

Budapesti Műszaki és Gazdaságtudományi Egyetem
Közlekedésmérnöki és Járműmérnöki Kar
Közlekedés- és Járműirányítási Tanszék

Modelljármű kormány szervójának áramköri- és
irányítástervezése

Készítette:
Baranyi Márk

Konzulens:
Bakos Ádám
Dr. Soumelidis Alexandros
Aradi Szilárd

Tudományos Diákköri Konferencia
Budapest, 2014
Utolsó módosítás dátuma: 2015.01.06.

Nyilatkozat önálló munkáról, hivatkozások átvételéről

Alulírott Baranyi Márk kijelentem, hogy a TDK dolgozatom magam készítettem és abban a feltüntetett forrásokat használtam fel. Minden olyan részt, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból vettem, egyértelműen, a forrás megadásával megjelöltem.

Budapest, 2014. november

Baranyi Márk

Köszönetnyilvánítás

Ezúton szeretném megköszönni Dr. Soumelidis Alexandros és Bakos Ádám tanár uraknak, hogy lehetőséget biztosítottak az MTA SZTAKI kutatólaboratóriumban végzett fejlesztésre.

Köszönöm az eddigi támogatást és fáradozást, ami a dolgozat teljesítéséhez kellett.

Továbbá köszönöm Aradi Szilárd tanár úrnak az együttműködést és a segítséget, amivel támogatta a munkát és helyet biztosított a BME Közlekedési és Járműirányítási tanszéken a mérésekhez, tesztekhez.

Összefoglaló

A Tudományos Diákköri Konferencia dolgozat témája a Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutatóintézetében található 1:5 méretarányú agymotoros modellautó kormány szervójának elemzéséről, specifikációjáról, mechanikai átalakításáról, NYÁK-tervezéséről, és végül, de nem utolsósorban a hardveresen kialakított környezet alapján a program, azaz a megtervezett és modellezett irányítás realizációjáról szól.

A laboratóriumban végzett munka célja egy már korábban megkezdett és alapjaiban megtervezett agymotoros hajtású jármű lehetőségeinek a megismerése és továbbfejlesztése egy lehetséges szabályozás megtervezésével.

A dolgozat során ismertetni fogom a felhasznált eszközöket, illetve a jármű szervójával szembeni követelményeket is. A minőségi kritériumokat az intézet határozta meg. A megtervezett kormány szervó segítségével megvalósulhat egy felsőbb szintű irányítás megtervezése is, ami a végső célok egyike és ennek egy modell szintű előtervezését is szeretném bemutatni.

Tartalomjegyzék

1.	Bevezetés.....	6
1.1	Motiváció.....	6
1.2	A probléma leírása.....	6
1.3	Célkitűzés.....	6
2.	Meglévő szervó bemutatása és a feladat ismertetése.....	8
2.1	A digitális szervó tulajdonságai	8
2.2	A feladat részletezése.....	8
3.	Szervó mechanikai, hardveres módosításai	9
3.1	Fizikai módosítások	9
3.2	Szabályozáshoz szükséges módosítások.....	12
3.3	A NYÁK-tervezés alaplépései	13
3.4	Vezérlési modellhez használt alapkapcsolás	15
3.5	A NYÁK-tervezése	17
4.	A szervó felkészítése az irányítás megtervezéséhez	19
4.1	Élesztés, tesztelés	19
4.2	Az kontroller programozása	19
4.3	Szögjeladó.....	20
4.4	CAN kommunikáció	20
4.5	Analóg mérések	21
4.6	A fordulatszám mérése	22
4.7	A vezérlési modell megvalósítása.....	22
5.	A DC motor irányítási lehetőségeinek ismertetése	23
5.1	A fekete doboz módszer.....	23
5.2	Az irányítás megtervezése egytárolós rendszert feltételezve.....	24
5.3	Motorszabályozás tervezése a DC motor dinamikus modellje alapján.....	25
5.4	Robusztus irányítás tervezés	27
6.	Rendszer identifikációs lépések az egytárolós átviteli függvény meghatározásához és az irányítás megtervezése	27
6.1	Az identifikációhoz szükséges eszközök ismertetése	27
6.2	A rendszer átviteli függvényének meghatározása	30
6.3	A zárt rendszer modellezése és tesztelése.....	33
6.4	Pozíció szabályozás modellezése és tesztelése	36
7.	A modell továbbfejlesztése és a lineáris kvadratikus szabályozás modellezése	40

7.1	A motor dinamikai modelljének meghatározása	40
7.2	Áttérés állapotterez felírásra.....	43
7.3	LQR irányítástervezés	45
8.	Eredmények összegzése	49
8.1	Kitekintés	49
8.2	Összefoglalás	50
9.	Ábrajegyzék	50
10.	Irodalomjegyzék	51
11.	Melléklet.....	52

1. Bevezetés

A dolgozat a kutatóintézetben fellelhető modellautó kormányzásáért felelős aktuátor elemzéséről, specifikációjáról, módosításáról, modelljének identifikációról és az alapján egy irányítás megtervezéséről és teszteléséről szól. Lényeges azt kiemelni, hogy a szervó központi eleme a szabályozó elektronika, amit újratervezünk. A dolgozat alapvetően e tervezési feladat köré épül.

1.1 Motiváció

Manapság az egész járműipar területén fontos kérdések közé tartozik az emisszió csökkentése és a hatékonyság növelése. Ezen belül is kiemelkedő jelentőségű a hajtásrendszer, erőgépek környezetre gyakorolt hatásuk csökkentése. Maga a hajtáslánc jelentős súlytöbbletet ad a gépjárműnek, nem beszélve a belső égésű motorok hatásfokáról. Ezzel szemben a villamos hajtású gépeket magas hatásfok tartományban tudjuk működtetni. Ha veszünk négy darab motort, és elhelyezzük megfelelően a kerék agyába, akkor egy olyan hajtásblokkot kapunk, ahol a négy kerék külön-külön kapja meg a hajtónyomatékot, ráadásul hajtáslánc sem kell hozzá. Ezzel csökkenthető a jármű össztömege, növelhető a helykihasználás az autó terén belül (például nagyobb méretű akkumulátor helyezhető el), illetve a közvetlen káros anyag kibocsátás gyakorlatilag a nullával egyenlő.

1.2 A probléma leírása

Jelen esetünkben egy 1:5 méretarányú BAJA 5B típusú, a kutatóintézet által beépített agymotorokkal és egy kereskedelmi forgalomban is kapható szervó egységgel rendelkező modellautó áll a rendelkezésünkre. A szervó problémája az, hogy nem szabályozható megfelelően és diagnosztikai támogatottságot sem tudunk vele elérni egyszerűsége miatt.

1.3 Célkitűzés

A dolgozat megírásával a végső cél az, hogy egy kommunikációs csatornával (CAN) ellátott, s azon kapott referencia jel alapján a jármű gyorsan, stabilan és pontosan kormányozható legyen, és az esetleges hibákat detektálni tudjuk a működés során. Tehát pontos pozíciószabályozást szeretnénk végrehajtani. Ezzel gyakorlatilag az egész rendszer CAN

kommunikációval kapcsolódna a központi ECU-hoz, ami jelentősen megkönnyítené a felsőszintű irányítás realizációját a későbbiekben.

Ez voltaképpen azt jelenti, hogy egy adott DC motorhoz kell egy komplex áramköri tervezést megvalósítani. Nemcsak egy NYÁK- tervezésről van szó ugyanis, hanem egy szabályzó elektronika irányítás célú megvalósításáról.

2. Meglévő szervó bemutatása és a feladat ismertetése

2.1 A digitális szervó tulajdonságai

A feladathoz kapott szervó egy XQ-Power s5040d típusú, kereskedelmi forgalomba is kapható eszköz.

Alapfunkciói a következők:

- magasfeszültségen is működtethető,
- titánium ötvözetű fogaskerekek,
- hőálló és korrózióálló,
- merev alumínium ház,
- DC motort tartalmaz.

Specifikus leírás(felsorolás szinten):

- Irányítás: PWM jellel
- Hőmérséklet tartomány: $-25\text{ }^{\circ}\text{C} \sim +70\text{ }^{\circ}\text{C}$
- Csapágyazás: kétsoros golyóscsapágyak
- Programozható: Nem
- Feszültségtartomány: 6V-7.2V
- 6V-on a fordulatszám: 1.3 1/sec (7.2V-on 1.5 1/sec)
- 6V-on a maximális nyomaték: 3.285 Nm (7.2V-on 3.9 Nm)
- Méret: 60.1*30.1*60 mm
- Súly: 177.5g

2.2 A feladat részletezése

Ez az eszköz otthoni céloknak kiválóan megfelel, viszont semmi információt nem ad számunkra a szervó állapotáról, beállási időről, egyedül egy potenciométer által szolgáltatott szöghelyzetet tudunk kinyerni belőle. Ez viszont nem elegendő egy finoman hangolt irányítás megtervezéséhez. Az intézet számára egy olyan kormányzás szükséges, ahol szükség szerint az eszköz átprogramozható és diagnosztikai adatokat is szolgáltat a fejlesztés érdekében, illetve képes referencijeleket fogadni a központi feldolgozó egység felől.



1.ábra: Az áttervezendő szervó kezdeti állapota(Forrás: xq-power.com)

Ahhoz, hogy az eszköz alkalmazható legyen, drasztikus átalakításokat kell rajta végezni mind mechanikai, mind elektrotechnikai szempontból is. A benne lévő elektronika helyett egy teljesen új, akár több emeletes NYÁK-ot kell tervezni és ehhez mechanikai átalakításokat is kell végezni.

A tervezett vezérlő a hardveres specifikációi a következők:

- 1) Tudjon a 6,4-8,4V feszültségtartományban lévő tápfeszültségről működni.
- 2) Képes legyen a CAN kommunikációra.
- 3) Valós idejű szabályozás megvalósítható legyen. (Pozíció szabályozás!)
- 4) 32 bites, lebegőpontos számolásokat is tudjon végezni.
- 5) Minden állapot mérve legyen:
 - Tápfeszültség
 - Motor áram (~nyomaték) VIR
 - Back EMF (Electromotive force) mérése (fordulatszámérés)
 - Szöghelyzet mérése
- 6) Fordulatszám, szöghelyzet szabályozható legyen!

Ezek közül a mechanikai, hardveres átalakításokat fogom a következőkben ismertetni.

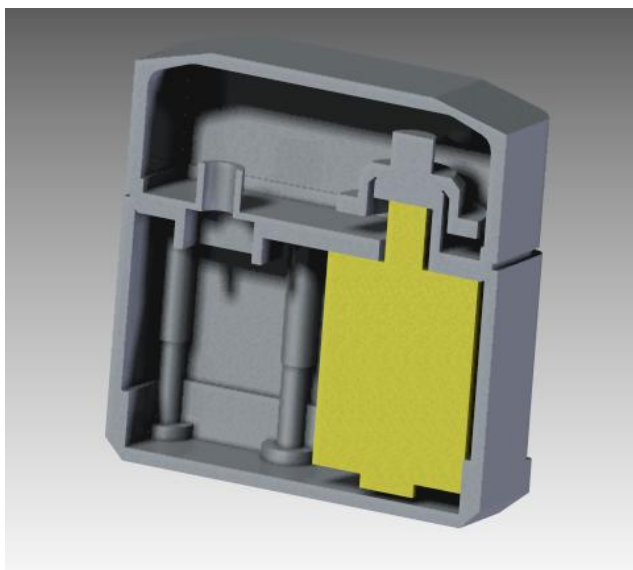
3. Szervó mechanikai, hardveres módosításai

3.1 Fizikai módosítások

A munkadarab belsejéből a DC motoron, fogaskerék áttételezésen kívül gyakorlatilag mindent eltávolítottunk. Ezenfelül a közepén található tartóoszlop eltávolításra került, mivel az a régebbi potenciométert tartotta.

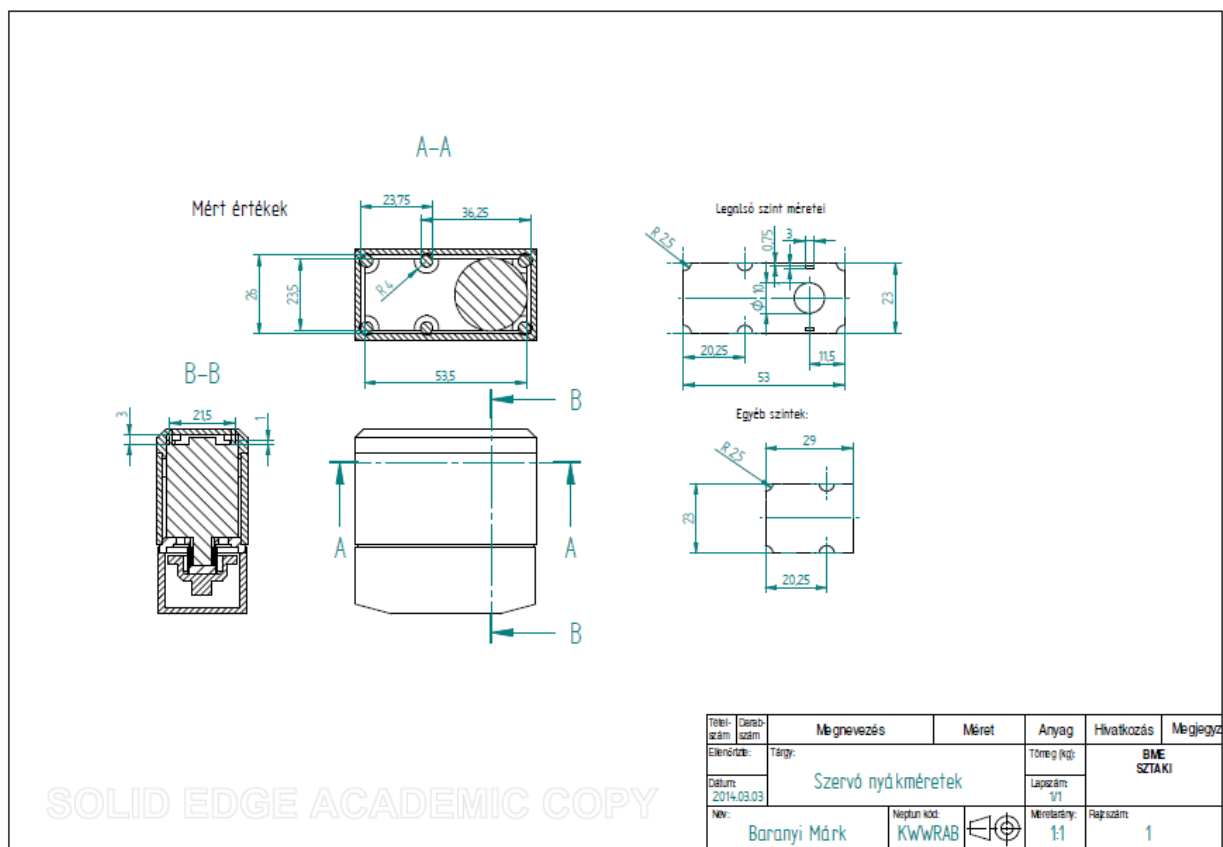
A házba becsatlakozik majd egy CAN kábel, illetve egy tápegység is. Így 2-2 db csatlakozó lábat kell elhelyezni a házon.

