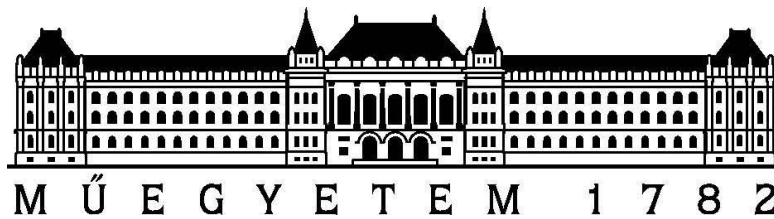


BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
IRÁNYÍTÁSTECHNIKAI ÉS INFORMATIKAI TANSZÉK



Két kontaktus ponttal rendelkező differenciális meghajtású
mobilis robot tervezése, megvalósítása és matematikai
modelljének verifikálása

TDK dolgozat

Odry Ákos
BME-VIK, IIT BSc, 4. évfolyam

Konzulens:
Dr. Harmati István

Tartalmi kivonat

A dolgozat egy differenciál meghajtású robot szerkezet fejlesztési folyamatát mutatja be. A fejlesztés célja egy olyan elektro-mechanikai konstrukció kialakítása, amely a későbbiekben többhurkos (pid, fuzzy, stb.) szabályozók tervezését és implementálását, valamint kritikus egyensúlyozási feladatok megoldását teszi lehetővé. A munka magába foglalja a szerkezet matematikai modellezését-, az elektronikai rendszer kialakítását, a felhasználói- és beágyazott szoftverek megírását, továbbá a szerkezet matematikai modelljének mérésekkel alátámasztott verifikációját.

A fejlesztés során az első feladat a robot matematikai modelljének meghatározása volt. A matematikai modell ismerete lehetővé teszi a robot dinamikus viselkedésének szimulálását, valamint az elméletileg megalapozott szabályozások tervezését. A modell a mechanikai szerkezet CAD rajzából, illetve a robotot meghajtó motorokra-, kerekekre- és a robottestre felírt differenciálegyenletekből tevődik össze. Az egyenletek megoldása és az eredmények kiértékelése MATLAB segítségével történt.

A modellezést az elektronikai rendszer megtervezése, megépítése és felélesztése követte. A robotot meghajtó elektronika a motorvezérlő H-hidakból, a vezeték nélküli kommunikációt biztosító Bluetooth egységből, a számításokat végző mikrokontrolleres felületből, a dinamikai tulajdonságok mérését biztosító szenzor felületből és egy USB-UART interfészből áll. A szenzor felület a pillanatnyi rotoráramot mérő áramkörökből, a forgási irányt és sebességet meghatározó forgó jeladókból, valamint a dőlésszöget meghatározó giroszkóp és gyorsulásmérő szenzorokból áll. A teljes berendezés 3.3V-on üzemel.

A folytatásban a működőképes hardver mikrovezérlős felületének felprogramozása és a felhasználói szoftver megírása következik. A felhasználói szoftver feladata a robot irányítása, valamint a szenzorok mérési eredményeinek rögzítése. A mikrokontrolleren futó beágyazott szoftver a számítógéptől fogadott parancsokat hajtja végre, valamint a szenzorok mérési eredményeit dolgozza fel. Ezt követően a szerkezet teljes dinamikus viselkedése mérhetővé- (mintavételezhetővé-), a matematikai modell pedig verifikálhatóvá válik.

Abstract

This paper presents the development process of a differential driven robot construction. The aim of the development is to build an electro-mechanical construction, which gives the opportunity to design and implement control systems, on the other hand due to the robot's mechanical design, it is capable of solving critical balancing tasks as well. The development process involved mathematical modelling of the construction, building the electronic system, programming the application and the embedded software, as well as the verification of the mathematical model by the help of the measurement results.

During the development process the first task was to define the mathematical model of the robot. Knowing the mathematical model enables simulation of the robot's dynamic behaviour, as well as planning the theoretically grounded control. The model consists of the CAD graphic of the mechanical structure and the differential equations of the engines, wheels and the body of the robot. The equations have been solved and the results have been evaluated using MATLAB.

The next step after the modelling was planning, building and testing of the electronic system. The electronics driving the robot consists of two motor controlling H-bridges, the Bluetooth unit providing wireless communication, the microcontroller interface performing calculations, the sensor interface measuring the dynamic behaviour, and an USB-UART interface for secondary communication. The entire electronic unit operates on 3.3V.

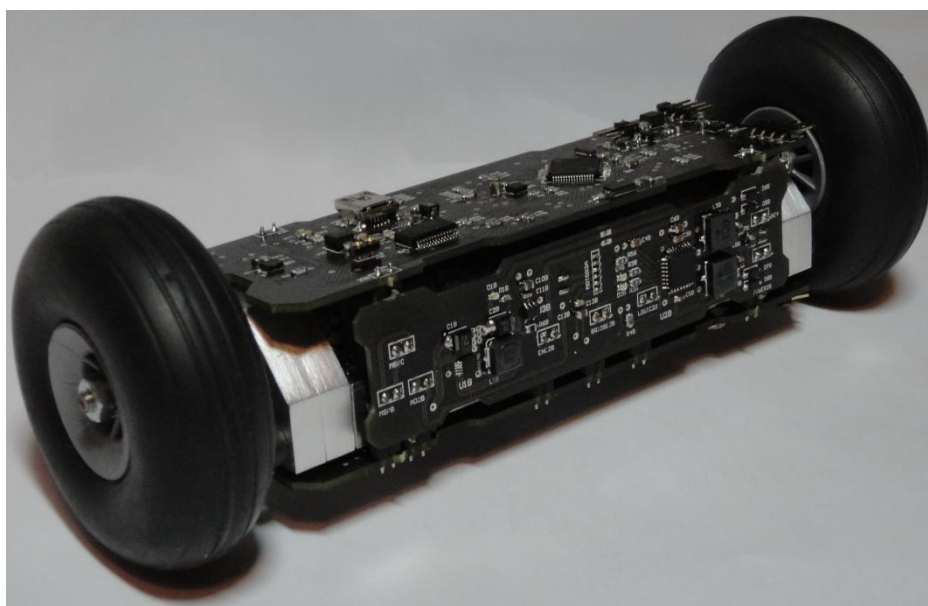
The following step of development will be the programming of the operational hardware's microcontroller interface and the application software. The task of the application software is to direct the robot and to record the measurements of the sensor interface. The embedded software running on the microcontroller executes commands received from the computer, as well as processes signals from the sensor interface. Due to this expansion, the entire dynamic behaviour of the construction will be measurable (sampled) and the mathematic model will be verifiable.

1. Bevezetés

A robotizált rendszerek alkalmazása az iparban egyre intenzívebbé vált az utóbbi években. Ilyenek a gyártósori robotkarok, az űrtechnológiában használt, ismeretlen vagy veszélyes területet felderítő robotok, vagy az olyan önegyensúlyozó, kétkerekű személyszállító mobilis robotok, mint a Segway. A robotok tehát egyre több helyen segítik a különböző feladatok elvégzését. Az emberek és a robotok mind jobban kiegészítik egymást.

A dolgozat egy speciális differenciál meghajtású kerek robot fejlesztési folyamatát mutatja be. A megépített elektro-mechanikai szerkezet két kontaktusponttal rendelkezik, harmadik kereket vagy alátámasztást nem tartalmaz. A mechanikai kialakítás differenciál hajtást valósít meg, így nincs szükség külön kormányzási mechanizmusra, a két külön-külön hajtott kerék biztosítja a haladó mozgást és az elfordulást is. A két kereket összekötő közbenső test átmérője kisebb-, viszont tehetetlenségi nyomatéka nagyobb, mint a kereké. A tömegközéppont a motor tengelyek alatt helyezkedik el, így a robot translációs mozgása során a belső test egy fizikai ingaként modellezhető. A kialakított mechanikai szerkezet egy gyengén csillapított rendszer, melynek stabilizálásában az irányítási és szabályozási algoritmusok kulcsszerepet játszanak. Ezen szerkezettel a többhurkos (*PID*, *fuzzy*) szabályozók tervezését, implementálását és tesztelését szeretném vizsgálni a kritikus egyensúlyozási feladatok megoldására.

A fejlesztési folyamat magába foglalja a szerkezet matematikai modellezését-, az elektronikai rendszer kialakítását, valamint a felhasználói- és beágyazott szoftverek megírását. A dolgozat olvasásával az olvasó végig követheti a robot építés lépéseit, az ötlettől a konkrét megvalósításig (*fizikai egyenletek felírása és matematikai modellezés, áramköri elemek kiválasztása és rendszertervezés, beágyazott szoftverek írása és a robot mozgásának megvalósítása*). A dolgozat végére érve egy működőképes elektro-mechanikai robot szerkezet és a hozzá tartozó verifikált matematikai modell áll rendelkezésre, amely a továbbiakban teljes mértékben alkalmas komplex szabályozástechnikai- és irányítástechnikai feladatok beágyazására és tesztelésére.



1.1. ábra – A megépített robot szerkezet bemutatása

1.1 Irodalomkutatás, a rendszer megépítésének indokoltsága

A kétkerekű mobilis robot platformok közül a témához legközelebb álló szerkezeteket szeretném röviden bemutatni.

1. *Personal Riding-type Wheeled Mobile Platform (PMP) [27]:*

A *PMP* a személyszállító eszközök új koncepciója, melynek részei a két kerék és egy bázis, amelyre a vezető áll. A két kerék független meghajtású (*differenciális hajtás*), az előre és hátra irányuló mozgás, valamint a kormányozás eléréséhez a tömegközéppontját kell megváltoztatnia a bázison álló vezetőnek. A berendezés előnyei: az egyszerű szerkezetnek köszönhetően kisebb az össztömeg, a kormányzó egység nélkül helytakarékos a kialakítás, az utazás irányítása a vezető tömegközéppontja segítségével történik [27].



1.2. ábra - *Personal Riding-type Wheeled Mobile Platform [27]*

2. *Two-wheeled Pendular Mobile Robotic Device [25]:*

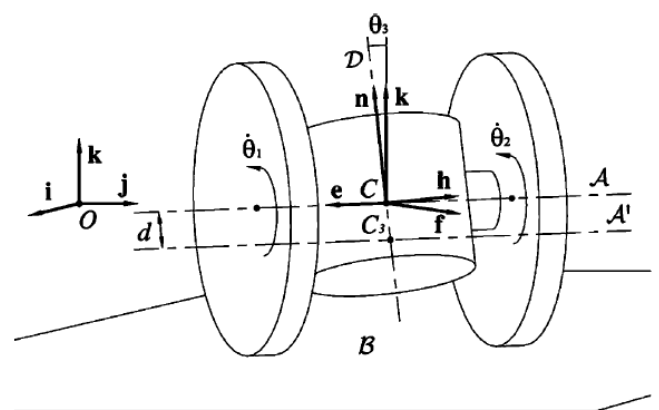
A robotot arra fejlesztették ki, hogy flexibilisebb kültéri navigáció mellett is egy olyan stabil szenzor platform legyen, amely elég erős ahhoz, hogy elbírnjon egy felnőttet. A robot egy 4 mm vastag acéllemez vázból (*kerek között*) áll, amely egy ingát képez. Ily módon a robot képes egyensúlyozni és nagyon stabil helyzetben tartani magát, anélkül, hogy szükség lenne harmadik kerékre. A háromkerékű robotok igen elterjedt szerkezeti megoldások, de hajlamosak elakadni a padló egyenletlenségeinél, például szőnyegek összetételénél, réseknél, vagy a lépcsők előtt. A bemutatott megoldás ezzel szemben, könnyen legyőzi az említett akadályokat, továbbá eltolható vagy eldobható bármely irányba, mivel automatikusan visszaáll egyensúlyi helyzetébe. A kétkerekű robot szerkezet képes ingaként egyensúlyba hozni önmagát, mozogni és manőverezni úgy, hogy a tömegközéppontját megváltoztatva a kívánt irányba gurul. Jellemző rá az alacsonyan elhelyezkedő tömegközéppont, a téglatest alakú vázhoz pedig két nagy hegyikerékpár kerék csatlakozik. A kerekek egyenáramú motorokhoz kapcsolódnak, melyeket egy autóakkumulátor hajt meg, amit a szerkezet alján helyeztek el, hogy az inga nehéz ellensúlyaként szolgáljon [25].



1.3. ábra - Two-heeled Pendular Mobile Robotic Device [25]

3. Quasimoro [28, 29]

A Quasimoro (*QUASIHolonomic MOBILE ROBOT*) a kétkerekű mobilis robotok új családjának első prototípusa. A szerkezet két koaxiálisan elhelyezett kerékből és egy közbenső testből áll, amely a hasznos terhet szállítja. Ezt a családot a *kvázi holonómia* jellemzi, melyet a mechanikai és geometriai szimmetria révén érnek el. A QH robotok olyan anholonóm (*nonholonomic*) rendszerek, ahol a kinematikai korlátozásokból származó Coriolis- és centrifugális erők eltűnnek. A következmény az, hogy a QH robot és a holonom rendszer dinamikus modellje illetve mobilitása megegyezik. A *kvázi holonómia* megkönnyíti az irányítást, így a dinamikai modell sokkal egyszerűbbé válik, mint egyébként.



1.4. ábra – Quasimoro [28, 29]

A szerkezetnek számos lehetséges felhasználása létezik. Ilyenek: az ipari robot anyagmozgatásra, segédeszköz a mozgáskorlátozottak számára, kiszolgálórobot a

házimunkák végzéséhez. A QH robot megépítésének másik jelentős hatása az RHA (*robotics for human augmentation*) területén jelentkeznek. Az RHA a robotikának olyan ága, amely specifikus problémákat kíván megoldani, melyek a fogyatékkal élő felhasználók problémáihoz kapcsolódnak, és melyekben az ember és a robot kiegészítik egymást adott feladaton belül. A Quasimoro-t úgy fejlesztették ki, hogy segédeszközként szolgáljon a mozgáskorlátozottak számára [28, 29].

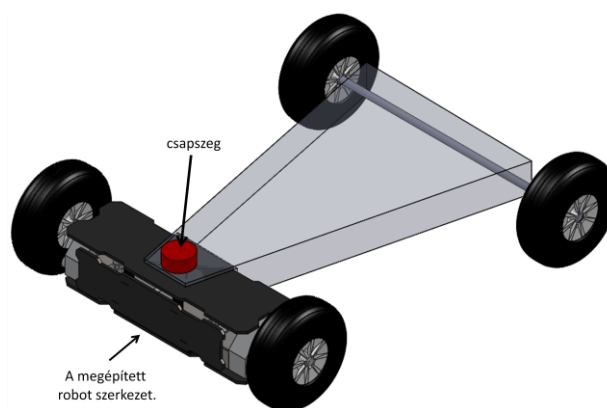
A dolgozatban bemutatásra kerülő mobilis robot nagyon hasonlít a fenti szerkezetekre. A lényeges eltérés az, hogy a lengő test tömegközéppontja és a felfüggesztési pont közötti távolság a robot méreteihez képest lényegesen kisebb (*mindössze 9 mm*), mint a publikált társainál. Ennek következtében instabilabb és érzékenyebb a dinamikai határosokra, egy jó felület kritikus szabályozási algoritmusok hatékonyságának tesztelése. A szerkezet nem csak fizikai ingaként, hanem inverz ingaként is működhet. Mindez mellett, eltolható vagy eldobható bármely irányba, mivel a tömegközéppont mindig visszaállítja az egyensúlyi helyzetébe. A felépítés nyújtotta konstrukciós előnyök (*pl. kis hely csak két kontaktuspont*) lengő rendszert hoznak létre, amelyet szabályozástechnikával lehet kézben tartani.

Egy további ötlet, felhasználási terület:

1. Alkalmazásával az elektromos járművek (*autók*) mechanikai szerkezete és meghajtása nagyságrendileg leegyszerűsödne a hasonló dinamikával rendelkező belső égésű járművekhez képest.

Miért?

Abban az esetben, ha az elektromos autó kötött első tengellyel rendelkező, differenciális meghajtású szerkezet, akkor a kritikus hajtási feladatok kutatására és tesztelésére a dolgozatban szereplő berendezés minden további nélkül alkalmas. A kötött első tengellyel rendelkező négykerekű elektromos autónál ez úgy kivitelezhető, hogy a dolgozatban tárgyalt szerkezet egy csapszegen keresztül kapcsolható az autó testéhez és így a kifejlesztett szerkezet integrált kormánymű- és meghajtóként szerepel rendszerben az *1.5. ábra* alapján.



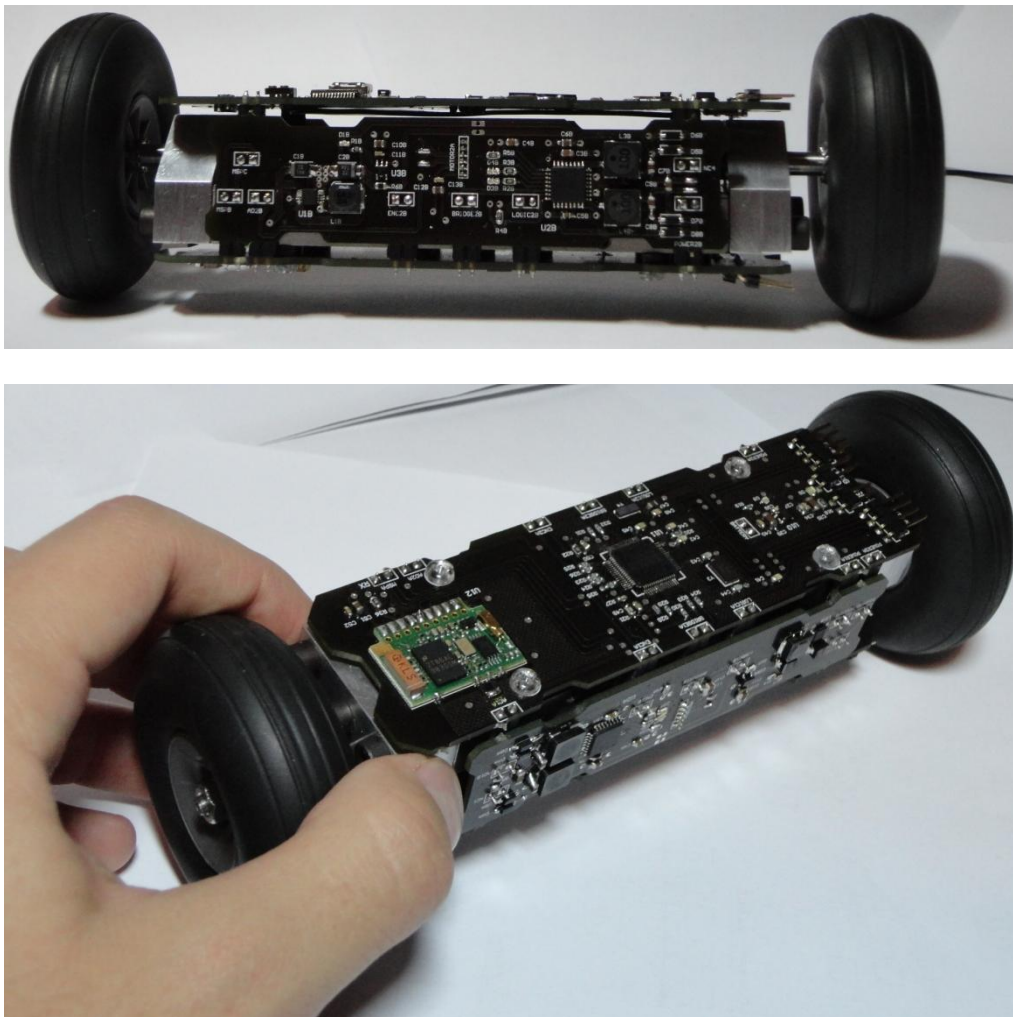
1.5. ábra – A robot, mint integrált kormánymű

Az ábrán látható négykerekű szerkezetben a csapszeg hordozza a legnagyobb terhelést. A hajtás közben jelentkező impulzusszerű terhelések és vibrációk az egész gépkocsi szerkezetébe nézve a legterhelőbbek.

A következőkben ismertetésre kerülő robot szerkezetet a fenti hajtásmechanizmus tanulmányozása is alkalmas. A szerkezet tömegközéppontja a motorok tengelyéhez kellően közel helyezkedik el, de további ellensúly alkalmazásával a távolság még jobban csökkenthető. Ebből kifolyólag a kerekek között elhelyezkedő közbenső test könnyen lengésbe kerül, akár foroghat is a kerekekhez viszonyítva.

Az elvárt dinamika és minimális tehetetlenség eléréséhez kifinomult hajtásokat kell megvalósítani. Ezen hajtások feltárásával a csapszeget érő rángató (*vibráló*) terhelés minimalizálható. Későbbi terveim közé tartozik, hogy ilyen hajtások kutatásával foglalkozzak. A megépített szerkezet folyamatosan tovább fejleszthető, hogy mind jobban közelítsen az elektromos gépkocsiknál jelentkező megoldásokhoz.

A következőkben a két kontaktusponttal rendelkező differenciális meghajtású mobilis robot tervezési- és fejlesztési folyamatát mutatom be. A dolgozat bemutatása során a céлом, hogy egy cél áramkört, egy működőképes mobil robotot építsek meg, és mellékeljem a hozzá tartozó *verifikált* matematikai modellt. Így a következő fejlesztés során az elméletben megalapozott szabályozási- és irányítási algoritmusok tervezésére a matematikai modell-, az implementációhoz és a teszteléshez pedig a megépített szerkezet áll rendelkezésre.

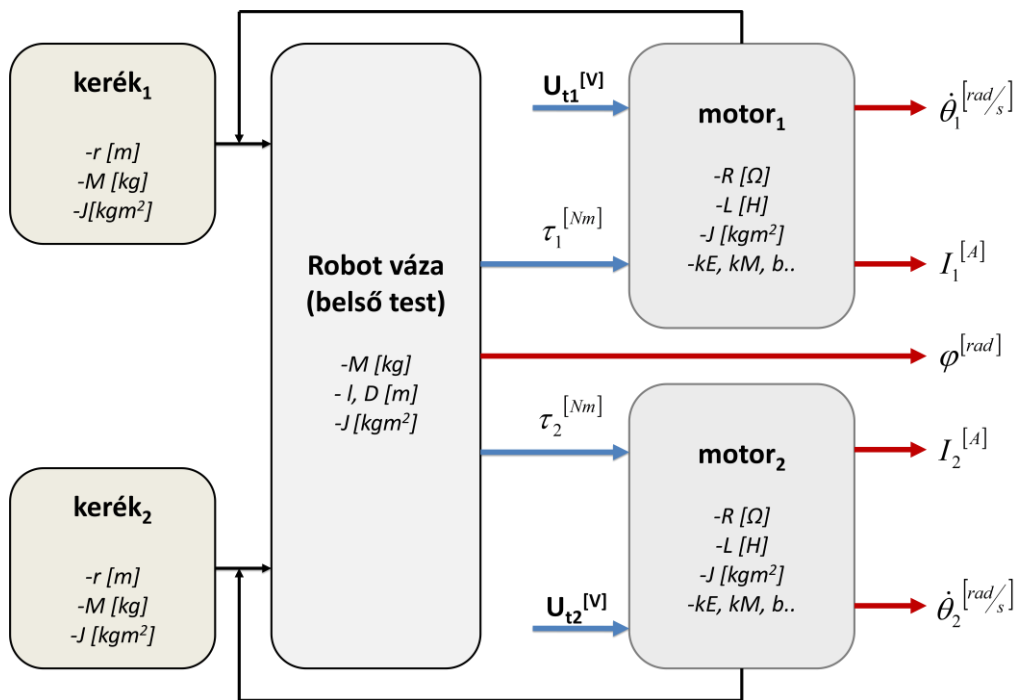


1.6. ábra – A kifejlesztett elektro-mechanikai szerkezet

2. Modellezés

A robot dinamikus viselkedését a kiválasztott motorok és a mechanikai konstrukció határozza meg. A szerkezet megépítése előtt célszerű kiszámítani, hogy milyen dinamikai tulajdonságokat biztosít a választott mechanikai konstrukció. A modellezés előnye, hogy amennyiben a dinamikai eredmények nem felelnek meg az elvárásoknak, még mindig nem késő a mechanikai szerkezeten változtatni. Továbbá a matematikai modell ismerete lehetővé teszi a hatékony szabályozók tervezését.

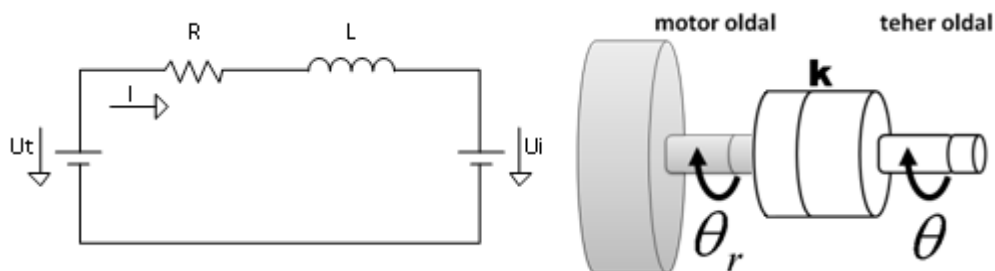
A robot matematikai modellje, a kiválasztott motorok-, a robotot meghajtó kerekek- és a robot vázának (*belső test*) matematikai egyenleteiből tevődik össze.



2.1. ábra – A modellezési feladat ismertetése

2.1 A DC motor matematikai modellje

Az állandó mágnessel gerjesztett, homogén mágneses térben forgó motor helyettesítő képe a 2.2. ábrán látható, ahol a forgórész ellenállása és induktivitása rendre R és L , továbbá a rotoráram I , a motor kapocsfeszültsége pedig U_t .



2.2. ábra – A DC motor szerkezeti vázlata

Lenz-törvény szerint a forgás hatására a mágneses térben U_t -vel szembeható U_i belső feszültség indukálódik. Másrészt *Biot-Savart-törvény* szerint a rotorban folyó áram hatására az árammal arányos erő keletkezik, amely a motor (*motoroldali*) tengelyén τ_r nyomatékot hoz létre [26].

A rendelkezésre álló motor paramétereit:

$R = 8.7 [\Omega]$	a forgórész ellenállása
$L = 28 [\mu H]$	a forgórész induktivitása
$J_r = 4 [nkgm^2]$	a forgórésznek a motor tengelyére vett tehetetlenségi nyomatéka
$b = 100 [pNms]$	a súrlódásból származó veszteségi együttható indukált feszültség állandó, arányos a szögsebességgel nyomaték állandó, arányos az áramerősséggel
$k_E = 1.93 [mVs]$	
$k_M = 1.93 [mNmA^{-1}]$	
$k = 256$	áttétel a motor- és a teher oldal között
τ_r	a motor oldalon keletkező nyomaték a teher oldalon keletkező nyomaték motoroldali szögsebesség teheroldali szögsebesség
τ	
$\dot{\theta}_r$	
$\dot{\theta}$	

A DC motor egyenleteinek felírásához *Kirchoff-* és *Newton-törvényét* kell felírni a 2.2. ábrán látható motor szerkezeti vázlatára.

Kirchoff-törvénye:

Bármely zárt áramhurokban a részfeszültségek előjelhelyes összege zérus.

$$\begin{aligned}
 U_t - U_i &= RI + L \frac{dI}{dt} \\
 U_t - k_E \dot{\theta}_r &= RI + L \frac{dI}{dt}
 \end{aligned}
 \tag{2.1}$$

Newton-törvénye forgómozgás esetére:

A forgómozgást végző test szöggyorsulása egyenesen arányos a testre ható erők forgatónyomatékainak eredőjével és fordítottan arányos a test tehetetlenségi nyomatékával.

$$\begin{aligned}
 J_r \ddot{\theta}_r &= \tau_r - b \dot{\theta}_r \\
 k_M I &= J_r \ddot{\theta}_r + b \dot{\theta}_r
 \end{aligned}
 \tag{2.2}$$

Az (2.1) és (2.2) szerinti differenciálegyenlet rendszer megoldása az egyenletek *Laplace-transzformálásával* a következő:

$$U_t(s) - s k_E \theta_r(s) = RI(s) + sLI(s)
 \tag{2.3}$$

$$k_M I(s) = s^2 J_r \theta_r(s) + s b \theta_r(s)
 \tag{2.4}$$

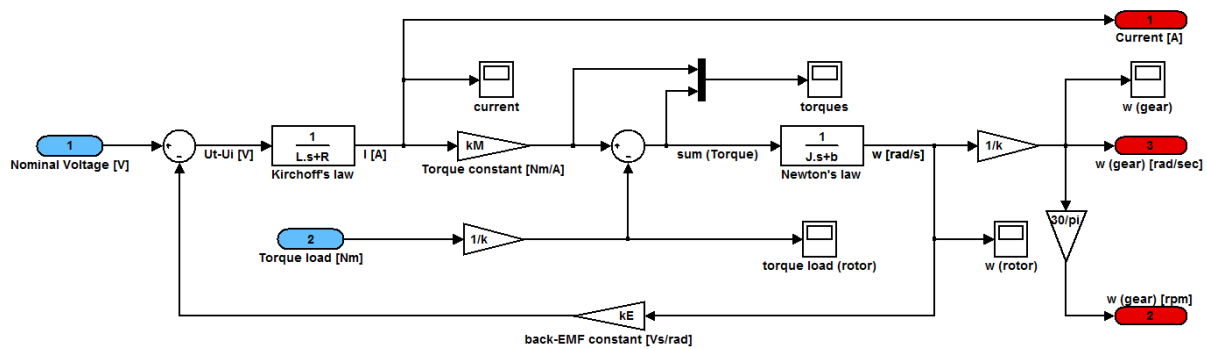
A DC motor egyenlete a (2.3) szerinti áram egyenlet (2.4) szerinti egyenletébe helyettesítésével adható meg.

$$k_M \frac{U_t(s) - s k_E \theta_r(s)}{R + sL} = s \theta_r(s) (b + sJ_r)$$

A fenti egyenlet rendezésével a DC motor hatásvázlatát leíró egyenlet:

$$\left[k_M \frac{U_t(s) - s k_E \theta_r(s)}{R + sL} \right] \cdot \frac{1}{b + sJ_r} = s \theta_r(s) \quad (2.5)$$

A DC motor hatásvázlata:



2.3. ábra – A DC motor MATLAB/Simulink modellje

Az egyenletek levezetése során alkalmazott szabályok:

$$\alpha \left\{ \frac{d}{dt} f(t) \right\} = sF(s) - f(0)$$

$$\alpha \left\{ \frac{d^2}{dt^2} f(t) \right\} = s^2 F(s) - sf(0) - \frac{d}{dt} f(0)$$

A 2.3. ábra a DC motor működését mutatja be. A modellben megjelenő k paraméter teremt kapcsolatot a motor- és teheroldali szögsebességek és nyomatékok között. A motor k áttételén keresztül hajtja meg a tengelyére helyezett terhet. Az áttétel a szögsebességet csökkenti a teher oldalon. A teher- és motoroldali szögsebességek között egy k nagyságú szorzó teremt kapcsolatot a hatásvázlatban, ahol:

$$k = \frac{\dot{\theta}_r}{\dot{\theta}} \quad (2.6)$$

Ideális esetben a motor veszteségmentes. Ekkor a teljesítmények megegyeznek az áttétel két oldalán [26]:

$$\text{teljesítmény} = \text{nyomaték} \times \text{szögsebesség}$$

$$\tau_r \dot{\theta}_r = \tau \dot{\theta} \quad (2.7)$$

A (2.7) szerint, az áttétel a motoroldali nyomatékot növeli a teher oldalon:

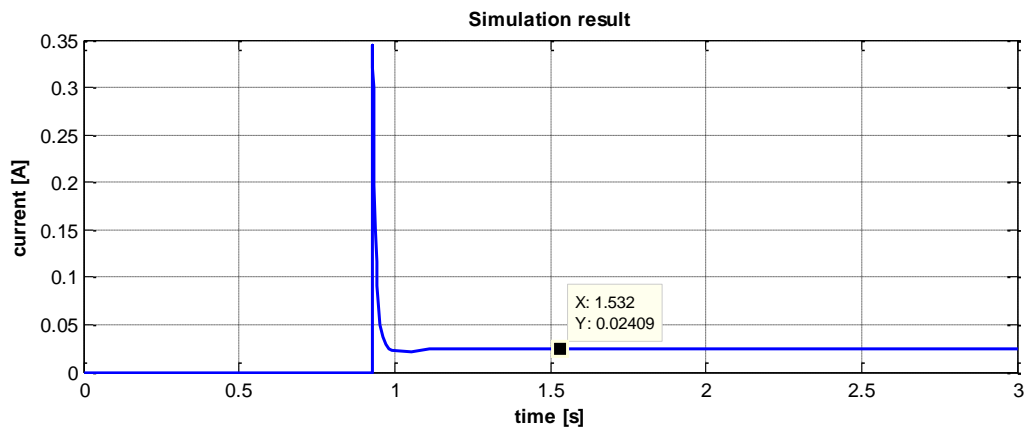
$$\tau = \tau_r \frac{\dot{\theta}_r}{\dot{\theta}} = \tau_r \cdot k, \quad (2.8)$$

tehát, a teheroldali nyomaték az áttétellel egyenesen arányos.

