Aszinkron gépek irányítási algoritmusainak fejlesztése

Horváth Krisztián

Széchenyi István Egyetem

„A villamos hajtású járművek (e-mobilitás) jelene és jövője”
szakmai nap

2017. május 3.

Bevezető

Az előadás célja bemutatni

- egy modellalapú fejlesztési módszert,
- egy moduláris, újrafelhasználható modellarchitektúrát,
- a MATLAB/Simulink által biztosított funkciók kiterjesztett alkalmazását,
- egy szigorú, következetes koncepciót

kalickás forgórészű aszinkron gépek irányítási algoritmusainak fejlesztéséhez.

Az előadás tartalma:

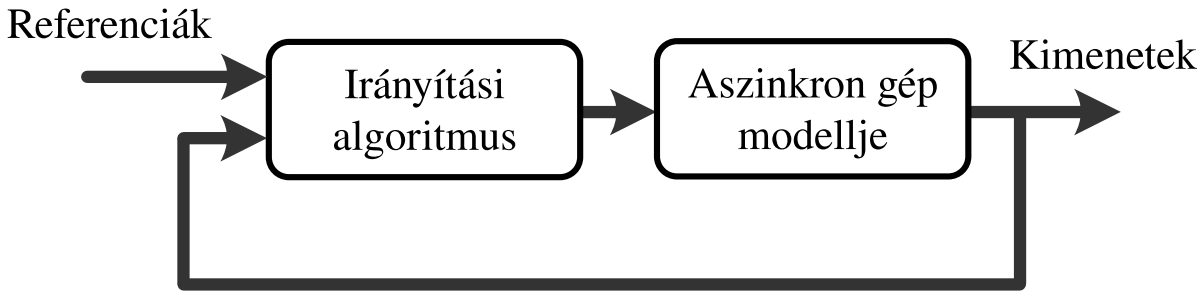
- Alkalmazott modellek és algoritmusok
- Implementációs módszerek és modellarchitektúra
- Implementációs példa
- Következtetések és kitekintés

Alkalmazott modellek és algoritmusok

Alkalmazott modellek és algoritmusok

A modellalapú fejlesztés olyan módszer, amely gyorsítja a fejlesztést és csökkenti a hibák előfordulásának valószínűségét. Alkalmazásához szükség van a gép és az irányítási algoritmus modelljére:

- Aszinkron gép modellje
  - Villamos modell
  - Mechanikus modell
- Mezőorientált szabályozási algoritmus



A szabályozási kör egyszerűsített hatásvázlata

Kalickás forgórészű aszinkron gép modellje[1]

Feszültségegyenletek:

$$v_{ds} = R_s i_{ds} + \frac{\mathrm{d}}{\mathrm{dt}}\psi_{ds} - \omega\psi_{qs} \qquad 0 = R_r i_{dr} + \frac{\mathrm{d}}{\mathrm{dt}}\psi_{dr} - (\omega - \omega_r)\psi_{qr}$$

$$v_{qs} = R_s i_{qs} + \frac{\mathrm{d}}{\mathrm{dt}}\psi_{qs} + \omega\psi_{ds} \qquad 0 = R_r i_{qr} + \frac{\mathrm{d}}{\mathrm{dt}}\psi_{qr} + (\omega - \omega_r)\psi_{dr}$$

Fluxusegyenletek:

$$\psi_{ds} = L_s i_{ds} + L_m i_{dr} \qquad \psi_{dr} = L_r i_{dr} + L_m i_{ds}$$

