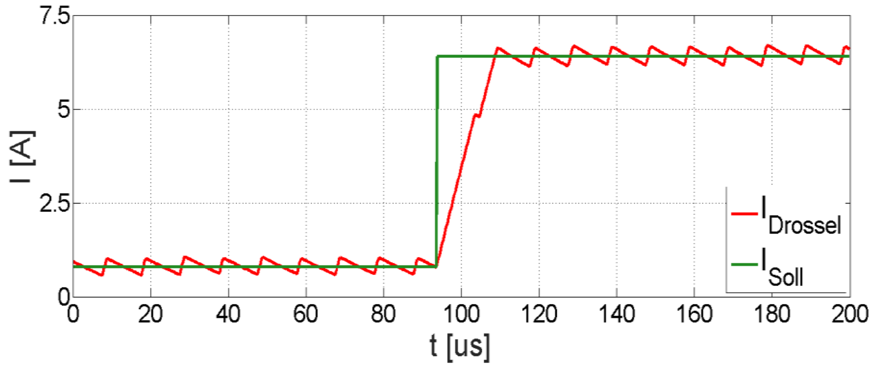
An Easily Adaptive Current Control Method for DC-DC Converters

Dr. Andreas Liske

Elektrotechnisches Institut (ETI) – Power Electronics & Drives



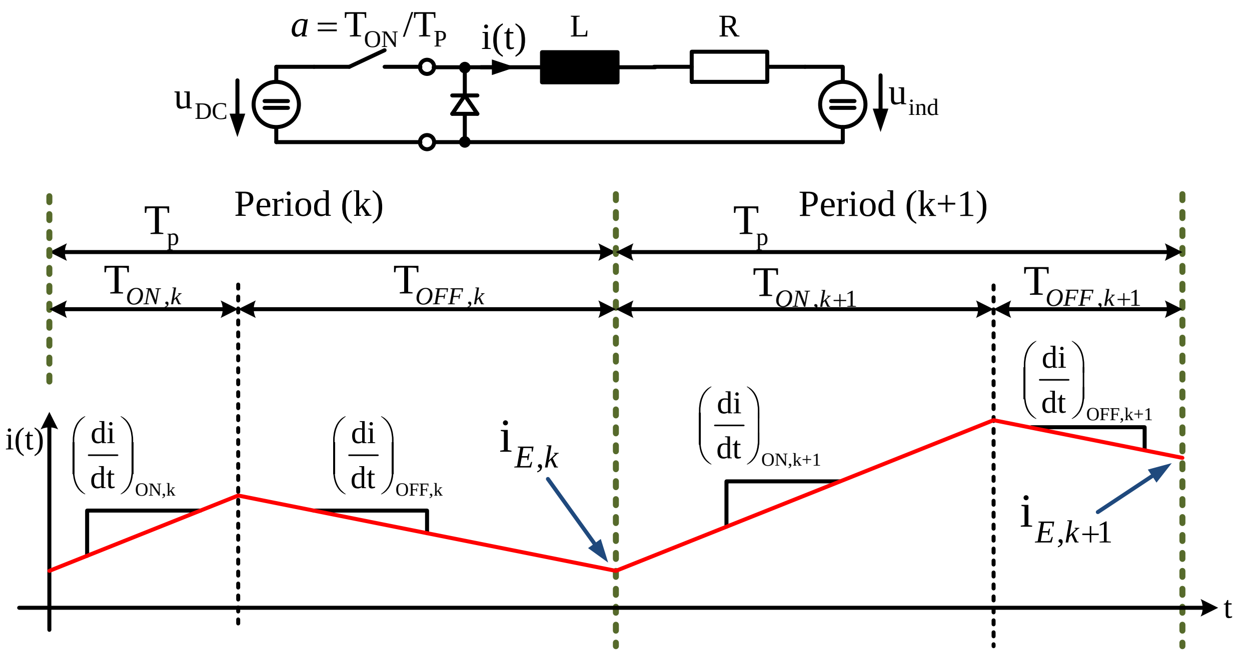
Wangener Automotive Symposium – Inverter Trends & Technology

September, 15th 2022

Content

I. Direct Adaptive Current Control

Basic Principle



$$i_{E,k+1} = i_{E,k} + T_{ON,k+1}\left(\frac{di}{dt}\right)_{ON,k} + T_{OFF,k+1}\left(\frac{di}{dt}\right)_{OFF,k}$$

with u_{DC}, L, R, $u_{ind} \approx$ const. for $2 \cdot T_P$

Knowledge of the ***current slopes***, depending on each applied ***switching state***

Duty cycle for a given current setpoint can be calculated.

No control path parameters required!

I. Direct Adaptive Current Control

Features & Challenges

Features of the DACC:

- ✓ No control path parameters required
- ✓ Completely adaptive
- ✓ Excellent control quality
- ✓ Dead-Beat characteristics
- ✓ Inherently stable
- ✓ Suitable for all kinds of DC/DC-converters
- ✓ Adaptable for 3-phase machine control

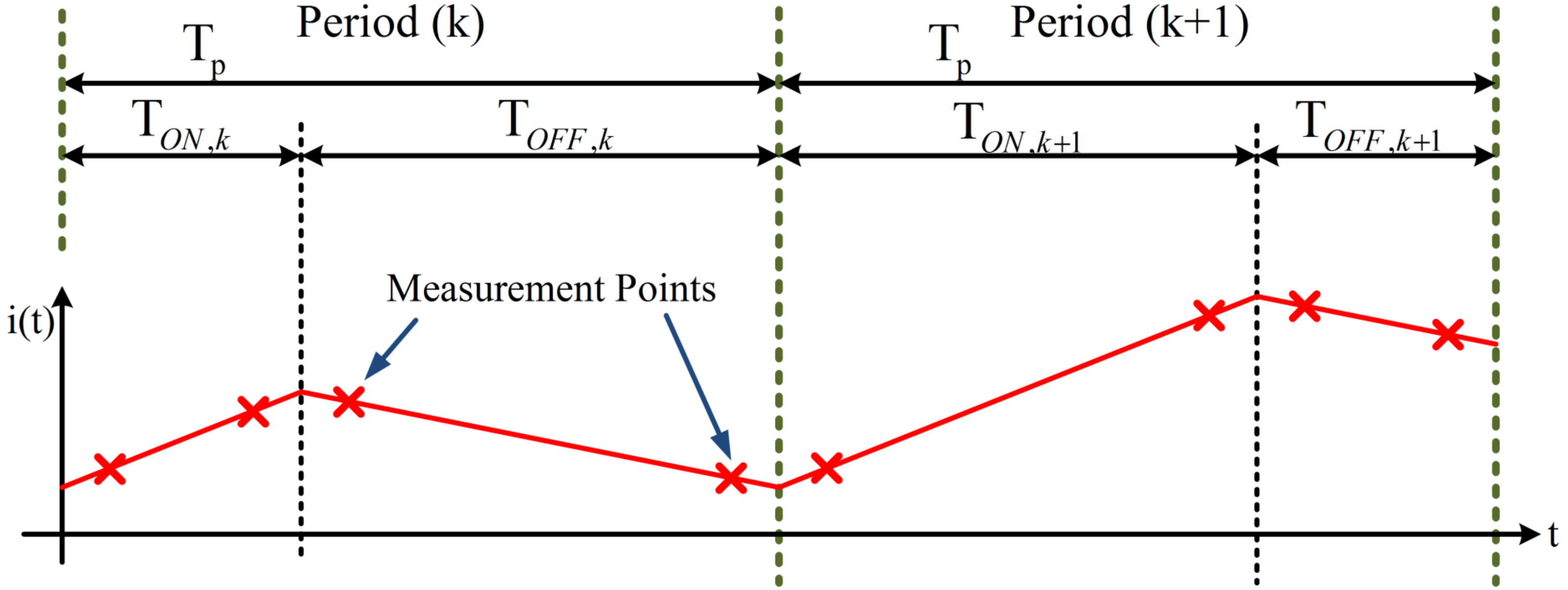
} **current slopes of each switching state must be known**

Challenges with fast switching DC-DC-Converters:

- short measurement time
- noise
- low-cost demand

II. Easy Current Slope Detection (ECSD)

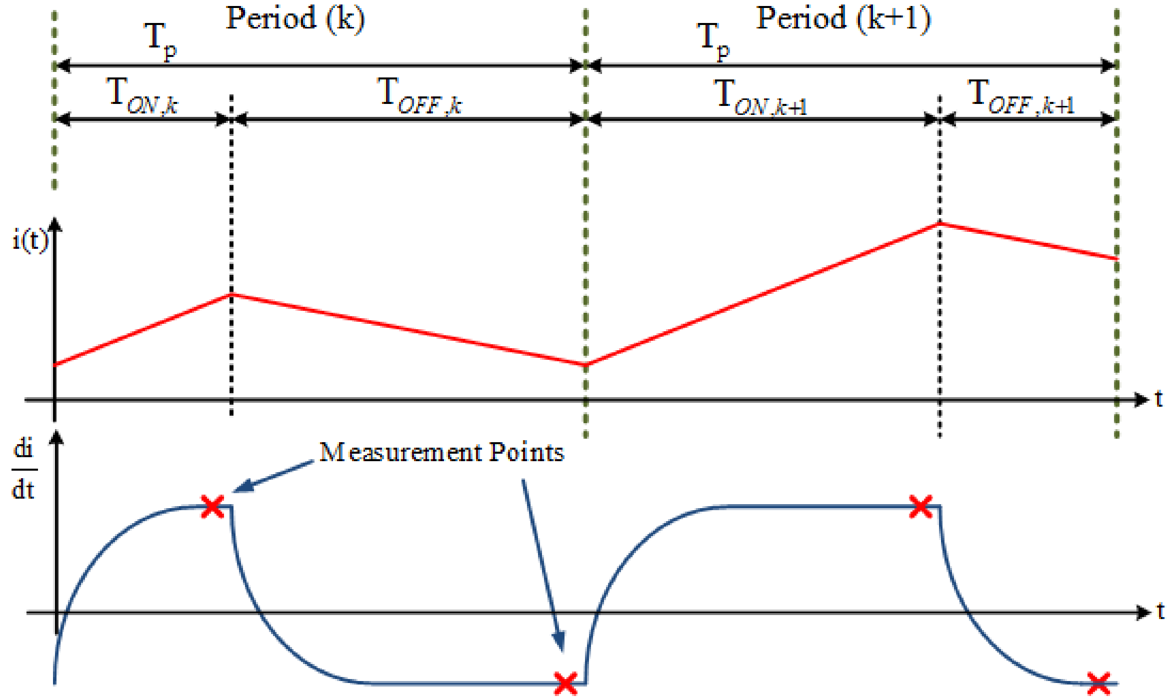
Measuring the current slopes – „2-point-method“



- ✓ Simple
- ✗ very sensitive to noise
- ✗ difficult with small duty cycles / high switching frequency
- ✗ Special trigger necessary

II. Easy Current Slope Detection (ECSD)

Measuring the current slopes – di/dt-sensor



✓ direct measurement of the current slopes with a di/dt-Sensor or analog differentiator circuit

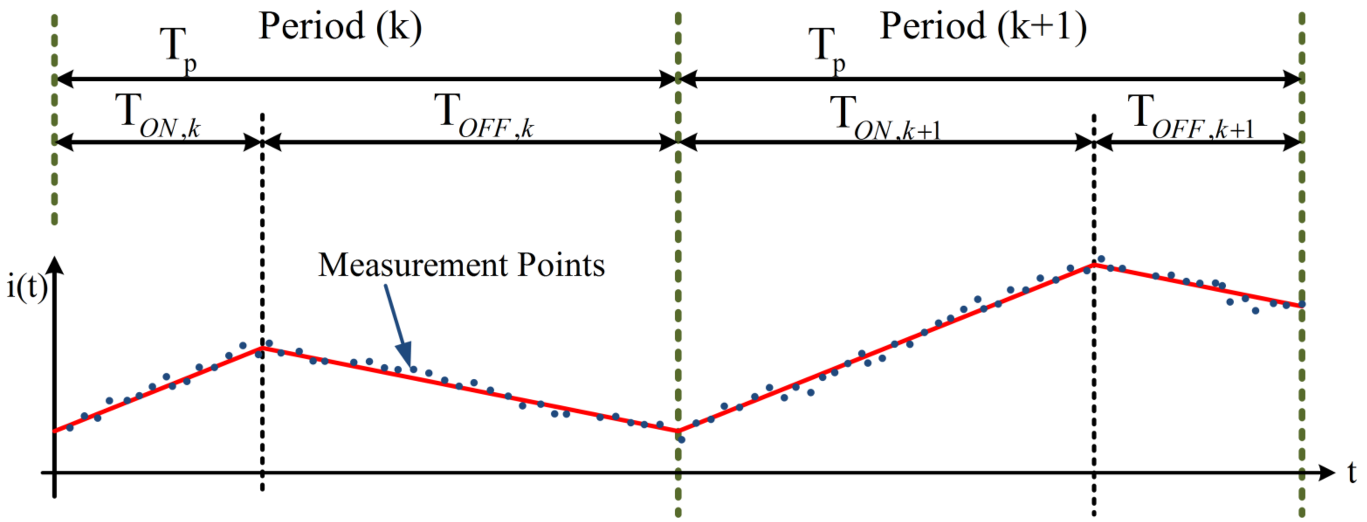
✗ special hardware necessary

✗ not ideal for high switching frequencies

II. Easy Current Slope Detection (ECSD)

Measuring the current slopes – Oversampling

- Measurement with fast oversampling
- Calculation of the current slopes with a Least-Squares-Estimator