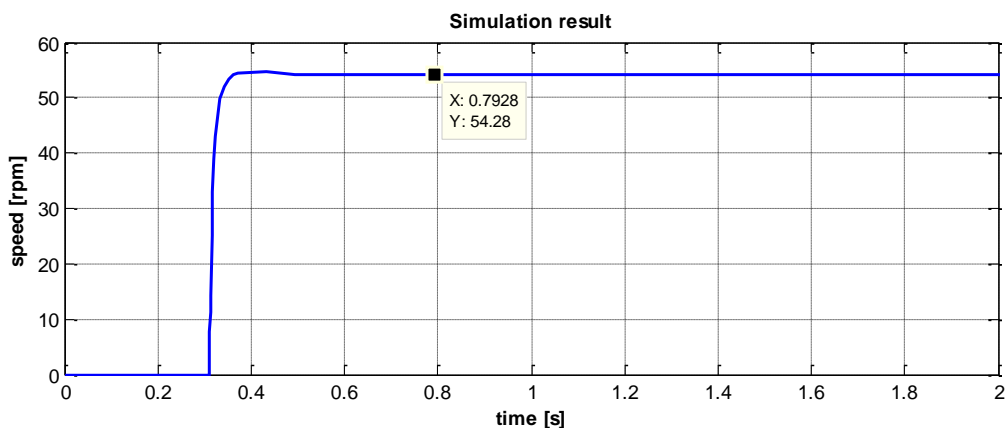
A DC motor matematikai modelljének bemenete a beavatkozó U_t feszültség, valamint a τ_{LOAD} terhelő nyomaték, kimenete pedig a keletkező I rotoráram és $\dot{\theta}$ szögsebesség. A robotot az U_t bemenő feszültség vezérli, melynek megfelelő értékét a beágyazott elektronikai rendszer biztosítja.

A kimeneten megjelenő rotoráram az elektronikai berendezés tervezésénél játszik szerepet. Az elektronikai rendszer tápegységének nagyságát a rotoráram maximális értéke határozza meg. A keletkező szögsebesség és a terhelő nyomaték pedig a további részmodellek (*robot kereke és váza*) egyenleteinek felírásához szükségesek.

Szimulációs eredmények: $U_t = 3V$



2.4. ábra – A keletkező rotoráram



2.5. ábra – A teher oldalon megjelenő maximális forgási sebesség

2.2 A mechanikai szerkezet modellezése

A motorok modellje és működése ismert. A kérdés az, hogy milyen paraméterekkel kell, hogy rendelkezzen a motorokhoz illesztett váz és a motorok tengelyére szerelt kerék, ahhoz, hogy a robot dinamikai tulajdonságai kielégítőek legyenek. A kérdésre a mechanikai szerkezet modellezése ad választ.

A modellezés két lépésből áll. Az első lépés a kerék és a váz megrajzolása CAD rendszerben. A CAD szoftver segítségével kiszámíthatóak a kerék és a váz paraméterei (*tömeg, inercia*). A második lépés a CAD szoftver eredményeinek felhasználása a dinamikai egyenletekben.

2.2.1 A kerekek CAD modellje

A CAD szoftver segítségével megrajzolt kerék és paraméterei a 2.6. ábrán láthatók.

Mass properties of WHEEL

Mass = 31.60 grams
Volume = 50305.09 cubic millimeters
Surface area = 10089.36 millimeters²

Center of mass: (millimeters)

X = 0.00
Y = 0.00
Z = 0.00

Principal axes of inertia and principal moments of inertia: (grams * square millimeters)

Taken at the center of mass.

I_x = 8077.48
I_y = 8077.48
I_z = 14169.81



2.6. ábra – A kerék CAD-modellje és paraméterei

A későbbi egyenletek felírásához szükséges paraméterek:

$$\begin{array}{l|l} r = 3 [cm] & \text{a kerék sugara} \\ M_k = 31.6 [g] & \text{a kerék tömege} \\ J_k = 14169.81 [gmm^2] & \text{a kerék tehetlenségi nyomatéka} \end{array}$$

2.2.2 A belső test (váz) CAD modellje

A robot mechanikai szerkezetének fő része a belső test. Ennek paramétereit úgy kell megválasztani, hogy relatív kis kilengés mellett, az elvárt dinamikus tulajdonságok érvényben maradjanak. A váz minimális kilengését a későbbiekben tervezendő szabályozási algoritmus fogja biztosítani. A CAD-modell és paraméterei a 2.8. ábrán láthatók.

Mass properties of CHASSIS

Mass = 420 grams

Center of mass: (millimeters)

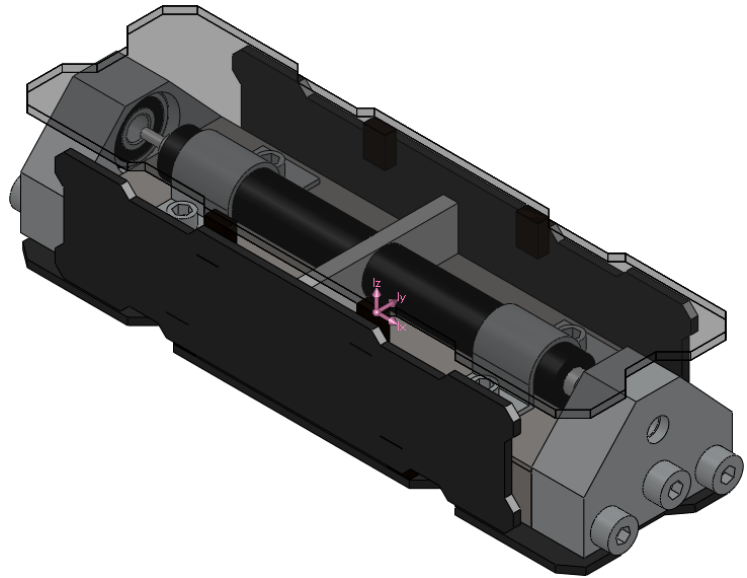
X = 0.00
Y = -8.78
Z = 0.00

Principal axes of inertia and principal moments of inertia: (grams * square millimeters) Taken at the center of mass.

I_x = 72781.43
I_y = 579637.08
I_z = 600885.22

Moments of inertia: (grams * square millimeters) Taken at the output coordinate system.

I_{xx} = 105179.17
I_{yy} = 600885.22
I_{zz} = 612034.82



2.8. ábra – A váz CAD-modellje és paraméterei

A belső test paraméterei:

$$\begin{array}{l|l} l = 8.78 [mm] & \text{a felfüggesztés pontja és a tömegközéppont közti távolság} \\ M_w = 420 [g] & \text{a belső test tömege} \\ J_w = 105179.17 [gmm^2] & \text{a belső test tehetetlenségi nyomatéka} \end{array}$$

2.3 A robot matematikai matematikai modellje

A matematikai modellezés első fázisában az a cél, hogy az előbbieken ismertett mechanikai konstrukció alkalmazásával egy durvább becslést adjunk a dinamikai viselkedésre (*a konkrét megvalósítás előtt*). Tehát nem az a cél, hogy felállítsuk a legkifinomultabb kinematikai és dinamikai egyenletrendszereket (*ez a matematikai modellezés második fázisa – a megvalósítás után*), hanem egy egyszerűsített fizikai elrendezést vizsgálva a legalapvetőbb kérdésekre választ kapjuk.

A két legalapvetőbb kérdés:

- Képes lesz a robot haladó mozgást végezni?
- Ha igen, várhatóan milyen lesz a belső test kilengésének idődiagramja?

Tehát a matematikai modellezés egy referenciát ad (*biztonsági érzetet is*) a robot építő személynek arról, hogy mire számíthat az adott mechanikai konstrukció alkalmazásával.

A két tervezési hiba ismertetése: (*nem szabályozott hatások vizsgálata*)

1. „félelmi” pont

A robot képtelen haladó mozgást végezni, mert:

- A belső test tömege túl könnyűre lett méretezve és a tömegközéppont túl közel esik a motorok tengelyéhez (*felfüggesztési pont*). Ekkor a belső test tehetetlenségi

nyomatéka kisebb lehet a kerek tehetetlenségi nyomatékánál. A következmény adott gerjesztés hatására: a robot egy helyben áll (*mozgásképtelen*), a belső test pedig a motorok tengelye körül forog.

2. „félelmi” pont

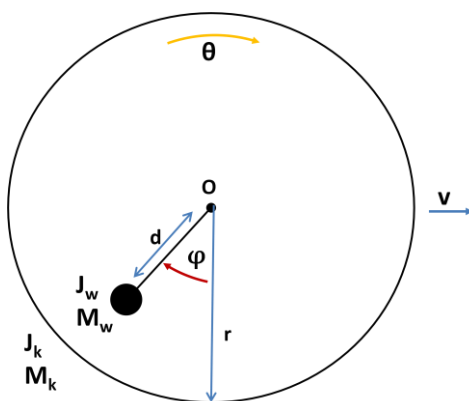
A robot képtelen haladó mozgást végezni, mert:

- A lengő test tömege túl nagy a motor teljesítményéhez képest. Következmény: a robot mozgásképtelen.

A modellezéssel a két félelmi pont közötti optimumot keressük. Az optimum megtalálását az egyszerűsített fizikai elrendezés teszi lehetővé. A következőkben a dinamikus viselkedést leíró egyenletek meghatározására kerül sor.

Tekintsük a következő elrendezést lineáris mozgás esetén:

A két kereket meghajtó motorok gerjesztése egyforma, ezért mindkét kerék θ szögsebességgel forog. A kerekek közötti fizikai ingát M_w tömeg, J_w tehetetlenségi nyomaték és d felfüggesztési pont jellemzi. Az inga lengési szögét φ jelöli a mozgás közben. A motorok által leadott nyomaték τ . Az alkalmazott kerekek sugara r , tehetetlenségi nyomatéka J_k , tömege pedig M_k . A dinamikai egyenletek felírásánál a 2.9. ábrán látható elrendezés szolgál referenciául.



2.9. ábra – Az egyszerűsített modell lineáris mozgás esetén

A dinamikus modellt leíró egyenletek meghatározására a Lagrange-egyenletet használjuk fel, amely alapján, ha $L = K - P$ a Lagrange-függvény és $q = (q_1, \dots, q_n)^T$ jelöli az általánosított koordinátákat, $\dot{q} = (\dot{q}_1, \dots, \dot{q}_n)^T$ az általánosított koordináták sebességét, $\tau = (\tau_1, \dots, \tau_n)^T$ pedig az általánosított erőket, akkor [26]

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i, i = 1, \dots, n. \quad (2.9)$$

Ennek ekvivalens alakja

$$\frac{d}{dt} \frac{\partial K}{\partial \dot{q}_i} - \frac{\partial K}{\partial q_i} + \frac{\partial P}{\partial q_i} = \tau_i, i = 1, \dots, n. \quad (2.10)$$

A fenti egyenletben szereplő K paraméter a test teljes kinetikus energiáját jelöli. A teljes kinetikus energia a test haladási energiájának és a forgási energiájának összegével egyenlő. Definíció szerint:

$$\sum K = K_t + K_r = \frac{1}{2}mv_{TKP}^2 + \frac{1}{2}J\omega^2, \quad (2.11)$$

ahol m a test tömegét, v_{TKP} a tömeg-középpontja sebességét, J a tehetetlenségi nyomatékát, míg ω a szögsebességét jelöli.

A fenti egyenletben szereplő P paraméter pedig a test gravitációs potenciális energiája. Definíció szerint:

$$P = mgh, \quad (2.12)$$

ahol g a nehézségi gyorsulás, az m tömegű test pedig h magasságban helyezkedik el.

A teljes kinetikus energia a 2.9. ábrán látható ingára és kerékre felírt kinetikus energiák összegéből áll:

$$K = K_k + K_w. \quad (2.13)$$

A kerék kinetikus energiája a következő:

$$K_k = K_{k,t} + K_{k,r} = \frac{1}{2}M_k(r\dot{\alpha})^2 + \frac{1}{2}J_k\dot{\theta}^2, \quad (2.14)$$

ahol $\alpha = (\theta - \varphi)$ a kerék szögelfordulása a talajhoz képest. Az $\alpha = (\theta - \varphi)$ helyettesítéssel a (2.15) egyenlethez jutunk:

$$K_k = \frac{1}{2}(M_k r^2 + J_k)\dot{\theta}^2 + \frac{1}{2}M_k r^2 \dot{\varphi}^2 - M_k r^2 \dot{\theta} \dot{\varphi}. \quad (2.15)$$

Hasonlóan a kerékre felírt egyenlethez, az inga kinetikus energiája:

$$K_w = K_{w,t} + K_{w,r} = \frac{1}{2}M_w(r\dot{\alpha})^2 + \frac{1}{2}J_w\dot{\varphi}^2. \quad (2.16)$$

Az $\alpha = (\theta - \varphi)$ helyettesítéssel a (2.17) egyenlethez jutunk:

$$K_w = \frac{1}{2}M_w r^2 \dot{\theta}^2 + \frac{1}{2}(M_w r^2 + J_w)\dot{\varphi}^2 - M_w r^2 \dot{\theta} \dot{\varphi}. \quad (2.17)$$

A (2.15) és (2.17) egyenletek összege a rendszer teljes kinetikus energiáját adja meg:

$$K = \frac{1}{2}((M_k + M_w)r^2 + J_k)\dot{\theta}^2 + \frac{1}{2}((M_k + M_w)r^2 + J_w)\dot{\varphi}^2 - (M_k + M_w)r^2 \dot{\theta} \dot{\varphi}. \quad (2.18)$$

A kerék O középpontját nulla potenciálnak tekintve, a rendszer potenciális energiája:

$$P = -M_w g d \cos \varphi. \quad (2.19)$$

A *Lagrange-egyenletek* használata: általános koordinátákként választható $q = (q_1, q_2)^T := (\varphi, \theta)^T$. A *Lagrange-egyenletekben* szereplő $\frac{\partial}{\partial q_i}$, $\frac{d}{dt} \frac{\partial}{\partial q_i}$ és $\frac{\partial}{\partial q_i}$ deriválások eredményei a következők:

$$\begin{aligned} \frac{\partial K}{\partial \dot{\varphi}} &= ((M_k + M_w)r^2 + J_w)\dot{\varphi} - (M_k + M_w)r^2\dot{\theta} \\ \frac{d}{dt} \frac{\partial K}{\partial \dot{\varphi}} &= ((M_k + M_w)r^2 + J_w)\ddot{\varphi} - (M_k + M_w)r^2\ddot{\theta} \\ \frac{\partial K}{\partial \varphi} &= 0 \\ \frac{\partial P}{\partial \varphi} &= M_w g d \sin \varphi \end{aligned}$$

$$\begin{aligned} \frac{\partial K}{\partial \dot{\theta}} &= ((M_k + M_w)r^2 + J_k)\dot{\theta} - (M_k + M_w)r^2\dot{\varphi} \\ \frac{d}{dt} \frac{\partial K}{\partial \dot{\theta}} &= ((M_k + M_w)r^2 + J_k)\ddot{\theta} - (M_k + M_w)r^2\ddot{\varphi} \\ \frac{\partial K}{\partial \theta} &= 0 \\ \frac{\partial P}{\partial \theta} &= 0 \end{aligned}$$

A fenti eredmények felhasználásával a következő mozgásegyenletek írhatók fel:

$$\frac{d}{dt} \frac{\partial K}{\partial \dot{\varphi}} - \frac{\partial K}{\partial \varphi} + \frac{\partial P}{\partial \varphi} = ((M_k + M_w)r^2 + J_w)\ddot{\varphi} - (M_k + M_w)r^2\ddot{\theta} + M_w g d \sin \varphi = 0, \quad (2.20)$$

$$\frac{d}{dt} \frac{\partial K}{\partial \dot{\theta}} - \frac{\partial K}{\partial \theta} + \frac{\partial P}{\partial \theta} = ((M_k + M_w)r^2 + J_k)\ddot{\theta} - (M_k + M_w)r^2\ddot{\varphi} = \tau. \quad (2.21)$$

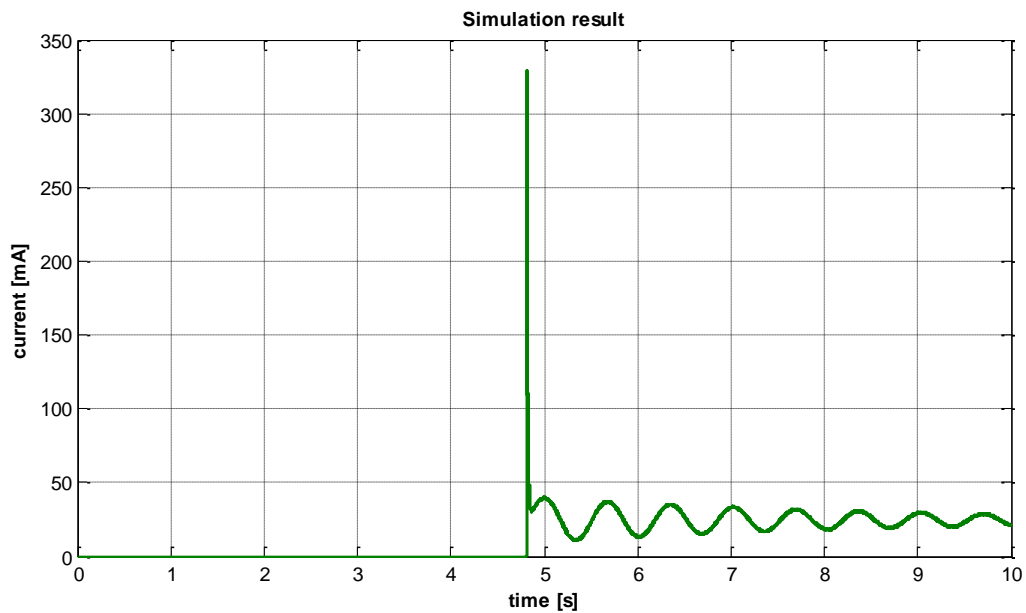
A (2.20) és (2.21) egyenletek MATLAB/Simulink környezetbe implementálásával a robot teljes dinamikus viselkedése szimulálható. Az implementáció során elegendő egyetlen motor modell használata, hiszen az egyszerűsített lineáris mozgás során a két motor szögsebessége megegyezik.

A robot MATLAB/Simulink modelljének:

- bemenete
 - a jobb- és bal motor kapcsolófeszültsége
 - és a kerék-talaj között fellépő súrlódás
- kimenete pedig a keletkező
 - rotoráram, forgási sebesség
 - a lineáris mozgás során megtett út és alkalmazott sebesség
 - valamint a belső test kilengése a translációs mozgás közben

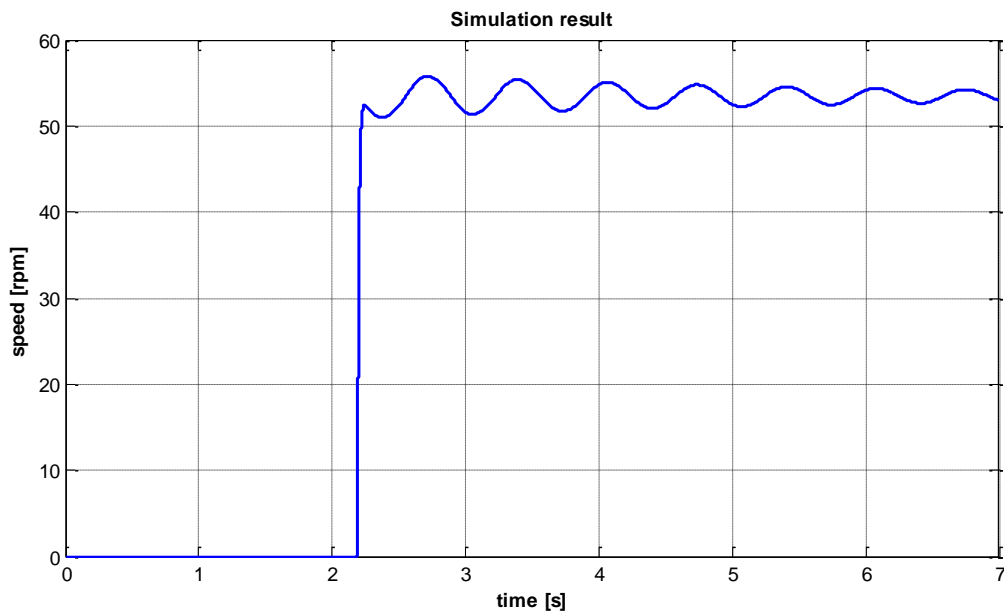
2.4 Szimulációs eredmények ($U_t = 3V$)

2.4.1 A motorok áramfelvétele a robot transzlációs mozgása közben



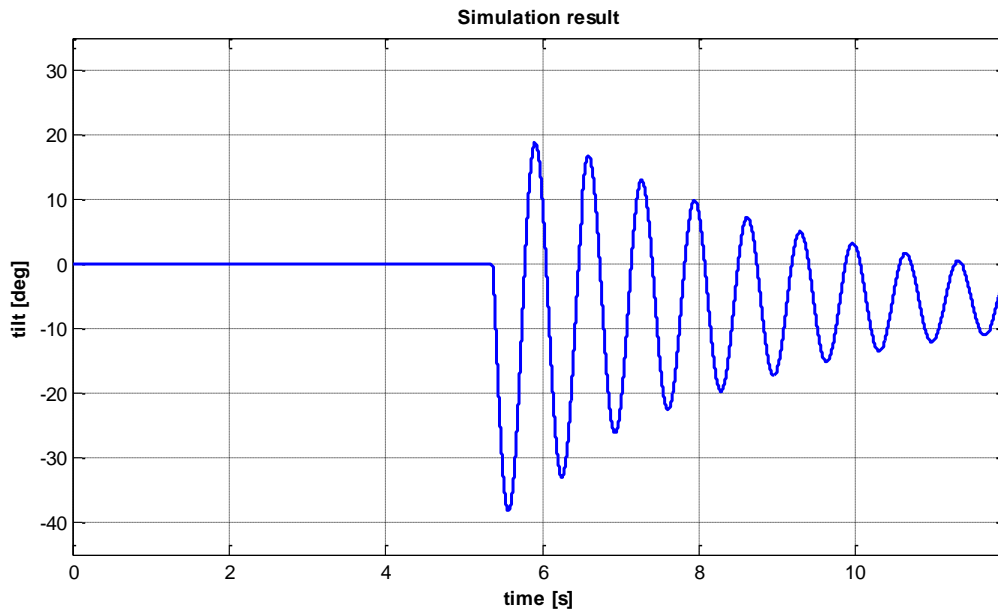
2.10. ábra – Szimulációs eredmény: a robotot meghajtó motorok áramfelvétele

2.4.2 A motorok forgási sebessége a robot transzlációs mozgása közben



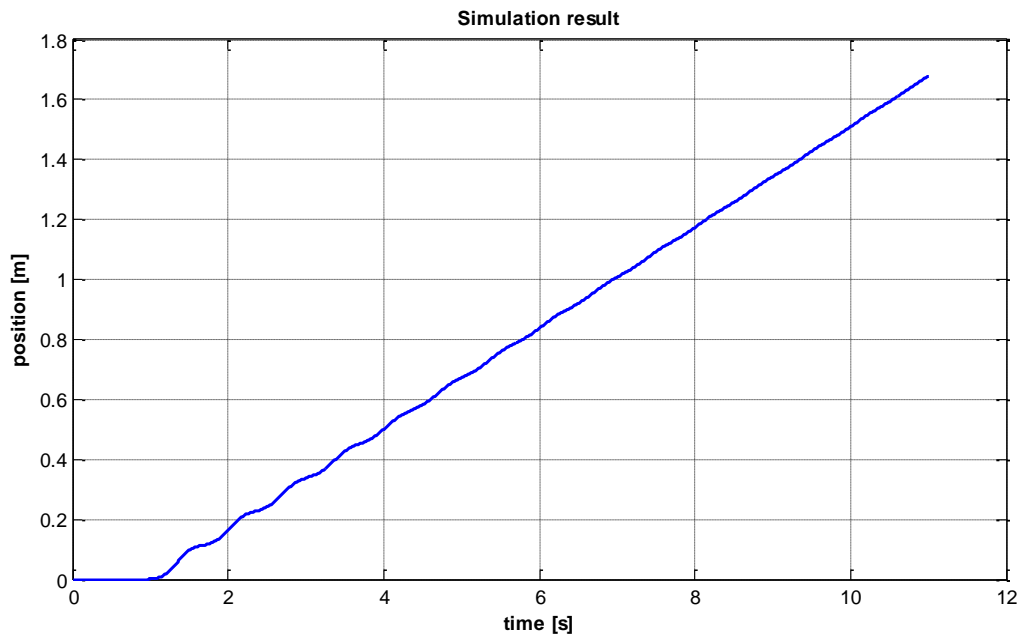
2.11. ábra – Szimulációs eredmény: a robotot meghajtó motorok forgási sebessége

2.4.3 A belső test kilengése a translációs mozgás közben



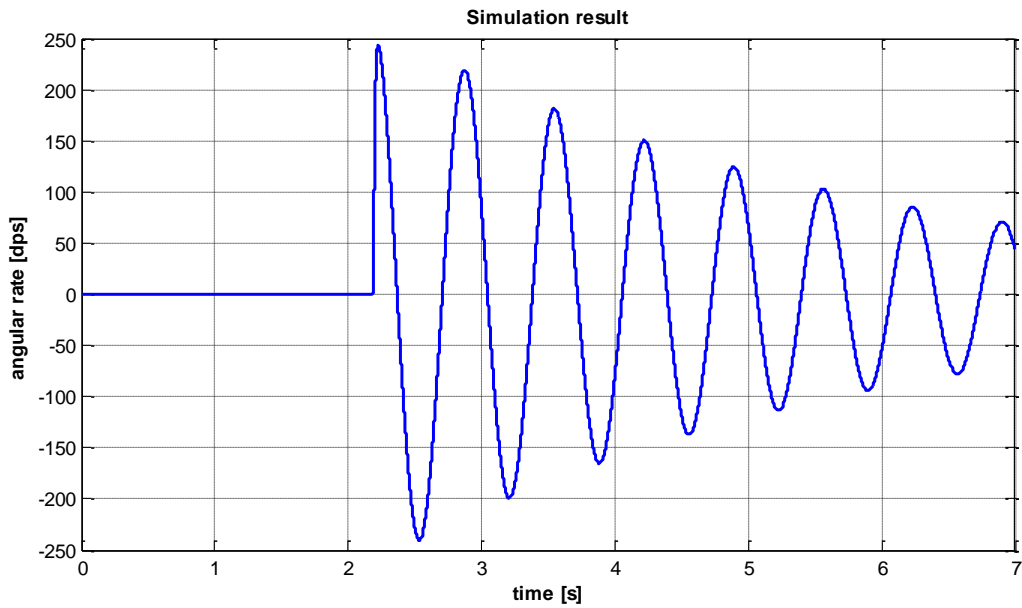
2.13. ábra – Szimulációs eredmény: a belső test kilengésének idődiagramja

2.4.4 A robot lineáris mozgása során megtett út

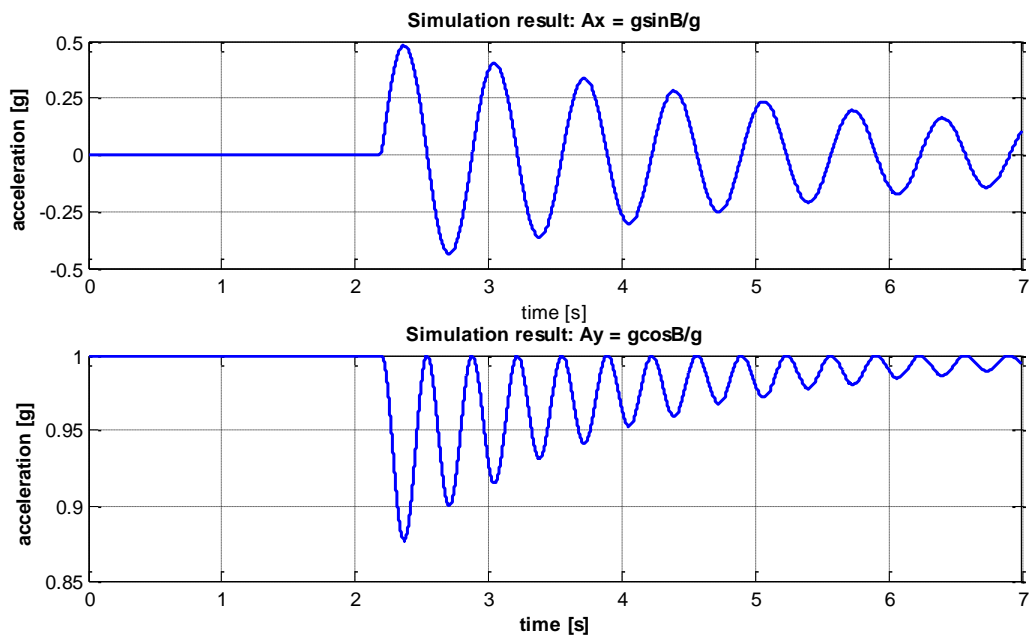


2.14. ábra – Szimulációs eredmény: a lineáris mozgás során megtett út

2.4.5 További szimulációs eredmények



2.15. ábra – *Szimulációs eredmény: a belső test kilengésének szögsebessége (későbbiekben: a megfelelő giroszkóp szenzor kiválasztásánál játszik szerepet)*



2.16. ábra – *Szimulációs eredmény: a belső test kilengése során keletkező gyorsulások (későbbiekben: a megfelelő gyorsulásmérő szenzor kiválasztásánál játszik szerepet)*

2.5 Összegzés

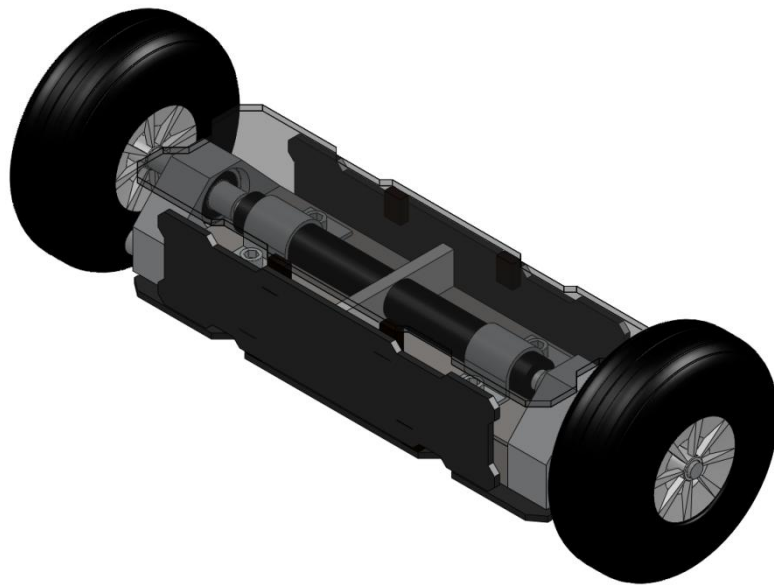
A 2.10. - 2.16. ábrák alapján az a következtetés vonható le, hogy a robot adott vezérlőfeszültség hatására mozgást végez és belső testének kilengése elfogadhatóan kicsi, az idő múlásával lecseng. A dinamikus tulajdonságok maximális bemenő feszültség mellett:

- translációs sebesség: $\dot{x} \cong 16.9 \frac{cm}{s}$,
- csillapodó kilengés: $\varphi \leq 40^\circ$.