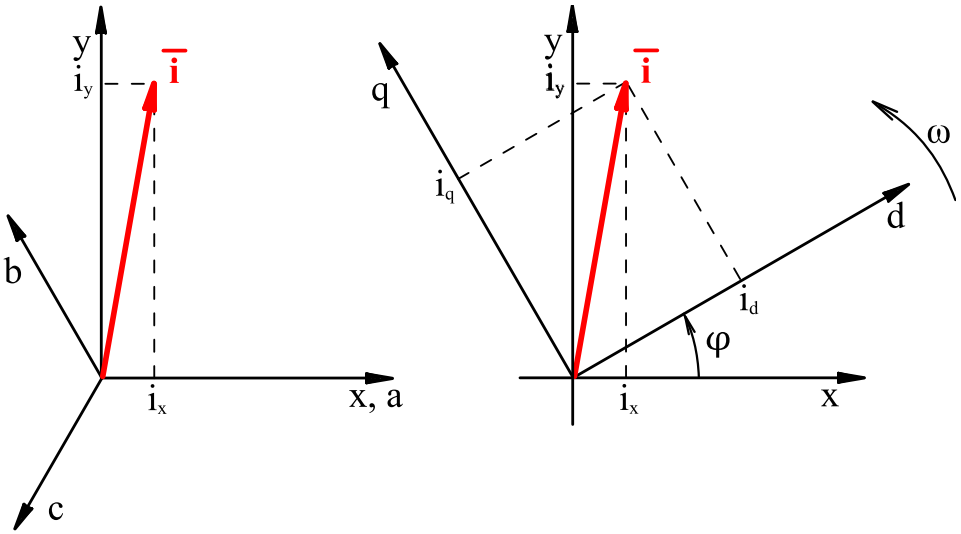
$$\psi_{qs} = L_s i_{qs} + L_m i_{qr} \qquad \psi_{qr} = L_r i_{qr} + L_m i_{qs}$$

Villamos nyomaték:

$$T_e = \frac{3}{2} N L_m (i_{qs} i_{dr} - i_{ds} i_{qr})$$

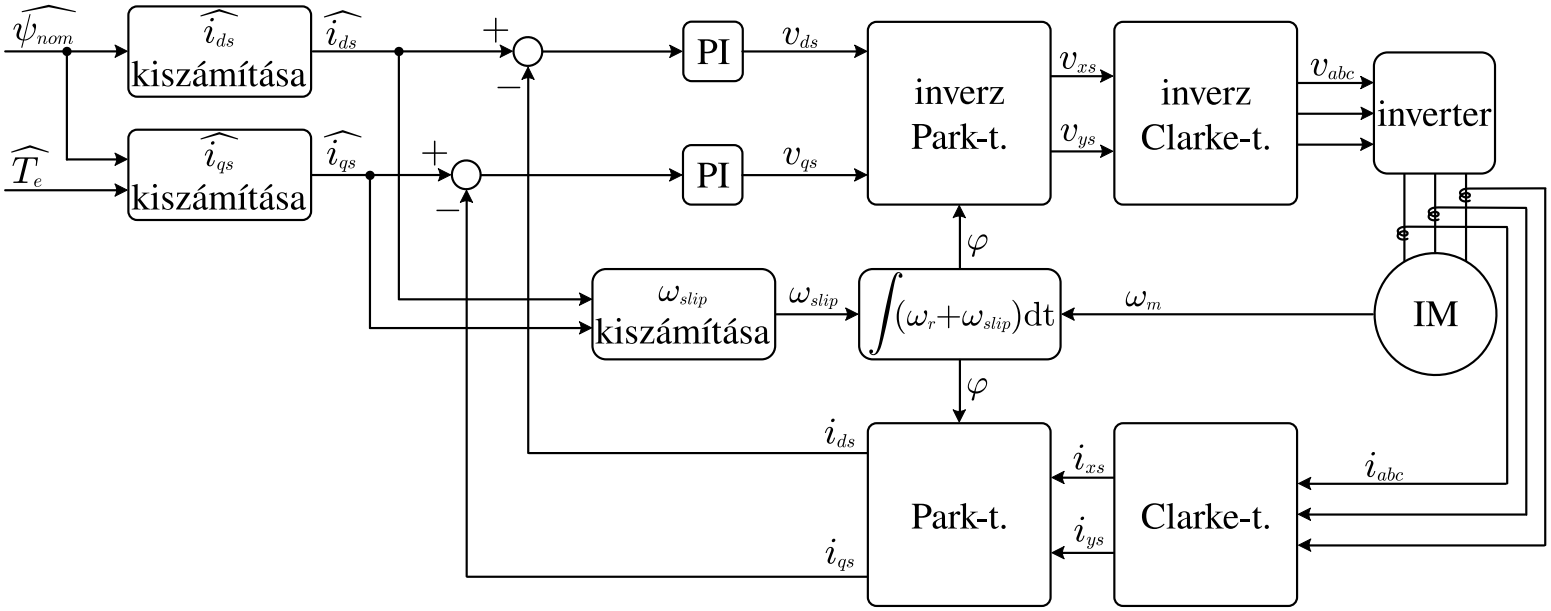
Mozgásegyenlet:

$$\frac{\mathrm{d}}{\mathrm{dt}}\omega_m = \frac{T_e - T_{load} - D\omega_m}{\theta}$$



[1] S. Filizadeh, *Electric Machines and Drives: Principles, Control, Modeling, and Simulation*. Taylor & Francis, 2013.

