NEUMANN JÁNOS
INFORMATIKAI KAR

SZAKDOLGOZAT

OE-NIK Hallgató neve: **Győri Ferenc Norbert**

2023 Hallgató törzskönyvi száma: **T/005641/FI12904/N**

NEUMANN JÁNOS
INFORMATIKAI KAR

SZAKDOLGOZAT
FELADATLAP

Hallgató neve: **Győri Ferenc Norbert**
Törzskönyvi száma: T/005641/FI12904/N
Neptun kódja: MSV4T4

A szakdolgozat címe:

Furuta inga tervezése és implementálása szenzorfúziós kísérleti alkalmazáshoz

Design and implementation of a Furuta pendulum for sensor fusion research applications

Intézményi konzulens: Dr. Kuti József

Beadási határidő: 2021. május 15.

A feladat

Szabályozási és szenzorfúziós módszerek kísérleti vizsgálata céljából tervezzen meg és készítse el egy ún. Furuta inga teljes mechatronikai megvalósítását. A tervezés során számításokkal ellenőrizze, hogy a használni kívánt részegységekkel megvalósíthatóak-e az irányítási célkitűzések. A mechanikai rendszer mozgásállapotát, a felhasználási célja okán, több különböző szenzorkombinációval is lehessen mérni. Tervezze meg és valósítsa meg ezen szenzorok kommunikációs sémáját a szabályozást végző egységgel. Dolgozza ki a rendszer nyomatékvezérelt pozíciószabályozását az egyensúlyi állapotra. Vizsgálja meg, hogy különböző szenzor kombinációk mért értékeiből milyen mértékben állíthatóak elő a Furuta inga szabályozásához szükséges jelek és vonja le következtetéseit.

A szakdolgozatnak tartalmaznia kell:

- Irodalomkutatás
- A Furuta inga mechanikai modellének bemutatása a szakirodalomnak megfelelően
- A mechanikai modell alapján szimulációk készítése a rendszer megvalósításához szükséges építőelemek kiválasztásához
- Szenzorfúzió vizsgálatához alkalmazandó szenzorok kiválasztása
- Kommunikációs megoldás kiválasztása
- Megvalósítás részletes leírása
- Tesztesetek megalkotása, a mérések megvalósítása
- Eredmények kiértékelése

Ph.

.....
Dr. Eigner György
intézetigazgató

A szakdolgozat elévülésének határideje: **2022. december 15.**
(OE TVSz 55.§ szerint)

A szakdolgozat beadásra alkalmasnak tartom:

.....
intézményi konzulens

NEUMANN JÁNOS
INFORMATIKAI KAR

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat/diplomunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladat kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 20. . .....

.....

hallagtó aláírása

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani konzulenseimnek Kuti Józsefnek és Galalambos Péternek a témáért és a megvalósítás során felmerülő problémák megoldásában való segítségnyújtásért, továbbá köszönet a rendszer megvalósításához szükséges eszközök rendselkezésre bocsájtásáért a teljes Bejczy Antal iRobottechnikai Központnak.

Külön köszönet Garamvölgyi Tivadarnak és Ládi Alexandernek, hogy segítséget nyújtottak a rendszer fizikai megtervezésében és megépítésében.

Hálával tartozom továbbá szüleimnek, testvéremnek és barátnőmnek Mányi Helénának, akik nélkül ez a szakdolgozat nem jöhetett volna létre. Köszönöm nekik, hogy tanulmányaim során türelemmel és megértéssel támogattak, és minden helyzetben mellettem álltak.

A szakdolgozat, a Nemzeti Kutatási Fejlesztési és Innovációs Alapból biztosított támogatással, a 2019-1.3.1-KK pályázati program finanszírozásában valósult meg a 2019-1.3.1-KK-2019-00007 számú projekt keretében.

Köszönet mindenkinek!

Tartalomjegyzék

Absztrakt

Szakdolgozatom során egy Furuta inga megtervezését és megvalósítását végeztem el olyan módon, hogy a feladatkiírásnak megfelelően alkalmas legyen szabályzási törvények megvalósítására és szenzorfúziós kísérletek elvégzésére. A rendszeren lineáris szabályozását valósítottam meg, amely képes az ingát fordított pozícióban tartani.

A tervezés magában foglalta a mechanikai rendszer megismerését. A modellt úgy választottam, hogy a szabályozási célnak megfelelő legyen. Kiválasztottam a megvalósításhoz szükséges beavatkozót, szenzorokat és mikrovezérlőket. Az Autocad Inventorban elkészítettem a rendszer 3D modelljét és ez alapján megépítettem a tervezett rendszert. Implementáltam a rendszer mérését végző mikrokontrollerek programját. Az eszközök között vezetékes és vezeték nélküli kommunikációt is megvalósítottam.

Matlab környezet segítségével identifikáltam a rendszert és megterveztem a szabályzót a matematikai modellje alapján. Implementáltam a szabályzást végző programot C++ nyelven majd ennek segítségével teszteltem a szabályozást a valós rendszeren.

Abstract

During my dissertation, I designed and implemented a Furuta pendulum in such a way that it would be suitable for the implementation of regulatory laws and sensor fusion experiments in accordance with the terms of reference. I implemented linear control on the system, which is able to keep the pendulum in the inverted position.

The design involved getting to know the mechanical system. I chose the model to fit the regulatory purpose. I selected the actuator, sensors and microcontrollers needed for the implementation. In Autocad Inventor I created a 3D model of the system and based on this I built the designed system. I implemented a program of microcontrollers measuring the system. I also implemented wired and wireless communication between the devices.

Using a Matlab environment, I identified the system and designed the controller based on its mathematical model. I implemented the control program in C ++ and then used it to test the control on the real system.

1. Bevezetés

Korunkban az ipari folyamatokban egyre jobban elterjed az automatizáció támogatása az informatika adta lehetőségekkel. A termelési folyamatokat a humán erőforrástól egyre inkább az autonóm működő gépi rendszerek veszik át.

Az ilyen jellegű törekvések rengeteg megoldandó problémát vetnek fel. Például meg kell oldani a gépek közötti kommunikációt, az emberi beavatkozást nem igénylő döntéshozatalt, továbbá azt, hogy képesek legyenek érzékelni az őket körülvevő környezetet, valamint a saját állapotukat. Ahhoz, hogy a gépek információt szerezzenek a környezetükről, illetve saját magukról ahhoz szenzorokkal kell felszerelnünk ezen berendezéseket. A szenzorok különböző elvek szerint képesek a fizikai világ mennyiségeinek mérésére és a mért adatokat számítógépek által feldolgozható analóg vagy digitális jelekké alakitani.

A szenzorok általi méréseket úgynevezett mérési zaj terheli, megjelenik rajtuk a szenzor dinamikája és állandó hiba is. Ezen felül fontos tényező lehet a mintavételi frekvencia valamit a kommunikációs és jelfeldolgozási időkésés, így nem mindig használhatók közvetlenül a mért értékek arra, hogy általuk a rendszer elvárt és megfelelő módon szabályozható legyen. Ilyen esetekben szükséges a szenzorok által adott jelek szoftveres szűrése, mely segítségével a valós érékekhez közelebb álló mért értékeket kaphatunk.

Bonyolultabb rendszerek esetén a rendszer állapotának méréséhez több különböző szenzort használhatunk fel, és a szenzorok által mért különböző adatok szoftveres összegzése alapján becsülhető meg a rendszer állapota. Ezt az eljárást szenzorfúziónak[1] nevezzük.

Jelenleg a Bejczy Antal iRobottechnikai Központban egy átalános célú szenzorfúzió megvalósítására képes szoftver könyvtár elkészítése zajlik. A szenzor könyvtár működésének tesztelése céljából szükség van egy mechanikai rendszerre mely úgy van megtervezve, hogy annak állapotát többféle szenzor kombinációjával lehessen mérni, és különböző szabályozási feladatok során tesztelni.

Mechanikai tesztrendszernek a Furuta inga került kiválasztásra, mivel mozgása többféle módon mérhető, valamit sokféle szabályozási megoldást lehet rajta kipróbálni. Szakdolgozatom célja a tesztelésre szánt Furuta ingát megtervezzem és megépítsem.Ezen felül az inga fordiott pozícióban való megtartása lineáris szabályozás hatására.

2. Furuta inga matematikai modellezése

A Furuta ingát[2] vagy forgó fordított ingát először a Tokiói Műszaki Intézetben alkották meg Katsuhisa Furuta és munkatársai 1991-ben.

A inga két henger alakú kartagból álló mechanikai rendszer. Az első kartag vízszintes síkban mozoghat. Ehhez a kartaghoz közvetlenül csatlakozik egy DC motor, így tehát mozgását közvetlenül tudjuk befolyásolni. A második kartag az első kartag végéhez csatlakozik és az arra merőleges síkban képes szabadon forgó mozgást végezni, ezen mozgását csak közvetve az első kartag segítségével tudjuk befolyásolni. Az ingának két különböző típusú egyensúlyi állapota van, egyik amikor a második kartag lefele lévő pozícióban van és a másik amikor fölfelé. A második egyensúlyi állapot instabil mivel, ha az ingát ebben az állapotban hatás éri akkor kitér és elkezd az első egyensúlyi állapotba mozogni.

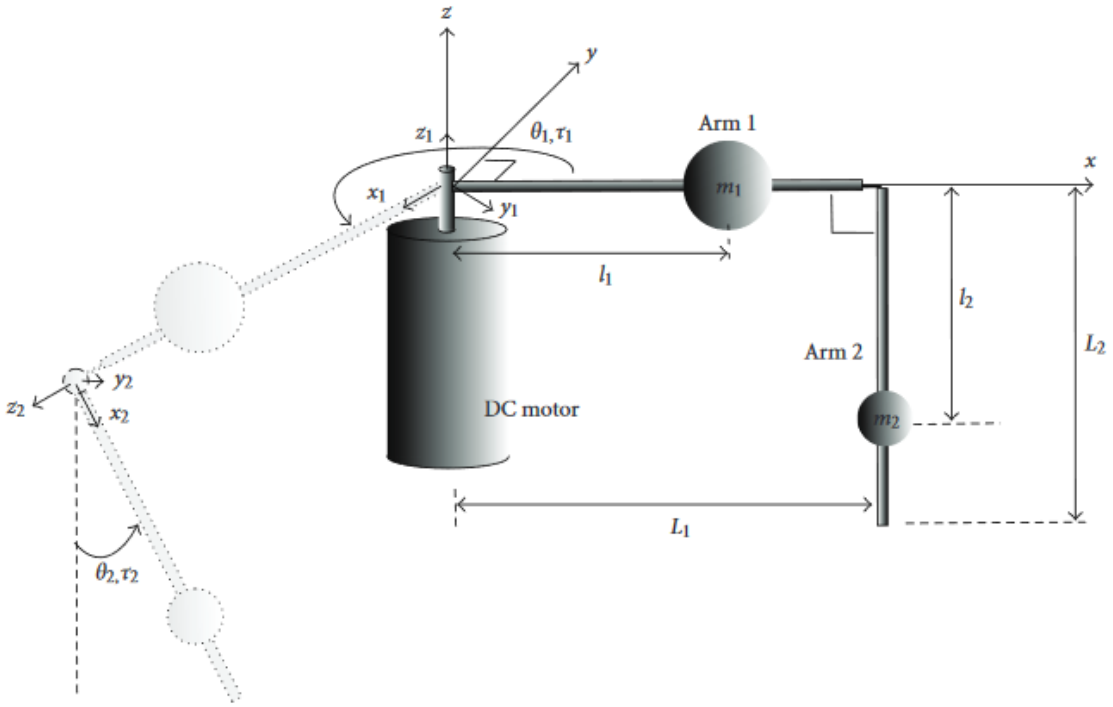
Az ingán két szabályozási célt lehet megfogalmazni. Egyik ilyen cél a fellendítés, amikor az ingát az alsó egyensúlyi állapotából a felső egyensúlyi állapot közelébe szeretnénk juttatni, a másik cél pedig felfele álló pozícióban lévő stabilizálás. Munkám során az ingán ez utóbbi célt valósítottam meg.