További szimulációs eredmények: *(a fejlesztés során még hivatkozunk ezekre az adatokra)*

- a motorok maximális áramfelvétele: $I < 350 \text{ mA}$ *(a tápegységek méretezése),*
- a kilengés maximális szögsebessége: $\omega < 250 \text{ dps}$ *(a giroszkóp kiválasztása),*
- a kilengés során keretkező maximális gyorsulás komponens: $a = 1 \text{ g}$ *(gyorsulásmérő kiválasztása).*

A választott mechanikai konstrukció az elvárt dinamikai tulajdonságoknak eleget tett. A fejlesztés következő lépése az elektronikai- és mechanikai berendezés fizikai megvalósítása, valamint ezek után a robot összerakása és felélesztése.



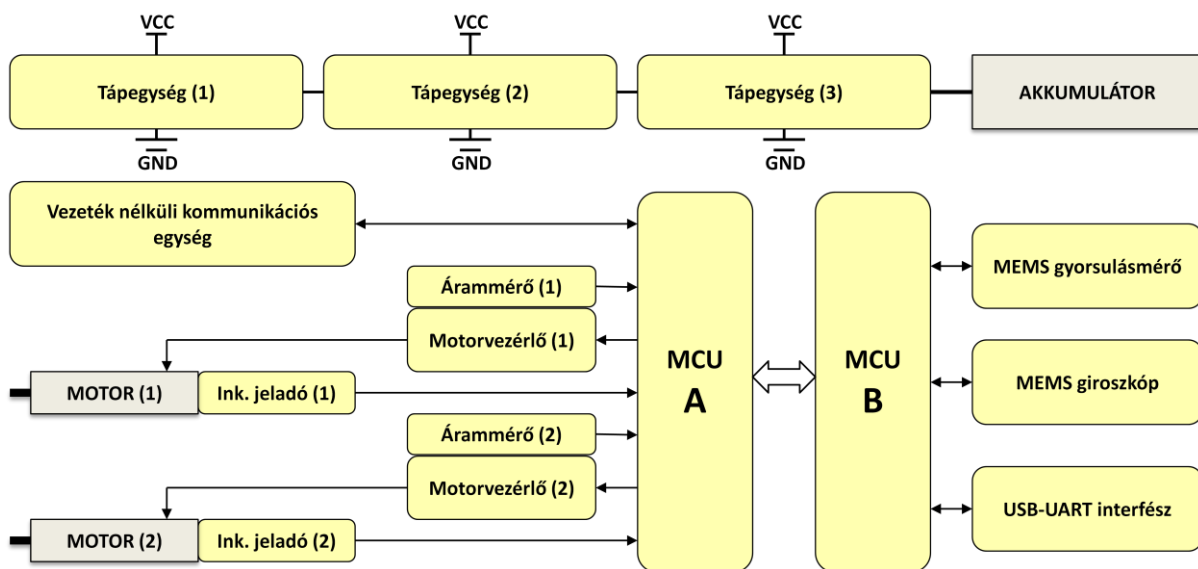
2.17. ábra – Az összerakott robot CAD modellje *(ezt szeretnénk megépíteni)*

3. A robot platform megvalósítása

Ez a fejezet a robot szerkezetet működtető elektronikai berendezés felépítését, a hozzá tartozó beágyazott szoftver működési elvét és a számítógépen futó felhasználói felületet mutatja be. Az áramköri felépítés tárgyalásánál a megvalósítási lehetőségek is ismertetve lesznek.

3.1 Az elektronikai konstrukció

Az elektronikai rendszer felépítése a 3.1. ábrán látható. Az áramköri elemek táplálása akkumulátorról történik. A szabványos stabilizált tápfeszültséget a tápegységek állítják elő az akkumulátor feszültségéből. A beágyazott rendszer két mikrokontroller (*továbbiakban A- és B mikrokontroller*) köré épül fel. A *mikrokontroller_A* adminisztratívabb feladatokkal van felruházva. Ezen feladatok közé a motorok megfelelő vezérlése, az árammérő és inkrementális jeladó jeleinek feldolgozása és a mérési eredmények továbbítása tartozik. A *mikrokontroller_B* a szenzorok mérési eredményeinek feldolgozásán túl egy szűrő algoritmust futtat a belső test pillanatnyi dőlésszögének becslésére. Továbbá a *mikrokontroller_B* felülete egy nagysebességű USB interfészt is tartalmaz a mérési eredmények továbbítására a számítógép felé.



3.1. ábra – A robot platform elektronikai felépítése

3.1.1 A tápegységek

A szerkezet táplálása egy 3.7V-os Li-Po akkumulátorról történik. Ezek az akkumulátorok a Li-Po technológiának köszönhetően nagyobb áramerősséget tudnak leadni, mint a hagyományos NiCd vagy NiMh akkumulátorok és sokkal kisebb (*könnyebb*) méretben is kaphatók. Az akkumulátor három darab feszültség stabilizáló áramkörre (*tápegységre*) csatlakozik. A tápegységek közül kettő a motorvezérlőket táplálja külön-külön, míg egy további a maradó áramköri elemek (*szenzorok, kommunikációs egység, mikrokontrolleres felület*) működéséhez szükséges állandó feszültséget állítja elő.

A tápegység áramkörök ismertetése:

A tápegység áramkörök egyik kategóriájába a lineáris tápegységek tartoznak. Ezek felépítése igen egyszerű, a stabilizátor áramkörön túl alig igényelnek külső elemeket. Lineáris jellegükből adódóan pontos, kis zajszintű tápfeszültséget szolgáltatnak, többnyire túlterhelés és rövidzár védett áramkörök. Alkalmazásuk esetén a legnagyobb figyelmet a rajtuk keletkező veszteségteljesítményre kell fordítani, ami a nyugalmi áram elhanyagolásával:

$$P_D = (V_{IN} - V_{OUT}) \times I_T. \quad (3.1)$$

A fenti teljesítmény valamikor csak valamilyen hűtőfelület biztosításával viselhető el az áramkör számára [1].

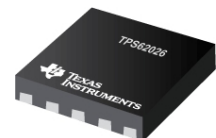
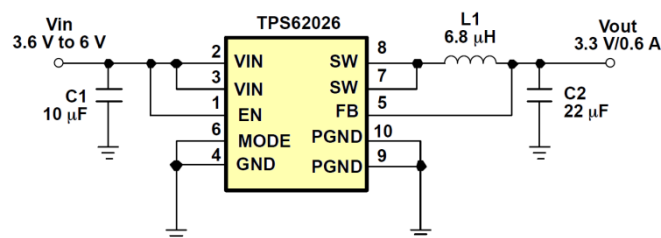
A másik kategóriába a kapcsoló üzemű tápegységek tartoznak. A kapcsoló üzemű tápegységek lényegesen jobb hatásfokkal rendelkeznek. A fix kapcsolási frekvenciával működő kapcsolások ma már egyetlen áramkörbe integrálva állnak rendelkezésre. Kisebb kapcsolások tápegységeként többnyire extra hűtőfelület nélkül is működőképesek, mivel hatásfokuk lényegesen kedvezőbb az előbbieken említett lineáris társaikénál [1].

A tápegység áramkörök kiválasztása:

A motorvezérlők tápegységének méretezését az egyenáramú motorok – mint legnagyobb áramköri fogyasztók teljesítményfelvétele határozza meg. A motorok kapocsfeszültsége 0 – 3V között változtatható, a keletkező rotoráram legfeljebb 400 mA lehet. Ezen határadatok figyelembevételével a Texas Instruments TPS62026-os magas hatásfokú DC-DC konvertere vált alkalmassá a stabilizált tápfeszültség biztosítására. Ez az áramkör kifejezetten kisméretű (*akkumulátoros*) berendezések optimális táplálására szolgál [2].

Az áramkör jellemzői [2]:

- a maximális kimeneti áram 600 mA,
- a bemeneti feszültség tartomány 3.6 – 6 V,
- a kimeneti állandó feszültség 3.3 V,
- védelem a rövidzár és melegedés ellen.



3.2. ábra – A motorvezérlők tápegységének kapcsolási rajza [2]

A harmadik tápegység méretezését a mikrokontrolleres felület köré épülő integrált áramköri elemek (*szenzorok, kommunikációs egység*) összefogyasztása határozza meg.

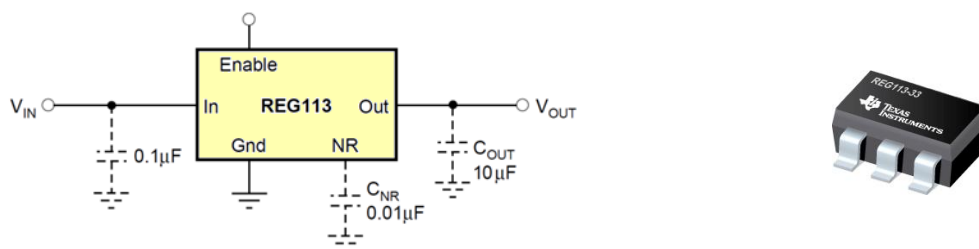
A figyelembe vett határadatok:

- a mikrokontroller köré épülő elektronika tápfeszültsége $3.3V$,
- az elektronikai rendszer összfogyasztása legfeljebb 100 mA lehet.

A fenti adatok figyelembevételével a Texas Instruments *REG113* magas hatásfokú, kiszajú, *kis-dropout* feszültségű lineáris tápegysége került kiválasztásra [3].

Az áramkör jellemzői [3]:

- a maximális kimeneti áram 400 mA ,
- a bemeneti feszültség tartomány $1.8V - 10V$,
- a kimeneti stabilizált feszültség $3.3V$,
- védelem a rövidzár és melegedés ellen.



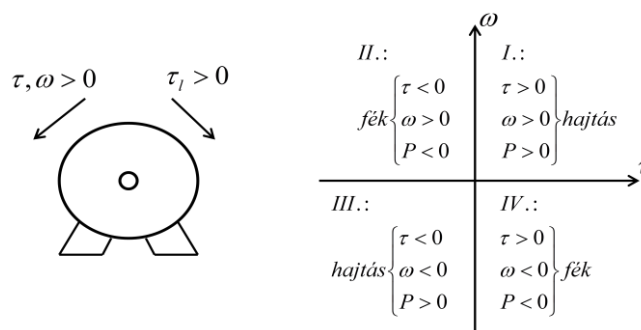
3.3. ábra – A harmadik tápegység kapcsolási rajza [3]

3.1.2 A motorvezérlők

A robot haladásának irányát és sebességét a két motor sebességének különbsége határozza meg. Így a motorvezérlő áramkörökkel szembeni elsődleges elvárás, hogy alkalmas vezérlésük esetén a forgásirány mellett a forgási sebességet is változtatni tudják. Továbbá olyan motorvezérlő áramkört kell kiválasztani, ami kisfeszültségű táplálás mellett biztosítani tudja a motorok működéséhez szükséges áramerősséget.

A villamos hajtások ismertetése [5]:

Legyenek a pozitív irányok a 3.4. ábrának megfelelőek, ahol a motor nyomatékát τ , a szögsebességét pedig $\omega = \dot{\theta}$ jelöli. Az ábrán a motor szögsebességét szabadon vettük fel, míg a motor nyomatékát akkor tekintjük pozitívnak, ha iránya megegyezik a szögsebesség választott pozitív irányával. A τ_l terhelőnyomaték pozitív irányát ehhez képest fordítva vesszük fel [5].



3.4. ábra – A pozitív irányok- és az $\omega(\tau)$ koordinátarendszer negyedeinek jelentése [5]

A pozitív irányokkal a hajtási üzemben a motor mechanikai teljesítménye pozitív, fékező üzemben mindig negatív lesz.

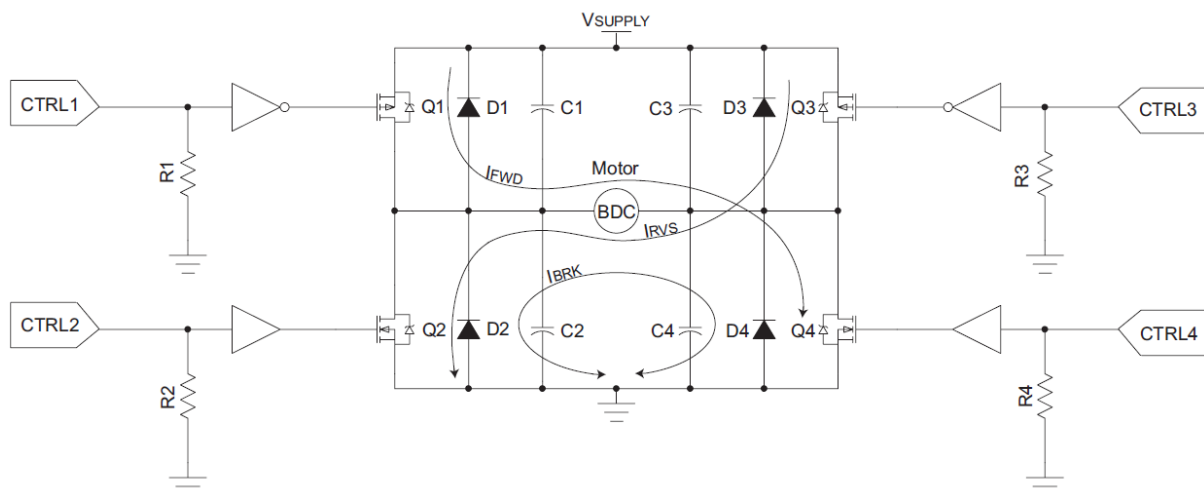
Az $\omega - \tau$ koordináta-rendszer negyedeinek jelentése [5]:

- Az első negyedben $\omega, \tau > 0$, ezért a motor mechanikai teljesítménye $P = \omega\tau > 0$, tehát a motor tengelynyomatéka a forgás irányába hat, a motor teljesítménye a motortól a tengelyre kerül. Ezt a negyedet hajtási negyednek nevezzük [5]. Hasonlóan az első negyedhez, a harmadik negyed is hajtást jelez, de a felvett pozitív irányval ellentétes forgás irányban.
- A második- és negyedik negyedben az ω és τ ellenkező előjelű, a nyomaték a forgási irányval szemben hat, a forgást gátolja, a motor fékez. A motor mechanikai teljesítménye negatív, a motor a tengelyén felveszi a teljesítményt.

Azt, hogy a motornak melyik negyedben kell tudnia üzemelni, a hajtott berendezés szabja meg. Pl. a ventilátor hajtásakor elegendő, ha a motor csak egy forgási irányban tud hajtani, de nem szükséges a motorral fékezni, elegendő tehát az első negyedben való ütemet megvalósítani. Szerszámgépek esetében rendszerint két forgásirányban kell hajtania a motornak. Ha a fékezés nem szükséges, akkor a motornak az első és a harmadik negyedben való üzemet kell lehetővé tennie (*kétnegyedes hajtás*). Ha a forgásirány-változásokat villamos fékezéssel akarjuk forszírozni, akkor négynegyedes hajtás szükséges [5].

A motorvezérlő áramkörök kiválasztása:

Az előbbieken ismertetett hajtások alapján, a robotot meghajtó motorokat négynegyedes üzemben szeretnénk használni. Ezt a hajtást ún. H-hidas hajtásnak nevezzük, a hajtással mindkét irányban lehet fékezni és hajtani a motorokat. A kapcsolás működése a 3.5. ábra alapján a következő.



3.5. ábra – A négynegyedes, H-hidas hajtás ismertetése [4]

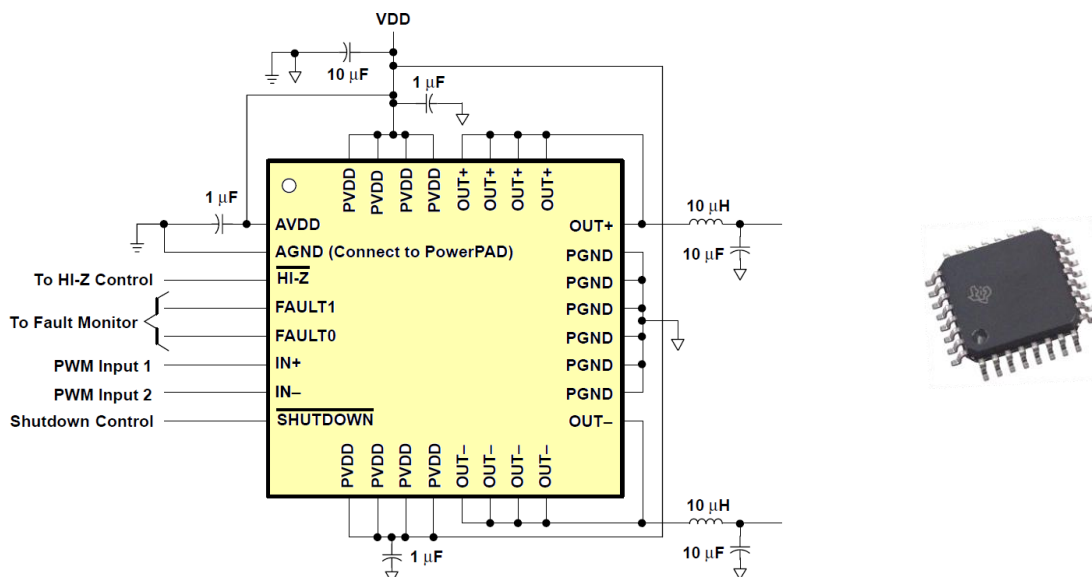
Példa az előre hajtásra:

- A $Q4$ MOSFET folyamatos bekapcsolt állapotban van, a $Q1$ MOSFET pedig PWM-el vezérelt. A kialakuló áram irányát I_{FWD} mutatja (*bekapcsolt állapotban*).
- A megforgatott motor fékezéséhez $Q1$ és $Q3$ MOSFET-ek kikapcsolt állapotban vannak, $Q4$ MOSFET folyamatosan bekapcsolt állapotban van, $Q2$ MOSFET pedig PWM jellel vezérelt.

A robot motorvezérlőinek a Texas Instruments $DRV592$, magas hatásfokú és nagyáramú H-híd áramkörei kerültek kiválasztásra [6].

Az áramkör jellemzői [6]:

- alacsony tápfeszültség $2.5 - 5.5 V$ között,
- magas kimeneti áramerősség $\pm 3 A$,
- külső vezérlés impulzus szélesség modulációval (PWM),
- nyugalmi áramerősség $1 mA$, aktív $5 mA$,
- védelem a túlterhelés és melegedés ellen.



3.6. ábra – A motorvezérlő kapcsolási rajza [6]

3.1.3 A kommunikációs egység

A szerkezet környezetével rádióhullámon kommunikál, iránytartását és sebességét közös instrukcióval kell megadni. A parancsok küldése és az eredmények fogadása (*feldolgozása*) számítógépen történik. A feladat megoldásához két kommunikációs egység illesztése szükséges, egy a mikrokontrolleres felülethez és egy a számítógép mellé.

A számítógépen egy felhasználói program fut (*ismertetése a későbbiekben*), ez a robotot irányítani, valamint a tőle érkező adatokat fogadni és feldolgozni képes. A működtető program egy virtuális soros portot használ, a portról olvassa az érkező- valamint a portra írja ki a küldendő adatokat. A soros port megvalósítására többféle megoldás létezik. Ebben az alkalmazásban egy általános célú Bluetooth adapter szolgál a számítógép-oldali

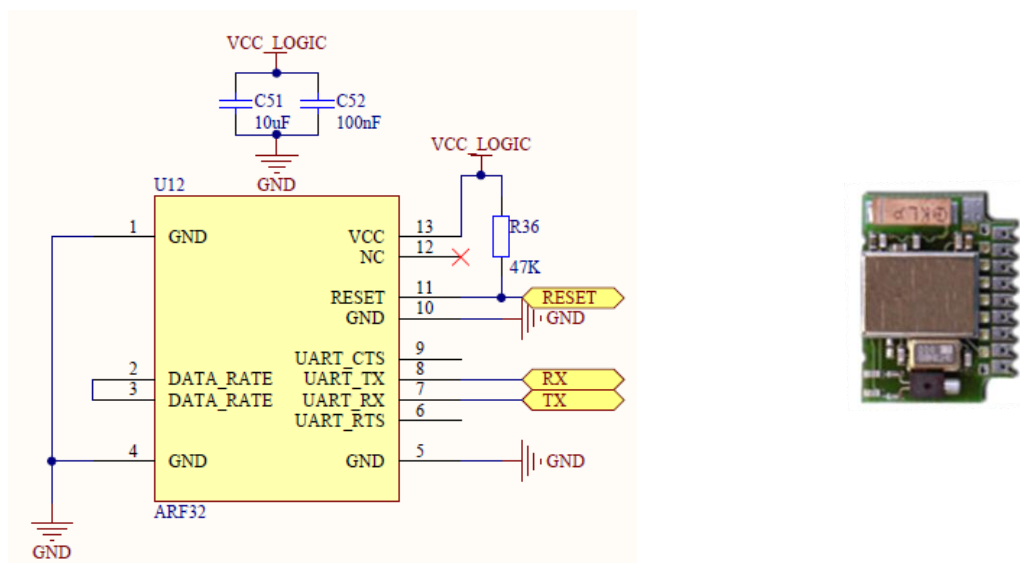
kommunikációs egység illesztésére. A Bluetooth adapter egy vezeték nélküli soros portot (*UART*) valósít meg, így nincs szükség külön hardvert építeni és azt a számítógéphez illeszteni (pl. *Bluetooth modul és FTDI chip-el kialakított áramkör*).

A robot oldalán a mikrokontrolleres felülethez egy Bluetooth modul illesztése szükséges. Az eszközzel szembeni elvárás, hogy soros port perifériával rendelkezzen (*UART*), és hogy a tápfeszültségét a kiválasztott tápegységek biztosítani tudják. Továbbá a robot mechanikai konstrukciójából következik, hogy a modul integrált antennával kell, hogy rendelkezzen (*hatótávolsága legalább 15 m*). A soros kapcsolat mindössze két vezetékot igényel (*rendre RX és TX vonalak*), ezek a mikrokontroller megfelelő periféria lábaira (*rendre TX és RX*) csatlakoznak. A környezettől érkező adat a mikrokontroller soros portjához tartozó megszakítási rutinban kerül feldolgozásra.

A robot-oldali kommunikáció biztosítására, a fenti feltételek kiértékelésével, az Adeunis *ARF32* Bluetooth modulja vált alkalmassá a feladatra [7].

Az eszköz jellemzői [7]:

- tápfeszültség 3.3 V,
- áramfelvétel 70 mA,
- full duplex Bluetooth adatátvitel,
- integrált antenna,
- hatótávolság 20 m,
- soros interfész 723kbps sebességig.



3.7. ábra – A kommunikációs egység kapcsolási rajza

3.1.4 A dinamikai tulajdonságok mérése

A robot dinamikus viselkedését a motorok áramfelvétele és sebessége, valamint a belső test kilengésének nagysága határozza meg. Ezen mennyiségeket kell mérésekkel meghatározni a második fejezetben bemutatott matematikai modell verifikációjához. (A későbbi fejlesztések során: a rendelkezésre álló mérési eredményekkel többhurkos kaszkád szabályozók tervezése és implementációja valósítható meg.)

A robotot működtető elektronikai berendezés négy fajta szenzort tartalmaz a dinamikai tulajdonságok mérésére. Ezen szenzorok közül az *A mikrokontrollerhez* az árammérő- és a sebességmérő szenzorfelület, míg a *B mikrokontrollerhez* a dőlésszög becsléséhez szükséges gyorsulásmérő- és giroszkóp mérőfelület tartozik (3.1. ábra alapján). A következőkben az említett áramkörök bemutatására kerül sor.

3.1.4.1 Az áramfelvétel mérése

A legelterjedtebb árammérő szenzorokat az 3.1. táblázat foglalja össze [8].

Árammérő szenzor	Jellemzők
Sönt ellenállás	pontos mérés, olcsó
Hall-effektus elvén működő árammérő szenzor	az áramkör a jelkondicionálást is tartalmazza, drága, pontatlanság a hőmérséklet-változás és zavaró mágneses fluxusok miatt
Árammérő transzformátor	AC mérési lehetőség (<i>erősáram</i>), galvanikusan izolált mérés

3.1. táblázat – A leggyakoribb árammérő szenzorok ismertetése

Az árammérés ismertetése:

Az *árammérő transzformátor* áttétele a primer- és szekunder tekercs menetszámával állítható be. Az áramátétel szerint, a szekunder oldalon folyó áramot a primer oldalon folyó áram és a menetszám áttétel szorzata adja meg:

$$I_2 = I_1 \times n, n = \frac{N_1}{N_2} \quad (3.2)$$

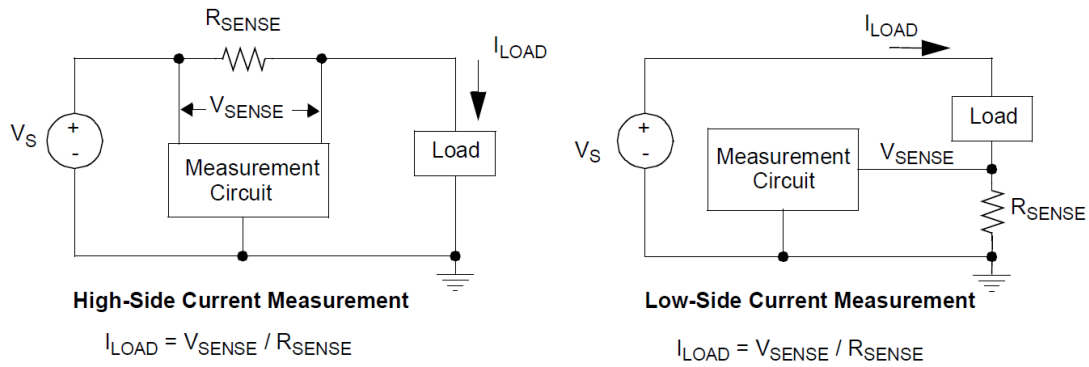
A mérni kívánt áramerősség a szekunder oldal terhelő ellenállásán keletkező feszültségből számítható ki.

A *Hall-szenzorok* működési elve azon alapul, hogy ha egy félvezető lapkában áram folyik és a lapka síkjára merőlegesen mágneses tér van jelen, akkor a tér és a mozgó töltések kölcsönhatása révén a lapkán az áram irányára merőlegesen a mágneses térrel arányos feszültség (*Hall-feszültség*) keletkezik. A Hall-feszültség a mérendő árammal arányos [9].

A *sönt ellenállás* használata során az árammérés feszültség mérésre vezethető vissza. A keresett áramerősség egyszerűen megkapható a mért feszültség és az ismert értékű sönt ellenállás hányadosaként.

A sönt ellenállás elhelyezkedésétől függően két különböző árammérési módszert különböztetünk meg:

- felsőoldali árammérés,
- és alsóoldali árammérés [8].



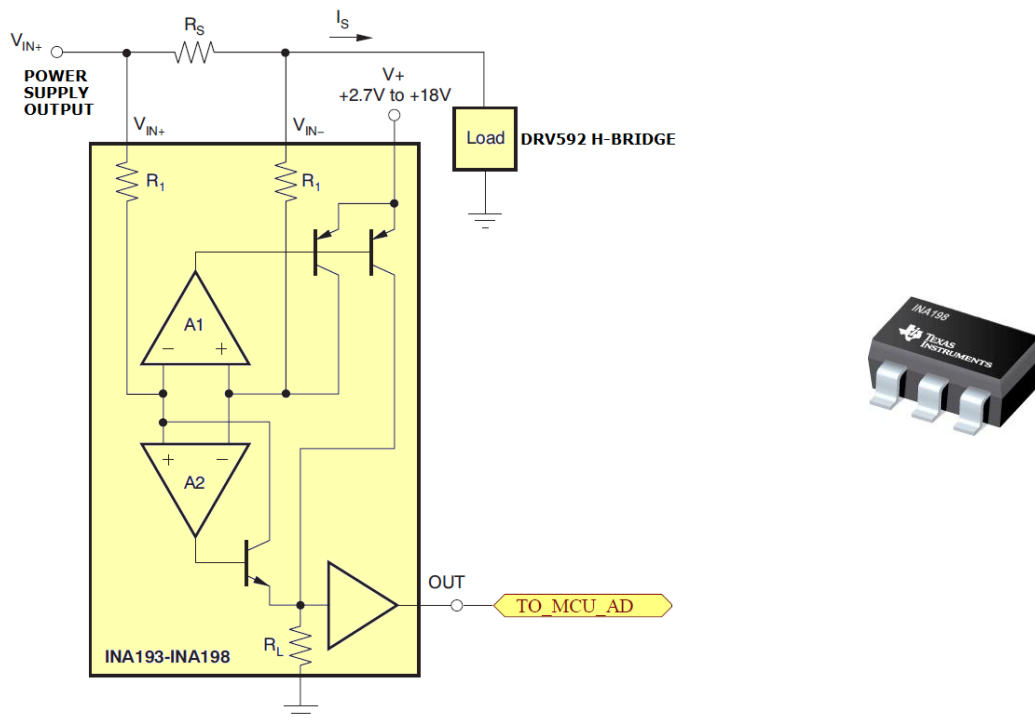
3.8. ábra – A felső- és alsóoldali árammérés ismertetése [8]

A felsőoldali árammérési elrendezést a sönt ellenállás és a feszültségforrás soros kapcsolása jellemzi. A söntellenállásra csatlakozó mérőáramkör differenciális mérést valósít meg (*műveleti erősítővel*). Az alkalmazott nagyfeszültségű műveleti erősítőt nagy közös módusú elnyomás jellemzi [8].

