



BERTÓK ZSANETT

PMSM hajtások paraméteridentifikációja

Konzulensek:

Dr. Budai Csaba

adjunktus (BME-GPK MOGI)

Dr. Stumpf Péter Pál

egyetemi docens (BME-VIK AUT)

Tudományos Diákköri Konferencia

Budapest, 2022

Tartalomjegyzék

Összefoglaló	3
Abstract	4
1. Bevezetés	5
2. Elméleti háttér	7
2.1. Az állandómágneses szinkron motor gépmodellje	7
2.2. Diszkrét modell létrehozása az identifikációhoz	10
2.3. Transzformációk a koordinátarendszerek között	11
2.4. Vivőfrekvenciás impulzusszélesség-moduláció	12
3. A paraméteridentifikációhoz használt módszerek	15
3.1. Rekurzív legkisebb négyzetek algoritmus	15
3.1.1. PMSM motor RLS identifikációja	17
3.2. ARMAX modell identifikációs módszer	18
3.2.1. A PMSM identifikációja az ARMAX modell predikciós hibájának becslésével	20
3.3. A <i>Trust Region</i> optimalizáció	21
3.3.1. A PMSM motor identifikációja <i>Trust Region</i> algoritmussal	22
4. Szimulációs eredmények	24
4.1. Az ellenállás és az induktivitás becslése	25
4.1.1. RLS algoritmus	25
4.1.2. ARMAX modell identifikáció	26
4.1.3. <i>Trust Region</i> optimalizáció	27
4.2. A pólusfluxus és nyomatékonstans becslése	29
4.2.1. ARMAX modell identifikáció	29
4.2.2. <i>Trust Region</i> optimalizáció	30
4.3. A mechanikai paraméterek becslése	32
4.3.1. RLS identifikáció	33
4.3.2. ARMAX modell identifikáció	33
4.3.3. A <i>Trust Region</i> algoritmus	34
4.3.4. Négyyszögjel gerjesztéssel történő identifikáció vizsgálata	35
4.4. A szimulációk alapján javasolt identifikációs eljárás	36
5. Hardveres megvalósítás	38
5.1. A rendszer áttekintése	38
5.2. ePWM modul	39

5.3.	A szögsebesség mérése	40
5.4.	Az áram mérése	41
5.5.	A program működése	42
5.5.1.	A kezdeti pozíció beállítása	43
5.6.	Az identifikáció elvégzése	44
5.7.	A zárt hurkú szabályozás megvalósítása	45
5.7.1.	Az áramszabályozás hangolása a villamos paraméterek alapján . . .	47
5.7.2.	A szögsebességszabályozás hangolása a mechanikai paramétereiből .	47
6.	A valós mérés eredményei	49
6.1.	<i>Trust Region</i> algoritmus	49
6.2.	RLS módszer	50
6.3.	A holtidő kompenzációja	51
6.3.1.	A <i>Trust Region</i> optimalizálás eredményei a holtidőkompenzáció után	54
6.3.2.	Az RLS optimalizálás eredményei a holtidőkompenzáció után	55
6.3.3.	A paraméterek alapján történő szögsebességszabályozás	56
7.	Konklúzió	57
	Köszönetnyilvánítás	58
	Hivatkozások	59

Összefoglaló

Digitálisan szabályozott teljesítményelektronikai átalakítóról táplált villamos hajtások egyre nagyobb szerepet játszanak a korszerű műszaki berendezésekben, a szerszámgépektől kezdve az ipari robotokon át a megújuló energiaforrásokat hasznosító rendszerekig, mivel velük költséghatékonyan megvalósítható a villamos és mechanikai energia oda-vissza történő alakítása. Az állandómágneses szinkrongép (Permanent Magnet Synchronous Machine) az egyik leggyakrabban használt géptípus a nagy minőségi követelményeknek eleget tevő szabályozható villamos hajtásokban számos előnyös tulajdonsága miatt, mint a magas hatásfok érték és a nagy teljesítménysűrűség.

PMSM zárt hurkú szabályozásához fontos a motor villamos és mechanikai paramétereinek minél pontosabb ismerete. Az egyik leggyakrabban használt szabályozási stratégia váltakozóáramú gépek, és így PMSM hajtások esetében, a mezőorientált szabályozás. A sebesség- és áramhurkok szabályozó paramétereinek kiszámításához szükség van a gép paramétereinek értékére, ami az esetek többségében nincs megadva a katalógusban. Az elmúlt évtizedben modell prediktív szabályozási módszereket egyre szélesebb körben kezdtek el alkalmazni PMSM hajtásokban. Prediktív szabályozási technikák esetén egy matematikai modell segítségével előre megjósolható a hajtás jövőbeni viselkedése, ami alapján kiszámítható, hogy melyik a legjobb következő beavatkozó lépés. Ebben az esetben is elengedhetetlen a motor paramétereinek minél pontosabb ismerete a megfelelő működéshez.

Dolgozatom célja PMSM hajtások paramétereinek becsléssel történő meghatározása és egy hatékony módszer kidolgozása az identifikáció folyamatára. A javasolt módszer először a motor villamos paramétereit azonosítja: a motor ellenállását, induktivitását, illetve pólusfluxus nagyságát. Ezt követően történik meg, a kapott eredmények felhasználásával, a mechanikai jellemzők, mint a tehetetlenségi nyomaték és a viszkózus csillapítás meghatározása.

A becsléshez használt eljárások közül olyan offline módszereket hasonlítok össze, mint a rekurzív legkisebb négyzetek módszere (RLS) algoritmus, a Trust Region optimalizáció, illetve az ARMAX modell identifikációs módszer. A különböző lehetőségeket első lépésben MATLAB környezetben szimulációk segítségével vizsgálom meg. Ezt követően bemutatom a módszerek működését valós környezetben, amihez egy DSP-t használok, mind az identifikációhoz, mind a PMSM hajtás zárt hurkú szabályozásának megvalósításához.

Abstract

Electric drives powered by digitally regulated power electric converters play an increasingly important role in modern technical equipment, from machine tools to industrial robots to renewable energy-powered systems, since conversion between electrical and mechanical energy could prove to be more cost-efficient using them. The permanent magnet synchronous machine is one of the most popular machine types among controllable electric drives meeting large quality requirements due to its numerous beneficial attributes such as its high efficiency and large power density.

In order to implement the closed-loop control of a PMSM, it is substantial to know its electrical and mechanical parameters as accurately as possible. Field-oriented control is one of the most common control strategies for alternating current motors and, thus, for PMSM drives. For the calculation of control parameters of speed and current loops, the value of machine parameters is needed, which are not specified in the motor catalogue in most cases. In the last decade, the use of model predictive control techniques for PMSM drives has become more widespread. These methods predict the future behaviour of the drive using a mathematical model, and then the next input step is calculated based on this prediction. To make this method feasible, it is also essential to know the motor parameters precisely.

In this thesis, I identify the parameters of PMSM drives through estimation and propose an efficient method for the identification process. First, the suggested method determines the electrical attributes of the motors: motor resistance, inductance and rotor flux. Afterwards, the mechanical parameters such as motor inertia and viscous damping are identified using the previously obtained results.

Among the methods used for the estimations, I compare offline techniques, including recursive least squares (RLS) algorithm, Trust Region optimization and ARMAX model identification method. The different approaches are examined through simulations in MATLAB, after which I present the operation of the methods in a real setting. For this, I use a DSP both for the identification and for implementing the closed-loop control of the PMSM drive.

1. Bevezetés

Az állandómágneses szinkron motorokat, nagy teljesítménysűrűségüknek, valamint megbízható üzemeltetésüknek köszönhetően, napjainkban egyre szélesebb körben alkalmazzák a műszaki életben. Költséghatékony és biztonságos működésük miatt előszeretettel használják gyártóberendezések, ipari robotok hajtásrendszereiben, illetve a korszerű villamos járművek hajtásláncában is fontos szerepet játszik ez a géptípus.

A PMSM (Permanent Magnet Synchronous Machine) táplálása jellemzően háromfázisú váltakozó árammal történik. A szinkron működés miatt a motor mechanikai forgási sebessége egyenesen arányos a forgó mágneses tér sebességével, amely sebesség a táplálás háromfázisú feszültségének frekvenciájával állítható. Ennek megvalósításához az iparban változtatható frekvenciájú hajtást (VFD) alkalmaznak. Fontos szempont, a megfelelő táplálási amplitúdó és frekvencia mellett, a veszteségmentes energiatovábbítás, így teljesítményelektronikai átalakítók felhasználása vált elterjedtté erre a célra. Amennyiben a táplálás egyenáramú feszültségről történik, leggyakrabban kétszintű, háromfázisú feszültséginverter segítségével valósítják meg a PMSM számára szükséges váltakozó feszültséggé alakítást. Dolgozatomban az inverter bemenő feszültségét ideális feszültségforrásként kezelem, így ennek előállítás módjára nem térek ki.

A teljesítményelektronikai átalakítóról táplált szinkron motorok zárt hurkú szabályozásához többféle különböző stratégia terjedt el, mint a mezőorientált szabályozás (Field-Oriented Control), közvetlen nyomatékszabályozás (Direct Torque Control) vagy szinuszos kommutáció. Manapság egyre nagyobb szerepet játszanak a modell prediktív szabályozási technikák is. Bármelyik szabályozási módszert alkalmazása esetén elengedhetetlen a gép villamos és mechanikai paramétereinek minél pontosabb ismerete. Ezen paraméterek becslésére adok hatékony eljárást, aminek eredményeivel azután a PMSM motorok szabályozása megvalósítható.

A villamos gépek paraméteridentifikációjához számos elterjedt módszer létezik, ezek közül az egyik leggyakoribb a rekurzív legkisebb négyzetek (RLS) algoritmus[5], ami kis számításigénnyel rendelkezik, így akár egy mikrovezérlőn végrehajtható. Ezen kívül két optimalizációs eljárást vizsgálok PMSM motor együtthatóinak becslésére: a predikciós hibabecslés alapú ARMAX modell identifikációt[8], valamint a Trust Region optimalizációs algoritmust[19].

A következő fejezetekben először bemutatom az állandómágneses szinkron motorok működésének egyenleteit, majd a használt algoritmusok elméleti hátterét, valamint az adott identifikációs problémára való alkalmazhatóságukat. Ezután MATLAB Simulink környezetben megvalósított szimulációk útján vizsgálom a paraméterek azonosításának hatékonyságát az egyes technikák esetén, majd ez alapján javaslatot teszek egy optimális identifikációs folyamatra. Ezt követően a kidolgozott stratégia eredményességét valós körülmények között is vizsgálom, amihez Hurst gyártmányú DMA0204024B101 PMSM

motoron végzem el az identifikációt. A későbbi fejezetek tárgyalják a folyamat fizikai megvalósítását, ami Texas Instruments LAUNCHXL-F28379D Launchpad digitális jelfeldolgozó processzor segítségével vezérelhető. A motor beavatkozójelét a DSP az ugyancsak Texas Instruments által gyártott BOOSTXL-3PhGaNInv háromfázisú feszültséginverteren keresztül biztosítja a PMSM számára.

Végezetül a meghatározott villamos és mechanikai paraméterek ismeretében elvégzem a motor zárt hurkú szabályozását, amihez a PMSM motorok esetén egyik legelterjedtebb stratégiát, a mezőorientált szabályozást fogom alkalmazni. A motor jellemzőiből kiszámíthatók az ehhez szükséges sebesség- és áramhurkok szabályozóparaméterei, így a kaszkád szabályozás a paraméterek azonosítása után azonnal megvalósíthatóvá válik a DSP-n keresztül.

2. Elméleti háttér

2.1. Az állandómágneses szinkron motor gépmmodellje

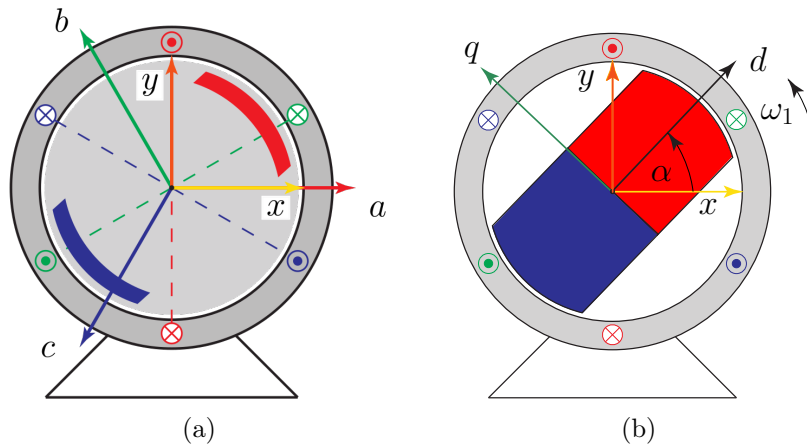
Az állandómágneses szinkron motorok forgórésze beágyazott mágneseket tartalmaz, és az ezen mágnesek által létrehozott állandó mágneses tér lép kölcsönhatásba az állórész háromfázisú tekercselése által gerjesztett forgó mágneses mezővel. A rotor pólusai együtt forognak a mágneses térrel, ezért a forgórész mechanikai szögsebessége, illetve szöghelyzete mindig közvetlenül arányos a mágneses tér forgási sebességével, illetve szögelfordulásával. A kapcsolatot a motor P póluspárszámával írhatjuk le:

$$\alpha = P \cdot \alpha_{\text{mech}} \quad (1)$$

$$\omega_1 = P \cdot \Omega \quad (2)$$

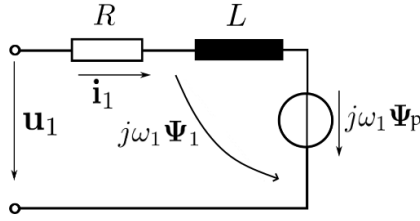
ahol α és α_{mech} a villamos és mechanikai szögelfordulás, ω_1 a villamos, Ω a mechanikai szögsebesség. A gép mozgását az álló és forgórészhez rögzített koordináta-rendszerben is leírhatjuk. Az álló- tengelyeket x - y betűkkel, a forgó koordináta-rendszer tengelyeit d - q -val jelöljük. Ezeket, valamint a három fázis a - b - c koordináta-rendszerét az 1. ábra szemlélteti.

Az 1a. ábra a három fázis és az álló koordináta-rendszer egymáshoz képesti elhelyezkedését mutatja be, míg a 1b. ábra az álló és a forgó koordináta-rendszereket, valamint a villamos szögelfordulást és szögsebességet.



1. ábra. A PMSM mozgását leíró koordináta-rendszerek

Ha a szinkron gép forgórésze szimmetrikus, modellezhetjük egy sorosan kapcsolt ellenállással, induktivitással és feszültséggenerátorral, utóbbival a forgórész által indukált feszültséget helyettesítjük.



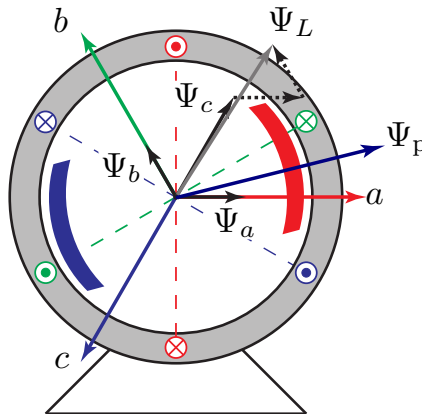
2. ábra. PMSM motor helyettesítő áramköri képe [13]

Az állórész villamos egyenletei álló koordináta-rendszerben az alábbi módon írhatók fel térvektoros formában

$$\mathbf{u}_1 = R_1 \mathbf{i}_1 + \frac{d\mathbf{\Psi}_1}{dt}, \quad (3)$$

ahol $\mathbf{u}_1$ és $\mathbf{i}_1$ az állórész feszültség- és áramvektora, R_1 az állórész-ellenállás, és $\mathbf{\Psi}_1$ állórész fluxusvektor. Ez utóbbi a 3. ábrán látható a háromfázisú tekercselésben folyó áram által létesített forgó $\mathbf{\Psi}_L = L\mathbf{i}_1$ fluxusvektor és az állandómágnes által gerjesztett, időben állandó, de a forgórészhez rögzített és így azzal együtt mechanikusan forgatott Ψ_p pólusfluxus vektoriális összege.

$$\mathbf{\Psi}_1 = L\mathbf{i}_1 + \mathbf{\Psi}_p = L\mathbf{i}_1 + \Psi_p e^{j\alpha} \quad (4)$$



3. ábra. A motor fluxusvektorai

Ezt az egyenletet behelyettesítve (3)-ba, az állórész feszültségre

$$\mathbf{u}_1 = R_1 \mathbf{i}_1 + L \frac{d\mathbf{i}_1}{dt} + \frac{d(\Psi_p e^{j\alpha})}{dt} \quad (5)$$

összefüggés adódik. Mivel Ψ_p az egyenletben konstans, az összetett függvény deriválási szabályát felhasználva

$$\frac{d(\Psi_p e^{j\alpha})}{dt} = \underbrace{\Psi_p e^{j\alpha} \cdot j}_{\omega_1} \frac{d\alpha}{dt}. \quad (6)$$

Transzformáljuk az 5. egyenletet forgó koordináta-rendszerbe, amihez mindkét oldalt szorozzuk meg $e^{-j\alpha}$ -val, majd vegyük figyelembe, hogy a forgó koordináta-rendszerben

$\mathbf{u}_{dq} = \mathbf{u}_1 e^{-j\alpha}$ és $\mathbf{i}_{dq} = \mathbf{i}_1 e^{-j\alpha}$:

$$\underbrace{\mathbf{u}_1 e^{-j\alpha}}_{\mathbf{u}_{dq}} = R_1 \underbrace{\mathbf{i}_1 e^{-j\alpha}}_{\mathbf{i}_{dq}} + L \underbrace{\frac{d\mathbf{i}_1}{dt}}_{f'(t)} \underbrace{e^{-j\alpha}}_{g(t)} + \Psi_p j\omega_1 \underbrace{e^{j\alpha} e^{-j\alpha}}_1 \quad (7)$$

Felhasználva, hogy

$$f'(t)g(t) = (f(t) \cdot g(t))' - f(t)g'(t) \quad (8)$$

a forgórészkeri feszültség felírható

$$\mathbf{u}_{dq} = R_1 \mathbf{i}_{dq} + \frac{d}{dt} L \underbrace{\mathbf{i}_1 e^{-j\alpha}}_{\mathbf{i}_{dq}} - L \mathbf{i}_1 \underbrace{\frac{d}{dt} e^{-j\alpha}}_{-j \frac{d\alpha}{dt} e^{-j\alpha}} + \Psi_p j\omega_1 \quad (9)$$

$$\mathbf{u}_{dq} = R_1 \mathbf{i}_{dq} + L \frac{d\mathbf{i}_{dq}}{dt} - j\omega_1 (L \mathbf{i}_{dq} + \Psi_p) \quad (10)$$

formában. Az egyenletet reális (d) és imaginárius (q) komponensekre bontva:

$$u_d = R_1 i_d + L \frac{di_d}{dt} - \omega_1 L i_q \quad (11)$$

$$u_q = R_1 i_q + L \frac{di_q}{dt} + \omega_1 (L i_d + \Psi_p) \quad (12)$$

Látható, hogy a modell nemlineáris, hiszen a megfelelő áramkomponens és a villamos szögsebesség szorzata is szerepel az egyenletekben.

A későbbiekben a rendszer kimeneteként az áramot figyelem, ezért érdemes az i_d és i_q áramkomponensek megváltozására felírni az egyenleteket.

$$\dot{i}_d = -\frac{R}{L} i_d + \omega_1 i_q + \frac{1}{L} u_d \quad (13)$$

$$\dot{i}_q = -\frac{R}{L} i_q - \omega_1 i_d + \frac{1}{L} u_q - \frac{\Psi_p}{L} \omega_1 \quad (14)$$

A gép elektromágneses nyomatéka az i_q áramkomponenssel szinkronban van.

$$M = \frac{3}{2} P \Psi_p i_q = K_t i_q \quad (15)$$

A motort a terhelőnyomaték és a viszkozus csillapítás lassítja, így a dinamika alaptételét felírva a következő mechanikai egyenlet adódik.

$$J \frac{d\Omega}{dt} = M - M_{\text{load}} - b \cdot \Omega \quad (16)$$

ahol J a motor hajtó tengelyére redukált tehetetlenségi nyomatéka. Az villamos szögsebesség és a mechanikai forgási sebesség közti kapcsolatot ismerve (2), valamint felhasználva (15)-t

$$\frac{J}{P} \frac{d\omega_1}{dt} = K_t i_q - M_{\text{load}} - b \cdot \frac{\omega_1}{P} \quad (17)$$

egyenlet írható fel.

2.2. Diszkrét modell létrehozása az identifikációhoz

Az előző részben felírt gépegyenletekből feltételezhetjük, hogy ismerjük a motor póluspárjainak számát, valamint a terhelést, amivel üzemeltetjük a gépet. Viszont a további jellemzői, mint az armatúra ellenállása, induktivitása, a rotor pólusfluxusa, valamint a mechanikai tulajdonságok, vagyis a motor tehetetlenségi nyomatéka és viszkózus csillapítása lehetnek bizonytalanok. Ezeket az adatokat, bár elengedhetetlenek a gép szabályozásához, nem feltétlenül tartalmazza a motorkatalógus.

Ezeket a paramétereket becsülhetjük mérés segítségével: a motorra meghatározott feszültségfüggvényeket adva a motor három fázisának áramerősségei, valamint feszültségei a használt inverterkártya segítségével mérhetők, illetve enkóderrel megvalósítható a mechanikai szögelfordulás és ezáltal, a szögsebesség mérése.