A teljes struktúra tehát két alrendszerből tevődik össze:

- A mechanikai alrendszer: a fentebb ismertetett két kartagból álló rendszer
- Az elektronikai alrendszer: DC motor, amely a rendszer mozgatásához szükséges nyomaték kifejtésére szolgál

2.1. Mechanikai alrendszer mozgásegyenlete

Ahhoz, hogy a struktúra mozgását leírhassuk első lépésként definiálnunk kell egy bázis koordinátarendszert. Ezt az első kartag és a motor forgástengelyének csatlakozási pontjában vesszük fel oly módon, hogy a motor forgástengelye és a koordinátarendszer z tengelye egy egyenesbe essenek. A könnyebb levezetés érdekében felveszünk további két koordinátarendszert, amelyek az egyes kartagok végéhez csatlakoznak és együtt mozognak velük. A koordináta rendszereket a 3-1. ábrán látható módon vesszük fel. Mindhárom koordinátarendszer jobbsodrású és derékszögű.



2-1. ábra. Furuta inga sematikus ábrája [3]

A rendszert a következő fizikai mennyiségekkel tudjuk leírni. A kartagok hosszát L_1 és L_2-vel, valamint a kartagok forgástengelyének és tömegközéppontjának távolságát l_1 és l_2 -vel jelöljük. Az egyes kartagok tömegét m_1 és m_2-vel jelöljük, a tehetetlenségi nyomatéki tenzorokat pedig J_1 és J_2 vel. A koordinátarendszereket úgy vettük fel, hogy azok tengelyei a forgás tehetetlenségi főtengelyei legyenek. Ezért a tehetetlenségi nyomatéki tenzorok diagonális mátrixok lesznek.

$$J_1 = \begin{bmatrix} J_{1xx} & 0 & 0 \\ 0 & J_{1yy} & 0 \\ 0 & 0 & J_{1zz} \end{bmatrix}, \quad J_2 = \begin{bmatrix} J_{2xx} & 0 & 0 \\ 0 & J_{2yy} & 0 \\ 0 & 0 & J_{2zz} \end{bmatrix}.$$

Az első kartag elfordulását θ_1-gyel jelöljük, pozitív iránya - a bázis koordinátarendszert a Z tengely felől nézve - az óramutatóval ellentétes irány. A második kartag elfordulását θ_2-vel jelöljük. Ezt az elfordulást a második kartag alsó egyensúlyi pozíciójához képest mérjük. Pozitívnak az óramutatójárásával ellentétes irányt vesszük, az első mozgó koordinátarendszer x tengelye felől. A koordináta rendszerek úgy lettek felvéve hogy θ_1 és θ_2 a rendszer általános koordinátái legyenek. A DC motor által az első kartagra kifejtett forgatónyomatékot τ_1, a második kartagra ható forgató nyomatékot τ_2-vel jelöljük. A

forgások kapcsolási pontjainál viszkózus csillapítás lép föl. Az első kartag motor általi csillapítását b_1 a második kartag csapágy általi csillapítását b_2 -vel jelöljük.

A modell felírásakor a következő feltételezésekkel élünk:

- A motor és az első kartag ideálisan kapcsolódnak egymáshoz
- Az első és második kartag merev test
- A mozgó koordinátarendszerek oly módon vannak felvéve, hogy a tehetetlenségi tenzorok diagonális mátrixok legyenek.
- A motor forgó részének tehetetlenségét elhanyagoljuk.
- Csak a viszkózus csillapitást vesszük figyelembe

2.2. Mozgásegyenlet levezetése másodfajú Lagrange-egyenlet alapján

A másodfajú Lagrange-egyenlet [4] alkalmazásával konzervatív holonom mechanikai rendszerek mozgásegyenleteit írhatjuk fel. Ehhez első lépésben a rendszer geometriája alapján fel kell írni a potenciális és kinetikus energiákat az álltalános koordináták felhasználásával. Az energiák különbsége ként irható fel a Lagrange-függvény.

$$L = T - V$$

Ha ezt a függvényt behelyetesítjük az Euler-Lagrange egyenletbe és elvégezzük a kijelölt deriválásokat akkor megkapjuk a redszer mozgásegyenletét. A következőkben ismertetem a rendszer Lagrange-függvényének levezetését.

2.2.1. Koordinátarendszerek közötti traszformációk

Az egyes kartagok tömegközéppontjának sebességét, pozícióját a kartaghoz rendelt koordinátarendszerben könnyű megadni, viszont szükséges, hogy az egyes koordinátarendszerekben leírt mennyiségeket át tudjunk vinni más koordinátarendszerekbe. Ehhez definiálunk két forgatási mátrixot [5].

Az egyes koordinátarendszerből a bázisba z tengely körüli θ_1-gyel való elforgatással juthatunk el.

$$\boldsymbol{R}_{1,0} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

A második koordináta rendszerből az elsőbe egy z tengely körüli θ_2-vel való elforgatással és egy y körüli 90 fokkal való elforgatással juthatunk.

$$\boldsymbol{R}_{2,1} = \begin{bmatrix} \cos(\theta_2) & \sin(\theta_2) & 0 \\ -\sin(\theta) & \cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \sin(\theta_2) & -\cos(\theta_2) \\ 0 & \cos(\theta_2) & \sin(\theta_2) \\ 1 & 0 & 0 \end{bmatrix}$$

2.2.2. Tömegközéppontok sebességei

A kinetikus energiák kifejezéséhez fel kell írnunk az egyes kartagok tömegközéppontjának sebességeit.

Az első kartag elfordulást csak a Z tengely körül végez θ_1-gyel így a tengelyek körüli elfordulás időszerinti első deriváltja, azaz a kartag szögsebessége a következő:

$$\boldsymbol{\omega}_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix}$$

Egy test bármely pontjának sebessége megadható, ha ismert a test egy pontjának sebessége és a test szögsebesége. Az első kartag azon pontja, amelyen átmegy a forgástengely nyugalomban van, tehát sebessége nulla. Ezek segítségével meghatározható az első kartag tömegközéppontjának sebessége:

$$\boldsymbol{v}_{1c} = 0 + \boldsymbol{\omega}_1 \times \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{\theta}_1 l_1 \\ 0 \end{bmatrix}$$

A második kartag forgó mozgása két komponensből áll, a második kartag elsőhöz képesti elfordulásából és az első kartag forgó mozgásából. Hogy felírhassuk a második kartag szögsebességét ahhoz ezt a két szögsebesség komponenst közös koordinátarendszerbe kell leírnunk, ehhez felhasználjuk az $R_{2,1}$ forgatási mátrixot. Ezek alapján a második kar szögsebessége:

$$\boldsymbol{\omega}_2 = \boldsymbol{R}_{2,1}\boldsymbol{\omega}_1 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} -\cos(\theta_2)\dot{\theta}_1 \\ \sin(\theta_2)\dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

Ahhoz, hogy a második kartag tömegközéppontjának sebességét meghatározhassuk fel kell írnunk az első és a második kartag kapcsolódási pontjának sebességét. Ezt az első

tömegközéppont sebeségének kiszámításához hasonlóan megtehetjük az 1-es koordinátarendszerben, amit utána R_2 felhasználásával átvihetünk a kettes koordinátarendszerbe.

$$\boldsymbol{v}_2 = \boldsymbol{R}_{2,1} \left(\boldsymbol{\omega}_1 \times \begin{bmatrix} 0 \\ 0 \\ L_1 \end{bmatrix} \right) = \begin{bmatrix} \dot{\theta}_1 L_1 \sin(\theta_2) \\ \dot{\theta}_1 L_1 \cos(\theta_2) \\ 0 \end{bmatrix}$$

A v_2-t és ω_2-t felhasználva már kifejezhető a keresett sebesség.

$$\boldsymbol{v}_{2c} = \boldsymbol{v}_2 + \boldsymbol{\omega}_2 \times \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1 L_1 \sin(\theta_2) \\ \dot{\theta}_1 L_1 \cos(\theta_2) + \dot{\theta}_2 l_2 \\ -\dot{\theta}_1 l_2 \sin(\theta_2) \end{bmatrix}$$

2.2.3. A rendszer energiái

A Lagrange függvény felírásához ki kell fejezzük az egyes kartagok kinetikus és potenciális energiáját.

Az első kartag tömeg középpontjának potenciális energiája 0, hiszen a kartag a bázis xy síkjában mozog.

$$V_1 = 0$$

Mozgási energiáját pedig a fent kifejezett sebesség és szögsebesség segítségével tudjuk felírni.

$$T_1 = \frac{1}{2} \left(\boldsymbol{v}_{1c}^T m_1 \boldsymbol{v}_{1c} + \boldsymbol{\omega}_1^T \boldsymbol{J}_1 \boldsymbol{\omega}_1 \right) = \frac{1}{2} \dot{\theta}_1^2 (m_1 l_1^2 + J_{1zz}).$$

A második kartag tömegközéppontjának potenciális energiája a rendszer geometriája alapján a következő szerint számítható:

$$V_2 = g m_2 l_2 (1 - \cos(\theta_2)).$$

A mozgási energia pedig a már kifejezett sebességeket felhasználva a következőként irható fel:

$$
\begin{aligned}
T_2 &= \frac{1}{2}\left(\boldsymbol{v}_{2c}^T m_2 \boldsymbol{v}_{2c} + \boldsymbol{\omega}_2^T \boldsymbol{J}_2 \boldsymbol{\omega}_2\right) \\
&= \frac{1}{2}\dot{\theta}_1^2\left(m_2 L_2^2 + (m_2 l_2^2 + J_{2yy})\sin^2(\theta_2) + J_{2xx}\cos^2(\theta_2)\right) \\
&+ \frac{1}{2}\dot{\theta}_2^2(J_{2zz} + m_2 l_2^2) + m_2 L_1 l_2 \cos(\theta_2)\dot{\theta}_1\dot{\theta}_2.
\end{aligned}
$$

Ezek felhasználásával már kifejezhető a rendszer összes kinetikai és potenciális energiája:

$$
\begin{aligned}
T &= T_1 + T_2, \\
V &= V_1 + V_2.
\end{aligned}
$$

2.2.4. Mozgásegyenlet meghatározása Euler-Lagrange egyenlettel

Az energiák felhasználásával most már meg tudjuk határozni a rendszer Lagrange függvényét. Ha ezt a függvényt behelyettesítjük az Euler-Lagrange egyenletbe [6] és elvégezük a kijelölt deriválásokat akkor megkapjuk a rendszer mozgásegyenletét. Az Euler-Lagrange egyenletet a következő formában fogjuk használni:

$$
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i.
$$

ahol q_i az álltalános koordináták, azaz θ_1 és θ_2, Q_i pedig az álltalános erők, amelyek a kartagokra hatnak. Az álltalános erők a kartagokra ható forgatónyomatékok és a fellépő viszkózus csillapítások különbözete ként irható fel.

$$
q = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \quad Q = \begin{bmatrix} \tau_1 - b_1\dot{\theta}_1 \\ \tau_2 - b_2\dot{\theta}_2 \end{bmatrix}.
$$

Az első általános koordináta szerint elvégezve a parciális deriválásokat a következő eredményeket kapjuk:

$$
\begin{aligned}
-\frac{\partial L}{\partial \theta_1} &= 0 \\
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) &= \ddot{\theta}_1(J_{1zz} + m_1 l_1^2 + m_2 L_1^2 \\
&\quad + (m_2 l_2^2 + J_{2yy})\sin^2(\theta_2) + J_{2xx}\cos^2(\theta_2)) \\
&\quad + m_2 L_1 l_2 \cos(\theta_2)\ddot{\theta}_2 - m_2 L_1 l_2 \sin(\theta_2)\dot{\theta}_2^2 \\
&\quad + \dot{\theta}_1\dot{\theta}_2 \sin(2\theta_2)(m_2 l_2^2 + J_{2yy} - J_{2xx}).
\end{aligned}
$$

Elvégezve a parciális deriválásokat a második általános koordináta szerint a következő egyenletet eredményezi:

$$\begin{aligned}
-\frac{\partial L}{\partial \theta_2} &= -\frac{1}{2}\dot{\theta}_1^2 \sin(2\theta_2)(m_2 l_2^2 + J_{2yy} - J_{2xx}) \\
&\quad + \dot{\theta}_1 \dot{\theta}_2 m_2 L_1 l_2 \sin(\theta_2) + g m_2 l_2 \sin(\theta_2) \\
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) &= \ddot{\theta}_1 m_2 L_1 l_2 \cos(\theta_2) + \ddot{\theta}_2 (J_{2zz} + m_2 l_2^2) \\
&\quad - \dot{\theta}_1 \dot{\theta}_2 m_2 L_1 l_2 \sin(\theta_2).
\end{aligned}$$