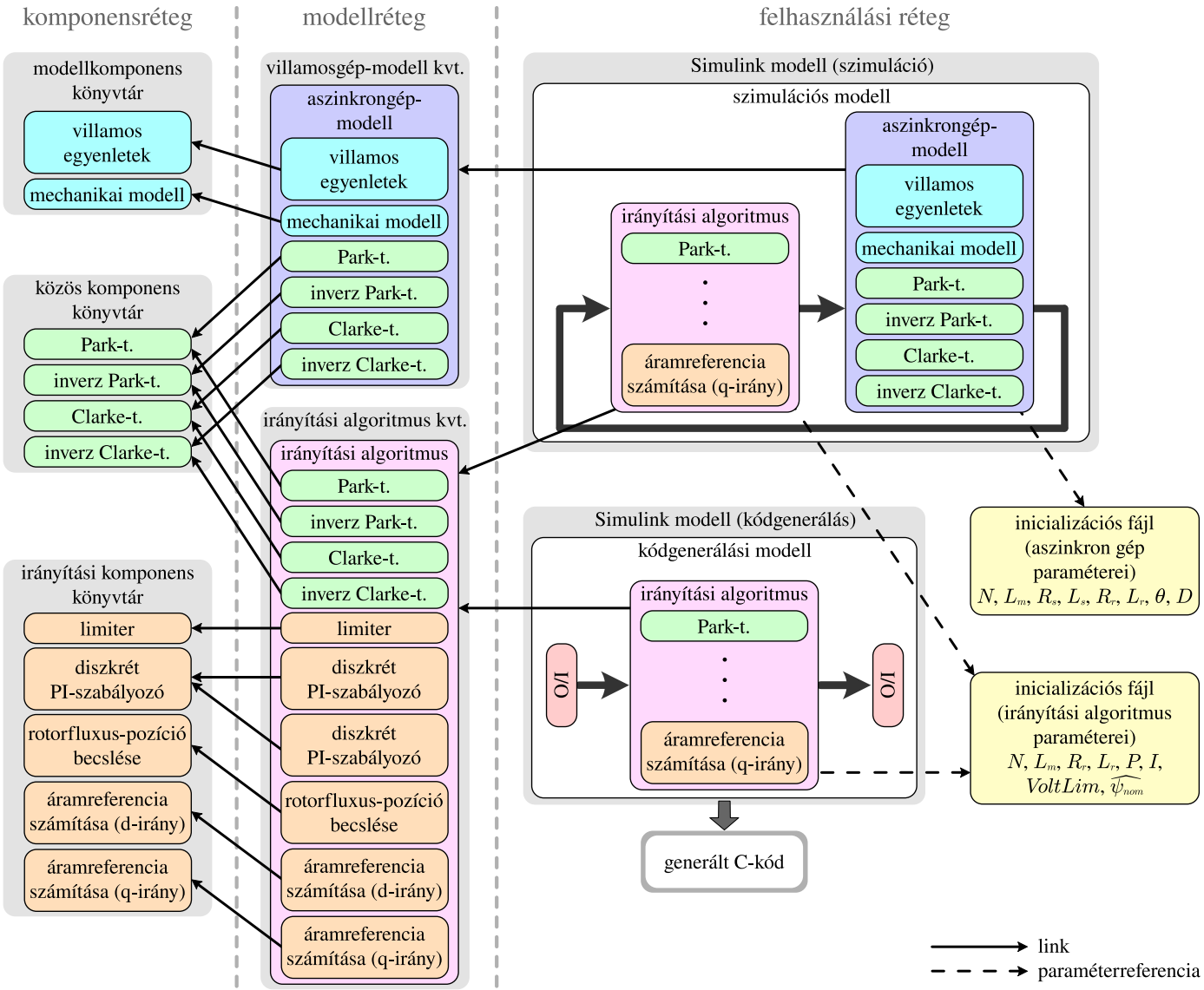
Mezőorientált szabályozási algoritmus



Az indirekt mezőorientált szabályozási algoritmus egy lehetséges blokkvázlata

Implementációs módszerek és modellarchitektúra

Implementációs módszerek és modellarchitektúra



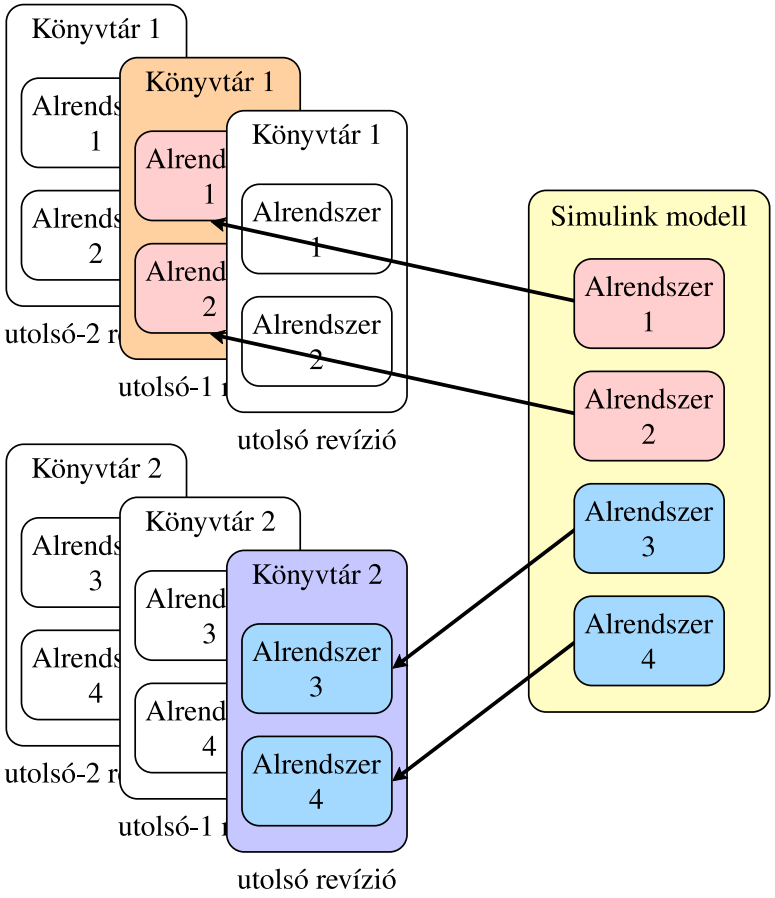
Blokk-könyvtárak és modellek architektúrája

Implementációs módszerek és modellarchitektúra

Az alkalmazott modellarchitektúra főbb előnyei:

- A MATLAB/Simulink olyan algoritmusfejlesztő eszközöket biztosít, amelyek hasonlóak az objektumorientált programozási módszerekhez.
- Különböző paraméterfájlokkal könnyen adaptálható a rendszer különböző forgógépekhez.
- Az alkalmazott architektúra megkönnyíti a szoftverkarbantartást.
- Az eszközök biztosítják a beágyazott rendszereken alkalmazható kód automatizált generálást.
- A komponensek többszörösen újrafelhasználhatóak.
- Verziókezelő rendszer alkalmazásával biztosítható a fejlesztés különböző lépésihez tartozó komponensek szétválasztása.