✓ very reliable and well-proven

✗ Fast AD-Conversion and calculations necessary

✗ not suitable for high switching frequencies

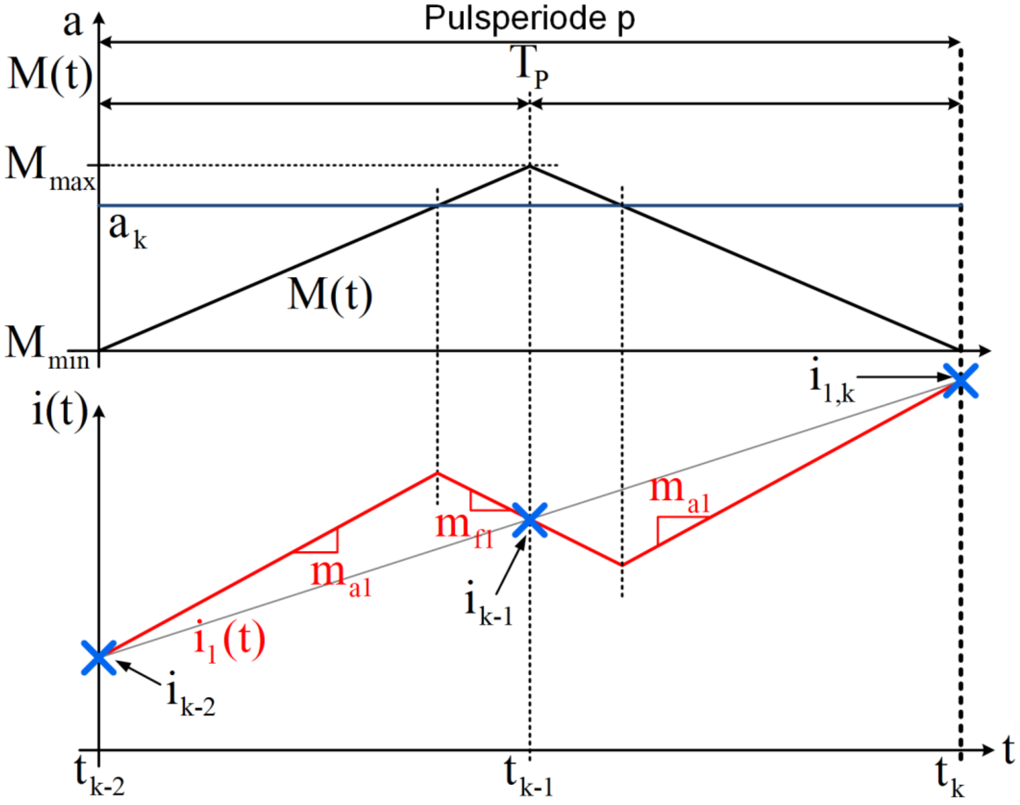
✗ not low cost

II. Easy Current Slope Detection (ECSD)

Standard features in Low-Cost Microcontrollers

- Measuring times:
  - M(t) = 0
  - M(t) = max

- Sampling times „farthest away" from duty cycle transitions



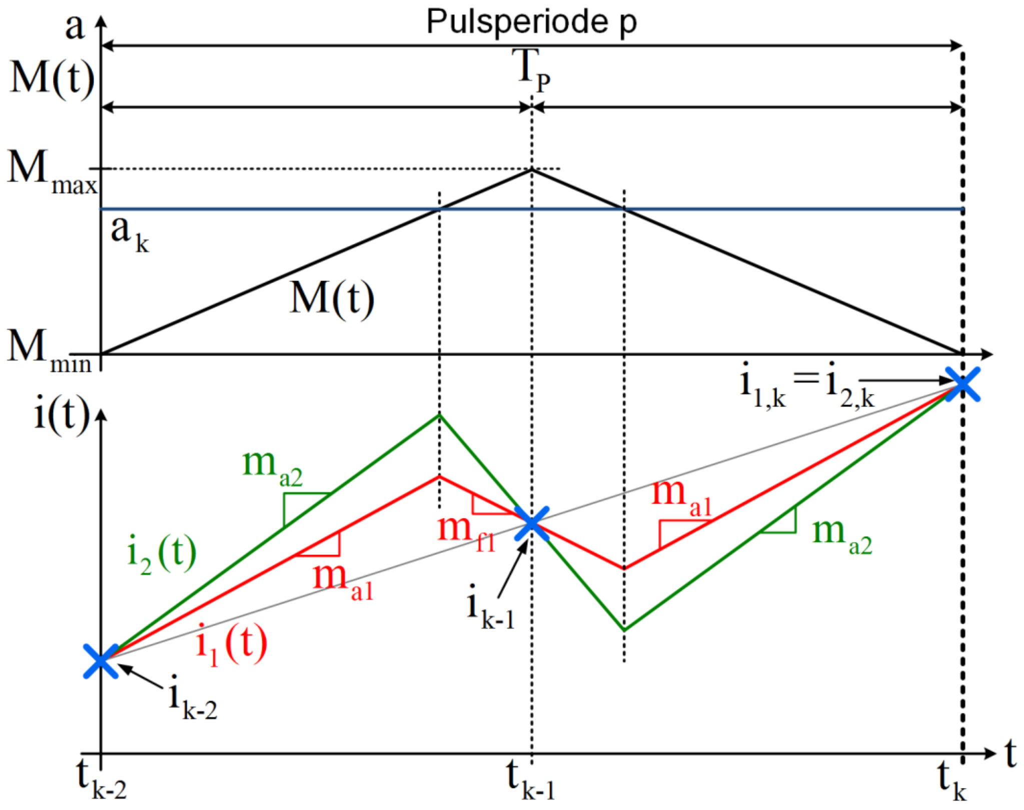
„a" → active
„f" → freewheeling

II. Easy Current Slope Detection (ECSD)

Standard features in Low-Cost Microcontrollers

- Measuring times:
  - M(t) = 0
  - M(t) = max
- Sampling times „farthest away“ from duty cycle transitions
- Different control paths
  - different current slopes
  - Identical measurement values

→ **Current slopes not detectable**!



„a“ → active
„f“ → freewheeling

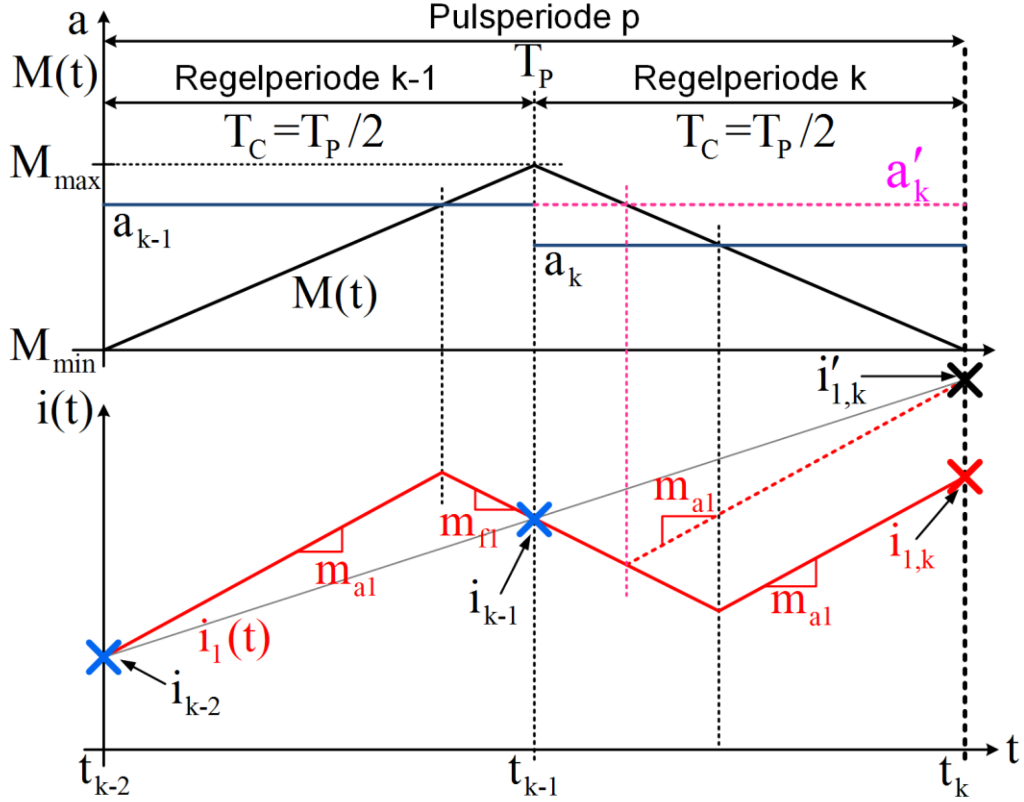
II. Easy Current Slope Detection (ECSD)

„The Trick“

Trick:

Variation of subsequent duty-cycles (Jitter)

$$a_{k-1} \neq a_k$$



II. Easy Current Slope Detection (ECSD)

„The Trick“

Trick:

Variation of subsequent duty-cycles (Jitter)

