Available online at www.sciencedirect.com

ScienceDirect

Procedia IUTAM 2 (2011) 275–289

Procedia IUTAM

www.elsevier.com/locate/procedia

2011 Symposium on Human Body Dynamics

Modeling and synthesis of human motion within the collaborative research center 588

Christian Simonidis[a], Günther Stelzner[a], Wolfgang Seemann[a,*], Fabian Bauer[a]

[a]*Karlsruhe Institute of Technology (KIT), Institute of Engineering Mechanics (ITM), Kaiserstr. 10, 76131 Karlsruhe, Germany*

Abstract

Within the Collaborative Research Center "humanoid robots" at KIT there is a strong interest in generating human-like motion. On one hand large-scale experimental studies are performed to gather insight into the variety and complexity of human motion on the other hand models for the kinematic and dynamic analysis as well as for the synthesis of movement are developed in order to understand the principles of human motion. Here we present a computational approach combining model-based kinematic and dynamic analysis of experimental data together with the synthesis of motion related to cost functions to analyze motion generation in humans.

Peer-review under responsibility of John McPhee and József Kövecses

Keywords: Motion capture; humanoid robot; motion synthesis; optimality

1. Introduction

The fast growing field of humanoid robotics produces a large variety of different robotic solutions and the systems get more and more complex and the simulation of the dynamics of the system becomes necessary not only for the design of the robot, e.g. control design, actuation forces, stress in components and contact forces but also for the understanding of human motion in order to transfer the motion and produce human-like robotic systems. Commercial software tools allow for the fast and comfortable analysis of mechanical systems, but their functional range is limited when it comes to the adoption of a custom solution, where source code manipulation is required, and particular dynamical routines are implemented.

In the collaborative research center 588 in Karlsruhe, a mechanics simulation environment was required in order to analyze existing humanoid robots and simulate new developments. Then, the environment was adopted to the mapping process of kinematical measurement data of a human subject onto the humanoid robot. Meanwhile, the software environment increased to a general mechanics simulation tool and, besides inverse and forward kinematical and dynamical analysis of individually defined models, it provides optimization based procedures for the integration of kinematical and dynamical measurement data into the model, e.g. data from motion capture systems. Control

*Corresponding author. Tel: +49-721-608-46824; fax: +49 721 608-46070
Email address: seemann@kit.edu (Wolfgang Seemann)

2210-9838
doi:10.1016/j.piutam.2011.04.025

approaches have been implemented and the system allows for the design of model based controllers as well as for testing simulations of existing controls. Another feature is the computation and analysis of optimal control problems, which are studied in order to understand human motion and synthesize human-like motion for humanoid robots.

Firstly, the multibody approach and its implementation is presented following the basics and application examples of individual joint kinematics. Then, the optimization based inverse kinematics approach and the related marker based mapping is introduced following the optimization based inverse dynamics approach. Examples of control implementation and ease of controller design is shown subsequently. The numerical solution of optimal control problems and motion generation based on several optimization criteria is presented. It will be shown that in general the motion due to minimal torque, minimal torque change and other optimization functions lead to motions which ressemble the measured motion. However it can be expected that the real motion depends on each subject in such a way that it is a combination of the above mentioned criteria. The contribution closes with discussion and future work.

2. Multibody approach

The general approach for kinematical and dynamical analysis of robotic and human-like systems is the multibody approach [1, 2] and a large variety of formulations with different types of coordinate representation (e.g. natural coordinates, relative coordinates) and derivation of the equations of motion are available. This approach aims on an efficient numerical generation of the kinematics of the system and the equations of motion and it was designed for use with the Matlab environment and its powerful ODE suite. Therefore, the equations of motion are generated in minimal coordinate form with recursive computational procedures [3]. In the following the basic underlying formulation is presented.

Considering two bodies i and j of a rigid body chain with local coordinate systems K_i and K_j, their absolute position is described by $\mathbf{R}_i$ and $\mathbf{R}_j$ in relation to the inertial system K_0 and their relative position is defined by $\mathbf{r}_{ij}$. The transformation matrices $\mathbf{A}_i$, $\mathbf{A}_j$ and $\mathbf{A}_{ij}$ describe absolute and relative orientation respectively (Figure 1). Therefore, the recursion equations for position and orientation are given by

$$\mathbf{R}_j = \mathbf{R}_i + \mathbf{A}_i \mathbf{r}_{ij}^i, \tag{1}$$

$$\mathbf{A}_j = \mathbf{A}_i \mathbf{A}_{ij}. \tag{2}$$

Keep in mind that $\mathbf{r}_{ij}$ and $\mathbf{A}_{ij}$ depend from defined (and independent) joint coordinates $\mathbf{q}_{ij}$ of the joint G_{ij}. This context will be described below. The recursion equations of velocity and angular velocity are derived by the time derivatives of equations (1) and (2) to

$$\dot{\mathbf{R}}_j = \dot{\mathbf{R}}_i + \tilde{\boldsymbol{\omega}}_i \mathbf{r}_{ij} + \mathbf{A}_i \dot{\mathbf{r}}_{ij}^i, \tag{3}$$

$$\boldsymbol{\omega}_j = \boldsymbol{\omega}_i + \mathbf{A}_i \boldsymbol{\omega}_{ij}^i = \boldsymbol{\omega}_i + \boldsymbol{\omega}_{ij}, \tag{4}$$

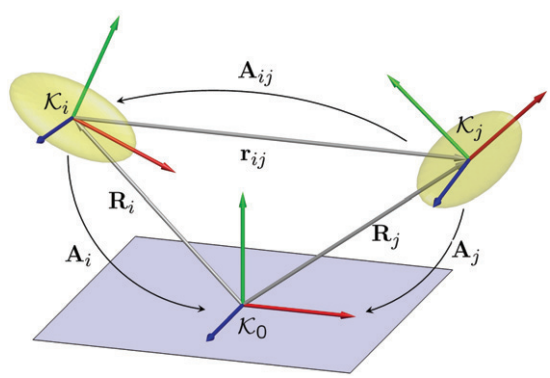


Fig. 1: Definition of absolute and relative kinematics

where $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_j$ are the absolute angular velocities of the respective body, $\boldsymbol{\omega}_{ij}$ is the relative angular velocity and $\tilde{\boldsymbol{\omega}}$ denotes the skew symmetric matrix. Still, the relative velocities $\mathbf{r}_{ij}$ and $\boldsymbol{\omega}_{ij}$ depend on the defined general coordinates of joint G_{ij}. Analogously, the acceleration and angular acceleration are obtained from the equations (3) and (4) to

$$\ddot{\mathbf{R}}_j = \ddot{\mathbf{R}}_i - \tilde{\mathbf{r}}_{ij}\dot{\boldsymbol{\omega}}_i + \tilde{\boldsymbol{\omega}}_i\tilde{\boldsymbol{\omega}}_i\mathbf{r}_{ij} + 2\tilde{\boldsymbol{\omega}}_i\mathbf{A}_i\dot{\mathbf{r}}^i_{ij} + \mathbf{A}_i\ddot{\mathbf{r}}^i_{ij}, \tag{5}$$

$$\dot{\boldsymbol{\omega}}_j = \dot{\boldsymbol{\omega}}_i + \tilde{\boldsymbol{\omega}}_i\boldsymbol{\omega}_{ij} + \mathbf{A}_i\dot{\boldsymbol{\omega}}^i_{ij}. \tag{6}$$