Mivel a motor vezérlése egy digitális jelfeldolgozó processzoron keresztül történik, a beavatkozások és a mérések is időben diszkrétan zajlanak le. A rendelkezésre álló identifikációs eljárások is mintavételezett modelleken dolgoznak, így szükséges a korábban meghatározott gépegyenleteket diszkrétizálni. Ehhez az előrelépő Euler-módszert alkalmazva az áramkomponensek deriváltjai

$$\frac{di_q}{dt} \approx \frac{i_q[k+1] - i_q[k]}{T_s} \text{ és } \frac{di_d}{dt} \approx \frac{i_d[k+1] - i_d[k]}{T_s} \quad (18)$$

alakban írhatók fel.

Induljunk ki a 13. és 14., áramkomponenseket tartalmazó, egyenletekből. Ha a mintavételezési idő megfelelően kicsi, a vizsgált időtartam alatt a szögsebességet állandónak tekinthetjük, mivel a rendszer mechanikai tranziensei jóval lassabban zajlódnak le az áramváltozásoknál, így adott k pillanatbeli értéke $\omega_1[k]$. A beavatkozójelet nulladrendű tartószerv alkalmazásával biztosítjuk a rendszernek, vagyis T_s ideig állandó, így $u[k]$ feszültség k időpillanatbeli értékével írható fel diszkrét időben. A fenti egyenleteket diszkrét formára átalakítva

$$\frac{i_d[k+1] - i_d[k]}{T_s} = -\frac{R}{L}i_d[k] + \omega_1[k]i_q[k] + \frac{1}{L}u_d[k] \quad (19)$$

$$\frac{i_q[k+1] - i_q[k]}{T_s} = -\frac{R}{L}i_q[k] - \omega_1[k]i_d[k] + \frac{1}{L}u_q[k] - \frac{\Psi_p}{L}\omega_1[k] \quad (20)$$

majd kifejezve belőle a $(k+1)$. időpillanatbeli áramértékeket

$$i_d[k+1] = \left(1 - \frac{R}{L}T_s\right)i_d[k] + T_s\omega_1[k]i_q[k] + \frac{T_s}{L}u_d[k] \quad (21)$$

$$i_q[k+1] = \left(1 - \frac{R}{L}T_s\right)i_q[k] - T_s\omega_1[k]i_d[k] + \frac{T_s}{L}u_q[k] - \frac{T_s\Psi_p}{L}\omega_1[k] \quad (22)$$

diszkrét egyenleteket kapjuk.

Hasonlóan diszkretizálható a mechanikai egyenlet is, ha

$$\frac{d\omega_1}{dt} \approx \frac{\omega_1[k+1] - \omega_1[k]}{T_s} \quad (23)$$

formában írjuk fel a villamos szögsebesség deriváltját. Így a diszkrét idejű egyenlet:

$$\frac{J}{P} \frac{\omega_1[k+1] - \omega_1[k]}{T_s} = K_t i_q[k] - M_{\text{load}}[k] - b \cdot \frac{\omega_1[k]}{P} \quad (24)$$

$$\omega_1[k+1] = \frac{PK_t T_s}{J} i_q[k] - \frac{PT_s}{J} M_{\text{load}}[k] + \left(1 - \frac{bT_s}{J}\right) \omega_1[k] \quad (25)$$

A mintavételezett rendszer egyenleteinek segítségével már többféle identifikációs módszerrel becsülhetők a mért jellemzőkhöz tartozó együtthatók, majd ezekből a mintavételezési idő és póluspárszám ismeretében azonosíthatók a keresett villamos és mechanikai tulajdonságok.

2.3. Transzformációk a koordinátarendszerek között

Az identifikációhoz az előző fejezetben előállított diszkrét egyenletekből fogok kiindulni, viszont a motor beavatkozójele az 1a. ábrán látható a - b - c koordinátarendszerbeli feszültség lesz, ezért szükséges megállapítani a kapcsolatot a d - q koordinátarendszerben felírt feszültségkomponensek és a tényleges háromfázisú beavatkozójel között. Ehhez először a forgó koordinátarendszerből transzformáljuk a feszültségeket az állórészhez rögzített x - y koordinátarendszerbe az 1. ábra alapján:

$$u_x = \cos \alpha \cdot u_d - \sin \alpha \cdot u_q \quad (26)$$

$$u_y = \cos \alpha \cdot u_q + \sin \alpha \cdot u_d \quad (27)$$

ahol α a villamos szögelfordulás. Ez további koordinátatranszformációval leírható a háromfázisú koordinátarendszerben:

$$u_a = u_x \quad (28)$$

$$u_b = \frac{\sqrt{3}}{2} u_y - \frac{1}{2} u_x \quad (29)$$

$$u_c = -\frac{\sqrt{3}}{2} u_y - \frac{1}{2} u_x \quad (30)$$

Ezeknek az összefüggéseknek a segítségével a d - q feszültségkomponensekből felírható a motor három fázisára adandó feszültség. Mivel az inverter csak U_{DC} vagy $-U_{DC}$ feszültséget tud kapcsolni az egyes fázisokra, ezért a beavatkozás impulzusszélesség-modulációval oldható meg, amit a következő alfejezet részletez.

Az identifikációhoz szükség lesz az i_d és i_q áramkomponensek mérésére is, viszont ezek közvetlenül nem mérhetők. Az inverterkártya az áramot az a - b - c fázisokon képes mérni,

ezért a kiértékeléshez ezeket is transzformálni kell a forgórészhez rögzített koordináta-rendszerbe. Először transzformáljuk álló koordináta-rendszerbe az áramokat:

$$i_x = i_a \quad (31)$$

$$i_y = \frac{1}{\sqrt{3}}i_a + \frac{2}{\sqrt{3}}i_b \quad (32)$$

ami után elvégezhető a $d - q$ koordinátákra transzformálás is:

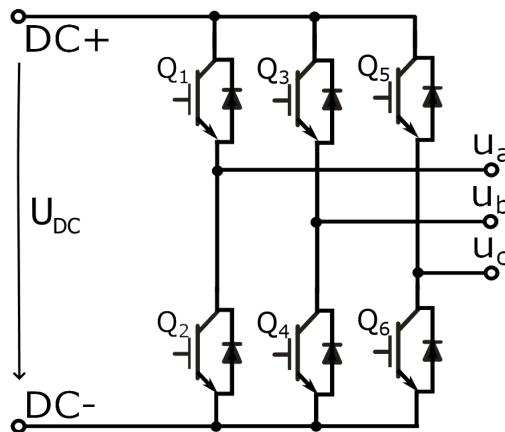
$$i_q = \cos \alpha \cdot i_y - \sin \alpha \cdot i_x \quad (33)$$

$$i_d = \cos \alpha \cdot i_x + \sin \alpha \cdot i_y \quad (34)$$

Látható, hogy a $d-q$ áramkomponensek átszámításához nem volt szükség mind a három fázis mérésére, ezért megvalósításkor sem fogom az i_c áramkomponenst mérni.

2.4. Vivőfrekvenciás impulzusszélesség-moduláció

A motort háromfázisú, kétszintű feszültséginverteren keresztül tápláljuk, ami három félhídból áll. Ezek feladata, hogy a terhelés három fázisát a DC csatlakozó pozitív vagy negatív sínjére kapcsolják a 4. ábrán látható módon. A kapcsolóelemek jellemzően nagy teljesítményszintű alkalmazásokban IGBT-k, közepes és alacsony teljesítményszintűen főleg Si-MOSFET-ek. Manapság egyre gyakrabban használnak szilícium alapú félvezetők helyett széles tiltott sávú anyagokat a teljesítményelektronikai átalakítókban, például GaN és SiC alapú MOSFET-eket. A hardveres megvalósításban Texas Instruments gyártmányú, GaN alapú invertert fogok használni.



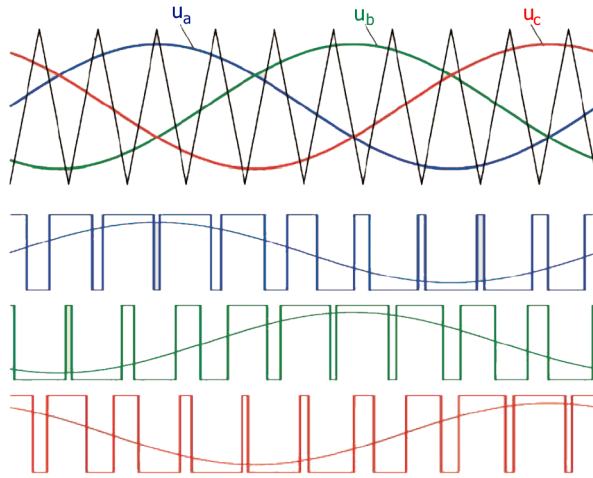
4. ábra. Háromfázisú feszültséginverter kapcsolási rajza

A felső-, illetve alsó oldali tranzisztorok mindig ellentétes vezérlésűek, vagyis a kapcsolójelek egymás negáltjai:

$$Q_1 = \bar{Q}_2, Q_3 = \bar{Q}_4 \text{ és } Q_5 = \bar{Q}_6 \quad (35)$$

A kapcsoláshoz szükséges jeleket a valós mérés során a TI LaunchPad PWM modulja fogja biztosítani, amihez az előző alfejezetben kiszámított feszültségeket modulálni kell. Ezt vivőfrekvenciás impulzusszélesség-modulációval hajtom végre, ami a DSP ePWM perifériájával könnyen implementálható, valamint nem nagy számításigénnyel rendelkező eljárás.

A vivőfrekvenciás impulzusszélesség-moduláció esetében a kapcsolójel egy referenciajel és egy háromszögjel komparálásából adódik, ahol a referenciát a transzformációk segítségével az egyes fázisokra kiszámított feszültségkomponens képzi. Az 5. ábra szemlélteti, hogy háromfázisú szinuszos feszültségjel esetén hogyan zajlana a moduláció.

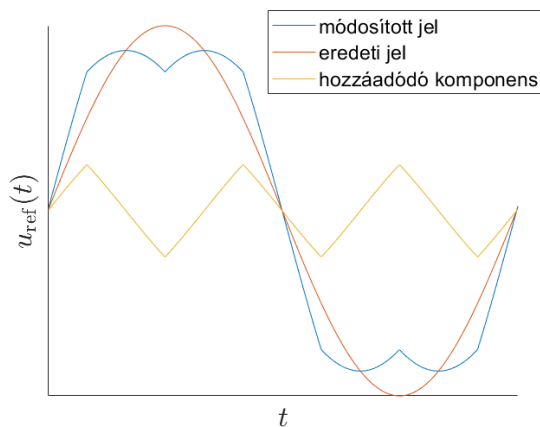


5. ábra. Háromfázisú szinuszos jel vivőfrekvenciás impulzusszélesség-modulációja [15]

Hátránya a módszernek, hogy lineáris modulációval a legnagyobb előállítható kimenő fázisfeszültség amplitúdója csak $\frac{U_{DC}}{2}$ lehet, így ha ennél nagyobb kimenő feszültséget szeretnénk előállítani, felléphet a túlmoduláció jelensége. Ekkor megjelennek alacsony sorrendű felharmonikusok a jelben és nagyobb lesz a motor harmonikus vesztesége. A lineáris modulációs tartomány kiterjesztéséhez az ún. min-max módszert alkalmazom, ami a referenciajeleket úgy módosítja, hogy mindegyik jelhez hozzáad egy bizonyos u_h feszültséget, ami az aktuális legkisebb és legnagyobb feszültség átlagából adódik:

$$u_h = -[(1 - 2\mu) + \mu u_M + (1 - \mu)u_m] = -\frac{u_M + u_m}{2} \quad (36)$$

ahol u_M a legnagyobb, u_m a legkisebb az adott időpillanatban a három fázis feszültsége közül, valamint μ értéke tipikusan 0,5. Így a módosítást végrehajtva egy szinuszos jelből a 6. ábrán látható alakot kapjuk, csökkentve a harmonikus veszteséget.



6. ábra. Feszültségkomponensek módosítása min-max módszerrel

Ez a módszer a térvektor moduláció vivőfrekvenciás megvalósításának tekinthető, viszont számításigénye jóval kisebb, mint a térvektoros megközelítésnek.

3. A paraméteridentifikációhoz használt módszerek

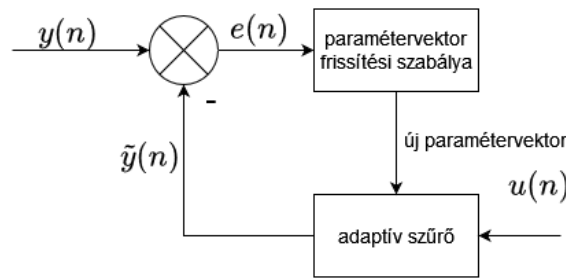
3.1. Rekurzív legkisebb négyzetek algoritmus

A rekurzív legkisebb négyzetek (RLS) algoritmus egy adaptív szűrő, ami akkor alkalmazható, ha az u bemenet és y kimenet közötti kapcsolat leírható a következő alakban:

$$y(n) = \Theta_n^\top u(n), \quad (37)$$

ahol a meghatározandó együtthatókat egy Θ_n paramétervektorban tároljuk. Az algoritmus megadott bemenő- és kimenőjel esetén keresi ezeket az együtthatókat, miközben minden lépésben új adatokat kap, ahogy a mért adatok elérhetővé válnak. Megvizsgálja, hogy az adott paraméterkombináció mellett milyen $\tilde{y}(n)$ kimenetet vár a bemenőjel függvényében, majd a várt és a tényleges kimenőjeltől hibafüggvényt képez.

$$e(n) = y(n) - \tilde{y}(n) \quad (38)$$



7. ábra. Negatívan visszacsatolt RLS algoritmus vázlata

[4] és [5] irodalmak tárgyalják a hibajel alapján történő frissítés szabályát, ezek alapján a módszer először a hibajelből képez lineáris, súlyozott költségfüggvényt úgy, hogy az előző n darab hibajel négyzetét veszi figyelembe exponenciálisan kisebb súllyal.

$$C(\Theta_n) = \sum_{i=0}^n \lambda^{n-1} e^2(i) \quad (39)$$

Az, hogy a szűrő mennyire érzékeny a régebbi mintákra a $0 < \lambda \leq 1$ *forgetting factor* segítségével állítható be. Az így kapott költségfüggvénynek keressük a minimumát, amihez a keresett együtthatók szerinti parciális deriváltak segítségével juthatunk:

$$\frac{\partial C(\Theta_n)}{\partial \Theta_n(k)} = \sum_{i=0}^n 2\lambda^{n-1} e(i) \cdot \frac{\partial e(i)}{\partial \Theta_n(k)} = - \sum_{i=0}^n 2\lambda^{n-1} e(i) u(i-k) = 0, \quad (40)$$

ahol $k = 0, 1, \dots$ egész szám. Mivel a hibajelet mint a valós és az elvárt kimenet különbsége definiáltuk, a hibajel

$$e(i) = y(i) - \tilde{y}(i) = y(i) - \sum_{l=0}^m \Theta_n(l) u(i-l) \quad (41)$$

alakban írható fel, ahol az elvárt kimenet az m darab múltbeli mintától függ. Behelyettesítve (41)-t és kiemelve (40)-ből a $\sum_{i=0}^n 2\lambda^{n-1}u(i-k)$ tagot az egyenlet a következőképp rendezhető:

$$\sum_{i=0}^n 2\lambda^{n-1} \left[y(i) - \sum_{l=0}^m \Theta_n(l)u(i-l) \right] u(i-k) = 0 \quad (42)$$

Átrendezve az egyenletet úgy, hogy az egyik oldalra kerüljenek a valós kimenő jeltől függő tagok:

$$\left[\sum_{i=0}^n 2\lambda^{n-1}u(i-l)u(i-k) \right] \sum_{l=0}^m \Theta_n(l) = \sum_{i=0}^n 2\lambda^{n-1}y(i)u(i-k) \quad (43)$$

Ezeket a szorzatokat felírhatjuk mátrixos formában, ha a $\sum_{i=0}^n 2\lambda^{n-1}u(i-l)u(i-k)$ tagot elnevezzük $\mathbf{P}$ kovariancia-mátrixnak és $\sum_{i=0}^n 2\lambda^{n-1}y(i)u(i-k)$ tagot az $\tilde{y}(n)$ és $u(n)$ közötti $\mathbf{p}$ keresztkovariancia-mátrixnak. Így a

$$\mathbf{P}(n)\boldsymbol{\Theta}_n = \mathbf{p}(n) \quad (44)$$

alakot kapjuk, amiből kifejezhető az a paramétervektor, ami az adott lépésbeli költségfüggvényt minimalizálja:

$$\boldsymbol{\Theta}_n = \mathbf{P}^{-1}(n)\mathbf{p}(n) \quad (45)$$

Ez lesz az algoritmus frissítési szabálya.

A fent látható alak esetén az új $\boldsymbol{\Theta}_n$ paramétervektor kiszámításakor minden lépésben kvadratikus mátrix invertálására van szükség, ezért a gyakorlatban az algoritmusnak másik alakjait használják, a kisebb számításigény érdekében. Dolgozatomban én a módszer [18] szerzői által alkalmazott alakját használom:

$$e(n) = y(n) - \mathbf{u}^\top(n)\boldsymbol{\Theta}(n-1) \quad (46)$$

$$\boldsymbol{\gamma} = \frac{\mathbf{P}(n-1)\mathbf{u}(n)}{\lambda + \mathbf{u}^\top(n)\mathbf{P}(n-1)\mathbf{u}(n)} \quad (47)$$

$$\mathbf{P}(n) = \frac{1}{\lambda}[\mathbf{P}(n-1) - \boldsymbol{\gamma}\mathbf{u}^\top(n)\mathbf{P}(n-1)] \quad (48)$$

$$\boldsymbol{\Theta}(n) = \boldsymbol{\Theta}(n-1) + \boldsymbol{\gamma}e(n) \quad (49)$$

ahol a $\boldsymbol{\gamma}$ erősítésvektor adja meg a hibajel és a frissített paramétervektor közti kapcsolatot.

Rekurzívan végrehajtva a fenti algoritmust, a paramétervektor értékei a keresett együtthatókhoz konvergálnak, így megkapható az összefüggés a kimenő- és bemenőjel között.

A módszer előnye, hogy nincs szükség nagy mennyiségű adat tárolására, így a DSP segítségével is végrehajtható az identifikáció, nem szükséges a nagyobb számítási kapacitású külső számítógéppel kommunikálni. Valamint, ha az együtthatók konvergálása során elérték az elvárt pontosságot, befejezhető az eljárás, ellentétben a következőkben bemutatott két eljárással, ahol a lépésszám előre meghatározott kell legyen.

3.1.1. PMSM motor RLS identifikációja

Ahhoz, hogy a módszert alkalmazhassuk a PMSM hajtás paramétereinek azonosításához, elengedhetetlen egy olyan modell, amire érvényes, hogy a kimenőjel felírható bemenetből és a keresett együtthatókból az alábbi módon:

$$y(n) = \Theta_{\mathbf{n}}^{\top} u(n), \quad (50)$$

Ehhez a 2.2. részben meghatározott diszkrét egyenleteket tudjuk felhasználni.

A villamos együtthatók meghatározásához [18]-ból indultam ki. Az együtthatók a 2.2. diszkrét formában felírt egyenlet alapján azonosíthatók. Mivel a bemenő feszültség és az áramkomponensek közötti kapcsolat nemlineáris, a kapott forma nem használható közvetlenül az RLS eljáráshoz, továbbá ez az algoritmus nem képes több bemenet hatásának együtthatóit azonosítani. Ezért az $\omega_1[k]$ villamos szögsebességtől függő tagokat el kell hanyagolnunk, így a diszkrét villamos egyenlet egyszerűbb formája:

$$i_q[k+1] = \left(1 - \frac{R}{L}T_s\right) i_q[k] + \frac{T_s}{L} u_q[k] \quad (51)$$

Ez az egyenlet azonban csak akkor írja le precízen a működést, ha a gép le van fogva, vagyis $J = \infty$. Ekkor a motor szögsebessége zérus, így valóban érvényes az egyszerűsített egyenlet a PMSM hajtásra, lefogás nélküli esetben közelítésnek alkalmazható.

