Aalto University
School of Electrical
Engineering

Module 3
State Feedback Current Control: Discrete-Time Design

Marko Hinkkanen

Politecnico di Torino, February 2017

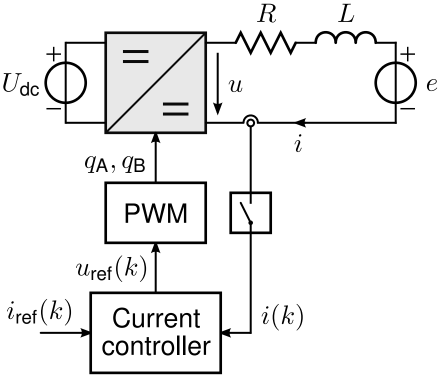
Outline

Introduction

- Switching losses in the converter are proportional to the switching frequency
- Sampling frequency depends on the switching frequency
- If the sampling frequency is very high (as compared to the highest frequencies to be controlled), we may simply discretize the controller designed in the continuous-time domain
- If high sampling frequencies cannot be reached, sampling and hold should be properly taken into account to avoid poor performance
- Direct discrete-time design is an efficient method, considered in this course
- Specifications can still be given in the continuous-time domain, but they are mapped to the discrete-time domain

Digital Current Control System

- Current is sampled at the discrete time instants, in synchronism with the PWM
- PWM behaves as the zero-order hold (ZOH)
- Plant model will be mapped into the discrete-time domain using closed-form expressions and physical parameters (R and L)
- This makes it easier to include the magnetic saturation effects in the discrete-time algorithm later

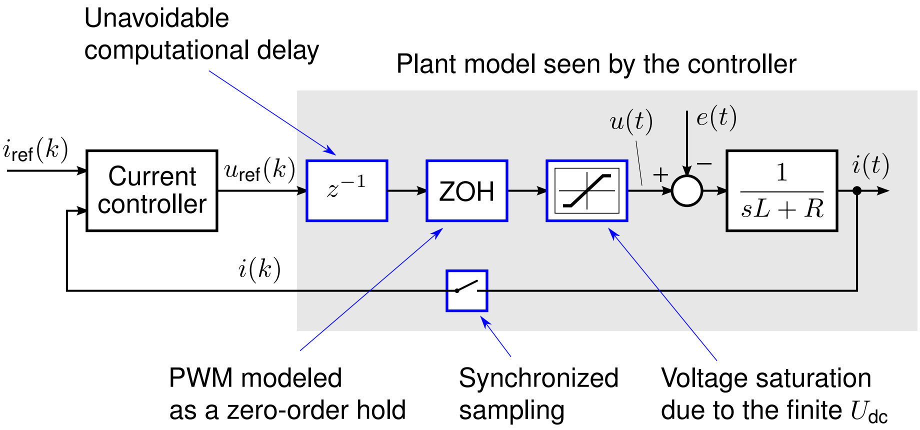


Discrete-Time Control Design Approaches

1. Discretized continuous-time design
   - Design the controller in the continuous-time domain
   - Discretize the controller using the Tustin or Euler method
   - Delays are difficult to handle
   - Use high sampling frequency (at least 10…20 times the closed-loop bandwidth)
2. Direct discrete-time design
   - Develop the hold-equivalent discrete-time plant model
     (exact model or series expansion, closed-form or numerical)
   - Delays can be easily included in the discrete-time model
   - Design the controller directly in discrete-time domain
   - Low sampling frequencies can be used (down to the Nyquist frequency)

The choice depends on the application.

Nonidealities due to Digital Implementation and Actuator Saturation



Outline

Euler and Tustin Methods

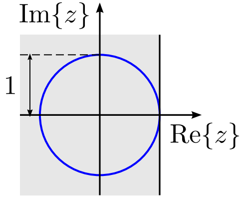
- Common methods to discretize continuous-time algorithms
- Euler method is very simple

$$\frac{\mathrm{d}x}{\mathrm{d}t} \leftarrow \frac{x(k+1) - x(k)}{T_\mathrm{s}} \qquad \text{or} \qquad s \leftarrow \frac{z-1}{T_\mathrm{s}}$$

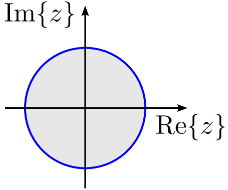
- Tustin method is more complicated

$$\frac{\mathrm{d}x}{\mathrm{d}t} \leftarrow \frac{2}{T_\mathrm{s}} \frac{x(k+1) - x(k)}{x(k+1) + x(k)} \qquad \text{or} \qquad s \leftarrow \frac{2}{T_\mathrm{s}} \frac{z-1}{z+1}$$

