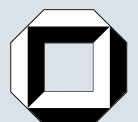
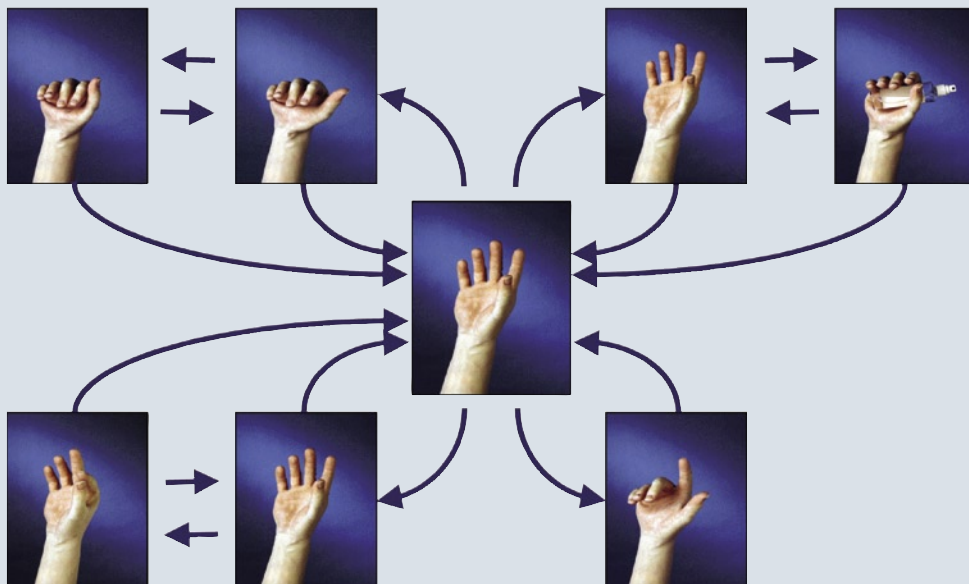


Markus Reischl

Ein Verfahren zum automatischen Entwurf von Mensch-Maschine- Schnittstellen am Beispiel myo- elektrischer Handprothesen



Markus Reischl

Ein Verfahren zum automatischen Entwurf von Mensch-Maschine-Schnittstellen am Beispiel myoelektrischer Handprothesen

Schriftenreihe des

Instituts für Angewandte Informatik / Automatisierungstechnik

an der Universität Karlsruhe (TH)

Band 13

Ein Verfahren zum automatischen Entwurf von Mensch-Maschine- Schnittstellen am Beispiel myo- elektrischer Handprothesen

von
Markus Reischl



universitätsverlag karlsruhe

Dissertation, Universität Karlsruhe (TH)
Fakultät für Maschinenbau, 2006

Impressum

Universitätsverlag Karlsruhe
c/o Universitätsbibliothek
Straße am Forum 2
D-76131 Karlsruhe
www.uvka.de



Dieses Werk ist unter folgender Creative Commons-Lizenz
lizenziert: <http://creativecommons.org/licenses/by-nc-nd/2.0/de/>

Universitätsverlag Karlsruhe 2006
Print on Demand

ISSN 1614-5267
ISBN 3-86644-014-6

Ein Verfahren zum automatischen Entwurf von Mensch-Maschine-Schnittstellen am Beispiel myoelektrischer Handprothesen

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
von der Fakultät für Maschinenbau der
Universität Karlsruhe

eingereichte
Dissertation
von

Dipl.-Ing. Markus Reischl
geboren am 22. Juni 1976 in Lahr/Schwarzwald

Hauptreferent:
Korreferent:

Prof. Dr.-Ing. habil. G. Bretthauer
Prof. Dr.-Ing. habil. R. Dillmann

Tag der Einreichung:
Tag der mündlichen Prüfung:

22. 12. 2005
03. 02. 2006

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit im Institut für Angewandte Informatik des Forschungszentrums Karlsruhe.

Für die Möglichkeit an einem sehr interessanten und umfangreichen Gebiet forschen zu können, möchte ich Professor Georg Bretthauer herzlich danken. Bereits während des Studiums gab er mir die Möglichkeit, Eindrücke auf dem Gebiet der Biosignalverarbeitung zu gewinnen, die ich im Rahmen meiner Dissertation vertiefen konnte. Stets war ich mir seiner Unterstützung sicher und fand jederzeit ein offenes Ohr, um Probleme und angestrebte Lösungen konstruktiv zu diskutieren.

Weiterhin danke ich Professor Rüdiger Dillmann für die Übernahme des Korreferats. Mit Interesse hat er den Fortschritt der Arbeit verfolgt und so zu deren Gelingen beigetragen.

Besonderer Dank gilt Herrn Ralf Mikut, der mir seit meinen ersten Tagen im Institut durch angeregte Diskussionen eine hervorragende fachliche Betreuung zukommen ließ. Ebenso möchte ich mich bei Herrn Lutz Gröll für zahlreiche Anregungen, wertvolle Hinweise und guten Kaffee bedanken.

Für die Mithilfe bei der Akquisition von Patientendaten möchte ich mich bei den Mitarbeitern der Orthopädischen Universitätsklinik Heidelberg und insbesondere bei Herrn Rüdiger Rupp bedanken. Sein großer Erfahrungsschatz im Umgang mit Patienten, dem Messen bioelektrischer Signale und der Veredelung metallischer Oberflächen waren mir eine große Hilfe.

Für die Unterstützung bei physiologischen Fragestellungen danke ich Herrn Christian Pylatiuk und Herrn Hagen Malberg, die stets bereit waren, ihr Wissen mit mir zu teilen, sowie Herrn Stefan Schulz für die Möglichkeiten zur praktischen Erprobung. Weiterhin gilt ein besonderer Dank Herrn Oliver Rebholz, der die Sicht des Patienten durch anregende Gespräche in diese Arbeit einfließen ließ.

Für wertvolle und unterhaltsame Diskussionen gilt mein Dank meinen Zimmerkollegen Herrn Sebastian Beck, Herrn Ole Burmeister und Herrn Arne Lehmann, sowie allen anderen Mitarbeitern des Instituts, die zum Gelingen dieser Arbeit beigetragen haben.

Weiterhin danke ich allen Studenten, die im Rahmen von Praktika, Studien- oder Diplomarbeiten einen Beitrag zu dieser Arbeit geleistet haben. Insbesondere möchte ich mich bei Herrn Daniel Kraut und Herrn Bernd Voelkel für die fruchtbare Zusammenarbeit bedanken.

Mein herzlicher Dank gilt schließlich meinen Eltern Christa und Werner sowie meiner Freundin Jasmin für Rückhalt, Verständnis und fleißiges Korrekturlesen.

Inhaltsverzeichnis

Inhaltsverzeichnis	v
1 Einleitung	1
1.1 Bedeutung der Arbeit	1
1.2 Darstellung des Entwicklungsstandes	2
1.2.1 Überblick Mensch-Maschine-Schnittstellen	2
1.2.2 Bewegungsplanung und Signalfluss	4
1.2.3 Signalübertragung	6
1.2.4 Mensch-Maschine-Schnittstellen zur Bewegungsdetektion	7
1.2.5 Bewertung der Signalquellen für den Einsatz in Mensch-Maschine-Schnittstellen	10
1.2.6 Signalverarbeitung und -auswertung	12
1.2.7 Stand der Technik: Myoelektrische Handprothesen	12
1.2.8 Automatischer Entwurf von Schnittstellen für biologische Systeme	16
1.2.9 Offene Probleme	17
1.3 Ziele und Aufgaben	18
2 Neue Entwurfsmethodik für biologische Mensch-Maschine-Schnittstellen	20
2.1 Neues Konzept	20
2.2 Universell anwendbarer Zustandsautomat	22
2.3 Ablauf zur Klassifikation von Schaltsignalen	24
2.4 Neues Bewertungsmaß zur Beurteilung von Mensch-Maschine-Schnittstellen	28
2.5 Entwurf eines Benchmarkdatensatzes	31
2.6 Merkmalsextraktion	33
2.6.1 Übersicht	33
2.6.2 Merkmale für Aktivitätssignale zum Schalten mit Zeitversatz	34
2.7 Merkmalsselektion	37
2.8 Merkmalsaggregation	40
2.9 Konstruktion der Entscheidungsregel	41
2.9.1 Übersicht	41
2.9.2 Fuzzy-Regelbasen	42
2.9.3 Künstliche Neuronale Netze (KNN)	45
2.9.4 k-Nearest-Neighbour-Klassifikatoren (k-NN)	47
2.9.5 Support-Vektor-Maschinen (SVM)	48
2.9.6 Approximative Maximum-Likelihood-Schätzung (AML)	51
2.10 Vergleich	53
2.10.1 Validierungstechniken	53
2.10.2 Zeitpunkt der Merkmalerzeugung	55
2.10.3 Umfang des Lerndatensatzes	56

2.10.4	Ergebnisse	57
3	Neue Verfahren für den Entwurf von Klassifikatoren	62
3.1	Übersicht	62
3.2	Einschrittverfahren: Optimierung der Prozessschritte	64
3.2.1	Modifizierte Merkmalsauswahl: Modifikation der Selektion (MM)	64
3.2.2	Modifizierte Diskriminanzanalyse: Modifikation der Projektion (MD)	67
3.2.3	Modifizierte Entscheidungsregel: Modifikation der Metrik (MAML)	69
3.3	Mehrschrittverfahren	70
3.3.1	Übersicht	70
3.3.2	Hierarchische Klassifikation nach Klassen (HKK)	72
3.3.3	Hierarchische Klassifikation nach Datentupeln (HKT)	77
3.3.4	Turnier- und Knock-Out-Klassifikation (TN, KO)	82
3.4	Robuster Entwurf	85
3.5	Diskussion	88
4	Implementierung	92
4.1	Übersicht	92
4.2	Werkzeuge	94
4.2.1	Datenauswertung und Verarbeitung (DAVE)	94
4.2.2	DAVE-Design	95
4.2.3	Mikrocontroller-Umgebung	97
4.3	Quelltextexport	98
4.3.1	Merkmalsextraktion und Merkmalsauswahl	98
4.3.2	Merkmalsaggregation und Entscheidungsregel	100
4.4	Bewertung	101
5	Anwendungen	102
5.1	Übersicht	102
5.2	Myoelektrische Handprothesen	102
5.2.1	Anforderungen	102
5.2.2	Umsetzbare Schaltsignale	104
5.2.3	Anwendungsspezifische Anpassung für Handprothesen	105
5.2.4	Spezifikation der Steuerungseinheit	109
5.2.5	Versuchsdesign und Patientenkollektiv	111
5.2.6	Schaltsignaldauer und Reproduzierbarkeit	113
5.2.7	Auswirkungen von Training	115
5.2.8	Umfang des Lerndatensatzes	115
5.2.9	Güte der Klassifikatoren	116
5.2.10	Robustheit	118
5.2.11	Resultat	123
5.3	Brain-Computer-Interface	124
5.3.1	Datensätze	124
5.3.2	Methoden	125
5.3.3	Ergebnisse und Diskussion	128
5.4	Bewertung	129
6	Zusammenfassung	131

Literaturverzeichnis	133
7 Anhang	146
7.1 Methodik	146
7.1.1 Vorgehensweise zum Entwurf des Benchmarkdatensatzes	146
7.1.2 Robuster Entwurf durch Erweitern des Lerndatensatzes	147
7.2 Klassifikationsergebnisse	152
7.2.1 Patientendatensätze und Entscheidungsregeln	152
7.2.2 Benchmarkdatensätze	154
7.3 Programmierte Funktionen	156
7.3.1 DAVE	156
7.3.2 Mikrocontroller	156
7.3.3 DAVE-Design	157
7.4 Quelltextgenerierung und Implementierung von Fuzzy-Regelbasen	160
7.5 Merkmalsrelevanzen	163
Bildverzeichnis	166
Tabellenverzeichnis	169

Symbolverzeichnis

A	Transformationsmatrix zur Aggregation
A	abstraktes Schaltsignal
a	Filterkonstante
$a(\mathbf{e}; \mathbf{g})$	Ausgangssignal eines künstlichen Neurons
a_{bench}	Filterkonstante zum Entwurf des Benchmarkbeispiels
$\mathbf{A}_{\text{hierch}}$	Transformationsmatrix beim HKK-Verfahren
$\mathbf{a}_{\text{hierch},l}$	Spaltenvektoren von $\mathbf{A}_{\text{hierch}}$
$\mathbf{A}_{\text{hierch,opt}}$	optimale Transformationsmatrix beim HKK-Verfahren
\mathbf{a}_i	Transformationsvektor i
$a_j^{(\alpha)}$	Ausgangssignal des Neurons j der Schicht α
$a_{j,i}$	Zuordnung einer Amplitude des Signals j zum Zeitpunkt i
$a_{j\max,i}$	maximale Amplitude des Signals j zum Zeitpunkt i
a_L	Filterkonstante für langsames Filter
\mathbf{A}_m	Transformationsmatrix für das HKT-Verfahren
\mathbf{A}_{opt}	optimale Transformationsmatrix zur Merkmalsaggregation
a_S	Filterkonstante für schnelles Filter
a_σ	Filterkonstante für die kontinuierliche Schätzung der Standardabweichung
A/D	analog/digital
AAR	adaptives autoregressives Modell
ANOVA	univariate Analyse der Varianzen
AML	approximierte Maximum-Likelihood-Regel
AR	Autoregression
AV	amplitudenvariiertes Schaltsignal
$\mathbf{B} _{\mathcal{I}}$	Zwischenklassenvariationsmatrix ausgewählter Merkmale
BCI	Brain-Computer-Interface
bestLDF	beste lineare Diskriminanzfunktion
BP	Bandleistung (bandpower)
C	frei wählbarer Parameter für SVM
\mathcal{C}	Indexmenge abgespaltener Klassen beim HKK-Verfahren
$C3, C4, Cz$	standardisierte Positionen auf der Kopfoberfläche zur Ableitung von EEG-Signalen
c_G	Konstante zur Berechnung von Q_G
c_K	Konstante zur Berechnung von Q_K
$c_{L,i}$	Konstanten zur Berechnung von Q_L
$c_{M,i}$	Konstanten zur Berechnung von Q_M
$c_{P,i}$	Konstanten zur Berechnung von Q_P
card	Kardinalität einer Menge
D	Divergenzmatrix
$d(\mathbf{x}_1, \mathbf{x}_2)$	Distanzmaß
D_i	zeitvarianter Benchmarktestdatensatz i

d_{ij}	Divergenz zwischen Klasse i und Klasse j
d_j	Entscheidungsmaß für Klasse j
d_{\max}	maximale Divergenz
DAVE	Werkzeug zur DatenAuswertung und VErarbeitung
e	Eingangsvektor eines Künstlichen Neuronalen Netzes
$E_{i,j}$	Ereignisse im Zustandsautomaten für die Handprothesensteuerung
ECoG	Elektrokortikogramm
EEG	Elektroencephalogramm
EMG	Elektromyogramm
ENG	Elektroneurogramm
$f(z_n)$	Aktivierungsfunktion eines künstlichen Neurons
f_a	Abtastfrequenz
f_j	Wahrscheinlichkeitsdichte der Klasse j
Fuzzy	Fuzzy-Regelbasis
g	Gewichtsvektor eines Künstlichen Neuronalen Netzes
\mathcal{G}_i	Entscheidungsbereich i einer Entscheidungsregel
g_j	Ungleichungssystem für Klasse j
HKA	Hauptkomponentenanalyse
HKK	hierarchische Klassifikation nach Klassen
HKT	hierarchische Klassifikation nach Datentupeln
i	Laufvariable
I	intuitives Schaltsignal
I	Einheitsmatrix
\mathcal{I}	Indexmenge ausgewählter Merkmale
$\mathcal{I}_{\text{cand}}$	Indexmenge von Kandidaten zur Merkmalsauswahl
i_j	Index eines ausgewählten Merkmals
$I_j[k]$	Indikator zur Extremwertsuche in Signal j
i_M	Parameter zur Beschreibung des Mikrocontrollers
\mathcal{I}_r	Indexmenge von r ausgewählten Merkmalen bei HKK
IIR	Infinite-Input-Response
j	Laufvariable
j_1, j_2	Sensorbezeichner
$\mathcal{J}_{j,m}$	nicht klassifizierte Datentupel der Klasse j in Hierarchiestufe m
$\mathcal{J}_{j,m}^*$	Indexmenge zum Anlernen verwendeter Datentupel der Klasse j in Hierarchiestufe m
\mathcal{J}_m	nicht klassifizierte Datentupel in Hierarchiestufe m
\mathcal{J}_m^*	Indexmenge zum Anlernen verwendeter Datentupel in Hierarchiestufe m
K	kombiniertes Schaltsignal
k	Anzahl nächster Nachbarn bei k-Nearest-Neighbour-Klassifikatoren
k	Abtastzeitpunkt
K	Klassenanzahl
$K(\mathbf{x}_1, \mathbf{x}_2)$	Kernfunktion
$k_{e,ji}$	Zeitpunkt des Extremwerts i von Signal j
$k_{\text{end},ji}$	vorläufiger Bezeichner für den Zeitpunkt eines Bereichsendes
$k_{h,ji}$	Zeitpunkt des Erkennens des Extremwerts i von Signal j
k_{\min}	erster Zeitpunkt mit einer Merkmalsrelevanz größer 0.4 bei BCI
k_{\max}	maximale Wartezeit, nach der Merkmale automatisch generiert werden
$k_{M\max}$	Zeitpunkt mit maximalem MANOVA-Wert bei BCI
K_{org}	Anzahl aufgenommener Schaltsignalklassen

k-NN	k-Nearest-Neighbour-Klassifikator
KL	Kullback-Leibler
KNN	Künstliches Neuronales Netz
KO	KO-Klassifikation
KO _{agg}	KO-Klassifikation mit aggregierten Merkmalen
KO _{ind}	KO-Klassifikation mit individueller Merkmalsauswahl im Zweiklassenproblem
KO _{sel}	KO-Klassifikation mit selektierten Merkmalen
l	Laufvariable
LDA	lineare Diskriminanzanalyse
LDS	Lerndatensatz
m	Hierarchiestufe
M	Zielfunktion zur Merkmalsauswahl
\mathcal{M}	Merkmalsraum
m_a	maximal mögliche Amplitude beim Benchmarkentwurf
m_{\max}	maximale Anzahl an Hierarchiestufen bei HKT
M^{ξ_G}	wertediskretes Kriterium zur Berechnung der Merkmalsauswahl
$M^{\xi_G, \text{mod}}$	modifiziertes Kriterium zur Berechnung der Merkmalsauswahl
$M^{\xi_G, \text{stetig}}$	kontinuierliches Kriterium zur Berechnung der Merkmalsauswahl
MAML	modifizierter AML-Klassifikator
MAN/DI	AML-Klassifikator mit MANOVA und LDA, $s_m = 8, s_d = 4$
MAN/MD	AML-Klassifikator mit MANOVA und MD, $s_m = 6, s_d = 3$
MANOVA	multivariate Analyse der Varianzen
MAXMIN	Maximum-Minimum-Inferenz
MD	modifizierte Diskriminanz
ML	Maximum-Likelihood
MLP	Multilayer-Perzeptron-Netz
MM	Modifikation der Merkmalsselektion
MM/DI	AML-Klassifikator mit MM und LDA, $s_m = 6, s_d = 3$
MM/MD	AML-Klassifikator mit MM und MD, $s_m = 6, s_d = 3$
MMI	Mensch-Maschine-Schnittstelle (Man-Machine-Interface)
n	Anzahl aller Datentupel
\mathbb{N}	Raum der natürlichen Zahlen
n_b	Anzahl der Zeitpunkte zur Synthetisierung eines Benchmarksignals
n_{Boot}	Anzahl der für den Bootstrap gezogenen Datentupel
n_c	Mindestanzahl zu verwendender Datentupel beim Anlernen von HKT
n_{cross}	Anzahl an Lerndatentupeln bei Crossvalidierung
n_e	Anzahl zu suchender Extrema
n_j	Anzahl an Datentupeln pro Klasse j
n_K	Anzahl an Multiplikationen zur Berechnung eines Kerns
n_n	Anzahl an Zwischenschichtneuronen bei Künstlichen Neuronalen Netzen
n_r	Anzahl an Multiplikationen zur Durchführung einer Transformation $\Phi(\mathbf{x})$ bei SVM
n_s	Anzahl an Zeitreihen im Schaltsignal
n_s^*	Anzahl aller Zeitreihen (einschließlich kombinierter Zeitreihen)
n_{sup}	Anzahl der Support-Vektoren
n/a	nicht verfügbar (not available)
NM	Fuzzy-Zugehörigkeitsfunktion "negativ mittel"
NK	Fuzzy-Zugehörigkeitsfunktion "negativ klein"
NSK	Fuzzy-Zugehörigkeitsfunktion "negativ sehr klein"

\mathbf{O}_j	symmetrische Matrix zur Optimierung der Kovarianzmatrix von Klasse j
p	Nummer des gestörten Schaltsignals
\mathbf{P}	In Schemazeichnungen: Pumpe
\hat{P}	geschätzte Wahrscheinlichkeit
\hat{P}_A	geschätzte Wahrscheinlichkeit bei Optimierung der Transformationsmatrix \mathbf{A}
p_j	Anzahl notwendiger Parameter zur Klassifikation der Klasse j beim AML-Klassifikator
$\hat{P}_{j,\text{all}}$	Gesamtzugehörigkeit zur Klasse j
$p_{j,l}$	Ergebnis des Klassifikators l für Klasse j bei KO- und TN-Verfahren
$\hat{P}_{j,m}$	geschätzte Klassenzugehörigkeit zu Klasse j in Hierarchiestufe m bei HKT
p_{max}	Anzahl gestörter Schaltsignale
PG	Fuzzy-Zugehörigkeitsfunktion "positiv groß"
PK	Fuzzy-Zugehörigkeitsfunktion "positiv klein"
PM	Fuzzy-Zugehörigkeitsfunktion "positiv mittel"
PSG	Fuzzy-Zugehörigkeitsfunktion "positiv sehr groß"
Q	Kriterium zur Berechnung des MMI-Index
Q_a	Kriterium zur Maximierung der Abstände von einer Trennebene bei SVM
q_c	Mindestkonfidenz zum Extrahieren eines Datentupels bei HKT
Q_C	Kriterium "Mikrocontroller-Kategorie"
Q_G	Kriterium "Güte der Klassenzuweisung"
Q_j	Kriterium zur Bewertung der gefundenen Parameter für Klasse j bei HKK
Q_K	Kriterium "Anzahl umsetzbarer Zustände"
Q_L	Kriterium "Engineering-Kosten"
Q_{Lag}	Q_a mit Lagrange'schen Nebenbedingungen
$Q_{L,1}$	Kriterium "Entwurf in einer Sitzung"
$Q_{L,2}$	Kriterium "numerischer Entwurf"
Q_M	Kriterium "Anzahl benötigter Multiplikationen"
Q_{max}	obere Schranke zum Abspalten von Klassen
Q_P	Kriterium "Anzahl benötigter Speicherplätze"
Q_T	Kriterium "Trainingsaufwand für den Patienten"
Q_{Trans}	Kriterium zur Berechnung der bestLDF-Transformation
r	Kardinalität der Menge ausgewählter Merkmale
$\mathbb{R}, \mathbb{R}^s, \mathbb{R}^{n \times s}$	Raum der reellen Zahlen
RBF	Radial-Basis-Funktion-Netz
s	Anzahl der Merkmale
\mathbf{S}	Distanzmatrix
\mathcal{S}	Indexmenge der Support-Vektoren
S_1	Abbildung zur Merkmalsextraktion
S_2	Abbildung zur Merkmalsselektion
S_2^*	Abbildung zum Entwurf der Merkmalsauswahl
$S_{2,m}$	Abbildung zur Merkmalsselektion in Hierarchiestufe m
S_3	Abbildung zur Merkmalsaggregation
S_3^*	Abbildung zum Entwurf der Merkmalsaggregation
$S_{3,m}$	Abbildung zur Merkmalsaggregation in Hierarchiestufe m
S_4	Abbildung zur Entscheidungsfindung
S_4^*	Abbildung zum Entwurf der Entscheidungsregel
$S_{4,m}$	Abbildung zur Entscheidungsfindung in Hierarchiestufe m
s_d	Anzahl aggregierter Merkmale
s_m	Anzahl ausgewählter Merkmale

s_m^{\min}	Mindestanzahl zu verwendender Merkmale bei HKT
s_u	Summe der Anzahl der Zeitpunkte aller Zeitreihen eines Schaltsignals
$s_{u,j}$	Anzahl an Zeitpunkten der Zeitreihe j eines Schaltsignals
s_Φ	Anzahl an Multiplikationen zur Berechnung von $\mathbf{w}^T \mathbf{x}$
SOM	Kohonenkarte (Self Organizing Map)
sp	Spur einer Matrix
SUMPROD	Summe-Produkt-Inferenz
SVM	Support-Vektor-Maschine
T	Gesamtdauer eines Benchmarksignals
\bar{T}	durchschnittliche Dauer zur Ausführung eines Schaltsignals
$\mathbf{T} _{\mathcal{I}}$	Gesamtvariationsmatrix ausgewählter Merkmale
T_a	Abtastzeit
$T_{A,B}$	Transition zum Übergang von Zustand A zu B
T_D	Dauer der Pumpenaktivierung
T_e	Dauer des Entwurfs der Entscheidungsregel
$t_{e,ji}$	Zeitpunkt des Extremwerts i von Signal j
t_h	Zeitpunkt beim Benchmarkentwurf
$t_{h,ji}$	Zeitpunkt des Erkennens des Extremwerts i von Signal j
$t_{j,i}$	Zeitpunkt des Signals j zum Zeitpunkt i beim Benchmarkentwurf
T_L	Konstante zur Trendberechnung
t_{\min}	erster Zeitpunkt mit einer Merkmalsrelevanz größer 0.4
T_o	maximale Dauer zur Ausführung eines Schaltsignals
t_p	Hilfsvariable
T_{s_m}	Dauer der Merkmalsauswahl
T_{s_d}	Dauer der Merkmalsaggregation
T_S	Konstante zur Trendberechnung
T_u	minimale Dauer zur Ausführung eines Schaltsignals
T&E	Test- und Evaluierung
TDS	Testdatensatz
TN	Turnier-Klassifikation
TN _{sel}	Turnier-Klassifikation mit selektierten Merkmalen
TN _{ind}	Turnier-Klassifikation mit individueller Merkmalsauswahl im Zweiklassenproblem
TN _{agg}	Turnier-Klassifikation mit aggregierten Merkmalen
u	Rohdaten eines Schaltsignals
\mathbf{u}	Vektor aller Rohdaten
$u_{D,j}[k]$	Trend des Signals j
u_{disk}	Diskretisiertes Benchmarksignal
$u_{\text{EMG,max}}$	maximale EMG-Sensorspannung
$u_{f,j}[k]$	gefiltertes Signal von Signal j
$u_{f,v}[k]$	Zeitreihe aus Verhältnissen von Signalen
$u_{f,w}[k]$	Zeitreihe aus Differenzen von Signalen
$\bar{u}_{i,p}[k]$	gestörtes Schaltsignal
$u_j[k]$	Signal von Sensor j
$u_{L,j}[k]$	langsam gefiltertes Signal $u_j[k]$
$u_{\text{Max},h}$	Hysterese bei der Maximumsuche
$u_{\text{Max},j}$	vorläufiges Maximum in Signal j
u_{\min}	Mindestaktivierung
$u_{\text{Min},h}$	Hysterese bei der Minimumsuche

$u_{\text{Min},j}$	vorläufiges Minimum in Signal j
u_{Pumpe}	Pumpenspannung
$u_{\text{Pumpe,max}}$	maximale Pumpenspannung
$u_{s,j}[k]$	Summensignal des Signals j
$u_{S,j}[k]$	schnell gefiltertes Signal $u_j[k]$
u_{Schwell}	Schwellwert zur Erkennung von Bewegungssignalen
$u_{sp,j}[k]$	Anzahl der Filterschnittpunkte
$u_{\text{Steuer}}[k]$	Steuersignal
$u_{\text{tmp}}[k]$	diskretisierte Zeitreihe
$u_{\sigma,j}[k]$	Standardabweichung des Signals j
v	Parameter für die Crossvalidierung
v_a	Index der Zeitreihen aus Verhältnissen von Signalen
$v_{ja,i}$	Parameter für die Ausprägung von Zufallsgrößen bei der Amplitudenmodellierung
$v_{jk,i}$	Parameter für die Ausprägung von Zufallsgrößen bei der Zeitpunktmodellierung
V1, ..., V6	Ventile 1-6
w	zu wichtender Wert
\mathbf{w}	Richtungsvektor für SVM
$\mathbf{W} _{\mathcal{I}}$	Innerklassenvariationsmatrix ausgewählter Merkmale
w_0	Parameter für SVM
w_a	Index der Zeitreihen aus Differenzen von Signalen
\mathbf{x}	Vektor aller Merkmale
\mathbf{X}	Matrix aller Merkmale
$\bar{\mathbf{x}} _{\mathcal{I}}$	Gesamtmittelwert ausgewählter Merkmale
$\mathbf{X} _{\mathcal{J}_m}$	Datenmatrix nicht klassifizierter Datentupel in Hierarchiestufe m
$\mathbf{x}_{D,j}$	Trendextremwertvektor des Signals j
$x_{D,j,i}$	Trendextremwert im Bereich i des Signals j
$\mathbf{x}_{e,j}$	Extremwertvektor des Signals j
$x_{e,j,i}$	Extremwert des Bereichs i des Signals j
$\mathbf{x}_{\mathcal{I}}$	Vektor der ausgewählten Merkmale
$\bar{\mathbf{x}}_j$	Klassenmittelwert der Klasse j
$\bar{\mathbf{x}}_j _{\mathcal{I}}$	Klassenmittelwert der Klasse j ausgewählter Merkmale
x_l	Beispiel für ein Merkmal
$\mathbf{x}_{r,j}$	relativer Extremwertvektor des Signals j
$x_{r,j,i}$	relativer Extremwert des Bereichs i des Signals j
$\mathbf{x}_{s,j}$	Summensignalvektor des Signals j
$x_{s,j,i}$	Summensignal zum Bereichsende des Bereichs i des Signals j
$\mathbf{x}_{S,j}$	Merkmalsvektor für Signal j (S: Sensorsignal)
$\mathbf{x}_{sp,j}$	Schnittpunktvektor des Signals j
$x_{sp,j,i}$	Anzahl an Signalschnittpunkten im Bereich i des Signals j
$\mathbf{x}_{t,j}$	zeitlicher Ausdehnungsvektor des Signals j
$x_{t,j,i}$	zeitliche Ausdehnung des Bereichs i des Signals j
$x_{\text{trans},l,i}$	intervalltransformiertes Merkmal
$\mathbf{x}_{u,j}$	relativer Summensignalvektor des Signals j
$x_{u,j,i}$	relatives Summensignal des Bereichs i des Signals j
$\mathbf{x}_{z,j}$	relativer zeitlicher Ausdehnungsvektor des Signals j
$x_{z,j,i}$	relative zeitliche Ausdehnung des Bereichs i des Signals j
$\mathbf{x}_{\sigma,j}$	Standardabweichungsvektor des Signals j
$x_{\sigma,j,i}$	gefilterte Standardabweichung zum Bereichsende des Bereichs i des Signals j

\mathbf{x}_ϕ	transformierte Merkmalsvariable bei SVM
$\mathbf{x}_{\phi,i}$	transformiertes Merkmal bei SVM
\mathbf{y}	Vektor der Klassenzugehörigkeiten
\hat{y}	geschätzte Klassenzugehörigkeit (Bewegungsabsicht)
$\mathbf{y} \mathcal{J}_m$	Klassenzugehörigkeitsmatrix nicht klassifizierter Datentupel in Hierarchiestufe m
y_i	Klassenzuordnung des Datentupels i
\hat{y}^{ML}	Maximum-Likelihood-Klassenschätzung
\mathbf{z}	Vektor der aggregierten Merkmale
\mathbf{Z}	Matrix der aggregierten Merkmale
$\bar{\mathbf{z}}_{j,l}$	Mittelwert der aggregierten Merkmale des Klassifikators l für Klasse j
z_n	Verknüpfung von Eingangssignalen bei Künstlichen Neuronalen Netzen
ZE	Fuzzy-Zugehörigkeitsfunktion "null"
ZV	zeitvariiertes Schaltsignal
α	Schichtnummer eines Künstlichen Neuronalen Netzes
α_{\max}	Anzahl Schichten eines Künstlichen Neuronalen Netzes
γ_{ji}	gleichverteilte Zufallsgröße beim Benchmarkentwurf
δ_{ji}	normalverteilte Zufallsgröße beim Benchmarkentwurf
$\epsilon_{a,p}$	gleichverteilte Zufallsgröße zur Modellierung von Amplituden bei zeitvarianten Signalen
$\epsilon_{\max,a}$	prozentuale Amplitudendehnung
$\epsilon_{\max,t}$	prozentuale Zeitdehnung
$\epsilon_{t,p}$	gleichverteilte Zufallsgröße zur Modellierung von Zeitdauern bei zeitvarianten Signalen
θ_j	Parameter des Klassifikators für Klasse j
$\hat{\theta}_j$	geschätzte Parameter des Klassifikators für Klasse j
κ	Konfusionsmatrix
κ_{ij}	Konfusion von Klasse i zu Klasse j
Λ	Bewertungsmaß MMI-Index
λ_i	Lagrange-Multiplikator (Eigenwert i)
$\mu(y = j)$	Zugehörigkeit zur Klasse j
$\boldsymbol{\mu}_j$	wahrer Mittelwertsvektor der Klasse j
$\boldsymbol{\nu}_i$	Eigenvektor i
ξ_G	Klassifikationsgüte
$\xi_{G,\text{all}}$	Klassifikationsgüte über allen Lerndaten
$\xi_{G,T}$	Klassifikationsgüte über Testdaten
ξ_L	Anlerndauer des Klassifikators
ξ_M	Anzahl an Multiplikationen
ξ_P	Anzahl an Speicherplätzen
$\rho[k]$	normalverteiltes Zufallsgrößensignal
$\boldsymbol{\Sigma}_j$	wahre Klassenkovarianzmatrix für Klasse j
$\hat{\boldsymbol{\Sigma}}_j$	geschätzte Klassenkovarianzmatrix für Klasse j
$\hat{\boldsymbol{\Sigma}}_j \mathcal{I}$	geschätzte Klassenkovarianzmatrix ausgewählter Merkmale für Klasse j
$\hat{\boldsymbol{\Sigma}}_{j,l}$	geschätzte Kovarianzmatrix des Klassifikators l für Klasse j
$\Phi(\mathbf{x})$	frei wählbare Merkmalstransformation bei SVM
χ	Proportionalitätskonstante
$\psi(w)$	Wichtungsfunktion
$\Psi_{A,B}$	Schaltsignal zur Definition von $T_{A,B}$
Ψ_N	Schaltsignal für neutralen Zustand
$\bar{\Psi}_{i,p}$	gestörtes Schaltsignal p entstanden aus Schaltsignal i

1 Einleitung

1.1 Bedeutung der Arbeit

Myoelektrische (elektromyographische (EMG-)) Handprothesen sind künstliche Gliedmaßen für Menschen mit Unter- oder Oberarmamputationen, die ihren Trägern die Funktionalität ihrer verlorenen Hand teilweise ersetzen [125, 129, 131]. Der Prothesenträger steuert seine Prothese durch Kontraktionen der Muskeln seines verbliebenen Armstumpfs, deren Aktionspotenziale oberflächlich als Summensignale erfasst werden. Über einfache Schwellwertdetektionen oder proportionale Steuerungen werden so meist ein bis zwei Freiheitsgrade bedient (z. B. Hand öffnen/schließen, Handgelenk drehen). Ähnlich funktionieren z. B. Brain-Computer-Interfaces (BCI) zur Verknüpfung von aktivierten Hirnarealen mit Cursorbewegungen auf einem Computermonitor [19, 116, 177, 185, 209]. Sowohl EMG-Prothesen als auch BCIs reagieren dabei auf bewusste Aktionen des Anwenders.

Um die bewussten Aktionen des Trägers auszuwerten und zu interpretieren, kommen Mensch-Maschine-Schnittstellen (Man-Machine-Interface: MMI) zum Einsatz. Das MMI erfasst hierbei menschliche Aktionen in Form von Sensorsignalen, wertet diese aus und leitet das Ergebnis an die angeschlossene Maschine weiter. Bei EMG-Prothesen setzt es dabei die Kontraktion der Armstumpfmuskulatur des Anwenders gemäß einem vorgegebenen Schema in Fingerbewegungen einer Prothese um. Um dem MMI eine einheitliche Auswertung zu ermöglichen, muss der Träger einer EMG-Prothese in der Lage sein, seine Stumpfmuskulatur reproduzierbar (bewusst oder unbewusst) zu kontrahieren. Der Träger eines BCIs muss in der Lage sein, Hirnregionen reproduzierbar zu aktivieren.

Die fortschreitende Miniaturisierung der vergangenen Jahrzehnte brachte neue Greifsysteme mit umfangreichen Bewegungsmöglichkeiten hervor [32, 42, 92, 185]. Um dem Anwender diese Bewegungsmöglichkeiten zugänglich zu machen, ist jedoch eine Erweiterung bestehender MMIs notwendig. Ein manueller Entwurf des erweiterten MMIs führt jedoch zu einer erheblichen Steigerung der Komplexität von Parametrierung, Sensorplatzierung, etc., zudem steigt der Implementierungs- und Rechenaufwand für die Umsetzung des MMIs.

Deshalb stellt diese Arbeit ein neues Verfahren zum automatischen Entwurf von Mensch-Maschine-Schnittstellen basierend auf biometrischen Beispielsignalen vor, das insbesondere einer hohen Zuverlässigkeit und recheffizienten Echtzeit-Implementierung Rechnung trägt. Das Verfahren parametriert MMIs anhand von vom Patienten gemessenen Signalen. Mit Hilfe einer grafischen Benutzeroberfläche ist es möglich, patientenindividuell und automatisch die Steuerung zu entwerfen. Die Implementierung in eine Echtzeitumgebung wird durch Quelltextgenerierung und -export bewerkstelligt. Das Verfahren wird beispielhaft zur Umsetzung einer Steuerung für myoelektrische Handprothesen angewandt und anhand von Patientendaten evaluiert. Abschließend demonstriert der Entwurf eines Brain-Computer-Interfaces die Übertragbarkeit auf andere MMIs.

1.2 Darstellung des Entwicklungsstandes

1.2.1 Überblick Mensch-Maschine-Schnittstellen

Definition 1.1 Als Mensch-Maschine-Schnittstelle werden alle wahrnehmbaren Komponenten eines technischen Systems (kurz: Maschine) bezeichnet, die der Kommunikation mit ihren Benutzern dienen. Die notwendigen Funktionen sind dabei nach ergonomischen Gesichtspunkten auf Mensch bzw. Maschine zu verteilen [34].

Einfache Mensch-Maschine-Schnittstellen (MMI) verarbeiten z. B. Eingaben über Bedienelemente (z. B. Schalter, Hebel), kompliziertere werten eingehende Signale aus und interpretieren die Handlungsabsicht des Bedieners. Das Schema zur Kommunikation zwischen Mensch (Anwender) und Maschine ist in Bild 1.1 dargestellt. Die Mensch-Maschine-Schnittstelle erfasst menschliche Aktionen in Form von Sensorsignalen und macht sie der Maschine in Form von Steuersignalen verfügbar. Die Transformation der Sensorsignale in Steuersignale erfolgt durch die Signalanalyse, die eingehende Signale bewertet und interpretiert. Informationen über die Maschine (Signale, Zustände oder Status) können dem Menschen wiederum über eine Maschine-Mensch-Schnittstelle mitgeteilt werden. Bei Neuroprothesen (Freehand-System) generiert der Anwender z. B. mittels eines in der Schulter implantierten Joysticks Bewegungssignale [171]. Die Signalanalyse generiert daraus ein Steuersignal, das sie einer Maschine zur Elektrostimulation der Handmuskulatur zuführt.

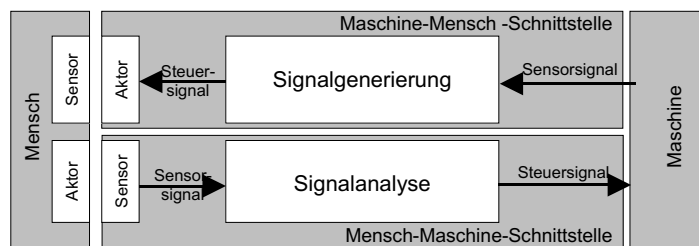


Bild 1.1: Mensch-Maschine-Kommunikation

Bei einfachen Bedienelementen ist die Signalanalyse oft trivial und umfasst beispielsweise nur das Auswerten einer Schalterstellung. Im Bereich der Medizintechnik dienen jedoch häufiger biometrische Signale als Eingang für das MMI, die z. B. als elektrische Größen Spannung oder Strom, seltener als akustische oder mechanische Größen vorliegen [10, 41, 57, 117]. Ein biometrisches Signal bezeichnet hier ein Signal, das einem biologischen System entstammt und das Informationen über den Zustand desselben enthält. Das Verarbeiten der oft komplexen, biometrischen Signale spezifiziert die Signalanalyse zur Biosignalanalyse.

MMIs lassen sich dann danach unterscheiden, ob die eingehenden biometrischen Sensorsignale bewussten oder unbewussten Aktionen des Menschen entspringen. So sind MMIs basierend auf unbewussten Größen wie Herzfrequenz, Blutdruck oder Magensaftkonzentration denkbar, auf die der Mensch jedoch keinen unmittelbaren Einfluss hat. Das Ansteuern technischer Systeme und Interagieren mit der Umgebung erfordert hingegen bewusste Signale des Menschen.

Bild 1.2 stellt den Signalfluss und die Schnittstellen für eine solche Interaktion mit einem technischen System dar und zeigt, wie sich ein MMI in dieses System integriert. Als Eingang seitens des Anwenders

erhält das System eine Handlungsabsicht in Form eines biometrischen Signals. Gebräuchlich sind elektrische Sensorsignale von Hirn-, Nerven- und Muskelsignalen. Eine Handlungsabsicht bezeichnet den Wunsch, eine Aktion auszuführen und wird mitgeteilt durch die bewusste Generierung des entsprechenden biometrischen Signals. Jede Handlungsabsicht zeichnet sich durch einen charakteristischen Verlauf des biometrischen Signals aus. Dieser Verlauf kann im Falle von Prothesen der physiologisch entsprechende sein, kann aber auch angelernt sein. Biometrische Sensoren transformieren die biometrischen Signale in elektrische Signale und führen ggf. eine Vorverarbeitung durch (z. B. Gleichrichtung, Filterung). Das so entstehende Signal wird einer Auswerteeinheit zugeführt, die das Signal interpretiert, eine Handlungsabsicht erkennt und Sollvorgaben für die geplante Aktion generiert. Eine Steuerungs- bzw. Regelungseinheit generiert nun Stell- bzw. Führungsgrößen für die angeschlossene Aktorik. Daraus resultieren physikalische Größen (z. B. Kräfte oder Momente, Positionen auf Bildschirm, etc.), die zur Interaktion mit der Umgebung dienen, oder wie z. B. im Falle von Neuroprothesen mit dem Anwender selbst durch Elektrostimulation seiner Extremitäten. Ein Feedback über den Zustand des MMIs kann der Anwender visuell oder akustisch erhalten, entweder durch das Beurteilen der Bewegung der Aktoren oder durch die Auswirkung des Systems auf seine Umgebung.

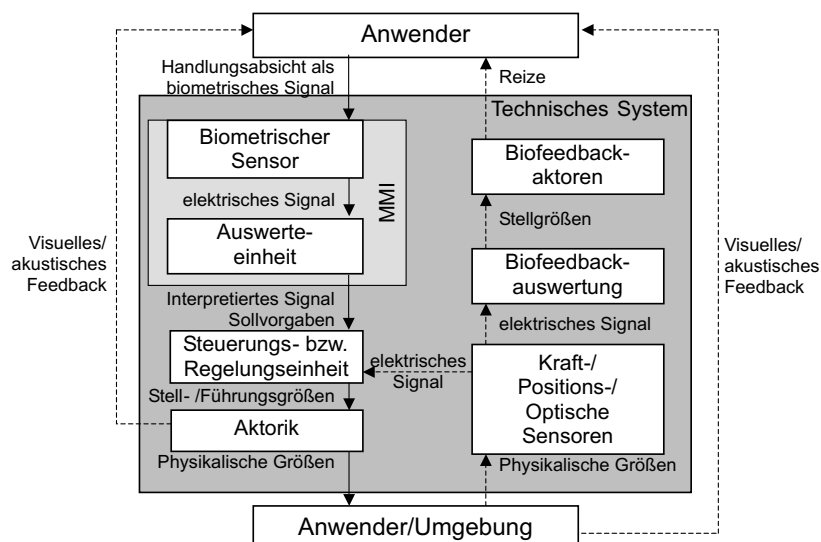


Bild 1.2: Steuerungs- bzw. Regelungskonzept für technische Systeme

Sensorsysteme erfassen physikalische Größen, die aus der Umgebung auf das System einwirken. Sie führen das entsprechende elektrische Signal an die Steuerungs- bzw. Regelungseinheit zurück oder optional einer Biofeedbackauswertung zu. Letztere erstellt wiederum Stellgrößen für Biofeedbackaktoren, die den Zustand des Systems durch Reize an den Anwender übermitteln.

Als Signalquelle (Aktor) zum Generieren der Bewegungsabsicht und zum Steuern technischer Systeme sind alle reproduzierbar generierbaren biometrischen Signale verwendbar, insbesondere sollen in dieser Arbeit die in die Bewegungsplanung und -ausführung beliebig gearteter Bewegungen eingebundenen Systeme untersucht werden.

1.2.2 Bewegungsplanung und Signalfluss

Zum Verständnis der Bewegungsplanung bedarf es der Unterteilung des Begriffs Bewegung in die Kategorien reflexgesteuerte Bewegung, programmgesteuerte Bewegung sowie Willkürbewegung [180, 206]. Reflexgesteuerte Bewegungen bezeichnen unwillkürliche, zweckgerichtete Reaktionen des zentralen Nervensystems (ZNS, z. B. Kniesehnenreflex). Die Steuerung der Bewegung wird vom Rückenmark bzw. Stammhirn übernommen, eine Verarbeitung durch das Großhirn ist nicht zwingend erforderlich. Im Gegensatz zu Reflexen werden laut [180] programmgesteuerte Bewegungen ohne Zutun äußerer Reize vom ZNS gesteuert, eine Verarbeitung durch das Großhirn ist ebenfalls nicht nötig. Als Beispiele lassen sich die Atmung und das Laufen anführen. Die Willkürbewegung bezeichnet hingegen einen sowohl in Planung als auch Ausführung bewussten Vorgang. Zu dieser Gruppe zählt auch die willentliche Bewegung der Hand.

Die physiologische Umsetzung einer Willkürbewegung besteht aus einem komplexen Zusammenspiel verschiedener Hirnareale. Besondere Bedeutung kommt dem Kortex (Hirnrinde) und insbesondere den motorischen Kortizes (supplementär-motorisches Areal, prämotorischer Kortex, primär motorischer Kortex) zu.

Chronologisch betrachtet besteht eine Willkürbewegung gemäß Bild 1.3 aus der Entschlussphase (innerer Handlungsantrieb - impulsiver Gedanke oder Reaktion), dem Entwurf der Bewegungsstrategie (Strategieentscheidung), dem Entwurf des Bewegungsprogramms und der Bewegungsdurchführung (eigentliche Bewegung) [84, 179].

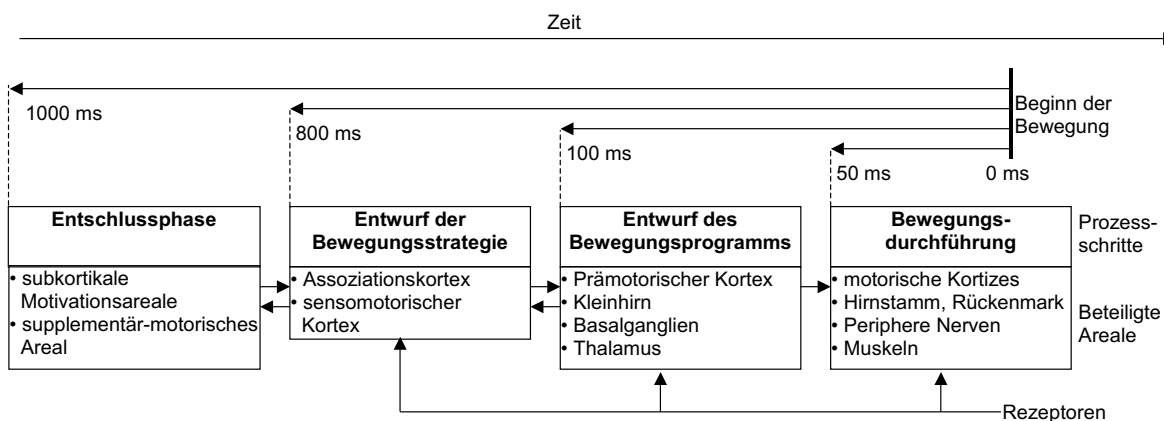


Bild 1.3: Ablauf der Zielmotorik [204]

In der Entschlussphase entsteht in den subkortikalen Motivationsarealen, den Assoziationskortizes und dem supplementär-motorischen Areal der Handlungsantrieb: Etwa 1s - je nach Schwierigkeitsgrad der Bewegung zwischen 0.3s und 3s - vor Beginn einer Bewegung zeigt sich das kortikale Bereitschaftspotenzial. Dieses spiegelt den Wunsch wieder, eine Bewegung durchzuführen. Auch wenn die Bewegung nicht ausgeführt wird, sind diese Regionen aktiv [179, 181].

Der Entschlussphase folgt der Entwurf der Bewegungsstrategie zur Durchführung der Bewegung. Dieser findet im Assoziations- und im sensomotorischen Kortex statt. Die Strategie bezeichnet eine mögliche Umsetzung der Bewegung. Zur Entwicklung der Bewegungsstrategie erhalten die Eingänge des Kortex Signale aus den Sinnesorganen (Rezeptoren). Der Thalamus (Teil des Zwischenhirns, Umschaltstation für Afferenzen) dient dabei als Verbindungsstation. Die Assoziationskortizes erhalten Signale

von den motorischen Kortizes. Die sensorischen Informationen werden dort in Zusammenarbeit mit den limbischen Strukturen ausgewertet und die motorischen Aktionen dadurch strategisch vorbereitet (geplant) [84].

Das Kleinhirn und die Basalganglien nutzen den prämotorischen Kortex zum Entwurf des Bewegungsprogramms aus der Bewegungsstrategie. Dieses stellt eine neuronale Repräsentation der geplanten Bewegung dar [179]. Beteiligte Systeme werden spezifiziert sowie Stärke und zeitliche Abfolge festgelegt.

Das vollendete Programm wird schließlich an den primär motorischen Kortex weitergeleitet. Dieser ist für die Kraftabstufung und die Bewegungsrichtung bei der Durchführung der Bewegung zuständig. Im Zuge der Selektion werden dort spinale Neuronen aktiviert, die der Umsetzung des Bewegungsprogramms dienen. Die Signalübermittlung vom Kortex weg wird als efferent bezeichnet, die Signalübermittlung zum Kortex hin als afferent.

Zur Bewegungsdurchführung übertragen die kortikospinalen Bahnen (Axone der Pyramidenzellen, Bild 1.4), die im motorischen Kortex entspringen, die Signale über den Hirnstamm und die Interneurone im Rückenmark zu den Motoneuronen. Zur Regulierung der Bewegungsabläufe greifen die entsprechenden Kortizes in Abhängigkeit von Rückmeldemechanismen bereits vor Beginn und während der Bewegung korrigierend in Bewegungsprogramm und -durchführung ein (Reafferenz).

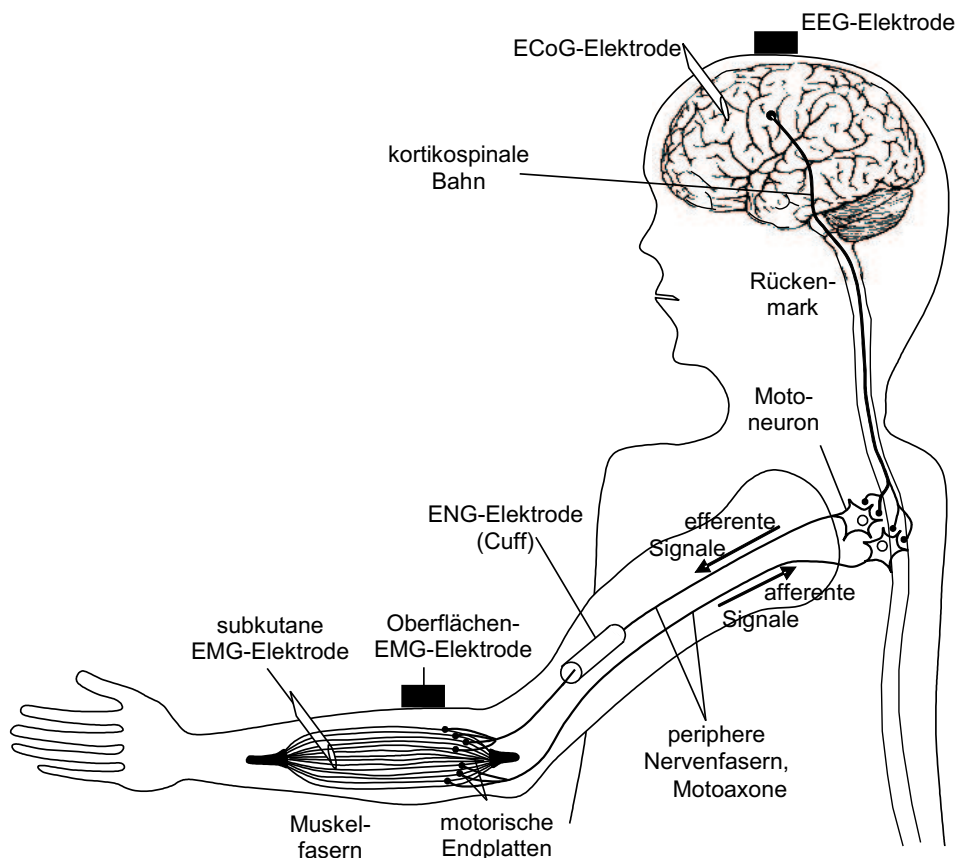


Bild 1.4: Neuromuskuläres System, zentrales und peripheres Nervensystem sowie Möglichkeiten zur Erfassung biometrischer Signale

Von den Motoneuronen gelangt das Signal über die α -Motoaxone (Nervenbahnen, die am Motoneuron

entspringen und an der Endplatte von Muskelfasern enden) zu den motorischen Endplatten der Muskeln. Die elektrischen Nervenreizimpulse verursachen ein Aktionspotenzial (AP) an der motorischen Endplatte des Muskels.

Muskelkontraktionen entstehen durch die motorischen Einheiten, die aus den α -Motoneuronen und den von ihnen innervierten Muskelfasern bestehen. Auf einen einzelnen Reiz hin entsteht eine Einzelzuckung der Muskelfasern. Eine motorische Einheit entlädt sich bei zunehmender Kraftentfaltung mit steigender Frequenz. Durch die Überlagerung der Einzelzuckungen kommt es zu einer Kontraktion des Muskels [84].

Das Kleinhirn erhält ständig Informationen über den Zustand der angestrebten Bewegung und kann so Korrekturen des Bewegungsablaufs vornehmen. Die Informationen erhält es über Rezeptoren in den Muskeln und Sehnen. Diese Rezeptoren sind über das periphere Nervensystem mit den afferenten Motoneuronen verbunden. Von dort aus führt die afferente Pyramidenbahn zum Kleinhirn.

Existieren im Nervensystem Läsionen (z. B. durch Querschnittlähmung, Amputation, Locked-In Syndrom), so ist eine Ausführung der entsprechenden motorischen Funktionen nicht mehr möglich. In diesem Fall besteht die Möglichkeit, durch ein MMI entweder die Verbindung zu überbrücken (z. B. Freehand-System, Blasenschrittmacher, etc.) oder ein technisches System als Ersatz für die motorische Funktion anzusteuern (z. B. Prothese, Computer). Hierzu muss das MMI an das Reizübertragungssystem vor der Läsionsstelle angeschlossen werden.

1.2.3 Signalübertragung

Die Reizübertragungssysteme des menschlichen Körpers basieren sowohl auf chemischen als auch auf elektrischen Prozessen. Beispielsweise gibt der synaptische Spalt an der Endigung eines Axons (Fortsatz der Nervenzelle) Informationen durch Neurotransmitter an den anschließenden Effektor (oder eine Zelle) weiter (chemische Übertragung). Die Reizweiterleitung innerhalb von Nerven- oder Muskelfasern erfolgt hingegen auf der Basis von Aktionspotenzialen (elektrische Übertragung).

Ausgehend vom Ruhepotenzial (-90mV bis -70mV) entstehen diese durch Einströmen von Na^+ -Ionen durch die Zellmembran. Die Dauer bis zum Erreichen ihres Maximums (+10mV bis +30mV) wird als Depolarisation, die Rückkehr zum Ruhepotenzial als Repolarisation bezeichnet.

Ein Aktionspotenzial wird ausgelöst, wenn die Zellmembran ein Schwellenpotenzial von etwa -50mV überschreitet. Dies führt zur kurzzeitigen Öffnung von Na^+ -Kanälen und somit zur weiteren Depolarisation, die als Erregung bezeichnet wird. Zeitverzögert öffnen sich K^+ -Kanäle, und die Zelle repolarisiert durch einen K^+ -Ausstrom. Nach der Repolarisation beginnt ca. 1-2ms lang die Refraktärphase, in der keine weitere Depolarisation möglich ist. Diese Refraktärität begrenzt die maximale Frequenz des Auftretens von Aktionspotenzialen auf 500-1000Hz¹.

Die Fortleitung des Aktionspotenzials entlang der Faser geschieht durch die Spannungsdifferenz zwischen erregter und nicht erregter Zelle, die zur Depolarisierung der nicht erregten Zelle führt. Die Geschwindigkeit, mit der Aktionspotenziale übertragen werden, hängt einerseits von der Zelle selbst, andererseits von der Faser ab. Während sich das Aktionspotenzial in dicken, markhaltigen Fasern saltatorisch fortbewegt und Leitungsgeschwindigkeiten von bis zu 100m/s erreicht, beträgt die Geschwindigkeit in dünnen, marklosen Fasern 1m/s und weniger [179, 187].

¹Höherfrequente Anteile entstehen durch Oberwellen in den Aktionspotenzialen.

1.2.4 Mensch-Maschine-Schnittstellen zur Bewegungskontrolle

Um die Handlungsabsicht zu erfassen, können chemische oder elektrische Signale in Kortex, Nerven und Muskeln zur Interpretation herangezogen werden. Da die chemische Erfassung aufwändig ist (kleine Synapsen, nur punktuell vorhanden) und für elektrische Signale vielfältige Aufnahmemöglichkeiten bestehen, beschränkt sich die Erfassung fast ausschließlich auf die Detektion von Aktionspotenzialen. Unterschieden werden die Verfahren zur invasiven, direkten Messung der Aktionspotenziale (Elektrokortikographie (Hirn), Elektroneurographie (Nerven)) und die Verfahren zur nichtinvasiven, indirekten Messung (Elektroencephalographie (Hirn), Elektromyogramm (Muskel)). Mögliche Elektrodenpositionen zeigt Bild 1.4. Ein Vorteil der invasiven Verfahren ist die bessere Signalqualität und Selektivität, dem Einsatz stehen allerdings das hohe Infektionsrisiko sowie die notwendige kabellose Energie- und Signalübertragung entgegen. Um einen dauerhaften Einsatz zu gewährleisten, sind noch Biokompatibilität nachzuweisen und Telemetriesysteme zu entwickeln. Die Eigenschaften einiger gängiger Biosignale stellt Tabelle 1.1 dar. Hierbei ist zu beachten, dass bei der Ableitung u. U. eine Vielzahl von Zeitreihen durch den Einsatz mehrerer Sensoren erhalten werden kann.

Bioelektrisches Signal		Frequenz in Hz	Spannung in mV
EEG	Elektroencephalographie	1.0 - 70	0.005 - 0.1
ECoG	Elektrokortikographie	1.0 - 100	0.05 - 1
ENG	Elektroneurographie	0-50000	50 - 130
EMG	Elektromyographie (oberflächliche)	10-1000	0.1 - 5
-	Elektromyographie (subkutane)	10-10000	0.05 - 5

Tabelle 1.1: Frequenzen und Amplituden ausgewählter bioelektrischer Signale [50, 96]

Die von Hans Berger 1929 eingeführte Elektroencephalographie (EEG, [17]) misst mittels knopfförmiger Elektroden oberflächlich Potentialschwankungen auf der Kopfhaut eines Patienten. Diese beschreiben ein Summensignal der Aktionspotenziale der Hirnneuronen und zeichnen sich durch eine Amplitude von 5-100 μ V und einer Frequenz von 1-70Hz aus. Entsprechend werden große Amplituden nur dann erhalten, wenn eine synchrone Aktivierung der Neuronen vorliegt [181]. Aufgrund der indirekten Messung misst die Elektroencephalographie bei einer Elektrodenfläche von 1mm² bis zu 1.000.000 Neuronen. Die Platzierung der Elektroden ist an die zu untersuchenden Hirnareale (Abschnitt 1.2.2) anzupassen. Das EEG ist in Ruhe - besonders im Schlaf - niederfrequent und synchronisiert. Bei Hirnaktivität ist es desynchronisiert und höherfrequent.

Die Entwicklung sogenannter Brain-Computer-Interfaces (BCI) auf der Basis von EEG-Signalen erfolgt in erster Linie für Menschen mit schwersten motorischen Beeinträchtigungen. Von großem Vorteil erweist sich die nichtinvasive Signalaufnahme sowie die Unabhängigkeit von jeglicher Muskelkontraktion. BCIs stellen allerdings hohe Anforderungen an die Unterdrückung von Störsignalen durch elektromagnetische Felder. Derzeit werden dem Anwender Systeme zur Verfügung gestellt, mittels derer binäre Entscheidungen (links/rechts, ja/nein) durch reproduzierbare Aktivierungen von kortikalen Bereichen getroffen werden [19, 21, 116, 147, 209]. Diese Aktivierungen können beispielsweise durch das Denken an linke bzw. rechte Handbewegungen erzeugt werden. Die Vielzahl möglicher Umsetzungsmöglichkeiten beschreibt [104]. Allen Anwendungen ist jedoch gemein, dass nur wenige Steuerungssignale (meist 2) vom Patienten reproduzierbar generiert werden können und deren Auswertung schwierig, aufwändig und oft nicht eindeutig ist. Die größten Probleme gehen von unspezifischen Signalen aus. Deswegen sind langwierige Trainingsperioden notwendig, um eine zuverlässige Ansteuerung zu gewährleisten [19, 67]. Aktuelle Arbeiten auf dem Gebiet befassen sich mit der Umsetzung von

Schreibgeräten durch die sukzessive Auswahl von Buchstaben sowie der Ansteuerung von Handorthesen [19, 21, 90, 209]. In [134, 141, 147] werden EEG-Daten verarbeitet, um eine Tastatur zu bedienen. Durch das Denken an linke und rechte Handbewegungen wurde das System auch zur Steuerung einer Prothese verwendet [67]. Ein Klassifikationsfehler von 10% bei Auswahldauern von sechs Sekunden lässt jedoch die Schwierigkeiten im Alltagseinsatz vermuten. Eine neue Anwendung für querschnittgelähmte Patienten ist in [172] dargestellt. Hier dient das Ausgangssignal des Brain-Computer-Interfaces zur Elektrostimulation von gelähmten Gliedmaßen.

Die direkte Messung der Aktionspotenziale der Hirnneuronen mittels implantierter Elektroden wird als Elektrokortikographie (engl.: Electrocorticography, ECoG) bezeichnet. Sie zeichnet sich durch etwas höhere Amplituden und eine bessere Frequenz- und Ortsauflösung aus. Gemessen werden entweder lokale Feldpotenziale (engl.: Local field potential) mittels penetrierender Elektroden oder epikortikale Feldpotenziale auf der Oberfläche des Hirns. Die Anfänge der Bewegungsanalyse mittels ECoG gehen auf Humphrey et al. zurück, die nachwies, dass Aktivitäten von kortikalen Neuronen mit bewussten Bewegungen von Gliedmaßen korrelieren [76]. Aktuelle Forschungsschwerpunkte auf dem Gebiet der Brain-Machine-Interfaces auf der Basis elektrokortikaler Signale befassen sich mit der Ansteuerung künstlicher Gliedmaßen mittels im Tierversuch gewonnener Signale [33, 54, 119, 196]. Probleme ergeben sich einerseits durch Infektionen im Bereich der Elektroden, andererseits durch die Isolierung der wenigen Neuronen, die für die Bewegung zuständig sind. Problematisch sind vor allem Umstrukturierungen kortikaler Gebiete und Positionsveränderungen der Elektroden. [33] wertet Neuronen im motorischen Kortex von Ratten zur Steuerung eines einfachen Roboterarms aus, dabei wird ausschließlich zwischen Aktivierung und Deaktivierung unterschieden. Die Weiterentwicklung dieses Brain-Machine-Interfaces beschreiben [118, 119]. Über mehrere Monate hinweg konnte eine kleine Anzahl kortikaler Neuronen eines Affen erfasst werden. Die Ergebnisse dienen zur Echtzeitvoraussage von Bewegungstrajektorien, um die Steuerung eines Roboterarms zu ermöglichen. In [196] werden Makaken² trainiert, durch ein Brain-Machine-Interface einen Cursor zu steuern. Eine virtuelle Realität generiert acht verschiedene Objekte, die vom Versuchstier gegriffen werden sollen. Die Klassifikationsgüten lassen darauf schließen, dass der hier verwendete Ansatz funktioniert, eine zuverlässige Steuerung kann bislang jedoch nicht erstellt werden. Ähnliche Ansätze finden sich in [20]. Ein alternativer Ansatz ist die Erfassung und Verarbeitung großer Bereiche von Neuronen [54, 139]. Als Versuchsobjekte werden Affen verwendet, die verschiedene Muster erlernt haben, um einen Roboterarm zu steuern. Hierbei werden Aktivitäten der Interneuronen im Rückenmark bei Flexionen-/Extensionen des Handgelenks untersucht. [7] beschreibt eine fortgeschrittene Arbeit, die epikortikale Feldpotenziale eines menschlichen Probanden verwendet, um eine Aussage über dessen Zeigefingerbewegung abzuleiten.

Sollen die zu einer Bewegung gehörigen Aktionspotenziale nicht im zentralen Nervensystem, sondern an peripheren Nervenfasern gemessen werden, so bietet sich die Elektroneurographie (ENG) an. Als Sensoren kommen den Nerv umschließende Cuff-Elektroden, vom Nerv durchwachsene Sieb-Elektroden oder den Nerv penetrierende Brush-Elektroden zum Einsatz [193]. Während Sieb- und Brush-Elektroden eine Vielzahl von Nervenfasern getrennt messen, erfassen Cuff-Elektroden nur die Auswirkungen des die Nervenfasern durchfließenden Stroms. Ähnlich zum EEG wird ein Summensignal vieler Nervenfasern erhalten. Als vorteilhaft erweisen sich jedoch das geringe Risiko und der geringe Aufwand beim Einbringen und Entfernen der Elektroden. Cuff-Elektroden an den Nervenenden sind im experimentellen, teils auch im klinischen Einsatz [171], zwei- und dreidimensionale Brush-Elektroden und Sieb-Elektroden befinden sich in der Testphase [173]. Es gilt es zu beachten, dass eingebrachte Sensoren sowohl der efferenten und afferenten Ableitung der Nervensignale als auch der efferenten und afferenten Stimulation der Nervenfasern dienen können (Bild 1.5). Die efferente Stimulation dient der Kontraktion des mit dem Nerv verbundenen Körperteils, die afferente Stimulation führt Meldungen über die Bewegungsausfüh-

²Meerkatzenartige Primaten

rung an das Gehirn zurück [203]. Die afferente Ableitung erfasst körpereigene sensorische Informationen des mit dem Nerv verbundenen Körperteils, die efferente Ableitung erfasst eine Bewegungsabsicht des Probanden.

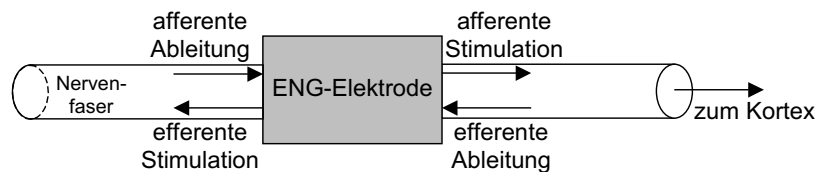


Bild 1.5: Ableitung und Stimulation an einer Nervenfasern mittels einer ENG-Elektrode

Um eine künstliche Hand zu kontrollieren (öffnen/schließen) hat sich Warwick in einem ersten Humanversuch eine efferent messende Brush-Elektrode in den nervus medialis einsetzen lassen [205]. Bereits nach wenigen Wochen wurde das Implantat jedoch abgestoßen. Um das Zusammenspiel zwischen Mensch und künstlichem System zu optimieren, wird neben dem Ableiten efferenter Nervensignale versucht, für Mensch und/oder Maschine bzw. MMI Feedbacksignale durch das Stimulieren und/oder das Ableiten von afferenten Nerven zu generieren. Zum Einsatz kommt dieses Verfahren bereits im Tierversuch [24, 25], klinische Studien mit Probanden werden in [189] beschrieben. Großer Beliebtheit erfreut sich die efferente Nervenstimulation (funktionelle Elektrostimulation, FES) zur Stimulation von Muskelgewebe [5, 133, 192, 194, 212]. So existieren bereits kommerzielle Systeme zur Stimulation von gelähmten Gliedmaßen, wie der Hand (Freehand, [171]). Ein ähnliches Forschungsthema befasst sich mit der Wiederherstellung von Bewegungsmustern durch FES, z. B. zum Stehen und Gehen (Stand-Gang-Neuroprothese [143]). Weitere kommerzielle Anwendungen bezüglich der afferenten Stimulation finden sich im Bereich der Sinnesreizung von Augen und Ohren. So werden spezielle Formen der Blindheit gelindert, indem ein Chip mittels Mikroelektroden den Sehnerv auf der Retina stimuliert [65, 69]. Um die verlorene Funktion der Haarzellen des Hörnervs zu ersetzen, werden Cochleaimplantate eingesetzt, mit denen sich der Hörnerv direkt reizen lässt [108]. Einen Schritt weiter gehen Hirnstammimplantate (engl.: Auditory Brainstem Implant), die im Gegensatz zum Cochleaimplantat direkt an die akustisch relevanten Areale des Hirnstamms angreifen [132, 191].

Ebenso wie Aktionspotenziale aus dem Nervensystem können Aktionspotenziale der Muskelfasern als Signalträger dienen (Elektromyographie, EMG). Die Messung findet oberflächlich oder intramuskulär mittels Nadel-Elektroden statt. Intramuskulär können myoelektrische Potenziale, respektive Spannungen der motorischen Einheiten direkt gemessen werden. Die oberflächige Ableitung führt zu qualitativ minderwertigen Signalen, da nur ein Summensignal mehrerer Muskelfasern und/oder mehrerer Muskelgruppen gemessen wird. Weiterhin wirkt das Gewebe zwischen Muskel und Sensor als Filter für hohe Frequenzanteile [97]. Die oberflächige Abtastung schränkt den auswertbaren Frequenzbereich myoelektrischer Signale auf ca. 10 bis 1000Hz ein [50, 96]. Im Falle einer Läsion der abgegriffenen Muskelgruppen verschlechtert sich die Signalqualität weiterhin. Damit gehen zwar für die Interpretation des Signals wichtige Informationen verloren, jedoch eröffnet sich der Vorteil einer sehr unkompliziert anwendbaren Technik. Untersuchungen mit subkutanen bzw. implantierten Elektroden zeigen am kurzen Amputationsstumpf bei der Ansteuerung von Prothesen keine signifikanten Vorteile gegenüber Oberflächensensoren. Bei längeren Armstümpfen können sich hingegen Verbesserungen zeigen [36].

Prinzipiell existiert zu jeder elektrischen Erfassungsart ein magnetisches Äquivalent (Magnetomyographie, Magnetoencephalographie, ...). Die durch elektrische Ladungsverschiebungen entstehenden

magnetischen Felder lassen sich mit der SQUID (Superconducting Quantum Interference Device)-Technologie erfassen. Im klinischen Einsatz befindet sich größtenteils die Magnetoencephalographie (MEG). Durch MEG erhaltene Signale ähneln denen des EEGs, jedoch ist die räumliche Auflösung der aktivierten Areale um ein Vielfaches besser. Aufgrund der komplizierten Ableitung und der nicht mobilen Gerätschaften verzichtet diese Arbeit jedoch auf eine weitere Betrachtung des MEG.

Zur Umsetzung von Prothesenbewegungen auf der Basis von efferenten biometrischen Signalen eignet sich insbesondere das EMG aufgrund der einfachen Zugänglichkeit der Muskulatur. Hier hat sich eine Vielzahl verschiedener Signalerfassungsmethoden entwickelt. In den Anfängen der Handprothetik wurde die verbliebene Stumpfmuskulatur kanalisiert und mit Stiften versehen, die den Muskelhub direkt auf die Bewegung der Prothese übertragen haben [174]. Diese als Myocineplastie bezeichnete Methodik ist allerdings für den Patienten unangenehm und kann zu Entzündungen führen. Die Myoakustik misst mit piezoelektrischen Kristallmikrofonen die Emissionen akustischer Signale, die bei der Kontraktion eines Skelettmuskels entstehen. Von Vorteil sind die berührungslose Signalerfassung sowie die Unabhängigkeit bei Impedanzänderungen der Haut [10, 136]. In [198] konnte eine lineare Beziehung zwischen Beißkraft und myoakustischem Signal nachgewiesen werden. Eine ausreichende Genauigkeit bei der Positionierung einer prothetischen Hand konnte bislang hingegen nicht nachgewiesen werden [11, 210]. So ergeben sich unerwünschte Effekte durch eine große Anzahl von Einflussfaktoren, wie z.B. Form des Muskels, Gewicht, Länge, Temperatur, etc.

Die myokinemetrische oder myovolumetrische Abtastung befasst sich mit der oberflächlich erfassbaren Veränderung der Form sich kontrahierender Muskeln. Volumetrische Änderungen können durch verschiedene Wirkprinzipien erfasst werden:

- Kenney et al. benutzen Magnetfelder, um Veränderungen von Muskelpositionen zu detektieren. Probleme ergeben sich hierbei mit der Dynamik des Systems. Untersuchungen mit Menschen, die einen Arm verloren haben, zeigen, dass das entstehende Signal den Ansprüchen zur Steuerung einer Prothese genügt, die Experimente wurden allerdings im Sitzen und ohne Einflüsse von Störgrößen durchgeführt [80].
- Mit Hilfe einer bewegungssensiblen Codierungstechnik werden Oberflächendeformationen durch ein Lichtabtabverfahren erfasst. Oberflächendeformationen eines Messobjektes werden in einen charakteristischen digitalen Code transformiert, als Anwendungsgebiete werden Prothesen genannt [126].
- Weiterhin können Drucksensoren dazu verwendet werden, charakteristische Druckmuster aufzuzeichnen [41, 57]. Als erheblicher Nachteil erweist sich bei der Unterarmbeugung die Kontraktion der Unterarmbeuger und die damit verbundenen Volumenänderungen des Stumpfs [23]. Die erste und einzig klinisch angewandte myokinemetrische Prothese wurden 1953 von Wilms entwickelt [208]. Hierbei wurde eine pneumatische Blase verwendet, um Ausbuchtungen des abgetasteten Muskels zu detektieren.

Vergleichbar zur myokinemetrischen Abtastung kann in der myosensorischen Abtastung auch die Steifheit eines Muskels Aussagen über den jeweiligen Kontraktionszustand liefern. Mit zunehmender Kontraktion versteift sich ein Muskel und setzt Körpern, die mit definierter Kraft auf ihn einwirken, einen erhöhten Verformungswiderstand entgegen [117]. Die Erprobung an Patienten hat allerdings ergeben, dass sich die elastischen Eigenschaften des Muskels in verschiedenen Stellungen verändern [23].

1.2.5 Bewertung der Signalquellen für den Einsatz in Mensch-Maschine-Schnittstellen

Das Steuern eines MMIs setzt die efferente Ableitung biometrischer Signale voraus, da hiermit Handlungsabsichten vom Anwender übermittelt werden. Ein Signaltyp eignet sich dann besonders gut, wenn

- dieser Signaltyp vom Anwender bewusst generiert,
- physiologisch ähnliche Signale verwendet,
- eine hohe Signalqualität und geringe Störeinflüsse erreicht,
- ein invasiver Eingriff vermieden oder zumindest auf ein Minimum reduziert und
- die Auswertung des Signals recheneffizient vorgenommen

werden kann. Insbesondere zur Steuerung künstlicher Extremitäten (Prothesen) sind Signale zusätzlich lokal, d. h. nahe der Extremität aufzunehmen, die physiologische Ähnlichkeit ist dann anwendungsabhängig zu beurteilen.

Elektrokortikographische Signale sind zur sicheren und gefahrlosen Steuerung von MMIs bislang nicht einsetzbar, ein möglicher Einsatz ist noch nicht abzusehen [118]. Zwar existieren zahlreiche internationale Forschungsprojekte, als grundlegende Herausforderung ist jedoch vorab die Entwicklung von Elektroden und chirurgischen Methoden zu sehen, die eine sichere und langfristige Aufzeichnung ermöglichen [149].

Große Anstrengungen werden unternommen, um Fehlerquoten und Zeitverzögerungen der EEG-Schnittstellen zu vermindern. Der derzeitige Entwicklungsstand ermöglicht zwar den Einsatz von EEG-Signalen für Schnittstellen zwischen Mensch und Maschine. Er lohnt sich jedoch nur dann, wenn eine vollständige Lähmung des Patienten vorliegt (Locked-in Syndrom). Für den Einsatz in anderen MMIs ist der Ansatz ungeeignet.

Bei der peripheren Ableitung von Nervensignalen handelt sich um eine vielversprechende Methode. Für den permanenten Einsatz im menschlichen Körper müssen jedoch vorab die Sensoren miniaturisiert, die Bioverträglichkeit anhand von Tierversuchen nachgewiesen, die vermuteten Vorteile (hohe Auflösung und Effektivität) verifiziert und die Langzeit-Biokompatibilität erreicht werden [173].

Als Eingangsgröße eines MMIs kommen (abgesehen von Spezialanwendungen) für einen dauerhaften Einsatz gegenwärtig ausschließlich Signale in Frage, die auf Muskelkontraktionen beruhen.

In Bezug auf die Kriterien Signalqualität, Invasivität und Recheneffizienz fasst Tabelle 1.2 die vorangegangenen Überlegungen über die Eignung der angeführten muskelbasierten Verfahren für den Einsatz in einem MMI zusammen. Demnach ist das oberflächige EMG als Signal zu bevorzugen.

Methode	Signalqualität	Invasivität	Recheneffizienz
EMG, implantiert	+	-	+
EMG, oberflächig	o	+	+
Myoakustik	o	+	-
Myocineplastie	+	-	+
Myokinemetrie	o	+	o
Myosensorik	o	+	o

Tabelle 1.2: Bewertung der Signalqualität muskulärer Signale zur Steuerung eines MMIs

1.2.6 Signalverarbeitung und -auswertung

Eine schwierige Aufgabe stellt die Interpretation eines biometrischen Signals dar. Einerseits ist das Signal in aussagefähige und leicht zu verarbeitende Größen zu transformieren (Merkmalsextraktion), andererseits ist die Handlungsabsicht des Anwenders aus diesen Größen zu ermitteln. Die Auswahl eines entsprechenden Verfahrens ergibt sich oft aus den Vorlieben des Entwicklers. Viele Arbeiten nutzen Zeit-Frequenz-Transformationen, um Merkmale zu berechnen. Verfahren zum Ermitteln der Handlungsabsicht entstammen fast ausschließlich der Klasse der lernenden bzw. parametrierbaren Systeme (Künstliche Neuronale Netze, Fuzzy-Regeln, statistische Verfahren, etc.). Diese berechnen auf der Basis von Beispielsignalen Parameter für ihre individuellen Funktionsansätze, die in ihrer Gesamtheit eine Entscheidungsregel formen. Zum exemplarischen Vergleich der Eignung von MMIs bezüglich der Auswertung von EEG-Daten findet jährlich ein Wettbewerb statt [22]. Ein systematischer Vergleich in verschiedenen Einsatzszenarien existiert für die Berechnung von Merkmalen bedingt, für den Einsatz von lernenden Systemen hingegen nicht. Ebenso wenig finden sich Anforderungsanalysen, die insbesondere in Echtzeitsystemen von Bedeutung sind.

1.2.7 Stand der Technik: Myoelektrische Handprothesen

Ein besonders weit entwickeltes, umfassendes und wichtiges Gebiet, in dem MMIs zum Einsatz kommen, ist die kommerzielle Entwicklung von funktionellen Handprothesen [1, 127]. Mittels der verbliebenen Muskulatur im Armstumpf steuert der Anwender einen oder zwei Freiheitsgrade seines künstlichen Gliedmaßes (z. B. Handgelenk drehen und Greifen). Unter Zuhilfenahme mechanischer Hilfsmittel können im Einzelfall sogar bis zu drei Freiheitsgrade bedient werden. Die Anwender verlangen jedoch nach Handprothesen mit hoher Funktionalität und natürlichem Erscheinungsbild. Letzteres erfüllen zwar sogenannte kosmetische Prothesen, hingegen sind Bewegungen nicht möglich. Internationale Untersuchungen in [6, 145, 188] ergaben, dass mindestens jede dritte Handprothese nicht regelmäßig getragen wird. Ein ausschlaggebender Punkt ist laut den Anwendern die eingeschränkte Funktionalität kommerziell erhältlicher Prothesen, die dem Anwender oft nur eine Griffart (Zylindergriff) bieten [125, 129, 131, 135]. Für Anwendungen in der Robotik und in der Raumfahrt wurden zwar zahlreiche anthropomorphe Greifsysteme mit einem höheren Grad an Funktionalität entwickelt, die komplexe Aufgaben wie das Manipulieren eines Gegenstandes zwischen den Fingerspitzen bewältigen können [29, 64, 101]. Jedoch werden diese Systeme den Anforderungen an Prothesen nicht gerecht, da sie zu groß und zu schwer sind und zudem einen zu hohen Energiebedarf aufweisen.

Die vergangenen Jahre brachten zunehmend Greifsysteme mit einer Fülle von neuen Bewegungsmöglichkeiten auf engstem Bauraum hervor [32, 42, 92, 185]. Um dem Anwender diese Bewegungsmöglichkeiten zugänglich zu machen, versuchen [53, 73, 75, 122], die Schnittstelle zwischen Mensch und Maschine zu erweitern. Es handelt sich hierbei jedoch um Vorversuche in Laborumgebungen, wobei sich die Untersuchungen hauptsächlich auf gesunde (nichtamputierte) Probanden beziehen. Diese steuern ihre Prothese durch EMG-Signale der für die Bewegung zuständigen Muskeln (Leitmuskeln). Zur Beschreibung zugehöriger MMIs stellt diese Arbeit vorab die Eigenschaften der myoelektrischen Signalaufnahme eingehender dar.

Die Weiterleitung eines Aktionspotenzials im Muskel kann vereinfacht als wandernder Dipol interpretiert werden, dessen Feldlinien sich im Körper ausbreiten und oberflächlich detektiert werden können. Als myoelektrische Spannung wird die Potentialdifferenz zwischen der aktuellen Größe des Aktionspotenzials und dem Ruhepotential einer Muskelzelle bezeichnet. Da jedoch das Anspannen eines Muskels auf der Kontraktion mehrerer Muskelzellen und -fasern beruht, wird gemeinhin das gemessene (und evtl. gefilterte) Summensignal als myoelektrische Spannung bezeichnet. Eine Messung des Ruhepotenzials

einer Zelle ist i. A. nicht möglich, deshalb kommen Ersatzpotenziale zum Einsatz. In Abhängigkeit von der Wahl des Ersatzpotenzials wird zwischen mono- und bipolarer Ableitung unterschieden.

Die monopolare Ableitung erfasst oberflächlich messbare Potenziale der Muskelzellen und setzt diese in Bezug zum Körperpotenzial. Als vorteilhaft stellt sich hierbei die Unabhängigkeit des Messergebnisses von der Richtung des Muskels heraus. Allerdings überlagern Störungen das Signal, die zwischen Signal- und Bezugselektrode auftreten.

Die bipolare Ableitung setzt die Potenziale zweier Elektroden in Bezug, die etwa 1-3cm voneinander entfernt liegen und üblicherweise den gleichen Muskel abgreifen. Eine Bezugselektrode verbindet die Masse des Messverstärkers mit dem Körperpotenzial. Eine Überlagerung mit Störsignalen wie im Falle der monopolen Ableitung besteht kaum.

Aufgrund der niedrigen Störanfälligkeit der bipolaren Ableitung hat sich diese Methode für den Einsatz in der Prothetik durchgesetzt. Sie geht allerdings mit folgenden Einschränkungen einher:

- Das oberflächige Abtasten von Potenzialfeldern führt zu Summensignalen aller unter dem Sensor befindlichen Muskelfasern und -gruppen. Tiefergelegene Muskelgruppen können nicht erfasst werden [82, 83]. Weiterhin wirkt das zwischen Sensor und Muskel liegende Gewebe als Tiefpassfilter [97].
- Die erfassten Spannungen sind klein und sehr anfällig für Störungen, wie z.B. Bewegungsartefakte durch Verschieben der Elektroden sowie elektrische Einstreuungen durch Niederfrequenzstörer (z. B. Stromnetz) und Hochfrequenzstörer (z. B. Mobiltelefon) [79].
- Das myoelektrische Signal variiert in Abhängigkeit vom Abtastort sowie des Punktes der Ankopplung der Muskeln an die Nerven stark. Dieser Punkt liegt individuell verschieden, und somit können keine allgemeingültigen Aussagen über den Signalverlauf bei unterschiedlichen Personen gemacht werden.
- Schweißabsonderungen unter dem Sensor (zeitvariante Hautübergangswiderstände) und Ermüdungserscheinungen der abgetasteten Muskeln führen zu Variationen in der Amplitude des gemessenen EMG-Signals [46, 140].

Die Kraft eines Muskels ergibt sich durch die Innervationsrate der motorischen Einheiten und die Anzahl der an einer Kontraktion beteiligten motorischen Einheiten. Folglich bestimmen zu Anfang einer Muskelkontraktion die Aktivierungen einzelner motorischer Einheiten das Signalbild. Mit steigender Kraft überlagern sich die Aktivierungen aller unter dem Sensor befindlichen Muskelfasern und -gruppen, und es ergibt sich ein Summensignal (vgl. Bild 1.6). Eine ausführliche Übersicht geben [12, 43, 44].

Der Einsatz von Oberflächen-EMG-Sensoren erfordert die Abtastung oberflächennaher Muskelgruppen. Im Falle von Unterarmläsionen eignen sich Signale der Strecker und Beuger des Handgelenks (*musculus extensor digitorum* und *musculus flexor carpi radialis*), bei Oberarmläsionen der Strecker und Beuger des Ellbogens (*musculus triceps brachii* und *musculus biceps brachii*). Eine Signalaufnahme der für die ursprüngliche Bewegung physiologisch sinnvollen Muskeln ist in den meisten Fällen unmöglich, da

- die zugehörigen Muskeln meist lädiert oder gar nicht vorhanden sind,
- die Stumpfmuskulatur ohne Training verkümmert, somit niedrigere oder gar keine Signalamplituden generiert werden können und folglich nur starke Muskelgruppen erfasst werden können, und
- bei der Amputation i. A. die Stumpfmuskeln miteinander vernäht werden und somit keine eigenständigen Signale mehr generieren können [207].

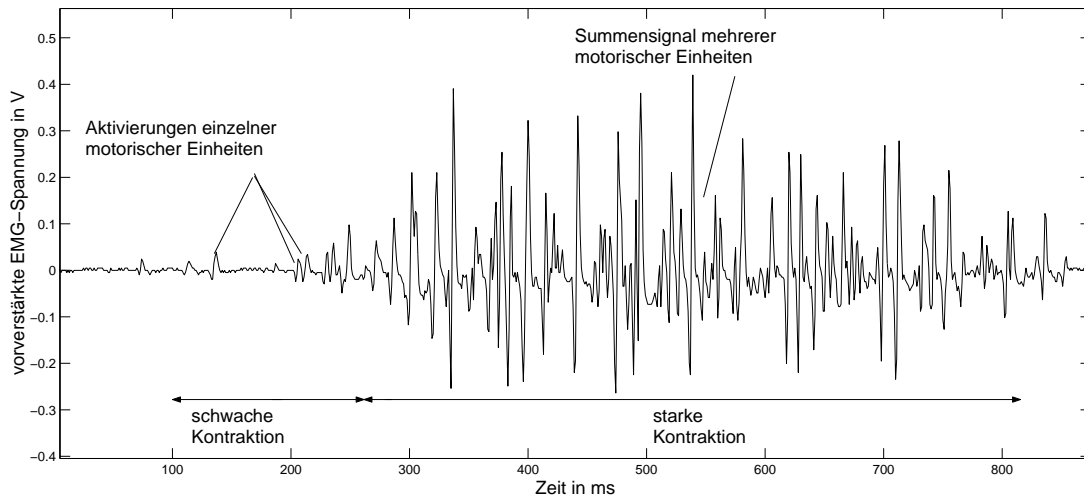


Bild 1.6: Bipolares EMG-Signal von musculus extensor digitorum

Letztgenannter Punkt führt sogar dazu, dass das Zusammenziehen einer Muskelgruppe ein Auseinanderziehen der anderen zur Folge hat. Dieser Vorgang wird als Übersprechen bezeichnet und führt bei einer Messung zu Signalen auf beiden Sensoren.