A NYÁK összesen kétemeletes: Az alsó NYÁK teljesen az aljában, szorosan fog illeszkedni az alsó részhez, kitöltve a legtöbb helyet. A motort pedig rátűzve be lehet forrasztani, ezáltal is még merevebbé téve a szerkezetet. A második emelet kb. 4-5 mm-el feljebb, szorosan a motor tengelyéhez közel helyezkedik el. Mivel nagyon szorosra terveztük a NYÁK-okat, pontos méreteket kellett venni, CAD szoftverrel megrajzolva 3D-ben (2. ábra), majd műszaki rajzzal beméretezve adtuk meg a pontos méreteit a NYÁK-nak (lásd 3. ábra).



2.ábra: A szervó 3D metszeti ábrája(saját készítés)

A modellezés során elegendő volt csak a belső részt részletesen lemérni és megrajzolni. A 3D ábra alapján a NYÁK-ok méretei a következő képpen adódtak ki:



3.ábra: Szervó NYÁK műszaki rajz.(saját készítés)

A műszaki rajznak köszönhetően egyszerűbben és pontosabban meg lehetett határozni a NYÁK-nak a méreteit. Mint utólag kiderült nagyon szorosan és megfelelően illeszkedik is a dobozba. A rajzon jelölve van, hogy a legtöbb szinteken kívül a többi azonos mérettel rendelkezett volna, de nem volt szükség kettőnél több szintre.

Ezek után már a méretre szabott NYÁK-ok két oldalára lehetett megtervezni a szükséges áramköröket. A rajzot [Solid Edge ST6 Student Edition](#) programmal készítettem.

3.2 Szabályozáshoz szükséges módosítások

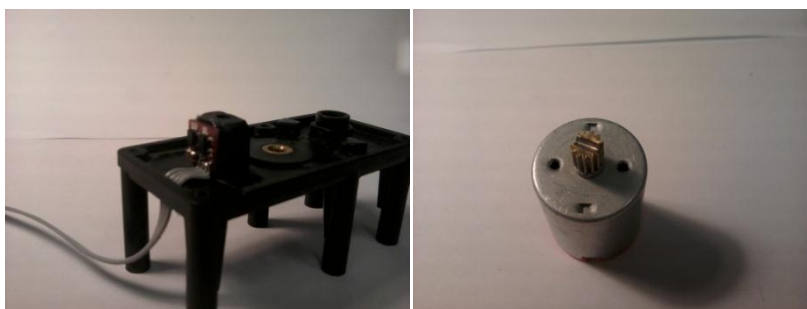
A szabályzó elektronika megtervezésének végső célja a szöghelyzet pontos szabályozása. Ehhez a motort vezérelni kell tudni és az állapotokat is szükséges mérnünk. Ezen felül kommunikálnia is tudni kell.

A motor PWM vezérlésű, melynek kapcsolása a NYÁK-on lesz kialakítva. A tápfeszültség, EMF, áram méréséhez szükséges kapcsolások is a NYÁK-on kapnak helyet. A szabályozáshoz közvetlenül az EMF és a szöghelyzet mérési eredményekre van szükség. Ezek közül a szög méréséhez kellett mechanikai módosítást végezni. A mérés mágneses szöghelyzet érzékelésen alapul az áttervezés után, melyet az áttételezés után mérünk egy beszerelt forgó mágnes és a hozzátartozó szögjeladó segítségével.



4. ábra: A szögjeladó elhelyezése

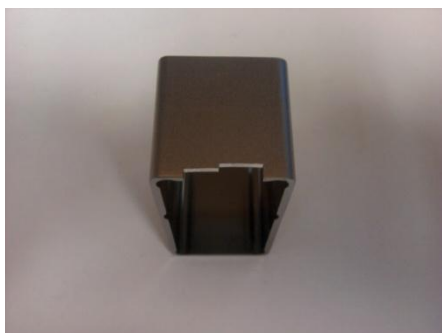
Egy alternatív megoldást is biztosítunk a fordulatszám méréséhez. Egy mágneset helyeztünk el a motor kimenő tengelyén lévő fogaskerékbe és ennek a közelében helyeztünk el két darab Hall elemet, melyekkel a fordulatszámot és a forgásirányt is meg tudjuk határozni.



5. ábra: Alternatív fordulatszámméréshez szükséges módosítások

Mint az kiderült, szükség is volt erre a megoldásra, mert ennél a motornál az EMF mérés megbízhatatlannak bizonyult.

A becsatlakozásoknak a helyeit kimarattuk az alumínium házon a 6. ábrán látható módon.



6.ábra: Becsatlakozási pontoknak kialakított hely

3.3 A NYÁK-tervezés alaplépései

A tervezés egyik legfontosabb része a hardveres háttér megfelelő biztosítása volt. Figyelembe kellett venni a hardveres specifikációkat, ezen kívül NYÁK-tervezés szempontjából a következő követelményeknek is meg kellett felelni:

- Megfelelő hűtőfelület: 3,3V-os rendszert egy körülbelül 7,4V-os táppal látunk el, amit egy feszültség stabilizátorral szabályozunk.
- Vastag vezeték a tápnak: teljesítménykapcsoláson keresztül kap feszültséget a motor, és az áramot közvetlenül a tápról kapja.
- Kondenzátorok rendeltetési helyükhöz közel: erre külön figyelni kell, mert rengeteg hidegítő kondenzátor kerül a NYÁK-ra
- Felső alsó csatlakozók: statikailag stabilnak kell lennie a két NYÁK-nak, ezért megfelelően kell elhelyezni a csatlakozókat, mert támasztó funkciót is betöltenek
- CAN, Vcc, GND becsatlakozás megfelelő kialakítása

A legelső lépés az áramkörü elemek, azok közül is a megfelelő mikrovezérlő kiválasztása volt. Két terméket találtunk, ami megfelelne az elvárásainknak, ez pedig az Atmel által gyártott AT32UC3B1512 és a Texas Instruments mikrovezérlője, a Hercules TMS470M.

AT32UC3B1512 lényeges tulajdonságai:

- maximum 60Mhz (1.89 DMIPS/s), RISC architektúra,
- három csatornás időzítő/számláló,
- hét csatornás PWM,
- SPI kontroller/interfész,
- TWI interfész,
- nyolc csatornás 10 bit-es A/D konverter,
- 48 pin, kis méret,
- JTAG interfész.

Hercules TMS470M főbb paraméterei:

- 80 Mhz Cortex™-M3 CPU,
- 2 db CAN interfész,
- 16 csatornás időzítő,
- 16 csatornás 10 bites A/D konverter,
- 2 db SPI interfész,
- extra biztonsági funkciók (ECC, CRC),
- 100 pin, nagy méret,
- JTAG interfész.

Mindkét vezérlő elegendő A/D csatornával rendelkezik, amikre szükség is lesz, mivel az fordulatszámot, nyomatékot, tápfeszültséget mérni kell analóg jelekből. Az Atmel által gyártottat ki kell egészíteni egy CAN vezérlővel is, mivel nincsen ilyen belső funkciója. A hétcsatornás PWM viszont szükséges a motor megfelelő vezérléséhez (4 csatornát le is foglal). Egy TWI kommunikációs interfésszel ellátott áramköri elemet is tudunk működtetni vele (pl. szögjeladó). Ha egy SPI CAN vezérlőt alkalmazunk, akkor szinte minden perifériát kihasználunk a mikrovezérlőn. Extraképp be tudunk még rakni 2 Hall elemet, alternatív fordulatszám mérés céljából időzítő vagy külső megszakítás funkcióval rendelkező lábra. További előnye, hogy szoftveresen erősen támogatott és már általunk ismert mikrovezérlőről van szó.

A Hercules vezérlő előnye, hogy nem kell külön CAN vezérlőt rakni a NYÁK-ra, és a két darab SPI interfészt is ki tudjuk használni. Még az intézet számára nem ismert mikrovezérlő családnál viszont elég nagy vállalkozás a szabályozást megvalósítani.

Mindezt végiggondolva az AT32UC3B MCU mellett döntöttünk.

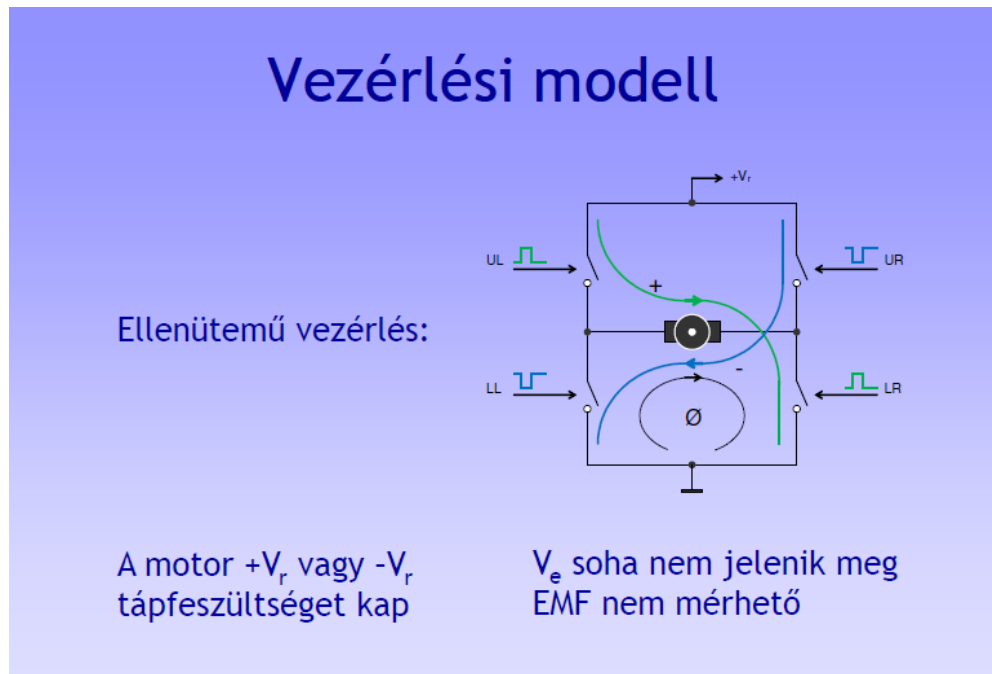
Kiválasztott áramköri elemek, amik a NYÁK-ra kerülnek:

- AT32UC3B1512
- 2-2 db. n és p csatornás FET
- 2 db. műveleti erősítő
- 1 db feszültség stabilizátor
- 2 db FET meghajtó
- 1 SPI perifériával rendelkező CAN vezérlő, hozzá transceiver
- 1 TWI perifériával rendelkező mágneses szögjeladó
- 2 Hall elem
- 1 dual schimdt trigger buffer
- csatlakozók, ellenállások, kondenzátorok
- 16 Mhz kristály oszcillátor

A bekötési tervek és az áramköri elemek pontos típusai megtalálhatóak a dolgozat mellékletei között.

3.4 Vezérlési modellhez használt alapkapsolás

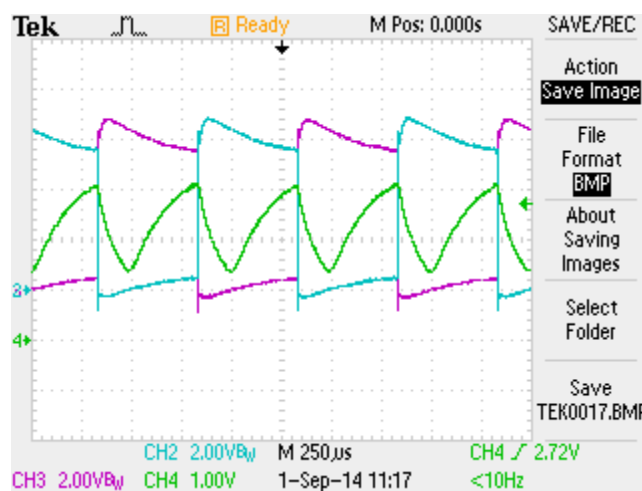
A motor irányításához ellenütemű vezérlést alkalmazunk a következők szerint (7. ábra):



7.ábra: A motor ellenütemű vezérlése (Forrás: Soumelidis Alexandros előadásanyaga)

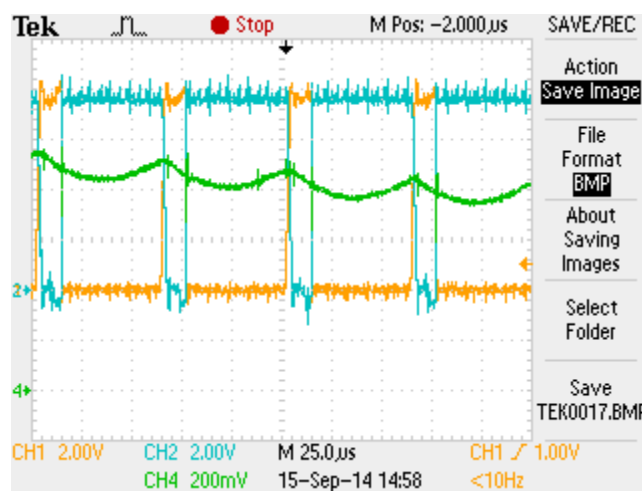
A képen is látható, hogy a motort a két irányba két-két PWM-jellel vezéreljük. Ez bonyolultabb vezérlést ad számunkra, ráadásul EMF mérést sem tudunk végezni a fordulatszám meghatározásához. Ennek oka, hogy a motornak az áramhullámossága túlságosan nagy volt alacsony PWM frekvencián (1000 Hz), ahol még tudtunk volna EMF mérést végezni.

A mérés eredménye pedig így néz ki (8. ábra):



8.ábra: Hídágban folyó áram mérése 1

Ha magasabb frekvenciájú PWM-et (16kHz) alkalmazunk, akkor a következő eredményre jutunk:



9.ábra: Hídágban folyó áram mérése 2

A mérést egy Tektronix digitális oszcilloszkóppal végeztük el. A zöld a hídágban folyó árammal arányos feszültség. A lila és a világoskék jel pedig az ellenütemű vezérlés két ellentétes PWM jele. A mérés alapján a maximális és a minimális áram különbsége 5 A nagyságúra adódott ki, ami igazolja azt, hogy az unipoláris vezérlés alkalmatlan lenne. Ezzel együtt az EMF mérés sem megfelelő a fordulatszám méréshez így az alternatív megoldást választottuk. Magasabb frekvenciát alkalmazva az áramhullámosság csökken és négyzetjelek alakja is elfogadhatóbb.

A mért jelek rövidítéseit és szükségességét a vezérléshez az alábbiak írják le:

BEMF: Back electromotive force, lényege az, hogy az impulzusok szüneteiben visszamaradó, a motor kapcsain mérhető feszültség arányos a fordulatszámmal. Ezt végül nem használjuk.

VBAT: Viszonylag nagy tartományban változó tápfeszültség, melytől függ a fordulatszám is, mert a FET-ek erre a tápra kapcsolódnak, ezért ezt a változást kompenzálni kell.