Az RLS módszer alkalmazásához szükségünk lesz a meghatározni kívánt paraméterek vektorára. Az algoritmus során az i_q áramkomponens a válaszjel, az u_q feszültségkomponens a bemenőjel, így a keresett együtthatók:

$$i_q[k+1] = \underbrace{\left(1 - \frac{R}{L}T_s\right)}_{a_1} i_q[k] + \underbrace{\frac{T_s}{L}}_{b_1} u_q[k] \quad (52)$$

$$a_1 = 1 - \frac{R}{L}T_s \text{ és } b_1 = \frac{T_s}{L} \quad (53)$$

Tehát az algoritmushoz használt paramétervektor:

$$\Theta_{\mathbf{n}} = \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \quad (54)$$

Ekkor $n = 2$, így a bemenő- és a kimenőjel utolsó mért értékére lesz szükség:

$$\mathbf{u}(n) = \begin{bmatrix} y(n-1) \\ u(n-1) \end{bmatrix} \quad (55)$$

Ekkor a kovariancia-mátrix is 2×2 -es, kiinduláskor értéke legyen: $\mathbf{P}(0) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, míg az erősítésvektor $\gamma(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

Ha a paraméterek konvergálnak, a kapott együtthatókból az armatúraellenállás és induktivitás visszaszámítható:

$$R = \frac{1 - a_1}{b_1} \text{ és } L = \frac{a_1}{b_1} T_s \quad (56)$$

A mechanikai paraméterek meghatározásához a 25. egyenletből indulunk ki, úgy, hogy a bemenőjelnek ezúttal az i_q áramkomponenst, kimenetnek a villamos szögsebességet választjuk. Továbbra sem tudjuk a módszert több bemenet esetén alkalmazni, ezért kikötjük, hogy $M_{\text{load}} = 0$, vagyis terhelés nélkül üzemeltetjük a rendszert. Így a paramétervektor együtthatói a mechanikai egyenletből:

$$\omega_1[k+1] = \underbrace{\left(1 - \frac{bT_s}{J}\right)}_{d_1} \omega_1[k] + \underbrace{\frac{PK_t T_s}{J}}_{e_1} i_q[k] \quad (57)$$

$$\Theta_n = \begin{bmatrix} d_1 \\ e_1 \end{bmatrix} \quad (58)$$

Továbbra is érvényes, hogy egytárolós rendszerként modellezzük a motort, tehát az adott időpillanatbeli értéken kívül egy múltbeli értéket használunk az algoritmushoz, így a mátrixok rangja, illetve kezdeti értékei változatlanok a villamos identifikációhoz képest.

3.2. ARMAX modell identifikációs módszer

Az ARMAX modell a bemeneti-kimeneti polinomiális modellek családjába tartozik. Ezek a kimenet és bemenet közti kapcsolatot olyan polinomok segítségével írják le, amelyek változója a z^{-1} eltolási operátor. Ez [1] alapján a következő alakban fejezhető ki:

$$A(z^{-1})y(t) = \sum_{i=1}^{n_u} \left[\frac{B_i(z^{-1})}{F_i(z^{-1})} u_i(t - nk_i) \right] + \frac{C(z^{-1})}{D(z^{-1})} n(t) \quad (59)$$

ahol $u(t)$ a bemeneti, $y(t)$ a kimeneti jel, $n(t)$ a zaj; A , B , C , D és F z^{-1} alapú polinomok. A bemenetek száma n_u , nk az egyes bemeneti jelekhez tartozó időkéscés.

Formájából is látható, hogy az RLS algoritmussal ellentétben ez a modell képes több bemenetet is kezelni, sőt, egyszerre több, egymástól is függő kimenet is leírható a modellel a következő egyenletben látható módon.

$$\sum_{j=1}^{n_y} [A_{lj}(z^{-1})y_j(t)] = \sum_{i=1}^{n_u} \left[\frac{B_{li}(z^{-1})}{F_{li}(z^{-1})} u_i(t - nk_i) \right] + \frac{C_l(z^{-1})}{D_l(z^{-1})} n_l(t) \quad (60)$$

A gyakorlatban nem szokás mind az A , B , C , D és F polinomokat felhasználni a modellalkotáshoz, a rendszer dinamikus viselkedésétől és a zajtól függően egyszerűsíthető az egyenlet alakja. Az egyszerűsítések különböző modelltípusokhoz vezetnek:

- ARX modell

$$A(z^{-1})y(t) = \sum_{i=1}^{n_u} \left[\frac{B_i(z^{-1})}{F_i(z^{-1})} u_i(t - nk_i) \right] + n(t) \quad (61)$$

Az ARX modell esetében a zaj nem modellezhető függetlenül, így ezt akkor érdemes használni egy rendszer leírásához, hogyha az adott rendszer jó jel-zaj viszonytal rendelkezik.

- ARIX modell

$$A(z^{-1})y(t) = \sum_{i=1}^{n_u} [B_i(z^{-1})u_i(t - nk_i)] + \frac{1}{1 - z^{-1}}n(t) \quad (62)$$

Az ARIX modell a zajforráshoz egy integrátort kapcsol, így abban az esetben használható jól, ha nemstacionáris zajjal terhelt a rendszer.

- ARMAX modell

$$A(z^{-1})y(t) = \sum_{i=1}^{n_u} [B_i(z^{-1})u_i(t - nk_i)] + C(z^{-1})n(t) \quad (63)$$

Ez a modell a zaj mozgóátlagát képi a C paraméterben, ezáltal az ARX modellhez képest nagyobb rugalmasságot ad a zaj modellezéséhez.

- ARIMAX modell

$$A(z^{-1})y(t) = \sum_{i=1}^{n_u} [B_i(z^{-1})u_i(t - nk_i)] + \frac{C}{1 - z^{-1}}(z^{-1})n(t) \quad (64)$$

Az ARIMAX modell az ARMAX kiterjesztése nemstacionárius zaj esetére.

Mivel a PMSM hajtás mérésből származó zaja jelentős, viszont stacionáriusnak tekinthető, dolgozatomban az ARMAX modellel foglalkozom.

A modellek paramétereit a MATLAB *polyest* függvényének segítségével optimalizálom, ami egy iteratív végrehajtott predikciós hibabecslési eljárást hajt végre. A [8] és [12] források szerzői a modellt, mint a következő pillanatbeli kimenőjel prediktorfüggvénye írják le az eljárás elvégzéséhez:

$$\tilde{y}(t|t+1) = f(\varphi_{t-1}) \quad (65)$$

ahol $\tilde{y}(t|t+1)$ a következő pillanatbeli kimenetet adja meg a jelenlegihez képest. Ehhez az összes múltban megfigyelt adatot felhasználja, amit φ_{t-1} tartalmaz:

$$\varphi_{t-1}^T = \{y_{t-1}, \dots, y_{t-n_a}, u_{t-1}, \dots, u_{t-n_b}\}^T \in \mathbb{R}^{n_a+n_b}, \forall t \quad (66)$$

A prediktor paraméterezhető Θ paramétervektorral:

$$\tilde{y}(t|\Theta) = f_{t|t+1}(\Theta) = f(\varphi_{t-1}, \Theta) \quad (67)$$

Ekkor a predikciós hiba

$$e(t, \Theta) = y(t) - f_{t|t+1}(\Theta) \quad (68)$$

alakban írható fel.

Az ARMAX modell általános formája:

$$y(t) = G(z^{-1}, \Theta)u(t) + H(z^{-1}, \Theta)n(t) \quad (69)$$

Ez alapján az optimális prediktor formája:

$$\hat{f}_{t|t+1}(\Theta) = L_1(z^{-1}, \Theta)u(t) + L_2(z^{-1}, \Theta)n(t) \quad (70)$$

ahol L_1 és L_2 a modell G és H polinomjainak megfelelő leképezései. A módszer a Θ paramétervektort a predikció pontosságát optimalizálva határozza meg

$$\underset{\Theta}{\operatorname{argmin}} \frac{1}{n} \sum_{t=1}^n e^2(t, \Theta) \quad (71)$$

összefüggés használatával.

A MATLAB-nak ehhez szükséges megadni a modell polinomiális együtthatóinak, A , B és C -nek fokszámait: n_a -t, n_b -t és n_c -t, valamint a bemenet kimenethez képesti időkésését, $nk - t$.

3.2.1. A PMSM identifikációja az ARMAX modell predikciós hibájának becslésével

A villamos paraméterek becslésénél több lehetőség áll előttünk:

- Felhasználva az RLS algoritmushoz előállított egyszerűsített modellt (52) kereshetjük az a_1 és b_1 (53) együtthatók értékeit. Ebben az esetben az identifikálandó modellt a z tartományban felírva, ha az $i_q[k+1]$ -et vesszük az aktuális pillanatbeli kimenőjelnek,

$$\underbrace{i_q[k+1]}_{i_q} = a_1 \underbrace{i_q[k]}_{z^{-1}i_q} + b_1 \underbrace{u_q[k]}_{z^{-1}u_q} \quad (72)$$

$$(1 - a_1 z^{-1})i_q = b_1 z^{-1}u_q \quad (73)$$

alakot használhatjuk. Így az A és B polinom fokszáma is 1, hiszen csak z^{-1} -es tagok szerepelnek. Így $n_a = 1$ és $n_b = 1$ paraméterekkel futtatandó a görbeillesztés. Mivel az előreható Euler-módszert használtuk diszkretizációra, a bemenet hatása késleltetve jelenik meg a kimeneten, így $nk = 1$. A zaj hatását szintén elsőfokú polinommal modellezem, $n_c = 1$ választással.

- A második lehetőség az eredeti 22. egyenletből kiindulva több bemenettel rendelkező modell identifikációja. Ebben az esetben a $\omega_1 i_d$ szorzatot nem tudja kezelni a modell, hiszen a bemutatott polinomiális modellek szigorúan lineáris rendszerek leírásával foglalkoznak, így ezt a tagot el kell hanyagolnunk.

Így a z-tartománybeli egyenlet:

$$(1 - a_1 z^{-1})i_q = b_1 z^{-1}u_q + c_1 z^{-1}\omega_1 \quad (74)$$

ahol $c_1 = -\frac{T_s \Psi_p}{L}$.

Így az előző esethez képest n_a foksám nem változik, viszont a két bemenet miatt $n_{b_1} = 1$ és $n_{b_2} = 1$ paraméterekre lesz szükség. Valamint a ω_1 bemenet is egy időpillanattal késleltetve jelenik meg a kimeneten, így $nk_1 = 1$ és $nk_2 = 1$ érvényes.

A mechanikai paraméterek azonosításakor a 57. egyenletet használom fel, így a z-tartománybeli egyenlet:

$$(1 - d_1 z^{-1})\omega_1 = e_1 z^{-1} i_q \quad (75)$$

Hasonlóan az előzőekhez egytárolós rendszert használunk modellnek, így a foksámok megegyeznek a 73. egyenletnél látott foksámokkal.

3.3. A *Trust Region* optimalizáció

A *Trust Region* algoritmus egy numerikus, feltétel nélküli optimalizációs eljárás, ami azon alapul, hogy minden lépésben definiál egy megbízhatósági tartományt (*trust region*-t) a jelenlegi legjobb megoldás körül. Ebben a régióban egy, általában kvadratikus, modell pontosnak tekinthető, így ebben keresi a módszer a következő közelítő megoldást egy célfüggvény minimumaként. Ha az adott közelítés jól írja le a célfüggvényt, akkor a régiót növeli, különben csökkenti.

A módszer működésének részletes leírásával [11], [19] és [6] foglalkozik, ezek alapján az algoritmus kiindulása egy tetszőleges $\mathbf{x}_k \in \mathbb{R}$ vektor és egy $\Delta_k > 0$ paraméter, ami a megbízhatósági tartomány sugarát adja meg, így a Ω_k megbízhatósági tartomány az alábbi módon írható le:

$$\Omega_k = \{x : \|x - x_k\| \leq \Delta_k\} \quad (76)$$

Az $\mathbf{x}_k$ pont körüli Δ_k sugarú tartományban keresi az algoritmus a következő $\mathbf{x}_{k+1} = x_k + s_k$ közelítést, amihez optimalizációs feladatot hajt végre: a $q_k(s)$ kvadratikus modell minimumát keresi a megadott tartományon belül:

$$\min\{q(s)\} = \min \left\{ \frac{1}{2} s^\top H s + s^\top g \right\} \quad (77)$$

ahol g az f függvény gradiense x pontban, H a Hesse-mátrix. Ha H pozitív definit, akkor az optimalizációs feladat s_k megoldása gyorsan megtalálható. Az s_k érték szemléletesen a régió kiterjesztésének irányát adja meg.

Ezután értékeli, hogy az adott iterációs lépés mennyire tekinthető sikeresnek. Ehhez számítja a következő paramétert, ami megmutatja, az identifikálandó függvény tényleges csökkenése mekkora a jelzett csökkenéshez képest:

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{f(x_k) - q_k(s_k)} \quad (78)$$

Ha $\rho_k < \eta_2$, ahol η_2 tipikus értéke 0,25, akkor sikertelennek tekintjük a lépést és a tartományt szűkítjük:

$$\Delta_{k+1} = 0,25 \cdot \Delta_k \quad (79)$$

és $x_{k+1} = x_k$.

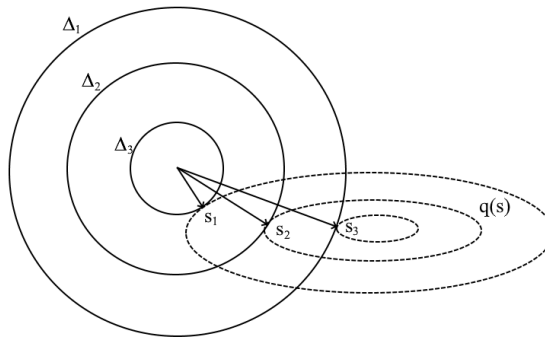
Ha $\rho_k > \eta_2$, már jó közelítésnek tekinthető a modell, így $x_{k+1} = x_k + s_k$ -re frissül az aktuális legjobb megoldás.

Ha $\eta_2 < \rho_k < \eta_3$ tartományba esik, ahol általában η_3 -at 0,75-re választják, akkor a megbízhatósági tartomány változatlan marad:

$$\Delta_{k+1} = \Delta_k \quad (80)$$

Ha $\rho_k > \eta_3$, akkor növelhető a tartomány:

$$\Delta_{k+1} = 2\Delta_k \quad (81)$$



8. ábra. *Trust Region* algoritmus szemléltetése

A módszer előnye, hogy az identifikálandó függvény ebben az esetben rendelkezhet több bemenettel, illetve nemlineáris alak esetén is képes kielégítő megoldást találni, így ez az egyik leggyakrabban használt optimalizációs algoritmus nemlineáris programozási problémák esetén.

3.3.1. A PMSM motor identifikációja *Trust Region* algoritmussal

Az algoritmus implementálásához a MATLAB többféle beépített optimalizációs függvényt tartalmaz, én ezek közül az *fminunc* függvényt választottam, ami megadott f célfüggvény minimumát keresi *Trust Region* algoritmussal. Ezt a célfüggvényt, az RLS algoritmushoz hasonlóan, az adott bemenetre adott tényleges és számított válasz közti hibából képezem. A módszer hátránya, hogy mivel egy célfüggvény minimumkeresésére képes, az adott célfüggvényt a meglévő összes mintából kell összeállítani, így nagy komplexitású problémát kell megoldania, és nincs lehetőség az algoritmus megállítására, amennyiben a megoldás már konvergál.

A célfüggvény összeállításához ebben az esetben is a diszkretizált modellből indulhatunk ki. Az ellenállás és induktivitás identifikációjához már használt 22. egyenletet többféle módon használhatjuk, hogy megtaláljuk a legoptimálisabb futtatást:

- egyrészt felhasználhatjuk a teljes egyenletet, így párhuzamosan keresve az armatúra ellenállását, induktivitását és a motor fluxusát. Ekkor az eljárással meghatározott együtthatók praktikusán:

$$a_1 = 1 - \frac{R}{L}T_s, \quad b_1 = \frac{T_s}{L} \quad \text{és} \quad c_1 = -\frac{T_s\Psi_p}{L} \quad (82)$$

- De kereshetjük először az egyszerűsített 52. egyenlet alapján a megoldást a_1 -re és b_1 -re, majd ezek meghatározása után az optimalizációt újra végrehajtva úgy, hogy már csak c_1 együtthatót keressük.

A célfüggvényt ezek alapján úgy állítom elő, hogy minden mintánál kiszámolom a tényleges kimenet és a számított kimenet különbségét paraméteresen, majd az így keletkezett hibák négyzetét összeadom. Így az első megközelítésben kimenet a q irányú áramkomponens, a bemenetek az u_q beavatkozó feszültség, az ω_1 villamos szögsebesség és a d irányú áramkomponens, így n minta esetén a felírható célfüggvény:

$$C = \sum_{i=0}^n e^2(i) = \sum_{i=0}^n [i_q(i) - (a_1 i_q(i-1) - T_s \omega_1(i-1) i_d(i-1) + b_1 u_q(i-1) + c_1 \omega_1(i-1))]^2 \quad (83)$$

Ennek a függvénynek az a_1 , b_1 és c_1 paramétereit keresi az *fminunc* függvény *trust region* algoritmussal.

A második megközelítés költségfüggvénye a motor egyszerűsített, lefogás esetén pontos, modelljéből:

$$C = \sum_{i=0}^n e^2(i) = \sum_{i=0}^n [i_q(i) - (a_1 i_q(i-1) + b_1 u_q(i-1))]^2 \quad (84)$$

Ekkor az a_1 és b_1 együtthatók azonosíthatók, majd az előző (83) függvénnyel a c_1 együttható is megkereshető, úgy, hogy a_1 és b_1 paramétereket már ismerjük.

A mechanikus paraméterek becsléséhez a 57. egyenletből indulok ki, ahol a bemenet a q irányú áramkomponens, a kimenet a villamos szögsebesség. Így a *Trust Region* algoritmussal keresett együtthatók:

$$d_1 = 1 - \frac{bT_s}{J} \quad \text{és} \quad e_1 = \frac{PK_t T_s}{J} \quad (85)$$

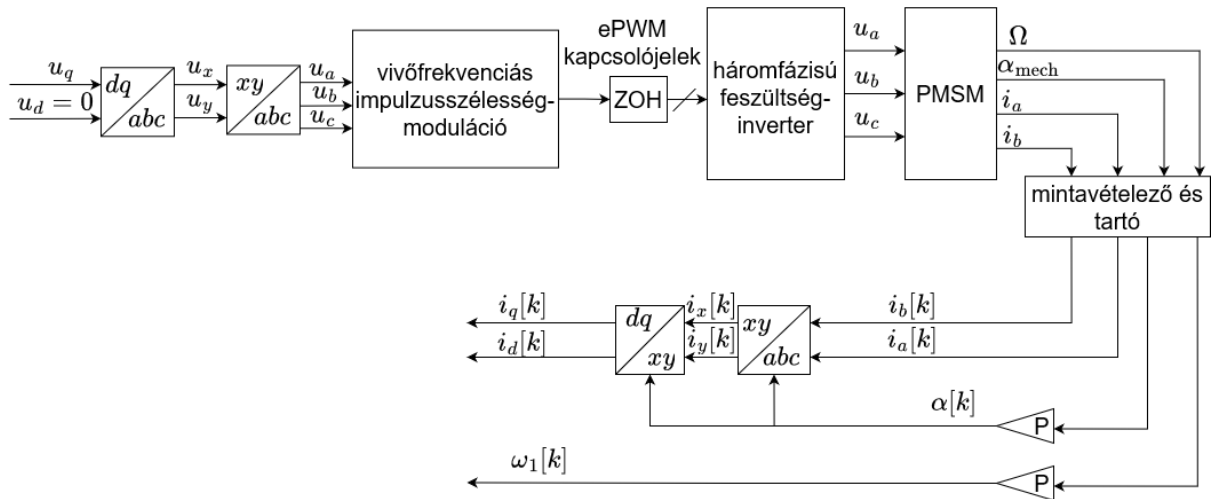
Az együtthatók segítségével a villamos paraméterek meghatározásához hasonlóan felírható a használt költségfüggvény:

$$C = \sum_{i=0}^n e^2(i) = \sum_{i=0}^n [\omega_1(i) - (d_1 \omega_1(i-1) + e_1 i_q(i-1))]^2 \quad (86)$$

4. Szimulációs eredmények

A szimuláció felépítése a 9. ábrán látható. A 2.3. fejezetben részletezett koordinátatranszformációk segítségével állítható elő a háromfázisú referenciajel az 2.4. fejezetben leírt vivőfrekvenciás impulzusszélesség moduláció számára. Ez biztosítja kapcsolójeleket az inverter számára, így képezve a PMSM beavatkozójelét.

Az inverterkártyával mérhető az i_a és i_b áramkomponens, valamint enkóderrel megvalósítható a mechanikai szögelfordulás és szögsebesség mérése, ezeket megadott frekvenciával mintavételezzük. Ezután az áramkomponensekből a 2.3. fejezetben látott koordinátatranszformáció segítségével előállíthatók az identifikációhoz szükséges $d-q$ áramkomponensek, valamint a 2. egyenletek alapján P -vel való szorzással a villamos szögelfordulás és szögsebesség is számítható.



9. ábra. A paraméteridentifikációhoz használt modell felépítése

Ez alapján elkészíthető a PMSM motor modellje MATLAB Simulink környezetben.

A szimuláció során a PMSM motor paramétereinek a valós mérésekhez rendelkezésre álló Hurst gyártmányú MA0204024B101 motor katalógusadatait használtam. Ezek az 1. táblázatban olvashatók. A motor viszkózus csillapítása nincs megadva a katalógusban, ezért ennek $B = 1 \cdot 10^{-3}$ Ns/m értéket állítottam be a teszteléshez, ami hasonló méretű motorok esetén tipikusnak tekinthető.

1. táblázat. A motor katalógusadatai

Tulajdonság	Értéke
Póluspárok száma (P)	5
DC feszültség (U_{DC})	24 V
Ellenállás (R)	0,285 Ω
Induktivitás (L)	$3,2 \cdot 10^{-4}$ H
Pólusfluxus (ψ_p)	$7,8901 \cdot 10^{-3}$ Wb
Nyomatékkonstans (K_t)	59,176 mNm/A
Tehetetlenségi nyomaték (J)	$1,7724 \cdot 10^{-5}$ kgm ²

4.1. Az ellenállás és az induktivitás becslése

Az ellenállás és az induktivitás meghatározása során a 52. egyenletnek megfelelően a kimenet a q irányú áramkomponens, míg a bemenet az u_q feszültségkomponens. A szimuláció biztosítja minden lépéshez a bemenő- és kimenőjel értékpárokat, ami alapján az identifikáció bármelyik módszerrel elvégezhető.