and can be rearranged to

$$\underbrace{\begin{bmatrix}\ddot{\mathbf{R}}_j\\ \dot{\boldsymbol{\omega}}_j\end{bmatrix}}_{\ddot{\boldsymbol{\alpha}}_j} = \underbrace{\begin{bmatrix}\mathbf{I} & -\tilde{\mathbf{r}}_{ij}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}}_{\boldsymbol{\Phi}_{ij}}\underbrace{\begin{bmatrix}\ddot{\mathbf{R}}_i\\ \dot{\boldsymbol{\omega}}_i\end{bmatrix}}_{\ddot{\boldsymbol{\alpha}}_i} + \underbrace{\begin{bmatrix}\mathbf{A}_i\mathbf{J}^r_{ij}\\ \mathbf{A}_i\mathbf{J}^\omega_{ij}\end{bmatrix}}_{\mathbf{J}_{ij}}\ddot{\mathbf{q}}_{ij} + \underbrace{\begin{bmatrix}\tilde{\boldsymbol{\omega}}_i\tilde{\boldsymbol{\omega}}_i\mathbf{r}_{ij} + 2\tilde{\boldsymbol{\omega}}_i\mathbf{A}_i\dot{\mathbf{r}}^i_{ij} + \mathbf{A}_i\boldsymbol{\Gamma}^r_{ij}\\ \tilde{\boldsymbol{\omega}}_i\boldsymbol{\omega}_{ij} + \mathbf{A}_i\boldsymbol{\Gamma}^\omega_{ij}\end{bmatrix}}_{\boldsymbol{\Theta}_{ij}} \tag{7}$$

and to the following short formulation respectively.

$$\ddot{\boldsymbol{\alpha}}_j = \boldsymbol{\Phi}_{ij}\ddot{\boldsymbol{\alpha}}_i + \mathbf{J}_{ij}\ddot{\mathbf{q}}_{ij} + \boldsymbol{\Theta}_{ij} \tag{8}$$

From the recursion formulation over n bodies $j = 1, ..., n$ the constraint equations are obtained in explicit matrix form

$$\ddot{\boldsymbol{\alpha}} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \boldsymbol{\Gamma}(\mathbf{q}, \dot{\mathbf{q}}), \tag{9}$$

where $\mathbf{J}$ is the Jacobian of the system and $\boldsymbol{\Gamma}$ are quadratic terms in joint coordinates $\mathbf{q}$ and derivatives $\dot{\mathbf{q}}$.

The equations of motion are derived with the principle of virtual work. The mass matrix of body i is defined by

$$\mathbf{M}_i = \begin{bmatrix} m_i\mathbf{E} & -m_i\tilde{\mathbf{r}}_S\\ m_i\tilde{\mathbf{r}}_S & \mathbf{I}_i\end{bmatrix}, \tag{10}$$

where E is the identity matrix and I is the inertial tensor of i. The vector of coriolis and centrifugal forces is defined by

$$\mathbf{Q}^Z_i = \begin{bmatrix} m_i\tilde{\boldsymbol{\omega}}_i\tilde{\boldsymbol{\omega}}_i\mathbf{r}_S\\ \tilde{\boldsymbol{\omega}}_i\mathbf{I}_i\boldsymbol{\omega}_i\end{bmatrix}. \tag{11}$$

External forces and torques on body i are defined as coordinate vector in absolute space

$$\mathbf{Q}^E_i = \begin{bmatrix} \sum_n^{n_F}\mathbf{F}_{i,n}\\ \sum_n^{n_F}\tilde{\mathbf{r}}_{i,n}\mathbf{F}_{i,n} + \sum_m^{n_M}\mathbf{T}_{i,m}\end{bmatrix}, \tag{12}$$

where $\mathbf{F}_i$ is the force component and $\mathbf{T}_i$ is the torque component in absolute coordinates respectively.

Linear and nonlinear spring and damper forces as well as muscle forces act between two arbitrary bodies K_i and K_j of the multibody system (Figure 2). The coordinates of vector $\mathbf{r}_{ij}$, which relates the force application points P_i and P_j are obtained by

$$\mathbf{r}_{ij} = \mathbf{R}_j + \mathbf{A}_j\mathbf{r}^j_j - \mathbf{R}_i - \mathbf{A}_i\mathbf{r}^i_i. \tag{13}$$

and the action of a viscoelastic system is formulated by a scalar function

$$F_{ve} = F\left(\ell, \dot{\ell}, t\right) \tag{14}$$

and the viscoelastic forces and moments on body i and j are given by the product of action and coordinate vector to

$$\mathbf{Q}^F_i = \frac{F_{ve}}{\ell}\begin{bmatrix}\mathbf{r}_{ij}\\ \tilde{\mathbf{r}}_i\mathbf{r}_{ij}\end{bmatrix} \text{ and } \mathbf{Q}^F_j = -\frac{F_{ve}}{\ell}\begin{bmatrix}\mathbf{r}_{ij}\\ \tilde{\mathbf{r}}_j\mathbf{r}_{ij}\end{bmatrix}. \tag{15}$$

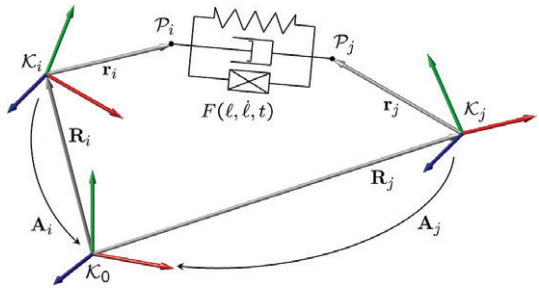


Fig. 2: Spring, damper and actuator forces

Rotational spring and damper torques between two bodies i and j can be obtained respectively and they are represented in absolute coordinates as

$$\mathbf{Q}_i^M = \begin{bmatrix} \mathbf{0} \\ T_{ve}\mathbf{u} \end{bmatrix} \text{ und } \mathbf{Q}_j^M = \begin{bmatrix} \mathbf{0} \\ -T_{ve}\mathbf{u} \end{bmatrix}, \tag{16}$$

where T_{ve} is the respective viscoelastic torque around the shared relative axis $\mathbf{u}$ of body i and j. Further, a vector $\mathbf{Q}_i^A$ is defined containing actuation forces and torques as well as joint limitations.

Then the equations of motion are derived with the principle of virtual work

$$\sum_i^{n_K} \delta\boldsymbol{\alpha}_i^T \left[\mathbf{M}_i\ddot{\boldsymbol{\alpha}}_i + \mathbf{Q}_i^Z - \mathbf{Q}_i^E - \mathbf{Q}_i^F - \mathbf{Q}_i^M - \mathbf{Q}_i^A\right] = 0. \tag{17}$$

to

$$\mathbf{J}^T\mathbf{M}\mathbf{J}\ddot{\mathbf{q}} = \mathbf{J}^T\mathbf{Q} - \mathbf{J}^T\mathbf{M}\boldsymbol{\Gamma}. \tag{18}$$