A kiszámított eredményeket az Euler-Lagrange egyenlet szerint összeírhatjuk egy kifejezéssé és ezzel megkaptuk a Furuta inga mozgásegyenletét:

$$\begin{bmatrix}
\begin{pmatrix}
\ddot{\theta}_1 (J_{1zz} + m_1 l_1^2 + m_2 L_1^2 \\
+ (m_2 l_2^2 + J_{2yy}) \sin^2(\theta_2) + J_{2xx} \cos^2(\theta_2)) \\
+ m_2 L_1 l_2 \cos(\theta_2) \ddot{\theta}_2 - m_2 L_1 l_2 \sin(\theta_2) \dot{\theta}_2^2 \\
+ \dot{\theta}_1 \dot{\theta}_2 \sin(2\theta_2)(m_2 l_2^2 + J_{2yy} - J_{2xx}) + b_1 \dot{\theta}_1
\end{pmatrix} \\
\begin{pmatrix}
\ddot{\theta}_1 m_2 L_1 l_2 \cos(\theta_2) + \ddot{\theta}_2 (J_{2zz} + m_2 l_2^2) \\
+ \frac{1}{2}\dot{\theta}_1^2 \sin(2\theta_2)(-m_2 l_2^2 - J_{2yy} + J_{2xx}) \\
+ g m_2 l_2 \sin(\theta_2) + b_2 \dot{\theta}_2
\end{pmatrix}
\end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}.$$

2.3. Egyszerűsítések

Mivel az inga kartagjai hosszúak így a kartagok hosszanti tengelye szerinti tehetetlenségi nyomatékot elhanyagolhatjuk, ezen kívül a kartagok forgásszimmetrikusak így a másik két tengely mentén a tehetetlenségi nyomatékok egyenlőek. Ezek alapján a tehetetlenségi nyomatéki tenzorok a következő alakkal közelíthetőek:

$$J_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & J_1 & 0 \\ 0 & 0 & J_1 \end{bmatrix}, \quad J_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_2 \end{bmatrix}.$$

Ezen kívül vezessük be a következő jelöléseket.

$$\hat{J}_1 = J_1 + m_1 l_1^2, \quad \hat{J}_2 = J_2 + m_2 l_2^2, \quad \hat{J}_0 = J_1 + m_1 l_1^2 + m_2 L_1^2. \tag{2.1}$$

Ezeket felhasználva a mozgásegyenlet egyszerűbb formára hozható:

$$\begin{bmatrix} \begin{pmatrix} \ddot{\theta}_1(\hat{J}_0 + \hat{J}_2\sin^2(\theta_2)) + \ddot{\theta}_2 m_2 L_1 l_2 \cos(\theta_2) \\ -m_2 L_1 l_2 \sin(\theta_2)\dot{\theta}_2^2 + \dot{\theta}_1\dot{\theta}_2\hat{J}_2\sin(2\theta_2) + b_1\dot{\theta}_1 \end{pmatrix} \\ \begin{pmatrix} \ddot{\theta}_1 m_2 L_1 l_2 \cos(\theta_2) + \ddot{\theta}_2\hat{J}_2 - \frac{1}{2}\dot{\theta}_1^2\hat{J}_2\sin(2\theta_2) \\ +b_2\dot{\theta}_2 + g m_2 l_2 \sin(\theta_2) \end{pmatrix} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}. \tag{2.2}$$

Algebrai átalakításokkal kifejezhetjük a fenti egyenletrendszerből $\ddot{\theta}_1$-et és $\ddot{\theta}_2$-t

$$\ddot{\theta}_1 = \frac{\left(\begin{bmatrix} -\hat{J}_2 b_1 \\ m_2 L_1 l_2 \cos(\theta_2) b_2 \\ -\hat{J}_2^2 \sin(2\theta_2) \\ \frac{1}{2}\hat{J}_2 m_2 L_1 l_2 \cos(\theta_2)\sin 2\theta_2 \\ \hat{J}_2 m_2 L_1 l_2 \sin(\theta_2) \end{bmatrix}^T \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_1\dot{\theta}_2 \\ \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{bmatrix} + \begin{bmatrix} \hat{J}_2 \\ -m_2 L_1 l_2 \cos(\theta_2) \\ \frac{1}{2} m_2^2 l_2^2 L_1 \sin(2\theta_2) \end{bmatrix}^T \begin{bmatrix} \tau_1 \\ \tau_2 \\ g \end{bmatrix} \right)}{\left(\hat{J}_0\hat{J}_2 + \hat{J}_2^2\sin^2(\theta_2) - m_2^2 L_1^2 l_2^2 \cos^2(\theta_2) \right)}$$

$$\ddot{\theta}_2 = \frac{\left(\begin{bmatrix} m_2 L_1 l_2 \cos(\theta_2) b_1 \\ -b_2(\hat{J}_0 + \hat{J}_2\sin^2(\theta_2)) \\ m_2 L_1 l_2 \hat{J}_2 \cos\theta_2 \sin 2\theta_2 \\ -\frac{1}{2}\sin(2\theta_2)(\hat{J}_0\hat{J}_2 + \hat{J}_2^2\sin^2(\theta_2)) \\ -\frac{1}{2} m_2^2 L_1^2 l_2^2 \sin(2\theta_2) \end{bmatrix}^T \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_1\dot{\theta}_2 \\ \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{bmatrix} + \begin{bmatrix} -m_2 L_1 l_2 \cos(\theta_2) \\ \hat{J}_0 + \hat{J}_2\sin^2(\theta_2) \\ -m_2 l_2 \sin(\theta_2)(\hat{J}_0 + \hat{J}_2\sin^2(\theta_2)) \end{bmatrix}^T \begin{bmatrix} \tau_1 \\ \tau_2 \\ g \end{bmatrix} \right)}{\left(\hat{J}_0\hat{J}_2 + \hat{J}_2^2\sin^2(\theta_2) - m_2^2 L_1^2 l_2^2 \cos^2(\theta_2) \right)}.$$

2.4. A rendszer állapotteres leírása

Egy rendszer állapotteres leírását két lineáris [7] egyenlettel adjuk meg: az első egyenlet egy differenciálegyenlet, ami leírja a rendszer dinamikáját, a második egyenlet egy algebrai egyenlet, ami megadja a rendszer kimenetét.

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

$$y = Cx + Du$$

Ebben a felírásban u függvény a rendszer bemenete, y függvény a rendszer mért kimenete x vektor pedig a rendszer állapot változói. Az A rendszermátrix, ami leírja az állapotváltozók és azok deriváltjai közötti összefüggést, B pedig a bemeneti mátrix, ami bemeneteknek a rendszer dinamikájára gyakorolt hatását írja le. C mátrix a rendszer mért kimenetei és az állapotváltozók közötti összefüggést írja le, D pedig a bemenetek és kimenetek közötti közvetlen összefüggést. Valós rendszerek esetén D=0 mivel a bemenetek csak közvetve, a rendszer dinamikáján keresztül jelenhetnek meg annak kimenetén.

Az inga dinamikáját ilyen formában írhatjuk fel és a későbbiekben ez alapján tervezhető meg a szabályozás. Ehez azonban az inga mozgás egyenletét linearizálnunk kell. Ezt megtehetjük az inga egyensúlyi állapotai körűl. Egy rendszer akkor van nyugalomban mikor gyorsulása nulla. Megvizsgálva a fenti egyenleteket látható, hogy ez akkor teljesül, ha:

$$\tau_1 = 0, \quad \tau_2 = 0, \quad \theta_1 \in \mathbb{R}, \quad \theta_2 = k\pi \quad k \in \mathbb{Z}, \quad \dot{\theta}_1 = 0, \quad \dot{\theta}_2 = 0.$$

Ezzel matematikailag is belátható, hogy az ingának valóbban két különböző egyensúlyi állapota van. A mozgásegyenletet ezen pontok körüli Taylor sorba fejtésével megkapható a rendszer egy linearizált modellje. A linearizált modell csak a sorba fejtés pontja körül ad helyes eredményt, de mivel a szabályozás célja, hogy a rendszert az adott pontban tartsuk így a rendszer ettől a ponttól csak kis kitéréseket fog tenni tehát a modell alkalmazható.

Elvégezve a Taylor sorba fejtést az első nem csak konstansokat tartalmazó tagokig, majd behelyettesítve a felfele álló pozíció szerinti értékeket, ezután változók szerint szétválogatva azt, a következő együtthatókat kapjuk:

$$\theta_1 = 0, \quad \theta_2 = \pi, \quad \dot{\theta}_1 = 0, \quad \dot{\theta}_2 = 0, \quad \tau_1 = 0, \quad \tau_2 = 0$$

$$A_{31} = 0, \quad A_{41} = 0, \quad B_{31} = \frac{\hat{J}_2}{\hat{J}_0\hat{J}_2 - m_2^2L_1^2l_2^2},$$

$$A_{32} = \frac{gm_2^2l_2^2L_1}{\hat{J}_0\hat{J}_2 - m_2^2L_1^2l_2^2}, \quad A_{42} = \frac{gm_2l_2L_1}{\hat{J}_0\hat{J}_2 - m_2^2L_1^2l_2^2}, \quad B_{41} = \frac{m_2L_1l_2}{\hat{J}_0\hat{J}_2 - m_2^2L_1^2l_2^2},$$

$$A_{33} = \frac{-b_1\hat{J}_2}{\hat{J}_0\hat{J}_2 - m_2^2L_1^2l_2^2}, \quad A_{43} = \frac{-b_1m_2l_2L_1}{\hat{J}_0\hat{J}_2 - m_2^2L_1^2l_2^2}, \quad B_{32} = \frac{m_2L_1l_2}{\hat{J}_0\hat{J}_2 - m_2^2L_1^2l_2^2},$$

$$A_{34} = \frac{-b_2m_2l_2L_1}{\hat{J}_0\hat{J}_2 - m_2^2L_1^2l_2^2}, \quad A_{44} = \frac{-b_2\hat{J}_0}{\hat{J}_0\hat{J}_2 - m_2^2L_1^2l_2^2}, \quad B_{42} = \frac{\hat{J}_0}{\hat{J}_0\hat{J}_2 - m_2^2L_1^2l_2^2}.$$

Ezeket az értékeket mátrix alakba rendezve megkapjuk a rendszer állapotteres leírását:

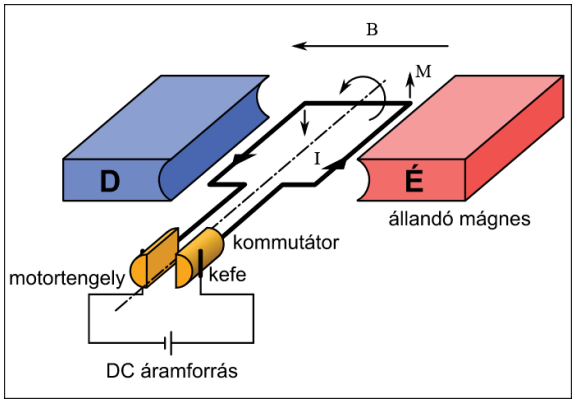
$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ B_{31} & B_{32} \\ B_{41} & B_{42} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

Ugyanezt elvégezve a lefele álló pozíció értékeivel ($\theta_1 = 0, \quad \theta_2 = 0, \quad \dot{\theta}_1 = 0, \quad \dot{\theta}_2 = 0, \quad \tau_1 = 0, \quad \tau_2 = 0$) azt tapasztalhatjuk, hogy hasonló eredmények jönnek ki, azonban a mátrixok egyes helyein a kifejezések mínusz egyszeresei szerepelnek. Ez igaz a A_{34}, A_{42}, A_{43}, B_{32} és B_{41} tagokra.

2.5. Az elektronikai alrendszer dinamikája

A Furuta inga mozgatását egy DC motor fogja végezni ezért a motor viselkedését is bele kell foglalnunk a rendszert leíró összefüggésbe. Napjainkban különböző szabályzási feladatok ellátására terjedtek el a szervomotorok. A szabályozandó rendszerhez viszonyítva dinamikájuk gyorsabb, valamint könnyen vezérelhetők.

A DC motorok az elektromos és mechanikai felépítés szempontjából az alábbi fontosabb részegységekből állnak: állórész állandó mágnes (vagy állórész tekercselés), állórész test, állórész pólusok, tekercselt forgórész, forgórész tengely, forgórész csapágyazás, kommutátor, kefék és kefetartók.