Mivel a valós identifikáció során a mért jelek zajosak lesznek, a szimulációban a kimenőjeleket zajjal terheltem, így modellezve a valós mérésből eredő zajt.

Ebben a fejezetben az 52. egyenletből indulok ki, ami csak lefogott forgórészű motor esetében modellezi tökéletesen a működést. Mivel a valós mérés során nem lesz lefogva a motor, fontos, hogy az u_d feszültségkomponens zérus legyen, ezzel csökkentve az i_d áramkomponenset, ami így kisebb szerepet játszik (22)-ben.

Mindhárom identifikációs módszer esetén az előző fejezetben definiált a_1 és b_1 együtthatók értékét keressük. A 82. egyenlet alapján az együtthatókból az ellenállás és az induktivitás kifejezhető, így

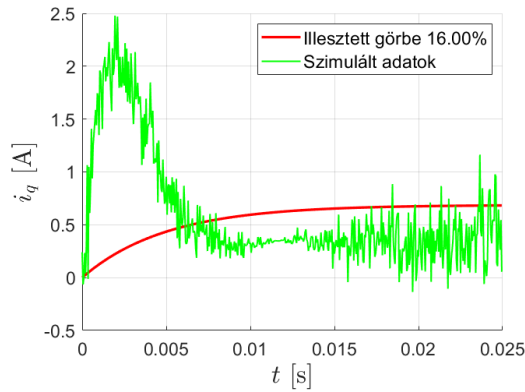
$$R = \frac{1 - a_1}{b_1} \text{ és } L = \frac{a_1}{b_1} T_s \quad (87)$$

képletek adódnak.

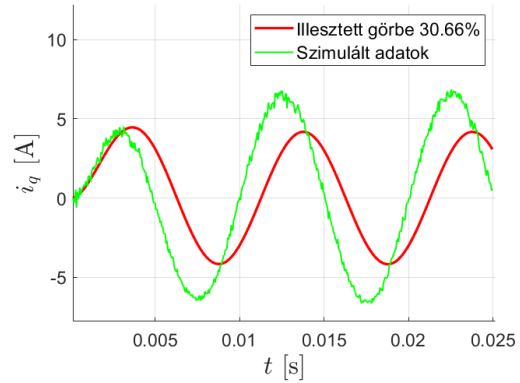
4.1.1. RLS algoritmus

A 52. egyenletből kiindulva az a_1 és b_1 paramétereket fogja tartalmazni a rekurzívan identifikált paramétervektor. Minden lépésben az új adatok érkezésekor elvégezve az RLS algoritmust ((46)-(49)), a paramétervektor értékei egy idő után konvergálnak, így megkaphatók az együtthatók. Ebből a 87. egyenlet segítségével kapható meg az ellenállás és induktivitás értéke.

A szimulációt elvégeztem $u_q = 1$ V-os gerjesztőfeszültség, valamint szinuszos $u_q = \sin(2\pi f_g t)$ gerjesztés esetén is. A paramétervektort megkapva kiszámoltam az adott gerjesztések esetén az identifikált paraméterek által adott válaszfüggvényt, majd ezt összevetettem a szimulált válaszfüggvénnyel. A kapott eredményeket a 10. ábra tartalmazza.



(a) Konstans $u_q = 1$ V gerjesztőfeszültségre adott válasz $R = 1,4544 \, \Omega$, $L = 0,0072 \, \text{H}$



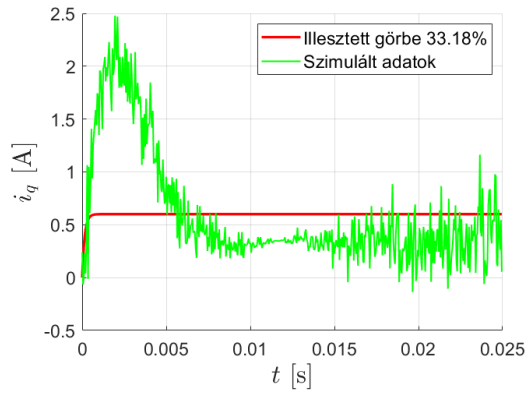
(b) Szinuszos gerjesztőfeszültségre adott válasz $R = 0,3402 \, \Omega$, $L = 5,2870 \cdot 10^{-4} \, \text{H}$

10. ábra. Az i_q áramkomponens időfüggvényei

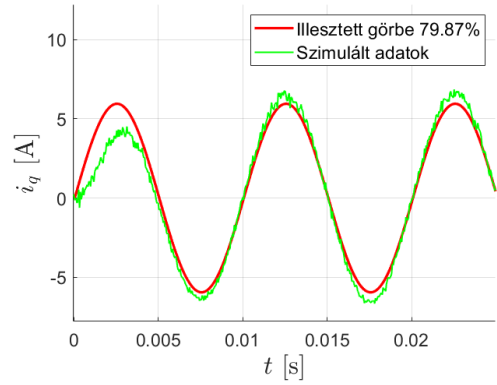
Látható, hogy a szinuszos esetben közelítenek a kapott eredmények a keresett motor paraméterekhez, viszont konstans gerjesztés esetén a kapott görbe teljesen eltér a szimulált válaszfüggvénnytől. Ennek megfelelően ebben az esetben a kapott paraméterek sem megbízhatók, így az RLS algoritmussal meghatározott villamos paraméterek értékeihez a szinuszos gerjesztés vizsgálatára hagyatkozom a továbbiakban.

4.1.2. ARMAX modell identifikáció

Először (73)-ból kiindulva vizsgálom az a_1 és b_1 együtthatókat. Ezek felhasználásával az RLS algoritmushoz hasonlóan 10. egyenlettel meghatározhatók a keresett paraméterértékek. A predikciós hibabecslő eljárással végrehajtott görbeillesztés eredményei a következő ábrán láthatók konstans, illetve szinuszos gerjesztés esetén.



(a) Konstans $u_q = 1$ V gerjesztőfeszültségre adott válasz $R = 1,6650 \, \Omega$, $L = 2,0226 \cdot 10^{-4}$ H



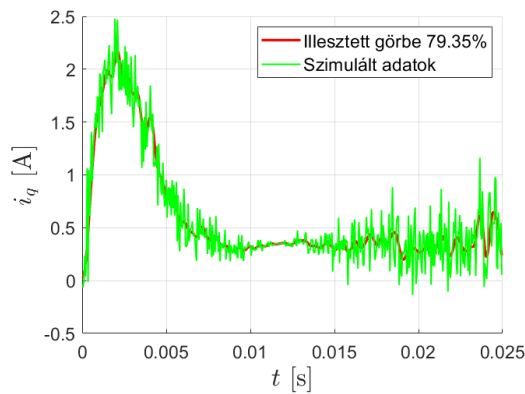
(b) Szinuszos gerjesztőfeszültségre adott válasz $R = 0,3363 \, \Omega$, $L = 4,8197 \cdot 10^{-6}$ H

11. ábra. Az i_q áramkomponens időfüggvényei

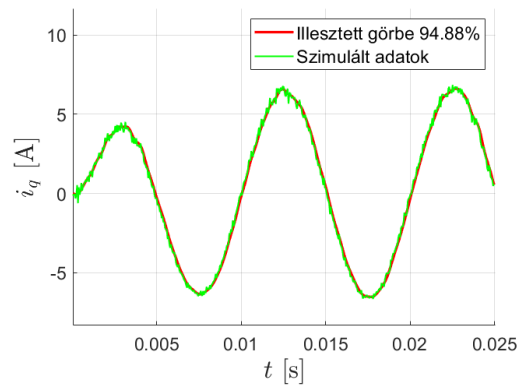
Ezzel a módszerrel már valamivel közelebbi időalakot kapunk a szimulációs adatokhoz konstans gerjesztés esetén is, de a kapott paraméterek ebben az esetben is használhatatlanok. Szinuszos u_q feszültség esetén közelebbi értékek adódnak az elvárt ellenállás és induktivitás értékekhez.

4.1.3. Trust Region optimalizáció

A *Trust Region* módszer esetén az előzőekhez hasonlóan az 52. egyenletből indulok ki és megvizsgálom a szimulációs eredményeket. A 84. egyenlet költségfüggvényét minimalizálva az algoritmus a keresett paraméterekre a 12. ábrán látható eredményeket adja.



(a) Konstans $u_q = 1$ V gerjesztőfeszültségre adott válasz $R = 1,3757 \, \Omega$, $L = 0,0311$ H

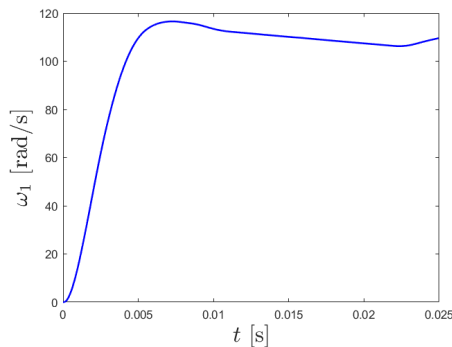


(b) Szinuszos gerjesztőfeszültségre adott válasz $R = 0,3282 \, \Omega$, $L = 5,1625 \cdot 10^{-4}$ H

12. ábra. Az i_q áramkomponens időfüggvényei lefogás nélkül

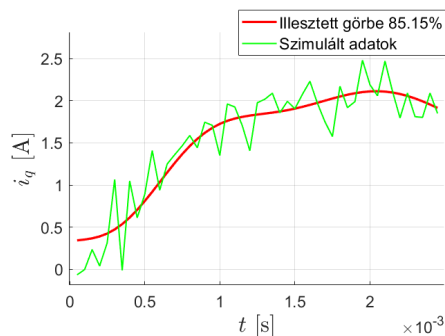
A szinuszos esetben megint az elvárthoz közeli eredményeket kaptunk, viszont konstans

teszt feszültség esetén jelentősen eltérő értékeket tapasztalunk. Ennek oka, hogy ebben az esetben a motor szögsebessége beáll $\omega_{\text{mech}} \approx 25 \text{ rpm} \approx 2,6 \text{ rad/s}$ körüli értékre a következő ábrán látható módon.



13. ábra. A mechanikai szögsebesség időfüggvénye konstans gerjesztőfeszültség esetén

Ezért a 22. egyenlet ω_1 -val rendelkező tagjai nem elhanyagolhatók, így az egyszerűsített modellünk használhatósága mindhárom felhasznált identifikációs módszer esetén korlátozódik. Ha csak a szimuláció elejét vizsgáljuk, amíg a szögsebesség felfut, akkor a becsült paraméterek pontosabbak tekinthetők.



14. ábra. Az i_q áramkomponens időfüggvénye és az optimalizáció által illesztett görbe a mérés elején

Ha csak a szimuláció első tizede alapján futtatom a *Trust Region* algoritmust az $R = 0,4372 \Omega$ és $L = 4,9745 \cdot 10^{-4} \text{ H}$ értékek adódnak, amik nem állnak messze a várakozásoktól. Ebben az esetben tehát a paraméterértékek nemhogy konvergálnak az idő múlásával, hanem egyre tévesebbek lesznek. Ezt láttuk az RLS algoritmus és az ARMAX identifikáció konstans gerjesztésű esetében is. Viszont az első 50 mintára hagyatkozni egy motor identifikációjához nem célszerű, hiszen egy valós esetben ebben a tartományban közrejátszhat a tapadási súrlódás és egyéb indítási jelenségek, amikkel a szimuláció során nem számolunk. Érdemesebb lehet kizárólag a szinuszos gerjesztésű eset eredményeire hagyatkozni, vagy egy bonyolultabb modellel, a nemlinearitások figyelembe vételével megközelíteni a problémát.

4.2. A pólusfluxus és nyomatékkonstans becslése

A pólusfluxus azonosításához a c_1 együtthatók értékét keressük ARMAX és *Trust Region* módszerekkel. Ebből (82) alapján

$$\psi_p = -\frac{L}{c_1 T_s} \quad (88)$$

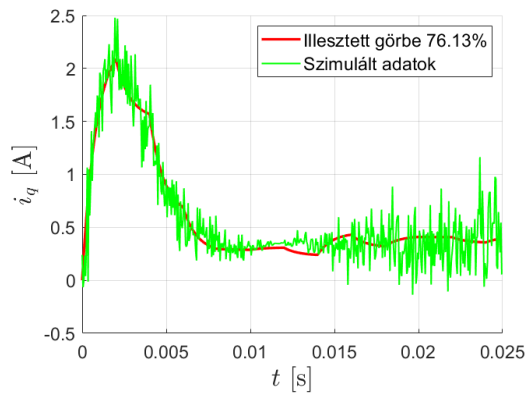
képlettel meghatározható a pólusfluxus. Ebből a nyomatékkonstans is meghatározható a 15. egyenlet alapján:

$$K_t = \frac{3}{2} P \psi_p \quad (89)$$

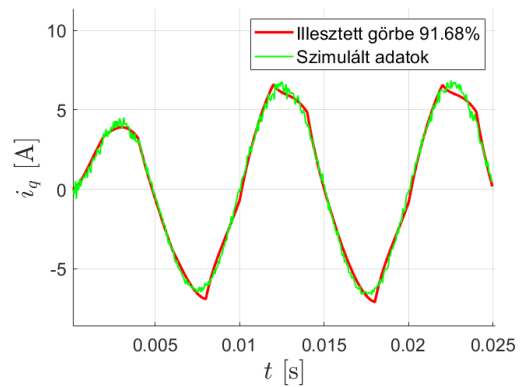
Ebben az esetben viszont már szükséges mérni a mechanikai szögsebességet is. Valós környezetben a mérést digitálisan végezzük, a szögsebességet enkódertárcsával mérjük, így a szögsebesség csak kisebb mintavételi frekvenciával vizsgálható (ennek részletes indoklása az 5.3. fejezetben olvasható). A szögsebesség érzékeléséhez 0,5 kHz-es mintavételezést fogunk használni, ezért érdemes a szimulációban is már ezzel a frekvenciával mérni a szögsebességet. Ebben az esetben 40 árammérésre jut egy szögsebesség adat, aminek feldolgozásához nulladrendű tartót alkalmazok.

4.2.1. ARMAX modell identifikáció

A fenti két paraméter mellett (74) felhasználásával becsülhető a pólusfluxus értéke is. A két bemenettel végrehajtott identifikáció eredményei a 15. ábrán láthatók.



(a) Konstans $u_q = 1$ V gerjesztőfeszültségre adott válasz $R = 0,4271 \Omega$, $L = 3,4850 \cdot 10^{-4}$ H, $\psi_p = 0,0073$ Wb, $K_t = 0,0548$ Nm/A



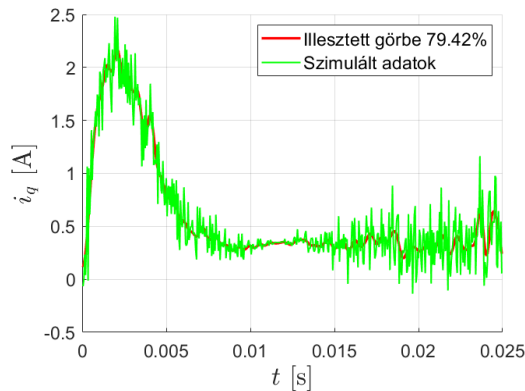
(b) Szinuszos gerjesztőfeszültségre adott válasz $R = 0,4060 \Omega$, $L = 2,7581 \cdot 10^{-4}$ H, $\psi_p = 0,0074$ Wb, $K_t = 0,0555$ Nm/A

15. ábra. Az i_q áramkomponens időfüggvényei a komplexebb modell esetén

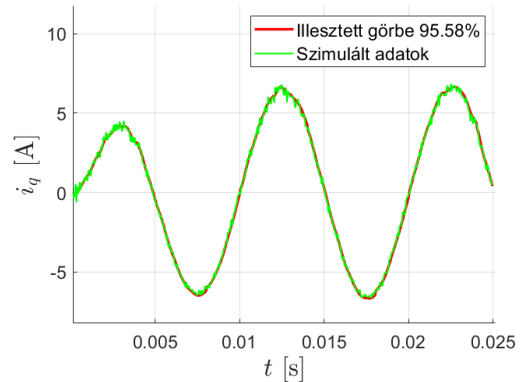
Két bemenetes ARMAX modell alkalmazásával mind konstans, mind harmonikus gerjesztés esetén közelítőleg helyes eredményeket kaptunk az 1. táblázatban látott motoradatokhoz képest.

4.2.2. *Trust Region* optimalizáció

A 4.1. fejezetben meghatározott ellenállás- és induktivitásérték ismeretében (83) alapján a c_1 együttható meghatározható a költségfüggvény minimalizálásával. Az ábrán láthatók a *Trust Region* algoritmus által illesztett görbék a zajjal terhelt szimulációs áramalakokkal összevetve.



(a) Konstans $u_q = 1$ V gerjesztőfeszültségre adott válasz $\psi_p = 0,0050$ Wb, $K_t = 0,0375$ Nm/A



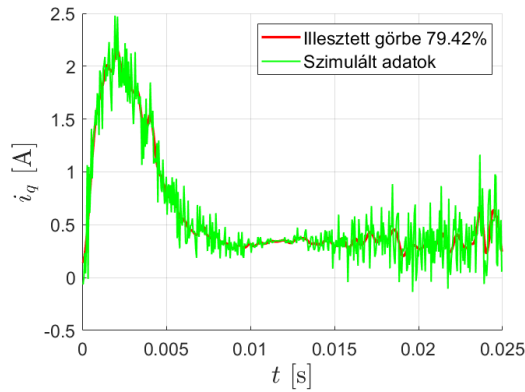
(b) Szinuszos gerjesztőfeszültségre adott válasz $\psi_p = 0,0063$ Wb, $K_t = 0,0473$ Nm/A

16. ábra. Az i_q áramkomponens időfüggvényei, feltételezve, hogy az ellenállás és az induktivitás ismert

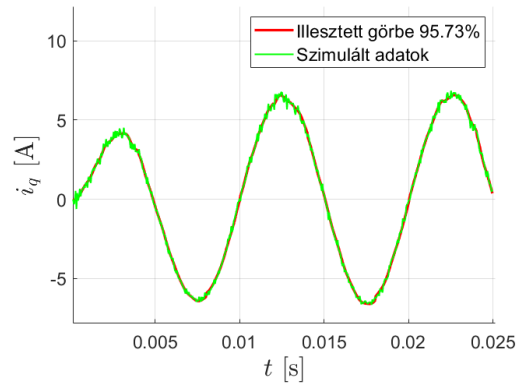
Látható, hogy az eredmények az ARMAX modell identifikációhoz képest pontatlanabban.

Az ellenállás, induktivitás és a pólusfluxus együttes becslése

Mivel a pólusfluxus becslése nem volt kielégítő az egyszerűsített modell esetén, vizsgáljuk meg a komplexebb, az a_1 , b_1 és c_1 paramétereket egyszerre identifikálandó költségfüggvényen futtatott *Trust Region* optimalizációt. Ekkor a költségfüggvény a 83. egyenletben látható függvény, amiből a paraméterek (87), (88) és (89) alapján határozhatók meg. Ekkor a 17. ábrán láthatók az optimalizáció által illesztett görbék.



(a) Konstans $u_q = 1$ V gerjesztőfeszültségre adott válasz $R = 0,4144 \Omega$, $L = 4,0543 \cdot 10^{-4}$ H, $\psi_p = 0,0075$ Wb, $K_t = 0,0561$ Nm/A



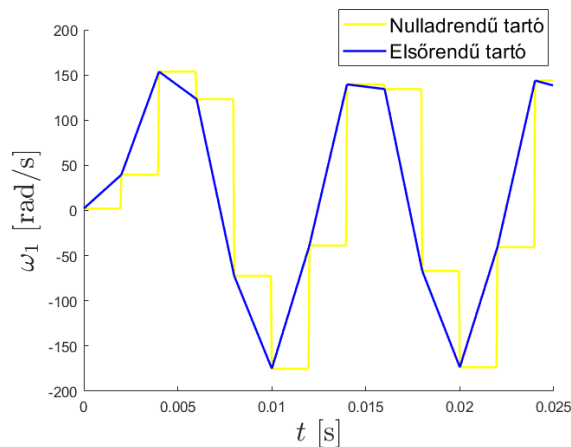
(b) Szinuszos gerjesztőfeszültségre adott válasz $R = 0,4008 \Omega$, $L = 3,0278 \cdot 10^{-4}$ H, $\psi_p = 0,0071$ Wb, $K_t = 0,0535$ Nm/A

17. ábra. Az i_q áramkomponens időfüggvényei a komplexebb modell esetén

Bár az ellenállás értéke pontatlanabb lett, mint amikor külön vizsgáltuk, az induktivitás, a pólusfluxus és a nyomatékkonstans értéke közelebb került a várakozásokhoz a komplexebb modell alkalmazásával.

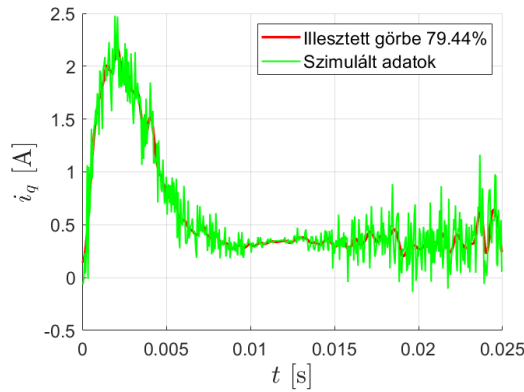
A szögsebességadatok korrekciója

Mivel a szögsebességértékek mintavételi frekvenciája jóval kisebb, mint az árammérésé, az identifikáció során 40 mintán keresztül a szögsebességet állandónak tekintettük. Ez nagyon eltér a valós működéstől, így rontja a becslés pontosságát. Ezért vizsgáljuk meg, hogy a nulladrendű tartószerv helyett két mért szögsebesség érték közé egyenest illesztve, lineáris interpolációval adjuk meg a szögsebesség értékeket.