VIR: Túláram mérése, biztonsági szempont

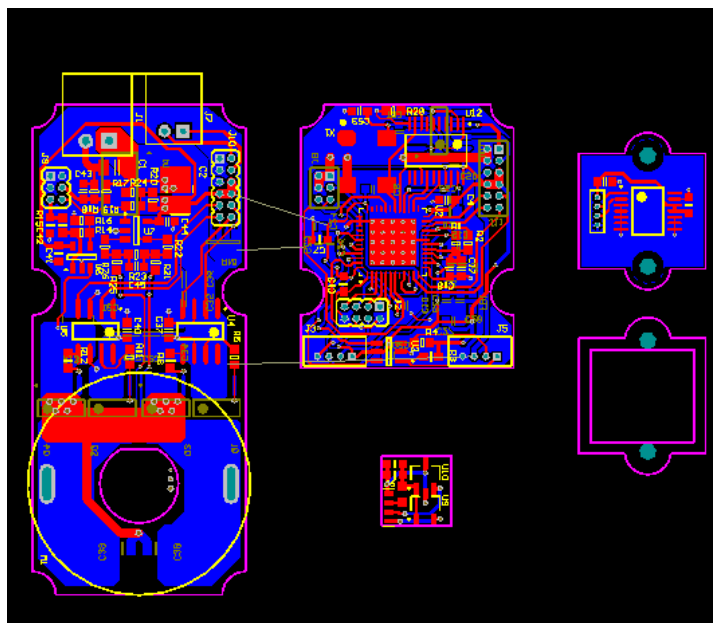
A szöghelyzetet pedig egy szögjeladó biztosítja nekünk.

3.5 A NYÁK-tervezése

A nyomtatott áramkörök, azaz a PCB-k (printed circuit board) áramköri tervező program segítségével készítettük el. Maga a szoftver és a NYÁK-tervezés megtanulása, elsajátítása külön időt vett igénybe, ráadásul nagyon kicsi 53x23mm-es és egy 29x23mm-es méretű lapkára kellett megcsinálni. Mindez körülbelül 3 hónapot (1 félév) vett igénybe, ez jelzi is a tervezés nehézségét is egyben. A NYÁK tervezésében Bakos Ádám tanár úr segített.

Összesen 4 db különálló NYÁK készült: A legelső szint, rajta a kisebb első emelet és külön készült a szögjeladónak és a Hall elemeknek is. A NYÁK-ot a könnyebb forrasztás érdekében forrasztásgátló lakkal vonták be ott, ahol nem szabad megtapadnia az ónnak. A méreteket (vonalvastagság) a gyártó ajánlásai alapján végeztük, hogy az árajánlat megfelelő legyen.

A végeredmény a 10. ábrán látható:



10.ábra: Kész PCB tervek (saját készítés)

A csatlakozók úgy lettek elhelyezve, hogy a felső NYÁK megtámasztva legyen. A motor szintjén a teljesítménykapcsolások, erősítő kapcsolások, feszültség stabilizátor, becsatlakozások kaptak helyet a NYÁK-on. Zajcsökkentés céljából az analóg eszközök (erősítők, szűrők) az alsó panelre kerültek, külön blokként, elválasztva a teljesítményelektronikai és a digitális részekről, alatta földkitöltéssel. Külön vannak a MOSFET-ek, és hozzájuk közel a meghajtóik. Talán érdekes lehet még az is, hogy a motorhoz menő vezetékek azért töltik ki teljesen a rendelkezésre álló helyet, mert egyben a FET-ek hűtőfelületeként is szolgálnak. Ezzel a kritikus és zajos elemek a mikrovezérlőtől távol kerültek és megfelelő elhelyezéssel az alsó NYÁK-on nem okoz problémát. A mikrovezérlő szintjére került a JTAG csatlakozó, a Hall elemek, a szögjeladó csatlakozásai, az MCP2515 CAN kontroller és a kristály oszcillátor. A szokásos, gyártó által javasolt hidegítő kondenzátorok elhelyezésre kerültek.

Beforrasztás szempontjából egy kis gyakorlás, kezűgyesség és türelem kellett, mivel mindezt kézzel csináltuk. A mikrovezérlőt hőlégfúvó segítségével rögzítettük a helyére, majd gyanta tartalmú ón segítségével beforrasztottuk a helyére az egyébként egymáshoz nagyon közel elhelyezkedő lábakat.

Kész állapotban, alumínium váz nélkül, tesztelésre készen így fest (11. ábra).



11. ábra: A szervó összeszerelt állapotban

4. A szervó felkészítése az irányítás megtervezéséhez

4.1 Élesztés, tesztelés

A beforrasztás utáni lépések már az irányítás megtervezését készítik elő. Először is ellenőrizni kell, hogy a mikrovezérlő megfelelően működik. Ehhez tápfeszültséget kell kapcsolni az eszközre és fel kell programozni. A tesztelést egy kiszámolt 1000 Hz frekvenciájú PWM jellel végeztük el. Ezzel az úgynevezett PLL (Phase Locked Loop) modul megfelelő beállításait és a beállított órajelet is ellenőrizni tudjuk, illetve ha ezt a motor egyik FET meghajtójára kapcsoljuk rá, akkor egyértelmű életjelet is kapunk. Ezzel az alapl működést le is ellenőriztük.

A perifériák tesztelése a programozás során történt.

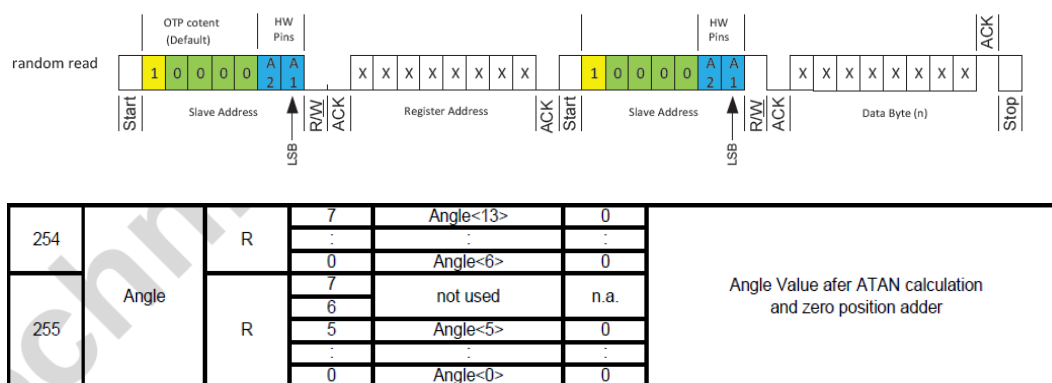
4.2 Az kontroller programozása

Az eszköz szegmenseit csak programozás során tudjuk alaposan letesztelni. Ezek pedig szögjeladó, CAN kommunikáció és maguk a mérések. Az egyéb perifériák nem kerülnek itt ismertetésre, mert azok általános felhasználásúak és nem igényelnek speciális kiegészítést a kódban sem (GPIO, PWM, Timer/Counter). A programozáshoz az [Atmel Studio](#) [Atmel Software Frameworkot](#) használtuk, amellyel ez a programozási folyamat felgyorsítható és leegyszerűsíthető. A következőkben ezeket fogom bemutatni.

4.3 Szögjeladó

A szöghelyzet mérésére egy AS5048B típusú mágneses szögjeladót használunk, amely 14 bites pontosságú, ahol egy bit eltérés 0.0219° -ot jelent. Ez elegendő pontosságú az irányításhoz szükséges egyik állapot méréséhez. Az eszköz támogatja a nagysebességű TWI kommunikációt, amivel lekérdezhetjük a pillanatnyi szöget.

A szög lekérdezéséhez elegendő ismerni a szolga címét és kiolvasandó regiszterek címét. Ezek pedig a következők (12. ábra):



12.ábra: A szögjeladó lekérdezése(Forrás: ams.com)

A kommunikáció sebessége az SCL órajelnek megfelelően adódik ki. A kiolvasott értéket pedig fel tudjuk használni az irányításhoz.

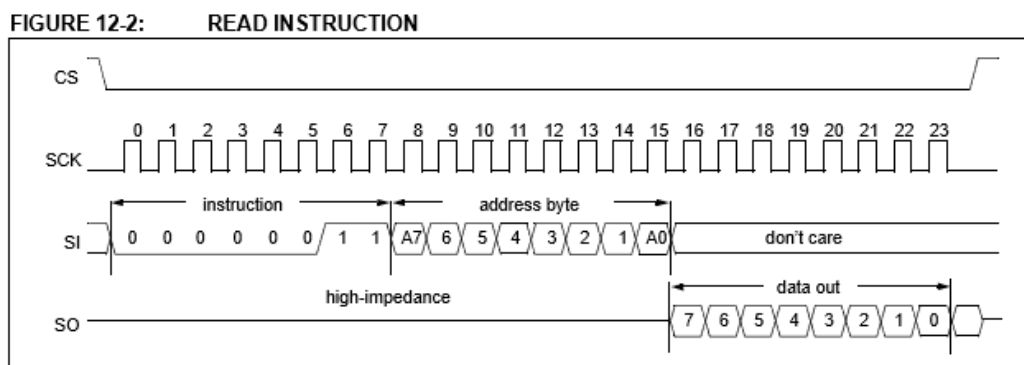
4.4 CAN kommunikáció

A CAN kommunikáció fontos szerepet tölt be a szabályozás megtervezésében, hiszen például referencia jelet ezen keresztül kapunk majd, illetve fejlesztéshez, diagnosztikához elengedhetetlen. Mivel az AT32UC3B nem rendelkezik CAN perifériával, így egy különálló CAN vezérlőt használunk, melyet SPI kommunikáció segítségével tudunk konfigurálni. Az eszköz neve MCP2515. Nagy előnye, hogy megfelelő beállítás után elegendő a megérkezett adatot kiolvasni, illetve az elküldendő adatot feltölteni a bufferbe. Gyakorlatilag így csak az SPI kommunikációra kell fektetni a hangsúlyt a megfelelő működéshez. A kommunikációt úgy oldottam meg, hogy a mikrovezérlő a lehető leggyorsabban feldolgozza az adatot. Ehhez pedig a megszakítás kezelés beállítása kellett. Az SPI kommunikáció sajátossága, hogy minden elküldött bájtot egy fogadott bájt követ. Tehát ha az UC3B SPI Transmit Data regiszterébe írunk, utána mindenképp kapunk adatot és erre tudunk megszakítást kérni. Így egy nagyon gyors írási-olvasási ciklust tudunk megvalósítani. Az SPI-on kiküldött első bájt

határozza meg mindig, hogy milyen típusú utasítást hajtson végre a CAN vezérlő. Ezek felsorolás szinten a következők:

- Reset (0xC0)
- Read (0x03)
- Read RX buffer (0b10010nm0)
- Write (0x02)
- Load TX buffer (0b01000abc)
- Request to send (0x81)
- Read status (0xA0)
- RX status (0xB0)
- Bit modify (0x05)

Ezek közül elegendő volt számomra a Read, Read RX buffer, Load TX buffer, Write, Request to Send parancsokat használni. Egy példa a használatra:



13.ábra: Példa az SPI kommunikációra (Forrás: microchip.com)

Egy kommunikációs ciklus maximum 3 bájt hosszú. A buffer írása/olvasása csak 2 bájt hosszú (DLC=1 esetén). Maga a küldés kérése pedig csak 1 bájt hosszú.

4.5 Analóg mérések

Mivel a mért jeleink egy része folytonos, ezért azokat digitális jelekké kell átalakítani. Összesen 3 db ilyen jelünk van: A BEMF, a tápfeszültség és a motor árama. Ezek közül a tápfeszültséget elegendő 100 ms időközönként mintavételezni, mivel csak az a fordulatszám szabályozás kompenzálásához kell és értéke nem változik gyorsan. A BEMF mérést az előzőekben említett okok miatt nem alkalmazzuk.

A VIR (áram) méréshez meg kell nézni, hogy magas frekvencián milyen jelet ad ki. Ha közel konstans feszültséget ad, akkor nem szükséges a mérés ütemezésének beállítása a PWM jelhez. Ezt korábban már megvizsgáltuk, ahol az áramhullámosságot mértük meg, és ezzel együtt elfogadható értéket kaptunk.

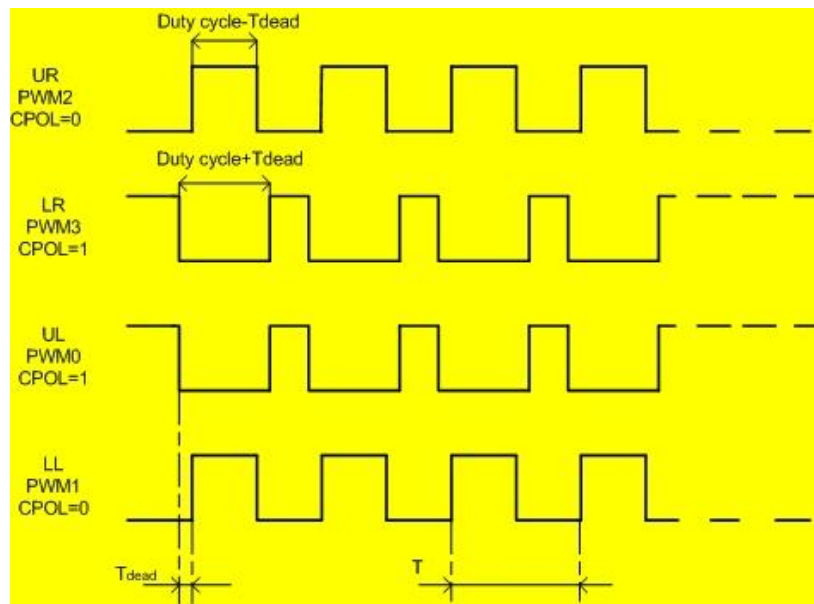
4.6 A fordulatszám mérése

A fordulatszám méréshez a vezérlő egység input capture funkcióját használjuk, mint alternatív megoldást. Időmérés alapú fordulatszám meghatározást választottunk, mivel ezt a perifériát 24 MHz frekvenciával működtetjük és a fordulatszám meg se közelíti ezt az értéket (maximum 350 Hz). Így egy kellően pontos mérést tudunk alkalmazni. A mérés nem igényelt speciális kiegészítést a kódban.

4.7 A vezérlési modell megvalósítása

Mint már az korábban említettük, a szabályozáshoz az ellenütemű vezérlést használjuk. Ehhez a mikrovezérlő segítségével négy darab PWM jelet kell generálni, hogy a FET meghajtókon keresztül kapcsolgathassuk a vezérlő áramkört. A nehézség abban rejlik, hogy egyszerre egy hídágban található teljesítménykapcsoló nem nyithat, ezért ezt ütemezni kell. Ráadásul a kapcsolási idővel is számolni kell, tehát a kapcsolások között holtidőt kell beiktatni.

A megvalósítás pedig így néz ki (14. ábra):



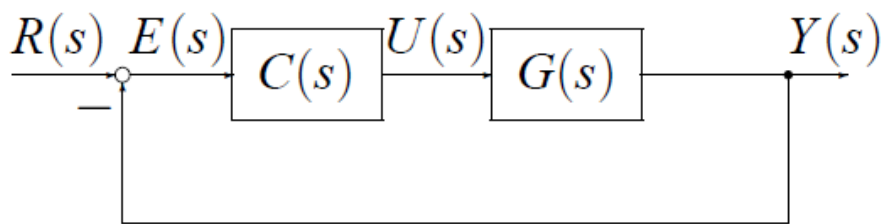
14.ábra: A négy PWM ütemezése(saját rajz)

Ezzel a megoldással nem melegednek a FET-ek, ráadásul mindegyik periódusideje azonos. A holtidőnek valójában a kétszeresét vesszük és az egyik polaritásából kivonjuk, a másikhoz pedig hozzáadjuk. Így meg tudunk határozni egy közepes kitöltési tényezőt, ami megkönnyíti a négy jel vezérlését is.