- Grey regions in the figure show how stable s-plane poles are mapped into the z-plane
- Tustin method maps the poles well, but time delays are still a remaining problem



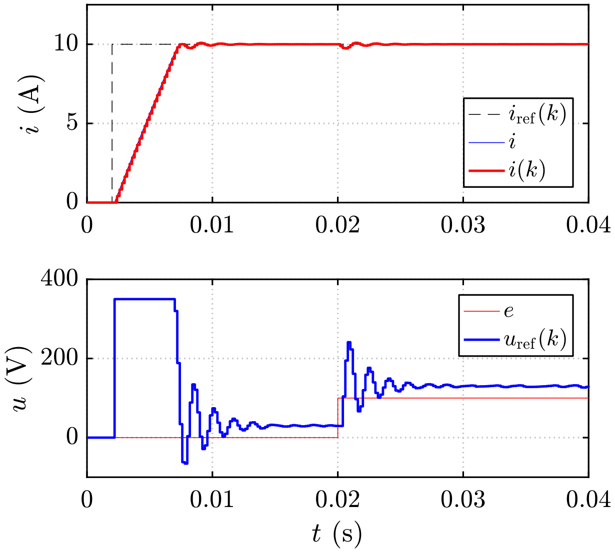
Euler



Tustin

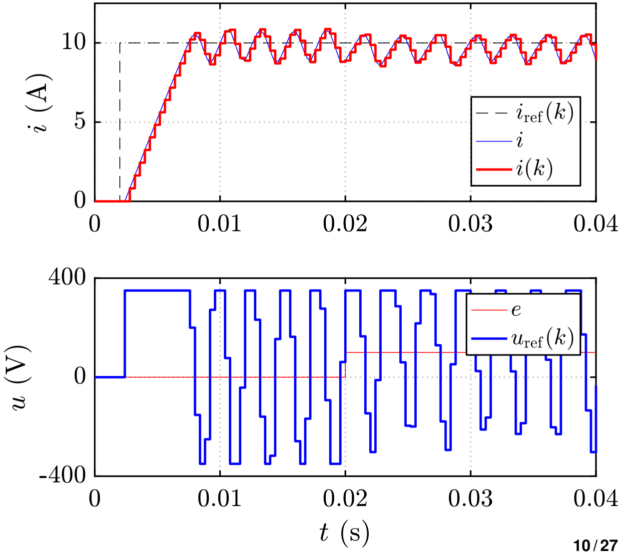
Exercise 3.1

- Augment your simulation model with the converter model, controlled with the unipolar PWM. Solver time step should be set a few decades shorter than the switching period for good accuracy.
- Discretize the controller using the Euler method and include the computational delay
- Simulate the model with $T_\mathrm{s} = 200\ \mu$s and $T_\mathrm{sw} = 400\ \mu$s



Exercise 3.2

- Reparametrize your model using $T_\mathsf{s} = T_\mathsf{sw} = 400\ \mu$s (single-update PWM)
- Simulate the model
- Discretization of the continuous-time controller deteriorates the control performance at this long sampling periods
- Note that the results are essentially the same even if you replace the PWM with the ZOH



Outline

Sampled-Data System

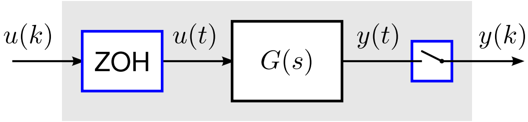
- Continuous-time system

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}u(t)$$
$$y(t) = \boldsymbol{C}\boldsymbol{x}(t)$$

- Integrating the state equation from t_0 to t gives

$$\boldsymbol{x}(t) = \mathrm{e}^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} \mathrm{e}^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}u(\tau)\mathrm{d}\tau$$

Sampled-data system $G(z)$



- Let us change the notation

$$t_0 = kT_\mathsf{s} \quad \text{and} \quad t = (k+1)T_\mathsf{s}$$

- Assume the ZOH at the control input

$$u(\tau) = u(kT_\mathsf{s}) \qquad kT_\mathsf{s} \leq \tau \leq (k+1)T_\mathsf{s}$$

Hold-Equivalent Discrete Model

- In digital control systems, assuming the ZOH at the control input typically agrees with the physical reality (also when the actuator is a power converter)
- Previous expressions lead to the discrete-time system model

$$\boldsymbol{x}(k+1) = \boldsymbol{\Phi}\boldsymbol{x}(k) + \boldsymbol{\Gamma}u(k)$$
$$y(k) = \boldsymbol{C}\boldsymbol{x}(k)$$

with exact hold-equivalent discrete-time matrices

$$\boldsymbol{\Phi} = \mathrm{e}^{\boldsymbol{A}T_\mathrm{s}} \qquad \boldsymbol{\Gamma} = \int_0^{T_\mathrm{s}} \mathrm{e}^{\boldsymbol{A}\tau}\mathrm{d}\tau\boldsymbol{B}$$