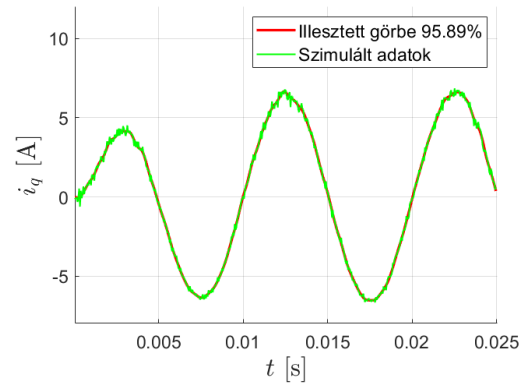


18. ábra. A szögsebesség becslése

Ha így közelítjük a szögsebességet a költségfüggvény összeállítása során, akkor a következő eredményeket kapjuk:



(a) Konstans $u_q = 1$ V gerjesztőfeszültségre adott válasz $R = 0,3015 \Omega$, $L = 4,0557 \cdot 10^{-4}$ H, $\psi_p = 0,0079$ Wb, $K_t = 0,0594$ Nm/A



(b) Szinuszos gerjesztőfeszültségre adott válasz $R = 0,2888 \Omega$, $L = 3,1573 \cdot 10^{-4}$ H, $\psi_p = 0,0082$ Wb, $K_t = 0,0612$

19. ábra. Az i_q áramkomponens időfüggvényei, a szögsebesség lineáris interpolációja után

Ezzel a módszerrel mindkét gerjesztés esetén pontosnak tekinthető paraméterértékeket kaptunk. Az ellenállás és induktivitás szinuszos gerjesztés esetén nagyon jól közelíti az elvárt $R = 0,285 \Omega$, $L = 3,2 \cdot 10^{-4}$ H értékeket, valamint a pólusfluxus és nyomatékkonstans értéke is elfogadható.

4.3. A mechanikai paraméterek becslése

A szögsebesség érzékeléséhez, a pólusfluxus azonosításához hasonlóan 0,5 kHz-es mintavételezést fogok használni, ezért érdemes a szimulációban is már ezzel a frekvenciával mérni a szögsebességet.

A mechanikai folyamatok jóval hosszabb lefutásúak, mint az elektromosak, ezért ebben az esetben jóval hosszabb időintervallumot érdemes vizsgálni, mint az előző mérések során. Mivel az árammérés ilyen időtartam során nagy memóriaterületet igényelne, ezért az áram mintavételezési frekvenciáját is csökkentem, amikor a mechanikai jellemzőket mérem.

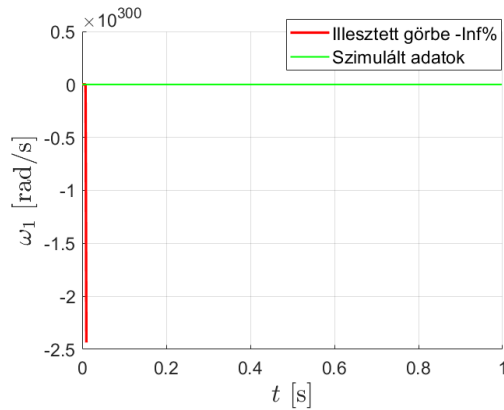
Mivel a mintavételi frekvencia csökkent, a szinuszos gerjesztés frekvenciáját is csökkentem, hiszen a mechanikai egyenlet időállandója jóval nagyobb a villamos egyenleténél, így a nagyfrekvenciás gerjesztésre nem kapnánk értelmezhető választ.

A mechanikai paraméterek becsléséhez a d_1 és e_1 együttható értékeket identificaljuk, úgy, hogy a kimenőjel a szimuláció által biztosított ω_1 villamos szögsebesség, a bemenőjel az i_q áramkomponens. A motor tehetetlenségi nyomatéka és viszkózus csillapítása a következőképpen fejezhető ki a megkapott együtthatókból:

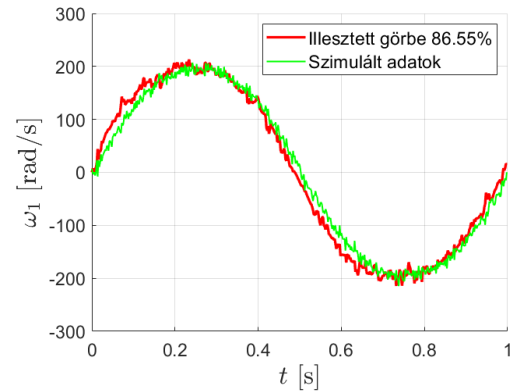
$$J = \frac{PK_t T_s}{e_1} \text{ és } b = \frac{(1 - d_1)J}{T_s} \quad (90)$$

4.3.1. RLS identifikáció

(57)-ből kiindulva d_1 és e_1 együtthatók határozhatók meg az RLS algoritmussal. A csökkentett mintavételi frekvenciával futtatott szimulációs adatokon elvégezve az RLS identifikációt, a következő görbéket kapjuk.



(a) Konstans $u_q = 1$ V gerjesztőfeszültségre adott válasz $J = -1,9183 \cdot 10^{-66}$ kgm², $b = 6,4833 \cdot 10^{-4}$ Ns/m



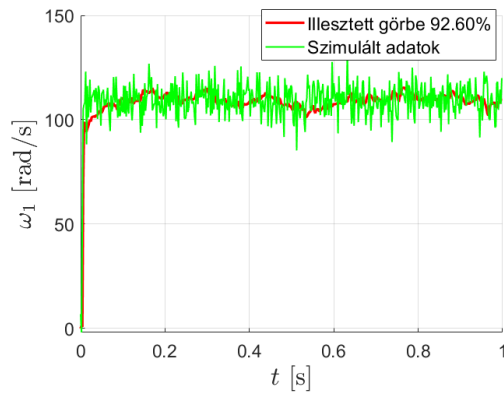
(b) Szinuszos gerjesztőfeszültségre adott válasz $J = 3,0106 \cdot 10^{-6}$ kgm², $b = 6,2950 \cdot 10^{-4}$ Ns/m,

20. ábra. Az ω_1 villamos szögsebesség időfüggvényei

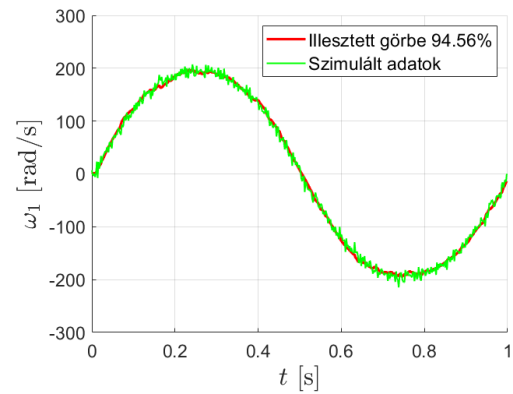
Állandó gerjesztésű esetben a paramétervektor értékei divergálnak, így az általuk kapott grafikon, valamint együtthatók egyáltalán nem fedik a valóságot. A harmonikus esetben a szimulációban beállított J tehetetlenségi nyomatékhoz és viszkózus csillapításhoz, bár pontatlan, de közelebbi értékeket kapunk.

4.3.2. ARMAX modell identifikáció

Bár az RLS-es szimulációval ellentétben konstans gerjesztés esetén is képes jellegre megfelelő görbét illeszteni az algoritmus, a kapott J nagyságrendekkel nagyobb a beállítottnál. Harmonikus gerjesztés esetén a tehetetlenség nyomaték értéke nagy pontossággal adódik ki, míg a viszkózus csillapítási együttható jelentősen nagyobb a várakozásoknál.



(a) Konstans $u_q = 1$ V gerjesztőfeszültségre adott válasz $J = 1,8403 \text{ kgm}^2$, $b = 0,0177 \text{ Ns/m}$



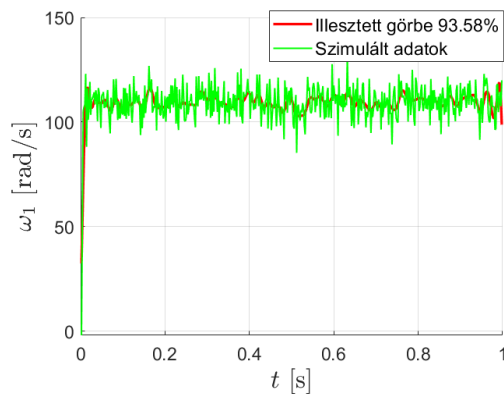
(b) Szinuszos gerjesztőfeszültségre adott válasz $J = 1,4418 \cdot 10^{-5} \text{ kgm}^2$, $b = 0,0138 \text{ Ns/m}$,

21. ábra. Az ω_1 villamos szögsebesség időfüggvényei

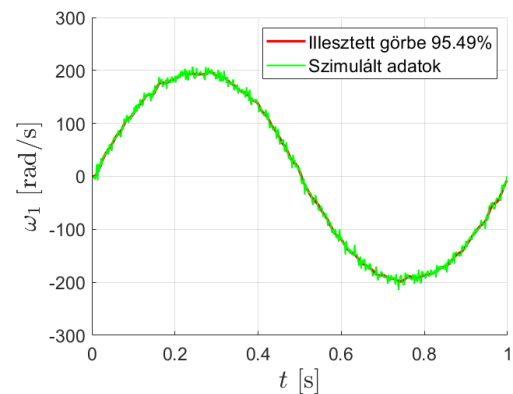
4.3.3. A *Trust Region* algoritmus

Ebben az esetben a használt költségfüggvény a 86. egyenletben található, amiből a d_1 és e_1 együtthatókat identifikálva a keresett mechanikai paraméterek értéke (90) alapján adódik.

Vizsgáljuk meg a következő ábrán látható eredményeket, állandó és harmonikus gerjesztésű esetben:



(a) Konstans $u_q = 1$ V gerjesztőfeszültségre adott válasz $J = 1,7493 \cdot 10^{-5} \text{ kgm}^2$, $b = 6,7339 \cdot 10^{-4} \text{ Ns/m}$



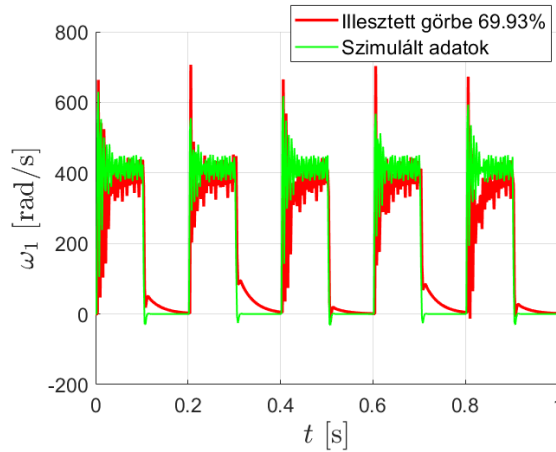
(b) Szinuszos gerjesztőfeszültségre adott válasz $J = 1,5291 \cdot 10^{-5} \text{ kgm}^2$, $b = 6,4343 \cdot 10^{-4} \text{ Ns/m}$,

22. ábra. Az ω_1 villamos szögsebesség időfüggvényei

A *Trust Region* algoritmust alkalmazva mindkét esetben hozzávetőlegesen pontos J és b értékeket kapunk.

4.3.4. Négyszögjel gerjesztéssel történő identifikáció vizsgálata

[16] szerzői a mechanikai paraméterek becsléséhez négyszögjel gerjesztést javasolnak. Ezért végrehajtom ezzel a gerjesztéssel is a méréseket, vagyis a motor feszültségét bizonyos időközönként ki- és bekapcsolva szimulálok. Ha ilyen jelet használok bemenetnek, akkor a görbeillesztés a 23. ábrán látható módon alakul az RLS algoritmus esetén.

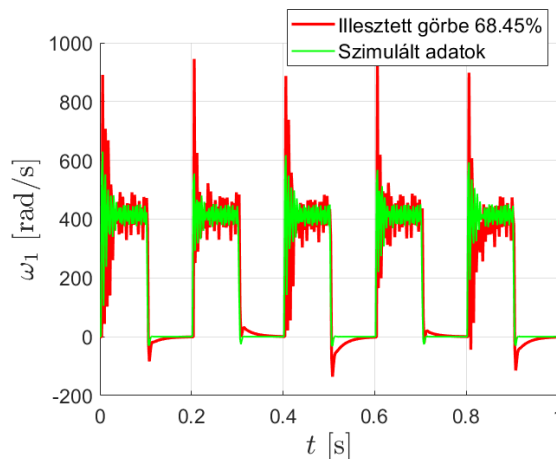


23. ábra. A ω_1 szögsebesség

$$J = 1,6993 \cdot 10^{-5} \text{ kgm}^2, b = 5,4203 \cdot 10^{-4} \text{ Ns/m}$$

A J érték pontosnak tekinthető és a viszkózus csillapítás is elfogadható a beállított értékhez képest.

A következő ábrán láthatók az ARMAX identifikáció eredményei:

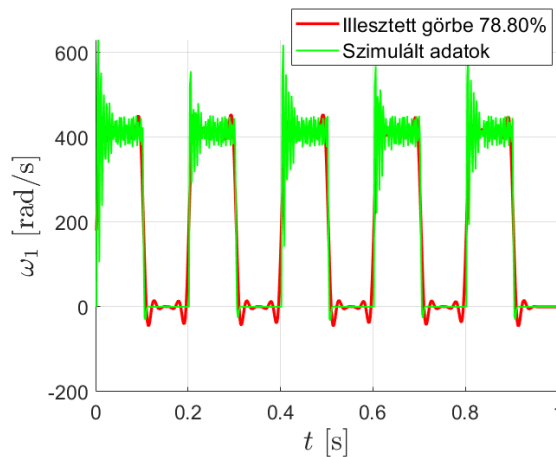


24. ábra. A ω_1 szögsebesség időfüggvénye

$$J = 1,2659 \cdot 10^{-5} \text{ kgm}^2, b = 0,0121 \text{ Ns/m}$$

A tehetetlenségi nyomaték precízen közelíti a beállított $J = 1,77 \cdot 10^{-5} \text{ kgm}^2$ -es értéket, a viszkózus csillapítás viszont egy nagyságrenddel nagyobb az elvártnál.

A *Trust Region* optimalizációval végrehajtott szimuláció eredményei:



25. ábra. A ω_1 szögsebesség időfüggvénye
 $J = 1,5954 \cdot 10^{-5} \text{ kgm}^2$, $b = 0,0012 \text{ Ns/m}$

Ebben az esetben a kapott értékek megint pontosnak tekinthetők.

4.4. A szimulációk alapján javasolt identifikációs eljárás

A vizsgált három algoritmus közül a rekurzív legkisebb négyzetek módszere az egyetlen, ami nem igényel nagy számítási kapacitást, így akár magán a DSP-n is megvalósítható. Ezzel viszont a pólusfluxus identifikációja nem valósítható meg. További fontos különbség a másik két módszerhez képest, hogy az RLS algoritmus megállítható, amennyiben a paramétervektor értékei konvergálnak, így futásideje is kisebb. A másik két algoritmus esetében kísérletekkel határozható meg az identifikációhoz szükséges mintaszám.

A villamos paraméterek meghatározásához a szinuszos gerjesztés bizonyult hatékonyabbnak, az egyes módszerek által kapott eredményeket szemlélteti a 2. táblázat.

2. táblázat. Villamos paraméterekre kapott eredmények a különböző módszerek esetén, harmonikus gerjesztés mellett

Paraméter	Elvárt	RLS	ARMAX		Trust Region			mérték- egység
			egy bem.	két bem.	lin. modell	nemlin. modell	+elsőr. tartó	
R	0,285	0,3402	0,3363	0,4271	0,3282	0,4008	0,2888	Ω
L	3,2	5,2870	4,8197	3,4850	5,1625	3,0278	3,1573	10^{-4} H
Ψ_p	0,0079	-	-	0,0073	0,0050	0,0071	0,0082	Wb
K_t	0,059	-	-	0,0548	0,0375	0,0535	0,0612	Nm/A
Illeszkedés		30,66	79,87	91,66	94,88	95,73	95,89	%

A pólusfluxus azonosítása csak a *Trust Region* és ARMAX optimalizáció segítségével valósítható meg kielégítően. Az ellenállás és induktivitás tekintetében az előbbi a legpontosabbnak mondható eljárás, amennyiben a nemlinearitást figyelembe vevő modellt használva ezeket együtt becsüljük a pólusfluxussal, valamint elsőrendű tartószervert alkalmazunk a szögsebesség mintavételezéséhez. Hátránya, hogy a számításigénye ennek az algoritmusnak legnagyobb, hiszen nagy bonyolultságú költségfüggvény optimalizációja zajlik a háttérben.

A mechanikus paraméterekhez a konstans és négyszögjel típusú gerjesztés bizonyult a hatékonyabbnak, ezek összehasonlítása a 3. táblázatban található. A viszkózus csillapítás értéke a *Trust Region* módszer esetén, négyszögjeles gerjesztés mellett közelítette legjobban a beállított paramétert.

3. táblázat. Mechanikai paraméterekre kapott eredmények a különböző módszerekkel

Paraméter	Elvárt	RLS	ARMAX		Trust Region		mértékegység
		négyszj. gerj.	konst. gerj.	négyszj. gerj.	konst. gerj.	négyszj. gerj.	
J	1,77	1,6993	1,8403	1,2659	1,7493	1,5954	10^{-5} kgm ²
b	1	0,5420	17,7	12,1	0,6734	1,15	10^{-3} Ns/m
Illeszkedés		69,93	92,60	68,45	93,58	78,80	%

Ezek alapján a identifikáció javasolt módja a következő:

- A villamos paraméterek meghatározása a kívánt pontosság függvényében elvégezhető a mikroprocesszoron RLS algoritmus online történő végrehajtásával, szinuszos u_q feszültség ráadása melletti q irányú áramkomponens mérésével, vagy a PC-vel történő kommunikáció segítségével a pontosabb *Trust Region* optimalizációval.
- A pólusfluxus azonosítása a PC-n futtatott *Trust Region* algoritmussal történik, az ellenállással és induktivitással együtt. Ezt a mérést szinuszosan gerjesztett rendszer esetén mind az áramokat, mind a szögsebességet mérve végzem, hiszen ekkor kaptuk a legprecízebb paraméterértékeket. A szögsebességet az előző fejezetben látható módon lineáris interpolációval közelítve történik a legpontosabb becslés.
- A mechanikai együtthatókhoz az inverter ki- és bekapcsolásával állítunk elő négyszögjel alakú u_q bemenő feszültséget, majd ezeket is a külső számítógépen *Trust Region* módszerrel határozzuk meg. Emellett annak érdekében, hogy az identifikáció minél nagyobb része megvalósítható legyen a PC-vel való kommunikáció nélkül, a tehetetlenségi nyomaték és viszkózus csillapítás RLS algoritmussal történő azonosítását is implementálok, ezzel a folyamat számítási igénye csökkenthető.

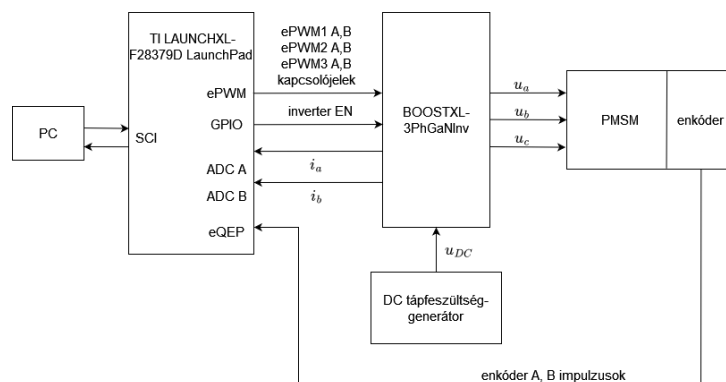
5. Hardveres megvalósítás

Az előző fejezetbeli eredményekből kiindulva valós környezetben két módszert valósítok meg:

- az RLS algoritmust, mivel ez lehetőséget ad a PC közbenjárása nélküli identifikációra és a
- *Trust Region* optimalizációs identifikációt, mivel ez a nemlineáris modellek kezelése miatt nagyon pontos szimulációs eredményeket produkált.

5.1. A rendszer áttekintése

A 26. ábra szemlélteti a rendszer összeállítását. Mind a paraméteridentifikációhoz szükséges mérések, illetve az RLS algoritmus esetében maga a paraméterek meghatározása is, mind az az alapján történő szabályozás digitálisan kerül megvalósításra, ami egy, a Texas Instruments által gyártott, LAUNCHXL-F28379D LaunchPad[17] típusú mikrovezérlőn fog futni. Az eszköz a TI TMS320F28379D jelfeldolgozó processzorját tartalmazza.