2-2. ábra. DC motor müködési elve[8]

Az ábrán látható módon a motor rotorjául szolgáló elektromágnes egy permanens mágnes biztosította mágneses térben forog. A forgó mozgást az azonos mágneses pólusok közti taszító, illetve az ellentétes mágneses pólusok közti vonzó erők forgatónyomatékai eredményezik. Amennyiben a forgórész tekercsében folyó áram iránya változatlan, úgy maximum 180°-os elfordulás lehetséges, majd a rotor megáll, mivel az ellentétes mágneses pólusok szembenállásával a forgatónyomaték megszűnik. Amennyiben a holtpont elérésekor a forgórész elektromágnesében megváltoztatjuk az áram irányát, s ezzel a mágneses pólusok polaritását, úgy a rotor forgása folyamatossá tehető. Az áram irányának periodikus megváltoztatásáról a forgórész forgástengelyére szerelt kommutátor gondoskodik. A kommutátor két egymástól elszigetelt vezető félgyűrű, amelyekhez a tekercs két végét csatlakoztatják. Az áram bevezetése kefék segítségével történik. A kefék csak elhanyagolható mértékű súrlódási veszteséggel terhelik a folyamatos forgómozgás létrejöttét. Az

állórész (armatúra) és a forgórész (rotor) mágneses póluspárjainak növelésével a forgás egyenletesebbé tehető. Több póluspár esetén természetesen a kommutátor szegmenseinek számát is növelni kell.

A DC motor egy sorba kötött ideális tekercccsel és egy ellenállással modellezhető [9]. Így a Kirchhoff törvényt, valamint Newton második törvényét felhasználva, a dinamikája a következő egyenlőségekkel írható le:

$$J\ddot{\theta} + b\dot{\theta} = K_m i,$$
$$L\dot{i} + Ri + K_e\dot{\theta} = V.$$

A kifejezésben i a motoron átfolyó áramot, θ a motor szöghelyzetét, V a motorra kapcsolt feszültséget jelenti. A további paraméterek a motor állandói, L a tekercs induktivitás, R ellenállás, K_m a motor nyomatékállandója,K_e a visszaható elektromotoros erő, b a csillapítás és J a motor tehetetlenségi nyomatéka.

A motor által kifejtett nyomaték a Biot-Savart törvény értelmében egyenesen arányos a motoron átfolyó árammal. Ez a nyomaték az első kartagot érő hatás. Ezek a következő képpen foglalhatók össze egyenletként:

$$\tau = K_m i.$$

A fenti kifejezések alapján a motor modelljét az alábbi alakban tudjuk felírni, ahol a rendszer kimenete a kifejtett nyomaték:

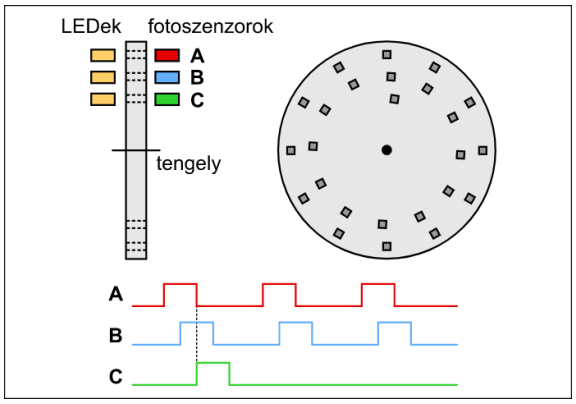
$$\frac{d}{dt}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} = \begin{bmatrix}-\frac{b}{J} & \frac{K_t}{J}\\ -\frac{K_e}{L} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{L}\end{bmatrix}V,$$
$$\tau = \begin{bmatrix}0 & K_m\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix}.$$

3. Szenzorok és beavatkozók

A célkitűzéshez igazodva a rendszer állapotát többféle módon tervezem mérni. Ebben a fejezetben áttekintem, hogy milyen eszközök lehetnek alkalmasak a Furuta inga mozgásának mérésére [10].

3.1. Optikai enkóder

Az optikai elven működő forgójeladóknál egy átlátszó műanyagból vagy üvegből készült tárcsára egyenletes eloszlásban világos (átlátszó) és sötét (átlátszatlan) osztásokat visznek fel. Az így kapott sávot a tárcsa egyik oldaláról jól fókuszált, keskeny, a tárcsára merőleges fénynyalábbal világítják át. Fényforrásként általában az infravörös hullámhossztartományban üzemelő LED-et vagy lézerdiódát alkalmaznak. A fénynyalábot a tárcsa másik oldalán egy optoelektronikus érzékelő, általában egy fototranzisztor fogadja. (A fényforrás (adó) és a detektor (vevő) együttesét optokapunak is nevezzük.) Optikai enkóderből létezik inkrementális és abszolút működésű is.



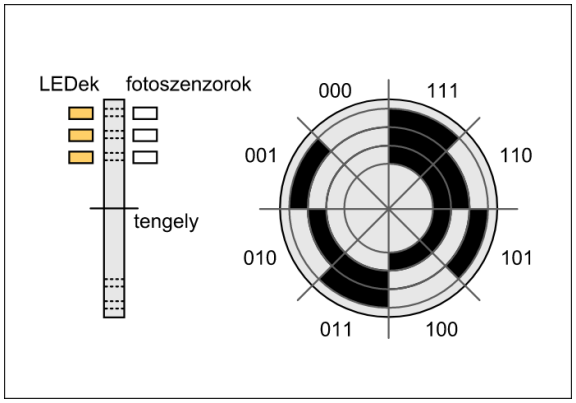
3-3. ábra. Inkrementális optikai enkóder működése[**?**]

Az inkrementális enkóder esetén a tárcsa elfordulásának hatására az osztások a fényutat megszaggatják, s azt az optoelektronikai érzékelő feszültségimpulzusokká (leggyakrabban TTL jelsorozattá) alakítja, amit elektronikus számláló áramkörökkel megszámlálhatunk. Ha a tárcsán egy teljes sávban N_0 sötét osztást alakítottak ki, és a számlálás során N_α impulzust számláltunk, akkor a tárcsa elfordulásának szöge (fokokban kifejezve):

$$\alpha = 360^\circ \frac{N_\alpha}{N_0}$$

A feszültségimpulzusok számlálása alapján a forgásirányt nem lehet megállapítani. A forgásirány maghatározásához a tárcsára egy második sávot is felvisznek, amelynek osztásai

90°-os fáziskülönbséggel helyezkednek el az első sáv osztásaihoz képest. Ezt a második sávot egy újabb LED-del világítják át, illetve egy újabb fototranzisztorral detektálják a fényintenzitást a tárcsa másik oldalán. A tárcsa abszolút pozíciójának detektálásához egy harmadik C csatornát is felvisznek, ami azonban csak egy osztást tartalmaz. A C csatorna jelét index-változónak is hívják. Manapság az osztások száma $100 \leqslant N_0 \leqslant 10.000$ között változhat, ami $3,6° \leqslant \Delta\alpha \leqslant 0,036°$ szögfelbontású méréseket tesz lehetővé. Az inkrementális enkóderek kedvezőtlen tulajdonsága, hogy segítségükkel abszolút szöghelyzet csak az index ponthoz viszonyítva határozható meg. Ez azt jelenti, hogy egy készülék bekapcsolása után a forgástengelyhez kapcsolt inkrementális enkóderrel csak az indexpozícióra (referenciapontra) állás után lehet követni a tengely abszolút pozícióját.



3-4. ábra. Abszolút optikai enkóder működése[**?**]

Az inkrementális enkóderekkel szemben az abszolút enkódereket úgy tervezték, hogy kiolvasásukkal minden egyes időpillanatban ismert legyen a tárcsa (és így a forgástengely) abszolút pozíciója. Működésük alapelve azonos az inkrementális enkódernél ismertetett sötét és világos szegmensek optoelektronikai érzékelésével, csak az optokapuk számában és a kódolás módjában van különbség. A forgó tárcsa koncentrikus gyűrűkre van felosztva úgy, hogy a belső körgyűrű két részt, egy sötét és egy világos szegmenst tartalmaz. Ez a belső gyűrű adja a bináris kód legnagyobb helyiértékű bitjét. A sugár mentén kifelé haladva az egymást követő gyűrűk szegmenseinek száma megduplázódik az előzőhöz képest. A külső gyűrű adja a bináris kód legkisebb helyiértékű bitjét. A gyűrűk kódjának kiolvasása sugárirányban elhelyezett optokapuk segítségével történik. Az ilyen módon kódolt tárcsa esetén egy adott pozícióból a szomszédosba forgatva a tárcsát egyszerre több bit is megváltozik. Egy-egy bit hibás kiolvasása ezért nagy hibát okozhat az abszolút pozíció meghatározása során. Ezért bináris kódolású tárcsákat csak a kis felbontású forgójeladókban használnak. A kiolvasási hiba csökkentésére vezették be a Gray-kód tár-

csákat, amelyek esetén a szomszédos szögpozíciókba való átlépés során a kódban csak egy bit változik. A legtöbb optikai elven működő abszolút enkóderben ezt a kódolási módszert használják.

3.2. Körpályás potenciométerek

Amennyiben a vezető réteget egy körív mentén alakítják ki, úgy körpályás potenciométerről beszélhetünk. Ebben az esetben az elektromos kontaktust biztosító csúszkát a körív középpontján átmenő tengely forgatja az adott körpályán, így a potenciométer szögelfordulás detektálására alkalmas. Egy α_0 szögtartományú vezetőpályát tartalmazó potenciométer esetén a szögelfordulás és a megfelelő ellenállások közötti kapcsolat:

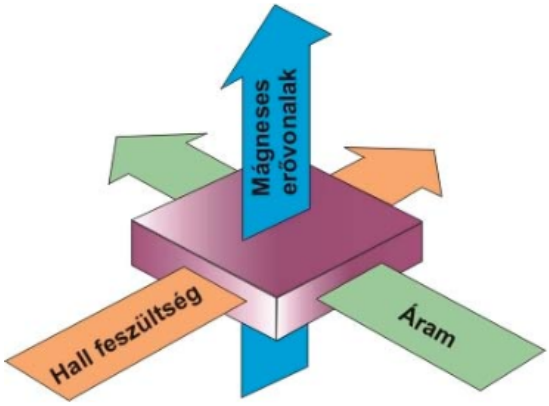
$$\alpha = \alpha_0 \frac{R_{AC}}{R_{AB}}$$

Amennyiben a csúszkát egy csavarmenetes mechanika egy hengeres felületre felcsévélt huzal mentén helikális pályán mozgatja, úgy helikális potenciométerhez jutunk, ami szintén a szögelfordulás mérésére alkalmas. Az ilyen potenciométerekkel többszörös körbe fordulás lehetősége miatt nagyobb szögtartományban mérhetünk elfordulást.
A potenciométeres útadók, helyzetmérők legnagyobb hátránya, hogy működtetésük során az ellenálláspálya és a súrlódó csúszka között történik az áramátvezetés. Ezek a mérőeszközök precíziós mérésre nem alkalmasak, mivel a csuszka súrlódása miatt hiszterézissel (késleltetéssel) rendelkeznek.

3.3. Mágneses inkrementális jeladók

Mivel az optikai eszközök bizonyos környezeti feltételek mellett korlátozottan, vagy egyáltalán nem alkalmazhatók, olyan eszközöket kellett kifejleszteni, melyek hasonló kimenő paraméterekkel rendelkeznek, de környezettel szembeni tűrőképességük lényegesen magasabb. Ezt a célt szolgálják az egyre jobban elterjedő, árban és műszaki paramétereiben is versenyképes mágneses elven működő forgójeladók. A mágneses elven működő inkrementális forgójeladókban is egy, a tengelyhez rögzített tárcsa elfordulását érzékelik. Általában két metódus közül választanak: vagy a forgó tárcsa peremén elhelyezett mágnesezhető gyűrűt sűrűosztásokkal felmágnesezik és a pólusok szenzor előtti elmozdulásából adódó mágneses tér változást érzékelik, vagy egy fixen elhelyezett, állandó mágnes által létrehozott mágneses mezőben keletkező változást érzékelik, amely változást egy sűrű fogazású acél tárcsa fogainak elfordulása okozza. A mágneses mező változása az érzékelőkben szinuszos jelet generál, amelyből egy speciális áramkör segítségével nagy felbontású

négyszög alakú jelet állítanak elő. A mágneses érzékelők fizikai elrendezése –hasonlóan az optikai eszközökéhez – biztosítja a fázisban 90°-al eltolt kimenő csatornákat. A mágneses mező változásának érzékelése leggyakrabban Hall elemes, vagy magnetorezisztív szenzorokkal történik. Ha egy félvezető lapkán áram folyik keresztül és a lapkát rá merőleges mágneses térbe helyezik, az áram folyására merőleges irányban, a lapkán feszültség keletkezik. Ez a feszültség a Hall feszültség és ezt a fizikai hatást hívják Hall effektusnak.