- System can also be expressed in the z-domain

$$G(z) = \frac{y(z)}{u(z)} = \boldsymbol{C}(z\mathbf{I} - \boldsymbol{\Phi})^{-1}\boldsymbol{\Gamma}$$

Series Expansion

- Series expansion

$$\boldsymbol{\Phi} = \mathbf{I} + T_{\mathsf{s}} \boldsymbol{\Psi} \boldsymbol{A} \qquad\qquad \boldsymbol{\Gamma} = T_{\mathsf{s}} \boldsymbol{\Psi} \boldsymbol{B}$$

where $\mathbf{I}$ is the identity matrix and

$$\boldsymbol{\Psi} = \mathbf{I} + \frac{T_{\mathsf{s}} \boldsymbol{A}}{2!} + \frac{T_{\mathsf{s}}^2 \boldsymbol{A}^2}{3!} + \dots$$

- Note that infinite series yields also exact model
- Choosing $\boldsymbol{\Psi} = \mathbf{I}$ yields the Euler approximation

Selection of State Variables

- State variables can be chosen arbitrarily in linear models (and in resulting controllers and observers) and they can be easily changed afterwards
- Different selections lead to mathematically equivalent control laws
- If the system is nonlinear, suitable selection of state variables may reduce the complexity of the model significantly
- Later we will consider the effects of magnetic saturation
- Saturation effects are easier to handle if the flux linkage ψ is the state variable
- For this reason, the flux linkage ψ will be chosen as a state variable (even if we first consider a magnetically linear system)

Hold-Equivalent Discrete Model: Example System

- Continuous-time system model

$$\frac{\mathrm{d}\psi(t)}{\mathrm{d}t} = -(R/L)\psi(t) + u(t) - e(t)$$
$$i(t) = \psi(t)/L$$

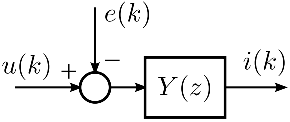
- Discrete-time system model

$$\psi(k+1) = \phi\psi(k) + \gamma[u(k) - e(k)]$$
$$i(k) = \psi(k)/L$$

- Hold-equivalent discrete-time coefficients

$$\phi = \mathrm{e}^{-(R/L)T_\mathrm{s}} \qquad \gamma = \int_0^{T_\mathrm{s}} \mathrm{e}^{-(R/L)\tau}\mathrm{d}\tau = \frac{1-\mathrm{e}^{-(R/L)T_\mathrm{s}}}{R/L}$$

ZOH is assumed for disturbance input $e(t)$ as well for simplicity



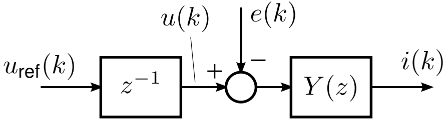
$$Y(z) = \frac{1}{L}\frac{\gamma}{z-\phi}$$

Inclusion of the Computational Delay

- Computational delay can be added as an extra state

$$\begin{aligned} \psi(k+1) &= \phi\psi(k) + \gamma[u(k) - e(k)] \\ u(k+1) &= u_{\text{ref}}(k) \\ i(k) &= \psi(k)/L \end{aligned}$$

- Both states are readily available for feedback



Transfer Function

- Transfer function from the reference voltage to the measured current

$$\frac{i(z)}{u_{\mathrm{ref}}(z)} = z^{-1}Y(z) = \frac{1}{L}\frac{\gamma}{z(z-\phi)}$$

- Open-loop poles at $z = \phi = \mathrm{e}^{-(R/L)T_{\mathrm{s}}}$ and at $z = 0$