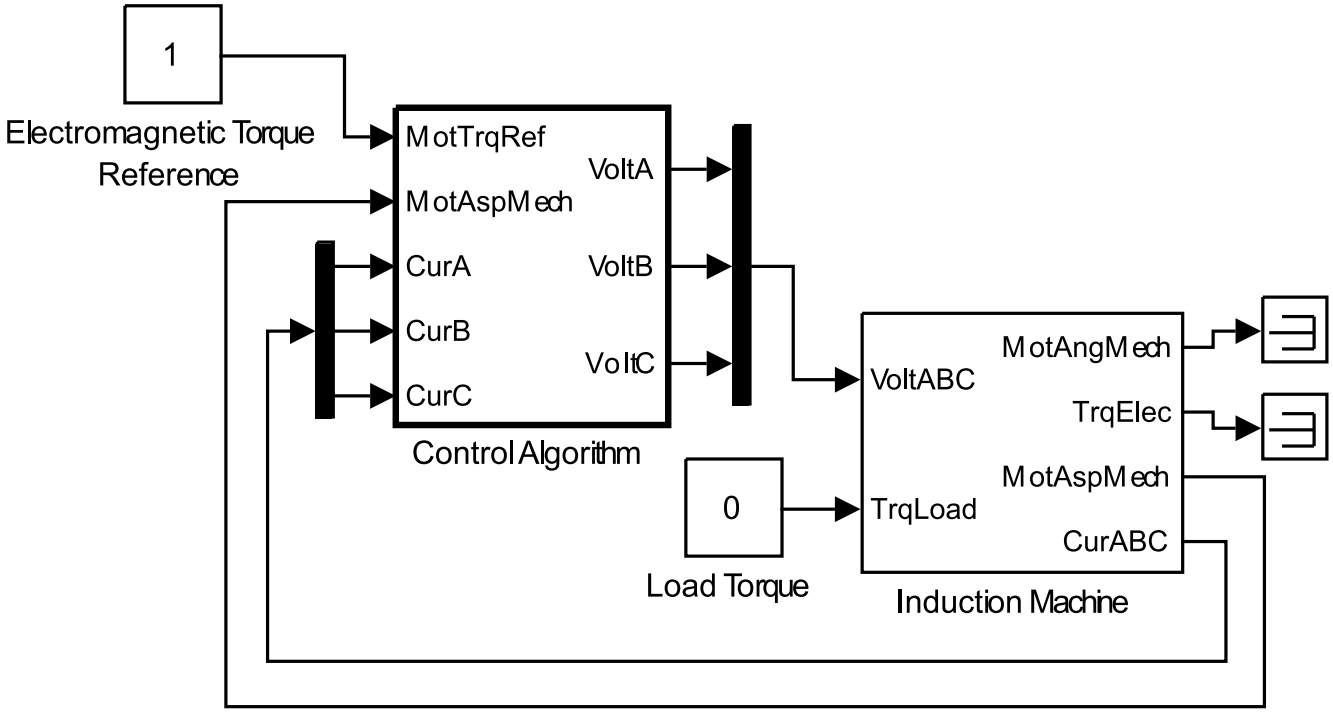
Implementációs módszerek és modellarchitektúra