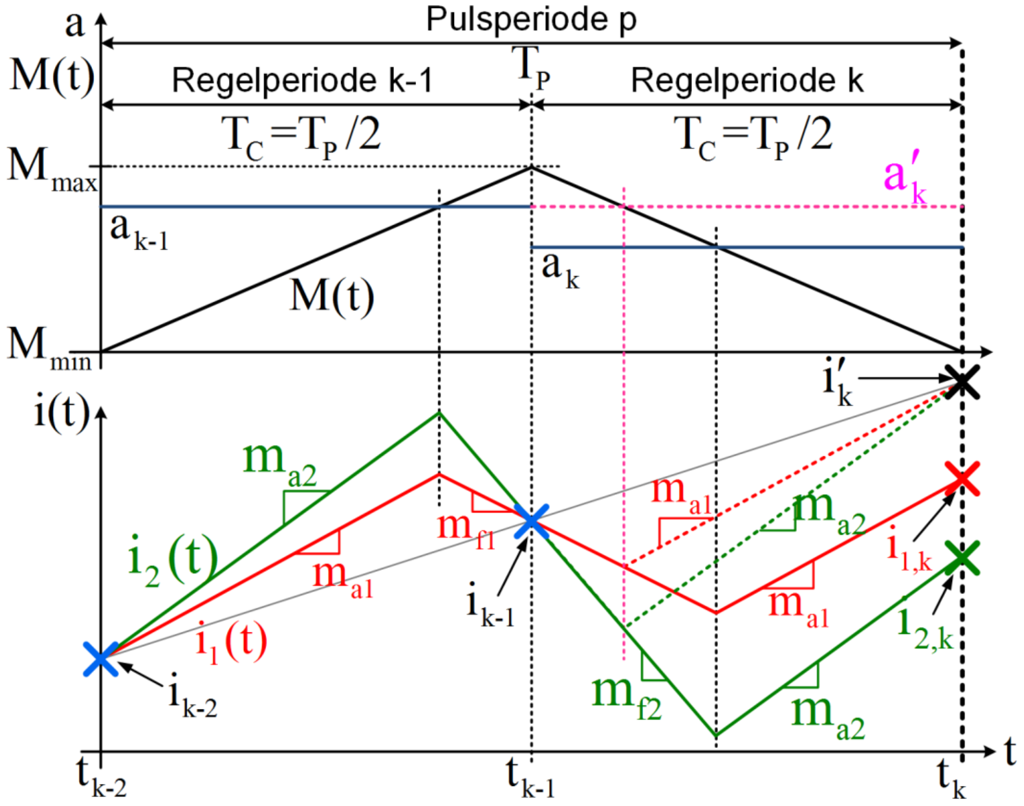
$$a_{k-1} \neq a_k$$

Different control paths

→ different current slopes

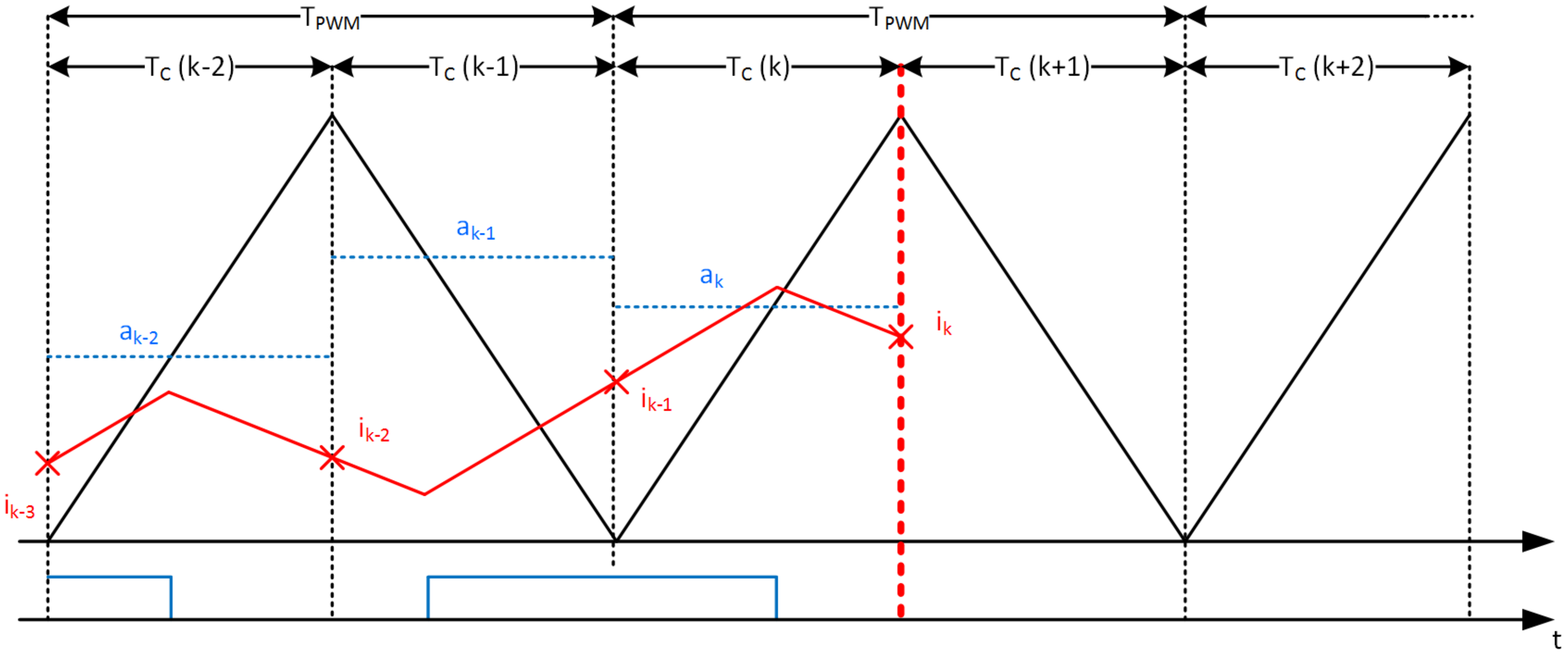
→ ***Different measurement values***

→ Current slopes can be identified!



II. Easy Current Slope Detection (ECSD)

Derivation of the ECSD-Algorithm



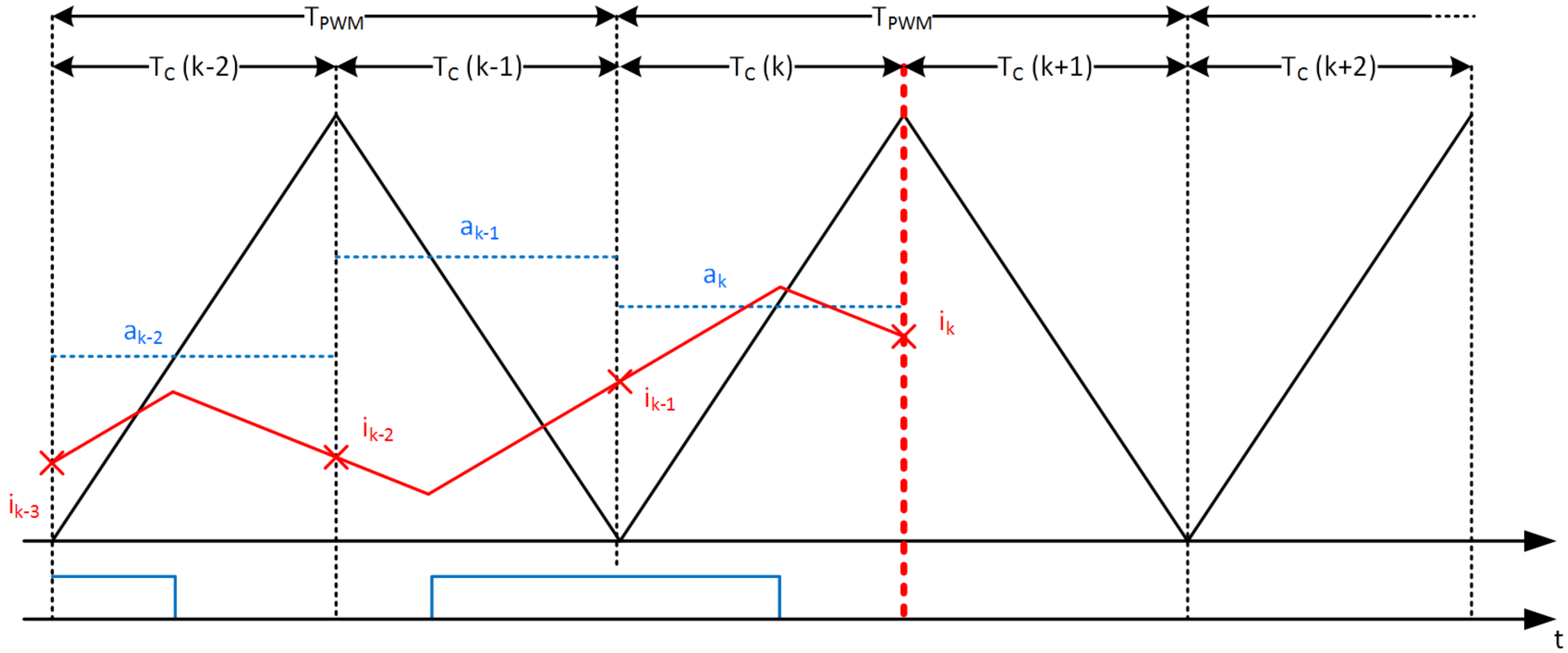
$$i_k = i_{k-1} + a_k \cdot T_C m_a + \left(1 - a_k\right) \cdot T_C m_f$$

$$i_{k-1} = i_{k-2} + a_{k-1} \cdot T_C m_a + \left(1 - a_{k-1}\right) \cdot T_C m_f$$

2 equations
2 unknowns

II. Easy Current Slope Detection (ECSD)

Derivation of the ECSD-Algorithm

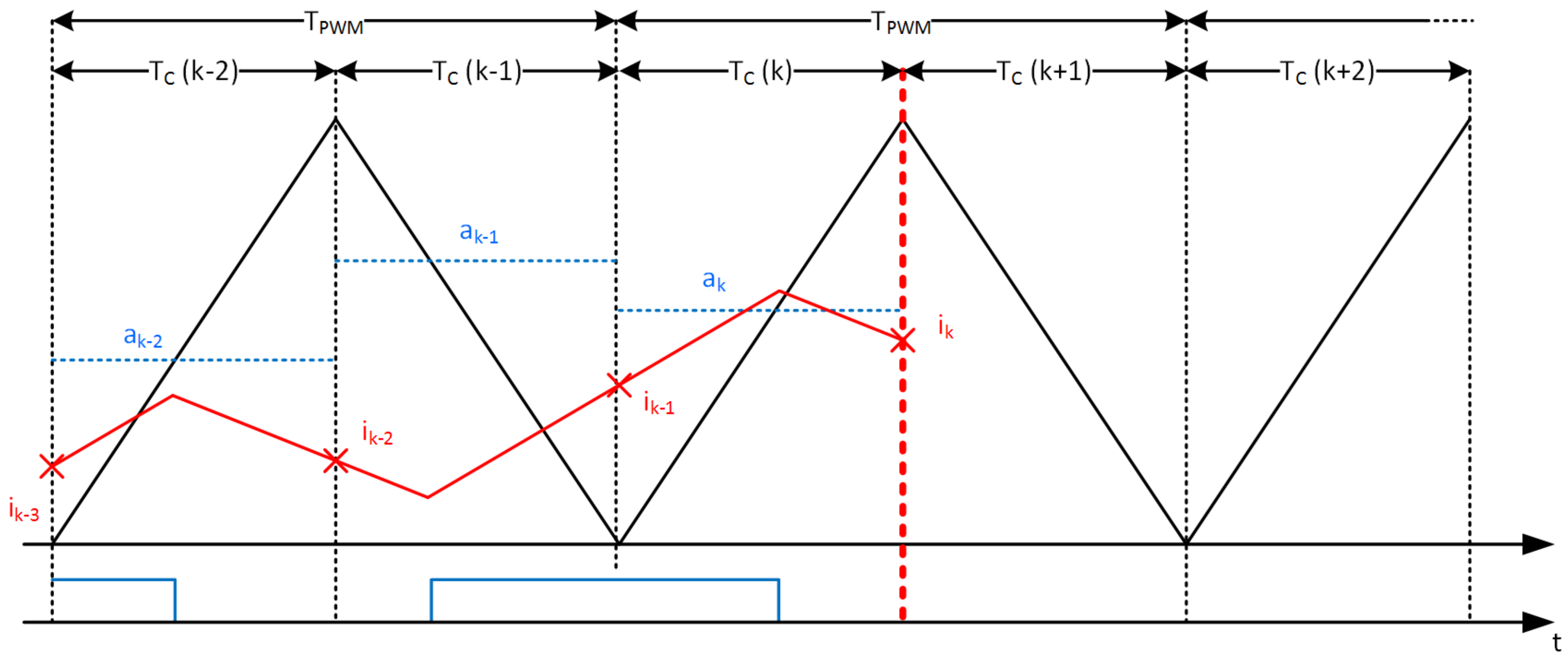


$$m_f = \frac{1}{T_C} \cdot \frac{a_{k-1}(i_{k-1} - i_k) + a_k(i_{k-1} - i_{k-2})}{a_k - a_{k-1}}$$

$$m_a = m_f + \frac{1}{T_C} \cdot \frac{i_{k-2} - 2i_{k-1} + i_k}{a_k - a_{k-1}}$$