3-5. ábra. Hall effektus[**?**]

Ezt a jelenséget alkalmazzák a mágneses forgásjeladóknál.

3.4. Sebességmérés tachométerekkel

Az egyenfeszültségű tachométer dinamók felépítése hasonló az egyenáramú kommutátoros motor felépítésével, működésük azonban pont annak a fordítottja, vagyis az elektromágnest tartalmazó rotor külső mágneses térben való forgatása következtében a rotorban feszültség indukálódik. Az indukált váltakozó feszültséget a kommutátor egyenirányítója, így a tachométer dinamó a fordulatszámmal arányos egyenfeszültséget ad. Az egyenfeszültség polaritásából a forgásirány is meghatározható.

3.5. Gyorsulásmérés

Egy tehetetlen tömeg gyorsításához erő szükséges. Az erő mérésével, Newton II. törvénye alapján a gyorsulás is meghatározható. Amennyiben az erő rugalmas testek deformációját okozza, úgy a deformáció mérése történhet nyúlásmérő bélyeggel, LVDT-vel vagy differenciál kondenzátorral is. A piezoelektromos effektussal történő erőmérés szintén alkalmas a gyorsulás meghatározására. A kapacitív és piezoelektromos elven működő

gyorsulásérzékelők különösen alkalmasak a miniatürizálásra. Napjainkban a gyorsulásérzékelők nagy része az ún. MEMS technológiák alapján készülnek.

3.6. Inerciális mérőegyég

Az IMU [8] (Inertial Measurement Unit - Inerciális mérőegység) a lineáris gyorsulást, a szögsebességet, és esetleg magnetométerrel kiegészítve digitális iránytűként a mágneses pólusokat méri. Az alacsony költségű IMU-k általában a MEMS technológiát alkalmazzák a miniatűr mechanikai elemek mozgásának mérésére az IMU-n belül. Az IMU-k 1, 2 vagy 3 tengelyben tudnak információt szolgáltatni. A legtöbb alkalmazáshoz 3 mérési tengelyre van szükség. Ahhoz, hogy megkapjuk a lineáris és a szöghelyzet becsléseket a nyers IMU adatokat integrálni kell, ez az integrálás folyamatos eltolódást okoz a kapott becslésekben, mivel minden helytelen nyers mérés a sebesség pillanatnyi és állandó hibáját okozza. Számos tényező befolyásolja a gyorsulásmérőkből és giroszkópokból származó nyers mérések pontosságát. Például az IMU-nak nem teljesen lineáris a karakterisztikája, és a különböző szenzorokból származó jelek különböző mértékben zajosak lehetnek. Mégis nagyon praktikus eszközök mert használatukkal elég sok információ tudható meg a megfigyelt test állapotairól.

4. Eszközök közötti kommunikációs megoldások

A rendszer szabályozását több eszközzel fogjuk megvalósítani. Ehhez az eszközöknek képesnek kell lenniük az egymással való kommunikációra. Az eszközök közötti kommunikáció kétféleképpen valósulhat meg: vezetéken keresztüli és vezeték nélkül kommunikációval. Mechanikai rendszerek szabályozásánál tipikusan vezetékes kommunikációt szoktak alkalmazni mivel ez a módszer nagyobb kommunikációs sebességet és kisebb késleltetést eredményez. Első fázisában vezetékes kommunikáció segítségével valósítom meg a célkitűzést.

A továbbiakban kísérletet tettem a szenzor adatok vezeték nélküli begyűjtésére hiszen így a kísérletek során nem merül fel a vezetékek feltekeredésének problematikája. Ez kifejezetten a második kartagra szerelt szenzorok esetén jelenthet problémát, mivel ennek a mozgását közvetlenül nem tudjuk befolyásolni, és hibás szabályozás hatására előfordulhat, hogy ez a kartag többször is körbe fordul a tengelye körül, ami vezetékes kapcsolat esetben kárt tehet a berendezésben. Ez a probléma az első kartag esetén nem olyan jelentős mivel a motor által biztosított mozgásteret szoftveresen könnyen limitálhatjuk. Azonban ebben az esetben is hasznos lehet a vezeték nélküli kommunikáció alkalmazása, mivel így nem korlátozódik a rendszer mozgástere. A következőkben röviden bemutatom az inga megvalósításához kiválasztott vezetékes és vezeték nélküli technológiákat.

4.1. UART kommunikáció

A mikrokontrollerek között az egyik legelterjedtebb kommunikációs protokoll az UART (Universal Asynchronous Receiver Transmitter) [11] , mely soros, aszinkron átvitelt tesz lehetővé. Alapértelmezetten három vezeték szükséges a működéséhez: egy fogadó (RXD), egy küldő (TXD), valamint egy földelés (GND) vezeték. A vonalak feszültségszintjei a szabványos TTL (Tranzisztor - Tranzisztor Logika) jelszinteknek felelnek meg, azaz a logikai 0, vagyis az alacsony jelszint: 0 V – 0,8 V közötti tartománynak, míg a logikai 1, azaz a magas jelszint a 2,4 V – 5 V közötti feszültségtartománynak felel meg.

Mivel az UART aszinkron protokoll, vagyis nincsen közös szinkronizáló órajel a felek között a kommunikáció alatt, így rendkívül fontos a kommunikációban résztvevő két fél megfelelő beállítása, máskülönben nem tudják értelmezni az egymásnak küldött adatokat illetve vezérlő jeleket. Alapvetően a kommunikáció paraméterei a következők:

- Adatátviteli sebesség, amely azt határozza meg, hogy a bitek milyen gyorsan követik egymást, azaz egy bit időegységben milyen „hosszú”. A szabványos értékek a következők szoktak lenni: 150, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200 bit/s.
- Adatbitek száma, azaz egy üzenetben hány darab adatbit található. A szabványos értékek a következők szoktak lenni: 5, 6, 7, 8, 9.
- Paritásbit, azaz páros, páratlan vagy semmilyen paritás bit alkalmazandó.
- Stop bitek száma, azaz milyen „hosszú” legyen egy stop bit. A szabványos értékek a következők szoktak lenni: 1 bit, 1,5 bit, 2 bit.

Az UART illetve az RS-232 üzenetformátumot is definiál, melyet komolyan befolyásolnak a kommunikációra vonatkozó fentebb leírt beállítási lehetőségek

4.2. Vezeték nélküli hálózati kommunikáció

A Wi-Fi[12] vezeték nélküli mikrohullámú kommunikációt megvalósító szabvány. Segítségével lehetőség van vezeték nélküli módon felcsatlakozni internet hálózatra. A Wi-Fi két rádióhullám frekvenciatartományban használható, a 2,4 GHz-es és az 5 GHz tartományan. Több Wi-Fi szabvány is létezik melyek különböző kommunikációs sebességet és hatótávolságot biztosítanak.

Az internet két fő protokollt használ a szállítási rétegben, az egyik összeköttetés-alapú, a másik összeköttetés nélküli. A két protokoll kiegészíti egymást. Az összeköttetés nélküli protokoll az UDP, amely csomagokat továbbít alkalmazások között, rájuk bízva, hogy erre egy olyan protokollt építsenek, amilyenre szükségük van. Az összeköttetés-alapú protokoll pedig a TCP, amely összeköttetéseket hoz létre, újra küldéssel megbízhatóvá teszi azokat, emellett forgalomszabályozást és torlódáskezelést is megvalósít, mindezt az alkalmazás helyett teszi.

Mivel az UDP kommunikáció nem végez semmilyen plusz elenőrzést így használatával gyorsabb kommunikáció valósítható meg ezért a megvalósítás során ezt a protokolt fogom használni. Az UDP olyan szegmenseket használ az átvitelhez, amelyek egy 8 bájtos fejrészből, valamint a felhasználói adatokból állnak. A két port a végpontok forrás- és a célgépen belüli azonosítására szolgál. Az UDP használatának tulajdonképpen az a legnagyobb előnye a nyers IP használatával szemben, hogy a fejrészben megtalálható a feladó és a címzett port száma is. A port számokat tartalmazó mezők nélkül a szállítási réteg

nem tudná, hogy mit kezdjen a csomaggal. A segítségükkel azonban a beágyazott szegmenseket a megfelelő alkalmazásoknak tudja kézbesíteni.

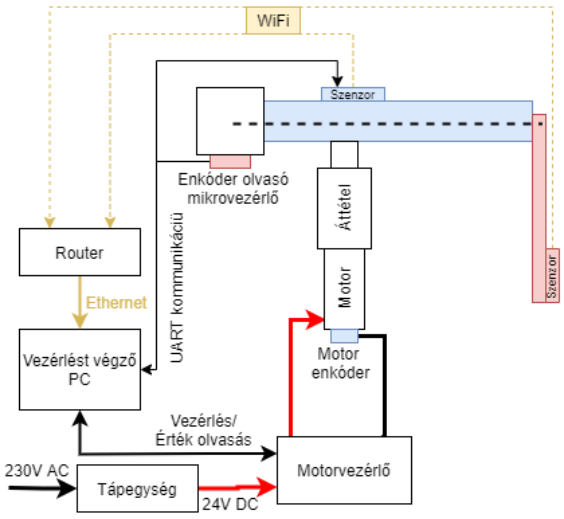
Érdemes néhány olyan feladatot külön is megemlíteni, amit az UDP nem végez el. Az UDP nem végez forgalomszabályozást, torlódáskezelést vagy újra küldést egy rossz szegmens vétele esetén. Ez mind a felhasználói folyamatokon múlik. Amit elvégez: egy interfészt biztosít az IP-protokoll használatához, azzal a többletképességgel, hogy a portok használatával egyszerre több folyamatot képes demultiplexelni, valamint szükség esetén hibajelzést biztosítani végponttól végpontig.

5. Rendszerterv

A rendszer teljes struktúrájának megtervezésénél szempont volt, hogy az inga állapotát többféle szenzorkonfigurációval lehessen majd mérni, a szenzorfúziós alkalmazások tesztelése miatt.

Első lépésként a rendszer teljes állapotát a körülményekhez mérten a lehető legpontosabban mérve valósítom meg a szabályozási célt. Ehhez a két kartag pozícióját nagy felbontású enkóderekkel mérem, a sebességeiket pedig a szabályozást végző programmal számítom a pozíciókból, tehát ebben a lépésben még nem használok állapotbecslést.

A szabályozást végző programot futtató számítógép USB-n keresztül lesz összekötve a motor vezérlővel, valamint egy mikrovezérlővel, amely továbbítja az enkóderek által mért értékeket a PC felé.



5-6. ábra. Rendszer felépítése

Következő lépésben mindkét kartagon elhelyezek egy-egy IMU-t és a szabályozást az ezekből származó adatok alapján próbálom megvalósítani. A második kartagra való szenzor elhelyezésénél arra a döntésre jutottam, hogy az inga szabad mozgásának megtartása okán a szenzor az adatokat vezetéknélküli kommunikációval továbbítsa a feldolgozást végző PC felé. Az első kartag IMU-val való mérése kapcsán is szeretném ezt a módszert alkalmazni, hiszen így bejárható marad az inga teljes mozgástartománya.

6. A rendszer megvalósítása

Ebben a fejezetben ismertetem a rendszer megépítéséhez felhasznált hardveres eszközöket. Az eszközöket úgy választottam ki, hogy a lehető legjobban teljesíteni tudjam a rendszertervben meghatározott struktúrát.

6.1. Kiválasztott eszközök

6.1.1. Beavatkozó

Az inga mozgatását egy DC motor biztosítja. A motor kiválasztásánál fontos szempont volt, hogy képes legyen a rendszer mozgatásához megfelelő nyomatékot kifejteni, valamint enkóderrel legyen felszerelve, ezáltal képes legyen az első kartag pozíciójának mérésére. Matlabban előzetesen szimuláció útján vizsgáltam meg, hogy mekkora nyomatékokra lehet szükség a szabályzás megvalósításához. A szabályzót próbáltam minél agresszívabbra tervezni, így maximalizálni a szabályozás során fellépő nyomatékot.