Különböző változatokhoz tartozó komponensek alkalmazása

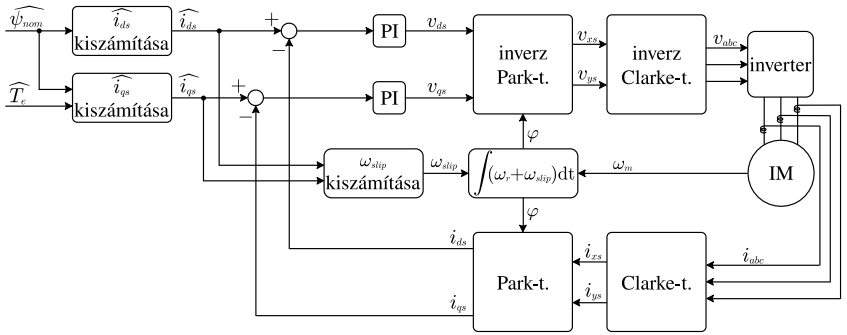
Implementációs példa

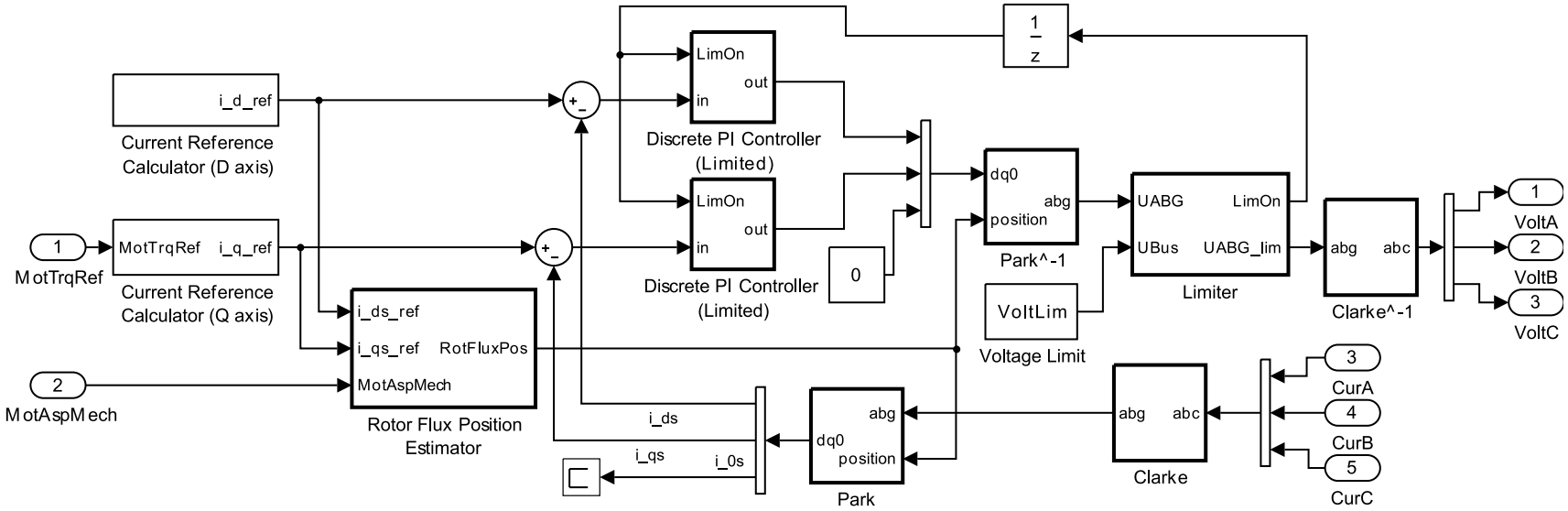
Implementációs példa



A felső szintű szimulációs modell implementációja Simulink környezetben

Implementációs példa

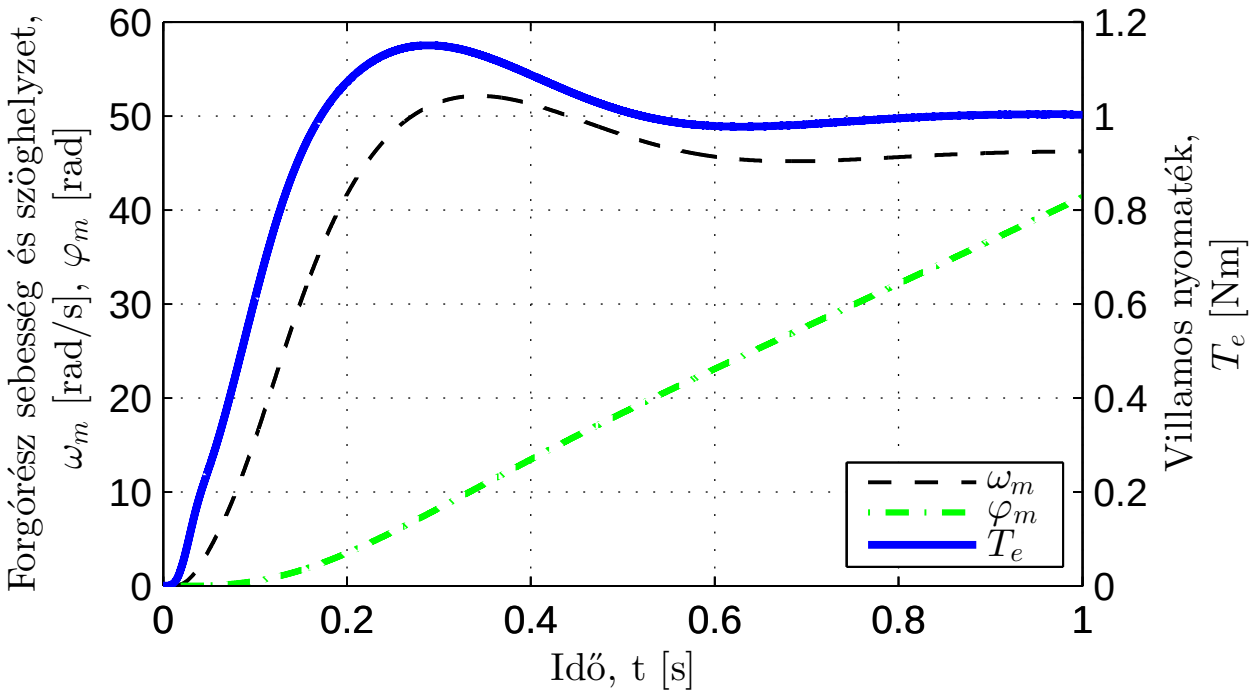




Az IFOC algoritmus implementációja Simulink környezetben

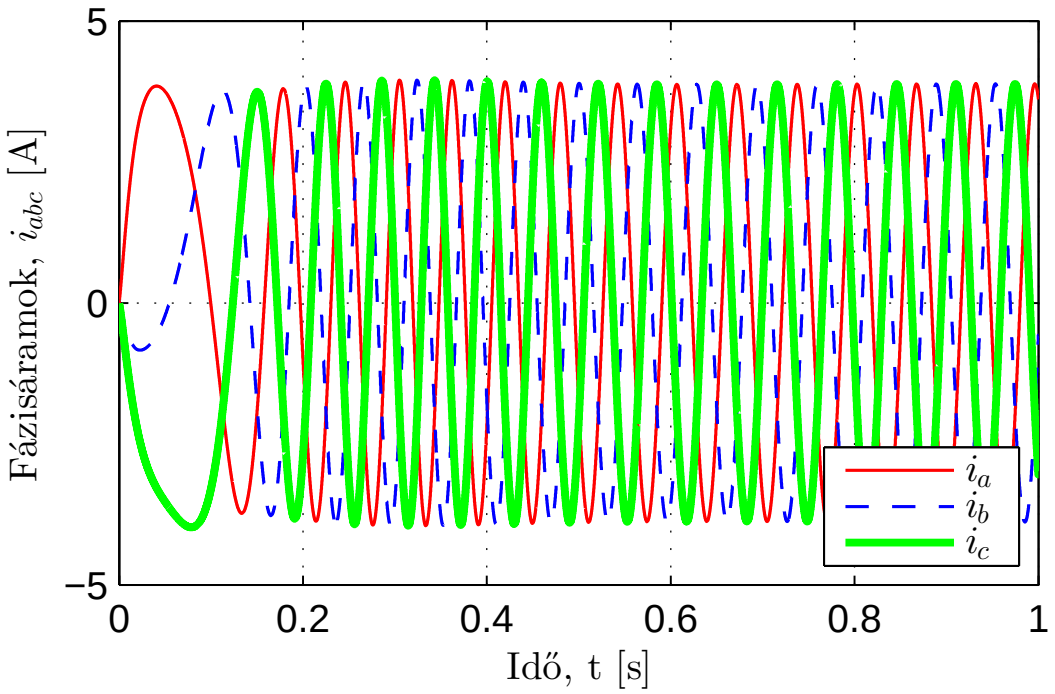
Implementációs példa

A szimulációs példa során a refencianyomaték: $\widehat{T_e} = 1\,\mathrm{Nm}$



A szabályozott aszinkron gép villamos nyomatéka, forgórészének szögsebessége és pozíciója

Implementációs példa



Az aszinkron gép fázisáramai

Következtetések és kitekintés

Következtetések és kitekintés

Az előadáson láthattunk

- egy alkalmazást a modellalapú tervezési módszertanra,
- egy moduláris, újrafelhasználható modellarchitektúrát,
- egy implementációs példát

aszinkron gépek irányítási algoritmusainak fejlesztése kapcsán.

A megkezdett munkát folytatva, a jövőbeli kutatási célok közt sebességszenzor nélküli és robusztus irányítási prototípus-algoritmusok kidolgozása szerepel. Az előadásban ismertetett módszer hatékonyan alkalmazható az ehhez kapcsolódó feladatok megoldásához.

Köszönöm a figyelmet!