Equation (18) can be rewritten into the compact minimal coordinate form

$$\mathbf{M}_\mathbf{q}\ddot{\mathbf{q}} - \mathbf{Q}_\mathbf{q} = \mathbf{T}, \tag{19}$$

where $\mathbf{M}_\mathbf{q}$ is the mass matrix of the system, $\mathbf{Q}_\mathbf{q} = \mathbf{J}^T\left(\mathbf{Q}^Z + \mathbf{Q}^E + \mathbf{Q}^F + \mathbf{Q}^M\right)$ and $\mathbf{T} = \mathbf{J}^T\mathbf{Q}^A$ are the generalized forces acting on the generalized coordinates $\mathbf{q}$. Equation (19) is valid for multibody systems with tree structure. For systems with closed kinematic loops, the coordinates $\mathbf{q}$ are not independent any more. Therefore, the closing conditions are formulated in explicit form and dependent coordinates are eliminated, which results again in a system of ordinary differential equations. An analytical solution is possible only in a few certain cases and a numerical solution is usually desired. For the closed loop condition

$$\mathbf{c}\left(\mathbf{R}_i, \mathbf{A}_i, \mathbf{R}_j, \mathbf{A}_j\right) = \mathbf{0} \tag{20}$$

the constraint equation on acceleration level

$$\mathbf{C}_\mathbf{q}\ddot{\mathbf{q}} + \dot{\mathbf{C}}_\mathbf{q}\dot{\mathbf{q}} = \mathbf{0}$$

is derived with dependent coordinates $\mathbf{q}$ and a DAE-System

$$\begin{bmatrix} \mathbf{M}_\mathbf{q} & \mathbf{C}_\mathbf{q}^T \\ \mathbf{C}_\mathbf{q} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_\mathbf{q} \\ -\dot{\mathbf{C}}_\mathbf{q}\dot{\mathbf{q}} \end{bmatrix}.$$

is obtained. The transition to ordinary differential equations is then performed by explicit closing conditions.

3. Joint kinematics

In this section, the relative context of $\mathbf{r}_{ij}$, $\mathbf{A}_{ij}$ and the generalized coordinates $\mathbf{q}_{ij}$ of a joint G_{ij} between the bodies K_i and K_j is explained and examples are given. A joint can be individually defined and it is not limited to standard

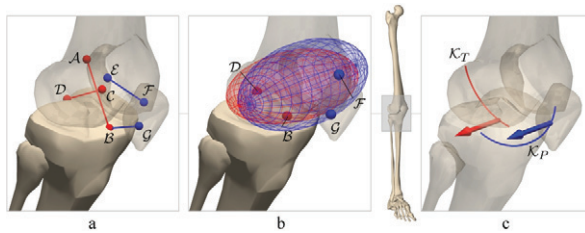


Fig. 3: Example: Individual kinematic definition of 1 dof knee joint, (a) four bar mechanism, (b) ellipsoid, (c) spline coupling

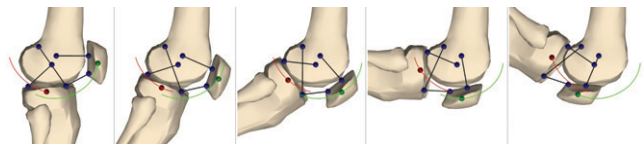


Fig. 4: Example: Different positions of the 1 dof knee joint modeled by cubic spline approximation

joints, e.g. rotational or spherical joints. The relative position and the relative orientation can be defined by a generally nonlinear functional context

$$\mathbf{r}_{ij}^{i} = \mathbf{f}^{r}\left(\mathbf{q}_{ij}\right) \tag{21}$$

$$\mathbf{w}_{ij} = \mathbf{f}^{w}\left(\mathbf{q}_{ij}\right) \tag{22}$$

This context allows one to define standard joints such as revolute, translational or spherical joints as well as any other desired joint behavior, e.g. coupling of coordinates $\mathbf{r}_{ij} = [r_{ij}^{x}(q), r_{ij}^{y}(q), r_{ij}^{z}(q)]^{T}$ of joint G_{ij} within one independent coordinate q or even coupling of variables between joints G_{ij} and G_{jk} with $\mathbf{r}_{ij}(q)$ and $\mathbf{r}_{jk}(q)$ with one common independent coordinate q. The maximum number of defined generalized coordinates is six per joint, which is related to the six independent directions and orientations in space.

3.1. Example: coupling in one joint

Figure 3 illustrates three kinematic models of a human knee joint based on biomedical data [4], where one independent joint coordinate q_1 couples the three dependent position coordinates $\mathbf{r}_{ij}(q) = [r_{ij}^{x}(q), r_{ij}^{y}(q), r_{ij}^{z}(q)]^{T}$. In a) a four bar mechanism is modeled with the formulation, in b) points fo the femur and the tibia body move relative on an elliptic surface and in c) the independent movement of femur and tibia have been first captured by a motion capture system and therefore, the six trajectories in $\mathbf{r}_{ij}(t)$ and $\mathbf{A}_{ij}(t)$ are known. Then, the independent joint coordinate q was introduced and the trajectories of $\mathbf{r}_{ij}(t)$ and $\mathbf{A}_{ij}(t)$ have been modeled by cubic spline approximation dependent on q (Figure 4).

3.2. Example: coupling of several joints

On the development of a humanoid spine, torso motion was studied with different kinds of kinematical models in order to find a suitable kinematical context (Figure 5) for human-like torso motion [5]. Therefore, a detailed spinal model with 24 vertebrae and 72 joint variables based on biomechanical data [6] was established and the variables where coupled to six independent joint coordinates $q_i,\ i = 1..6$ (Figure 6 a)). Further models (Figure 6 b)-e))have been investigated and a kinematical context for $\mathbf{r}_{ij}(t)$ and $\mathbf{A}_{ij}(t)$ was found which is able to reproduce human-like torso motion with only three independent joint coordinates $[q_1, q_2, q_3]$.

4. Optimization-based inverse kinematics

Inverse kinematic problems are commonly formulated as a system of nonlinear equations of the form

$$\begin{aligned} \mathbf{R}_{E}\left(\mathbf{q}\right) - \mathbf{R}_{Z} &= \mathbf{0}, \\ \mathbf{A}_{E}\left(\mathbf{q}\right) - \mathbf{A}_{Z} &= \mathbf{0}, \end{aligned} \tag{23}$$

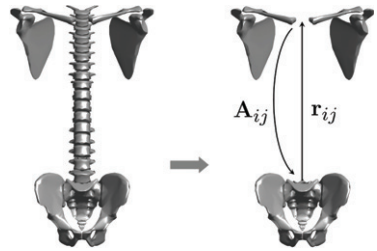


Fig. 5: Example: Mapping of torso motion

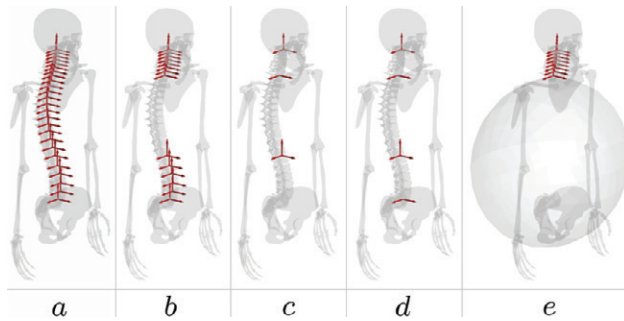


Fig. 6: Example: Different kinds of spinal models

where $\mathbf{R}_z$ and $\mathbf{A}_z$ are the desired end-effector position and orientation, $\mathbf{R}_E(\mathbf{q})$ and $\mathbf{A}_E(\mathbf{q})$ describes the absolute position and orientation of a coordinate system K_E. The solution of (23) are the generalized joint coordinates.