Meist werden unnatürliche, künstliche Bewegungsmuster verwendet, um die Freiheitsgrade einer Prothese zu bedienen. Zwar sind Patienten so gezwungen, die Prothesensteuerung zu erlernen, jedoch kann auf diese Weise die größtmögliche Robustheit der Steuerung garantiert werden. Die zu verwendenden EMG-Signale sind sinnvollerweise dem Stumpf zu entnehmen, können aber auch unphysiologisch von anderen Muskelgruppen abgeleitet werden.

Die Forschung im Bereich der EMG-Prothesen begann im Jahre 1948 mit der Einführung der Elektrokunsthand [166]. Seinem Vorbild folgten hybride Systeme, die eine Kombination aus mechanischer und myoelektrischer Steuerung anstrebten. Es wurden einfache Methoden entwickelt, um mehr als zwei Funktionen mit zwei EMG-Sensoren auszuführen, z. B. indem Bereichen der Signalamplitude verschiedene Funktionen zugewiesen wurden [47]. Die Methoden konnten sich jedoch aufgrund von Robustheitsproblemen nicht durchsetzen.

1967 schlug Finley [55] ein neues Konzept zur multifunktionalen Steuerung vor, das auf der Anwendung von Mustererkennungsalgorithmen basierte. Diesem Ansatz folgten viele Forschungsarbeiten, die bekanntesten sind die Arbeiten von Herberts [72] und Hudgins [75], vgl. Tabelle 1.3. Während Herberts versuchte, eine lineare Diskriminanzanalyse (LDA) auf einfache Merkmale wie gleichgerichtete und bandpassgefilterte Sensorsignale anzuwenden, befasste sich Hudgins mit der Extraktion von informationsträchtigen Merkmalen. Die Klassifikation dieser Merkmale übernahm ein Künstliches Neuronales Netz (KNN), in diesem Fall ein Multilayer-Perzeptron-Netz (MLP). Ein wichtiger Unterschied zu Herberts war, dass Hudgins seine Merkmale zu Beginn einer Kontraktion in einem vorgegebenen Zeitfenster generierte und dann eine Klassifikationsentscheidung erzwang. Herberts hingegen führte zu jedem Abtastzeitpunkt eine Klassifikation durch.

Zwar zeigten diese Bemühungen die prinzipielle Machbarkeit einer multifunktionalen Steuerung, für den Einsatz am Patienten waren die Systeme jedoch zu spezifisch und unzuverlässig. Kommerzielle myoelektrische Handprothesen besitzen deshalb nur ein bis drei Freiheitsgrade und lassen sich mittels

zweier gleichgerichteter und tiefpassgefilterter Sensoren über einfache Schwellwertstrategien bedienen. Anbieter dieser Systeme sind u. a. Otto Bock, Motion Control, Hosmer und Steeper [129, 130, 131, 135].

Die zugrundeliegenden Steuerungen gliedern sich in digitale und proportionale Steuerungen. Die einfach zu bedienenden digitalen Steuerungen (Zweikanalsteuerung: Griffartenauswahl durch Muskelgruppe, Doppelkanalsteuerung: Griffartenauswahl durch Kontraktionsamplitude, Systeme mit Totbandunterdrückung) zeichnen sich durch konstante Schließ- bzw. Öffnungsgeschwindigkeiten nach dem Überschreiten eines Schwellwerts aus. Proportionale Steuerungen generieren zum EMG-Signal proportionale Geschwindigkeiten. Bei integrierter Greifkraftsteuerung schaltet die Prothese bei Überschreiten eines Greifkraftschwellwerts in einen Kraftsteuermodus, in dem die Greifkraft proportional zum EMG-Signal gesteuert wird [144]. Hierzu sind dem System Abhängigkeiten zwischen EMG-Signal und Motorstrom in Form von Tabellen hinterlegt. Zusätzlich können Greifkräfte bei Vorhandensein eines Kraftsensors in den Fingerspitzen sogar geregelt werden. Durch eine Kokontraktion (gleichzeitige Kontraktion beider Muskeln) kann zwischen zwei Bewegungszuständen umgeschaltet werden. Auf diese Weise können gleiche Muskeln zur Steuerung weiterer Bewegungen verwendet werden - im Fall der Otto Bock-Prothese zur Drehung des Handgelenks. Weitere Informationen finden sich in [170].

Bisher gibt es keine kommerzielle Handprothese, in der ein taktiles Biofeedback integriert ist, weswegen eine visuelle Kontrolle beim Greifvorgang notwendig ist. Feedbackgebende Systeme befinden sich jedoch in der Entwicklung [103]. Hierfür werden taktile Sensoren in den Fingerspitzen und kompakte Winkelsensoren in die Gelenke der Finger integriert [115].

Die individuelle Anpassung kommerzieller Prothesen geschieht durch die Anfertigung eines Prothesenschafts als Negativ des Armstumpfs sowie die patientenspezifische Suche nach Elektrodenkontaktstellen und dem Training entsprechender Muskeln. Hierfür kommen mobile Geräte zur LED-Anzeige der EMG-Spannungen zum Einsatz (Otto Bock: MyoBoy, Motion Control: Myolab II). Die Güte einer Position ergibt sich aus der Amplitude des Signals, der Aktivierbarkeit durch den Patienten, der Unabhängigkeit von anderen Sensormesspositionen sowie der Lokalisierung im Bereich des Prothesenschafts [79, 135]. Ist eine geeignete Position gefunden, kann schließlich die Ausgangsspannung des Sensors durch ein Potentiometer angepasst werden.

Zahlreiche Forschungsvorhaben der letzten Jahre versuchen Steuerungen zu entwickeln, um dem Patienten mehr Bewegungsarten zu ermöglichen. Im Vergleich zu den kommerziellen Anbietern kamen

- eine erhöhte Elektrodenanzahl,
- aufwändigere Methoden der Signalvorverarbeitung (z.B. Spektralanalysen, bandpassgefilterte Signale, Korrelationsanalysen, Parameterschätzungen mit autoregressiven Modellen (AR), Wavelets) oder
- aufwändigere Klassifikatoren (Künstliche Neuronale Netze, statistische Klassifikatoren, Fuzzy-Regelbasen)

zum Einsatz.

Einige Projekte, die sich mit der prothetischen Umsetzung einer myoelektrischen Steuerung befassen, stellt Tabelle 1.3 vor. Allen ist gemein, dass sie entweder nach dem Prinzip von Herberts oder dem von Hudgins arbeiten.

Die Arbeiten versuchen durch hohe Abtastfrequenzen, den Frequenzinhalt von mindestens zwei myoelektrischen Signalen auszubeuten. Dies führt einerseits zu rechenaufwändigen Verfahren und andererseits zu kurzen Zeitfenstern, in denen Berechnungen möglich sind. Eine Auswertung auf mobilen Mikrocontrollersystemen, wie sie in Prothesen eingesetzt werden, liegt jedoch noch in weiter Ferne. Zudem

	Jahr	Ziel	Patienten	Elektroden	Griffarten	f_a (Hz)	Merkmale	Klassifikator
Herberts [72]	1978	Prothesensteuerung	4 amputiert	6	6	n/a	Gleichrichter, Bandpassfilter	LDA
Hudgins [75]	1993	Prothesensteuerung	6 amputiert, 9 gesund	2	4	1000	einfache algebraische Operationen	KNN
Eriksson [53]	1998	Tracking v. Fingerbewegungen	1 amputiert	8	5	8000	Bandpassfilter	SOM
Huang [73]	2000	Prothesensteuerung	1 gesund	3	8	2500	Frequenz-Modelle, Merkmale aus [75]	KNN
Nishikawa [122, 123]	2001	Prothesensteuerung	1 amputiert, 5 gesund	2	8	1600	Frequenz-Modelle	KNN
Barrero [9]	2001	Prothesensteuerung	1 gesund	4	7	n/a	Fourier-Transformation	KNN
Peleg [138]	2002	Erkennen v. Fingerbewegungen	4 gesund	2	5	500	Fourier-Transformation, AR-Modelle	KNN
Cunha [61]	2002	Prothesensteuerung	n/a	2	4	1600	Bandpassfilter	KNN
Ajiboye [3]	2002	Prothesensteuerung	1 amputiert, 1 gesund	4	4	1400	Bandpassfilter	Neuro-Fuzzy
Lamounier [93]	2002	Prothesensteuerung	n/a	n/a	4	n/a	AR-Koeffizienten	KNN
Micera [109, 214]	2002	Prothesensteuerung	6 gesund	7	3	1000	AR-Koeffizienten	Fuzzy, LDA
Englehart [51, 52]	2003	Prothesensteuerung	11 gesund	4	6	1000	Wavelet-Methoden	KNN, LDA

Tabelle 1.3: Forschungsschwerpunkte auf dem Gebiet der myoelektrischen Prothetik (n/a: keine Angaben, f_a : Abtastfrequenz)

führt eine Erhöhung der Anzahl der Oberflächenelektroden inklusive der dafür benötigten miniaturisierten Filter und Vorverstärker einerseits zu deutlich höheren Kosten für das Gesamtsystem andererseits zu einem hohen Rechenaufwand. Die meisten Arbeiten bewerten ihre Verfahren zudem nur anhand von gesunden Probanden.

1.2.8 Automatischer Entwurf von Schnittstellen für biologische Systeme

Der automatische Entwurf zur Verarbeitung biologischer Signale geschieht datenbasiert anhand von Beispielsignalen bzw. Datentupeln, die von einem Patienten oder Probanden gemessen und kategorisiert werden. Der Entwurf muss dann für jeden Patienten automatisch

- recheneffiziente, trennstarke Merkmale aus den Signalen extrahieren,
- Klassifikatoren für Merkmale berechnen, bewerten und auswählen und
- für den gesamten Ablauf echtzeitfähigen Quelltext generieren³.

Das Extrahieren von Merkmalen ist für jede Anwendung individuell an die Signale anzupassen. Hier existieren nur wenige wissenschaftliche Vorgaben, die nicht verbindlich und signalübergreifend zu ver-

³Echtzeitfähigkeit bezeichnet die Bereitschaft, anfallende Daten in einer vorgegebenen Zeitspanne garantiert zu bearbeiten.

wenden sind [26, 51]. Es handelt sich um einen iterativen Prozess, der die Vorlieben des Entwicklers widerspiegelt. So existieren Arbeiten, die Frequenzspektren in EEG- oder EMG-Signalen verwenden [51, 177], Einzelmerkmale (Extrema, Mittelwerte, etc.) für Bewegungsanalysen [99], oder einfache algebraische Operationen zur Charakterisierung von Handbewegungen [75].

Zur Zuordnung von Datentupeln zu einer Klasse (z. B. Griffklasse bei Prothesen) kommen Klassifikatoren zum Einsatz. Ein Klassifikator ist ein parametrierbares Verfahren, dessen Parameter anhand von Beispielen (z. B. von Patienten) angepasst wurden. So ist er in der Lage, über unbekanntes Beispiele zu extrapolieren. Bekannte Klassifikatoren sind Fuzzy-Regelbasen [8, 81, 213], Künstliche Neuronale Netze [71, 167, 168], k-Nearest-Neighbour-Verfahren [38], Support-Vektor-Maschinen [199, 200] oder approximative Maximum-Likelihood (AML)-Klassifikatoren [195]. Klassifikatoren liefern grundsätzlich bessere Ergebnisse, je weniger Klassen ein Datensatz enthält und je mehr Beispiele pro Klasse zur Verfügung stehen. Weiterhin sind alle Klassifikatoren darauf angewiesen, dass die vorliegenden Datentupel bestimmte Voraussetzungen erfüllen, z. B. achsparallele Klassifikationsgebiete bei Fuzzy-Regelbasen oder normalverteilte Daten bei AML-Klassifikatoren. Insbesondere bei Patientendaten treffen viele dieser Voraussetzungen nicht zu. Grundsätzlich sind nur wenige Beispiele vieler Klassen vorhanden, da einerseits der Patient beim Sammeln der Beispiele nicht überlastet werden, andererseits die Funktionalität des MMIs so hoch wie möglich sein soll. Anwenden lassen sich die Klassifikatoren zwar trotzdem, es steigt allerdings die Anzahl der fehlerhaften Klassifikationen, weil die Beispiele auswendig gelernt wurden und eine Generalisierungsfähigkeit der Klassifikatoren nicht gegeben ist. Umfassende Untersuchungen über eine spezielle Anpassung der Klassifikatoren an die mit Patientendaten auftretenden Randbedingungen existieren nicht. Ein weiteres Problem stellt die Implementierung eines angelegten Klassifikators in einer echtzeitfähigen, mobilen Umgebung (z. B. Mikrocontroller) dar. Oft ist sie sehr rechenaufwändig, so dass notwendige Abtastfrequenzen der Signale nicht mehr gewährleistet werden können. Hier existieren ebenfalls keine systematischen Untersuchungen, welche Voraussetzung ein Klassifikator erfüllen muss, um implementierbar zu bleiben.

Schließlich ist es notwendig, den individuellen Entwurf der Merkmalsextraktion und des verwendeten Klassifikators automatisch in Quelltext zu übertragen und diesen beispielsweise auf einem Mikrocontroller zu implementieren. Hierbei ist darauf zu achten, dass manuelle Anpassungen des Quelltextes vermieden werden, um Fehlerquellen auszuschließen und den Aufwand zu reduzieren. Zwar existieren spezielle Anwendungen, die Klassifikatoren vollautomatisch in Echtzeitumgebungen portieren [177], hierbei handelt es sich jedoch um strukturfeste Entwürfe. Eine Übertragung von der Entwurfsplattform (z. B. PC) auf eine mobile Plattform (z. B. Mikrocontroller) findet nicht statt. Gerade diese Übertragung ist jedoch ein kritischer Punkt, da eine strukturelle Anpassung und effiziente Quelltextgenerierung eine Implementierung erst ermöglichen. Verfahren zur Quelltextgenerierung von Klassifikatoren sind in der Literatur bekannt [60, 66], eine Übertragung auf die individuelle Merkmalsextraktion, -selektion, -transformation und -klassifikation auf biologische Systeme zur Steuerung von MMIs existiert nicht.

1.2.9 Offene Probleme

Zusammenfassend ergeben sich aus den vorangegangenen Abschnitten folgende offene Probleme, die den Entwurf von MMIs sehr erschweren und oft scheitern lassen:

1. *Entwurf der Mensch-Maschine-Schnittstelle:*

Ein einheitliches Entwurfskonzept für individuelle Mensch-Maschine-Schnittstellen existiert nicht. Kommerzielle Systeme beschränken zwar die Einstellung auf wenige Parameter, aufgrund der festgelegten Struktur ist jedoch keine Individualität des MMIs gegeben. Für patientenindividuelle MMIs hingegen existiert keine einheitliche Entwurfsmethodik, der Entwurfsablauf besteht

aus einer Folge von manuell zu strukturierenden und parametrierenden Arbeitsschritten. Eine einheitliche Umsetzung für beliebige Anwender und Signale besteht nicht.

2. *Auswahl und Optimierung der Signalverarbeitung und -auswertung:*

Die Auswertung biometrischer Signale erfolgt bislang durch heuristische Funktionsansätze, die im Wesentlichen die Vorlieben des Entwicklers widerspiegeln. Anforderungen des Anwenders und der Hardware bleiben im Systementwurf weitgehend unberücksichtigt, zumal ein systematischer Überblick nicht existiert. Ebenso fehlen systematische Analysen geeigneter Auswertelgorithmen sowie der Vergleich und die Bewertung für verschiedene Einsatzszenarien. Alle in der Literatur erwähnten Systeme entwirft und parametriert ein Fachmann, die Implementierung wird stückweise entwickelt und zusammengesetzt. Es existieren keine automatischen Verfahren, die in einem Schritt Daten auswerten, optimale Strukturen für das individuelle MMI entwerfen und dieses rechenineffizient auf einen mikrocontrollertauglichen Quelltext transformieren.

3. *Anforderungsgerechte Anpassung der Signalauswertung:*

Bei der Auswahl der Algorithmen zur Signalauswertung spielen derzeit Plattform- und Qualitätsanforderungen eine untergeordnete Rolle. Insbesondere weisen die eingesetzten Algorithmen Defizite bei der Verarbeitung der Signale und Daten auf. Eine Optimierung der Algorithmen im Sinne dieser Defizite (z. B. bezüglich Rechenzeit, Speicherplatz, Klassifikationsgüte) findet nicht statt.

4. *Grafische Benutzeroberflächen für Entwurf, Simulation, Training und Evaluation:*

Der Entwurf von MMIs erfordert vom Entwickler umfassendes Methodenwissen, das zeitaufwändig erworben, umgesetzt und in seinen Einzelheiten verstanden werden muss. Es fehlen grafische Benutzeroberflächen, die das Verständnis erleichtern und Auswirkungen von Parametrierungen zum Methodenvergleich darstellen. Ebenso beruht das Simulieren von MMIs bislang auf einfachen grafischen Darstellungen, die nur sehr eingeschränkte Möglichkeiten zur Evaluierung von Parametereinstellungen bieten. Selbiges gilt für Trainingsumgebungen, die nur bedingt zur Verfügung stehen.

1.3 Ziele und Aufgaben

Das Ziel dieser Arbeit ist die Entwicklung eines Verfahrens, das den einfachen, zuverlässigen und strukturierten Entwurf eines MMIs ermöglicht. Die Funktionsweise des Verfahrens soll anhand des Entwurfs einer myoelektrischen Steuerung für Handprothesen verifiziert werden, die Übertragbarkeit des Verfahrens auf andere MMIs soll der Entwurf eines Brain-Computer-Interfaces belegen.

Die Entwicklung des Verfahrens erfordert folgende Entwicklungsschritte:

1. Ein Steuerungskonzept für MMIs ist zu entwickeln. Dieses muss universell einsetzbar und an Patient und Problemstellung adaptierbar sein. Ein Baukastensystem dient dann der Umsetzung unterlagerter Routinen.
2. Die Beurteilung der Güte eines MMIs erfordert eine systematische Analyse und Bewertung bekannter Verfahren und Algorithmen. Nötig sind hierfür Kriterien zu deren Quantifizierung, die in einem Gütemaß vereint werden.
3. Besonderheiten in biometrischen Datensätzen führen mit herkömmlichen Verfahren meist nur zu befriedigenden Ergebnissen. Deshalb sind neue Kriterien zu entwickeln, die diesen Ursachen entgegenwirken. Auf der Basis der neuen Kriterien sind Vorgehensweisen, Methoden und Algorithmen zu erstellen, die das Gütemaß verbessern.

4. Um dem Entwickler einen Einblick in den Entwurfsprozess zu verschaffen, ist eine grafische Benutzeroberfläche zu entwickeln, die mögliche Verfahren in sich vereint und Ergebnisse aussagekräftig und interpretierbar darstellt.
5. Zur einfachen Implementierung und zum Nachweis der Funktionalität sind Exportroutinen für das MMI sowie eine Echtzeitumgebung für das MMI und eine Echtzeitumgebung für die Ausführung bereitzustellen.

Mit Hinblick auf eine Handprothesensteuerung ist das Verfahren zu spezifizieren und folgende Arbeitspakete sind zu realisieren:

1. In Abhängigkeit der Möglichkeiten zur Muskelkontraktion bei armamputierten Menschen ist das Steuerungskonzept so zu spezifizieren, dass sich eine einfache multifunktionale Steuerung der Prothese ergibt.
2. Eine komfortable Benutzeroberfläche zur Aufnahme von Daten sowie zur Anpassung und Simulation der Prothese ist zu entwickeln. Insbesondere Patienten, die den Umgang mit ihrer Muskulatur verlernt haben, benötigen motivierendes Training, das ihnen z. B. durch grafische Benutzeroberflächen gegeben werden kann.
3. Anhand der Untersuchung eines Patientenkollektivs ist die Überlegenheit der neuen Kriterien nachzuweisen. Ebenso sind Untersuchungen über Möglichkeiten des Patienten, Auswirkungen von Training und Güte der Klassifikatoren anzustreben.
4. Zum Nachweis der Funktionalität sind Untersuchungen über das Echtzeitverhalten der Algorithmen durchzuführen. Die Ergebnisse geben Auskunft über die Qualität der Steuerung im Alltagsbetrieb.

Kapitel 2 stellt die Umsetzung des Steuerungskonzepts und der unterlagerten Algorithmen vor. Anforderungen an MMIs werden systematisiert und ein Gütemaß zur Quantifizierung möglicher Lösungen eingeführt. Anhand eines Benchmarkbeispiels auf der Basis biometrischer Zeitreihen findet eine Bewertung der vorgestellten Lösungen statt. Kapitel 3 führt Möglichkeiten zur anforderungsgerechten Modifikation ein. Zum Einsatz kommen neue Kriterien für den Entwurfsprozess und neue hierarchische Vorgehensweisen zur Klassifikation. Darauf aufbauend vereint Kapitel 4 alle vorgestellten Methoden in einer grafischen Benutzeroberfläche, die um Werkzeuge zur individuellen, automatischen Quelltextgenerierung erweitert ist. Kapitel 5 führt den vollständigen Entwurf eines MMIs anhand einer myoelektrische Handprothese vor und zeigt dessen Übertragbarkeit anhand des Entwurfs eines Brain-Computer-Interfaces. Ergebnisse werden anhand der Daten von insgesamt 18 Patienten und zwei Probanden diskutiert. Abgeschlossen wird die Arbeit von einer Zusammenfassung und einem Ausblick in Kapitel 6.

2 Neue Entwurfsmethodik für biologische Mensch-Maschine-Schnittstellen

2.1 Neues Konzept

Die willentliche Steuerung eines medizintechnischen Systems mittels eines MMIs setzt dessen patientenindividuellen Entwurf voraus. Hierbei werden die Auswertelgorithmen der Steuerung (oder Regelung) entsprechend den biometrischen Signalen des Patienten angepasst. Der Entwurf geschieht bislang größtenteils manuell, ist sehr zeitaufwändig und beschränkt sich auf die Anpassung von Parametern. Deshalb führt diese Arbeit eine Entwurfsmethodik ein, die eine automatische Anpassung von Parametern und Strukturen der Auswertelgorithmen ermöglicht.

Gemäß Bild 2.1a wird beim manuellen Entwurf von MMIs dem Anwender nach einer Belehrung ein Paradigma zur Steuerung eines medizintechnischen Systems (z. B. Prothese) vorgegeben. Auf das vom Patienten umgesetzte Paradigma wird das vorläufige MMI angewendet, i. A. ohne dabei die zugrunde liegenden Daten zu speichern. Auf der Basis einer subjektiven Bewertung findet eine manuelle Parameteranpassung (z.B. durch einen Medizintechniker) statt und das vorgegebene Paradigma wird verändert. Dieser Vorgang wird wiederholt, bis das Bewertungsmaß eine Mindestgüte des MMIs signalisiert. Das so entworfene MMI kann schließlich noch in einem Langzeitversuch auf seine Funktionsfähigkeit geprüft werden.

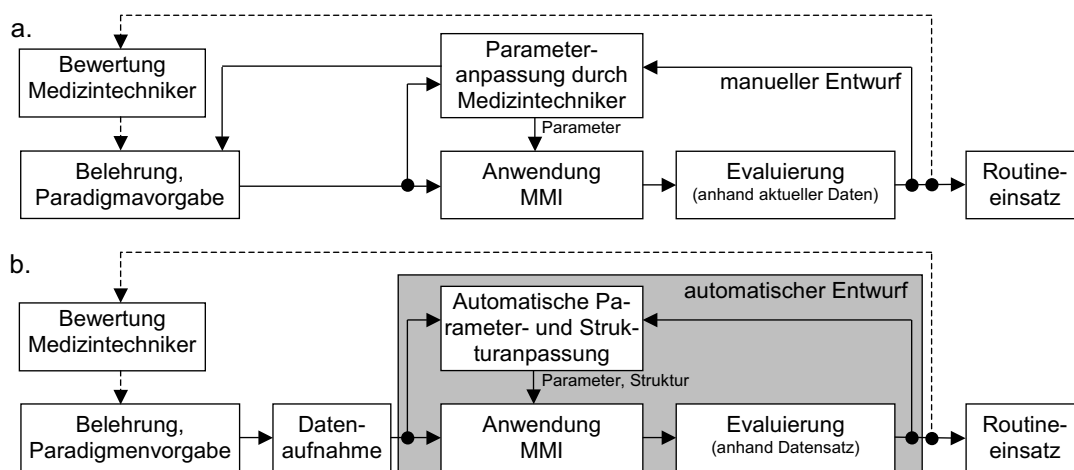


Bild 2.1: Manueller und automatischer Entwurf von biologischen Mensch-Maschine-Schnittstellen

Das Maß zur Bewertung basiert hierbei auf der Güte der Klassenzuweisung bzw. der Güte des aufgenommenen Signals. Eine Beurteilung der Struktur des MMIs ist nicht erforderlich, da dieses strukturfest und

i. A. bereits in die Zielplattform integriert ist. Einen großen Nachteil dieser Entwurfsphilosophie stellt die sequentielle und damit zeitaufwändige Rückkopplung auf die Trainingsparadigmen dar, die die fortwährende Anwesenheit eines Experten erfordert. Weiterhin kann nur eine partielle Anpassung stattfinden, da die Strukturen der Auswertalgorithmen bereits vorgegeben sind.

Der in dieser Arbeit vorgeschlagene automatische Entwurf von MMIs (Bild 2.1b) beginnt ebenfalls mit einer Belehrung. Es folgt die parallele Vorgabe mehrerer möglicher Paradigmen, die in ihrer Komplexität den Paradigmen des manuellen Entwurfs entsprechen. Für alle Paradigmen werden Daten aufgenommen, und äquivalent zum manuellen Entwurf findet die Anwendung des vorläufigen MMIs statt, welches in diesem Fall struktur- und parametervariabel ist. Im Gegensatz zum manuellen Entwurf muss hier die Bewertung automatisch durchgeführt werden. Da die Strukturfreiheit des Entwurfs auch nicht implementierbare Lösungen liefern kann, wird ein Kriterium benötigt, das neben einer hohen Güte der gefundenen Lösung auch deren Echtzeitfähigkeit sicherstellt und die Komplexität der zu verwendenden Strukturen beschränkt. Zwar befinden sich Systeme zum automatischen Entwurf von MMIs im Forschungsstadium (z. B. [75, 177]), diese sind jedoch strukturfest und beschränken sich auf die Anpassung der Parameter. Eine Beurteilung der Implementierbarkeit entfällt, und der Routineeinsatz findet meist nicht statt oder beschränkt sich auf nicht portable Lösungen.

Um einen automatischen Entwurf durchzuführen, ist somit gemäß Bild 2.1b die Art der Anwendung des MMIs zu spezifizieren und ein Bewertungsmaß zu definieren, um so Verfahren unterschiedlicher Struktur zur automatischen Parameteranpassung diskutieren zu können.

Die Anwendung eines MMIs erfordert die Spezifikation der Kommunikation zwischen Mensch und Maschine. Hierbei wird davon ausgegangen, dass das Bedienen eines MMIs durch vordefinierte Aktionen des Anwenders geschieht, die den Zustand (Aktionszustand) des MMIs festlegen. Um eine Kommunikation zwischen Mensch und Maschine auf der Basis solcher Aktionen zu ermöglichen, wird in Abschnitt 2.2 ein einheitlich anwendbarer Zustandsautomat eingeführt, mittels dessen der Anwender durch die Aktionszustände des MMIs navigieren kann. Beschrieben wird unter welchen Voraussetzungen sich dieser Automat vereinfachen lässt und wie sich experimentelle und kommerzielle Systeme in den Automaten einordnen lassen.

Im Gegensatz zu MMIs ohne Lernbedarf, wie z. B. Schaltern, etc., muss das hier zu erstellende MMI Aktionen des Anwenders interpretieren können. Diese lernt es anhand von Beispielen. Hierfür wird ein universeller Ablauf zum systematischen Anlernen und Ausführen von Klassifikationsalgorithmen vorgeschlagen (Abschnitt 2.3).

Auf der Basis des Zustandsautomaten und des einheitlichen Ablaufs zum Anlernen ist das Bewertungsmaß für den automatischen Entwurf zu entwickeln. Es soll den schnellen und zuverlässigen Entwurf des MMIs sicherstellen und dabei die Strukturfreiheit des Entwurfs so einschränken, dass die gefundenen Lösungen in Echtzeitsysteme implementierbar sind. Dem Entscheidungsmaß liegen Kriterien zugrunde, die die notwendigen Eigenschaften des MMIs quantifizieren (Abschnitt 2.4).

Um die Qualität des neuen Konzepts zu belegen, sind Experimente mit einer großen Anzahl an Realisierungen notwendig. Aufgrund der Anstrengung und des Zeitaufwands ist das Generieren dieser Realisierungen einem Patienten jedoch nicht zumutbar. Deshalb schließt sich der Entwurf eines Benchmarkdatensatzes auf der Basis eines künstlichen, biometrisch nachempfundenen Signals an. Dieser Datensatz gibt dann die Möglichkeit, Klassifikationsalgorithmen systematisch zu evaluieren. Betrachtet werden dabei Verfahren für den Entwurf von Merkmalsextraktion, -selektion, -aggregation und für die Entscheidungsfindung (Abschnitt 2.5).

Der Merkmalsextraktion kommt dabei eine entscheidende Bedeutung zu, weil deren Qualität die Ergebnisse der nachfolgenden Verfahren stark beeinflusst. Es wird deshalb eine Routine für eine echtzeitfähige

Merkmalsextraktion vorgeschlagen, die sich für einen weiten Bereich von biometrischen Signalen einsetzen lässt (Abschnitt 2.6).

Mögliche Verfahren für Merkmalsselektion (Abschnitt 2.7), -aggregation (Abschnitt 2.8) und Entscheidungsfindung (Abschnitt 2.9) werden anschließend vorgestellt und hinsichtlich ihrer Güte bezüglich der formulierten Entscheidungshilfen evaluiert.

Zum Abschluss des Kapitels wird der entwickelte Benchmarkdatensatz auf die vorgestellten Verfahren angewandt und eine Bewertung anhand des neuen MMI-Index durchgeführt (Abschnitt 2.10).

Die detaillierte Beschreibung des strukturvariablen Entwurfs verdeutlicht schließlich Kapitel 4.

2.2 Universell anwendbarer Zustandsautomat

Der Signalfluss innerhalb eines MMIs lässt sich als Zustandsautomat mit $K + 1$ Zuständen darstellen - K Zustände, die Aktionen der Maschine zugeordnet sind (Aktionszustände, Klassen - z. B. Bewegungen einer Prothese, Ereignisse auf einem Bildschirm) sowie einem optionalen, neutralen Zustand N ohne zugeordnete Aktionen. Die Transitionen zwischen den Zuständen basieren auf Schaltsignalen, der Übergang von Zustand A nach Zustand B erfolgt durch die Transition $T_{A,B}$ ¹, die durch ein Schaltsignal $\Psi_{A,B}$ ² definiert ist (Bild 2.2).

Ein Schaltsignal bezeichnet ein biometrisches Signal, das willkürlich vom Anwender erzeugt werden kann, um im Zustandsautomaten zu navigieren. Es wird dargestellt durch n_s Zeitreihen, die Sensoren oder Signalverarbeitungsalgorithmen entstammen können.

Die Navigation im Zustandsautomaten kann durch Schaltsignale ohne Zeitversatz oder durch Schaltsignale mit Zeitversatz geschehen. Ziel des Schaltens ohne Zeitversatz ist es, das biometrische Signal bereits während seines Auftretens zu interpretieren und einem Aktionszustand zuzuweisen. Vorteil des Schaltens ohne Zeitversatz ist die schnell reagierende Ansteuerung des Automaten, jedoch sind Auswertelgorithmen meist nicht robust. Beim Schalten mit Zeitversatz besteht das Schaltsignal aus einem abgeschlossenen biometrischen Signal, das erst nach dessen Ende einem Aktionszustand zugewiesen wird. Zwar reagiert das MMI verzögert, es ist aber robuster und leichter erweiterbar.

Das Schalten mit Zeitversatz geschieht durch zeitpunktbasierte oder ereignisbasierte Zustandsübergänge. Erstere interpretieren fortlaufend (zu jedem Abtastzeitpunkt) eingehende Signale als Schaltsignale, werten sie aus und setzen sie in Aktionen der Maschine um. Letztere interpretieren nur abgeschlossene, chronologische Folgen von Eingangssignalen als Schaltsignal und setzen die entsprechende Aktion um. Systeme mit zeitpunktbasierten Zustandsübergängen finden in Brain-Computer-Interfaces [19, 21, 147, 209] Anwendung. Hier wird zu jedem Abtastzeitpunkt eine Entscheidung über den Signalinhalt abgeleitet und als Schaltsignal verwendet. Ähnliche Verfahren existieren in der EMG-Analyse, z. B. bei Herberts [3, 51, 72]. Die Auswertung einer abgeschlossenen, chronologischen Folge von EMG-Eingangssignalen führte Hudgins ein [75], weitere Projekte beschreiben [73, 109, 214].

Oft basieren die beiden Übergänge zwischen zwei Zuständen auf unterschiedlichen Schaltsignalen $\Psi_{A,B} \neq \Psi_{B,A}$. Innerhalb der Zustände befinden sich Signalverarbeitungseinheiten, die u. a. aus weiteren Zustandsautomaten bestehen können. Die zum Einsatz kommenden Signale $\Psi_{A,A}$ dürfen nicht mit Schaltsignalen verwechselt werden (Bild 2.2).

¹In der Literatur wird der Übergang von Zustand A nach Zustand B oft mit entgegengesetzten Indizes dargestellt.

² $\Psi_{A,B}$ ergibt sich aus einer oder mehreren, parallel aufgenommenen Zeitreihen und kann somit Vektor als auch Matrix sein.

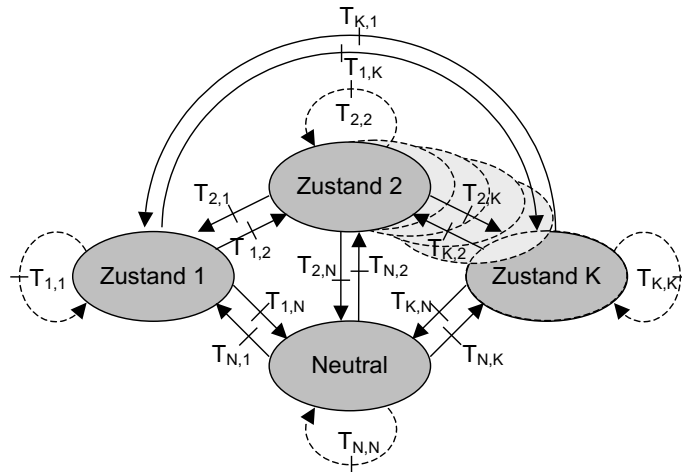


Bild 2.2: Zustandsautomat für ein MMI

Das Generieren der Schaltsignale ist Aufgabe des Anwenders, wobei ihn das MMI durch lernende Algorithmen, Feedback über den aktuellen Zustand, etc. unterstützt. Wichtig ist es, den Anwender nicht mit der Fülle von Aktionsmöglichkeiten zu überlasten, sondern die Auswahl so zu beschränken, dass einerseits ein komfortables Ansteuern gewährleistet, andererseits Mindestanforderungen an die Robustheit erfüllt sind.

Beispielsweise beschränkt das Entfernen aller Transitionen $T_{A,B}$; $A \neq N, B \neq N$ zwar die Möglichkeit, zwischen den Aktionszuständen direkt zu wechseln, die Anzahl zu beherrschender Schaltsignale vermindert sich jedoch von $K(K + 1)$ auf $2K$ (pro Zustand 2 Schaltsignale, Bild 2.3). Eine weitere Vereinfachung eröffnet sich durch ein einheitliches Schaltsignal $\Psi_{j,N} = \Psi_N$, um aus einem beliebigen Zustand $j \neq N$ in den neutralen Zustand zurückzukehren. Die Anzahl notwendiger Schaltsignale vermindert sich auf $K + 1$ (pro Zustand ein Schaltsignal + ein einheitliches Schaltsignal zum Zurückschalten). Durch das Generieren der Schaltsignale Ψ_N und $\Psi_{N,i}$ hintereinander kann jedoch die gleiche Funktionalität des Automaten aus Bild 2.2 erreicht werden.

Sollen nur wenige Schaltsignale ($K \leq 3$) umgesetzt werden, dann empfiehlt sich das Beibehalten der Transitionen zwischen den Aktionszuständen und das Entfernen des neutralen Zustandes, um schneller und einfacher zu navigieren. Insbesondere die Umsetzung von nur zwei Zuständen führt oft zu Schaltsignalen $\Psi_{1,2} = \Psi_{2,1}$.

Gemäß dieser Kategorisierung lässt sich das MMI der Otto Bock-Handprothese zur Auswertung von EMG-Signalen als System mit zwei Zuständen (Hand öffnen/schließen, Handgelenk drehen) einordnen. Der Zustand wird durch ein einheitliches Schaltsignal ohne Zeitversatz ereignisgesteuert umgeschaltet (Kokontraktion). Die Transitionen $T_{A,A}$ beinhalten weitere unterlagerte Zustandsautomaten, die das Ausführen der Bewegung (Öffnen und Schließen) gewährleisten.

Im Folgenden betrachtet diese Arbeit den Entwurf von MMIs mit ereignisgesteuerten Zustandsübergängen, da deren Umsetzung wesentlich weniger Rechenleistung erfordert. Im Zustandsautomat wird auf Transitionen zwischen den Aktionszuständen verzichtet (Bild 2.3). Der Übergang von jedem Aktionszustand j in den neutralen Zustand soll durch eine einheitliche Transition $T_{j,N} = T_N$ bzw. ein einheitliches Schaltsignal Ψ_N geschehen.

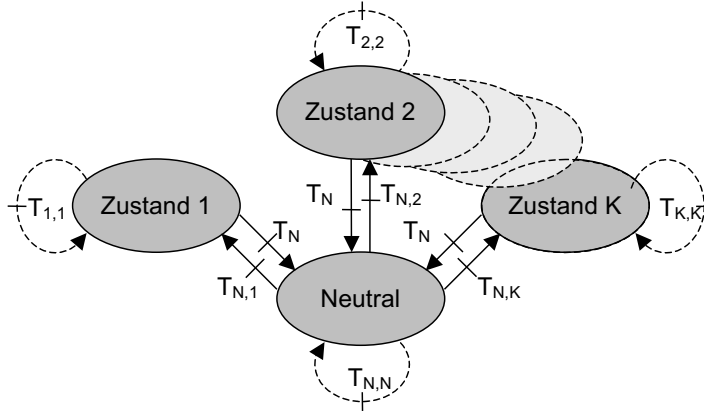


Bild 2.3: Eingeschränkter Zustandsautomat für ein MMI

2.3 Ablauf zur Klassifikation von Schaltsignalen

Realisierungen der vom Anwender generierten Schaltsignale sind oftmals nicht identisch. Ein System, das diese Schaltsignale auswerten soll, muss somit trotz Abweichungen von dem als optimal angesehenen Schaltsignal die richtige Entscheidung treffen, d. h. das Schaltsignal der richtigen Schaltsignalklasse zuordnen. Ein solches System soll fortan als Klassifikationssystem oder auch Klassifikator bezeichnet werden. Der Klassifikator soll nur die im neutralen Zustand auftretenden K Schaltsignale $\Psi_{N,j}, j \in \{1 \dots K\}$ klassifizieren, da sich das in den neutralen Zustand zurückführende Schaltsignal Ψ_N oft durch einfache Berechnungen erkennen lässt.

Es ist somit die Aktionsabsicht \hat{y} des Anwenders aus einem generierten Schaltsignal Ψ zu interpretieren. Das Schaltsignal setzt sich zusammen aus n_s Zeitreihen $u_j[k], j = 1, \dots, n_s$ mit jeweils $s_{u,j}$ Abtastwerten, es ergeben sich die Rohdaten u mit insgesamt s_u Abtastwerten:

$$S: \Psi = (u_1, \dots, u_{s_u})^T \mapsto \hat{y} \in \{1, \dots, K\} \quad (2.1)$$

$$\text{mit } s_u = \sum_{j=1}^{n_s} s_{u,j}, \quad (u_1, \dots, u_{s_u})^T = (u_1[1], \dots, u_1[s_{u,1}], u_2[1], \dots, u_{n_s}[s_{u,n_s}]), \quad K \in \mathbb{N}^+.$$

Der Ablauf der Klassifikation setzt sich nach Bild 2.4 aus den Teilschritten Merkmalsextraktion S_1 , Merkmalsauswahl S_2 , Merkmalsaggregation S_3 und der Entscheidungsfindung mittels eines Entscheidungsmaßes S_4 zusammen. Diese schrittweise Strategie hat sich bewährt, da es überaus schwer ist, direkt einen Klassifikator zu ermitteln. Suboptimalität durch separaten Entwurf der Teilschritte wird dabei in Kauf genommen.

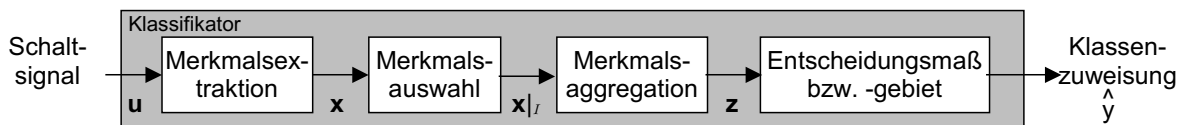


Bild 2.4: Schema zur Klassifikation von Schaltsignalen

Die Merkmalsextraktion dient der Ableitung bedeutungstragender Merkmale x_i aus den Rohdaten \mathbf{u} . Es ergibt sich eine Transformation

$$S_1 : \Psi = (u_1, \dots, u_{s_u})^T \mapsto \mathbf{x} = (x_1, \dots, x_s)^T, \quad \mathbf{x} \in \mathbb{R}^s. \quad (2.2)$$

Der entstehende Merkmalsraum ist häufig redundant und enthält u. a. Merkmale, die keine für die Klassifikation taugliche Information enthalten. Dieser Problematik nimmt sich die Merkmalsauswahl an. Sie sucht aus den extrahierten s Merkmalen die s_m aussagekräftigsten gemäß einem Kriterium heraus. Eine Übersicht ist in [2, 113] gegeben. Mit der Bezeichnung \mathcal{I} für die Indexmenge der ausgewählten Merkmale ergibt sich die Abbildung

$$S_2 : \mathbf{x} = (x_1, \dots, x_s)^T \mapsto \mathbf{x}|_{\mathcal{I}} = (x_{i_1}, \dots, x_{i_{s_m}})^T, \quad s \geq s_m. \quad (2.3)$$

Selbst in dem Fall, dass die ausgewählten Merkmale für eine Klassifikation geeignet sind, muss mit diesen der Entwurf einer Entscheidungsregel nicht zu einem guten Klassifikator führen, wenn der hohen Dimension des Merkmalsraums eine geringe Anzahl von Lerndaten gegenübersteht. Eine zuverlässige Parametrierung gelingt dann nicht und macht eine Merkmalsaggregation erforderlich. Das Ziel der Merkmalsaggregation besteht darin, eine Transformation vom s_m -dimensionalen Merkmalsraum in einen s_d -dimensionalen Raum, $s_m \geq s_d$, mit einem geringen Verlust an Diskriminanzinformation zu finden:

$$S_3 : \mathbf{x}|_{\mathcal{I}} = (x_{i_1}, \dots, x_{i_{s_m}})^T \mapsto \mathbf{z} = (z_1, \dots, z_{s_d})^T, \quad s_m \geq s_d, \quad \mathbf{z} \in \mathbb{R}^{s_d}. \quad (2.4)$$

Eine Übersicht über Merkmalsaggregationen ist [77] zu entnehmen. Ohne Beschränkung der Allgemeinheit findet die Aggregation durch eine lineare Transformation statt. Sind nichtlineare Transformationen erwünscht, so können diese über die Merkmalsextraktion miteinbezogen werden. Eine Transformationsmatrix \mathbf{A} dient dann zur Aggregation der Merkmale [195]:

$$\mathbf{z} = \mathbf{A}^T \cdot \mathbf{x}|_{\mathcal{I}}, \quad \mathbf{A} \in \mathbb{R}^{s_m \times s_d}. \quad (2.5)$$

Die Entscheidungsregel weist schließlich den so entstandenen Merkmalsatz einer Klasse y zu, dieser Schätzwert soll als \hat{y} bezeichnet werden:

$$S_4 : \mathbf{z} = (z_1, \dots, z_{s_d})^T \mapsto \hat{y} \in \{1, \dots, K\}, \quad K \in \mathbb{N}^+. \quad (2.6)$$

Die Konstruktion der Entscheidungsregel kann nach unterschiedlichen Prinzipien erfolgen, unabhängig davon, ob zuvor eine Merkmalsauswahl und/oder -aggregation vorgenommen wurde. Die Entscheidungsregel bildet gewissermaßen die finale Abbildung, die letztlich die Klassenzuordnung bewerkstelligt. Sie ist somit selbst ein Klassifikator, aber gleichzeitig auch Teil eines Klassifikators. Eine Übersicht geben [186, 200].

Neben der hier vorgestellten vierstufigen Vorgehensweise fassen viele Klassifikatorentwurfsverfahren die Merkmalsaggregation und die Konstruktion der Entscheidungsregel in einem Schritt zusammen. Andere verzichten auf die Merkmalsauswahl.

Idealerweise ist ein Klassifikator individuell an den Patienten angepasst. Diese Anpassung lässt sich zwar manuell durchführen, ist jedoch mit hohem Zeitaufwand verbunden und meist nur suboptimal. Um eine solche Anpassung automatisch durchzuführen, können Parameter auf der Basis von Beispielschaltsignalen automatisch geschätzt werden. Im vorliegenden Fall ist hierfür ein Verbund von $n = \sum_{j=1}^K n_j$ Beispielschaltsignalen pro Klasse $j \in \{1, \dots, K\}$ notwendig.

Es werden folgende Bezeichner eingeführt:

$$n_j \quad \text{Anzahl an Datentupeln pro Klasse } j, \quad (2.7)$$

$$n = \sum_{j=1}^K n_j \quad \text{Anzahl aller Datentupel,} \quad (2.8)$$

$$\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T \quad \text{Matrix aller Merkmale,} \quad (2.9)$$

$$\mathbf{y} = (y_1, \dots, y_n)^T; y_i \in \{1, \dots, K\} \quad \text{Vektor der Klassenzugehörigkeiten und} \quad (2.10)$$

$$\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_n)^T \quad \text{Matrix der aggregierten Merkmale.} \quad (2.11)$$

Der Entwurf der Merkmalsextraktion wird i. A. einmalig manuell durchgeführt und nicht an den individuellen Patienten angepasst. Hingegen finden der Entwurf von Merkmalsauswahl, -aggregation und Entscheidungsregel automatisch und individuell statt.

Unter Vorgabe der richtigen Klassenzuordnungen \mathbf{y} muss die Indexmenge \mathcal{I} der zu selektierenden Merkmale gefunden werden:

$$S_{2^*} : \mathbf{X}, \mathbf{y} \mapsto \mathcal{I}. \quad (2.12)$$

Die Abbildung S_{2^*} wird fortan als Entwurf der Merkmalsauswahl bezeichnet.

Der Datensatz mit ausgewählten Merkmalen $\mathbf{X}|_{\mathcal{I}}$ kann in Verbindung mit den Klassenzuordnungen \mathbf{y} zur Suche nach einer Aggregations- oder Transformationsmatrix \mathbf{A} verwendet werden:

$$S_{3^*} : \mathbf{X}|_{\mathcal{I}}, \mathbf{y} \mapsto \mathbf{A}, \quad \mathbf{A} \in \mathbb{R}^{s_m \times s_d}. \quad (2.13)$$

Die Abbildung S_{3^*} wird fortan als Entwurf der Aggregationsvorschrift bezeichnet.

Ausgehend vom Datensatz der aggregierten Merkmale nach (2.5) ist für jede Klasse j der Parametersatz θ_j zu finden, mittels dessen eine richtige Klassenzuordnung gelingt:

$$S_{4^*} : \mathbf{Z}, \mathbf{y} \mapsto \theta_1, \dots, \theta_K. \quad (2.14)$$

Die Abbildung S_{4^*} soll als Entwurf der Entscheidungsregel bezeichnet werden.

Ein Unterscheidungsmerkmal für die Entwurfsverfahren stellen die dem Entwurf zu Grunde liegenden mathematischen Modelle dar [121], vgl. Bild 2.5.

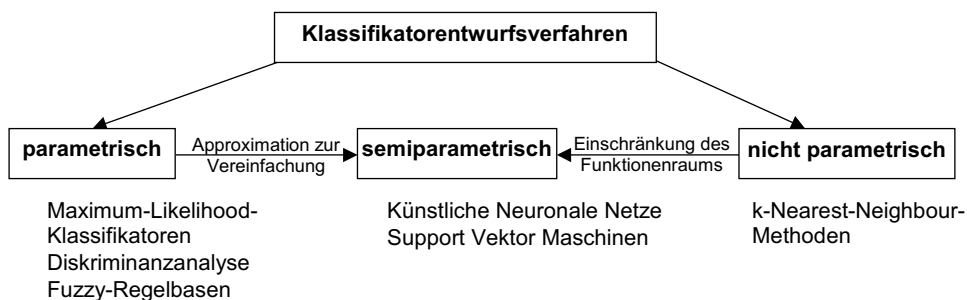


Bild 2.5: Klassifikatorentwurfsverfahren

Parametrisch werden Modelle genannt, wenn sie einen funktionalen Zusammenhang durch einen Formel-ausdruck beschreiben, in dem eine endliche Anzahl an Parametern vorkommt. Üblicherweise wird dabei die Struktur des Zusammenhangs als bekannt angenommen (als wahr postuliert), weshalb die Parameter

gemeinhin interpretierbar sind. Falls aus Gründen einer einfacheren mathematischen Behandlung statt des parametrischen Modells eine Approximation desselben benutzt werden muss, sind die Parameter des approximierenden Systems nicht mehr interpretierbar, sie werden als Semiparameter bezeichnet.

Nichtparametrisch werden Modelle genannt, wenn sie einen funktionalen Zusammenhang durch eine unendliche Menge (z. B. Funktionsgraphen), eine endliche Menge (Menge, die durch Aufzählungen charakterisiert wird) oder eine Menge von Kennwerten (z. B. Werte von Funktionalen) beschreiben. Da das Arbeiten mit unendlichen und großen endlichen Mengen schwer fällt, wird durch den Übergang auf eine endliche parametrische Approximation ausgewichen. Das nichtparametrische Modell wird dadurch zu einem semiparametrischen Modell.

Die Verwendung eines Klassifikatorentwurfsverfahrens erfolgt in Abhängigkeit von der Problemstellung. Es existieren Arbeiten, die die Merkmalsextraktion bewerten und Empfehlungen für zu extrahierende Merkmale im Sinne hoher Klassifikationsgüten aussprechen [26, 51]. Eine Bewertung des gesamten Ablaufs bestehend aus Merkmalsextraktion, -auswahl, -aggregation und Entscheidungsregel bezüglich Kriterien, die für ein Echtzeitsystem von Bedeutung sind, existiert nicht.

Es werden folgende Kriterien als Entscheidungshilfe definiert:

1. Oberste Priorität für ein MMI hat die *Onlinefähigkeit/Echtzeitfähigkeit*. Nach DIN 44300³ ist Echtzeitfähigkeit gewährleistet, wenn das System ständig betriebsbereit ist, um anfallende Daten innerhalb einer vorgegebenen Zeitspanne zu verarbeiten. Die Echtzeitfähigkeit eines Systems hängt vom Rechenaufwand zum Zuweisen einer Entscheidung (Merkmalsextraktion, -auswahl, -aggregation, Auswerten der Entscheidungsregel) ab. Sie ist systemabhängig und wird in dieser Arbeit auf kostengünstige Mikrocontrollersysteme bezogen. Ist ein Verfahren nicht in Echtzeit implementierbar, so kann es nicht verwendet werden. Zusätzlich zur Echtzeitfähigkeit ist die Anzahl der notwendigen Speicherplätze für die Onlinefähigkeit eines Verfahrens wichtig, da Speicherplatz i. A. nur beschränkt zur Verfügung steht.
2. Die *Klassifikationsgüte* gibt an, ob das Verfahren das vorgegebene Lernziel (z. B. Optimierungskriterium) erfüllt und ob es in der Lage ist, über dem Lerndatensatz zu generalisieren und unbekannte Datensätze richtig zuzuweisen.
3. Die *Interpretierbarkeit* gibt Aufschluss, inwiefern die Parameter des Verfahrens zur Beschreibung des Problems genutzt werden können. Dies ist insbesondere dann wichtig, wenn Wirkungsmechanismen aufzudecken sind.
4. Die *Adaptionmöglichkeiten* ermöglichen die Adaption der Parameter des Algorithmus in Echtzeit, um dem System auch die erfolgreiche Auswertung zeitvarianter Schaltsignale zu ermöglichen.
5. Die *Robustheit* des Verfahrens bezeichnet seine Fähigkeit, auf der Basis eines realistischen Lerndatensatzes (wenige Datentupel im Lerndatensatz, Ausreißer, Redundanzen, Störungen, etc.) über unbekanntes Daten gute Ergebnisse zu liefern. Insbesondere biometrische Zeitreihen zeichnen sich durch systematische Änderungen in verschiedenen Aufnahmesitzungen aus. Da dies vielen Verfahren zum Entwurf der Entscheidungsregel Probleme bereitet, sind für solche Fälle besonders robuste Merkmale zu finden.
6. Der *Lernaufwand* bezeichnet die Komplexität des Lernverfahrens sowie die Geschwindigkeit des Anlernvorgangs (Parameterbestimmung, Validierung, etc.).
7. Der *Trainingsaufwand* für den Patienten erfasst die notwendige Anzahl an Beispielsignalen, die ein Patient zum erfolgreichen Anlernen des Klassifikators generieren muss. Somit beeinflusst diese Anzahl wiederum die Kriterien Klassifikationsgüte und Robustheit. Insbesondere beim Entwurf

³Die DIN 44300 wurde mittlerweile zurückgezogen, eine neue Fassung ist derzeit in Bearbeitung.

von MMIs ist darauf zu achten, dass wenige Sitzungen für das Aufnehmen der Beispielschaltssignale benötigt werden.

2.4 Neues Bewertungsmaß zur Beurteilung von Mensch-Maschine-Schnittstellen

Für den Entwurf eines Klassifikators sind die vorgeschlagenen Kriterien anwendungsspezifisch zu bewerten. Insbesondere für den Entwurf eines MMIs ist die Interpretierbarkeit eines Klassifikators aus der Sicht des Anwenders nicht von Belang. Adaptionsmöglichkeiten und Robustheit der Algorithmen sind schlecht im Vorfeld evaluierbar. Deshalb wird in dieser Arbeit ein neues Bewertungsmaß vorgeschlagen, das die Teilkriterien mit Hinblick auf die Patientenzufriedenheit beurteilt. Es wird als MMI-Index $\Lambda \in [0, 1]$ bezeichnet und liefert große Werte für eine hohe voraussichtliche Patientenzufriedenheit. Ausgangspunkt ist ein Anwender, der die Qualität seiner Steuerung nach der Güte der Auswertalgorithmen und nach dem Preis für das Gesamtsystem bewertet. Beide Kategorien enthalten Kriterien $Q \in [0, 1]$, aus denen sich der MMI-Index dann zusammensetzt. Die Güte der Auswertalgorithmen lässt sich durch Kriterien für

- Güte der Klassenzuweisung (Q_G),
- Anzahl umsetzbarer Zustände (Schaltssignalkategorien) (Q_K) und
- Trainingsaufwand für den Patienten (Q_T),

darstellen. Einfluss auf den Preis für das Gesamtsystem haben

- die Engineering-Kosten (Q_L), die sich aus dem Aufwand zum Anlernen des Klassifikators ergeben und
- die Mikrocontroller-Kategorie (Q_C), die sich aus notwendigem Arbeitsspeicher und Rechenleistung ableitet.

Unter der Voraussetzung eines konstanten Trainingsaufwands für den Patienten ($Q_T = \text{konst.}$) ergibt sich der MMI-Index zu

$$\Lambda = Q_G \cdot Q_K \cdot Q_L \cdot Q_C. \quad (2.15)$$

Zwar ergeben sich durch die multiplikative Verknüpfung u. U. sehr kleine Werte, doch so können die einzelnen Kriterien bei Unterschreiten der Mindestanforderung ($Q = 0$) den MMI-Index zu 0 setzen, um die gefundene Lösung zu verwerfen.

Um die Kriterien Q zu quantifizieren, sind sie mit einer Bewertungsfunktion zu hinterlegen, die i. A. nichtlinear ist. Die Auswahl einer geeigneten Bewertungsfunktion geschieht in dieser Arbeit durch die Vorgabe qualitativer Eigenschaften, die anschließend mit heuristischen Funktionsansätzen erfüllt werden. Eine Parametrierung dient dann der Anpassung an ein gegebenes Problem.

Das Kriterium "Güte der Klassenzuweisung" Q_G ergibt sich aus der Klassifikationsgüte ξ_G in % durch

$$Q_G(\xi_G) = \begin{cases} \left(\frac{\xi_G - c_G}{100 - c_G} \right)^4 & \text{wenn } \xi_G \geq c_G \text{ mit } 0 \leq c_G < 100 \\ 0 & \text{sonst.} \end{cases} \quad (2.16)$$

Demnach ist ein Klassifikator zu verwerfen, wenn eine Mindestklassifikationsgüte c_G unterschritten wird. Der Funktionsansatz wurde so gewählt, dass eine Verbesserung der Klassifikationsgüte einen überproportionalen Einfluss auf Q_G hat. Klassifikationsgüten $\xi_G > 95\%$ sind besonders erstrebenswert, was

durch den Wert 4 des Exponenten modelliert wird, Klassifikationsgüten $\xi_G < 80\%$ sind zu verwerfen. Mit $c_G = 80$ ergibt sich Bild 2.6a.

Das Kriterium "Anzahl umsetzbarer Zustände" Q_K berechnet sich aus der Klassenanzahl K durch

$$Q_K(K) = \frac{K - 1}{c_K + K - 1}, \quad c_K > 0, \quad K \geq 1. \quad (2.17)$$

Das Kriterium steigt monoton an, je mehr Schaltsignale unterschieden werden können, und nähert sich der Asymptote $Q_K = 1$. Der Einfluss einer Erhöhung der Klassenanzahl auf Q_K fällt hierbei mit steigender Klassenanzahl, da meist nicht so viele Klassen benötigt werden. Der Parameter c_K bestimmt die Steigung der Kurve im Punkt (1,0). Für $c_K = 1$ ergibt sich Bild 2.6b.

Beim Kriterium "Engineering-Kosten" Q_L werden zwei Funktionen $Q_{L,1}$ und $Q_{L,2}$ modelliert, die von der Anlerndauer des Klassifikators ξ_L (in Sekunden) abhängen. Einerseits soll der Entwurf in der gleichen Sitzung stattfinden können, in der Beispielsignale aufgenommen werden (beschrieben durch $Q_{L,1}$), andererseits ist ein sehr schneller Entwurf von Vorteil, um Parameter des Klassifikators anhand der Klassifikationsgüten numerisch zu bestimmen (beschrieben durch $Q_{L,2}$). Kann Letzteres nicht gewährleistet werden, so ist nur $Q_{L,1}$ zu betrachten. Das Kriterium ergibt sich dann zu

$$Q_L(\xi_L) = \max(Q_{L,1}, Q_{L,2}) = \max\left(\frac{c_{L,1}}{c_{L,2}\xi_L + 1}, \frac{1}{c_{L,0}\xi_L + 1}\right) \quad (2.18)$$

mit $0 < c_{L,1} < 1$ und $0 < c_{L,2} < c_{L,0}$.

Beide Kurven fallen hyperbolisch ab, da kleine Anlerndauern besonders erstrebenswert sind. Die Parameter $c_{L,0}$ und $c_{L,2}$ beschreiben die Streckung der Kurven in x-Richtung, $c_{L,1}$ entspricht der Ordinate von $Q_{L,1}$ für $\xi_L = 0$. Die Parameter sind nun so zu wählen, dass

- geringe Anlernzeiten besonders bevorzugt werden,
- die Geschwindigkeit des Klassifikatorentwurfs bei sehr geringen Anlernzeiten sehr großen Einfluss hat und
- bei einer Anlerndauer von $\xi_L = 0$ $Q_{L,2}$ mindestens dreimal größer ist als $Q_{L,1}$.

Zweckmäßige Parameter sind $c_{L,0} = 2$, $c_{L,1} = 0.3$, $c_{L,2} = 0.01$ (Bild 2.6c).

Das Kriterium "Mikrocontroller-Kategorie" Q_C wird modelliert aus einem Kriterium für die Anzahl benötigter Speicherplätze Q_{P,i_M} und einem Kriterium für die Anzahl notwendiger Multiplikationen zur Herbeiführung einer Klassifikationsentscheidung Q_{M,i_M} :

$$Q_C(\xi_M, \xi_P) = Q_{P,i_M}(\xi_P) \cdot Q_{M,i_M}(\xi_M). \quad (2.19)$$

Der Parameter i_M beschreibt den gewählten Mikrocontroller. Stehen mehrere Controller zur Verfügung, so ergibt sich eine Kurvenschar mit einem Scharparameter $0 < i \leq 1$, deren Verlauf abhängig von dem gewählten Controller ist. Jedes i spezifiziert einen möglichen Controller, der preiswerteste Controller ergibt sich für $i = 1$. Je teurer ein Modell ist, desto kleiner wird i . Es ist dann derjenige Controller mit i_M zu wählen, für den $Q_{P,i}(\xi_P) \cdot Q_{M,i}(\xi_M)$ maximal wird:

$$i_M = \max_i(Q_{P,i}(\xi_P) \cdot Q_{M,i}(\xi_M)). \quad (2.20)$$

$Q_{P,i}$ bzw. $Q_{M,i}$ bewerten die Anzahl an Speicherplätzen ξ_P und Multiplikationen ξ_M danach, wieviel Rechenleistung und Speicherplatz auf dem Controller zur Verfügung stehen. In Bezug auf die Rechenleistung ist es erstrebenswert, dass alle Operationen innerhalb eines Abtastzyklus durchgeführt werden

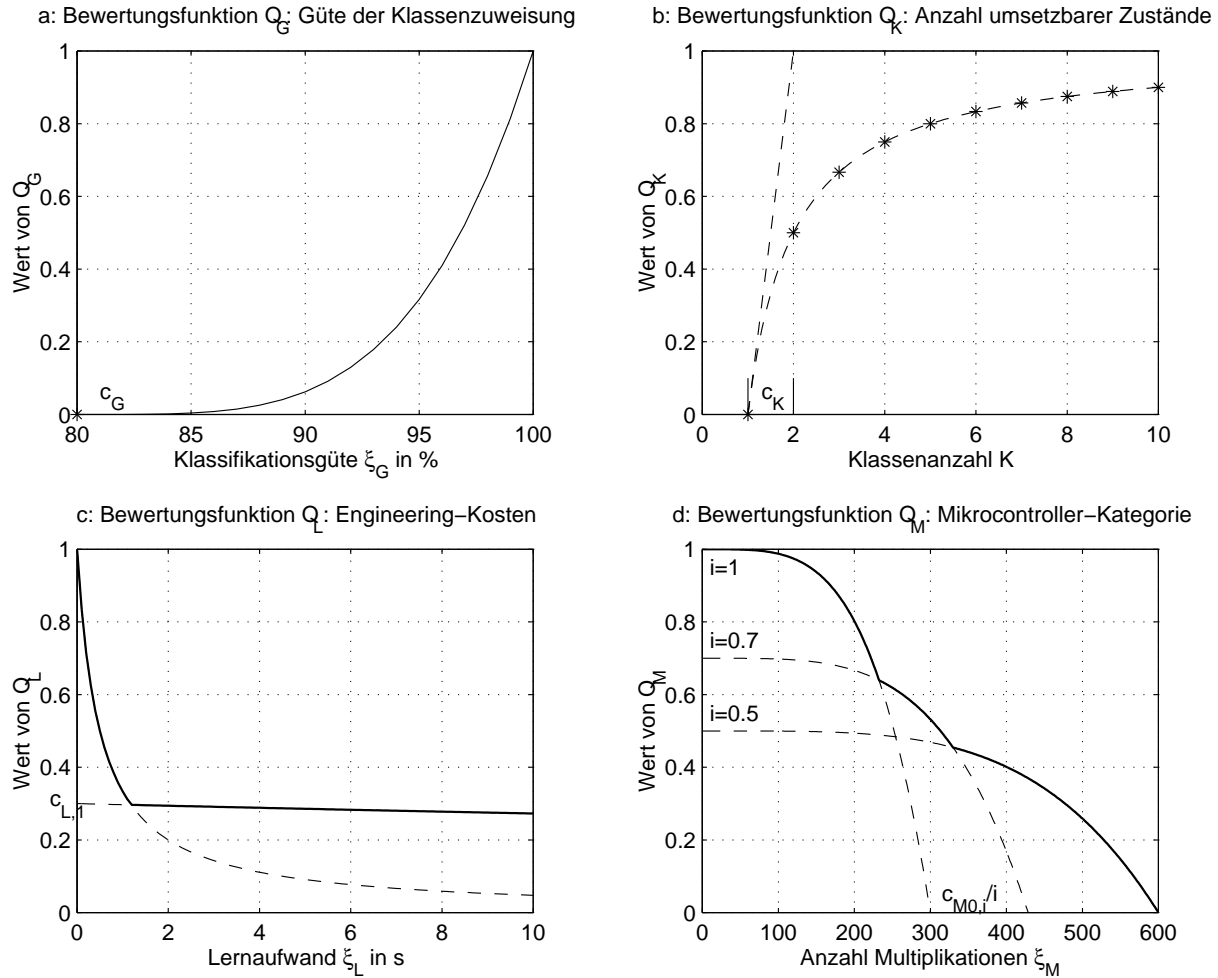


Bild 2.6: Bewertungsfunktionen zur Berechnung des MMI-Index, a: Kriterium "Güte der Klassenzuweisung" $Q_G(\xi_G)$, b: Kriterium "Anzahl umsetzbarer Zustände" $Q_K(K)$, c: Kriterium "Engineering-Kosten" $Q_L(\xi_L)$, d: Kriterium "Anzahl benötigter Multiplikationen" $Q_{M,i}(\xi_M)$ mit $i = \{1, 0.7, 0.5\}$

können. Diese Arbeit beurteilt die Rechenleistung anhand der Anzahl an Multiplikationen, die innerhalb dieser Zeit möglich sind. Additionen spielen hier eine untergeordnete Rolle, der Aufwand zur Auswertung der Exponentialfunktion entspricht in etwa dem Aufwand zur Auswertung von 20 Multiplikationen.

Ist der Controller kaum ausgelastet, so führt eine Erhöhung der Speicherplatz- oder Multiplikationsanzahl kaum zu Einbußen in der Leistungsfähigkeit des Controllers - die Kurve fällt nur flach ab. In Bereichen für ξ_P bzw. ξ_M , in denen der Controller an seine Kapazitätsgrenzen gelangt, fällt die Kurve steil ab, da deren Änderung starke Auswirkungen auf die verbleibenden Ressourcen hat. Das modellierte Kriterium ergibt sich zu

$$Q_{P,i}(\xi_P) = \begin{cases} i \left(1 - \left(\frac{i \cdot \xi_P}{c_{P0,i}} \right)^{c_{P1,i}} \right) & \text{für } i \cdot \xi_P < c_{P0,i} \\ 0 & \text{sonst} \end{cases} \quad \text{mit } c_{P0,i} > 0 \text{ und } c_{P1,i} \geq 1, \quad (2.21)$$

$$Q_{M,i}(\xi_M) = \begin{cases} i \left(1 - \left(\frac{i \cdot \xi_M}{c_{M0,i}}\right)^{c_{M1,i}}\right) & \text{für } i \cdot \xi_M < c_{M0,i} \\ 0 & \text{sonst} \end{cases} \quad \text{mit } c_{M0,i} > 0 \quad \text{und} \quad c_{M1,i} \geq 1. \quad (2.22)$$

Der Startwert von $Q_{P,i}$ bzw. $Q_{M,i}$ für $\xi_P = \xi_M = 0$ ergibt sich zu

$$Q_{P,i}(0) = Q_{M,i}(0) = i. \quad (2.23)$$

Der Schnitt mit der Abszisse errechnet sich aus $c_{P0,i}/i$ bzw. $c_{M0,i}/i$. Der Parameter $c_{P1,i}$ bzw. $c_{M1,i}$ beschreibt die Krümmung der Kurve. Für leistungsfähige, preiswerte Mikrocontroller ($i = 1$) gilt etwa $c_{M0,i} = c_{M0} = 300$ und $c_{M1,i} = c_{M1} = 1$. Bild 2.6d stellt eine Kurvenschar für $Q_{M,i}$ dar. Unter der Voraussetzung von $Q_{P,i} = Q_P = 1$ zeigt die durchgezogene Linie den Wert von Q_C nach (2.20).

Es existieren natürlich auch MMIs, für die eine Umsetzung auf Desktop-PCs akzeptabel ist (z. B. Brain-Computer-Interfaces). Die Parameter $c_{M0,i}$, $c_{P0,i}$ sind dann zwei bis vier Größenordnungen höher anzusetzen.

Alle Parameter sind frei wählbar und anwendungsspezifisch an den Typ des zu entwickelnden MMIs anzupassen.

2.5 Entwurf eines Benchmarkdatensatzes

Zur Evaluierung und Parametrierung möglicher Entwurfsverfahren S_{1^*} bis S_{4^*} wird in diesem Abschnitt die Entwicklung eines Benchmarkdatensatzes vorgeschlagen. So werden Großzahlexperimente möglich, die eine systematische Analyse der Auswirkungen von Variationen in den Merkmalen und der Anzahl notwendiger Lerndaten zulassen. Der Benchmark synthetisiert Schaltsignale zum Schalten mit Zeitversatz auf der Basis von Aktivitätssignalen, die einem biometrischen Signal nachempfunden wurden⁴.

Ein Aktivitätssignal bezeichnet ein biometrisches Signal, dessen Amplitude die Aktivität des abgetasteten biologischen Systems widerspiegelt. Die zugrundeliegenden Werte sind stets positiv. Die Form des Zeit-Amplituden-Verlaufs gibt Auskunft über das Schaltsignal.

Sofern das zu untersuchende Signal nicht selbst ein Aktivitätssignal ist, lässt es sich in ein solches konvertieren. Liegt der Informationsinhalt des Signals beispielsweise in den auftretenden Frequenzanteilen, so können diese Anteile (z. B. durch Bandpässe, autoregressive Modelle, Waveletanalysen, etc.) extrahiert, gleichgerichtet, tiefpassgefiltert und evtl. (z. B. durch 50Hz Sperrfilter) entstört werden.

Aus einem einzelnen biometrischen Signal lassen sich u. U. mehrere Aktivitätssignale extrahieren (nähere Erläuterungen finden sich in Abschnitt 2.6).

Die modellierten Schaltsignale bestehen aus Aktivitätssignalen auf zwei Kanälen, die z. B. der Aktivität von zwei Sensoren oder zwei Frequenzbereichen entspringen können. Sie setzen sich zusammen aus einer oder zwei aufeinander folgenden Aktivierungen, die in Dauer und/oder Amplitude variieren.

Folgende physiologische Besonderheiten wurden integriert:

1. Aufgrund der Verzögerung durch eine Tiefpassfilterung sinkt die Amplitude einer Aktivierung, wenn deren Dauer eine Schwelle unterschreitet.

⁴Den Algorithmus zur Modellierung zeigt Anhang 7.1.1.

2. Ein starkes Aktivitätssignal führt oft zu einem Vollausschlag der zugehörigen Sensorspannung. Entsprechende Realisierungen differieren nur wenig in ihrer maximalen Amplitude. Hingegen sind schwache Aktivierungen für den Anwender schwieriger zu dosieren. Realisierungen mit schwacher Aktivierung werden deshalb mit größeren Schwankungen ihrer maximalen Amplitude beaufschlagt.
3. Wünscht der Anwender das Schaltsignal ausschließlich durch Aktivität auf einem Kanal zu geben, so ist es ihm i. A. nicht möglich, Aktivität auf dem anderen Kanal vollständig zu unterdrücken. Das Benchmarkbeispiel modelliert deshalb immer zumindest eine schwache Aktivität auf dem anderen Kanal.

Die Vorgehensweise des Entwurfs, notwendige Parameter sowie eine Visualisierung der entworfenen Schaltsignale stellt Anhang 7.1.1 dar, ein repräsentatives Beispiel zeigt Bild 2.7.

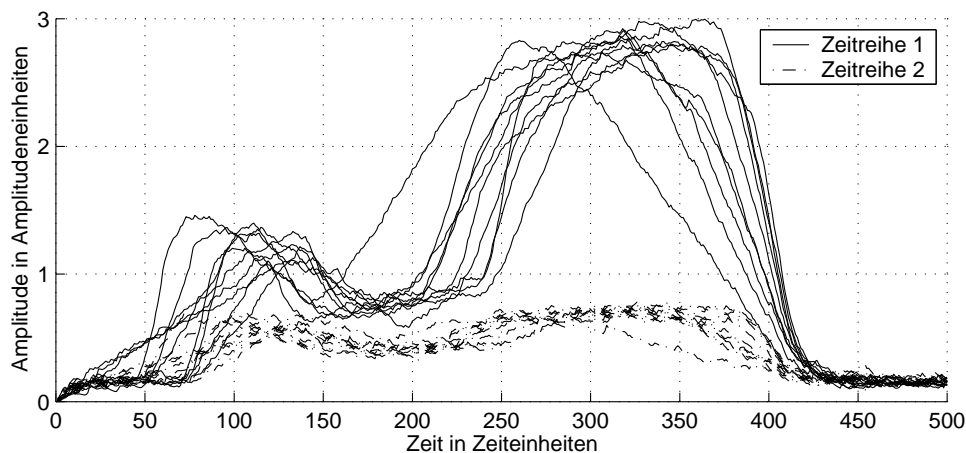


Bild 2.7: Repräsentativer Verlauf von zehn Realisierungen des gleichen Benchmarkschaltsignaltyps

Zur Verdeutlichung und Visualisierung der nachfolgenden Verfahren wird ein Datensatz mit $n_j = 10, j = 1 \dots K$, Realisierungen entworfen (Datensatz A). Zum Anlernen möglicher Klassifikatoren empfiehlt sich ebenfalls ein Datensatz mit einer geringen Anzahl an Realisierungen (ähnlich zu Datensatz A), da von einem echten Anwender kaum mehr Daten zu erhalten sind - der Anwender darf bei einer Aufnahmesitzung nicht überanstrengt werden. Die Ergebnisse, die einem Klassifikator basierend auf einem einzigen Lerndatensatz entstammen, sind jedoch statistisch nicht signifikant. Datensatz B vereint deshalb $n_j = 1000, j = 1 \dots K$, Realisierungen pro Klasse, aus denen zum Anlernen von Klassifikatoren jeweils zufällig $n_j = 10$ Realisierungen pro Klasse gezogen werden. Es ergibt sich ein (Lern-) Datensatz L. Datensatz C dient schließlich zum Testen der Güte entworfener Klassifikatoren. Er enthält $n_j = 1000, j = 1 \dots K$, Realisierungen pro Klasse, um eine verlässliche Aussage über Klassifikationsergebnisse zu erhalten. Um nachzuweisen, ob das MMI in der Lage ist, viele Bewegungen zu unterscheiden, werden alle Datensätze mit einer großen Anzahl an Schaltsignalklassen ($K = 8$) entworfen.

Um die Robustheit bei zeitvarianten Änderungen zu evaluieren, werden drei weitere Testdatensätze D_1, D_2, D_3 mit kleinen, mittleren und großen zeitvarianten Änderungen entworfen (vgl. Anhang 7.1.2). Identisch zu Datensatz C vereinen alle drei Datensätze $K = 8$ Klassen mit jeweils $n_j = 1000, j = 1, \dots, K$ Beispielen.

Zu Darstellungszwecken leitet Abschnitt 2.8 den Datensatz A^* aus Datensatz A ab, indem die wichtigsten

$s_m = 6$ Merkmale ausgewählt, zu $s_d = 2$ Merkmalen linear kombiniert und das Problem auf $K = 6$ vermindert wird.

2.6 Merkmalsextraktion

2.6.1 Übersicht

Ein Merkmal bezeichnet eine aus einer Zeitreihe abgeleitete Größe, die zur Abstraktion des zugehörigen Schaltsignals verwendet wird. Es ist Informationsträger über einen Zustand oder eine Eigenschaft des Schaltsignals und dient zu dessen Interpretation im Sinne der Klassifikation. Idealerweise lässt die Gesamtheit aller extrahierten Merkmale eine Rekonstruktion des Schaltsignals zu.

Für echtzeitfähige und informationsträchtige Merkmale werden folgende Forderungen formuliert:

1. Der Berechnungsaufwand muss möglichst gering sein, um die Onlinefähigkeit zu garantieren.
2. Die Merkmale müssen jedes für sich gut geeignet sein, um die Schaltsignalklassen voneinander zu trennen. Diese Eigenschaft beschreibt die univariate Trenngüte.
3. Die Kombination von Merkmalen muss gut geeignet sein, um die Schaltsignalklassen voneinander zu trennen. Diese Eigenschaft beschreibt die multivariate Trenngüte.
4. Die Merkmale müssen eine geringe Empfindlichkeit gegenüber zeitvarianten Änderungen, Signalrauschen und Messfehlern aufweisen.
5. Der Wert eines Merkmals darf nicht von Größen, die unabhängig vom Schaltsignal sind (wie z.B. die Vorlaufzeit bis zum Generieren des Schaltsignals), beeinflusst werden.

Die Forderung 1 kann anhand des Kriteriums Q_M validiert werden. Die Forderungen 2-5 werden anhand der in Abschnitt 2.7 eingeführten Verfahren ANOVA und MANOVA validiert. Eine Auswertung für die in dieser Arbeit gewählten Merkmale zeigt Abschnitt 2.10.2.

Grundsätzlich lassen sich aus einer Zeitreihe Zeitreihenmerkmale oder Einzelmerkmale extrahieren. Zeitreihenmerkmale werden zu jedem Zeitpunkt berechnet und ergeben sich aus dem aktuellen Wert von gemessenen oder daraus transformierten Zeitreihen. Einzelmerkmale ergeben sich aus charakteristischen Signalabschnitten definierter Länge oder werden nur zu festgelegten Zeitpunkten berechnet. Sie können auf der Basis von gemessenen Zeitreihen berechnet werden (z. B. Dauer von charakteristischen Bereichen, Anzahl Nullstellen, etc.) oder ebenfalls einer Transformation der Zeitreihe (Trendberechnung, Standardabweichung, etc.) entstammen.

Unabhängig davon, ob Zeitreihen- oder Einzelmerkmale extrahiert werden sollen, sind solche Transformationen in der Biosignalanalyse von großer praktischer Bedeutung, die informationstragende Frequenzbänder aus den Signalen extrahieren [26, 176]. Je nach Anwendung und Ursprung der Zeitreihe kommen ARMAX-Modelle, Wavelet-, Fourier- oder Gabor-Transformationen oder einfache Bandpassfilterungen zum Einsatz [37, 63, 100, 102, 120].

Das Erfassen eines kompletten Frequenzspektrums stellt hohe Anforderungen an die Abtastsysteme, verkürzt die Zeitintervalle, in denen Auswertoperationen möglich sind, und ist meist sehr rechenintensiv. Deswegen ist die Extraktion von Frequenzbändern durch einfache algebraische Operationen, wie z. B. Bandpassfilter oder die diskrete Wavelet-Transformation zu bevorzugen.

Im folgenden Abschnitt wird auf die Extraktion von Einzelmerkmalen aus Schaltsignalen zum Schalten mit Zeitversatz eingegangen. Zwar sind die zu extrahierenden Merkmale immer problemspezifisch zu

wählen, MMIs auf der Basis von Aktivitätssignalen zum Schalten mit Zeitversatz lassen sich jedoch fast immer umsetzen und sind somit generell einsetzbar.

Bei der Merkmalsextraktion ist darauf zu achten, einen großen Vorrat potenzieller Merkmale zu erhalten, um dann problemspezifisch die wichtigsten zu verwenden. Um mit zeitvarianten Änderungen umzugehen, sind Merkmale und Zeitreihen in Relation zueinander zu setzen.

2.6.2 Merkmale für Aktivitätssignale zum Schalten mit Zeitversatz

Schaltssignale mit Zeitversatz zeichnen sich oftmals durch charakteristische Zeitbereiche aus, die durch die Aktivität des abgegriffenen biometrischen Systems bestimmt sind (Bild 2.8). Ausgeprägte Minima und Maxima charakterisieren den Signalverlauf. Somit bietet sich das Unterteilen des Signalverlaufs in Bereiche an, die separat zur Extraktion von Einzelmerkmalen herangezogen werden. Da biometrische Signale jedoch häufig Störungen enthalten, ist eine Online-Suche nach Extrema schwierig und entsprechend anfällig - hier bietet sich das Tiefpass-Filtern des Signals an. Im Gegensatz zu Rechteckfiltern, die auf eine umfassende Archivierung vergangener Signalwerte angewiesen sind, bieten IIR-Filter rekursive Implementierungen, die Archivierung beschränkt sich auf wenige Werte. Mit einem IIR-Filter erster Ordnung wird ein Signal $u_j[k]$ gemäß

$$u_{f,j}[k] = a \cdot u_{f,j}[k-1] + (1-a) \cdot u_j[k], \quad u_{f,j}[0] = 0, \quad 0 < a < 1. \quad (2.24)$$

tiefpassgefiltert (Filterparameter $a = 0.99$, Abtastfrequenz $f_a = T_a^{-1} = 100\text{Hz}$, Abtastzeit $T_a = 0.01\text{s}$)⁵. Das zusätzliche Einführen von Hysteresen $u_{\text{Min},h}$, $u_{\text{Max},h}$ dient dem Verhindern von Fehlentscheidungen. Bei einer Differenz von $u_{\text{Min},h}$ zwischen aktuellem Filterwert $u_{f,j}[k]$ und vermutetem Minimum $u_{\text{Min},j}$ wird das Minimum erkannt und die Suche nach einem Maximum beginnt. Äquivalent wird bei einer Differenz von $u_{\text{Max},h}$ zwischen vermutetem Maximum $u_{\text{Max},j}$ und aktuellem Filterwert $u_{f,j}[k]$ das Maximum erkannt.

Mit dem Indikator $I_j[k]$ ($I_j[k] = 1$ für eine Maximumsuche, $I_j[k] = 0$ für eine Minimumsuche) ergibt sich

$$I_j[k] = \begin{cases} 0 & \text{wenn } (I_j[k-1] = 1) \cap (u_{\text{Max},j} - u_{f,j}[k] > u_{\text{Max},h}) \\ 1 & \text{wenn } (I_j[k-1] = 0) \cap (u_{f,j}[k] - u_{\text{Min},j} > u_{\text{Min},h}) \\ I_j[k-1] & \text{sonst} \end{cases} \quad (2.25)$$

mit

$$u_{\text{Min},j} = \begin{cases} u_{f,j}[k] & \text{wenn } ((I_j[k] = 0) \cap (I_j[k-1] = 0) \cap (u_{f,j}[k] < u_{\text{Min},j})) \\ & \cup \\ & ((I_j[k] = 0) \cap (I_j[k-1] = 1)), \end{cases} \quad (2.26)$$

$$u_{\text{Max},j} = \begin{cases} u_{f,j}[k] & \text{wenn } ((I_j[k] = 1) \cap (I_j[k-1] = 1) \cap (u_{f,j}[k] > u_{\text{Max},j})) \\ & \cup \\ & ((I_j[k] = 1) \cap (I_j[k-1] = 0)). \end{cases} \quad (2.27)$$

Eine grafische Darstellung findet sich in Bild 2.8, den zugehörigen Automaten zeigt Bild 4.8 in Kapitel 4.

⁵Sofern nicht anders kenntlich gemacht, werden alle nachfolgenden Beispiele mit dieser Parametrierung durchgeführt.

Zum Festlegen von Start- bzw. Endzeitpunkt eines Bereichs ergeben sich somit zwei Möglichkeiten. Die Zeitpunkte des Auftretens des i -ten Extremums werden im Weiteren als $k_{e,ji}$ (bzw. $t_{e,ji}$), die Zeitpunkte des eindeutigen Erkennens des Extremums durch die Hysterese als $k_{h,ji}$ ($t_{h,ji}$) bezeichnet. Für jeden Bereich i gilt

$$k_{h,j} = \{k | I_j[k+1] \neq I_j[k]\}. \quad (2.28)$$

Welcher dieser Zeitpunkte trennschärfere Merkmale erzeugt, wird im weiteren Verlauf mit einer univariaten bzw. multivariaten Analyse der Varianzen in Abhängigkeit der verwendeten Merkmale untersucht. Der Zeitpunkt des Bereichsendes sei solange mit $k_{\text{end},ji}$ bezeichnet. Die Beziehung zwischen Abtastzeitpunkt k und Zeitpunkt t ist gegeben durch

$$t = kT_a. \quad (2.29)$$

Die Anzahl der zu suchenden Extrema wird für das Benchmarkbeispiel auf $n_e = 4$ beschränkt. Dies führt zum Auffinden von fünf Bereichen:

- Bereich I (bis zum ersten Minimum),
- Bereich II (vom ersten Minimum zum ersten Maximum),
- Bereich III (vom ersten Maximum zum zweiten Minimum),
- Bereich IV (vom zweiten Minimum zum zweiten Maximum) und
- Bereich V (nach dem zweiten Maximum).

Der erste und fünfte Bereich eignen sich jedoch für die Merkmalerzeugung nicht. In Bereich I ist die Aktivierung des Signals zu klein, um signifikante Merkmale zu ergeben. Bereich V bildet lediglich die Filtereigenschaften der Soft- und evtl. Hardwarefilter ab (Bild 2.8).

Neben den n_s Sensorsignalen ergeben sich Merkmale aus weiteren synthetischen Zeitreihen, die Verhältnissen und Differenzen für alle Kombinationen zweier Sensoren j_1, j_2 entstammen. Mit $j_1 = 1, \dots, n_s - 1$ und $j_2 = j_1 + 1, \dots, n_s$ gilt

$$u_{f,v_a}[k] = \frac{u_{f,j_1}[k]}{(u_{f,j_2}[k] + u_{f,j_1}[k])}, \quad v_a = n_s + \sum_{i=2}^{j_1} (n_s - i + 1) + (j_2 - j_1), \quad (2.30)$$

$$u_{f,w_a}[k] = u_{f,j_1}[k] - u_{f,j_2}[k], \quad w_a = n_s + \frac{1}{2}n_s(n_s - 1) + \sum_{i=2}^{j_1} (n_s - i + 1) + (j_2 - j_1). \quad (2.31)$$

Dies führt zu insgesamt $n_s^* = n_s^2$ Zeitreihen.

Potenzielle Merkmale für jeden Bereich $i \in [2, n_e]$ und jede Zeitreihe $j \in [1, n_s^*]$ sind:

- Extremum $x_{e,ji}$ des gefilterten Signals $u_{f,j}[k]$ (Maximum in Bereich II and IV, Minimum in III),
- relatives Extremum $x_{r,ji} = x_{e,ji} / \max_i(x_{e,ji})$,
- zeitliche Ausdehnung des Bereichs $x_{t,ji} = k_{\text{end},ji} - k_{\text{end},j(i-1)}$,
- relative zeitliche Ausdehnung $x_{z,ji} = x_{t,ji} / (k_{\text{end},jn_e} - k_{\text{end},j1})$,
- Summensignal⁶ zum Bereichsende $x_{s,ji} = u_{s,j}[k_{\text{end},ji}]$ mit $u_{s,j}[k] = \sum_{l=k_{\text{end},j(i-1)}}^k u_{f,j}[l]$,
- relatives Summensignal $x_{u,ji} = x_{s,ji} / (k_{\text{end},jn_e} - k_{\text{end},j1})$,
- Anzahl von Schnittpunkten $x_{sp,ji}$ zwischen Signal $u_j[k]$ und Filter $u_{f,j}[k]$,

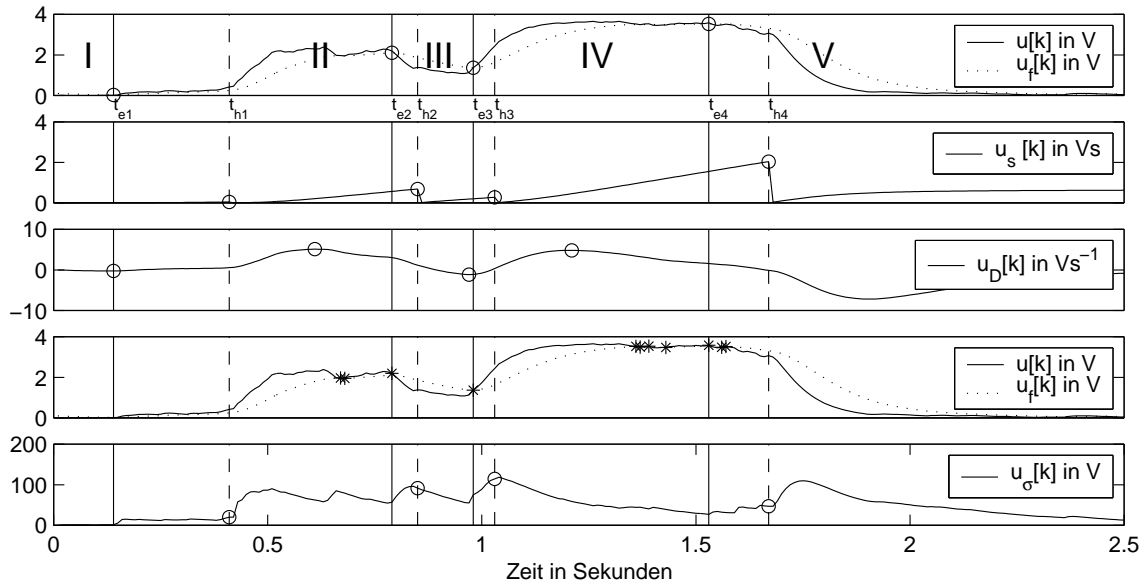


Bild 2.8: Generierte Zeitreihen und erzeugte Merkmale (durch 'o' gekennzeichnet) anhand einer Zeitreihe $u[k]$, von oben nach unten: Sensorsignal $u[k]$ und Filter $u_f[k]$, Summensignal $u_s[k]$, Trend $u_D[k]$, Schnittpunkte (durch '*' gekennzeichnet) zwischen Sensorsignal $u[k]$ und Filter $u_f[k]$ sowie gefilterte Standardabweichung $u_\sigma[k]$, Zeitbereiche I, II, ..., V

- Extremum $x_{D,ji}$ eines Trends $u_{D,j}[k]$ (Maximum in Bereich II and IV, Minimum in III) sowie
- gefilterte Standardabweichung zum Bereichsende $x_{\sigma,ji} = u_{\sigma,j}[k_{\text{end},ji}]$.

