

Szabályozott villamos hajtások szimulációja Phyton alapú környezetben

SZABÓ VENCEL

BME GPK Mechatonikai mérnök MSc

Témavezető: Stumpf Péter

Projektfeladat védeés



MŰEGYETEM 1782

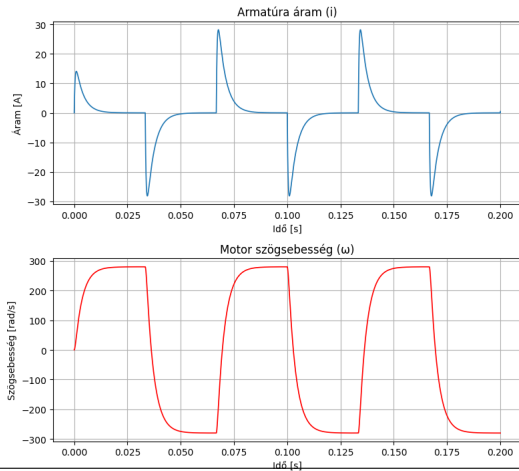
Modell: Az egyenáramú motor viselkedését az alábbi egyenletrendszer írja le:

$$\begin{aligned}\frac{di(t)}{dt} &= \frac{u(t) - Ri(t) - k\omega(t)}{L} \\ \frac{d\omega(t)}{dt} &= \frac{ki(t) - b\omega(t) - T_{\text{load}}}{J}\end{aligned}$$

Szabályozás és szimuláció:

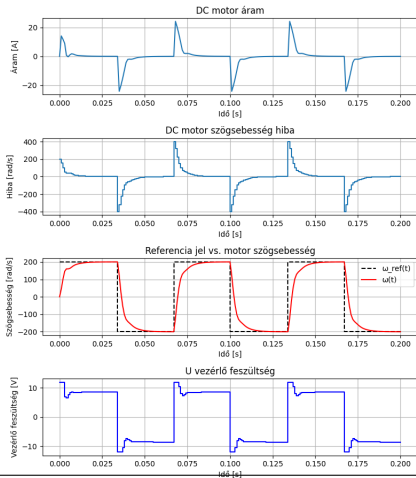
- ▶ Egyszerű P-szabályzó: $u(t) = K_p \cdot (\omega_{\text{ref}}(t) - \omega(t))$
- ▶ A referenciajel négyszögjel ± 300 rad/s között.
- ▶ A szimuláció során minden szabályozási lépés után a motor dinamikáját integráljuk.
- ▶ A rendszer numerikus megoldása az RK45 metódussal történik.

DC motor szimuláció (négysszögjel bemenettel)



DC motor P szabályzóval

PI szabályzó



Cél: A DC motor PI-alapú sebesség- és áramvezérlése **STM32 mikrokontrollerrel.**

- ▶ **PWM kimenet:** A TIM1 modul biztosítja a motor meghajtásához szükséges PWM jelet.
- ▶ **Encoder bemenet:** A TIM2 számláló az inkrementális enkóderből olvassa a pozíciót és a sebességet.
- ▶ **Árammérés:** A ADC1 analóg bemenetről olvassa az armatúra áramát.
- ▶ **Időzítés:** A TIM3 időzítő 100 ms-onként megszakítást generál (sebességméréshez).
- ▶ **Két szabályozási hurok:**
 - ▶ **Sebesség PI:** 1 ms időzítéssel fut (referencia alapján kiszámítja az áramreferenciát)
 - ▶ **Áram PI:** 0.1 ms időzítéssel fut (referenciaáram alapján szabályozza a PWM duty-t)

A szabályozás lényege: **sebesség** → **áram referencia** → **áram szabályozás** → **PWM jelek.**

1. Sebesség szabályozó hurok (külső hurok):

- ▶ Cél: a motor szögsebessége kövesse a referenciajel (pl. négyszögjel) értékét.
- ▶ A hiba: $\omega_{\text{ref}} - \omega_{\text{measured}}$
- ▶ A PI szabályozó kiszámítja az áram referencia értékét: i_{ref}
- ▶ Frissítési periódus: $T_s = 1 \text{ ms}$

2. Áram szabályozó hurok (belső hurok):

- ▶ Cél: a tényleges motoráram kövesse az i_{ref} értéket.
- ▶ A hiba: $i_{\text{ref}} - i_{\text{measured}}$
- ▶ A PI szabályozó meghatározza a vezérlő feszültséget \rightarrow PWM duty cycle
- ▶ Frissítési periódus: $T_s = 0.1 \text{ ms}$

A gyorsabb belső hurok (áram) garantálja a stabilitást, míg a lassabb külső hurok (sebesség) a kívánt dinamikát szabja meg.

Sebesség szabályozó:

$$u_{\text{speed}}(k) = K_p^{\text{speed}} \cdot \left(e(k) + K_i^{\text{speed}} \cdot T_s^{\text{speed}} \cdot \sum_{i=0}^k e(i) \right)$$

$$e(k) = \omega_{\text{ref}}(k) - \omega(k)$$

Áram szabályozó:

$$u_{\text{current}}(k) = K_p^{\text{current}} \cdot \left(e_i(k) + K_i^{\text{current}} \cdot T_s^{\text{current}} \cdot \sum_{j=0}^k e_i(j) \right)$$

$$e_i(k) = i_{\text{ref}}(k) - i_{\text{measured}}(k)$$

Szaturáció:

$$u(k) = \begin{cases} V_{\max} & \text{ha } u(k) > V_{\max} \\ u(k) & \text{ha } -V_{\max} \leq u(k) \leq V_{\max} \\ -V_{\max} & \text{ha } u(k) < -V_{\max} \end{cases}$$

Anti-windup logika:

- ▶ Ha a szabályozó kimenete eléri a szaturációs határt, akkor:

integráció letiltva $\Rightarrow \sum e(k)$ nem frissül

- ▶ Csak akkor integrál, ha $|u(k)| < V_{\max}$

PWM duty számítása:

$$\text{duty} = \frac{u_{\text{current}}}{V_{\max}}$$