Az így kapott eredmények segítségével lett kiválasztva a Maxon DCX35L 24 voltos kefés DC motor, mely 121 mNm nyomaték folyamatos kifejtésére képes. A motor nyomaték állandója 29.3 mNm/A, névleges sebessége 7160 fordulat per perc. A nyomaték növelésének érdekében a motorra felszereltem egy Maxon GPX37 26:1 arányú áttétet. A motor és ezáltal az első kartag pozíciójának mérésére a HEDL 5540 inkrementális enkóder lett kiválasztva. Az enkóder felbontása 500 osztás per fordulatú. A motorra felszerelt áttét miatt azonban ezt meg kel szorozni az áttét arányával, így az első kartag pozícióját 13000 osztás per fordulat pontossággal fogom tudni mérni.

6-7. ábra. Maxon DCX35L motor

Motor vezérlőnek a Maxon Epos4 50/5[13] lett kiválasztva, ehhez csatlakozik majd a motor és a rajta található enkóder. A vezérlő rendelkezik beépített pozíció és sebesség szabályozóval, valamint nyomaték szabályozás is végezhető a segítségével. Az eszköz rendelkezik EtherCAT, CAN és USB 2.0 interfészekkel. Ezen felül a vezérlőhöz tartozik egy program, az EPOS Studio melynek segítségével irányítható a motor mozgása. Ezen felül hozzáférhetők a program által is használt DLL-ek melyek segítségével saját programból is adható utasítás a vezérlőnek.

6-8. ábra. Maxon Epos4 50/5

A motorvezérlő és a motor árammal való ellátásához egy LRS-150-24[14] 24 voltos 156W teljesítményű tápegységet szeretnék használni.

6.1.2. Szenzorok

A második kartag pozícióját szintén egy inkrementális enkóderrel fogom mérni. Erre a célra a HEDS 9040[15] optokaput választottam ki, valamint egy 2000-es felbontású enkóder tárcsát. A kiválasztott olvasófej 3 csatornás, így képes a tárcsa teljes körbefordulását is detektálni.

Az enkóder működtetését egy Arduino Nano[16] mikrovezérlővel fogom megvalósítani. A Nano működési feszültsége 5 Volt, magja egy ATmega328 mely 16MHz órajelen működik. Erre a mikrovezérlőre egyszerű programozhatósága, alacsony ára és kis mérete miatt esett a választás. Nem elhanyagolható szempont az sem, hogy rendelkezik 2 darab hardveres megszakítású kivezetéssel, amely segítségével jobban garantálható, hogy az enkóder olvasása közben ne veszítsen lépést, pozíciót. Továbbá az eszköz támogatja az I2C,

SPI, valamint UART kommunikációt.

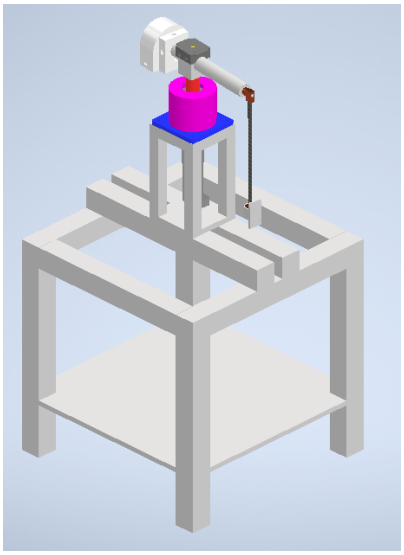
A rendszerterv második pontjában megfogalmazott célok kielégítésére kompakt megoldást próbáltam találni. Így esett a választás az M5Stick-C[17] fejlesztői boardra. Az eszköz alapját egy ESP-32 2 magos 240 MHz -es mikrovezérlő adja. A mikrovezérlő támogatja a Wifi és Bluetooth-os kommunikációt is. Maga a teljes modul tartalmaz egy 6 tengelyes MPU6886 IMU-t. A kiválasztás legfontosabb szempontja az volt, hogy az egység tartalmaz egy beépített 95 mAh elemet. Ennek segítségével teljes egészében megvalósíthatóvá válik a Furuta inga állapot mérése vezetéknélküli módon. Az eszközön továbbá található egy USB-C és egy GROVE port, valamint 3 szabadon felhasználható input-output pin. Ezen interfészek segítségével az M5Stick-C könnyen csatlakoztathat PC-hez vagy más mikrovezérlőkhöz. Az eszköz programozására rendelkezésre áll egy programkönyvtár melynek segítségével egyszerű módon használhatók ki az eszköz nyújtotta lehetőségek.

6-9. ábra. M5Stick-C fejlesztői board

6.2. A rendszer megépítése

Az ingáról a megépítés előtt 3D modell készült az Autodesk Inventor programmal. A program segítségével meghatározhatóvá válnak egyes fizikai paraméterek (pl: tömegközéppont távolsága), amelyek a későbbiekben a matematikai modell alapján készülő szimulációknál lesznek felhasználva. Ezen kívül az egyes alkatrészek legyártásához is szük-

ség volt az modellek előzetes elkészítésére.
A végleges modell a következő ábrán látható:

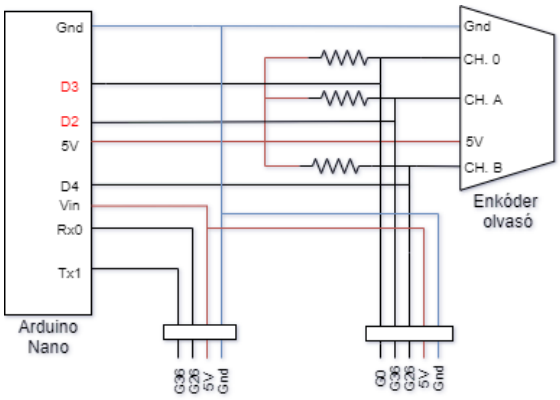


6-10. ábra. Furuta inga 3D modellje

Első körben a matematikai modellel is leírt két kartag modellezése és megépítése valósult meg. Az inga első kartagja egy 25 mm átmérőjű alumínium csőből készült el, melynek mind a két végére egy-egy csapágyat helyeztünk el és ezeken keresztül vezettünk egy acélrudat. Ennek az acélrúdnak az egyik feléhez rögzítettük a második kartagot, míg a másik felére az előzőekben bemutatott enkóder tárcsát szereltük fel. A második kartagot karbon szálból készítettük el melynek végére egy rézből készített súlyt helyeztünk.

Az enkóder tárcsa védelme érdekében az első kartagra felerősítettünk egy 3D nyomtatással készített védőházat is, melynek további szerepe, hogy erre az elemre erősítettük fel az enkóder olvasó elektronikáját is.

Az enkóder működéséhez szükséges elektronikát egy nyáklapra forrasztottuk fel. Ezen található egy tüskesor aljzat melybe csatlakoztatható a fent bemutatott Arduino Nano mikrovezérlő. A nyáklapra lett felforrasztva az optokapu és az működéséhez szükséges ellenállások. Az enkóder 1 -es csatornája, valamint a teljes körbefordulást detektáló 0-s csatornájának kivezetései a mikrovezérlő megszakítás lábaira lettek rákötve.

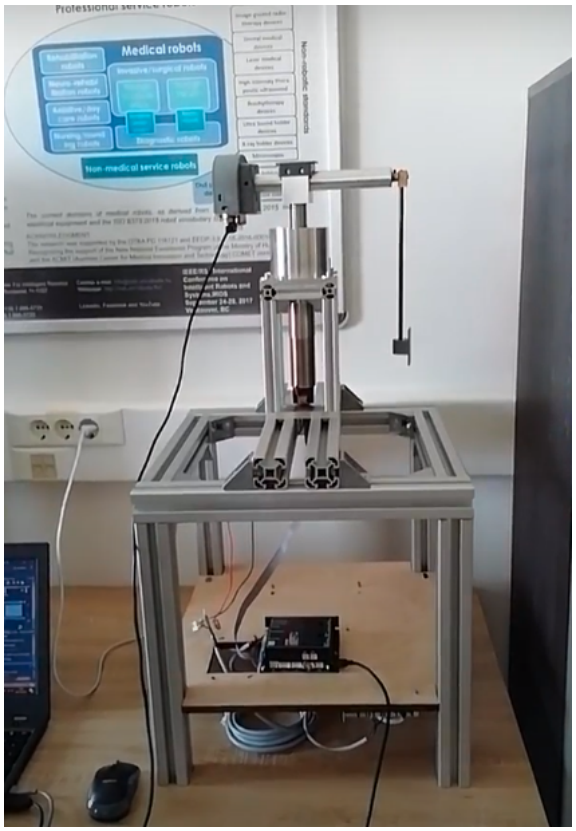


6-11. ábra. Enkóder olvasó elektronika

Ahogy az rajzon is látható a nyáklapon elhelyeztünk további két tüskesort, ezek teret adnak ahhoz, hogy az enkóder értékeit többféle úton olvashassuk ki. Egyik ilyen megoldás mikor az olvasást a Nano végzi és UART kommunikáció segítségével továbbítja a pozíció értékeket. A Nano a 0 és 1-es lábai szolgálnak az UART kommunikáció kiszolgálására. Ezek a lábak összeköttetésben álnak a boardon található USB-B csatlakozóval is, így azonos programot futtatva a mikrovezérlő képes az általa mért pozíció értékeket vagy a szabályozást végző számítógép, vagy az első kartagon elhelyezett másik mikrovezérlő felé továbbítani. Másik lehetőség, hogy az M5Stick-C közvetlenül olvassa az enkóder értékeket és továbbítja a számítógép felé vezetékes vagy vezeték nélküli módon.

A használni kívánt IMU-k felszereléséhez szintén 3D nyomtatással készült el egy-egy tartó elem. Egyik ilyen tartó elemet az ábrán látható módon az első kartag rögzítő elemére helyeztük el. Innen könnyedén megoldható, hogy a rászerelt M5Sstick-C-t összekössük az enkóder olvasó Arduino Nanoval. A másik tartó elemet a második kartagon található súlyra erősítettük fel.

Az elkészült ingához készítettünk egy keretet mely stabil alapul szolgál számára. Ezen felül a működéshez szükséges elektronikai alkatrészeket is itt helyeztük el. A váz 40 centiméter élű kocka, melyet ITEM profil elemekből állítottunk össze. A kockán belül elhelyeztünk egy rétegelt lemezlapból kivágott polcot és az alkatrészeket ehhez erőssítettük hozzá.



6-12. ábra. Megépített rendszer

6.3. Rendszer működését szolgáló programok

A rendszer egyes elemeinek működéséhez különböző programok szükségesek. Ezek a mérési adatok begyűjtését és továbbítását végző mikrovezérlők, valamint a szabályozást végző PC. Ebben a fejezetben röviden bemutatom ezen programok feladatát, felépítésüket.

6.3.1. Mikrovezérlőkön futtatott programok

Első körben tekintsük át a mikrovezérlőkre elkészített programokat. Ezeket a programokat a PlatformIO fejlesztő környezet segítségével készítettem el és töltöttem fel a mikrovezérlőkre. A környezet előnye, hogy rengeteg boardot támogat és könnyű vele külső csomagokat a saját projektünkhöz hozzáadni.

A Nano-ra megírt program nagyon egyszerű hiszen feladata csupán az enkóder kezelése és az adatok továbbítása soros kommunikáció útján a rendszer további elemei felé. A program elején lefutó setup függvényben elindítjuk a soros kommunikációt, amihez be

kell állítani, hogy a kommunikáció milyen sebességgel történjen. Ezt a board által támogatott maximális 115200 bit per másodpercre állítottam be, továbbá úgy konfiguráltam a kommunikációt, hogy 8 adat és 2 stopbitet használjon páratlan paritás figyelés mellett. A setup függvényben továbbá beállításra került a megszakításlábakhoz való függvények hozzárendelése. A 2-es megszakítás láb (melyhez az enkóder A csatornáját csatlakoztattuk) állapotának megváltozása esetén lefutó függvény kiolvassa az enkóder A és B csatornájának állapotát és ez alapján eldönti, hogy az enkóder melyik irányba fordult és ennek alapján növeli a pozíciót tároló változó érékét. A másik megszakítás láb az enkóder teljes körbefordulását figyeli és annak megtörténésekor nullázza a pozíció változót.