Az alsóoldali árammérőnél viszont a sönt ellenállás a föld-visszavezetésen helyezkedik el. Az alkalmazott műveleti erősítő kisfeszültségű is lehet. A mérés során nincs szükség differenciális mérésre [8].

Az árammérő kiválasztása:

A motorok áramfelvételének mérése az előbbieken ismertetett felsőoldali sönt ellenállás segítségével valósult meg. Az alkalmazott mérőáramkör a Texas Instruments *INA198* integrált áramkör. Az áramkör kapcsolási rajza a 3.9. ábrán látható.



3.9. ábra – Az árammérő kapcsolási rajza [10]

A választott áramköri elem jellemzői [10]:

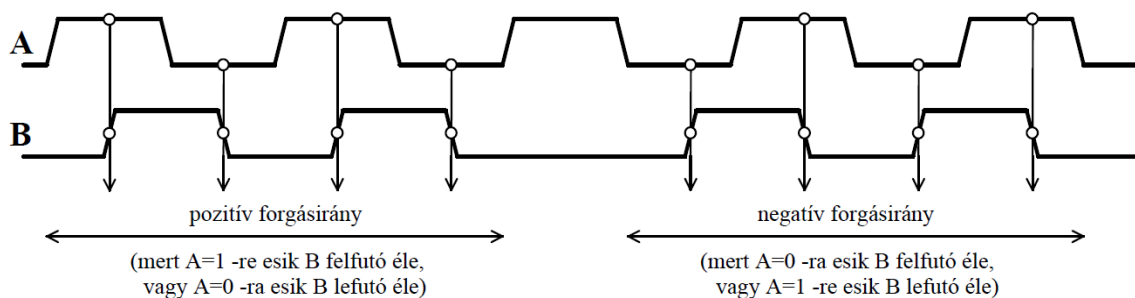
- tápfeszültség +2.7 – 18V között,
- analóg (feszültség) kimenet,
- feszültségerősítés $A = 100 V/V$,
- és az áramfelvétel $900 \mu A$.

A második fejezetben bemutatott szimulációs eredmények alapján a rotoráramok az $I_S = (25, 355) mA$ tartományba esnek. Amennyiben a mérőáramkör sönt ellenállásának értéke $R_S = 0.068 \Omega$ (tolerancia $\pm 1\%$, méret SMD 0805, hőteljesítmény $330 mW$), a mikrokontroller A/D bemenetére jutó feszültségtartomány:

$$U_{A/D} = A \times (R_S \times I_S) = (0.17, 2.414) V. \quad (3.3)$$

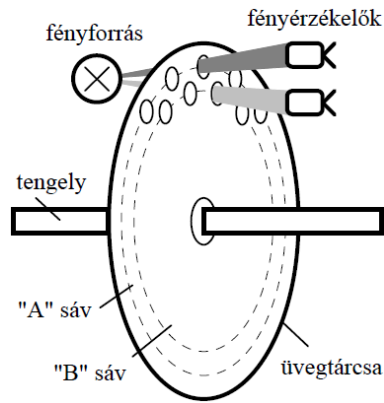
3.1.4.2 A forgási sebesség mérése

A forgási sebesség (és pozíció) mérésére leggyakrabban inkrementális szöghelyzet adókat használnak [1, 8]. Az inkrementális adó kettő, egymáshoz képest fázisban eltoltsorozatot szolgáltat (A és B). Az információ hordozója a jelek frekvenciája és a két jel egymáshoz mért fázisszöge, ami csak ± 90 fokos érték lehet. A frekvencia az adó tengelyének szögsebességével arányos, míg a fázisszög a tengely forgásirányára vonatkozó információt hordozza [1].



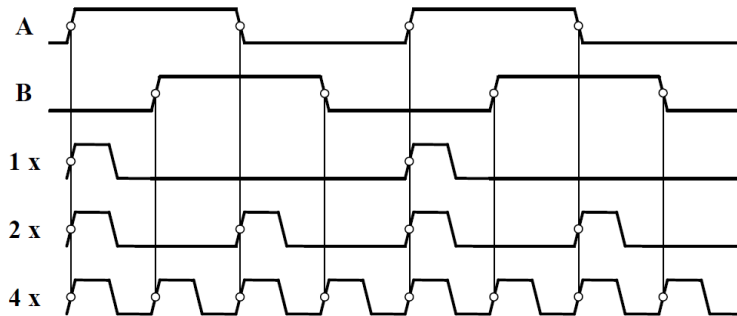
3.10. ábra – Az inkrementális jeladó jeleinek értelmezése [1]

Fizikailag az inkrementális adó egy zárt házba épített néhány cm átműrű „lyuktárcsából” és a hozzá kapcsolódó elektronikából áll. A lyuktárcsa valójában egy üvegtárcsa, amire vákuumgőzöléssel krómot hordanak fel és így egy egyenletes rácsszerkezetet alakítanak ki, amely optikailag váltakozva átlátszó ill. átláthatatlan. Erre a megoldásra azért van szükség, mivel a megkívánt „lyukszám” egyenletes mechanikai kifúrása technológiailag nehezebb feladat lenne [1].



3.11. ábra – Az inkrementális jeladó vázlatos belső felépítése [1]

A jelekben található információtartalom több annál, semhogy a számlálót csak impulzus-periódusonként növelnénk vagy csökkentenénk 1-gyel. A fogadó/kiértékelő áramkörök többsége megkülönböztet ún. 1-, 2- vagy 4-szeres kiértékelést a következők szerint [1]:



3.12. ábra – Az inkrementális jeladó jeleinek 1-, 2- és 4-szeres kiértékelése [1]

Az 1-szeres kiértékelés esetén egy impulzus („lyuk”) egyetlen éle kerül kiértékelésre, a teljes körfordulást a lyukak számával egyező részre tudjuk felbontani. A beérkező jelekből bizonyos feltételekkel azonban ennél több információ is kinyerhető: amennyiben biztosítjuk, hogy az impulzusok 50%-os kitöltésűek, akkor egyetlen impulzus fel- és lefutó élét is feldolgozva már 2-szeres kiértékelésről beszélünk, felbontásunk a lyukszám duplájára nőtt. Ha emellett biztosítani tudjuk a másik (eltolt) impulzussorozat pontosan 90 fokos eltolását, akkor abból nem csak forgásirány információját nyerhetjük ki, hanem még egyszer duplázzhatjuk a felbontásunkat – ilyenkor beszélünk 4-szeres kiértékelésről [1]. Az inkrementális adók előnye a nagy felbontásból adódó pontosságuk, hátrányuk viszont az abszolút pozíció információ hiánya.

A motorok fordulatszámának méréséhez (ebben az alkalmazásban) nem volt szükség a lyuktárcsa és a hozzá tartozó elektronika beszerzésére (fototranzisztorok, emitter diódák), valamint a mérőfelület kialakítására. A robotot meghajtó motorokhoz ugyanis ez a mérőfelület hozzá van illesztve, így a feladat megoldásához a rendelkezésre álló A és B csatornák jeleit kell feldolgozni és a szenzorok működéséhez szükséges tápfeszültséget biztosítani. Az A és B csatornák a mikrokontroller érzékeny bemeneteire csatlakoznak. A kiértékelés szoftveresen történik.

A motorra szerelt optikai szenzor jellemzői:

- a felbontás 100 *lyuk/fordulat*,
- rendelkezésre álló A és B csatorna,
- digitális kimenet,
- tápfeszültség 2.7 – 3.3 V,
- és az áramkör fogyasztása 8 mA.

3.1.4.3 A dőlésszög (*kilengés*) mérése

A robot dinamikai tulajdonságait a motorok forgási sebessége és áramfelvétele mellett a belső test kilengése határozza meg.

A dőlésszög mérésére az egyik megoldás a dőlésmérő szenzor (*ún. inclinometer*) használata. Ezeket a szenzorokat nagy pontosságuk és magas előállítási árak jellemzik. Mivel a dőlésmérő szenzor ára túl magas (*és ritka a kisméretű integrált áramköri kialakítása*), ezért a dőlésszög meghatározására legelterjedtebb megoldás a szenzorfüzió alkalmazása. A szenzorfüzió során két vagy több szenzort kombinálva és megfelelő szűrő algoritmus implementálva pontos mérési adatok határozhatók meg. A következőkben a szenzorfüzió során használatos szenzorok bemutatására kerül sor.

A dőlésszög meghatározását az *ún. MEMS (mikro-elektromechanikai)* inerciális mérőfelületek (*IMU*) is lehetővé teszik. A MEMS IMU az inerciális szenzortechnikába tartozó (*több dimenziós*) MEMS giroszkóp és gyorsulásmérő áramkörök kombinációjából áll. A giroszkóp és gyorsulásmérő szenzorok közös tulajdonsága, hogy differenciális működésűek, azaz csak két időpont közötti relatív mozgásváltozást képesek megmérni. A technológia előnyei [14]:

- kis méret és súly,
- robusztus kialakítás,
- kis teljesítmény-felvétel,
- rövid indítási idő,
- olcsó előállítás,
- és magas megbízhatóság.

A giroszkóp és gyorsulásmérő áramkörök hátránya viszont az, hogy önmagukban nem alkalmasak a dőlésszög pontos mérésére, minkét szenzor más-más tekintetben pontos illetve pontatlan (*ismertetésük a későbbiekben*). A megoldás a vázolt problémára: az *ún. Kálmán-szűrő* alkalmazása a szenzorok mérési eredményeire. A Kálmán-szűrő a lineáris dinamikus rendszerek esetében egy optimális algoritmus, amely kombinálja az alkalmazott szenzorok előnyeit, és alkalmas a dőlésszög pontos becslésére.

A dőlésszög meghatározása mellett, az IMU további információkat is szolgáltat a dinamikai viselkedésről. Az IMU segítségével a robot teljes dinamikai és kinematikai tartománya mintavételezhetővé válik. (Előfordulhat az az eset, hogy a robot olyan ismeretlen terepre téved, ahol a talaj csúszik. Ha a robotot meghajtó motorok gerjesztetlen állapotban vannak, akkor az inkrementális jeladó a robot egy helyben állásáról ad visszacsatolást – a valóságban viszont a robot haladó (transzlációs) mozgást végez a csúszó talaj által. Ez a jelenség az IMU mérőfelülettel detektálható. Továbbá az IMU felületet szabadesés detektálási funkcióval is rendelkezik.)

Összegezve: a dőlésszög meghatározásához ki kell választani a megfelelő szenzorokat (*gyorsulásmérő és giroszkóp*), a szenzorok mérési eredményeit fel kell dolgozni és alkalmazni a Kálmán-szűrő algoritmust a dőlésszög becslésére. A Kálmán-szűrőt a következő fejezet ismerteti.

3.1.4.3.1 A Kálmán-szűrő

A villamos jelek szűrése sok esetben nélkülözhetetlen. A szűrő algoritmus feladata, hogy eltávolítsa a zajt a feldolgozandó jelről úgy, hogy a hasznos információ megmaradjon. A Kálmán-szűrőt gyakran alkalmazzák beágyazott szabályozó rendszerekben, mert az algoritmus a vezérlésekhez szükséges állapotok pontos becslését teszi lehetővé.

A Kálmán-szűrő elvének kidolgozása Kálmán Rudolf Emil nevéhez fűződik, aki 1960-ban dolgozta ki a matematikai statisztikai szűrés elméleti alapját, és ezen keresztül a nemlineáris diszkrét rendszerek állapotterére alkalmazható matematikai eljárást, a Kálmán-szűrő elvét. Ennek segítségével a különféle adatsorokból kiszűrhető a „zaj”, s hiányos információk esetében is – rekurzív paraméterbecsléssel – megállapítható az adott rendszer időállapota. A Kálmán-szűrő hatékony számítási eljárással becsülni tudja a folyamat állapotát úgy, hogy eközben minimalizálja az *átlagos négyzetes hibát*. A szűrő algoritmus lehetővé teszi a múlt-, jelen és jövőbeli állapotok becslését még akkor is, ha a rendszer viselkedésének pontos modellje ismeretlen [11, 13].

A Kálmán-szűrő az \hat{x}_k állapot becslésénél már a rendelkezésre álló y_k kimenetet veszi figyelembe (és az előző u_{k-1} bemenetet). Ezáltal az $u_k = -K_k \hat{x}_k$ állapotvisszacsatolásban számíthatóvá válik u_k (mert csak az előző u_{k-1} szükséges hozzá), de már figyelembe lesz véve az aktuális y_k mérési eredmény, ami nyilván jobb irányítási tulajdonságok elérését teszi lehetővé [12].

Tekintsük a zajokkal terhelt diszkrétidejű lineáris rendszert [11, 12]:

$$\begin{aligned} x_k &= Ax_{k-1} + Bu_{k-1} + w_{k-1}, \\ y_k &= Cx_k + v_k. \end{aligned} \quad (3.4)$$

Az állapotegyenletben szereplő $x \in \mathfrak{R}^n$ az állapotváltozó vektor, $u \in \mathfrak{R}^l$ bemenő jel vektor és $y \in \mathfrak{R}^m$ kimenő jel vektor. Az állapotegyenlet $A_{n \times n}$ mátrixát állapotmátrixnak, $B_{n \times l}$ mátrixát bemeneti mátrixnak és a $C_{m \times n}$ mátrixát pedig kimeneti mátrixnak nevezzük.

Gyakorlati alkalmazás során az állapotegyenletben szereplő x állapot általában nem mérhető, ezért helyettesíteni kell valamilyen \hat{x} becslésével. A becslést y mérés segítségével lehet előállítani, azonban a mérés zajjal terhelt. A fenti egyenletekben szereplő w_k és v_k paraméterek a folyamatban és a mérés során megjelenő véletlenszerű *fehérzajt* jelentik. Ezek a paraméterek függetlenek egymástól és *Gauss-féle normális eloszlással* rendelkeznek [11]:

$$p(w) \sim N(m, \sigma^2) = N(0, Q), Q = E[w_k w_k^T], \quad (3.5)$$

$$p(v) \sim N(m, \sigma^2) = N(0, R), R = E[v_k v_k^T]. \quad (3.6)$$

Jelölje $\hat{x}_k^- \in \mathfrak{R}^n$ a jóslott (*jövendőlt, a priori*) állapotbecslést, amely a k . lépés előtti folyamat ismeretében határozható meg, és jelölje $\hat{x}_k \in \mathfrak{R}^n$ a frissített (*javított, a posteriori*)

állapotbecslést, amely az y_k mérés segítségével határozható meg. Ekkor a becslési hibák és azok kovarianciái [11]:

$$\begin{array}{l|l} e_k^- = x_k - \hat{x}_k^-, & P_k^- = E[e_k^- e_k^{-T}], \\ e_k = x_k - \hat{x}_k, & P_k = E[e_k e_k^T]. \end{array}$$

Definiáljuk a mérési újítást (*residuum*). A mérési újítás a mérési jóslat $C\hat{x}_k^-$ és az aktuális y_k mérés különbségét adja meg [11]:

$$(y_k - C\hat{x}_k^-).$$

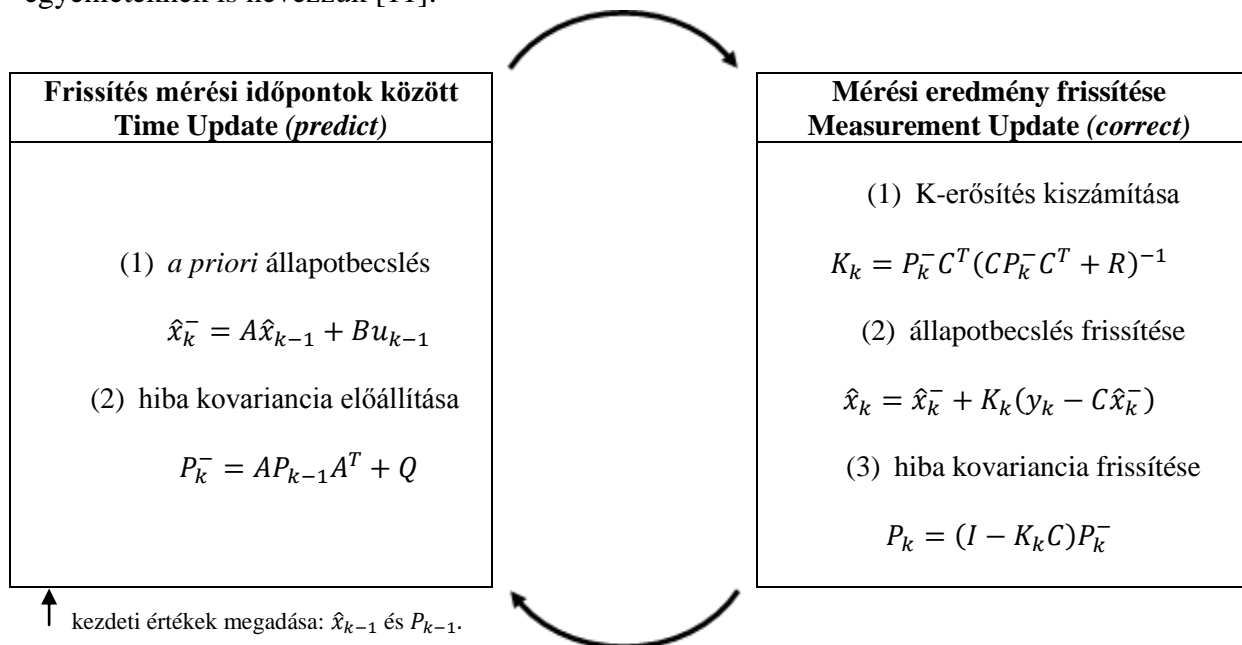
Az \hat{x}_k javított állapotbecslés előállítható az \hat{x}_k^- jövendőlt állapotbecslés és a súlyozott mérési újítás lineáris kombinációjaként [11]:

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-), \quad (3.7)$$

ahol a $K_{n \times m}$ mátrixot Kálmán-erősítésnek nevezzük. A Kálmán-erősítés minimalizálja a P_k becslési hiba kovariancia mátrixát, értéke [11]:

$$K_k = P_k^- C^T (C P_k^- C^T + R)^{-1}. \quad (3.8)$$

Az algoritmus az állapotok becslésére egy visszacsatolási mechanizmust használ. A szűrő időközönként becsléseket végez (*a priori állapot*), majd a mérési eredmények segítségével a becsléseket frissíti (*a posteriori állapot*). Az algoritmus kétfajta egyenlet típusra osztható fel: idő követő (*time update*) egyenletekre és mérés követő (*measurement update*) egyenletekre. Az idő követő egyenletek az aktuális állapotbecslésért és a becslési hiba kovarianciájának előállításáért, míg a mérés követő egyenletek az aktuális mérési eredményt és az állapotbecslést kombinálva a javított állapotbecslés előállításáért felelősek. Az idő követő egyenleteket előrejelző- (*predictor*), míg a mérés követő egyenleteket korrigáló (*corrector*) egyenleteknek is nevezzük [11].



3.13. ábra – A Kálmán-szűrő működési elve [11]

3.1.4.3.2 A MEMS gyorsulásmérő

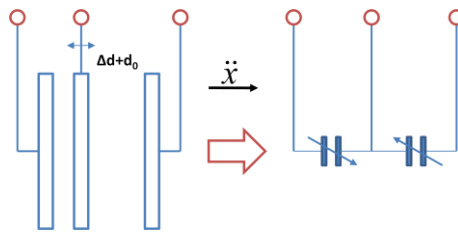
A gyorsulásmérő segítségével adott tengely (vagy tengelyek) mentén alkalmazott gyorsulás mérhető meg. Az alapelv a következő:

Egy egy-dimenzióban elmozdulni képes m tömegű test a mozgás irányában fel van függesztve egy rugóra. Ha a mérőegység a tengelye irányában gyorsulni kezd, akkor a rugó vagy megnyúlik, vagy összenyomódik. A rugó nyúltságának nagysága lineáris kapcsolatban van a szenzort ért gyorsulással [14, 15].

A MEMS gyorsulásmérő működési elve egy kondenzátor elektródáján mérhető kapacitásváltozás érzékelésén alapul. A kondenzátor C kapacitása:

$$C = \varepsilon \frac{A}{d}, \quad (3.9)$$

ahol ε a lemezeket elválasztó közeg permittivitása, A a lemez felülete, d pedig a lemezek közötti távolság. A félvezető-technológiával kialakított MEMS gyorsulásmérőnél az elektródák egyike mozgatható, és az elmozdulás nagysága arányos a gyorsító erővel. Az alkalmazott gyorsulással lineáris mérési eredmény szolgáltatását a következő elrendezés teszi lehetővé:



3.14. ábra – A MEMS gyorsulásmérő működési elve

Alkalmazzuk *Newton II. törvényét*:

$$F = m \cdot \ddot{x}, \quad (3.10)$$

ahol F a gyorsító erő, m a tömeg és \ddot{x} a gyorsulás. A mozgó tömeg rugós felfüggesztése révén az $F = K \cdot \Delta d$ erő következtében Δd elmozdulás keletkezik:

$$\Delta d = \frac{m \cdot \ddot{x}}{K}, \quad (3.11)$$

ahol K a rugóállandó. A kapacitív elvű gyorsulásérzékelőre vonatkozó összefüggés:

$$\frac{C_1 - C_2}{C_1 + C_2} = \frac{\Delta d}{d_0} = \frac{m \cdot \ddot{x}}{K \cdot d_0}. \quad (3.12)$$

A félvezető MEMS-szerzők a közvetlen mikromechanikai részegységek mellett további áramköröket tartalmaznak, melyek feladata a kapacitásváltozással arányos feszültségváltozás előállítás, jelfeldolgozása, szűrése és hőkompenzációja.

A gyorsulásmérő szenzor kiválasztása:

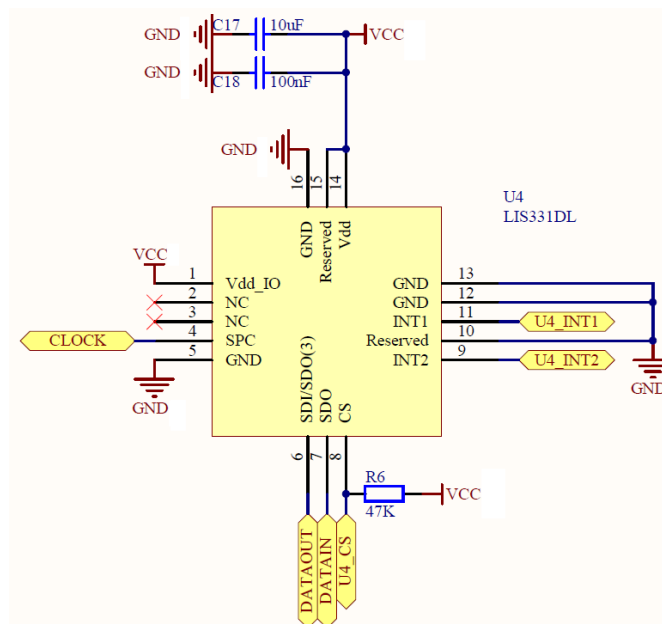
A pillanatnyi gyorsulás X, Y és Z tengely mentén (azaz három dimenzióban) mérendő. Elvárások a választott gyorsulásmérővel szemben:

- rendelkezzen digitális kimenettel és választható mérési tartományokkal,
- háromtengelyes mérést tegyen lehetővé (későbbi fejlesztésekhez),
- alacsony fogyasztású, kis tápfeszültséget igénylő áramkör legyen,

A fenti feltételeket az STMicroelectronics *LIS331DLH* típusú gyorsulásmérő áramköre elégítette ki. A gyorsulásmérő digitális kimeneti interfésszel rendelkezik és háromtengelyes mérést tesz lehetővé.

Az áramkör jellemzői [16]:

- a választható mérési tartományok: ± 2 és $8g$,
- I^2C/SPI digitális kimeneti interfész,
- a mérési eredmény ábrázolása 16 biten,
- választható mintavételezési sebességek: 100 és 400Hz,
- alacsony fogyasztás: $0.4mA$, és kis tápfeszültség: $2.5V - 3.6V$ között.



3.15. ábra – A MEMS gyorsulásmérő kapcsolási rajza

3.1.4.3.3 A MEMS giroszkóp

A giroszkóp adott tengely (vagy tengelyek) körüli szögsebesség mérését teszi lehetővé. A MEMS giroszkóp működési elve azon alapszik, hogy egy forgó koordináta rendszerben szilíciumhordozón kialakított rezgő tömegre a szögsebességgel arányos *Coriolis-erő* hat. A hordozó és a hozzá rögzített rezgő mechanikai rendszer forgása közben, a szögsebességgel

arányos, a rezgés irányára merőleges erő hat, így a tömeg a rezgés irányára merőlegesen elmozdul. Az elmozdulás kapacitív módon mérhető [14].

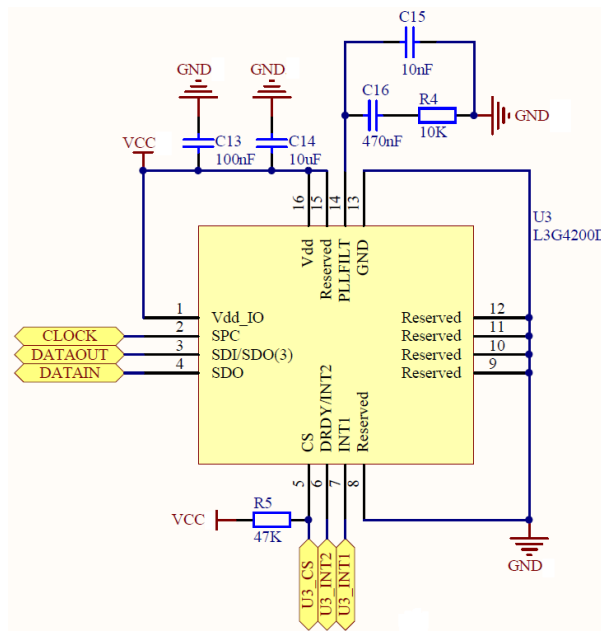
A pillanatnyi szögsebességet X, Y és Z tengely körül szükséges mérni. A gyorsulásmérőhöz hasonlóan, az elvárások a választott giroszkóppal szemben:

- háromtengelyes mérés (*későbbi fejlesztésekhez*) és választható mérési tartományok,
- digitális kimenet (*analóg kimenet esetén szűrő áramkör illesztése szükséges*),
- kis tápfeszültségű működés és alacsony fogyasztás,
- és figyelembe véve a második fejezet szimulációs eredményeit, a kilengés során várható maximális szögsebesség: $\omega < 250 \text{ dps}$.

A fenti követelményeket az STMicroelectronics L3G4200D típusú giroszkóp áramköre elégítette ki. A giroszkóp háromtengelyes mérést tesz lehetővé és digitális kimenettel rendelkezik.

Az áramkör jellemzői [17]:

- választható mérési tartományok: $\pm 250, 500$ és 2000 dps között,
- I^2C/SPI digitális kimeneti interfész és a mérési eredmény ábrázolása 16 biten,
- a tápfeszültség: $2.4 - 3.6V$, alacsony fogyasztás: *aktív* – $6.1mA$,
- választható mintavételezési sebességek: $100, 200, 400$ és $800Hz$.



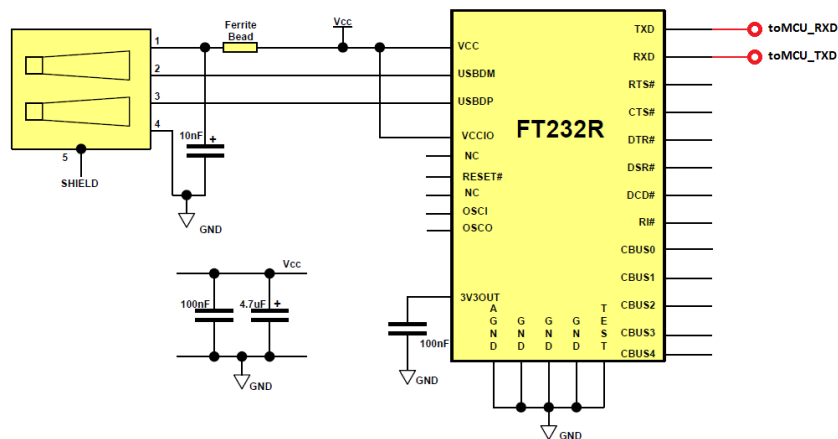
3.16. ábra – A MEMS giroszkóp kapcsolási rajza

3.1.4.3.4 Összegzés

A dinamikai tulajdonságok méréséhez szükséges áramkörök kiválasztása megtörtént. A szenzorok mérési eredményeinek kiértékeléséről a későbbiekben még részletesen beszélünk. A következőkben az USB interfész és a mikrokontrollerek bemutatására kerül sor.

3.1.5 USB-UART interfész

A számítási eredmények megjelenítéséhez és a szűrő algoritmus paramétereinek kiszámításához egy kétirányú, nagy sebességű adatátviteli vonal kialakítása is szükséges a mikrokontroller és a számítógép között. Az *FTDI232R USB to serial UART* interfész egy olyan céláramkör, melynek egyik oldalán a szabványos USB felület található (*teljes mértékben lekezelve az USB protokollt*), másik oldalán pedig egy soros felületet áll rendelkezésre ami a mikrokontrollerhez csatlakozik. A mikrokontroller és a számítógép a kommunikációs felületet virtuális soros portként kezeli, a két oldal között csak a kommunikációs sebességet kell összehangolni. Aszinkron soros kapcsolat esetén maximum *3Mbaud* adatátviteli sebesség valósítható meg [1, 18].



3.17. ábra – A kommunikációs egység kapcsolási rajza

3.1.6 Mikrokontrollerek

A fejlesztés első fázisában kitűzött cél a mobilis robot vezérlésének megoldása, valamint a dinamikai tulajdonságokat (*áramok, sebességek és dőlésszög*) mérő szenzorok jeleinek feldolgozása és továbbítása a számítógépen futó felhasználói programnak. A vezérlés során a számítógép parancsokat küld a robotnak. A parancsok tartalmazzák az irányt és a translációs sebességet. A parancs fogadását követően a robot mozgást végez az előírások szerint. Eközben a felhasználó program a rotoráramokat, forgási sebességeket és a dőlésszöget minden pillanatban rögzíti. Ez lehetővé teszi a verifikációt. (*A fejlesztés második fázisa a szabályozó tervezése és implementálása – jövőbeli terv.*)

A feladat megoldásához egy mikrokontrolleres felület illesztése szükséges, amely *A és B* mikrokontrollerekből áll a *3.1. ábra* alapján. A mikrokontrolleres felület az elektronika központi részét képezi. A felület táplálását a kisebb teljesítményű tápegység biztosítja.