5. A DC motor irányítási lehetőségeinek ismertetése

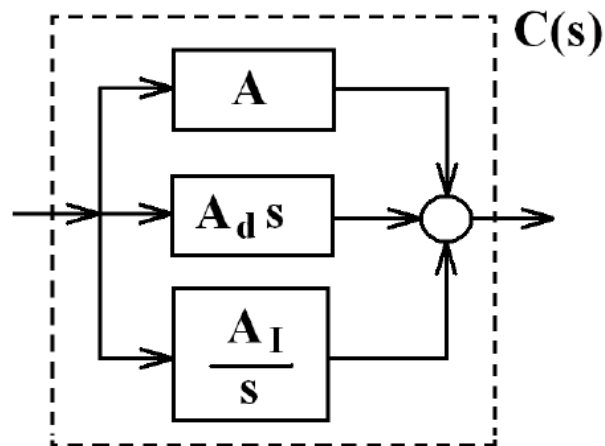
5.1 A fekete doboz módszer

A gyakorlatban nagyon elterjedt tervezési módszer. Lényege, hogy magát a rendszert nem, vagy csak kis mértékben ismerjük. Ezért a rendszert úgy tudjuk a kívánt módon hangolni, hogy beállítjuk a szabályzót egy tetszőleges kezdeti értékre, majd megfigyeljük a rendszer kimenetét. Ezután a kimenet alapján finomítjuk a szabályzót, amíg el nem érjük a kívánt hatást.



15. ábra: Zárt szabályozási kör (Forrás: egyetemi jegyzet)

A mi esetünkben a $G(s)$ nem ismert, viszont a $C(s)$ -t mi tudjuk beállítani. Ezt a módszert csak tesztelésre használtuk. A $C(s)$ pedig 3 tagból áll:



16. ábra: $C(s)$ átviteli függvény (Forrás: egyetemi jegyzet)

Lényegében egy arányos, differenciáló és egy integráló tag összege. Ezek közül mi az arányos és az integráló tagot használjuk. Csak teszteléshez alkalmaztuk ezt a módszert.

5.2 Az irányítás megtervezése egytárolós rendszert feltételezve

Tegyük fel hogy a $G(s)$ motor átviteli függvénye a következő alakú, aminek a bemenete a motor kapcsaira adott feszültség és a kimenet pedig a motor tengelyén kimenő fordulatszám:

$$G(s) = \frac{A}{1 + Ts}$$

Ezzel leegyszerűsítjük a problémát az A erősítési tényező és a T időállandó meghatározására. Az erősítést úgy tudjuk meghatározni, hogy nyugalmi állapotban megvizsgáljuk, hogy adott bemenetre mekkora kimenetet produkál a motor és vesszük a kettő hányadosát. Az időállandó meghatározásához pedig egy egységugrást adunk a rendszerre és megvizsgáljuk a beállási időt. Mindezt Matlab és egy CAN-es mérőprogram segítségével határozzuk majd meg. Hozzá kell tenni, hogy ez a fordulatszám szabályozásra vonatkozik, a pozíció szabályozáshoz egy $\frac{1}{s}$ taggal kell bővíteni a rendszert, illetve az áttételi arányt is figyelembe kell venni a kimeneten.

5.3 Motorszabályozás tervezése a DC motor dinamikus modellje alapján

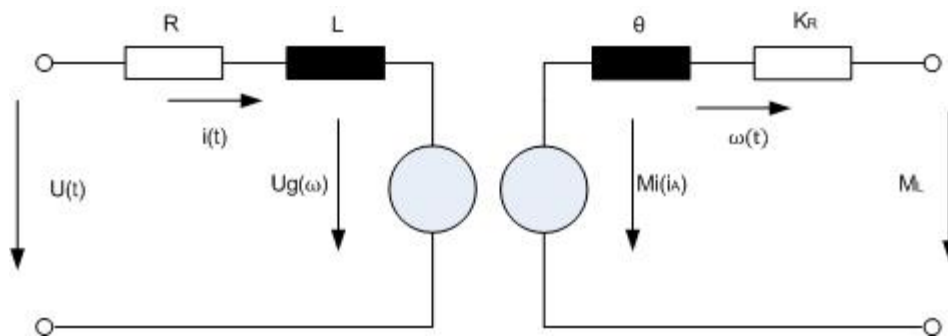
A DC motort két egyenlet segítségével írhatjuk le:

$$(1) \quad L_r \times \frac{d}{dt} \times i_r + R_r \times i_r = v_r(t) - k_m \times \omega$$

$$(2) \quad J_r \times \frac{d}{dt} \times \omega + \mu_m \times \omega = k_m \times i_r$$

$$T_m = k_m \times i_r$$

$$v_e = k_m \times \omega$$



17.ábra: DC motor helyettesítő kapcsolása (saját rajz)

Az első egyenlet a motor elektromos viselkedését, a második pedig a mechanikai viselkedést írja le.

A jelölések a következőket jelentik:

- L_r : A forgórész induktivitása
- i_r : A forgórész árama
- R_r : A forgórész villamos ellenállása
- v_r : A forgórész kapcsolófeszültsége
- k_m : Nyomatéktényező
- ω : A forgórész szögsebessége
- J_r : A forgórész tehetetlenségi nyomatéka
- μ_m : Viszkózus súrlódási tényező

Mindezt felhasználva felírhatjuk a rendszer átviteli függvényét, illetve az állapotteres alakját is. Az átviteli függvény felírása után megtervezhető a zárt rendszer átviteli függvénye a C(s) erősítési tagjainak meghatározásával. Ez a módszer hasonló a második pontban (5.2) leírtakkal, csak egy bonyolultabb, pontosabb megoldása az irányításnak, ami egy kéttárolós átviteli függvényt eredményez. Ezt az alkalmazást jelen tárgyalásból kihagyom, helyette az állapottér reprezentációból valósítok meg lineáris kvadratikus szabályozást (LQR), mivel az állapotvektor minden eleme mérhető, ráadásul Matlab/Simulink környezetben könnyen tervezhető, modellezhető a rendszer. Azonban ehhez ugyanúgy szükséges a kéttárolós átviteli függvény paramétereinek meghatározása, mert abból határozzuk meg az állapottér reprezentáció elemeit.

Az állapotteres leírás ω , i_r állapotvektorokkal:

$$\frac{d}{dt}\omega = -\frac{\mu_m}{J_r}\omega + \frac{k_m}{J_r}i_r$$

$$\frac{d}{dt}i_r = -\frac{k_m}{L_r}\omega - \frac{R_r}{L_r}i_r + \frac{1}{L_r}v_r(t)$$

Ezekkel a tervezési módszerekkel összehasonlítási alapot kaphatunk az egyszerűsített irányítástervezés és a bővebb, állapottér reprezentáción alapuló irányítás realizáció között.

5.4 Robusztus irányítás tervezés

Említés szinten pár szót ejteni érdemes erről a módszerről is. Egy szabályozási rendszer esetén a gyakorlatban nem ismerjük a $G(s)$ rendszer átviteli függvény paramétereinek pontos értékeit, helyette egy közelítő $G_N(s)$ névleges modellt alkalmazunk. A $C(s)$ szabályzót úgy kell beállítani, hogy a valós rendszer stabilitása is biztosítva legyen. Ezt nevezzük robusztus stabilitásnak.

Ha ismerjük az eredeti és a névleges modell közti eltérés legnagyobb értéket akkor tudjuk ezt a tervezési metódust alkalmazni. Jelen dolgozatban nem foglalkozunk ezzel a módszerrel.

6. Rendszer identifikációs lépések az egytárolós átviteli függvény meghatározásához és az irányítás megtervezése

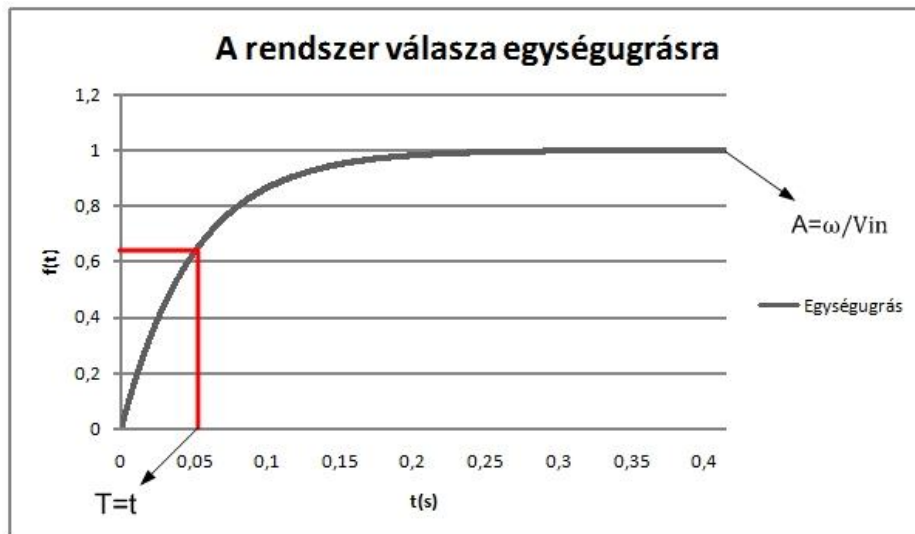
6.1 Az identifikációhoz szükséges eszközök ismertetése

Legelőször az egytárolós átviteli függvény paramétereinek meghatározását tűztem ki célul, mert maga a módszer átlátható és akár kézileg is leellenőrizhető. Nézzük a fordulatszámra vonatkozó átviteli függvény egységugrásra adott válaszát időtartományban (átmeneti függvény az eredmény):

$$\omega(t) = V_{in} * A * (1 - e^{\frac{-t}{T}})$$

Az egyenletet inverz Laplace transzformációval kapjuk meg.

Egy egyszerűbb ábrával illusztrálva ezt:



18.ábra: Az egytárolós átviteli függvény átmeneti függvénye(saját rajz)

Belátható hogy $t \rightarrow \infty$ esetén a rendszer A erősítését tudjuk megállapítani. Az exponenciális függvény alapján pedig van olyan pont ahol $t=T$. Ezt pedig a $0,632 \cdot \omega(t \rightarrow \infty)$ helyen tudjuk leolvasni.

Így már minden információ adott a rendszer átviteli függvényének meghatározásához. Azonban ehhez mérni is tudni kell ezt. Szükség van a fordulatszám mérésére kellő sűrűséggel, miközben a PWM kitöltési tényezőt is vizsgálni kell, amiből a közép feszültséget számoljuk. A megoldás kézenfekvő: Mivel egy CAN kommunikációra képes eszközről van szó így ki tudunk alakítani egy olyan tesztkörnyezetet, ami valós idejű adatot biztosít nekünk és el is tudjuk azt tárolni a későbbi feldolgozáshoz. Ehhez Soumelidis Alexandros tanár úr segítségét kaptam, aki egy olyan szoftvert készített, ami képes ennek lebonyolítására. Ez a tesztkörnyezet a DCMcan elnevezést kapta és LabWindows/CVI segítségével lett megírva. A program lényege, hogy felhasználói felületet biztosít az egyes beavatkozó jelek generálásához és így akár egységugrást is tudunk generálni.



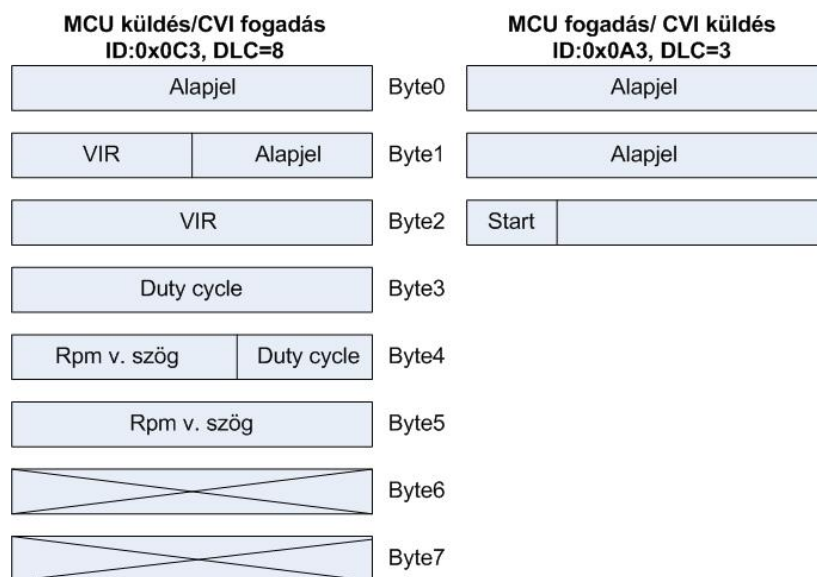
19.ábra: DCMcan felhasználó felülete

A programot (mikrovezérlőt és a tesztkörnyezetét egyaránt) úgy alakítottam át, hogy a megfelelő CAN azonosítókat használják, megfelelő adatformátummal. Ezenkívül a teszteléshez egy úgynevezett CANboardot is igénybe vettem az alapfunkciók teszteléséhez.



20.ábra: Tesztkörnyezet

A kialakított adatkeretek a következők:

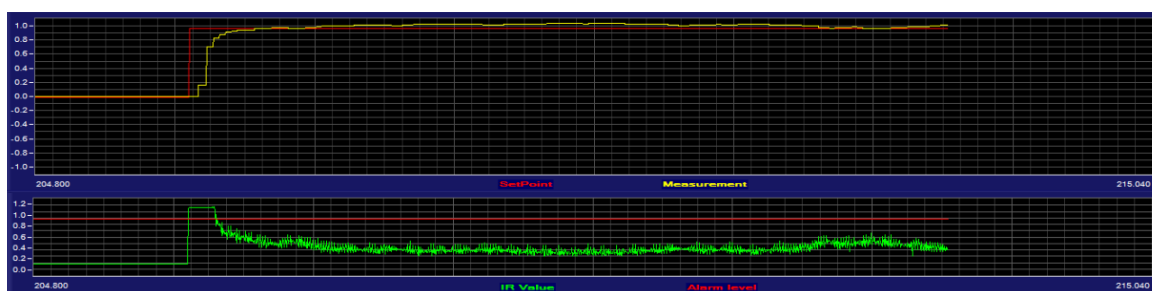


21.ábra: A méréshez kialakított adatok(saját rajz)

Ezen kívül még egy NI USB-8473 CAN csatlakozót is használunk, hogy a számítógép és a szervó között a kommunikációt biztosítsuk. Az LCD kijelzős tesztboardra továbbra is szükség volt, mert az alapján ellenőriztem a helyes mérést, másrészt szükség volt a rajta található lezáró ellenállásra, hogy működjön a kommunikáció. Ezzel minden adott a mérés lebonyolításához és az egyik tervezési módszerhez szükséges identifikációhoz.