26. ábra. A rendszer vázlata

A DSP a TI BoosterPack interfészen keresztül csatlakozik a TI BOOSTXL-3PHGANINV[3] típusú kétszintű, háromfázisú inverterhez, aminek kapcsolóelemei GaN alapú tranzisztorok. Ezen keresztül történik a PMSM motor meghajtása: három félhídja segítségével kapcsolja a motor három fázisára a DC feszültséget, amit egy egyenáramú MATRIX MPS-3005 lab típusú feszültségforrás biztosít. Az inverterkártya kapcsolójeleit a LaunchPad ePWM pinjeiről kapja, továbbá a DSP egyik GPIO-jának segítségével engedélyezhető az inverter működése.

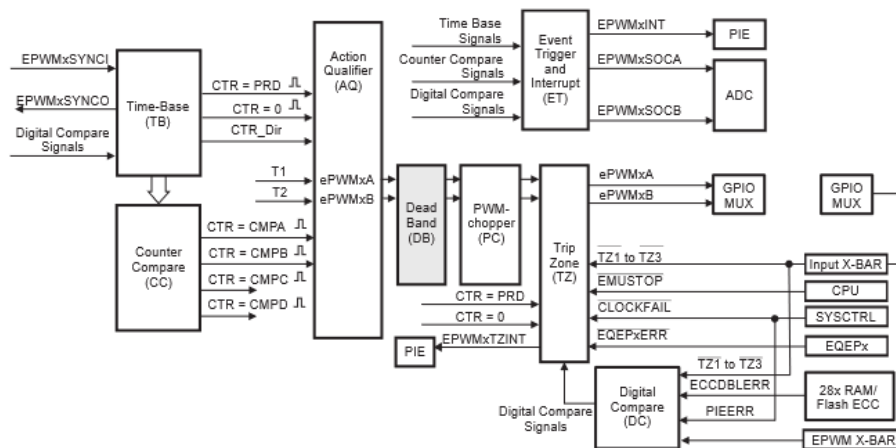
A 3PHGANINV segítségével megvalósítható a PMSM áramának és feszültségének mérése, amiket a mikrovezérlő analóg konverziós modulja fogad. Esetünkben az i_a és i_b fázisok áramát mérjük ennek segítségével, hiszen ezek alapján előállíthatók a $d - q$ irányú áramkomponensek a számításokhoz.

Maga a motor egy Hurst gyártmányú DMA0204024B10 motor, amihez inkrementális relatív enkóder tartozik. Az enkóder jelei a LaunchPad eQEP (*enhanced Quadrature Encoder Pulse*) perifériáján keresztül dolgozhatók fel, így mérhető a PMSM mechanikai szögelfordulása. A szögelfordulásból a DSP számítja az identifikációhoz szükséges mechanikai, majd a villamos szögsebességet.

A folyamatok irányítását egy külső számítógép végzi, ami a mikroprocesszorral soros UART kommunikációt folytat, amit a TI SCI moduljával valósítok meg. A PC-ről érkeznek az elvégzendő feladatok parancsai, a mikrovezérlőről pedig áramerősség és szögsebesség adatok továbbítódnak a PC felé.

5.2. ePWM modul

Az inverter kapcsolójeleit a DSP ePWM modulja biztosítja. Ennek felépítését szemlélteti a 27. ábra.



27. ábra. Az ePWM periféria felépítése [17]

A *Time-Base* almodullal konfigurálható a PWM frekvenciája, ami esetünkben a mintavételezési frekvenciával megegyezik, hiszen ez fogja kiváltani a az AD konverziót és így az azt feldolgozó megszakításrutint is. Az áram mintavételezése nagy frekvenciával kell történjen, mivel a rendszer villamos időállandója μs nagyságrendű, ezért a PWM frekvenciáját 20 kHz-nek választom.

A *Counter Compare* almodullal konfigurálhatók minden mintavételezési ciklusban a komparátor regiszterek, amik a vivőfrekvenciás impulzusszélesség-moduláció referenciajelet adják. A komparálás folyamatát az *Action-Qualifier* modul végzi, így ezzel billenthetők az ePWM periféria kimeneti jelei magas és alacsony szintre.

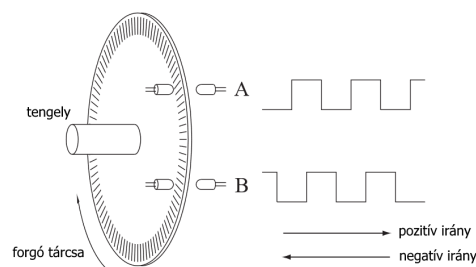
A tranzisztorok esetén általános, hogy a kikapcsolás folyamata lassabban zajlik le, mint a bekapcsolás. Mivel az inverter félhídjait komplementer kapcsolójelek vezérlik, előfordulhat, hogy egy rövid ideig mindkét tranzisztor vezet, ezért rövidzárlat lép fel. Ekkor

a tranzisztoron energia disszipálódik hő formájában, ami a félvezető eszközök élettartamát csökkenti és a kapcsolás hatékonyságát is rontja. Ennek elkerülése érdekében fontos, hogy az egyik tranzisztor egy bizonyos idővel korábban kezdjen el kikapcsolni, mint, hogy a másik bekapcsol, vagyis a két kapcsolás között holtidőt kell vinni a rendszerbe. Ezt az ePWM modul *Dead Band* rendszerével oldhatjuk meg, aminek RED és FED regisztereinek beállításával biztosítható, hogy az egyik tranzisztor lefutó ága és a másik felfutó ága között legyen holtidő, így semmiképpen nem tud rövidzárlat fellépni.

A holtidő megjelenése miatt nem valósítható meg ideális működés, vagyis valamivel kisebb feszültségek fognak kerülni a motor fázisaira, így a szimulációhoz képest kisebb áramalakokat kapunk.

5.3. A szögsebesség mérése

Az inkrementális jeladók, vagy enkóderek az egyik leggyakoribb szögelfordulás és szögsebesség mérésére alkalmas eszközök. Az enkódertárcsa a motor tengelyére van szerelve és annak elfordulásakor digitális impulzusokat ad ki a kimenetein. Ezekből az impulzusokból a motor szögelfordulásán kívül a tengely elfordulásának iránya is meghatározható. Az enkódertárcsa felépítését a 28. ábra mutatja be.



28. ábra. Az enkóder működési elve

Az enkódertárcsa a pereme közelében átlátszó ablakkal rendelkezik, így azok forgás közben vagy átengedik vagy meggátolják a fény útját. Optokaput elhelyezve a tárcsa kerülete mentén négyszögjel kimenetet kapunk. Az A és B optokaput 90° fáziseltéréssel elhelyezve az elfordulás iránya is meghatározható.

Az általunk használt enkódertárcsán az A és B kimeneti jelek egyenként 250 impulzust adnak egy elfordulás alatt, így a felfutó és lefutó éleket figyelve a két csatornán összesen 1000 jelváltozás figyelhető meg egy elfordulás alatt.

Így az enkóder felbontása

$$d_\alpha = \frac{2\pi}{1000} = 6,283 \cdot 10^{-3} \text{rad} = 0,36^\circ \quad (91)$$

A szögsebességet bizonyos mintavételezési frekvenciával figyelve az adott időtartam alatti

mechanikai szögelfordulásból számíthatjuk:

$$\Omega = \frac{\Delta\alpha_{\text{mech}}}{\Delta t} \quad (92)$$

A mintavételezési időt úgy kell megválasztani, hogy a megállapított felbontásnál nagyobb legyen az elfordulás az általunk használt sebességtartományban. Ezért a szögsebesség mérésénél 0,5 kHz-es mintavételi frekvenciát fogok használni.

A mechanikai szögelfordulás az inkrementációk számából, amit az eQEP periféria QPOSCNT regisztere tárol,

$$\alpha_{\text{mech}} = \frac{\text{QPOSCNT} \cdot 2\pi}{1000} \quad (93)$$

képlettel számítható. Ebből a villamos szögelfordulás a póluspárok számának ismeretében:

$$\alpha = P \cdot \alpha_{\text{mech}} \quad (94)$$

A Δt mintavételi idő 0,5 kHz-es frekvencia esetén $\Delta t = \frac{1}{500} = 0,002$ s, így a mechanikai szögsebesség az adott idő alatti inkrementációk számából

$$\Omega = \frac{\frac{\text{QPOSCNT} \cdot 2\pi}{1000}}{0,002} = 31,415 \cdot \text{QPOSCNT} \quad (95)$$

aminek számításakor figyelembe kell venni a forgás irányát, valamint azt, amikor a pozícióregiszter QPOSCNT = 1000 értéknél túlszordul és 0-ról folytatja a számolást. Ez alapján a villamos szögsebesség is meghatározható:

$$\omega_1 = P \cdot \Omega \quad (96)$$

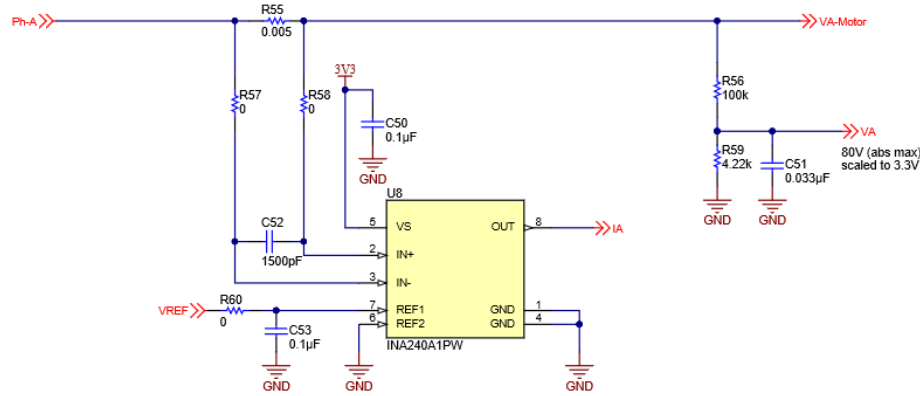
Mivel a használt enkóder nem abszolút enkóder, ahhoz, hogy a szögelfordulás megfelelően mérhető legyen, elengedhetetlen, hogy legyen egy referenciapont, amihez képest figyeljük a szöget. Ezért a motor inicializálásakor egy előre meghatározott helyzetbe forgatom a PMSM-et, így a mérés mindig ugyanabból a helyzetből indul. Ennek folyamata a 5.5.1. részben olvasható.

5.4. Az áram mérése

Az a és b fázisú áramkomponensek mérése analóg módon történik az inverterkártyán, majd ezt az analóg jelet a DSP analóg-digitális konverziós perifériája dolgozza fel. Mivel a mikrovezérlő három ADC modullal rendelkezik, képesek vagyunk párhuzamosan mérni mindkét áramjelet. Az ADC-k 12 bites üzemmódban végzik a konverziót, amit az ePWM periféria időzítője triggerel, így 20 kHz-es mintavételi frekvenciával dolgozik.

A LaunchPad analóg jeleket 0 és 3 V között képes fogadni, ezért szintillesztő áramkörre van szükség, amit az inverter áramköri rajzában lehet megtalálni. Az fázisáramok szintillesztő áramköre a 29. ábrán látható.

Phase Current/Voltage Sense



29. ábra. Az áramjelek szintillesztő áramköre [2]

Az áramokat egy $0,005\ \Omega$ értékű ellenálláson keresztül méri az INA240A1[7] típusú műveleti erősítő, ami 20-as erősítéssel rendelkezik. A V_{ref} referencia feszültség $1,65\ \text{V}$. Ezeknek az adatoknak ismeretében az erősítő kimenete felírható:

$$u_{ki} = 1,65 - 20 \cdot 0,005 \cdot i_f \quad (97)$$

ahol $f = \{a; b\}$.

Mivel az ADC felbontása 4095 és a bemeneti feszültségtartományának szélessége $3\ \text{V}$, az ADC kimenete és a fázisáramok közti összefüggés

$$ADC_{ki} = (1,65[\text{V}] - 20 \cdot 0,005[\Omega] \cdot i_f) \frac{4095}{3[\text{V}]} \quad (98)$$

amiből kifejezhetők a fázisáramok:

$$i_f = \frac{1}{0,1[\Omega]} \left(1,65[\text{V}] - \frac{ADC_{ki}}{4095} 3[\text{V}] \right) \quad (99)$$

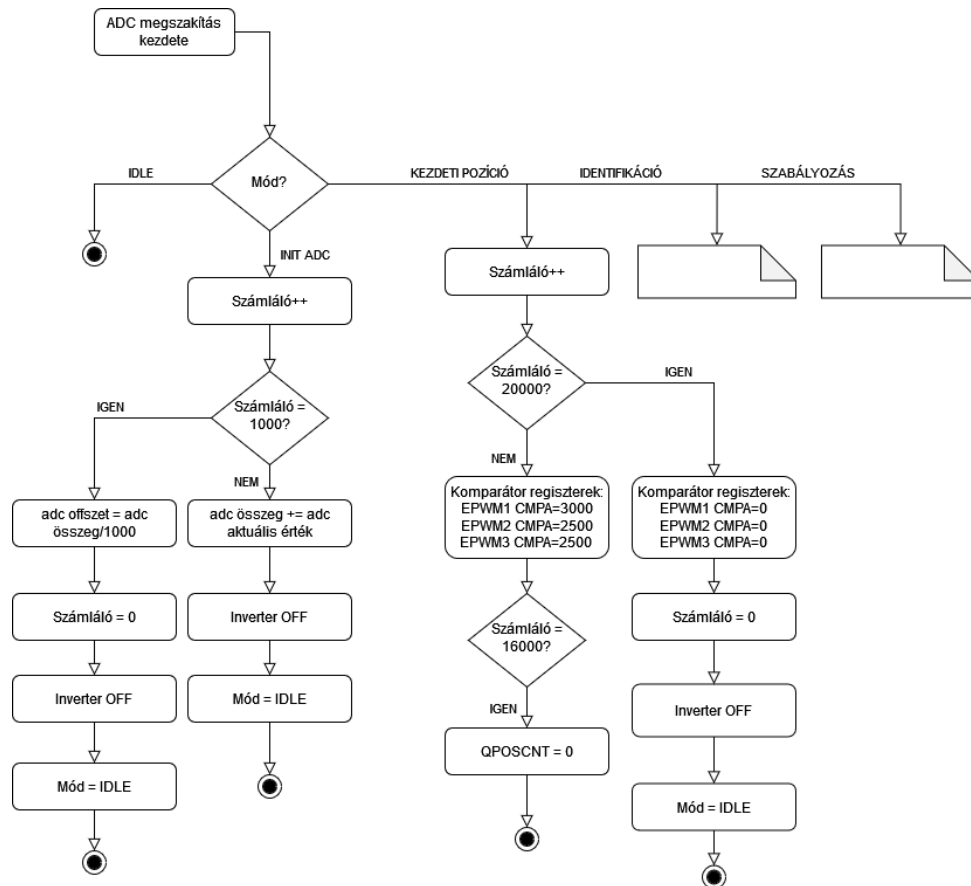
Ennek segítségével az áramértékek kiszámíthatók a konverzió után.

5.5. A program működése

A TI TMS320F28379D LaunchPaden futó program a gyártó C2000Ware szoftverfejlesztő csomagjával készült C nyelven, a TI Code Composer Studio 12.1.0. környezetben.

A perifériák a main() függvényben kerülnek inicializálásra, valamint ez indítja el a kommunikációt a PC-vel és a továbbiakban figyeli a soros porton keresztül kapott parancsokat. A fő funkciók, amik ezen parancsok hatására hajtódnak végre, az ADC megszakításkezelő rutinjában történnek, amit az ePWM periféria triggerrel $20\ \text{kHz}$ -es frekvenciával. Ez folyamatosan megszakítja a main() függvényt.

Az interrupt rutinon belül egy állapotgép felelős az aktuális mód beállításáért. A program lehetséges módjait a 30. ábra szemlélteti:



30. ábra. A program módjai

- IDLE mód: Ilyenkor utasításra vár a program.
- INIT ADC mód: Az ADC modul rendelkezik egy bizonyos offsettel, ami az árammérést eltolja. Annak érdekében, hogy az árammérés pontos legyen, mielőtt mérni kezdünk, ki kell mérni ezt az offsetet, hogy a későbbiekben az árammérést korrigálni tudjuk. Ehhez 1000 ciklus által mért ADC érték átlagát veszem offsetnek.
- INIT POZÍCIÓ mód: Az 5.3. résznek megfelelően a motort be kell forgatni egy meghatározott állapotba, hogy utána az enkóderrel pontosan mérhessük a pozíciót.
- IDENTIFIKÁCIÓ mód: Ebben a módban zajlik a tesztjelek kiadása és a rendszer erre adott válaszainak mérése. A kapott adatokat feldolgozva a motor paraméterei azonosíthatók.
- SZABÁLYOZÁS mód: A sikeres identifikáció után a kapott paraméterértékek ismeretében zajlik a hajtás mezőorientált szabályozása.

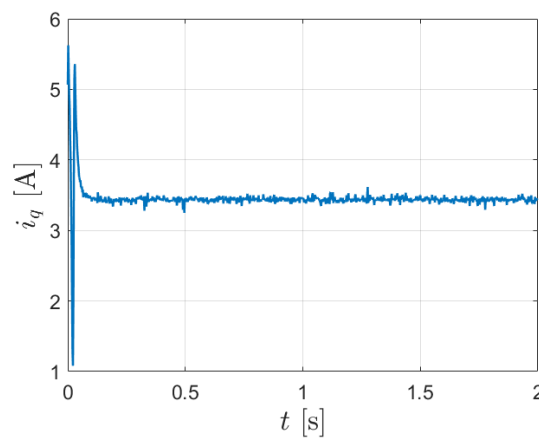
5.5.1. A kezdeti pozíció beállítása

Mivel az enkóder csak relatív elfordulást képes mérni, az identifikáció és a szabályozás előtt be kell forgatni egy nulla pozícióba, ahol a villamos szögelfordulás zérus. Mivel a

használt PMSM $P = 5$ pólussal rendelkezik, öt olyan pozíció van, ami ennek eleget tesz.

A $\varphi = 0$ villamos szög helyzet megegyezik az a fázis által definiált iránnyal, tehát egy a irányba mutató feszültség ráadásával a tengely megfelelő helyzetbe hozható. Hogy ne terheljük túl a motort, a névleges DC feszültséghez képest tizedakkora nagyságú, vagyis $u_a = 2,4$ V feszültséget kapcsolunk az a fázisra. Ez a vivőfrekvenciás impulzusszélesség-moduláció után azt jelenti, hogy a EPWM1-es modul CMPA=3000 referenciát kap, míg a másik két modul CMPA=2500-at.

A motorra folyamatosan ráadva a fenti PWM jelet, a mért áramerősség a következő ábrán látható módon beáll állandósult állapotra. Leolvasható, hogy a gép tengelye már 0,1 s alatt képes beforogni a zérus pozícióba.



31. ábra. Kezdeti pozíció beállításának áramalakja

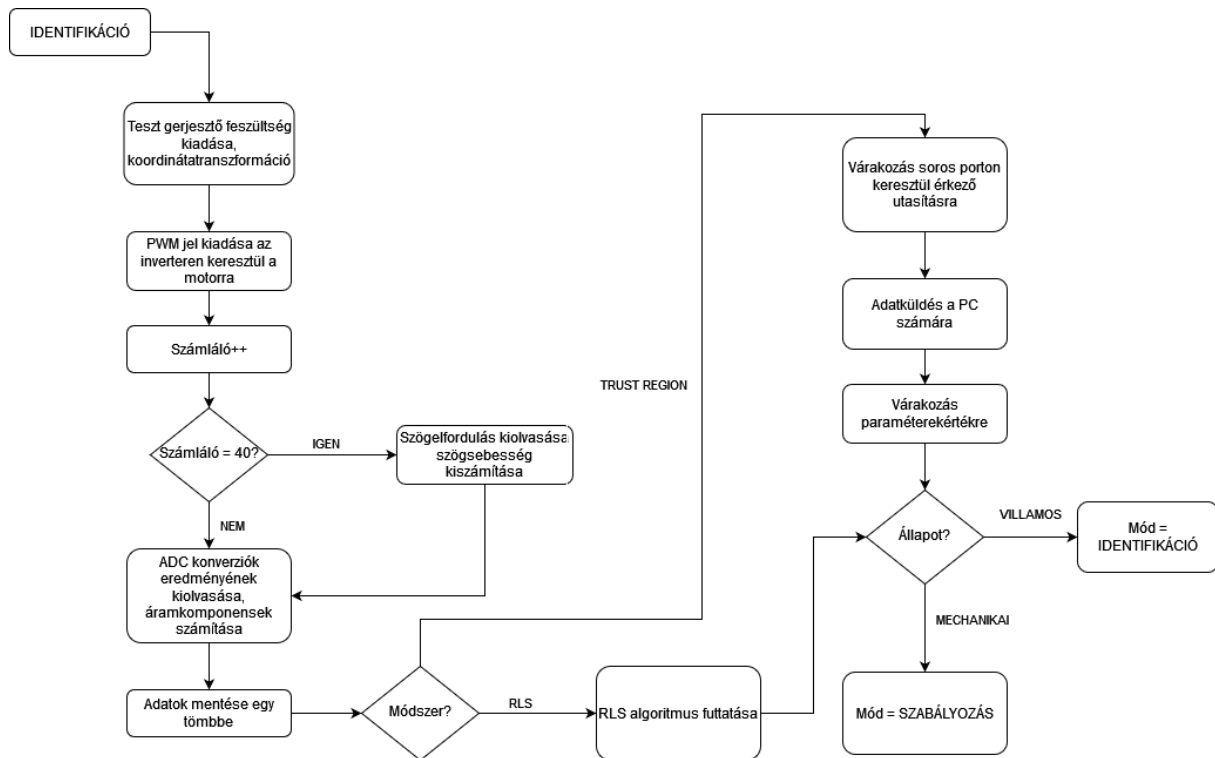
5.6. Az identifikáció elvégzése

Az identifikáció folyamata a 32. ábrán látható. Minden mintavételi ciklusban a tesztelő feszültség ráadásával kezdődik a rutin, amihez először a $d - q$ koordinátarendszerben megadott feszültségeket $x - y$, majd $a - b - c$ koordinátarendszerbe transzformálja a függvény. Ezután valósul meg a vivőfrekvenciás impulzusszélesség-moduláció, ami 0 és 1 közé normalizált referenciafeszültségeket eredményez. Ezek az értékek lesznek az ePWM modulok kimeneteinek kitöltési tényezői.