3.1.6.1 Mikrokontroller (A)

Az eszköz segítségével a következő feladatokat kell megoldani:

- a számítógéptől érkező parancsok (*Bluetooth UART*) fogadása és feldolgozása, a parancs hatására a H-hidak vezérlése,
- a robot irányának, sebességének és motorok áramfelvételének mintavételezése,

- az aktuális irány, sebesség, áramfelvétel és a pillanatnyi kilengés értékének visszaküldése a számítógép felé (*Bluetooth UART*).

A mikrovezérlő a következő tulajdonságokkal kell, hogy rendelkezzen:

- alacsony tápfeszültség és alacsony fogyasztás az akkumulátoros táplálás miatt,
- soros kommunikációs interfész (*Bluetooth modulhoz, UART mód*) az adatok küldésére és fogadására, és egy további soros kommunikációs interfész (*B mikrokontrollerhez, UART mód*) a dőlésszög értékének fogadására,
- számláló modul összehasonlító (*compare*) funcióval a PWM jelek előállítása miatt,
- digitális be- és kimenetek a H-hidak működésének beállítására ill. ellenőrzésére,
- érzékeny digitális bemenetek a jeladók jeleinek fogadására,
- és A/D átalakító a rotoráramok mintavételezéséhez.

3.1.6.2 Mikrokontroller (B)

Az eszköz segítségével a következő feladatokat kell megoldani:

- a MEMS giroszkóp és gyorsulásmérő áramkörök felprogramozása (*SPI mód*),
- a mérési eredmények ciklikus kiolvasása és továbbítása a számítógép felé: USB-UART interfészen (*MATLAB környezetben a szűrő paraméterek kiszámítása, a szűrő tesztelése → kis túllövés, gyors válaszidő*) valamint az *A mikrokontrolleren keresztül (a szűrő paraméterek beállítása után)*,
- a szűrő algoritmus implementálása, a kiszámított dőlésszög folyamatos továbbítása az *A mikrokontrollernek*.

A mikrovezérlő a következő tulajdonságokkal kell, hogy rendelkezzen:

- alacsony tápfeszültség és kis fogyasztás az akkumulátoros táplálás miatt,
- *SPI* adatátviteli interfész: a szenzorok programozása/eredmények kiolvasása,
- *UART* adatátviteli interfész a mérési eredmények továbbítására az USB interfész felé,
- *UART* adatátviteli interfész a mérési eredmények továbbítására az *A mikrokontroller* felé, valamint digitális ki- és bemenetek a szenzorok működésének ellenőrzésére.

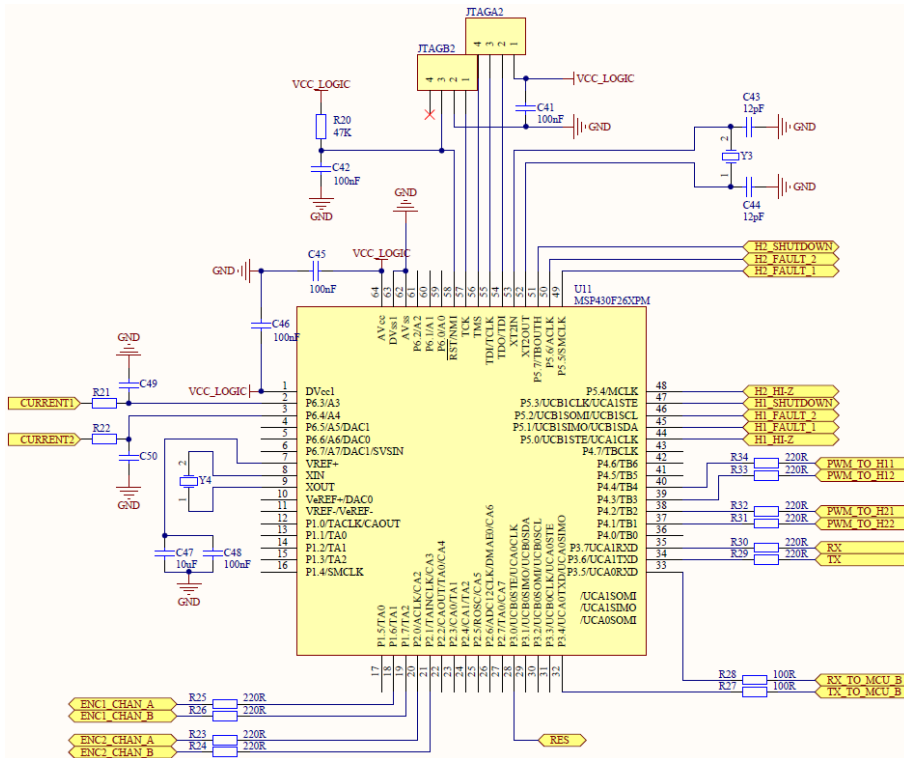
3.1.6.3 A választott mikrokontrollerek

A fenti 3.1.6.2 és 3.1.6.3 fejezetekben tárgyalt elvárások kiértékelésével két egyforma, a Texas Instruments *MSP430*-as processzorcsalád 16 bites ultra-alacsony fogyasztású *F2618* mikrokontrollere vált alkalmassá a feladat megoldására.

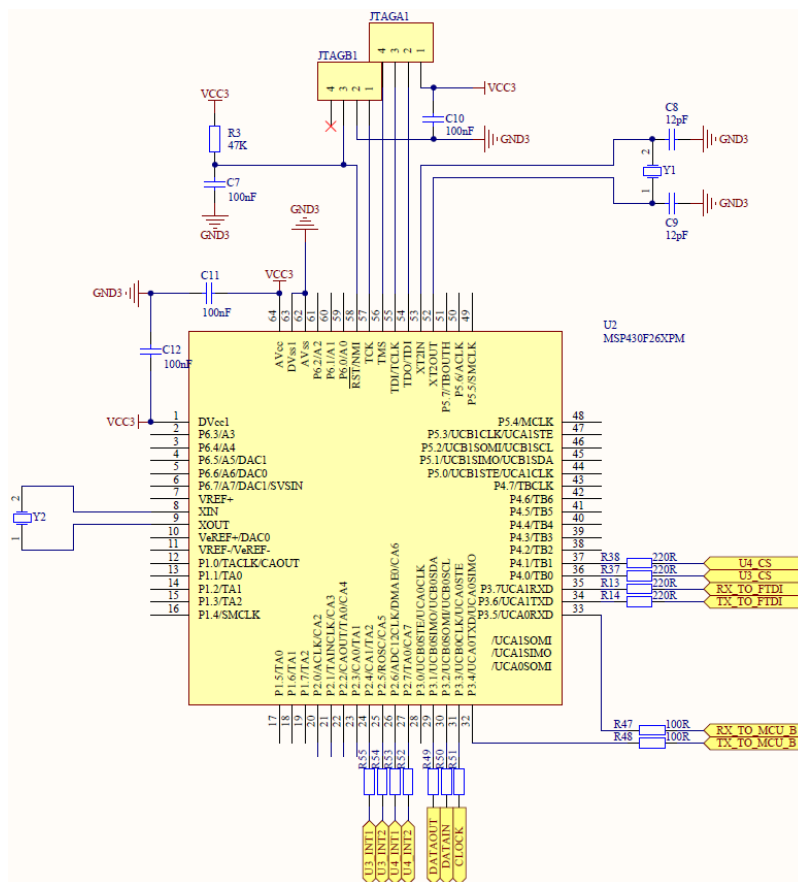
A mikrokontrollerek jellemzői [19]:

- alacsony tápfeszültség: 1.8 – 3.6 V,
- ultra-alacsony fogyasztás: *aktív* – 365 μ A, *standby* – 0.5 μ A,
- 16 bites RISC architektúra, 62.5ns-os műveletvégrehajtási idő,
- 12 bites *analóg/digitális* átalakító belső referencia feszültséggel
- 16 bites *Timer A* számláló három összehasonlító regiszterrel
- 16 bites *Timer B* számláló hét összehasonlító regiszterrel
- négy univerzális soros interfész (2 *UART*, 2 *SPI*).





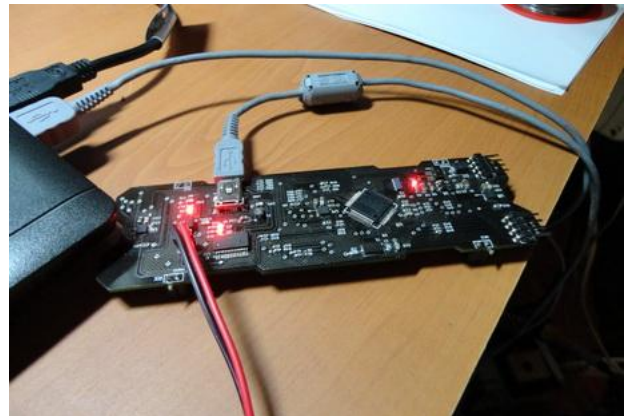
3.18. ábra – Az A mikrokontroller kapcsolási rajza



3.19. ábra – A B mikrokontroller kapcsolási rajza

3.1.7 Összegzés

Az elektronikai berendezés kapcsolási rajzának és nyomtatott áramköri tervének kialakítását (a NYÁK-ok legyártása után), az alkatrészek felforrasztása és az elektronika felélesztése követte. A 3.20. ábra néhány fejlesztési képet tartalmaz.



3.20. ábra – A felélesztett elektronikai rendszer bemutatása

A következőkben a működőképes hardver mikrovezérlős felületének felprogramozására és a felhasználói szoftver megírására került sor.

3.2. A felhasználói szoftver

A felhasználói szoftver feladata a robot irányítása, valamint a robot irányának, pillanatnyi sebességének, a motorok áramfelvételének és a belső test kilengésének rögzítése. A program felületén az irányítás gombokkal történik, a sebesség csúszkával állítható, a mérési eredmények pedig a felület közepén található ábrán kerülnek rögzítésre. A felhasználói program MATLAB-ban készült.

A program indítását és a komponensek inicializálását követően az első lépés a vezeték nélküli kapcsolat kialakítása. A program virtuális soros porton keresztül küldi és fogadja az adatokat. E célra alkalmas COM-port nevet a Bluetooth adapter határozza meg. A robothoz csatlakozás az alkalmas port név kiválasztásával és a *Connect* gomb megnyomásával történik. Ezt a következő forráskód igazolja:

```
function butConnect_Callback(hObject, eventdata, handles)

port = strcat('COM', num2str(get(handles.listPorts, 'Value')));
handles.s = serial(port);
if strcmp(handles.s.Status, 'closed'),
    fopen(handles.s);
else
    disp('Can''t open the COM-Port!');
end

guidata(hObject, handles);
```

3.2.1 Parancsok küldése a robotnak

A robotnak küldött parancsokat két kategóriába oszthatjuk:

- vezérlési parancsok,
- mérési parancsok.

A vezérlési parancsok a robot irányítására szolgálnak, míg a egy-egy mérési parancs azt határozza meg, hogy melyik dinamikai tulajdonságot mérjük.

A robotnak küldött parancsok 4 bájton vannak ábrázolva. A négy bájtból az első és az utolsó a keret elejét (*frame start*) és a keret végét (*frame stop*) jelöli ki. A maradék a hasznos információ, ami attól függően hogy vezérlés vagy mérés két bájtkombinációja. Vezérlés esetén ebből az első az aktuális irányt, második az alkalmazott sebességet jelenti, míg a mérés kiválasztásánál ez három darab két bájtból álló karakterkombináció. Az, hogy egy adott parancs fogadása a túloldalon vezérlést vagy mérést fog jelöl, azt az *A mikrokontroller* állapotgépe határozza meg.

Példa egy vezérlő gomb megnyomására:

Egy gomb megnyomásával a gombhoz- és a kiválasztott sebességhez tartozó karakterek kombinációja kerül elküldésre. A kiválasztott sebesség a PWM jel kitöltési tényezőjét határozza meg (*H-hidak meghajtása*), az irány pedig azt, hogy melyik motort milyen irányba vezéreljük. Minden egyes gomb megnyomásakor a következő folyamat játszódik le (*pl. mozgás előre*):

```
function moveForward_Callback(hObject, eventdata, handles)

pwm = get(handles.listSpeed, 'Value');
command = strcat('!F', pwm, '#');
fwrite(handles.s, command);

guidata(hObject, handles);
```

A fenti kódrészletből látható, hogy a keret elejét és végét a „!” és „#” karakterek jelzik. A keret kezdetét jelző karaktert rendre az irány (*F karakter*) és a PWM jel kitöltési tényezője követi. A robot oldalán a karakter sorozatot az *A mikrokontroller* állapotgépe dolgozza fel és (*részletezése a későbbiekben*).

3.1.2 Eredmények fogadása a robottól

A robot-oldali *A mikrokontroller* a kiválasztott dinamikai tulajdonságot meghatározott időközönként méri és továbbítja a számítógép felé. A dinamikai tulajdonságok mérése közé az inkrementális jeladók jelei alapján a pozíciók számolása, az A/D bemenetre kerülő feszültségek mérése és a pillanatnyi dőlésszög értékének fogadása (*a B mikrokontrollertől*) tartozik.

A folyamatosan érkező adatcsomagok mérete 6 bájt. A 6 bájtól hasonlóan a küldési folyamathoz, itt is az első és az utolsó a keretet jelöli ki. A 4 bájtnyi hasznos információ pedig attól függően, hogy melyik mérés van kiválasztva más és más feldolgozást jelent. A felhasználói programban a feladat megoldásához egy *switch-case* szerkezet tartozik, ami a kiválasztott mérést jelző változó alapján értékeli ki a fogadott adatcsomagot. A felhasználói program a soros port bemeneti bufferébe tárolja az érkező adatokat és adott bájtnyi bejövő adat után egy eseményt generál (*event*), ahol az adatcsomagok feldolgozására kerül sor.

```
handles.s.BytesAvailableFcnCount = 6*10; % minden 60 bajt érkezésekor @getNewData
handles.s.BytesAvailableFcn = @getNewData(hObject, eventdata, handles); % hívodik
handles.s.BytesAvailableFcnMode = 'byte';

%...

function getNewData(hObject, eventdata, handles) % újabb 60 bajt érkezett meg

    Data = fread(handles.s);
    Data = reshape(Data, 6, 10)';

    % 1 2 3 4 5 6
    % 7 8 9 10 11 12
    % 13 14 15 16 17 18...

    switch handles.state % a Data vektor mit tartalmaz?
        case 1
            % a pillanatnyi aramok ertekeit
            % Data vektor feldolgozasa
        case 2
            % a pillanatnyi sebességek es irányok ertekeit
            % Data vektor feldolgozasa
        case 3
            % a pillanatnyi dolesszog erteket
            % Data vektor feldolgozasa
    end

    guidata(hObject, handles);
```

3.1.2.1 A forgási sebesség meghatározása

A kerekek pillanatnyi forgási sebességének meghatározásához a fizikában tanult képletet használjuk, azaz a sebesség a Δt időköz alatt megtett Δs útnak és a Δt időköznek hányadosa. A képletben a megtett útnak az időköz alatt bejövő inkremensek száma felel meg, míg az időköz a két mérés között eltelt idő. Ennek megfelelően a kerekek pillanatnyi sebessége:

$$v[\text{rpm}] \triangleq \frac{\Delta s}{\Delta t} := \frac{i[\text{inc}] \times n \left[\frac{\text{rot}}{\text{inc}} \right]}{t[\text{min}]} \times \frac{1}{k}, \quad (3.13)$$

ahol $i[inc]$ a (két mérés között összeszámolt) inkremensek száma, $t[min]$ a mérések között eltelt idő, $n \left[\frac{rot}{inc} \right]$ az egy rotor fordulat alatt generálódó inkremensek száma és k pedig az áttétel.

A fenti paraméterek értékei:

- $t = 0.01 \cdot \frac{1}{60} [min]$, (azaz a mintavételi frekvencia 100 Hz)
- $n = \frac{1}{200} \left[\frac{rot}{inc} \right]$, a lyuktárcsa osztása
- $k = 256$, az áttétel (a motor adataiból).

3.1.2.2 Az áramfelvétel meghatározása

A rotoráramok pillanatnyi értékének meghatározásához az egyszerű Ohm-törvényt használjuk. Mivel az A/D átalakító bemenetén mérhető feszültség a sönt ellenálláson eső feszültség és az árammérő áramkör erősítésének szorzata, ezért:

$$I[A] = \frac{U_{AD}[V]}{R_s[\Omega] \times A'} \quad (3.14)$$

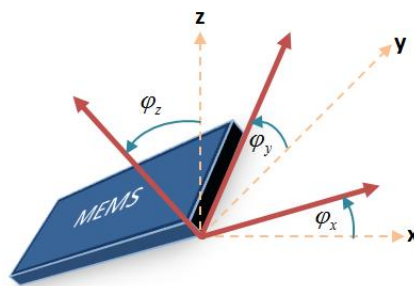
ahol $U_{AD}[V]$ az A/D átalakító bemenetére jutó feszültség, $R_s[\Omega]$ a sönt ellenállás és A pedig az árammérő erősítése.

A paraméterek értékei:

- $R_s = 0.068 [\Omega]$,
- $A = 100$.

3.1.2.3 A dőlésszög meghatározása

Jelölje rendre φ_x, φ_y és φ_z a földi koordináta rendszer X, Y és Z tengelyeitől mérhető szögelfordulásokat a 3.21. ábra alapján.



3.21. ábra – A koordináta rendszer dőlésszögei

3.1.2.3.1 A dőlésszög mérése giroszkóppal

A giroszkóp kimenetén az adott tengely körül érzékelt szögsebesség nagysága jelenik meg. A dőlésszög a pillanatnyi szögsebesség integrálásával kapható meg.

A diszkrét integrálás: a dőlésszög k . értéke egy összeg, ahol az egyik összeadandó a dőlésszög $(k - 1)$. értéke, a másik összeadandó pedig a giroszkóp k . értékének és a mintavételi időnek szorzatával egyenlő. Jelölje a giroszkóp kimenetét ω_x , ω_y és ω_z , továbbá a mintavételezési időt T . Ekkor a földi koordináta rendszer tengelyeitől mért szögelfordulások:

$$\begin{aligned}\varphi_{x,k} &= \varphi_{x,k-1} + \omega_y T, \\ \varphi_{y,k} &= \varphi_{y,k-1} - \omega_x T, \\ \varphi_{z,k} &= \varphi_{z,k-1} + \omega_z T.\end{aligned}\tag{3.15}$$

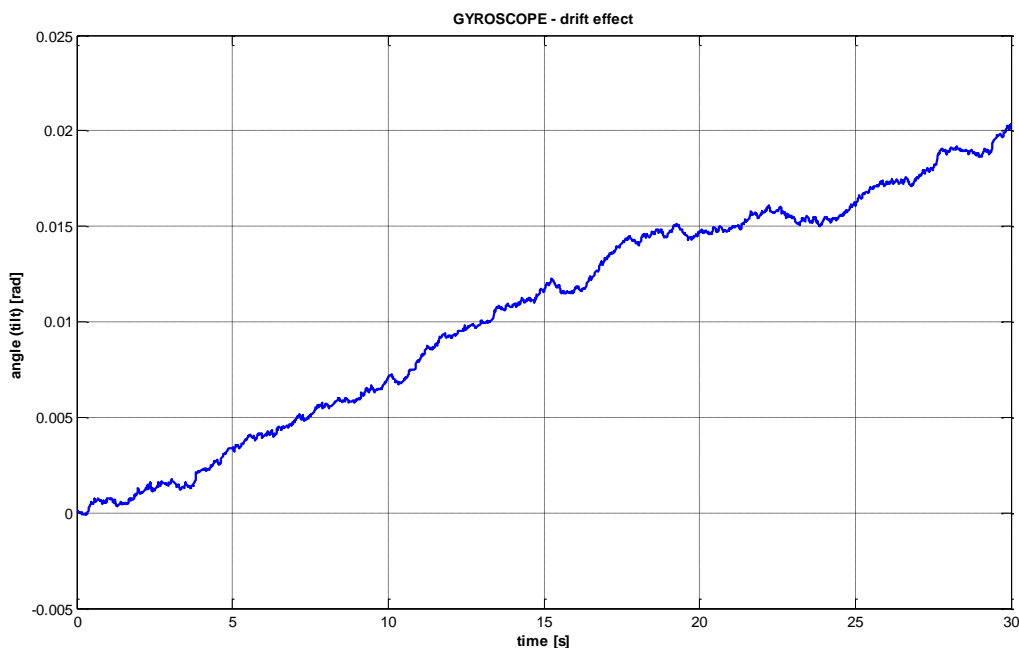
Drift jelenség:

A giroszkóp bias értékének (*kimeneti ofszet*) a kimenet átlagértékét nevezzük, akkor, ha a szenzor nyugalmi (*mozdulatlan*) állapotban van. A giroszkóp kimenetének integrálása során $\tilde{\varphi}$ dőlésszög hiba keletkezik, amely az eltelt idővel lineárisan változik [14]:

$$\tilde{\varphi}(t) = b \cdot t,\tag{3.16}$$

ahol b a giroszkóp bias értékét (*csúszási szögsebességét*), t pedig az időt jelöli.

A csúszási és egyéb jelenségek (*hőmérséklet változás, kalibrációs hiba, ofszet hibák stb.*) miatt a giroszkóp önmagában nem alkalmas a dőlésszög pontos meghatározására. A jelentkező hibák a gyorsulásmérővel korrigálhatók.

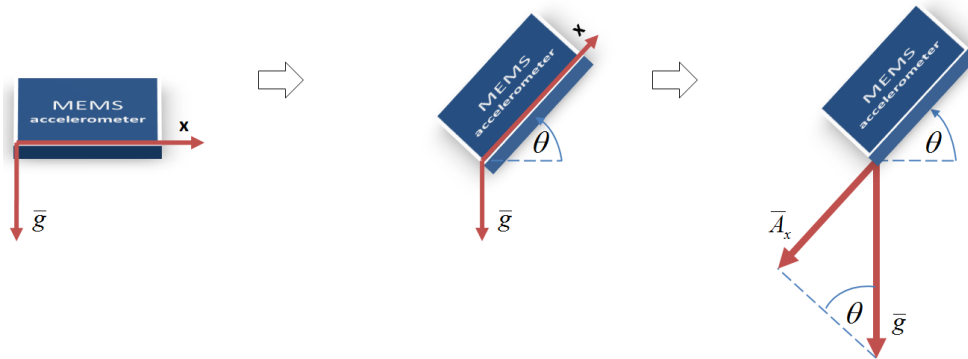


3.22. ábra – Mérési eredmény: a drift jelenség bemutatása

3.1.2.3.2 Dőlésszög mérése gyorsulásmérővel

A gyorsulásmérő a g gravitációs gyorsulás (*nehézségi gyorsulás*) X, Y és Z irányú komponensét (*vetületét*) méri. A komponensek értékéből a dőlésszög iránya és nagysága határozható meg. A következőkben az *egy- és kéttengelyes* dőlésszög meghatározási módszer lesz bemutatva [21, 22, 23].

Dőlésszög meghatározása egytengelyes méréssel:



3.23. ábra – Az egytengelyes mérés bemutatása

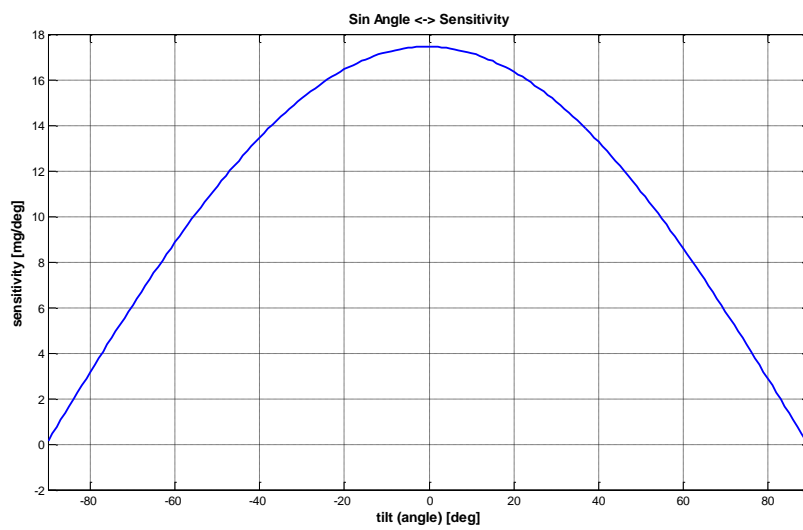
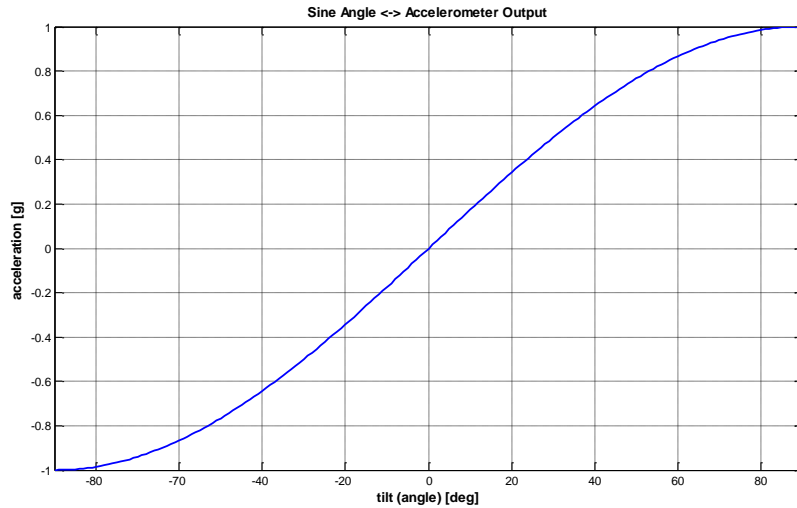
Az gyorsulásmérő által érzékelt gyorsulás nagysága a nehézségi gyorsulás és a dőlésszög szinuszának szorzatával egyenlő [21]:

$$A_x = g \cdot \sin(\theta). \quad (3.17)$$

A dőlésszög a fenti egyenletből:

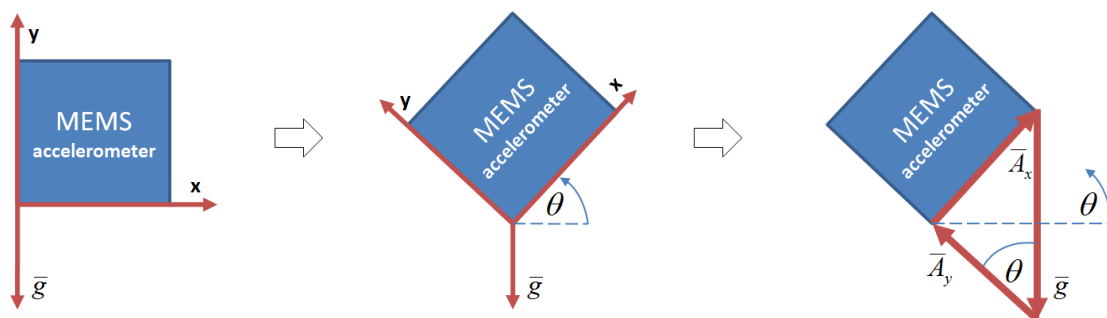
$$\theta = \sin^{-1}\left(\frac{A_x}{g}\right). \quad (3.18)$$

Az egytengelyes mérés egyik hátránya, hogy a dőlésszög mérés érzékenysége a $(-90^\circ, -45^\circ)$ és $(45^\circ, 90^\circ)$ tartományokban csökken. Ez a számítási eljárás a dőlésszög eredményét pontatlanná teszi, ha a gyorsulásmérő kimenete a $\pm 1g$ közelében van a 3.24. ábra alapján. A számítási eljárás másik hátránya, hogy a dőlési irány (*a dőlésszög előjele*) nem határozható meg (*ha a mérési eredmény $0.5g$, akkor a dőlésszög 30° vagy 150° is lehet*) [21]. A kéttengelyes mérési elrendezéssel az említett problémák kiküszöbölhetők.



3.24. ábra – Az egytengelyes mérés érzékenysége a dőlésszög függvényében

A dőlésszög meghatározása kéttengelyes méréssel:



3.25. ábra - A kéttengelyes mérés bemutatása

A X és Y tengelyek mentén érzékelt gyorsulások:

$$\begin{aligned} A_x &= g \cdot \sin(\theta), \\ A_y &= g \cdot \cos(\theta). \end{aligned} \tag{3.19}$$

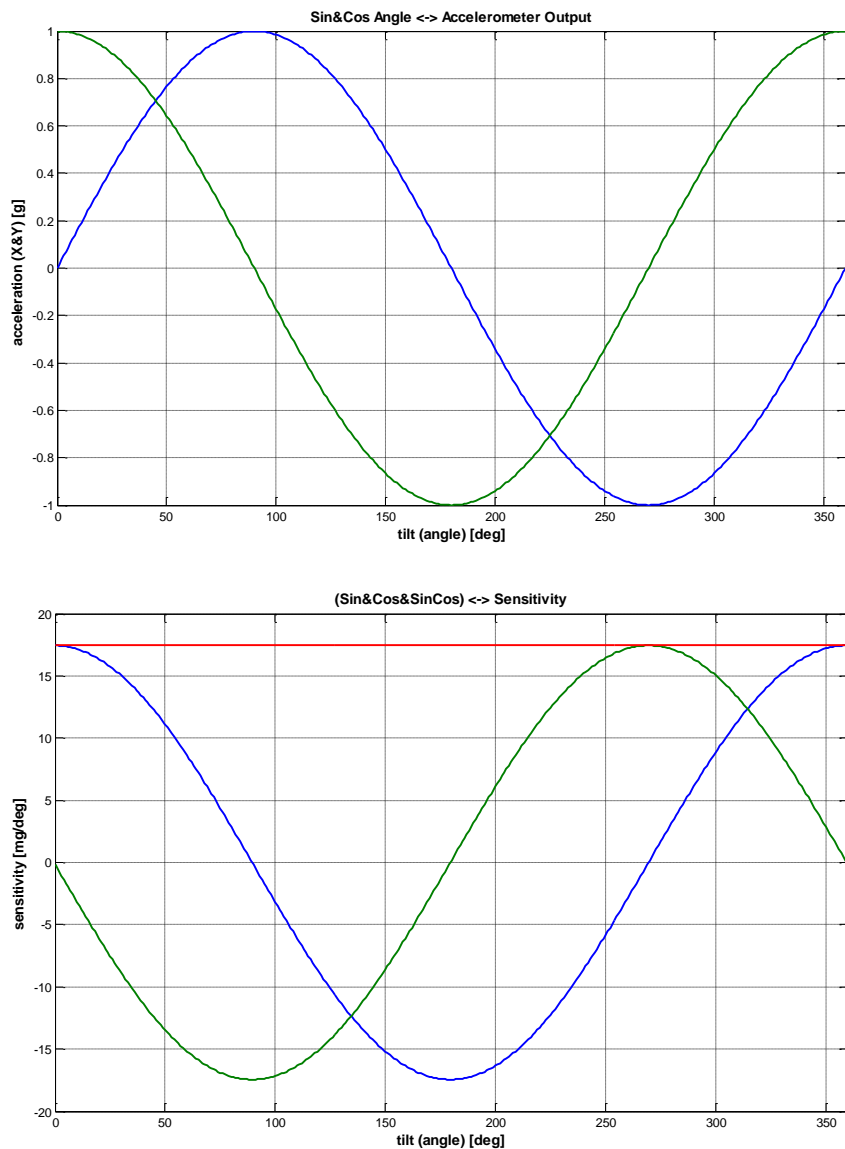
A fenti gyorsulások hányadosa a dőlésszög tangensét adja meg:

$$\tan(\theta) = \frac{A_x}{A_y}. \quad (3.20)$$

A dőlésszög nagysága kéttengelyes mérési elrendezés esetén:

$$\theta = \tan^{-1}\left(\frac{A_x}{A_y}\right). \quad (3.21)$$

A 3.26. ábra egy 360°-os szögelfordulás során keletkező X és Y tengelyek menti gyorsulásokból (rendre A_x és A_y) számított érzékenységeket mutatja.