Die Berechnung der Merkmale wird im Folgenden beschrieben: Die Anzahl der Schnittpunkte $x_{sp,j}[k]$ zwischen Signal und Filter ergibt sich durch Vorzeichenwechsel deren Differenz und wird nur gewertet, wenn eine minimale Schwelle u_{\min} überschritten ist:

$$u_{sp,j}[k] := \begin{cases} u_{sp,j}[k-1] + 1 & \text{wenn } ((u_{f,j}[k] \leq u_j[k] \wedge u_{f,j}[k-1] > u_j[k-1]) \vee \\ & (u_{f,j}[k] \geq u_j[k] \wedge u_{f,j}[k-1] < u_j[k-1])) \wedge \\ & (u_{f,j}[k] > u_{\min}) \\ 0 & \text{wenn } I_j[k] \neq I_j[k-1] \\ u_{sp,j}[k-1] & \text{sonst} \end{cases} \quad (2.32)$$

mit

$$u_{sp,j}[0] = 0. \quad (2.33)$$

Das Merkmal $x_{sp,ji}$ ergibt sich dann zu

$$x_{sp,ji} = u_{sp,j}[k_{\text{end},ji}] - u_{sp,j}[k_{\text{end},j(i-1)}]. \quad (2.34)$$

Ein Trend beruht auf der Differenzbildung zweier IIR-Filter $u_S[k]$, $u_L[k]$ (mit $a = a_S$ bzw. $a = a_L$, $a_S < a_L$) gemäß (2.24) und beschreibt die Steigung des Signals. Das Einführen der Größen T_N und T_M

⁶Mittelwert und Summensignal eines Bereichs sind zueinander proportional, eine Berechnung von beiden liefert keine Erhöhung der Trenngüte. Eine Möglichkeit zur effizienten rekursiven Berechnung zeigt Bild 4.8 in Kapitel 4.

bzw. deren Differenz führt zur Möglichkeit, den Trend zu normieren [111]:

$$u_{D,j}[k] = \frac{u_{S,j}[k] - u_{L,j}[k]}{T_S - T_L} \quad (2.35)$$

mit

$$T_S - T_L = T_a \cdot \frac{a_L - a_S}{(1 - a_L)(1 - a_S)}. \quad (2.36)$$

Die gefilterte Standardabweichung kann unter der Voraussetzung einer normalverteilten Störung durch ein IIR-Filter mit Filterparameter a_σ gemäß (2.24) berechnet werden:

$$u_{\sigma,j}^2[k] = a_\sigma \cdot u_{\sigma,j}^2[k-1] + (1 - a_\sigma) \cdot \frac{1}{2} \cdot (u_j[k] - u_j[k-1] - u_{D,j}[k] \cdot T_a)^2. \quad (2.37)$$

Bei Vorhandensein eines Trends $u_{D,j}[k] \neq 0$ im Signal treten Fehler bei der Berechnung der Standardabweichung auf, da $u_j[k]$ durchschnittlich um $u_{D,j}[k] \cdot T_a$ größer ist als $u_j[k-1]$. Dieser Fehler wird durch eine Subtraktion von $u_{D,j}[k] \cdot T_a$ zumindest teilweise kompensiert [150].

Die angeführten Berechnungsvorschriften (2.24)-(2.37) gelten für konstante Abtastzeiten bzw. Frequenzen. Bei zeitvarianten Abtastzeiten ($T_a = T_a[k]$) müssen Anpassungen vorgenommen werden. Für alle folgenden Merkmalsberechnungen gelten $a_\sigma = 0.99$, $a_S = 0.98$, $a_L = 0.99$, $u_{\min} = 0.2V$, $u_{\max,h} = u_{\min,h} = 0.2V^7$.

Bild 2.8 zeigt am Beispiel einer Zeitreihe die extrahierten Merkmale sowie die Zeitpunkte $t_{e,ji} = t_{e,i}$ bzw. $t_{h,ji} = t_{h,i}$, die zur Wahl der Bereiche verwendet werden können. Sofern weniger als n_e Extrema vorliegen, stoppt der Algorithmus nach k_{\max} Abtastzeitpunkten und setzt nicht berechnete Merkmale zu Null. Einen Algorithmus zur Echtzeitberechnung der Merkmale führt Abschnitt 4.3 ein.

Zur Klassifikation von Schaltsignalen ist es für Verfahren, die auf richtungsneutralen Distanzmaßen basieren, von Vorteil, wenn ein Merkmal x_l nur in einem vorgegebenen Intervall streut (z. B. $x_l \in \{0, \dots, 1\}$). Bei Vorliegen von n Schaltsignalrealisierungen, geschieht diese Intervalltransformation durch

$$x_{\text{trans},l,i} = \frac{x_{l,i} - \min_i x_{l,i}}{\max_i x_{l,i} - \min_i x_{l,i}} \quad i \in 1, \dots, n. \quad (2.38)$$

Die Wahl des Zeitpunktes zur Berechnung der Merkmale wird in Abschnitt 2.10.2 getroffen, anhand der in den nachfolgenden Abschnitten eingeführten Evaluierungsverfahren. Ebenso werden dann die expliziten Berechnungsvorschriften vorgestellt.

2.7 Merkmalsselektion

Das Beurteilen des Signalverlaufs gestaltet sich umso schwieriger, je mehr redundante Merkmale zur Verfügung stehen. Mitunter führt eine zu große Anzahl an der Entscheidungsfindung teilnehmender Merkmale zu fehlerhaften Entscheidungen [40]. Insbesondere das vorgestellte Verfahren zur Merkmalsextraktion liefert

- eine große Anzahl an Merkmalen mit starken Korrelationen,
- teilweise Merkmale ohne Streuung für ausgewählte Klassen (z. B. relative Extrema) und

⁷Die Parameter wurden heuristisch eingestellt mit der Forderung, über dem verwendeten Benchmarkdatensatz gute Glättungseigenschaften zu haben und Extrema zuverlässig zu erkennen.

- Merkmale mit Mittelwert Null ohne Streuung für ausgewählte Klassen (z. B. in Bereich IV, wenn nur drei Bereiche vorhanden sind).

Weiterhin stellt das Berechnen von Merkmalen in Echtzeit eine erhebliche Belastung für das MMI dar. Deshalb ist es wichtig, einen offline generierten Merkmalsatz nicht im Ganzen zu beurteilen, sondern nur aussagekräftige Merkmale auszuwählen, ohne dabei zu viel trennstärke Information zu verlieren. Diese Dimensionsreduktion führt zu einer schnelleren und speichereffizienteren Berechnung und evtl. zu einem Einsparen von Merkmalskosten (wenn das Messen von Merkmalen aufwändig ist).

Unter Vorgabe einer Anzahl s_m zu selektierender Merkmale aus den s vorhandenen Merkmalen ist eine Zielfunktion M zu entwerfen, die für trennstärkere Merkmalskombinationen monoton steigt. Gesucht ist nun die Indexmenge $\mathcal{I} = \{i_1, \dots, i_{s_m}\}$ aus der Menge aller möglichen Indexmengen $\mathcal{I}_{\text{cand}}$ für die M maximal wird:

$$\mathcal{I} = \underset{\mathcal{I}_{\text{cand}}}{\operatorname{argmax}} M(\mathcal{I}_{\text{cand}}) \quad \text{mit} \quad \operatorname{card}(\mathcal{I}_{\text{cand}}) = s_m. \quad (2.39)$$

Die Kardinalität (card) bezeichnet die Anzahl der in $\mathcal{I}_{\text{cand}}$ enthaltenen Elemente.

In [39] konnte nachgewiesen werden, dass nur die Suche über alle $\binom{s}{s_m}$ möglichen Merkmalskombinationen mit Sicherheit zum besten Ergebnis führt. Um dieses rechenintensive Problem zu umgehen, kommen suboptimale Verfahren zum Einsatz. So kann die Menge \mathcal{I} ausgehend von der leeren Menge schrittweise um ein Merkmal ergänzt werden (Vorwärtssuche), oder sie kann ausgehend von der vollbesetzten Menge schrittweise um ein Merkmal verringert werden (Rückwärtssuche), so dass M maximal wird. Eine Kombination aus beiden Verfahren ist ebenfalls möglich.

Geschieht die Konstruktion der Zielfunktion M unabhängig von dem zu entwerfenden Klassifikator, so handelt es sich um einen Filter-Ansatz. Geht hingegen die Klassifikationsgüte des nachgeschalteten Klassifikators in die Zielfunktion ein, so wird von einem Wrapper-Ansatz gesprochen. Letzterer liefert i. A. bessere Klassifikationsgüten, der Entwurf ist jedoch wesentlich rechenzeitintensiver [68, 86].

Zur Berechnung der Zielfunktion M kommt häufig die Multivariate Analyse der Varianzen (MANOVA, [2, 195]) zum Einsatz, die Konstruktion der Zielfunktion M geschieht folgendermaßen: Für eine potenzielle Merkmalskombination mit der Indexmenge \mathcal{I} werden Innerklassen- und Zwischenklassenvariationsmatrizen berechnet.

Es wird vereinbart:

$$\bar{\mathbf{x}}|_{\mathcal{I}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i|_{\mathcal{I}} \quad \text{Gesamtmittelwert,} \quad (2.40)$$

$$\bar{\mathbf{x}}_j|_{\mathcal{I}} = \frac{1}{n_j} \sum_{i:y_i=j} \mathbf{x}_i|_{\mathcal{I}} \quad \text{Klassenmittelwerte der Klasse } j, \quad (2.41)$$

$$\hat{\Sigma}_j|_{\mathcal{I}} = \frac{1}{n_j} \sum_{i:y_i=j} (\mathbf{x}_i|_{\mathcal{I}} - \bar{\mathbf{x}}_j|_{\mathcal{I}})(\mathbf{x}_i|_{\mathcal{I}} - \bar{\mathbf{x}}_j|_{\mathcal{I}})^T \quad \text{geschätzte Klassenkovarianzmatrizen,} \quad (2.42)$$

$$\mathbf{B}|_{\mathcal{I}} = \sum_{j=1}^K n_j (\bar{\mathbf{x}}_j|_{\mathcal{I}} - \bar{\mathbf{x}}|_{\mathcal{I}})(\bar{\mathbf{x}}_j|_{\mathcal{I}} - \bar{\mathbf{x}}|_{\mathcal{I}})^T \quad \text{Zwischenklassenvariationsmatrix,} \quad (2.43)$$

$$\mathbf{W}|_{\mathcal{I}} = \sum_{j=1}^K n_j \cdot \hat{\Sigma}_j|_{\mathcal{I}} \quad \text{Innerklassenvariationsmatrix und} \quad (2.44)$$

$$\mathbf{T}|_{\mathcal{I}} = \mathbf{W}|_{\mathcal{I}} + \mathbf{B}|_{\mathcal{I}} \quad \text{Gesamtvariationsmatrix.} \quad (2.45)$$

Die Notation $|\mathcal{I}$ meint, dass nur die in der Menge \mathcal{I} spezifizierten Merkmale verwendet werden, um die betreffenden Vektoren und Matrizen zu bilden.

Es besteht nun das Ziel, im dimensionsreduzierten Raum die Zwischenklassenvariationsmatrix \mathbf{B} bei Fixierung der Innerklassenvariationsmatrix \mathbf{W} zu maximieren. Grob gesprochen wird das Verhältnis aus Klassenabständen und Klassenstreuungen im niederdimensionalen Raum maximiert. Mit dem Bezeichner \mathbf{I} für die Einheitsmatrix ergibt sich folgendes Eigenwertproblem (Eigenvektoren $\boldsymbol{\nu}_i$):

$$(\mathbf{W}|\mathcal{I}^{-1}\mathbf{B}|\mathcal{I} - \lambda_i\mathbf{I})\boldsymbol{\nu}_i = \mathbf{0}. \tag{2.46}$$

Aus den sortierten Eigenwerten λ_i lassen sich nun Maße konstruieren, die als Zielfunktion zur Merkmalsselektion geeignet sind. In der vorliegenden Arbeit wird das Likelihood-Quotienten-Kriterium

$$M(\mathcal{I}) = 1 - \prod_{i=1}^{s_m} \frac{1}{1 + \lambda_i} \tag{2.47}$$

zum Bewerten der Merkmalskombination mit der Indexmenge \mathcal{I} verwendet. Ein Beispiel für diese Vorgehensweise findet sich in [113]. Bild 2.9 zeigt die Histogramme über den Klassen von drei Merkmalen des Benchmarkdatensatzes A und deren Bewertung. Demnach lässt ein Merkmal mit hohem λ oft nicht eine Unterscheidung aller Klassen zu. Das Verfahren wählt evtl. trennschwache Merkmale aus, wenn für wenige Klassen keine Streuung auftritt, deren Mittelwerte jedoch verschieden sind.

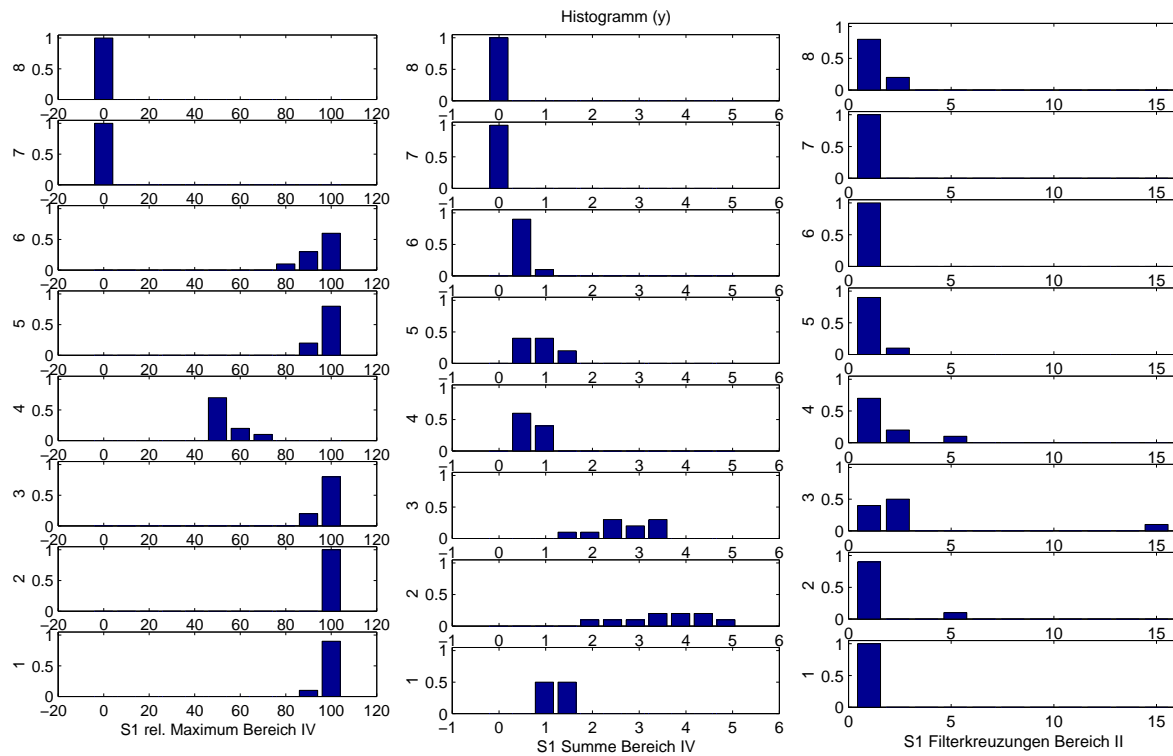


Bild 2.9: Univariate Analyse der Varianzen anhand dreier Merkmale aus Benchmarkdatensatz A. Links: Merkmal x_{12} : $\lambda_1 = 0.951$, Mitte: Merkmal x_{15} : $\lambda_1 = 0.882$, rechts: Merkmal x_{19} : $\lambda_1 = 0.185$, vgl. Anhang 7.5.

Ein Vergleich der Komplexität zwischen voller Suche und Vorwärtssuche ergibt, dass bei der vollen Suche $\binom{s}{s_m}$ -mal das s_m -dimensionale Eigenwertproblem gelöst werden muss, während zur Vorwärtssuche lediglich s -mal das eindimensionale, $(s - 1)$ -mal das zweidimensionale, ..., $(s - s_m + 1)$ -mal das

s_m -dimensionale Eigenwertproblem gelöst werden muss. Aus Aufwandsgründen ist hier somit klar die suboptimale Vorwärtssuche zu bevorzugen⁸.

Zur Vereinfachung der Schreibweise von $\mathbf{x}|_{\mathcal{I}}$, $\mathbf{X}|_{\mathcal{I}}$, $\mathbf{B}|_{\mathcal{I}}$, $\mathbf{W}|_{\mathcal{I}}$, $\mathbf{T}|_{\mathcal{I}}$ wird im Folgenden auf die Notation $|_{\mathcal{I}}$ verzichtet.

2.8 Merkmalsaggregation

Im Gegensatz zur Merkmalsselektion reduziert die Merkmalsaggregation die Dimension des Merkmalsraums, ohne dabei Merkmale völlig außer Acht zu lassen. Vielmehr findet ein Wichten der Merkmale statt. Das Ziel ist wiederum das Einsparen von Rechenzeit, weil die nachfolgenden Entscheidungsregeln in einem niederdimensionalen Raum arbeiten. Es existiert eine Vielzahl von Verfahren, die linear/nichtlinear, überwacht/unüberwacht und auf der Basis von Maßen/Wahrscheinlichkeiten/Heuristiken arbeiten. Hierbei können klassische statistische Verfahren, Künstliche Neuronale Netze, Kernfunktions-Methoden, etc. zum Einsatz kommen. Übersichten sind [31, 77, 182] zu entnehmen.

Die bekanntesten Verfahren zur linearen Aggregation mit einer Transformations- bzw. Projektionsmatrix \mathbf{A} (nach (2.5)) gemäß

$$\mathbf{Z} = \mathbf{X}\mathbf{A} \quad \text{bzw.} \quad \mathbf{z}^T = \mathbf{x}^T \mathbf{A} = (\mathbf{A}^T \mathbf{x})^T \quad (2.48)$$

sind die Hauptkomponentenanalyse (HKA) und die lineare Diskriminanzanalyse (LDA).

Die Hauptkomponentenanalyse setzt voraus, dass der gesamte Datensatz normalverteilt ist. Ohne Kenntnis der Klassenzugehörigkeiten versucht sie, die Streuung im aggregierten Merkmalsraum zu maximieren, mit der Restriktion orthogonale Transformationsvektoren zu berechnen. Mit $\mathbf{A} \in \mathbb{R}^{s_m \times s_d}$ ergibt sich die Zielfunktion

$$\text{sp}(\mathbf{A}^T \mathbf{T} \mathbf{A}) \rightarrow \max_{\mathbf{A}} \quad \text{bez.} \quad \mathbf{A}^T \mathbf{A} = \mathbf{I}_{s_d} \quad (2.49)$$

und das daraus folgende Eigenwertproblem

$$(\mathbf{T} - \lambda_i \mathbf{I}) \boldsymbol{\nu}_i = \mathbf{0}. \quad (2.50)$$

Bei der linearen Diskriminanzanalyse sind im Gegensatz zur Hauptkomponentenanalyse die Klassenbezeichner der Datentupel bekannt. Für den Zweiklassenfall wurde sie erstmals in [56] und für den Mehrklassenfall in [148] beschrieben.

Äquivalent zu (2.46) ist es das Ziel, das Verhältnis von Zwischenklassenvariation zu Innerklassenvariation zu maximieren. In diesem Fall müssen allerdings die Zwischenklassenvariationsmatrix der aggregierten Merkmale $\mathbf{A}^T \mathbf{B} \mathbf{A}$ und die Innerklassenvariationsmatrix $\mathbf{A}^T \mathbf{W} \mathbf{A}$ der aggregierten Merkmale betrachtet werden. Konkret ist zu lösen:

$$\text{sp}(\mathbf{A}^T \mathbf{B} \mathbf{A}) \rightarrow \max_{\mathbf{A}} \quad \text{bez.} \quad \mathbf{A}^T \mathbf{W} \mathbf{A} = \mathbf{I}_{s_d}. \quad (2.51)$$

Es ergibt sich folgendes Eigenwertproblem

$$(\mathbf{W}^{-1} \mathbf{B} - \lambda_i \mathbf{I}) \boldsymbol{\nu}_i = \mathbf{0}. \quad (2.52)$$

Die Lösung für die beiden Eigenwertprobleme (2.50) und (2.52) lautet

$$\mathbf{A}_{\text{opt}} = (\mathbf{a}_1, \dots, \mathbf{a}_{s_d}) \quad \text{mit} \quad \mathbf{a}_i = \boldsymbol{\nu}_i, \quad (2.53)$$

⁸Bereits bei suboptimaler Auswahl von $s_m = 4$ Merkmalen beträgt $M(\mathcal{I}) = 1 - 7 \cdot 10^{-7}$ bei Datensatz A, das Optimum beträgt 1.

wobei die \mathbf{a}_i die zu den geordneten Eigenwerten ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{s_d} \geq \dots \geq \lambda_{s_m}$) korrespondierenden Eigenvektoren \mathbf{v}_i von \mathbf{T} bzw. $\mathbf{W}^{-1}\mathbf{B}$ bezeichnen. Bei der LDA gilt die strenge Ungleichheit mit Wahrscheinlichkeit 1, für alle λ_i mit $i < K$. Weiterhin gilt

$$\lambda_i = 0 \quad \text{für} \quad i \geq K. \tag{2.54}$$

In Bild 2.10 bzw. Bild 2.11 sind die durch HKA bzw. LDA aggregierten Merkmalsräume anhand von Benchmarkdatensatz A dargestellt. Die HKA liefert im Gegensatz zur LDA aufgrund der Unkenntnis der Klasseninformation nicht zusammenhängende Klassen im aggregierten Merkmalsraum (z. B. ist Klasse 3 nicht kompakt in Bild 2.10 im Gegensatz zu Bild 2.11). Sie eignet sich somit schlechter zur Aggregation bei Kenntnis der Klassenzugehörigkeiten und wird im Weiteren nicht mehr betrachtet.

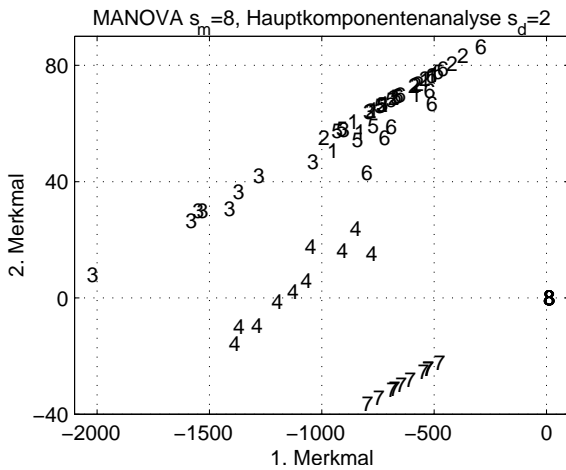


Bild 2.10: Reduktion von Benchmarkdatensatz A auf zwei Dimensionen mittels HKA

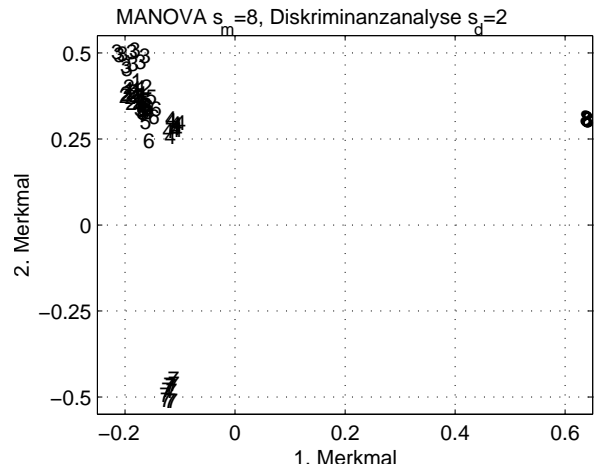


Bild 2.11: Reduktion von Benchmarkdatensatz A auf zwei Dimensionen mittels LDA

2.9 Konstruktion der Entscheidungsregel

2.9.1 Übersicht

Die Entscheidungsregel bildet den Merkmalsraum \mathcal{M} in die Menge der K möglichen Klassen ab. Diese Abbildung beruht entweder auf

- Entscheidungsmaßen $d_j(\mathbf{z})$ (statistisch bzw. geometrisch motiviert) oder
- einer Zuordnung zu Entscheidungsbereichen $\mathcal{G}_j(\boldsymbol{\theta}_j)$ mit $\mathcal{G}_i \cap \mathcal{G}_j = \emptyset$ für $i \neq j$ und $\bigcup_{j=1}^K \mathcal{G}_j = \mathcal{M}$ im Merkmalsraum.

Bei der Klassifikation mittels Entscheidungsmaß ist ein Datentupel \mathbf{z} derjenigen Klasse y zuzuordnen, für die das Maß minimal wird, d. h. als Klassenschätzung \hat{y} ergibt sich

$$\hat{y}(\mathbf{z}) = \operatorname{argmin}_{1 \leq j \leq K} d_j(\mathbf{z}; \boldsymbol{\theta}_j). \tag{2.55}$$

Die Beziehung (2.55) heißt Entscheidungsregel. θ_j fasst die Parameter des Entscheidungsmaßes in einem Vektor zusammen. Dieser Vektor kann beispielsweise die Elemente der Klassenmittelwertvektoren und der geschätzten Klassenkovarianzmatrizen enthalten.

Bei der Klassifikation mittels Entscheidungsbereichen ist ein Datentupel \mathbf{z} derjenigen Klasse $\hat{y} = j$ zuzuordnen, in deren Entscheidungsbereich sich \mathbf{z} befindet. Die Entscheidungsregel lautet

$$\hat{y}(\mathbf{z}) = j \quad \text{für } \mathbf{z} \in \mathcal{G}_j(\theta_j). \quad (2.56)$$

θ_j bezeichnet die zur Beschreibung des Entscheidungsbereichs \mathcal{G}_j notwendigen Parameter.

Für einfache Fälle lassen sich die Gebiete \mathcal{G}_j durch klassenspezifische Ungleichungssysteme darstellen⁹, wobei alle Ungleichungen erfüllt sein müssen, d. h.

$$\hat{y}(\mathbf{z}) = j, \text{ wenn } \mathbf{g}_j(\mathbf{z}) \geq \mathbf{0} \text{ und } \mathbf{g}_i(\mathbf{z}) < \mathbf{0} \text{ für mindestens eine Ungleichung und alle } i \neq j. \quad (2.57)$$

Ist nur eine Ungleichung nicht erfüllt, kann keine Zuordnung zur entsprechenden Klasse stattfinden. Ein einzelnes Gleichungssystem $\mathbf{g}_j(\mathbf{z}) = \mathbf{0}$ markiert eine Grenze zwischen zwei Klassen und wird als Diskriminanzfunktion bezeichnet.

Der Vorteil der Entscheidungsmaße ist, dass die aufwändige Auswertung der Ungleichungssysteme entfällt. Eine explizite Formulierung der Diskriminanzfunktionen und damit der Entscheidungsbereiche wird nicht benötigt. Dennoch ist es möglich, Diskriminanzfunktionen zur Trennung der Klassen i, j über $d_i(\mathbf{z}) = d_j(\mathbf{z})$ zu definieren. Allerdings lassen sich diese nicht zwingend durch geschlossene Formel­ausdrücke darstellen.

Als Vorteil der Entscheidungsbereiche erweist sich die Umsetzbarkeit beliebig komplexer Klassentrennungen. Eine gleichwertige Darstellung durch Entscheidungsmaße gelingt häufig nicht.

Das Schätzen der Parameter $\theta_1, \dots, \theta_K$ geschieht auf der Basis des Lerndatensatzes. In Abhängigkeit von der Art der Entscheidungsregel kommen verschiedene Verfahren zum Einsatz, die im Folgenden beschrieben werden. Bei allen Verfahren kann mit der vollen Anzahl Merkmale, mit ausgewählten oder aggregierten Merkmalen gearbeitet werden. Ohne Beschränkung der Allgemeinheit werden die folgenden Verfahren anhand des Datensatzes mit s Merkmalen beschrieben. Eine zweidimensionale Darstellung von Diskriminanzfunktionen geschieht anhand des Benchmarkdatensatzes A^* mit reduzierter Klassenanzahl ($K = 6$) und vorgeschalteter Merkmalsauswahl nach MANOVA mit $s_m = 8$ und Merkmalsaggregation durch die Diskriminanzanalyse mit $s_d = 2$.

2.9.2 Fuzzy-Regelbasen

Die Fuzzy-Logik stellt die Verallgemeinerung der klassischen Logik zur mehrwertigen Logik dar [213]. Während in der klassischen Logik für einen Ausdruck nur die Zugehörigkeitswerte 0 und 1 existieren, sind in der Fuzzy-Logik auch Zwischenwerte möglich. So lassen sich natürlich-sprachliche Begriffe mit einer hohen Interpretierbarkeit umsetzen.

Eine Fuzzy-Regelbasis nutzt die Fuzzy-Logik, um die quantitativen Werte der Eingangsgrößen (Merkmale x_i) in quantitative Werte der Ausgangsgrößen (Klassenschätzung \hat{y}) zu überführen. Sie besteht

⁹Gilt für ein zu klassifizierendes Datentupel $\mathbf{g}_i(\mathbf{z}) = \mathbf{g}_j(\mathbf{z}) = \mathbf{0}$, so erhält die Klasse mit der kleineren Klassenbezeichnung Vorrang.

aus Regeln der Form (WENN natürlich-sprachliche Bedingung DANN natürlich-sprachliche Anweisung). Den Einzugsbereich einer Regel aus dem Benchmarkdatensatz zeigt Bild 2.12 (vgl. Anhang 7.4):

WENN (S1 relatives Minimum Bereich III=POSITIV KLEIN)
 UND (S2 Maximum Bereich IV=POSITIV MITTEL, GROSS ODER SEHR GROSS)
 DANN $y=2$.

Die Regel deckt alle Beispiele der Klasse 2 und fälschlicherweise zwei Beispiele der Klasse 1 ab.

Die Auswertung natürlich-sprachlicher Regeln geschieht durch Fuzzifizierung, Inferenz und Defuzzifizierung [8, 78, 81, 111].

Die Fuzzifizierung weist dem Wert eines Merkmals x_i einen Zugehörigkeitswert $\mu_T(x_i)$ zu einem linguistischen Term T (z. B. 'GROSS', 'WARM', etc.) zu. Dazu dienen Zugehörigkeitsfunktionen (ZGF), die für jeden linguistischen Term definiert sind und Werte zwischen 0 und 1 annehmen können. Sie werden bestimmt durch ihre Form (dreiecks-, trapezförmig, etc.) und den Bereich eines Merkmals, den sie abdecken. Im Falle von dreiecksförmigen Zugehörigkeitsfunktionen ergibt sich dieser aus einem Stützpunkt sowie durch die Stützpunkte ihrer Nachbarn.

Die Inferenz beschreibt die Auswertung von logischen Bedingungen und Regeln der Form WENN (Teilprämisse 1 UND ... UND Teilprämisse T) DANN (Konklusion). Sie besteht aus den Schritten Prämissenauswertung, Aktivierung und Akkumulation. Aufgrund der Erweiterung der klassischen Logik auf Zugehörigkeitswerte zwischen 0 und 1 müssen vorab die klassischen logischen Verknüpfungen erweitert werden. Verbreitete Inferenzstrategien sind SUMPROD (Konjunktion: Produkt, Disjunktion: beschränkte Summe) und MAXMIN (Konjunktion: Maximum, Disjunktion: Minimum). Die Prämissenauswertung nutzt die Inferenzstrategie, um den Zugehörigkeitswert der gesamten Prämisse aus Zugehörigkeitswerten der Teilprämissen zu errechnen. Die Aktivierung weist der Regelkonklusion den Zugehörigkeitswert der Regelprämisse zu. Anschließend werden alle Zugehörigkeitswerte gleicher Konklusionen der Regelbasis ODER-verknüpft (Akkumulation I). Die Akkumulation II verknüpft die Zugehörigkeitswerte der Konklusion mit deren Zugehörigkeitsfunktion durch eine UND-Verknüpfung. Schließlich werden die erhaltenen, modifizierten Zugehörigkeitsfunktionen zu einer Fuzzy-Menge durch eine weitere ODER-Verknüpfung vereinigt (Akkumulation III). Bei Einsatz von Singletons als Ausgangs-Zugehörigkeitsfunktionen entfallen Akkumulation II und III.

Aus der Fuzzy-Menge kann ein scharfer, nicht linguistischer Ausgangswert gewonnen werden (Defuzzifizierung), indem beispielsweise das Maximum oder der Schwerpunkt der Fuzzy-Menge bestimmt wird. Je nach Problemstellung variiert die Art der Berechnung.

Einsatz finden Fuzzy-Regeln aufgrund ihres interpretierbaren Zugangs oft zum Modellieren von Expertenwissen oder zur Wissensextraktion aus Datensätzen. Eine aus Datensätzen extrahierte Regelbasis kann schließlich zur Klassifikation verwendet werden. Grundpfeiler der Regelbasisgenerierung sind die Hypothesengenerierung und deren Bewertung. Die Hypothesengenerierung kann z. B. durch Entscheidungsbäume, Clusteralgorithmen oder evolutionäre Algorithmen geschehen, die Bewertung z. B. durch Klassifikationsgüte, Entscheidungskosten oder Konfidenzintervalle [14, 87, 112].

In dieser Arbeit werden zum Entwurf von Fuzzy-Regelbasen mittels eines baumorientierten Vorgehens (WENN, DANN)-Regeln aus Beispieldaten abgeleitet [112, 146].

Zu Beginn wird die Anzahl zu verwendender Zugehörigkeitsfunktionen festgelegt. Als Zugehörigkeitsfunktionen kommen für jedes Merkmal Trapezfunktionen in Randbereichen und Dreiecksfunktionen zum Einsatz. Der notwendige Stützpunkt jeder Funktion wird für die beiden Trapezfunktionen durch den minimalen und den maximalen Funktionswert des Merkmals und für die Dreiecksfunktionen durch gleichmäßige Abdeckung der Beispiele bestimmt und nachfolgend mit einem speziellen Verfahren gerundet.

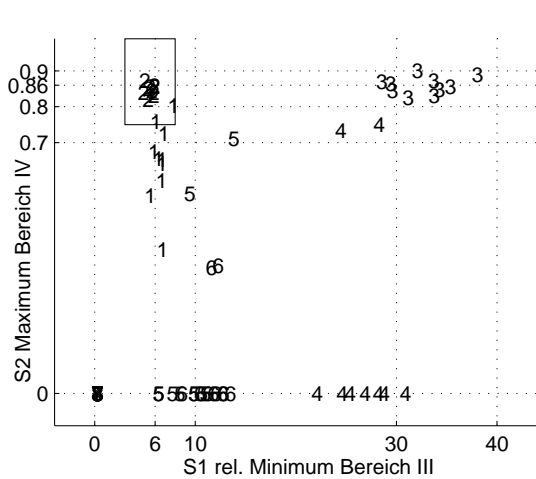


Bild 2.12: Fuzzy-Regel im Merkmalsraum
(Benchmark-Datensatz A)

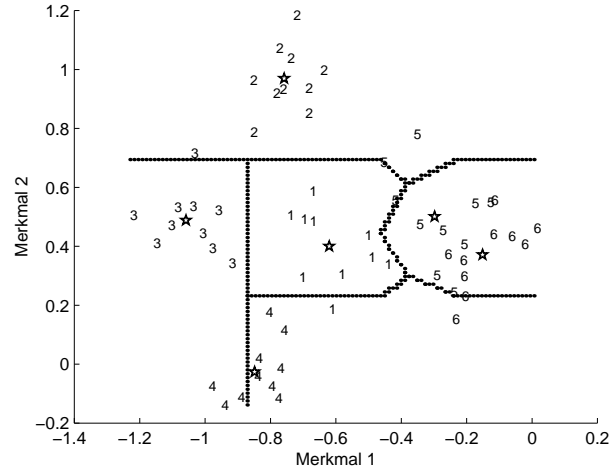


Bild 2.13: Diskriminanzfunktionen einer Fuzzy-
Regelbasis (Benchmarkdatensatz A*)

Zur Extraktion der Fuzzy-Regeln werden Entscheidungsbäume entworfen. Die Wurzel eines Baumes enthält das Merkmal, das auf der Basis der vorliegenden Beispiele die sicherste Klassifikation zulässt. Der Baum spaltet sich entsprechend der Anzahl der linguistischen Terme auf und es werden Merkmale gesucht, die in Kombination mit der im Wurzelknoten getroffenen Entscheidung die beste Klassifikation ermöglichen. Der Vorgang wird abgebrochen, wenn homogene Klassifikationen erreicht werden, oder wenn der relative Fehler einen vorgegebenen Wert unterschreitet. Die auf diese Weise extrahierbaren Regeln sind UND-verknüpft. Es folgt das Pruning der Regeln, d. h. das Erweitern der Teilprämissen um Nachbar-Zugehörigkeitsfunktionen oder das Streichen von Teilprämissen. Abschließend entscheidet eine Bewertungsfunktion über die Verwendung der so entstandenen Regeln. Um den Eingangsraum vollständig durch die Regelbasis abzudecken, wird noch eine Rückweisungsregel (SONST) zugefügt.

Im Ergebnis entstehen relevante Einzelregeln, die zu einer relativ kleinen Anzahl kooperierender Regeln in einer Regelbasis zusammengefasst sind. Anwendung findet dieses Verfahren u. a. bei medizinischen Diagnoseproblemen [113], bei denen die Interpretierbarkeit der Ergebnisse im Vordergrund steht. Hierfür wird bewusst eine Verschlechterung des Klassifikationsergebnisses in Kauf genommen.

Diskriminanzfunktionen und Entscheidungsregeln ergeben sich aus dem Zusammenwirken aller gefundenen Regeln. Eine explizite Darstellung ist zwar möglich, jedoch nicht angestrebt, da hierdurch die Interpretierbarkeit der Regeln verloren geht.

Den Verlauf der Diskriminanzfunktionen im aggregierten, zweidimensionalen Merkmalsraum zeigt Bild 2.13 anhand von Datensatz A*. Diese trennen allerdings schlecht (die Klassen 4, 5 und 6 werden nur unzureichend erkannt), da das Verfahren mehr als zwei Merkmale für hohe Klassifikationsgüten benötigt.

Von Vorteil erweist sich dieses Verfahren bei der Implementierung in Echtzeitsysteme, Anhang 7.4 stellt einen schnellen und effizienten Algorithmus vor. Die für eine Klassifikation notwendige Anzahl an Speicherplätzen und Multiplikationen lässt sich nicht pauschal angeben, da die entworfenen Regelbasen keine einheitliche Struktur aufweisen. Die für Datensatz A entworfene Beispiel-Regelbasis findet sich in Anhang 7.4.

2.9.3 Künstliche Neuronale Netze (KNN)

Künstliche Neuronale Netze sind ein Zusammenschluss parallel arbeitender Funktionseinheiten, die in Anlehnung an biologische neuronale Netze Neuronen oder künstliche Neuronen genannt werden [71, 167, 168]. Ein Neuron verknüpft w Eingangssignale $\mathbf{e} = (e_1, \dots, e_w)^T$ mit Gewichten $\mathbf{g} = (g_1, \dots, g_w)^T$ und Bias g_0 (Bild 2.14). Beliebte Verknüpfungen sind

$$z_n = \mathbf{e}^T \mathbf{g} + g_0 \quad (\text{gewichtete Summe}), \quad (2.58)$$

$$z_n = \sqrt{g_0(\mathbf{e} - \mathbf{g})^T(\mathbf{e} - \mathbf{g})} \quad (\text{korrigierter euklidischer Abstand}). \quad (2.59)$$

Die Ergebnisse z_n werden einer Aktivierungsfunktion $f(z_n) = f(z_n(\mathbf{e}; \mathbf{g}))$ zugeführt, wie z. B.

$$f(z_n) = z_n \quad (\text{lineare Aktivierungsfunktion}), \quad (2.60)$$

$$f(z_n) = \frac{1}{1 + e^{-z_n}} \quad (\text{Sigmoid-Aktivierungsfunktion}), \quad (2.61)$$

$$f(z_n) = \frac{2}{1 + e^{-2z_n}} - 1 \quad (\text{Tansig-Aktivierungsfunktion}), \quad (2.62)$$

$$f(z_n) = e^{-z_n^2} \quad (\text{Radial-Basis-Aktivierungsfunktion}). \quad (2.63)$$

Deren Funktionswert bestimmt das Ausgangssignal $a(\mathbf{e}; \mathbf{g}) = f(z_n) = f(\mathbf{e}; \mathbf{g})$ des Neurons.

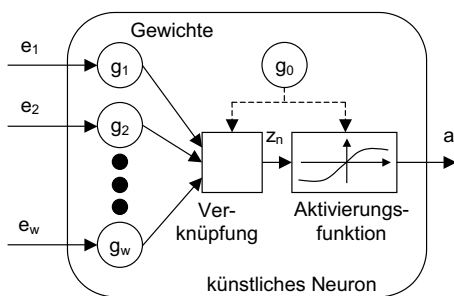


Bild 2.14: Aufbau eines einzelnen Neurons

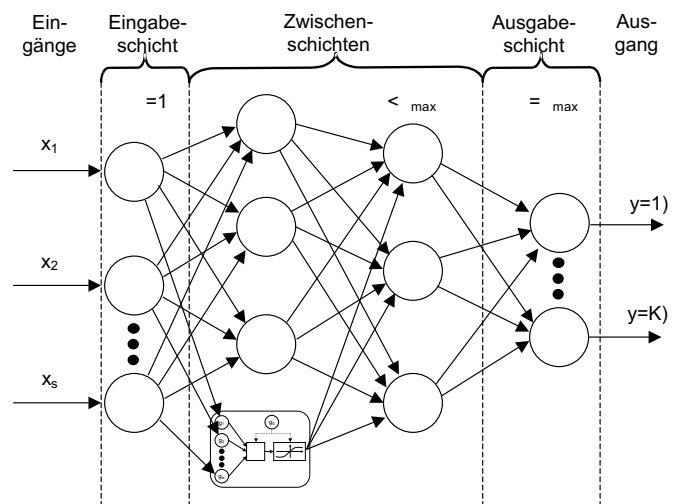


Bild 2.15: Schematischer Aufbau eines Künstlichen Neuronales Netzes mit zwei Zwischenschichten

Das Künstliche Neuronale Netz ergibt sich durch das Verknüpfen einzelner Neuronen, die oftmals in Schichten $\alpha \in \{1 \dots \alpha_{\max}\}$ angeordnet werden. Meist gibt es dabei gemäß Bild 2.15 eine Eingabeschicht ($\alpha = 1$), eine oder mehrere Zwischenschichten ($2 < \alpha < \alpha_{\max}$) und eine Ausgabeschicht ($\alpha = \alpha_{\max}$). Das Ausgangssignal des j -ten Neurons der Schicht α wird mit $a_j^{(\alpha)}$ bezeichnet. Die Verknüpfung der Ausgangssignale eines Neurons mit den Eingängen anderer Neuronen beschreibt die Netztopologie. Es existieren vorwärtsgerichtete Verbindungen (Neuronen der Schicht α erhalten nur Eingangssignale aus der Schicht $1, \dots, \alpha - 1$), laterale (Neuronen der Schicht α erhalten Eingangssignale aus dergleichen Schicht α) und rückgekoppelte (Neuronen können von Neuronen der Schichten $\alpha + 1, \dots, \alpha_{\max}$ Eingangssignale erhalten).

Für Klassifikationsaufgaben kommen oft vorwärtsgerichtete Multilayer-Perzeptron-Netze (MLP) und Radial-Basis-Funktion-Netze (RBF) zum Einsatz. Ein- und Ausgabeschicht bestehen meist aus Neuronen mit linearer Aktivierungsfunktion nach (2.60) und gewichteten Eingangssignalen nach (2.58), die Zwischenschichten aus Perzeptron- oder Radial-Basis-Funktion-Neuronen. Ein Neuron der Schicht α erhält Ausgabewerte aller Neuronen der $(\alpha - 1)$ -ten Schicht als Eingang. Die Neuronen der Eingabeschicht haben nur einen Eingang, und jedes Neuron erhält genau ein Merkmal als Eingangssignal. Es sind somit s Neuronen nötig, die dann die Eingangssignale auf die nachfolgenden Schichten verteilen:

$$e_j^{(1)} = x_j, \quad j \in \{1, \dots, s\}. \quad (2.64)$$

Die Ausgabe des Netzes ergibt sich aus der Ausgabe der Neuronen der Ausgabeschicht. In [75, 114] konnte gezeigt werden, dass zur Klassifikation biometrischer Signale K Ausgabeneuronen von Vorteil sind. Das j -te Neuron kann dann die Zugehörigkeit μ zur Klasse j beschreiben. Die Klassenschätzung ergibt sich zu

$$\hat{y}(\mathbf{x}) = \operatorname{argmax}_{1 \leq j \leq K} \mu(y = j) = \operatorname{argmax}_{1 \leq j \leq K} a_j^{(\alpha_{\max})}(\mathbf{x}). \quad (2.65)$$

Das Perzeptron-Neuron wurde erstmals in [169] beschrieben. Es wichtet seine Eingangssignale nach (2.58) und verwendet oft Tansig-Funktionen nach (2.62) als Aktivierungsfunktion. Die Ausgangsgröße lautet dann

$$a(\mathbf{e}; \mathbf{g}) = \frac{2}{1 + e^{-2(\mathbf{e}^T \mathbf{g} + g_0)}} - 1. \quad (2.66)$$

Das Radial-Basis-Funktion-Neuron ermittelt die Distanz zwischen Eingang und Gewichten nach (2.59) und verwendet (2.63) als Aktivierungsfunktion:

$$a(\mathbf{e}; \mathbf{g}) = e^{-g_0(\mathbf{e}-\mathbf{g})^T(\mathbf{e}-\mathbf{g})}. \quad (2.67)$$

Mit zunehmender Distanz sinkt der ausgegebene Funktionswert. Entsprechend sind RBF-Neuronen unanfällig gegenüber Ausreißern. Aufgrund der Distanzberechnung ist für RBF-Netze eine Intervalltransformation der Merkmale zu bevorzugen, um einen skaleninvarianten Klassifikator zu erhalten.

Um MLPs oder RBFs für Klassifikationsaufgaben zu entwerfen, sind vorab Art und Anzahl der Neuronen sowie die Anzahl der Schichten festzulegen. In der vorliegenden Arbeit werden stets Ein- und Ausgabeschicht mit linearer Aktivierungsfunktion sowie eine Zwischenschicht mit n_n Perzeptron- bzw. Radial-Basis-Funktion-Neuronen versehen. Die Netze lernen durch das Anpassen der Gewichte g_i . Dies geschieht in der Regel durch den Backpropagation-Algorithmus. Ausgehend von zufällig initialisierten Gewichten propagiert der Algorithmus die Netzausgabe rückwärts durch das Netz und versucht durch das Ändern der Gewichte den Ausgangsfehler zu minimieren. Dieses Optimierungsproblem kann durch Gradientenverfahren gelöst werden, besonders geeignet ist das Levenberg-Marquardt-Verfahren, das Polynome zweiter Ordnung aus dem Gradienten schätzt und so sehr schnell konvergiert. Einen detaillierten Überblick über mögliche Lernverfahren gibt [215]. Bild 2.16 stellt die Diskriminanzfunktionen eines MLP-Netzes über dem Benchmarkdatensatz A^* dar. Es kommen $n_n = 12$ Zwischenschichtneuronen mit Bias und Tansig-Aktivierungsfunktion sowie K Ausgangsneuronen mit linearer Aktivierungsfunktion zum Einsatz. Die Implementierung geschieht mit der MATLAB-Toolbox Neural Networks [106]. In dem durch die Daten abgedeckten Bereich des Merkmalsraums ergeben sich sechs Bereiche, die die Klassen gut trennen.

Zum Entwurf eines MLP-Netzes sind $n_n(s + K + 1) + K$ Speicherplätze¹⁰ und zum Auswerten $n_n(s + K + 1)$ Multiplikationen¹¹ sowie n_n Berechnungen der Exponentialfunktion notwendig. Eine

¹⁰Eingabeschicht: keine Speicherplätze, Zwischenschicht: $(s + 1)$ pro Neuron, Ausgabeschicht: $(n_n + 1)$ pro Neuron (u. U. weitere $2s$ Speicherplätze für Intervalltransformation)

¹¹Zwischenschicht: $(s + 2)$ pro Neuron, Ausgabeschicht: n_n pro Neuron

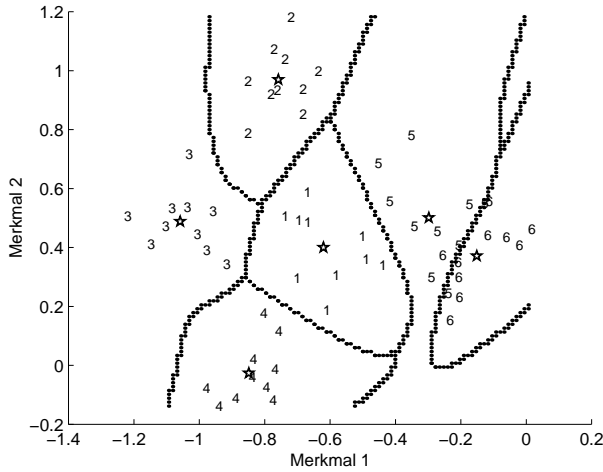


Bild 2.16: Diskriminanzfunktionen eines MLPs mit $n_n = 12$ Neuronen, Benchmarkdatensatz A^*

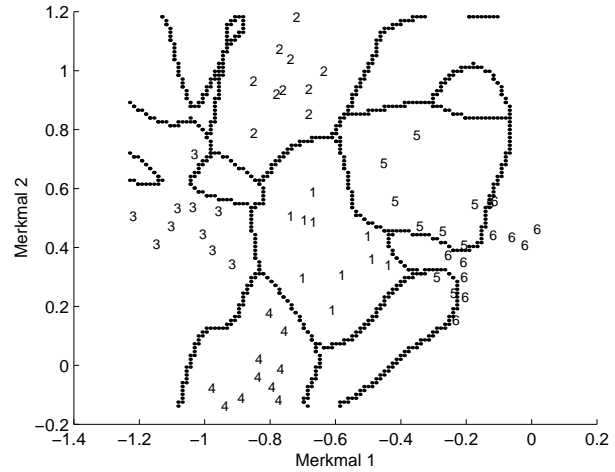


Bild 2.17: Diskriminanzfunktionen eines überlernten MLPs mit $n_n = 30$ Neuronen, Benchmarkdatensatz A^*

Klassifikationsentscheidung ist somit speicher- und rechenzeitaufwändig. Aufgrund der hohen Parameteranzahl konvergieren die Verfahren zudem oft in Nebenminima - das Netz ist überlernt¹². Die Konsequenzen sind ein Auswendiglernen des Datensatzes - die Generalisierungsfähigkeit des Netzes lässt stark nach. In diesem Fall sind die Diskriminanzfunktionen zu sehr an die Lerndaten angepasst. In Bereichen, in denen keine Lerndaten existieren, ergeben sich jedoch unsinnige Verläufe (Bild 2.17). Weiterhin ist der durch Daten abgedeckte Bereich des Merkmalsraums in viele Bereiche geteilt, in denen verschiedene Klassenzuweisungen stattfinden. Dies führt bei kleinen Änderungen in den Merkmalen zu Fehlklassifikationen. Eine Beschränkung der Parameter durch Verwendung weniger Zwischenschichtneuronen und einer Auswahl von Merkmalen leistet Abhilfe. Ein weiterer Nachteil sind schlecht interpretierbare Ergebnisse.

2.9.4 k-Nearest-Neighbour-Klassifikatoren (k-NN)

Ein k-Nearest-Neighbour-Klassifikator beurteilt die Klassenzugehörigkeit eines unbekanntem Datentupels \mathbf{x} anhand der Klassenzugehörigkeiten y_i seiner Nachbarn $\mathbf{x}_i, i = 1, \dots, n$ im Lerndatensatz. Der Parameter k bestimmt die Anzahl zu betrachtender Nachbarn. Für $k=1$ wird \mathbf{x} die Klassenzugehörigkeit seines nächsten Nachbarn zugewiesen. Die Klassenschätzung¹³ lautet

$$\hat{y}(\mathbf{x}) = y_i \quad \text{für} \quad d(\mathbf{x}, \mathbf{x}_i) < d(\mathbf{x}, \mathbf{x}_j), \quad j = 1, \dots, n, \quad i \neq j, \quad (2.68)$$

für $k > 1$ folgt eine Zuweisung zur häufigsten Klasse:

$$\hat{y}(\mathbf{x}) = \underset{1 \leq j \leq K}{\operatorname{argmax}} \operatorname{card}(\{l | y_{i_l} = j, \quad l = 1, \dots, k\}) \quad \text{für} \quad d(\mathbf{x}, \mathbf{x}_{i_1}) < \dots < d(\mathbf{x}, \mathbf{x}_{i_k}) < d(\mathbf{x}, \mathbf{x}_j),$$

mit $j = 1, \dots, n, \quad j \notin \{i_1, \dots, i_k\}$. (2.69)

Im Allgemeinen wird zur Distanzberechnung die euklidische Metrik

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2)} \quad (2.70)$$

¹²Ein Überlernen tritt ebenfalls ein, wenn das globale Optimum zwar gefunden, aber die Werte der Datentupel auswendig gelernt wurden.

¹³Gilt nicht die strenge Ungleichheit, so erhält die Klasse mit der kleineren Klassenbezeichnung Vorrang.

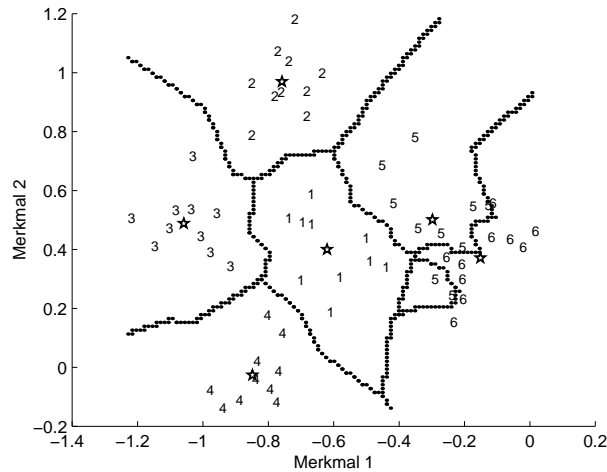


Bild 2.18: Diskriminanzfunktionen eines k-NN-Klassifikators mit $k=1$, Benchmarkdatensatz A^*

verwendet, eine Intervalltransformation nach (2.38) ist demnach zu empfehlen. Andere Metriken sind ebenfalls möglich (Mahalanobis, Manhattan, etc.) [38].

Das Anlernen des Klassifikators geschieht durch das Abspeichern aller Datentupel des Lerndatensatzes \mathbf{x}_i mit Klassenzugehörigkeiten y_i und ggf. durch das Parametrieren einer Intervalltransformation nach (2.38) und eines Distanzmaßes. Der Parameter k wird heuristisch festgelegt. Bild 2.18 stellt die Diskriminanzfunktionen eines k-NN-Klassifikators über dem Benchmarkdatensatz A^* dar. Es zeigen sich sieben Bereiche für die sechs Klassen. Alle Datentupel werden richtig klassifiziert, da Lern- und Testdaten identisch sind und somit jedes Datentupel sich selbst als nächsten Nachbarn betrachtet.

Es wird nach $k = 1$ Nachbarn gesucht, die Implementierung erfolgt mit Hilfe der MATLAB-Toolbox "SVM and Kernel Methods Matlab Toolbox" [30].

k-Nearest-Neighbour-Klassifikatoren sind einfach anzulernen, bieten einen leichten Zugang und sind auch für nicht kompakte Klassengebiete geeignet. Jedoch weisen sie Instabilitäten im Bereich von Klassengrenzen auf, insbesondere dann, wenn k klein ist oder Ausreißer im Datensatz vorhanden sind. Eine Implementierung in Echtzeitsysteme ist problembehaftet. Bei Verwendung der euklidischen Distanz sind $(n + 2) \cdot s + 1$ Speicherplätze¹⁴ zu belegen und für jede Klassifikation $(n + 1) \cdot s$ Multiplikationen¹⁵ durchzuführen. Aufgrund der Unabhängigkeit von K eignet sich der Klassifikator für Probleme mit vielen Klassen. Die Verwendung anderer Metriken erhöht den Aufwand erheblich (Mahalanobis Distanz: $ns(s + 1)$ Multiplikationen). Auch hier empfiehlt sich die Einschränkung des Merkmalsraums.

2.9.5 Support-Vektor-Maschinen (SVM)

Ausgangspunkt für die in [199, 200] eingeführten Support-Vektor-Maschinen sind Zweiklassenprobleme mit den Klassenbezeichnungen $y_i = -1$ (Klasse 1) oder $y_i = 1$ (Klasse 2) für das i -te Datentupel \mathbf{x}_i . Der Grundgedanke besteht darin, eine Hyperebene zur optimalen Trennung der beiden Klassen zu finden. Da

¹⁴ $2s$ Speicherplätze für die Intervalltransformation, ns Speicherplätze um den Lerndatensatz zu speichern und ein Speicherplatz, um k festzulegen

¹⁵ s Multiplikationen für die Intervalltransformation, ns Multiplikationen für die Distanzberechnung, eine explizite Berechnung der Wurzel ist aufgrund der Monotonie nicht nötig.

jedoch eine lineare Trennung vieler Probleme nicht möglich ist, wird der Merkmalsraum vorab in einen höherdimensionalen Raum transformiert. Diese frei wählbare Transformation Φ ist i. A. nichtlinear:

$$\mathbf{x}_\phi = \Phi(\mathbf{x}). \quad (2.71)$$

Die Trennung im hochdimensionalen Raum findet durch eine Hyperebene

$$\mathbf{w}^T \mathbf{x}_\phi + w_0 = 0 \quad (2.72)$$

statt, das Vorzeichen des Ausdrucks $(\mathbf{w}^T \mathbf{x}_\phi + w_0)$ gibt Auskunft über die Klassenzugehörigkeit eines Datentupels:

$$\hat{y} = \begin{cases} 1 & \text{wenn } \mathbf{w}^T \mathbf{x}_\phi + w_0 \geq 0 \\ -1 & \text{sonst.} \end{cases} \quad (2.73)$$

Zur Berechnung von \mathbf{w} wird zunächst von einer fehlerfreien Klassifikation aller Datentupel $\mathbf{x}_{\phi,i}$ ausgegangen:

$$y_i(\mathbf{w}^T \mathbf{x}_{\phi,i} + w_0) - 1 \geq 0, \quad i = 1, \dots, n. \quad (2.74)$$

Die zur Trennebene nächstgelegenen Datentupel erfüllen hierbei die Gleichheit und haben somit einen Abstand von $1/\|\mathbf{w}\|$ zur Trennebene. Nun wird versucht, die Abstände dieser Datentupel zur Trennebene zu maximieren

$$Q_a = \frac{2}{\|\mathbf{w}\|} \rightarrow \max_{\mathbf{w}}. \quad (2.75)$$

Mit den Nebenbedingungen (2.74) führt der Lagrange'sche Lösungsansatz mit den Lagrange-Multiplikatoren λ_i zur Lagrange-Funktion

$$Q_{\text{Lag}} = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \lambda_i (y_i(\mathbf{w}^T \mathbf{x}_{\phi,i} + w_0) - 1), \quad \lambda_i \geq 0 \quad (2.76)$$

mit

$$Q_{\text{Lag}} \rightarrow \min_{\substack{\mathbf{w}, w_0 \\ \lambda_i \geq 0 \\ \frac{\partial Q_{\text{Lag}}}{\partial \lambda_i} = 0}} \quad \text{bzw.} \quad Q_{\text{Lag}} \rightarrow \max_{\substack{\mathbf{w}, w_0 \\ \lambda_i \geq 0 \\ \frac{\partial Q_{\text{Lag}}}{\partial \mathbf{w}} = 0, \frac{\partial Q_{\text{Lag}}}{\partial w_0} = 0}}. \quad (2.77)$$

Ableiten nach \mathbf{w} und w_0 , Nullsetzen und Einsetzen in (2.76) liefert folgendes Optimierungsproblem

$$Q = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j \mathbf{x}_{\phi,i} \mathbf{x}_{\phi,j} \lambda_i \lambda_j \rightarrow \max_{\lambda_1, \dots, \lambda_n} \quad (2.78)$$

$$\text{mit } 0 \leq \lambda_i, \quad i = 1, \dots, n \quad (2.79)$$

$$\sum_{i=1}^n \lambda_i y_i = 0. \quad (2.80)$$

Das Aufgeben der Nebenbedingung der fehlerfreien Klassifikation führt nach [27] zu einem Ersetzen von (2.79) durch

$$0 \leq \lambda_i \leq C, \quad i = 1, \dots, n. \quad (2.81)$$

Je größer der frei wählbare Parameter C gewählt wird, desto weniger Klassifikationsfehler toleriert der Algorithmus. Das Problem kann z. B. durch den Sequential-Minimal-Optimization-Algorithmus gelöst werden [142]. Alle zu $0 \leq \lambda_i \leq C$ gehörigen $\mathbf{x}_{\phi,i}$ werden als Support-Vektoren bezeichnet, ihre Indizes seien in einer Indexmenge \mathcal{S} enthalten

$$\mathcal{S} = \{i | 0 < \lambda_i \leq C\} \quad \text{mit} \quad \text{card}(\mathcal{S}) = n_{\text{sup}}. \quad (2.82)$$

Der Summand w_0 und Richtungsvektor \mathbf{w} ergeben sich dann aus

$$w_0 = \frac{1}{y_i} - \mathbf{w}^T \mathbf{x}_{\phi,i} \quad \text{für } i \text{ beliebig mit } 0 < \lambda_i < C, \quad (2.83)$$

$$\mathbf{w}^T \mathbf{x}_\phi = \sum_{i \in \mathcal{S}} \overbrace{\lambda_i y_i \mathbf{x}_{\phi,i}}^{\mathbf{w}^T} \mathbf{x}_\phi = \sum_{i \in \mathcal{S}} \lambda_i y_i \Phi(\mathbf{x}_i) \Phi(\mathbf{x}). \quad (2.84)$$

Von Vorteil in (2.84) ist, dass \mathbf{w} explizit berechnet werden kann. Da die Berechnung der Transformation Φ jedoch meist aufwändig ist, kann sie so gewählt werden, dass gilt:

$$\Phi(\mathbf{x}_i) \Phi(\mathbf{x}) = K(\mathbf{x}_i, \mathbf{x}). \quad (2.85)$$

$K(\mathbf{x}, \mathbf{y})$ bezeichnet hierbei eine Kernfunktion, wodurch (2.84) alternativ mit

$$\mathbf{w}^T \mathbf{x}_\phi = \sum_{i \in \mathcal{S}} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) \quad (2.86)$$

berechnet werden kann. (2.86) bietet den Vorteil, dass die Transformation in den hochdimensionalen Raum nicht mehr explizit ausgeführt werden muss, hingegen lässt sich \mathbf{w} i. A. nur noch implizit als $\mathbf{w}^T \mathbf{x}_\phi$ darstellen, wenn (2.85) in (2.84) eingesetzt wird. Die Klassenschätzung geschieht nun gemäß

$$\hat{y}(\mathbf{x}) = \begin{cases} -1 & \text{für } \sum_{i \in \mathcal{S}} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) + w_0 < 0 \\ 1 & \text{für } \sum_{i \in \mathcal{S}} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) + w_0 \geq 0. \end{cases} \quad (2.87)$$

Für die Wahl des Kerns existieren keine allgemeingültigen Kriterien, sie muss der Problemstellung angepasst werden. Ein in der Praxis oft eingesetzter Kern ist der Polynomkern zweiter Ordnung¹⁶ mit

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + 1)^2 \quad \text{bzw.} \quad \Phi(\mathbf{x})^T = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2). \quad (2.88)$$

Weitere Kerne sind [40] zu entnehmen.

Der Übergang vom Zweiklassenproblem auf ein Mehrklassenproblem geschieht entweder durch das Gegenüberstellen einer Klasse gegen alle anderen (engl.: One-against-all¹⁷) oder durch das Betrachten aller paarweisen Klassenkombinationen (engl.: One-against-one) sowie dem Verschmelzen der erhaltenen Ergebnisse. Letztere Methode erzeugt explizit die in (2.57) dargestellten Ungleichungssysteme. Die Beschreibung der genauen Vorgehensweise folgt in Abschnitt 3.3.4.

Bei der Anwendung der SVM zeigt sich, dass auch hier eine Intervalltransformation nach (2.38) von Vorteil ist.

Bild 2.19 stellt die Diskriminanzfunktionen einer SVM über dem Benchmarkdatensatz A^* dar. Verwendet wurde ein Polynomkern zweiter Ordnung bei Betrachtung von paarweisen Klassenkombinationen und $C = 1000$. Die Implementierung erfolgte mit der MATLAB-Toolbox "SVM and Kernel Methods Matlab Toolbox" [30]. Es sind sechs Bereiche zu sehen, die den Datensatz gut klassifizieren. Supportvektoren sind durch Kreise gekennzeichnet, Supportvektoren können für die Zweiklassenprobleme mehrfach auftreten.

¹⁶Die Anzahl transformierter Merkmale beträgt $s_\Phi = \frac{1}{2}(s^2 + 3s + 2)$, die Anzahl der Multiplikationen zur Transformation beträgt $n_r = s(s + 1)$.

¹⁷One-against-all liefert i. A. schlechtere Ergebnisse als One-against-one.

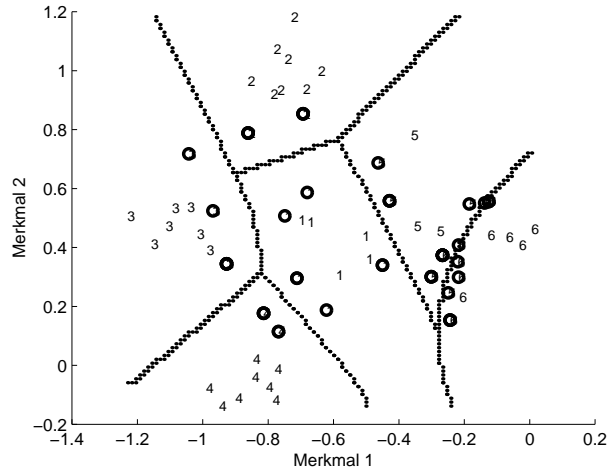


Bild 2.19: Diskriminanzfunktionen einer SVM, Benchmarkdatensatz A^* , Kern aus (2.88), Support-Vektoren durch "o" gekennzeichnet

Zur Implementierung einer SVM zum Unterscheiden von zwei Klassen sind bei Verwendung eines Polynomkerns zweiten Grades $n_{\text{sup}}(s + 1) + 1 + 2s$ Speicherplätze¹⁸ zu speichern und zur Auswertung $n_{\text{sup}}(n_K + 1) + s$ Multiplikationen¹⁹ zu tätigen. n_K beschreibt hierbei die Anzahl der Multiplikationen zur Berechnung des Kerns, für $K(\mathbf{x}, \mathbf{y})$ aus (2.88) gilt $n_K = s + 1$. Der Verzicht auf die Kernfunktion durch Verwendung von \mathbf{w} erfordert $s_{\Phi} + 1 + 2s$ Speicherplätze und $s_{\Phi} + n_r + s$ Multiplikationen²⁰.

Im Mehrklassenfall ($K > 2$) sind K -mal (one-against-all) bzw. $1/2 \cdot K(K - 1)$ -mal (one-against-one) mehr Speicherplätze und Multiplikationen nötig (n_{sup} sei für alle Zweiklassenprobleme gleich). Auch hier ist somit eine Merkmalsauswahl anzustreben. Die Echtzeitfähigkeit des Verfahrens ist stark eingeschränkt, aber je nach Problemstellung zu bewerten. Nachteilig wirkt sich der schlecht interpretierbare Zugang aus (Wahl der Support-Vektoren, große Anzahl von Semiparametern).

2.9.6 Approximative Maximum-Likelihood-Schätzung (AML)

Bei der Konstruktion der Entscheidungsregel nach dem Maximum-Likelihood (ML)-Prinzip wird für jede Klasse j eine Verteilungsannahme getroffen, die durch eine klassenspezifische Dichtefunktion f_j mit den Parametern θ_j beschrieben wird. Unter der Voraussetzung stochastisch unabhängiger Datentupel ergibt sich die Likelihood-Funktion zu

$$L(\theta_1, \dots, \theta_K, y; \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{x}) = f_y(\mathbf{x}; \theta_y) \prod_{j=1}^K \prod_{i: y_i=j} f_j(\mathbf{x}_i; \theta_j). \quad (2.89)$$

Diese Funktion ist bezüglich $\theta_1, \dots, \theta_K$ und der Klassenzuordnung y des zu klassifizierenden Datentupels \mathbf{x} zu maximieren²¹. Die Zuordnung des Datentupels \mathbf{x} zu einer Klasse j beeinflusst somit die

¹⁸Für n_{sup} Support-Vektoren i sind jeweils ein $\lambda_i \cdot y_i$ sowie s Merkmale, zusätzlich w_0 und weitere $2s$ Speicherplätze für die Intervalltransformation notwendig.

¹⁹ $n_K + 1$ Multiplikationen für die Auswertung jedes Support-Vektors, s Multiplikationen für die Intervalltransformation

²⁰ n_r Multiplikationen zur Transformation, s_{Φ} Multiplikationen zur Auswertung von $\mathbf{w}^T \mathbf{x}_{\Phi}$, s Multiplikationen für die Intervalltransformation

²¹Im Gegensatz zur Klassifikation ist bei der Clustering von Daten die Likelihood-Funktion über den θ_j und allen y_i zu maximieren.

Parameter θ_j der auf der Basis der Lerndaten \mathbf{x}_i bestimmten Dichtefunktion f_j . Die ML-Schätzung für die Klassenzugehörigkeit y von \mathbf{x} ergibt sich dann zu

$$\hat{y}^{ML}(\mathbf{x}) = \underset{\substack{y \in \{1, \dots, K\} \\ \theta_1 \in \mathbb{R}^{p_1}, \dots, \theta_K \in \mathbb{R}^{p_K}}}{\operatorname{argmax}} f_y(\mathbf{x}; \theta_y) \prod_{j=1}^K \prod_{i: y_i=j} f_j(\mathbf{x}_i; \theta_j). \quad (2.90)$$

Es bezeichnet p_j die Dimension des Parametervektors θ_j der Verteilungsdichte der Klasse j . Unter der Voraussetzung $n_j \gg \max_{1 \leq j \leq K} p_j$ werden die Schätzungen für θ_j nur noch unwesentlich von dem neu zu klassifizierenden Datentupel \mathbf{x} beeinflusst und können so durch die ML-Schätzer $\hat{\theta}_j$ ersetzt werden. Wird zudem für alle Klassen dieselbe Familie von Verteilungen unterstellt, so sind alle p_j und f_j gleich ($f_j = f$). Mit dieser Näherung zerfällt das Problem in eine Auswertung bezüglich der Werte der klassenspezifischen Dichten an der Stelle \mathbf{x} :

$$\hat{y}^{ML}(\mathbf{x}) \approx \underset{j \in \{1, \dots, K\}}{\operatorname{argmax}} f(\mathbf{x}; \hat{\theta}_j) \quad \text{für } n_j \gg \max_{1 \leq j \leq K} p_j. \quad (2.91)$$

Eine ausführliche Herleitung findet sich in [154].

Im Fall der K -Klassen-Normalverteilung ergibt sich die geschätzte Klassenzugehörigkeit durch die approximative ML-Entscheidungsregel²² (AML-Regel):

$$\hat{y}(\mathbf{x}) = \underset{1 \leq j \leq K}{\operatorname{argmax}} f(\mathbf{x}; \bar{\mathbf{x}}_j, \hat{\Sigma}_j) \quad (2.92)$$

$$= \underset{1 \leq j \leq K}{\operatorname{argmax}} \frac{1}{(2\pi)^{\frac{s}{2}} \sqrt{\det \hat{\Sigma}_j}} e^{-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_j)^T \hat{\Sigma}_j^{-1} (\mathbf{x} - \bar{\mathbf{x}}_j)} \quad (2.93)$$

$$= \underset{1 \leq j \leq K}{\operatorname{argmin}} \left(-2 \ln f(\mathbf{x}; \bar{\mathbf{x}}_j, \hat{\Sigma}_j) \right) \quad (2.94)$$

$$= \underset{1 \leq j \leq K}{\operatorname{argmin}} \ln \det \hat{\Sigma}_j + (\mathbf{x} - \bar{\mathbf{x}}_j)^T \hat{\Sigma}_j^{-1} (\mathbf{x} - \bar{\mathbf{x}}_j). \quad (2.95)$$

Bild 2.20 stellt die Diskriminanzfunktionen des AML-Klassifikators über dem Benchmarkdatensatz A* dar. Die Diskriminanzfunktionen gehen aus den Schnittlinien der Dichtefunktionen in Bild 2.21 hervor. Je höher dabei das Maximum der Dichtefunktion liegt, desto kleiner ist der Einzugsbereich der entsprechenden Klasse (z. B. Klasse 6).

Die notwendigen Parameter $\hat{\theta}_j$ zur Klassifikation der Klasse j setzen sich aus den Klassenmittelwerten $\bar{\mathbf{x}}_j$ und klassenspezifischen Kovarianzmatrizen $\hat{\Sigma}_j$ zusammen. Die Klassifikation eines Datentupels \mathbf{x} erfordert somit $K(s(s+3)/2 + 1)$ Speicherplätze²³, und mit

$$\mathbf{a}^T \mathbf{S} \mathbf{a} = \sum_{i=1}^s a_i^2 s_{ii} + 2 \sum_{i=1}^s \sum_{j=i+1}^s a_i a_j s_{ij} \quad (2.96)$$

werden $K \cdot (1 + s(s+1))$ Multiplikationen²⁴ benötigt. Die Auswertung der Exponentialfunktion entfällt. Hier ist ebenfalls eine Reduktion der Merkmale anzustreben, da sowohl die Anzahl der Speicherplätze als auch die der Multiplikationen mit s^2 steigt.

²²Bei der Umformung von (2.92) zu (2.95) sind zwar die Argumente der arg-Funktionen nicht gleich, deren Ergebnisse sind es jedoch. Der Vorzeichenwechsel des Arguments in (2.94) wird kompensiert durch das Ersetzen der argmax- durch die argmin-Funktion.

²³pro Klasse: 1 Speicherplatz für $\ln \det \hat{\Sigma}_j$, s Speicherplätze für $\bar{\mathbf{x}}_j$ und $\frac{1}{2}s(s+1)$ Speicherplätze für $\hat{\Sigma}_j^{-1}$ (hier gilt $\sigma_{ij} = \sigma_{ji}$)

²⁴Für den ersten Term aus (2.96) ergeben sich $2s$ Multiplikationen, für den zweiten sind $s(s-1) + 1$ Multiplikationen nötig. Für den Sonderfall $s=1$ sinkt die Anzahl um 1.

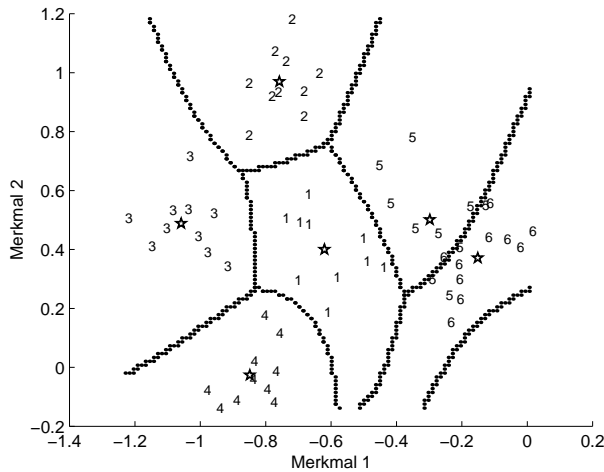


Bild 2.20: Diskriminanzfunktionen des AML-Klassifikators, Sterne bezeichnen Klassenmittelpunkte (Benchmarkdatensatz A^*)

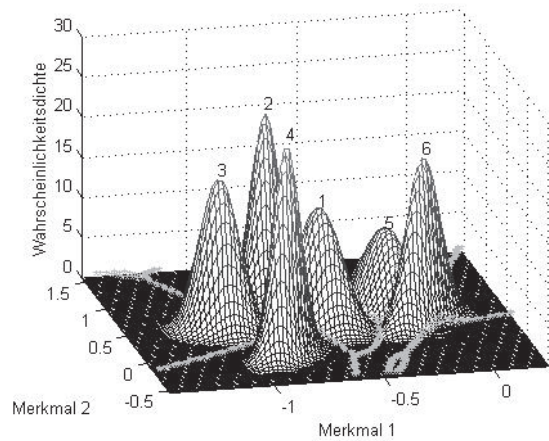


Bild 2.21: Wahrscheinlichkeitsdichten und Diskriminanzfunktionen im Merkmalsraum anhand von Benchmarkdatensatz A^*

Eine Möglichkeit zur Verminderung der Speicherplatzanzahl ergibt sich durch die Überführung von (2.95) in eine Distanz d . Dies geschieht durch Ersetzen der Kovarianzmatrix durch eine einheitliche Matrix für alle Klassen:

$$\hat{\Sigma} = \hat{\Sigma}_j = \mathbf{S}^{-1}. \tag{2.97}$$

Es gilt dann

$$\hat{y}(\mathbf{x}) = \operatorname{argmin}_{1 \leq j \leq K} d^2(\mathbf{x}, \bar{\mathbf{x}}_j) = \operatorname{argmin}_{1 \leq j \leq K} (\mathbf{x} - \bar{\mathbf{x}}_j)^T \mathbf{S} (\mathbf{x} - \bar{\mathbf{x}}_j) = \operatorname{argmin}_{1 \leq j \leq K} (\mathbf{x} - \bar{\mathbf{x}}_j)^T \hat{\Sigma}^{-1} (\mathbf{x} - \bar{\mathbf{x}}_j). \tag{2.98}$$

Gebräuchlich sind Klassifikatoren auf der Basis der Mahalanobis Distanz mit (2.99) und der euklidischen Distanz mit (2.100):

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T, \tag{2.99}$$

$$\hat{\Sigma} = I_{(s,s)}. \tag{2.100}$$

Die Anzahl notwendiger Speicherplätze beträgt bei (2.99) $ks + s(s + 1)/2$, bei (2.100) sind ks Speicherplätze nötig. Die Klassifikation mittels dieser Distanzmaße liefert jedoch i. A. schlechtere Ergebnisse als der AML-Klassifikator und wird für die folgenden Untersuchungen außer Acht gelassen.

2.10 Vergleich

2.10.1 Validierungstechniken

Um die Klassifikationsgüte eines Verfahrens quantitativ zu beurteilen, bieten sich statistische Fehlerabschätzungen an. Da der Datensatz jedoch meistens erforderliche Voraussetzungen verletzt (Wahrscheinlichkeitsverteilung, stochastische Unabhängigkeit, etc.), kommen Verfahren zum Einsatz, die Datensätze

konstruieren und Klassifikatoren auf diese anwenden. Diese Verfahren setzen lediglich voraus, dass die Datentupel der gleichen Verteilung entstammen. Die bekanntesten Validierungstechniken sind [77, 186]:

1. Reklassifikation oder Testen über Lerndaten: Das angelernte Verfahren wird sowohl über seinen Lerndaten angelernt als auch getestet. Die Reklassifikation neigt dazu, den Klassifikationsfehler zu unterschätzen.
2. Testen über Testdaten: Das angelernte Verfahren wird auf einem unbekanntem Datensatz getestet. Das Ergebnis ist repräsentativ (sofern der Datenumfang groß genug ist), benötigt wird jedoch ein unbekannter Testdatensatz, der oft nicht zur Verfügung steht. Zudem führt das Testen über Testdaten zu hohen Informationsverlusten, da dem Anlernvorgang die Informationen des Testdatensatzes vorenthalten werden.
3. Crossvalidierung: Die v -fache Crossvalidierung nutzt $v - 1$ Teile des Datensatzes zum Anlernen (mit einem Umfang von $n_{\text{cross}} = n/(v - 1)$) und einen Teil zum Testen. Das Verfahren wird v -mal wiederholt, bis jedes Datentupel genau einmal zum Testen verwendet wurde. Die Aufteilung des Datensatzes geschieht zufällig. Fehlerquoten der Crossvalidierung sind im Schnitt zu hoch, da der Umfang des Lerndatensatzes verkleinert ist und somit Informationen verloren gehen. Zudem ist das Verfahren sehr rechenaufwändig.
4. Auslass-Methode oder Leave-One-Out: Die Auslass-Methode ist ein Spezialfall der Crossvalidierung für $v = n$.
5. Bootstrap: Die Bootstrap-Methode zieht mit Zurücklegen $n_{\text{Boot}} = n$ Datentupel aus dem Datensatz und verwendet diese zum Anlernen. Alle nicht gezogenen Datentupel des Datensatzes vereint das Verfahren zu einem Testdatensatz. Die Bootstrap-Methode liefert ebenfalls eine pessimistische Schätzung. Um die Schätzung zu verbessern, bietet sich die 632-Bootstrap-Methode nach [49] an. Diese berechnet die Klassifikationsgüte ξ_G nach

$$\xi_G = 0.632\xi_{G,T} + 0.368\xi_{G,\text{all}} \quad (2.101)$$

aus der Klassifikationsgüte über Testdaten $\xi_{G,T}$ und der Klassifikationsgüte bei Reklassifikation über dem gesamten Datensatz $\xi_{G,\text{all}}$. Der Algorithmus liefert zwar realistische Schätzungen, zeigt sich jedoch anfällig gegenüber Entscheidungsregeln, die sich den Lerndatensatz merken und somit $\xi_{G,\text{all}} = 1$ erfüllen [85].

Da die Reklassifikation und die Bootstrap-Methode für die zu untersuchenden Verfahren unzuverlässige Ergebnisse liefern, kommen in dieser Arbeit das Testen über Testdaten zur Evaluierung des Benchmarkbeispiels und die Crossvalidierung zur Evaluierung von realen Daten zum Einsatz.

Eine detaillierte Aufschlüsselung des Klassifikationsergebnisses liefert die sogenannte Konfusionsmatrix [77]:

$$\boldsymbol{\kappa} = \begin{pmatrix} \kappa_{11} & \kappa_{12} & \dots & \kappa_{1k} \\ \kappa_{21} & \kappa_{22} & & \vdots \\ \vdots & & \ddots & \\ \kappa_{k1} & \dots & & \kappa_{kk} \end{pmatrix}. \quad (2.102)$$

Die Elemente κ_{ij} geben Auskunft darüber, wieviele Datentupel der Klasse i der Klasse j zugeordnet wurden.

2.10.2 Zeitpunkt der Merkmalerzeugung

Der Zeitpunkt der Merkmalerzeugung (t_h oder t_e) kann nach folgenden Kriterien gewählt werden:

- Rechen- und Speichereffizienz zur Erstellung des Merkmals,
- univariate Trenngüte (ANOVA) und
- multivariate Trenngüte (MANOVA).

Rechen- und speicheraufwändig sind Merkmale dann, wenn zu ihrer Berechnung eine Vielzahl vergangener Werte (wie z.B. Zeitpunkt des aktuell angenommenen Extremums) gespeichert und aktualisiert werden müssen. Das Berechnen von Trenngüten mittels ANOVA liefert im univariaten Fall Aussagen über die Trennschärfe einzelner Merkmale, im multivariaten Fall (MANOVA, vgl. Abschnitt 2.7) über die Trennschärfe einer Kombination von Merkmalen [195]. Kombiniert werden jeweils drei gleiche Merkmalstypen (z. B. Extrema der Bereiche II, III und IV).

	Rechenaufwand		Güte ANOVA		Güte MANOVA	
	$t_{h,ji}$	$t_{e,ji}$	$t_{h,ji}$	$t_{e,ji}$	$t_{h,ji}$	$t_{e,ji}$
lokale Extrema $x_{e,ji}$	+	+	++	++	++	++
zeitliche Ausdehnung $x_{t,ji}$	++	+	+	o	+	o
Summensignal $x_{s,ji}$	+	o	++	+	++	+
Trendextrema $x_{D,ji}$	o	-	+	+	+	+
Anzahl Filterkreuzungen $x_{sp,ji}$	+	-	-	--	-	--
gefilterte Standardabweichung $x_{\sigma,ji}$	-	--	o	+	o	+

Tabelle 2.1: Merkmalsgenerierung: Bewertung des Zeitpunktes t_h im Vergleich zu t_e , vgl. Anhang 7.5, Tabellen 7.18 und 7.19

Tabelle 2.1 zeigt die Vorteile der Zeitpunkte $t_{h,ji}$ im Vergleich zu $t_{e,ji}$ anhand der Auswertung von Benchmarkdatensatz A - auf eine Betrachtung der relativen Merkmale wird hierbei verzichtet. In Bezug auf den Rechenaufwand ist grundsätzlich $t_{h,ji}$ zu bevorzugen, da hierbei keine Werte zwischengespeichert werden müssen. Die Merkmale Trendextrema, Anzahl Filterkreuzungen und gefilterte Standardabweichung benötigen aufgrund ihrer Berechnungsvorschriften einen größeren Rechenaufwand. Sie sind zudem anfälliger gegenüber Störungen und liefern deshalb geringere Trennschärfen. Das ANOVA und das MANOVA-Verfahren favorisieren $t_{h,ji}$ leicht, um Klassen zu separieren (ausführliche Ergebnisse in Anhang 7.5). Folglich ist das Berechnen aller Merkmale zu $t_{h,ji}$ zu bevorzugen. Extrema können weiterhin bei $t_{e,ji}$ berechnet werden. Trennschärfe, Rechenaufwand und Robustheit der Extrema sind nahezu identisch, kleine Abweichungen können aus Diskretisierungsfehlern entspringen.

Es werden folgende Merkmalssätze erstellt (mit jeweils $n_e - 1$ Elementen):

$$\mathbf{x}_{e,j}^T = (u_{f,j}[k_{e,j2}], \dots, u_{f,j}[k_{e,jn_e}]), \quad (2.103)$$

$$\mathbf{x}_{r,j}^T = \frac{\mathbf{x}_{e,j}}{\max_{i \in \{2, \dots, n_e - 1\}} (x_{e,ji})}, \quad (2.104)$$

$$\mathbf{x}_{t,j}^T = (k_{h,j2} - k_{h,j1}, \dots, k_{h,jn_e} - k_{h,j(n_e-1)}), \quad (2.105)$$

$$\mathbf{x}_{z,j}^T = \frac{\mathbf{x}_{t,j}}{(k_{h,jn_e} - k_{h,j1})}, \quad (2.106)$$

$$\mathbf{x}_{s,j}^T = (u_{s,j}[k_{h,j2}], \dots, u_{s,j}[k_{h,jn_e}]), \quad (2.107)$$

$$\mathbf{x}_{u,j}^T = \frac{\mathbf{x}_{s,j}}{(k_{h,jn_e} - k_{h,j1})}, \quad (2.108)$$

$$\mathbf{x}_{sp,j}^T = (u_{sp,j}[k_{h,j2}], \dots, u_{sp,j}[k_{h,jn_e}]), \quad (2.109)$$

$$\mathbf{x}_{D,j}^T = \left(\max_{k \in \{k_{h,j1}, \dots, k_{h,j2}\}} u_{D,j}[k], \dots, \max_{k \in \{k_{h,j(n_e-1)}, \dots, k_{h,jn_e}\}} u_{D,j}[k] \right) \text{ und} \quad (2.110)$$

$$\mathbf{x}_{\sigma,j}^T = (u_{\sigma,j}[k_{h,j2}], \dots, u_{\sigma,j}[k_{h,jn_e}]). \quad (2.111)$$

Für jedes Sensorsignal bzw. jede Sensorkombination wird nun ein Merkmalsatz mit je $9(n_e - 1)$ Merkmalen berechnet:

$$\mathbf{x}_{S,j}^T = (\mathbf{x}_{e,j}^T, \mathbf{x}_r^T, \mathbf{x}_l^T, \mathbf{x}_z^T, \mathbf{x}_s^T, \mathbf{x}_u^T, \mathbf{x}_{sp,j}^T, \mathbf{x}_{D,j}^T, \mathbf{x}_{\sigma,j}^T). \quad (2.112)$$

Es wird dann jedes Schaltsignal durch einen Merkmalsatz beschrieben, der die Merkmalsätze für die einzelnen Sensorsignale enthält:

$$\mathbf{x}^T = (\mathbf{x}_{S,1}^T, \dots, \mathbf{x}_{S,n_s}^T). \quad (2.113)$$

2.10.3 Umfang des Lerndatensatzes

Einerseits benötigen die Klassifikatoren des MMIs zum Anlernen möglichst viele Beispiele, um gute Klassifikationsresultate zu liefern, andererseits dürfen Patienten nicht durch anstrengende Trainings- und Aufnahmesitzungen überlastet werden.