6.2 A rendszer átviteli függvényének meghatározása

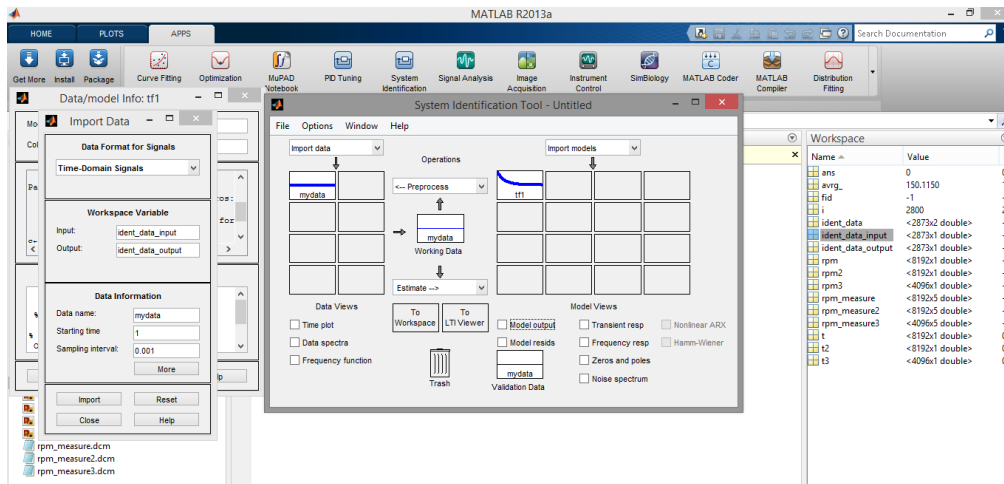
A mérést elvégezve a következő adatsorozatot kaptuk:



22. ábra: Az egységugrás mérési eredménye

Ehhez hasonló adatsor több is készült, a megfelelően pontos számítás érdekében. Az eredményeket a program elmenti olyan formátumban, hogy Matlab segítségével könnyen feldolgozhatjuk. Mivel rendelkezésünkre áll a bemenet és a hozzátartozó kimeneti adatsor, ezért alkalmazhatjuk a Matlab rendszer identifikációs programcsomagját. A program

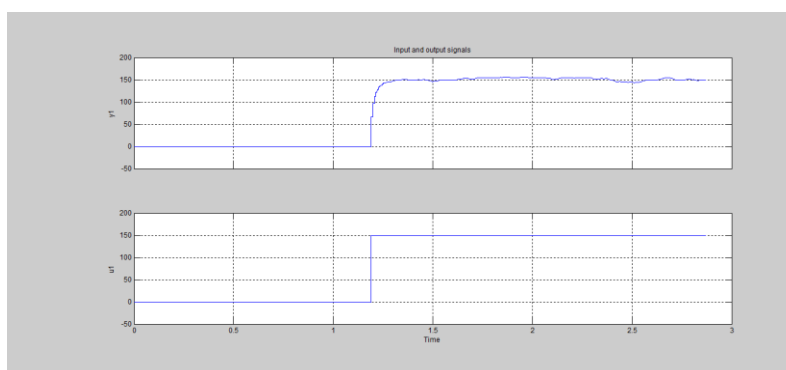
alapbeállításainál az ARX modell struktúrát használtam. A időlépték adott 1ms-nek lett beállítva, mivel az adatokat ilyen sűrűséggel kaptuk meg. Maga az automatizált identifikáció így néz ki a 23. ábrán.



23.ábra: Matlab használata az átviteli függvény megállapításához

Az egész programcsomag úgynevezett drag and drop rendszerű. A mért adatokat beolvassuk, meg kell adni, hogy milyen időintervallumban mértünk, aztán ki kell választani, hogy az átviteli függvényt szeretnénk meghatározni, ami egytárolós alakú. Ezután a program kiszámolja a mért adatokhoz leginkább illeszkedő átviteli függvényt. Utána a megkapott eredményt értelmezni és elemezni kell.

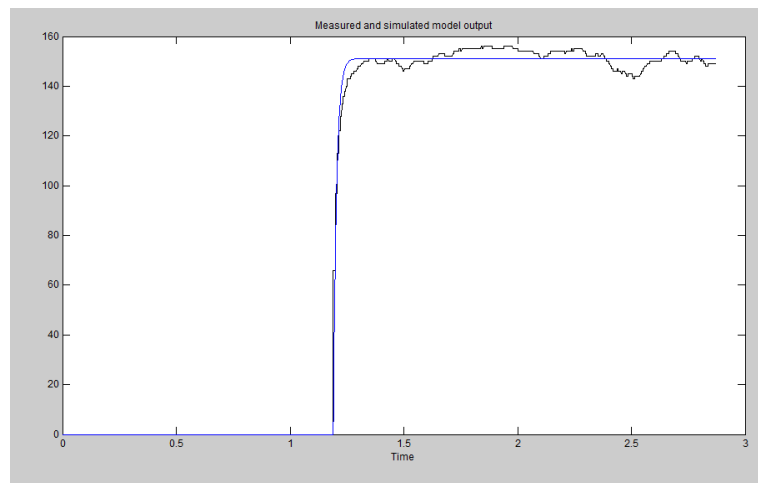
Több eredmény feldolgozása után a legjobban illeszkedőt választottam ki. Az eredmények közül két ábrát elemeztem. A megadott adatsort, hogy azt dolgozta-e fel, amit megadtam, illetve az összehasonlító ábrát.



24.ábra: A bevitt adatsorok

Ennek ellenőrzésére azért volt szükség, mert előfordulhat, hogy nem megfelelően adtuk meg az adatsort.

Végül pedig a leginkább illeszkedő átviteli függvény a következő:

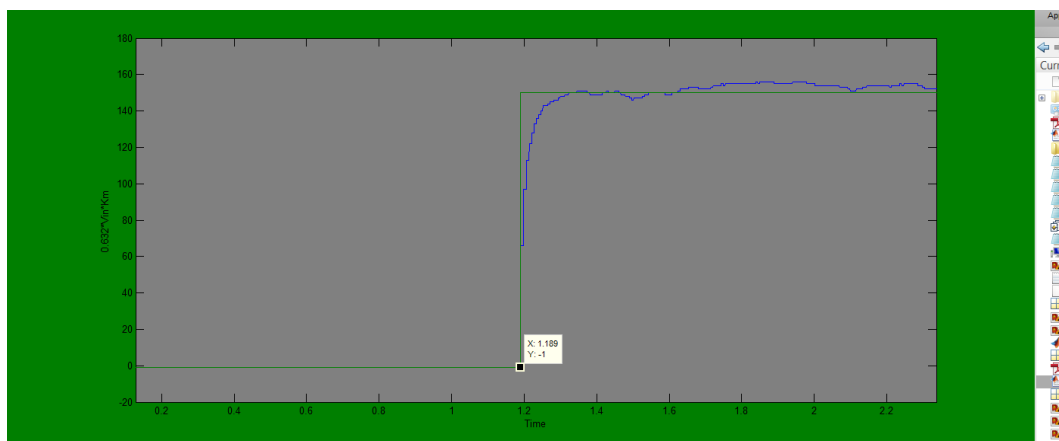


25. ábra: A mért és a szimulált modell átmeneti függvénye

Az átviteli függvénynek a következőt számolta ki a program:

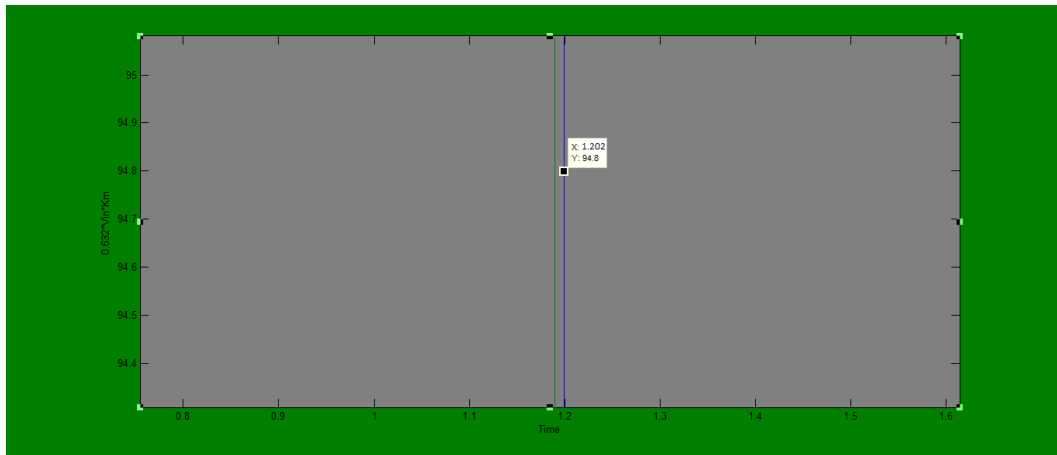
$$G(s) = \frac{23,25}{1 + 0,038s}$$

Ellenőrzésnek a kézi számítást választottam. Az exponenciális függvény paramétereinek meghatározásához a bevitt adatokat használtam fel. A kiindulási pont a következő:



26. ábra: Kézi ellenőrzéshez használt adatsor

A T időállandó meghatározásához megnéztem a függvény kezdeti értékét, aztán pedig leolvastam azt a helyet, ahol $t=T$. Előtte azonban a beállási értéket kellett meghatározni, ami egyszerű átlagszámítással 150,1-re jött ki. Az időállandó pedig 0,038-re jött ki.



27. ábra: Az időállandó leolvasása

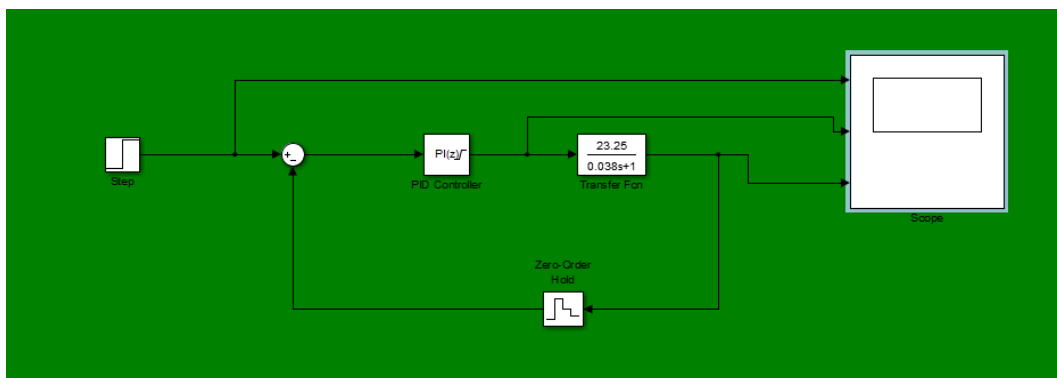
Kézi számítással ezután a következő átviteli függvényt kaptuk:

$$G(s) = \frac{25.4}{1 + 0,04s}$$

A két eredmény nagyon közel áll egymáshoz, így ez elfogadható az irányítás megtervezéséhez a következőkben. Ez alapján modellezés segítségével már meg tudjuk határozni a zárt rendszer átviteli függvényét. Ha a modellezés és a tesztelés sikeres, át lehet térni a rendszer megtervezésére is a dinamikai egyenletek alapján.

6.3 A zárt rendszer modellezése és tesztelése

A modellezés Simulink segítségével történt.

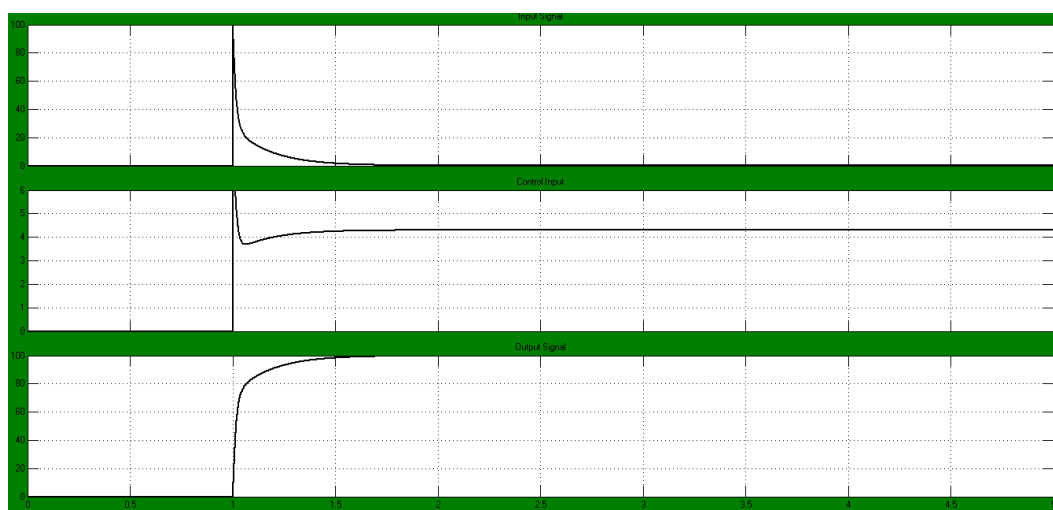


28. ábra: A fordulatszám szabályozás modellezése

Mivel a modellezést minél jobban szeretnénk a valósághoz közelíteni, ezért alkalmazunk a kimeneti jel visszacsatolásánál egy nulladrendű tartószervet, ahol a mintavételi idő 1 ms, illetve a PID blokkban pedig a beavatkozó jelet korlátozzuk, mivel az ismert. A modellben is a szabályozás is diszkrét idejű. A P és az I tagot PID tuner nevű eszköz segítségével tudjuk finomhangolni, ahol lehetőség van az időtartományi és frekvenciatartományi vizsgálatra is.

Az időtartományi vizsgálat esetén a válaszidőt és a tranziens lezajlását tudjuk befolyásolni. Frekvenciatartományi vizsgálat esetén a vágási frekvenciát és a fázistartalékot lehet módosítani. Jelen esetünkben legalább 60° fázistartalék szükséges, mivel a modellbizonytalanság nagy és kezdetben a beállási időt is érdemes kisebbre venni.

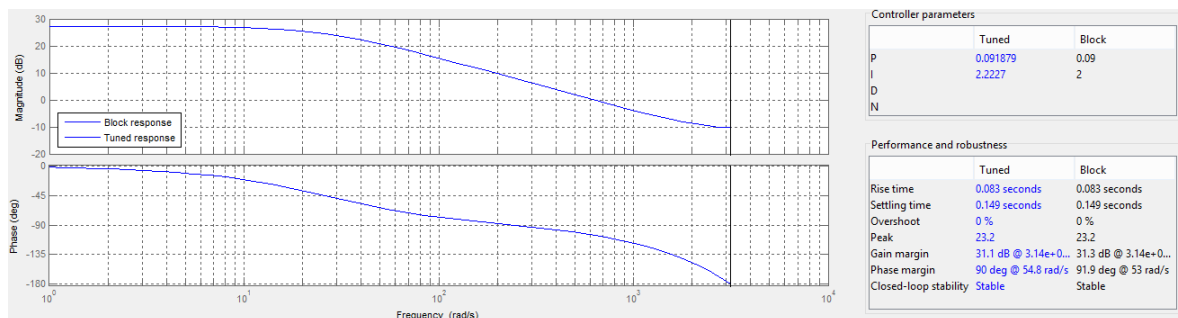
Lefuttatva a szimulációt:



29.ábra: Szimulációs eredmények fordulatszám szabályozás esetén

A beállási idő viszonylag nagy, azonban a rendszer stabil és a valós környezetben is tesztelhető. Ez természetesen nem jelenti azt, hogy implementálva a programban is pontosan ugyanezt kapjuk. A fogaskerék áttétel és maga a motor jelentős nemlinearitást visz a rendszerbe, emiatt nagy valószínűséggel a modellezett irányítást hangolni kell.