II. Easy Current Slope Detection (ECSD)

Derivation of the ECSD-Algorithm



$$\Delta i_f = T_C m_f = \frac{a_{k-1}(i_{k-1} - i_k) + a_k(i_{k-1} - i_{k-2})}{a_k - a_{k-1}}$$

$$\Delta i_a = T_C m_a = \Delta i_f + \frac{i_{k-2} - 2i_{k-1} + i_k}{a_k - a_{k-1}}$$

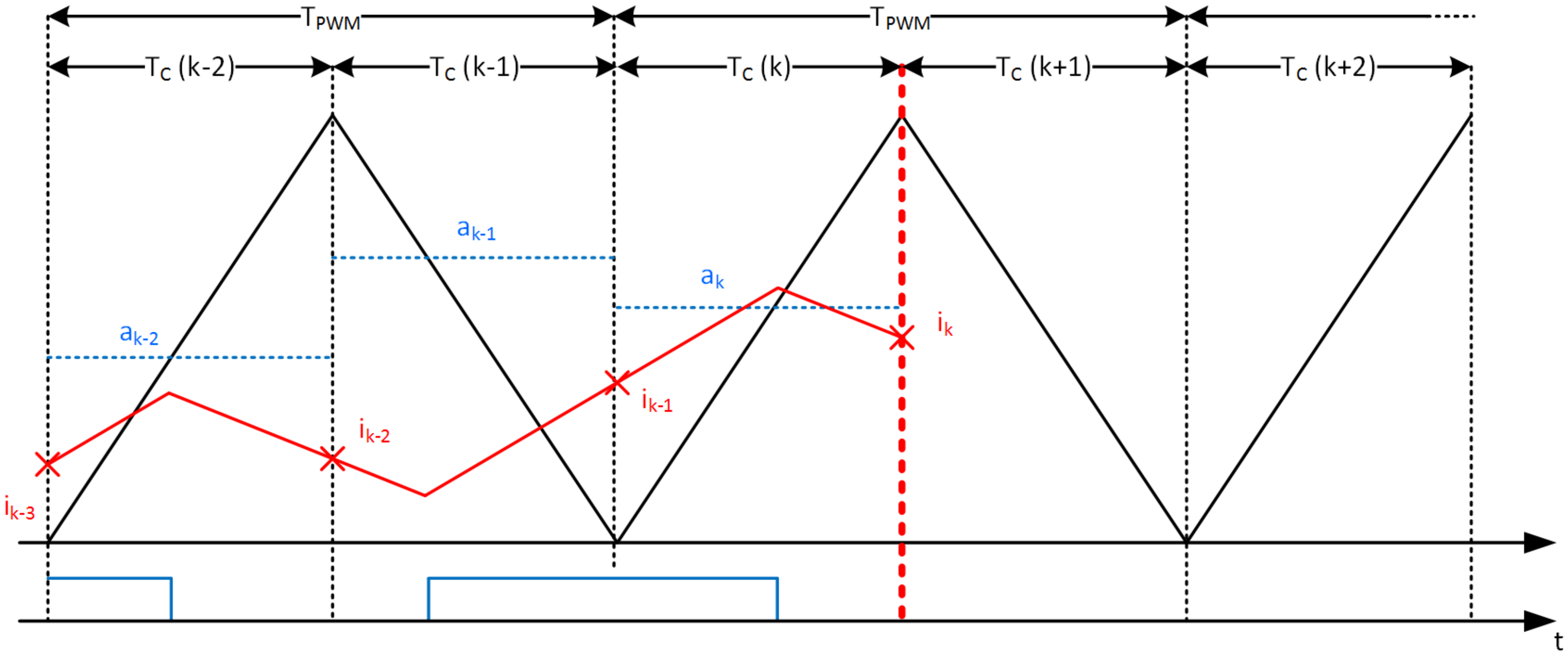
„Current Gradients“:

$$\Delta i_f = T_C \cdot m_f$$

$$\Delta i_a = T_C \cdot m_a$$

II. Easy Current Slope Detection (ECSD)

Derivation of the ECSD-Algorithm



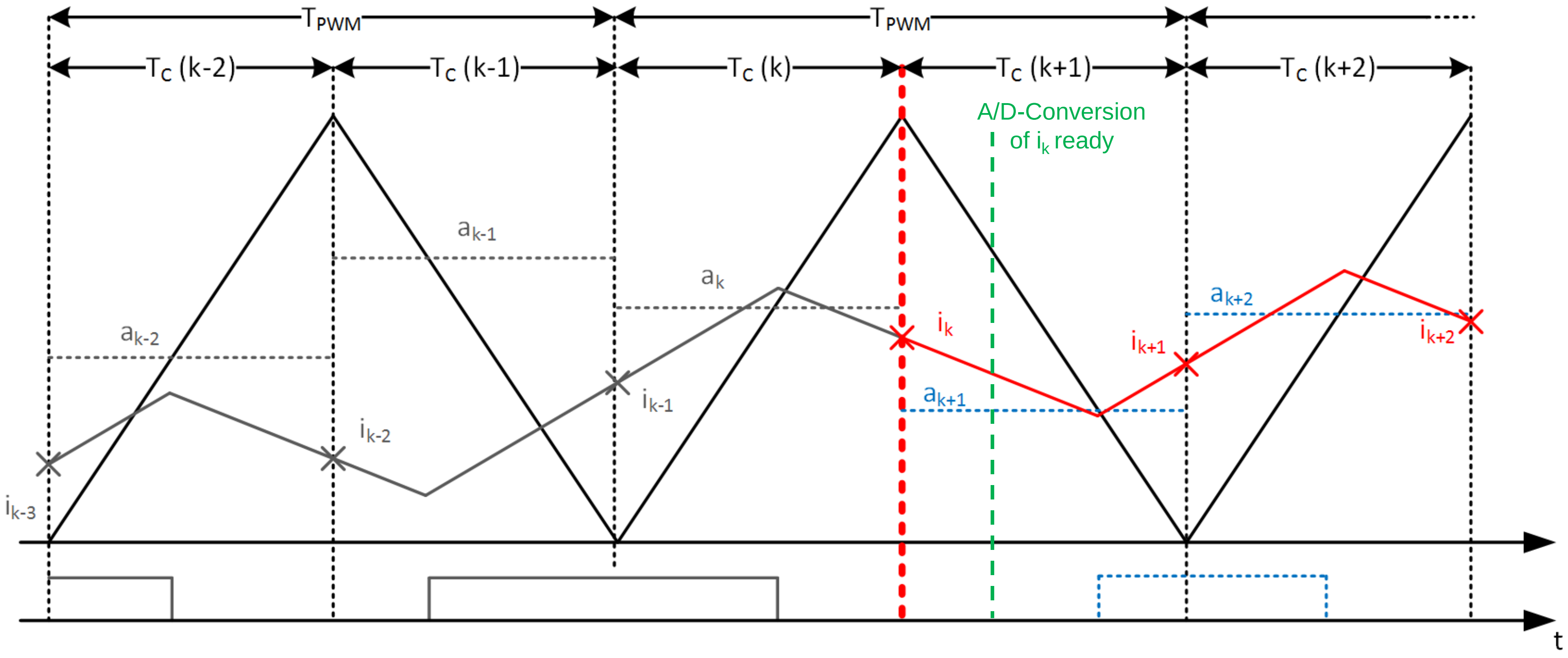
$$\Delta i_f = T_C m_f = \frac{a_{k-1}(i_{k-1} - i_k) + a_k(i_{k-1} - i_{k-2})}{a_k - a_{k-1}}$$

$$\Delta i_a = T_C m_a = \Delta i_f + \frac{i_{k-2} - 2i_{k-1} + i_k}{a_k - a_{k-1}}$$