Miután megtörtént a feszültségek ráadása, kiolvassuk az ADC konverzió eredményét és meghatározzuk belőle a fázisáramokat a 99. egyenlet segítségével. Ezután álló, majd forgó koordinátarendszerbe transzformálva ezeket megkapjuk az i_q és i_d áramkomponenseket.

Minden 40. mintavételi ciklusban kiolvassuk a QPOSCNT regiszter értékét is és ebből kiszámítjuk (95) és (96) segítségével a szükséges szögsebességeket. Így lesz a szögsebességmérés mintavételi frekvenciája $\frac{20[\text{kHz}]}{40} = 0,5$ kHz.

Ezután attól függően, hogy melyik identifikációs eljárást használjuk, vagy elmenti a program az adatokat, vagy lefuttatja az RLS algoritmus egy lépését. Amennyiben a



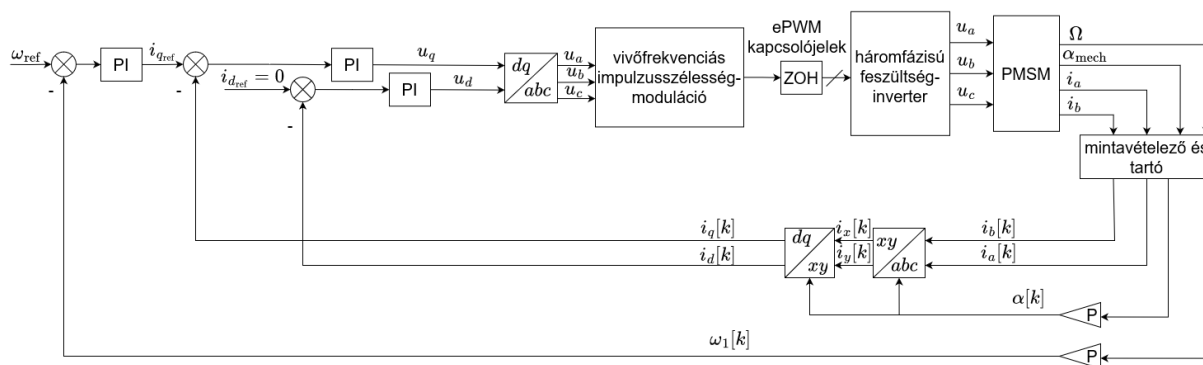
32. ábra. Az i_q áramkomponens mért adatai és a rá illesztett görbe

Trust region optimalizálást szeretnénk elvégezni, a program 1000 adatot ment egy tömbbe, majd a program vár, amíg a PC nem kéri az adatokat. Ekkor UART kommunikáció segítségével elküldi az adatokat, majd vár a külső számítógép válaszára, ami az azonosított paramétereket fogja tartalmazni. Az RLS módszer esetén addig marad azonosítás állapota, amíg a paramétervektor értékei nem konvergálnak. Ha sikeresen megbecsültük a motorparamétereket, a vezérlést végző állapotgép átkerülhet a szabályozás állapotába.

5.7. A zárt hurkú szabályozás megvalósítása

A kapott paraméterek alapján a motor szögsebesség- és áramszabályozása megoldható. Ehhez kaszkád struktúrával rendelkező mezőorientált szabályozást fogok megvalósítani. A 33. ábra mutatja be a szabályozókör felépítését, amit [13] alapján valósítok meg. A szabályozás belső köre az áramot szabályozza, amihez a referencia áramértéket a külső szögsebességszabályozó kör adja. Mivel a mechanikai rendszer tranziensei jóval lassabban zajlanak, mint a villamos tranziensek, a szögsebességmérés történhet kisebb frekvenciával. A mért sebesség, valamint a beállított szögsebességreferencia különbségjeléből PI szabályozóval képezzük a referenciát a q irányú áramkomponens számára. A d irányú áramkomponenset normál üzemi tartományban zérusra szabályozzuk.

A mért és a referencia i_q és i_d áram különbségéből PI szabályozók segítségével képezzük az u_q és u_d feszültségkomponenseket, amiből a bemutatott koordinátatranszformációk és vivőfrekvenciás impulzusszélesség-moduláció végrehajtásával állítható elő a motor szá-

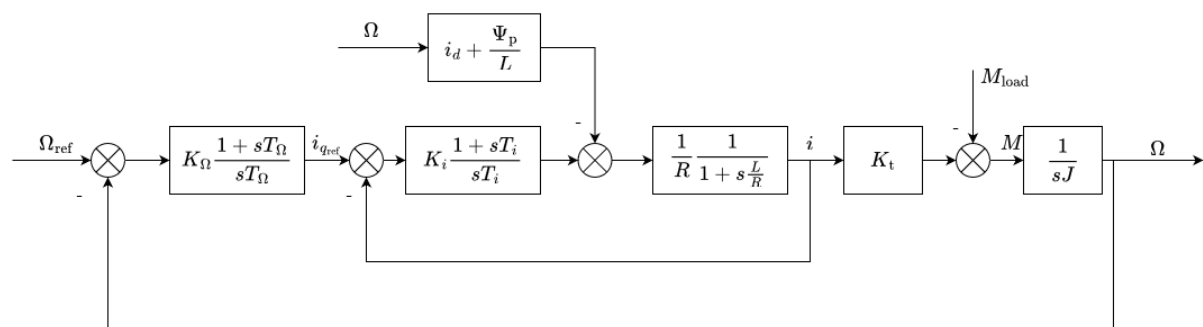


33. ábra. Mezőorientált szabályozókör felépítése

mára a beavatkozójel, amit a DSP PWM perifériája ad ki a gépnek.

Az áram a már korábban látott módon az inverterkártya segítségével mérhető, ebből koordinátatranszformációk után megkaphatók az i_q és i_d áramok. A szögsebességhez enkóder segítségével mérem a szögelfordulást, majd ebből deriválással képezhető a mechanikai szögsebesség, amiből a villamos szögsebesség is számítható.

A szabályozókör hatásvázlatos formában a 34. ábrán látható:



34. ábra. A zárt szabályozókör hatásvázlata

Ez a szabályozókör az i_q áramkomponensre vonatkozik. Az i_d áram szabályozását az ábrán látható áramhurokkal lehet modellezni, úgy, hogy $i_{d_{\text{ref}}} = 0$.

Az áramhurokban a szögsebességtől függő tagokat a szabályozókör szempontjából zavarásnak modellezem, hasonlóan a terhelőnyomaték is zavarásként kerül a rendszerbe. Ezeknek a hatását a szabályozókör kompenzálja.

A PI szabályozókat szaturációval láttam el, hogy az áramszabályozó esetében ne lehessen U_{DC} és $-U_{DC}$ értékénél nagyobb, illetve kisebb feszültséget kiadni beavatkozójelnek, valamint a szögsebességszabályozásból adódó áramreferencia ne lépesse túl a névleges áramot.

5.7.1. Az áramszabályozás hangolása a villamos paraméterek alapján

A kaszkád struktúra belső szabályozóköreinek paraméterei [14] alapján pólus-zéruskiejtés elvégzésével határozhatók meg, amihez

$$T_i = \frac{L}{R} \quad (100)$$

értéket szükséges az integráló tag időállandójának választani. Így az áramszabályozás nyílt hurkának átviteli függvénye a következőképp számítható:

$$W_{i_o} = K_i \frac{1 + s\frac{L}{R}}{s\frac{L}{R}} \frac{1}{R} \frac{1}{1 + s\frac{L}{R}} = K_i \frac{1}{R \cdot s\frac{L}{R}} = \frac{K_i}{L} \frac{1}{s} \quad (101)$$

Ez alapján a zárt szabályozó kör átviteli függvénye

$$W_{i_{cl}} = \frac{W_{i_o}}{1 + W_{i_o}} = \frac{K_i}{Ls + K_i} = \frac{1}{\frac{L}{K_i}s + 1} \quad (102)$$

amiből az egytárolós jellegű kör összidőállandója $T = \frac{L}{K_i}$. Ebből a beállási idő

$$t_{\alpha\%} = T \ln \left(\frac{100}{\alpha} \right) \quad (103)$$

képlettel számítható. Előre meghatározott beállási idő beállításához a megfelelő

$$K_i = \frac{L}{t_{\alpha\%}} \ln \left(\frac{100}{\alpha} \right) \quad (104)$$

érték választható arányos tagnak. Ebből a szabályozó integráló tagja

$$I_i = \frac{K_i}{T_i} = \frac{K_i R}{L} \quad (105)$$

képlettel számítható.

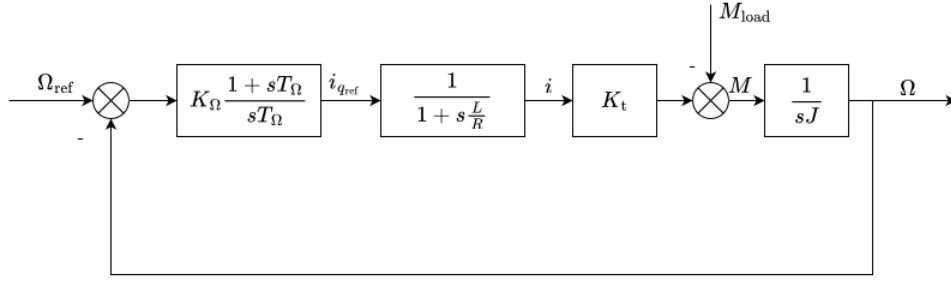
Így a zárt áramkör átviteli függvénye

$$W_{i_{cl}} = \frac{W_{i_o}}{1 + W_{i_o}} = \frac{1}{1 + \frac{L}{K_i}s} \quad (106)$$

ami alapján látható, hogy az eredeti villamos rendszer egytárolós jellege megmaradt, amellett, hogy a beállítás ideje és pontossága az adott alkalmazásnak megfelelően módosítható.

5.7.2. A szögsebességszabályozás hangolása a mechanikai paraméterekből

A fenti hatásvázlatot egyszerűsítsük az áramszabályozó zárt körének átviteli függvénye alapján:

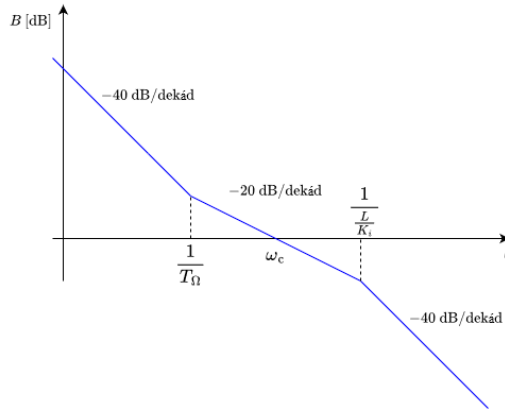


35. ábra. A sebességszabályozás hatásvázlata

Így a felnyitott kör átviteli függvénye:

$$W_{\Omega_o} = K_{\Omega} \frac{1 + sT_{\Omega}}{sT_{\Omega}} \frac{1}{1 + s\frac{L}{K_i}} \frac{K_t}{sJ} \quad (107)$$

Ebből a PI paramétereket szimmetrikus optimumkereső eljárással hangolhatjuk. Ehhez a nyílt kör amplitúdódiagramja a következő ábrán látható:



36. ábra. A felnyitott kör amplitúdódiagramja

A $T_{\Omega} = 10 \frac{L}{K_i}$ választással iktatható be a két -40 dB/dekádos közé a -20 dB/dekádos meredekségű szakasz. A vágási körfrekvenciát úgy állítom be, hogy a -20 dB/dekád meredekségű szakasz közepére essen, vagyis:

$$\omega_c = \frac{1}{\sqrt{T_{\Omega} \frac{L}{K_i}}} \quad (108)$$

Ehhez az arányos tag

$$K_{\Omega} = \frac{\frac{J}{K_t}}{\sqrt{T_{\Omega} \frac{L}{K_i}}} \quad (109)$$

beállítása szükséges, ezzel elérve, hogy a $\frac{K_t}{sJ}$ integráló tag $\frac{J}{K_t}$ vágási körfrekvenciája megfelelő helyre essen. Ez alapján a szögsebességszabályozás integráló tagjának együtthatója:

$$I_{\Omega} = \frac{K_{\Omega}}{T_{\Omega}} = \frac{\frac{J}{K_t}}{10 \frac{L}{K_i}} = \sqrt{10} \frac{J}{K_t} \quad (110)$$

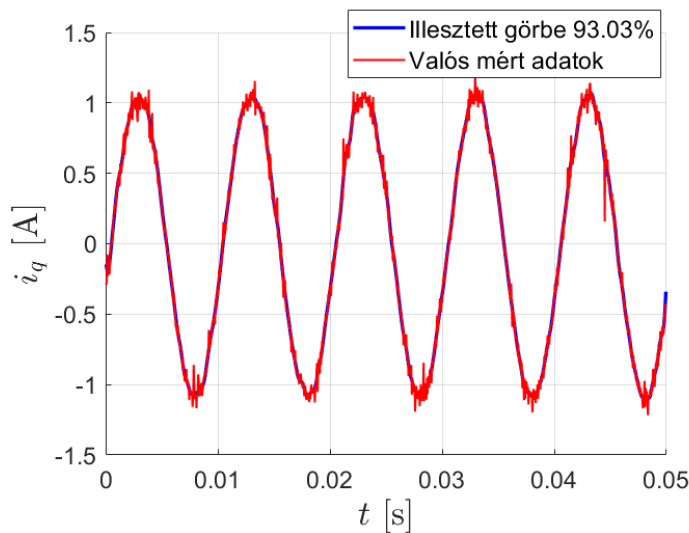
6. A valós mérés eredményei

6.1. *Trust Region* algoritmus

Valós körülmények között tesztelve a javasolt paraméterbecslés folyamatát, először a *Trust Region* algoritmus segítségével identifikálom az ellenállás, az induktivitás és a pólusfluxus értékét harmonikus u_q bemenő feszültséget adva a rendszerre.

A pólusfluxus becsléséhez a 20 kHz-es mintavételi frekvenciával mért i_q és i_d áramok mellett a szögsebességet is mérem 0,5 kHz-el. 4.4 alapján a szögsebességet az 50 μ s-os időlépés eléréséhez lineáris interpolációval közelítem azokon a helyeken, ahova nem tartozik minta.

Ekkor a mért i_q áramkomponens alakja az ábrán látható módon alakul, az optimalizáció által illesztett görbe 93,03%-os egyezést mutat a valós adatokkal.



37. ábra. Az i_q áramkomponens mért adatai és a rá illesztett görbe

Ekkor az ellenállás

$$R = 0,8865 \, \Omega \quad (111)$$

értéknek adódik. Az induktivitás pedig

$$L = 2,8533 \cdot 10^{-4} \, \text{H} \quad (112)$$

értékű. Látható, hogy az induktivitás értéke nem áll távol a várt $L = 3,2 \cdot 10^{-4} \, \text{H}$ értéktől, viszont az ellenállás nagyjából háromszoros nagyságú a motorkatalógusban szereplő $R = 0,285 \, \Omega$ -s ellenállásához képest.

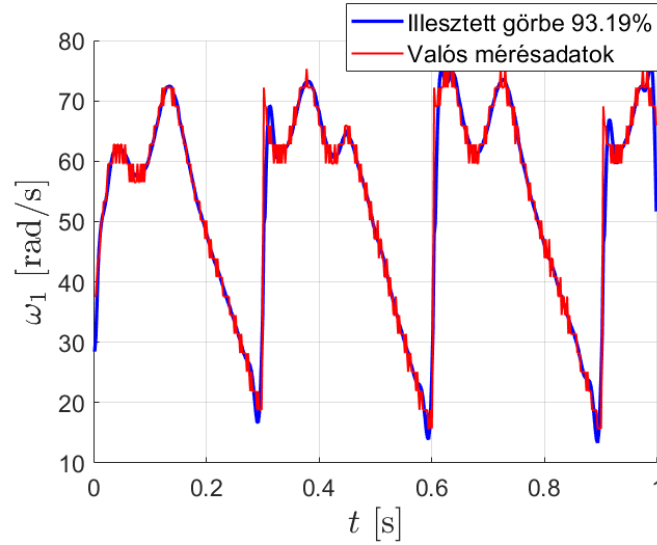
A pólusfluxus

$$\Psi_p = 6,8605 \cdot 10^{-4} \, \text{Wb} \quad (113)$$

értéknek adódik. Ebből a nyomatékonstans a katalógusnak hozzávetőlegesen megfelelő

$$K_t = 0,0051 \, \text{Nm/A} \quad (114)$$

A mechanikai jellemzők meghatározásához négyszögjel bemenetet használva a szögsebesség időfüggvénye és az arra illesztett görbe, 0,5 kHz-es mintavételi frekvencia mellett:



38. ábra. Az ω_1 villamos szögsebesség mért adatai és a rá illesztett görbe

Ekkor a tehetetlenségi nyomaték

$$J = 5,7130 \cdot 10^{-4} \text{ kgm}^2 \quad (115)$$

értéknek adódik, ami a katalógusban megadott $1,77 \cdot 10^{-5} \text{ kgm}^2$ értéknél valamivel nagyobb. A kapott viszkózus csillapítás

$$b = 0,0040 \text{ Ns/m} \quad (116)$$

Ez az érték nincs megadva a katalógusban, viszont a hasonló méretű motorok jellemző értékeihez képest nagyságrendileg reális a kapott eredmény.

6.2. RLS módszer

Az RLS algoritmust i_q kimenőjel és u_q bemenőjel esetén, minden új minta megérkezésekor elvégezve a (46)-(49) lépéseket, a paramétervektor Θ elemeire a következő értékek adódnak:

$$a_1 = 0,35045 \text{ és } b_1 = 0,60874 \quad (117)$$

adódnak. Ezekből

$$R = 1,06704 \text{ } \Omega \quad (118)$$

$$L = 2,8785 \cdot 10^{-5} \text{ H} \quad (119)$$

villamos paraméterek számíthatók.

A mechanikai jellemzők méréséhez a (57)-ből kiindulva a Θ paramétervektor együtthatói:

$$d_1 = 0,95854 \text{ és } e_1 = 40,06875 \quad (120)$$

amiből

$$J = 1,4725 \cdot 10^{-5} \text{ kgm}^2 \quad (121)$$

és

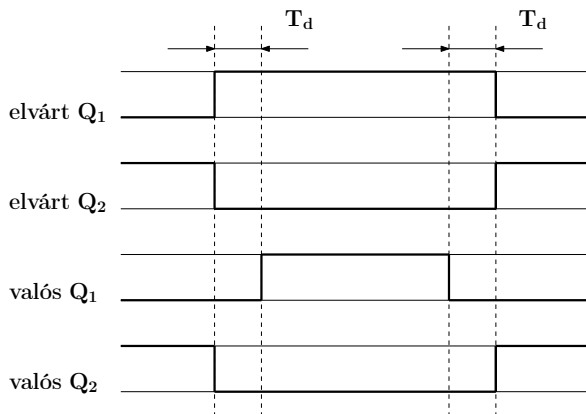
$$b = 3,0524 \cdot 10^{-4} \text{ Ns/m} \quad (122)$$

értékek számíthatók.

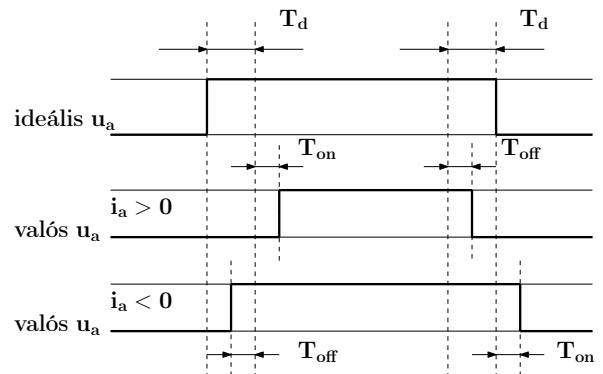
Az ellenállásértéket leszámítva ebben az esetben is reális értékeket kaptunk.

6.3. A holtidő kompenzációja

Az előzőekben látott mérés során az i_q áram alakja jelentősen eltér a szimulációs, ideálisnak tekinthető, grafikonoktól. Ennek egyik oka a PWM kimeneti jelei között beállított holtidő, amivel a 5.2. részben foglalkoztam. A holtidő bevezetése után az EPWM kimeneti jele a következőképpen változik meg:



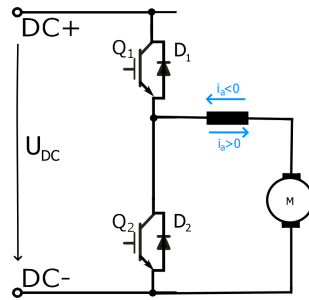
(a) Az EPWM periféria kimeneti jelei, a kapcsolójelek a T_d holtidővel



(b) Az a fázis feszültsége ideális és holtidős esetben, figyelembe véve a tranzisztorok be- és kikapcsolási idejét

39. ábra

Így olyan időtartam lép fel, amikor egyik tranzisztor sem vezet. Vizsgáljuk meg ezt a jelenséget a 40. ábra alapján egy fázis esetén, az áram irányától függően.