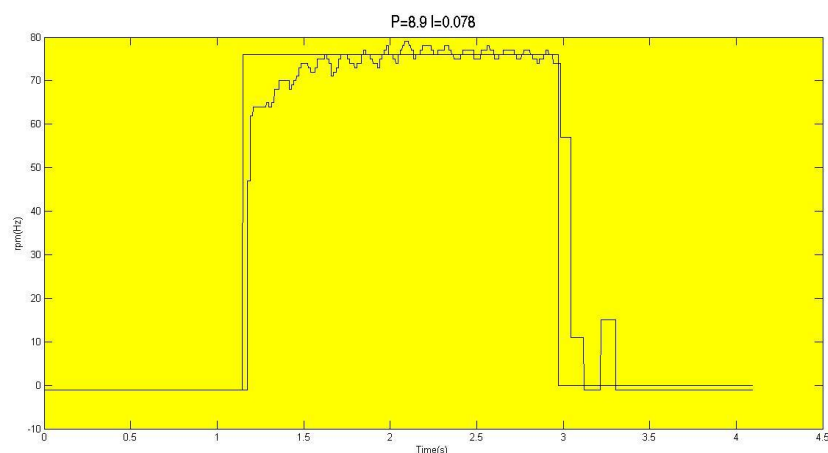
Érdemes megnézni a zárt hurok Bode-diagramját frekvenciatartományi vizsgálat és ellenőrzés céljából:



30.ábra: A modellezett zárt hurok Bode-diagramja

A rendszer fázistartaléka nem meglepő módon nagy, később az erősítési értékek módosítva lesznek a pozíció szabályozás hangolásakor.

Mivel minden rendelkezésre áll a valós környezetben történő tesztelésre, rögtön ki is próbáltam. A P és az I tag hangolásával a következő eredmény kaptam:



31.ábra: Tesztelés valós környezetben

A különbség tisztán kivehető. Egyrészt holtidő is van a rendszerben, másrészt a rendszerben kisebb lengések vannak. A nemlineáris viselkedés miatt alacsony fordulatszámon és magas fordulatszámon is másképp viselkedik. Ugyanis alacsony fordulatszámot adva referencia jelnek a rendszer instabil, magas fordulatszámot megadva pedig beáll és stabilan tartja azt. Emiatt érdemes egy olyan technikát alkalmazni, hogy alacsony és magas fordulatszámon is más-más erősítési értékeket alkalmazunk. A kisebb lengések oka a fogaskerék áttétel miatt van, azonban ezt is bent kellett hagyni a rendszer átviteli függvényének meghatározásakor, mert akkor más eredményt kapunk. A fogaskerek kapcsolószáma 3 körüli a magas fogszám

miatt, de tranziensek esetén ütköznek így is egymáshoz. Beálláskor pedig a fogaskerekek forgását folyamatosan fent kell tartani és ez jelentős energiát vesz igénybe, mert a fogaskerekek inerciája nagyobb, mint maga a motoré.

Mindezeket átgondolva érdemes a szabályozást több tartományra felosztani, mégpedig egy alacsony fordulatszám szabályozásra és a normál szabályozásra. A lengés nem számottevő, de foglalkozni érdemes lesz vele később.

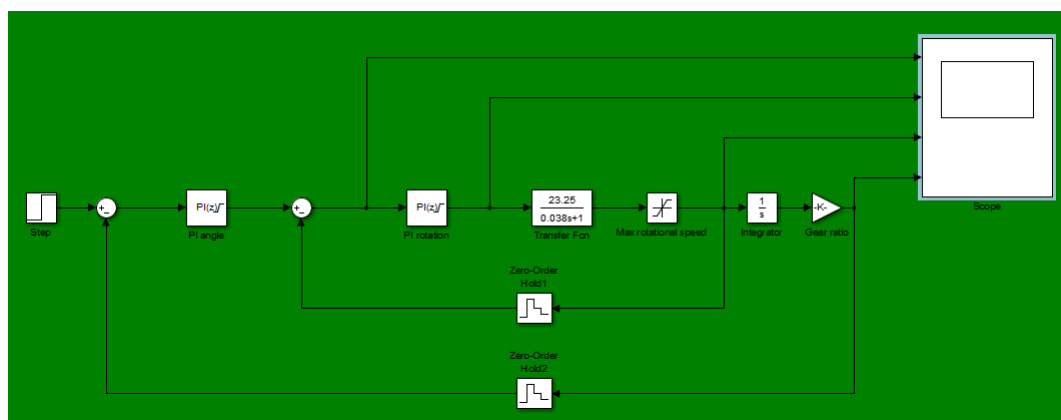
6.4 Pozíció szabályozás modellezése és tesztelése

Mivel a végső cél a szöghelyzet pontos szabályozása, ezért ezt külön kell modellezni és tesztelni. Gyakorlatilag az eddigi rendszert kell kiegészíteni egy integráló taggal, és ezután újra modellezhetjük a zárt rendszert.

A rendszer így integráló jellegű, aminél akár elég egy arányos tag megfelelő beállítása is.

Azonban érdemes letesztelni a PI szabályozást is.

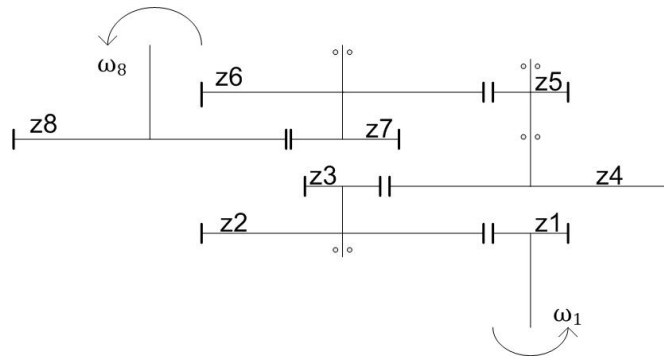
Kis módosítással az előző blokksémát, megkapjuk a zárt hurkú szabályozási modellt erre az esetre is:



32. ábra: A pozíció szabályozás modellezése

Gyakorlatilag az eddigi szabályozást egy külső hurokkal egészítettük ki. Fontos volt az áttételi arány pontos értéke, ehhez a régimódi fogszám meghatározását alkalmaztam. Összesen 8 darab fogaskereke van, amivel jelentős nagyságú módosítás érhető el. Az eredmények:

- $z_1=13$
- $z_2=55$, melyhez $z_3=13$ fogszámú fogaskerék rögzítve
- $z_4=55$, melyhez $z_5=13$ fogszámú fogaskerék rögzítve
- $z_6=55$, melyhez $z_7=15$ fogszámú fogaskerék rögzítve
- $z_8=48$



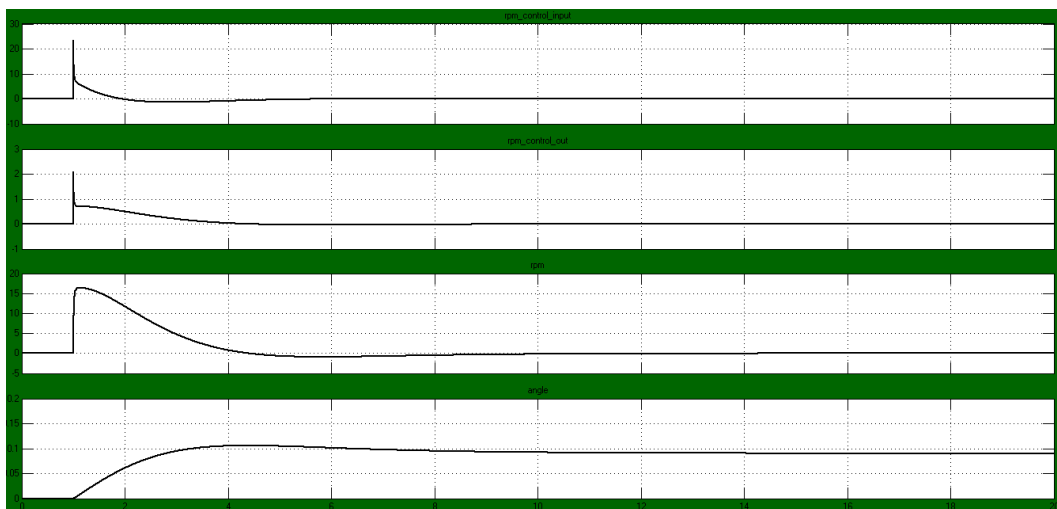
33. ábra: A szervó fogaskerék áttételezésének vázlata(saját rajz)

Ebből a kimeneti fordulatszám:

$$\omega_8 = \omega_1 \times \frac{z_1}{z_2} \times \frac{z_3}{z_4} \times \frac{z_5}{z_6} \times \frac{z_7}{z_8} = \frac{1}{242,33}$$

Nem véletlenül lett választva ez a szervó, ugyanis nagyon nagy nyomatékmódosítása miatt stabilan kormányozható a jármű terhelés alatt is.

Ugyanazon a módszeren végigfutva, felhasználva segítségül a PID tunert az alábbi eredményt kapjuk:



34. ábra: Szimulációs eredmények pozíciószabályozás esetén

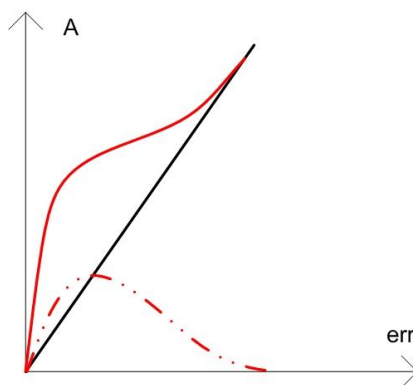
Az ellenütemű vezérlés nagy hátránya, hogy kisebb kimeneti fordulatszám érhető el, mint unipoláris vezérlés esetén, mivel meg kellett határozni egy maximális kitöltési tényezőt a dead time-ok miatt.

Mielőtt még magán az eszközön is letesztelnénk a szabályozást, gondoskodni kell arról, hogy a belső hurok, azaz a fordulatszám irányítása megfelelően gyors legyen, mert az fogja meghatározni az egész rendszer gyorsaságát. Azonban felvetődött egy olyan, már említett

probléma, és a tesztelések során ezért nem lett gyorsabb tisztán PI szabályozás megvalósítva, hogy alacsony fordulatszám tartományon az irányítás instabil állapotba kerül.

Emiatt konzultálás során arra a döntésre jutottunk, hogy érdeemes a lineáris szabályozástól eltérően irányítani a rendszert egy bizonyos fordulatszám alatt. Nyilván ez is csak egy bizonyos szintig elég, mivel egy fordulatonként csak egy impulzus jön fordulatszám mérésakor, ami megnehezíti ezt, de egy ilyen módszerrel valószínűleg javíthatók a tulajdonságok.

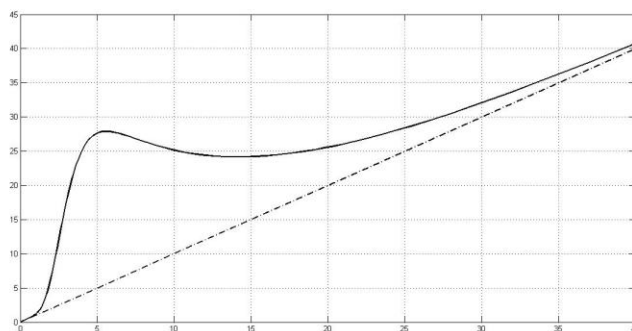
Nézzük az elgondolást a következő ábrán:



35.ábra: Nemlineáris erősítési értékek alacsony fordulatszámon(saját rajz)

Az eddigi erősítést egy exponenciális függvénnyel egészítjük ki. Ezzel induláskor lényegesen nagyobb, akár háromszorosos erősítést kapunk. Ez már valószínűleg segíteni fogja a motort abban, hogy induláskor könnyebben legyőzze a tapadási súrlódást és kellő perdületet adjon a forgó elemeknek úgy, hogy ne legyen a normál tartományba való váltáskor ugrás.

$$A_{új} = A(err) + G \times e^{-(err-5+e^{-err+5})/10}$$



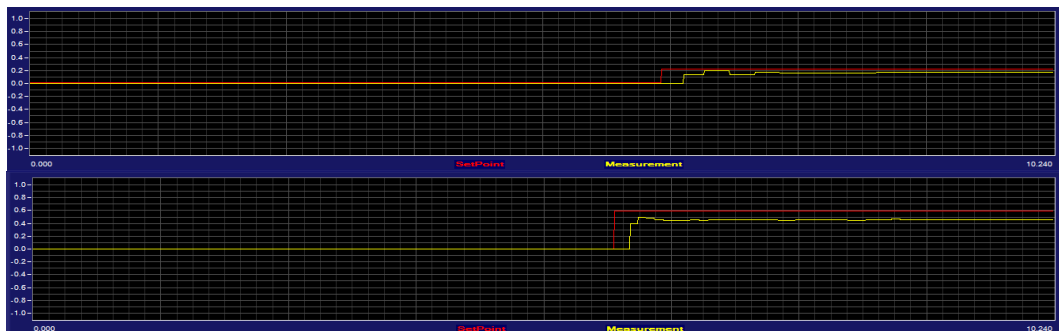
36. ábra A keresett exponenciális függvény(saját rajz)

Természetesen alaposan utána kellett nézni, hogy melyik az a függvény ami ezt a kívánt hatást jól megközelíti. Ez pedig a Gumbel valószínűségi sűrűségfüggvénye. Magát az eredeti

eloszlásfüggvényt elég volt 5-el jobbra tolni és a paramétereken finomítani. Így a G nemlineáris torzításon kell csak változtatni a későbbi teszteken. Az AT32UC3B rendelkezik lebegőpontos számoláshoz szükséges emulátorral, így nagy valószínűséggel ezzel se lesz probléma.

A diagramban (35. ábra), képletben jelölt err hiba pedig lehet akár a szög és akár a fordulatszám is.

Letesztelve mindezt fordulatszám szabályozás esetén, agresszívabb kontrollt alkalmazva:

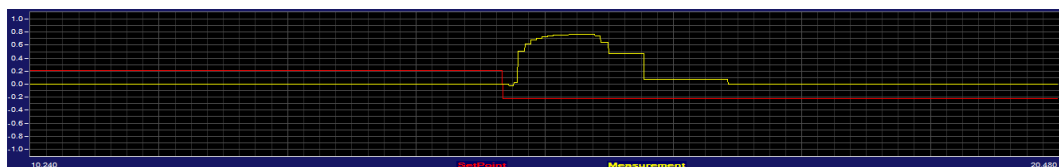


37. ábra: Nemlineáris irányítás alacsony és magas fordulatszámon

Így kis hiba esetén nagy erősítés van, de 0 felé közelítve meredeken csökken és nagy hiba esetén pedig közel az eredeti erősítést kapjuk. Ez azt igazolja, hogy a fordulatszám szabályozással volt gond, nem lehetett elég alacsony fordulatszámot stabilan tartatni a motorral. Gyakorlatilag ez azt jelenti, hogy adunk neki egy nagyobb impulzust, hogy elinduljon. Az I tag 0.06-ra, a P tag pedig 3-ra lett állítva, ami ki lett egészítve az exponenciális függvényvel:

$$P_{új} = 3 + 5 \times e^{-(err-5+e^{-err+5})/10}$$

Ezután letesztelve a pozíciósabályozást, feltételezve, hogy a belső hurok elegendően gyors és stabil:



38. ábra: A pozíciósabályozás tesztelése

A teszteléskor a pozíció is mérve lett, ami megfelelő és gyors, viszont a fordulatszám-mérés több információt tartalmaz. A holtidő sajnos a mérésből fakadóan benne marad, ami inkább

lényeges, hogy a fordulatszám lecsengése ennél gyorsabb, mint a mért eredmény. Emiatt érdemes a mérésen finomítani, illetve esetleg egy végtelen válaszüjű szűrőt is bele rakni a mérésbe szoftveresen.