Um den Einfluss des Umfangs der Lerndaten auf die Klassifikationsresultate abzuschätzen, wertet dieser Abschnitt den Klassifikationsfehler in Abhängigkeit vom Umfang des Lerndatensatzes aus. Hierzu werden für jeden Klassifikator nacheinander Lerndatensätze bestehend aus $n_j = 2, n_j = 3, \dots, n_j = 30$ Datentupeln pro Klasse aus Datensatz B zufällig gezogen und über Datensatz C getestet. Der Vorgang wird zehn Mal wiederholt. Plausibilitätsuntersuchungen aus den bisherigen Abschnitten führen zu folgenden Parametrierungen:

- KNN: Verwendung einer MLP-Zwischenschicht mit $n_n = 12$ Neuronen mit Bias und Tansig-Aktivierungsfunktion nach (2.62) sowie K Ausgangsneuronen mit linearer Aktivierungsfunktion, Gewichtung durch gewichteten euklidischen Abstand nach (2.58), Implementierung mit Hilfe von [106],
- k-NN: Suche nach $k=1$ nächsten Nachbarn, Distanzberechnung mit euklidischem Abstand, Implementierung mit Hilfe von [30] und
- SVM: Polynomkern zweiter Ordnung nach (2.88), $C = 1000$, Betrachtung von paarweisen Klassenkombinationen, Implementierung mit Hilfe von [30].

Ohne Beschränkung der Allgemeinheit findet eine Intervalltransformation aller Merkmale auf das Intervall $[0; 1]$ (nach (2.38)) statt, um bessere Ergebnisse für k-NN und SVM zu erhalten. Der AML-Klassifikator verbessert sich dadurch nicht, die KNN und die Fuzzy-Regelbasen in den meisten Fällen ebenfalls nicht.

Die gemittelten Ergebnisse stellt Bild 2.22 dar.

Alle Klassifikatoren liefern tendenziell bessere Ergebnisse, je mehr Lerndaten zur Verfügung stehen. Ab ca. 25 Datentupel pro Klasse ändern sich die Ergebnisse kaum noch. Für die notwendige Größe des Lerndatensatzes empfiehlt sich ein Umfang von $n_j = 10 \dots 15$ Datensätzen pro Klasse. In diesem Bereich ist der Klassifikationsfehler bereits klein genug ($1 - \xi_G = 11\%$ für Fuzzy mit $n_j = 15$, $1 - \xi_G = 4\%$ für SVM mit $n_j = 15$), und der Aufwand für den Patienten, d. h. die Anzahl zu generierender Schaltsignale ist beherrschbar.

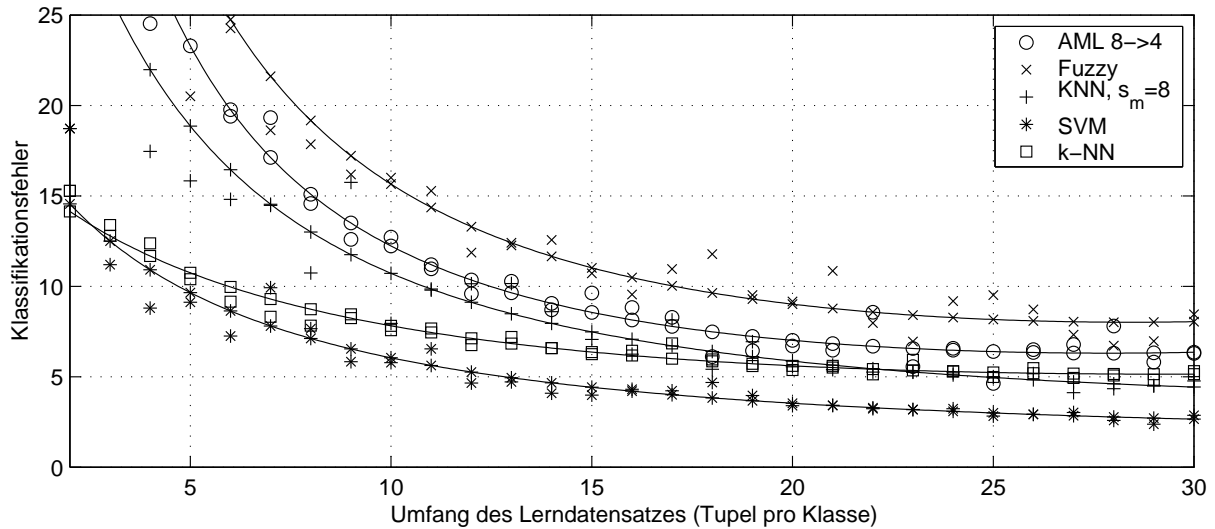


Bild 2.22: Klassifikationsfehler mit Interpolationen (Exponentialfunktionen) in Abhängigkeit vom Umfang an Lerndaten für verschiedene Klassifikatoren

2.10.4 Ergebnisse

Um anhand des MMI-Index die vorgestellten Klassifikationsverfahren zu evaluieren, sind die Kriterien Klassifikationsgüte Q_G , Anzahl umsetzbarer Zustände Q_K , Mikrocontroller-Kategorie Q_C und Engineering-Kosten Q_L anhand des Benchmarks zu bewerten.

Dabei werden die zu extrahierenden Merkmale als gegeben vorausgesetzt. Die Verfahren zur Konstruktion einer Entscheidungsregel werden in Abhängigkeit von ausgewählten und aggregierten Merkmalen beurteilt. Es gelten die Parametrierungen aus Abschnitt 2.10.3. Alle nachfolgenden Ergebnisse basieren auf einem Anlernen eines aus Datensatz B wiederholt zufällig gezogenen Lerndatensatzes L mit $n_j = 10, K = 8$.

Zur Berechnung der Klassifikationsgüte ξ_G werden die angelernten Verfahren auf Datensatz C ($n_j = 1000, K = 8$) getestet. Diese Vorgehensweise wird 100-mal wiederholt, die Ergebnisse werden gemittelt. Aus Gründen der Übersichtlichkeit stellt dieser Abschnitt nicht die Klassifikationsgüte, sondern den relativen Klassifikationsfehler $100 - \xi_G$ bei Variation der Anzahl ausgewählter bzw. aggregierter Merkmale dar (Tabelle 2.2).

Die Fuzzy-Regelbasen, k-Nearest-Neighbour-Klassifikatoren und Support-Vektor-Maschinen vermindern ihren Klassifikationsfehler, je mehr Merkmale zur Verfügung stehen. Die beiden anderen Verfahren besitzen ein Optimum für sechs bis acht ausgewählte Merkmale. Sehr sensitiv reagiert der AML-Klassifikator, da die Merkmalsanzahl quadratisch auf die Anzahl zu schätzender Elemente der Kovarianzmatrix einwirkt, und somit mit gleichen Daten wesentlich mehr Parameter geschätzt werden müssen. Die Klassifikationsgüte fällt ab. Ähnliches gilt für die Künstlichen Neuronalen Netze. Während die Merkmalsaggregation den AML-Klassifikator aus o. g. Gründen stark verbessert, schränkt sie bei den anderen Verfahren den Merkmalsraum zu sehr ein, so dass der Klassifikationsfehler steigt. Einzig die k-Nearest-Neighbour-Klassifikatoren liefern vergleichbare Ergebnisse. Einen negativen Einfluss auf die Ergebnisse des AML-Klassifikators übt die Verletzung der Normalverteilungsannahme im Lerndatensatz aus. Die anderen Verfahren werden hierdurch nicht beeinflusst.

Es sei bemerkt, dass diese Ergebnisse nur für die vorliegende Problemstellung repräsentativ sind. Es

Anzahl Merkmale	Fuzzy	KNN	k-NN	SVM	AML
$s_m = 3$	43.2	30.7	33.7	29.2	31.2
$s_m = 4$	32.6	20.2	20.5	19.2	19.0
$s_m = 6$	19.2	9.6	9.8	8.7	12.4
$s_m = 8$	16.5	9.1	8.2	7.3	15.6
alle Merkmale	15.5	34.9	7.8	6.1	64.8
$s_m = 8, s_d = 1$	63.5	37.5	38.8	36.9	33.9
$s_m = 8, s_d = 2$	51.3	22.4	23.5	22.4	22.0
$s_m = 8, s_d = 3$	38.7	16.4	17.6	16.4	16.2
$s_m = 8, s_d = 4$	24.2	12.0	11.4	10.4	12.0
$s_m = 8, s_d = 6$	16.0	9.2	7.8	7.7	11.7

Tabelle 2.2: Klassifikationsfehler $100 - \xi_G$ in % bei Variation der auszuwählenden Merkmale s_m und der zu aggregierenden Merkmale s_d beim Testen über Testdatensatz C, Bestwerte sind fett hervorgehoben.

gibt kein Verfahren, das auf allen Datensätzen die anderen Verfahren überflügelt, da die Eigenschaften eines Datensatzes immer bestimmte Stärken und Schwächen eines Verfahrens hervorheben. Einen großen Verfahrenvergleich anhand verschiedener Benchmarkdatensätze und Problemstellungen stellt [110] vor.

Um das Kriterium "Engineering-Kosten" zu beurteilen, werden die Anlernzeiten für Merkmalsselektion T_{s_m} , Merkmalsaggregation T_{s_d} und Konstruktion der Entscheidungsregel t_e auf einem PC-System mit Pentium IV mit 2600MHz und 1GB Arbeitsspeicher gemessen.

Die Dauer zur Auswahl von s_m Merkmalen ist in Tabelle 2.3 dargestellt. Sie steigt nahezu linear mit der Anzahl auszuwählender Merkmale. Die Dauer zur Berechnung der Transformationsvorschrift für die Merkmalsaggregation ist nahezu konstant und beträgt $T_{s_d} = 0.016s$. Die Dauer T_e zum Anlernen der Entscheidungsregeln in Abhängigkeit von der Anzahl zu verwendender Merkmale ist in Tabelle 2.4 dargestellt. Bei den Fuzzy-Regelbasen und den Künstlichen Neuronalen Netzen steigt T_e mit steigender Merkmalsanzahl s_m . Die k-Nearest-Neighbour-Klassifikatoren benötigen gar keine Anlernzeit, da lediglich der Lerndatensatz beibehalten werden muss. Unregelmäßigkeiten der Anlerndauer der SVM sind durch Instabilitäten des Anlernalgorithmus bei Verwendung weniger Merkmale zu begründen. Der AML-Klassifikator ist geschlossen lösbar und benötigt deshalb nur eine sehr geringe Anlerndauer, wobei diese aufgrund der rechentechnischen Verarbeitung in einigen Fällen zu Null gesetzt wird. Lediglich bei der Verwendung aller Merkmale fällt diese ins Gewicht.

Anzahl Merkmale	2	3	4	6	8
Dauer in s	0.41	0.67	0.89	1.36	1.81

Tabelle 2.3: Dauer T_{s_m} in Sekunden zur Auswahl von s_m Merkmalen von Benchmarkdatensatz A

Die notwendige Anzahl an Speicherplätzen und Multiplikationen für das Kriterium "Mikrocontroller-Kategorie" stellt Tabelle 2.5 dar. Aufgrund der Auswertung von n_n Exponentialfunktionen werden die Künstlichen Neuronalen Netze mit $20 \cdot n_n$ zusätzlichen Multiplikationen beaufschlagt. Die Anzahl notwendiger Speicherplätze und Multiplikationen steigt bei den Künstlichen Neuronalen Netzen, den k-Nearest-Neighbour-Klassifikatoren und dem AML-Klassifikator monoton mit steigender Merkmalsanzahl. Für Fuzzy-Regelbasen und Support-Vektor-Maschinen ist diese Anzahl bei gleicher Klassen-

Anzahl Merkmale	Fuzzy	KNN	k-NN	SVM	AML
1	0.17	1.88	0	0.61	0
2	0.44	2.00	0	0.53	0
3	0.48	2.34	0	0.19	0
4	1.69	2.59	0	0.09	0
6	3.17	3.30	0	0.08	0
8	4.02	4.17	0	0.12	0
alle	10.41	294.7	0	0.16	0.39

Tabelle 2.4: Dauer T_e in Sekunden zum Anlernen der Entscheidungsregeln bei Variation der verwendeten Merkmale s (Benchmarkdatensatz A), Bestwerte sind fett hervorgehoben.

Merkmals- und Lerndatenanzahl problemspezifisch. Der erhöhte Aufwand zur Auswertung der Fuzzy-Regelbasis bei $s_m = 6$ Merkmalen resultiert aus einer suboptimalen Merkmalsauswahl. So werden teilweise redundante Regeln erstellt, deren Auswertung aufwändig ist. Der niedrige Aufwand bei $s_m = 8, s_d = 3$ basiert auf Regeln mit wenigen Teilprämissen. Es sind zwar relativ viele Regeln auszuwerten, der Rechenaufwand zum Auswerten jeder Regel ist aber sehr gering. Bei Support-Vektor-Maschinen steigt die Anzahl der Parameter und Multiplikationen nicht monoton aufgrund von nicht garantierten Konvergenzeigenschaften im Anlernalgorithmus.

Anzahl Merkmale	Fuzzy	KNN	k-NN	SVM	AML
$s_m = 3$	9/28	397 / 152	243/ 247	819 / 687	104/ 80
$s_m = 4$	10/27	409 / 164	324/ 329	501 / 435	168/ 120
$s_m = 6$	27/49	433 / 188	486/ 493	807 / 725	344/ 224
$s_m = 8$	9/36	457 / 212	648/ 657	1086 / 997	584/ 360
alle Merkmale	30/57	1657/1412	8748/8857	17091/ 17047	94184/47960
$s_m = 8, s_d = 1$	10/16	381 / 136	89 / 91	805 / 608	32/ 32
$s_m = 8, s_d = 2$	19/28	401 / 156	178 /181	543 / 441	72/ 64
$s_m = 8, s_d = 3$	10/35	421 / 176	267/ 271	519 / 441	128/ 104
$s_m = 8, s_d = 4$	49/73	441 / 196	356 /361	547 / 479	200/ 152
$s_m = 8, s_d = 6$	72/102	481 / 236	534 /541	873 / 789	392/ 272

Tabelle 2.5: Anzahl Multiplikationen/Speicherplätze der vorgestellten Verfahren bei Variation der Anzahl ausgewählter Merkmale s_m und aggregierter Merkmale s_d auf der Basis von Benchmarkdatensatz A, Bestwerte sind fett hervorgehoben.

Um aus den Ergebnissen den MMI-Index zu berechnen, wird folgende Parametrierung gemäß Abschnitt 2.4 vereinbart:

- $c_G = 80$ (nur Verfahren mit Klassifikationsgüten von mehr als 80% sind akzeptabel),
- $c_K = 1$ (beliebiger Parameter, da die Klassenanzahl konstant ist),
- $c_{L,0} = 2, c_{L,1} = 0.3, c_{L,2} = 0.01$ ($Q_{L,1}$ fällt 200 mal flacher als $Q_{L,2}$ ab) und
- $i = 1$ (es wird nur ein Mikrocontroller betrachtet), $c_{M0} = 300$ und $c_{M1} = 4, c_{P0} = 300$ und $c_{P1} = 4$ (maximal 300 Multiplikationen und 300 Speicherplätze sind erlaubt).

Anzahl Merkmale	Fuzzy	KNN	k-NN	SVM	AML
$s_m = 3$	0	0	0	0	0
$s_m = 4$	0	0	0	0	0.01
$s_m = 6$	0.01	0	0	0	0
$s_m = 8$	2.29	0	0	0	0
alle Merkmale	5.83	0	0	0	0
$s_m = 8, s_d = 1$	0	0	0	0	0
$s_m = 8, s_d = 2$	0	0	0	0	0
$s_m = 8, s_d = 3$	0	0	0.02	0	2.40
$s_m = 8, s_d = 4$	0	0	0	0	27.25
$s_m = 8, s_d = 6$	3.33	0	0	0	0

Tabelle 2.6: Auswertung des MMI-Index als $10000 \cdot \Lambda$, Bestwerte sind fett hervorgehoben.

Den höchsten MMI-Index erreicht der AML-Klassifikator mit Auswahl von $s_m = 8$ Merkmalen und einer Aggregation auf $s_d = 4$ Merkmale (Tabelle 2.6). Gute Ergebnisse liefern auch ein AML-Klassifikator unter Verwendung von $s_m = 3$ Merkmalen und Fuzzy-Regelbasen unter Verwendung aller Merkmale. Alle anderen Verfahren sind für die angenommene Parametrierung nicht einsetzbar, da ihre Auswertung zu aufwändig ist ($\Lambda = 0$).

Tabelle 2.7 evaluiert die Robustheit der Verfahren anhand des Testens über dem zeitvarianten Testdatensatz D_2 . Alle Verfahren liefern so schlechte Ergebnisse, dass der MMI-Index auf Null fällt. Im Vergleich zu Tabelle 2.2 scheint die Fuzzy-Regelbasis am robustesten zu sein, der Klassifikationsfehler liegt jedoch trotzdem zu hoch.

Klassifikator	Fuzzy	KNN	k-NN	SVM	AML
Klassifikationsfehler	24.0	20.3	19.2	17.1	28.7
MMI-Index	0	0	0	0	0

Tabelle 2.7: Klassifikationsfehler $100 - \xi_G$ und MMI-Index über zeitvarianten Testdaten (Testdatensatz D_2 , Auswahl von $s_m = 8$ Merkmalen, 100 Wiederholungen)

Zusammenfassend ergibt sich folgende Bewertung der vorgestellten Verfahren (Tabelle 2.8): Fuzzy-Regelbasen eignen sich gut für den Einsatz in Echtzeitsystemen und sind aufgrund ihrer Interpretierbarkeit auch zur Wissensextraktion geeignet. Damit einher geht allerdings ein kaum akzeptabler Verlust an Klassifikationsgüte – insbesondere der Lernaufwand ist sehr hoch. Große Nachteile von Künstlichen Neuronalen Netzen sind der hohe Lernaufwand bis alle Gewichte angepasst sind sowie nicht optimale Lösungen für die Parameter. k-Nearest-Neighbour-Klassifikatoren sind einfach anzulernen und liefern hohe Klassifikationsgüten, sie reagieren jedoch sensibel auf schlecht konditionierte Daten und sind in Echtzeitsystemen wegen ihrer schlechten Onlinefähigkeit nachteilig. Support-Vektor-Maschinen zeichnen sich durch sehr hohe Klassifikationsgüten aus, der Lernvorgang ist hingegen aufwändig und eine Implementierung in Echtzeitsysteme fast nicht möglich. Der Approximative-Maximum-Likelihood-Klassifikator verfügt neben einer guten Klassifikationsgüte über einen niedrigen Lernaufwand und onlinefähige Auswertalgorithmen. Keines der angeführten Verfahren ist bislang in der Lage, über zeitvarianten Testdaten robuste Ergebnisse zu liefern.

Methode	Fuzzy	KNN	k-NN	SVM	AML
Onlinefähigkeit	++	-	-	--	+
Klassifikationsgüte	-	+	+	++	o
Lernaufwand	--	--	++	o	+
Interpretierbarkeit	+	--	o	-	o
Robustheit	-	--	--	--	--

Tabelle 2.8: Bewertung der Entscheidungsregeln (- -: sehr schlecht, -: schlecht, o: befriedigend, +: gut, ++: sehr gut)

Die vorgestellten Verfahren liefern über dem synthetischen Benchmarkdatensatz zufriedenstellende Klassifikationsergebnisse. Für einen zuverlässigen Einsatz (z. B. für Prothesen) sind sie jedoch noch zu schlecht, und eine Implementierung in kostengünstige Echtzeitsysteme wird durch die hohe Anzahl notwendiger Speicherplätze sowie den Rechenaufwand erschwert. Zudem sind Klassifikationsergebnisse nicht robust, wenn zeitvariante Änderungen auftreten.

Weiterhin ist keines der Verfahren in der Lage, die datensatzspezifischen Hürden (vgl. Abschnitt 2.7) zu meistern. Es existieren keine Möglichkeiten, einzelne, isolierte Klassen zu separieren und einen Entwurf über dem verbleibenden Datensatz durchzuführen. Insbesondere für Merkmalsauswahl und -aggregation stellt dies ein erhebliches Hindernis dar. Zwar finden sich in der Literatur etliche Ansätze zum hierarchischen Klassifizieren von Datentupeln, eine Zusammenführung dieser Verfahren zum Erzielen von Synergieeffekten besteht nicht. Folglich existiert auch keine einheitliche Implementierungsplattform zum automatischen Entwurf und Vergleich der Verfahren und zur Generierung des entsprechenden mikrocontroller-tauglichen Quelltextes.

Bezüglich der erarbeiteten Kriterien für MMIs heben sich Fuzzy-Regelbasen und AML-Klassifikatoren als besonders geeignet hervor. Aufgrund seiner Eignung für alle Kriterien und Einsatzszenarien ist der AML-Klassifikator für den Einsatz in MMIs zu bevorzugen. Allerdings krankt er an schlechteren Klassifikationsgütern. Eine Analyse der Ursachen sowie eine gezielte Verbesserung sind notwendig. Im folgenden Kapitel werden deshalb neue Kriterien und Verfahren zum Entwurf von Merkmalsauswahl, -aggregation und Entscheidungsregel basierend auf dem Einsatz eines AML-Klassifikators entwickelt.

3 Neue Verfahren für den Entwurf von Klassifikatoren

3.1 Übersicht

Beim Einsatz von MANOVA zur Merkmalsauswahl, linearer Diskriminanzanalyse (LDA) zur Merkmalsaggregation und AML-Prinzip zur Konstruktion der Entscheidungsregel treten im Vergleich zu den KNN-, k-NN- und SVM-Verfahren erhöhte Klassifikationsfehler auf. Jedoch ist die AML-Entscheidungsregel das einzige gemäß dem MMI-Index implementierbare Verfahren. Gründe für den höheren Klassifikationsfehler lassen sich in vier Kategorien einordnen:

1. Die Kriterien von MANOVA und LDA bevorzugen tendenziell Merkmale, die eine einzelne Klasse weit von den übrigen positionieren, ohne dabei auf die Trennbarkeit näher zusammenliegender Klassen Rücksicht zu nehmen. Ursache sind die zugrundeliegenden Kriterien (2.47) bzw. (2.51), die bessere Werte liefern, wenn der Abstand der Mittelwerte erkennbarer Klassen von den Mittelwerten der anderen Klassen groß ist. Dies verbessert jedoch nicht die Klassifikationsgüte und erhöht im Allgemeinen den Klassifikationsfehler. Eine Illustration der Problematik zeigt Bild 3.1. Die Klassen 7 und 8 wurden gut von den übrigen getrennt, die Konfusionsmatrix zeigt jedoch deutlich, dass hierdurch eine Unterscheidung der Klassen 1 und 2 sowie 5 und 6 kaum möglich ist.
2. Bedingt durch die geringe Datenanzahl ergeben sich schlechte Schätzungen von Mittelwerten und Kovarianzmatrizen.
3. Im Allgemeinen verletzt der Datensatz die Normalverteilungsannahme.
4. Insbesondere, wenn Lern- und Testdatensätze nicht den gleichen Aufnahmesitzungen entstammen, führen zeitvariante Änderungen durch unbekannte Zufallsprozesse zu systematischen Unterschieden in Lern- und Testdaten.

Aufbauend auf Bild 2.4 ergeben sich die in Bild 3.2 dargestellten Modifikationsmöglichkeiten für den Klassifikator.

Nach [74] lässt sich die Klassifikationsgüte durch eine Auswahl geeigneter Datentupel auf der Basis von Konfidenzintervallen bereits vor dem Entwurf des Klassifikators verbessern. So kann Problem 3 abgemildert werden. Den gleichen Ansatz verfolgt eine Erweiterung der Merkmale um deren Kombination untereinander. So erhöht sich die Chance, dass die modifizierten Merkmale annähernd normalverteilt sind. Das Fehlen eines umfangreichen Lerndatensatzes verhindert jedoch diese Vorgehensweisen und führt oft zu nicht-robusten Schätzungen.

Im Gegensatz zu den in Kapitel 2 vorgestellten Filteransätzen kann zur Bewertung und Modifikation von Merkmalsauswahl, -aggregation und Entscheidungsregel z. B. auch der Klassifikationsfehler und somit die Entscheidungsregel selbst als Black-Box angewandt werden. Auf diese Weise ergeben sich die Parameter aus der Güte des zugehörigen Klassifikationssystems. In Erweiterung zum Klassifikationsfehler erstellt diese Arbeit verschiedene Kriterien basierend auf dem Entscheidungsmaß zur

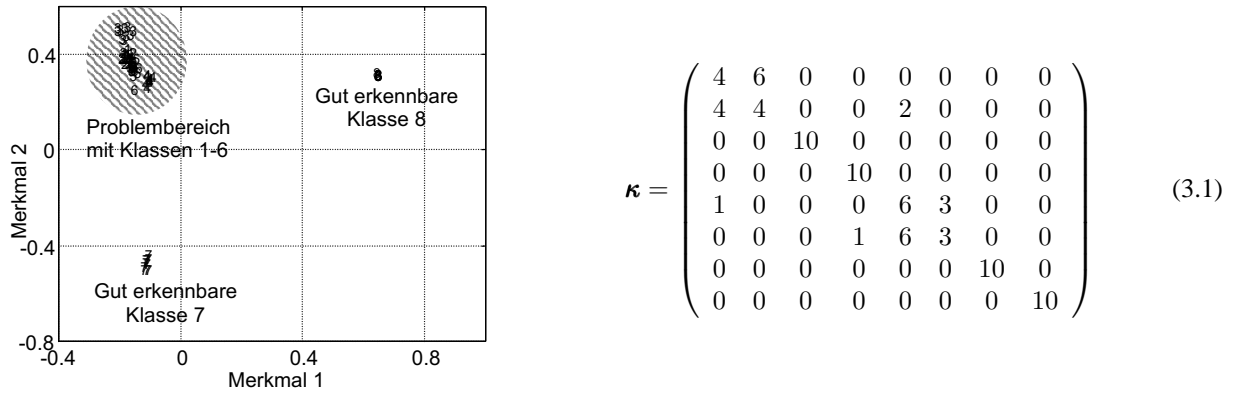


Bild 3.1: Links: Klassifikationsproblematik im aggregierten Merkmalsraum (Datensatz A, $s_m = 8$, $s_d = 2$), rechts: zugehörige Konfusionsmatrix bei Leave-One-Out Validierung

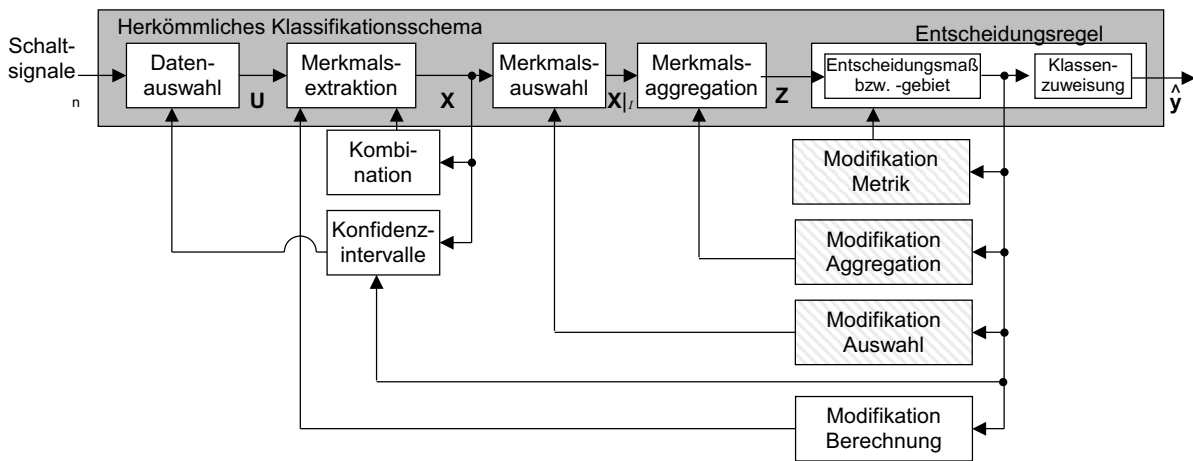


Bild 3.2: Möglichkeiten der Modifikation des Klassifikationsschemas

- Modifikation der Auswahl (Optimierung der Merkmalsauswahl, verbessert Positionierung der Klassen, siehe Problem 1),
- Modifikation der Aggregation (Optimierung der Merkmalsaggregation, verbessert Positionierung der Klassen und vermindert Auswirkungen fehlerhafter Verteilungsannahme, Probleme 1 und 3) und
- Modifikation der Metrik (Optimierung des Entscheidungsmaßes, vermindert Auswirkungen fehlerhafter Verteilungsannahme, Problem 3).

Die Berechnung der Klassifikationsgüte kann einem Bootstrap oder einer Crossvalidierung entstammen (Wrapper-Ansatz), die erforderliche Rechenleistung ist jedoch immens und eine Evaluierung des Datensatzes meist nicht mehr möglich. Deshalb wird hier die Klassifikationsgüte über Lerndaten betrachtet. Eine Wichtungsfunktion forciert korrekte Klassifikationsergebnisse. Da die Parameteranpassung nicht geschlossen lösbar ist, kommt ein numerischer Optimierer zum Einsatz, für den gute Startschätzungen aus vereinfachten geschlossen lösbaren Verfahren (z. B. aus Abschnitt 2.8) notwendig sind (Abschnitt 3.2).

Die Verwendung eines solchen Kriteriums lässt sich auf die Modifikation der Berechnung der Merkmalsextraktion (z. B. Filterkonstanten) und der Bestimmung von Konfidenzintervallen für die Datenauswahl erweitern, der Rechenaufwand für eine fortwährende Merkmalsextraktion ist jedoch oftmals zu hoch.

Das Klassifizieren eines Datentupels durch einmaliges Ausführen der Prozessschritte Merkmalsauswahl, -aggregation und Entscheidungsregel sei als Einschrittverfahren bezeichnet. In Erweiterung trifft ein Mehrschrittverfahren mehrfach eine Auswahl an Datentupeln und entwirft für diese einen individuellen Klassifikator. Alle so entstehenden Klassifikatoren werden dann in einem abschließenden Schritt wieder zu einer Ausgangsgröße zusammengefasst. Zum Entwurf der individuellen Klassifikatoren können sowohl die herkömmlichen als auch modifizierte Verfahren zum Einsatz kommen. Mehrschrittverfahren beugen den schlechten Schätzungen von Mittelwerten und Kovarianzmatrizen (Problem 2) vor, da Klassifikatoren nur für gut zu erkennende Datentupel entworfen werden. Auf diese Weise vermindern sich ebenfalls die Verletzungen der Normalverteilungsannahme (Problem 3). Abschnitt 3.3 stellt auf die Problemstellung angepasste Mehrschrittverfahren vor.

Um Problem 4 zu bewältigen, sind repräsentative Merkmale und Datensätze automatisch zu finden, die nur wenig von den zeitvarianten Änderungen beeinflusst werden. Dies kann bei Kenntnis des Zufallsprozesses unter Verwendung von a-priori Merkmalsrelevanzen geschehen [99]. Notwendig sind hierfür im Vorfeld Aussagen über die Robustheit von Merkmalen. Da diese Kenntnis meist jedoch nicht gegeben ist, wird hier der Lerndatensatz erweitert (Abschnitt 3.4). Dies erfordert u. U. eine qualitative Kenntnis über die Art der zeitvarianten Änderung oder zusätzliche Lerndaten nach einer zeitvarianten Änderung. Anhand des Benchmarkbeispiels wird die Güte des Verfahrens belegt. Abschnitt 3.5 bewertet die entwickelten Verfahren nach den in Kapitel 2 vorgestellten Kriterien.

3.2 Einschrittverfahren: Optimierung der Prozessschritte

3.2.1 Modifizierte Merkmalsauswahl: Modifikation der Selektion (MM)

Die ausgewählten Merkmale i_1, \dots, i_{s_m} sind für Klassifikationsaufgaben nur suboptimal, da das Likelihood-Quotienten-Kriterium (2.47) die auszuwählenden Kandidaten nicht nach deren Klassifikationsgüte bezüglich der nachgeschalteten Entscheidungsregel bewertet.

Ein Wrapper-Ansatz leistet Abhilfe, indem er für die Kandidatenkombinationen $\mathcal{I}_{\text{cand}}$ einen Klassifikator entwirft, dessen Klassifikationsgüte ξ_G schätzt und die Kandidaten bewertet. Das Maß für die Merkmalsauswahl gemäß (2.39) ergibt sich dann zu

$$M^{\xi_G}(\mathcal{I}) = \xi_G = \frac{1}{n} \sum_{i=1}^n \xi_{G,i}(\mathbf{x}_i|\mathcal{I}) \quad \text{mit} \quad (3.2)$$

$$\xi_{G,i}(\mathbf{x}_i|\mathcal{I}) = \begin{cases} 1 & \text{für } f(\mathbf{x}_i|\mathcal{I}; \bar{\mathbf{x}}_{y_i}|\mathcal{I}, \hat{\Sigma}_{y_i}|\mathcal{I}) > f(\mathbf{x}_i|\mathcal{I}; \bar{\mathbf{x}}_j|\mathcal{I}, \hat{\Sigma}_j|\mathcal{I}), \quad j = 1, \dots, K, \quad j \neq y_i \\ 0 & \text{sonst.} \end{cases} \quad (3.3)$$

Zur Berechnung von $f(\mathbf{x}_i|\mathcal{I}; \bar{\mathbf{x}}_{y_i}, \hat{\Sigma}_{y_i})$ verwenden in der Literatur gängige Wrapper-Ansätze den Bootstrap oder die Crossvalidierung (vgl. Abschnitt 2.10.1, [68, 86]). Beide Validierungstechniken berechnen jedoch die Parameter des Klassifikators auf der Basis von zufällig gewählten Datentupeln des Lerndatensatzes und liefern somit streuungsbehaftete Ergebnisse. Für verlässliche Aussagen über die Klassifikationsgüte sind die Verfahren mehrfach zu wiederholen und die Ergebnisse zu mitteln - die Dauer

zur Auswahl der Merkmale ist entsprechend hoch. Zudem sinkt bei kleinen Datensätzen, wie im vorliegenden Fall, die Güte der Parameterschätzungen. Deshalb schlägt diese Arbeit eine Berechnung der Klassifikationsgüte über den Lerndaten vor und passt das Kriterium (3.2) an.

Der Nachteil des Kriteriums (3.2) ist es, dass $\xi_{G,i}(\mathbf{x}_i|\mathcal{I})$ diskret und die Zielfunktion somit unstetig ist. Um ein stetiges Kriterium zu erhalten, wird der Wert des Entscheidungsmaßes für die wahre Klassenzuordnung $f(\mathbf{x}_i|\mathcal{I}; \bar{\mathbf{x}}_{y_i}, \hat{\Sigma}_{y_i})$ zur Summe der Entscheidungsmaße aller Klassenzuordnungen ins Verhältnis gesetzt. Das zugehörige Maß $M_{\mathcal{I}}^{\xi_G, \text{stetig}}$ ergibt sich dann zu

$$M^{\xi_G, \text{stetig}}(\mathcal{I}) = \frac{1}{n} \sum_{i=1}^n \hat{P}(\mathbf{x}_i|\mathcal{I}) \quad \text{mit} \quad (3.4)$$

$$\hat{P}(\mathbf{x}_i|\mathcal{I}) = \frac{f(\mathbf{x}_i; \bar{\mathbf{x}}_{y_i}|\mathcal{I}, \hat{\Sigma}_{y_i}|\mathcal{I})}{\sum_{j=1}^K f(\mathbf{x}_i; \bar{\mathbf{x}}_j|\mathcal{I}, \hat{\Sigma}_j|\mathcal{I})}. \quad (3.5)$$

Die Summe aller Verhältnisse der $\hat{P}(\mathbf{x}_i|\mathcal{I})$ über den Lerndaten liefert eine stetige Approximation für die Klassifikationsgüte, die Summe der Werte $1 - \hat{P}(\mathbf{x}_i|\mathcal{I})$ ein Kriterium für den Klassifikationsfehler. Im ersten Fall ist bezüglich \mathcal{I} zu maximieren, im zweiten zu minimieren, die Ergebnisse sind identisch. In der praktischen Anwendung hat sich jedoch gezeigt, dass es zweckmäßig ist, von einer Gleichgewichtung der Fehlklassifikationswerte $1 - \hat{P}(\mathbf{x}_i|\mathcal{I})$ abzuweichen. Fehlklassifikationen implizieren stets Verhältnisse $\hat{P}(\mathbf{x}_i|\mathcal{I}) < 0.5$; die Umkehrung gilt nicht. Werte $\hat{P}(\mathbf{x}_i|\mathcal{I}) < 0.5$ weisen aber auf einen nicht unbeträchtlichen Einfluss der anderen Klassen hin.

Es wird deshalb eine Gewichtsfunktion $\psi(w)$ eingeführt. Mit der Vorgabe, Fehlklassifikationen $w = \hat{P}(\mathbf{x}_i|\mathcal{I}) < 0.5$ stärker zu gewichten und für $w = \hat{P}(\mathbf{x}_i|\mathcal{I}) = 0$ die Gewichtung zu verdoppeln ($\psi(w) = 2$), ergibt sich das Kriterium

$$M^{\xi_G, \text{mod}}(\mathcal{I}) = \sum_{i=1}^n \psi(\hat{P}(\mathbf{x}_i|\mathcal{I})) \quad \text{mit} \quad (3.6)$$

$$\psi(w) = \begin{cases} 2 - 3w & \text{für } w < 0.5 \\ 1 - w & \text{für } w \geq 0.5. \end{cases} \quad (3.7)$$

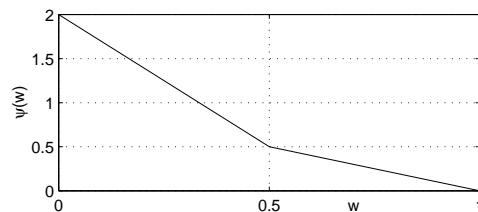


Bild 3.3: Wichtungsfunktion nach (3.7)

Die Wichtungsfunktion zeigt Bild 3.3. Auf diese Weise werden die Merkmale so ausgewählt, dass richtige Klassifikationsergebnisse zwar weniger sicher werden, aber so gleichzeitig Klassifikationsfehler vermieden werden. Die Merkmalsauswahl ist dann vorzunehmen nach

$$\mathcal{I} = \underset{\mathcal{I}_{\text{cand}}}{\operatorname{argmin}} M^{\xi_G, \text{mod}}(\mathcal{I}_{\text{cand}}) \quad \text{mit} \quad \operatorname{card}(\mathcal{I}_{\text{cand}}) = s_m. \quad (3.8)$$

Die Bilder 3.4 und 3.5 zeigen einen Vergleich von MANOVA und der modifizierten Merkmalsauswahl (MM) nach Kriterium (3.6). MANOVA trennt die Klassen 7 und 8 ab und minimiert deren Streuung. Klasse 4 kann noch erkannt werden, die anderen Klassen sind nicht mehr zu unterscheiden. Im Gegensatz dazu verzichtet Kriterium (3.6) auf eine perfekte Erkennung der Klassen 7 und 8 und konzentriert sich darauf, alle Klassen gleichermaßen gut zu erkennen. Tabellen der ausgewählten Merkmale und deren Relevanzen $M^{\xi_G, \text{mod}}$ zeigt Anhang 7.5.

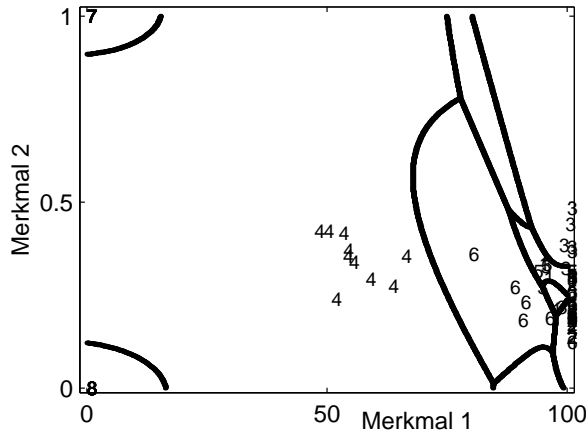


Bild 3.4: Merkmalsraum und Diskriminanzfunktionen: Auswahl von 2 Merkmalen nach MANOVA (Datensatz A)

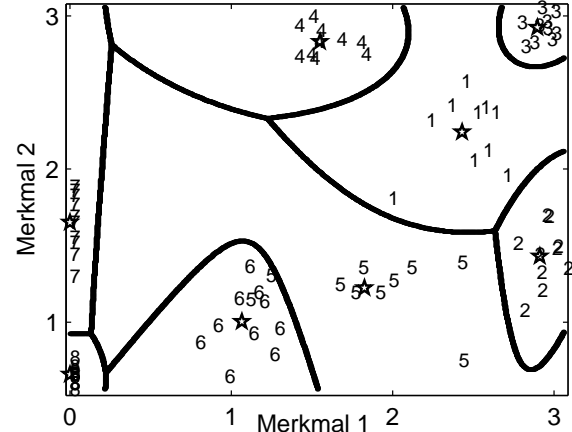


Bild 3.5: Merkmalsraum und Diskriminanzfunktionen: Auswahl von 2 Merkmalen nach (3.6) (Datensatz A)

Parametrierung	MANOVA nach (2.39, 2.47)	Merkmalsauswahl nach (3.2)	Merkmalsauswahl nach (3.4)	Merkmalsauswahl nach (3.6), (MM)
$s_m = 2$	46.5	12.0	10.6	10.7
$s_m = 3$	31.2	10.9	10.3	10.0
$s_m = 4$	19.0	10.9	9.8	9.9
$s_m = 6, s_d = 2$	20.5	17.6	16.9	16.6
$s_m = 6, s_d = 3$	16.4	11.3	11.2	10.5
$s_m = 6, s_d = 4$	11.8	10.6	10.2	9.4
$s_m = 8, s_d = 2$	22.0	20.3	18.8	17.9
$s_m = 8, s_d = 3$	16.2	12.9	12.3	11.8
$s_m = 8, s_d = 4$	12.0	10.5	10.3	10.0
$s_m = 10, s_d = 2$	20.5	20.7	18.9	18.8
$s_m = 10, s_d = 3$	17.1	12.5	12.4	11.9
$s_m = 10, s_d = 4$	12.3	9.4	10.0	10.0

Tabelle 3.1: Klassifikationsfehler der AML-Entscheidungsregel ohne/mit LDA über Testdaten (Testdatensatz C, 100 Wiederholungen, siehe Abschnitt 2.10.4) für verschiedene Parametrierungen der vorgestellten Merkmalsauswahl-Verfahren, Bestwerte sind fett hervorgehoben.

Tabelle 3.1 zeigt die Klassifikationsfehler in Abhängigkeit von den drei Kriterien (3.2), (3.4) und (3.6).

Es zeigt sich, dass alle in diesem Kapitel vorgestellten Verfahren den Klassifikationsfehler verkleinern, und dass insbesondere das neue Kriterium (3.6) die besten Ergebnisse liefert. Ist s_m klein ($s_m \leq 4$), so wählen die Kriterien (3.4) und (3.6) meist identische Merkmale aus, die Klassifikationsgüten liegen entsprechend nahe beisammen. Der Vorsprung gegenüber dem herkömmlichen MANOVA-Verfahren wird umso höher, je kleiner die Merkmalsdimension s_m gewählt wird. Somit ergeben sich bessere Klassifikationsgüten in niederdimensionalen Räumen, und es vermindern sich die Echtzeitanforderungen.

Grundsätzlich werden Klassifikationsergebnisse über Lerndaten mit steigender Merkmalsanzahl s_m und steigender Merkmalsdimension s_d besser (bzw. verschlechtern sich nicht), wenn die Lerndaten normalverteilt sind. Das gleiche gilt über Testdaten, wenn Lern- und Testdaten den gleichen Verteilungen entstammen und der Lerndatensatz umfangreich genug ist, um das Problem zu beschreiben. Da es sich bei den vorliegenden Experimenten jedoch um Ziehungen aus einer geringen Zahl von Lerndaten handelt und weiterhin nicht von normalverteilten Daten ausgegangen werden kann, können die Verfahren ihre Parameter nur grob an das Problem anpassen. Daraus folgt ein globales Minimum der Klassifikationsgüte über Testdaten in Abhängigkeit von s_m und s_d . Idealerweise soll das Minimum bei kleinen s_m und s_d auftreten, da ein Verfahren dann weniger Parameter benötigt. Da die verwendete Parameteroptimierung die Parameter besser an das Problem anpasst als die einfache Annahme der Normalverteilung (wie bei MANOVA) verschiebt sich hier das Minimum der Klassifikationsgüte in Richtung kleinerer Parameteranzahlen. In Tabelle 3.1 deutet sich beispielsweise ein Minimum durch den stagnierenden Klassifikationsfehler für $s_m = 3$ und $s_m = 4$ an. Noch deutlicher sichtbar wird dieser Umstand in den nachfolgenden Abschnitten.

3.2.2 Modifizierte Diskriminanzanalyse: Modifikation der Projektion (MD)

In die AML-Entscheidungsregel gemäß (2.94) gehen bei der Klassifikation die transformierten Merkmale $\mathbf{A}^T \mathbf{x}_i |_{\mathcal{I}}$ bzw. $\mathbf{A}^T \mathbf{x}_i$ ein. Die hierfür bestimmte Transformationsmatrix \mathbf{A} ist bezüglich des Klassifikators ebenfalls suboptimal, da sie nach dem Kriterium (2.51) berechnet wurde. Äquivalent zu Abschnitt 3.2.1 ist hier ein Wrapper-Ansatz möglich. Die in der Literatur bekannten Wrapper-Verfahren scheitern jedoch, da für die Elemente von \mathbf{A} nun nicht aus einer bestehenden Menge möglicher Werte ausgewählt werden kann, sondern kontinuierlich minimiert werden muss [197]. Der Aufwand zur Berechnung der Crossvalidierungs- oder Bootstrap-Ergebnisse steigt enorm. Wird die Klassifikationsgüte jedoch wie in Abschnitt 3.2.1 vorgeschlagen über den Lerndaten berechnet und das Kriterium modifiziert, so führt dies zu einem beherrschbaren Problem. Es gilt

$$\mathbf{A}_{\text{opt}} = \underset{\mathbf{A}}{\operatorname{argmin}} \sum_{i=1}^n \psi(\hat{P}_A(\mathbf{A}^T \mathbf{x}_i)) \quad \text{mit} \quad \psi(w) \quad \text{aus} \quad (3.7) \quad \text{und} \quad (3.9)$$

$$\hat{P}_A(\mathbf{A}^T \mathbf{x}_i) = \frac{f(\mathbf{A}^T \mathbf{x}_i; \mathbf{A}^T \bar{\mathbf{x}}_{y_i}, \mathbf{A}^T \hat{\Sigma}_{y_i} \mathbf{A})}{\sum_{j=1}^K f(\mathbf{A}^T \mathbf{x}_i; \mathbf{A}^T \bar{\mathbf{x}}_j, \mathbf{A}^T \hat{\Sigma}_j \mathbf{A})}; \quad \hat{P}_A(\mathbf{A}^T \mathbf{x}_i) \in [0, 1]. \quad (3.10)$$

\mathbf{A}_{opt} wird durch numerische Optimierung bestimmt. Als Startwert dient die aus der LDA stammende Matrix (2.53). Die Bilder 3.6 und 3.7 zeigen den Vergleich von LDA zu modifizierter Diskriminanzanalyse (MD) anhand des Benchmarkdatensatzes A. Während beide Verfahren die Klassen 3 und 4 gut erkennen, krankt die LDA an der Unterscheidung der Klassen 1, 2, 5 und 6. Die MD projiziert die Merkmale so, dass die Trennung der Klasse 4 schlechter wird, kann dann aber die Klassen 1, 2, 5 und 6 im zweidimensionalen Raum kompakt mit minimaler Überlagerung darstellen.

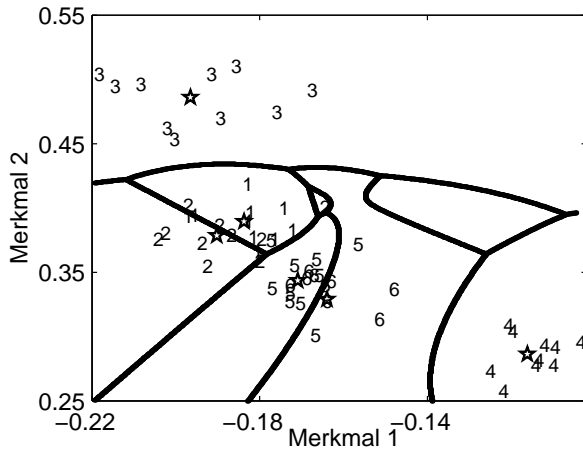


Bild 3.6: Transformierter Merkmalsraum mit A_{opt} nach (2.53), Problembereich aus Bild 3.1 (Datensatz A)

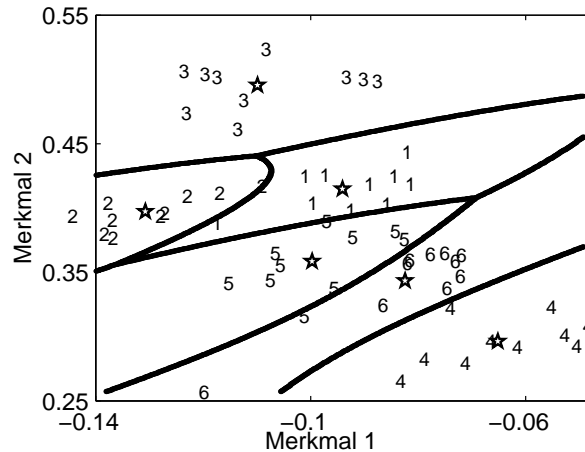


Bild 3.7: Transformierter Merkmalsraum mit A_{opt} nach (3.9), Problembereich aus Bild 3.1 (Datensatz A)

Parametrierung	MANOVA LDA	MANOVA MD	MM LDA	MM MD
$s_m = 6, s_d = 2$	20.5	13.4	16.6	11.4
$s_m = 6, s_d = 3$	16.4	12.0	10.5	9.8
$s_m = 6, s_d = 4$	11.8	11.3	9.4	10.1
$s_m = 8, s_d = 2$	22.0	12.8	17.9	12.4
$s_m = 8, s_d = 3$	16.2	12.0	11.8	10.4
$s_m = 8, s_d = 4$	12.0	11.3	10.0	9.9
$s_m = 10, s_d = 2$	20.5	11.8	18.8	13.4
$s_m = 10, s_d = 3$	17.1	11.3	11.9	11.4
$s_m = 10, s_d = 4$	12.3	11.8	10.0	10.6

Tabelle 3.2: Klassifikationsfehler über Testdaten (AML-Entscheidungsregel, Testdatensatz C, 100 Wiederholungen, siehe Abschnitt 2.10.4): Vergleich der modifizierten Verfahren (MM: modifizierte Merkmalsauswahl nach (3.6), MD: modifizierte Diskriminanzanalyse nach (3.9)), Bestwerte sind fett hervorgehoben.

Ein Vorteil der vorgestellten modifizierten Diskriminanzanalyse liegt darin, dass sie hinsichtlich klassenspezifischer Kosten einer Fehlklassifikation über die Bewertungsfunktion $\psi(w)$ erweitert werden kann.

Die Ergebnisse der modifizierten Diskriminanzanalyse über dem Benchmarkbeispiel stellt Tabelle 3.2 dar. Demnach verbessert das Verfahren die Klassifikationsgüte, insbesondere dann, wenn kein optimiertes Verfahren zur Merkmalsauswahl vorgeschaltet ist. Bei modifizierter Merkmalsauswahl nach (3.6) (MM) liefert das Verfahren nur leicht bessere Werte als die LDA, da die MM das gleiche Kriterium optimiert. Es zeigt sich jedoch, dass sich insbesondere bei niederdimensionalen Transformationen ($s_d = 2$) Vorteile ergeben - damit sinken die Echtzeitanforderungen des Systems. Ursache ist die in Abschnitt 3.2.1 diskutierte Verschiebung des Minimums des Klassifikationsfehlers. Je stärker ein Verfahren an die Daten angepasst ist, desto stärker wandert das Minimum in Richtung kleinerer Parameteranzahlen. Besonders

deutlich zeigt dies Tabelle 3.2 für $s_m = 6$ im Vergleich von MM/LDA zu MM/MD.

Steigt s_d an, verringern sich die Vorteile der modifizierten Diskriminanzanalyse, da die Modellordnung dann hoch genug ist, um mit der LDA ähnliche Klassifikationsgüten zu erreichen. Der Einfluss der Anzahl ausgewählter Merkmale s_m ist im vorliegenden Beispiel nicht einheitlich. Abschnitt 3.5 stellt den MMI-Index für die Kombination aus modifizierter Merkmalsauswahl und modifizierter Diskriminanzanalyse dar sowie die dafür notwendigen Größen Anlerndauer, Anzahl Multiplikationen und Anzahl Speicherplätze.

3.2.3 Modifizierte Entscheidungsregel: Modifikation der Metrik (MAML)

Äquivalent zu den vorangegangenen Abschnitten lassen sich Mittelwerte und Kovarianzmatrizen der Entscheidungsregel optimieren. Je nach Merkmalsraum, auf den die Entscheidungsregel angewandt wird, können hierbei s , s_m oder s_d Merkmale verwendet werden. Ohne Beschränkung der Allgemeinheit werden die folgenden Formeln in einem s -dimensionalen Merkmalsraum dargestellt. Um die Symmetrie und positive Definitheit der Kovarianzmatrizen zu garantieren, lässt sich eine Matrix \mathbf{O}_i einführen, mit

$$\hat{\Sigma}_j = \mathbf{O}_j^T \mathbf{O}_j, \quad \mathbf{O}_j \in \mathbb{R}^{s \times s}. \quad (3.11)$$

Zur Berechnung von $\bar{\mathbf{x}}_j$ und $\hat{\Sigma}_j$ ergibt sich äquivalent zu Abschnitt 3.2.1 folgendes Kriterium:

$$Q = \sum_{i=1}^n \psi \left(\frac{f(\mathbf{x}_i; \bar{\mathbf{x}}_{y_i}, \mathbf{O}_{y_i}^T \mathbf{O}_{y_i})}{\sum_{j=1}^K f(\mathbf{x}_i; \bar{\mathbf{x}}_j, \mathbf{O}_j^T \mathbf{O}_j)} \right) \rightarrow \min_{\substack{\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_K \\ \mathbf{O}_1, \dots, \mathbf{O}_K}} \quad \text{mit} \quad \psi(w) \text{ aus (3.7)}. \quad (3.12)$$

Im vorliegenden Fall lassen sich durch das Verfahren kaum Verbesserungen erreichen, da der modifizierte AML-Klassifikator (MAML) nur die Diskriminanzfunktionen verschieben kann. Diese sind aber im reduzierten Merkmalsraum durch den AML-Klassifikator hinreichend genau (Bild 3.8 und 3.9).

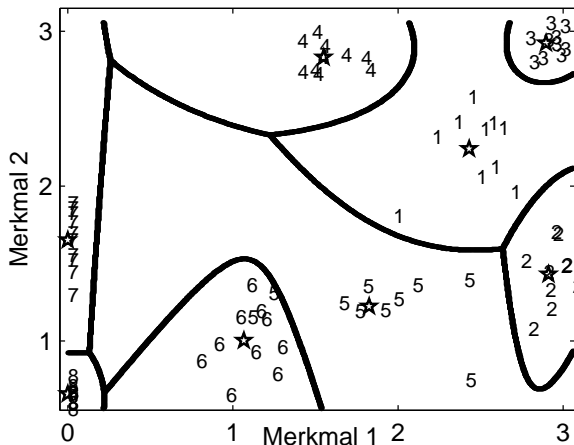


Bild 3.8: Diskriminanzfunktionen des herkömmlichen AML-Klassifikators gemäß (2.95), Datensatz A

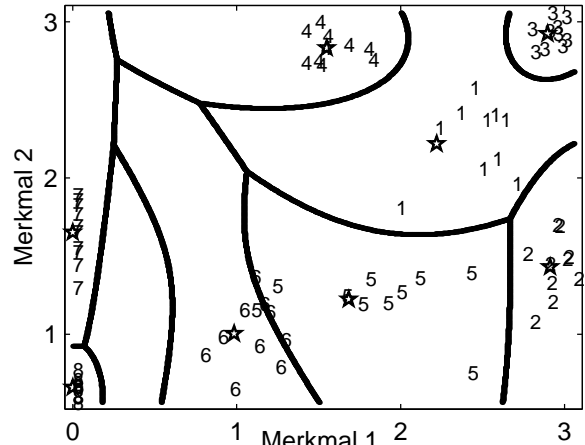


Bild 3.9: Diskriminanzfunktionen des modifizierten AML-Klassifikators gemäß (3.12), Datensatz A

Außerdem ist der Umfang des Datensatzes zu klein, um robuste Aussagen über optimale Klassengrenzen zu erhalten. Tabelle 3.3 stellt den Klassifikationsfehler über diversen Parametrierungen dar. Bei einer Vorverarbeitung durch MANOVA und LDA liefert das AML-Verfahren bessere Ergebnisse als das

MAML-Verfahren, da MANOVA und LDA die Daten bereits so auswählen und transformieren, dass der Ansatz einer Normalverteilung für die AML-Entscheidungsregel gerechtfertigt ist. Bei Verwendung von MM und/oder MD zeigt sich wiederum die in Abschnitt 3.2.1 diskutierte Verschiebung des Minimums des Klassifikationsfehlers in Richtung sinkender Parameteranzahlen. Durch MM und MD sind die Daten jedoch bereits so stark angepasst, dass die Verschiebung nur minimal ausfällt und eine Verbesserung des Klassifikationsfehlers nicht möglich ist. Das MAML-Verfahren liefert somit zwar gute Ergebnisse, ist jedoch nicht besser als das AML-Verfahren: Das MAML-Verfahren eignet sich für dieses Beispiel nicht. Das Verfahren eignet sich hingegen sehr gut, wenn eine gute Einteilung des Merkmalsraums in Klassen möglich ist, aber die Normalverteilungsannahme stark verletzt wird.

Parametrierung	MANOVA	MM	MANOVA	MM	MM
	LDA	MD	LDA	LDA	MD
	AML	AML	MAML	MAML	MAML
$s_m = 6, s_d = 2$	20.5	11.4	22.0	17.8	12.1
$s_m = 6, s_d = 3$	16.4	9.8	17.8	12.0	9.5
$s_m = 6, s_d = 4$	11.8	10.1	13.6	9.7	9.6
$s_m = 8, s_d = 2$	22.0	12.4	23.8	18.4	12.5
$s_m = 8, s_d = 3$	16.2	10.4	16.8	12.3	10.3
$s_m = 8, s_d = 4$	12.0	9.9	13.4	10.8	10.2
$s_m = 10, s_d = 2$	20.5	13.4	22.8	19.3	13.2
$s_m = 10, s_d = 3$	17.1	11.4	17.1	13.8	11.0
$s_m = 10, s_d = 4$	12.3	10.6	13.0	10.2	10.2

Tabelle 3.3: Klassifikationsfehler von AML- und MAML-Verfahren über Testdaten (Testdatensatz C, 100 Wiederholungen, siehe Abschnitt 2.10.4), Bestwerte sind fett hervorgehoben.

3.3 Mehrschrittverfahren

3.3.1 Übersicht

Während zur direkten Lösung des Mehrklassenproblems (Polychotomie) für jedes Datentupel \mathbf{x} die gleichen Abbildungen S_2 bis S_4 zum Einsatz kommen, bilden Mehrschrittverfahren Hierarchien und betrachten auf der Hierarchiestufe m individuelle Abbildungen $S_{2,m}$ bis $S_{4,m}$. Beruhen diese Abbildungen auf der Extraktion von Datensubgruppen, so kommen sogenannte Teile-und-Herrsche (engl. Separate-and-Conquer, Divide-and-Conquer)-Algorithmen zum Einsatz [62]. Eine Aufteilung des Klassifikationsproblems im Sinne des Erkennens bzw. Extrahierens von Klassensubgruppen (Reduktion des Problems, in der Anwendungsphase Verschmelzen der Ergebnisse aller Probleme) führt zur Anwendung von Aufschlussverfahren (engl. decomposition methods) [105, 211]. Besondere Aufmerksamkeit hat hierbei die Aufteilung in Zweiklassenprobleme (Dichotomien) erlangt [107], insbesondere die paarweise Kombination (engl. pairwise coupling) [70].

Diese Arbeit stellt drei neue Mehrschrittverfahren vor. Die Hierarchie entsteht dabei durch

1. ein Abspalten aller Datentupel einer ausgewählten Klasse $j \in \{1, \dots, K\}$ (bezüglich des Mehrklassenproblems). Findet keine Abspaltung statt, folgt in der nächsten Hierarchiestufe die Betrachtung eines zusätzlichen Merkmals und wiederum der Versuch des Abspaltens einer Klasse. Sobald

ein unbekanntes Datentupel zu einer abgespaltenen Klasse als zugehörig befunden wird, folgt die finale Zuweisung des Datentupels zu dieser Klasse.

2. ein Abspalten von ausgewählten Datentupeln mit einer Mindestzugehörigkeit q_c zu einer Klasse j . Verbleibende Datentupel der Klasse j werden von dieser Entscheidung nicht beeinflusst. Ein Abspalten bedingt eine finale Zuweisung eines Datentupels zur Klasse j .
3. eine Dekomposition des Lerndatensatzes in $\binom{K}{2}$ -Zweiklassenprobleme und eine problemspezifische Durchführung von Merkmalsauswahl und -aggregation. Beim Turnier-Kriterium werden die Entscheidungen aller möglichen Zweiklassenprobleme betrachtet. Beim KO-Kriterium führen die Entscheidungen einzelner Zweiklassenprobleme zur Auswahl des nächsten zu betrachtenden Zweiklassenproblems, die unterlegene Klasse wird endgültig verworfen.

Methode 1 strebt danach, komplette Klassen zuverlässig zu erkennen (Hierarchische Klassifikation von Klassen, Abschnitt 3.3.2). Methode 2 sucht nach einzelnen zuverlässig zu erkennenden Datentupeln (Hierarchische Klassifikation von Datentupeln, Abschnitt 3.3.3). Methode 3 versucht durch eine Hierarchie von Zweiklassenproblemen eine Klassifikationsentscheidung herbeizuführen. Die Betrachtung von Klassensubgruppen führt zum Entwurf von Turnier- oder KO-Klassifikatoren (Abschnitt 3.3.4).

Alle vorgestellten Verfahren stützen sich auf die angeführten statistischen Methoden, da hierbei ein leicht verständlicher Parameterzugang und ein geringer Berechnungsaufwand gegeben sind. Allerdings eignen sie sich prinzipiell für beliebige Entscheidungsregeln.

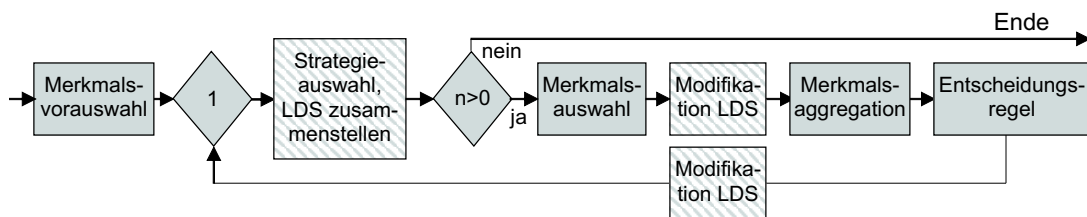


Bild 3.10: Vorgehensweise bei Mehrschrittverfahren, LDS: Lerndatensatz

Um Mehrschrittverfahren einheitlich zu entwerfen und auszuwerten, wird nun ein neuer Algorithmus eingeführt, der sich gemäß folgendem Schema gliedert (Bild 3.10 und 3.11, [157]): Zu Beginn besteht optional die Möglichkeit einer Merkmalsvorauswahl (Schritt 1), um die Komplexität für den nachfolgenden Klassifikatorentwurf zu reduzieren - alle folgenden Schritte beziehen sich dann auf einen Lerndatensatz mit diesen gewählten Merkmalen. Es folgt eine Endlosschleife (Schritt 2), wobei jeder Schleifendurchgang einer Erhöhung der Hierarchiestufe m entspricht. In den Hierarchiestufen selbst ist dann eine eigenständige Folge von Abbildungen $S_{2,m}$ bis $S_{4,m}$ zu entwerfen: Vorab findet eine Auswahl der Klassifikationsstrategie inklusive des entsprechenden Lerndatensatzes (Schritt 2a) statt. Sofern keine Bedingung für einen Abbruch gegeben ist, folgt eine Merkmalsauswahl (Schritt 2b). Eine Modifikation des Lerndatensatzes (Schritt 2c) kann nun für die folgende Merkmalsaggregation (Schritt 2d) notwendig sein, um Parameterschätzungen zu optimieren. Es folgt der Entwurf einer Entscheidungsregel (Schritt 2e) und darauf basierend eine erneute Modifikation des Lerndatensatzes (Schritt 2f).

Die Klassifikation eines unbekanntes Datentupels (Anwendung des Klassifikators) beschränkt sich dann auf die Auswahl der in Schritt 1 berechneten Merkmalsindizes (Schritt 3) sowie innerhalb einer Schleife (Schritt 4) auf die Auswahl eines Klassifikators (Schritt 4a) und die Anwendung der zugehörigen Merkmalsauswahl, -aggregation und Entscheidungsregel (Schritt 4b). Die Schleife wird dann abgebrochen,

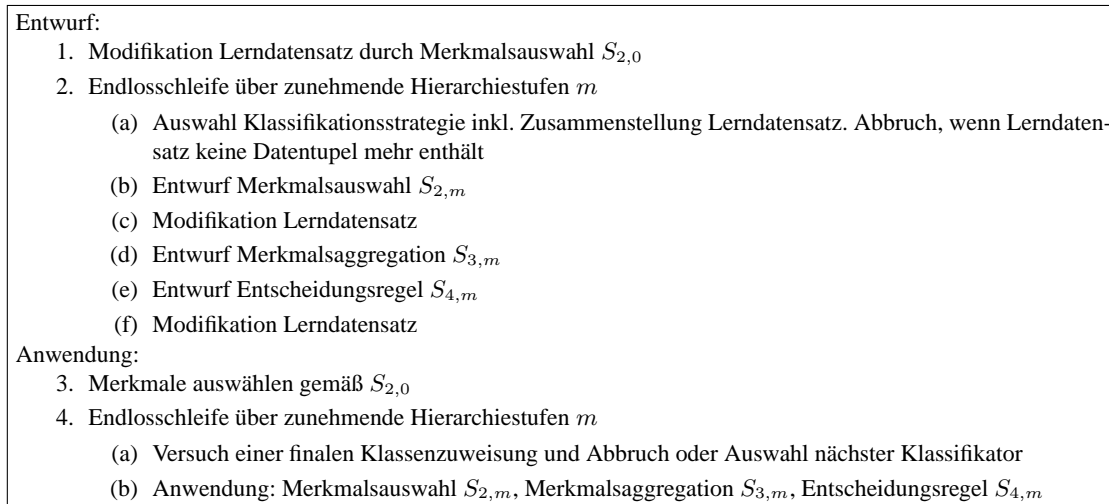


Bild 3.11: Algorithmus zum modifizierten Klassifikatorentwurf

wenn eine Klassenzuweisung geschehen ist. Diese Klassenzuweisung wird in der letzten Hierarchiestufe forciert.

Zum Entwurf von Merkmalsauswahl (Schritt 2b), Merkmalsaggregation (Schritt 2d) und der Entscheidungsregel (Schritt 2e) in den Hierarchiestufen ist eine Anwendung der modifizierten Einschnittverfahren (Abschnitt 3.2) wenig sinnvoll, da diese Verfahren insbesondere für die sofortige Klassifikation von Mehrklassenproblemen entworfen wurden. Hingegen kann der Einsatz einer modifizierten Merkmalsauswahl (MM) in Schritt 1 Vorteile bringen.

3.3.2 Hierarchische Klassifikation nach Klassen (HKK)

Die hierarchische Klassifikation nach Klassen (HKK) basiert in jeder Hierarchiestufe m auf einer Merkmalsauswahl und einer Merkmalsaggregation mit dem Ziel, mindestens eine Klasse gut zu erkennen, um sie abtrennen zu können. Wenn sich eine Klasse im Sinne eines Kriteriums deutlich von den anderen unterscheidet, ist der Weg frei für die Konstruktion eines binären Klassifikators, der in die separierbare Klasse j und die Restklassen trennt. So kann für die Restklassen ein individueller Klassifikator entworfen werden, der den großen Einfluss der Klasse j nicht berücksichtigt. Neue Kombinationen von Merkmalen, die auf eine Trennung der Klasse j verzichten können, ergeben dann bessere Schätzungen für Mittelwerte und Kovarianzmatrizen der Restklassen. Die der Klasse j angehörenden n_j Datentupel werden hierzu aus dem Datensatz eliminiert. Im verbleibenden Datensatz wird wiederum nach einer Klasse gesucht, die sich abtrennen lässt. Dies wird solange fortgesetzt, bis alle Klassen separiert wurden oder bis das Kriterium signalisiert, dass eine weitere Klassenabtrennung eine starke Erhöhung des Klassifikationsfehlers bedingt. In diesem Fall wird das MD-Verfahren für einen Klassifikatorentwurf für die verbleibenden Klassen verwendet. Ebenso besteht die Möglichkeit, die Anzahl der zu verwendenden Merkmale durch eine Merkmalsvorauswahl nach MANOVA oder MM nach (3.6) zu beschränken.

Separate-and-Conquer(Teile-und-Herrsche)-Algorithmen teilen den gesamten Merkmalsraum in Gebiete ein und weisen diesen eine Ausgangsgröße zu. Anschließend führen sie eine Bewertung der entstehenden Entscheidungsregeln durch. Die besten Regeln werden verwendet und bilden die Hierarchie. Es findet somit eine Anpassung der Regeln an den Datensatz statt. Das hier beschriebene Verfahren hingegen

arbeitet mit einer Regelbasis mit jeweils einer Regel pro Ausgangsklasse. Im Gegensatz zu den Teile-und-Herrsche-Algorithmen wird hier der Merkmalsraum so transformiert, dass sich möglichst gute Regeln ergeben. Die Bewertung der Regeln geschieht durch das neue Kriterium (3.6).

<p>Entwurf:</p> <ol style="list-style-type: none"> 1. Merkmalsvorauswahl durch MANOVA oder MM gemäß (3.6) 2. Endlosschleife über zunehmende Hierarchiestufen m <ol style="list-style-type: none"> (a) Wenn zuviele Merkmale, dann Datensätze direkt gemäß (2.3)-(2.6) klassifizieren (b) finde neues Merkmal, maximiere Abstand der Klassen nach (3.21) und (3.22) (c) - (d) maximiere Abstand der Klassen nach (3.23) (e) AML-Entscheidungsregel (f) wenn eine Klasse abgespalten wurde, entsprechende Datentupel aus LDS entfernen <p>Anwendung:</p> <ol style="list-style-type: none"> 3. Merkmale auswählen gemäß $S_{2,0}$ 4. Endlosschleife über zunehmende Hierarchiestufen m <ol style="list-style-type: none"> (a) Versuch einer finalen Klassenzuweisung (wenn Klassenzugehörigkeit = vorgegebene Klassenzugehörigkeit der Hierarchiestufe) und Abbruch oder Auswahl nächster Klassifikator (b) Anwendung: Merkmalsauswahl $S_{2,m}$, Merkmalsaggregation $S_{3,m}$, Entscheidungsregel $S_{4,m}$

Bild 3.12: Hierarchischer Klassifikatorentwurf gemäß Bild 3.11 zum Abtrennen von Klassen

Der Algorithmus (Bild 3.12, [155]) startet mit einer Merkmalsauswahl (MANOVA oder modifiziertes Verfahren, Schritt 1) zur Beschränkung des Lerndatensatzes auf s_m relevante Merkmale. Innerhalb der Endlosschleife (Schritt 2) wird dann zunächst geprüft, ob bereits alle Lerndaten einer Klasse zugeordnet wurden oder ob die Anzahl auszuwählender Merkmale ein vorgegebenes Maximum erreicht hat (Schritt 2a). Im ersten Fall bricht der Algorithmus ab, im zweiten werden die verbliebenen Datensätze gemäß (2.3)-(2.6) direkt klassifiziert. Ansonsten findet in Schritt 2b die Merkmalsauswahl auf der Basis der verbliebenen Datentupel statt. Die Anzahl zu betrachtender Merkmale wird sukzessive in jedem Schleifendurchlauf erhöht. Ziel der Merkmalsauswahl ist es, mindestens eine Klasse gut zu erkennen, um sie abtrennen zu können. Der folgende Klassifikatorentwurf verwendet nun den vorhandenen Lerndatensatz (Schritt 2c entfällt), um vorab mit der Merkmalsaggregation auf s_d Merkmale nochmals eine Klasse von den anderen abzusetzen (Schritt 2d). Auf den aggregierten Merkmalen baut eine AML-Entscheidungsregel auf (Schritt 2e). Wenn sich nun eine Klasse im Sinne eines Kriteriums deutlich von den anderen unterscheidet, wird ein Klassifikator entworfen, der die verbliebenen Datensätze der aktuellen Hierarchiestufe in die separierbare Klasse j und die Restklassen trennt. Die der Klasse j angehörenden n_j Datentupel werden danach aus dem Datensatz eliminiert (Schritt 2f).

Die Anwendung des hierarchischen Klassifikators wählt in jeder Hierarchiestufe den entsprechenden Klassifikator aus. Eine Klassenzuweisung geschieht, wenn die in der Hierarchiestufe zu bestimmende Klasse tatsächlich zugewiesen wurde oder wenn die letzte Hierarchiestufe erreicht ist.

Es soll nun detailliert auf die Funktionsweise des Algorithmus eingegangen werden. Zunächst ist ein Maß zu finden, das die Separation zweier Wahrscheinlichkeitsdichten f_i, f_j beschreibt. Dabei ist einerseits der Abstand der Mittelpunkte zu betrachten, andererseits die Richtung der Streuellipsoide. Oft werden hierfür die Divergenz nach Kullback-Leibler [91]

$$d(f_i, f_j) = \int f_i \ln \frac{f_i}{f_j} + \int f_j \ln \frac{f_j}{f_i} \quad (3.13)$$

bzw. das Ähnlichkeitsmaß nach Bhattacharyya [18]

$$d(f_i, f_j) = - \int \ln \sqrt{f_i f_j} \quad (3.14)$$

verwendet. Alternativen hierzu finden sich in [16]. Für s -variate Normalverteilungsdichten f_i, f_j ergibt sich das Maß nach Kullback-Leibler zu

$$d_{ij}^{KL} = (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \frac{\boldsymbol{\Sigma}_i^{-1} + \boldsymbol{\Sigma}_j^{-1}}{2} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \frac{1}{2} \text{sp}(\boldsymbol{\Sigma}_i \boldsymbol{\Sigma}_j^{-1} + \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\Sigma}_j - 2\mathbf{I}_s) \quad (3.15)$$

und das Maß nach Bhattacharyya zu

$$d_{ij}^B = \frac{1}{8} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \left(\frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2} \right)^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \frac{1}{2} \ln \frac{\det \frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2}}{\sqrt{\det(\boldsymbol{\Sigma}_i \boldsymbol{\Sigma}_j)}}. \quad (3.16)$$

Da die Dreiecksungleichung für beide Maße nicht erfüllt ist, handelt es sich nicht um Abstandsmaße, sondern um Ähnlichkeitsmaße im Sinne einer Halbmetrik.

Grob gesprochen beschreibt die Kullback-Leibler-Divergenz das kumulierte Verhältnis von richtig zu falsch klassifizierten Realisierungen bezüglich der Klasse i und bezüglich der Klasse j . Dies kann zu unerwünschten Schlussfolgerungen aus dem Maß insbesondere bei Abweichungen von der angenommenen Verteilungsdichte führen¹. Das Aufspalten der Summe in (3.13) leistet Abhilfe:

$$d(f_i, f_j) = \min \left\{ \int f_i \ln \frac{f_i}{f_j}, \int f_j \ln \frac{f_j}{f_i} \right\}. \quad (3.17)$$

Wird (3.17) für Normalverteilungen f_i und f_j bestimmt und werden Erwartungswert bzw. Kovarianzmatrix durch die betreffenden Schätzungen ersetzt, ergibt sich das folgende Separationsmaß

$$d_{ij}(\mathcal{I}) = \min \left\{ \frac{1}{2} (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)^T \hat{\boldsymbol{\Sigma}}_i^{-1} (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) + \frac{1}{2} \text{sp}(\hat{\boldsymbol{\Sigma}}_i^{-1} \hat{\boldsymbol{\Sigma}}_j - \mathbf{I}_r), \right. \\ \left. \frac{1}{2} (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)^T \hat{\boldsymbol{\Sigma}}_j^{-1} (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) + \frac{1}{2} \text{sp}(\hat{\boldsymbol{\Sigma}}_j^{-1} \hat{\boldsymbol{\Sigma}}_i - \mathbf{I}_r) \right\} \Big|_{\mathcal{I}}. \quad (3.18)$$

In Formel (3.18) bezeichnet $\mathcal{I} = \{i_1, \dots, i_r\}$ eine geordnete Indexmenge. Die Notation $|_{\mathcal{I}}$ meint, dass nur die in der Menge \mathcal{I} spezifizierten Merkmale verwendet werden, um die betreffenden Vektoren und Matrizen zu bilden. Weiterhin steht r für die Kardinalität von \mathcal{I} .

Die Divergenzmatrix $\mathbf{D}(\mathcal{I})$ definiert nun, wie groß die Unterschiede der einzelnen Klassen gemäß (3.18) zueinander sind

$$\mathbf{D}(\mathcal{I}) = \begin{pmatrix} 0 & d_{12}(\mathcal{I}) & \dots & d_{1K}(\mathcal{I}) \\ d_{21}(\mathcal{I}) & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & d_{K-1,K}(\mathcal{I}) \\ d_{K1}(\mathcal{I}) & \dots & d_{K,K-1}(\mathcal{I}) & 0 \end{pmatrix} \Big|_{\mathcal{C}} \quad \text{mit } d_{ij} = d_{ji}. \quad (3.19)$$

\mathcal{C} ist die Indexmenge bereits abgespaltener Klassen. Die Notation $|_{\mathcal{C}}$ bedeutet, dass die in der Menge \mathcal{C} enthaltenen Klassen nicht in die Berechnung einfließen. Zu Beginn ist $\mathcal{C} = \emptyset$.

¹Es findet keine Unterscheidung statt, ob alle Fehlklassifikationen von Klasse j stammen oder ob beide Klassen gleichermaßen zu den Fehlklassifikationen beitragen.

Für das Beispiel aus den Bildern 3.1 und 3.6 ergibt sich folgende Divergenzmatrix

$$\mathbf{D}_{1,2} = \begin{pmatrix} 0 & 2 & 28 & 106 & 13 & 10 & 2835 & 9359 \\ 2 & 0 & 34 & 98 & 12 & 16 & 2876 & 5530 \\ 28 & 34 & 0 & 133 & 61 & 41 & 2677 & 2862 \\ 106 & 98 & 133 & 0 & 106 & 44 & 1570 & 7871 \\ 13 & 12 & 61 & 106 & 0 & 2 & 1649 & 9037 \\ 10 & 16 & 41 & 44 & 2 & 0 & 732 & 8886 \\ 2835 & 2876 & 2677 & 1570 & 1649 & 732 & 0 & 15794 \\ 9359 & 5530 & 2862 & 7871 & 9037 & 8886 & 15794 & 0 \end{pmatrix}. \quad (3.20)$$

Folglich lassen sich Klasse 1 und Klasse 2 nur sehr schlecht voneinander trennen, hingegen hat Klasse 8 den größten Abstand von den übrigen. Ihr nächster Nachbar ist gemäß dem gewählten Maß Klasse 3. Für jedes Merkmal kann aus einer jeden solchen Matrix die am stärksten separierte Klasse $j \notin \mathcal{C}$ und deren Abstand zu ihrem nächsten Nachbarn über

$$d_{\max}(\mathcal{I}) = \max_{\substack{1 \leq j \leq K \\ j \notin \mathcal{C}}} \min_{\substack{1 \leq i \leq K \\ i \neq j \\ i \notin \mathcal{C}}} d_{ij}(\mathcal{I}) \quad (3.21)$$

bestimmt werden. Die Größe von $d_{\max}(\mathcal{I})$ gibt einen Hinweis, ob die Indizes \mathcal{I} Merkmale spezifizieren, die eine Klasse gut separieren. Für das Beispiel in (3.20) gilt folglich $d_{\max} = 2862$ für $j = 8$. Es ist dasjenige \mathcal{I} bei einer vorgegebenen Kardinalität r zu wählen, das $d_{\max}(\mathcal{I})$ über \mathcal{I} maximiert. Bei s Merkmalen sind demzufolge $\binom{s}{r}$ Matrizen $\mathbf{D}(\mathcal{I})$ zu berechnen. Ein suboptimales Verfahren erweitert die Indexmenge \mathcal{I} mit einem Erweiterungselement i_{r+1} sequentiell nach dem Prinzip

$$\mathcal{I}_{r+1} = \{\mathcal{I}_r, i_{r+1}\} \quad \text{mit } i_{r+1} = \underset{\substack{1 \leq l \leq s_m \\ l \notin \mathcal{I}_r}}{\operatorname{argmax}} d_{\max}(\{\mathcal{I}_r, l\}) \text{ und } \mathcal{I}_0 = \emptyset. \quad (3.22)$$

Hohe Kardinalitäten r bedingen hochdimensionale Merkmalsräume und somit eine große Anzahl zu schätzender Parameter beim Entwurf der Entscheidungsregel. Es ist sinnvoll, eine lineare Merkmalsaggregation mit der Transformationsmatrix $\mathbf{A}_{\text{hierch, opt}}$ durchzuführen, wenn die Kardinalität r eine vorgegebene Dimension s_d überschreitet. Die Transformationsmatrix ist dann so zu optimieren, dass die maximale Divergenz zweier Klassen maximiert wird:

$$\mathbf{A}_{\text{hierch, opt}} = \underset{\mathbf{A}_{\text{hierch}}}{\operatorname{argmax}} d_{\max}(\mathbf{A}_{\text{hierch}}, \mathcal{I}_{r+1}). \quad (3.23)$$

Als Startwerte für $\mathbf{A}_{\text{hierch}}$ eignen sich für $s_d < K$ die Verbindungsvektoren $\mathbf{a}_{\text{hierch}, i}$ vom Mittelwert der Klasse mit maximaler Divergenz (gemäß (3.18)) zu den Mittelwerten der übrigen Klassen². Es gilt $\mathbf{A}_{\text{hierch}} = (\mathbf{a}_{\text{hierch}, 1}, \dots, \mathbf{a}_{\text{hierch}, s_d})$. Für $s_d \geq K$ kann (2.53) als Startwert herangezogen werden. Ist $r \leq s_d$ wird $\mathbf{A}_{\text{hierch, opt}} = \mathbf{I}_{r,r}$ gesetzt.

Ob \mathcal{I}_{r+1} und $\mathbf{A}_{\text{hierch, opt}}$ ausreichen, um eine Klasse zu separieren, wird mit dem Kriterium Q_j geprüft, in das (3.10) einzusetzen ist. Für jede Klasse $j \notin \mathcal{C}$ beschreibt es die Güte der gefundenen Parameter

$$Q_j = \frac{1}{n_j} \sum_{\nu: y_\nu = j \notin \mathcal{C}} \psi(\hat{P}_A(\mathbf{A}_{\text{hierch, opt}}^T \mathbf{x}_\nu)) \quad \text{mit} \quad (3.7). \quad (3.24)$$

Klassen, deren Gütewert Q_j eine vorgegebene Fehlertoleranz Q_{\max} unterschreiten, werden extrahiert, ihre Indizes der Menge \mathcal{C} zugefügt. Anschließend wird der Algorithmus erneut durchlaufen. Sind alle zur

²Geordnet werden die Vektoren nach fallender Divergenz zur ausgewählten Klasse.

Gegeben: k, s_m, s_d, Q_{\max}
 $r = 0$: Anzahl ausgewählter Merkmale,
 $\mathcal{C} = \emptyset$: Indexmenge abgespaltener Klassen
 $\mathcal{I}_r = \emptyset$: Indexmenge verwendeter Merkmale

1. Wähle s_m Merkmale nach MANOVA oder MM gemäß (3.6) aus
2. Solange 1
3. Wenn $\text{card}(\mathcal{C}) = K - 1$, Ende.
4. Wenn $r - 1 = s_m$, klassifiziere alle Klassen $j \notin \mathcal{C}$ gemäß Abschnitt (3.2.2). Ende.
5. Für $l = 1$ bis s_m , $l \notin \mathcal{I}_r$ (übrige Merkmale)
6. Berechne Divergenzmatrix für Klassen $j \notin \mathcal{C}$ gemäß (3.19): $\mathbf{D}(\{\mathcal{I}_r, l\})$.
7. Berechne maximale Nächste-Nachbar-Divergenz gemäß (3.21): $d_{\max}(\{\mathcal{I}_r, l\})$.
8. Erweitere bestehenden Merkmalsatz gemäß (3.22): $\mathcal{I}_{r+1} = \{\mathcal{I}_r, i_{r+1}\}$.
9. Erhöhe Anzahl ausgewählter Merkmale: $r = r + 1$.
10. Wenn $r > s_d$ (dann Merkmalsaggregation)
11. Startparameter festlegen: $\mathbf{A}_{\text{hierch}} = (\mathbf{a}_{\text{hierch},1}, \dots, \mathbf{a}_{\text{hierch},s_d})$.
12. nach Kriterium (3.23) optimieren: $\mathbf{A}_{\text{hierch, opt}}$.
13. sonst $\mathbf{A}_{\text{hierch, opt}} = \mathbf{I}_{r,r}$.
14. Entwerfe den AML-Klassifikator
15. Berechne Güte Q_j der Klassifikation jeder Klasse $j \notin \mathcal{C}$ gemäß (3.24).
16. Extrahiere alle Klassen $j \notin \mathcal{C}$ mit $Q_j < Q_{\max}$: Setze $\mathcal{C} = \{j | Q_j < Q_{\max}\}$.
17. Ende.

Bild 3.13: Algorithmus zum hierarchischen Klassifikatorentwurf nach HKK

Verfügung stehenden Merkmale bereits verwendet worden, so wird der Algorithmus durch eine Aggregation der übrigen Klassen gemäß Abschnitt 3.2.2 und eine anschließende Klassifikation abgebrochen.

Die Anwendung des Verfahrens beschränkt sich auf die Schritte 9 bzw. 16 in Bild 3.13.

Der Algorithmus ist suboptimal, da

- Merkmale zu bisher gefundenen Merkmalskombinationen hinzugefügt und diese nicht neu berechnet werden und
- die Merkmalsauswahl erst nach der Reduktion der Dimension stattfinden darf.

Aus Aufwandsgründen wurde jedoch auf ein optimales Vorgehen verzichtet. Ebenso wie in Abschnitt 3.2.2 besteht bei diesem Verfahren die Möglichkeit, Klassifikationskosten zu integrieren.

Einen Vergleich der hierarchischen Klassifikation nach Klassen mit den bisher untersuchten Verfahren liefert Tabelle 3.4. Demnach ist bei der Merkmalsvorauswahl (Schritt 1 in Bild 3.13) stets die modifizierte Merkmalsauswahl (MM) nach (3.6) zu bevorzugen. Das Verfahren liefert bei gleicher Parametrierung bessere Ergebnisse als alle beschriebenen Einschrittverfahren. Der Einfluss der ausgewählten Merkmale s_m ist im vorliegenden Beispiel bei Verwendung des MM-Verfahrens unwesentlich, in Abhängigkeit von der Anzahl aggregierter Merkmale ergeben sich die besten Klassifikationsergebnisse für $s_d = 3$. Wiederum zeigt sich das in Abschnitt 3.2.1 diskutierte Vorhandensein des Minimums des Klassifikationsfehlers sowie dessen Verschiebung in Richtung sinkender Parameteranzahlen für das MM-Verfahren. Der Parameter Q_{\max} spiegelt die Fehlersensitivität des Verfahrens wider: Für große Q_{\max} werden Fehler zugunsten einer geringen Anzahl an Hierarchiestufen (Speicherplätze, Multiplikationen) toleriert. Wird Q_{\max} klein gesetzt, so werden nur sehr gut erkennbare Klassen extrahiert und das verbleibende Problem mit den Einschrittverfahren gelöst. Die besten Ergebnisse liefert das Verfahren für $Q_{\max} = 0.001$. Der

Parametrierung	MANOVA	MANOVA	MANOVA	MM	MM	MM	MM
	LDA AML	$Q_{\max} = 10^{-4}$	$Q_{\max} = 10^{-3}$	$Q_{\max} = 10^{-4}$	$Q_{\max} = 10^{-3}$	$Q_{\max} = 0.01$	$Q_{\max} = 0.1$
$s_m = 6, s_d = 2$	20.5	11.3	12.1	10.0	10.0	9.6	10.4
$s_m = 6, s_d = 3$	16.4	11.4	10.7	8.9	9.0	9.6	10.1
$s_m = 6, s_d = 4$	11.8	11.6	11.5	9.5	9.6	10.0	10.6
$s_m = 8, s_d = 2$	22.0	10.8	10.3	9.9	9.2	9.8	10.3
$s_m = 8, s_d = 3$	16.2	10.1	11.0	9.4	8.9	9.1	11.1
$s_m = 8, s_d = 4$	12.0	10.8	11.1	9.8	9.4	10.2	10.7
$s_m = 10, s_d = 2$	20.5	10.7	10.1	9.4	9.3	9.3	10.3
$s_m = 10, s_d = 3$	17.1	9.7	10.1	9.1	9.3	9.6	10.6
$s_m = 10, s_d = 4$	12.3	11.1	10.8	9.5	9.5	10.3	11.2

Tabelle 3.4: Klassifikationsfehler des HKK-Verfahrens über Testdaten (Testdatensatz C, 100 Wiederholungen, siehe Abschnitt 2.10.4): Vergleich der hierarchischen Klassifikation nach Klassen (HKK) mit Merkmalsvorauswahl (Schritt 1) nach MANOVA und modifizierter Merkmalsauswahl (MM) nach (3.6) bei Variation des Parameters Q_{\max} , vgl. Tabelle 3.1, Bestwerte sind fett hervorgehoben.