„Easy Current Slope Detection"

III. Implementing the ECSD for the DACC

Using the Current Gradients for the DACC

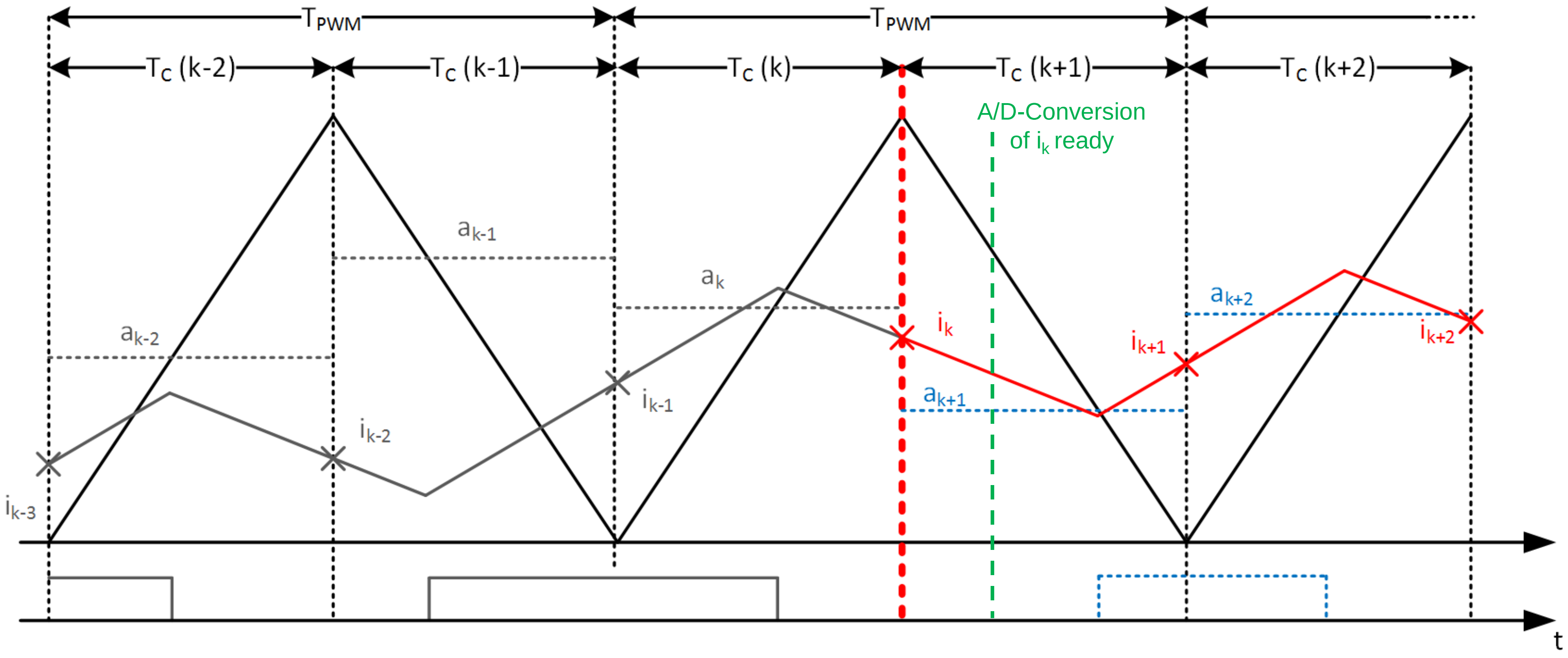


$$i_{k+1} = i_k + a_{k+1} \cdot \Delta i_a + \left(1 - a_{k+1}\right) \cdot \Delta i_f$$

$$i_{k+2} = i_{k+1} + a_{k+2} \cdot \Delta i_a + \left(1 - a_{k+2}\right) \cdot \Delta i_f$$

III. Implementing the ECSD for the DACC

Using the Current Gradients for the DACC

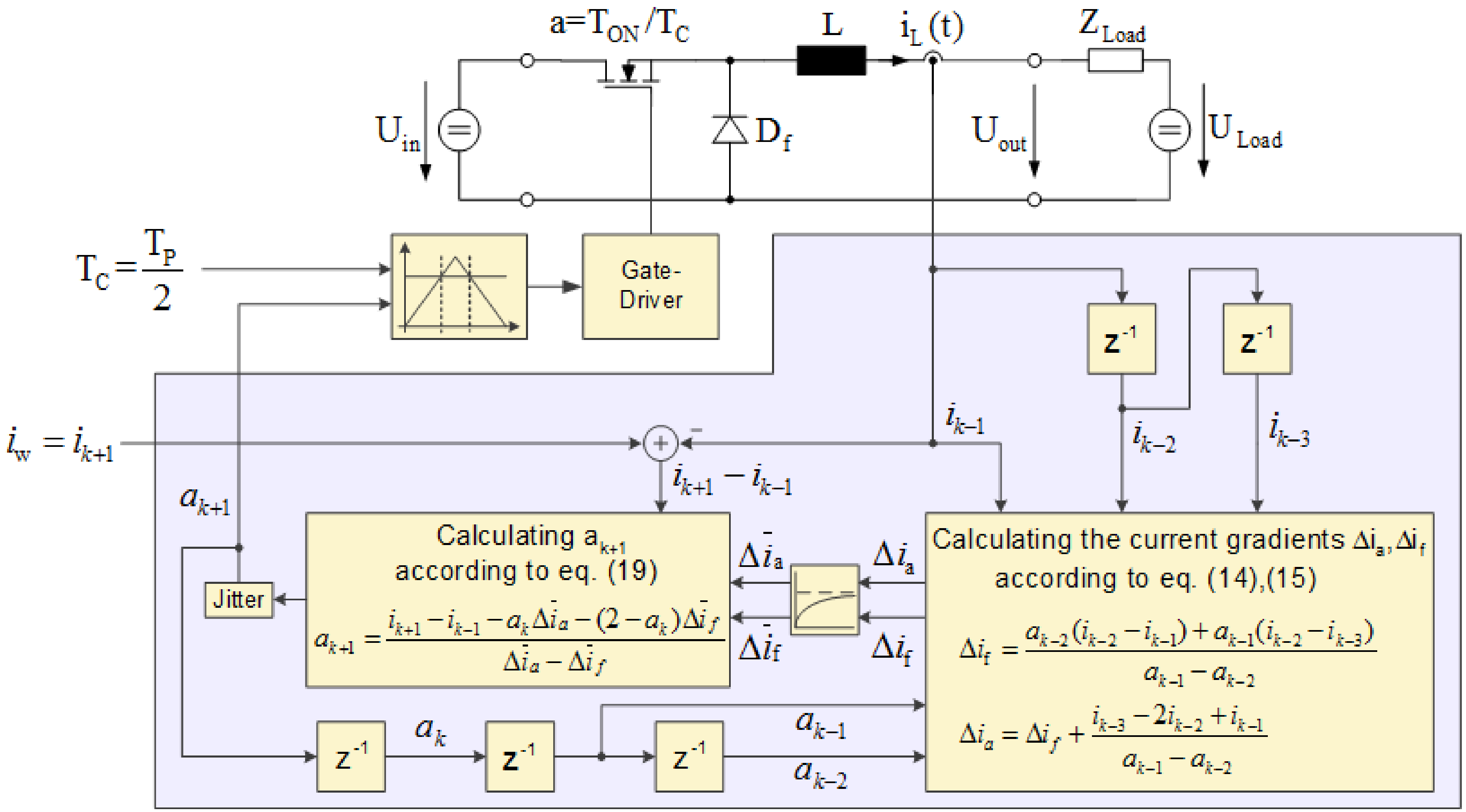


$$a_{k+2} = \frac{i_{k+2} - i_k - a_{k+1} \cdot \Delta i_a - \left(2 - a_{k+1}\right)\Delta i_f}{\Delta i_a - \Delta i_f}$$

Direct Adaptive Current Control
for
DC/DC-Converters

III. Implementing the ECSD for the DACC

Block Diagram



$$a_{k+1} = \frac{i_{k+1} - i_{k-1} - a_k \Delta \bar{i}_a - (2 - a_k)\Delta \bar{i}_f}{\Delta \bar{i}_a - \Delta \bar{i}_f}$$

$$\Delta i_f = \frac{a_{k-2}(i_{k-2} - i_{k-1}) + a_{k-1}(i_{k-2} - i_{k-3})}{a_{k-1} - a_{k-2}}$$

$$\Delta i_a = \Delta i_f + \frac{i_{k-3} - 2i_{k-2} + i_{k-1}}{a_{k-1} - a_{k-2}}$$