A szabályozás gyors (1,2~1,5 sec/360°), pontos (+-1°), ráadásul hibatűrő, mivel a fordulatszám-mérés során impulzusokat is kap a motortól, és stabilan visszaáll a referenciaértékre. Elmondható, hogy ez már kiváló alapot nyújt egy jármű irányításának megvalósításához.

7. A modell továbbfejlesztése és a lineáris kvadratikus szabályozás modellezése

7.1 A motor dinamikai modelljének meghatározása

A motor dinamikai modelljének meghatározásával egy pontosabb képet kaphatunk a motorról. Az eddigi mérésekből érdemes kiindulni, ráadásul a gyártó oldalán is megtalálható pár információ a termékről, illetve motorparaméterekkel kapcsolatos diagramokat is meg lehetett találni. Ezeket összevetve még egy ellenőrzési lehetőségünk van arra, hogy az identifikált modell megfelelő-e.

Induljunk ki először is az 5.3 pontban szereplő (1) és (2) egyenletekből. Alkalmazva a Laplace-transzformációt:

$$(J_r s + \mu_m) \Omega(s) = k_m I_r(s)$$

$$(L_r s + R_r) I_r(s) = V_r(s) - k_m \Omega(s)$$

Ezekből az egyenletekből pedig $\Omega(s)$, $I_r(s)$:

$$I_r(s) = \frac{(J_r s + \mu_m) \Omega(s)}{k_m}$$

Ezt felhasználva a második egyenletben, kisebb átalakítás után:

$$\Omega(s) = \frac{k_m}{R_r \mu_m + k_m^2} \times \frac{1}{1 + \frac{J_r R_r + L_r \mu_m}{R_r \mu_m + k_m^2} s + \frac{J_r L_r}{R_r \mu_m + k_m^2} s^2} V_r(s)$$

Ezzel megkaptuk a szögsebességre vonatkozó átviteli függvényt, ami láthatóan kéttárolós alak (2TP). Kicsit átláthatóbbá téve ezt az egyenletet:

$$\Omega(s) = \frac{G}{1 + bs + as^2}$$

ahol:

$a = T^2$ (ahol T a sarokkőrfrekvencia reciproka)

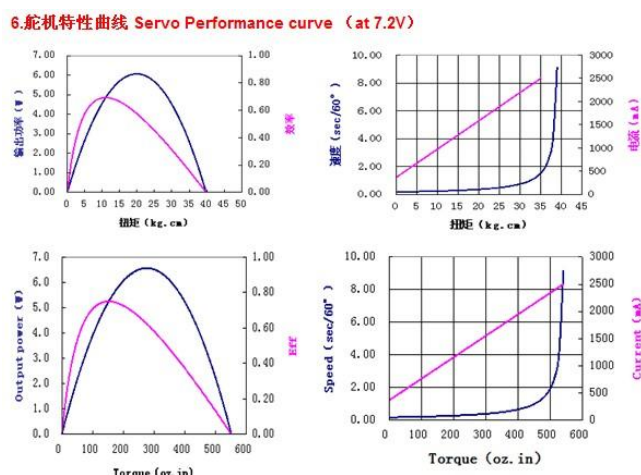
$b = 2T\xi$ (ahol ξ a csillapítás)

A G , a , b értékei értelemszerűen adódnak. Ezek után feltételezve, hogy az egytárolós átviteli függvény meghatározásakor kapott értékeket felhasználhatjuk:

$$G = 23.25 \times 2\pi \approx 146$$

$$b = 0.03 \text{ (Későbbi mérésekből átlagolt)}$$

További adatot szolgáltat még a gyártói oldal is:



39. ábra: A DC motor karakterisztikái (Forrás: xqpower.com)

Lehetőségünk van még a motorral kapcsolatban méréseket végezni a laborban fellelhető digitális multiméterrel, amellyel megmérve a rotor ellenállását és induktivitását több helyzetben is, majd átlagolva:

$$R_r = 2.4 \, \Omega$$

$$L_r = 0.387 \, mH$$

A gyártói adatok alapján meghatározva a nyomatéktényezőt:

$$k_m = 6,5 \times 10^{-3} \frac{N}{A}$$

Ezekkel az adatokkal a motor többi paraméterei is becsülhetők, mégpedig a viszkózus súrlódási tényező és a tehetetlenségi nyomaték (inercia). Ezeket a paramétereket mérés és számolás esetén is fenntartással kell kezelni. A szögsebességre vonatkozó átviteli függvény alapján ezt a kért értéket az eddigi meghatározott paraméterek alapján becsüljük:

$$\mu_m = \frac{k_m}{G \times R_r} \times \frac{k_m^2}{R_r} = 9,354 \times 10^{-7} \, Nms$$

$$J_r = \frac{b(R_r \mu_m + k_m^2) - L_r \mu_m}{R_r} = 5.56 \times 10^{-7} \, kgm^2$$

Ezekből az "a" paraméter már számolható:

$$a = \frac{J_r L_r}{R_r \mu_m + k_m^2} = 4,8 \times 10^{-6}$$

Ezzel igazolva lett az, hogy ha egytárolós tagként kezeljük a rendszert, akkor nem hibázunk sokat. Az 'a' paraméter értéke 4 nagyságrenddel kisebb, mint a b paraméteré. Az állapotter reprezentáció során azonban ezt is figyelembe vesszük. Fontos azonban azt még megjegyezni, hogy a felállított paraméterek alapján hasonló eredményre kell jussunk, ha az új, dinamikai egyenletek útján felállított átviteli függvényvel modelleznénk le egy PI irányítást.

7.2 Áttérés állapotter felírásra

Az állapotter reprezentáció általános alakja:

$$\dot{x} = Ax + Bu$$

$$y = C^T x$$

Fordulatszám szabályozáshoz az állapotter reprezentáció állapotai legyenek a következők:

$$x = \begin{bmatrix} \alpha \\ \dot{\alpha} \\ i \\ z \end{bmatrix}$$

$$\text{Ahol: } z = \int_0^t (r - y) dt \quad \text{és: } \dot{\alpha} = \omega$$

Ezzel egy integráló típusú soros kompenzátort keresünk a visszacsatolásba, biztosítva a jelkövetést. Az állapotok megválasztásánál a rendszer kimenetét választottuk meg első állapotként. A szabályozás elvileg akkor is jelkövető lesz, ha a rendszer kimenetét közvetlenül csatoljuk vissza, azaz egy soros kompenzálást kombinálunk állapot-visszacsatolással. Most általános esetként lesz kezeljük, így kétszeresen integráló tulajdonságú a rendszer, mert a fordulatszámérés kis fordulatszámokon pontatlan és egy integráló taggal a megfelelő jelkövetést valószínűleg nem tudjuk biztosítani. Ráadásul a PI szabályzásnál is bevittünk integráló tagot, és az volt a tapasztalat, hogy érdemes ezt az utat követni.

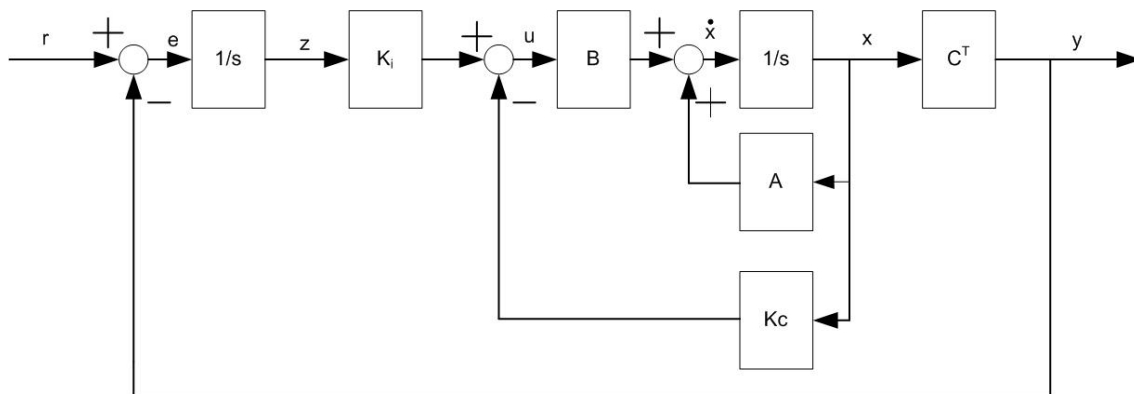
(A,B,C^T) reprezentáció 5.3-ban szereplő dinamikai egyenletek alapján, bővítve:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\mu_m}{J_r} & \frac{k_m}{J_r} & 0 \\ 0 & \frac{k_m}{L_r} & \frac{R_r}{L_r} & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \frac{1}{L_r} \end{bmatrix}$$

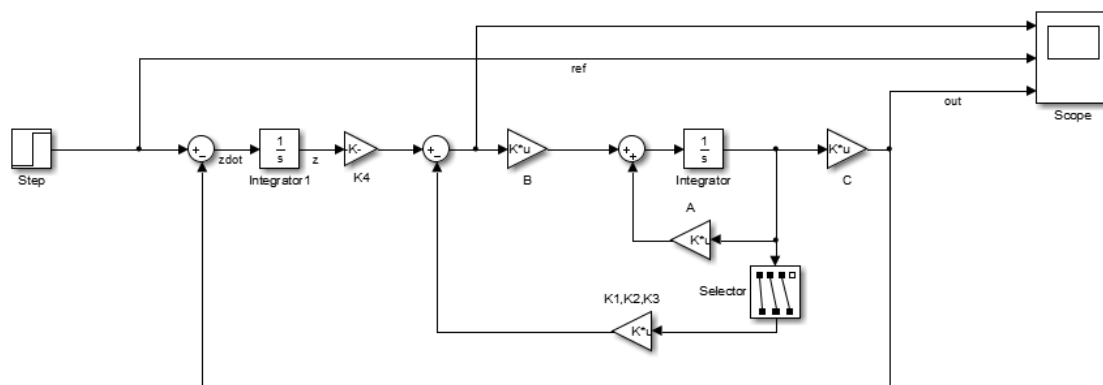
$$C^T = [1 \ 0 \ 0 \ 0]$$

A bővített állapottér-reprezentáció blokkdiagramja:



40. ábra: LQ irányítástervezéshez használt blokkféma(saját ábra)

Ez a jelkövető irányítástervezéshez használt standard blokkféma, amit a továbbiakban használunk. A megtervezett K erősítéset ezen teszteljük. Fontos még az is, hogy minden állapotot mérünk, ezért nem kell állapot-megfigyelőt tervezni a szabályozáshoz.



41. ábra: LQ Simulink blokk

7.3 LQR irányítástervezés

A Matlab segítségével lehetőségünk van ezen tervezési metódus automatizált megoldására. Nézzük a minimalizálandó LQR költségfüggvényt:

$$J(x, u) = \frac{1}{2} \int_0^T [x(t)^T Q x(t) + R u(t)^2] dt$$

A rendszer minőségi tulajdonságait a Q és az R mátrix segítségével tudjuk hangolni. A tervezés nagy hátránya, hogy ezen büntető mátrixok értékeinek megválasztása nem egyértelmű, ezért szokás először a főátlókba 1-et írni és az alapján vizsgálat alá vetni a rendszert. Egy másik lehetőség, hogy ha ismerjük az állapotok korlátjait, akkor az inverz négyzetek módszerét is ki lehet próbálni, és az alapján következtetni a súlyok hangolására. A megoldást a minőségi tulajdonságok közti kapcsolat jelenti, amiket a tervezés során folyamatosan figyelni kell, ezzel tudjuk majd hangolni, finomítani a súlyozást.

Tehát a következő lesz az eljárás:

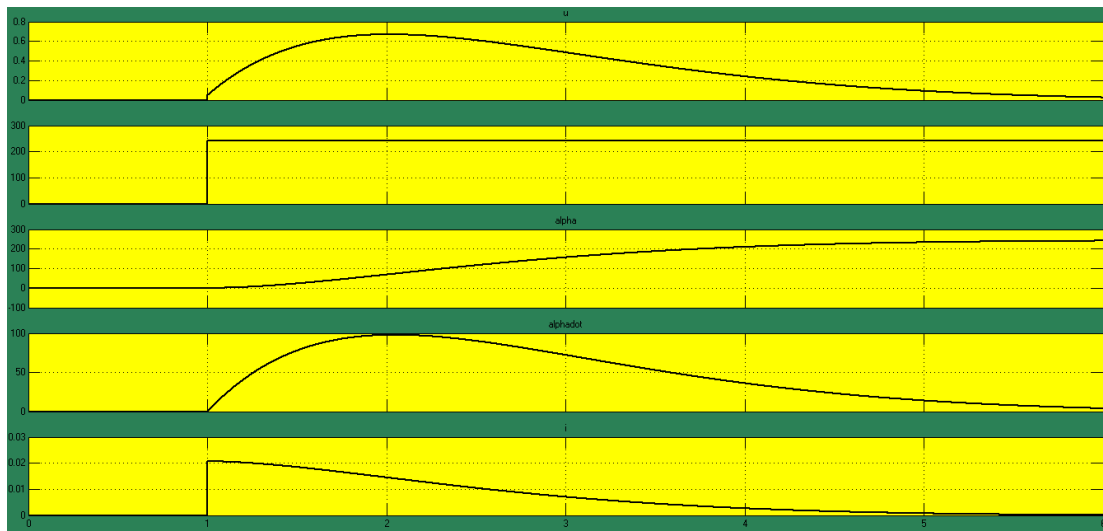
- Irányíthatóság ellenőrzése. A rendszer bővítése miatt érdemes ezt is előtte elvégezni.
- A rendszer a fent említett módon való szimulálása, nagyobb hangsúlyt fektetve az inverz négyzetek módszerére.
- Megfelelő állapotokra zérus súly megválasztása, azaz a követendő kimenetet "szabadjára" engedjük, feltételezve, hogy "józan" referenciajelet alkalmazunk és így futtatjuk le a tesztet.
- A súlyozás hangolása mindaddig, amíg eleget nem teszünk a minőségi követelményeknek.

A felsorolás alapján ellenőrizzük az irányíthatósági mátrix rangját. Ezt Matlab-ban egyszerűen a rank és a ctrb paranccsal tudjuk megtenni:

$$\text{rank}(\text{ctrb}(A, B)) = 4$$

Ez megegyezik az állapottér dimenziójával, így az állapottér reprezentáció minimális.

Nézzük $[q_1, q_2, q_3, q_4] = [1, 1, 1, 1]$ és $[r] = [1]$ súlyozást először (42. ábra).