Wert ist klein genug, um Fehlklassifikationen zu vermeiden, begrenzt aber auch die Anzahl der Hierarchiestufen und somit die Anzahl der zu schätzenden Parameter.

Die Anzahl notwendiger Speicherplätze sowie der Aufwand zur Berechnung des Klassifikationsergebnisses sind problemspezifisch, da die Anzahl der entworfenen Klassifikatoren in den Hierarchiestufen vom Datensatz abhängt. Abschnitt 3.5 stellt die Größen Anlerndauer, Anzahl Multiplikationen und Anzahl Speicherplätze bei variierter Anzahl ausgewählter und aggregierter Merkmale dar und errechnet daraus den MMI-Index. Sofern nicht anders kenntlich gemacht, gilt im Folgenden $Q_{\max} = 10^{-3}$ für das HKK-Verfahren.

3.3.3 Hierarchische Klassifikation nach Datentupeln (HKT)

Die hierarchische Klassifikation nach Datentupeln (HKT) versucht schrittweise, gut zu erkennende Datentupel aus dem Lerndatensatz zu extrahieren und auf der Basis der verbliebenen Datentupel sowie zusätzlicher Merkmale einen neuen Klassifikator zum Abspalten weiterer Datentupel zu entwerfen (Bild 3.14). Sie verzichtet dabei auf eine Vorauswahl von Merkmalen gemäß Schritt 1. Der Entwurf neuer Hierarchiestufen bricht ab, wenn die maximale Hierarchiestufe m_{\max} erreicht wurde oder wenn alle Datentupel klassifiziert wurden. Innerhalb einer Hierarchiestufe erweitert die modifizierte Merkmalsauswahl nach (3.6) die bestehende Merkmalsauswahl um ein aussagekräftiges Merkmal auf der Basis des verbliebenen Lerndatensatzes. Da die folgenden Schritte Entwurf der Merkmalsaggregation (Schritt 2d) und Entscheidungsregel (Schritt 2e) für robuste Schätzungen eine Mindestanzahl an Lerndaten benötigen, gibt Schritt 2c die Möglichkeit, den Lerndatensatz zu modifizieren. Schließlich werden alle verbliebenen Datentupel aus dem Lerndatensatz entfernt, die eine vorgegebene minimale Klassenzugehörigkeit überschreiten (Schritt 2f).

Bei der Anwendung des HKT-Verfahrens findet in jeder Hierarchiestufe die Auswahl des entsprechenden Klassifikators statt. Eine finale Klassenzuweisung geschieht, wenn die Klassenzugehörigkeit des unbe-

1. -
 2. Endlosschleife über zunehmende Hierarchiestufen m
 - (a) Wenn maximale Hierarchiestufe erreicht, dann Datensätze direkt gemäß (2.3)-(2.6) klassifizieren
 - (b) Modifizierte Merkmalsauswahl nach (3.6) mit $s_m = s_m^{\min} + m - 1$ Merkmalen
 - (c) Erweiterung Lerndatensatz durch bereits klassifizierte Datentupel aus dem Lerndatensatz der Hierarchiestufe $m - 1$ für robuste Schätzung, wenn zu wenige Datentupel für bestimmte Klassen vorhanden sind
 - (d) Diskriminanzanalyse
 - (e) AML-Entscheidungsregel
 - (f) Entfernen erfolgreich klassifizierter Datentupel aus dem alten Lerndatensatz
- Anwendung:
3. -
 4. Endlosschleife über zunehmende Hierarchiestufen m
 - (a) finale Klassenzuweisung (wenn Klassenzugehörigkeit $>$ min. Klassenzugehörigkeit) und Abbruch oder Auswahl Klassifikator für Hierarchiestufe m
 - (b) Anwendung: Merkmalsauswahl $S_{2,m}$, Merkmalsaggregation $S_{3,m}$, Entscheidungsregel $S_{4,m}$

Bild 3.14: Hierarchischer Klassifikatorentwurf gemäß Bild 3.11 zur Extraktion von Datentupeln

kannten Datentupels eine minimale Klassenzugehörigkeit übersteigt oder wenn die letzte Hierarchiestufe erreicht ist.

Es soll nun detailliert auf die Funktionsweise des Algorithmus eingegangen werden. Die Indexmenge \mathcal{J}_m enthalte die Indizes aller noch nicht klassifizierten Datentupel der Hierarchiestufe m und ist somit zu Anfang

$$\mathcal{J}_m = \{1, \dots, n\}. \quad (3.25)$$

Weiterhin enthalte die Indexmenge $\mathcal{J}_{j,m}$ die noch nicht klassifizierten Datentupel der Klasse j

$$\mathcal{J}_{j,m} = \{i \in \mathcal{J}_m | y_i = j\}. \quad (3.26)$$

Auf der Basis der bislang nicht klassifizierten Datentupel $\mathbf{X}|_{\mathcal{J}_m}$ wird nun die Indexmenge aussagekräftiger Merkmale nach (3.6) um ein Merkmal erweitert

$$\mathbf{X}|_{\mathcal{J}_m}, \mathbf{y}|_{\mathcal{J}_m}, \mathcal{I}_{m-1} \mapsto \mathcal{I}_m, \quad \text{card}(\mathcal{I}_m) = s_m. \quad (3.27)$$

Um anschließend die Transformationsmatrix und die Entscheidungsregel zu entwerfen, ist im Gegensatz zu (3.27) sicherzustellen, dass jede Klasse durch eine Mindestanzahl an Datentupeln n_c vertreten ist. Zwar passt sich der zu entwerfende Klassifikator nicht mehr individuell an die verbliebenen Datentupel an, jedoch sind die Schätzungen seiner Parameter wesentlich robuster.

Die Indizes der Klassenvertreter der Klasse j im m -ten Durchlauf befinden sich in $\mathcal{J}_{j,m}$, die Klassenvertreter zum Anlernen heißen $\mathcal{J}_{j,m}^*$. Die Gesamtheit aller Datentupel zum Anlernen sei als \mathcal{J}_m^* bezeichnet. Eine Darstellung dieser Mengen zeigt Bild 3.15. Es gilt

$$\mathcal{J}_{j,m}^* = \begin{cases} \mathcal{J}_{j,m-1}^* & \text{für } 0 < \text{card}(\mathcal{J}_{j,m}) < n_c \\ \mathcal{J}_{j,m} & \text{sonst,} \end{cases} \quad (3.28)$$

$$\mathcal{J}_m^* = \bigcup_{j=1}^K \mathcal{J}_{j,m}^*. \quad (3.29)$$

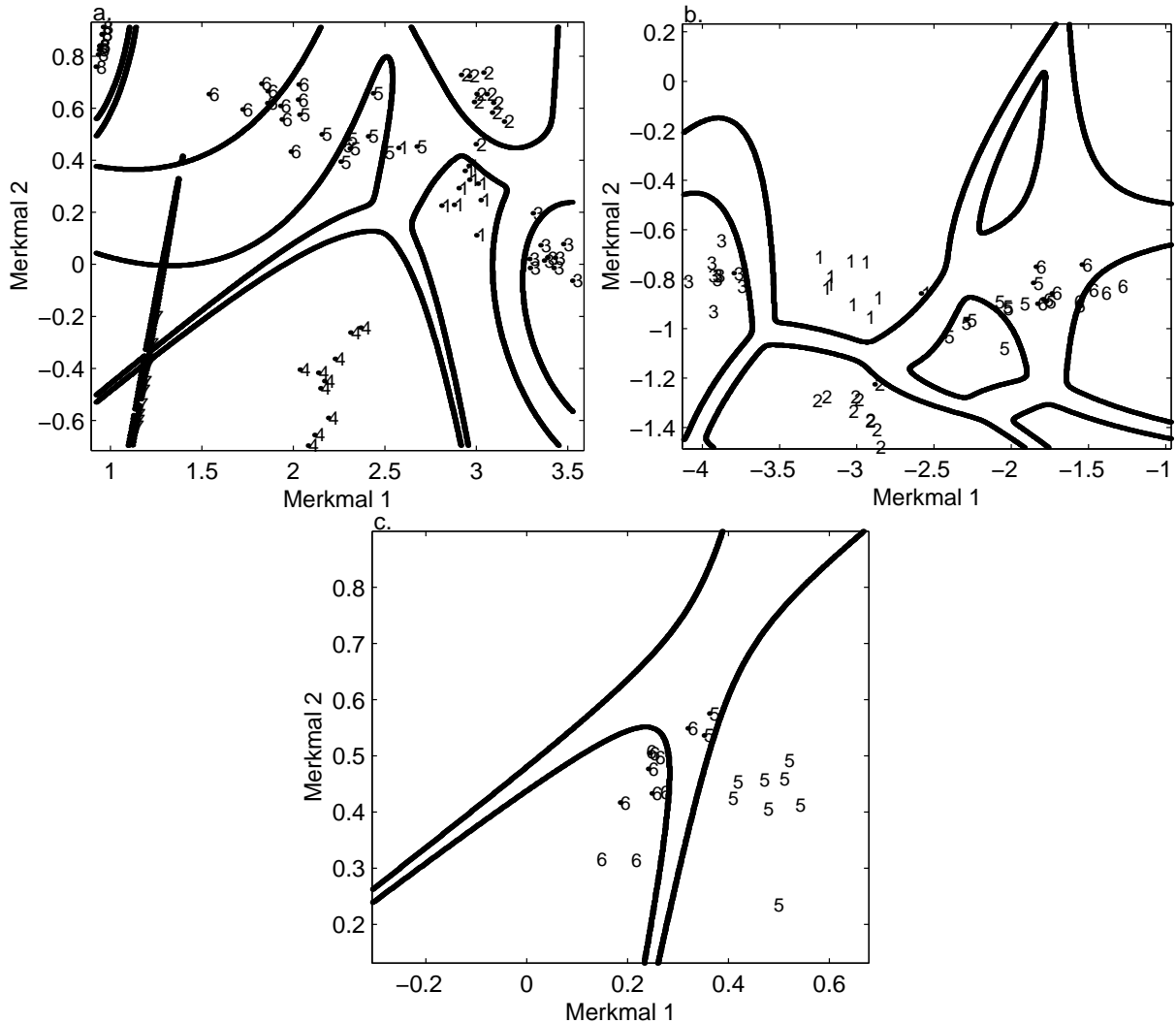


Bild 3.15: Hierarchische Klassifikation von Datentupeln (Datensatz A). a: 1. Hierarchiestufe mit Auswahl von sechs Merkmalen, Reduktion auf zwei, b: 2. Hierarchiestufe, Auswahl von sieben Merkmalen, Reduktion auf zwei, c: 3. Hierarchiestufe, Auswahl von acht Merkmalen, Reduktion auf zwei. Alle Datentupel mit Nummer entsprechen $\mathcal{J}_{j,m}^*$ bzw. \mathcal{J}_m^* , alle Datentupel mit Punkt und Nummer entsprechen $\mathcal{J}_{j,m}$ bzw. \mathcal{J}_m . Parametrierung: $s_m^{\min} = 6$, Aggregation auf $s_d = 2$ Merkmale, Anzahl Durchläufe $m_{\max} = 3$, Mindestkonfidenz $q_c = 0.9$, Mindestdatentupelanzahl $n_c = 10$.

Auf der Basis der ausgewählten Merkmale \mathcal{I}_m und der zu verwendenden Datentupel \mathcal{J}_m^* wird nun eine Transformationsmatrix \mathbf{A}_m für die Merkmalsaggregation gesucht. Sofern die Anzahl ausgewählter Merkmale die Anzahl zu aggregierender Merkmale übersteigt, d. h. $s_m > s_d$, ist \mathbf{A}_m durch eine Diskriminanzanalyse zu bestimmen, ansonsten wird sie als Einheitsmatrix angenommen:

$$\mathbf{X}|_{\mathcal{I}_m, \mathcal{J}_m^*}, \mathbf{y}|_{\mathcal{J}_m^*} \mapsto \mathbf{A}_m. \quad (3.30)$$

Nach Durchführung der Transformation (2.5) mit $\mathbf{A} = \mathbf{A}_m$ und $n = \text{card}(\mathcal{J}_m^*)$ können schließlich die Parameter der Entscheidungsregel $\bar{\mathbf{z}}_{j,m}$ und $\hat{\Sigma}_{j,m}$ geschätzt werden:

$$\mathbf{Z}|_{\mathcal{J}_m^*}, \mathbf{y}|_{\mathcal{J}_m^*} \mapsto \bar{\mathbf{z}}_{j,m}, \hat{\Sigma}_{j,m}. \quad (3.31)$$

Für alle in \mathcal{J}_m enthaltenen Datenindizes i wird nun der Wert des Entscheidungsmaßes $f_j(\mathbf{z}_i; \bar{\mathbf{z}}_{j,m}, \hat{\Sigma}_{j,m})$ in (2.94) für die wahre Klassenzuordnung zur Summe der Entscheidungsmaße aller Klassenzuordnungen ins Verhältnis gesetzt:

$$\hat{P}_{j,m}(\mathbf{z}_i) = \frac{f_j(\mathbf{z}_i; \bar{\mathbf{z}}_{y_i,m}, \hat{\Sigma}_{y_i,m})}{\sum_{j=1}^K f_j(\mathbf{z}_i; \bar{\mathbf{z}}_{j,m}, \hat{\Sigma}_{j,m})}, \quad i \in \mathcal{J}_m, \quad \hat{P}_{j,m}(\mathbf{z}_i) \in [0, 1]. \quad (3.32)$$

Übersteigt die maximale relative Klassenzugehörigkeit

$$q(\mathbf{z}_i) = \max_j \hat{P}_{j,m}(\mathbf{z}_i) \quad (3.33)$$

eine vorgegebene Konfidenz $q_c \in [0.5; 1]$, so kann i aus der Indexmenge zu betrachtender Datentupel \mathcal{J}_m entfernt werden:

$$\mathcal{J}_m = \begin{cases} \{\mathcal{J}_m \setminus i\} & \text{für } q(\mathbf{z}_i) > q_c \\ \mathcal{J}_m & \text{sonst} \end{cases}, \quad i \in \mathcal{J}_m. \quad (3.34)$$

Ist die maximale Hierarchiestufe m_{\max} erreicht, so werden alle in \mathcal{J}_m verbliebenen Datentupel in die noch nicht vollständig extrahierten Klassen gemäß (2.3) - (2.6) eingeordnet. Ist dies nicht der Fall, so erhöht sich der Zähler m um Eins und der Algorithmus wird von vorne gestartet.

Bild 3.15 visualisiert die Vorgehensweise des Algorithmus. In Bild a werden für die erste Hierarchiestufe Konfidenzbereiche berechnet, gültige Datentupel klassifiziert und aus dem Lerndatensatz entfernt (z. B. Klasse 8) oder für nachfolgende Hierarchiestufen zum Anlernen beibehalten (z. B. Klasse 2). Bild b zeigt die Konfidenzbereiche und Datentupel zum Anlernen für die zweite Hierarchiestufe, sowie Datentupel, die noch nicht klassifiziert worden sind. Alle bislang nicht klassifizierten Datentupel der Klassen 1, 2 und 3 werden richtig zugewiesen, so dass in der dritten Hierarchiestufe (Bild c) nur noch die Klassen 5 und 6 voneinander unterschieden werden müssen.

Den vollständigen Algorithmus gibt Bild 3.16 wider.

Tabelle 3.5 stellt den Klassifikationsfehler für verschiedene Parametrierungen über dem Benchmarkbeispiel dar. Im Mittel sinkt der Klassifikationsfehler je größer q_c gewählt wird. Tendenziell deutet sich jedoch für jeden Parametersatz ein Minimum in Abhängigkeit von q_c an. Die Ursache ist eine Erstellung von zu wenigen Hierarchiestufen für kleine q_c bzw. ein Überlernen aufgrund von zu vielen Hierarchiestufen auf der Basis eines zu geringen Lerndatensatzes für große q_c . Grundsätzlich gilt: Mit kleiner werdendem q_c liefern kleinere s_m^{\min} bessere Ergebnisse. Der Grund ist die niedrige Rückweisungsquote bei einer geringen Anzahl freier Parameter. In wenigen Hierarchiestufen werden somit nur aussagekräftige Merkmale verwendet, wenige Parameter geschätzt und ein Überlernen ist kaum möglich. Für kleine q_c

<p>Gegeben:</p> <p>$m = 1$: Hierarchiestufe,</p> <p>m_{\max}: maximale Anzahl Durchläufe,</p> <p>s_m^{\min}: Mindestanzahl auszuwählender Merkmale,</p> <p>$s_m = s_m^{\min}$: Anzahl auszuwählender Merkmale,</p> <p>s_d: Anzahl zu aggregierender Merkmale,</p> <p>n_c: Minimal erlaubte Kardinalität einer Klasse j,</p> <p>q_c: Minimale Konfidenz zum Erkennen einer Klasse.</p> <ol style="list-style-type: none"> 1. Solange 1 2. Wenn $\text{card}(\mathcal{J}_m) = 0$, Ende. 3. Wenn maximale Hierarchiestufe überschritten: $m = m_{\max} + 1$, dann 4. Klassifikation aller verbliebenen Datentupel mit $s_m = s_m - 1$ gemäß (2.3)-(2.6). Ende. 5. Finde verbliebene Datentupel $\mathcal{J}_{j,m}$ jeder Klasse j. 6. Finde neues relevantes Merkmal zur Klassifikation der verbliebenen Datentupel: \mathcal{I}_m. 7. Für jede Klasse j 8. Wenn Anzahl Datentupel zu klein: $0 < \text{card}(\mathcal{J}_{j,m}) < n_c$, dann 9. Lerndatensatz der Stufe $m-1$ verwenden: $\mathcal{J}_{j,m}^* = \mathcal{J}_{j,m-1}^*$ 10. sonst $\mathcal{J}_{j,m}^* = \mathcal{J}_{j,m}$. 11. Lerndatensätze aller Klassen verschmelzen: \mathcal{J}_m^*. 12. Transformationsmatrix mit $\mathbf{X} _{\mathcal{I}_m, \mathcal{J}_m^*}$ für Merkmalsaggregation berechnen: \mathbf{A}_m. 13. Transformation für verbliebene Datentupel durchführen: $\mathbf{Z} _{\mathcal{J}_m^*}$. 14. Parameter für Entscheidungsregel berechnen: $\bar{\mathbf{z}}_{j,m}, \hat{\Sigma}_{j,m}$. 15. Für alle bislang nicht klassifizierte Datentupel $i \in \mathcal{J}_m$ 16. Gewissheit der Klassenzuordnung berechnen: $q(\mathbf{z}_i)$. 17. Wenn Klassenzuordnung gewiss: $q(\mathbf{z}_i) > q_c$, dann 18. Index des Datentupels i aus \mathcal{J}_m entfernen. 19. Hierarchiestufe erhöhen: $m = m + 1$ 20. Anzahl zu betrachtender Merkmale erhöhen: $s_m = s_m + 1$. 21. Ende.
--

Bild 3.16: Algorithmus zum hierarchischen Klassifikatorentwurf nach HKT

zeigt sich weiterhin, dass die Ergebnisse mit steigender Anzahl aggregierter Merkmale s_d besser werden. Aufgrund der niedrigen Rückweisungsquote versucht das Verfahren möglichst viele Datentupel direkt zu klassifizieren - bei Problemfällen ist hierfür ein höherdimensionaler Merkmalsraum von Vorteil.

Wird q_c sehr groß gewählt, so drehen sich die Verhältnisse. Durch die niedrige Rückweisungsquote sind niederdimensionale Merkmalsräume aufgrund der geringen Anzahl zu schätzender Parameter von Vorteil, Problemfälle werden nicht mehr direkt klassifiziert, sondern in die nächste Hierarchiestufe verschoben. Eine optimale Wahl von s_m^{\min} ist dann allerdings schwer zu treffen.

Für $q_c = 0.99$ sind die Ergebnisse vergleichbar mit der hierarchischen Klassifikation nach Klassen. Von Vorteil ist, dass das Verfahren auch für Zweiklassenprobleme einsetzbar ist. Im vorliegenden Fall ist die Datentupelanzahl des Datensatzes jedoch zu klein ($n_j \leq n_c$), so dass die Vorteile des Verfahrens kaum zum Tragen kommen.

Erweitern lässt sich der beschriebene Algorithmus, indem in aufeinander folgenden Hierarchiestufen keine Erweiterung der Merkmalsauswahl stattfindet. Auf diese Weise können die gleichen Merkmale verwendet werden, um mit modifizierten Diskriminanzfunktionen andere Datentupel zu extrahieren. Hierfür ist jedoch ein größerer Umfang an Lerndaten Voraussetzung.

Parametrierung	Ref	$q_c = 0.6$	$q_c = 0.8$	$q_c = 0.9$	$q_c = 0.95$	$q_c = 0.99$
$s_m^{\min} = 5, m_{\max} = 4, s_d = 2$	22.0	13.9	11.5	10.1	10.0	9.2
$s_m^{\min} = 5, m_{\max} = 4, s_d = 3$	16.2	9.8	9.7	9.6	9.8	9.6
$s_m^{\min} = 5, m_{\max} = 4, s_d = 4$	12.0	9.3	9.5	9.6	9.7	10.1
$s_m^{\min} = 6, m_{\max} = 3, s_d = 2$	22.0	14.3	10.9	10.5	10.5	9.3
$s_m^{\min} = 6, m_{\max} = 3, s_d = 3$	16.2	10.1	9.6	9.4	9.5	9.7
$s_m^{\min} = 6, m_{\max} = 3, s_d = 4$	12.0	9.6	8.8	9.6	10.1	9.6
$s_m^{\min} = 7, m_{\max} = 2, s_d = 2$	22.0	15.3	11.9	11.9	10.7	11.2
$s_m^{\min} = 7, m_{\max} = 2, s_d = 3$	16.2	11.2	10.2	9.8	10.2	9.6
$s_m^{\min} = 7, m_{\max} = 2, s_d = 4$	12.0	9.5	10.1	9.4	9.6	9.5

Tabelle 3.5: Klassifikationsfehler des HKT-Verfahrens über Testdaten (Testdatensatz C, 100 Wiederholungen, siehe Abschnitt 2.10.4): Einfluss der Parametrierung. Vergleich mit Referenzverfahren (Ref): MANOVA mit $s_m = 8$, Diskriminanzanalyse (s_d in Tabelle), AML-Entscheidungsregel, Bestwerte sind fett hervorgehoben.

Die Anzahl notwendiger Speicherplätze sowie der Aufwand zur Berechnung des Klassifikationsergebnisses sind problemspezifisch. Die Größen Anlerndauer, Anzahl Multiplikationen und Anzahl Speicherplätze bei variierter Anzahl ausgewählter und aggregierter Merkmale finden sich in Abschnitt 3.5, ebenso wie der zugehörige MMI-Index. Im Folgenden wird mit $q_c = 0.99$ gearbeitet, da gute Klassifikationsergebnisse erreicht werden und die Gefahr eines Überlernens durch Anpassen der anderen Parameter umgangen werden kann.

3.3.4 Turnier- und Knock-Out-Klassifikation (TN, KO)

Die Turnier- bzw. Knock-Out-Klassifikation (KO) beginnt nach Bild 3.17 mit einer Merkmalsauswahl nach MANOVA oder dem MM-Verfahren nach (3.6), mittels der der gesamte Lerndatensatz auf seine wesentlichen s_m Merkmale reduziert wird (Schritt 1). Ziel ist es, Klassifikatoren für sämtliche $\frac{1}{2}K(K-1)$ Kombinationen von zwei Klassen zu finden und somit die Polychotomie in eine Serie von Dichotomien zu überführen. Sofern also noch nicht alle paarweisen Klassifikatoren entwickelt wurden, werden nur die Datentupel zweier Klassen i, j als Lerndatensatz ausgewählt (Schritt 2a). Auf deren Basis wählt das MANOVA-Kriterium optional s_m^* wichtige Merkmale aus (Schritt 2b). Das MANOVA-Kriterium ist im Zweiklassenfall geeignet, da es versucht eine Klasse weit weg zu schieben. Unter Verwendung desselben Lerndatensatzes (Schritt 2c) wird eine Merkmalsaggregation auf s_d Merkmale durchgeführt, mit dem Ziel, eine möglichst gute lineare Diskriminanzfunktion [124] zu erhalten (Schritt 2d). Anschließend wird die AML-Entscheidungsregel entworfen (Schritt 2e). Die KO-Klassifikation fordert ebenfalls den Entwurf aller paarweisen Klassifikatoren, weil sich erst in der Anwendung entscheidet, welche Paare auszuwerten sind.

Beim Anwenden des Klassifikators werden aus einem unbekanntem Datentupel die Merkmale aus Schritt 1 ausgewählt. Anschließend wird der Klassifikator für die Klassenkombination i, j gewählt. Bei der Turnier-Klassifikation werden alle Klassenkombinationen durchgespielt. Beim KO-System treten in jedem Schleifendurchlauf zwei Klassen gegeneinander an, wobei die unterlegene Klasse final verworfen wird. Im ersten Fall wird eine Klassenzugehörigkeit des Datentupels auf der Basis von kumulierten Klassenzugehörigkeiten nach Durchlauf aller Klassenkombinationen gefällt (Turnier-Kriterium), im zweiten Fall bestimmt sich die Klassenzugehörigkeit durch den Gewinner des KO-Systems (KO-

Kriterium). Die einzelnen Klassenzugehörigkeiten berechnen sich aus der entsprechenden Merkmalsauswahl, -aggregation und dem Anwenden der Entscheidungsregel (Schritt 4b).

<p>Entwurf:</p> <ol style="list-style-type: none"> 1. Merkmalsauswahl nach MANOVA oder nach (3.6): $S_{2,0}$ 2. Endlosschleife über zunehmende Hierarchiestufen m <ol style="list-style-type: none"> (a) Wenn alle Klassenkombinationen i, j getestet, dann Abbruch. Auswahl von allen Datensätzen \mathbf{x}_l der Klassen $y_l = i$ oder $y_l = j, i \neq j$ (b) MANOVA (c) - (d) finde beste lineare Diskriminanzfunktion (e) AML-Entscheidungsregel (f) - <p>Anwendung:</p> <ol style="list-style-type: none"> 3. Merkmale auswählen gemäß $S_{2,0}$ 4. Endlosschleife über zunehmende Hierarchiestufen m <ol style="list-style-type: none"> (a) Versuch einer finalen Klassenzuweisung (Turnier: Zuweisung kumulierte Klassenzugehörigkeit, wenn alle Klassenkombinationen getestet wurden; KO: wenn alle Klassen außer einer final verworfen wurden) und Abbruch oder Auswahl nächster Klassifikator (Auswahl Klassifikator für die Klassen i, j, Turnier: Weiterzählen, KO: Klassenkombination aus noch nicht verworfener Klasse gleicher Hierarchiestufe) (b) Anwendung: Merkmalsauswahl $S_{2,m}$, Merkmalsaggregation $S_{3,m}$, Entscheidungsregel $S_{4,m}$

Bild 3.17: Hierarchischer Klassifikatorentwurf gemäß Bild 3.11 zur Turnier- und KO-Klassifikation

Gemäß diesem Vorgehen und in Abhängigkeit vom Vorverarbeitungszustand der Daten zum Zeitpunkt des Entwurfs der paarweisen Klassifikatoren werden drei verschiedene Systeme entworfen und betrachtet (Bild 3.18). Allen ist gemeinsam, dass im Zweiklassenfall nur noch eine Merkmalsdimension betrachtet wird, um Speicherplätze und Rechenaufwand einzusparen. Suboptimalität in der Auswertung der Entscheidungsregel wird in Kauf genommen. Die Transformation auf eine Dimension geschieht durch das bestLDF-Verfahren, das im hochdimensionalen Raum die beste lineare Diskriminanzfunktion sucht. Im eindimensionalen Raum ist der AML-Klassifikator dann aufgrund der Transformation optimal. Der Transformationsvektor \mathbf{a} ergibt sich aus der Überlegung, wie weit der Mittelwert einer Klasse gemäß der Streuung der anderen Klasse (und umgekehrt) entfernt liegt

$$Q_{\text{Trans}} = \min \left(\frac{(\mathbf{a}^T(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j))^2}{\mathbf{a}^T \hat{\Sigma}_i \mathbf{a}}, \frac{(\mathbf{a}^T(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j))^2}{\mathbf{a}^T \hat{\Sigma}_j \mathbf{a}} \right) \rightarrow \max_{\mathbf{a}}, \quad (3.35)$$

wobei $\bar{\mathbf{x}}_i$ der Mittelwertvektor der Klasse i vor der Aggregation und $\hat{\Sigma}_i$ die Kovarianzmatrix darstellt.

Auf diese Weise werden $\frac{1}{2}K(K-1)$ Klassifikatoren entworfen. Als Entscheidungsregel kommt wiederum die AML-Regel zum Einsatz. Andere Entscheidungsregeln sind ebenfalls denkbar, aufgrund ihrer Anlerndauern und des benötigten Speicherplatzes jedoch nicht praktikabel.

Das Anwenden des l -ten Klassifikators für die Klassen i, j auf ein unbekanntes Datentupel $\mathbf{z} = \mathbf{a}^T \mathbf{x}$ führt zu zwei relativen Klassenzugehörigkeiten $\hat{P}_{j,l}(\mathbf{z})$ und $\hat{P}_{i,l}(\mathbf{z}) = 1 - \hat{P}_{j,l}(\mathbf{z})$ (siehe (2.93)) mit

$$\hat{P}_{j,l}(\mathbf{z}) = \frac{f(\mathbf{z}; \bar{\mathbf{z}}_{j,l}, \hat{\Sigma}_{j,l})}{f(\mathbf{z}; \bar{\mathbf{z}}_{j,l}, \hat{\Sigma}_{j,l}) + f(\mathbf{z}; \bar{\mathbf{z}}_{i,l}, \hat{\Sigma}_{i,l})}. \quad (3.36)$$

Für die Turnier-Klassifikation werden die Zugehörigkeiten $\hat{P}_{n,l}(\mathbf{z})$ mit $n \notin \{i, j\}$ auf 0 gesetzt. Auf diese Weise kann eine Gesamtzugehörigkeit zur j -ten Klasse aus allen Paarungen durch

$$\hat{P}_{j,\text{all}}(\mathbf{z}) = \sum_{l=1}^{\frac{1}{2}K(K-1)} \hat{P}_{j,l}(\mathbf{z}). \quad (3.37)$$

berechnet werden. Die Schätzung der Klasse geschieht dann durch

$$\hat{y} = \operatorname{argmax}_{1 \leq j \leq K} \hat{P}_{j,\text{all}}. \quad (3.38)$$

Als großer Nachteil dieses Verfahrens erweist sich die Berechnung von zwei Exponentialfunktionen in (3.36). Dies erhöht den Rechenaufwand im Mikrocontrollerbetrieb enorm. Deshalb wird im Fall der KO-Klassifikation ein Baum aus Zweiklassenpaarungen (Klasse i , Klasse j) entworfen, so dass jede Klasse in einer Paarung vorkommt. Der größere der beiden Werte $\ln f(\mathbf{z}; \bar{\mathbf{z}}_{i,l}, \hat{\Sigma}_{i,l})$ und $\ln f(\mathbf{z}; \bar{\mathbf{z}}_{j,l}, \hat{\Sigma}_{j,l})$ bestimmt die Siegerklassen, die wiederum gegeneinander antreten. Das Datentupel wird der Klasse zugewiesen, die übrig bleibt, eine Berechnung der Exponentialfunktionen entfällt.

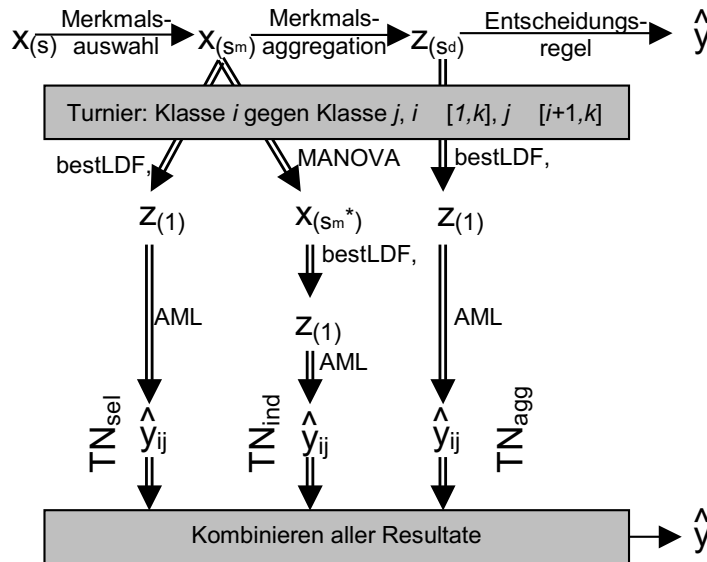


Bild 3.18: Vorgehen zur Klassifikation mittels Turnier- und KO-Klassifikatoren, gezeigt anhand der Dimension von Merkmalsvektoren

Die entworfenen Systeme unterscheiden sich danach, wie und wann eine Merkmalsauswahl stattfindet und ergeben sich gemäß Bild 3.18 wie folgt³:

1. s_m Merkmale werden im Mehrklassenfall gewählt und bestLDF vermindert die Merkmalsdimension im Zweiklassenfall auf Eins (TN_{sel} bzw. KO_{sel}). TN benötigt $\frac{1}{2}K(K-1)(s_m+4)$ Speicherplätze⁴ und $\frac{1}{2}K(K-1)(s_m+5)$ Multiplikationen⁵ und $K(K-1)$ Exponentialfunktionen, KO

³Da die Entwurfsphase für Turnier- und KO-Klassifikation identisch ist, ist eine getrennte Namensgebung redundant. Um die Bezeichner für die Anwendungsphase einzuführen, wird sie trotzdem an dieser Stelle vorgenommen.

⁴Für $\frac{1}{2}K(K-1)$ Klassenkombinationen jeweils ein Transformationsvektor mit s_m Elementen sowie 2 Mittelwerte und Standardabweichungen

⁵Für $\frac{1}{2}K(K-1)$ Klassenkombinationen jeweils s_m Multiplikationen für die Merkmalsaggregation, 4 Multiplikationen zur Auswertung von zwei Entscheidungsregeln, 1 Multiplikation zur Berechnung von (3.36)

benötigt $\frac{1}{2}K(K-1)(s_m+4)$ Speicherplätze und $(K-1)(s_m+4)$ Multiplikationen⁶.

2. s_m Merkmale werden im Mehrklassenfall gewählt, eine weitere Merkmalsauswahl wählt dann im Zweiklassenfall die s_m^* besten Merkmale⁷. Der bestLDF-Algorithmus reduziert die Merkmalsdimension auf Eins (TN_{ind} bzw. KO_{ind}). TN benötigt $\frac{1}{2}K(K-1)(s_m^*+4)$ Speicherplätze und $\frac{1}{2}K(K-1)(s_m^*+5)$ Multiplikationen und $K(K-1)$ Exponentialfunktionen, KO benötigt $\frac{1}{2}K(K-1)(s_m^*+4)$ Speicherplätze und $(K-1)(s_m^*+4)$ Multiplikationen.
3. s_m Merkmale werden im Mehrklassenfall gewählt und zu s_d aggregiert (Ansatz aus Abschnitt 3.2.2). BestLDF vermindert die Merkmalsdimension in Zweiklassenfall auf Eins (TN_{agg} bzw. KO_{agg}). TN benötigt $s_m s_d + \frac{1}{2}K(K-1)(s_d+4)$ Speicherplätze und $s_m s_d + \frac{1}{2}K(K-1)(s_d+5)$ Multiplikationen und $K(K-1)$ Exponentialfunktionen, KO benötigt $s_m s_d + \frac{1}{2}K(K-1)(s_d+4)$ Speicherplätze und $s_m s_d + (K-1)(s_d+4)$ Multiplikationen.

Findet die Auswertung des Zweiklassenproblems im eindimensionalen Merkmalsraum statt, so reicht die Auswertung eines einfachen Schwellwerts bei der KO-Klassifikation aus. Entsprechend sinkt die Anzahl der Multiplikationen und der notwendigen Speicherplätze. Um die Berechnung auf eine beliebige Anzahl von Merkmalsdimensionen anwenden zu können, wird dieser positive Umstand nicht berücksichtigt. Ein zugehöriger MMI-Index stellt somit immer eine untere Schranke dar.

Ergebnisse für KO- und Turnier-Klassifikation zeigt Tabelle 3.6. Für die gleiche Parametrierung liefert das KO-Verfahren geringfügig schlechtere Ergebnisse als das TN-Verfahren. Grundsätzlich ist auf eine Aggregation vor der Turnier-Klassifikation zu verzichten, und ebenso wie in den vorangegangenen Kapiteln liefert die modifizierte Merkmalsauswahl nach (3.6) bessere Ergebnisse als MANOVA. Wiederum zeigt sich ein Minimum des Klassifikationsfehlers, das bei Verwendung des MM-Verfahrens in Richtung kleinerer Parameteranzahlen verschoben ist. Als Merkmalsanzahl wird $s_m = 8$ vorgeschlagen.

Die besten Ergebnisse erreicht das TN_{ind}-Verfahren mit einem Klassifikationsfehler von 6.8%. Mit fallender Anzahl ausgewählter Merkmale steigt der Klassifikationsfehler.

Nachteilig wirken sich die schlechten Visualisierungsmöglichkeiten aus, ebenso wie die Beschränkung des Verfahrens auf Mehrklassenprobleme ($K > 2$). Abschnitt 3.5 diskutiert die Größen Anlerndauer, Anzahl Multiplikationen und Anzahl Speicherplätze.

3.4 Robuster Entwurf

Die Parameter der Klassifikatoren können meist nur anhand der Lerndaten aus einer Sitzung geschätzt werden, um den Patienten nicht zu überlasten. Neben den bereits diskutierten Nachteilen eines geringen Umfangs der Lerndatensätze ergibt sich ein weiteres Problem: Die Lerndatensätze spiegeln oft Eigenheiten der Aufnahmesitzung wider. Da die zu Grunde liegenden Biosignale jedoch häufig zeitvarianten Veränderungen unterliegen - die Bewegungsgewohnheiten des Patienten, Platzierung und Übertragungsverhalten der Sensoren sind sitzungsabhängig - ist der Lerndatensatz nicht repräsentativ für die spätere Anwendung des MMI. Es werden nicht-optimale Merkmale gewählt und der Klassifikationsfehler erhöht sich über unbekanntem Testdatensätzen drastisch.

Um das Problem zu umgehen, bestehen vier Möglichkeiten:

⁶Für eine Paarung: s_m Multiplikationen für die Transformation, weitere vier für die Entscheidungsregel (siehe Abschnitt 2.9.6, möglich ist auch die Einführung eines einzigen Schwellwerts). $\frac{K}{2}$ Klassifikatorpaarungen in der ersten Hierarchiestufe, $\frac{K}{4}$ in der zweiten, etc. Ist $K \neq 2^q$, $q \in \mathbb{N}$, so erhalten zufällig gewählte Klassen Freilose in der ersten Runde.

⁷ s_m^* ist zwar ein Freiheitsgrad des Klassifikationsproblems, erhöht die Anzahl der Speicherplätze jedoch nicht, da ein automatischer Quelltextexport die notwendigen Merkmale festsetzen kann.

Parametrierung	Ref	TN _{sel}	TN _{ind}	TN _{agg}	KO _{sel}	KO _{ind}	KO _{agg}
MANOVA, $s_m = 6, s_m^* = 3$	16.4	9.5	8.9	14.8	9.4	9.5	15.1
MANOVA, $s_m = 8, s_m^* = 3$	16.2	8.4	7.4	15.4	8.4	7.5	15.6
MANOVA, $s_m = 10, s_m^* = 3$	17.1	8.0	7.5	15.4	8.4	7.5	16.6
MM, $s_m = 6, s_m^* = 3$	10.5	7.9	7.3	10.2	8.0	7.6	9.7
MM, $s_m = 8, s_m^* = 3$	11.8	7.7	7.3	10.1	8.6	7.6	10.7
MM, $s_m = 10, s_m^* = 3$	11.9	8.9	6.8	11.5	9.3	7.8	12.2

Tabelle 3.6: Klassifikationsfehler der TN- und KO-Verfahren über Testdaten (Testdatensatz C, 100 Wiederholungen, siehe Abschnitt 2.10.4), TN_{agg}: $s_d = 3$. Vergleich mit Referenzverfahren (Ref): MANOVA mit $s_m = 8$, Diskriminanzanalyse mit $s_d = 3$, AML-Entscheidungsregel, Bestwerte sind fett hervorgehoben.

- Die zu klassifizierende Klassenanzahl wird zugunsten der Robustheit gesenkt.
- Der Entwurf der Merkmalsauswahl wird nach [99] mit a-priori Merkmalsrelevanzen erweitert.
- Der Klassifikator wird über mehreren, voneinander unabhängigen Aufnahmesitzungen angelernet.
- Der Lerndatensatz wird gestört und künstlich erweitert, um dem Entwurfsprozess die Möglichkeit zu geben, selbstständig einen robusten Klassifikator zu entwerfen.

Das Senken der Klassenanzahl ist eine gute Alternative, solange genügend Klassen zur Verfügung stehen. Eine geringere Klassenanzahl führt zu einer geringeren Anzahl zu schätzender Parameter und somit zu robusteren Schätzungen. Die Anwendung empfiehlt sich jedoch nur dann, wenn sich die Robustheit durch kein anderes Verfahren befriedigend realisieren lässt.

Die Vorgabe von a-priori Merkmalsrelevanzen fordert detailliertes Prozesswissen und widerspricht der Idee eines vollautomatischen Klassifikatorentwurfs. Außerdem profitieren der Entwurf von Merkmalsaggregation und Entscheidungsregel davon nicht.

Die Verwendung von mehreren Aufnahmesitzungen erhöht zwar die Dauer des Entwurfs des MMIs, die Anpassung des Klassifikators erfolgt im Gegenzug vollautomatisch.

Der Erfolg des Erweiterns des Lerndatensatzes mit gestörten Schaltsignalen, die sich aus den ursprünglichen Schaltsignalen ableiten, hängt vom jeweiligen Datensatz ab. Von Vorteil ist, dass der gesamte Entwurfsprozess an der Anpassung an zeitvariante Signale teilnimmt. Eine detaillierte Erläuterung findet sich in Anhang 7.1.2.

Um die letztgenannten beiden Verfahren zu evaluieren, werden sie über den Testdatensätzen D_1, D_2, D_3 getestet.

Für einige ausgewählte Klassifikatoren zeigt Tabelle 3.7 den Klassifikationsfehler über den Testdatensätzen beim herkömmlichen Entwurf nach Abschnitt 2.10.4 an. Der Klassifikationsfehler steigt für alle Klassifikatoren rapide an, sobald eine Störung vorliegt. Je größer die Störung ist, desto größer wird der Klassifikationsfehler. Als relativ robust erweisen sich die modifizierte Merkmalsauswahl nach (3.6) ohne nachgeschaltete Aggregation und die Turnier-Klassifikation.

Für die gleichen Klassifikatorbeispiele zeigt Tabelle 3.8 den Klassifikationsfehler bei Verwendung des robusten Entwurfs durch künstliche Erweiterung des Lerndatensatzes. Bereits über dem ursprünglichen Benchmarkbeispiel C zeigt sich eine Verbesserung von 3 - 11%. Bei der Modellierung des zeitvarianten Verhaltens durch Datensatz D_3 ergeben sich für die modifizierte Merkmalsauswahl nach (3.6) ohne

Datensatz	MANOVA, LDA $s_m = 8, s_d = 4$	MM $s_m = 4$	MM, MD $s_m = 8, s_d = 4$	HKK, MM $s_m = 8, s_d = 4$	TN _{ind} , MM $s_m = 8, s_m^* = 3$
C	12.0	9.9	9.9	9.4	7.3
D ₁	21.0	12.7	16.3	15.1	13.8
D ₂	23.2	15.9	19.3	16.4	14.8
D ₃	32.2	24.5	28.2	27.2	22.3

Tabelle 3.7: Herkömmlicher Entwurf mit Datensatz L: Durchschnittlicher Klassifikationsfehler (100 Wiederholungen, siehe Abschnitt 2.10.4) über Datensatz D₁, D₂, D₃ im Vergleich zu Datensatz C für einige ausgewählte Klassifikatorbeispiele, Bestwerte sind fett hervorgehoben.

Datensatz	MANOVA, LDA $s_m = 8, s_d = 4$	MM $s_m = 4$	MM, MD $s_m = 8, s_d = 4$	HKK, MM $s_m = 8, s_d = 4$	TN _{ind} , MM $s_m = 8, s_m^* = 3$
C	10.6	8.8	8.8	8.4	6.7
D ₁	10.9	9.8	9.2	8.7	7.2
D ₂	11.7	10.8	10.1	10.1	7.8
D ₃	18.8	18.2	16.3	16.2	13.4

Tabelle 3.8: Robuster Entwurf durch künstliches Erweitern des Lerndatensatzes (Datensatz L*): Durchschnittlicher Klassifikationsfehler (100 Wiederholungen, siehe Abschnitt 2.10.4) über Datensatz D₁, D₂, D₃ im Vergleich zu Datensatz C für einige ausgewählte Klassifikatorbeispiele, Bestwerte sind fett hervorgehoben.

nachgeschaltete Aggregation eine Verbesserung von 26%, alle anderen Verfahren zeigen Verbesserungen von ca. 40%.

Tabelle 3.9 zeigt den Klassifikationsfehler der gleichen Verfahren beim robusten Entwurf durch Verwenden von Datensatz B und D₁ zum Anlernen. Der Lerndatensatz setzt sich aus jeweils zehn Ziehungen von Datentupeln aus den Datensätzen B und D₁ zusammen. Der Klassifikationsfehler sinkt bereits über den Lerndaten und zeigt auch bei der Anwendung über den Datensätzen D₂ und D₃ bessere Ergebnisse. Somit ist das Anlernen über zwei Datensätzen zum Erreichen robuster Ergebnisse zu bevorzugen.

Datensatz	MANOVA, LDA $s_m = 8, s_d = 4$	MM $s_m = 4$	MM, MD $s_m = 8, s_d = 4$	HKK, MM $s_m = 8, s_d = 4$	TN _{ind} , MM $s_m = 8, s_m^* = 3$
C	7.3	5.4	5.3	4.9	4.5
D ₂	9.7	8.2	8.2	7.3	5.7
D ₃	18.8	17.1	16.3	15.1	12.8

Tabelle 3.9: Robuster Entwurf durch Verwenden von zwei separaten Lerndatensätzen: Durchschnittlicher Klassifikationsfehler (100 Wiederholungen, siehe Abschnitt 2.10.4) über Datensatz D₂, D₃ im Vergleich zu Datensatz C für einige ausgewählte Klassifikatorbeispiele, Bestwerte sind fett hervorgehoben.

Die geringere Verbesserung der Klassifikationsgüte des reinen MM-Verfahrens ist auf eine Unteranpas-

sung an das Problem aufgrund zu weniger freier Parameter zurückzuführen - das Verfahren wird im Folgenden nicht mehr ohne Aggregation betrachtet.

3.5 Diskussion

Die vorangegangenen Abschnitte zeigen die Notwendigkeit der Modifikation herkömmlicher Entwurfskriterien für den Entwurf von Klassifikatoren auf. Für die betrachtete Problemstellung werden neue Einschnittverfahren zur modifizierten Merkmalsauswahl (MM, Abschnitt 3.2.1), modifizierten Diskriminanzanalyse (MD, Abschnitt 3.2.2) und modifizierten AML-Entscheidungsregel (MAML, Abschnitt 3.2.3) entwickelt. Der Vorteil der Einschnittverfahren ist die strukturelle Äquivalenz zu den in Kapitel 2 vorgestellten statistischen Verfahren - die Anwendung des Klassifikators erfolgt äquivalent. Ein weiterer Vorteil ist die Integration der neuen Kriterien in Mehrschrittverfahren. Die vorgestellten Mehrschrittverfahren unterscheiden sich vor allem in ihren Anwendungsbereichen. Die hierarchische Klassifikation von Klassen (HKK, Abschnitt 3.3.2) liefert gute Ergebnisse und lässt sich gut einsetzen, um Klassenverteilungen und Diskriminanzfunktionen im Zweidimensionalen zu visualisieren. Sie ist allerdings beschränkt auf Mehrklassenprobleme. Die hierarchische Klassifikation von Datentupeln (HKT, Abschnitt 3.3.3) ist auch im Zweiklassenfall einsetzbar, liefert jedoch über dem Benchmarkbeispiel aufgrund der geringen Datentupelanzahl nur zufriedenstellende Ergebnisse. Die mit Abstand besten Ergebnisse erreichen Turnier-Klassifikatoren (TN, Abschnitt 3.3.4), eine Visualisierung ist jedoch schwer möglich und die Auswertung des Klassifikators aufwändig. KO-Klassifikatoren liefern nur geringfügig schlechtere Ergebnisse bei geringem Rechenaufwand.

Verfahren	Problem 1 Trennbarkeit	Problem 2 geringe Datenanzahl	Problem 3 keine Normalverteilung	Problem 4 zeitvariante Änderungen
MM	+	+	+	+
MD	+	+	+	o
MAML	o	o	(+)	o
HKK	++	o	+	o
HKT	+	o	o	o
KO, TN	+	+	+	+
Robuster Entwurf	o	+	o	++

Tabelle 3.10: Probleme aus Abschnitt 3.1 und Lösungsstrategien

In Bezug auf die in Abschnitt 3.1 diskutierten datensatzspezifischen vier Probleme stellt Tabelle 3.10 die Verbesserungen durch die neuen Verfahren dar. Problem 1, die Trennbarkeit von eng beisammen liegenden Klassen, wird von MM und MD sowie allen Mehrschrittverfahren gelöst. Besonders gut schneidet sich das HKK-Verfahren ab, da hier eine gesonderte Betrachtung dieser Klassen stattfindet. Problem 2, die schlechten Schätzungen aufgrund der geringen Datenanzahl, wird ebenfalls von MM und MD (durch eine niedrigere Merkmalsdimension) sowie den TN und KO-Klassifikatoren gelöst. Durch die Erweiterung des Datensatzes beim robusten Entwurf ergeben sich ebenfalls bessere Schätzungen. Problem 3, das Versagen der Klassifikatoren bei Verletzung der Normalverteilung, lösen alle Verfahren, außer HKT und dem robusten Entwurf. Der Erfolg des MAML-Verfahren hängt davon ab, ob der Merkmalsraum eine Trennung zulässt. Problem 4, die zeitvarianten Änderungen, lässt sich durch den robusten Entwurf sehr gut lösen, ebenfalls einsetzbar sind TN und KO.

Datensatz	MM, MD	HKK, MM	HKT, MM	TN _{ind} , MM	KO _{ind} , MM
$s_m = 6, s_d = 2$	16.6	10.0	9.2 ($s_m^{\min} = 5, s_d = 2$)	7.3	7.6
$s_m = 6, s_d = 3$	10.5	9.0	9.6 ($s_m^{\min} = 5, s_d = 3$)		
$s_m = 6, s_d = 4$	9.4	9.6	10.1 ($s_m^{\min} = 5, s_d = 4$)		
$s_m = 8, s_d = 2$	17.9	9.2	9.3 ($s_m^{\min} = 6, s_d = 2$)	7.3	7.6
$s_m = 8, s_d = 3$	11.8	8.9	9.7 ($s_m^{\min} = 6, s_d = 3$)		
$s_m = 8, s_d = 4$	10.0	9.4	9.6 ($s_m^{\min} = 6, s_d = 4$)		
$s_m = 10, s_d = 2$	18.8	9.3	11.2 ($s_m^{\min} = 7, s_d = 2$)	6.8	7.8
$s_m = 10, s_d = 3$	11.9	9.3	9.6 ($s_m^{\min} = 7, s_d = 3$)		
$s_m = 10, s_d = 4$	10.0	9.5	9.5 ($s_m^{\min} = 7, s_d = 4$)		

Tabelle 3.11: Klassifikationsfehler der modifizierten Verfahren über Testdaten (Testdatensatz C, 100 Wiederholungen, siehe Abschnitt 2.10.4, abweichende Parametrierung für HKT), Bestwerte sind fett hervorgehoben.

Neben den qualitativen Vor- und Nachteilen der vorgestellten Verfahren soll nun der MMI-Index eine quantitative Evaluierung ermöglichen. Tabelle 3.11 zeigt den Klassifikationsfehler der modifizierten Verfahren, es gelten die in den jeweiligen Kapiteln beschriebenen Parametrierungen. Auf eine Betrachtung des robusten Entwurfs wird verzichtet, da die qualitativen Ergebnisse die gleichen sind. Die Mehrschrittverfahren sind den Einschrittverfahren überlegen. Das TN_{ind}-Verfahren liefert die besten Ergebnisse, gefolgt von KO_{ind}, HKK, HKT und den Einschrittverfahren.

Datensatz	MM, MD	HKK, MM	HKT, MM	TN _{ind} , MM	KO _{ind} , MM
$s_m = 6, s_d = 2$	7.6	8.1	3.8	5.8	
$s_m = 6, s_d = 3$	6.0	8.1	3.9		
$s_m = 6, s_d = 4$	5.2	7.8	3.6		
$s_m = 8, s_d = 2$	12.0	14.6	4.7	7.4	
$s_m = 8, s_d = 3$	9.0	13.2	4.4		
$s_m = 8, s_d = 4$	7.2	13.2	4.2		
$s_m = 10, s_d = 2$	15.9	25.3	5.1	9.2	
$s_m = 10, s_d = 3$	11.9	19.0	5.0		
$s_m = 10, s_d = 4$	9.7	27.6	4.9		

Tabelle 3.12: Gemittelte Dauer T_{s_m} in Sekunden zum Anlernen der modifizierten Verfahren (100 Wiederholungen, abweichende Parametrierung für HKT, siehe Tabelle 3.11), Bestwerte sind fett hervorgehoben.

Tabelle 3.12 stellt die Dauer zum Anlernen der einzelnen Verfahren auf einem PC-System mit Pentium IV mit 2600MHz und 1GB Arbeitsspeicher dar. Die dargestellten Werte sind Mittelwerte über 100 Entwürfen jedes Klassifikators, da die enthaltenen numerischen Optimierungen datensatzabhängige Laufzeiten aufweisen. Hohe Anlerndauern benötigen die Verfahren, die viele numerische Optimierungen und/oder Merkmalsauswahlen enthalten. Mit steigender Anzahl ausgewählter Merkmale nimmt die Anlerndauer bei allen Verfahren zu. Hingegen sinkt die Anlerndauer oft, wenn die Anzahl aggregierter Merkmale erhöht wird. Dies hat für die vorliegenden Verfahren zwei Gründe: Für die Einschrittverfahren ist die Klassifikationsgüte in einem höherdimensionalen, aggregierten Merkmalsraum bereits so gut, dass die

Möglichkeiten zur Optimierung beschränkt sind (vgl. Tabelle 3.2). Bei den beiden hierarchischen Verfahren werden im höherdimensionalen Raum Datentupel besser abgeschieden und damit zusätzlich zur verkürzten Optimierungszeit weniger Hierarchiestufen durchlaufen. Einen Sonderfall stellt das HKK-Verfahren bei Auswahl von $s_m = 10$ Merkmalen dar. Hier führt die Verwendung von vier aggregierten Merkmalen zu einer Erhöhung der Optimierungsdauer von (3.23), da eine Vielzahl ($s_m \cdot s_d$) nichtoptimaler Startwerte zu verbessern ist.

Datensatz	MM, MD	HKK, MM	HKT, MM	TN _{ind} , MM	KO _{ind} , MM
$s_m = 6, s_d = 2$	68 / 60	145 / 142	180 / 145	1264/196	49 /196
$s_m = 6, s_d = 3$	122 / 98	228 / 202	211 / 163		
$s_m = 6, s_d = 4$	192 / 144	308 / 252	275 / 202		
$s_m = 8, s_d = 2$	72 / 64	187 / 182	168 / 135	1264/196	49 /196
$s_m = 8, s_d = 3$	128 / 104	238 / 212	230 / 178		
$s_m = 8, s_d = 4$	200 / 152	382 / 315	247 / 182		
$s_m = 10, s_d = 2$	76 / 68	194 / 191	130 / 105	1264/196	49 /196
$s_m = 10, s_d = 3$	134 / 110	230 / 206	200 / 156		
$s_m = 10, s_d = 4$	208 / 160	349 / 296	261 / 194		

Tabelle 3.13: Gemittelte Anzahl Multiplikationen/Speicherplätze der modifizierten Verfahren (100 Wiederholungen, abweichende Parametrierung für HKT, siehe Tabelle 3.11), Bestwerte sind fett hervorgehoben.

Tabelle 3.13 stellt die notwendige Anzahl an Multiplikationen und Speicherplätzen der modifizierten Verfahren dar. Die Ergebnisse sind wiederum über 100 Realisierung gemittelt worden. Hierbei ist zu beachten, dass die Anzahl Multiplikationen des HKK- und des HKT-Verfahrens die maximale Anzahl Multiplikation darstellt (Durchlauf bis zur letzten Hierarchiestufe), die für eine Entscheidung notwendig ist. Kann ein Datentupel bereits in einer vorhergehenden Hierarchiestufe klassifiziert werden, so sinkt die Anzahl entsprechend. Eine Betrachtung der maximalen Anzahl ist jedoch sinnvoll, da ein Mikrocontroller in der Lage sein muss, diese Rechenleistung zu liefern. Das TN_{ind}- und das KO_{ind}-Verfahren weisen Werte für Multiplikationen und Speicherplätze auf, die unabhängig von der dargestellten Parametrierung sind. Dies liegt an der internen Auswahl von $s_m^* = 3$ Merkmalen aus den bereits ausgewählten s_m Merkmalen. So wird für jegliche Merkmalsauswahl der gleiche Rechenaufwand benötigt. Das TN_{ind}-Verfahren ist hier sehr rechenintensiv, da viele Exponentialfunktionen (jeweils mit 20 Multiplikationen gewertet, vgl. Kapitel 4) ausgewertet werden müssen.

Die Einschrittverfahren zeigen den niedrigsten Aufwand an Multiplikationen und Speicherplätzen, das HKK-Verfahren für große s_d den größten. Gleichwohl sind die Zahlen für die hierarchischen Verfahren überraschend niedrig. Der Grund ist ein oftmaliges Verschwinden vieler Klassen in der zweiten Hierarchiestufe, was zu einem sehr niedrigen Rechen- und Speicheraufwand in den folgenden Hierarchiestufen führt. Das TN_{ind}-Verfahren sticht durch sehr großen Rechenaufwand aufgrund der hohen Anzahl an auszuwertenden Exponentialfunktionen (mit jeweils 20 Multiplikationen angesetzt) hervor. Hingegen ist der niedrige Rechenaufwand für das KO_{ind}-Verfahren auffällig. Dies liegt an der Reduktion auf sehr wenige Merkmale in den einzelnen Zweiklassenproblemen. Zu bemerken ist, dass sich Speicherplätze- und Multiplikationsanzahl für TN_{ind} mit K^2 verändern. Bei weniger als $K = 8$ Klassen sinkt der Rechenaufwand schnell, bei mehr Klassen steigt er schnell.

Tabelle 3.14 zeigt den MMI-Index der betrachteten Verfahren. Grundsätzlich weisen alle in diesem Kapitel eingeführten Verfahren (abgesehen von TN_{ind}) eine wesentliche Verbesserung des MMI-Index im

Datensatz	MM, MD	HKK, MM	HKT, MM	TN _{ind} , MM	KO _{ind} , MM
$s_m = 6, s_d = 2$	1.86	90.25	105.47	0	204.52
$s_m = 6, s_d = 3$	93.99	51.31	65.79		
$s_m = 6, s_d = 4$	89.46	0	13.28		
$s_m = 8, s_d = 2$	0.26	75.26	112.44	0	201.47
$s_m = 8, s_d = 3$	48.97	40.82	47.24		
$s_m = 8, s_d = 4$	63.20	0	37.50		
$s_m = 10, s_d = 2$	0.03	59.38	66.72	0	185.68
$s_m = 10, s_d = 3$	43.73	39.37	74.09		
$s_m = 10, s_d = 4$	55.57	0	26.89		

Tabelle 3.14: Auswertung des MMI-Index, dargestellt ist $10000 \cdot \Lambda$, abweichende Parametrierung für HKT, siehe Tabelle 3.11, Bestwerte sind fett hervorgehoben.

Gegensatz zum besten in Kapitel 2 dargestellten Verfahren auf (MMI-Index: 27.25). Nach Tabelle 3.14 ist für eine Implementierung, die unabhängig von den eingangs erwähnten qualitativen Vorteilen stattfinden soll, das KO_{ind}-Verfahren mit $s_m = 6$ zu bevorzugen. Sein MMI-Index ist 651% größer als der des besten Verfahrens aus Kapitel 2. Für das HKK-Verfahren mit $s_d = 4$ und das TN_{ind}-Verfahren fällt der Index auf Null, da die Anzahl notwendiger Multiplikationen das zulässige Maximum übersteigt.

Für die Verfahren mit dem besten MMI-Index aus Tabelle 3.14 zeigt Tabelle 3.15 den Klassifikationsfehler und den resultierenden MMI-Index nach robustem Anlernen über den zwei Datensätzen B und D₁ gemäß Abschnitt 3.4 und dem Testen über Testdatensatz D₂. Alle Verfahren liefern bessere Ergebnisse als die Verfahren in Kapitel 2 und sind für einen Einsatz in MMIs geeignet.

Klassifikator	MM, MD $s_m = 6$ $s_d = 3$	HKK, MM $s_m = 6$ $s_d = 2$	HKT, MM $s_m^{\min} = 5$ $s_d = 2$	TN _{ind} , MM $s_m = 10$	KO _{ind} , MM $s_m = 6$
Klassifikationsfehler	8.8	8.5	13.1	5.6	6.7
MMI-Index	181.58	159.94	15.08	0	270.68

Tabelle 3.15: Klassifikationsfehler der modifizierten Verfahren über Testdaten (Testdatensatz D₂, 100 Wiederholungen, siehe Abschnitt 2.10.4, abweichende Parametrierung für HKT und MMI-Index ($10000 \cdot \Lambda$))

Die vorgestellten Verfahren wurden speziell für die in der Medizintechnik auftretenden Datensätze und Merkmale entworfen. Sie lassen sich jedoch grundsätzlich mit Erfolg auf alle Probleme anwenden, die

- viele (redundante) Merkmale und Klassen und wenige Beispieldatensätze enthalten,
- vorgegebene Datensätze enthalten, die den Voraussetzungen des AML-Klassifikators (z. B. Normalverteilung) widersprechen und
- die geringe Parameteranzahlen und Rechenzeiten für Klassifikatoren erfordern.

Eine Verallgemeinerung für beliebige Datensätze ist schwierig. Tabelle 7.12 im Anhang zeigt jedoch, dass die Verfahren auch auf in der Literatur bekannte Benchmarkdatensätze angewendet werden können und gute Ergebnisse liefern. Insbesondere zeigt sich auch hier eine Verschiebung des Minimums des Klassifikationsfehlers in Richtung geringerer Parameteranzahlen.

4 Implementierung

4.1 Übersicht

Das Ziel der Implementierung eines MMIs ist das Bereitstellen einer lauffähigen Umsetzung des MMIs in einer portablen Umgebung (z. B. Mikrocontroller). Diese Arbeit setzt die Implementierung eines MMIs aus der Basisimplementierung, der anwendungsspezifischen Implementierung und der patientenindividuellen Implementierung (Bild 4.1) zusammen. Die Basisimplementierung ist für jede Art von MMI (z. B. BCI, EMG-Prothese, etc.) und portabler Umgebung identisch und beinhaltet grundsätzliche Ablaufschemata, wie z. B. die Umsetzung des Zustandsautomaten aus Abschnitt 2.2. Innerhalb der Basisimplementierung befindet sich die anwendungsspezifische Implementierung, die durch die Art des MMIs und die Art der portablen Umgebung spezifiziert wird, aber für alle Patienten gleich ist. Sie beinhaltet Ein- und Ausgabefunktionalitäten des MMIs, z. B. die Umsetzung der Aktionen in den Zuständen des Zustandsautomaten und muss für jede Art von MMI und portabler Umgebung einmalig entworfen werden. Die patientenindividuelle Implementierung setzt sich aus einem strukturvariablen und einem strukturfesten, parametervariablen Teil zusammen. Sie enthält die Klassifikationsalgorithmen aus den Kapiteln 2 und 3, die für jeden Patienten individuell entworfen und angepasst werden. Bei fester anwendungsspezifischer Implementierung kann sie automatisch als Quelltext erstellt werden.

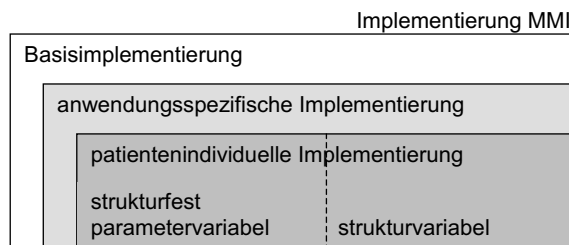


Bild 4.1: Struktur der Implementierung

Um einen sicheren und zuverlässigen Betrieb des MMIs zu gewährleisten, gliedert diese Arbeit den Entwurf eines MMIs in zwei Einheiten (Bild 4.2). In der Entwurfseinheit wird die anwendungsspezifische und die patientenindividuelle Implementierung entworfen, beurteilt und ggf. angepasst, um dann in der Betriebseinheit auf einem Mikrocontroller zur Steuerung eines medizinischen Gerätes (z. B. Handprothese) eingesetzt zu werden.

In der Entwurfseinheit wird ein Aufnahme- und Simulationswerkzeug benötigt, das Daten aufnimmt, online verarbeitet und die Ergebnisse visualisiert. Die prototypische Umsetzung dieses Werkzeuges wird als DAVE (DatenAuswertung und VErarbeitung) bezeichnet. Es dient dem Entwickler des MMIs (z. B. Medizintechniker) und dem Patienten als Unterstützung durch die visuelle Rückmeldung über die Qualität

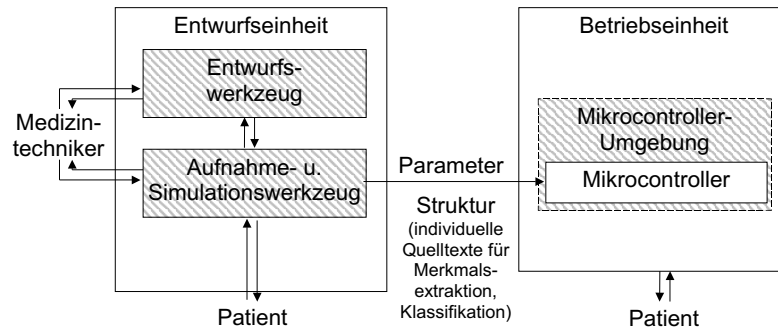


Bild 4.2: Schematische Darstellung zum Entwurf von MMIs

der Steuerung und die Auswirkungen von Änderungen. Da die individuelle Struktur- und Parameteranpassung beim Klassifikatorentwurf rechenaufwändig ist, kann sie i. A. nicht online durchgeführt werden. Deshalb wird ein Entwurfswerkzeug benötigt, das die individuelle Anpassung übernimmt. Die in dieser Arbeit entstandene prototypische Umsetzung wird als DAVE-Design bezeichnet.

Ist der Patient mit der Steuerung zufrieden, findet ein Export der angepassten Strukturen und eingestellten Parameter der anwendungsspezifischen und der patientenindividuellen Implementierung auf einen Mikrocontroller statt. Äquivalent zur Visualisierung in der Entwurfseinheit kann der Patient in der Betriebseinheit das MMI bedienen (z. B. zur Steuerung von Geräten oder weiteren Simulationen). Um den Mikrocontroller komfortabel zu programmieren, zu testen und zu visualisieren, besteht eine Mikrocontrollerumgebung, in die er eingefügt werden kann. Diese Umgebung beinhaltet Ein- und Ausgabefunktionalitäten zur Kommunikation mit dem Patienten und eine Simulation des medizinischen Gerätes.

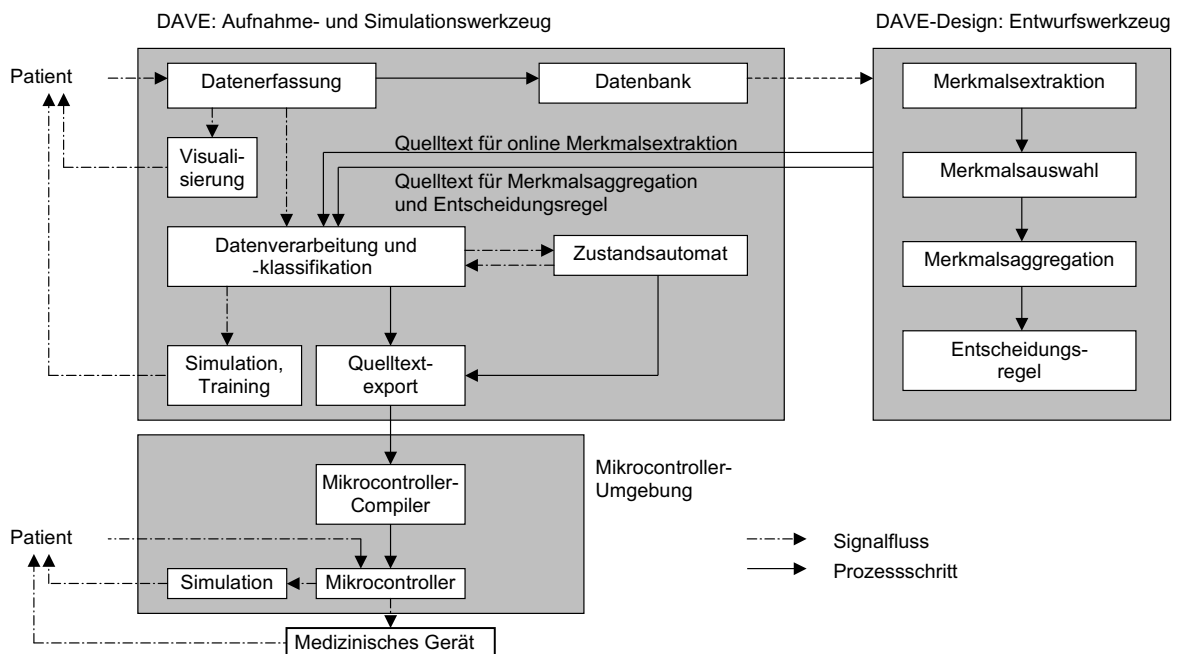


Bild 4.3: Prozessschritte und Signalfluss zum Entwurf einer patientenindividuellen Steuerung

Die Kommunikation der drei Verarbeitungseinheiten untereinander und mit dem Patienten stellt Bild 4.3 dar. Die vom Patienten generierten Biosignale/Daten werden von DAVE erfasst und einer Datenbank zugeführt. Diese stellt die Kompatibilität der Daten zu DAVE-Design sicher. DAVE-Design extrahiert aus den Daten Merkmale und entwirft die Merkmalsauswahl, -aggregation und Entscheidungsregel. Als Ergebnis liefert DAVE-Design nicht nur die berechneten Parameter, sondern den kompletten, individuell angepassten Quelltext für Merkmalsextraktion, -auswahl, -aggregation und Entscheidungsregel an DAVE zurück. Der Quelltext fungiert als Bauteil und kann ohne jegliche Anpassung in DAVE integriert werden. Eine Datenverarbeitungs- und Klassifikationsroutine wendet den Quelltext des exportierten Klassifikators auf vom Patienten generierte Signale an. Eine Visualisierung ist ebenfalls möglich. In Kombination mit dem Zustandsautomaten ergibt sich die Steuerung. Diese visualisiert DAVE in einer Simulations- oder Trainingseinheit und generiert dem Patienten ein Feedback über ihre Funktionsweise und Qualität. Ist diese zufriedenstellend, folgt die Implementierung in eine Mikrocontroller-Umgebung. Der Quelltext wird dazu in einen Mikrocontroller-Compiler exportiert und auf einen Mikrocontroller gebrannt. Dieser kann entweder über eine Simulation mit dem Patienten interagieren oder von der Entwurfseinheit getrennt in einem medizinischen Gerät (z. B. Handprothese) eingesetzt werden.

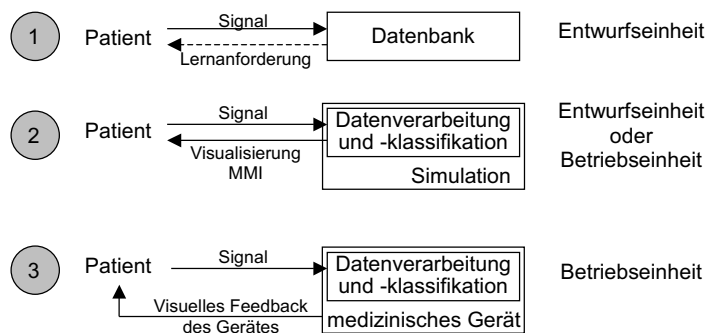


Bild 4.4: Schema des Entwurfsvorgangs

Der Entwurfsprozess eines MMIs gliedert sich gemäß Bild 4.4. Im ersten Schritt werden dem Patienten Lernanforderungen vorgegeben, gemäß denen er Signale generiert, die in einer Datenbank gespeichert werden. Dieser Schritt findet in der Entwurfseinheit statt. Im zweiten Schritt steuert der Patient eine Simulation des MMIs und erhält eine Visualisierung der Reaktion des MMIs. Die Datenverarbeitungs- und -klassifikationsalgorithmen basieren auf den in der Datenbank gespeicherten Signalen. Dieser Schritt kann sowohl in der Entwurfs- als auch in der Betriebseinheit stattfinden. Im dritten Schritt arbeitet das MMI autark (z. B. in Form eines Mikrocontrollers) und steuert das medizinische Gerät in Abhängigkeit von den Signalen des Patienten. Der Patient erhält ein visuelles Feedback durch den Zustand des medizinischen Gerätes. Dieser Schritt findet ausschließlich in der Betriebseinheit statt. Im Folgenden wird die Umsetzung der drei dargestellten Werkzeuge für Entwurfs- und Betriebseinheit diskutiert.

4.2 Werkzeuge

4.2.1 Datenauswertung und Verarbeitung (DAVE)

Die in Visual C++ entworfene Aufnahme- und Simulationseinheit DAVE ist als grafische Benutzeroberfläche implementiert und enthält Routinen zur komfortablen Aufnahme und Visualisierung von Biosignalen, zur Simulation, zum Training und zur Evaluierung von MMI-Steuerungen (Bild 4.5). Die Namen, Aufgaben und den Umfang der einzelnen Funktionen stellt Anhang 7.3, Tabelle 7.13 dar.

Die Datenerfassung mit DAVE basiert auf einer Analog-Digital(A/D)-Wandler-Karte, die bis zu acht analoge Sensoren parallel mit einer maximalen Abtastfrequenz $f_a = 10000\text{Hz}$ abtastet. Die abgetasteten Signale werden in eine DAVE-Design-kompatible Datenbank geschrieben.

Als Visualisierung kann zwischen der Anzeige des Signals als Zeitreihe (inklusive Vergrößerungsoptionen), der Anzeige als fortlaufende Kurve und der alphanumerischen Darstellung der aktuellen Amplituden mit einer Maximal-Minimalwert-Berechnung gewählt werden.

Die Simulationsplattform stellt das Zusammenspiel von Klassifikator und Zustandsautomat dar. Hierbei lassen sich Filterparameter, etc. online manipulieren. Die Zustände werden durch Zustandsbilder des entsprechenden medizinischen Gerätes visualisiert. Als Beispiel für ein Gerät ist in Bild 4.5 eine Handprothese dargestellt. Die Simulation ist modular aufgebaut, so dass sie durch einfaches Ersetzen der Bilder mit jedem MMI verwendet werden kann.

Die Trainingsplattform gliedert sich in zwei Teile. Erstens können Schaltsignale in zufälliger Reihenfolge vorgegeben werden, die dann vom Anwender zu wiederholen sind. Zweitens ist eine Variante des Spiels Tetris auf der Basis bioelektrischer Signale integriert. Die Evaluierung des von DAVE-Design generierten Quelltextes kann objektiv anhand mehrfacher Vorgabe nachzuahmender Schaltsignale oder subjektiv anhand der Zufriedenheit des Patienten geschehen. Nähere Angaben finden sich in [159, 160, 161, 201].

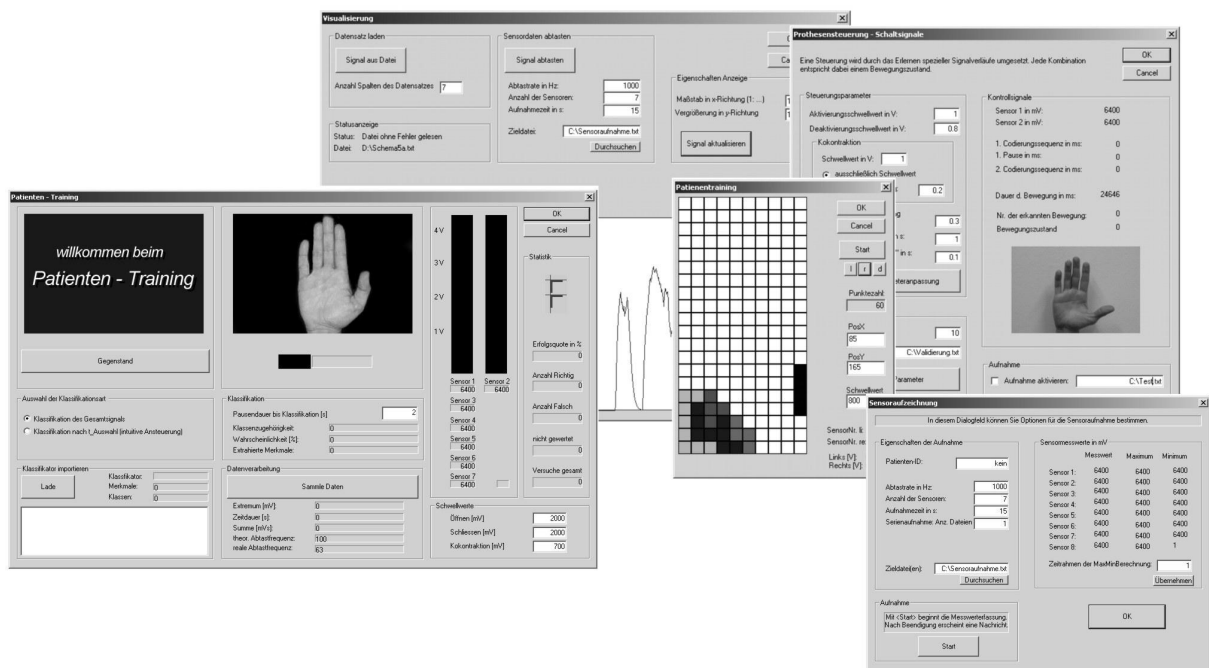


Bild 4.5: DAVE: Plattform zur Auslegung und Evaluierung von MMIs

4.2.2 DAVE-Design

Das in MATLAB entwickelte Entwurfswerkzeug DAVE-Design wurde als MATLAB-Toolbox entworfen und vereint alle Entwurfsverfahren der Kapitel 2 und 3 sowie Import-, Visualisierungs-, Konvertierungs- und Validierungsverfahren. Zudem sind Routinen zum Quelltextexport für die Merkmalsextraktion und

für die wichtigsten Entscheidungsregeln implementiert. Alle Funktionen sind über eine grafische Benutzeroberfläche zugänglich (Bild 4.6). Die Toolbox kann jedoch auch unabhängig von der Oberfläche für beliebige Klassifikationsaufgaben verwendet werden. Die Namen, Aufgaben und den Umfang aller Funktionen stellt Anhang 7.3, Tabelle 7.15 dar.

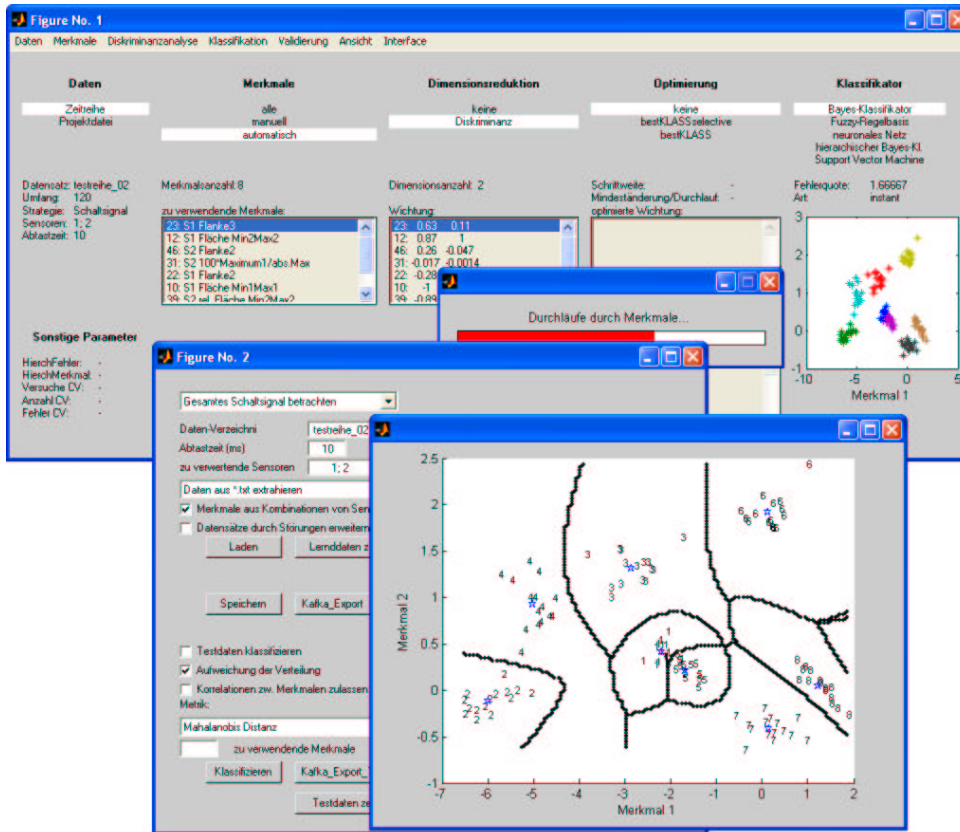


Bild 4.6: DAVE-Design: Plattform zum Entwurf von Klassifikatoren auf der Basis von importierten Zeitreihen aus DAVE

DAVE-Design baut auf der MATLAB-Toolbox KAFKA auf [113]. Im Vergleich zu KAFKA erweitert DAVE-Design den Umfang der Klassifikationsroutinen und führt ein einheitliches Schema zum Entwurf von Klassifikatoren für bioelektrische Zeitreihen ein. So können Klassifikatoren modular aufgebaut, trainiert und getestet werden.

Zur Übernahme von Daten aus DAVE existieren Importroutinen. Daten lassen sich als Zeitreihe und im Merkmalsraum visualisieren. Ebenso kann der Merkmalsraum um Klassifikatorgrenzen erweitert werden.

Konvertierungsroutinen eröffnen die Möglichkeit, eine Datenauswahl oder -erweiterung für den Klassifikatorentwurf durchzuführen. Ebenso ist eine Erweiterung der Anzahl der Merkmale oder der Anzahl der Klassen (durch Clusterung) möglich. Weiterhin sind alle in Abschnitt 2.10.1 dargestellten Validierungstechniken implementiert.

Für die Merkmalsextraktion ist eine rechen- und speichereffiziente Routine zur automatischen Quelltextgenerierung implementiert. Ebenso existieren Routinen zur Quelltextgenerierung für die Merk-

malsaggregation und für die wichtigsten Entscheidungsregeln, die Fuzzy-Regelbasis, die AML-Entscheidungsregel sowie für die modifizierten Mehrschrittverfahren TN_{ind} und KO_{ind} .

4.2.3 Mikrocontroller-Umgebung

Die Mikrocontroller-Umgebung besteht aus den Elementen Mikrocontroller-Compiler und der entwickelten Plattform T&E-Box (Bild 4.7), die den Mikrocontroller beinhaltet. Die T&E-Box simuliert den Betrieb des Mikrocontrollers in einem medizinischen Gerät und stellt hierfür Ein- und Ausgabe-funktionalitäten zur Verfügung sowie Schnittstellen zur Programmierung oder Auswertung mittels eines externen Rechners. Die Namen, Aufgaben und den Umfang der Funktionen für die Mikrocontrollerprogrammierung stellt Anhang 7.3, Tabelle 7.14 dar.

Als Mikrocontroller kommt ein 8-Bit ATMEL ATmega128 zum Einsatz, der als Mikrocontrollermodul, bestückt mit Speicher und Quarz, erhältlich ist (Abmessungen: $45 \times 46 \text{mm}$, Stromverbrauch: max. 30mA). Der Controller ist mit 20MHz getaktet und benötigt für eine Addition 0.008ms , für eine Multiplikation 0.022ms und für eine Berechnung der Exponentialfunktion 0.42ms ¹. Er bietet 128kB Flash-Speicher für Programmcode sowie 4kB RAM. Gewählt wurde das Modul aufgrund seines geringen Preises, seiner geringen Leistungsaufnahme sowie des Befehlssatzes des Controllers, der für Hochsprachen optimiert ist. Im Bereich der 8-Bit Controller markiert er das obere Ende der Leistungsfähigkeit.

Bild 4.7 zeigt den Aufbau der T&E-Box sowie das integrierte Mikrocontrollermodul. Es ist auf eine Sockelplatine aufgesetzt, um bei Bedarf andere Controller einsetzen zu können. Mittels des Notebookanschlusses lässt sich die T&E-Box mit einem externen Computer verbinden, Eingabemöglichkeiten sind durch Schalter und Taster gegeben. Ausgaben werden zusätzlich zu einem LC-Display durch Leuchtdioden dargestellt.

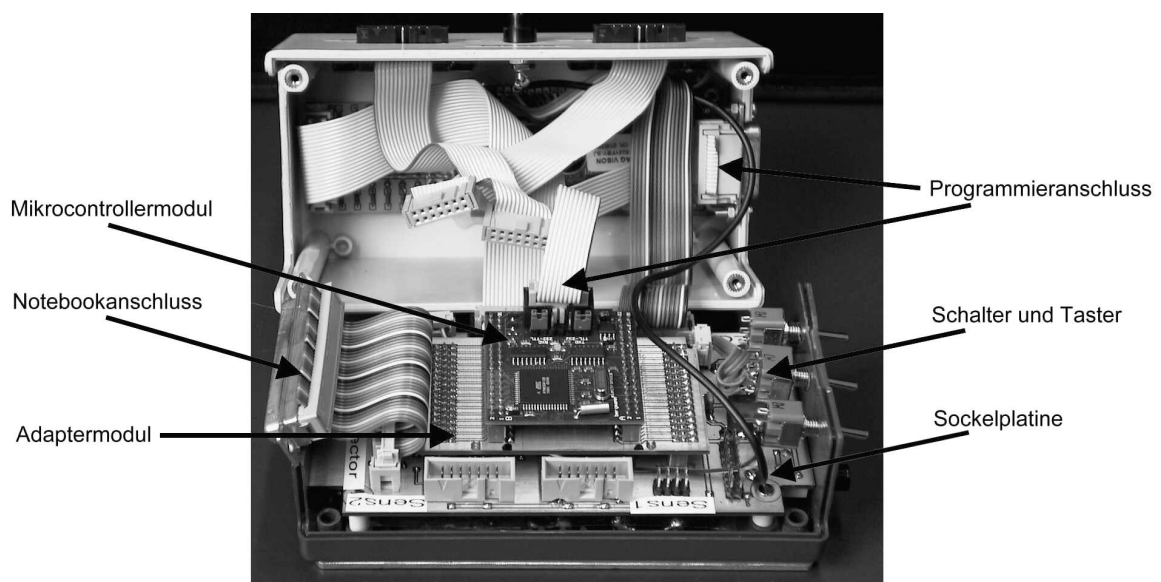


Bild 4.7: Integration der Module der Testumgebung im Gehäuse der T&E-Box [88]

¹Der Ansatz der vorangegangenen Kapitel, den Rechenaufwand einer Exponentialfunktion gleich dem von 20 Multiplikationen zu setzen, ist damit gerechtfertigt.

An die T&E-Box lassen sich direkt Sensoren anschließen, deren Signale bei Bedarf an einen externen Computer weitergeleitet werden können (z. B. zur Datenaufnahme mit DAVE) oder die direkt die aufgespielte Steuerung bedienen. Die T&E-Box arbeitet dann autark entsprechend Schritt 2 in Bild 4.4 und dient als Versuchsumgebung für die Funktionalität der Programmierung des Mikrocontrollers. Bei Bedarf kann dieser einfach entfernt und in das dafür vorgesehene medizinische Gerät integriert werden. Einzelheiten sind [88] zu entnehmen.

4.3 Quelltextexport

4.3.1 Merkmalsextraktion und Merkmalsauswahl

Während die anwendungsspezifische Implementierung für eine Kombination aus Art des MMIs und Art des Mikrocontrollers einmalig manuell entworfen werden muss, ist die patientenindividuelle Implementierung für jeden Patienten neu zu entwickeln. Um Entwicklungsaufwand einzusparen, bietet sich der Export des patientenindividuell angepassten Codes an. Von Vorteil ist hierbei, dass nur benötigte Teile exportiert werden und somit Rechenzeit und Speicherplätze im Online-Betrieb des Mikrocontrollers eingespart werden.