Robotic and human motion can be recorded by motion capture systems. A well known method is the infrared marker based approach, which captures the markers on the subject with several cameras and reconstructs the three-dimensional position of any marker in a defined absolute coordinate reference frame. The issue is now to transfer the captured motion on a defined multibody model. Among other methods the optimization based inverse kinematics approach is a method, which fits the model into the captured data while minimizing the residuals between model markers $\mathbf{R}_m$ and the corresponding measured marker position $\bar{\mathbf{R}}_m(t)$ (Figure 7). The marker $\mathcal{M}_m$ is defined by its relative position $\mathbf{r}_m^b$ with respect to the body-fixed reference frame of body K_b and its absolute position is obtained by

$$\mathbf{R}_m = \mathbf{R}_b + \mathbf{A}_b \mathbf{r}_m^b. \tag{24}$$

From the time derivative of equation (24) the Jacobian of marker $\mathcal{M}_m$ can be obtained, which is useful for gradient based optimization algorithms. The optimization problem can then be formulated by

$$\min_{\mathbf{q}} \tfrac{1}{2} \|\mathbf{F}(\mathbf{q})\|_2^2 \tag{25}$$

with the $n_M \times 1$ vector function

$$\mathbf{F}(\mathbf{q}) = \begin{bmatrix} \mathbf{f}_1(\mathbf{q}) \\ \vdots \\ \mathbf{f}_{n_M}(\mathbf{q}) \end{bmatrix}, \tag{26}$$

where n_M is the number of markers. The single vector functions $\mathbf{f}_m(\mathbf{q}) = \bar{\mathbf{R}}_m - \mathbf{R}_m(\mathbf{q})$ form the residual between the measured markers $\bar{\mathbf{R}}_m$ and the absolute position $\mathbf{R}_m(\mathbf{q})$ of the model marker $\mathcal{M}_m$. The coordinates $\mathbf{q}$ are then calculated by a nonlinear least-squares algorithm.

An optimization based approach is robust against poor marker placement and it is further useful for mapping on models with different kinematical structures, e.g. the same measured marker set can be used to map the motion on a human-like model with 106 dof as well as on a humanoid model with 24 dof. Figure 8 shows mapping results for a human-like and a humanoid system.

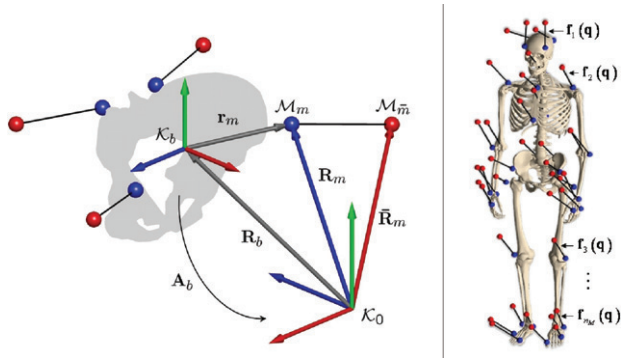


Fig. 7: Definition of marker and residuals

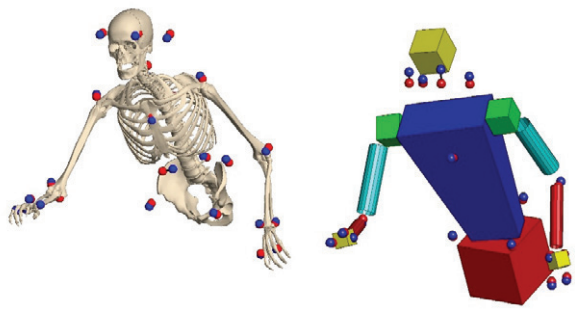


Fig. 8: Mapping the same measurement data on models with different number of dof

5. Optimization-based inverse dynamics

The inverse dynamics can be calculated if the kinematics of the system is determined, for example, the generalized forces of the system can be determined from motion capture data. But in case of additional available dynamical data such as force plate measurements or data of grip force sensors the inverse dynamical problem becomes over-determined. In the opposite case of several contact points to the environment, where the affecting contact forces are of interest, the inverse problem becomes even under-determined. Both problems appear in research of human and robotic systems. Considering the first case one may want to integrate both, kinematical and dynamical measurements into the system in order to keep the modeling error small [7] while avoiding possible appearing residual forces and moments on one body. The second case appears if one may want to know the applied forces and torques on several contact points with present kinematical measurements and only few or not sufficient dynamical measurements. Further, there is always a certain error in measurement, which might be considered, too.

The solution of the inverse dynamic problem in this approach is performed with control-based forward dynamics integration of an optimization based inverse dynamics solution based on weighted least squares similar to [7, 8, 9].

In the equation

$$\mathbf{M_q}\ddot{\mathbf{q}} - \mathbf{Q_q} = \mathbf{T} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{bmatrix} + \sum_{i=1}^{n_b} \mathbf{J}_i^{\mathbf{T}} \mathbf{I}_i^r \mathbf{F}_\mathbf{i}^\mathbf{6} \tag{27}$$

$n_\mathbf{q}$ is the number of dof in the system, the vector of generalized forces $\mathbf{T}$ is of size $n_\mathbf{q} \times 1$ and it is equal to the vector of actuator forces and torques $[\mathbf{0}^\mathbf{T}, \boldsymbol{\tau}^\mathbf{T}]^\mathbf{T}$ plus the sum of n_b external applied forces and torques, where $\mathbf{J}_i$ is the Jacobian of body i and $\mathbf{I}_i^r$ transforms the vector of measurements or applied forces $\mathbf{F}_i^\mathbf{6} = [M_x, M_y, M_z, F_x, F_y F_z]^\mathbf{T}$ on the system. The skew matrix $\tilde{\mathbf{r}}_i$ contains the elements of the relative vector $\mathbf{r}_i$ between local coordinate system K_i and

force application point on body i in absolute coordinates and $\mathbf{E}$ is the $\mathbf{3 \times 3}$ identity matrix.

$$\mathbf{I}_i^r = \begin{bmatrix} \mathbf{E} & \tilde{\mathbf{r}}_i \\ \mathbf{0} & \mathbf{E} \end{bmatrix}, \quad \mathbf{F}_i^{\mathbf{6}} = \begin{bmatrix} \mathbf{M}_i \\ \mathbf{F}_i \end{bmatrix} \tag{28}$$

The first six equations from the matrix equation (27) show a direct context between generalized coordinates and the applied forces which is expressed by

$$\mathbf{I}_c \mathbf{T} = \mathbf{I}_c \sum_{i=1}^{n_b} \mathbf{J}_i^{\mathbf{T}} \mathbf{I}_i^r \mathbf{F}_{\mathbf{i}}^{\mathbf{6}} \tag{29}$$

with $\mathbf{I}_c = [\mathbf{E}_{6\times6}, \mathbf{0}]^{\mathbf{T}}$. Rewriting equation (27) to

$$\ddot{\mathbf{q}} - \mathbf{M}_{\mathbf{q}}^{-1} \mathbf{Q}_{\mathbf{q}} = \mathbf{M}_{\mathbf{q}}^{-1} \mathbf{T} \tag{30}$$

the optimization problem can be formulated with equations (29) and (30) to

$$\underbrace{\begin{bmatrix} \mathbf{I}_c \\ \mathbf{M}_{\mathbf{q}}^{-1} \end{bmatrix}}_{\mathbf{A}} \underbrace{\mathbf{T}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{I}_c \sum\limits_{i=1}^{n_b} \mathbf{J}_i^{\mathbf{T}} \mathbf{I}_i^r \mathbf{F}_{\mathbf{i}}^{\mathbf{6}} \\ \ddot{\mathbf{q}} - \mathbf{M}_{\mathbf{q}}^{-1} \mathbf{Q}_{\mathbf{q}} \end{bmatrix}}_{\mathbf{b}} \tag{31}$$

Equation (31) has the basic form $\mathbf{Ax} = \mathbf{b}$ of a least squares problem for static optimization, where $\mathbf{A}$ is the in general non-quadratic pseudoinverse matrix and it can be solved for $\mathbf{x}$. The problem can be converted in a weighted least squares problem of the form

$$\mathbf{x} = \left(\mathbf{A}^{\mathbf{T}} \mathbf{W}^{-1} \mathbf{A} \right)^{-1} \mathbf{A}^{\mathbf{T}} \mathbf{W}^{-1} \mathbf{b} \tag{32}$$

with the time dependent covariance matrix $\mathbf{W}$, which may consist of the quality factors of the measurements and the optimal generalized forces of the system can be estimated for present kinematical measurements $\mathbf{q}, \dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ and dynamical measurements $\mathbf{F}^{\mathbf{6}}$ and even for imperfect or missing measurements.