A program főciklusán belül a pozíció változó soros kommunikációra való kiküldése történik. A ciklusban az adat küldést mindig azonos időközönként valósítjuk meg. A 16 bites integer változóként tárolt pozíciót 2 bájtként küldöm el.

```
void loop()
{
  now = micros();
  if (now >= next)
    {
      buf[0] = counter & 255;
      buf[1] = (counter >> 8)  & 255;
      Serial.write(buf, sizeof(buf));
      next = next + measurement_period;
    }
}
```

6-13. ábra. Időzítést végző ciklus

Az első kartagra szerelt M5Stick-C feladata, hogy a benne található IMU segítségével mérje a kartag állapotát, fogadja az Arduino Nano által elküldött pozíció adatokat és mindezeket WiFi-s kommunikáció útján továbbítsa a PC felé.

A program megírásához felhasználásra került az M5Stick-C saját könyvtára melynek segítségével egyszerű módon használhatóak az eszköz nyújtotta lehetőségek. A program indulásakor az eszköz megpróbál csatlakozni a WiFi hálózathoz, amint ez sikeres a folytatja a program futását. Következő lépésben meghívja az IMU inicializálását végző metódust. Az IMU segítségével mérhető az eszköz szögsebessége, gyorsulása és szöghelyzete 3 tengely mentén, de ezen értékek kiolvasásához időre van szükség. Mivel az M5Stick-C két magos processzorral rendelkezik így lehetőség van arra, hogy az IMU értékeinek kiolvasását külön szálon végezhessem, és ezzel gyorsabbá tehessem a program futását.

Minden az IMU-ból kiolvasott értéket 4 bájtos lebegőpontos változókba tároltam el. A setup függvény utolsó lépéseként elindítom a soros kommunikációt, ám a Nano-tól eltérően az M5Stick-C esetén megadható, hogy mely lábakat használja erre a célra az eszköz. A 26-os és 36-os szabadon felhasználható lábak lettek kiválasztva erre a célra, minden más beállítás azonos a fent leírtakkal.

A főciklusban bonyolítom le kommunikációt. Ellenőrzésre kerül, hogy érkezett e csomag a soros kommunikáción, ha igen, letárolásra kerülnek a kapott értékeket. Ezután UDP-n keresztül byte tömb ként továbbítódnak a megkapott enkóder értékek és az IMU mérésének eredményei. Minden UDP-n elküldött csomag 38 bájtot tartalmaz a következő formában:

1-2 byte	3-14 byte
Enkóder	Gyorsulás(X,Y,Z)
15-26 byte	27-38 byte
Szögsebesség(X,Y,Z)	Szögelfordulás (roll,pitch,yaw)

A második kartagra szerelt mikrovezérlőn futó program a fentivel megegyező leszámítva, hogy ez az eszköz nem végez soros kommunikációt.

6.3.2. Rendszer vezérlését végző program

A fent leírt elemek működését egy számítógép és a rajta futtatott vezérlést végző program fogja össze. A program a CMake felhasználásával készített C++ nyelven írt konzol alkalmazás. A CMake egy projekt környezet előállítását végző eszköz, melynek segítségével kezelhetjük a projekt számára szükséges függőségeket és különböző build-elési szabályokat hozhatunk. Használata nem operációs rendszerhez és fejlesztői környezethez kötött, így bármilyen platformon előállítható vele a fejleszthető és futtatható projekt a forrásfájlokból.

A projekt fontosabb függőségei:

- EposCmd: a motorvezérlő parancsait tartalmazó könyvtár melynek segítségével irányítható a motor működése
- SensorFusion: A tesztelni kívánt szenzorfúziós könyvtár, ami implementálja az állapotbecsléshez szükséges Kálmán szűrő algoritmust és absztrakt osztályokon keresztül definiálható benne a rendszerdinamika.

A program feladata az egyes szenzorok felől érkező adatok fogadása a különböző kommunikációs csatornákon keresztül, a beavatkozó jel értékének meghatározása és a motor irányítása a motorvezérlőn keresztül. Ezen kívül a program segítségével loggolhatóvá válik a rendszer állapota, ezáltal a későbbiekben vizsgálhatóvá válik a viselkedése is. Ezen feladatok ellátását egy-egy külön osztály biztosítja. Röviden tekintsük át ennek működését.

A MotorHandler a motorvezérlő könyvtár wrapper osztálya. A motor vezérlő könyvtár rengeteg lehetőséget biztosít a motor vezérlésére és konfigurálására, a wrapper osztály ezeket teszi egyszerű módon elérhetővé, amelyekre a projekt működéséhez is szükség van. Ezek a vezérlővel való kommunikáció elindítása, a motor állapotok kiolvasása (pozíció, sebesség, áram), valamint a motor különböző módokon való vezérlése, pozicionálása. A motor működéséhez szükséges konfigurációkat előzetesen az EPOS Studio nevű program segítségével tudom elvégezni egyszerű grafikus felületen és ezeket a beállításokat a motorvezérlő tárolja, így ezekre a konfigurációs eljárásokra a szabályzó programon belül nincs szükség. A motorvezérlő három fontosabb mozgatási módot tesz lehetővé a motoron: adott pozícióra mozgatást, megadott sebességgel való mozgatást és előírt áram hatására történő mozgatást. Ezen módokban történő mozgatások a wrapper osztályon keresztül is elérhetők, de a szabályzási cél megvalósításához a motort a rákapcsolt áram előírásával szeretném mozgatni.

Következő a SensorListener nevű osztály, amely elindítási módtól függően fogadja és tárolja a szenzor adatokat. Az osztály biztosítja egy UDP szerver futtatását, amely IP cím alapján külön objektumokba tárolja a beérkező szenzor adatokat. Ezen felül lehetőséget biztosít a soros kommunikáción érkező értékek fogadására. Mindkét mód esetében a program a kommunikációt külön szálon figyeli.

Ezen két osztály példányait használja a ControllLoop osztály melyben definiálhatóak a különböző feltételek szerinti szabályzások és mérések. A szabályozást végző metódus engedélyezi a motor vezérlőt, majd 0 pozícióra állítja a motort. Ezután beállítástól függően megnyitja a szenzorok felé a szükséges kommunikációs csatornát. Mint ahogy azt a matematikai modellezésnél láthattuk a stabilizálást végző lineáris szabályzó csak kis kitérés esetén működik ezért a program várakozik míg a második kartagot a felső pozíció közeli állapotba mozgatjuk. Ha ez sikeresen megtörtént elindul a szabályozást végző ciklus. Itt azonos időközönként a szenzorértékek alapján kiszámolásra kerül a szükséges beavatkozó jel.

A program fő fájljában a fenti osztályok példányosítása és egy egyszerű konzolos menü megjelenítése történik. A menü lehetőséget nyújt a szabályozás elindítására, valamint az egyes osztályok funkcióinak manuális meghívására (pl: motor pozícióra állítása, UDP kommunikáció megjelenítése stb.).

7. Szabályzó tervezés és a rendszer szimulációja

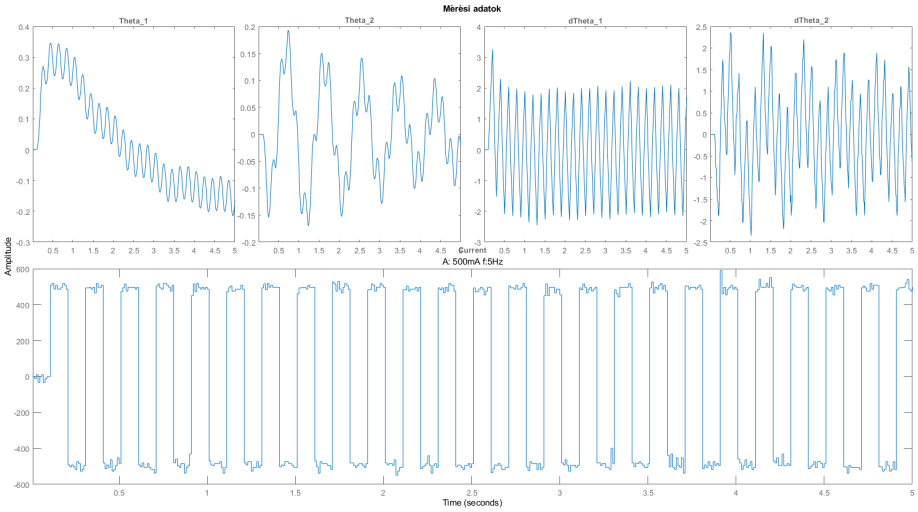
A matematikai modell és a megépített struktúra paramétereinek ismeretében elvégezhető a rendszer viselkedésének vizsgálata és a szabályzó tervezés. A szabályzás hatását szimulációk útján ellenőrizhetjük a tényleges rendszeren végzett kísérletek előtt. Ezen feladatokat Matlab és Simulink segítségével fogom elvégezni.

7.1. A rendszer identifikálása

A szimulációhoz és a szabályzó tervezéshez szükséges modell paramétereit a rendszeren elvégzett mérések alapján határoztam meg a Matlab System Identification Toolbox[18] segítségével. Ez az eszköz képes rendszerek mért bemenete és kimenete alapján lineáris és nem lineáris modellek becslésére. A modell becslés kétféle elv szerint végezhető el, ezek közül az egyik a fekete a másik a szürke doboz modell megközelítés. A fekete doboz modell esetében nem ismerjük a rendszer belső összefüggéseit. Ez a megközelítés alkalmas lehet, ha becsülni kívánt rendszer bonyolult vagy ha számunkra nem fontos a belső működésének ismerete csupán az, hogy a modell kimenete közel azonos legyen a valós rendszer kimenetével.

A szürke doboz modell esetén, van információnk a rendszer belső összefüggéseiről, azonban a modellt leíró egyenletek paraméterei nem, vagy csak részben ismertek. Mivel a Furuta inga esetén a dinamikát leíró összefüggés ismert ezért ezzel a szürke doboz modell módszerrel határoztam meg a matematikai levezetést tartalmazó fejezetben bemutatott lineáris modell paramétereit.

A paraméter becsléshez szükséges adatokat a megépített rendszeren elvégzett mérésekkel kaptam meg. A rendszer bemenetre különböző amplitúdójú és frekvenciájú négyszögjeleket kötöttem miközben logoltam a kartagok pozícióját és az ebből számított sebességét, valamint a tényleges bemenetet, azaz a motorra érkező áramot. A méréseket az inga alsó egyensúlyi állapotából indítottam el és a bemenetül szolgáló négyszög jelek paramétereit úgy állítottam be, hogy az inga csak kis kitéréseket tegyen a kiindulási állapottól.



7-14. ábra. Valós rendszer be- és kimenetei

A szürke doboz modell paramétereit a következő alakban kerestem:

$$A_b = \begin{bmatrix} \bar{\bar{A}}_f & [C_m, 0] * B_f \\ 0 & \begin{matrix} A_{m11} - B_{m11} * C_{m11} * P & A_{m12} - B_{m11} * C_{m12} * P & B_1 1 * I \\ A_{m21} - B_{m21} * C_{m11} * P & A_{m22} - B_{m21} * C_{m12} * P & B_2 1 * I \\ -C_{m11} & -C_{m12} & 0 \end{matrix} \end{bmatrix},$$

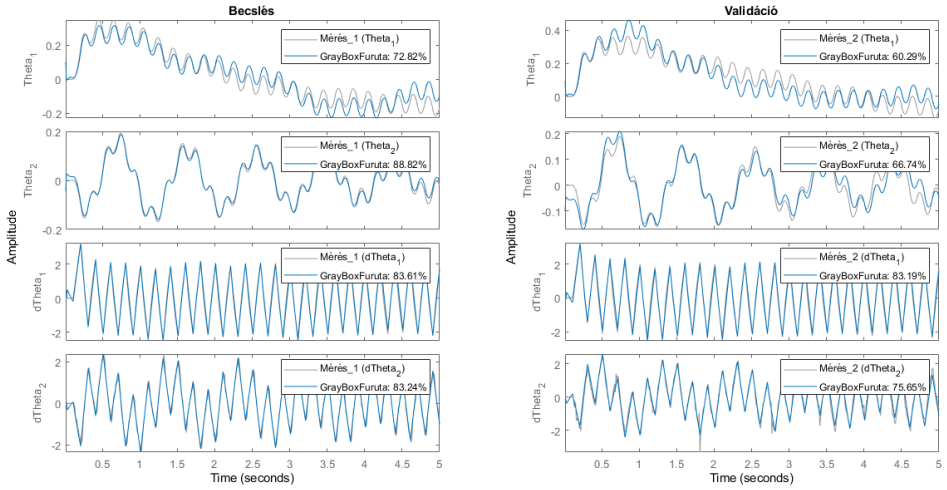
$$B_b = \begin{bmatrix} 0_{4*1} \\ B_{m11} * P \\ B_{m21} * P \\ 1 \end{bmatrix} \quad C_{becsls} = \begin{bmatrix} I_4 & 0_{4*3} \end{bmatrix} \quad D_{becsls} = 0.$$

Ez a leírás tartalmazza a mechanika rendszer (f alsó index) és a motor (m alsó index) állapotteres modelljét. A motorvezérlő a motor áramot egy PI szabályzóval állítja elő, ez szintén megtalálható a fenti modellben. A motor paramétereit annak katalógusából, a motorvezérlő szabályozási paraméterei pedig az EPOS Stúdió segítségével kaptam meg. A motor paraméterei:

Js=0.00000966 kgm^2	b=0.0042778 Nms
K_e=0,00306748 V/s	K_t=0.0293 Nm/A
R=0.346 Ω	L=0.000121 H
P=861.342 mV/A	I=3040.005 mV/Ams

Ezekből a motor állapot teres modellje a harmadik fejezet alapján írható fel.