Einen Automaten zur Online-Berechnung der in Abschnitt 2.6.2 vorgeschlagenen Merkmale zeigt Bild 4.8. Dieser Automat sei als vollständig bezeichnet, da alle Merkmale ohne Berücksichtigung einer Merkmalsauswahl berechnet werden. Der Automat teilt sich in zwei symmetrische Hälften, wobei die rechte Hälfte die Minimumsuche und die linke Hälfte die Maximumsuche beschreibt.

Der Aufruf des Automaten startet mit einer Suche nach dem ersten Minimum für den Bereich $i = 1$. Hierbei wird zu jedem Zeitpunkt k das Sensorsignal $u[k]$ eingelesen, zu $u_f[k]$ gefiltert und bei Bedarf die Zahl der Filterschnittpunkte $u_{sp}[k]$ des Bereichs i erhöht. Es folgt eine Aktualisierung von Summensignal $u_S[k]$, Trend $u_D[k]$ und gefilterter Standardabweichung $u_\sigma[k]$. Ist $u_f[k]$ kleiner als das bisherige Minimum $u_{f,\text{Min}}$, so wird Letzteres aktualisiert. Bei Bedarf wird das Minimum des Trends $u_{D,\text{Min}}$ ebenfalls aktualisiert.

Wird das Ende eines Bereichs i aufgrund des Überschreitens der Hysterese $u_{\text{Min},h}$ erreicht, so werden die Merkmale gespeichert, alle Variablen zur Zwischenspeicherung neu initialisiert und der Bereichszähler i inkrementiert. Der Zustand des Automaten wechselt von einer Minimumsuche zu einer Maximumsuche. Beendet wird der Kreislauf, wenn die maximale Anzahl $i_{\text{max}} = n_e$ gesuchter Extremwerte erreicht oder für eine Dauer k_{max} kein Merkmal mehr generiert wurde.

Für die Auswertung eines Schaltsignals ist für jede Zeitreihe $j \in [1, \dots, n_s^*]$ eine Instanz dieses Automaten mit unabhängigen Variablen und Zuständen zu implementieren.

Um einen recheneffizienten Quelltext zu generieren, ist die Funktion zur Berechnung der Merkmalsautomaten parametervariabel zu entwerfen. Parametervariabel umgesetzt wird die Anpassung der Sensoranzahl n_s , der Zeitreihenanzahl n_s^* , der Bereichsanzahl n_e sowie alle zur Berechnung der Merkmale notwendigen Parameter. So kann ausschließlich die Berechnung von ausgewählten Merkmalen in jeder Automateninstanz durchgeführt werden. Jedoch bestehen Abhängigkeiten bei der Extraktion von Merkmalen - Merkmalsberechnungen können nicht beliebig ausgelassen werden. Nach Bild 4.9 bauen alle Merkmalsberechnungen auf der Berechnung von Extrema auf. Kommt beispielsweise die gefilterte Standardabweichung als Merkmal zum Einsatz, so müssen auch Extrema und Extrema des Trends berechnet werden. Eine ausführliche Beschreibung gibt [89].

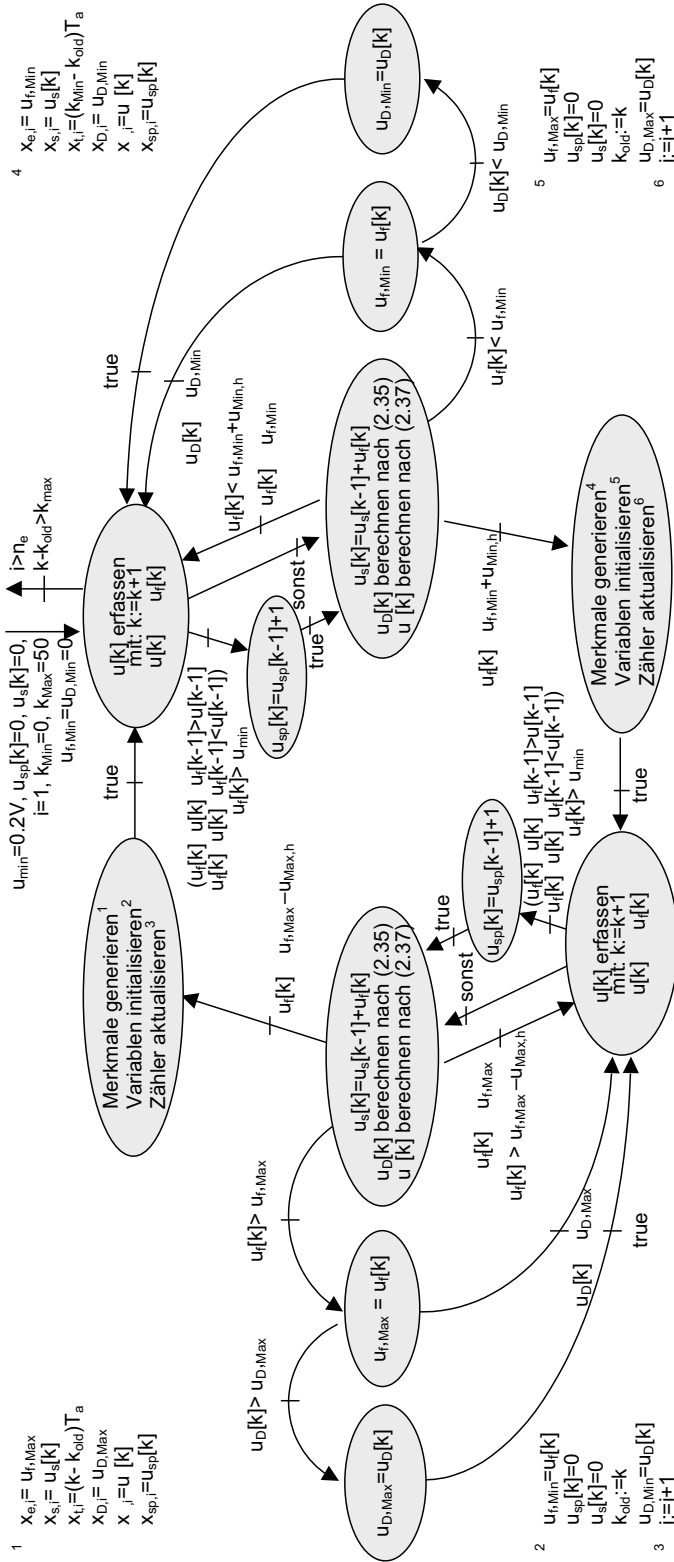


Bild 4.8: Automat zur Berechnung von Merkmalen im zeitdiskreten Signal aus einer Zeitreihe $u[k]$

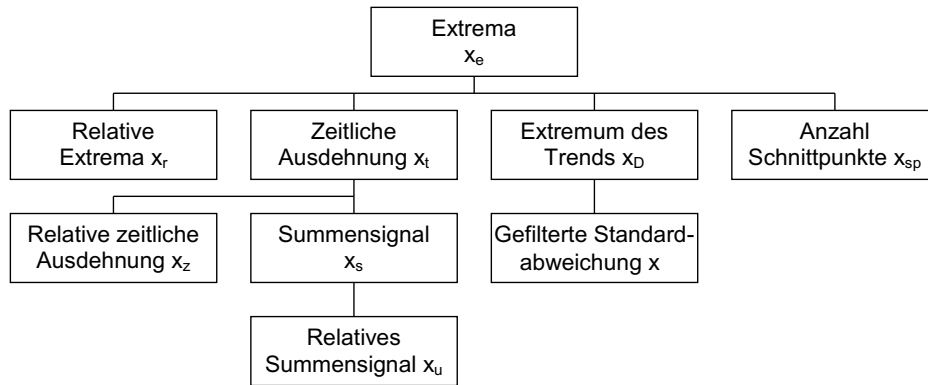


Bild 4.9: Abhängigkeiten der Merkmalsgenerierung [156]

Die maximale Anzahl an Multiplikationen und Speicherplätzen entsteht, wenn für alle n_s^* Zeitreihen Merkmale in gleichen Zeitbereichen i extrahiert werden. Das rechenintensivste Merkmal ist hierbei die gefilterte Standardabweichung. Die maximale Anzahl notwendiger Multiplikationen beträgt dann $11n_s^{*2}$.

Das speicheraufwändigste Merkmal ist ebenfalls die gefilterte Standardabweichung. Die maximale Anzahl notwendiger Speicherplätze ergibt sich zu $14n_s^{*3}$.

4.3.2 Merkmalsaggregation und Entscheidungsregel

Die Routinen zum Quelltextexport von Merkmalsextraktion und -auswahl bestehen aus struktur- und parametervariablen Teilen, die sich in einen festen Rahmen fügen. Exportroutinen sind für die beiden besten Verfahren aus Kapitel 2, die Fuzzy-Regelbasen und AML-Entscheidungsregeln (und somit für alle Einschrittverfahren aus Abschnitt 3.2) entworfen worden, ebenso für die TN- und KO-Verfahren.

Bild 4.10 zeigt exemplarisch den automatisch generierten, individuellen Quelltext eines auf der Basis von Datensatz A entworfenen AML-Klassifikators ($s_m = 8, s_d = 3$). Alle Parameter entstammen der zugrundeliegenden MATLAB-Routine und werden in den Quelltext übertragen. In Abhängigkeit von Klassenanzahl K und Anzahl aggregierter Merkmale s_d werden die Parameterdeklarationen strukturell angepasst. Selbiges gilt für die Ausführung der Merkmalsaggregation. Bei Verzicht auf eine solche entfällt der entsprechende Quelltext ersatzlos. Die Berechnung der Entscheidungsregel erfolgt ebenfalls strukturvariabel, bei der Variation der Metrik ändern sich die For-Schleifen-Konstrukte. Die Auswertung der Entscheidungsmaße ist für alle AML-Klassifikatoren identisch.

²Für jede Automateninstanz ist zu jedem Zeitpunkt eine Filterung des Signals (2 Multiplikationen), eine Berechnung des Trends (5 Multiplikationen) und die Berechnung der gefilterten Standardabweichung (4 Multiplikationen) zu bewerkstelligen. Für $s_m \geq n_s^*$ müssen keine weiteren Merkmale durch Multiplikationen berechnet werden.

³Alle Konstanten werden fest im Quelltext vorgegeben. Benötigt werden 7 Speicherplätze für $u[k], u[k-1], u_f[k], u_{f,\min}$ bzw. $u_{f,\max}, k_{\text{old}}, k, i$ und für die Berechnung der gefilterten Standardabweichung 6 Speicherplätze $u_\sigma[k], u_\sigma[k-1], u_S[k], u_S[k-1], u_L[k], u_L[k-1]$ sowie ein Speicherplatz zum Speichern des Ergebnisses. Für $s_m \geq n_s^*$ beträgt die maximale Anzahl $14n_s^* + 10(s_m - n_s^*)$ (zweitaufwändigstes Merkmal: Trend, jedoch schon berechnet, nächstaufwändigere Merkmale: x_{sp} und x_s mit zwei Hilfsvariablen).

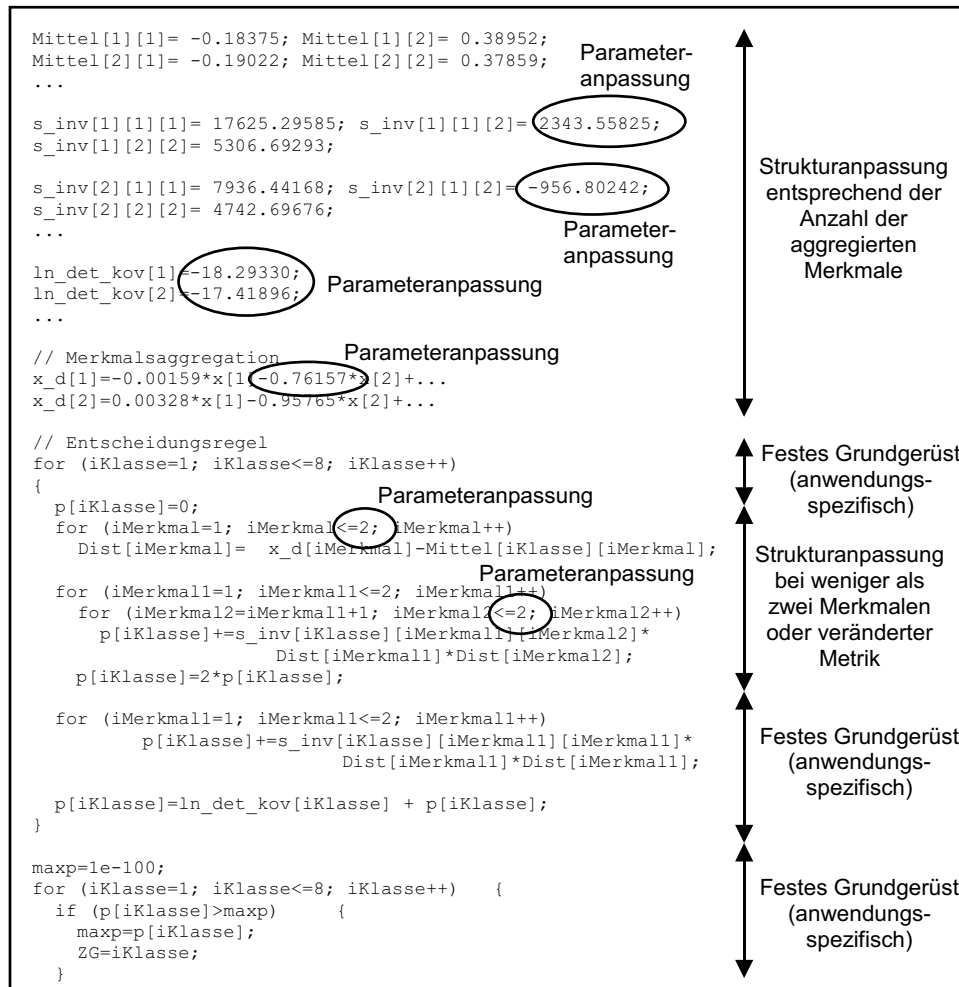


Bild 4.10: Patientenindividuelle Implementierung: Beispiele für angepasste Parameter und Strukturen beim automatischen parameter- und strukturvariablen Quelltextexport

4.4 Bewertung

Die vorgestellten Entwurfsschemata, Werkzeuge und patientenindividuellen Implementierungsmöglichkeiten lassen sich für den Entwurf beliebiger, auf Biosignalen basierender MMIs einsetzen. Manuell zu realisieren ist nur das einmalige Anpassen der anwendungsspezifischen Implementierung, um die individuellen Vorgaben von medizinischem Gerät und Mikrocontroller zu erfüllen. Bei Handprothesen beinhaltet dies das Festlegen der Art und Anzahl der auszuführenden Griffarten (Trajektorien für Fingerbewegungen, Zuordnung von Schaltsignalen zu Griffarten) sowie das Festlegen der Betriebsparameter des Mikrocontrollers (Taktfrequenz, Abtastfrequenz, etc.). Die patientenindividuelle Implementierung kann dann vollautomatisch generiert und auf den Mikrocontroller übertragen werden. Ein manuelles Eingreifen in den Entwurfsprozess ist selbst bei Anpassungen von Merkmalen und/oder Klassifikatoren nicht mehr notwendig.

5 Anwendungen

5.1 Übersicht

Dieses Kapitel zeigt am Beispiel der Steuerung myoelektrischer Handprothesen, wie der vollständige Entwurf eines MMIs mit dem neuen Konzept durchzuführen ist (Abschnitt 5.2).

Zur Umsetzung des MMIs werden Anforderungen definiert (Abschnitt 5.2.1) und anatomische Möglichkeiten des Patienten erörtert (Abschnitt 5.2.2). Es folgt die anwendungsspezifische Anpassung des MMIs (Abschnitt 5.2.3). Die Integration des angepassten MMIs in die Prothese (technisches System gemäß Bild 1.2) erfolgt durch die Spezifikation der Steuerungseinheit, die die vorhandene Aktorik bedient (Abschnitt 5.2.4).

Die Güte des entworfenen MMIs wird anhand der Daten von 17 armamputierten Patienten und zwei Probanden belegt. Es finden Untersuchungen zu Reproduzierbarkeit, Auswirkungen von Training, Leistungsfähigkeit der Klassifikatoren und Robustheit statt (Abschnitt 5.2.5 bis 5.2.10).

Die Entwicklung eines zweiten MMIs auf der Basis von EEG-Signalen zeigt die Übertragbarkeit des Konzepts auf andere MMIs (Abschnitt 5.3).

5.2 Myoelektrische Handprothesen

5.2.1 Anforderungen

Bereits 1992 legte [35] die Ansprüche an eine ideale Handprothesensteuerung fest, die nach [207] noch immer als wünschenswerte Ziele angesehen werden. Dies sind:

1. Unabhängige Bedienbarkeit der multifunktionellen Steuerung: Mögliche Griffarten sollen unabhängig vom Ausgangszustand angesteuert werden können.
2. Geringe geistige Belastung: Der Anwender muss sich nicht explizit auf seine Prothese konzentrieren.
3. Benutzerfreundliche und/oder einfach zu erlernende Steuerung: Der Trainingsaufwand minimiert sich.
4. Keine Beeinträchtigung der übrigen Körperfunktionen: Zum Steuern der Prothese werden nur Muskeln des Armstumpfs verwendet.
5. Natürliches Aussehen: Die Koordination der Prothesenfinger muss beim Schließen entsprechend der menschlichen Hand geschehen.

6. Simultane Steuerung mehrerer Funktionen: Das Kombinieren einzelner Elemente der multifunktionalen Steuerung soll möglich sein.
7. Direkter Zugang und sofortige Antwort: Die Steuerung muss ohne merkbare Zeitverzögerung auf die Aktionen des Anwenders reagieren.

Zwar erhöhen 6. und 7. den Komfort für den Anwender, jedoch reduzieren sie die Robustheit einer Steuerung enorm. Gängige Forschungsvorhaben setzen die Punkte nicht gemeinsam um. Die vorliegende Arbeit verzichtet ebenfalls darauf.

Bislang erfordert der Einsatz von kommerziellen myoelektrischen Handprothesen ein Trainingsprogramm von mehreren Wochen und viele Patienten geben an, dass eine zufriedenstellende Steuerbarkeit der Prothese erst nach mehreren Monaten erreicht wird. Dies ist ein einschränkender Punkt für den Einsatz und die Akzeptanz herkömmlicher EMG-Prothesen [13].

Ferner ist das Anpassen der Prothese an den Patienten aufwändig, da Sensormessstellen, Verstärkungsfaktoren, etc. heuristisch gesucht und eingestellt werden müssen.

Bei der Umsetzung der Bewegung beschränken sich viele Arbeiten darauf, dem Patienten nicht einzelne Fingerbewegungen, sondern ganze Griffarten zugänglich zu machen. Allen ist gemein, dass die Bewegungsabsicht aus dem ersten Zeitabschnitt des EMG-Signals geschätzt wird (Schalten ohne Zeitversatz, [52, 61, 75, 122]). Dieses Verfahren funktioniert gut für gesunde Probanden und kann auch angewendet werden, wenn der Patient eine Phantomhand¹ besitzt. Es scheitert jedoch, wenn der Patient unter einem kongenitalen² Verlust seines Arms leidet. Die Kontraktion einzelner Muskelgruppen ist dann zwar möglich, die Kombination der Muskelgruppen, die für eine Bewegung notwendig ist, kann jedoch nicht reproduziert werden. Eine Griffartenauswahl muss jedoch für alle Patienten gleichermaßen möglich sein.

Die angeführten anpassungsspezifischen Probleme führen somit zu einer Erweiterung der Anforderungen um

- eine individuell an den Patienten adaptierte Steuerung, die den langwierigen Lernprozess seitens des Patienten verkürzt,
- eine Sensoranordnung, die wenig sensibel auf Abweichungen in Position oder Einstellung reagiert und somit für alle Patienten anwendbar ist und
- ein Steuerungskonzept, das von jedem Patienten unabhängig der Anatomie seines Armstumpfs bedient werden kann.

Neben den erweiterten Anforderungen ist die Bewegungsabsicht des Patienten so auszuwerten, dass zumindest die fünf wichtigsten Griffarten umgesetzt werden können, die der Mensch im Alltag verwendet [190]. Dies sind nach Bild 5.1

- der Hakengriff, ein Griff zum Tragen schwerer Gegenstände,
- die isolierte Streckung des Zeigefingers, z. B. um eine Tastatur zu bedienen,
- der Lateralgriff, zum Halten eines flachen Objekts, wie etwa eines Schlüssels oder einer Kreditkarte,
- der sphärische Griff (Zylindergriff), ein Kraftgriff, bei dem alle fünf Finger zum Greifen mit einbezogen sind und
- der Pinzettengriff, ein Präzisionsgriff zum Halten kleiner Objekte.

Sofern es der Patient wünscht, muss die Möglichkeit der Umsetzung weiterer Griffarten gegeben sein.

¹Der Patient fühlt seine verlorene Hand noch und kann diese auch virtuell bewegen.

²von Geburt an

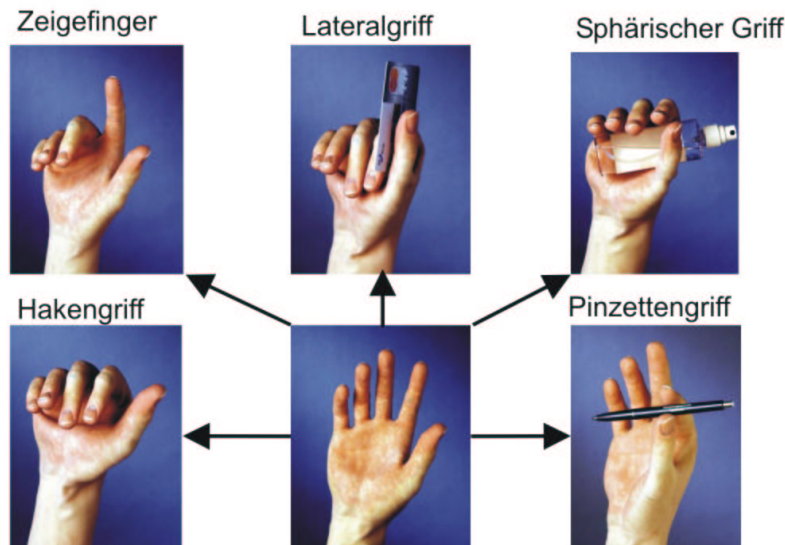


Bild 5.1: Umsetzung von Griffarten in einer Prothese

5.2.2 Umsetzbare Schaltsignale

Durch das adaptive System zum Entwurf von MMIs unter Verwendung des Schaltens mit Zeitversatz können die diskutierten Anforderungen gut erfüllt werden. Als Schaltsignale kommen Aktivitätssignale von willentlich reproduzierten Muskelkontraktionen zum Einsatz. Abgesehen von extrem kurzen Armstümpfen verfügt ein Patient über mindestens zwei Stumpfmuskeln (z. B. Flexoren und Extensoren des Handgelenks), in manchen Fällen sogar drei oder mehr. Sofern er diese trainiert, besitzt er folgende Möglichkeiten, Kontraktionen zu generieren:

1. Jede Muskelgruppe kann abgesehen von einem Übersprechen der Muskelgruppen separat kontrahiert werden.
2. Die Dauer einer Kontraktion kann zeitlich variiert und bis auf wenige zehntel Sekunden verkürzt werden.
3. Die Intensität einer Kontraktion kann in ihrer Amplitude variiert werden und bestimmt den Ausschlag des gemessenen EMG-Signals.
4. In vielen Fällen können Drehbewegungen des proximalen Gelenks durchgeführt werden (bei Verlust einer Hand somit die Drehbewegung des Ellbogens).
5. Durch die sequentielle Ausführung von Kontraktionen ergeben sich Kontraktionssequenzen.
6. Eine Kombination von Muskelgruppen kann kontrahiert werden, es ergeben sich Kokontraktionen.

Gemäß dieser Vorgaben lassen sich Kontraktionsmuster zur Verwendung als Schaltsignale in Schaltsignalkategorien einteilen [164, 202], jede Schaltsignalkategorie setzt sich wiederum aus Schaltsignaltypen zusammen:

- Zeitvariierte Schaltsignale enthalten Einzelkontraktionen, die sich in ihrer zeitlichen Ausdehnung und der dazwischenliegenden Pausen unterscheiden. So sind Bewegungsabsichten z.B. durch Schaltsignaltypen wie "kurz", "lang", "kurz-lang", "kurz-kurz", "lang-kurz" oder "lang-lang" kodiert.

- Amplitudenvariierte Schaltsignale bestehen aus zwei Einzelkontraktionen, die sich in ihrer Amplitude unterscheiden, beispielsweise sind Schaltsignaltypen wie "schwach-stark" oder "stark-schwach" denkbar.
- Bei intuitiven Schaltsignalen handelt es sich um Schaltsignale, die bestimmte Bewegungen des Armstumpfs repräsentieren. So ist es naheliegend und physiologisch sinnvoll, eine Pronation/Supination des Armstumpfs direkt in eine Prothesenbewegung "Handgelenksdrehung links/rechts" umzusetzen, oder mittels einer Handgelenksflexion/-extension ein Schließen bzw. Öffnen des Griffmusters anzusteuern.
- Abstrakte Schaltsignale bezeichnen einen beliebigen Verlauf der Sensorspannung, den der Patient zuverlässig wiederholen kann.
- Kombinierte Schaltsignale beschreiben Mischformen aus den oben genannten Kontraktionsmustern, Beispiele für Schaltsignaltypen sind "kurz/schwach-lang/stark" oder "lang/stark-lang/schwach", etc.

Alle oben genannten Schaltsignale sind darüber hinaus auch in Kombination mehrerer Muskelgruppen möglich. Zum Beispiel kann bei der Verwendung je eines Sensors an Handgelenksflexor und -extensor ein Kontraktionsmuster "Flexor kurz - Extensor kurz" generiert werden.

5.2.3 Anwendungsspezifische Anpassung für Handprothesen

Die multifunktionale Steuerung einer Handprothese stellt hohe Anforderungen an den Anwender, da bei allen Forschungsarbeiten gleichzeitig sowohl Bewegungsart als auch -geschwindigkeit bestimmt werden müssen. Dies mit nur einer Muskelkontraktion zu erreichen, fällt gerade Patienten schwer, die unter einem kongenitalen Verlust ihrer Hand leiden. Um allen Patienten eine multifunktionale Steuerung zur Verfügung zu stellen und gleichzeitig viele Bewegungen mit einer geringen Anzahl an Muskeln ausführen zu können, eignet sich das Schalten mit Zeitversatz sehr gut. Die Auswahl der Griffart und die Ausführung der Griffart (Auswahl der Bewegungsgeschwindigkeit) erfolgen im Gegensatz zu anderen Arbeiten voneinander getrennt [152, 162]. Zwar erhöht sich die Dauer der Griffartenauswahl, gleichzeitig verbessert sich hingegen deren Zuverlässigkeit.

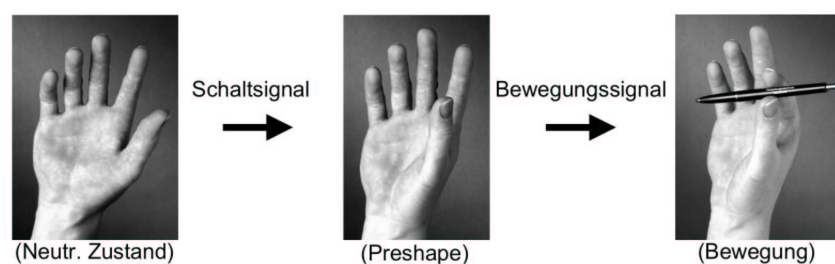


Bild 5.2: Entkopplung von Bewegungsauswahl und -steuerung: Ausgehend vom neutralen Zustand wird durch ein Schaltsignal ein Preshape (Vorbewegungszustand) aktiviert, Bewegungssignale steuern das Öffnen bzw. Schließen der Bewegung

Das Umsetzen des Schaltens mit Zeitversatz erfolgt durch das Abtasten von zwei antagonistischen Muskeln des Armstumpfs mit EMG-Sensoren. Die Auswahl der Griffart geschieht durch Schaltsignale, die durch die beiden abgetasteten Muskelsignale dargestellt werden. Das Erkennen eines Schaltsignals führt

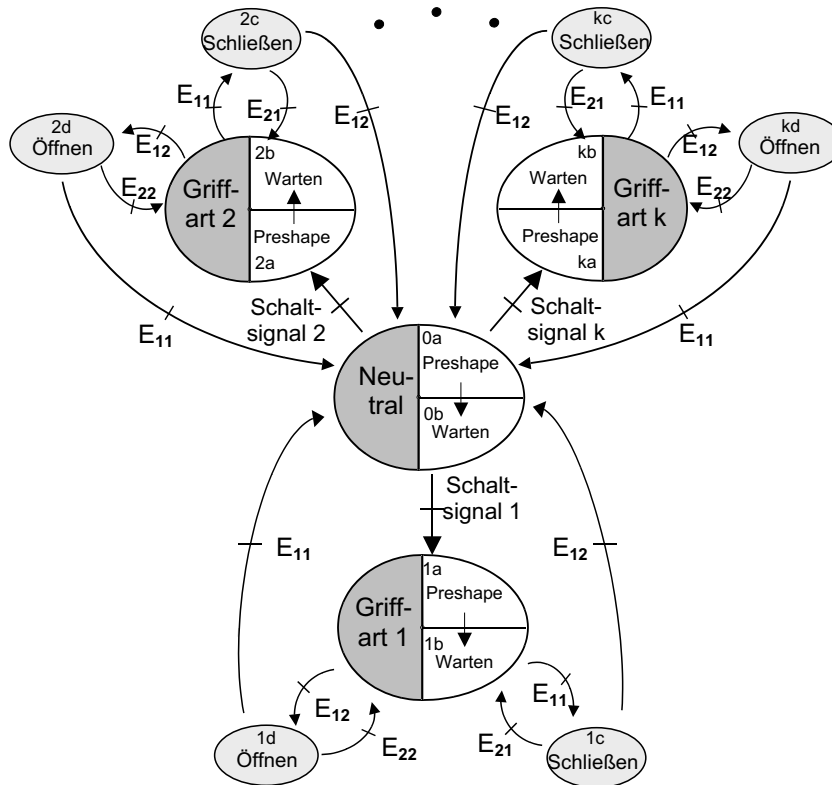


Bild 5.3: Zustandsautomat zur Umsetzung von k Griffarten unter Verwendung von zwei EMG-Sensoren. E_{11} : Kontraktion von Muskel 1, E_{12} : Kontraktion von Muskel 2, E_{21} : Relaxation von Muskel 1, E_{22} : Relaxation von Muskel 2, $E_{11} \wedge E_{12}$: Kokontraktion

in einen Vorbewegungszustand, den sogenannten Preshape. Das Ausführen der Bewegung wird durch Bewegungssignale gestartet (Bild 5.2)³. Ein Bewegungssignal bezeichnet den Verlauf der EMG-Spannung, der bei ausgewählter Bewegung zum Schließen bzw. Öffnen der Bewegung dient. Die physikalische Größe zur Darstellung der Signalinformation ist beim Bewegungssignal die aktuelle Amplitude des größeren der zwei EMG-Sensoren. So können das Öffnen und das Schließen der Prothese getrennt voneinander bedient werden, indem einer der beiden Muskeln kontrahiert wird.

In [159] wird der Zustandsautomat beschrieben, der die Umsetzung in Prothesenbewegungen koordiniert (Bild 5.3, 5.4 und 5.5). Nach Abschnitt 2.2 handelt es sich um einen eingeschränkten Zustandsautomaten mit insgesamt $K + 1$ Zuständen (Zustände $1 \dots K$ (Bewegungszustände), neutraler Zustand). Die Zustandsübergänge sind ereignisgesteuert, für die vom neutralen Zustand erreichbaren K Zustände kommen Schalt-signale zum Schalten mit Zeitversatz als Transition zum Einsatz. Der Übergang zurück in den neutralen Zustand findet durch eine Kokontraktion der beiden abgetasteten Muskeln statt. Jeder Bewegungszustand setzt sich aus vier Unterzuständen (Preshape, Warten, Öffnen, Schließen) zusammen und die Transitions werden in den ersten beiden Fällen durch das Überschreiten einer Zeitdauer, in den letzten beiden Fällen durch Bewegungssignale dargestellt.

Der Ablauf für den Patienten gestaltet sich wie folgt: Nach der Reproduktion eines Schalt-signal bewegt sich die Prothese vom neutralen Zustand in einen Vorbewegungszustand (Preshape). In diesem nimmt

³In der Regel besteht kein Zusammenhang mit Muskelkontraktionsmustern eines gesunden Arms zum Ausführen derselben Bewegung.

die Handprothese eine für die folgende Bewegung optimale Startposition (Preshape) ein (Hakengriff, Lateralgriff: Beugen aller Finger außer Daumen; Zeigefinger: Beugen von Mittel-, Ring- und Zeigefinger; Pinzettengriff, Zylindergriff: Oppositionsstellung des Daumens). An der Stellung der Finger in diesem Preshape-Zustand erkennt der Anwender, dass ein Schaltsignal erkannt und ob es richtig interpretiert wurde.

Anschließend generiert der Anwender ein Bewegungssignal durch Kontraktionen des antagonistischen Muskelpaars (z. B. Flexoren und Extensoren des Unterarms, Biceps und Triceps des Oberarms), das den gewählten Griff schließt bzw. öffnet. Hierfür werden die der Griffart entsprechenden Freiheitsgrade der Hand gekoppelt, so dass effektiv nur ein Freiheitsgrad gesteuert werden muss⁴. Die zwei abgetasteten Muskeln dienen so zum Schließen bzw. Öffnen der Griffart, die Amplitude der Sensorsignale kann als Signal zur Spezifikation der Bewegung dienen.

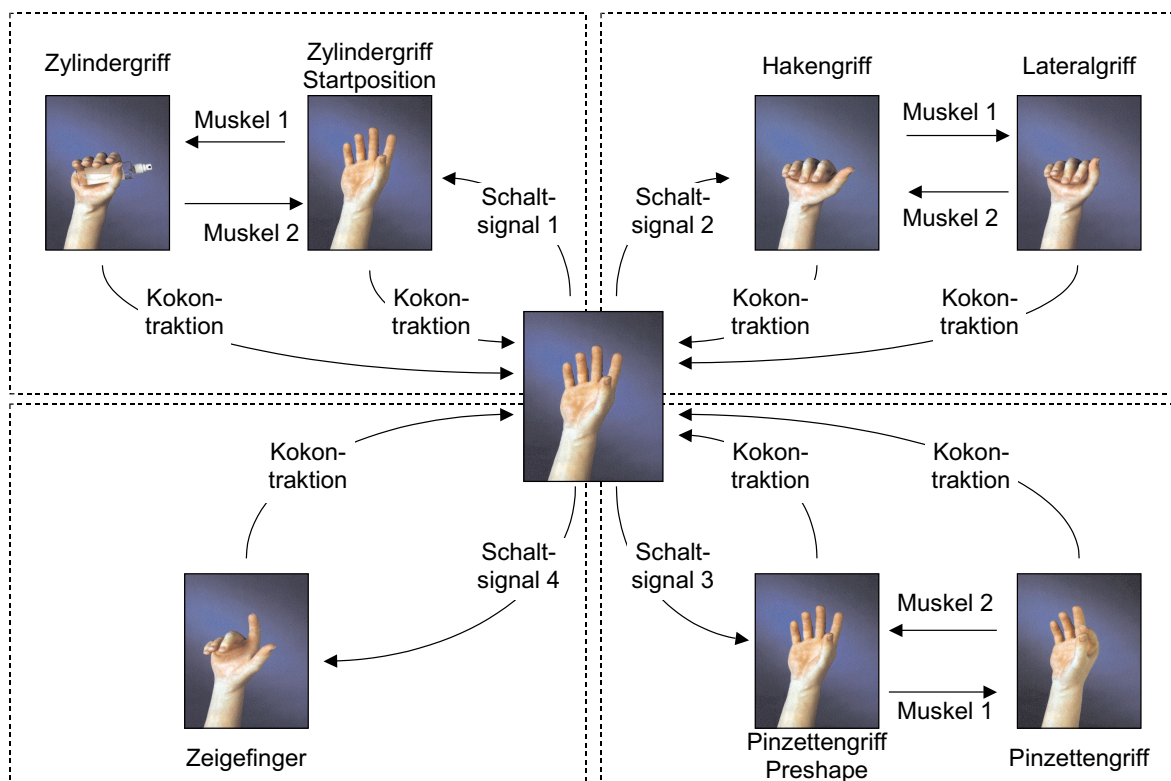


Bild 5.4: Beispielhafte Darstellung der implementierten Steuerung zur Umsetzung von fünf Griffarten, die fünfte Griffart ist der Preshape der zweiten Griffart, damit $K = 4$.

Die Bewegungsrichtung und -geschwindigkeit hängt von der Art des kontrahierten Muskels sowie der Stärke der Kontraktion ab. Im Bewegungszustand steuern Ereignisse am ersten Sensor $E_{1,1}$ (Start) das Schließen der Bewegung für die jeweilige Griffart. Das Ende der Kontraktion wird durch $E_{2,1}$ (Stopp) angezeigt, was die Rückkehr in den Ausgangszustand der Griffart bewirkt. Analog bewirken die Ereignisse des zweiten Sensors $E_{1,2}$, $E_{2,2}$ das Öffnen der Griffart. So können beispielsweise die durch den ersten Sensor erfassten Flexoren im Armstumpf zum Schließen und die Extensoren (zweiter Sensor)

⁴Das menschliche Greifverhalten läuft ähnlich ab [137]. Vor dem Greifen eines Objektes bewegt sich die Hand in die für die kommende Bewegung optimale Stellung.

zum Öffnen einer Bewegung verwendet werden. Die Ausführungsgeschwindigkeit der Bewegung wird dabei der Amplitude des Signals angepasst.

Eine Kokontraktion führt in den neutralen Zustand zurück, die Prothese öffnet sich wieder und wartet auf ein weiteres Schaltsignal. Um eine Griffart zu wechseln, ist somit eine Kokontraktion und ein Schaltsignal zu generieren. Die Implementierung des Automaten zur Umsetzung der eingangs erwähnten fünf Griffarten zeigt Bild 5.4. Ein beispielhaftes Signal mit den zugehörigen Zuständen des Zustandsautomaten stellt Bild 5.5 dar.

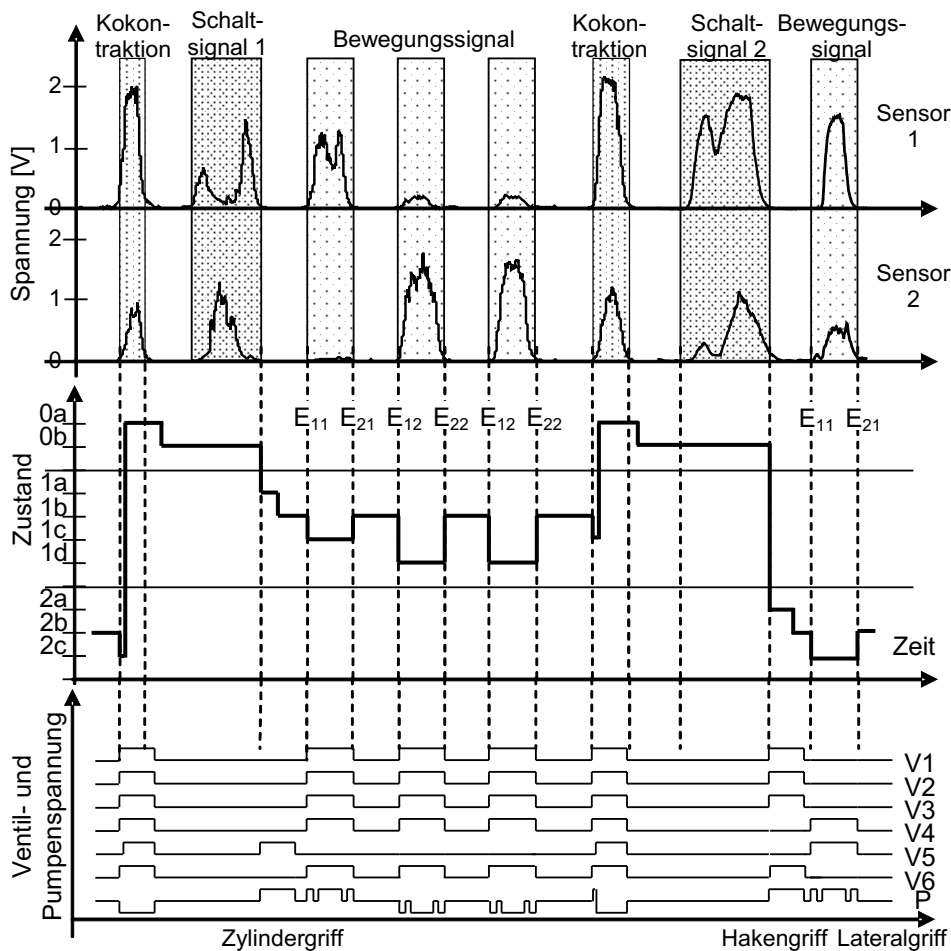


Bild 5.5: Myoelektrisches Signal zur Umsetzung von Bewegungsmustern unter Verwendung von zwei Sensoren

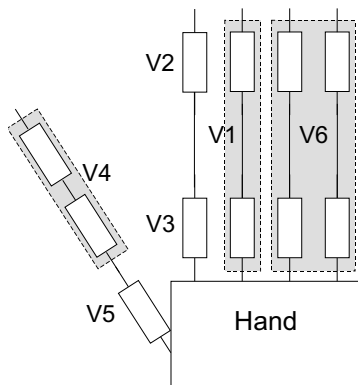
Durch diesen Ansatz kann jegliches Schaltsignal implementiert werden, sofern Schaltsignale vom Patienten gut reproduziert werden können und die extrahierten Merkmale zu seiner Beschreibung ausreichen.

Die Ausgangsgrößen des in diesem Abschnitt angepassten MMIs sind die zu den Schalt- und Bewegungssignalen zugehörigen Bewegungsmuster für die Prothese. In Abhängigkeit von der Art der Prothese muss ihre Steuerungseinheit (Bild 1.2) individuell angepasst werden, um auf die Ausgangsgrößen des MMIs korrekt zu reagieren.

5.2.4 Spezifikation der Steuerungseinheit

Eine neue Prothese mit natürlichem Erscheinungsbild und hoher Beweglichkeit beschreibt [185]. Die verwendete Antriebstechnik besteht aus miniaturisierten flexiblen Fluidaktoren [184]. Diese ermöglichen eine hohe Funktionalität bei reduziertem Gewicht und Bauraum sowie einem "weichen" Handgriff. Jeder Finger enthält zwei Fluidaktoren. Der Daumen verfügt über einen weiteren Fluidaktor zum Erreichen der Oppositionsstellung, so dass insgesamt elf Fluidaktoren in die Gelenke der Finger integriert sind. Jeder Fluidaktor übernimmt sowohl Beugung als auch Streckung des jeweiligen Gelenks. Um die Fluidaktoren mit möglichst wenigen Ventilen zu steuern, werden die Freiheitsgrade der Finger gekoppelt (Bild 5.6). Bei insgesamt elf Freiheitsgraden ergeben sich sechs unabhängige Freiheitsgrade. Die Druckerzeugungseinheit besteht aus einer Miniaturpumpe und sechs Miniaturventilen. Das Öffnen von Ventil V5 führt bei anliegendem Druck zur Oppositionsstellung des Daumens, alle anderen Ventile dienen zum Beugen des jeweiligen Gelenks (Bild 5.6). Sensoren zur Rückmeldung über den Zustand der Hand (vgl. Bild 1.2) sind nicht integriert, weshalb die Umsetzung von Bewegungen zeitgesteuert gesteuert geschehen muss.

Das Umsetzen einer erkannten Bewegungsabsicht aus einem Schaltsignal bedarf einer Zuordnung des Schaltsignals zu einer Griffart und eine Zuordnung der Griffart zu den Freiheitsgraden der Prothese. Wichtig ist es, den Anwender nicht mit der Fülle möglicher Bewegungsmöglichkeiten zu überlasten, sondern die für die Bewegung notwendigen Freiheitsgrade zu koppeln, so dass nur ein Freiheitsgrad vom Anwender bedient werden muss. Für eine Griffart sind sowohl die Ventile für den Preshape als auch für die eigentliche Bewegung zu belegen (Bild 5.6). Für die vorliegende sensorlose Prothese wird dies manuell durchgeführt, für kompliziertere Manipulatoren mit integrierten Sensoren eignet sich eine automatisierte Trajektoriengenerierung besser [45, 94].



Griff	Preshape	Bewegung
Zylindergriff	V5	alle außer V5
Lateralgriff	V1, V2, V3, V6	V4, V5
Pinzettengriff	V5	V3
Zeigefingerstellung	V1, V6	keine

Bild 5.6: Ventilzusammengehörigkeiten und -belegung der Prothese: V1, V4, V6 bedienen mehrere Fluidaktoren gleichzeitig, das Öffnen zusammengehöriger Ventile bei anliegendem Druck führt zur Ausführung von Griffarten, die Griffart Zeigefingerstellung wird nicht durch Bewegungssignale beeinflusst, der Hakengriff ist der Preshape des Lateralgriffs [151].

Zur Integration der Griffarten sind die Ventil-Codierungen aus Bild 5.6 im Zustandsautomaten zu hinterlegen und bei der Ausführung von Preshape-Bewegungen oder dem Öffnen/Schließen zu schalten. Besteht der Zustandsautomat aus mehr als vier Bewegungszuständen, lassen sich weitere Griffarten durch Hinzufügen entsprechender Ventil-Codierungen realisieren. Beim Übergang in den neutralen Zustand sind alle Ventile geöffnet, und die Pumpe läuft rückwärts. In Wartezuständen und im neutralen Zustand sind alle Ventile geschlossen und die Pumpe ist ausgeschaltet.

Das Ausführen des Preshapes muss ohne Einfluss des Patienten geschehen. Deswegen ist die Pumpe bei geöffneten Ventilen mit maximal zulässiger Spannung $u_{\text{Pumpe,max}}$ über einen festgelegten Zeitraum zu beaufschlagen (Bild 5.10).

Zum Ausführen der Bewegung selbst ist eine Proportionalsteuerung zweckmäßig, d. h. Systemgrößen wie z. B. Winkel, Winkelgeschwindigkeiten oder Greifkraft sollen sich proportional zur aktuellen EMG-Spannung des Bewegungssignals ändern. Die Proportionalsteuerung ist ohne Objektkontakt positionsbezogen umzusetzen. Bei Objektkontakt ist eine kraftbezogene Proportionalsteuerung anzustreben. Die vorliegende Prothese realisiert diese Umschaltung aufgrund ihrer elastischen Elemente automatisch, deshalb reicht die reine Umsetzung einer Antriebssteuerung aus (Pumpenspannung \sim EMG-Spannung). Eine ausführliche Diskussion findet sich in [15, 95].

Diese kann lage- oder geschwindigkeitsgesteuert geschehen [207]. Unter der Annahme, dass Pumpenspannung⁵ und Aktormoment proportional sind, setzt die Lagesteuerung Aktormomente proportional zu einem Steuersignal (vgl. Bild 1.1) um. Die Geschwindigkeitssteuerung führt Änderungen der Aktormomente proportional zum aktuellen Steuersignal durch.

Bei der Lagesteuerung reicht das Bewegungssignal von einem Muskel aus, um die Griffart zu öffnen und schließen. Jedoch ist zum Halten eines Griffs ein kontinuierlicher Kraftaufwand seitens des Patienten nötig oder eine Verriegelung ist zu integrieren. Zum komfortablen Halten eines Griffs hat sich deshalb die Geschwindigkeitssteuerung durchgesetzt [48], obwohl sie nicht der biologischen Bewegung entspricht. In der vorliegenden Prothese verhindern die kinetischen Eigenschaften der Aktoren die Umsetzung einer reinen Geschwindigkeitssteuerung - die Aktoren kommen in einen Sättigungsbereich, in dem keine weitere Winkelzunahme mehr möglich ist.

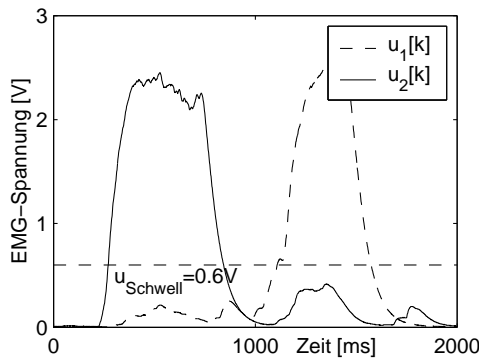


Bild 5.7: Bewegungssignale von zwei EMG-Sensoren

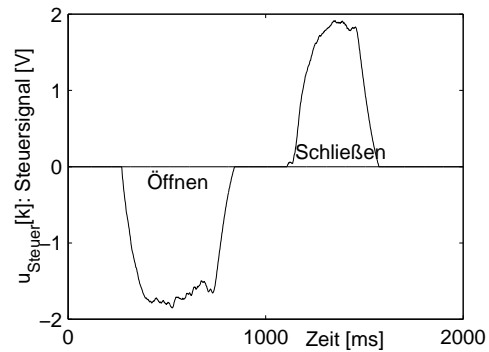


Bild 5.8: Steuersignal zum Öffnen und Schließen

Die Bestimmung des Steuersignals $u_{\text{Steuer}}[k]$ aus den EMG-Signalen zweier antagonistischer Muskeln zeigen die Bilder 5.7 und 5.8. Das Überschreiten eines Schwellwerts u_{Schwell} eines EMG-Sensors führt zur Berechnung des Steuersignals. Für zwei EMG-Sensoren mit den Spannungen $u_1[k]$ und $u_2[k]$ gilt

$$u_{\text{Steuer}}[k] = \begin{cases} (u_1[k] - u_{\text{Schwell}}) & \text{wenn } u_1[k] > u_{\text{Schwell}} \text{ und } u_1[k] > u_2[k] \\ (-u_2[k] + u_{\text{Schwell}}) & \text{wenn } u_2[k] > u_{\text{Schwell}} \text{ und } u_2[k] > u_1[k]. \end{cases} \quad (5.1)$$

Zur Berechnung der zugehörigen Pumpenspannung u_{Pumpe} muss eine Proportionalitätskonstante χ eingefügt werden, die die maximal mögliche Spannung eines EMG-Sensors $u_{\text{EMG,max}}$ berücksichtigt:

$$u_{\text{Pumpe}} = \chi \cdot u_{\text{Steuer}} \quad \text{mit} \quad \chi = \frac{u_{\text{Pumpe,max}}}{u_{\text{EMG,max}} - u_{\text{Schwell}}} \quad (5.2)$$

Die Pumpenspannung ist dann pulsweitenmoduliert umzusetzen. Den Verlauf von Ventil- und Pumpenspannung bei einer Sequenz von Muskelkontraktionen zeigt Bild 5.5.

⁵Die Pumpenspannung liegt in der Regel nicht analog vor, sondern wird pulsweitenmoduliert.

Zum Schutz der Aktoren vor einem Überdruck und der Pumpe vor Überlastung aufgrund des Fehlens einer sensorischen Ausstattung der verwendeten Prothese verzichtet eine erste Implementierung auf die Proportionalsteuerung und setzt die Bewegungen zeitgesteuert um. Nach Bild 5.10 wird die Pumpe über einen griffartabhängigen Zeitraum T_D mit maximaler Spannung beaufschlagt. Den modifizierten Zustandsautomat stellt Bild 5.9 dar.

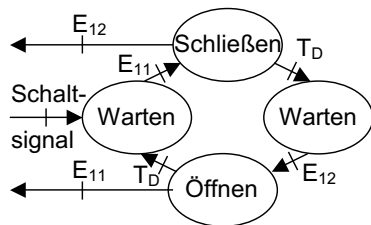


Bild 5.9: Zeitgesteuerte Implementierung einer Griffart

	Preshape	Bewegung
Zylindergriff	300	1500
Zeigefingerstellung	1200	-
Lateralgriff	1500	500
Pinzettengriff	300	600

Bild 5.10: Dauer maximaler Pumpenspannung in ms zur Ausführung der Griffarten bei Verzicht auf eine Proportionalsteuerung

5.2.5 Versuchsdesign und Patientenkollektiv

Um Aussagen über die Anwendbarkeit der Prothesensteuerung insbesondere in Bezug auf die vorgestellten modifizierten Klassifikationsverfahren zu erhalten, sind vorab folgende Fragen zu beantworten:

- Sind amputierte Menschen in der Lage, Schaltsignale zuverlässig zu generieren?
- Gibt es eine Beziehung zwischen Schaltsignaltyp und dessen Reproduzierbarkeit?
- Wie viele Realisierungen sind mindestens notwendig, um robuste Klassifikatoren zu entwerfen?
- Wie wirkt sich Training auf die Qualität der Schaltsignale aus?
- Wie robust ist das MMI bei Verwendung in unterschiedlichen Sitzungen?

Zur Beantwortung dieser Fragen sind Patientensitzungen zu konzipieren, die sich in Anlernsitzungen und Validierungssitzungen einteilen lassen. Nach Bild 5.11 beginnt eine Anlernsitzung mit dem Festlegen von zu wiederholenden Schaltsignaltypen. Die Anzahl ist so hoch zu wählen, dass einige verworfen werden können, wenn der Patient sie nicht wiederholen kann oder sich dabei unwohl fühlt. Alle Patienten generieren mindestens fünf Schaltsignaltypen. Anlernsitzungen finden mit 17 Patienten (2 weibliche, 15 männliche, 12 bis 64 Jahre alt) und zwei Probanden statt. Eine Übersicht gibt Tabelle 5.1. Tabelle 5.2 zeigt die Einteilung der Schaltsignaltypen in Schaltsignalkategorien, die von den Patienten generiert werden. Je nach Fragestellung besteht die Möglichkeit, die festgelegten Schaltsignaltypen zu trainieren, bevor eine Aufnahme stattfindet. Eine Bewertung der gewonnenen Daten findet anhand der Klassifikationsergebnisse über Lerndaten oder Crossvalidierung eines auf der Basis der Daten entworfenen Klassifikators statt. Je nach Eignung eines Schaltsignaltyps kann dieser aus dem Datensatz entfernt und der Klassifikator neu angelern werden. Den reproduzierbaren Schaltsignaltypen können schließlich für den Prothesenbetrieb Griffarten zugewiesen werden und der Quelltext kann exportiert werden. Um einen Echtzeitbetrieb zu simulieren, kann in einer weiteren, sogenannten Validierungssitzung der entworfenen Klassifikator verwendet werden, um aufgenommene Signale zu validieren.

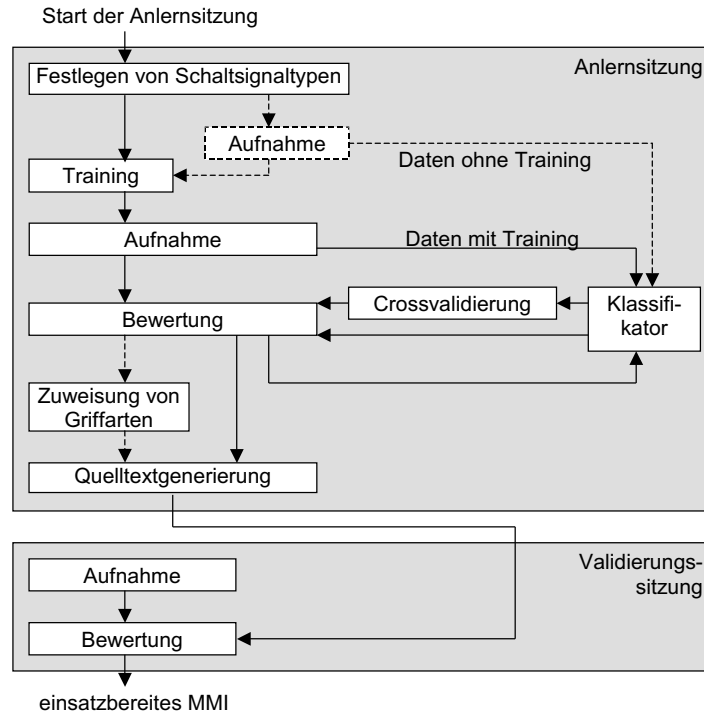


Bild 5.11: Vorgehensweise bei Anlernsitzungen und Validierungssitzungen

Die Aufnahme der Schaltsignale erfolgt unter Verwendung von zwei myoelektrischen Sensoren (Otto Bock MYOBOCK, $f_a=100\text{Hz}$), die auf dem Armstumpf des Patienten platziert werden. Bei den Patienten E und J werden musculus biceps brachii und musculus triceps brachii, bei allen anderen musculus extensor digitorum and musculus flexor carpi radialis gemessen.

17 der 19 Probanden nehmen nur an einer Anlernsitzung teil, Patient F und Proband S nehmen an mehreren Validierungssitzungen teil. Fünf Patienten generieren zusätzlich Schaltsignale ohne Training, alle anderen erhalten Training vor der Aufnahme.

Alle Klassifikatoren werden unter Zuhilfenahme der MATLAB-Toolbox DAVE-Design angelemt. Das Berechnen von Klassifikationsgüten erfolgt durch eine 10-fache Crossvalidierung. Die Verwendung der Klassifikatoren beruht auf folgenden Parametrierungen (gemäß maximalem MMI-Index aus den Kapiteln 2 und 3) und Abkürzungen:

- MAN/DI: Merkmalsauswahl durch MANOVA mit $s_m = 8$, Merkmalsaggregation durch lineare Diskriminanzanalyse mit $s_d = 4$,
- MM/DI: Merkmalsauswahl durch MM mit $s_m = 6$, Merkmalsaggregation durch lineare Diskriminanzanalyse mit $s_d = 3$,
- MAN/MD: Merkmalsauswahl durch MANOVA mit $s_m = 6$, Merkmalsaggregation durch modifizierte Diskriminanzanalyse (MD) mit $s_d = 3$,
- MM/MD: Merkmalsauswahl durch MM mit $s_m = 6$, Merkmalsaggregation durch modifizierte Diskriminanzanalyse (MD) mit $s_d = 3$,

Patient	Alter	Verlust	Geschlecht	Amputation	Prothesenversorgung
A	41	im Alter von 19	m	Unterarm, links	7 Jahre
B	12	im Säuglingsalter	m	Unterarm, rechts	einige Jahre
C	34	kongenital	m	Unterarm, rechts	viele Jahre
D	31	kongenital	w	Unterarm, links	13 Jahre
E	62	im Alter von 22	m	Oberarm, links	18 Jahre
F	44	im Alter von 13	m	Unterarm, rechts	26 Jahre
G	31	im Alter von 16	m	Unterarm, links	viele Jahre
H	36	im Alter von 34	m	Unterarm, links	2 Jahre
I	64	im Alter von 60	m	Unterarm, rechts	4 Jahre
J	34	im Säuglingsalter	m	Oberarm, links	keine
K	16	kongenital	m	Unterarm, links	keine
L	31	im Alter von 16	m	Unterarm, rechts	15 Jahre
M	61	im Alter von 38	w	Unterarm, rechts	22 Jahre
N	33	im Alter von 21	m	Unterarm, rechts	10 Jahre
O	40	im Alter von 19	m	Unterarm, rechts	21 Jahre
P	59	im Alter von 56	m	Unterarm, rechts	3 Jahre
Q	20	im Alter von 2	m	Hand, rechts	4 Jahre
R	27	gesund	m	-	-
S	28	gesund	m	-	-

Tabelle 5.1: Patientenbeschreibung [165]

- HKK: Merkmalsvorauswahl durch MM mit $s_m = 6$, Merkmalsaggregation innerhalb der Hierarchie auf $s_d = 2$, $Q_{\max} = 10^{-3}$,
- HKT: Merkmalsauswahl innerhalb der Hierarchie durch MM mit $s_m^{\min} = 6$, $m_{\max} = 3$, $q_c = 0.99$ und $s_d = 2$ und
- KO: Merkmalsauswahl durch MM mit $s_m = 6$, Merkmalsauswahl im Zweiklassenproblem durch MANOVA mit $s_m^* = 3$.

5.2.6 Schaltsignaldauer und Reproduzierbarkeit

Dieser Abschnitt wertet die Fähigkeit der Patienten aus, diverse Schaltsignaltypen sicher zu reproduzieren, und diskutiert die Dauer zum Generieren eines Schaltsignals.

Für einen Wechsel der Griffart muss der Patient vorab eine Kokontraktion generieren. Hierbei entsteht eine durchschnittliche Zeitverzögerung von 0.55s (0.28s - 0.91s). Während sich die Prothese öffnet, kann bereits das nachfolgende Schaltsignal gegeben werden. Die Dauer zum Generieren eines Schaltsignals variiert nach Tabelle 5.2 von 0.18s (Patient F) bis 3.63s (Proband S). Die durchschnittliche Dauer beträgt 1.01s. Bezogen auf die von der Prothese benötigte Dauer zum Erreichen des Preshapes (Bild 5.10) lassen sich bequem schnellen Schaltsignalen wichtige Griffarten zuordnen, so dass für diese nur eine geringe Zeitverzögerung entsteht.

68% (13 von 19) aller Patienten und Probanden waren in der Lage, alle der gewählten Schaltsignalkategorien zu reproduzieren. Insgesamt konnten 91% (112 von 123) der verwendeten Schaltsignaltypen zur multifunktionalen Steuerung eingesetzt werden (Tabelle 5.2). Alle Patienten konnten mindestens vier Schaltsignaltypen wiederholen, eine Umsetzung der in Abschnitt 5.2.1 beschriebenen fünf Griffarten ist somit möglich. Es folgt eine kurze Darstellung der patientenindividuellen Probleme.

Patient	ZV	AV	I	A	K	$K(K_{\text{org}})$	T_u	T_o	\bar{T}
A	2	1	2	1	-	6	0.78	2.63	1.23
B	3	1	0(1)	-	-	4(5)	0.62	2.29	1.12
C	4	2	-	-	2	8	0.42	3.62	1.89
D	4	2	1	1	-	8	0.41	3.05	1.15
E	4	2	-	1	-	7	0.19	1.73	0.84
F	5(7)	-	-	-	-	5(7)	0.18	2.04	0.63
G	5	-	-	-	-	5	0.80	2.86	1.53
H	2	-	2	-	1	5	0.31	1.74	0.79
I	3	-	2	-	-	5	0.48	2.61	1.35
J	4	1	1	-	-	6	0.44	3.46	1.88
K	0(2)	2	2	0(3)	-	4(9)	0.52	2.69	1.30
L	2	2	2	1	-	7	0.37	2.24	1.07
M	2	1	1	1	-	5	0.20	2.14	0.90
N	2	1	1(2)	1	-	5(6)	0.21	1.35	0.62
O	2	1	2	1	-	6	0.44	3.12	1.03
P	2	0(1)	2	-	-	4(5)	0.20	2.84	0.60
Q	2	1	1(2)	-	-	4(5)	0.23	1.26	0.58
R	5	2	1	1	-	9	0.19	2.27	1.19
S	4	2	2	1	-	9	0.18	3.63	0.97
Σ	57(61)	21(22)	22(25)	9(12)	3(3)	112(123)	\emptyset :0.38	\emptyset :2.50	\emptyset :1.09

Tabelle 5.2: Verwendete Schaltsignaltypen (ZV: zeitvariiert, AV: amplitudenvariiert, I: intuitiv, A: abstrakt, K: kombiniert, in Klammern: festgelegte Schaltsignaltypen), K : Anzahl der verwendeten Schaltsignalklassen. T_u : minimale Dauer, T_o : maximale Dauer, \bar{T} : durchschnittliche Dauer in Sekunden zum Ausführen eines Schaltsignals, K_{org} : Anzahl aufgenommener Schaltsignalklassen

Patient B hatte aufgrund seines sehr kurzen Armstumpfs (ca. 5cm) Probleme beim Ausführen einer Pronation (Schaltsignalkategorie I). Die resultierende EMG-Spannung war zu gering, um vom System als Muskelkontraktion erkannt zu werden.

Patient F fühlte sich unwohl beim Kontrahieren der Flexor-Muskeln. Daraufhin wurden zwei Schaltsignaltypen verworfen.

Patient K hatte Probleme, kurze und lange Kontraktionen unterscheidbar zu generieren. Weiterhin konnte er ein abstraktes Schaltsignal nicht sicher reproduzieren. Der Grund war die Unerfahrenheit des Patienten im Umgang mit myoelektrischen Signalen. Eine prothetische Versorgung sowie die dafür notwendige Konstitution war bislang nicht gegeben. Trotzdem konnten vier Schaltsignaltypen verwendet werden. Zur Steigerung der Ergebnisse sind weitere Sitzungen notwendig, längere Trainingsprogramme führen hier zu verbesserten Ergebnissen.

Patient P zeigte Fehler beim Unterscheiden der Schaltsignale "schwach-stark" und "kurz-lang". Es zeigt sich, dass Patienten beim Versuch, eine kurze Kontraktion zu geben, gelegentlich aus Versehen ein schwaches Signal generieren. Ebenso verleiten stärkere Kontraktionen zu einer Verlängerung der Kontraktionsdauer.

Die Patienten N und Q waren nicht in der Lage, unterscheidbare Kontraktionsmuster für Pronationen und Supinationen zu generieren. In beiden Fällen wurde je einer der beiden Schaltsignaltypen verwendet.

Die Probleme beim Generieren von Schaltsignalen basieren entweder auf anatomischen Gegebenheiten oder auf der vollautomatischen Routine zur Merkmalsextraktion. Die Probleme können vermieden werden, indem Schwellwerte und Filter angepasst werden. Hierdurch wird beispielsweise die Identifikation von ähnlichen Schaltsignalen begünstigt, die Robustheit sinkt allerdings. Deshalb verwendet diese Arbeit nur Schaltsignaltypen, die der Patient ohne Probleme generieren kann, alle anderen werden verworfen.

5.2.7 Auswirkungen von Training

Dieser Abschnitt beurteilt die Fähigkeit von Patienten, Schaltsignale ohne und mit Training zu reproduzieren und vergleicht die Ergebnisse anhand der Klassifikationsfehler des MAN/DI-Verfahrens⁶.

Patient	ohne Training		mit Training	
	Klassifikationsfehler	(K/n_j)	Klassifikationsfehler	(K/n_j)
B	15.8 ± 3.1	(5/15)	7.4 ± 2.0	(5/10)
D	18.8 ± 2.1	(8/15)	6.1 ± 1.7	(8/15)
E	8.4 ± 2.1	(7/15)	11.7 ± 1.6	(7/10)
J	25.0 ± 3.9	(6/15)	1.5 ± 1.5	(6/10)
K	12.2 ± 2.5	(4/15)	7.6 ± 1.5	(4/15)

Tabelle 5.3: Vergleich der Klassifikationsfehler des MAN/DI-Verfahrens für fünf Patienten ohne/mit Training

Hierzu werden Schaltsignale ohne Training aufgenommen, der Klassifikator wird angelernt und für die Ergebnisse eine Crossvalidierung berechnet. Anschließend erhält der Patient die Gelegenheit, die Schaltsignale mit DAVE für 5-10 Minuten zu trainieren, indem die Zeit/Amplitudenverläufe visualisiert werden. Es folgt die Wiederholung des Experiments.

Während des Trainings beurteilt der Versuchsleiter die Ähnlichkeit von Schaltsignalen gleichen Schaltsignaltyps anhand der Signaldarstellung in DAVE. Das Training wird beendet, wenn der Patient in der Lage ist, Schaltsignale so zu reproduzieren, dass sie sich in ihrem Zeitverlauf ähneln.

Tabelle 5.3 zeigt den Klassifikationsfehler des MAN/DI-Verfahrens bei Betrachtung von 50 Wiederholungen einer 10-fachen Crossvalidierung. Die Patienten B, D, K und J konnten eine starke Verbesserung durch das Training erreichen, da sie durch die Visualisierung des Trainings ein Gefühl für Dauer und Stärke einer Kontraktion bekamen. Insbesondere Patient J zeigte eine erhebliche Verbesserung, da er einerseits oberarmamputiert und andererseits nie mit einer Prothese versorgt war. Die Verschlechterung der Ergebnisse von Patient E ist auf mangelnde Motivation beim Ausführen der Schaltsignale zurückzuführen. Alle folgenden Experimente werden mit Patienten durchgeführt, die 5-10 Minuten Training erhalten haben.

5.2.8 Umfang des Lerndatensatzes

Dieser Abschnitt evaluiert den anhand des Benchmarkdatensatzes festgelegten Umfang für Patientendatensätze (vgl. Abschnitt 2.10.3). Exemplarisch wird der Datensatz von Patient D ($n_j = 15$) verwendet.

⁶Alle anderen Klassifikatoren liefern qualitativ gleichwertige Ergebnisse [202].

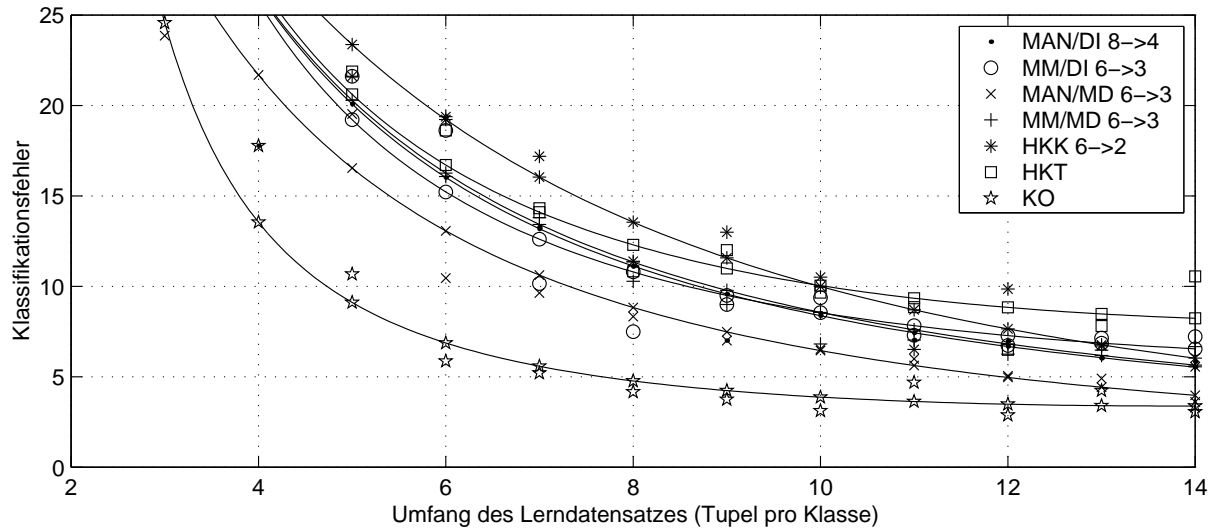


Bild 5.12: Klassifikationsfehler mit Interpolationen (Exponentialfunktionen) in Abhängigkeit vom Umfang des Lerndatensatzes für verschiedene Klassifikatoren

Lerndatensätze bestehend aus $n_j = 2, n_j = 3, \dots, n_j = 14$ Datentupeln pro Klasse werden zufällig erstellt und über den nicht verwendeten Datentupeln getestet. Der Vorgang wird solange wiederholt bis jedes Datentupel 5 Mal als Testdatentupel verwendet wurde.

Die Ergebnisse sind in Bild 5.12 dargestellt. Demnach werden die Ergebnisse aus Abschnitt 2.10.3 bestätigt: Eine Mindestanzahl von $n_j = 10$ Lerndatensätzen pro Klasse ist notwendig, um einen Klassifikationsfehler kleiner 10% zu erhalten. Durch die Verwendung einer größeren Anzahl an Lerndaten sinkt der Klassifikationsfehler nur noch wenig. Die mit Abstand besten Ergebnisse liefert der KO-, die schlechtesten der HKT-Klassifikator. Der HKK-Klassifikator zeigt eine deutliche Tendenz, bei der Verwendung eines umfangreicheren Datensatzes bessere Ergebnisse zu liefern. Die modifizierten Einschnittverfahren liefern gute Ergebnisse, besonders das MAN/MD-Verfahren sticht hervor. Obwohl das MAN/DI-Verfahren mit einer großen Anzahl an Parametern eher in der Lage ist, sich Problemen mit umfangreichen Lerndatensätzen anzupassen, liefern die modifizierten Verfahren trotz der geringen Anzahl an Parametern vergleichbare bzw. bessere Ergebnisse.

5.2.9 Güte der Klassifikatoren

Dieser Abschnitt evaluiert anhand von Tabelle 5.4 die Fähigkeit der beschriebenen Klassifikatoren, sich an die aufgenommenen Patientendatensätze anzupassen. Es kommen ausschließlich Datensätze mit Training zum Einsatz. Nicht reproduzierbare Schaltsignaltypen (Abschnitt 5.2.6) werden aus den Datensätzen entfernt, die Datensätze werden nicht von falsch ausgeführten Schaltsignalen gesäubert, um beim Testen die Realität im Prothesenbetrieb abzubilden. Verwendet werden die Parametrierungen der Klassifikationsverfahren mit dem maximalen MMI-Index aus den Kapiteln 2 und 3. Parametrierungen zum Erreichen der maximalen Klassifikationsgüte und deren Ergebnisse zeigt Anhang 7.2.1.

Die Berechnung des durchschnittlichen MMI-Index für alle Datensätze bezieht sich zur besseren Vergleichbarkeit mit den Kapiteln 2 und 3 auf $K = 8$ Klassen und nicht auf eine Mittelung der Klassenanzahl über alle Datensätze.

Pat.	ZV	AV	I	A	K	K	n	MAN/DI	MAN/MD	MM/DI	MM/MD	HKK	HKT	KO	
A	2	1	2	1	-	6	15	5.5 ± 1.3	3.9 ± 1.5	6.4 ± 1.9	4.3 ± 1.6	2.6 ± 0.9	5.7 ± 2.5	2.3 ± 0.8	
B	3	1	-	-	-	4	15	7.4 ± 2.0	8.2 ± 2.0	11.0 ± 3.3	11.1 ± 3.9	12.8 ± 3.8	11.4 ± 3.7	9.6 ± 2.5	
C	4	2	-	-	2	8	10	11.2 ± 2.8	11.9 ± 2.2	16.0 ± 2.9	13.4 ± 3.1	15.7 ± 3.1	16.1 ± 3.6	7.5 ± 2.5	
D	4	2	1	1	-	8	15	6.1 ± 1.7	4.8 ± 2.0	6.7 ± 2.0	5.6 ± 1.8	4.8 ± 2.0	7.4 ± 2.4	2.9 ± 1.5	
E	4	2	-	1	-	7	15	11.7 ± 1.6	12.5 ± 1.8	10.7 ± 2.2	10.5 ± 2.2	10.5 ± 2.2	11.4 ± 2.4	7.5 ± 2.1	
F	5	-	-	-	-	5	15	5.8 ± 0.5	6.1 ± 0.9	8.8 ± 1.5	8.3 ± 1.8	9.3 ± 1.7	9.2 ± 1.9	7.5 ± 1.6	
G	5	-	-	-	-	5	10	9.2 ± 1.9	8.0 ± 1.7	6.1 ± 2.4	5.9 ± 2.2	5.1 ± 2.7	6.8 ± 2.7	4.2 ± 2.2	
H	2	-	2	-	1	5	10	8.6 ± 1.6	8.2 ± 2.0	6.5 ± 3.1	7.4 ± 2.7	6.4 ± 2.5	7.2 ± 3.2	6.3 ± 2.7	
I	3	-	2	-	-	5	10	0.0 ± 0.3	0.2 ± 0.6	4.2 ± 2.9	5.0 ± 3.0	4.9 ± 3.0	4.5 ± 2.4	2.6 ± 2.0	
J	4	1	1	-	-	6	15	1.5 ± 1.5	1.8 ± 1.5	3.6 ± 1.5	3.0 ± 1.9	2.6 ± 1.6	5.3 ± 2.3	2.2 ± 1.7	
K	-	2	2	0	-	4	15	7.6 ± 1.5	6.7 ± 1.8	8.0 ± 2.3	8.7 ± 2.6	8.2 ± 2.6	9.1 ± 2.9	6.6 ± 1.8	
L	2	2	2	1	-	7	15	5.4 ± 0.9	4.4 ± 1.0	4.1 ± 1.2	3.4 ± 1.1	3.7 ± 1.1	3.6 ± 1.2	4.1 ± 0.7	
M	2	1	1	1	-	5	15	7.5 ± 1.7	7.9 ± 2.0	6.6 ± 1.9	7.4 ± 2.0	7.3 ± 2.0	8.3 ± 2.3	5.9 ± 1.8	
N	2	1	1	1	-	5	15	7.8 ± 2.1	8.7 ± 1.8	5.9 ± 2.4	5.7 ± 2.4	6.9 ± 2.7	7.0 ± 2.7	4.8 ± 1.7	
O	2	1	2	1	-	6	15	6.9 ± 2.3	7.1 ± 1.9	10.6 ± 2.6	10.2 ± 2.1	11.6 ± 3.0	12.5 ± 2.6	8.9 ± 1.9	
P	2	-	2	-	-	4	15	4.5 ± 1.6	4.6 ± 1.6	4.5 ± 1.5	4.5 ± 1.7	4.4 ± 1.6	4.1 ± 1.4	3.9 ± 1.4	
Q	2	1	1	-	-	4	15	8.7 ± 1.9	6.5 ± 2.2	8.5 ± 2.7	8.2 ± 2.7	7.5 ± 2.5	7.8 ± 2.4	7.3 ± 2.8	
R	5	2	1	1	-	9	15	5.9 ± 1.0	4.3 ± 1.5	6.2 ± 2.1	4.1 ± 2.1	4.8 ± 2.1	7.2 ± 2.3	2.2 ± 1.5	
S	4	2	2	1	-	9	15	1.8 ± 0.7	1.8 ± 0.5	5.4 ± 2.4	1.5 ± 1.1	1.9 ± 1.1	3.1 ± 1.5	0.5 ± 0.6	
	∅								6.5	6.2	7.4	6.7	6.9	7.8	5.1
mittlerer MMI-Index (10000 · Λ)								221	419	291	361	271	188	426	

Tabelle 5.4: Klassifikationsfehler der Klassifikatoren bei Verwendung von Patientendaten mit Training (50 Wiederholungen einer 10-fachen Crossvalidierung), ZV: zeitvariant, AV: amplitudenvariant, I: intuitiv, A: abstrakt, K: kombiniert

Alle in dieser Arbeit modifizierten Verfahren mit Ausnahme der hierarchischen Klassifikation nach Datentupeln (HKT) weisen einen höheren MMI-Index auf als das beste in Kapitel 2 erarbeitete Verfahren. Die Ursache für diese Ausnahme liegt in den Verletzungen von Normalverteilungsannahmen begründet, sowie in der geringen Datentupelanzahl pro Datensatz, durch die das Verfahren seine Stärken nicht ausspielen kann. Mit einer Steigerung des MMI-Index um 93% ist die KO-Klassifikation das beste Verfahren.

Anhang 7.2.1 zeigt weitere Parametrierungsbeispiele, die belegen, dass die modifizierten Verfahren die herkömmlichen bei gleicher Parametrierung auch in der Klassifikationsgüte übertreffen.

Bezogen auf den mittleren Klassifikationsfehler zeigt sich, dass insbesondere die Einschrittverfahren mit einer modifizierten Merkmalsauswahl (MM) nach (3.6) schlechtere Ergebnisse liefern. Hingegen liefert die modifizierte Diskriminanzanalyse (MD) gute Ergebnisse. Problematisch ist für alle Verfahren das Vorhandensein von falschen Klassenbezeichnungen aufgrund verwechelter Schaltsignale seitens des Patienten (Patienten C, F und O). Weitere Probleme ergeben sich für das MM-Verfahren durch numerische Instabilitäten aufgrund nicht vorhandener Streuung für bestimmte Merkmale und Klassen (zu kleiner Datensatz) - das Verfahren neigt dazu, diese Merkmale zu verwenden (Patient I). Ein drittes Problem tritt auf, wenn die Trennung zweier Klassen kaum möglich ist (Patient B). Da allen Mehrschrittverfahren das MM-Verfahren zugrunde liegt, sind die Ergebnisse für die genannten Datensätze verfälscht. Anhang 7.2.1 zeigt Ergebnisse auf der Basis des MANOVA-Verfahrens.

Bei allen legitimen Lerndatensätzen überflügeln die modifizierten Verfahren das herkömmliche Verfahren auch bezüglich der Klassifikationsgüte. Insbesondere die hierarchische Klassifikation nach Klassen (HKK) und die KO-Klassifikation ergeben in Kombination mit den modifizierten Einschrittverfahren sehr gute Ergebnisse. Diese werden umso besser, je mehr Klassen im Lerndatensatz enthalten sind (Patient D, Probanden R und S).

Als weiteres Ergebnis lässt sich festhalten, dass die über dem Benchmarkdatensatz parametrierten Verfahren auch auf den realen Patientendaten gute Ergebnisse liefern. Der Benchmarkdatensatz liefert also eine gute Approximation zur Parametrierung der modifizierten Verfahren. Zur Implementierung sind nur minimale Nachbesserungen erforderlich, die der Präferenz des Entwicklers entspringen (z. B. zum Erreichen höherer Klassifikationsgüten, vgl. Anhang 7.2.1).

5.2.10 Robustheit

Eine Crossvalidierung führt zu einer optimistischen Schätzung der Klassifikationsgüte, obwohl nur 90% der Daten zum Lernen verwendet werden (auf der Basis der 10-fachen Crossvalidierung). Einflüsse auf den Echtzeitbetrieb bei der nachfolgenden Routinenutzung wie Tagesform, nicht exakt reproduzierbare Sensorpositionen, Vergessen von Griffmustern, etc. gehen nicht in die Ergebnisse ein.

Deswegen modelliert dieser Abschnitt die Güte der Klassifikatoren bei den genannten Störungen im täglichen Leben auf der Basis von mehreren voneinander unabhängigen Testdatensätzen, die in Validierungssitzungen gewonnen werden. Von Patient F und Proband S werden hierfür Daten aus drei bzw. sechs Sitzungen gesammelt, die jeweils im Abstand von 3-7 Tagen stattfinden. Das Reproduzieren von Sensorpositionen in jeder Sitzung geschieht absichtlich nur durch eine grobe Schätzung und nicht durch Messung. Weiterhin gibt es keine Möglichkeit zum Trainieren der Schaltsignale vor oder zwischen den Sitzungen. Zur Erinnerung an die Aufgabe werden den Patienten lediglich kurz handschriftliche Visualisierungen ihrer EMG-Signale präsentiert. Es folgt die automatische Aufzeichnung der EMG-Signale von 50 bis 100 Schaltsignalen, die DAVE in zufälliger Reihenfolge vorgibt.

Zu Validierungszwecken findet vorab ein Anlernen ohne Verwendung der Methoden des robusten Entwurfs aus Abschnitt 3.4 statt. Die Ergebnisse des Anlernens über den Daten der ersten Sitzung und dem Testen über den nachfolgenden Sitzungen zeigen die Tabellen 5.5 und 5.6. Die Verfahren werden so parametrisiert, dass die maximale Klassifikationsgüte über den Lerndaten erreicht wird. Dabei ergeben sich wiederum die Parametrisierungen aus Abschnitt 5.2.5, für den KO-Klassifikator wird sogar die Merkmalsdimension auf $s_m = 4$ vermindert.

Klassifikator	MAN/DI	MAN/MD	MM/DI	MM/MD	HKK	KO	Ø
Sitzung 2	18.7	18.7	18.7	17.3	16.0	52.0	23.6
Sitzung 3	25.3	49.3	38.7	25.3	28.0	40.0	34.4
Ø	22.0	34.0	28.7	21.3	22.0	46.0	29.0

Tabelle 5.5: Klassifikationsfehler bei Patient F ohne robusten Entwurf, Anlernen über Daten von Sitzung 1, Testen über Daten von Sitzung 2 und 3

Klassifikator	MAN/DI	MAN/MD	MM/DI	MM/MD	HKK	KO	Ø
Sitzung 2	43.3	53.3	56.7	58.9	51.1	41.4	50.8
Sitzung 3	43.7	38.5	51.1	52.6	44.1	30.4	43.4
Sitzung 4	61.1	66.7	56.7	65.6	65.6	38.9	59.1
Sitzung 5	48.9	46.7	66.7	68.9	51.1	45.6	54.6
Sitzung 6	45.6	50.0	51.1	58.9	50.0	51.1	51.1
Ø	48.5	51.0	56.5	61.0	52.4	41.5	51.8

Tabelle 5.6: Klassifikationsfehler bei Proband S ohne robusten Entwurf, Anlernen über Daten von Sitzung 1, Testen über Daten von Sitzung 2, 3, 4, 5 und 6

Die Tabellen zeigen durchweg hohe Klassifikationsfehler der Klassifikatoren ohne robusten Entwurf. Die besten Ergebnisse ergeben sich zwar durch die modifizierten Verfahren, an einen Einsatz in Prothesen ist jedoch nicht zu denken. Auffällig sind die schlechten Ergebnisse des KO-Klassifikators bei Patient F. Der Grund ist die Beschränkung des KO-Klassifikators auf $s_m^* = 3$ Merkmale. Die drei wichtigsten Merkmale der Anlernsitzung geben nur geringe Information über das Klassifikationsproblem für Sitzung 2. Die übrigen Verfahren schöpfen ihre Güte aus den Merkmalen 4 bis 8. In Sitzung 3 sind die Merkmale 4 bis 8 trennschwächer, die drei besten Merkmale beschreiben die Daten aus Sitzung 3 jedoch besser als die aus Sitzung 2. Deshalb liefert der KO-Klassifikator hier bessere Ergebnisse, während die Ergebnisse über den anderen Verfahren abfallen. Bei Proband S gilt diese Aussage nicht, da die in den Lerndaten gewählten Merkmale für alle Verfahren schlechte Ergebnisse liefern, wobei der KO-Klassifikator hier wieder die besten Ergebnisse liefert. Der höhere Klassifikationsfehler im Vergleich zu Patient F resultiert hierbei aus einer größeren Klassenanzahl ($K = 9$). Eine ausführliche Diskussion über Merkmalsselektion bei zeitvarianten Signalen findet sich in [153].

Die Bilder 5.13 und 5.14 zeigen eine zweidimensionale Projektion der angelernten Diskriminanzfunktionen und den Lerndatensatz im Vergleich zum unbekanntem Testdatensatz (kleine Zahlen). Es ist zu beachten, dass diese Bilder nicht das originale Klassifikationsproblem darstellen, sondern aufgrund der schlechten Darstellbarkeit im höherdimensionalen Raum nur eine vereinfachte Darstellung als zweidimensionale Projektion zeigen. Zu sehen ist in beiden Bildern eine systematische Verschiebung der Klassen des Testdatensatzes im Vergleich zum Lerndatensatz. Die Diskriminanzfunktionen spiegeln somit

nicht die Testdaten wider, und es ergeben sich ein großer Klassifikationsfehler. Besonders auffällig ist hierbei Klasse 1 in Bild 5.13 und Klasse 3 in Bild 5.14. Im ersten Fall werden alle Testdaten von Klasse 1 der Klasse 3 zugewiesen, im zweiten Fall werden 14 von 15 Testdatentupeln von Klasse 3 der Klasse 9 zugewiesen. Dies führt zu vielen Klassifikationsfehlern.

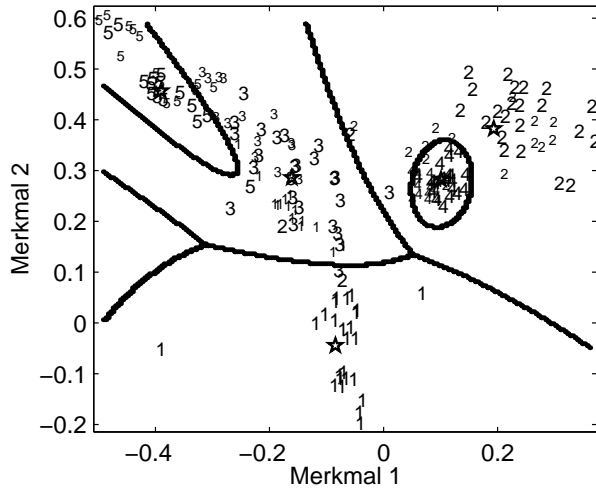


Bild 5.13: Darstellung von Lern- (große Zahlen) und Testdaten (kleine Zahlen) im zweidimensionalen Merkmalsraum, ohne robusten Entwurf, Patient F

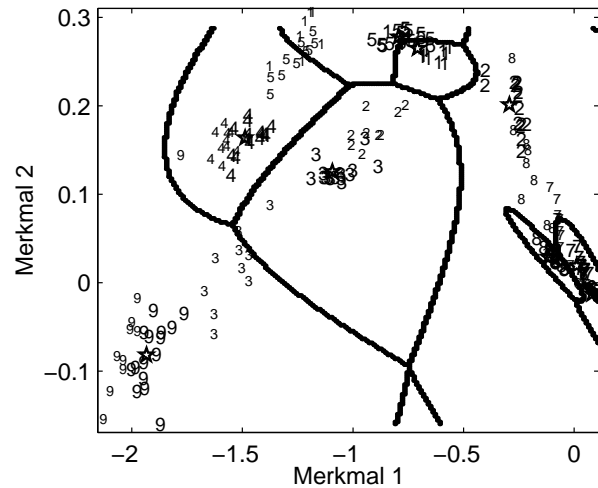


Bild 5.14: Darstellung von Lern- (große Zahlen) und Testdaten (kleine Zahlen) im zweidimensionalen Merkmalsraum, ohne robusten Entwurf, Proband S

Der hohe Klassifikationsfehler ergibt sich aus zwei Ursachen:

1. Viele der verwendeten Merkmale weisen in ihren Realisierungen strukturelle Unterschiede auf, wenn der Patient Schaltsignale nicht gut reproduziert. Wird für ein Schaltsignal bestehend aus zwei Kontraktionen die zweite Kontraktion zu schwach oder zu schnell ausgeführt, so erkennt die Merkmalsextraktion nicht drei Bereiche, sondern nur einen. Die relativen Merkmale für Bereich II sind dann immer gleich 1 und alle Merkmale für Bereich IV gleich 0. Dies führt zu nicht kompakten Klassen im Merkmalsraum (Klassen 2 und 6 in den Bildern 5.15 und 5.16).
2. Insbesondere die modifizierte Merkmalsauswahl (MM) neigt dazu, bei Vorhandensein strukturvariabler Merkmale einen überlernten Klassifikator zu liefern (z. B. für Klasse 6 in Bild 5.15). Ursache ist die nichtkompakte Klassenverteilung bei strukturvariablen Merkmalen, die zu einer niedrigen, aber nur flach abfallenden Wahrscheinlichkeitsdichtefunktion führt. Das gute Erkennen der Klasse über Lerndaten führt zur Auswahl des entsprechenden Merkmals, die Klassifikation widerspricht jedoch den philosophischen Ansätzen des AML-Klassifikators und versagt über unbekannten Testdaten. In Bild 5.15 sind zwei ausgewählte Merkmale des MM-Verfahrens dargestellt. Für die Klasse 6 rechtfertigt das Merkmal "S2 rel. Dauer Bereich II" den Ansatz einer Normalverteilung nicht, da nur die Werte 0 oder 1 angenommen werden und somit nicht-kompakte Klassen entstehen.