The problem is now transferred to a dynamical optimization problem by introducing the control law

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_m + \mathbf{K}_v(\dot{\mathbf{q}} - \dot{\mathbf{q}}_{\mathbf{m}}) + \mathbf{K}_p(\mathbf{q} - \mathbf{q}_{\mathbf{m}}), \tag{33}$$

where the index m denotes data from kinematical measurements and $\mathbf{K}_v, \mathbf{K}_p$ are critically damped diagonal matrices. Then, the problem is solved by the integration of the controlled static optimization problem and the optimal kinematical and dynamical data are obtained.

The inverse problem of the optimal inverse dynamics solution is now solved for the distribution of the applied forces and torques for the contact points. Therefore, another pseudoinverse problem is solved which determines the optimal applied forces and torques with the minimum variance between calculated and measured dynamical data in presence of the covariance matrix $\mathbf{W}$. The optimal actuator forces and torques can be calculated from equation (31) with given optimal $\mathbf{T}$ and $\mathbf{F}^{\mathbf{6}}$. Subsequently to the rigid reconstruction a compensation for residuals in highly dynamic movements using wobbling masses can be performed, as described in [10].

5.1. Example: tap-dancing

Time synchronous kinematical and dynamical data were recorded for tap-dancing motion with a Vicon motion capture system and two AMTI force plates and integrated into the model with the optimization based inverse dynamics approach. Figure 9 shows the assembly of a trial in the simulation. The left and the right force plate can be seen with a reference point at their center as well as the labeled model-based and measured markers. The turquoise vector starting at the left foot denotes the measured force vector at the time instance.

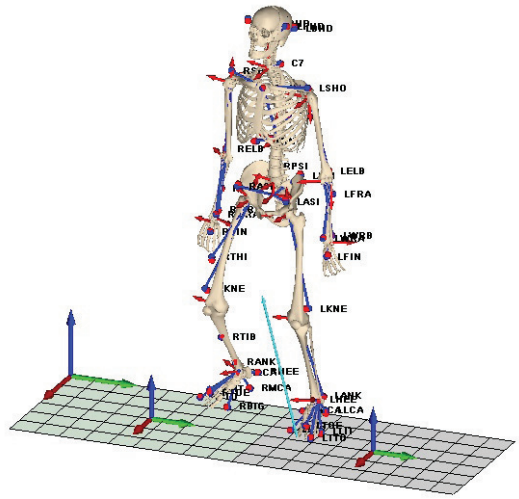


Fig. 9: Tap-dancing, recorded with motion capture and force plates

6. Control

The defined models allow for ease of control development and simulation implementation. They can be used as plant models in the testing simulation of an existing control or the model matrices can be extracted and used in model based controllers.

The problem statement in robotics is often to control a system of ordinary differential equations of the form

$$\mathbf{M}(\mathbf{q}, t)\ddot{\mathbf{q}} - \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{d}(t) = \mathbf{T}(t), \tag{34}$$

where $\mathbf{M}$ is the mass matrix of the system and $\mathbf{Q}$ contains gravitational, centrifugal and Coriolis terms as well as friction and other nonlinear terms and $\mathbf{d}$ is the external disturbance of the system, e.g. impact or system change. Besides a linear PID controller, two model based controllers have been implemented, yet and they are explained in the following examples

6.1. Example: Computed torque controller

The computed-torque controller generates the control torque on a model based approach [11]. Therefore, the position control error is defined as $\mathbf{e} = \mathbf{q}_d - \mathbf{q}$, where $\mathbf{q}_d$ denotes the desired value of the joint coordinate and $\mathbf{q}$ is the actual value. The error dynamics can then be obtained by

$$\ddot{\mathbf{q}}(t) = \ddot{\mathbf{q}}_d(t) + \mathbf{K}_v\dot{\mathbf{e}}(t) + \mathbf{K}_p\dot{\mathbf{e}}(t) \tag{35}$$

with critically damped gains $\mathbf{K}_v = \mathbf{I} \cdot 2c$, $\mathbf{K}_p = \mathbf{I} \cdot c^2$, $c = const.$ and the equations of the computed torque controller are given by

$$\mathbf{T}(t) = \hat{\mathbf{M}}(\mathbf{q}, \mathbf{t})\left[\ddot{\mathbf{q}}_d(t) + \mathbf{K}_v\dot{\mathbf{q}}(t) + \mathbf{K}_p\dot{\mathbf{q}}(t)\right] + \hat{\mathbf{Q}}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{t}) \tag{36}$$

where $\hat{\mathbf{M}}$ and $\hat{\mathbf{Q}}$ are some (more or less accurate) models of the plant.

6.2. Example: Recursive controller

Recursive control laws often show fast performance and robustness against disturbance [12, 13], e.g. receiving additional mass during a manipulation task, which might not be included in the model matrices. The recursive control law of a discrete time based controller is given by

$$\mathbf{T}(k+1) = \mathbf{T}(k) - \sigma(k)/\frac{\partial\sigma(k)}{\partial\mathbf{T}(k)} \tag{37}$$

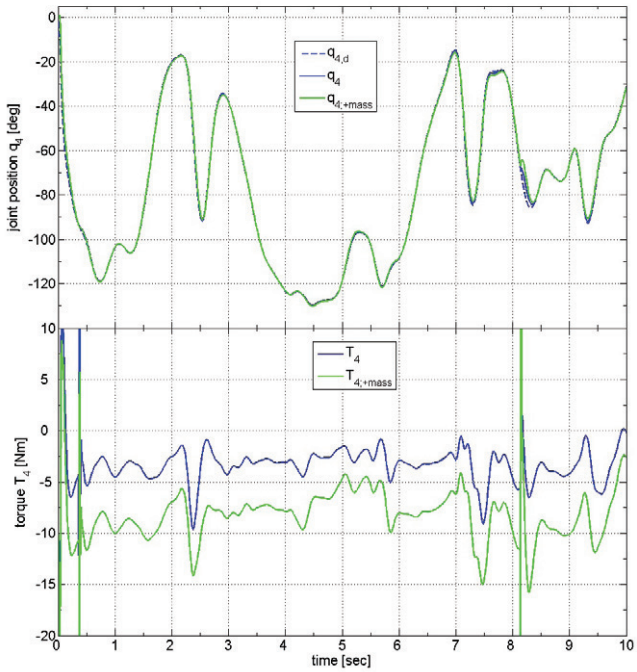


Fig. 10: Performance of a Newton-Controller

where torque $\mathbf{T}(k+1)$ at discrete time $k+1$ is calculated by a Newton-Raphson method, $\boldsymbol{\sigma}(k)$ is some desired error dynamics $\boldsymbol{\sigma}(k) = \ddot{\mathbf{e}}(k) + \mathbf{K}_v\dot{\mathbf{e}}(k) + \mathbf{K}_p\dot{\mathbf{e}}(k)$ and actuator torque is limited to $\mathbf{T}(k+1) \in [\mathbf{T}_{min},\ \mathbf{T}_{max}]$. For model based $\boldsymbol{\sigma}$ the control law is obtained by

$$\mathbf{T}(k+1) = \mathbf{T}(k) + \hat{\mathbf{M}}(k)(\ddot{\mathbf{e}} + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) \tag{38}$$

and closed loop stability is given for the spectral radius $\rho(\mathbf{J}(f)) < 1$ with $\mathbf{J}(f) = \mathbf{I} - \hat{\mathbf{M}}\mathbf{M}^{-1}$, where $\hat{\mathbf{M}}$ is the estimated or model based mass matrix and $\mathbf{M}$ is the mass matrix of the plant [14]. Such a controller can be easily designed and simulated with the presented multibody approach. As an example, the Newton-controller was applied to a seven dof robotic arm. A defined human-like trajectory was selected as the desired trajectory and the controller had to position the robotic arm with different payload mass. The additional mass has not been purposefully considered in the mass matrix. Figure 10 shows the desired trajectory and the resulting well performing controller trajectories as well as the generated torque for the robotic arm with no mass and with a mass of 2kg for the elbow flexion joint.