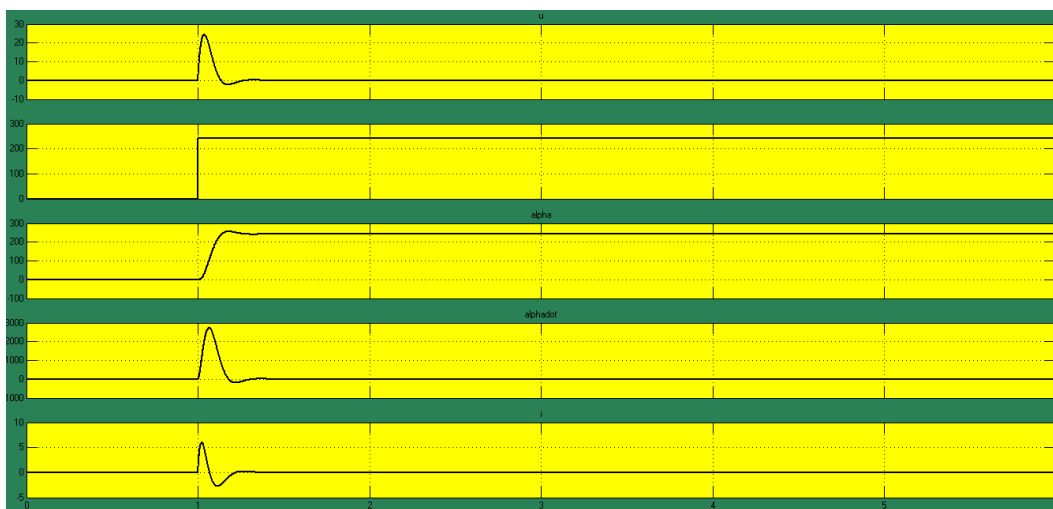


42. ábra: LQ szabályozás $q[1,1,1,1];r[1]$ esetén

A mért jelek fentről lefele a következők:

- Feszültség bemenet a motor kapcsain
- Szög referenciajel
- Motor tengelyén mérhető szöghelyzet (persze a mi esetünkben kimenő tengelyen mérünk, de ezzel felesleges bonyolítani a blokkot)
- Motor tengelyén mérhető fordulatszám
- Motor árama

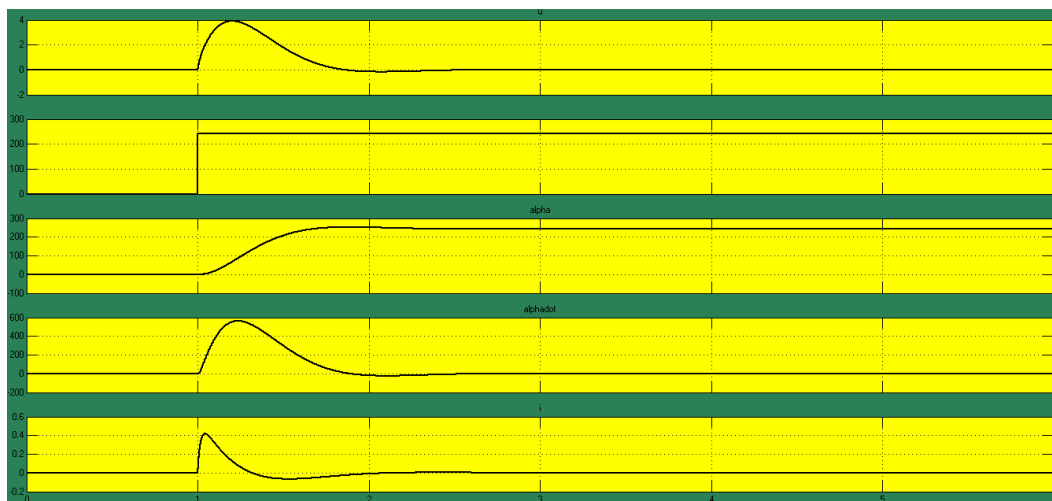
A beállási idő nagy, a szögsebesség értéke alacsony, ebből következik, hogy a bemenet is. Valószínűleg túl nagy büntető súlyokat kapnak az egyes állapotok, de ez várható volt. Az állapotok korlátjai ismertek, így ezen értékeket véve, négyzetre emelve majd invertálva a 43. ábrán látható diagramokat látjuk.



43. ábra: LQ szabályozás $q[0.000001,6.3e-10,0.04,1];r[0.015625]$ súlyozás esetén

Ebben az esetben pedig túl alacsony súlyozások lettek megadva. A feszültség bemenet túl magas és a fordulatszám is. Az áram viszont nem haladta meg a fizikai korlátot. De már megállapítható az, hogy az irányítójelhez, fordulatszámhoz és a "z" állapothoz tartozó súly az, ami a minőségi tulajdonságokat jelentősen befolyásolja. Az áram korlátozása nem érdemes, nem akarunk áram alapján korlátozni, így ezt is akár a "szabadon engedett" állapotok közé sorolhatjuk. Persze ez akár labilitáshoz is vezethet, így lehet, hogy elegendően alacsony súly lesz a célszerűbb. Még a bemenet és a szögsebesség között is megfigyelhető a kapcsolat.

Ezek alapján az eredmény:



44. ábra: LQ szabályozás $q[0,1e-6,1e-8,1e-4];r[1e-100]$ súlyozás esetén

Ez a gondolatmenet bevált. A fordulatszámra és az áramra alacsony súlyokat alkalmaztunk, ezzel kevésbé büntetve azokat a költségfüggvényben. Így a problémát leszűkítettük a bemenet és a "z" állapot hangolására. Ha a bemenetnek nagyobb súlyt adunk, akkor a beállítás lassabb lesz és az irányítójel maximuma is kisebb, ha pedig nagyobbat, akkor egy durvább, gyors beállási idejű irányítást kapunk. A "z" állapottal pedig az integráló hatást tudjuk befolyásolni, azaz ez pont a fordítottja: növelve a beállási idő csökken, csökkentve pedig nő.

A következő szimulációs ábra az eddigiek alapján történő iterációs lépések eredménye:



45. ábra: LQ szabályozás $q[0,1e-6,1e-8,0.01];r[0.01235]$ súlyozás esetén

Az input maximális értéke 6.5V, a fordulatszámnak pedig körülbelül 1000 rad/s. Ezek az értékek még az eszköz tartományain belül vannak. A referenciajel 242 radián egységugrás, mert a fogaskerék áttétel miatt ez a kimeneten pontosan 1 radiánra adódik ki. Az irányítás

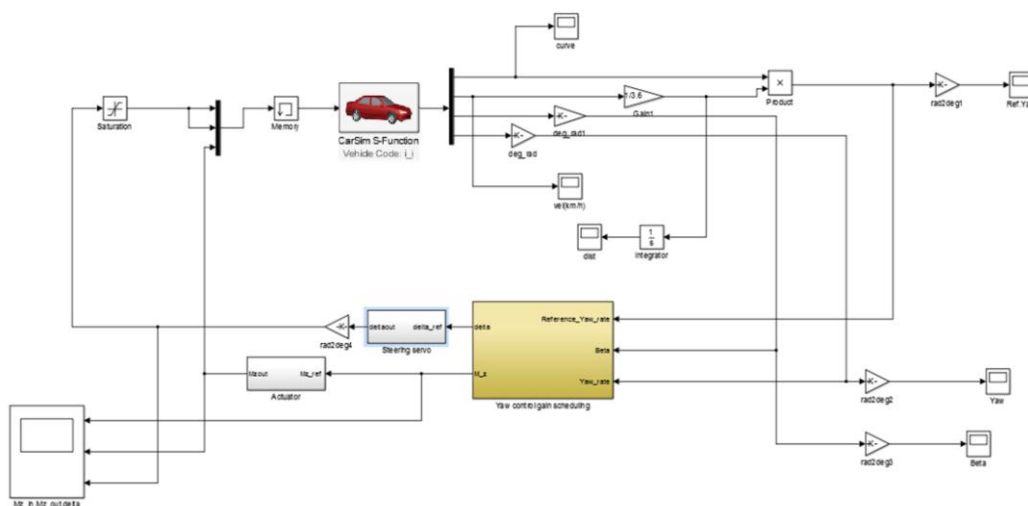
pedig 60 fokos tartományban fog működni. Ha túl is mennénk a tartományokon, akkor sem lenne gond, mert valós környezetben egyszerűen megállna a maximális értékeken. Azonban érdemes a reális tartományon belül maradni a tesztelés során, hogy jó összehasonlítási alap legyen a HIL (hardwer in the loop) tesztelés során. A beállási idő 400ms, ami nem gyorsabb a PI szabályzásnál, viszont nincs túllendülés, ráadásul nem használjuk ki a teljesítményét a motornak, így elmondható, hogy ez a megoldás optimális ebből a szempontból.

A későbbiekben sor kerül ennek a megoldásnak a valós környezetben történő kipróbálására is. Ha ez megtörtént, csak akkor érdemes esetlegesen az egyes paraméter értékeken módosítani a modellben. Tehát a jövőbeni terv ennek a modellnek a tesztelése és tovább hangolása, hogy egy olyan szabályozót kapjunk, ami egy pontos modell-identifikáción alapszik.

8. Eredmények összegzése

8.1 Kitekintés

Mivel már létezik a kormány szervónak egy valóságot megközelítő modellje és a realizációja is megfelelő, így akár a hierarchikus irányítási láncba is beilleszthető és tesztelhető. Ez alatt az értendő, hogy Matlab/Simulink és az úgynevezett CarSim jármű szimulációs szoftver segítségével tesztelhető a járműmodell és a kormány szervó viselkedése adott feltételek mellett. Tekintsünk meg egy ilyen példát:



46. ábra: Modelljármű felső szintű irányítási blokkvázlata (Mihály András tanár úr anyaga alapján)

Ezzel a sémával tesztelhető az agymotorok, a kormány szervó és maga a jármű együttes viselkedése modell szinten. Ráadásul fizikailag is beépíthető és tesztelhető a megtervezett

pozíció szabályzás a modelljárművön is. Mivel jelen tárgyalás a kormány szervó megtervezéséről szolt, így ezen tesztelést nem részletezem.

8.2 Összefoglalás

A megvásárolt szervó egy teljesen új szabályozó elektronikával jelentősen feljavítottuk úgy, hogy a minőségi kritériumoknak is megfeleljen az. Ezenfelül diagnosztikai műveleteket is el tudunk végezni rajta a CAN kommunikációnak köszönhetően. A nagy áttételezés miatt kiváló a modelljármű kormányzására, ráadásul a kis szögváltozásokra lett finomhangolva az eszköz, mivel a gyakorlatban általában a kormányzás nem szokott ugrásszerű lenni, inkább lassabb átmenetek a jellemzőbbek. Terhelés alatt pedig a pozíció is stabil marad.

Irányítási szempontból a PI szabályzás megfelelő, gyors. Az LQ szabályozás még vár a további tesztelésre, de az azonban már most megállapítható, hogy ez a modell jobb megoldást nyújthat akár, mint az előző. Nagy valószínűséggel a legjobb megoldást a robusztus irányítástervezés jelentené. Ugyanis ha vesszük például a motor tengelyén az inerciát(J_r), arra már tudunk egy alsó és felső értéket adni, amely korlátokon belül elhelyezkedik nagy valószínűséggel. Ebből felírhatunk két átviteli függvényt és ezen paraméter változására ezeken a tartományokon belül érzéketlenné tudjuk tenni a rendszert.

9. Ábrajegyzék

1. ábra: Az átervezendő szervó kezdeti állapota (Forrás: xq-power.com)	9
2. ábra: A szervó 3D metszeti ábrája (saját készítés)	10
3. ábra: Szervó NYÁK műszaki rajz. (saját készítés)	11
4. ábra: A szögjeladó elhelyezése	12
5. ábra: Alternatív fordulatszám-méréshez szükséges módosítások	12
6. ábra: Becsatlakozási pontoknak kialakított hely	13
7. ábra: A motor ellenütemű vezérlése (Forrás: Soumelidis Alexandros előadásanyaga)	15
8. ábra: Hídágban folyó áram mérése 1	16
9. ábra: Hídágban folyó áram mérése 2	16
10. ábra: Kész PCB tervek (saját készítés)	18
11. ábra: A szervó összeszerelt állapotban	19
12. ábra: A szögjeladó lekérdezése (Forrás: ams.com)	20
13. ábra: Példa az SPI kommunikációra (Forrás: microchip.com)	21
14. ábra: A négy PWM ütemezése (saját rajz)	23
15. ábra: Zárt szabályozási kör (Forrás: egyetemi jegyzet)	24
16. ábra: $C(s)$ átviteli függvény (Forrás: egyetemi jegyzet)	24
17. ábra: DC motor helyettesítő kapcsolása (saját rajz)	25

18. ábra: Az egytárolós átviteli függvény átmeneti függvénye(saját rajz)	28
19. ábra: DCMcan felhasználó felülete	29
20. ábra: Tesztkörnyezet	29
21. ábra: A méréshez kialakított adatok(saját rajz)	30
22. ábra: Az egységugrás mérési eredménye.....	30
23. ábra: Matlab használata az átviteli függvény megállapításához	31
24. ábra: A bevitt adatsorok.....	31
25. ábra: A mért és a szimulált modell átmeneti függvénye.....	32
26. ábra: Kézi ellenőrzéshez használt adatsor.....	32
27. ábra: Az időállandó leolvasása	33
28. ábra: A fordulatszám szabályozás modellezése	33
29. ábra: Szimulációs eredmények fordulatszám szabályozás esetén	34
30. ábra: A modellezett zárt hurok Bode-diagramja.....	35
31. ábra: Tesztelés valós környezetben	35
32. ábra: A pozíció szabályozás modellezése	36
33. ábra: A szervó fogaskerék áttételezésének vázlata(saját rajz)	37
34. ábra: Szimulációs eredmények pozíciószabályozás esetén	37
35. ábra: Nemlineáris erősítési értékek alacsony fordulatszámon(saját rajz)	38
36. ábra A keresett exponenciális függvény(saját rajz).....	38
37. ábra: Nemlineáris irányítás alacsony és magas fordulatszámon.....	39
38. ábra: A pozíciószabályozás tesztelése	39
39. ábra: A DC motor karakterisztikái (Forrás: xqpower.com).....	42
40. ábra: LQ irányítástervezéshez használt blokk-séma(saját ábra)	44
41. ábra: LQ Simulink blokk	44
42. ábra: LQ szabályozás $q[1,1,1,1];r[1]$ esetén	46
43. ábra:LQ szabályozás $q[0.000001,6.3e-10,0.04,1];r[0.015625]$ súlyozás esetén	46
44. ábra: LQ szabályozás $q[0,1e-6,1e-8,1e-4];r[1e-100]$ súlyozás esetén	48
45. ábra:LQ szabályozás $q[0,1e-6,1e-8,0.01];r[0.01235]$ súlyozás esetén	48
46. ábra: Modelljármű felsőszintű irányítási blokkvázlata (Mihály András tanár úr anyaga alapján) .	49

10. Irodalomjegyzék

- [1]**Dr. Bokor József - Dr. Gáspár Péter:** Irányítástechnika járműdinamikai alkalmazásokkal
- [2]**Dr. Soumelidis Alexandros:** Diszkrét PID szabályzó
- [3]**Dr. Soumelidis Alexandros:** Érzékelők és beavatkozók I.-II. előadás anyagai 2013.
- [4]**Lang András:** Agymotoros hajtású modellautó építése és szabályozása, Tudományos Diákköri Konferencia 2013
- [5]**Ingo Völlmecke:** Parameter Identification of DC Motors
- [6]**Mohammed S. Z. Salah:** Parameters identification of a permanent magnet DC motor
- [7]**Atmel Corporation:** AT32UC3B complete datasheet. www.atmel.com/Images/doc32059.pdf
- [8]**Microchip:** MCP2525 Stand-Alone Can Controller With SPI Interface
<http://ww1.microchip.com/downloads/en/DeviceDoc/21801d.pdf>

[9]AMS: AS5048B Magnetic Rotary Encoder www.ams.com/ger/content/.../AS5048_Datasheet.pdf

[10] Dr. Gáspár Péter: Járműirányítás I.-II. előadás anyagok 2013-2014

[11] Dr. Bauer Péter: Repülőgépek egyszerű referenciajel követő szabályozóinak tervezése LQ Servo módszerrel, Matlab/Simulink környezetben 2009.03

[12]Control Tutorials for Matlab/Simulink:

<http://ctms.engin.umich.edu/CTMS/index.php?aux=Home>

[13]Mathworks: <http://www.mathworks.com/>

11. Melléklet