Problem 1 lässt sich zwar durch eine kleinere Filterkonstanten a aus (2.24) umgehen, dies führt jedoch zu unzuverlässigeren Bereichsdetektionen bei oszillierendem Signal - eine Robustheitssteigerung ergibt sich nicht. Problem 2 lässt sich auf den Mangel an Daten zurückführen und kann durch die Einführung von a-priori-Merkmalrelevanzen gelöst werden. Da diese Relevanzen jedoch patienten- und sogar schaltsignalindividuell vorzugeben sind, verhindert diese Möglichkeit den vollautomatischen Entwurf.

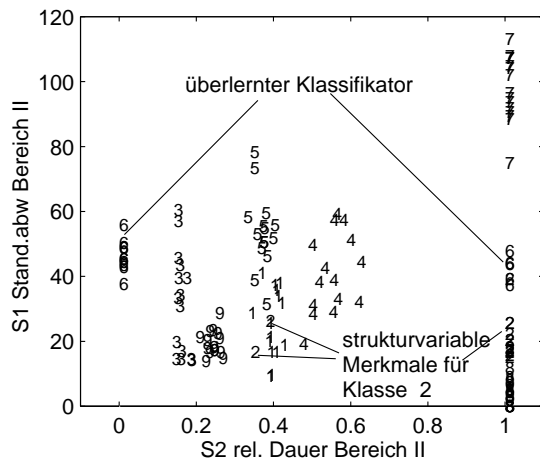


Bild 5.15: Effekte von Überlernen und strukturvariablen Merkmalen bei der Darstellung von zwei Merkmalen, Proband S, ein Datensatz

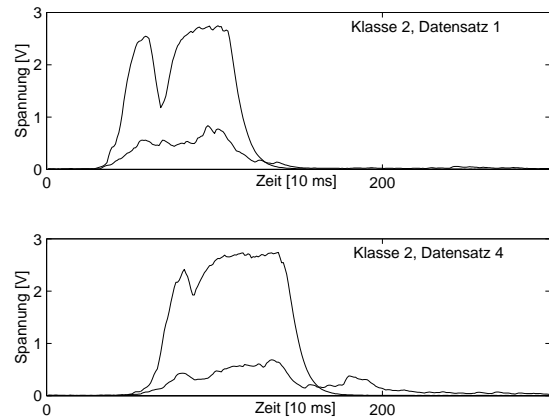


Bild 5.16: oben: korrekte Kontraktion, unten: zu schnell aufeinander folgende Kontraktionen, nur ein Bereich wird erkannt.

Um die Auswirkungen der Probleme zu beheben, bieten sich die Verfahren zum robusten Entwurf aus Abschnitt 3.4 an. Aufgrund der starken systematischen Änderungen versagt hier die künstliche Erweiterung des Lerndatensatzes. Eine einfache Anpassung an die vorliegenden Daten ist nicht möglich. Auch ein Entfernen von Klassen liefert nur beschränkten Erfolg und vermindert die Funktionalität des MMIs. Deshalb bietet sich ein Anlernen des Klassifikators auf der Basis eines Lerndatensatzes aus zwei voneinander unabhängigen Aufnahmesitzungen (Sitzung 1 und Sitzung 2) an.

Klassifikator	MAN/DI	MAN/MD	MM/DI	MM/MD	HKK	KO	Ø
Sitzung 3	16.0	12.0	8.0	14.5	10.7	10.7	12.0
Anlerndauer in s	1.4	2.7	3.4	5.1	8.6	4.6	
MMI-Index (10000Λ)	3	52	264	11	84	101	

Tabelle 5.7: Klassifikationsfehler bei Patient F, robuster Entwurf, Lerndatensatz aus Sitzung 1 und 2, Testdatensatz aus Sitzung 3

Klassifikator	MAN/DI	MAN/MD	MM/DI	MM/MD	HKK	KO	Ø
Sitzung 3	17.8	18.5	13.3	11.1	21.5	11.1	15.6
Sitzung 4	4.4	4.4	1.1	2.2	12.2	1.1	4.2
Sitzung 5	7.8	6.7	11.1	7.8	21.1	4.4	9.8
Sitzung 6	14.4	8.9	5.6	4.4	16.7	3.3	8.8
Ø	11.1	9.6	7.8	6.4	17.9	5.0	9.6
Anlerndauer in s	2.3	10.0	6.7	15.0	27.4	10.9	
MMI-Index (10000Λ)	32	123	240	344	0.1 ⁷	211	

Tabelle 5.8: Klassifikationsfehler bei Proband S, robuster Entwurf, Lerndatensatz aus Sitzung 1 und 2, Testdatensatz aus Sitzung 3-6

Die Tabellen 5.7 und 5.8 stellen den Klassifikationsfehler über den Daten der Sitzungen 3 bis 6 dar. Die Hinzunahme des zweiten Datensatzes führt zu wesentlich robusteren Ergebnissen, so dass sich die Klassifikatoren in eine Prothese integrieren lassen. Bei Patient F liefert das MM/DI-Verfahren mit 8% Klassifikationsfehler die besten Ergebnisse (unrobuster Entwurf: 28.7% Fehler), das herkömmliche MAN/DI-Verfahren mit 16% die schlechtesten.

Die besten Ergebnisse bei Proband S liefert im Mittel das KO-Verfahren mit 5.0% Klassifikationsfehler (unrobuster Entwurf: 41.5%). Alle modifizierten Verfahren weisen eine Steigerung der Klassifikationsgüte gegenüber dem herkömmlichen MAN/DI-Verfahren auf. Einzig das HKK-Verfahren liefert bei Patient F schlechte Werte, was allerdings in einer falschen Parametrierung begründet liegt⁸ und bereits in Tabelle 5.4 abzusehen war - die neun Klassen des Testdatensatzes lassen sich nicht in zwei Dimensionen unterscheiden. Klassifikationsgüten für andere Parametrierungen zeigt Anhang 7.2.1.

Die Betrachtung des MMI-Index zeigt sehr große Vorteile der modifizierten Verfahren gegenüber dem herkömmlichen MAN/DI-Verfahren. Die Einschnittverfahren erweisen sich aufgrund ihres geringen Berechnungsaufwands als sehr kompetent, auch wenn sie nicht die besten Klassifikationsgüten liefern. Der KO-Klassifikator fällt zwar im MMI-Index ab, da die Anzahl der Multiplikationen quadratisch von der großen Klassenanzahl abhängt, liefert aber die besten Klassifikationsergebnisse. Das HKK-Verfahren weist ebenfalls gute Werte auf.

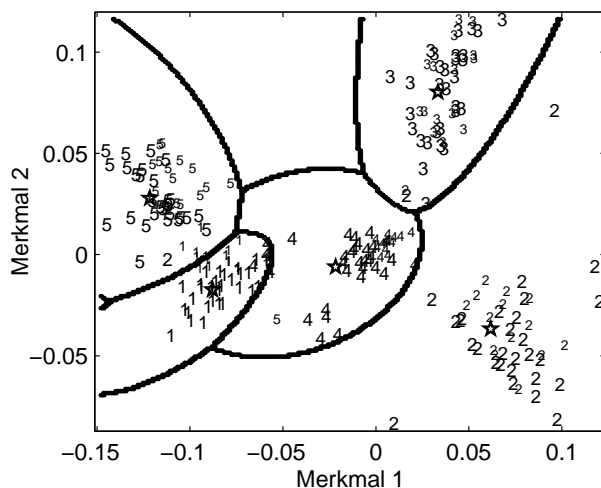


Bild 5.17: Darstellung von Lern- (große Zahlen) und Testdaten (kleine Zahlen) im zweidimensionalen Merkmalsraum, robuster Entwurf, Patient F

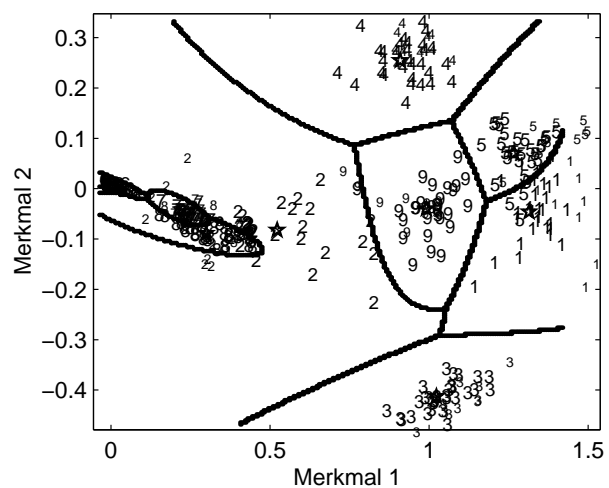


Bild 5.18: Darstellung von Lern- (nur Zahlen) und Testdaten (Zahlen in Kreisen) im zweidimensionalen Merkmalsraum, robuster Entwurf, Proband S

Eine Betrachtung von Lerndaten und angelegten Diskriminanzfunktionen im Vergleich zu den Testdaten geben die Bilder 5.17 und 5.18. Das dargestellte Problem der Drift der Testdaten ist durch den robusten Entwurf fast vollständig eliminiert worden. Die Testdaten decken nun den gleichen Bereich im Merkmalsraum ab wie die Lerndaten. Es ist wiederum zu beachten, dass hier nur zweidimensionale Projektionen und nicht das eigentliche höherdimensionale Klassifikationsproblem dargestellt sind.

⁷Der Klassifikator liefert bei individueller Parametrierung vergleichbare Ergebnisse wie die anderen Verfahren, s. u.

⁸Hier wurde mit der Parametrierung gearbeitet, die dem Benchmarkbeispiel entnommen wurde. Mit $s_m = 8$ und $s_d = 3$ ergibt sich beispielsweise ein durchschnittlicher Klassifikationsfehler von 9.3%.

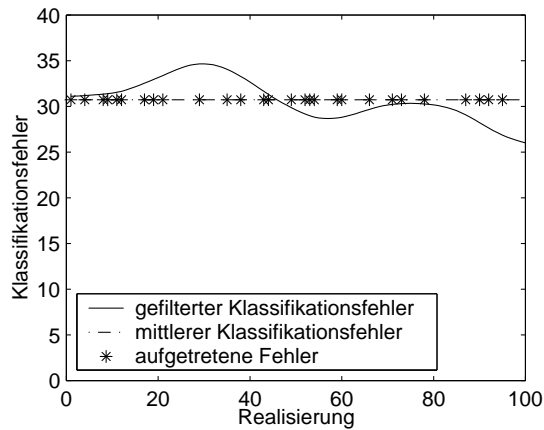


Bild 5.19: Gefilterter Verlauf des Klassifikationsfehlers für Patient F bei Simulation des Echtzeitbetriebs, ohne robusten Entwurf

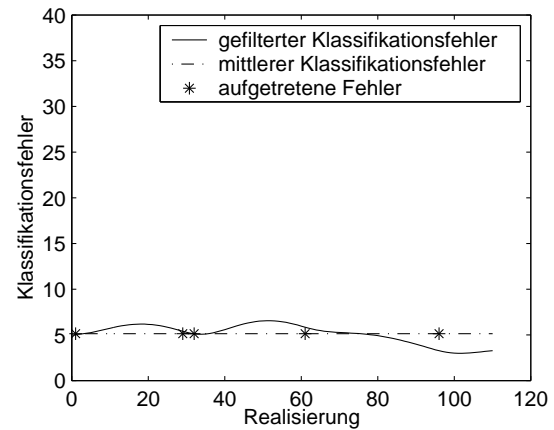


Bild 5.20: Gefilterter Verlauf des Klassifikationsfehlers für Patient F bei Simulation des Echtzeitbetriebs, robuster Entwurf

Zur Evaluation des Echtzeitbetriebs wurde das MAN/DI-Verfahren mit $s_m = 8$ und $s_d = 3$ aus DAVE-Design exportiert und in DAVE implementiert⁹. DAVE generiert zufällig Vorgaben für nachzunehmende Schaltsignale, die Patient F äquivalent zum Betrieb einer Prothese zu generieren versucht. Den gefilterten Verlauf des Klassifikationsfehlers mit Anlernen über einer und über zwei Sitzungen zeigen die Bilder 5.19 und 5.20.

Im Vergleich zum unrobusten Entwurf (mittlerer Klassifikationsfehler: 31%) liefert der robuste Entwurf (mittlerer Klassifikationsfehler: 5%) sehr gute Ergebnisse. Charakteristisch ist ein Abfallen des Klassifikationsfehlers, je länger der Patient mit diesem arbeitet, unabhängig davon, ob mit einem oder zwei Datensätzen angelernt wurde. Die Bilder belegen die Fähigkeit eines Patienten, eine Prothese auf der Basis der vorgestellten, modifizierten Verfahren zu bedienen. Weiterhin führt das Erlernen der Steuerung seitens des Patienten zu sinkenden Klassifikationsfehlern (im vorliegenden Fall auf 5%).

In einem weiteren Vorversuch zur Funktionalität von Schaltsignalen konnte Patient C bereits die Bedienbarkeit der Steuerung nachweisen (Bilder 5.21 und 5.22, Prothese am rechten Arm). Verwendet wurde eine Prothese, Typ 2 [183], die mittels vier zeitvarianter Schaltsignaltypen die fünf vorgegebenen Griffarten umsetzen konnte. Zum Einsatz kamen zwei Otto Bock MYOBOCK-Sensoren, die Abtastfrequenz betrug $f_a = 100\text{Hz}$. Eine einheitliche Aufzeichnung der Schaltsignale war zu diesem Zeitpunkt technisch noch nicht möglich.

5.2.11 Resultat

Die modifizierten Verfahren zeigen durchweg bessere Ergebnisse als die herkömmlichen Verfahren. Dabei steigt nicht nur der MMI-Index, sondern zudem auch die Klassifikationsgüte. Auf einen Einsatz des HKT-Verfahrens muss allerdings aufgrund des niedrigen Datensatzumfangs verzichtet werden. Sind viele Griffarten (Klassen) umzusetzen, so empfiehlt sich der Einsatz der modifizierten Einschnittverfahren aufgrund der niedrigen auszuwertenden Merkmalsdimension. In Abhängigkeit von der Anzahl ausgewählter strukturvariabler Merkmale ist hierbei patientenindividuell über den Einsatz des MM-Verfahrens

⁹Zum Zeitpunkt dieses Versuchs war der Export weiterer modifizierter Klassifikatoren noch nicht implementiert, für die dargestellten Versuche liegen nur die Klassifikationsergebnisse, nicht aber die Aufzeichnung der Schaltsignale vor.



Bild 5.21: Patient C mit Prothese, Lateralgriff zum Halten eines Löffels



Bild 5.22: Patient C mit Prothese, Zylindergriff zum Halten einer Dose

zu entscheiden. Sinkt die Anzahl der Griffarten, so lohnt sich der Einsatz des KO-Verfahrens. Insbesondere mit Blick auf hohe Klassifikationsgüten empfiehlt sich dieses Verfahren. Das HKK-Verfahren zeigt Vorteile, wenn der Datensatz einige Klassen enthält, die sich sehr gut abspalten lassen.

Die durchgeführten Patiententests zeigen, dass der vollautomatische Entwurf eines MMIs für Handprothesen sehr erfolgreich ist. Alle Patienten sind in der Lage, die geforderte Mindestanzahl an Schaltsignaltypen zu generieren. Für einen geringen Klassifikationsfehler ist nur ein Mindestmaß an Training erforderlich. Dann können auch die auf der Basis des Benchmarkdatensatzes entwickelten Parametrierungen zum Einsatz kommen. Um ein robustes MMI zu entwerfen, eignet sich insbesondere das Anlernen der Klassifikatoren über zwei separaten Sitzungen. Zudem zeigt sich im Testbetrieb eine schnelle Anpassung des Patienten an seine Prothesensteuerung.

Die Rechenkapazitäten der verwendeten Mikrocontroller (zuerst Infineon C164CI, dann ATMEL ATmega128) waren ausreichend, um die implementierten Systeme in Echtzeit zu betreiben.

5.3 Brain-Computer-Interface

5.3.1 Datensätze

Gemäß Abschnitt 1.2.4 basieren Brain-Computer-Interfaces (BCI) bislang größtenteils auf binären Entscheidungen, denen aufwändige Auswerteroutinen zugrundeliegen. Die wohl wichtigste Anwendung für BCIs auf der Basis von EEG-Signalen stellen virtuelle Tastaturen für Locked-In-Patienten (Lähmung aller Muskeln) dar, mittels derer sich für die Betroffenen eine Kommunikationsmöglichkeit mit ihrer Umwelt eröffnet. Durch das Denken an zwei verschiedene Ereignisse (z. B. linke und rechte Handbewegungen) wählt der Anwender aus den 26 Buchstaben des Alphabets 13 aus und wiederholt den Vorgang so lange, bis schließlich ein Buchstabe übrig bleibt (Bild 5.23, [19, 175]).

Jedem Zustand des Automaten aus Bild 5.23 (externer Automat) unterliegt ein Zustandsautomat mit zwei Aktionszuständen ohne neutralen Zustand gemäß der Kategorisierung aus Abschnitt 2.2. In diesen internen Zustandsautomaten kommen zeitpunktbasierte Schaltsignale zum Schalten ohne Zeitversatz zum Einsatz, d. h. es werden fortwährend Sensorinformationen ausgewertet und als Schaltsignale im

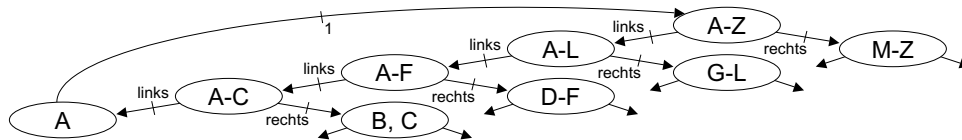


Bild 5.23: Automatengraph eines virtuellen Keyboards

Automaten verwendet. Der Verbleib im gleichen Zustand über einen vorgegebenen Zeitraum wird als Entscheidung interpretiert (z. B. Teilung des Alphabets) und der Übergang des externen Zustandsautomaten in einen neuen Zustand startet einen neuen internen Zustandsautomaten.

Um die Anstrengungen auf diesem Gebiet und die entstandenen Algorithmen zu vergleichen, findet im Jahrestakt der Wettbewerb "BCI-Competition" statt. Diverse Forschergruppen stellen hierbei ihre Datensätze zum Benchmarking zur Verfügung. Um die Übertragbarkeit des vorgeschlagenen MMI-Entwurfs zu belegen, führt diese Arbeit den MMI-Entwurf anhand des Datensatzes III der "BCI-Competition 2003" durch [128].

Der verwendete Datensatz stammt von einer gesunden Probandin (weiblich, 25 Jahre), die durch die Vorstellung von imaginären linken und rechten Handbewegungen ($K = 2$ Klassen) versucht, einen Balken auf einem Bildschirm zu steuern. Der Datensatz umfasst sieben Durchläufe mit jeweils 40 Versuchen. Ein Versuch besteht aus $n_s = 3$ bipolaren EEG-Signalen der Kanäle mit den Kurzbezeichnungen C3, Cz und C4¹⁰, die mit $f_a = 1000\text{Hz}$ über 9s aufgezeichnet werden. Die ersten zwei Sekunden besteht keine Aktivität, zum Zeitpunkt $t = 2\text{s}$ zeigt ein akustisches Signal den Beginn des Experiments an und ein visuelles Signal gibt die zu reproduzierende Vorgabe. Im Zeitraum $t = 3 - 9\text{s}$ versucht der Proband die Vorgabe zu reproduzieren. Die zufällige Aufteilung des erhaltenen Datensatzes führt zu einem Lern- und einem Testdatensatz (jeweils $n = 140$, $n_j = 70$, $j = 1, 2$).

Als Merkmale kommen geschätzte Parameter eines adaptiven autoregressiven Modells (AAR) zum Einsatz (AiCj: i -ter Parameter des j -ten Kanals) sowie die Bandleistung (BP) in zwei Frequenzbereichen (B10Cj: 10-12Hz, B16Cj: 16-24Hz).

Das Bestimmen der AAR-Parameter geschieht durch die Funktion AAR.m der MATLAB-Toolbox BIOSIG¹¹ [178]. Der Bandleistung liegt ein Butterworth-Filter 5. Ordnung zugrunde, dessen Wert quadriert und in einem Rechteckfenster der Länge 1s geglättet wird. Abschließend findet eine Logarithmierung statt. Auf diese Weise entstehen zu jedem Zeitpunkt $s = 15$ Merkmale (pro Kanal 3 AAR-Koeffizienten und 2 Bandleistungen).

5.3.2 Methoden

Zum Vergleich der Güte der Merkmalskategorien AAR und BP werden zusätzlich Datensätze betrachtet, die ausschließlich aus AAR-Merkmalen oder ausschließlich BP-Merkmalen bestehen. Alle folgenden Verarbeitungsschritte entstammen der in dieser Arbeit entwickelten MATLAB-Toolbox DAVE-Design. Zur Merkmalsauswahl kommt das MANOVA-Verfahren zum Einsatz. Für jeden Zeitpunkt k liefert es die Relevanzen $M(\mathcal{I}[k])$ für eine Kombination von Merkmalen $\mathcal{I}[k]$ ($s_m > 1$). Der Zeitpunkt $k_{M_{\max}}$ mit maximalem M dient der Konstruktion der AML-Entscheidungsregel auf der Basis der s_m Merkmale, die durch lineare Diskriminanzanalyse (LDA) zu $s_d = 1$ Merkmal aggregiert werden. Vergleichsweise findet auf der Basis der gleichen Merkmale der Entwurf einer Fuzzy-Regelbasis statt.

¹⁰Es handelt sich um standardisierte Bezeichner zur Detektion von Signalen des motorischen Kortex [96].

¹¹mode = [0, 0], Modellordnung p=3 und Update-Koeffizient UC = 0.0055

Der Einsatz der modifizierten Verfahren lohnt hier nicht, da das Problem hinreichend einfach ist ($K = 2$ Klassen, nahezu normalverteilt). Sowohl KO- als auch HKK zerfallen für das Zweiklassenproblem zu einem Einschnittverfahren, lediglich HKT kann hier Vorteile bringen. Weiterhin sind alle Verfahren (bis auf HKT) auf einen Einsatz für Mehrklassenprobleme und Implementierbarkeit optimiert und können deshalb hier keine Verbesserungen liefern. Zur besseren Vergleichbarkeit finden sich trotzdem Klassifikationsergebnisse in Tabelle 5.11.

Tabelle 5.9 zeigt die maximalen Werte der MANOVA-Relevanzen sowie die ausgewählten Merkmale. Es bezeichnet t_{min} den ersten Zeitpunkt mit einer Merkmalsrelevanz größer 0.4:

$$t_{min} = f_a^{k_{min}} \quad \text{mit } k_{min} : M(\mathcal{I}[k] \leq 0.4) \quad \text{und} \quad M(\mathcal{I}[k_{min}] > 0.4), \quad k = 1, \dots, k_{min} - 1. \quad (5.3)$$

Die Relevanzen für einen einzelnen Kanal sind signifikant kleiner als die Relevanzen für die Kombination der Kanäle C3 und C4. Ursache ist die Sensorpositionierung, die auf jedem Kanal nur Informationen über eine Klasse enthält. Cz enthält hingegen redundante Informationen für das vorgegebene Problem. Beide Merkmalsätze (AAR- und BP) führen zu guten Relevanzen, jedoch zeigt die Verwendung von AAR-Merkmalen erst gute Ergebnisse ab $s_m = 4$, die BP-Merkmale benötigen nur $s_m = 2$ Merkmale. Hingegen führt die Verwendung der AAR-Merkmale schneller zu hohen Relevanzen ($t_{min} = 4.4s$) als die Verwendung der BP-Merkmale ($t_{min} = 5.2s$). Bei Betrachtung nur eines Kanals wird $M(\mathcal{I}[k]) > 0.4$ nicht erreicht.

Merkmale (Kanal)	s	t_{min}	$s_m = 1$	$s_m = 2$	$s_m = 3$	$s_m = 4$	$s_m = 5$
AAR+BP (C3)	5	-	0.20 (B10C3)	0.21 (B16C3)	0.22 (A3C3)	0.22 (A2C3)	0.22 (A1C3)
AAR+BP (Cz)	5	-	0.07 (B10Cz)	0.09 (A3Cz)	0.11 (B16Cz)	0.17 (A1Cz)	0.18 (B16Cz)
AAR+BP (C4)	5	-	0.17 (A3C4)	0.30 (B16C4)	0.33 (B10C4)	0.33 (A1C4)	0.33 (A2C4)
AAR+BP (C3+Cz)	10	-	0.20 (B10C3)	0.26 (B10Cz)	0.29 (A3Cz)	0.32 (A2Cz)	0.36 (A1Cz)
AAR+BP (C3+C4)	10	4.1s	0.20 (B10C3)	0.51) (B10C4)	0.55 (A3C4)	0.57 (A2C4)	0.57 (A3C3)
AAR+BP (C4+Cz)	10	4.5s	0.17 (A3C4)	0.47 (B10Cz)	0.51 (B10C4)	0.52 (A2C4)	0.53 (A1Cz)
AAR (C3+C4)	6	4.4s	0.17 (A3C4)	0.34 (A3C3)	0.37 (A2C4)	0.47 (A1C3)	0.48 (A1C4)
BP (C3+C4)	4	5.2s	0.20 (B10C3)	0.51 (B10C4)	0.52 (B16C3)	0.52 (B16C4)	
AAR+BP (alle Ci)	15	4.1s	0.20 (B10C3)	0.51 (B10C4)	0.55 (A3C4)	0.57 (A2C4)	0.57 (B10Cz)

Tabelle 5.9: Maximale Relevanzen $M(\mathcal{I}[k_{Mmax}])$ für ausgewählte Merkmale über Lerndaten

Klassifikationsergebnisse zeigt Tabelle 5.10. Hierbei wird der Klassifikationsfehler pro Abtastzeitpunkt über alle 140 Realisierungen des Testdatensatzes gemittelt. Der minimale Klassifikationsfehler beschreibt das Minimum des gemittelten Klassifikationsfehlers über alle Abtastzeitpunkte. Die besten Ergebnisse liefert die AML-Entscheidungsregel mit AAR- und BP-Merkmalen und einer Auswahl von

$s_m = 3$ Merkmalen. Im Vergleich zu den neun am Wettbewerb teilnehmenden Forschergruppen entspricht dies dem drittbesten Ergebnis. Der minimale Klassifikationsfehler liegt in den besten 10% bezogen auf alle anderen Gruppen [163]. Zudem zeigt Tabelle 5.10, dass AAR- und BP-Merkmale nicht redundant sind und die ausschließliche Verwendung zu schlechteren Klassifikationsergebnisse führt (20.7% für AAR, 17.1% für BP). Weiterhin wird die Erkenntnis bestätigt, dass Fuzzy-Regelbasen zwar recheneffizient arbeiten, aber schlechtere Klassifikationsergebnisse liefern als ein AML-Klassifikator.

Entscheidungsregel	Merkmale (s_m)	min. Klassifikationsfehler (Pos.)
AML	AAR+BP (3)	14.3 (3)
AML	AAR (4)	20.7 (7)
AML	BP (2)	17.1 (4)
Fuzzy (6 Regeln)	AAR+BP (7)	25.7 (7)
Fuzzy (7 Regeln)	AAR (7)	28.6 (7)
Fuzzy (2 Regeln)	BP (2)	21.4 (7)
Ergebnisse in [177]		10.7-49.2 (1-9)

Tabelle 5.10: Vergleich von Entscheidungsregeln, Merkmalsätzen und Position in der "BCI-Competition 2003"

Tabelle 5.11 zeigt, dass die modifizierten Verfahren über dem EEG-Datensatz vergleichbare Ergebnisse liefern, wie die in Kapitel 5.3.2 dargestellten. Wiederum ist hier jedoch die Merkmalsdimension auf $s_d = 1$ erniedrigt, wodurch sich der Rechenaufwand verringert.

Datensatz	min. Klassifikationsfehler
MANOVA, MD, AML, $s_m = 3, s_d = 1$	15.0
MM, LDA, AML, $s_m = 3, s_d = 1$	15.0
MM, MD, AML, $s_m = 3, s_d = 1$	15.0
HKT, $s_m^{\min} = 3, m_{\max} = 2, s_d = 1$	16.4

Tabelle 5.11: Ergebnisse der modifizierten Verfahren über dem EEG-Datensatz

Bild 5.24 stellt die Merkmalsrelevanzen des MANOVA-Verfahrens bei Verwendung aller Merkmale über der Zeit und in Abhängigkeit von der Anzahl ausgewählter Merkmale dar. Es zeigt sich deutlich, dass die Hinzunahme eines vierten Merkmals kaum einen Gewinn in der maximalen Merkmalsrelevanz liefert. Ebenso zeigt sich, dass vor dem Zeitpunkt $t = 4s$ kaum Information extrahiert werden kann, obwohl der Proband bereits bei $t = 3s$ den Versuch startet. Dies lässt sich einerseits durch die Zeitverzögerung beim Generieren der Merkmale (Rechteckfilter, Schätzvorschriften), andererseits durch Reaktionszeit und Konzentrationsaufbau des Probanden begründen.

Bild 5.25 zeigt die mittlere Klassifikationsgüte über allen Testdaten. Beginnend mit $t = 4s$ steigt sie bis zu einem Maximum von 85.7% bei $t = 5.3s$ an und fällt dann wieder ab.

Fuzzy-Verfahren weisen etwas schlechtere Ergebnisse auf, da sie achsenparallele Einzugsgebiete erzeugen. Dies führt bei korrelierten Merkmalen und gedrehten Streuellipsoiden für beide Klassen zu Problemen. Der Klassifikatorentwurf muss (insbesondere bei AAR) mit relativ ungeeigneten Merkmalen auskommen. Das erzwingt z.T. viele Regeln und tendenziell eine zu starke Anpassung an die Lerndaten.

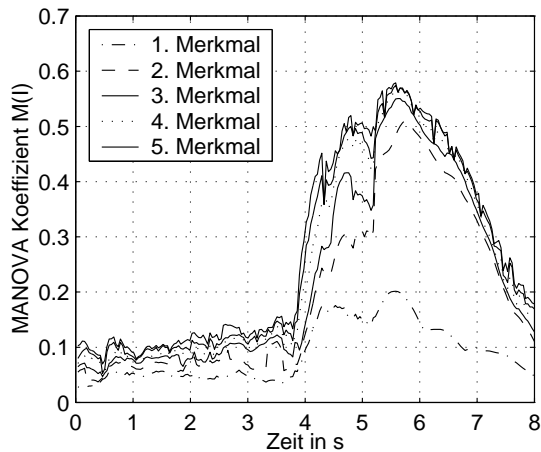


Bild 5.24: MANOVA-Koeffizienten, alle Merkmale

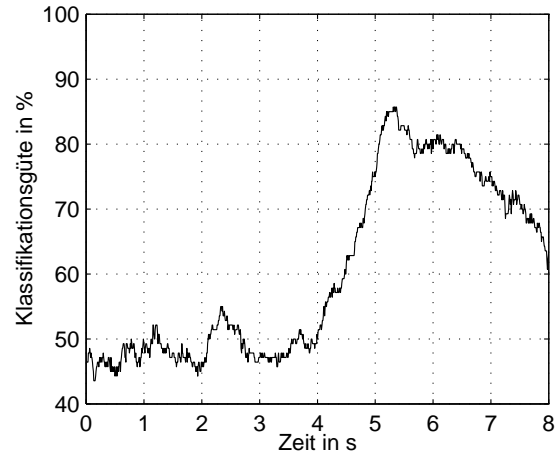


Bild 5.25: Klassifikationsgüte, alle Merkmale, Testdaten

Eine aus BP-Merkmalen entworfene Regelbasis arbeitet über dem Lerndatensatz mit nur einer Regel und der SONST-Regel mit Konklusion $y = 2$ (linke Handbewegung) und führt bei Crossvalidierung und über Testdaten zu den besten Ergebnissen. Bild 5.26 zeigt die gefundene Fuzzy-Regel:

WENN (B10C3 = NICHT NSK)
UND (B10C4 = NM ODER NK ODER NSK) DANN $y = \text{rechts}$.

mit den Bezeichnern NSK (negativ sehr klein), NK (negativ klein), NM (negativ mittel). Hierbei ist deutlich zu erkennen, dass eine nicht-achsparallele Klassifikation bessere Ergebnisse liefert.

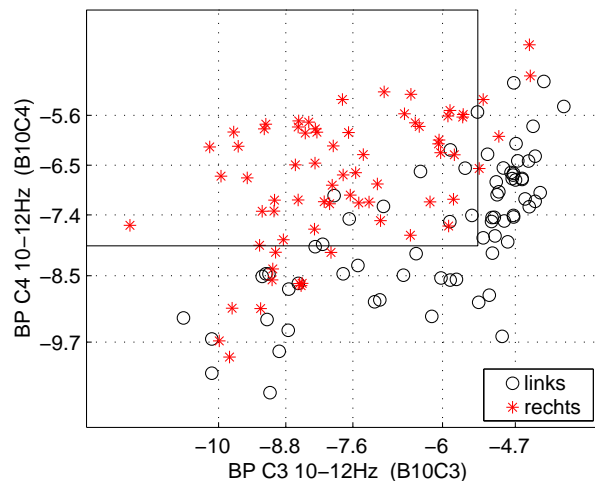


Bild 5.26: Fuzzy-Regel (Rechteck) und Parameter für die Zugehörigkeitsfunktionen (gestrichelte Linien für $\mu = 1$)

5.3.3 Ergebnisse und Diskussion

Die Merkmalsrelevanzen des MANOVA-Verfahrens über dem Lerndatensatz sind als Bewertungskriterium geeignet, um einen Klassifikator anzulernen. Die Berechnung ist schneller als ein möglicher

Wrapper-Ansatz und gibt zudem Aussagen über Distanzen zwischen einzelnen Klassen.

Der große Vorteil der Fuzzy-Regelbasis ist der Wissensgewinn bei einer Offline-Analyse des Problems. Zweidimensionale Darstellungen von Regeln und Regelbasen, wie in Bild 5.26 dargestellt, sind z. B. eine interessante Möglichkeit, um Patienten Feedback für ein verbessertes Training zukommen zu lassen. Leider führen sie zu einem höheren Klassifikationsfehler im Vergleich zur AML-Entscheidungsregel nach Merkmalsselektion und -aggregation. Hauptgründe hierfür sind die näherungsweise normalverteilten Merkmale mit starken Korrelationen, die Probleme für achsparallele Diskriminanzfunktionen darstellen. Da Merkmalstransformationen zur Minimierung des Klassifikationsfehlers dem Grundgedanken des interpretierbaren Fuzzy-Entwurfs widersprechen, wird darauf verzichtet.

Auf die Darstellung des MMI-Index wird ebenfalls verzichtet, da eine Echtzeit-Implementierung nur auf leistungsstarken Systemen durchgeführt werden kann und hierbei wesentlich mehr Rechenzeit in die Merkmalsextraktion als in die Klassifikation investiert werden muss.

Eine Anwendung des vorgeschlagenen MMI-Entwurfs auf der Basis von Wavelet-Methoden zeigt minimal schlechtere Ergebnisse, jedoch verringert sich die Zeitverzögerung bis zum Erreichen der maximalen Klassifikationsgüte [98].

Die vorangegangenen Abschnitte zeigen die Übertragbarkeit des MMI-Entwurfs auf ein Problem, das große strukturelle Unterschiede zum vorangegangenen Entwurf für Handprothesen aufweist. Bereits ohne Modifikation der Verfahren ergeben sich sehr gute Ergebnisse.

Bei der Teilnahme an der "BCI Competition III" (2005) konnte mit der vorgestellten Systematik der zweite von sieben Plätzen erreicht werden [28].

5.4 Bewertung

Die Funktionsweise des vorgeschlagenen MMI-Entwurfs wurde in diesem Kapitel anhand von zwei realen und praxisrelevanten Beispielen belegt. Im MMI der Prothesensteuerung verbessern die in dieser Arbeit entwickelten modifizierten Verfahren nicht nur den MMI-Index, sondern auch die Klassifikationsgüte. Für alle untersuchten 17 Patienten existiert somit die Möglichkeit des Entwurfs eines individuellen MMIs. Die Simulation des Echtzeitbetriebs belegt die Einsatzfähigkeit des Entwurfs bei einem Patienten und einem Probanden. Ein weiterer Patient war schließlich in der Lage, eine reale Prothese mittels des individuell angepassten MMIs zu bedienen. Der Entwurf des MMIs für Brain-Computer-Interfaces demonstriert die Übertragung auf andere auf bioelektrischen Signalen basierende MMIs. Ohne spezielle Anpassung erreicht das vorgeschlagene System über den Daten des Wettbewerbs "BCI-Competition 2003" den dritten von neun Plätzen und erreichte bei der Teilnahme an der "BCI Competition III" (2005) sogar den zweiten von sieben Plätzen.

Aus den Aufnahmen der Patientendaten und der realen Anwendung der vorgestellten MMIs lassen sich folgende generelle Aussagen ableiten:

- Patienten sind grundsätzlich in der Lage, Schaltsignale zu generieren und zu reproduzieren. Die Art des Schaltsignals muss im Einzelfall an die Anatomie des Patienten angepasst werden.
- Eine hohe Güte des MMIs kann gewährleistet werden, indem die Schaltsignale fünf bis zehn Minuten lang trainiert werden, bevor die Datenaufnahme durchgeführt wird.
- Ein Umfang von 10 Schaltsignalrealisierungen pro Klasse ist notwendig, um einen geringen Klassifikationsfehler zu erhalten.

- Um den Einfluss strukturvariabler Merkmale zu schmälern und ein robustes MMI zu erstellen, empfiehlt es sich, Daten aus mindestens zwei verschiedenen Aufnahmesitzungen zum Entwurf des MMIs zu verwenden.

Folgende Empfehlungen werden für die Parametrierung des Systems vorgeschlagen:

- Bei aufwändigen Problemen mit vielen Daten, zeitpunktbasierten Schaltsignalen und wenigen Klassen eignen sich die geschlossen lösbaren Verfahren MANOVA und lineare Diskriminanzanalyse (LDA).
- Beim Schalten mit Zeitversatz und einer höheren Klassenanzahl liefern die modifizierte Einschrittverfahren MM (modifizierte Merkmalsauswahl) und MD (modifizierte Diskriminanzanalyse) sehr gute Ergebnisse.
- Liegen strukturvariable Merkmale vor, so überflügeln Mehrschrittverfahren mit vorheriger Merkmalsauswahl durch MANOVA oder MM alle anderen Verfahren. Insbesondere bei großen strukturellen Unterschieden eignet sich die hierarchische Klassifikation nach Klassen (HKK).

Gemäß dieser Parametrierung steht ein leistungsfähiges Verfahren für den Entwurf beliebig gearteter MMIs zur Verfügung.

6 Zusammenfassung

Der Entwurf von Mensch-Maschine-Schnittstellen auf der Basis von biometrischen Signalen stellt hohe Anforderungen an den Entwickler, da selbst einfachste Steuerungskonzepte aufwändig und patientenindividuell angepasst werden müssen. Im Fall von kommerziell erhältlichen Handprothesen ist diese Anpassung zudem manuell durchzuführen.

Zur Erweiterung von Funktionalität und Handhabbarkeit schlägt diese Arbeit den automatischen Entwurf von Mensch-Maschine-Schnittstellen vor. Die Grundidee ist die Implementierung eines struktur- und parametervariablen Systems auf der Basis von patientenindividuellen Beispielsignalen. Neue Entwurfsverfahren eröffnen die Möglichkeit, nicht nur den Klassifikationsfehler zu senken, sondern quantifizieren auch die Größen Implementierbarkeit, Robustheit und Komplexität.

Die Kernpunkte dieser Arbeit umfassen

- die Entwicklung einer Entwurfsmethodik zur quantitativen Beschreibung biometrischer Signale sowie zur Auswahl und Bewertung bekannter Verfahren zur Signalauswertung gemäß einem neuen Kriterium für Mensch-Maschine-Schnittstellen (Kapitel 2),
- die Modifikation und die Erweiterung der Verfahren bezüglich des neuen Kriteriums sowie der Nachweis der Funktionalität (Kapitel 3),
- die Entwicklung einer automatischen Implementierungsstrategie zur Unterstützung von Anwender und Entwickler (Kapitel 4) sowie
- die Validierung der erarbeiteten Verfahren anhand eines umfangreichen Patientenkollektivs für zwei Arten von Mensch-Maschine-Schnittstellen (Kapitel 5).

Die wichtigsten Ergebnisse der Arbeit sind:

1. Entwicklung eines Konzeptes zum automatischen und systematischen Entwurf von Mensch-Maschine-Schnittstellen mit den Elementen Merkmalsextraktion aus biometrischen Zeitreihen, Merkmalsauswahl, Merkmalsaggregation und Entscheidungsregel.
2. Ableitung eines neuen Bewertungsmaßes zur quantitativen Beurteilung der Schnittstelle aus Patienten- und Entwicklersicht anhand von Klassifikationsgüte, Funktionalität, Implementierbarkeit und Entwicklungsaufwand.
3. Herleitung neuer Optimierungskriterien zur automatischen Anpassung von Merkmalsauswahl, -aggregation und Entscheidungsregel, die speziell auf Klassifikationsprobleme biometrischer Datensätze eingehen.
4. Spezifikation neuer Verfahrensabläufe (Mehrschrittverfahren) zum Entwurf von Klassifikatoren für Teilmengen eines Datensatzes unter Verwendung der neuen Optimierungskriterien zur Verminderung der Defizite herkömmlicher Einschrittverfahren.

5. Einführung des Schaltsignalkonzepts zum robusten Entwurf von Mensch-Maschine-Schnittstellen auf der Basis eines Zustandsautomaten zur Realisierung einer beliebigen Anzahl von Reaktionszuständen, Spezifikation für zwei MMIs und Nachweis der Funktionsfähigkeit.
6. Herleitung von robusten Strategien zur Parametrierung von MMIs bei zeitvarianten Signalen.
7. Entwicklung der grafischen Benutzeroberfläche DAVE zur patientengerechten Aufnahme, zum Entwurf, zur Simulation, zum Training und zur Modifikation von Mensch-Maschine-Schnittstellen.
8. Erstellung der neuen MATLAB-Toolbox DAVE-Design als Entwicklungs- und Evaluationsplattform zur Übertragung der erarbeiteten Algorithmen auf beliebige Problemstellungen.
9. Parametrierung einer neuen echtzeitfähigen Routine zur Extraktion von trennstarken Merkmalen aus biometrischen Aktivitätssignalen.
10. Entwicklung einer neuen Routine zur automatischen Quelltextgenerierung für Klassifikationssysteme zur Implementierung auf Mikrocontrollern.
11. Realisierung einer Mikrocontroller-Testumgebung zur Simulation entworfener MMIs auf Mikrocontrollern und zur anschließenden direkten Übertragung von Mikrocontrollern in Prothesen.
12. Nachweis der Funktionalität des neuen Konzeptes durch den kompletten Entwurf von Handprothesensteuerungen anhand realer Daten von 17 Patienten und zwei Probanden.
13. Nachweis von Robustheit und Echtzeitfähigkeit der entworfenen MMIs anhand von Daten mehrerer Aufnahmesitzungen eines Patienten und eines Probanden und anhand eines Patienten im Echtzeitversuch mit einer realen Prothese.
14. Nachweis der Übertragbarkeit des Entwurfsverfahrens auf beliebige MMIs anhand des Entwurfs eines Brain-Computer-Interfaces.

Das vorgestellte Verfahren ist vollständig in die dargestellten Entwicklungsumgebungen implementiert und kann auf beliebige Problemstellungen angewandt werden.

In Bezug auf die Prothesensteuerung müssen sich weiterführende Arbeiten mit dem Alltagseinsatz der vorgestellten Verfahren befassen und Probleme wie die tägliche Form des Anwenders, Positionierungsfehler von Sensoren, etc. evaluieren. Eine Erweiterung der Patientenbasis kann zudem dazu dienen, automatisch einen robusten Entwurf anhand von einem Lerndatensatz zu finden.

Derzeit finden Bemühungen statt, die Ergebnisse dieser Arbeit zur Erweiterung einer Mensch-Maschine-Schnittstelle auf der Basis von Nervensignalen zu verwenden. Hierfür sind neue Merkmalsextraktoren zu entwickeln und anhand von Patientendaten zu evaluieren.

Potenzial für weiterführende methodische Entwicklungen bieten die modifizierten Einschnitt- sowie die Mehrschrittverfahren. Hier muss untersucht werden, wie sich die Struktur der Verfahren automatisiert in Abhängigkeit vom zu verwendenden Datensatz festlegen lässt. Weiterhin haben bereits Arbeiten begonnen, die sich mit der Echtzeit-Adaption der Klassifikationsparameter befassen.

Literaturverzeichnis

- [1] EN 12523: Externe Gliedmaßenprothesen und externe Orthesen. 1999.
- [2] AHRENS, H.; LÄUTER, J.: *Mehrdimensionale Varianzanalyse: Hypothesenprüfung, Dimensionserniedrigung, Diskrimination bei multivariaten Beobachtungen*. Berlin: Akademie-Verlag, 1974.
- [3] AJIBOYE, A.; WEIR, R.; HECKATHORNE, C.; CHILDRESS, D.: Neurofuzzy logic as a control algorithm for an externally powered multifunctional hand prosthesis. In: *Proc., Myoelectric Controls/Powered Prosthetics Symposium*, S. 126–130, Fredericton, Canada: University of New Brunswick, 2002.
- [4] ANDERSON, E.: The irises of the gaspe peninsula. *Bulletin of the American Iris Society* 59 (1935), S. 2–5.
- [5] ANDREWS, B. J.; WARWICK, K.; GASSON, M. N.; JAMOUS, A.; HARWIN, W. S.; KYBERD, P. J.: Development of an implanted neural control interface for artificial limbs. In: *Proc., 10th World Congress of the International Society for Prosthetics and Orthotics ISPO2001*, S. TO 8.6, Glasgow, 2001.
- [6] ATKINS, D. J.; HEARD, D. C.; DONOVAN, W. H.: Epidemiologic overview of individuals with upper-limb loss and their reported research priorities. *Journal of Prosthetics and Orthotics* 8(1) (1996), S. 2–11.
- [7] BALL, T.; NAWROT, M.; PISTOHL, T.; AERTSEN, A.; SCHULZE-BONHAGE, A.; MEHRING, C.: Towards an implantable brain-machine interface based on epicortical field potentials. *Biomedizinische Technik* 49 (2) (2004), S. 756–759.
- [8] BANDEMER, H.; NÄTHER, W.: *Fuzzy data analysis*. Dordrecht: Kluwer Academic Publishers, 1992.
- [9] BARRERO, V.; GRISALES, V.; ROSAS, F.; SANCHEZ, C.; LEON, J.: Design and implementation of an intelligent interface for myoelectric controlled prosthesis. In: *Proc., 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (Cat. No.01CH37272)*, Bd. 4, S. 3352–3355, Piscataway, NJ, USA, 2001.
- [10] BARRY, D.; LEONHARD, J.; GITTER, A.; BALL, R.: Acoustic myography. *Journal of the Association of Childrens Prosthetic-Orthotic Clinics* 19 (3) (1984) No. Publication: US4571750, S. 54ff.
- [11] BARRY, D. T.; COLE, N. M.: Fluid mechanics of muscle vibrations. *Biophysical Journal* 53 (1988), S. 899–905.
- [12] BASMAJAN, J.; DELUCA, C.: *Muscles Alive*. Baltimore: The Williams and Wilkins Company, 1985.

- [13] BAUMGARTNER, R.; BOTTA, P.: *Amputation und Prothesenversorgung der oberen Extremität*, S. 161 ff. Enke Verlag, 1997.
- [14] BECK, S.: *Ein Konzept zur automatischen Lösung von Entscheidungsproblemen bei Unsicherheit mittels der Theorie der unscharfen Mengen und der Evidenztheorie*. Dissertation, Universitätsverlag, Universität Karlsruhe, 2005.
- [15] BECK, S.; REISCHL, M.; BRETTHAUER, G.: Steuerungs- und Regelungskonzepte für eine myoelektrische Handprothese. In: *37. Regelungstechnisches Kolloquium in Boppard*, S. 43, Epple, U., 2002.
- [16] BEN-BASSAT, M.: *Use of distance measures, information measures and error bounds for feature extraction*, Kap. 35, S. 773–791. North-Holland Publishing Company, 1982.
- [17] BERGER, H.: Über das Elektroencephalogramm des Menschen. *Archiv für Psychiatrie und Nervenkrankheiten* 87 (1929), S. 527–570.
- [18] BHATTACHARYYA, A.: On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of Calcutta Maths Society* 35 (1943), S. 99–110.
- [19] BIRBAUMER, N.; KÜBLER, A.; GHANAYIM, N.; HINTERBERGER, T.; PERELMOUTER, J.; KAISER, J. IVERSEN, I.; KOTCHOUBEY, B.; NEUMANN, N.; FLOR, H.: The thought translation device (TTD) for completely paralyzed patients. *IEEE Transactions on Rehabilitation Engineering* 8 (2000), S. 190–193.
- [20] BLACK, M.; SERRUYA, M.; BIENENSTOCK, E.; GAO, Y.; WU, W.; DONOGHUE, J.: Connecting brains with machines: The neural control of 2D cursor movement. In: *Proc., First International IEEE EMBS Conference on Neural Engineering*, S. 19–22, Capri Island, 2003.
- [21] BLANKERTZ, B.; CURIO, G.; MÜLLER, K.-R.: Classifying single trial EEG: Towards brain computer interfacing. In: *Advances in Neural Information Processing Systems* (DIETRICH, T. G.; BECKER, S.; GHAHRAMANI, Z., Hg.), S. 157–164, 2002.
- [22] BLANKERTZ, B.; MÜLLER, K.; CURIO, G.; VAUGHAN, T.; SCHALK, G.; WOLPAW, J.; SCHLÖGL, A.; NEUPER, C.; PFURTSCHELLER, G.; HINTERBERGER, T.; SCHRÖDER, M.; BIRBAUMER, N.: The BCI Competition 2003: Progress and perspectives in detection and discrimination of EEG single trials. *IEEE Transactions on Biomedical Engineering* 51 (6) (2004), S. 1044–1051.
- [23] BOENICK, U.: Ingenieuraufgaben in der technischen Orthopädie, dargestellt am Problem der Steuerung künstlicher Extremitäten. *VDI-Berichte* 175 (1971), S. 31–44.
- [24] BOGDAN, M.: *Signalverarbeitung biologischer Nervensignale zur Steuerung einer Prothese mit Hilfe künstlicher neuronaler Netze*. Dissertation, Universität Tübingen, Tübingen, 1998.
- [25] BOGDAN, M.: Nervengesteuerte Prothese. *Deutsches Ärzteblatt* 5 (2000), S. 36–37.
- [26] BOOSTANI, R.; MORADI, M. H.: Evaluation of the forearm EMG signal features for the control of a prosthetic hand. *Physiological Measurement* 24 (2003), S. 309–319.
- [27] BURGESS, C.: A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining* 2(2) (1998), S. 121–167.

- [28] BURMEISTER, O.; REISCHL, M.; MIKUT, R.: Zeitvariante Klassifikatoren für Brain Machine Interfaces und Neuroprothesen. In: *Proc., 15. Workshop Computational Intelligence* (MIKUT, R.; REISCHL, M., Hg.), S. 129–143, Karlsruhe: Universitätsverlag Karlsruhe, 2005.
- [29] BUTTERFASS, J.; GREBENSTEIN, M.; LIU, H.; HIRZINGER, G.: DLR-Hand II: Next generation of a dextrous robot hand. In: *Proc., 19. IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2001.
- [30] CANU, S.; GRANDVALET, Y.; RAKOTOMAMONJY, A.: SVM and kernel methods MATLAB toolbox. Perception Systèmes et Information, INSA de Rouen, Rouen, France, 2003.
- [31] CARREIRA-PERPINAN, M.: A review of dimension reduction techniques. Techn. Ber. CS-96-09, University of Sheffield, 1997.
- [32] CARROZZA, M.; SUPPO, C.; SEBASTIANI, F.; MASSA, B.; VECCHI, F.; LAZZARINI, R.; CUTKOSKY, M.; DARIO, P.: The SPRING hand: Development of a self-adaptive prosthesis for restoring natural grasping. *Journal of Autonomous Robots* 16 (2) (2003), S. 125–141.
- [33] CHAPIN, J. K.; MOXON, K. A.; MARKOWITZ, R. S.; NICOLELIS, M.: Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature Neuroscience* 2 (1999), S. 664–670.
- [34] CHARWAT, H. J.: *Lexikon der Mensch-Maschine-Kommunikation*. München, Wien: Oldenbourg Verlag, 1992.
- [35] CHILDRESS, D.: *Control of limb prostheses*, Kap. 6D, S. 175–199. Mosby-Year Book, Inc., 1992.
- [36] CHILDRESS, D.; GRAHN, E.; HECKATHORNE, C.: Direct muscle attachment: Multifunctional control of hands and arms. *DVA Progress Report 8/96* (1996).
- [37] COCHRAN, W. T.; COOLEY, J. W.; FAVIN, D. L.; HELMS, H. D.: What is the Fast Fourier Transform? *IEEE Transactions on Audio and Electroacoustics* AU-15 (1967), S. 45–55.
- [38] COVER, T.; HART, P.: Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* IT-13(1) (1967), S. 21–27.
- [39] COVER, T.; VAN CAMPENHOUT, J. M.: On the possible orderings in the measurement selection problem. *IEEE Transactions on Systems, Man and Cybernetics* 7 (9) (1977), S. 657–661.
- [40] CRISTIANNI, N.; SHAW-TAYLOR, J.: *Support vector machines*. Cambridge, United Kingdom: Cambridge University Press, 2000.
- [41] CURCIE, D. J.; FLINT, J. A.; CRAELIUS, W.: Biomimetic finger control by filtering of distributed forelimb pressures. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 9 (1) (2001), S. 69–75.
- [42] DA CUNHA, F.; SCHNEEBELI, H.; DYNNIKOV, V.: Development of anthropomorphic upper limb prostheses with human-like interphalanganian and interdigital couplings. *Artificial Organs* 24 (3) (2000), S. 193–197.
- [43] DE LUCA, C.: Physiology and mathematics of myoelectric signals. *IEEE Transactions on Biomedical Engineering* 26 (1979), S. 313–325.
- [44] DE LUCA, C.; KNAFFLITZ, M.: *Surface electromyography: What's new?* Turin: Neuromuscular Research Center, Boston University, 1992.

- [45] DILLMANN, R.; ROGALLA, O.; EHRENMANN, M.; ZÖLLNER, R.; BORDEGONI, M.: Learning robot behaviour and skills based on human demonstration and advice: The machine learning paradigm. In: *Proc., 9th International Symposium of Robotics Research*, S. 229–238, 1999.
- [46] DIMITROVA, N. A.; DIMITROV, G. V.: Interpretation of EMG changes with fatigue: Facts, pitfalls, and fallacies. *Journal of Electromyography and Kinesiology* 13 (2003), S. 13–36.
- [47] DORCAS, D. S.; SCOTT, R. N.: A three-state myo-electric control. *Medical and Biological Engineering* 4 (1966), S. 367–370.
- [48] DOUBLER, J. A.; CHILDRESS, D.: An analysis of extended physiological proprioception as a control technique. *Journal of Rehabilitation Research and Development* 21 (1) (1984), S. 5–18.
- [49] EFRON, B.; TIBSHIRANI, R.: Cross-validation and the bootstrap: Estimating the error rate of a prediction rule. Techn. Ber., Stanford-University, 1995.
- [50] EICHMEIER, J.: *Medizinische Elektronik - eine Einführung*. Berlin: Springer Verlag, 1997.
- [51] ENGLEHART, K.; HUDGINS, B.; CHAN, A.: Continuous multifunction myoelectric control using pattern recognition. *Technology and Disability* 15 (2003), S. 95–103.
- [52] ENGLEHART, K.; HUDGINS, B.; PARKER, P.: A wavelet-based continuous classification scheme for multifunctional myoelectric control. *IEEE Transactions on Biomedical Engineering* 48 (3) (2001), S. 302–311.
- [53] ERIKSSON, L.; SEBELIUS, F.; BALKENIUS, C.: Neural control of a virtual prosthesis. In: *Proc., International Conference on Artificial Neural Networks*, S. 905–910, Berlin: Springer Verlag, 1998.
- [54] FETZ, E.: Real-time control of a robotic arm by neuronal ensembles. *Nature Neuroscience* 2 (7) (1999), S. 583–584.
- [55] FINLEY, R.; WIRTA, R.: Myocoder-computer study of electromyographic patterns. *Archives of Physical Medicine and Rehabilitation* 48 (1967), S. 20–24.
- [56] FISHER, R. A.: The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7 (1936), S. 179–188.
- [57] FLINT, J. A.; PHILLIPS, S. L.; CRAELIUS, W.: Residual kinetic vectors for prosthetic control. In: *Proc., Myoelectric Controls/Powered Prosthetics Symposium*, S. 112–115, New Brunswick, 2002.
- [58] FORINA, M. ET AL: PARVUS- An extendible package for data exploration, classification, and correlation. Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno, 16147 Genoa, Italy, 1991.
- [59] FORSYTH, R. S.: Zoo database. 8 Grosvenor Avenue, Mapperley Park, Nottingham NG3 5DX, 1990.
- [60] FOX, G.; KOLLER, J.: Code generation by a generalized neural network: general principles and elementary examples. *Journal of Parallel and Distributed Computing* 6 (2) (1989), S. 388 – 410.
- [61] FRANCA, J.; ORTOLAN, R.; DA CUNHA, R.; DANNIKOV, V.; CLIQUET, A.: A specific control system for an anthropomorphic myoelectric hand prosthesis. In: *Nonlinear dynamics, chaos, control and their applications to engineering sciences* (BALTHASAR, J.; GONCALVES, P.; BRASIL, R.; CALDAS, I.; RIZATTO, F., Hg.), Kap. Applications of nonlinear phenomena, S. 356–365, 2002.

- [62] FÜRNKRANZ, J.: Separate-and-conquer rule learning. *Artificial Intelligence Review* 13 (1) (1999), S. 1–46.
- [63] GABOR, D.: Theory of communication. *Journal of the Institute of Electrical Engineers* 93 (1946), S. 429–441.
- [64] GAZEAU, J. P.; ARSICAULT, M.; LALLEMAND, J. P.: The L. M. S. mechanical hand. In: *Proc., Robots Manipulators Systems*, Paris, 1998.
- [65] GEKELER, F.; SCHWAHN, H.; STETT, A.; KOHLER, K.; ZRENNER, E.: *Subretinal microphotodiodes to replace photoreceptor-function. A review of the current state*, S. 77–95. Irvinn, 2001.
- [66] GRAY, A.; FISCHER, B.; SCHUMANN, J.; W., B.: *Advances in Neural Information Processing Systems*, Kap. Automatic derivation of statistical algorithms: The EM family and beyond. MIT Press, 2002.
- [67] GUGER, C.; HARKAM, W.; HERTNAES, C.; PFURTSCHELLER, G.: Prosthetic control by an EEG-based brain-computer interface (BCI). In: *Proc., AAATE 5th European Conference for the Advancement of Assistive Technology*, Düsseldorf (Germany), 1999.
- [68] HALL, M.: *Correlation-based feature selection for machine learning*. Dissertation, University of Waikato, Hamilton, New Zealand, 1999.
- [69] HÄMMERLE, H.; HÖFFLINGER, B.; SCHUBERT, M.; ZRENNER, E.: *Mikrophotodioden-Arrays als Retina-Implantate zum Ersatz der Netzhautphotorezeptoren*, Kap. Entwicklungsperspektiven für die Biomedizinische Technik aus der Sicht der Universitäten 1997, S. 27–33. 1997.
- [70] HASTIE, T.; TIBSHIRANI, R.: Classification by pairwise coupling. In: *Advances in Neural Information Processing Systems* (JORDAN, M.; KEARNS, M.; SOLLA, S., Hg.), Bd. 10, The MIT Press, 1998.
- [71] HAYKIN, S.: *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ: Prentice Hall, 1994.
- [72] HERBERTS, P.; ALMSTROEM, C.; CAINE, K.: Clinical application study of multifunctional prosthetic hands. *Journal of Bone and Joint Surgery* 60-B (4) (1978), S. 552–560.
- [73] HUANG, H.; CHIANG, C.: DSP-based controller for a multi-degree prosthetic hand. In: *Proc. IEEE International Conference on Robotics and Automation*, S. 1378–1383, San Francisco, 2000.
- [74] HUBERT, M.; VAN DRIESSEN, K.: Fast and robust discriminant analysis. *Computational Statistics and Data Analysis* 45 (2) (2002), S. 301–320.
- [75] HUDGINS, B.; PARKER, P.; SCOTT, R.: A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering* 40 (1993), S. 82–94.
- [76] HUMPHREY, D. R.; SCHMIDT, E. M.; THOMPSON, W. D.: Predicting measures of motor performances from multiple cortical spike trains. *Science* 170 (1970), S. 758–762.
- [77] JAIN, A.; DUIN, R.; MAO, J.: Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (1) (2000), S. 4–37.
- [78] JÄKEL, J.; MIKUT, R.; BRETTHAUER, G.: Fuzzy control systems, in controls systems, robotics, and automation. In: *Encyclopedia of Life Support Systems (EOLSS)* (UNBEHAUEN, H., Hg.), Oxford: Eolss Publishers, 2003.

- [79] KAMPAS, P.: Myoelektroden - optimal eingesetzt. *Medizinisch Orthopädische Technik* 1 (2001), S. 21–27.
- [80] KENNEY, L.; LISITSA, I.; BOWKER, P.; HEATH, G.; HOWARD, D.: Dimensional change in muscle as a control signal for powered upper limb prostheses: A pilot study. *Medical Engineering and Physics* 21 (1999), S. 589–597.
- [81] KIENDL, H.: Fuzzy Control. *Automatisierungstechnik* 41 (1993) 1, S. A1–A8.
- [82] KITZENMAIER, P.: *Entwicklung verbesserter Verfahren für die Ableitung und Verarbeitung myoelektrischer Signale zur Steuerung von Rehabilitationshilfen*. Berlin: Fachverlag Schiele u. Schön, 1995.
- [83] KITZENMAIER, P.; BOENICK, U.: Möglichkeiten der myoelektrischen Ansteuerung von Gliedmaßenprothesen. *Biomedizinische Technik* 37 (1992), S. 170–180.
- [84] KLINKE, R.; SILBERNAGL, S.: *Lehrbuch der Physiologie*. Stuttgart: Georg Thieme Verlag, 2001.
- [85] KOHAVI, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proc., International Joint Conference on Artificial Intelligence*, 1995.
- [86] KOHAVI, R.; JOHN, G.: Wrappers for feature subset selection. *Artificial Intelligence* 97 (1997), S. 273–324.
- [87] KRABS, M.; KIENDL, H.: Automatische Generierung von Fuzzy-Regeln mit dem ROSA-Verfahren. *VDI-Berichte* 1113 (1994), S. 29–39.
- [88] KRAUT, D.: *Entwicklung einer Testumgebung für mikrocontroller-basierte Steuerungen von myoelektrischen Handprothesen*. Diplomarbeit, Berufsakademie Karlsruhe, Forschungszentrum Karlsruhe, 2003.
- [89] KRAUT, D.: *Portierung und Validierung biosignalverarbeitender Algorithmen für Funktionsprothesen*. Studienarbeit, Fachhochschule Mannheim, 2004.
- [90] KREPKE, R.; BLANKERTZ, B.; CURIO, G.; MÜLLER, K.-R.: The Berlin Brain-Computer Interface (BBCI): Towards a new communication channel for online control of multimedia applications and computer games. In: *Proc., 9th International Conference on Distributed Multimedia Systems*, S. 237–244, 2003.
- [91] KULLBACK, S.; LEIBLER, R. A.: Information and sufficiency. *Annals of Mathematics and Statistics* 22 (1951), S. 79–86.
- [92] KYBERD, P.; LIGHT, C.; CHAPPELL, P.; NIGHTINGALE, J.; WHATLEY, D.; EVANS, M.: The design of anthropomorphic prosthetic hands: A study of the Southampton hand. *Robotica* 19 (2001), S. 593–600.
- [93] LAMOUNIER, E.; SOARES, A.; ANDRADE, A.; CARRIJO, R.: A virtual prosthesis control based on neural networks for EMG pattern classification. In: *Proc., 6th IASTED International Conference: Artificial Intelligence and Soft-Computing*, Bauff, Canada, 2002.
- [94] LEHMANN, A.; MIKUT, R.; ASFOUR, T.; OSSWALD, D.; BRETTHAUER, G.; DILLMANN, R.: Task coordination and task supervision for compliant anthropomorphic robot grippers. *Submitted to Automatisierungstechnik* (2006).

- [95] LEHMANN, A.; MIKUT, R.; MARTIN, J.; BRETTHAUER, G.: Online-Stabilitätsüberwachung strukturvariabler Roboterregelungen. In: *Robotik 2004*, S. 55–62, Düsseldorf: VDI Verlag, 2004.
- [96] LEHMANN, T.; MEYER ZU BEXTEN, E.: *Handbuch der medizinischen Informatik*. München: Hanser Verlag, 2002.
- [97] LINDSTRÖM, L. R.: *On the frequency spectrum of EMG signals*. Göteborg: Chalmers University of Technology, 1970.
- [98] LIPOVEI, M.: *Evaluation of biometric signal characteristics for movement classification*. Forschungszentrum Karlsruhe, 2004.
- [99] LOOSE, T.: *Konzept für eine modellgestützte Diagnostik mittels Data Mining am Beispiel der Bewegungsanalyse*. Dissertation, Universitätsverlag, Universität Karlsruhe, 2004.
- [100] LOUIS, A. K.; MAASS, P.; RIEDER, A.: *Wavelets - Theorie und Anwendungen*. Stuttgart: Teubner Verlag, 1994.
- [101] LOVCHIK, C. S.; DIFTLER, M. A.: The Robonaut hand: A dexterous robot hand for space. In: *Proc., IEEE International Conference on Robotics and Automation*, S. 907–913, Detroit, 1999.
- [102] MALLAT, S.: *A wavelet tour of signal processing*. San Diego: Academic Press, 1999.
- [103] MARTIN, J.: *Ein Beitrag zur Integration von Sensoren in eine anthropomorphe künstliche Hand mit flexiblen Fluidaktoren*. Dissertation, Universitätsverlag, Universität Karlsruhe, 2004.
- [104] MASON, S.; BIRCH, G.: A general framework for brain-computer interface design. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11 (1) (2003), S. 70–85.
- [105] MASULLI, F.; VALENTINI, G.: Comparing decomposition methods for classification. In: *Proc., 4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies* (HOWLETT, R.; JAIN, L., Hg.), S. 788–791, Piscataway, NJ, 2000.
- [106] MATHWORKS, INC.: Neural Network Toolbox, Version 3.0.1. 1998.
- [107] MAYORAZ, E.; MOREIRA, M.: On the decomposition of polychotomies into dichotomies. In: *Proc., 14th International Conference on Machine Learning*, S. 291–226, Nashville, TN: Morgan-Kaufmann, 1997.
- [108] MCDERMOTT, H. J.; MCKAY, C. M.; VANDALI, A. E.: A new portable sound processor for the University of Melbourne / Nucleus Limited multichannel cochlear implant. *Journal of the Acoustic Society of America* 91 (1992), S. 3367–3371.
- [109] MICERA, S.; SABATINI, A.; DARIO, P.; ROSSI, B.: A hybrid approach to EMG pattern analysis for classification of arm movements using statistical and fuzzy techniques. *Medical Engineering and Physics* 21 (1999), S. 303–311.
- [110] MICHIE, D.; SPIEGELHALTER, D. J.; TAYLOR, C. C.: *Machine Learning, Neural and Statistical Classification*. London: Ellis Horwood, 1994.
- [111] MIKUT, R.: *Modellgestützte on-line Stabilitätsüberwachung komplexer Systeme auf der Basis unscharfer Ljapunov-Funktionen*. Dissertation, Universität Karlsruhe, VDI-Verlag, 1999.
- [112] MIKUT, R.; JÄKEL, J.; GRÖLL, L.: Interpretability issues in data-based learning of fuzzy systems. *Fuzzy Sets and Systems* 150 (2005), S. 179–197.

- [113] MIKUT, R.; PETER, N.; MALBERG, H.; JÄKEL, J.; GRÖLL, L.; BRETTHAUER, G.; ABEL, R.; DÖDERLEIN, L.; RUPP, R.; SCHABLOWSKI, M.; GERNER, H.: Diagnoseunterstützung für die instrumentelle Ganganalyse (Projekt GANDI). *Bericht des Forschungszentrums Karlsruhe FZKA 6613* (2001).
- [114] MILEWSKI, K.: *Verarbeitung myoelektrischer Signale mittels künstlicher neuronaler Netze als Schnittstelle zwischen Mensch und Maschine*. Studienarbeit, Forschungszentrum Karlsruhe, 2003.
- [115] MOUNIER, S.: *Entwicklung einer realitätsnahen Krafrückkopplung bei fluidisch betriebenen Handprothesen*. Dissertation, Forschungszentrum Karlsruhe, FZKA 7004, Universität Karlsruhe, 2004.
- [116] MÜLLER, G.; NEUPER, C.; RUPP, R.; KEINRATH, C.; GERNER, H.; PFURTSCHELLER, G.: Event-related beta EEG changes during wrist movements induced by functional electrical stimulation of forearm muscles in man. *Neuroscience Letters* 340(2) (2003), S. 143–147.
- [117] NÄDER, M.: Erfahrungen und Beobachtungen mit Myobock I. Entwicklungsstand des Otto Bock - Myostat Systems. *Medizinische Technik* (1970), S. 393–399.
- [118] NICOLELIS, M.: Actions from thoughts. *Nature* 409 (2001), S. 403–407.
- [119] NICOLELIS, M.; RIBEIRO, S.: Multielectrode recordings: The next steps. *Current Opinion in Neurobiology* 12 (2002), S. 602–606.
- [120] NIEDERHOLZ, J.: *Anwendungen der Wavelet-Transformation in Übertragungssystemen*. Dissertation, Universität Duisburg, 1998.
- [121] NIEMANN, H.: *Klassifikation von Mustern*. Berlin: Springer Verlag, 1983.
- [122] NISHIKAWA, D.: *Studies on electromyogram to motion classifier*. Hokkaido University, 2001.
- [123] NISHIKAWA, D.; YU, W.; YOKOI, H.; KAKAZU, Y.: EMG prosthetic hand controller using real-time learning method. In: *Proc., IEEE International Conference on Systems, Man and Cybernetics*, Tokyo, 1999.
- [124] N.N.: *Discriminant Analysis and Clustering*. Washington, D.C.: National Academy Press, 1988.
- [125] N.N.: Technische Daten Otto Bock Hand. Techn. Ber., Otto Bock, 2000.
- [126] N.N.: <http://bmt1.kf.tu-berlin.de/fgbmt/bmt.htm>. 2003.
- [127] N.N.: *Leitlinie Unterarmprothesen*. Bundesinnungsverband für Orthopädie-Technik, 2003.
- [128] N.N.: <http://ida.first.fraunhofer.de/projects/bci/competition>. 2004.
- [129] N.N.: <http://www.hosmer.com>. 2004.
- [130] N.N.: <http://www.rslsteeper.com>. 2004.
- [131] N.N.: <http://www.utaharm.com>. 2004.
- [132] N.N.: <http://www.cochlearamericas.com>. 2005.
- [133] NORLIN, P.; KINDLUNDH, M.; MOURoux, A.; YOSHIDA, K.; JENSEN, W.; HOFMANN, U.: A 32-site neural recording probe fabricated by double-sided deep reactive ion etching of silicon-on-insulator substrates. *Journal of Micromechanics and Microengineering* 12 (4) (2001), S. 414–419.

- [134] OBERMAIER, B.; MÜLLER, G.; PFURTSCHELLER, G.: 'Virtual Keyboard' controlled by spontaneous EEG activity. In: *Proc., International Conference on Artificial Neural Networks*, S. 636–641, Vienna (Austria), 2001.
- [135] OTTO BOCK, ORTHOPÄDISCHE INDUSTRIE: *Patienteninformationen für SensorHand*. 1999.
- [136] OUAMER, M.; BOITEUX, M.; PETITJEAN, M.; TRAVENS, L.; SALÈS, A.: Acoustic myography during voluntary isometric contraction reveals non-propagative lateral vibration. *Journal of Biomechanics* 32 (12) (1999), S. 1279–1285.
- [137] PAULIGNAN, Y.; FRAK, V. G.; TONI, I.; JEANNEROD, M.: Influence of object position and size on human prehension movements. *Experimental Brain Research* 114 (2) (1997), S. 226–234.
- [138] PELEG, D.; BRAIMAN, E.; YOM-TOV, E.; INBAR, G.: Classification of finger activation for use in a robotic prosthesis arm. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 10(4) (2002), S. 290–293.
- [139] PERLMUTTER, S.; MAIER, M.; FETZ, E.: Activity and output linkages of spinal premotor interneurons during voluntary wrist movements in the monkey. *Journal of Neurophysiology* 80 (1998), S. 2475–2494.
- [140] PETROFSKY, J. S.: Frequency and amplitude analysis of the EMG during exercise on the bicycle ergometer. *European Journal of Applied Physiology* 41(1) (1979), S. 1–15.
- [141] PFURTSCHELLER, G.; GUGER, C.; MÜLLER, G.; KRAUSZ, G.; NEUPER, C.: Brain oscillations control hand orthosis in a tetraplegic. *Neuroscience Letters* 292 (2000), S. 211–214.
- [142] PLATT, J.: Sequential minimal optimization: A fast algorithm for training support vector machines. Techn. Ber. MSR-TR-98-14, Microsoft, 1998.
- [143] POPOVIC, M.; KELLER, T.; PAPPAS, I.; DIETZ, V.; MORARI, M.: Surface-stimulation technology for grasping and walking neuroprostheses. *IEEE Engineering in Medicine and Biology Magazine* 20 (1) (2001), S. 82–93.
- [144] PUCHHAMMER, G.: Der taktile Rutschsensor: Integration miniaturisierter Sensorik in einer Myo-Hand. *Orthopädie-Technik* 7 (1999), S. 564–569.
- [145] PYLATIUK, C.; SCHULZ, S.: Using the internet for an anonymous survey of myoelectrical prosthesis wearers. In: *Proc., Myoelectric Controls/Powered Prosthetics Symposium*, S. 255–257, Fredericton, Canada: University of New Brunswick, 2005.
- [146] QUINLAN, J. R.: Induction of Decision Trees. *Machine Learning* 1 (1986), S. 81–106.
- [147] RAMOSER, H.; MÜLLER-GERKING, J.; PFURTSCHELLER, G.: Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transaction on Rehabilitation Engineering* 8 (2000), S. 441–446.
- [148] RAO, C. R.: The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society, Series B* 10 (1948), S. 159–203.
- [149] REGALDO, A.: Bewusstsein für Maschinen. *VDI-Nachrichten* 8 (2001).
- [150] REISCHL, M.: *Signalaufbereitung und -klassifikation von biometrischen Patientendaten mit Mikrosensoren in der Rehabilitation*. Diplomarbeit, Universität Karlsruhe, 2001.

- [151] REISCHL, M.: Steuerungskonzept - Prothese. Techn. Ber., Forschungszentrum Karlsruhe, 2002.
- [152] REISCHL, M.; BECK, S.; MIKUT, R.; LEHMANN, A.; MARTIN, J.; SCHULZ, S.: Regelungskonzepte für fluidisch betriebene Handprothesen. In: *Proc., 4. Workshop "Automatisierungstechnische Methoden und Systeme für die Medizin"* (VOGES, U.; BRETTHAUER, G., Hg.), S. 16–17, 2003.
- [153] REISCHL, M.; BURMEISTER, O.; MIKUT, R.: Robust design of man machine interfaces for time-variant biosignals. *Biomedizinische Technik* 50 (E1/1) (2005), S. 774–775.
- [154] REISCHL, M.; GRÖLL, L.; MIKUT, R.: Optimierte Klassifikation für Mehrklassenprobleme am Beispiel der Bewegungssteuerung von Handprothesen. In: *Proc., 13. Workshop Fuzzy Systeme* (MIKUT, R.; REISCHL, M., Hg.), S. 124–143, Forschungszentrum Karlsruhe, 2003.
- [155] REISCHL, M.; GRÖLL, L.; MIKUT, R.: Optimized classification of multiclass problems applied to EMG-control of hand prostheses. In: *Proc., IEEE International Joint Conference on Neural Networks*, Budapest, 2004.
- [156] REISCHL, M.; KRAUT, D.: Interner Bericht: Online-Berechnung von Merkmalen für patientenindividuelle Man-Machine-Interfaces. 2004.
- [157] REISCHL, M.; MIKUT, R.: Validierung hierarchischer Klassifikatoren für Mehrklassenprobleme. In: *Proc., 14. Workshop Fuzzy-Systeme und Computational Intelligence*, S. 175–188, Universitätsverlag Karlsruhe, 2004.
- [158] REISCHL, M.; MIKUT, R.; PYLATIUK, C.; SCHULZ, S.: Control strategies for hand prostheses using myoelectric patterns. In: *Proc., 9th Zittau Fuzzy Colloquium*, S. 168–174, 2001.
- [159] REISCHL, M.; MIKUT, R.; PYLATIUK, C.; SCHULZ, S.: Erkennung von Bewegungsabsichten für myoelektrisch angesteuerte Handprothesen. In: *Proc., 11. Workshop Fuzzy Control des GMA-FA 5.22* (MIKUT, R.; REISCHL, M., Hg.), S. 106–119, Karlsruhe: Forschungszentrum Karlsruhe, 2001.
- [160] REISCHL, M.; MIKUT, R.; PYLATIUK, C.; SCHULZ, S.: Control and signal processing concepts for a multifunctional hand prosthesis. In: *Proc., Myoelectric Controls/Powered Prosthetics Symposium*, S. 116–119, Fredericton, Canada: University of New Brunswick, 2002.
- [161] REISCHL, M.; MIKUT, R.; PYLATIUK, C.; SCHULZ, S.: Erweiterung der Ansteuerungsmöglichkeiten myoelektrischer Handprothesen. *Biomedizinische Technik* 47 (E2) (2002), S. 868–870.
- [162] REISCHL, M.; MIKUT, R.; PYLATIUK, C.; SCHULZ, S.; BECK, S.; BRETTHAUER, G.: Steuerungs- und Signalverarbeitungskonzepte für eine multifunktionale Handprothese. *Automatisierungstechnik (at)* 50 (6) (2002), S. 279–286.
- [163] REISCHL, M.; MIKUT, R.; SCHLÖGL, A.: Comparison of fuzzy and statistical classifiers for brain-computer interfaces. *Biomedizinische Technik* 49 (E2) (2004), S. 762–763.
- [164] REISCHL, M.; PYLATIUK, C.; MIKUT, R.: Individual control concepts for multifunctional hand prostheses. *Orthopädie-Technik Quarterly* 1 (2005), S. 12–14.
- [165] REISCHL, M.; VOELKEL, B.; GRÖLL, L.; MIKUT, R.: An adaptive control scheme for multifunctional arm prostheses. *Submitted to IEEE Transactions on Neural Systems and Rehabilitation Engineering* (2006).
- [166] REITER, R.: Eine neue Elektrokunsthand. *Grenzgebiete der Medizin* 4 (1948), S. 133ff.

- [167] RITTER, H.; MARTINETZ, T.; SCHULTEN, K.: *Neuronale Netze*. München: Oldenbourg, 1994.
- [168] ROJAS, R.: *Neural networks: A systematic introduction*. Berlin: Springer Verlag, 1996.
- [169] ROSENBLATT, F.: *Principles of neurodynamics*. New York: Spartan, 1962.
- [170] ROSSDEUTSCHER, W.: Steuerungsmöglichkeiten in der Armprothetik. *Orthopädie-Technik* 10(51) (2000), S. 865–868.
- [171] RUPP, R.; FROMM, B.; GERNER, H. J.: Freehand - Moderne Aspekte der funktionellen Elektrostimulation (FES) bei Tetraplegikern. *Medizinisch Orthopädische Technik* 6 (2000), S. 190–194.
- [172] RUPP, R.; MÜLLER, G.; PFURTSCHELLER, G.; GERNER, H.: Gedankensteuerung der Greif-funktion eines Tetraplegikers: I. Funktionelle Elektrostimulation (FES). In: *Proc., Jahrestagung der deutschen, österreichischen u. schweizerischen Gesellschaft für biomedizinische Technik*, S. 300–301, Salzburg: Biomedizinische Technik (48), Ergänzungsband 1, 2003.
- [173] RUTTEN, W.; MOUVEROUX, J. M.; BUITENWEG, J.; HEIDA, C.; RUARDIJ, T.; MARANI, E.; LAKKE, E.: Neuroelectronic interfacing with cultured multielectrode arrays toward a cultured probe. *Proceedings of the IEEE* 89 (7) (2001), S. 1013–1029.
- [174] SAUERBRUCH, F.: *Die willkürlich bewegbare Hand, Band II*. Berlin: Springer Verlag, 1923.
- [175] SCHERER, R.; MÜLLER, G.; NEUPER, C.; GRAIMANN, B.; PFURTSCHELLER, G.: An asynchronously controlled EEG-based virtual keyboard: improvement of the spelling rate. *IEEE Transactions on Biomedical Engineering* 51 (6) (2004), S. 979–984.
- [176] SCHLITGEN, R.; STREITBERG, B.: *Zeitreihenanalyse*. München: Oldenbourg Wissenschaftsverlag GmbH, 2001.
- [177] SCHLÖGL, A.: Outcome of the BCI-competition 2003 on the Graz data set. In: <http://ida.first.fraunhofer.de/projects/bci/competition/>, 2003.
- [178] SCHLÖGL, A.: The BIOSIG project. <http://biosig.sf.net/>, 2003 - 2004.
- [179] SCHMIDT, F.; SCHAIBLE, H.: *Neuro- und Sinnesphysiologie*. Berlin: Springer Verlag, 2001.
- [180] SCHMIDT, R.; THEWS, G.: *Physiologie des Menschen*. Berlin: Springer Verlag, 1993.
- [181] SCHMIDT, R.; THEWS, G.; LANG, F.: *Physiologie des Menschen*. Berlin: Springer Verlag, 2000.
- [182] SCHÖLKOPF, B.; SMOLA, A.; MÜLLER, K.: Nonlinear component analysis using a kernel eigenvalue problem. *Techn. Ber.* 44, Max-Planck-Institut für biologische Kybernetik, 1996.
- [183] SCHULZ, S.: *Eine neue Adaptiv-Hand-Prothese auf der Basis flexibler Fluidaktoren*. Dissertation, Shaker-Verlag, Universität Karlsruhe, 2004.
- [184] SCHULZ, S.; PYLATIUK, C.; BRETTHAUER, G.: A new class of flexible fluidic actors and their applications in medical engineering. *Automatisierungstechnik-at* 47 (1999), S. 390–395.
- [185] SCHULZ, S.; PYLATIUK, C.; REISCHL, M.; MARTIN, J.; MIKUT, R.; BRETTHAUER, G.: A lightweight multifunctional prosthetic hand. *Robotica* 23(3) (2005), S. 293–299.
- [186] SCHÜRMAN, J.: *Pattern classification*. New York: John Wiley & Sons, 1996.

- [187] SILBERNAGL, S.; DESPOPOULOS, A.: *Taschenatlas der Physiologie*. Georg Thieme Verlag, 1991.
- [188] SILCOX, D. H.; ROOKS, M. D.; VOGEL, R.; FLEMING, L.: Myoelectric prostheses. *The Journal of Joint and Bone Surgery* 75A (12) (1993), S. 1781–1791.
- [189] SINKJAER, T.; HAUGLAND, M.; INMANN, A.; HANSEN, M.; NIELSEN, K.: Biopotentials as command and feedback signals in functional electrical stimulation systems. *Medical Engineering and Physics* 25 (2003), S. 29–40.
- [190] SOLLERMANN, C.: *The use of eight main hand grips in activities of daily living*, Kap. Handens greppfunktion. Göteborg, 1980.
- [191] SOLLMANN, W.; LASZIG, R.; MARANGOS, N.; CHARACHON, R.; RAMSDEN, R.; FRAYSSE, B.; LAZORTHES, Y.; STERKERS, O.; REY, A.; LENARZ, T.: Electrical stimulation of the cochlear nucleus. First results of the European auditory brainstem implant clinical trial. *Clinical Neurology and Neurosurgery* 99 (1) (1997), S. 109ff.
- [192] STIEGLITZ, T.: Flexible biomedical microdevices with double-sided electrode arrangements for neural applications. *Sensors and Actuators A* 90 (2001), S. 203–211.
- [193] STIEGLITZ, T.: Implantable microsystems for monitoring and neural rehabilitation. *Medical Device Technology* 1 (2002), S. 2–4.
- [194] STIEGLITZ, T.; BEUTEL, H.; BLAU, C.; MEYER, J.-U.: Flexible Multikanal-Mikroelektroden mit integrierten Zuleitungen zum Einsatz in der Neuroprothetik. *Biomedizinische Technik, Ergänzungsband* 42 (1997), S. 449–450.
- [195] TATSUOKA, M. M.: *Multivariate Analysis*. New York: Macmillan, 1988.
- [196] TAYLOR, D. M.; TILLERY, S. I.; SCHWARTZ, A. B.: Direct cortical control of 3D neuroprosthetic devices. *Science* 296 (2002), S. 1829–1832.
- [197] TORKKOLA, K.: Learning discriminative feature transforms to low dimensions in low dimensions. In: *Proc., Neural Information Processing Systems (NIPS) Conference*, Vancouver, BC, Canada: MIT Press, 2001.
- [198] TORTOPIDIS, D.; LYONS, M. F.; BAXENDALE, R. H.: Acoustic myography, electromyography and bite force in the masseter muscle. *Journal of Oral Rehabilitation* 25(12) (1998), S. 940–945.
- [199] VAPNIK, V.: *The nature of statistical learning theory*. Berlin: Springer Verlag, 1995.
- [200] VAPNIK, V.: *Statistical learning theory*. New York: John Wiley & Sons, 1998.
- [201] VOELKEL, B.: Entwicklung einer Softwareumgebung zur Analyse eines Man-Machine-Interfaces am Beispiel einer steuerbaren Handprothese. Techn. Ber., Forschungszentrum Karlsruhe, 2003.
- [202] VOELKEL, B.: *Auswertung von Patientendaten zur Evaluierung von EMG-Steuerungsstrategien*. Diplomarbeit, Forschungszentrum Karlsruhe, 2004.
- [203] VOSSIUS, G.: Principles of functional stimulation. *Zeitschrift für Orthopädie und ihre Grenzgebiete* 125 (6) (1987), S. 605–609.
- [204] WALENDA, C.: Praktikumsbericht zum 2. praktischen Studiensemester. Techn. Ber., Forschungszentrum Karlsruhe, 2003.

- [205] WARWICK, K.; GASSON, M.; HUTT, B.; GOODHEW, I.; KYBERD, P.; ANREWS, B.; TEDDY, P.; SHAD, A.: The application of implant technology for cybernetic systems. *Archives in Neurology* 60 (2003), S. 1369–1373.
- [206] WEHR, M.; WEINMANN, M.: *Die Hand - Werkzeug des Geistes*. Berlin: Spektrum Akademischer Verlag GmbH, 1999.
- [207] WEIR, R.: *Design of artificial arms and hands for prosthetic applications*, Kap. 32, S. 32.1–32.61. McGraw-Hill, 2002.
- [208] WILMS, E.: Die Technik der Vaduzer Prehensor. *Orthopädie Technik* 2 (1951), S. 7–9.
- [209] WOLPAW, J.; MCFARLAND, D.; VAUGHAN, T.; SCHALK, G.: The Wadsworth Center Brain-Computer Interface (BCI) research and development program. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11 (2003), S. 204–207.
- [210] WOOD, C.; BARRY, D.; ALTER, B.; AMANN, M.; MINOR, L.: Myoacoustic control of upper limb prostheses. *Journal of the Association of Childrens Prosthetic-Orthotic Clinics* 24 (1989), S. 2–3.
- [211] XU, L.; KRZYZAK, A.; SUEN, C.: Methods for combining multiple classifiers and their applications in handwritten character recognition. *IEEE Transactions on Systems, Man and Cybernetics* 22 (1992), S. 418–435.
- [212] YOSHIDA, K.; JENSEN, W.; NORLIN, P.; KINDLUNDH, M.; HOFMANN, U. G.: Characterization of silicon microelectrodes from the EU VSAMUEL project. *Biomedizinische Technik* 46 (1) (2001), S. 446–447.
- [213] ZADEH, L. A.: Fuzzy Sets. *Information and Control* 8 (1965), S. 338–353.
- [214] ZECCA, M.; MICERA, S.; CARROZZA, M.; DARIO, P.: On the control of multifunctional prosthetic hands by processing the electromyographic signal. *Critical Reviews in Biomedical Engineering* 30 (4-6) (2002), S. 459–485.
- [215] ZELL, R.: *Simulation neuronaler Netze*. München: Oldenbourg Verlag, 1994.