7. Optimal motion

The analysis of dynamical systems may lead to optimal control problems, especially in robotics research, where dynamical properties of the system (e.g. force, kinetic energy) have to be minimized and a control law is desired such that a given criterion is optimized by a cost function [15]. Human motion is investigated for certain optimality criteria in order to understand human motion [16] and, in humanoid robotics, transfer human motion to the robotic system.

An optimal control problem can be formulated on the basis of Pontryagin's minimum principle. It is used for the optimal control law taking the system from an initial state to a final state, especially in the presence of constraints, e.g. joint and actuator limits. The formulation is obtained from the Hamiltonian of the system

$$H(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), t) = \dot{\boldsymbol{\lambda}}(t)\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + C(\mathbf{x}(t), \mathbf{u}(t)), \tag{39}$$

where $\mathbf{x}(t)$ are the state variables of the system, $\mathbf{u}(t)$ is the control, $\boldsymbol{\lambda}(t)$ are the Lagrange-multipliers, $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$ is the dynamical system in state space form and $C(\mathbf{x}(t), \mathbf{u}(t))$ forms some optimality principles [17] for a scalar cost

function

$$J = \int_0^{t_f} C(\mathbf{x}(t), \mathbf{u}(t))dt. \tag{40}$$

The equations result in a system with a large number of variables for even few state variables. Different methods and algorithms are available, e.g. collocation methods [18], multiple shooting methods [19, 20] or temporal finite element methods [21, 22] to solve the problem and these methods are poor in computational performance.

The optimal control approach for problems with initial and target time conditions was also implemented in this project to study human and humanoid optimal motion for comparison with motion measurements of a human subject. The main focus lies on computational performance and ease of implementation.

Basically, human motion trajectories are smooth that is they are at least twice continuous differentiable. Given an initial time t^I and a target time t^F the initial and the final states of the system are given and therefore, the joint coordinates $\left[\mathbf{q}_{ij}^I, \mathbf{q}_{ij}^F\right], \left[\dot{\mathbf{q}}_{ij}^I, \dot{\mathbf{q}}_{ij}^F\right] \left[\ddot{\mathbf{q}}_{ij}^I, \ddot{\mathbf{q}}_{ij}^F\right]$ of any joint G_{ij} are known and the absolute position and orientation of each body i respectively. The solution in the time interval $[t^I, t^F]$ can then be discretized and piecewise interpolated between the supporting points with polynomials of order three or higher. Therefore, piecewise cubic Hermite spline interpolation is useful [23]. It is composed of two supporting points $q_{ij}^{t_k}$, $q_{ij}^{t_{k+1}}$ and their slopes $\dot{q}_{ij}^{t_k}$, $\dot{q}_{ij}^{t_{k+1}}$ at discrete time instances t_k, t_{k+1} so that the trajectory $q_{ij}(t)$ is obtained by the sum of the single cubic polynomials on the interval $t = [t_0, t_1, .., t_k, .., t_n]$ with $t_0 = t_{(k=0)} = t^I$, $t_n = t_{(k=n)} = t^F$ and $\Delta t_k = t_{k+1} - t_k$

$$q_{ij}(t) = \sum_{k=0}^{n-1} S_k(q_{ij}^{t_k}, \dot{q}_{ij}^{t_k}, t) \tag{41}$$

where S_k contains the time dependent piecewise cubic Hermite spline with its polynomial coefficients based on $q_{ij}^{t_k}$, $q_{ij}^{t_{k+1}}$, $\dot{q}_{ij}^{t_k}$ and $\dot{q}_{ij}^{t_{k+1}}$ The obtained trajectory $q_{ij}(t)$ is C^1 continuous over element boarders and its differentials are obtained by

$$\dot{q}_{ij}(t) = \sum_{k=0}^{n-1} \dot{S}_k(q_{ij}^{t_k}, \dot{q}_{ij}^{t_k}, t) \tag{42}$$

$$\ddot{q}_{ij}(t) = \sum_{k=0}^{n-1} \ddot{S}_k(q_{ij}^{t_k}, \dot{q}_{ij}^{t_k}, t) \tag{43}$$

For a system with N dof, equations (41), (42) and (43) can be rewritten to matrix form

$$\begin{aligned} \mathbf{q}(t) &= S(\mathbf{q}^{t_k}, \dot{\mathbf{q}}^{t_k}, t) \\ \dot{\mathbf{q}}(t) &= \dot{S}(\mathbf{q}^{t_k}, \dot{\mathbf{q}}^{t_k}, t) \\ \ddot{\mathbf{q}}(t) &= \ddot{S}(\mathbf{q}^{t_k}, \dot{\mathbf{q}}^{t_k}, t) \end{aligned} \tag{44}$$

The differential equations of a dynamical system are further discretized on the time interval $[t^I, t^F]$ with a mesh $\Delta t_l = t_{l+1} - t_l$ smaller than Δt_k to

$$\mathbf{T}^{t_l} = \mathbf{M}(\mathbf{q}^{t_l})\ddot{\mathbf{q}}^{t_l} - \mathbf{Q}(\mathbf{q}^{t_l}, \dot{\mathbf{q}}^{t_l}). \tag{45}$$

and equations (41), (42) and (43) inserted into (45) result to

$$\begin{aligned} \mathbf{T}^{t_l} = \mathbf{M}\left(\mathbf{S}(\mathbf{q}^{t_k}, \dot{\mathbf{q}}^{t_k}, t_l)\right) \cdot \ddot{\mathbf{S}}(\mathbf{q}^{t_k}, \dot{\mathbf{q}}^{t_k}, t_l) \\ -\mathbf{Q}\left(\mathbf{S}(\mathbf{q}^{t_k}, \dot{\mathbf{q}}^{t_k}, t_l), \dot{\mathbf{S}}(\mathbf{q}^{t_k}, \dot{\mathbf{q}}^{t_k}, t_l)\right). \end{aligned} \tag{46}$$

The optimal control problem can then be reduced to a parameter optimization problem of the form

$$min \sum_{l=0}^{n} \mathbf{F}^{\mathbf{T}}\left(t_l, \mathbf{q}^{t_k}, \dot{\mathbf{q}}^{t_k}\right) \mathbf{F}\left(t_l, \mathbf{q}^{t_k}, \dot{\mathbf{q}}^{t_k}\right), \; k = 0, .., n \tag{47}$$