3.26. ábra – Az egy- (kék, zöld) és kéttengelyes (piros) mérés érzékenysége a dőlésszög függvényében

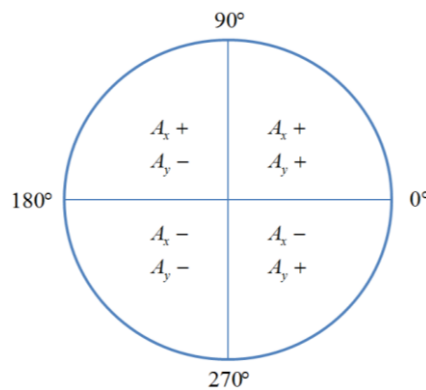
Látható, hogy az A_x és A_y -ből számított érzékenységek kombinációja állandó, értéke $17.45\text{mg}/^\circ$ (piros). Az érzékenység a gyorsulási értékek különbségéből kapható meg 1° -os dőlésszög változás esetén [21, 22]:

$$\begin{aligned}\Delta_x(\theta) &= A_x(\theta + 1^\circ) - A_x(\theta), \\ \Delta_y(\theta) &= A_y(\theta + 1^\circ) - A_y(\theta).\end{aligned}\quad (3.22)$$

Az X és Y tengelyek menti gyorsulásokból számított érzékenységek kombinációja pedig:

$$\Delta(\theta) = \sqrt{\Delta_x(\theta)^2 + \Delta_y(\theta)^2}.\quad (3.23)$$

A kéttengelyes mérési elrendezésnél továbbá számolni kell azzal, hogy egy 360° -os szögelfordulás során az első és harmadik síknegyed, valamint a második és negyedik síknegyed ugyanazon eredményeket adja a 3.27. ábra alapján [21].



3.27. ábra – Egy forgatás során keletkező gyorsulások előjelei

A következő algoritmus szolgál a pontos dőlésszög meghatározására az A_x és A_y gyorsulások előjele alapján [21]:

- ha az A_x és A_y gyorsulások előjele pozitív (*első síknegyed*), akkor a dőlésszög

$$\theta_1 = \tan^{-1}\left(\frac{A_x}{A_y}\right),\quad (3.24)$$

- ha az A_x gyorsulás előjele pozitív és az A_y gyorsulás előjele negatív (*második síknegyed*), valamint, ha az A_x és A_y gyorsulások előjele negatív (*harmadik síknegyed*), akkor a dőlésszög

$$\theta_2 = \theta_3 = \theta_1 + 180^\circ,\quad (3.25)$$

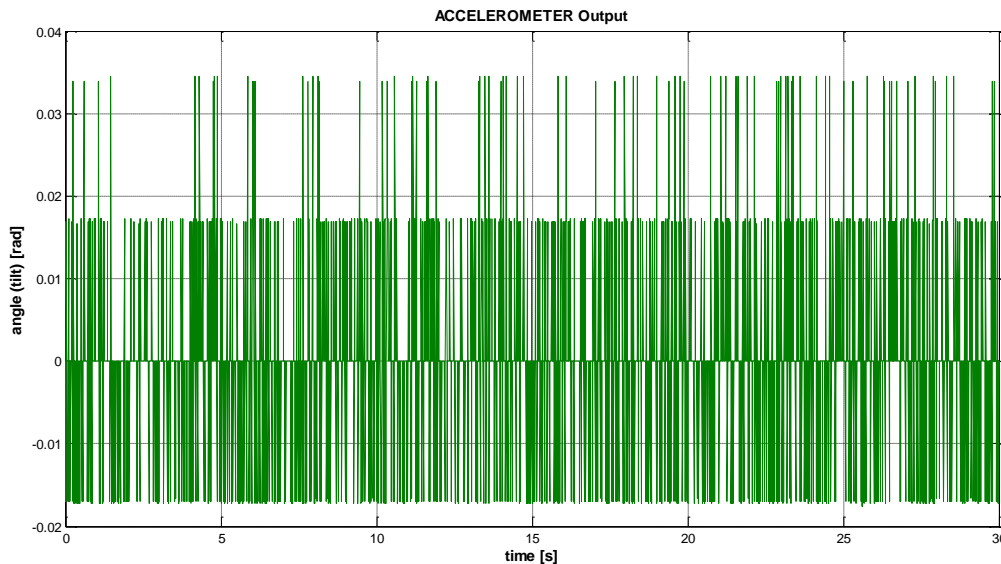
- és, ha az A_x gyorsulás előjele negatív, az A_y gyorsulás előjele pedig pozitív (*negyedik síknegyed*), akkor a dőlésszög

$$\theta_4 = \theta_1 + 360^\circ.\quad (3.26)$$

A gyorsulásmérő hibái:

A gyorsulásmérő sem alkalmas önmagában a dőlésszög pontos meghatározására, mert:

- a szenzor érzékeny a zajos környezetre és rezgésekre,
- a mérési eredményeinek nagy a szórása (*zajjal terhelt kimenet*).



3.29. ábra – Mérési eredmény: a gyorsulásmérő kimenetéből számított dőlésszög mozdulatlan (nyugalmi) állapotban

A gyorsulásmérő kimenetén megjelenő zaj és a zajból eredő dőlésszög-szórás a giroszkóp segítségével eliminálható.

3.1.2.3.3 A szenzorfüzió alkalmazása

Mivel se a giroszkóp se a gyorsulásmérő nem használható önmagában a dőlésszögek meghatározására, ajánlott a két szenzort kölcsönösen használni. A szenzorfüzió a Kálmán-szűrő segítségével oldható meg.

A szenzorfüzió következményei:

- a gyorsulásmérő korrigálja a giroszkóp alapértékének csúszását,
- a giroszkóp eliminálja a gyorsulásmérő zajos mérési eredményeit.

A φ dőlésszög becslése giroszkóp és gyorsulásmérő segítségével

A szenzorok mérési eredményei egy lineáris rendszert alkotnak. A lineáris rendszer állapotegyenletében az u gerjesztést a giroszkóp kimenete, míg az y kimenő jelet a gyorsulásmérő mérési eredményeiből számított dőlésszög adja meg:

$$\begin{aligned} u \left[\frac{\text{rad}}{\text{s}} \right] &= \text{gyro_rate}, \\ y[\text{rad}] &= \text{accel_to_tilt}. \end{aligned}$$

Az x állapotvektor a φ [rad] dőlésszögből és a giroszkóp b $\left[\frac{rad}{s}\right]$ csúszási sebességéből (*bias érték*) áll:

$$x = \begin{bmatrix} \varphi \\ b \end{bmatrix}.$$

A diszkrétidejű időben változó lineáris rendszer állapotegyenlete [15]:

$$\begin{aligned} x_k &= \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} dt \\ 0 \end{bmatrix} u_{k-1}, \\ y_k &= [1 \quad 0] x_k, \end{aligned} \quad (3.27)$$

ahol dt [s] a két egymást követő mérés között eltelt időt adja meg.

Az algoritmus futtatását a szűrő paraméterek meghatározása előzi meg, ezek a mérési zaj R varianciája, a folyamatban jelentkező zaj Q kovariancia mátrixa és az x_0, P_0 kezdőértékek. Ezt követően a Kálmán-szűrő a mérési modell statisztikai sajátosságait figyelembe véve rekurzív módon határozza meg a dőlésszög értékét.

Szűrő paraméterek:

A mérési zaj R [rad²] varianciája a 3.29. ábrán látható mérési eredmények tömbbe tárolásával és a `var()` MATLAB függvény segítségével határozható meg. A mérési zaj R varianciájának értéke:

$$R = 1.3128 \cdot 10^{-4} \text{ rad}^2. \quad (3.28)$$

A folyamatban megjelenő zaj Q kovariancia mátrixa az állapotvektor állapotait terhelő zajok varianciáiból áll:

$$Q = E[w_k w_k^T] = E \begin{bmatrix} \tilde{\varphi}^2 & \tilde{\varphi} \tilde{b} \\ \tilde{b} \tilde{\varphi} & \tilde{b}^2 \end{bmatrix}, \quad (3.29)$$

ahol $\tilde{\varphi}$ a dőlésszöget terhelő fehérzajt, \tilde{b} pedig a giroszkóp csúszási sebességében megjelenő fehérzajt adja meg.

A dőlésszöget terhelő zaj varianciája a mozdulatlan giroszkóp kimenetét terhelő zaj \tilde{u}^2 varianciájának és a mintavételi idő négyzetének szorzatával egyenlő:

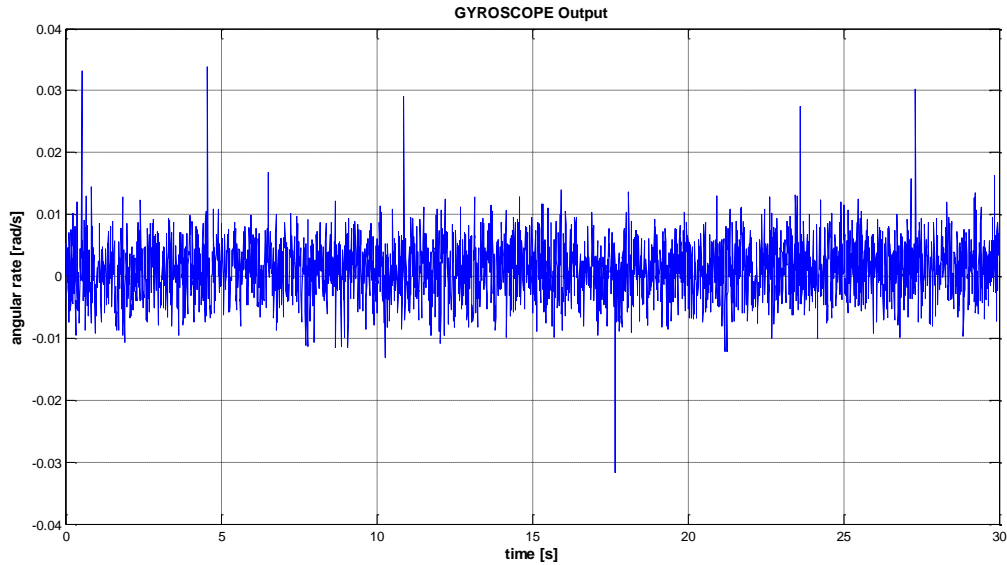
$$\tilde{\varphi}^2 = dt^2 \cdot \tilde{u}^2. \quad (3.30)$$

A mozdulatlan giroszkóp kimenetét terhelő zaj a 3.30. ábrán látható. Az ábra alapján a giroszkóp kimenetét terhelő fehérzaj varianciája:

$$\tilde{u}^2 = 2.0709 \cdot 10^{-5} \frac{\text{rad}^2}{\text{s}^2}. \quad (3.31)$$

A dőlésszöget terhelő zaj varianciája 10ms-os mintavételi idő esetén:

$$\tilde{\varphi}^2 = 2.0709 \cdot 10^{-9} \text{ rad}^2. \quad (3.32)$$



3.30. ábra – Mérési eredmény: a mozdulatlan giroszkóp kimenete

A mozdulatlan giroszkóp b csúszási sebessége a giroszkóp kimenetének aritmetikai középértéke (másrészt, a csúszási sebesség a 3.22. ábrán látható görbére illesztett egyenes meredeksége):

$$b = 6.7955 \cdot 10^{-4} \frac{\text{rad}}{\text{s}}. \quad (3.33)$$

Ebben az alkalmazásban, a csúszási sebességet terhelő fehérzaj varianciája a giroszkóp kimenetét terhelő fehérzaj varianciájával egyenlő:

$$\tilde{b}^2 = \tilde{u}^2 = 2.0709 \cdot 10^{-5} \frac{\text{rad}^2}{\text{s}^2}. \quad (3.34)$$

Feltételezve, hogy a dőlésszöget- és csúszási sebességet terhelő fehérzajok függetlenek, a folyamatban megjelenő zaj Q kovariancia mátrixa:

$$Q = \begin{bmatrix} 2.0709 \cdot 10^{-9} \text{ rad}^2 & 0 \\ 0 & 2.0709 \cdot 10^{-5} \frac{\text{rad}^2}{\text{s}^2} \end{bmatrix}. \quad (3.35)$$

A kezdőértékek megválasztása:

- a becslési hiba P kovariancia mátrixa és az állapotok kezdőértéke

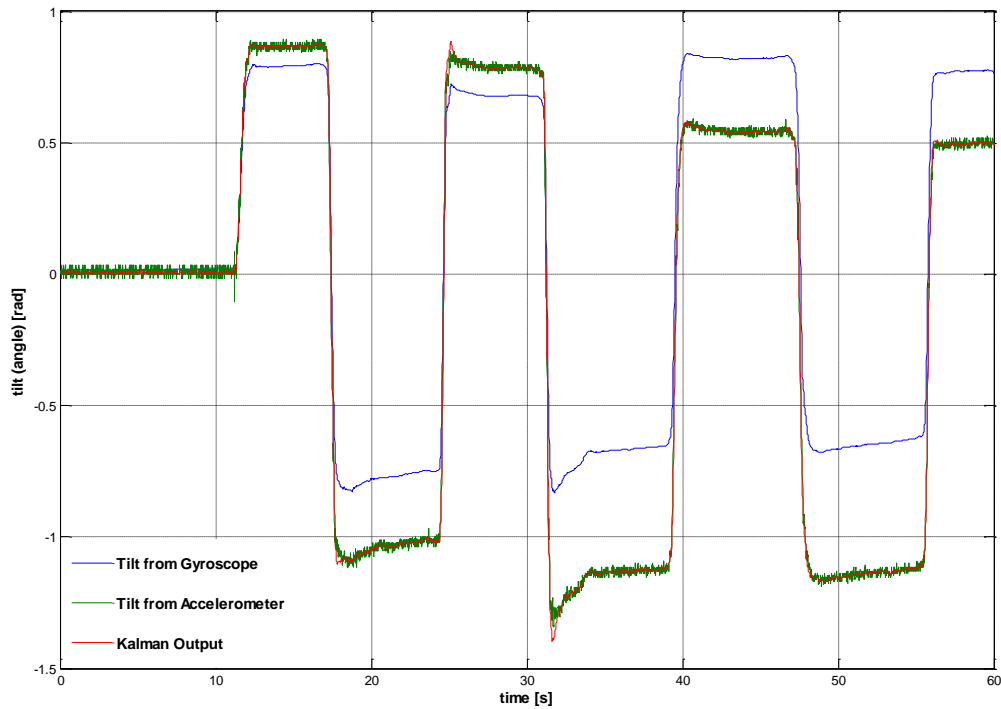
$$P_0 = Q, \quad (3.36)$$

$$x_0 = \begin{bmatrix} 0 \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 6.7955 \cdot 10^{-4} \frac{\text{rad}}{\text{s}} \end{bmatrix}.$$

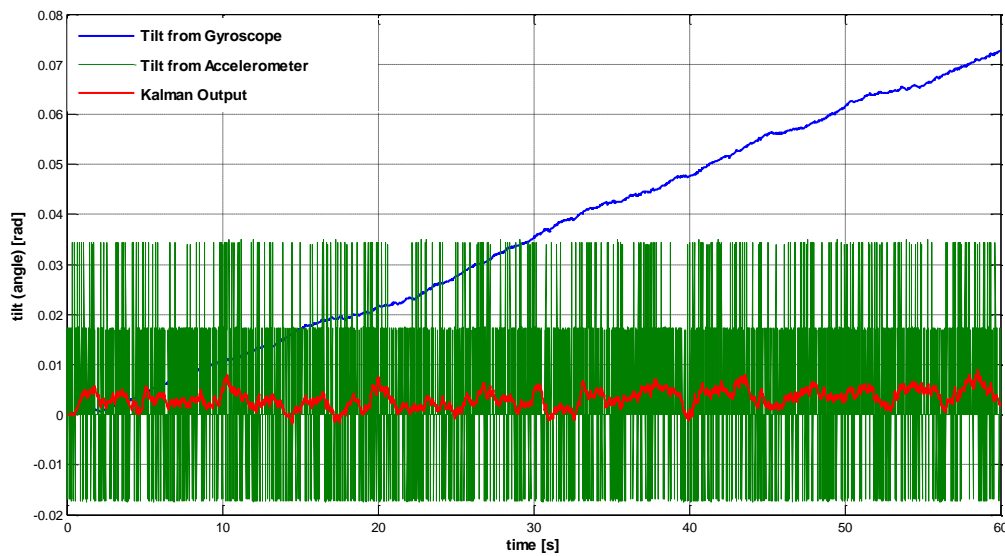
A paraméterek inicializálását a Kálmán-szűrő futtatási követi. A szimulációs eredmények a 3.31. és 3.32. ábrán láthatók. Az algoritmus egy ciklusa az 3.13. ábra alapján:

- *a priori* állapotok becslése és a becslési hiba kovariancia mátrixának kiszámítása,
- a Kálmán-erősítés kiszámítása, az *a posteriori* állapotbecslés előállítás és a becslési hiba kovariancia mátrixának frissítése.

3.1.2.3.4 Mérési eredmények



3.31. ábra – Mérési eredmény: a Kálmán-szűrés eredménye



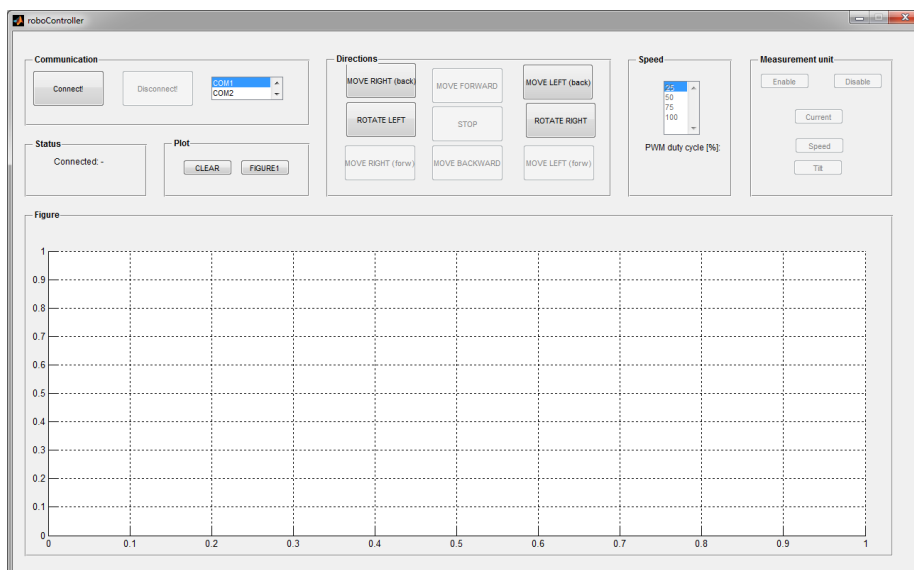
3.32. ábra – Mérési eredmény: a Kálmán-szűrés eredménye nyugalmi (mozdulatlan) állapotban

A szimulációs eredményeken látható, hogy önmagában a giroszkóp vagy gyorsulásmérő szenzor alkalmatlan a dőlésszög pontos meghatározására. A legkellemetlenebb hatások közé a giroszkóp csúszási jelensége és a gyorsulásmérő zajos kimenete tartozik. Az említett problémák a szenzorfüzióval és Kálmán-szűréssel orvosolhatók. A 3.31. ábra és 3.32. ábra kimutatta, hogy az állapotbecsléssel előállított Kálmán-kimenet pontosabb és sokkal alkalmasabb (megfelelőbb) dőlésszög meghatározást tesz lehetővé a magasabb szintű irányítások számára.

A szűrési eredmények:

- a kiszámolt és beállított Kálmán-paraméterekkel a kimeneten megjelenő dőlésszöget gyors válaszidő és kis túllövés jellemzi,
- az állapotbecsléssel előállított kimenet simább és kisebb a szórása, hiszen a Kálmán-kimenet \hat{x}_1 állapotának (*a* becsült dőlésszög) varianciája és a gyorsulásmérő *y* kimenetének *R* varianciája közötti nagyságrent:

$$\frac{R}{\text{VAR}(\hat{x}_1)} = \frac{1.3128 \cdot 10^{-4} \text{ rad}^2}{2.5578 \cdot 10^{-6} \text{ rad}^2} \cong 50. \quad (3.37)$$



3.33. ábra – A felhasználói program felülete

3.3 A beágyazott szoftverek

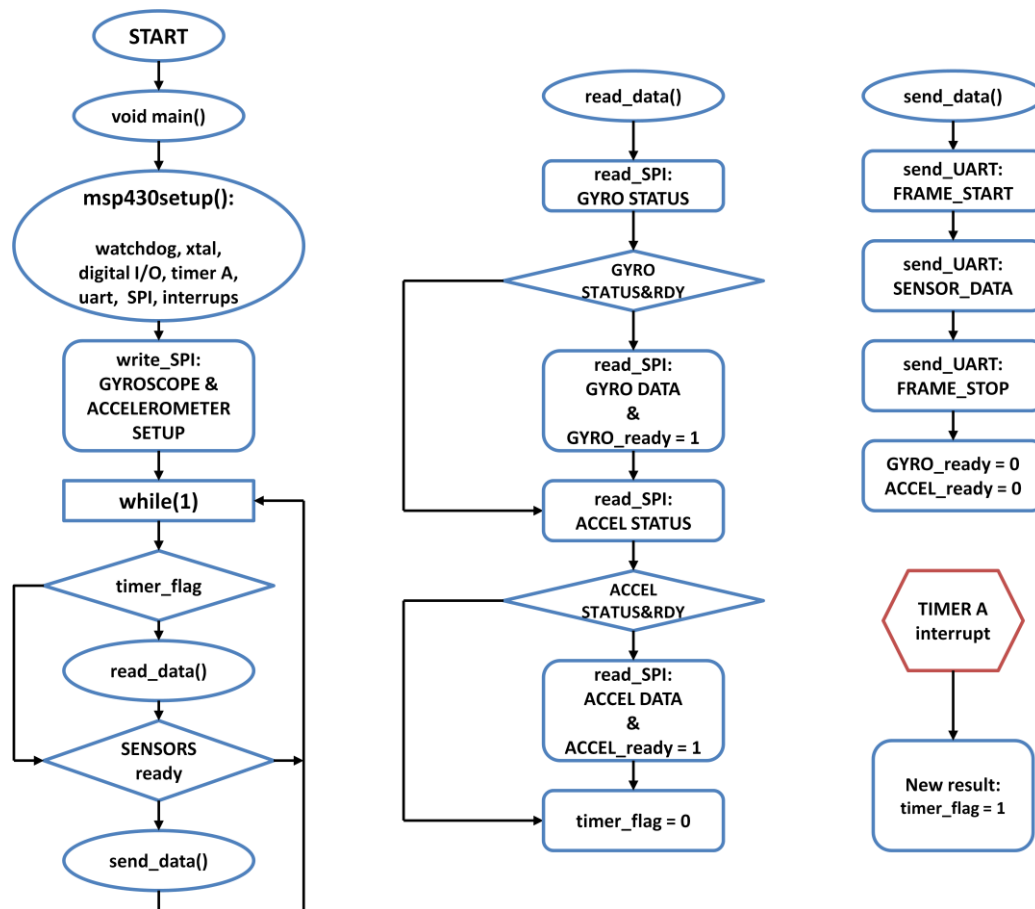
A mikrovezérlőkre írt beágyazott szoftverek folyamatábráit a 3.34. és 3.35. ábrák mutatják. A következőkben részletes ismertetésre kerülnek az *A* és *B* mikrokontrolleren futó programok működése.

3.3.1 A *B* mikrokontrolleren futó beágyazott szoftver

A fejlesztés első fázisában a mikrokontrolleren futó program nem tartalmazza a Kálmán-szűrő algoritmust. Ebben a fázisban a feladat a gyorsulásmérő és giroszkóp áramkörök felprogramozása, a *nyers* mérési eredmények ciklikus kiolvasása és továbbküldése. A

számítógép soros vonalán érkező mérési eredmények fogadása, feldolgozása és megjelenítése MATLAB segítségével történik. A mérési adatok feldolgozását és kiértékelését a Kálmán-szűrő paramétereinek meghatározása és a szűrő algoritmus implementálása követi MATLAB környezetben. A fejlesztés második fázisa a szűrő algoritmus beágyazása mikrokontrolleres rendszerbe, de ez a lépés csak akkor következik, ha a szűrő paraméterek és a szimulációs eredmények megfelelőek (*kis túllövés, gyors válaszdő, stb.*).

A mikrokontrolleren futó beágyazott program indulásakor az első feladat a perifériák inicializálása (*msp430setup*). A perifériák inicializálását a gyorsulásmérő és giroszkóp szenzorok beállítása követi (*méréstartomány, mintavételezési frekvencia, XYZ tengelyek engedélyezése, stb.*). Ennek hatására a szenzorok meghatározott időközönként új mérési eredményeket szolgáltatnak. A mikrokontroller feladata ezen mérési eredmények begyűjtése és továbbítása soros vonalon. Ez a feladat a program végtelen ciklusába hajtodik végre (*while*). Adott frekvenciával a számláló megszakítás egy olvasás engedélyező zászlót *igaz értékre* állít (*timer_flag*). Ennek következtében lehetőség nyílik a szenzorok státusz regiszterének olvasására és a mérési eredmények begyűjtésére. Ha mind a gyorsulásmérő mind pedig a giroszkóp aktuális mérési eredményét sikerült összegyűjteni egy erre a célra alkalmas vektor változóba, akkor a két szenzor zászlója logikai 1 értéket vesz fel (*GYRO_ready, ACCEL_ready*), ez engedélyezi az adatok továbbítást soros vonalon. Az adattovábbítási függvény pedig soros vonalon a hasznos információt *egy-egy bájtnyi* kerettel kiegészítve továbbítja. A beágyazott szoftver folyamatábrája a 3.34. ábrán látható.



3.34. ábra – A B mikrokontrolleren futó beágyazott szoftver folyamatábrája

wachdog_init():

Ebben az alkalmazásban a watchdog timer ki van kapcsolva. A fejlesztés során feltételezhető, hogy a rendszer nem fog végtelen váróciklusba kerülni.

digital_IO_init():

Ez a függvény szolgál a mikrokontroller portjainak beállítására. A rendelkezésre álló portok közül:

- A negyedik port megfelelő vonalai (*P4.0 - P4.1*) szolgálnak a szenzorok kiválasztó jelének generálására. Ezek a port lábak kimenetként vannak definiálva és a mikrokontroller mindaddig magas szinten tartja a vonalakat, míg egy szenzort meg nem szólít (*aktív alacsony*). Ekkor a megfelelő vonalat alacsony logikai értékre állítja.
- A harmadik port megfelelő vonalai periféria módban üzemelnek (*UART: P3.4 - P3.5, P3.6 - P3.7, SPI: P3.1 - P3.3*). Ezen vonalakkal alakítható ki a soros összeköttetés az *FTDI* interfésszel, az *A mikrokontrollerrel* és a *MEMS* szenzorokkal.

xtal_init():

A függvény a mikrokontroller külső órajelforrását inicializálja. Az órajelforrás egy magas frekvenciás kristály oszcillátor (*XT2*), amely *16 MHz*-en rezeg.

uart_init():

A függvény a soros kommunikáció konfigurálását teszi lehetővé a *B mikrokontroller* és az *FTDI*, valamint a *B mikrokontroller* és az *A mikrokontroller* között. A kommunikáció paraméterei:

- *UART0 (FTDI)*: adatátviteli sebesség: *115200 bps*
- *UART1 (MCU A)*: adatátviteli sebesség: *230400 bps*
- Adatbitek száma: *8*
- Paritás: *-*
- Stop bitek száma: *1*

spi_init():

A függvény a sorok kommunikáció konfigurálását teszi lehetővé a szenzorok és a *B mikrokontroller* között. Az *SPI* kommunikáció sebessége *8 MHz*.

timer_A_init():

A mikrokontroller két *16 bites* számlálóval rendelkezik, ezek *A és B számlálók*. Az *A* számláló a szenzorok státuszregisztereinek olvasási gyakoriságát határozza meg (*timer_flag felhúzása*). A számláló inicializálása a következő:

- Forrása a *16 MHz*-es órajel, regiszterének (*TAR*) két értéke között eltelt idő *62.5 ns*.
- Felfelé számláló (*up mode*) módban üzemel. Ennek következtében ismétlődően számol *0x0000*-től *CCR0* regiszter értékéig.
- A *TAR* regiszter program-megszakítást okoz, ha a *CCR0* értékéről nullára vált. Így *CCR0* értékének megválasztásával az olvasási gyakoriság állítható be.
- A szenzorok mintavételezési frekvenciáját *400 Hz*, a *8 kHz*-es olvasási gyakoriság elegendőnek bizonyult, így a *CCR0* értéke:

$$CCR0 = \frac{16MHz}{8kHz} = 2000. \quad (3.38)$$

interrupts_init():

A függvény az általános megszakításkéréseket engedélyezi.

gyro_accel_setup():

A függvény a giroszkóp és gyorsulásmérő szenzorok felprogramozását teszi lehetővé. Ennek során a megfelelő mintavételi frekvencia, megszakítás kérések, méréstartomány és az XYZ tengelyek engedélyezésére kerül sor. A felprogramozás a már beállított SPI interfészen keresztül történik a megfelelő engedélyező jel (*chip select*) alacsonyra logikai értékre állításával és az elvárt működést biztosító adatbájtok elküldésével. A szenzorok felprogramozásának eredményei:

- a mintavételezési frekvencia mindkét szenzornál: 400 Hz,
- a gyorsulásmérő méréstartománya: $\pm 2g$,
- a gyorsulásmérő Y és Z tengelyének engedélyezése,
- a giroszkóp méréstartománya: $\pm 250 dps$,
- a giroszkóp X tengelyének engedélyezése,
- és a megszakításkérő funkciók kikapcsolása.

read_data():

A függvény kiértékelésére akkor kerül sor, ha a számláló megszakítási rutinban az olvasást engedélyező zászló logikai magas értéket vesz fel (*timer_flag*). Ez minden 0.125 ms-ban (8 kHz) megtörténik. Ennek következtében kiolvashatók a gyorsulásmérő és giroszkóp szenzorok státuszregisztereinek értékei. Amennyiben új mérési eredmény áll rendelkezésre, akkor az erre alkalmas vektor változóban azok eltárolódnak és az új mérési eredmény létét jelző zászlók igaz értéket vesznek fel. A következőkben tárgyalt függvény ezen zászlók igaz értékére vár. Ha az aktuális *read_data* függvény híváskor nincs új mérési eredmény, a procedúra a következő 0.125 ms-ban ismét végrehajtódik. Ezt a következő programkód szemlélteti:

```
if(timer_flag)
{
    rx = spi_sread(GYRO_CS, STATUS_REG+SREAD);
    if((rx&0x000F) == 0x0F) && !gyro_flag
    {
        results[2] = spi_sread(GYRO_CS, OUT_X_L+SREAD);
        results[3] = spi_sread(GYRO_CS, OUT_X_H+SREAD);
        gyro_flag = 1;
    }

    rx = spi_sread(ACCEL_CS, STATUS_REG+SREAD);
    if((rx&0x000F) == 0x0F) && !accel_flag
    {
        results[0] = spi_sread(ACCEL_CS, OUT_Y+SREAD);
        results[1] = spi_sread(ACCEL_CS, OUT_Z+SREAD);
        accel_flag = 1;
    }
    timer_flag = 0;
}

// ...

// IT: TIMER_A @ 8 kHz
#pragma vector=TIMERAO_VECTOR
__interrupt void Timer_A (void)
{
    timer_flag = 1;
}
```

send_data():

A függvény az összegyűjtött mérési eredmények továbbítására szolgál. Akkor értékelődik ki, ha mind a giroszkóp, mind pedig a gyorsulásmérő mérési eredményét jelző zászló értéke igaz. A függvény kiértékelődésének gyakorisága (*körülbelül*) a szenzorok mintavételezési frekvenciájával egyezik meg. A függvényhez tartozó forráskód:

```
if(gyro_flag && accel_flag && transfer)
{
    while (!(IFG2&UCA0TXIFG));
    UCA0TXBUF = FRAME_START;

    for(i=0; i<4; i++)
    {
        while (!(IFG2&UCA0TXIFG));
        UCA0TXBUF = results[i];
    }

    while (!(IFG2&UCA0TXIFG));
    UCA0TXBUF = FRAME_STOP;

    accel_flag = 0;
    gyro_flag = 0;
}
```

3.3.2 Az A mikrokontrolleren futó beágyazott szoftver

Az A mikrokontrolleren futó beágyazott program feladata a felhasználói programból küldött parancsok végrehajtása (*motorok vezérlése*), a motorokra szerelt forgó jeladók jeleinek feldolgozása, az A/D átalakítók bemenétre kerülő feszültségeket mintavételezése, a B mikrokontrollertől érkező adatok fogadása és a mérési eredmények folyamatos továbbítása a számítógép felé.