DACC for DC/DC-Converters with easy identification of the current slopes

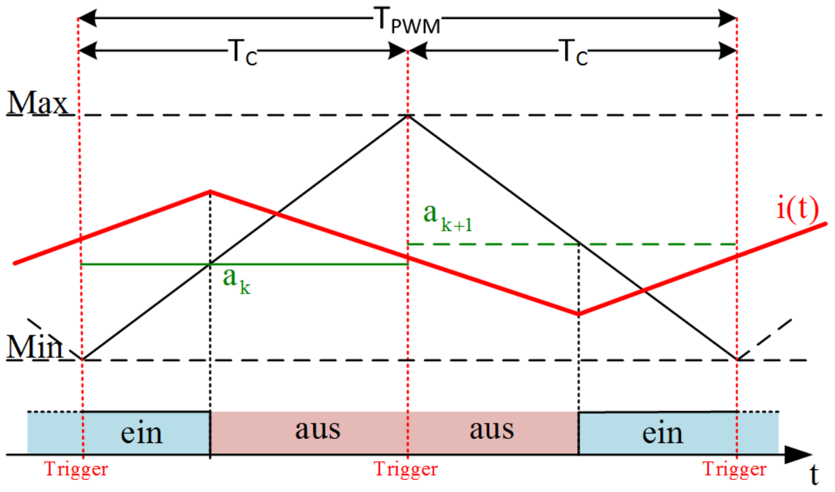
Noise & Sensitivity

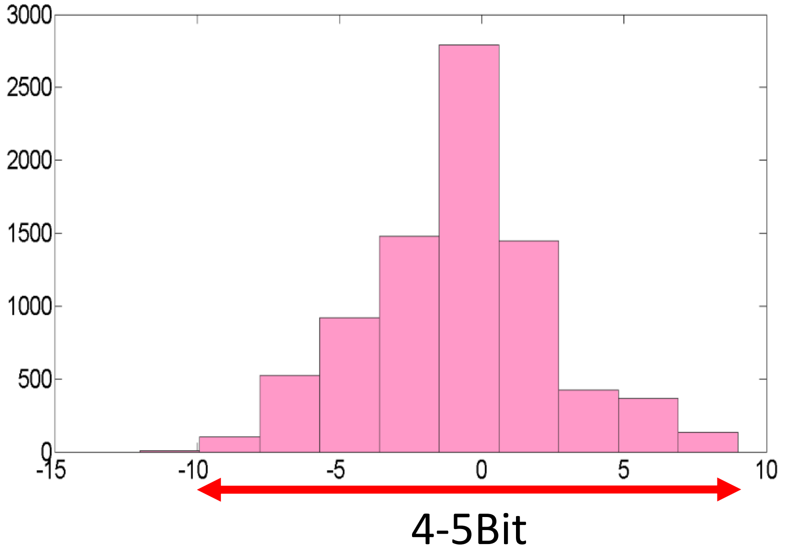
Sensitivity

- In steady state operation:
  Almost identical current samples
  → Difference $i_k - i_{k-1}$ is very small compared to the measuring range

- Critical:
  - Noise of the current signal
  - Noise of the AD-Converter

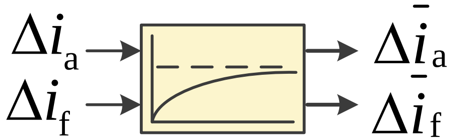
Original idea of controlling the current with the current slopes, calculated from period to period only did not work due to noise.

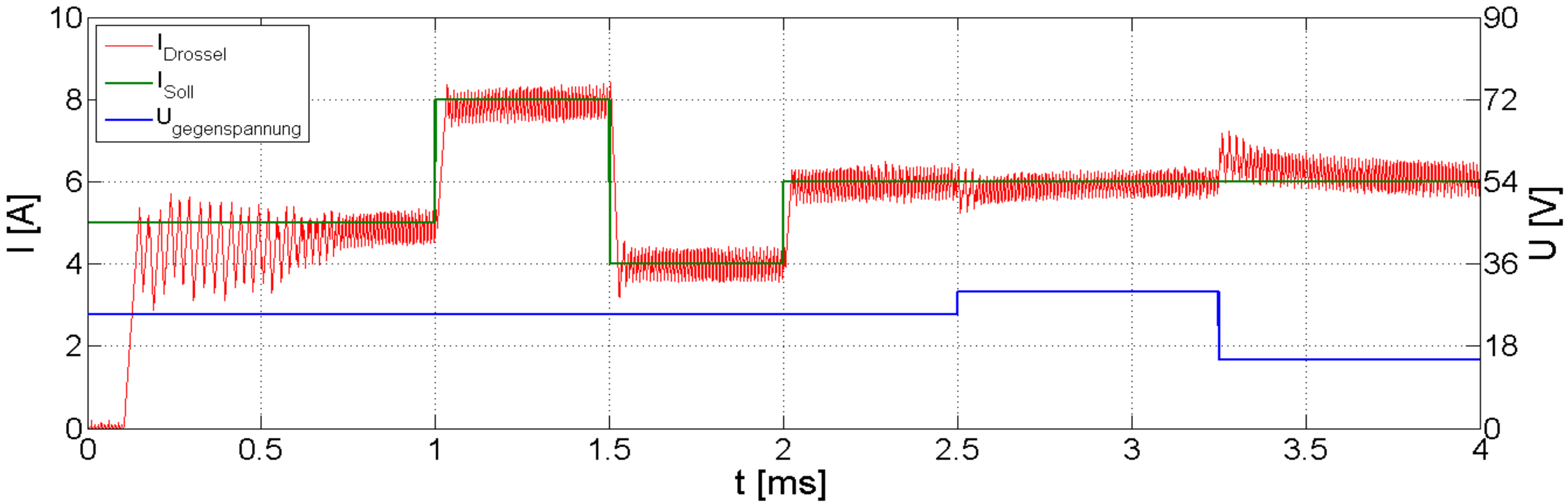




III. Implementing the ECSD for the DACC

DACC with filtered current gradients (DACC-F)

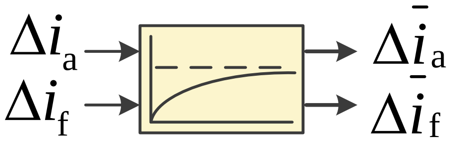


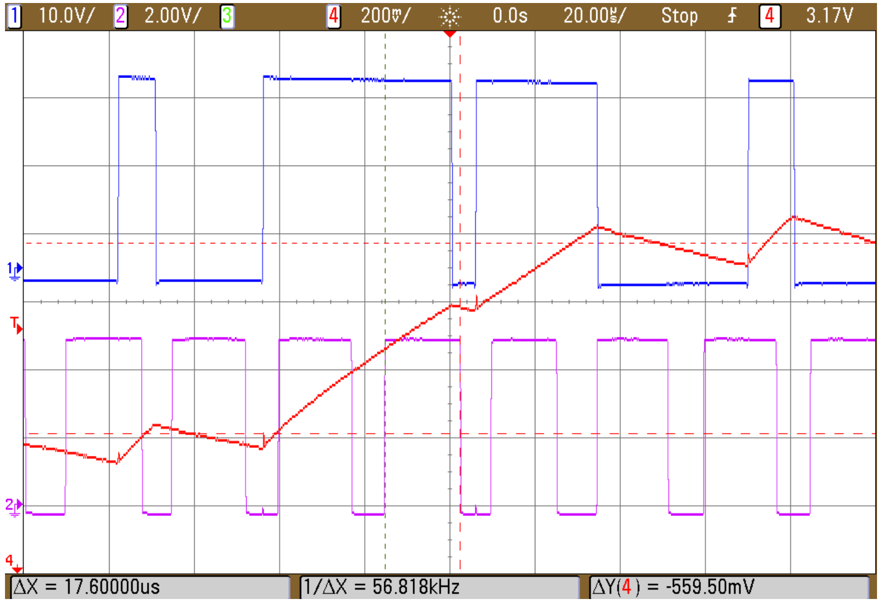


- ✓ only current measurement necessary
- ✓ self-parametrizing
- ✓ dead-beat-characteristics for setpoint steps
- ✗ without voltage feed-forward slow disturbance reaction