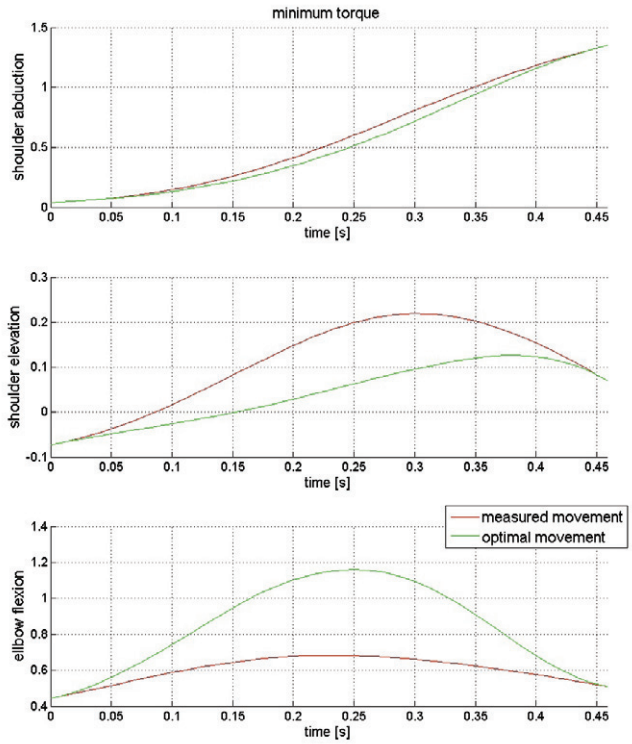


Fig. 11: Comparison of measured motion and motion due to minimum torque criterion

where $\mathbf{F}$ depends on the discrete time t_l, the supporting points $\mathbf{q}^{t_k}$ and the slopes $\dot{\mathbf{q}}^{t_k}$. Further, $\mathbf{F}$ is formulated as one or more optimality principles, e.g. for minimum torque, $\mathbf{F} = \mathbf{T}(t_l)$. Optimality principles of higher order are computed from equations (48) for kinematical dependent principles, e.g. minimum jerk

$$\dddot{\mathbf{q}}(t) = \dddot{S}(\mathbf{q}^{t_k}, \dot{\mathbf{q}}^{t_k}, t), \tag{48}$$

dynamical principles are computed by spline approximation, e.g. the differential of torque for the minimum torque change criterion

$$\dot{\mathbf{T}}(t) = \mathbf{S}_\mathbf{p}(\mathbf{T}^{t_l}, t). \tag{49}$$

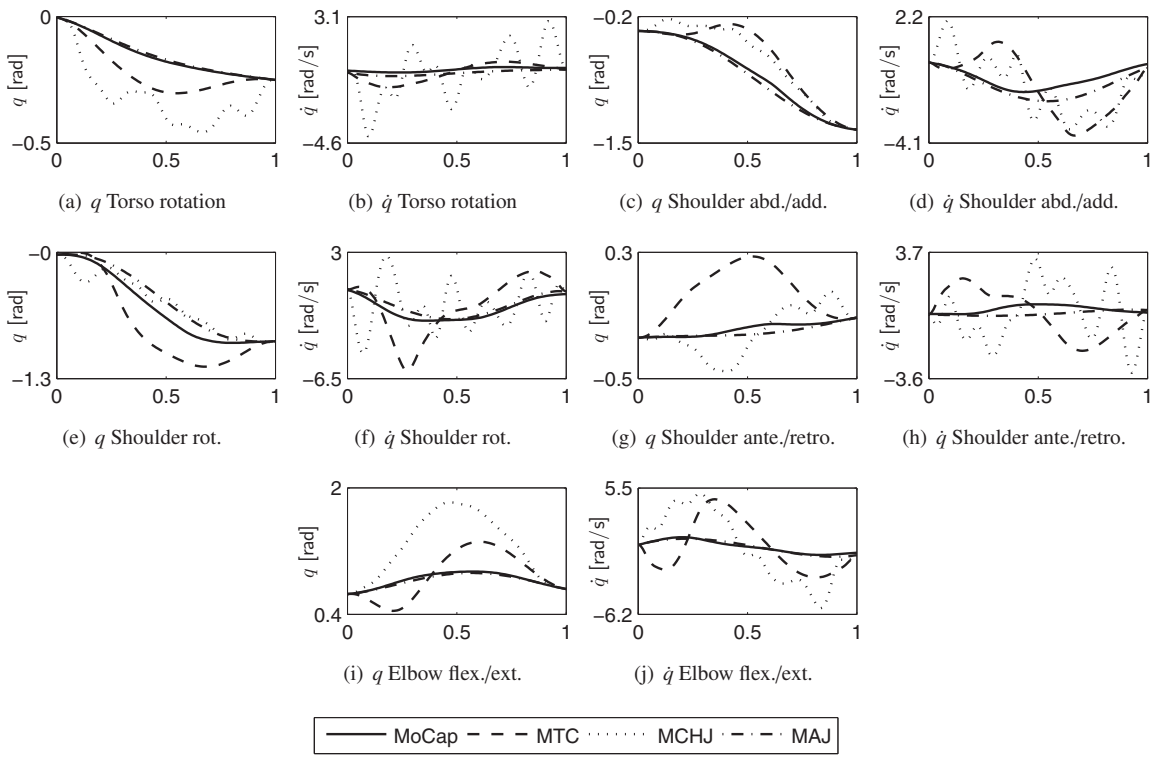
Finally, the problem is numerically solved for the n supporting points and the n slopes for N degrees of freedom q_{ij}, thus, the number of variables to solve for the optimization problem is $2 \cdot n \cdot N$.

Another approximation of the displacement dependent trajectories was realized with a polynomial of 9^{th} order with C^9 continuity, where the second order initial and final conditions are even fulfilled. Therefore, the ten coefficients $[a_0, a_1, a_2, ..., a_9]$ of the polynomial can be determined by a system of ten equations, where six equations are obtained by the initial and final conditions $\left[\mathbf{q}_{ij}^I, \dot{\mathbf{q}}_{ij}^I, \ddot{\mathbf{q}}_{ij}^I, \mathbf{q}_{ij}^F, \dot{\mathbf{q}}_{ij}^F, \ddot{\mathbf{q}}_{ij}^F\right]$. Four more equations are obtained on an equally spaced discrete time interval $t_k = [t_0 = t^I, t_1, t_2, t_3 = t^F]$ with supporting points $q_{ij}(t_1)$, $q_{ij}(t_2)$ and the slopes $\dot{q}_{ij}(t_1)$, $\dot{q}_{ij}(t_2)$. This leads to a number of only $4 \cdot N$ variables to solve for the optimization problem and continuous differentiable trajectories are obtained for displacement, velocity acceleration and jerk coordinates as well as for torque. The continuous shape of the trajectories is similarly compared to human ones and results compared to other approaches are plausible for small time intervals $\Delta t < 2s$.

Figure 11 shows some results for minimum torque simulation of pointing gestures. Data has been recorded on the human subject (red trajectories) and initial and final conditions have been extracted. The optimal motion was then calculated on the defined time interval (green trajectory). Results are plausible compared to other studies in literature [18, 16, 21] and computation time is small within seconds up to half an hour on actual CPUs.

The results due to Minimum Torque Change (MTC), Minimum Constrained Hand Jerk (MCHJ) and Minimum Angle Jerk (MAJ) are presented in Figure 12 [24]. The optimality criterion of MAJ seems to fit best, however the diagrams show that the simulated and measured trajectories still do have differences. Due to this fact it can be assumed that the real motion is generated using a combination of the different criteria using several weighting factors. However, this is subject of ongoing research and results will be shown in the future.

Another criterion, the metabolic cost function is not used here as it requires to model the muscle actuation and performance. This is beyond this research as the generation of motion for a humanoid robot is achieved by a limited number of discrete actuators and therefore optimization criteria like the ones shown above are estimated to be more suitable.