A program indulásakor a mikrokontroller perifériáit inicializálja (*msp430setup*). A perifériák beállítását a program végtelen ciklusa követi (*while*), amely két feltételes vezérlési szerkezetet tartalmaz.

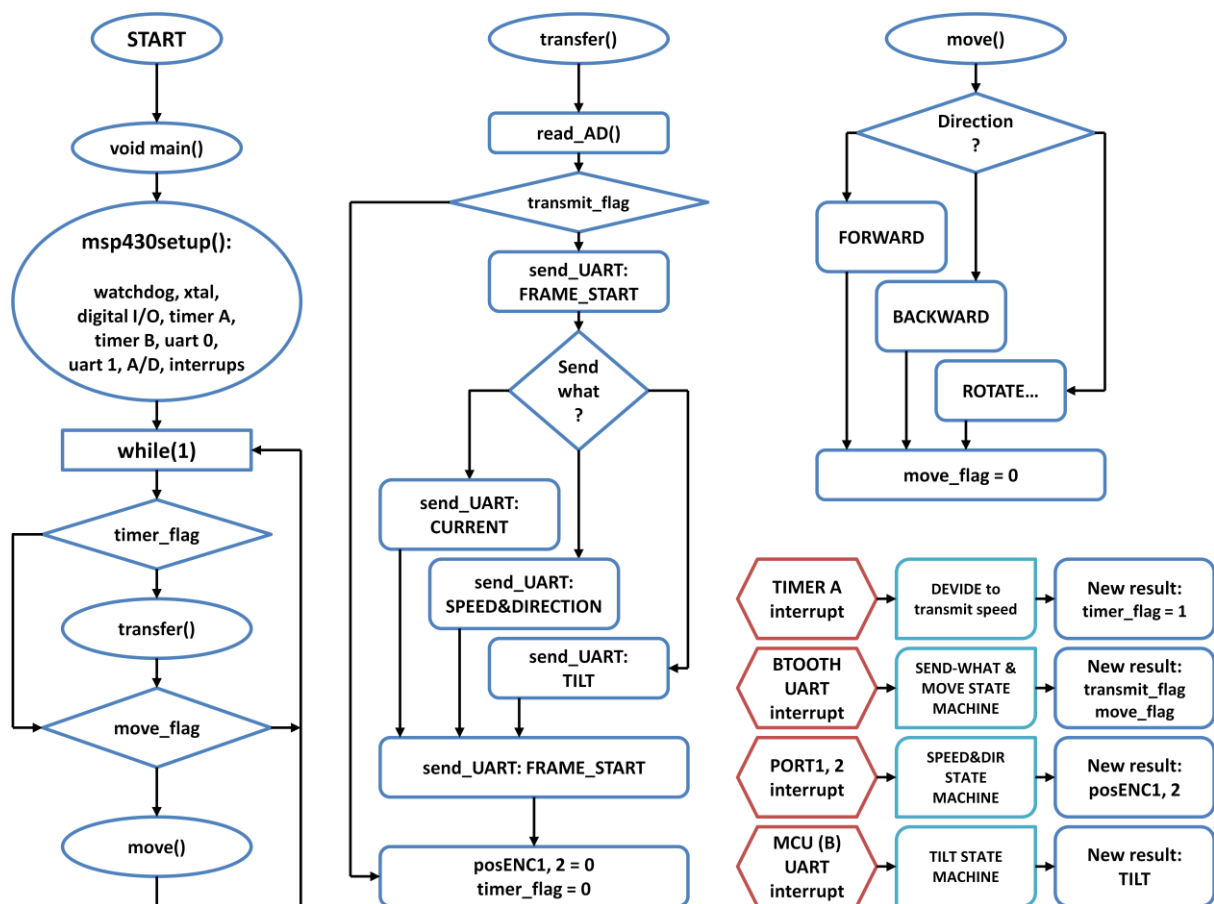
Az első feltételes szerkezet az adattovábbításért és az A/D bemenetek mintavételezéséért felelős (*transfer*). A függvény időzítését az A számláló megszakítási rutinja végzi (*a timer_flag magas logikai értékre állítása*), tehát az A számlálóval beállított megszakítási gyakoriság határozza meg mind az adattovábbítási gyakoriságot, mind pedig az A/D bemenetek mintavételi frekvenciáját. A függvény meghívásával az A/D bemenetek mintavételezésére és a felhasználó által kiválasztott dinamikai tulajdonságok elküldésére kerül sor (*feltéve, hogy ezt megelőzően a felhasználó engedélyezte az adattovábbítást – transmit_flag*).

A másik feltételes szerkezet a robot irányításáért felelős. A függvény meghívásakor a felhasználó által kiválasztott irány és sebesség függvényében a mikrokontroller előállítja a megfelelő PWM jeleket (*feltéve, hogy ezt megelőzően a felhasználó engedélyezte az irányítást – move_flag*).

A fenti eseményekkel párhuzamosan további megszakítások is érvénybe jutnak. Az előbbieken említett feltételes szerkezetek csak akkor fognak kiértékelődni, ha azt egy érvényes adatsor érkezése előzi meg a soros vonalon. A felhasználtól érkező adatok határása megszakítás keletkezik (*bluetooth uart interrupt*). Az érkező adatsor egy állapotgépen

keresztül értékelődik ki. Attól függően, hogy a felhasználó milyen adatsort küldött az állapotgép vagy az adattovábbítást (*transmit_flag*) vagy az irányítást engedélyezi (*move_flag*). Hasonlóan a *B mikrokontrollertől* érkező adatsor hatására is megszakítás keletkezik, ahol az adatsor (*gyorsulásmérő és giroszkóp mérési eredményei*) feldolgozása állapotgépen keresztül történik.

Mindez mellett, a motor tengelyének forgása közben a forgó jeladók impulzusokat juttatnak az *A mikrokontroller* érzékeny port-bemeneteire. A programban minden felfutó- és lefutó él (*4-szeres kiértékelés*) hatására megszakítás keletkezik (*port1,2 interrupt*). Minden megszakításkor, a megszakítás helyiértékének megfelelő inkremens számláló (*posENC1, 2*) értéke egyel nő vagy csökken. Az inkremens számlálók két adattovábbítási rutin között keletkező relatív elmozdulás értékét tartalmazzák. Minden adattovábbítási rutin végén ezen változók értéke nullára áll vissza. A beágyazott szoftver folyamatábráját a 3.35. ábra mutatja.



3.35. ábra – az *A mikrokontrolleren* futó beágyazott szoftver folyamatábrája

wachdog_init():

Ebben az alkalmazásban a watchdog timer ki van kapcsolva. A fejlesztés során feltételezhető, hogy a rendszer nem fog végtelen váróciklusba kerülni.

xtal_init():

A függvény a mikrokontroller külső órajelforrását inicializálja. Az órajelforrás egy magas frekvenciás kristály oszcillátor (*XT2*), amely 16 MHz-en rezeg.

digital_IO_init():

Ez a függvény szolgál a mikrokontroller portjainak beállítására. A rendelkezésre álló portok közül:

- Az első port vonalai (*P1.6 – P1.7*) bementként vannak definiálva, felfutó és lefutó érzékeny megszakítással. A *P1.x* szolgál a forgó jeladók jeleinek feldolgozására.
- A második port vonalai (*P2.0 – P2.1*) bementként vannak definiálva, felfutó érzékeny megszakítással. A *P2.x* szolgál a forgó jeladók jeleinek feldolgozására.
- A harmadik port vonalai (*P3.4 – P3.5, P3.6 – P3.7*) periféria módban üzemelnek. A *P3.x* segítségével alakítható ki a soros összeköttetés az *A mikrokontroller* és a kommunikációs egység, valamint az *A mikrokontroller* és a *B mikrokontroller* között.
- A negyedik port vonalai (*P4.1 - P4.4*) kimenetként üzemelnek periféria módban. A *P4.x* szolgál az impulzus szélesség modulált jelek előállítására.
- Az ötödik port vonalai (*P5.0 – P5.7*) kimenetként vannak definiálva. Ezen vonalak szolgálnak a motorvezérlők üzembe helyezéséért.
- A hatodik port vonalai (*P6.3 – P6.4*) periféria módban üzemelnek. Az A/D bemenetek szolgálnak a motorok áramának mintavételezésére.

uart_init():

A függvény a soros kommunikáció konfigurálását teszi lehetővé az *A mikrokontroller* és a *B mikrokontroller* között, valamint az *A mikrokontroller* és a Bluetooth egység között. A kommunikáció paramétereit:

- *UART0 (MCU B)*: adatátviteli sebesség: 230400 bps
- *UART1 (BLUETOOTH)*: adatátviteli sebesség: 9600 bps
- Adatbitek száma: 8
- Paritás: -
- Stop bitek száma: 1

timer_A_init():

A mikrovezérlő *A* számlálója az A/D bemenetek mintavételezési frekvenciáját és az adattovábbítás gyakoriságát határozza meg (*timer_flag felhúzása*). A számláló inicializálása a következő:

- Forrása a 16 MHz-es órajel, regiszterének (*TAR*) két értéke között eltelt idő 62.5 ns.
- Felfelé számláló (*up mode*) módban üzemel. Ennek következtében ismétlődően számol 0x0000-tól *CCR0* regiszter értékéig.
- A *TAR* regiszter program-megszakítást okoz, ha a *CCR0* értékéről nullára vált. Így *CCR0* értékének megválasztásával az olvasási gyakoriság állítható be.
- Az A/D bemenetek mintavételezési frekvenciája és az adattovábbítás gyakorisága 100 Hz. A megszakítási rutin gyakoriságát 500 Hz-re állítva, minden ötödik megszakítást szükséges vizsgálni. A *CCR0* értéke:

$$CCR0 = \frac{16 \text{ MHz}}{500 \text{ Hz}} = 32000. \quad (3.39)$$

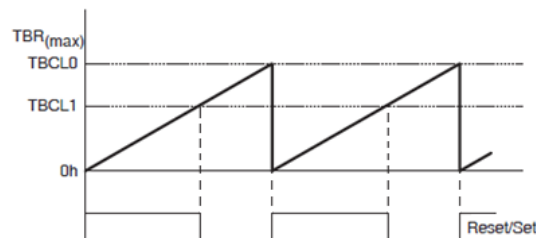
timer_B_init():

A függvény a mikrokontroller *B* számlálóját inicializálja. A számláló a PWM jelek előállítására szolgál, a *compare* mód segítségével.

A B számláló paraméterei:

- Forrása a 16 MHz-es órajel, regiszterének (*TBR*) két értéke között eltelt idő 62.5 ns.
- Felfelé (*up mode*) számláló üzemmód, melyben a számláló ismétlődően számol 0x0000-tól *TBCL0* (*compare latch*) értékéig. A *TBCL0* a PWM jel periódusát határozza meg.
- *TBCL0 = 16*, azaz $t = 1 \mu\text{s}$ szükséges ahhoz, hogy a *TBR* 0x0000-tól a *TBCL0* értékig számoljon el. Ennek megfelelően a PWM jelek frekvenciája 1 MHz.
- A *compare blokk* kimeneti egysége *reset/set* módban üzemel. A *reset/set* mód lehetővé teszi a PWM jelek előállítását. Az aktuális kimeneti vonal:
 - alacsony logikai értékre vált, amikor $TBR = TBCLx$
 - magas logikai értékre vált, amikor $TBR = TBCL0$,ahol *TBCLx* az aktuális kimeneti vonalhoz tartozó regiszter (*compare latch*).

Output Example – Timer in Up Mode



3.36. ábra – a PWM jel előállítása [20]

ad_init():

A függvény, az analóg/digitális átalakítót inicializálja. Az A/D bemenetekre kerülő feszültségből számítható ki a motorok pillanatnyi áramfelvétele. A konverzió a rendelkezésre álló belső referencia feszültséget $U_{ref} = 2.5V$ használja fel.

interrupts_init():

A függvény az általános megszakításkéréseket engedélyezi.

transfer():

A függvény akkor értékelődik ki, ha az A számláló megszakítási rutinban az adattovábbítást és az A/D bemenetek mintavételezését engedélyező zászló igaz értéket vett fel (*timer_flag*). Ez minden 2 ms-ban (500 Hz) lép érvénybe. Ennek következtében az A/D bemenetek mintavételezése és a kiválasztott dinamikai tulajdonság továbbítására kerül sor (*feltéve, hogy a felhasználó engedélyezte az adattovábbítást – transmit_flag*). A felhasználó a megfelelő adatsort továbbítva, ki tudja választani, hogy mit akar mérni. Ezt a túloldalon a mikrokontroller állapotgépe értékeli ki.

```
if(timer_flag)
{
    currents[0] = 0x00FF & ADC12MEM0; //L
    currents[1] = (0x0F00 & ADC12MEM0)>>8; //H
    currents[2] = 0x00FF & ADC12MEM1; //L
    currents[3] = (0x0F00 & ADC12MEM1)>>8; //H
    ADC12CTL0 |= ADC12SC;

    if(transmit_flag && ((transfer[0] == TRANS_C) ||
        (transfer[0] == TRANS_S) || (transfer[0] == TRANS_T)))
```

```

{
  while (!(UC1IFG&UCA1TXIFG));
  UCA1TXBUF = FRAME_START;

  switch(transfer[0])
  {
    case TRANS_C: // transmit: CURRENTS
      for(i=0; i<4; i++)
      {
        while (!(UC1IFG&UCA1TXIFG));
        UCA1TXBUF = currents[i];
      }
      break;
    case TRANS_S: // transmit: POSITIONS
      for(i=0; i<4; i++)
      {
        while (!(UC1IFG&UCA1TXIFG));
        UCA1TXBUF = positions[i];
      }
      break;

    case TRANS_T: // transmit: GYRO&ACCEL
      if(tilt[4])
      {
        for(i=0; i<4; i++)
        {
          while (!(UC1IFG&UCA1TXIFG));
          UCA1TXBUF = tilt[i];
        }
        tilt[4]=0;
      }
      break;
  }

  while (!(UC1IFG&UCA1TXIFG));
  UCA1TXBUF = FRAME_STOP;
}

posEnc1 = 0;
posEnc2 = 0;
timer_flag = 0;
}
}

// ...

// IT: TIMER_A @ 500 Hz
#pragma vector=TIMERA0_VECTOR
__interrupt void Timer_A (void)
{
  if(timer == 4) // 100 Hz
  {
    timer_flag = 1;
    timer = 0;
  }
  else
    timer++;
}
}

```

move():

A függvény akkor értékelődik ki, ha azt megelőzően a felhasználtól érkezett adatsor hatására az állapotgép az irányítást engedélyező zászlót logikai magas értékre állította (*move_flag*). Ennek hatására a felhasználó által kiválasztott irányú és sebességű mozgást fog végezni a robot. A függvény feladata a motorok vezérléséhez szükséges PWM jelek előállítását.

```

if(move_flag)
{
    switch(move[0])
    {
        case MOVE_F:           // MOVE: FORWARD
            TBCCR2 = move[1];  // PWM DUTY CYCLE
            TBCCR4 = move[1];
            TBCCR1 = 0;
            TBCCR3 = 0;
            break;
        case ROTATE:          // MOVE: ROTATE
            TBCCR2 = move[1];
            TBCCR3 = move[1];
            TBCCR1 = 0;
            TBCCR4 = 0;
            break;
        case MOVE_B:          // MOVE: BACKWARD
            TBCCR1 = move[1];
            TBCCR3 = move[1];
            TBCCR2 = 0;
            TBCCR4 = 0;
            break;

        // ...

        default:
            motor_stop();
    }
    move_flag = 0;
}

```

uart_rx_interrupt:

A soros vonalon érkező adat hatására megszakítás keletkezik. A megszakítási rutinban minden érkező adat egy állapotgépen halad végig. Megfelelő adatsor érkezését követően az állapotgép engedélyező jelet ad a kívánt parancs végrehajtására. Példa egy irányítási parancsot fogadó állapotgépre:

```

// IT: BLUETOOTH_UART_RX
#pragma vector=USCIAB1RX_VECTOR
__interrupt void USCI1RX_ISR(void)
{
    char rx;
    rx = UCA1RXBUF;
    rx_state(rx);
}

void rx_state(char rx)
{
    if((rec_stat == 0) && (rx == FRAME_START))
        rec_stat = 1;
    else if ((rec_stat == 1) && ((rx == MOVE_F) || (rx == MOVE_B) ||
        (rx == ROTATE)))
    {
        move[0] = rx;    // direction
        rec_stat = 2;
    }
    else if ((rec_stat == 2) && (rx > 64) && (rx < 69))
    {
        move[1] = rx;    // pwm duty
        rec_stat = 3;
    }
    else if ((rec_stat == 3) && (rx == FRAME_STOP))
    {
        move_flag = 1;    // data ready
        rec_stat = 0;
    }
}

```

```

}
else
{
    rec_stat = 0;
    motor_stop();
}
}

void motor_stop()
{
    TBCCR3 = 0;
    TBCCR1 = 0;
    TBCCR2 = 0;
    TBCCR4 = 0;
}

```

port1, 2 interrupt:

A forgó jeladók jelei az első és második portra csatlakoznak. Minden felfutó és lefutó él hatására megszakítás keletkezik. Az érzékeny megszakítási rutin megvizsgálja, hogy melyik vezetéken érkezett megszakítás, mely alapján növeli vagy csökkenti az inkremens számláló értékét. A forgó jeladó két jelet szolgáltat (*channel A és B*). Az A és B csatorna egymástól geometriailag 90 fokra helyezkedik el. A feldolgozott jelek segítségével a relatív pozíció és forgásirány határozható meg a 3.10. és 3.12. ábra alapján.

Az algoritmust a 3.2. táblázat írja le.

		A		B	
		0	1	0	1
A	0 → 1			posENC++	posENC--
	1 → 0			posENC--	posENC++
B	0 → 1	posENC--	posENC++		
	1 → 0	posENC++	posENC--		

3.2. táblázat – Az inkrementális jeladó jeleinek négyszeres kiértékelése

```

// Port2 interrupt service routine
#pragma vector = PORT2_VECTOR
__interrupt void Port2_ISR(void) {
    if (P2IFG & ENC2A) { // chanA IT <low-to-high, high-to-
low>
        if (P2IES & ENC2A){ // chanA IT @ high-to-low
            if (P2IN & ENC2B) posEnc2++; // dir1
            else posEnc2--; // dir2
        }
        else { // chanA IT @ low-to-high
            if (P2IN & ENC2B) posEnc2--; // dir2
            else posEnc2++; // dir1
        }
        P2IES ^= ENC2A; //invert
    }
    else { //chanB IT <low-to-high, high-to-low>
        if (P2IES & ENC2B){ //chanB IT @ high-to-low
            if (P2IN & ENC2A) posEnc2--; //dir2
            else posEnc2++; //dir1
        }
        else { //chanB IT @ low-to-high
            if (P2IN & ENC2A) posEnc2++; //dir1
            else posEnc2--; //dir2
        }
        P2IES ^= ENC2B; //invert
    }
}

```



```
}  
P2IFG = 0; //clear  
}
```

3.4 Összegzés

A beágyazott szoftver és a felhasználói felület a robot vezérlési feladatainak elvégzésére szolgál, valamint lehetővé teszi a robot dinamikai jellemzőinek rögzítését. A következő fejezet a 2. *fejezetben* bemutatott matematikai modell verifikációjáról szól. A megépített hardver és a hozzá tartozó beágyazott és felhasználói szoftverek segítségével mindazt amit az előzőekben egyenletekkel felírtunk és szimuláltunk, most mérésekkel is igazoljuk (*alátámasztjuk*).

4. A szimulációs eredmények verifikációja

A matematikai modell ismerete lehetővé teszi a robot dinamikai viselkedésének szimulálását, valamint az elméletileg megalapozott szabályozások tervezését.

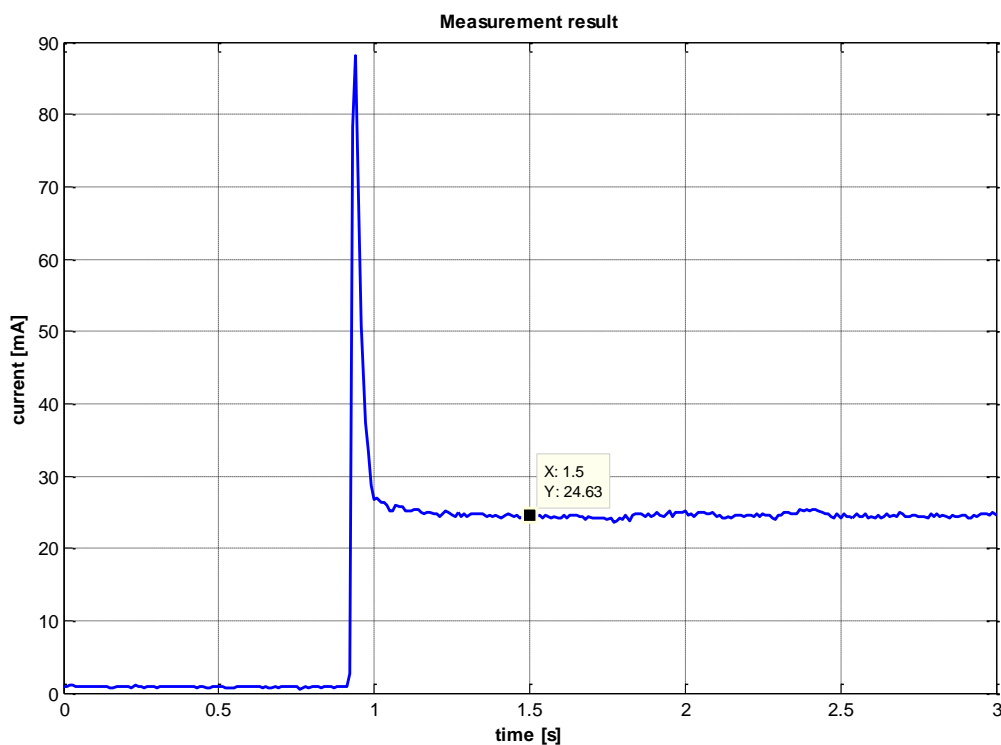
- De vajon a szimulációs eredmények pontosak? Attól még, hogy jellegre azt kaptuk, amit vártunk, még nagy eltérések lehetnek a valós eredmények és a szimulációs eredmények között.

Mivel a szerkezet teljes dinamikus viselkedése mérhető (*mintavételezhető*), ezért lehetőségünk van a matematikai modell verifikációjára. A verifikációval bebizonyítható az, hogy a szimulációs eredmények a valós működést írják le, így a továbbiakban a matematikai modell teljes egészében alkalmas szabályozók tervezésére.

A mérések során alkalmazott gerjesztő feszültség $U_t = 3V$, a mintavételi frekvencia pedig $f_s = 100 \text{ Hz}$. A következő ábrák a mérési eredményeket tartalmazzák (*összehasonlítva a szimulációs eredményekkel*).

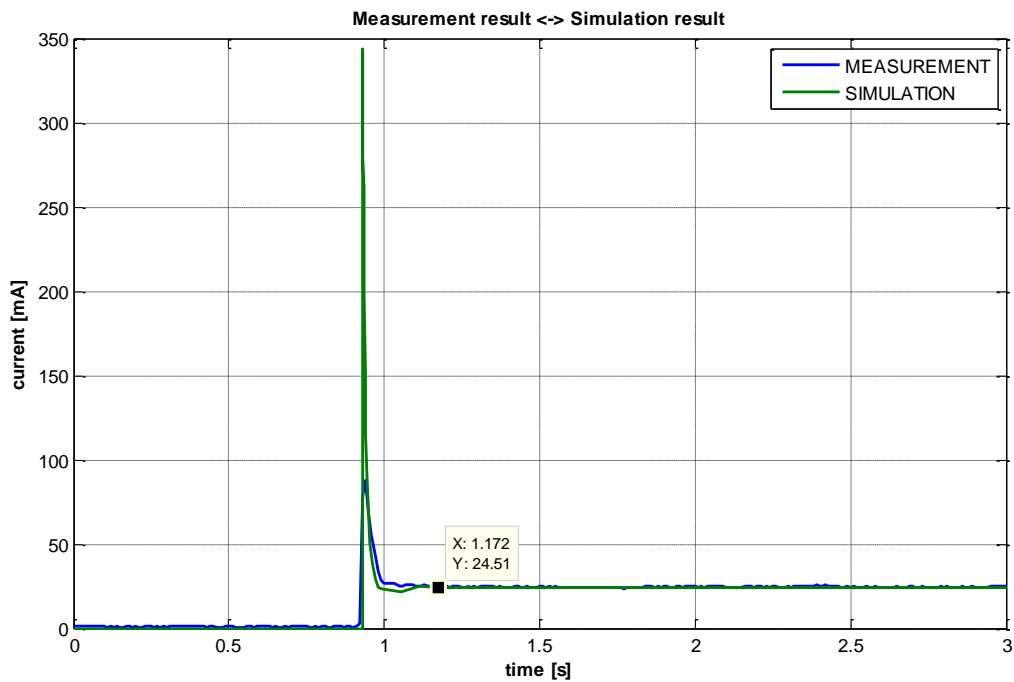
4.1 A motorok áramfelvétele üresjáratban

Mérési eredmény:



4.1. ábra – Mérési eredmény: a motorok áramfelvétele üresjáratban

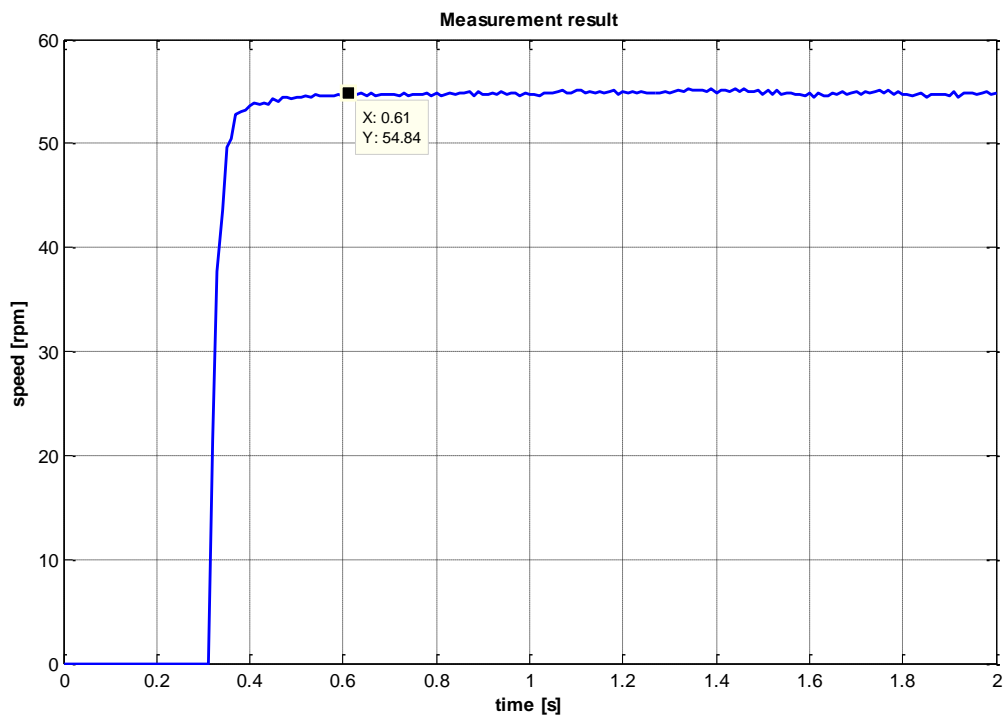
A mérési eredmény összevetése a szimulációs eredménnyel:



4.2. ábra – A szimulációs- és mérési eredmények: a motorok áramfelvétele

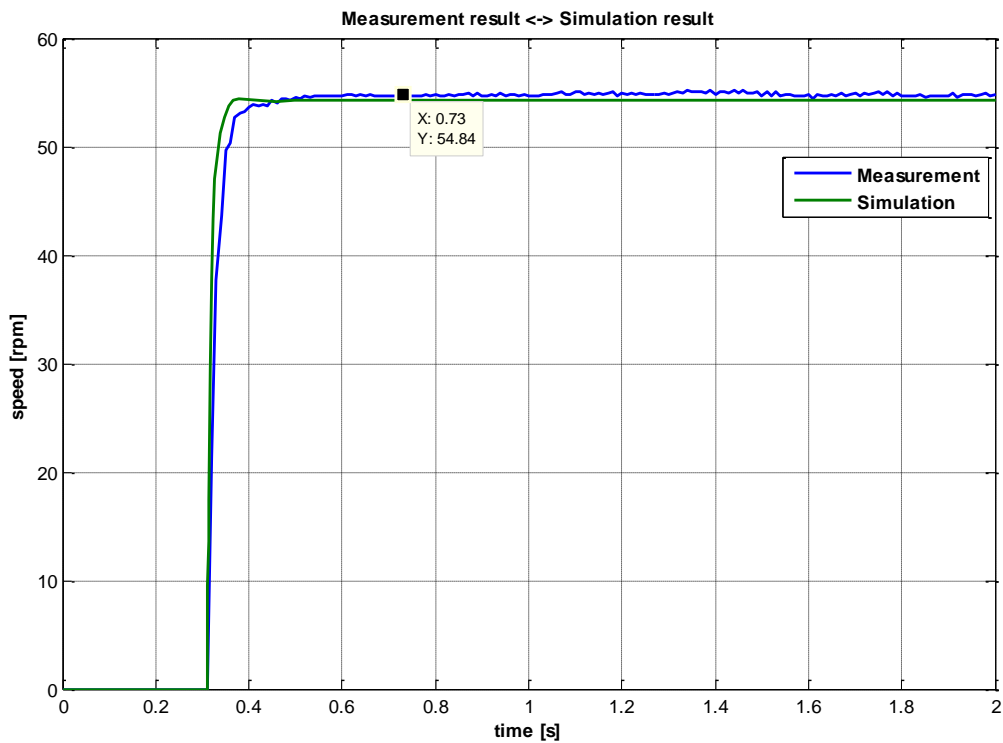
4.2 A motorok forgási sebessége üresjáratban

Mérési eredmény:



4.3. ábra – Mérési eredmény: a motorok forgási sebessége üresjáratban

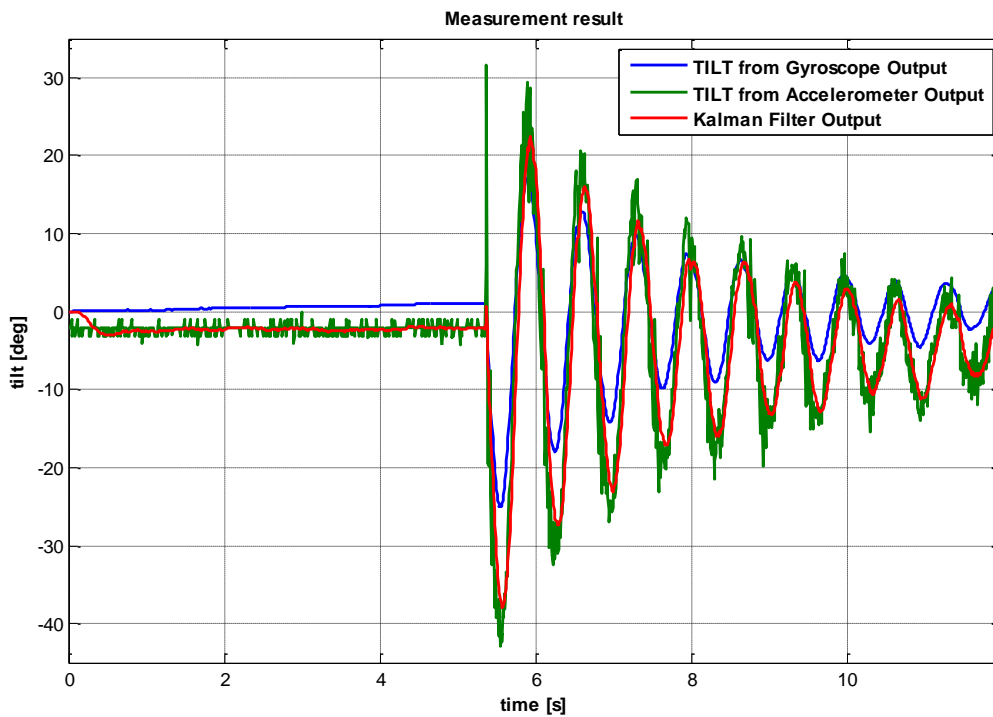
A mérési eredmény összevetése a szimulációs eredménnyel:



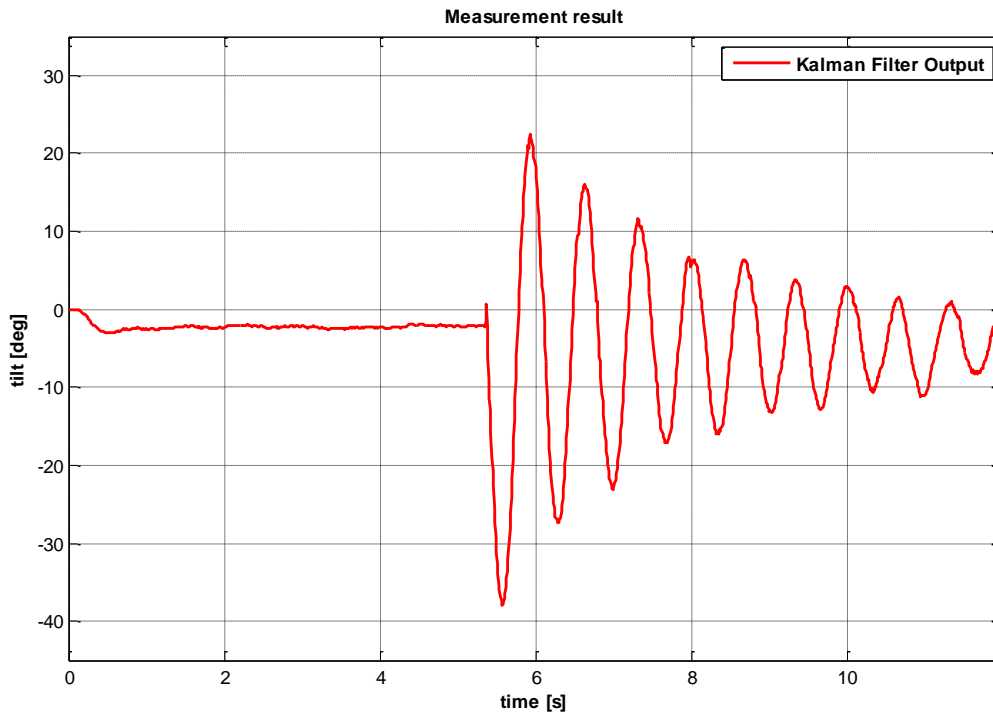
4.4. ábra – A szimulációs- és mérési eredmények: a motorok forgási sebessége

4.3 A belső test kilengése a robot transzlációs mozgása közben

Mérési eredmény:

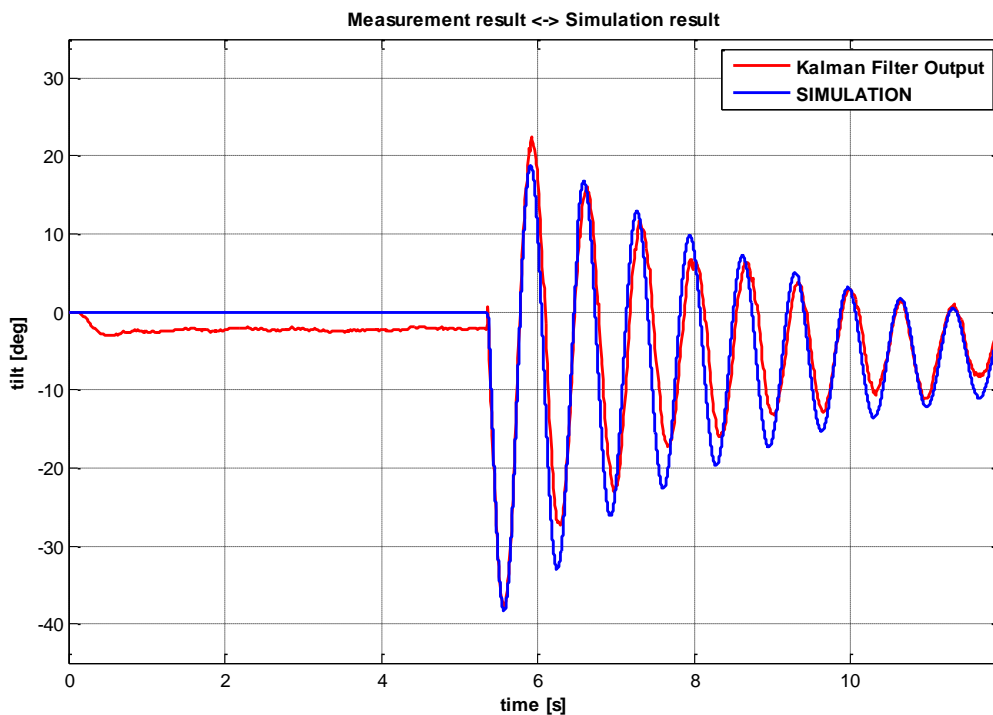


4.5. ábra – Mérési eredmény: a belső test kilengése (pillanatnyi dőlésszög mérése) (giroszkóp (kék), gyorsulásmérő (zöld), Kálmán (piros))



4.6. ábra – Mérési eredmény: a belső test kilengése (csak a Kálmán-szűrő kimenete)

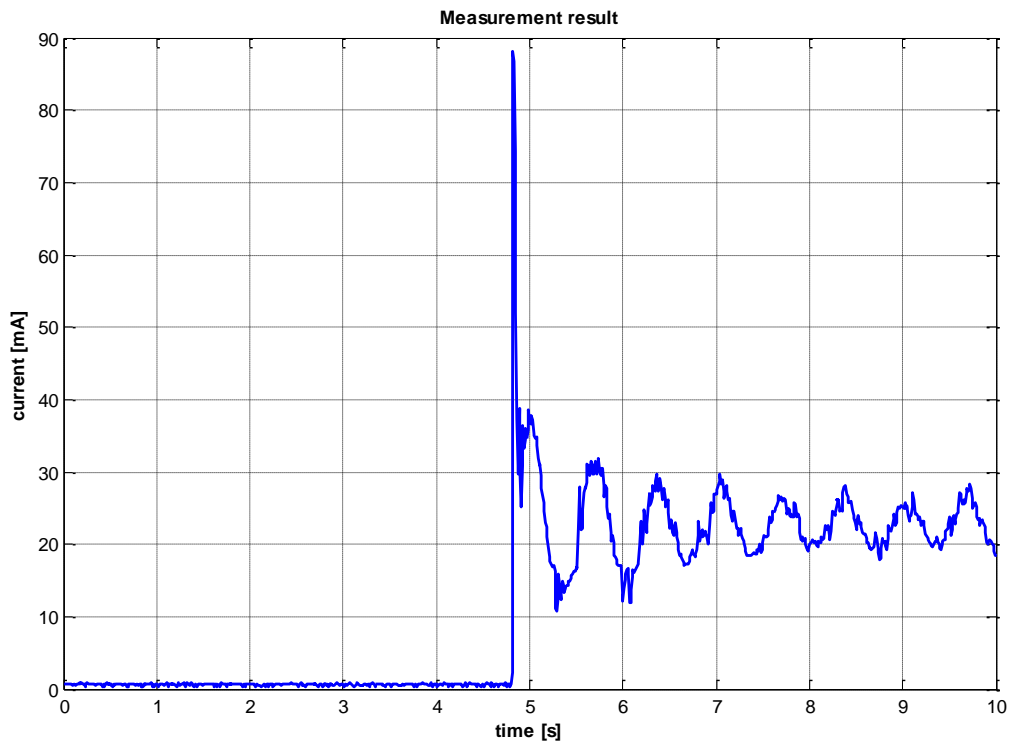
A mérési eredmény összevetése a szimulációs eredménnyel:



4.7. ábra – A szimulációs- és mérési eredmények: a belső test kilengése a robot transzlációs mozgása közben

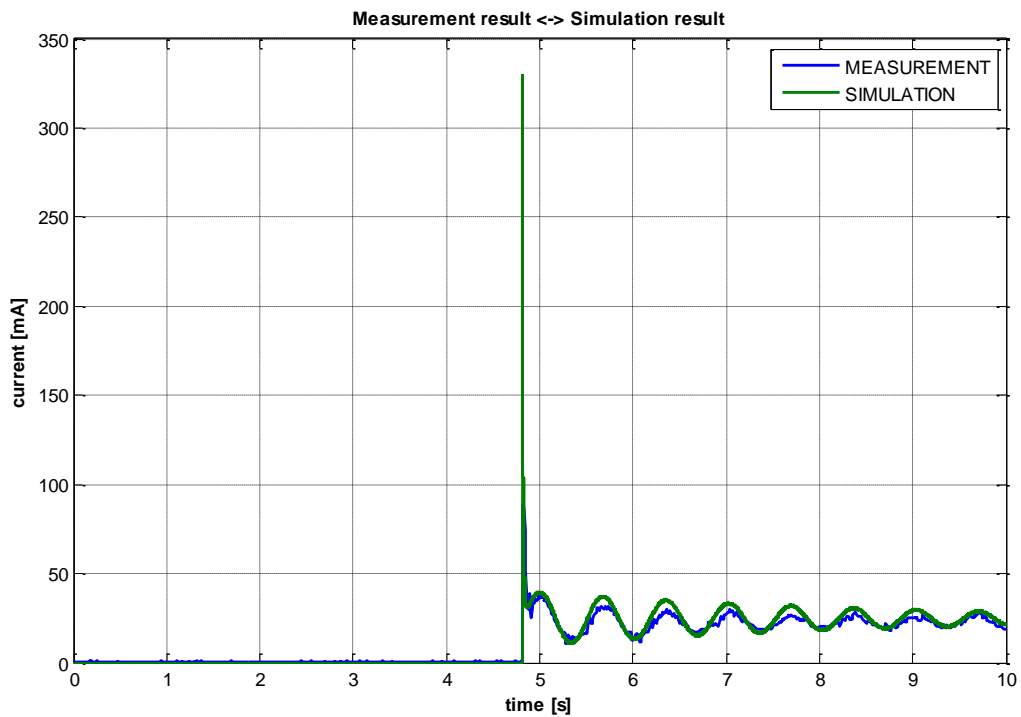
4.4 A motorok áramfelvétele mozgás közben

Mérési eredmény:



4.8. ábra – Mérési eredmény: a motorok áramfelvétele a robot transzlációs mozgása közben

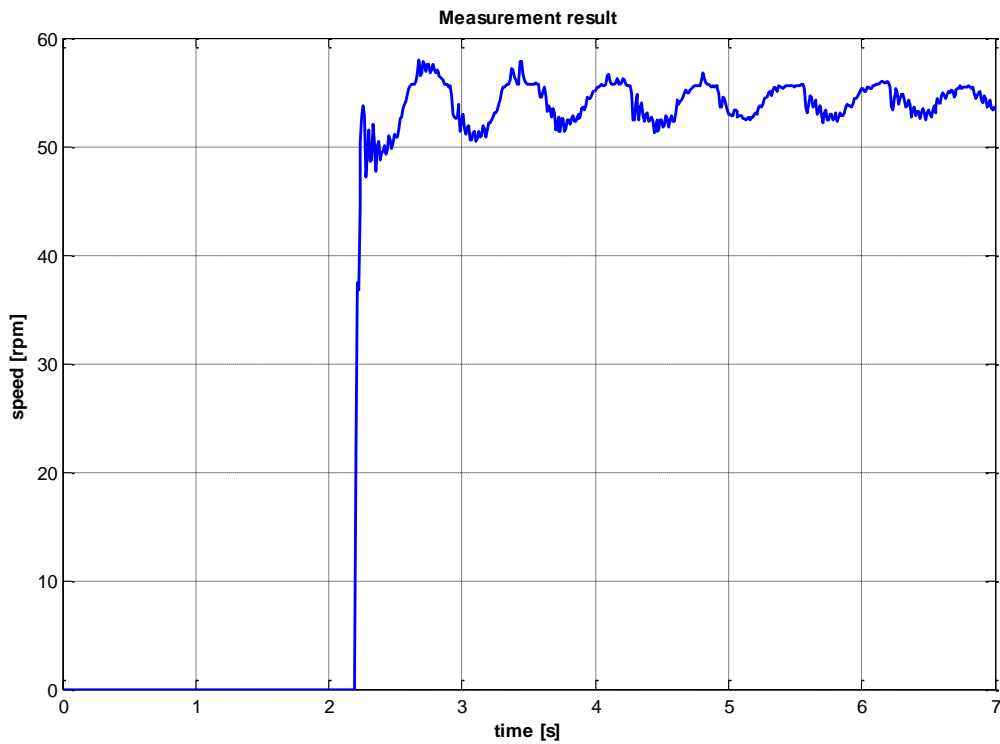
A mérési eredmény összevetése a szimulációs eredménnyel:



4.9. ábra – A szimulációs- és mérési eredmények: a motorok áramfelvétele mozgás közben

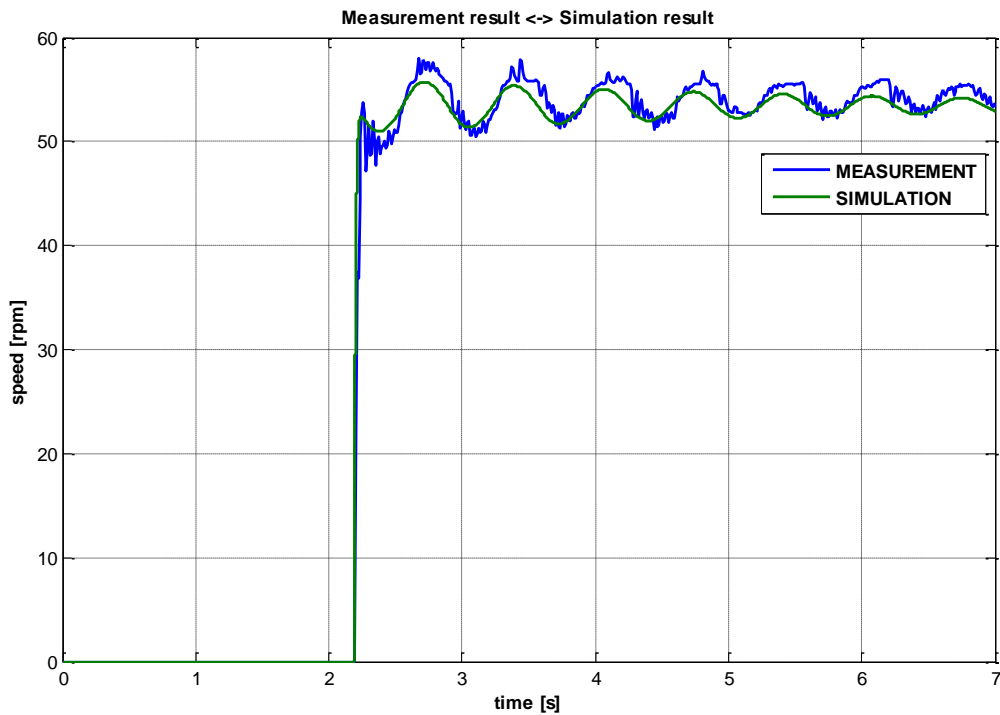
4.5 A motorok forgási sebessége mozgás közben

Mérési eredmény:



4.10. ábra – Mérési eredmény: a motor forgási sebessége a robot translációs mozgása közben

A mérési eredmény összevetése a szimulációs eredménnyel:



4.11. ábra – A szimulációs- és mérési eredmények: a motor forgási sebessége mozgás közben

4.6. Összegzés

A mérési eredmények igazolták (*bebizonyították*) a második fejezetben bemutatott matematikai modell helyességét. Eltérések ugyan jelentkeznek a mért- és szimulált eredmények között, de ez elhanyagolhatónak tekinthető. A verifikáció kimutatta, hogy a második fejezet matematikai modellje tökéletesen alkalmas az elméletben megalapozott szabályozók tervezéséhez.

5. Összegzés

Munkám során bemutattam, hogy melyek azok a fejlesztési fázisok, amelyek szükségesek egy mobilis robot megépítéséhez. A robot fejlesztési lépésein keresztül gondosan rögzítettem minden apró momentumot, a modellezéstől, illetve az általános döntések meghozatalától kezdve, egészen a konkrét megvalósításig.

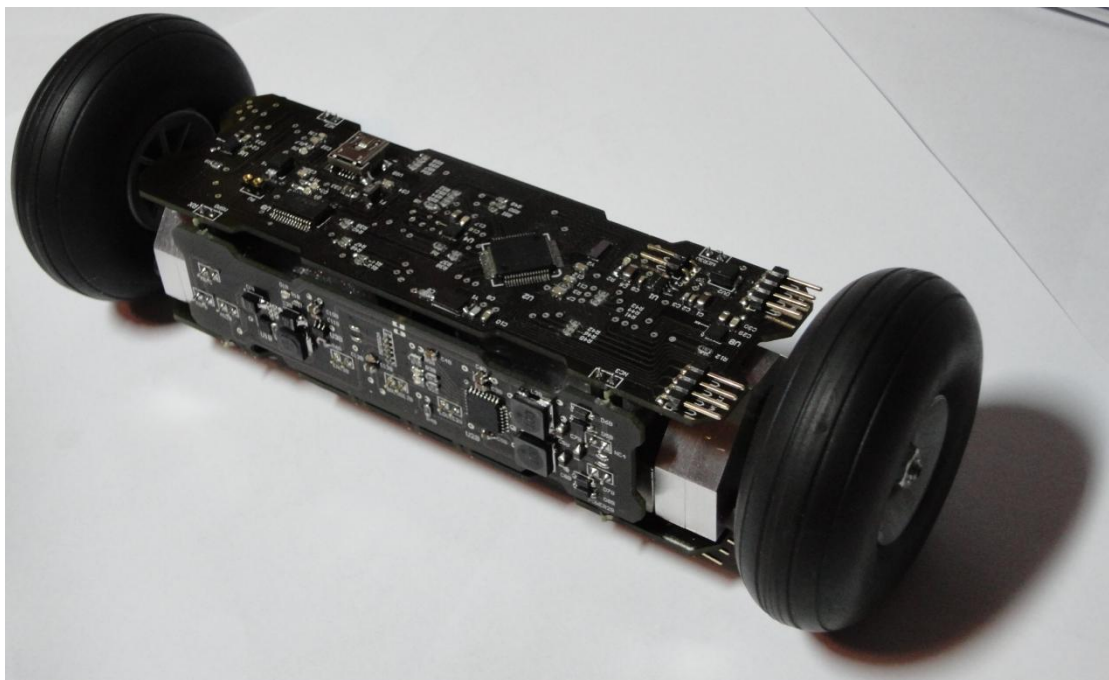
A mobilis robot működik, alkalmas összetett feladatok megoldására. A dolgozat keretein belül sikerült bebizonyítani a robot dinamikai viselkedését leíró matematikai modell helyességét, így ez lehetővé teszi a szabályozási algoritmusok elméleti tervezését és gyakorlati tesztelését (*implementálását*).

A jövőbeli fejlesztések:

A giroszkóp és gyorsulásmérő szenzorok kalibrációja. A kalibráció azért szükséges, mert a gyári beállítások mellett a mérési eredmények hibákat tartalmaznak (*tengelyek merőlegességi hibái, erősítéshiba, hőmérsékletfüggés, stb.*). A kalibráció a valós időben mért jeleket egy kompenzációs folyamat segítségével alkalmasabbá teszi a magasabb szintű szabályozások számára.

A robot teljes kinematikai- és dinamikai viselkedésének modellezése. A második fejezetben ismeretett matematikai modell általánosítása.

Mivel dinamikai jellemzők mintavételezhetők, a pillanatnyi rotoráram, forgási irány és sebesség, valamint a kilengés ismeretében egy négyhurkos kaszkádszabályozás implementálható a rendszerbe. A négyhurkos kaszkádszabályozás belső áramszabályozásból, középső sebesség- és pozíció szabályozásból, valamint külső kilengés szabályozásból áll.



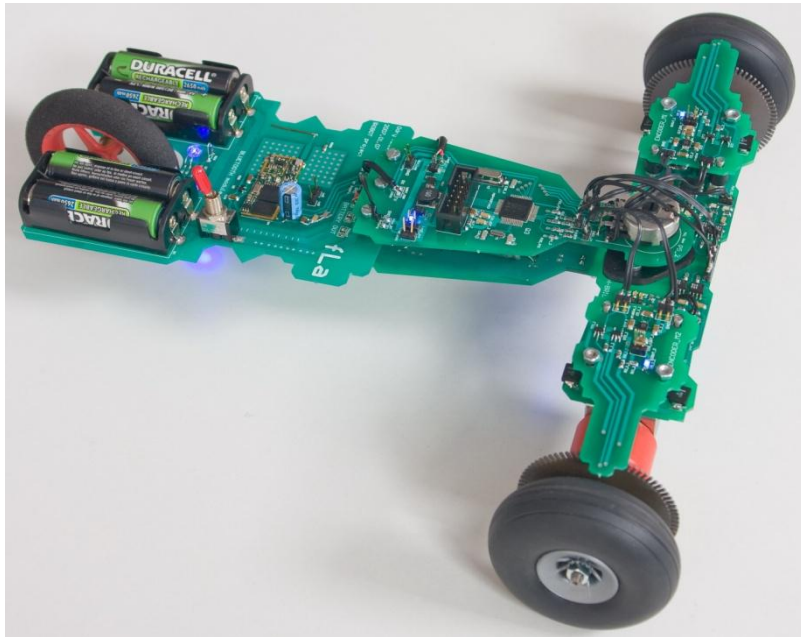
6. Irodalomjegyzék

- [1]: Dr. Tevesz Gábor, MIKROKONTROLLER ALAPÚ RENDSZEREK, ELEKTRONIKUS JEGYZET (2011)
- [2]: Texas Instruments, TPS62026, 600mA/1.25 MHz HIGH-EFFICIENCY STEP-DOWN CONVERTER Datasheet (2005)
- [3]: Texas Instruments, REG113, DMOS 400mA LOW-DROPOUT REGULATOR Datasheet (2005)
- [4]: Reston Condit, Microchip Technology Inc., AN905, BRUSHED DC MOTOR FUNDAMENTALS (2004)
- [5]: Dr. Halász Sándor, VILLAMOS HAJTÁSOK, ROTEL KFT. (2003). pp. 15 – 18.
- [6]: Texas Instruments, DRV592, ± 3 -A HIGH-EFFICIENCY H-BRIDGE Datasheet (2002)
- [7]: Adeunis RF, ARF32 DATA, CLASS2 BLUETOOTH MODULE FOR DATA TRANSMISSION Datasheet (2009)
- [8]: Jim Lepkowski, Microchip Technology Inc., AN894, MOTOR CONTROL SENSOR FEEDBACK CIRCUITS (2003)
- [9]: Dr. Péceli Gábor, Dr. Dabóczi Tamás, BME-VIK, HALLGATÓI SEGÉDLET A LABORATÓRIUM 2. C. TÁRGY MÉRÉSEIHEZ (2008)
- [10]: Texas Instruments, INA198, CURRENT SHUNT MONITOR, -16V TO +80V COMMON-MODE RANGE Datasheet (2010)
- [11]: Greg Welch, Gary Bishop, Department of Computer Science University of North Carolina at Chapel Hill, AN INTRODUCTION TO THE KALMAN FILTER (2006)
- [12]: Dr. Lantos Béla, IRÁNYÍTÁSI RENDSZEREK ELMÉLETE ÉS TERVEZÉSE II., KORSZERŰ SZABÁLYOZÁSI RENDSZEREK, AKADÉMIAI KIADÓ (2001)
- [13]: Dan Simon, Embedded Systems Programming, KALMAN FILTERING (2001)
- [14]: Oliver J. Woodman, UNIVERSITY OF CAMBRIDGE Computer Laboratory, AN INTRODUCTION TO INERTIAL NAVIGATION (2007)
- [15]: Kis László, Dr. Prohászka Zoltán, BME-VIK IIT Mesterséges Intelligencia labor, NAVIGÁCIÓ CÉLÚ JELFELDOLGOZÁS, MÉRÉSI ÚTMUTATÓ (2010)
- [16]: STMicroelectronics, LIS331DLH, MEMS DIGITAL OUTPUT MOTION SENSOR: ULTRA LOW-POWER HIGH PERFORMANCE 3-AXIS “NANO” ACCELEROMETER Datasheet (2009)

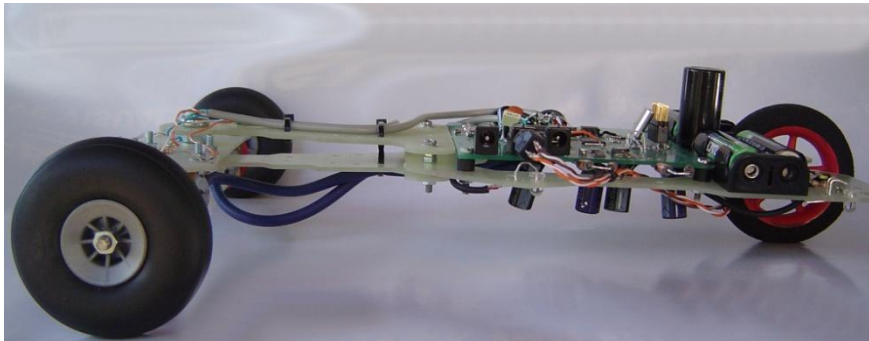
- [17]: STMicroelectronics, L3G4200D, MEMS MOTION SENSOR: ULTRA-STABLE THREE-AXIS DIGITAL OUTPUT GYROSCOPE Datasheet (2010)
- [18]: Future Technology Devices International Ltd., FT232R USB UART IC Datasheet (2010)
- [19]: Texas Instruments, MSP430F261x, MIXED SIGNAL MICROCONTROLLER Datasheet (2011)
- [20]: Texas Instruments, MSP430x2xx Family, USER'S GUIDE Datasheet (2011)
- [21]: Kimberly Tuck, Freescale Semiconductor, AN3461, TILT SENSING USING LINEAR ACCELEROMETERS (2007)
- [22]: STMicroelectronics, AN3182, TILT MEASUREMENT USING A LOW-G 3-AXIS ACCELEROMETER (2010)
- [23]: Kionix, AN005, TILT-SENSING WITH KIONIX MEMS ACCELEROMETERS (2007)
- [24]: Markus Haid, Jan Breitenbach, Applied Mathematics and Computation, LOW COST INERTIAL ORIENTATION TRACKING WITH KALMAN FILTER
- [25]: SIMOES, Eduardo do Valle, ICMC, TWO-WHEELED PENDULAR MOBILE ROBOTIC DEVICE (2008)
- [26]: Dr. Lantos Béla, IRÁNYÍTÁSI RENDSZEREK ELMÉLETE ÉS TERVEZÉSE I., EGYVÁLTOZÓS SZABÁLYOZÁSOK, AKADÉMIAI KIADÓ (2003). pp. 49 – 52., pp. 64-67.
- [27]: M. Sasaki, N. Yanagihara, O. Matsumoto, K. Komoriya, AIST, STEERING CONTROL OF THE PERSONAL RIDING-TYPE WHEELED MOBILE PLATFORM (PMP)
- [28]: Alessio Salerno, Jorge Angeles, A NEW FAMILY OF TWO-WHEELED MOBILE ROBOTS: MODELING AND CONTROLLABILITY (2007 IEEE)
- [29]: Alessio Salerno, Jorge Angeles, DESIGN AND IMPLEMENTATION OF A QUASIHOLONOMIC MOBILE ROBOT (2007 IEEE)

7. A dolgozat előzményei képekben

2007/2008



2006/2007



2005/2006