7 Anhang

7.1 Methodik

7.1.1 Vorgehensweise zum Entwurf des Benchmarkdatensatzes

Zur Modellierung des j -ten Aktivitätssignals eines Schaltsignals (z. B. j -ter Sensor) im Rahmen eines Benchmarks wird vorab von einer festen Zuordnung von Amplituden $a_{j,i}$ zu $n_b = 11$ vorgegebenen Zeitpunkten $t_{j,i}, i = 1, \dots, n_b$ ausgegangen. Die maximal auftretende Amplitude wird mit m_a bezeichnet, die Gesamtdauer des Schaltsignals mit T . Der Parameter $a_{j\max,i} \in [0, 1]$ bestimmt das zum Zeitpunkt $t_{j,i}$ maximal zulässige $a_{j,i}$ bezogen auf die Maximalamplitude m_a . Die so erhaltenen Zeit/Amplituden-Wertepaare werden mit Zufallsgrößen δ_{ji} ((0,1)-normalverteilt) für den Zeitwert und $\gamma_{ji} \in [0, 1]$ (gleichverteilt) für den Amplitudenwert beaufschlagt. Die frei wählbaren Parameter $v_{jk,i} \geq 0$ und $v_{ja,i} \geq 0$ spezifizieren die Ausprägung der Zufallsgrößen. Für $v_{jk,i} = 0$ existiert keine Zeitstreuung, für $v_{ja,i} = 0$ existiert keine Amplitudenstreuung. Die Werte für $a_{j,i}$ und $t_{j,i}$ ergeben sich nun zu

$$a_{j,i} = a_{j\max,i} m_a (1 - v_{ja,i} \gamma_{ji}), \quad (7.1)$$

$$t_{j,i} = (i - 1) \frac{T}{n_b - 1} + \frac{T}{n_b - 1} v_{jk,i} \delta_{ji}. \quad (7.2)$$

Die erhaltenen Amplituden-Zeit-Werte werden linear interpoliert (Bild 7.1). Es ergibt sich

$$u_{\text{tmp}}(t_h) = \frac{t_h - t_{j,i}}{t_{j,(i+1)} - t_{j,i}} \cdot (a_{j,(i+1)} - a_{j,i}) + a_{j,i} \quad \text{für } t_h \in [t_{j,i}, t_{j,(i+1)}]. \quad (7.3)$$

Es folgt eine Diskretisierung:

$$u_{\text{disk}}[k] = u_{\text{tmp}}(T_a \cdot k) \quad \text{mit } T_a = 0.01s. \quad (7.4)$$

Die Addition einer gleichgerichteten, (0, 0.2)-normalverteilten Zufallsgröße $\varrho[k]$ und anschließende Tiefpassfilterung (IIR-Filter, $a_{\text{bench}} = 0.9$) führt zum modellierten Aktivitätssignal $u[k]$:

$$u[k] = (1 - a_{\text{bench}})(u_{\text{disk}}[k] + |\varrho[k]|) + a_{\text{bench}}u[k - 1]. \quad (7.5)$$

Wird mehr als ein Aktivitätssignal zur Modellierung des Schaltsignals verwendet, besteht die Möglichkeit, die Zeitpunkte $k_{j,i}$ der Aktivitätssignale zu synchronisieren. Dann gilt

$$k_{j,i} = k_{1,i}. \quad (7.6)$$

Für die Datensätze A und B wird $T = 500$ und $m_a = 2.9$ gesetzt. Die Parametrierungen sind den Tabellen 7.1-7.8 zu entnehmen, eine Visualisierung der entworfenen Signale zeigt Bild 7.2.

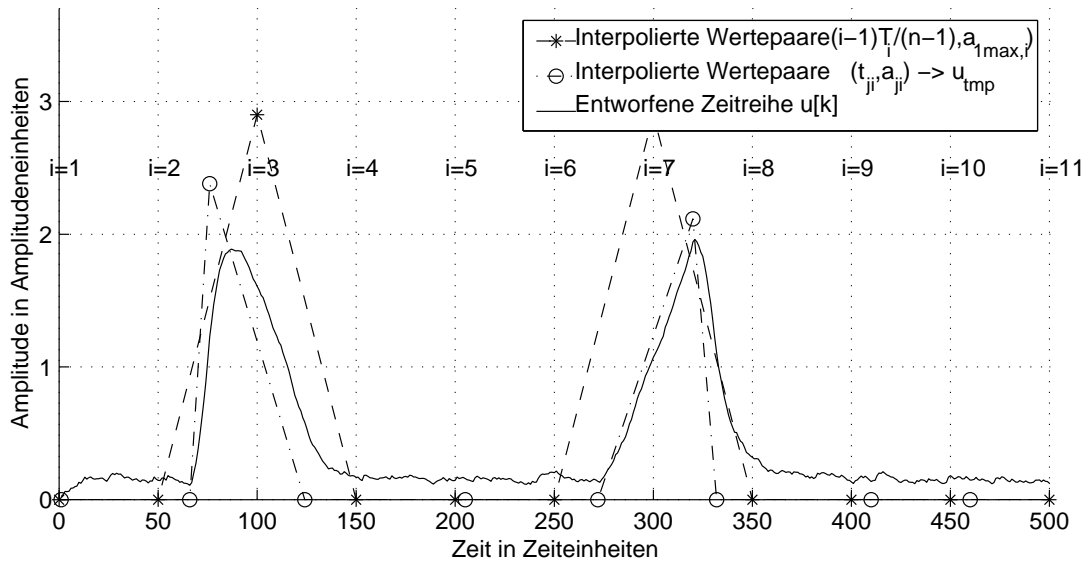


Bild 7.1: Erstellung eines künstlichen Schaltsignals

i	1	2	3	4	5	6	7	8	9	10	11
$a_{1max,i}$	0	0	1	0	0	0	1	0	0	0	0
$v_{1a,i}$	0	0	0.3	0	0	0	0.3	0	0	0	0
$v_{1k,i}$	0	0.6	0.6	0.6	0	0.6	0.6	0.6	0	0	0
$a_{2max,i}$	0	0	0.3	0	0	0	0.3	0	0	0	0
$v_{2a,i}$	0	0	0.3	0	0	0	0.3	0	0	0	0

Tabelle 7.1: Parameter für Schaltsignalklasse a (Signal 1 und Signal 2 synchronisiert)

i	1	2	3	4	5	6	7	8	9	10	11
$a_{1max,i}$	0	0	0.7	0	0	0	0.9	1	0.9	0	0
$v_{1a,i}$	0	0	0.5	0	0	0	0.2	0.1	0.2	0	0
$v_{1k,i}$	0	0.6	0.6	0.6	0	0.6	0.6	0.6	0.6	0.6	0
$a_{2max,i}$	0	0	0.3	0	0	0	0.3	0.3	0.3	0	0
$v_{2a,i}$	0	0	0.5	0	0	0	0.2	0.1	0.2	0	0
$v_{2k,i}$	0	0.6	0.6	0.6	0	0.6	0.6	0.6	0.6	0.6	0

Tabelle 7.2: Parameter für Schaltsignal b (Signal 1 und Signal 2 nicht synchronisiert)

7.1.2 Robuster Entwurf durch Erweitern des Lerndatensatzes

Ausgangspunkt für das Erweitern eines Lerndatensatzes ist die Annahme, dass sich die Zeitvarianz der Schaltsignale in jeder Aufnahmesitzung als Zufallsprozess mit sitzungsabhängigen Verteilungen und Parametern darstellen lässt. Die Integration von gestörten Schaltsignalen in den Lerndatensatz dient zu deren Modellierung. Ein gestörtes Schaltsignal kann dabei entweder auf der Basis der originalen Schaltsignale modelliert oder von ihnen direkt abgeleitet werden. Im ersten Fall besteht die Möglichkeit, aus

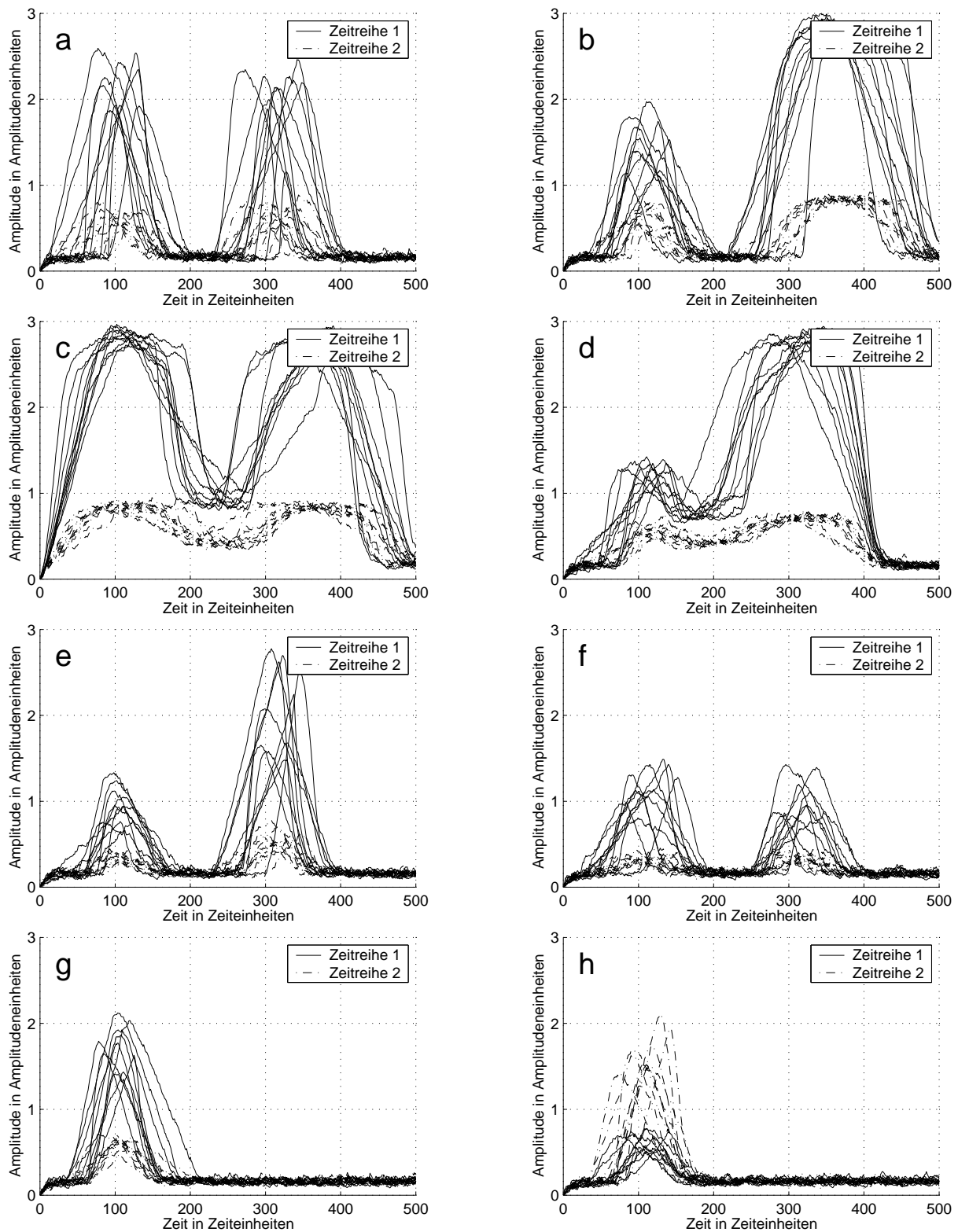


Bild 7.2: Simulierte Realisierungen von $K = 8$ Schaltsignalkategorien für Benchmarkdatensatz A

i	1	2	3	4	5	6	7	8	9	10	11
$a_{1\max,i}$	0	0.9	1	0.9	0.3	0.3	0.9	1	0.9	0	0
$v_{1a,i}$	0	0.3	0.1	0.3	0.3	0.3	0.3	0.1	0.3	0	0
$v_{1k,i}$	0	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0
$a_{2\max,i}$	0	0.3	0.3	0.3	0.1	0.1	0.3	0.3	0.3	0	0
$v_{2a,i}$	0	0.3	0.1	0.3	0.3	0.3	0.3	0.1	0.3	0	0
$v_{2k,i}$	0	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0

Tabelle 7.3: Parameter für Schaltsignal c (Signal 1 und Signal 2 nicht synchronisiert)

i	1	2	3	4	5	6	7	8	9	10	11
$a_{1\max,i}$	0	0	0.5	0.2	0.3	0.9	1	0.9	0	0	0
$v_{1a,i}$	0	0	0.3	0.3	0.3	0.3	0.1	0.3	0	0	0
$v_{1k,i}$	0	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0	0	0
$a_{2\max,i}$	0	0	0.2	0.1	0.1	0.2	0.2	0.2	0	0	0
$v_{2a,i}$	0	0	0.3	0.3	0.3	0.3	0.1	0.3	0	0	0

Tabelle 7.4: Parameter für Schaltsignal d (Signal 1 und Signal 2 synchronisiert)

i	1	2	3	4	5	6	7	8	9	10	11
$a_{1\max,i}$	0	0	0.5	0	0	0	1	0	0	0	0
$v_{1a,i}$	0	0	0.6	0	0	0	0.6	0	0	0	0
$v_{1k,i}$	0	0.6	0.6	0.6	0	0.6	0.6	0.6	0	0	0
$a_{2\max,i}$	0	0	0.1	0	0	0	0.3	0	0	0	0
$v_{2a,i}$	0	0	0.6	0	0	0	0.6	0	0	0	0

Tabelle 7.5: Parameter für Schaltsignal e (Signal 1 und Signal 2 synchronisiert)

i	1	2	3	4	5	6	7	8	9	10	11
$a_{1\max,i}$	0	0	0.5	0	0	0	0.5	0	0	0	0
$v_{1a,i}$	0	0	0.5	0	0	0	0.5	0	0	0	0
$v_{1k,i}$	0	0.6	0.6	0.6	0	0.6	0.6	0.6	0	0	0
$a_{2\max,i}$	0	0	0.1	0	0	0	0.1	0	0	0	0
$v_{2a,i}$	0	0	0.5	0	0	0	0.5	0	0	0	0

Tabelle 7.6: Parameter für Schaltsignal f (Signal 1 und Signal 2 synchronisiert)

dem Schaltsignal die Modellierungsparameter gemäß dem Benchmark zu extrahieren, zu stören und das Schaltsignal neu zu erstellen oder bei der Entstehung des Benchmarks die Parameter direkt zu stören. Für den Benchmarkdatensatz scheint dies praktikabel, für einen realen Datensatz ist allerdings Voraussetzung, dass das zugrundeliegende Modell bekannt ist. Deshalb schlägt diese Arbeit das direkte Ableiten der gestörten Schaltsignale aus den originalen Schaltsignalen vor. Zur Modellierung der Vielzahl mögli-

i	1	2	3	4	5	6	7	8	9	10	11
$a_{1\max,i}$	0	0	0.8	0	0	0	0	0	0	0	0
$v_{1a,i}$	0	0	0.4	0	0	0	0	0	0	0	0
$v_{1k,i}$	0	0.4	0.4	0.4	0	0	0	0	0	0	0
$a_{2\max,i}$	0	0	0.3	0	0	0	0	0	0	0	0
$v_{2a,i}$	0	0	0.4	0	0	0	0	0	0	0	0

Tabelle 7.7: Parameter für Schaltsignal g (Signal 1 und Signal 2 synchronisiert)

i	1	2	3	4	5	6	7	8	9	10	11
$a_{1\max,i}$	0	0	0.3	0	0	0	0	0	0	0	0
$v_{1a,i}$	0	0	0.4	0	0	0	0	0	0	0	0
$v_{1k,i}$	0	0.4	0.4	0.4	0	0	0	0	0	0	0
$a_{2\max,i}$	0	0	0.8	0	0	0	0	0	0	0	0
$v_{2a,i}$	0	0	0.4	0	0	0	0	0	0	0	0

Tabelle 7.8: Parameter für Schaltsignal h (Signal 1 und Signal 2 synchronisiert)

cher sitzungsabhängiger Verteilungen und Parameter geht ein gestörtes Schaltsignal aus dem originalen Schaltsignal durch eine zufällige, gleichverteilte Dehnung in Zeit und Amplitude hervor (Bild 7.3). Je nach Anwendung sind natürlich noch andere Störungen möglich, z. B. die Erweiterung mit nichtlinearen Dehnungs- und Amplitudenkurven.

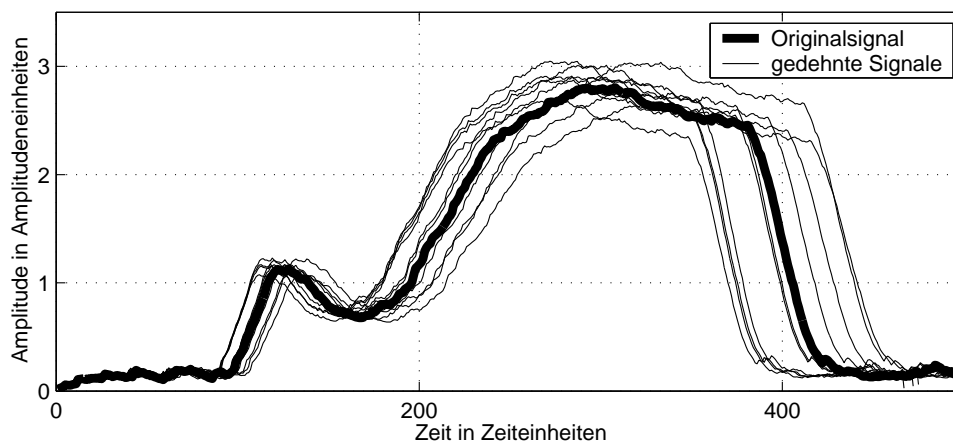


Bild 7.3: Originales Schaltsignal, zeit- und amplitudengedehnte Schaltsignale

Die maximale, prozentuale Dehnung beschreibt $\epsilon_{\max,t} \in [0, 1]$ für die Zeit und $\epsilon_{\max,a} \in [0, 1]$ für die Amplitude. Als Dehnungsfaktor für das p -te gestörte Schaltsignal $\bar{\Psi}_{i,p}$ ($p \in [1, p_{\max}]$) aus dem originalen Schaltsignal Ψ_i kommt eine Realisierung der gleichverteilten Größe $\epsilon_{t,p} \in [1 - \epsilon_{\max,t}, 1 + \epsilon_{\max,t}]$ für die Zeit und eine Realisierung der gleichverteilten Größe $\epsilon_{a,p} \in [1 - \epsilon_{\max,a}, 1 + \epsilon_{\max,a}]$ für die Amplitude zum Einsatz. Die Zeitreihe $\bar{u}_{i,p}$ des p -ten gestörten Schaltsignals $\bar{\Psi}_{i,p}$ ergibt sich aus $u_i(t)$ gemäß Bild 7.3

durch

$$\bar{u}_{i,p}(\epsilon_{t,p} \cdot t) = \epsilon_{a,p} \cdot u_i(t) \quad (7.7)$$

bzw. aus deren interpolierter Diskretisierung nach erneuter Abtastung mit T_a .

Um die Qualität des vorgeschlagenen Verfahrens zu belegen, werden drei neue Testdatensätze mit kleinen, mittleren und großen Dehnungen erstellt (Datensatz D_1 mit $\epsilon_{\max,t} = \epsilon_{\max,a} = 0.1$, Datensatz D_2 mit $\epsilon_{\max,t} = \epsilon_{\max,a} = 0.2$ und Datensatz D_3 mit $\epsilon_{\max,t} = \epsilon_{\max,a} = 0.4$). Identisch zu Datensatz C vereinen alle drei Datensätze $K = 8$ Klassen mit jeweils $n_j = 1000, j = 1, \dots, K$ Beispielen. Jedes Beispiel geht gemäß (7.7) aus einem originalen Schaltsignal hervor, die originalen Schaltsignale werden anschließend verworfen. Die Datensätze D_1, D_2, D_3 simulieren somit je 8000 Zufallsprozesse bzw. Verteilungen und werden als Näherung für die unendliche Anzahl möglicher Verteilungen verwendet.

Das Vorgehen zur Evaluierung des robusten Entwurfs zeigt Bild 7.4. Äquivalent zu Abschnitt 2.10.4 werden aus Datensatz B $n_j = 10$ Beispiele pro Klasse j zufällig gezogen und zu einem Lerndatensatz zusammengeführt (Datensatz L). Dieser Datensatz spiegelt die Verteilung für einen einzigen Zufallsprozess wider. Das neue Verfahren zum robusten Entwurf fügt dem Lerndatensatz für jedes Schaltsignal $p_{\max} = 10$ gestörte Realisierungen gemäß (7.7) hinzu mit $\epsilon_{\max,t} = \epsilon_{\max,a} = 0.1$ (insgesamt $n = 880$ Realisierungen, Datensatz L^*)¹. Der Datensatz simuliert somit elf Zufallsprozesse und enthält Informationen über deren Verteilungsfunktion. Um die Güte der beiden Lerndatensätze zu beurteilen, werden sie jeweils über dem ursprünglichen Testdatensatz C und über den neuen Testdatensätzen D_1, D_2, D_3 getestet.

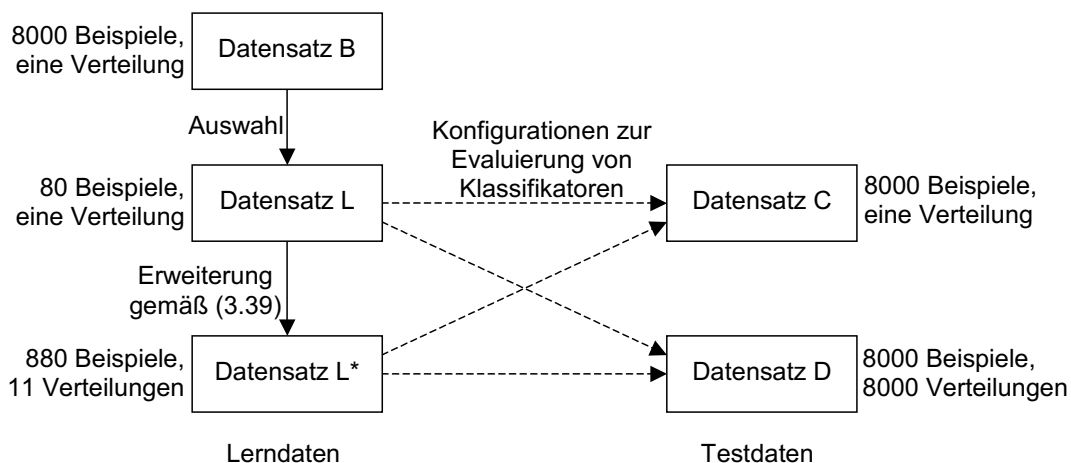


Bild 7.4: Konfiguration zur Evaluierung der Robustheit von Klassifikatoren

¹Hierbei ist zu beachten, dass der Testdatensatz D_1 zwar ebenfalls mit $\epsilon_{\max,t} = \epsilon_{\max,a} = 0.1$ entworfen wurde, dies jedoch für jedes der 8000 Beispiele separat durchgeführt wurde und die ungestörten Schaltsignale verworfen wurden. Im Gegensatz hierzu basiert der Lerndatensatz aus 800 gestörten Beispielen, die den 80 Beispielen des Originaldatensatzes entstammen. Eine Verwendung von Datensatz D_1 als Testdatensatz ist somit zulässig.

7.2 Klassifikationsergebnisse

7.2.1 Patientendatensätze und Entscheidungsregeln

Patient	Fuzzy	KNN	k-NN	SVM
A	7.0 ± 2.8	3.6 ± 1.5	2.6 ± 0.8	2.3 ± 0.4
B	5.7 ± 1.6	7.8 ± 2.7	15.2 ± 1.0	9.9 ± 1.2
C	10.6 ± 2.6	9.7 ± 3.0	6.1 ± 0.7	4.0 ± 0.7
D	10.6 ± 2.1	3.0 ± 1.2	3.5 ± 0.6	0.9 ± 0.2
E	11.3 ± 2.7	11.3 ± 1.9	10.1 ± 1.0	8.3 ± 1.2
F	8.5 ± 3.8	4.5 ± 1.8	5.7 ± 0.4	5.0 ± 0.6
G	6.4 ± 2.3	5.7 ± 1.8	3.7 ± 0.3	1.9 ± 0.0
H	15.6 ± 5.4	12.2 ± 2.8	6.1 ± 0.0	8.0 ± 1.1
I	11.5 ± 5.1	0.7 ± 1.1	0.0 ± 0.0	0.0 ± 0.0
J	24.0 ± 5.5	3.8 ± 2.0	4.5 ± 0.8	0.2 ± 0.6
K	9.7 ± 4.2	6.0 ± 2.2	6.1 ± 1.1	6.0 ± 1.1
L	6.2 ± 2.7	3.3 ± 0.8	1.9 ± 0.4	1.9 ± 0.0
M	8.8 ± 3.2	8.1 ± 2.2	5.9 ± 1.1	5.4 ± 1.0
N	9.7 ± 4.1	9.7 ± 2.1	1.4 ± 0.4	1.7 ± 0.9
O	17.4 ± 6.3	6.8 ± 2.2	5.4 ± 0.8	5.5 ± 1.0
P	4.3 ± 2.1	5.4 ± 1.9	10.0 ± 1.3	4.9 ± 0.9
Q	14.7 ± 4.2	4.5 ± 1.9	2.1 ± 1.0	3.1 ± 1.3
R	7.1 ± 2.2	1.5 ± 1.2	2.2 ± 0.0	2.2 ± 0.0
S	6.9 ± 2.1	0.9 ± 0.9	1.1 ± 0.5	0.8 ± 0.3
∅	10.3	5.7	4.5	3.9

Tabelle 7.9: Klassifikationsfehler diverser Entscheidungsregeln über Patientendaten (50-fache Wiederholung einer 10-fachen Crossvalidierung)

Tabelle 7.9 stellt den Klassifikationsfehler der Entscheidungsregeln Fuzzy-Regelbasis (Fuzzy), Künstliches Neuronales Netz (KNN), k-Nearest-Neighbour-Klassifikator (k-NN) und Support-Vektor-Maschine (SVM) über den Patientendaten bei 50-facher Wiederholung einer 10-fachen Crossvalidierung dar. Für das Künstliche Neuronale Netz werden $s_m = 8$ Merkmale mittels MANOVA ausgewählt, die anderen Verfahren verwenden alle Merkmale. Es zeigt sich qualitativ das gleiche Bild wie in Kapitel 2 anhand des Benchmarkdatensatzes.

Die Tabellen 7.10 und 7.11 zeigen die modifizierten Verfahren im Vergleich zum herkömmlichen Verfahren bei identischer Parametrierung. Tendenziell ergeben sich die gleichen Effekte wie in Abschnitt 5.2.9. Bei fehlerhaften Datensätzen und strukturvariablen Merkmalen liefert das MM-Verfahren schlechtere Ergebnisse. In diesem Fall ist auch für die Mehrschrittverfahren die Merkmalsauswahl nach MANOVA zu bevorzugen. Dies gilt insbesondere für Probleme mit vielen freien Parametern ($s_d = 4$). Hier neigen die modifizierten Verfahren dazu, den Datensatz zu überlernen. Fällt die Anzahl freier Parameter ($s_d = 3$), so liefern die modifizierten Verfahren wesentlich bessere Ergebnisse als das Referenzverfahren MAN/DI.

Pat.	MAN/DI	MAN/MD	MM/DI	MM/MD	HKK (MAN)	HKK (MM)	HKT (MAN)	KO (MAN)	KO (MM)
A	4.4 ± 1.1	4.0 ± 1.4	5.2 ± 1.8	4.6 ± 1.6	3.3 ± 1.1	2.4 ± 1.1	4.6 ± 1.6	3.4 ± 1.1	2.3 ± 0.7
B	9.0 ± 2.4	8.6 ± 2.5	10.4 ± 3.7	16.3 ± 3.9	6.5 ± 1.5	13.0 ± 4.3	7.2 ± 2.1	6.0 ± 1.9	9.6 ± 3.2
C	13.8 ± 2.9	10.7 ± 2.9	16.0 ± 3.1	14.7 ± 2.7	7.0 ± 1.7	11.0 ± 3.0	10.8 ± 2.6	5.5 ± 2.1	7.2 ± 2.1
D	5.1 ± 1.7	4.5 ± 1.5	7.4 ± 2.4	5.5 ± 1.9	2.8 ± 1.3	3.7 ± 1.5	5.7 ± 1.6	2.7 ± 1.1	2.9 ± 1.2
E	13.3 ± 2.1	11.3 ± 1.6	11.1 ± 2.8	11.4 ± 2.6	13.2 ± 2.8	11.2 ± 2.6	13.3 ± 2.0	9.9 ± 2.0	7.4 ± 1.8
F	6.1 ± 1.4	6.1 ± 1.7	8.4 ± 2.0	7.5 ± 1.5	6.7 ± 2.0	7.6 ± 1.8	6.3 ± 1.2	5.9 ± 1.0	7.1 ± 1.4
G	8.9 ± 1.3	8.1 ± 1.6	6.2 ± 2.6	6.0 ± 2.5	7.4 ± 1.7	5.8 ± 3.0	8.4 ± 1.8	5.9 ± 1.1	4.5 ± 2.2
H	8.5 ± 1.6	7.9 ± 2.0	8.2 ± 3.1	7.3 ± 2.7	4.6 ± 1.8	7.2 ± 2.8	8.4 ± 2.1	7.2 ± 1.4	7.1 ± 2.7
I	0.2 ± 0.7	0.4 ± 1.0	4.7 ± 2.9	4.4 ± 2.5	0.9 ± 1.5	4.8 ± 3.0	0.4 ± 0.9	0.4 ± 0.8	3.3 ± 2.2
J	2.8 ± 1.7	1.7 ± 1.5	3.3 ± 2.0	3.3 ± 2.1	0.9 ± 0.9	2.1 ± 1.4	2.4 ± 1.2	0.6 ± 0.7	2.0 ± 1.5
K	7.8 ± 1.8	7.3 ± 1.7	8.6 ± 2.6	8.0 ± 2.6	7.9 ± 1.6	7.6 ± 2.4	8.2 ± 1.8	4.9 ± 1.6	6.7 ± 2.1
L	5.7 ± 1.3	4.7 ± 1.0	4.7 ± 1.3	4.1 ± 1.0	2.7 ± 0.9	3.1 ± 0.8	5.8 ± 0.9	3.7 ± 0.6	4.0 ± 0.6
M	7.9 ± 2.1	8.0 ± 2.2	7.0 ± 2.1	7.9 ± 1.9	7.9 ± 2.2	6.6 ± 2.0	8.6 ± 2.3	8.2 ± 1.9	6.0 ± 1.7
N	17.2 ± 3.5	7.8 ± 2.1	4.6 ± 2.0	5.1 ± 2.1	8.3 ± 2.0	6.1 ± 2.2	8.0 ± 2.0	7.4 ± 2.0	4.7 ± 1.9
O	7.7 ± 2.5	7.5 ± 1.9	10.8 ± 2.7	11.0 ± 2.7	7.5 ± 2.4	9.6 ± 2.3	9.5 ± 2.7	6.4 ± 1.5	8.1 ± 1.6
P	4.6 ± 1.1	4.2 ± 1.7	4.1 ± 1.8	4.5 ± 1.9	4.7 ± 1.4	6.6 ± 3.1	4.2 ± 1.6	3.8 ± 1.0	3.4 ± 1.1
Q	5.3 ± 1.9	5.5 ± 1.6	7.6 ± 2.3	8.5 ± 2.7	4.5 ± 1.8	8.3 ± 2.8	4.4 ± 1.5	4.9 ± 1.8	6.1 ± 2.6
R	6.6 ± 1.7	3.8 ± 1.5	6.8 ± 2.5	4.7 ± 1.9	4.0 ± 1.5	5.0 ± 2.0	7.8 ± 1.8	1.1 ± 0.5	2.3 ± 1.6
S	2.8 ± 1.1	1.4 ± 0.7	4.8 ± 1.9	2.7 ± 1.6	1.2 ± 0.7	1.5 ± 0.9	2.2 ± 0.7	0.8 ± 0.6	0.5 ± 0.6
Ø	7.2	6.0	7.4	6.9	5.4	6.5	6.6	4.7	5.0

Tabelle 7.10: Klassifikationsfehler der modifizierten Verfahren bei Parameteranpassung, $s_m = 8$, $s_d = 3$, 50-fache Wiederholung einer 10-fachen Crossvalidierung

Patient	MAN/DI	MAN/MD	MM/DI	MM/MD	HKK (MAN)	HKK (MM)
A	5.5 ± 1.3	4.9 ± 1.2	4.3 ± 1.1	3.6 ± 1.4	3.4 ± 1.5	2.4 ± 1.1
B	7.4 ± 2.0	9.0 ± 2.4	12.4 ± 3.7	17.7 ± 4.7	6.8 ± 1.6	13.4 ± 3.4
C	11.2 ± 2.8	10.4 ± 2.4	14.1 ± 3.2	12.7 ± 3.3	9.3 ± 2.1	11.2 ± 2.6
D	6.1 ± 1.7	4.9 ± 1.5	5.8 ± 1.7	5.5 ± 1.9	4.0 ± 1.5	4.5 ± 1.9
E	11.7 ± 1.6	11.2 ± 2.0	11.0 ± 2.7	10.8 ± 2.7	12.5 ± 2.6	10.5 ± 2.3
F	5.8 ± 0.5	6.0 ± 0.7	8.1 ± 1.7	7.4 ± 1.5	6.9 ± 1.2	8.3 ± 1.7
G	9.2 ± 1.9	8.7 ± 1.5	6.6 ± 2.1	7.6 ± 2.8	6.2 ± 1.9	6.3 ± 3.1
H	8.6 ± 1.6	9.3 ± 1.9	7.4 ± 2.7	7.3 ± 2.8	3.9 ± 1.6	7.5 ± 3.0
I	0.0 ± 0.3	0.4 ± 0.9	6.0 ± 3.3	6.1 ± 3.1	0.5 ± 1.0	4.9 ± 3.3
J	1.5 ± 1.5	1.0 ± 1.0	2.9 ± 1.6	2.7 ± 2.0	1.5 ± 1.0	3.0 ± 2.0
K	7.6 ± 1.5	7.4 ± 1.8	7.4 ± 2.3	9.1 ± 3.3	8.0 ± 1.5	9.2 ± 2.7
L	5.4 ± 0.9	4.1 ± 0.8	4.2 ± 1.1	3.9 ± 1.3	2.8 ± 0.9	3.3 ± 0.8
M	7.5 ± 1.7	7.2 ± 1.8	6.6 ± 2.1	6.4 ± 2.2	7.8 ± 2.0	6.5 ± 1.7
N	7.8 ± 2.1	8.0 ± 1.8	5.2 ± 2.1	5.1 ± 2.0	8.7 ± 2.2	5.6 ± 2.1
O	6.9 ± 2.3	7.0 ± 2.0	10.2 ± 2.3	10.5 ± 2.1	7.5 ± 2.2	9.3 ± 2.0
P	4.5 ± 1.6	4.8 ± 1.7	4.8 ± 1.7	5.1 ± 2.1	4.8 ± 1.4	6.5 ± 2.5
Q	8.7 ± 1.9	7.0 ± 2.1	7.6 ± 2.7	7.7 ± 2.2	6.6 ± 1.8	8.5 ± 2.7
R	5.9 ± 1.0	3.7 ± 1.4	4.8 ± 1.9	4.6 ± 1.9	3.0 ± 1.0	6.2 ± 2.2
S	1.8 ± 0.7	1.3 ± 0.5	3.4 ± 1.3	2.2 ± 1.3	1.5 ± 0.7	1.3 ± 0.9
Ø	6.5	6.1	6.9	7.1	5.6	6.7

Tabelle 7.11: Klassifikationsfehler der modifizierten Verfahren bei Parameteranpassung, $s_m = 8$, $s_d = 4$, 50-fache Wiederholung einer 10-fachen Crossvalidierung

7.2.2 Benchmarkdatensätze

Der Entwurf der modifizierten Verfahren orientiert sich an Datensätzen mit Mehrklassenproblemen, relativ wenig Datensätzen mit strukturvariablen Merkmalen und versucht möglichst im niederdimensionalen Raum gute Ergebnisse zu liefern. Tabelle 7.12 zeigt die Ergebnisse verschiedener Parametrierungen über den gängigen Benchmarkdatensätzen WINE ($K = 3, n_1 = 59, n_2 = 71, n_3 = 48, s = 13$, [58]), ZOO ($K = 7, n_1 = 41, n_2 = 20, n_3 = 5, n_4 = 13, n_5 = 4, n_6 = 8, n_7 = 10, s = 16$, [59]) und IRIS ($K = 3, n_1 = n_2 = n_3 = 50, s = 4$, [4]).

Für den IRIS-Datensatz zeigen sich keine Verbesserungen durch die modifizierten Verfahren, da sowohl Klassenanzahl K als auch Merkmalsanzahl s_m zu klein sind. Beim WINE-Datensatz ergeben sich bereits bessere Ergebnisse als durch MANOVA/LDA, die jedoch aufgrund der geringen Merkmalsanzahl nicht deutlich ausfallen. Sehr gute Ergebnisse liefern die modifizierten Verfahren über dem ZOO-Datensatz, insbesondere im Fall niedriger Merkmalsdimension s_d , für den die Verfahren konstruiert worden sind.

²Für den IRIS-Datensatz findet keine Merkmalsauswahl statt, das HKT-Verfahren arbeitet mit $s_m^{\min} = 2, m_{\max} = 2$.

Datensatz	WINE	ZOO	IRIS ²
K	3	7	3
n_j	$n_1 = 59$ $n_2 = 71$ $n_3 = 48$	$n_1 = 41$ $n_2 = 20$ $n_3 = 5$ $n_4 = 13$ $n_5 = 4$ $n_6 = 8$ $n_7 = 10$	$n_1 = 50$ $n_2 = 50$ $n_3 = 50$
s	13	16	4
MANOVA, LDA, $s_m = 8, s_d = 1$	8.3 ± 0.9	24.9 ± 3.8	2.5 ± 0.4
MANOVA, LDA, $s_m = 8, s_d = 2$	2.8 ± 0.8	10.4 ± 2.6	2.7 ± 0.3
MANOVA, LDA, $s_m = 8, s_d = 3$	2.8 ± 0.8	7.3 ± 1.7	2.7 ± 0.4
MM, LDA, $s_m = 8, s_d = 1$	9.9 ± 1.7	41.1 ± 4.3	-
MM, LDA, $s_m = 8, s_d = 2$	2.2 ± 0.8	17.8 ± 2.9	-
MM, LDA, $s_m = 8, s_d = 3$	1.5 ± 0.7	10.7 ± 2.1	-
MANOVA, MD, $s_m = 8, s_d = 1$	8.1 ± 0.1	13.8 ± 2.5	2.6 ± 0.3
MANOVA, MD, $s_m = 8, s_d = 2$	2.8 ± 1.2	8.3 ± 1.9	3.0 ± 0.4
MANOVA, MD, $s_m = 8, s_d = 3$	2.6 ± 1.0	8.2 ± 2.0	3.0 ± 0.5
MM, MD, $s_m = 8, s_d = 1$	8.8 ± 1.2	13.5 ± 2.6	-
MM, MD, $s_m = 8, s_d = 2$	2.1 ± 0.7	10.1 ± 2.0	-
MM, MD, $s_m = 8, s_d = 3$	2.2 ± 0.9	11.1 ± 2.0	-
HKK, $s_m = 8, s_d = 1$	9.0 ± 1.5	12.9 ± 2.2	2.6 ± 0.3
HKK, $s_m = 8, s_d = 2$	2.7 ± 1.1	9.5 ± 2.8	2.9 ± 0.5
HKK, $s_m = 8, s_d = 3$	2.6 ± 1.0	9.0 ± 1.9	2.8 ± 0.5
HKK, MM, $s_m = 8, s_d = 1$	8.8 ± 1.4	13.6 ± 3.2	-
HKK, MM, $s_m = 8, s_d = 2$	2.7 ± 0.9	14.5 ± 3.2	-
HKK, MM, $s_m = 8, s_d = 3$	2.1 ± 0.8	13.8 ± 2.9	-
HKT, $s_m^{\min} = 6, m_{\max} = 3, s_d = 1$	12.6 ± 2.7	25.8 ± 4.0	3.1 ± 0.6
HKT, $s_m^{\min} = 6, m_{\max} = 3, s_d = 2$	2.8 ± 0.8	12.5 ± 2.7	3.4 ± 0.7
HKT, $s_m^{\min} = 6, m_{\max} = 3, s_d = 3$	2.8 ± 1.6	13.0 ± 2.4	3.8 ± 0.7
TN _{sel} , $s_m = 8$	3.3 ± 1.0	7.2 ± 1.7	2.9 ± 0.4
TN _{ind} , $s_m = 8$	3.9 ± 0.8	14.9 ± 2.1	3.8 ± 0.5
KO _{sel} , $s_m = 8$	3.3 ± 0.9	7.7 ± 2.0	2.9 ± 0.4
KO _{ind} , $s_m = 8$	3.8 ± 0.8	15.2 ± 2.2	3.6 ± 0.4
TN _{sel} , $s_m = 8, MM$	2.8 ± 1.0	16.4 ± 2.7	-
TN _{ind} , $s_m = 8, MM$	4.4 ± 1.0	27.9 ± 2.8	-
KO _{sel} , $s_m = 8, MM$	2.6 ± 1.0	16.6 ± 2.7	-
KO _{ind} , $s_m = 8, MM$	4.4 ± 1.0	27.7 ± 3.1	-

Tabelle 7.12: Benchmarkergebnisse für drei in der Literatur übliche Benchmarks (WINE [58], ZOO [59] und IRIS [4]), 50 Wiederholungen einer 10-fachen Crossvalidierung

7.3 Programmierete Funktionen

7.3.1 DAVE

Datei	Funktion	Zeilen
Aufnahme.cpp	Aufnahme von Biosignalen, Ansteuerung A/D-Wandler	276
ChildView.cpp	Koordinierung GUI	186
fließendeDarstellung.cpp	Anzeigefunktionen für Biosignale	339
HighscoreFenster.cpp	Punktzahlanzeige für Tetris	192
Klassifikation.cpp	Klassifikation, Parameterübergabe aus DAVE-Design	461
Morsecode.cpp	Trainingsprogramm für Schaltsignale, manuelle Adaption	627
Mainframe.cpp	Darstellung Fenster und Konstruktor	178
Signaldarstellung.cpp	Zeit-Amplituden-Darstellung, Bearbeitungsfunktionen	403
Steuerung.cpp	Umsetzung des Zustandsautomaten	315
Training.cpp	Trainieren von Schaltsignalen, automatische Adaption (ohne Merkmalsextraktor)	1319
Übung.cpp	Tetris	726

Tabelle 7.13: Visual C++ Source Code Files für DAVE, die im Rahmen dieser Arbeit entstanden sind

7.3.2 Mikrocontroller

Datei	Funktion	Zeilen
arm.c	komplette Prothesensteuerung (manuell)	1506
ausgaben.h	Ausgabe-Funktionen von Mikrocontroller und T&E-Box	135
export patient.c	individuelle Merkmalsextraktion	903
ansi.c	AML-Entscheidungsregel	91
adc test.c	A/D-Wandler des Mikrocontrollers	313
	sonstige Header-Dateien	ca. 500

Tabelle 7.14: C Source Code Files für Mikrocontroller, die im Rahmen dieser Arbeit entstanden sind

7.3.3 DAVE-Design

Dateiname	Funktion	Zeilen
Adapt_Online.m	Echtzeit-Adaption von Parametern	68
Ansicht.m	Aufbau von Dialogfenstern	111
Ansicht_Bayes_hierch.m	Dialogfenster für HKK	120
Ansicht_Bayes_hierch_Tup.m	Dialogfenster für HKT	182
Ansicht_bestklass_cov.m	Dialogfenster für MAML	70
Ansicht_bestklass_select.m	Dialogfenster für die Kostenoptimierung	70
Ansicht_cluster.m	Dialogfenster für Cluster-Algorithmen	61
Ansicht_Crossvalidierung.m	Dialogfenster für die Crossvalidierung	98
Ansicht_Laden.m	Dialogfenster für das Laden von Patientendaten	363
Ansicht_LZeichnenCodiert.m	Dialogfenster für die Darstellung des Datensatzes	61
Ansicht_LZeichnenSelektiv.m	Dialogfenster für die selektive Darstellung des Datensatzes	40
Ansicht_Merk_erweitern.m	Dialogfenster für die Kombination von Merkmalen	98
Ansicht_Merk_manuell.m	Dialogfenster für die manuelle Merkmalsauswahl	58
Ansicht_Merk_red.m	Dialogfenster für MANOVA	76
Ansicht_plot_merk.m	Dialogfenster für die Visualisierung der Merkmalsauswahl	40
Ansicht_red_dat.m	Dialogfenster für die Reduktion des Datensatzes	40
Autoregression.m	Berechnung von autoregressiven Modellen	37
Button_Bayes_hierch.m	Parametrierung und Start der HKK	38
Button_Bayes_hierch_tupel.m	Parametrierung und Start der HKT	43
Button_BestKlass.m	Parametrierung und Start der MD	40
Button_bestklass_cov.m	Parametrierung und Start der MAML	30
Button_BestKlass_select.m	Parametrierung und Start der Kostenoptimierung	39
Button_Bestldf.m	Parametrierung und Start der bestLDF	40
button_cluster.m	Parametrierung und Start von Cluster-Algorithmen	57
Button_CQuellMerkmale.m	Start des Quelltextexports für Merkmale	11
Button_Crossvalidierung.m	Parametrierung und Start der Crossvalidierung	14
Button_Diskriminanz.m	Parametrierung und Start der Diskriminanzanalyse	26
Button_Disk_loeschen.m	Entfernung der Einstellungen der Diskriminanzanalyse	35
Button_Externe_Testdaten.m	Start der Validierung über externe Testdaten	46
Button_Laden.m	Einlesen von Patientendatensätzen	101
Button_LZeichnen.m	Parametrierung und Start der Zeichenroutinen	14
Button_Manova.m	Parametrierung und Start der MANOVA	22
Button_Speichern.m	Sicherung aller Variablen	18
Button_TZeichnen.m	Zeichnen von Testdaten	10
Button_Uebern_Merkaus.m	Übernahme der Merkmalsauswahl	38
Button_Varimax.m	Parametrierung und Start der Drehung des Merkmalsraums	13
Button_ZeichnenCodiert.m	Parametrierung und Start der Zeichenroutine für Zeitreihenklassen	23

Fortsetzung auf der nächsten Seite

Dateiname	Funktion	Zeilen
calc_anz_param.m	Berechnung von Multiplikationen/Parametern bei HKK	36
calc_anz_par_hierch_tupel.m	Berechnung von Multiplikationen/Parametern bei HKT	34
choose_and_class_testdata.m	Validierungsroutine für Benchmarks	145
convert.m	Konvertierung von EEG-Daten	38
create_bench_8Sig.m	Entwurf eines Benchmarkdatensatzes	240
crossval_batch.m	Batch-Datei zum Ausführen von Crossvalidierungen	352
cvdiskri3.m	Crossvalidierungsroutine für echte Datensätze	347
DatensaeetzeLern.m	Auswertung der notwendigen Anzahl Lerndatentupel	63
daten_erweitern.m	Erweiterung des Datensatzes durch Streuungen	27
dialogfenster.m	Standardroutine zum Darstellen von Dialogfenstern	52
diskri_an.m	Anwenden der Diskriminanzanalyse	25
diskri_en.m	Entwerfen der Diskriminanzanalyse	46
eeg_crossval.m	Crossvalidierung in EEG-Zeitreihen	76
EinerGegenAlle_an.m	Anwenden der One-against-All-Aufteilung für beliebige Klassifikatoren	59
EinerGegenAlle_en.m	Entwerfen der One-against-All-Aufteilung für beliebige Klassifikatoren	49
Einlesen.m	Parametrierung der Einleseroutine für Patientendatensätze	69
export.m	Parameterexport eines Klassifikators für DAVE	7
export_ansic_bayes.m	Quelltextexport für AML-Klassifikatoren	238
export_ansic_turnier.m	Quelltextexport für Turnier- und KO-Klassifikatoren	115
Export_Merkmale_C.m	Quelltextexport für Schalten ohne Zeitversatz	249
export_Merk_C_gesamt.m	Quelltextexport für Schalten mit Zeitversatz	1037
Externe_Testdaten.m	Validierung von Klassifikatoren über externen Testdaten	68
gesamt_Ber_Merk_mult.m	Parametrierung der Merkmalsberechnung für Zeitpunkt t_h	33
gesamt_Ber_Merk_mult_t.m	Parametrierung der Merkmalsberechnung für Zeitpunkt t_e	32
gesamt_daten_mult_sens.m	Berechnung von Zeitreihen aus Sensorkombinationen	35
gesamt_Extr_Daten.m	Parametrierung der Merkmalsextraktion	84
gesamt_Merk_suchen.m	Extraktion von Zeitreihenmerkmalen zum Zeitpunkt t_h	186
gesamt_Merk_suchen_txtr.m	Extraktion von Zeitreihenmerkmalen zum Zeitpunkt t_e	171
guete_bestklass.m	Zielfunktion des MD-Verfahrens	18
guete_bestklass_select.m	Zielfunktion des MD-Verfahrens mit Kostenoptimierung	32
guete_bestldf.m	Zielfunktion des bestLDF-Verfahrens	44
guete_cov.m	Zielfunktion des MAML-Verfahrens	52
guete_hierch.m	Zielfunktion für das HKK-Verfahren	8
guete_varianz_log.m	Zielfunktion für logistische Diskriminanzanalyse	32
GUI.m	Grundfunktion der grafischen Benutzeroberfläche	685
gui_menu.m	Menüfunktionen der grafischen Benutzeroberfläche	141
IIRFilter.m	Implementierung eines IIR-Filters	12
klassdist.m	Bestimmung der Kullback-Leibler-Divergenz von Klassen	45
Klass_Bayes.m	Evaluierung des AML-Klassifikators	35

Fortsetzung auf der nächsten Seite

Dateiname	Funktion	Zeilen
Klass_Fuzzy.m	Evaluierung der Fuzzy-Regelbasis	42
klass_knneighbour.m	Evaluierung des k-Nearest-Neighbour-Klassifikators	37
Klass_KO.m	Evaluierung des KO-Verfahrens	38
klass_logdisc.m	Evaluierung der logistischen Diskriminanzanalyse	37
klass_nn.m	Evaluierung des Künstlichen Neuronalen Netzes	36
klass_svm.m	Evaluierung der Support-Vektor-Maschine	58
Klass_Turnier.m	Evaluierung des TN-Klassifikators	42
Klass_turnier_gegenalle.m	Evaluierung der One-against-all-Aufteilung	56
klf_an_hierch.m	Berechnung des HKK-Verfahrens	40
klf_an_hierch_dat.m	Berechnung des HKT-Verfahrens	62
klf_en_hierch.m	Entwurf des HKK-Verfahrens	147
klf_en_hierch_dat.m	Entwurf des HKT-Verfahrens	115
kosystem_an.m	Berechnung des KO-Verfahrens	57
load_projekt.m	Einladen einer Projektdatei	27
logdisc_an.m	Berechnung der logistischen Diskriminanzanalyse	6
logdisc_en.m	Entwurf der logistischen Diskriminanzanalyse	17
MaxMin.m	Automat zur Merkmalsberechnung	82
Merkmale.m	Vergleich von t_h und t_e	111
merkmale_erweitern.m	Berechnung neuer Merkmale	110
merk_red_bayes.m	modifizierte Merkmalsauswahl nach (3.4)	35
merk_red_bayes_mod.m	modifizierte Merkmalsauswahl nach (3.6)	47
merk_red_bayes_unstetig.m	modifizierte Merkmalsauswahl nach (3.2)	30
MMIIndex.m	Berechnung des MMI-Index	91
nn_an.m	Berechnung des Künstlichen Neuronalen Netzes	43
nn_en.m	Entwurf des Künstlichen Neuronalen Netzes	86
Optimiere.m	numerischer Optimierer	75
plot_hoehe.m	Zeichenroutine: Höhenlinien von Verteilungen	38
plot_klass.m	Zeichenroutine: Merkmalsraum mit Klassen	58
plot_klass_zg.m	Zeichenroutine: Merkmalsraum mit Klassenbereichen	15
plot_klass_zg_trenn.m	Zeichenroutine: Merkmalsraum mit Klassen und Grenzen	55
plot_mesh.m	Zeichenroutine: Merkmalsraum mit Dichtefunktionen	71
plot_reg.m	Zeichenroutine: Klassifikationsfehler über Lerndatentupel	35
plot_rueck.m	Basisroutine zum Berechnen von Klassengrenzen	37
refresh_window.m	Aktualisierung aller Fenster	20
reset_gui.m	Initialisierung der Benutzeroberfläche	32
robust.m	robuster Entwurf	46
rueckweisung.m	Rückweisungsroutine für HKT	32
save_projekt.m	Speichern eines Projekts	34
schalt_zeit.m	Berechnung von Schaltsignaldauern	35
Schnittpunkte.m	Berechnung des Schnittpunktmerkmals	11
set_choice.m	Parametrierungsfunktion der Benutzeroberfläche	31

Fortsetzung auf der nächsten Seite

Dateiname	Funktion	Zeilen
set_cost.m	Setzen der Kostenmatrix	20
Standardabweichung.m	Berechnung der Standardabweichung eines Signals	33
startup.m	Startroutine der grafischen Benutzeroberfläche	77
svm_an.m	Berechnung der Support-Vektor-Maschine	32
svm_en.m	Entwurf der Support-Vektor-Maschine	31
tauswahl_Extr_Daten.m	Parametrierung der Merkmalsextraktion beim Schalten ohne Zeitversatz	180
taus_Ber_Merkmale_mult.m	Parametrierung der Merkmalsberechnung beim Schalten ohne Zeitversatz	91
taus_Merkmale_suchen.m	Extraktion der Merkmale aus einer Zeitreihe beim Schalten ohne Zeitversatz	53
turnierklass_an.m	Berechnung des TN-Klassifikators	52
turnierklass_en.m	Entwurf des TN-Klassifikators	48
valid.m	Batch-Datei zur Evaluierung des Benchmarkdatensatzes	156
val_externetestdaten.m	Validierungsfunktion zur Berechnung der Anzahl notwendiger Datentupel	241
val_testdaten.m	Auswertung von Klassifikationsgütern mehrerer Sitzungen	167
varimax.m	Varimax-Transformation	37

Tabelle 7.15: MATLAB m-Files für DAVE-Design, die im Rahmen dieser Arbeit entstanden sind

7.4 Quelltextgenerierung und Implementierung von Fuzzy-Regelbasen

Für Benchmarkdatensatz A ergibt sich folgende Regelbasis:

R_1 :	WENN (S1-S2) rel. Dauer Bereich II=PSG UND ((S1/(S1+S2)) Dauer Bereich II=PK ODER PM)	DANN y=7
R_2 :	WENN (S1/(S1+S2)) Trendmaximum Bereich II=ZE	DANN y=8
R_3 :	WENN S1 rel. Minimum Bereich III=PK UND (S2 Maximum Bereich IV=PM ODER PG ODER PSG)	DANN y=2
R_4 :	WENN S1 rel. Minimum Bereich III=PK UND (S2 Maximum Bereich IV=PK ODER PM)	DANN y=1
R_5 :	WENN S1 Maximum Bereich II=PSG	DANN y=3
R_6 :	WENN S1 rel. Minimum Bereich III=PG UND (S1 rel. Summe Bereich II=PSK ODER PK ODER PM)	DANN y=4
R_7 :	WENN (S1-S2) Minimum Bereich III=NK UND (S1/(S1+S2)) rel. Dauer Bereich IV=PM	DANN y=5
R_8 :	WENN S1 rel. Minimum Bereich III=PK UND S2 Maximum Bereich IV=ZE	DANN y=5
R_9 :	WENN S1 Maximum Bereich IV=PK	DANN y=6
	SONST y=1	

Der Berechnungsaufwand wird für jede Regelbasis durch das Kombinieren von Fuzzifizierung und Aggregation verringert [158]. Disjunktionen von Termen werden als neue linguistische Terme mit trapezförmigen Zugehörigkeitsfunktionen interpretiert. Zugehörigkeitswerte aller Zugehörigkeitsfunktionen werden berechnet, indem begrenzende Geraden ($f(x)$ bzw. $g(x)$) eingeführt werden (Bild 7.5). Unter

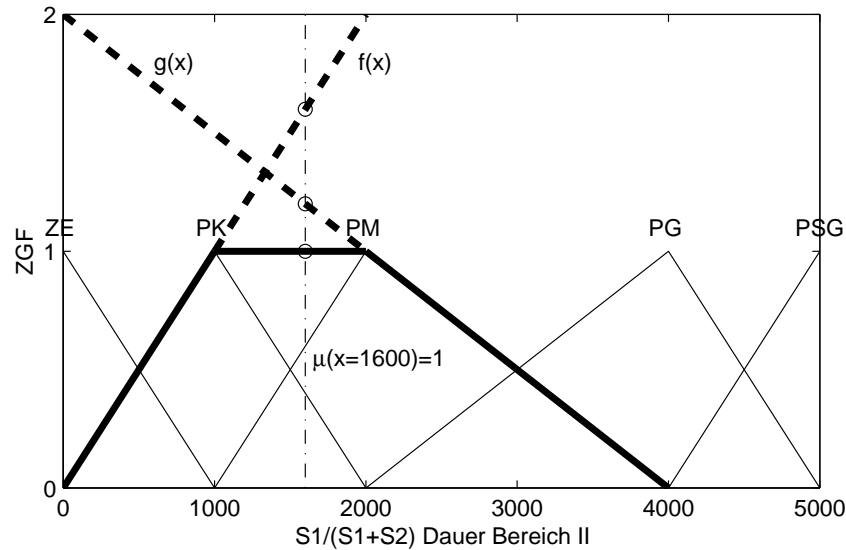


Bild 7.5: Schnelle Fuzzifizierung von dreiecks- und trapezförmigen Zugehörigkeitsfunktionen, Auswertung der Teilprämisse $(S1/(S1+S2)$ Dauer Bereich II=PK ODER PM) aus Regel R_1 :
 $f(x) = \frac{1}{1000}x$, $g(x) = -\frac{1}{2000}x + 2$, $\mu(x = 1600) = \min(1, 1.2, 1.6) = 1$

Verwendung der SUM-PROD Inferenz wird die Zugehörigkeit des resultierenden Terms $\mu(x)$ aus dem Minimum der beiden Funktionen $f(x)$ bzw. $g(x)$ berechnet und auf 1 limitiert:

$$\mu(x) = \begin{cases} 0, & \text{für } t_p(x) < 0 \\ 1, & \text{für } t_p(x) > 1 \\ t_p & \text{sonst} \end{cases} \quad \text{mit } t_p(x) = \min(f(x), g(x)). \quad (7.8)$$

Die Aggregation wird durch eine Multiplikation der Ergebnisse von (7.8) mit allen in den Regelprämissen verwendeten Merkmalen beendet (UND-Verknüpfung). Aktivierung und Akkumulation werden durch eine Addition aller Zugehörigkeitswerte gleicher Konklusionen durchgeführt, zur Defuzzifizierung wird die Maximum-Methode verwendet.

Der verwendete Algorithmus führt zu einem automatisch generierten, kompakten Quelltext, der sehr recheneffizient bezogen auf die Anzahl notwendiger Multiplikationen und Speicherplätze arbeitet.

Es folgt nun der automatisch generierte Quelltext für die Regelbasis zur Klassifikation von Benchmarkdatensatz A:

```

//Automatically generated C code of a fuzzy rulebase with 10 rules
//Project: bench_8kl_80.prj, 12-May-2005, 08:42

//Default value output variable: y=1
/*Rulebase:
  1. Regel (Q=+0.956, 0 Feh./ 10 Bsp.): WENN x58=5 UND (x82=2 ODER 3) DANN y=7
  2. Regel (Q=+0.804, 0 Feh./ 10 Bsp.): WENN x106=1 DANN y=8
  3. Regel (Q=+0.676, 2 Feh./ 12 Bsp.): WENN x11=2 UND (x36=3 ODER 4 ODER 5) DANN y=2
  4. Regel (Q=+0.591, 1 Feh./ 11 Bsp.): WENN x11=2 UND (x36=2 ODER 3) DANN y=1
  5. Regel (Q=+0.561, 1 Feh./ 11 Bsp.): WENN x7=5 DANN y=3
  6. Regel (Q=+0.537, 0 Feh./ 9 Bsp.): WENN x11=4 UND (x16=1 ODER 2 ODER 3) DANN y=4
  7. Regel (Q=+0.245, 0 Feh./ 5 Bsp.): WENN x62=2 UND x87=3 DANN y=5
  8. Regel (Q=+0.172, 0 Feh./ 3 Bsp.): WENN x11=2 UND x36=1 DANN y=5
  9. Regel (Q=+0.033, 5 Feh./ 15 Bsp.): WENN x9=2 DANN y=6
*/

//Input variables, program definition necessary: float x[109];
//Output variable's membership degrees, program definition necessary: float mu_y[9];

//Function for partial premises and fuzzification (triangular or trapezoid MBF's
float computeTp(float tp1,float tp2)
{
float tp;
if (tp1<tp2) tp=tp1; else tp=tp2;
if (tp<0) tp=0;
if (tp>1) tp=1;
return tp;
};

//Main function fuzzy_control, return value: y (defuzzified output value)
float fuzzy_control(float* x,float* mu_y)
{
float pr[11],sum_mu_y;
int i;
float y,zgf_y[]={1.000,2.000,3.000,4.000,5.000,6.000,7.000,8.000};

//Fuzzification and rule premises (one step!)
pr[1]=computeTp((x[58]-0.300000)/0.700000,1)*computeTp((x[82])/1000.000000,
(4000.000000-x[82])/2000.000000);
pr[2]=computeTp(1,(0.700000-x[106])/0.700000);
pr[3]=computeTp((x[11])/6.000000,(10.000000-x[11])/4.000000)*
computeTp((x[36]-0.700000)/0.100000,1);
pr[4]=computeTp((x[11])/6.000000,(10.000000-x[11])/4.000000)*
computeTp((x[36])/0.700000,(0.860000-x[36])/0.060000);
pr[5]=computeTp((x[7]-2.000000)/1.000000,1);
pr[6]=computeTp((x[11]-10.000000)/20.000000,(40.000000-x[11])/10.000000)*
computeTp(1,(0.400000-x[16])/0.100000);
pr[7]=computeTp((x[62]+0.500000)/0.460000,(-x[62])/0.040000)*
computeTp((x[87]-0.200000)/0.110000,(0.350000-x[87])/0.040000);
pr[8]=computeTp((x[11])/6.000000,(10.000000-x[11])/4.000000)*
computeTp(1,(0.700000-x[36])/0.700000);
pr[9]=computeTp((x[9])/1.000000,(2.000000-x[9])/1.000000);
//Rule conclusions (always SUM-PROD-inference, no overlapping correction);
mu_y[1]=pr[4]+pr[10];
mu_y[2]=pr[3];
mu_y[3]=pr[5];
mu_y[4]=pr[6];
mu_y[5]=pr[7]+pr[8];
mu_y[6]=pr[9];
mu_y[7]=pr[1];
mu_y[8]=pr[2];
//Sum-Prod-Inference
sum_mu_y=0;y=0;
for (i=1;i<9;i++)
{
y+=mu_y[i]*zgf_y[i-1];sum_mu_y+=mu_y[i];
};
return y;
};

```


7.5 Merkmalsrelevanzen

Position	Merkmal	Güte ($M_{\mathcal{I}}$)
1	S1 Minimum Bereich III	0.958
2	S1 rel. Maximum Bereich IV	0.951
3	S1 rel. Dauer Bereich II	0.947
4	S1 rel. Minimum Bereich III	0.943
5	S1 Maximum Bereich IV	0.935
6	S1 rel. Summe Bereich II	0.927
7	$(S1/(S1+S2))$ Maximum Bereich II	0.916
8	S2 Maximum Bereich II	0.902
9	$(S1/(S1+S2))$ rel. Maximum Bereich IV	0.900
10	$(S1/(S1+S2))$ rel. Dauer Bereich II	0.894

Tabelle 7.16: Merkmalsrelevanzen ANOVA ($M_{\mathcal{I}}$), Benchmarkdatensatz B

Position	Merkmal	Güte ($M_{\mathcal{I}}$)
1	S1 Minimum Bereich III	0.958
2	S1 rel. Maximum Bereich IV	0.998
3	S1 Maximum Bereich II	1.000
4	S2 Maximum Bereich II	1.000
5	S1 Maximum Bereich IV	1.000
6	$(S1/(S1+S2))$ rel. Summe Bereich II	1.000
7	S2 Summe Bereich IV	1.000
8	S1 rel. Maximum Bereich II	1.000
9	$(S1/(S1+S2))$ Dauer Bereich II	1.000
10	S2 rel. Dauer Bereich IV	1.000

Tabelle 7.17: Merkmalsrelevanzen MANOVA ($M_{\mathcal{I}}$, sukzessive Auswahl), Benchmarkdatensatz B

Merkmal	t_h	t_e
S1 Dauer Bereich III	0.855	0.826
S1 Dauer Bereich IV	0.841	0.789
S1 Dauer Bereich II	0.472	0.028
S1 Minimum Bereich III	0.958	0.958
S1 Maximum Bereich IV	0.935	0.935
S1 Maximum Bereich II	0.884	0.884
S1 Summe Bereich III	0.890	0.893
S1 Summe Bereich IV	0.861	0.838
S1 Summe Bereich II	0.794	0.730
S1 Filterkreuzungen Bereich III	0.278	0.278
S1 Filterkreuzungen Bereich IV	0.258	0.264
S1 Filterkreuzungen Bereich II	0.103	0.030
S1 Stand.abw Bereich IV	0.650	0.780
S1 Stand.abw Bereich II	0.494	0.579
S1 Stand.abw Bereich III	0.474	0.771
S1 Trendminimum Bereich III	0.851	0.851
S1 Trendmaximum Bereich IV	0.807	0.807
S1 Trendmaximum Bereich II	0.735	0.735

Tabelle 7.18: Merkmalsrelevanzen ANOVA (M_T), Vergleich t_h zu t_e , Benchmarkdatensatz B

Merkmal	t_h	t_e
S1 Dauer Bereich II	0.855	0.826
S1 Dauer Bereich III	0.971	0.954
S1 Dauer Bereich IV	0.988	0.981
S1 Maximum Bereich II	0.958	0.958
S1 Minimum Bereich III	0.997	0.997
S1 Maximum Bereich IV	1.000	1.000
S1 Summe Bereich II	0.890	0.893
S1 Summe Bereich III	0.978	0.976
S1 Summe Bereich IV	0.991	0.988
S1 Filterkreuzungen Bereich II	0.278	0.278
S1 Filterkreuzungen Bereich III	0.449	0.443
S1 Filterkreuzungen Bereich IV	0.513	0.464
S1 Stand.abw Bereich II	0.650	0.780
S1 Stand.abw Bereich III	0.812	0.934
S1 Stand.abw Bereich IV	0.879	0.969
S1 Trendmaximum Bereich II	0.851	0.851
S1 Trendminimum Bereich III	0.948	0.948
S1 Trendmaximum Bereich IV	0.981	0.981

Tabelle 7.19: Merkmalsrelevanzen MANOVA (M_T , sukzessive Auswahl), Vergleich t_h zu t_e , ausschließliche Betrachtung von drei Merkmalen der gleichen Kategorie, Benchmarkdatensatz B

Position	Merkmal	Güte ($M^{\xi_G, \text{mod}}$)
1	S1 rel. Minimum Bereich III	4226.414
2	S1 Maximum Bereich II	1205.433
3	S1 Maximum Bereich IV	682.746
4	(S1-S2) Maximum Bereich IV	348.660
5	(S1-S2) Summe Bereich II	203.381
6	S1 Summe Bereich IV	134.409
7	S2 Trendmaximum Bereich IV	116.431
8	(S1-S2) Trendminimum Bereich III	107.505
9	(S1-S2) Minimum Bereich III	99.697
10	S1 Summe Bereich III	95.293

Tabelle 7.20: Merkmalsrelevanzen MM-Verfahren, multivariat, Benchmarkdatensatz B

Position	Merkmal	Güte ($M^{\xi_G, \text{mod}}$)
1	S1 rel. Minimum Bereich III	4226.414
2	S1 Maximum Bereich IV	5893.126
3	S1 Summe Bereich IV	6503.505
4	S1 Minimum Bereich III	6605.273
5	S2 Maximum Bereich II	6660.559
6	S1 Trendminimum Bereich III	6765.245
7	S1 Summe Bereich III	6807.176
8	S1 Maximum Bereich II	6892.105
9	(S1-S2) Maximum Bereich IV	7014.839
10	(S1-S2) Summe Bereich IV	7028.000

Tabelle 7.21: Merkmalsrelevanzen MM-Verfahren, univariat, Benchmarkdatensatz B

Bildverzeichnis

1.1	Mensch-Maschine-Kommunikation	2
1.2	Steuerungs- bzw. Regelungskonzept für technische Systeme	3
1.3	Ablauf der Zielmotorik	4
1.4	Neuromuskuläres System	5
1.5	Ableitung und Stimulation an einer Nervenfasern mittels ENG-Elektrode	9
1.6	Bipolares EMG-Signal von musculus extensor digitorum	14
2.1	Manueller und automatischer Entwurf von biologischen Mensch-Maschine-Schnittstellen	20
2.2	Zustandsautomat für ein MMI	23
2.3	Eingeschränkter Zustandsautomat für ein MMI	24
2.4	Schema zur Klassifikation von Schaltsignalen	24
2.5	Klassifikatorentwurfsverfahren	26
2.6	Bewertungsfunktionen zur Berechnung des MMI-Index	30
2.7	Realisierungen eines Benchmarkschaltsignaltyps	32
2.8	Generierte Zeitreihen und erzeugte Merkmale	36
2.9	Univariate Analyse der Varianzen anhand des Benchmarkdatensatzes	39
2.10	Reduktion von Benchmarkdatensatz A auf zwei Dimensionen mittels HKA	41
2.11	Reduktion von Benchmarkdatensatz A auf zwei Dimensionen mittels LDA	41
2.12	Fuzzy-Regel im Merkmalsraum	44
2.13	Diskriminanzfunktionen einer Fuzzy-Regelbasis	44
2.14	Aufbau eines einzelnen Neurons	45
2.15	Schematischer Aufbau eines Künstlichen Neuronalen Netzes	45
2.16	Diskriminanzfunktionen eines MLPs	47
2.17	Diskriminanzfunktionen eines überlernten MLPs	47
2.18	Diskriminanzfunktionen eines k-NN-Klassifikators	48
2.19	Diskriminanzfunktionen einer SVM	51
2.20	Diskriminanzfunktionen eines AML-Klassifikators	53
2.21	Wahrscheinlichkeitsdichten und Diskriminanzfunktionen im Merkmalsraum	53
2.22	Klassifikationsfehler in Abhängigkeit von Lerndatensatzumfang und Entscheidungsregel	57
3.1	Klassifikationsproblematik im aggregierten Merkmalsraum	63

3.2	Möglichkeiten der Modifikation des Klassifikationsschemas	63
3.3	Funktion zur Wichtung des Klassifikationsfehlers	65
3.4	Auswahl von 2 Merkmalen gemäß MANOVA	66
3.5	Auswahl von 2 Merkmalen gemäß MM	66
3.6	Transformierter Merkmalsraum durch LDA	68
3.7	Transformierter Merkmalsraum durch MD	68
3.8	Diskriminanzfunktionen eines AML-Klassifikators	69
3.9	Diskriminanzfunktionen eines MAML-Klassifikators	69
3.10	Vorgehensweise bei Mehrschrittverfahren	71
3.11	Algorithmus zum modifizierten Klassifikatorentwurf	72
3.12	Hierarchischer Klassifikatorentwurf zum Abtrennen von Klassen	73
3.13	Algorithmus zum hierarchischen Klassifikatorentwurf nach HKK	76
3.14	Hierarchischer Klassifikatorentwurf zur Extraktion von Datentupeln	78
3.15	Hierarchische Klassifikation von Datentupeln	79
3.16	Algorithmus zum hierarchischen Klassifikatorentwurf nach HKT	81
3.17	Hierarchischer Klassifikatorentwurf zur Turnier- und KO-Klassifikation	83
3.18	Vorgehen zur Klassifikation mittels Turnier- und KO-Klassifikatoren	84
4.1	Struktur der Implementierung	92
4.2	Schematische Darstellung zum Entwurf von MMIs	93
4.3	Prozessschritte und Signalfluss zum Entwurf einer patientenindividuellen Steuerung	93
4.4	Schema des Entwurfsvorgangs	94
4.5	DAVE: Plattform zur Auslegung und Evaluierung von MMIs	95
4.6	DAVE-Design: Plattform zum Entwurf von Klassifikatoren	96
4.7	Integration der Module in die T&E-Box	97
4.8	Automat zur Berechnung von Merkmalen im zeitdiskreten Signal	99
4.9	Abhängigkeiten der Merkmalsgenerierung	100
4.10	Patientenindividuelle Quelltext-Implementierung	101
5.1	Umsetzung von Griffarten in einer Prothese	104
5.2	Entkopplung von Bewegungsauswahl und -steuerung	105
5.3	Zustandsautomat zur Umsetzung von k Griffarten	106
5.4	Steuerung zur Umsetzung von fünf Griffarten	107
5.5	Myoelektrisches Signal zur Umsetzung von Bewegungsmustern	108
5.6	Ventilzusammengehörigkeiten einer Prothese	109
5.7	Bewegungssignale von zwei EMG-Sensoren	110
5.8	Steuersignal zum Öffnen und Schließen	110
5.9	Zeitgesteuerte Implementierung einer Griffart	111
5.10	Dauer der Pumpenspannung in ms zur Ausführung von Griffarten	111

5.11	Vorgehensweise bei Anlernsitzungen und Validierungssitzungen	112
5.12	Klassifikationsfehler in Abhängigkeit vom Umfang des Lerndatensatzes	116
5.13	Darstellung von Lern- und Testdaten, ohne robusten Entwurf, Patient F	120
5.14	Darstellung von Lern- und Testdaten, ohne robusten Entwurf, Proband S	120
5.15	Problematik des nicht-robusten Entwurfs	121
5.16	Robustes und nicht-robustes Schaltsignal	121
5.17	Darstellung von Lern- und Testdaten, robuster Entwurf, Patient F	122
5.18	Darstellung von Lern- und Testdaten, robuster Entwurf, Proband S	122
5.19	Klassifikationsfehler im Echtzeitbetrieb, ohne robusten Entwurf	123
5.20	Klassifikationsfehler im Echtzeitbetrieb, robuster Entwurf	123
5.21	Patient mit Prothese, Lateralgriff	124
5.22	Patient mit Prothese, Zylindergriff	124
5.23	Automatengraph eines virtuellen Keyboards	125
5.24	MANOVA-Koeffizienten, alle Merkmale	128
5.25	Klassifikationsgüte, alle Merkmale, Testdaten	128
5.26	Fuzzy Regel für EEG-Daten	128
7.1	Erstellung eines künstlichen Schaltsignals	147
7.2	Simulierte Realisierungen von 8 Schaltsignalkategorien	148
7.3	Originales Schaltsignal, zeit- und amplitudengedehnte Schaltsignale	150
7.4	Konfiguration zur Evaluierung der Robustheit von Klassifikatoren	151
7.5	Schnelle Fuzzifizierung von dreiecks- und trapezförmigen ZGF	161

Tabellenverzeichnis

1.1	Frequenzen und Amplituden bioelektrischer Signale	7
1.2	Bewertung der Signalqualität muskulärer Signale	11
1.3	Forschungsschwerpunkte der myoelektrischen Prothetik	16
2.1	Bewertung des Zeitpunktes der Merkmalsgenerierung	55
2.2	Klassifikationsfehler verschiedener Entscheidungsregeln	58
2.3	Dauer zur Auswahl von Merkmalen	58
2.4	Dauer zum Anlernen verschiedener Entscheidungsregeln	59
2.5	Anzahl Multiplikationen/Speicherplätze verschiedener Entscheidungsregeln	59
2.6	Auswertung des MMI-Index für verschiedene Entscheidungsregeln	60
2.7	Klassifikationsfehler der modifizierten Verfahren	60
2.8	Bewertung von Entscheidungsregeln	61
3.1	Klassifikationsfehler in Abhängigkeit von der Merkmalsauswahl	66
3.2	Klassifikationsfehler in Abhängigkeit von der Merkmalsaggregation	68
3.3	Klassifikationsfehler von AML- und MAML-Verfahren	70
3.4	Klassifikationsfehler des HKK-Verfahrens	77
3.5	Klassifikationsfehler des HKT-Verfahrens	82
3.6	Klassifikationsfehler der TN- und KO-Verfahren	86
3.7	Herkömmlicher Entwurf bei Benchmarkdaten	87
3.8	Robuster Entwurf bei Benchmarkdaten durch Erweitern	87
3.9	Robuster Entwurf bei Benchmarkdaten über den Sitzungen B und D ₂	87
3.10	Problemlösungsstrategien	88
3.11	Klassifikationsfehler der modifizierten Verfahren	89
3.12	Dauer zum Anlernen der modifizierten Verfahren	89
3.13	Anzahl Multiplikationen/Speicherplätze der modifizierten Verfahren	90
3.14	Auswertung des MMI-Index	91
3.15	Klassifikationsfehler der modifizierten Verfahren	91
5.1	Patientenbeschreibung	113
5.2	Schaltsignaltypen und Schaltsignal dauern	114

5.3	Klassifikationsfehler mit und ohne Training	115
5.4	Klassifikationsfehler über Patientendaten	117
5.5	Klassifikationsfehler bei Patient F ohne robusten Entwurf	119
5.6	Klassifikationsfehler bei Proband S ohne robusten Entwurf	119
5.7	Klassifikationsfehler bei Patient F mit robustem Entwurf	121
5.8	Klassifikationsfehler bei Proband S mit robustem Entwurf	121
5.9	MANOVA-Relevanzen für den BCI-Entwurf	126
5.10	Ergebnisse über BCI-Daten	127
5.11	Ergebnisse der modifizierten Verfahren über dem EEG-Datensatz	127
7.1	Parameter für Schaltsignalklasse a (Signal 1 und Signal 2 synchronisiert)	147
7.2	Parameter für Schaltsignal b (Signal 1 und Signal 2 nicht synchronisiert)	147
7.3	Parameter für Schaltsignal c (Signal 1 und Signal 2 nicht synchronisiert)	149
7.4	Parameter für Schaltsignal d (Signal 1 und Signal 2 synchronisiert)	149
7.5	Parameter für Schaltsignal e (Signal 1 und Signal 2 synchronisiert)	149
7.6	Parameter für Schaltsignal f (Signal 1 und Signal 2 synchronisiert)	149
7.7	Parameter für Schaltsignal g (Signal 1 und Signal 2 synchronisiert)	150
7.8	Parameter für Schaltsignal h (Signal 1 und Signal 2 synchronisiert)	150
7.9	Klassifikationsfehler diverser Entscheidungsregeln über Patientendaten	152
7.10	Klassifikationsfehler der modifizierten Verfahren mit $s_m = 8$, $s_d = 3$	153
7.11	Klassifikationsfehler der modifizierten Verfahren mit $s_m = 8$, $s_d = 4$	154
7.12	Klassifikationsfehler für in der Literatur übliche Benchmarkdatensätze	155
7.13	Visual C++ Source Code Files für DAVE	156
7.14	C Source Code Files für Mikrocontroller	156
7.15	MATLAB m-Files für DAVE-Design	160
7.16	Merkmalsrelevanzen ANOVA ($M_{\mathcal{I}}$), Benchmarkdatensatz B	163
7.17	Merkmalsrelevanzen MANOVA ($M_{\mathcal{I}}$, sukzessive Auswahl), Benchmarkdatensatz B	163
7.18	Merkmalsrelevanzen ANOVA ($M_{\mathcal{I}}$), Vergleich t_h zu t_e , Benchmarkdatensatz B	164
7.19	Merkmalsrelevanzen MANOVA, Vergleich t_h zu t_e	164
7.20	Merkmalsrelevanzen MM-Verfahren, multivariat, Benchmarkdatensatz B	165
7.21	Merkmalsrelevanzen MM-Verfahren, univariat, Benchmarkdatensatz B	165

Index

- Ähnlichkeitsmaß, 74
- Übersprechen, 14
- 632-Bootstrap-Methode, 54

- AAR, 125
- Akkumulation, 43
- Aktionszustand, 22
- Aktivierung, 43
- Aktivierungsfunktion, 45
- Aktivitätssignal, 31
- AML-Klassifikator, 52
- AML-Klassifikator, modifiziert, 69
- AML-Regel, 52
- Amplitudendehnung, 146
- Anlernen, herkömmlich, 85
- Anlernen, robust, 85
- Anlernsitzung, 111
- ANOVA, 55
- Antriebssteuerung, 110
- Anzahl umsetzbarer Zustände, 28, 29
- Appr. Maximum-Likelihood-Schätzung, 51
- Auslass-Methode, 54

- Backpropagation, 46
- Bandleistung, 125
- Basisimplementierung, 92
- BCI-Competition, 125
- BCI-Entwurf, 124
- Benchmarkentwurf, 146
- Benchmarkergebnis, 152
- Benchmarking, 31, 85
- bestLDF, 83
- Betriebseinheit, 92
- Bewegung, programmgesteuert, 4
- Bewegung, reflexgesteuert, 4
- Bewegungsdetektion, 7
- Bewegungsplanung, 4
- Bewegungssignal, 105
- Bewegungsstrategie, Entwurf, 4

- Bewegungszustand, 106
- Bhattacharyya, 74
- Bootstrap, 54
- BP, 125
- Brain-Computer-Interface, 124

- Crossvalidierung, 54

- Datenerfassung, 95
- Datensatzumfang, 56, 115
- DAVE, 94, 156
- DAVE-Design, 95, 157
- Defuzzifizierung, 43
- Dichotomie, 70
- Diskriminanzanalyse, linear, 40
- Diskriminanzanalyse, modifiziert, 67
- Distanz, 53
- Divergenz, 74
- Divergenzmatrix, 74

- Echtzeit, 118
- Echtzeitbetrieb, 118
- Echtzeitfähigkeit, 21, 27
- ECoG, 8
- EEG, 7
- Eigenwertproblem, 40
- Einschrittverfahren, 64, 118
- Einzelmerkmal, 33
- Elektroencephalographie, 7
- Elektrokortikographie, 8
- Elektromyographie, 9
- Elektroneurographie, 8
- Elektrostimulation, funktionell, 9
- EMG, 9
- ENG, 8
- Engineering-Kosten, 28, 29
- Entscheidungsbereich, 41
- Entscheidungsmaß, 24, 41
- Entscheidungsregel, 25, 41

- Entschlussphase, 4
- Entwurf, robust, 85, 120, 146
- Entwurfseinheit, 92
- Entwurfsprozess, 94
- Entwurfsverfahren, 26
- euklidische Distanz, 52

- FES, 9
- Filter-Ansatz, 38
- Fluidaktor, 109
- Freiheitsgrad, 109
- Fuzzifizierung, 43
- Fuzzy-Logik, 42
- Fuzzy-Regelbasis, 42, 160

- Güte der Klassenzuweisung, 28
- Geschwindigkeitssteuerung, 110
- Griff, sphärisch, 103

- Hakengriff, 103
- Handprothese, myoelektrisch, 12, 102
- Hauptkomponentenanalyse, 40
- Hierarchie, 70
- Hierarchische Klassifikation nach Datentupeln, 77
- Hierarchische Klassifikation nach Klassen, 72
- HKK-Verfahren, 72
- HKT-Verfahren, 77

- Implementierung, 92
- Implementierung, anwendungsspezifisch, 92
- Implementierung, patientenindividuell, 92
- Inferenz, 43
- Inferenzstrategie, 43
- Innerklassenvariationsmatrix, 40
- Intervalltransformation, 37
- IRIS, 154

- k-Nearest-Neighbour-Klassifikator, 47
- Künstliches Neuronales Netz, 45
- Kern, 50
- Klassenkombination, paarweise, 50, 82
- Klassifikationsergebnis, 152
- Klassifikationsgüte, 28, 116
- Klassifikationskosten, 67, 76
- Klassifikationssystem, 24
- Klassifikator, 24
- Klassifikatorbewertung, qualitativ, 60
- Klassifikatorbewertung, quantitativ, 53
- KO-Klassifikation, 82

- Kokontraktion, 15, 108
- Kombination, paarweise, 70
- Konfidenzintervall, 62
- Konfusionsmatrix, 54
- kongenital, 103
- Konvertierungsroutine, 96
- Kostenoptimierung, 67, 76
- Kullback-Leibler, 73

- Lagesteuerung, 110
- Lateralgriff, 103
- LDA, 40
- Leave-One-Out, 54
- Lerndatensatz, 42
- Lerndatensatz, erweitern, 85
- Lerndatensatz, Umfang, 56, 115
- Likelihood-Quotienten-Kriterium, 39
- Locked-In-Syndrom, 6, 11, 124

- Mahalanobis Distanz, 48, 53
- MAML-Verfahren, 69
- MANOVA, 38, 55
- MD-Verfahren, 67
- Mehrschrittverfahren, 64, 70
- Mensch-Maschine-Schnittstelle, 2
- Merkmalsaggregation, 24, 40
- Merkmalsaggregation, modifiziert, 67
- Merkmalsauswahl, 24
- Merkmalsauswahl, modifiziert, 64
- Merkmalsberechnung, 99
- Merkmals erzeugung, Zeitpunkt, 55
- Merkmalsextraktion, 24, 33, 98
- Merkmalsrelevanzen, 163
- Merkmalsrelevanzen, a-priori, 85
- Merkmalsätze, 55
- Merkmalsselektion, 37
- Metrik, 47, 52, 69
- Metrik, Modifikation, 69
- Mikrocontroller, 97, 156
- Mikrocontroller-Kategorie, 28, 29
- Mikrocontroller-Umgebung, 97
- MLP, 46
- MM-Verfahren, 64
- MMI, 2
- MMI-Index, 28, 90, 116
- Modell, autoregressiv, 125
- Modell, nichtparametrisch, 27
- Modell, parametrisch, 26

- Modell, semiparametrisch, 27
- Modifikation, 62
- Multilayer-Perzeptron-Netz, 46
- Multiplikationen, Anzahl, 29

- Nearest-Neighbour-Klassifikator, 47
- Netz, Künstliches Neuronales, 45
- Normalverteilung, 52

- one-against-all, 50, 82
- one-against-one, 50, 82
- Online-Berechnung, 98

- pairwise coupling, 70
- parametervariabel, 100
- Parametrierung, 130
- Patientenbeschreibung, 112
- Patientendatensatz, 116, 152
- Patientenkollektiv, 111
- Perzeptron-Neuron, 46
- Physiologie, 4
- Pinzettengriff, 103
- Polychotomie, 70
- Polynomkern, 50
- Prämissenauswertung, 43
- Preshape, 106
- Problem, patientenindividuell, 113
- Problemlösungsstrategie, 88
- Projektion, 67
- Projektionsmatrix, 40, 67
- Proportionalsteuerung, 110

- Quelltextgenerierung, 98, 100, 160

- Rückwärtssuche, 38
- Radial-Basis-Funktion-Neuron, 46
- RBF, 46
- Reklassifikation, 54
- Reproduzierbarkeit, 113
- Robustheit, 118

- Schalten mit Zeitversatz, 22
- Schalten ohne Zeitversatz, 22
- Schaltsignal, 22, 105
- Schaltsignal, abstrakt, 105
- Schaltsignal, amplitudenvariiert, 105
- Schaltsignal, ereignisbasiert, 22
- Schaltsignal, gestört, 146
- Schaltsignal, intuitiv, 105
- Schaltsignal, kombiniert, 105
- Schaltsignal, zeitpunktbasiert, 22, 124
- Schaltsignal, zeitvariiert, 104
- Schaltsignaldauer, 113
- Schaltsignalkategorie, 104
- Schaltsignaltyp, 104
- Schnittpunkte, Anzahl, 36
- Separate-and-Conquer, 72
- Separationsmaß, 74
- Sequential-Minimal-Optimization, 49
- Signal, zeitvariant, 85
- Simulationsplattform, 95
- Speicherplätze, Anzahl, 29
- Spezifikation, 108
- Standardabweichung, gefiltert, 37
- Steuerungseinheit, 108
- Steuerungskonzept, 3
- strukturvariabel, 100
- Support-Vektor-Maschine, 48

- T&E-Box, 97
- Teile-und-Herrsche, 72
- Testen über Lerndaten, 54
- Testen über Testdaten, 54
- TN-Klassifikation, 82
- Training, 115
- Trainingsaufwand, 28
- Trainingsplattform, 95
- Transformationsmatrix, 25, 40, 67
- Trend, 36
- Trenngüte, multivariat, 55
- Trenngüte, univariat, 55
- Turnier-Klassifikation, 82

- Validierungssitzung, 111
- Validierungstechnik, 53
- Ventilzusammengehörigkeit, 109
- Versuchsdesign, 111
- Vorbewegungszustand, 106
- Vorwärtssuche, 38

- Wavelet-Methoden, 129
- Willkürbewegung, 4
- WINE, 154
- Wrapper, 62
- Wrapper-Ansatz, 38

- Zeigefingerstellung, 103

Zeit-Frequenz-Transformation, 33
Zeitdehnung, 146
Zeitreihenmerkmal, 33
Ziele, 18
Zielmotorik, 4
ZOO, 154
Zugehörigkeitsfunktion, 43
Zustandsautomat, 22, 106
Zustandsautomat, vereinfacht, 23
Zwischenklassenvariationsmatrix, 40
Zylindergriff, 103