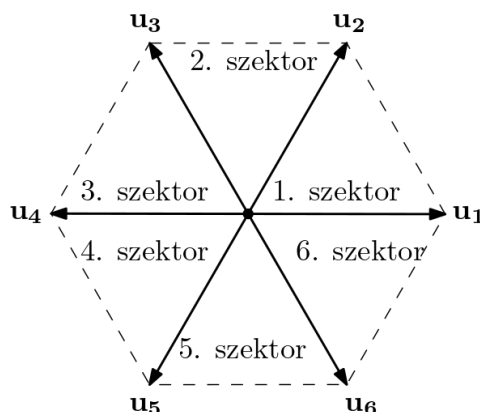


40. ábra. Az inverter egy fázisának áramköre

Abban az esetben, amikor az a fázis árama pozitív, eredetileg a Q_1 van bekapcsolva, a Q_2 kikapcsolva, az áram az inverter felől folyik át a motoron, így az motoros üzemben van. Ekkor ha nyitjuk Q_1 -et is, az áram iránya a motor induktivitása miatt nem változhat azonnal, így az áram a D_2 diódán keresztül továbbra is fenntartja az áramot. Így a motorra negatív feszültség esik, vagyis a pozitív feszültség tényleges vezetési ideje csökken, $T_d + T_{on} - T_{off}$ idővel.

Ha kezdetben negatív áram folyik, vagyis a Q_2 kapu van zárva alapesetben, a Q_1 nyitva, a motor generátoros üzemben van, vagyis a motor felől folyik az inverter felé az áram. Ha Q_2 -t is kikapcsoljuk, akkor az áramot a felsőoldali dióda vezeti tovább, a motoron pozitív feszültség esik. Így az pozitív feszültség valós vezetési ideje $T_d + T_{on} - T_{off}$ időtartammal nő.

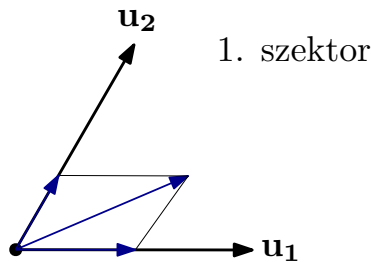
Ezt a két hatást szükséges kompenzálni a PWM kitöltési tényezőjének módosításával. Mivel a kompenzáció függ az áram irányától, ezért három fázisra $2^3 = 8$ esetet kell vizsgálni. Ehhez érdemes térvektoros formában felírni az összefüggéseket. Az inverter 8 állapotához 8 feszültségvektor tartozik, amiből 6 pozitív nagyságú vektor és 2 nullvektor. A térvektorok a síkot szektorokra osztják, a 41. ábrán látható módon.



41. ábra. Az inverter által kiadott háromfázisú feszültség térvektoros formában

A motorra kiadandó feszültség megadható annak a két térvektornak súlyozott összegeként, amik határolják a szektort, ahova a feszültség esik. Így amikor a motorra adandó

feszültségek közül az a fázis feszültsége pozitív, a b és a c negatív, akkor ez a feszültségvektor az első szektorban helyezkedik el. Ekkor a feszültségvektor a 42. ábrán alapján az $\mathbf{u}_1$ és $\mathbf{u}_2$ térvektorok súlyozott összegeként adódik. Így az adott mintavételi idő alatt bizonyos ideig az egyik, majd a másik térvektor kapcsolható a motorra, a maradék időben pedig valamelyik nullvektor van bekapcsolva. Így ezzel megadtuk az inverter vezérlését, ha adott első szektorbeli feszültséget szeretnénk kiadni a motorra.



42. ábra. Első szektorbeli feszültségvektor

Ha az első szektorban vizsgáljuk a PWM periféria holtidejének hatását, akkor megállapítható, hogy a b és c fázisáramok negatív irányúak, így mindkét fázisnak a pozitív feszültsége tart tovább a holtidőben. Így a kiadandó $\mathbf{u}_2$ térvektor ideje megegyezik a ténylegesen a motorra jutó $\mathbf{u}_2$ idejével ebben a szektorban. Az a fázis árama pozitív, így az a irányú motorra jutó pozitív feszültség ideje, és ezáltal a $\mathbf{u}_1$ térvektor kiadásának ideje, rövidül $\Delta T = 2(T_d + T_{on} - T_{off})$ -vel. Vagyis a valós térvektor $\Delta T \mathbf{u}_1$ -el kisebb lesz, mint az elvárt. Ezért a PWM kitöltési tényezőjét $\Delta T \mathbf{u}_1$ -nek megfelelően kell növelni.

A következő táblázat tartalmazza [10] alapján a lehetséges esetekben a kompenzálendő térvektor értékét:

Szektor	i_a	i_b	i_c	Szükséges kompenzáció
1	> 0	< 0	< 0	$\Delta T \mathbf{u}_1$
2	> 0	> 0	< 0	$\Delta T \mathbf{u}_2$
3	< 0	> 0	< 0	$\Delta T \mathbf{u}_3$
4	< 0	> 0	> 0	$\Delta T \mathbf{u}_4$
5	< 0	< 0	> 0	$\Delta T \mathbf{u}_5$
6	> 0	< 0	> 0	$\Delta T \mathbf{u}_6$

4. táblázat. A szükséges kompenzációk az egyes szektorokban

A kompenzálás megvalósításához ismerni kell a $\Delta T = 2(T_d + T_{on} - T_{off})$ értékét, ahol T_d a PWM periféria RED és FED regisztereiben beállított holtidő érték, esetünkben mérés is igazolva, $0,5 \mu s$. A tranzisztorok be- és kikapcsolási idejei azoknak katalógusából maximálisan 50 ns -osak, az egyik kikapcsolása és a másik bekapcsolása között eltelt idő

$t_{\text{on}} - t_{\text{off}} = 8 \text{ ns}$ a katalógus szerint[9]. Így $\Delta T = 2(T_d + T_{\text{on}} - T_{\text{off}}) = 2(0,5 \cdot 10^3 + 8) = 508 \text{ ns}$. Ezt az időt a periféria beállításainak megfelelően skálázva kell növelni a PWM jelek kitöltési tényezőjét az egyes esetekben.

A kompenzálás megvalósításához a programban szükség van a fázisáramok előjelvizsgálatára. Az identifikáció során csak az a és b komponenseket mérjük, mivel az i_c ezekből meghatározható:

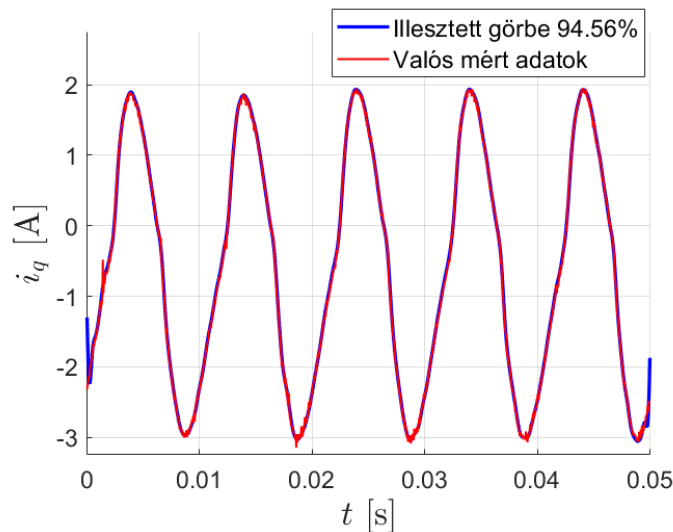
$$i_c = -i_a - i_b \quad (123)$$

Mivel a megvalósításban vivőfrekvenciás impulzusszélesség-modulációt használunk, a térvektoros formából vissza kell térni az egyes fázisokhoz. Így az $\mathbf{u}_1$ térvektor egybeesik az u_a komponenssel, így ekkor csak az a fázis PWM jelének kitöltési tényezőjét kell növelni. Az $\mathbf{u}_2$ térvektor az u_a és u_b komponensek összege, így ekkor az a és b fázis impulzusszélességét kell növelni. Ez alapján mindegyik áramkombináció esetén módosíthatók a kitöltési tényezők.

Az eddigiek alkalmazásával a 32. ábrán látható folyamatára úgy módosul, hogy minden megszakítás esetén kiszámítódik az i_c áramkomponens, majd a moduláció után a kitöltési tényezők az áramkomponensek előjelének megfelelően módosításra kerülnek.

6.3.1. A *Trust Region* optimalizálás eredményei a holtidőkompenzáció után

Az előzőekben leírt módosításokkal elvégezve az identifikációt a kompenzáció után az ellenállás és induktivitás identifikációjához mért q irányú áramkomponens és az arra *Trust Region* optimalizálással illesztett görbe a következő ábrán látható.



43. ábra. Az i_q áramkomponens mért adatai és a rá illesztett görbe

Ekkor az ellenállás

$$R = 0,2790 \, \Omega \quad (124)$$

míg az induktivitás

$$L = 4,7680 \cdot 10^{-4} \text{ H.} \quad (125)$$

értéknek adódik. A pólusfluxus

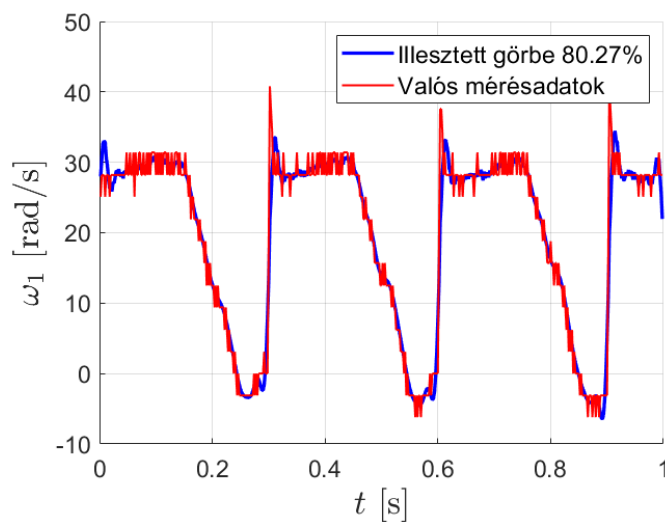
$$\Psi_p = 0,0057 \text{ Wb} \quad (126)$$

értékű, az ebből számított nyomatékonstans

$$K_t = 0,0428 \cdot \text{Nm/A} \quad (127)$$

Ezzel az ellenállás értéke is nagy pontossággal meghatározható lett.

Vizsgáljuk meg a mechanikai paraméterek becslését is holtidő kompenzáció elvégzésével.



44. ábra. Az ω_1 villamos szögsebesség mért adatai és a rá illesztett görbe

Ekkor a tehetetlenségi nyomaték

$$J = 1,4622 \cdot 10^{-4} \text{ kgm}^2 \quad (128)$$

értéknek adódik, míg a viszkózus csillapítás

$$b = 0,0012 \text{ Ns/m} \quad (129)$$

A tehetetlenségi nyomaték pontosan közelíti a katalógus alapján elvárt értéket és a viszkózus csillapításra is nagyságrendileg helyes értéket kaptunk.

6.3.2. Az RLS optimalizálás eredményei a holtidőkompenzáció után

Az RLS algoritmus esetén

$$R = 0,4995 \Omega \quad (130)$$

$$L = 4,8785 \cdot 10^{-4} \text{ H} \quad (131)$$

villamos paraméterek adódnak. A mechanikai paraméterekre 57. alapján

$$J = 3,5832 \cdot 10^{-5} \text{ kgm}^2 \quad (132)$$

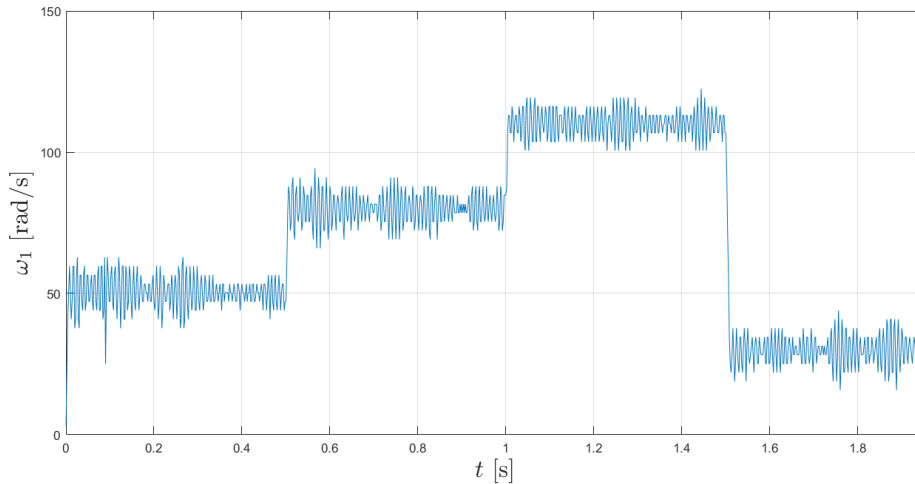
$$b = 5,4821 \cdot 10^{-4} \text{ Ns/m} \quad (133)$$

értékeket kaptam.

A holtidő kompenzálásával kapott paraméterértékek jó közelítéssel megfelelnek a valós adatoknak, így ezekre alapozva a szabályozás már elvégezhető.

6.3.3. A paraméterek alapján történő szögsebességszabályozás

A 5.7. fejezetben látottak alapján hangolhatók a kaszkád szabályozás áram- és szögsebesség-hurkának szabályozóparaméterei. Az ez alapján végrehajtott szögsebességszabályozás beállási függvénye a 45. ábrán látható. A vizsgált időtartamban a beállított szögsebesség referencia kezdetben $\omega_{\text{ref}} = 50 \text{ rad/s}$, majd 80 rad/s , ezt követően 110 rad/s , végül 30 rad/s .



45. ábra. A zárt szabályozókör válasza egymást követő konstans szögsebességreferenciákra

A szögsebesség beállása közelítőleg egytárolós jellegű lesz a mezőorientált szabályozás hatására. A szögsebességválaszok beállás idejei 10 ms körüli értékek.

7. Konklúzió

A dolgozatban többféle identifikációs módszert bemutattam a PMSM hajtások villamos és mechanikai paramétereinek becslésére. A használt módszerek közül a legprecízebben a *Trust Region* optimalizáció írta le a motor működését, viszont ennek a legnagyobb a számítási igénye is. A rekurzív legkisebb négyzetek algoritmus használata bár pontatlanabbnak bizonyult, kis erőforrásigénye miatt mégis indokolt lehet.

A szimulációs eredmények alapján kétféle identifikációs folyamatot dolgoztam ki. A pontosabb paraméterértékeket eredményező *Trust Region* algoritmust alkalmazva külső számítógép dolgozza fel a mérési adatokat és végzi az identifikációt. A villamos paraméterek meghatározásához a motor áramkomponenseit és szögsebességét mérve keresem a kapcsolatot az i_q áramkomponens és a bemeneti u_q feszültség, illetve a villamos szögsebesség között. Ezt az összefüggést egy előre meghatározott nemlineáris formában felírva az algoritmus képes megtalálni a kapcsolatot leíró együtthatókat, amik alapján az armatúra ellenállás, induktivitás és a rotor pólusfluxusa, valamint a nyomatékkonstans számítható.

Az RLS módszer esetén a villamos paraméterek azonosításkor csak a bemeneti feszültség, valamint a kimeneti i_q áram mért értéke használható, így a kapott ellenállás és induktivitásértékek pontatlanabbak, de közelítőleg helyesek, valamint a pólusfluxus értéke nem identifikálható. Alkalmazásának előnye, hogy a folyamat elvégezhető kizárólag a jelfeldolgozó processzoron.

A folyamat következő fázisa minden esetben a mechanikai paraméterek becslése. Ekkor mind *Trust Region* optimalizáció, mind RLS algoritmus esetén a villamos szögsebesség időfüggvényét állítom elő a q irányú áramkomponens alapján. Az összefüggést leíró együtthatókból ezúttal a motor tehetetlenségi nyomatéka és viszkózus csillapítási tényezője adódik.

A kidolgozott eljárásokat megvalósítottam valós mérési összeállításon is. Az identifikált motorparaméterek nem minden esetben álltak közel a katalógusban megadott értékekhez, így szükségessé vált az impulzusszélesség-moduláció kapcsolójeleiből adódó holtidő hatásának kompenzálása. A kompenzáció alkalmazásával a kapott görbék az ideális, szimulációs eredményeket közelítik és így az identifikációs eredmények jelentős mértékben pontosabbá válnak.

A *Trust Region* optimalizációs eredmények a szimulációs vizsgálatok alapján elvárt módon jóval precízebbnek bizonyultak az RLS algoritmus konvergált paramétereinél. Ez utóbbi esetében is nagyságrendileg helyes értékek adódtak, ezért a motor zárt hurkú szabályozása mindkét esetben megoldható volt.

Az alkalmazott kaszkád szabályozás belső áramszabályozóköre a villamos paraméterek, a külső szögsebességszabályozó kör a mechanikai paraméterek ismeretében hangolható. Kipróbálva a szabályozás beállítását az identifikációval kapott motorparaméterekkel, a szabályozott rendszer szögsebessége gyors és precíz beállást produkált.

Köszönetnyilvánítás

Szeretném megköszönni konzulenseimnek, Dr. Stumpf Péter Pálnak és Dr. Budai Csabának, a rengeteg segítséget, tanácsot, a folyamatos útmutatást, amiért bármikor fordulhatam hozzájuk, és nem utolsó sorban nagyon hálás vagyok az irántam tanúsított végtelen türelmükért.

Hivatkozások

- [1] Lantos Béla. *Irányítási rendszerek elmélete és tervezése I.* Akadémiai Kiadó, 2016. ISBN: 978 963 05 9848 4. DOI: 10.1556/9789630598484. URL: <https://mersz.hu/kiadvany/31>.
- [2] *BOOSTXL-3PhGaNInv Schematic*. Texas Instruments, TI Literature Number:SLURAY0A.
- [3] *BOOSTXL-3PhGaNInv User's Guide*. Texas Instruments, TI Literature Number: SLUUBP1A, June 2017 - Revised April 2018.
- [4] Katalin Hangos és Gábor Szederkényi. *Dinamikus rendszerek paramétereinek becslése*. Veszprém: Veszprémi Egyetemi K., 1999, 99. old. URL: <https://eprints.sztaki.hu/1877/>.
- [5] Monson H. Hayes. *Statistical Digital Signal Processing and Modeling*. 1996.
- [6] A. Házy. *Nemlineáris optimalizálás*. Miskolci Egyetem, 2014. URL: <https://gyires.inf.unideb.hu/KMITT/b14/>.
- [7] *INA240 Datasheet*. Texas Instruments, TI Literature Number: SBOS662B, July 2016 - Revised February 2018.
- [8] Lennart Ljung. "Prediction error estimation methods". *Circuits, Systems, and Signal Processing* 21 (2002. jan.), 11–21. old. DOI: 10.1007/BF01211648.
- [9] *LMG5200 Datasheet*. Texas Instruments, TI Literature Number: SNOSCY4E, March 2015 - Revised October 2018.
- [10] J. Heide M. Granström. "Simulation of a Current Controller with Dead-Time Compensation". (2021). URL: <https://liu.diva-portal.org/smash/get/diva2:1562448/FULLTEXT01.pdf>.
- [11] T. Nagy. *Feltétel nélküli optimalizálás algoritmusai*. Miskolci Egyetem, Alkalmazott Matematikai Tanszék, 2012.
- [12] K. Pelckman. *Prediction Error Methods*. Uppsala University, Department of Information Technology, 2012.
- [13] P. Stumpf. *Állandómágneses gépek*. Budapesti Műszaki- és Gazdaságtudományi Egyetem, Automatizálási és Alkalmazott Informatikai Tanszék, 2019.

- [14] P. Stumpf. *Szabályozott egyenáramú hajtás*. Budapesti Műszaki- és Gazdaságtudományi Egyetem, Automatizálási és Alkalmazott Informatikai Tanszék, 2019.
- [15] P. Stumpf. *Teljesítményelektronika és villamos hajtások (BMEVIAUM036): DC/AC átalakítók I. Kétszintű feszültséginverterek*. Budapesti Műszaki- és Gazdaságtudományi Egyetem, Automatizálási és Alkalmazott Informatikai Tanszék, 2019.
- [16] Guoyong Su és tsai. “Multiparameter Identification of Permanent Magnet Synchronous Motor Based on Model Reference Adaptive System, Simulated Annealing Particle Swarm Optimization Algorithm”. *Electronics* 11.1 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11010159. URL: <https://www.mdpi.com/2079-9292/11/1/159>.
- [17] *TMS320F2837xD Dual-Core Delfino Microcontrollers Technical Reference Manual*. Texas Instruments, TI Literature Number: SPRUHM8H, December 2013 - Revised January 2019.
- [18] Simon Wiedemann és Ralph M. Kennel. “Self-Commissioning of the Current Control Loop in AC Drives”. *PCIM Europe 2018; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*. 2018, 1–8. old.
- [19] W. Ye. *Trust-region methods*. Northwestern University, 2014. URL: https://optimization.mccormick.northwestern.edu/index.php/Trust-region_methods.