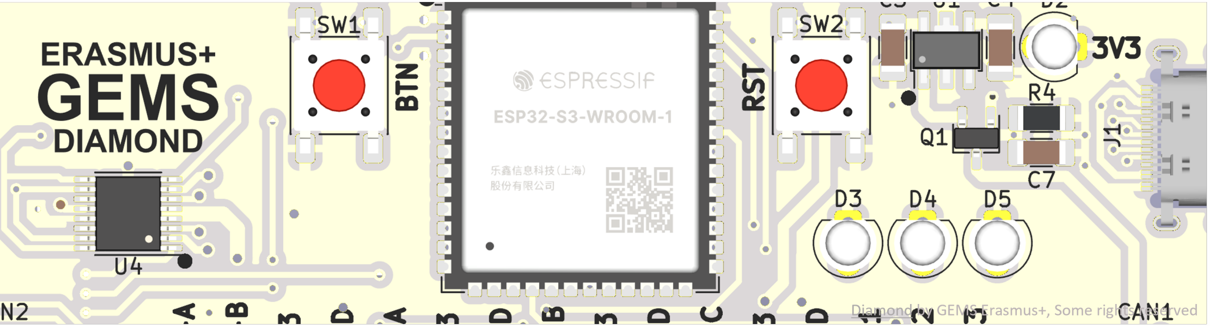


Tutorial 5

Implementation of Motor Model in a Microcontroller

Dr. ir. Sachin Yadav
DCE&S, ESE, TU Delft

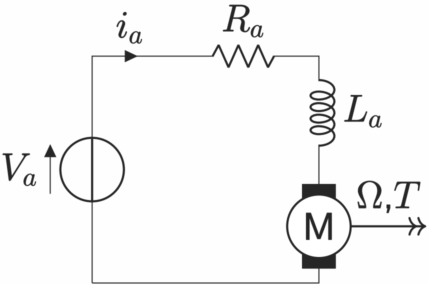
1. Motor model
2. Continuous to Discrete time domain conversion
3. Implementation in ESP32 microcontroller

1. Motor model

2. Continuous to Discrete time domain conversion

3. Implementation in ESP32 microcontroller

Motor model



$$\frac{\Omega(s)}{V_a(s)} = \frac{K}{(sL_a + R_a)(sJ + b) + K^2} \qquad (1)$$

1. Motor model
2. Continuous to Discrete time domain conversion
3. Implementation in ESP32 microcontroller

Conversion methods

Table: Continuous to discrete conversion methods [1].

Method	Use case/advantage
Zero-order hold	For staircase inputs
First-order hold	For piecewise linear inputs
Impulse-invariant mapping	For impulse train inputs
Tustin approximation	Frequency domain match, capture dynamics at particular frequencies
Zero-pole matching equivalents	SISO model, good frequency match
Least-squares	SISO model, good frequency match, capture fast dynamics

Tustin approximation

$$z = e^{sT_s} \tag{2}$$

$$= \frac{e^{\frac{sT_s}{2}}}{e^{\frac{-sT_s}{2}}} \tag{3}$$

$$\approx \frac{1 + sT_s/2}{1 - sT_s/2} \tag{4}$$

where, T_s is the sample time of the controller. Inverting, we get

$$s = \frac{2}{T_s}\frac{1 - z^{-1}}{1 + z^{-1}} \tag{5}$$

Motor model in discrete time domain

Substituting (5) in (1), we get:

$$\frac{\Omega(z)}{V_a(z)} = KT_s^2 \frac{1 + 2z^{-1} + z^{-2}}{a + bz^{-1} + cz^{-2}} \tag{6}$$

where,

$$\begin{aligned}
a =& (K^2 + bR_a)T_s^2 + 2T_s(JR_a + bL_a) + 4JL_a \\
b =& 2(K^2 + bR_a)T_s^2 - 4JL_a \\
c =& (K^2 + bR_a)T_s^2 - 2T_s(JR_a + bL_a) + 4JL_a
\end{aligned}$$

1. Motor model

2. Continuous to Discrete time domain conversion

3. Implementation in ESP32 microcontroller

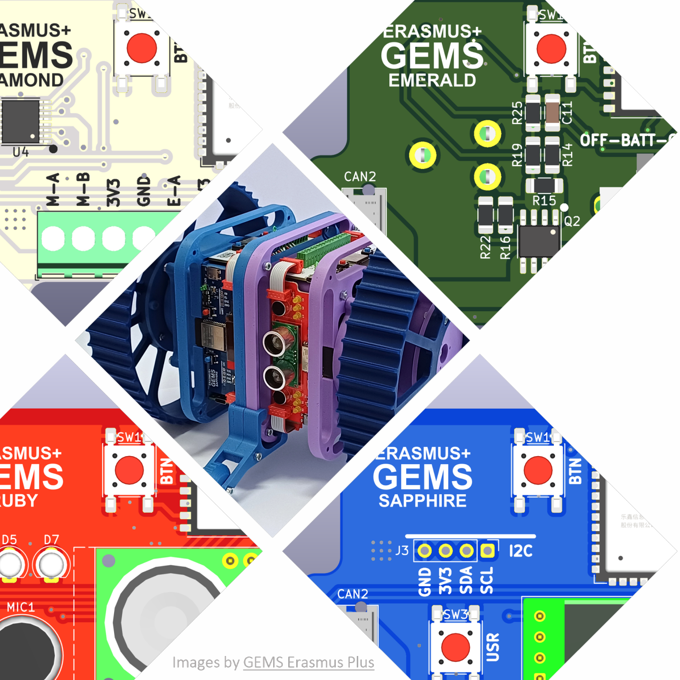
Difference equation

To implement the model in a microcontroller, a difference equation can be formed with z^{-1} as the one-sample delay operator.

$$\begin{aligned}\Omega[n] =& a_1 V_a[n] + a_2 V_a[n-1] + a_3 V_a[n-2] \\ & - b_1 \Omega[n-1] - b_2 \Omega[n-2]\end{aligned}$$

Conclusion

- Review of motor model.
- Review of voltage to speed transfer function.
- Continuous time domain to discrete time domain conversion.
- Creation of the difference equation.
- Implementation of difference equation into an ESP32 microcontroller.



Images by GEMS Erasmus Plus

Thank you for watching!

This video was created by Delft University of Technology for the GEMS Erasmus+ project (a collaboration between University of Ljubljana, University of Alcala, Teaching Factory Competence Center, and Delft University of Technology). It is released under a **Creative Commons Attribution Non Commercial Share Alike 4.0 International License**

Co-funded by the European Union

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CMEPIUS. Neither the European Union nor the granting authority can be held responsible for them.

References I

[1] Mathworks. *Continuous-Discrete Conversion Methods*. Oct. 2025. URL: https://www.mathworks.com/help/control/ug/continuous-discrete-conversion-methods.html.