(a) q Torso rotation (b) $\dot{q}$ Torso rotation (c) q Shoulder abd./add. (d) $\dot{q}$ Shoulder abd./add.

(e) q Shoulder rot. (f) $\dot{q}$ Shoulder rot. (g) q Shoulder ante./retro. (h) $\dot{q}$ Shoulder ante./retro.

(i) q Elbow flex./ext. (j) $\dot{q}$ Elbow flex./ext.

Fig. 12: Measured and synthesised trajectories with 5 degrees of freedom

8. Visualization

Matlab provides powerful visualization techniques and animation routines, which are used in the environment. Arbitrary geometries can be defined for any rigid body based on a mesh or a polygon surface and the rendering and representation in plots is taken by Matlab routines. This simplifies the animation of three dimensional elements, e.g. bones, which are available as polygons in the environment or CAD-geometries (Figure 13).

Coordinate systems, joint axes and paths can be plotted on demand. By the way, all skeleton and robot animation plots in this paper were generated directly with models from this approach. Further, the environment provides an interface to the open source Matlab-Toolbox Gait-CAD [25], which provides data visualization routines and methods for the analysis of time series and data mining problems.

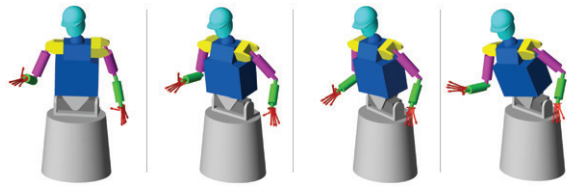


Fig. 13: Visualization of the robot from CAD-geometries

9. Conclusions and future works

The mechanics simulation environment presented in this contribution allows for customization of individual mechatronic problems and it is used within the collaborative research center 588 ”Humanoid Robots” at the University of Karlsruhe. The presented kinematical and dynamical techniques allow for simulation support in the development of the humanoid robots in the project and the analysis of human motion. Present applications are the analysis of manipulation tasks with data acquisition form motion capture and grip force sensors at each finger tip for the dynamic analysis of grasping. Figure 14 presents the visualization of recorded pouring motion with a bottle and a cup. In return motion generation due to different optimality criterion can be tested by a comparison of measured and simulated data. This is important to generate a human like motion of the humanoid.

Further implementations will be done with contact dynamics, e.g. soft-finger contact [26], in order to entirely simulate manipulation tasks with contact dynamics.

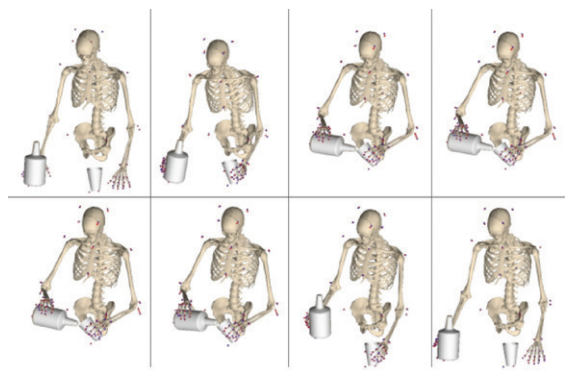


Fig. 14: Manipulation task: Pouring with bottle and cup

Acknowledgments

This work has been granted within the German collaborative research center ”SFB588 Humanoid Robots” granted by the Deutsche Forschungsgemeinschaft. We thank our colleagues T. Stein, A. Fischer, A. Richter, G. Strutzenberger and H. Schwameder, Institute of Sports and Sports Sciences, for their contribution of measurement equipment and data and great coorperation.

References

[1] A. Shabana, Computational Dynamics, Wiley-Interscience, 2001.
[2] G. T. Yamaguchi, Dynamic Modeling of Musculosceletal Motion A Vectorized Approach for Biomechanical Analysis in Three Dimensions, Kluwer Academic Publishers, 2001.
[3] J. Wittenburg, Dynamics of Systems of Rigid Bodies, B.G. Teubner Verlag Stuttgart, 1977.
[4] I. Kapandji, The Physiology of the Joints, volume 2: Lower Limb, Churchill Livingstone, Edinburgh.

[5] C. Simonidis, G. Stelzner, W. Seemann, A kinematic study of human torso motion, in: Proceedings of ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, 2007.
[6] A. White, M. Panjabi, Clinical biomechanics of the spine.
[7] A. Kuo, A least-squares estimation approach to improving the precision of inverse dynamics computations, J. Biomech. Eng 120 (1) (1998) 148–159.
[8] D. Thelen, F. Anderson, S. Delp, Generating dynamic simulations of movement using computed muscle control, Journal of Biomechanics 36 (3) (2003) 321–328.
[9] S. Delp, F. Anderson, A. Arnold, P. Loan, A. Habib, C. John, E. Guendelman, D. Thelen, OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement, IEEE Transactions on Biomedical Engineering 54 (11).
[10] C. Simonidis, W. Seemann, Using wobbling masses and optimization to compensate for residuals in highly dynamic movements, in: 1st Joint International Conference on Multibody System Dynamics, Lappeenranta, Finland, 2010.
[11] F. Lewis, D. Dawson, C. Abdallah, Robot Manipulator Control: Theory and Practice, CRC Press, 2004.
[12] M. Ostojic, Numerical Approach to Nonlinear Control Design, Journal of Dynamic Systems, Measurement, and Control 118 (1996) 332.
[13] M. Ostojic, I. Links, Tracking Control of Chaotic Systems by using Numerical Procedures, 22 nd IASTED International Conference on Modelling, Identification, and Control (2003) 214–218.
[14] M. Ostojic, Recursive control of robotic motion, International Journal of Control 64 (5) (1996) 775–787.
[15] D. Hull, Optimal Control Theory for Applications, Springer, 2003.
[16] T. Flash, N. Hogan, The coordination of arm movements: an experimentally confirmed mathematical model, Journal of Neuroscience 5 (7) (1985) 1688–1703.
[17] E. Todorov, Optimality principles in sensorimotor control (review), Nature neuroscience 7 (9) (2004) 907.
[18] M. A. Admiraal, M. Kusters, S. Gielen, Modeling kinematics and dynamics of human arm movements, Motor Control 8 (2004) 312–338.
[19] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, Springer, 2002.
[20] M. Stelzer, O. von Stryk, Efficient forward dynamics simulation and optimization of human body dynamics, Journal of Applied Mathematics and Mechanics 86 (10) (2006) 828–840.
[21] A. Eriksson, Optimization in target movement simulations, Computer Methods in Applied Mechanics and Engineering.
URL http://www.sciencedirect.com/science/article/B6V29-4SD29KH-1/1/d34571456041c2419dfd509cc07ccd72
[22] A. Eriksson, Temporal finite elements for target control dynamics of mechanisms, Computers and Structures 85 (17-18) (2007) 1399–1408.
[23] R. Bartels, J. Beatty, B. Barsky, Hermite and Cubic Spline Interpolation, An Introduction to Splines for Use in Computer Graphics and Geometric Modelling.
[24] C. Simonidis, Methoden zur analyse und synthese menschlicher bewegungen unter anwendung von mehrkörpersystemen und optimierungsverfahren, Ph.D. thesis (2010).
[25] R. Mikut, O. Burmeister, M. Braun, S.and Reischl, The open source matlab toolbox gait-cad and its application to bioelectric signal processing proc., in: Potsdam, 2008.
[26] R. Keppler, W. Seemann, A dynamic model for the plane contact with rigid contact areas, in: IUTAM-Symposium on Multiscale Problems in Multibody System Contacts, Stuttgart, 2006.