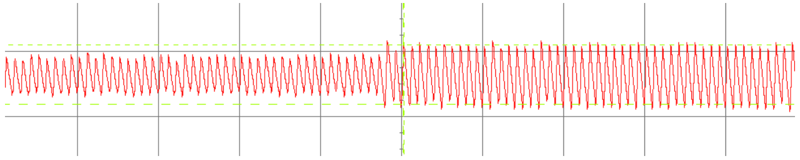
III. Implementing the ECSD for the DACC

DACC with filtered current gradients (DACC-F) – with low-cost microcontroller

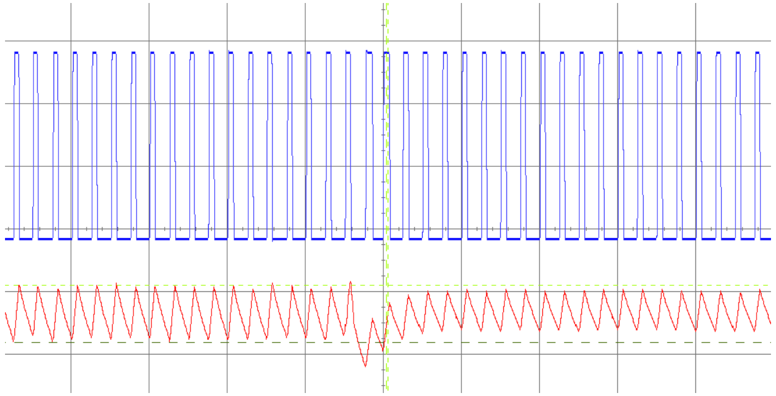




Setpoint Step 1A to 3A



Load step: 800nH → 500nH



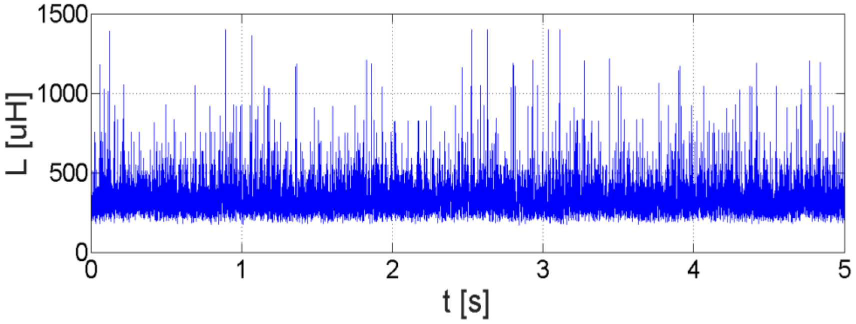
Load step: 600nH → 800nH

III. Implementing the ECSD for the DACC

Using the ECSD to calculate L

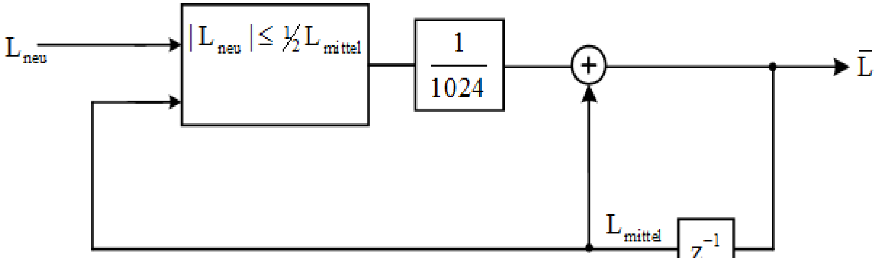
Calculation of L:

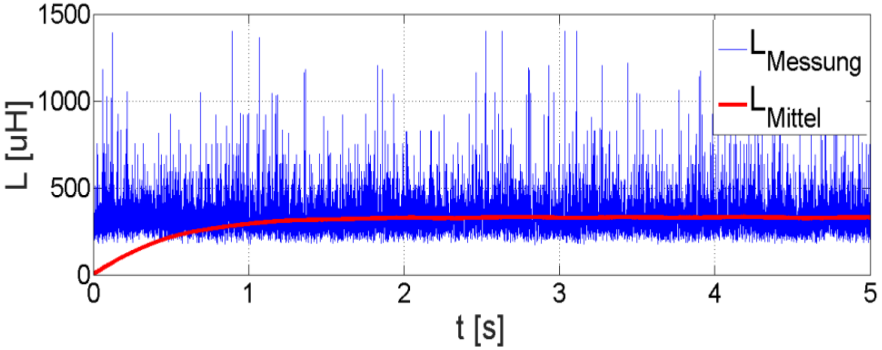
$$L = \frac{-U_a \cdot (a_k - a_{k-1})}{a_{k-1} \cdot (i_{k-1} - i_k) + a_k \cdot (i_{k-1} - i_{k-2})}$$



Filtering L:

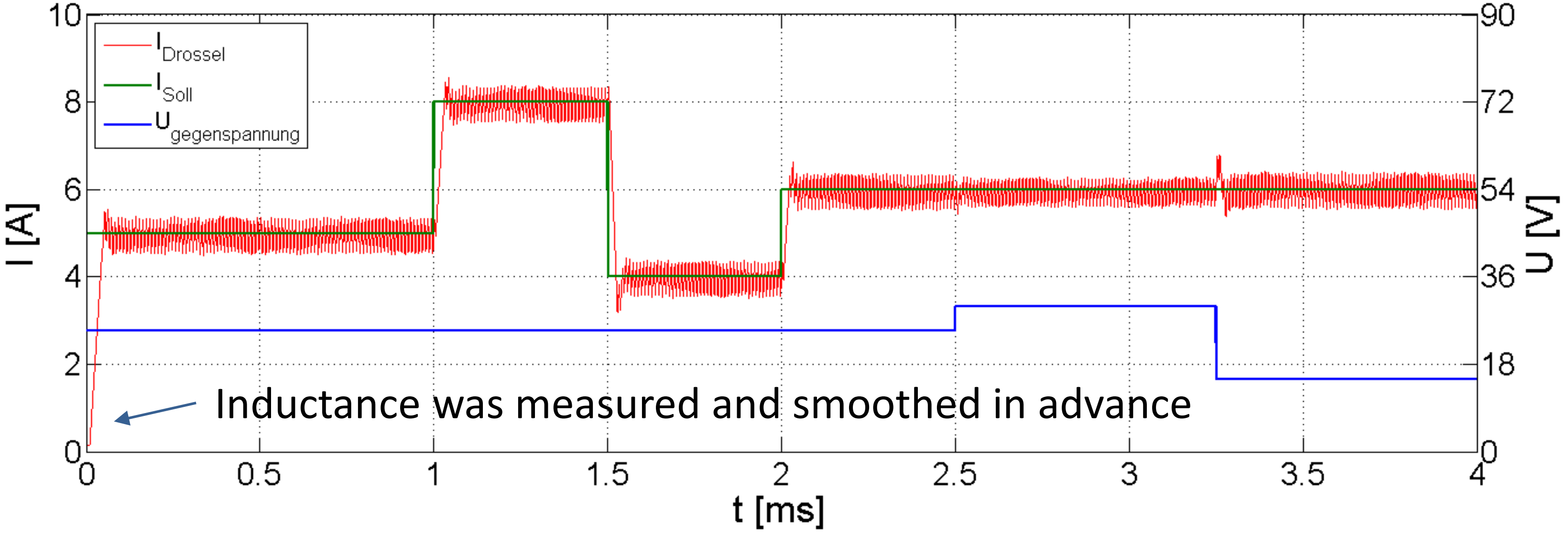
- Noise is one-sided
- Special Filter





III. Implementing the ECSD for the DACC

MPC with filtered L (DACC-MPC)



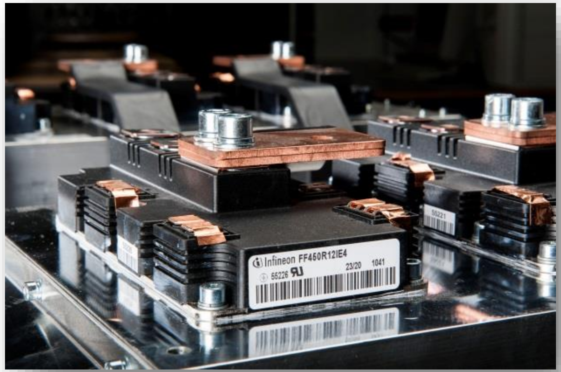
- ✓ Excellent control quality and good disturbance reaction
- ✗ Additional voltage-measurement necessary
- ✗ Smoothing out L takes relatively long (~1s)

IV. Conclusion

1. **Easy & efficient algorithm to identify the current slopes (ECSD)**
   - ✓ simple & low cost
   - ✓ easy to implement
   - ✓ no external hardware needed
   - ✓ just two samples per period at times where switching noise is minimal
   - ✓ perfect for high switching frequency applications

2. **Combination of the ECSD with the Direct Adaptive Current Control (DACC)**
   - ✓ Enables the DACC in high switching frequency SMPS
   - ✓ Realized as DACC-F and MPC
   - ✓ Deadbeat characteristics in setpoint steps
   - ✓ despite jitter good steady state accuracy
   - ✓ self-initializing & low cost
   - ✓ inherent stable

Thank you for your attention!



Dr. Andreas Liske

Kaiserstr. 12 • 76131 Karlsruhe • Germany

Phone: +49 / (0)721 / 608 - 46848

Email: Andreas.Liske@kit.edu

Web: www.eti.kit.edu