Outline

Discrete-Time Controller

- Full state-feedback control law

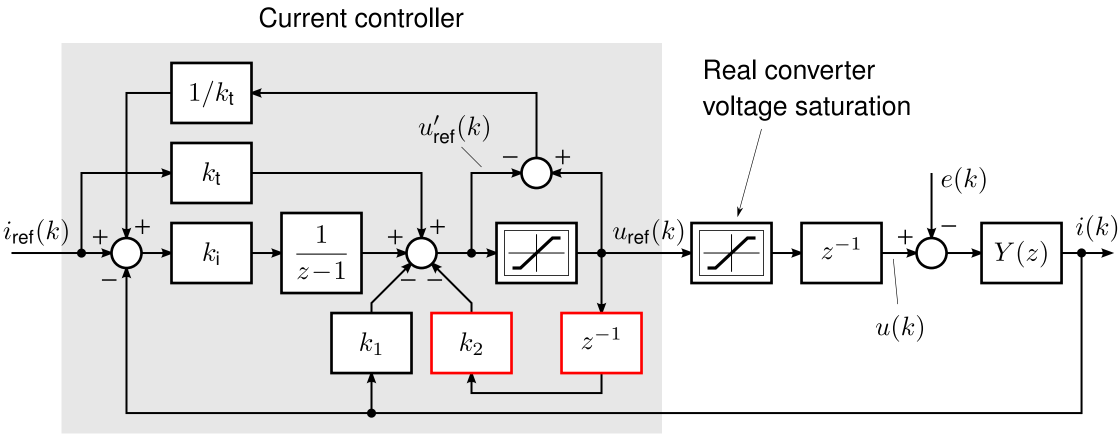
$$\begin{aligned} u_{\mathsf{i}}(k+1) &= u_{\mathsf{i}}(k) + k_{\mathsf{i}}[i_{\mathsf{ref}}(k) - i(k)] \\ u(k+1) &= u_{\mathsf{ref}}(k) \\ u_{\mathsf{ref}}(k) &= k_{\mathsf{t}} i_{\mathsf{ref}}(k) - k_1 i(k) - k_2 u(k) + u_{\mathsf{i}}(k) \end{aligned}$$

- Control law in the z-domain

$$u_{\mathsf{ref}}(z) = k_{\mathsf{t}} i_{\mathsf{ref}}(z) - k_1 i(z) + \frac{k_{\mathsf{i}}}{z-1}[i_{\mathsf{ref}}(z) - i(z)] - k_2 z^{-1} u_{\mathsf{ref}}(z)$$

- Antiwindup can be added similarly as in the continuous-time case
- Choosing $k_t = k_1$ and $k_2 = 0$ yields the PI controller

Discrete-Time Controller



- Only the red blocks have to be added to the PI controller
- Poles can now be arbitrarily placed

Closed-Loop System

- Closed-loop transfer function from the reference to the measured current

$$\frac{i(z)}{i_{\mathsf{ref}}(z)} = \frac{b_1 z + b_0}{z^3 + a_2 z^2 + a_1 z + a_0}$$

- Coefficients depend on the controller gains

$$a_2 = k_2 - \phi - 1 \qquad a_1 = \phi(1 - k_2) - k_2 + \frac{\gamma k_1}{L} \qquad a_0 = k_2\phi + \frac{\gamma(k_{\mathsf{i}} - k_1)}{L}$$

$$b_1 = \frac{\gamma k_{\mathsf{t}}}{L} \qquad b_0 = \frac{\gamma(k_{\mathsf{i}} - k_{\mathsf{t}})}{L}$$

Pole and Zero Placement

- To simplify notation, accurate parameters are assumed
- Coeffients (poles and zero) can be arbitrarily placed

$$k_{\mathrm{t}} = \frac{b_1 L}{\gamma} \qquad k_2 = a_2 + \phi + 1 \qquad k_1 = \frac{[a_1 - \phi(1 - k_2) + k_2]L}{\gamma} \qquad k_{\mathrm{i}} = k_1 + \frac{(a_0 - k_2\phi)L}{\gamma}$$

- Let us choose $a_0 = 0$ due to the computational delay
- Example selection for other coefficients

$$a_1 = \beta^2 \qquad a_2 = -2\beta \qquad b_1 = 1 - \beta$$

- Resulting closed-loop system

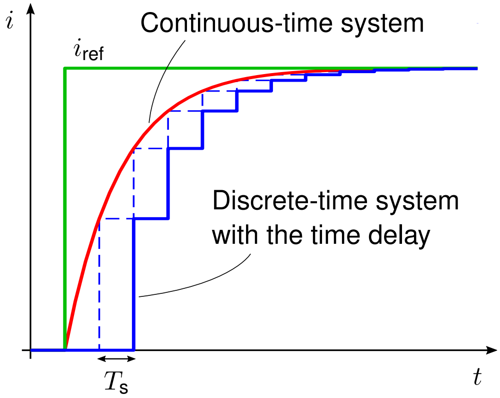
$$\frac{i(z)}{i_{\mathrm{ref}}(z)} = \frac{(1-\beta)(z-\beta)}{z(z-\beta)^2} = \frac{1-\beta}{z(z-\beta)}$$

Desired Closed-Loop Reference Tracking

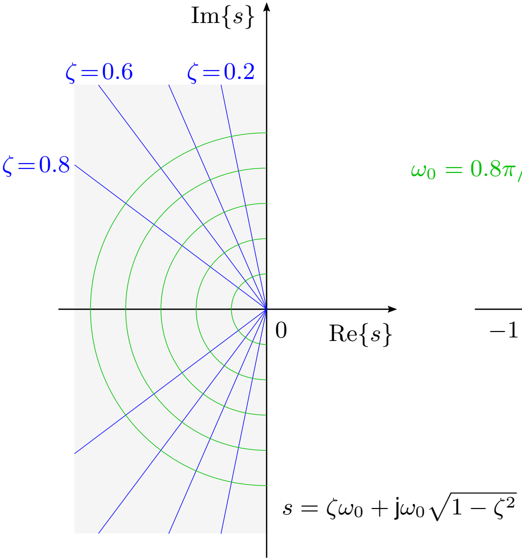
- 1st-order unity-gain LPF $+$ time delay

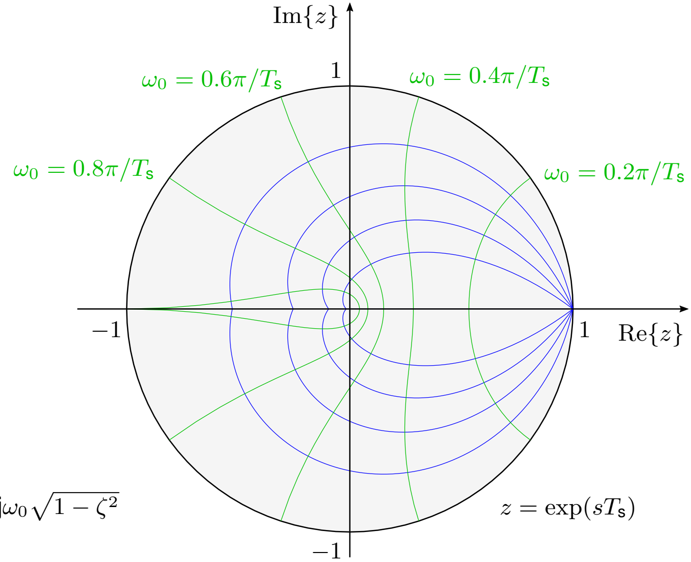
$$\frac{i(z)}{i_{\mathrm{ref}}(z)} = \frac{1-\beta}{z(z-\beta)}$$

- $\beta = \mathrm{e}^{-\alpha_{\mathrm{c}} T_{\mathrm{s}}}$ and α_{c} is the desired continuous-time real pole
- Delay z^{-1} cannot be avoided in practice
- Same "user input" as in the continuous-time case (α_{c}, R, L)



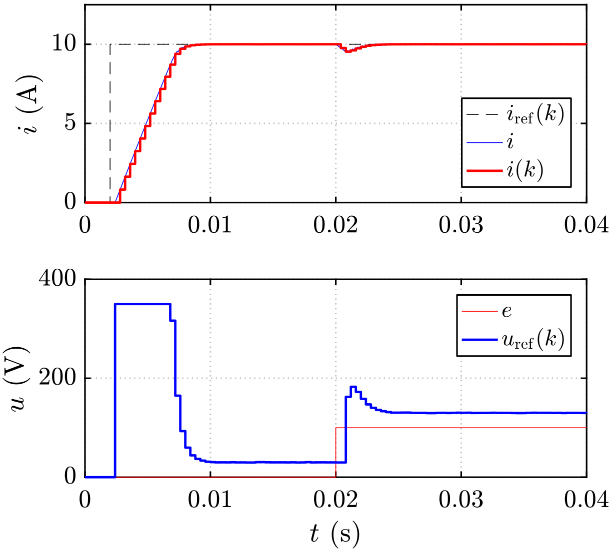
Mapping from s-Plane to z-Plane





Exercise 3.3

- Modify the current controller according to the presented direct discrete-time design approach
- Simulate the model with $\alpha_\mathrm{c} = 2\pi \cdot 300$ rad/s and $T_\mathrm{s} = T_\mathrm{sw} = 400\ \mu$s
- Test also higher bandwidths, model parameter errors, and sensitivity to the measurement noise



Further Reading

- G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*. Menlo Park, CA: Addison-Wesley, 1997.
- M. Hinkkanen, H. A. A. Awan, Z. Qu, T. Tuovinen, and F. Briz, “Current control for synchronous motor drives: Direct discrete-time pole-place- ment design,” *IEEE Trans. Ind. Applicat.*, 2016.