A becslést a Matlab *greyest* függvényével végeztem el, egy a rendszeren mért adathalmaz alapján, majd egy másik független mérés segítségével ellenőriztem annak jóságát.



7-15. ábra. Identifikáció eredménye

Látható, hogy a becsült modell a valós rendszerhez hasonló kimeneteket eredményez.

Azonban az így megkapott modell az alsó egyensúlyi helyzet körüli lineáris modell, míg számomra a szabályozás megtervezéséhez a felső állapot szerinti modell szükséges. Ezt könnyen megkaphatjuk a fent ismertetett összefüggések segítségével a becsült modellből. Továbbiakban a becsléssel meghatározott mechanikai rendszer modellt fogom felhasználni a szabályzó tervezéshez.

$$A_f = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 5.51183667241668 & -1.42556513239451 & -0.997793787116279 \\ 0 & 46.9300893457984 & -1.03289596221695 & -0.349243771491141 \end{bmatrix},$$

$$B_f = \begin{bmatrix} 0 \\ 0 \\ 0.0861095537139821 \\ 0.0560928185594264 \end{bmatrix}.$$

7.2. Szabályzó tervezés

Mivel a szabályozást digitális úton fogom megvalósítani ezért a szabályzó tervezéshez a fent megkapott modellt diszkrét idejű modellé kell alakitani. A diszkrét idejű állapot teres leírás[7] hasonló alakú mint a folytonos, a különbség, hogy a rendszer dinamikáját ebben az esetben differenciaegyenletekkel adjuk meg a következő alakban.

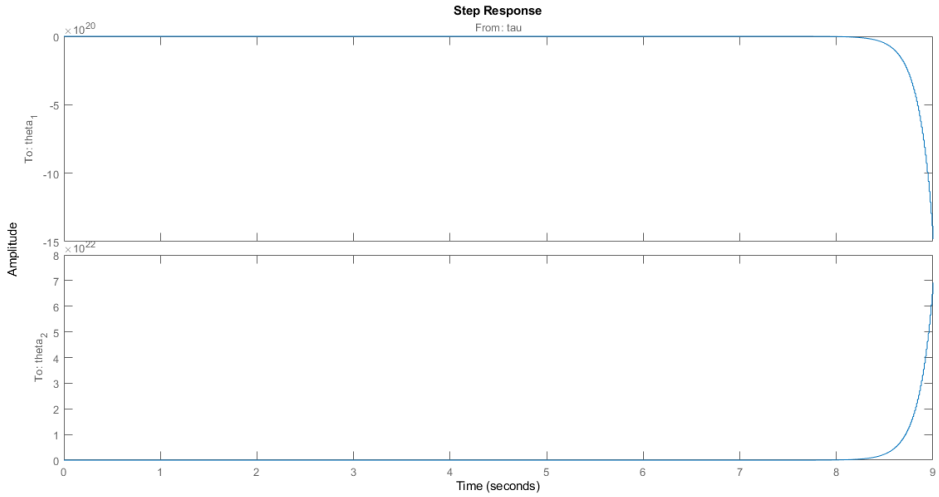
$$x[k+1] = Ax[k] + Bu[k],$$
$$y[k] = Cx[k] + Du[k].$$

Folytonos rendszer diszkretizálására többféle megoldás lehetséges. A leggyakrabban használt közelítés, amikor azt írjuk elő, hogy a diszkrét idejű rendszer ugrásválasza egyezzen meg a folytonos idejű rendszer ugrásválaszával a mintavételi időpontokban. Ezt egységugrás ekvivalens áttérésnek nevezzük. Ez a konverzió elvégezhető a Matlab c2d függvényével. A konverzióhoz meg kell adni a diszkretizációs metódust és a rendszer mintavételi idejét. Nullad rendű tartószervvel és 0.01 s mintavételi idővel a következő diszkrét állapotteres modellt kaptam.

$$A_{fd} = \begin{bmatrix} 1 & 0.000266 & 0.009929 & -0.000048 \\ 0 & 1.002343 & -0.000051 & 0.009990 \\ 0 & 0.052442 & 0.985895 & -0.009623 \\ 0 & 0.468573 & -0.010245 & 0.998905 \end{bmatrix},$$

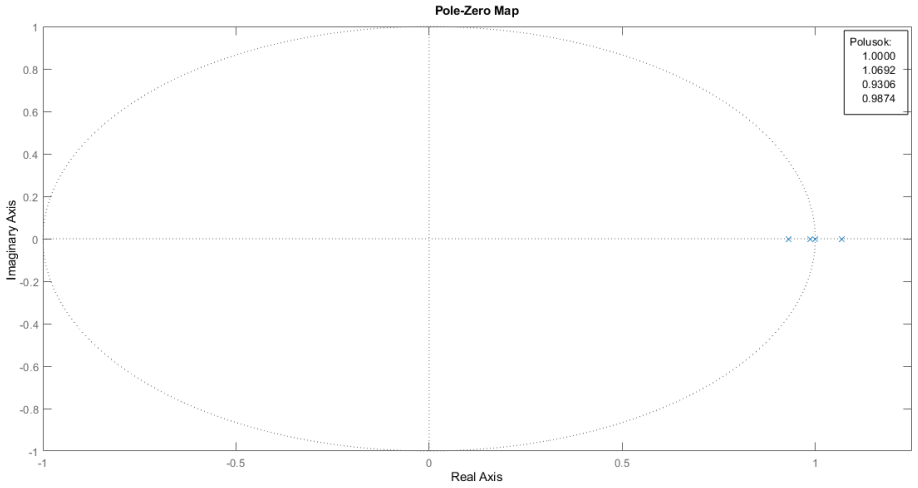
$$B_{fd} = \begin{bmatrix} 4.27596685111938e-06 \\ 2.78773631767872e-06 \\ 0.000852269574795282 \\ 0.000555974686279129 \end{bmatrix}.$$

Megvizsgálva a modell ugrás válaszát azt tapasztalhatjuk, hogy nem stabil, vagyis az állapotok exponenciálisan tartanak végtelenhez.



7-16. ábra. Nyilt kör ugrásválasza

Az instabilitás abból is látszik, hogy van olyan pólusa, ami az egység körön kívül található.



7-17. ábra. Nyilt kör pólusai

A rendszer stabillá tehető teljes állapot visszacsatolás segítségével. Ez azt jelenti, hogy a rendszer állapotait egy erősítő tényezőn keresztül visszacsatoljuk a bementére:

$$u[k] = -Kx[k] + r[k].$$

Ha u-t behelyettesítjük a dinamikát leíró modellbe akkor a következőt kapjuk:

$$x[k+1] = (A - BK^T)x[k] + Br.$$

Láthatjuk, hogy az így kapott rendszer dinamikáját leíró mátrixban szerepel az álltalunk megadott K paraméter, vagyis ennek segítségével befolyásolhatjuk a rendszer viselkedését. Ez azonban csak akkor tehető meg ha a rendszer irányítható, azaz a bemenetek és a dinamika alapján lehetséges a rendszert véges idő alatt tetszőleges állapotba eljuttatni. Ez a rendszer irányíthatósági mátrixának vizsgálatával ellenőrizhető.

$$C_o = \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix}$$

Ha az irányíthatósági mátrix rangja megegyezik a rendszer állapotának számával akkor a rendszer irányítható. A modell esetén ez teljesül, így lehetséges a teljes állapot visszacsatolás. A kérdés, hogy hogy válaszuk meg K vektor értékeit.

Több módszerrel határozható meg a K vektor értéke. Egyik módszer az úgynevezett pólus áthelyezés. Ebben az esetben meghatározzuk a zárt kör kívánt pólusait és az Ackermann formula[19] segítségével megkaphatjuk ezeket felhasználva K értékét. A zárt kör annál gyorsabb lesz, minél távolabb helyezzük a pólusokat a bal félsíkon a képzetes tengelytől. A módszer hátránya, hogy nehéz felfedezni a kapcsolatot a pólusok és a rendszer állapotainak viselkedése között, valamint szabályozáshoz szükséges beavatkozó jel nagysága között.

A másik módszer az LQ szabályozás[20]. A módszer lényege, hogy definiáljuk az alábbi költségfüggvényt:

$$J = \sum_{n=1}^{\infty} (x[k]^T Q x[k] + u[k]^T R u[k]),$$

ahol a Q és R szabadon választható négyzetes diagonális súlyozó mátrixok, melyek a rendszer eltérését az egyensúlyi állapottól és a beavatkozó munkáját büntetik. Az optimalizációs feladat célja, hogy ezt a költséget minimalizáljuk, ez pedig a megfelelő u-tól fog függeni. A K értékét ebben az estben:

$$K = (R + B^T B)^{-1} B^T P A,$$

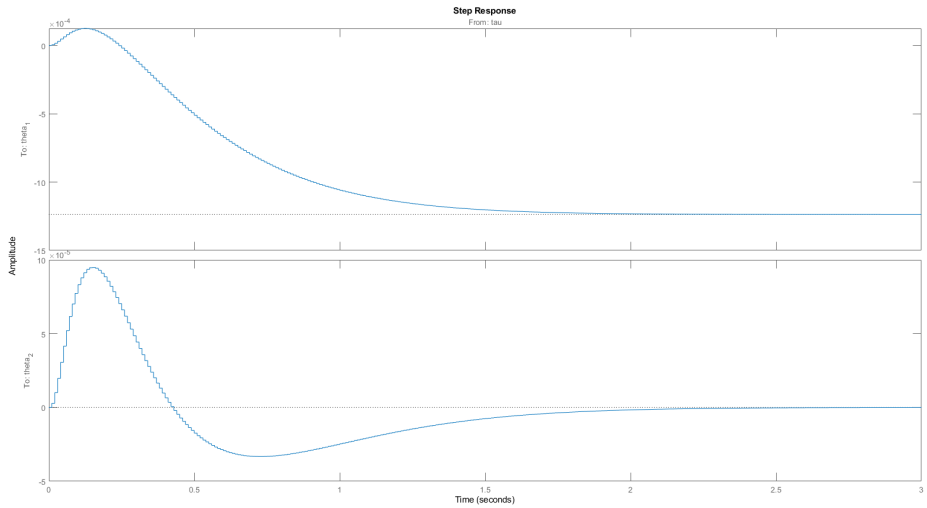
ahol P a diszkrét algebrai Riccati egyenlet[21] megoldása :

$$P = A^T PA - A^T PB(R + B^T PB)^{-1} B^T PA + Q.$$

A rendszer szabályzásához szükséges K-t az LQR segítségivel határoztam meg a következő súlyozó mátrixokkal:

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R = 0.0001, \quad K^T = \begin{bmatrix} -808.231266773576 \\ 10429.0457774995 \\ -582.481964757167 \\ 1551.69545581374 \end{bmatrix}.$$

Ahhoz, hogy ellenőrizhessük a szabályozás hatását a rendszeren vizsgáljuk meg a zárt kör ugrás válaszát.

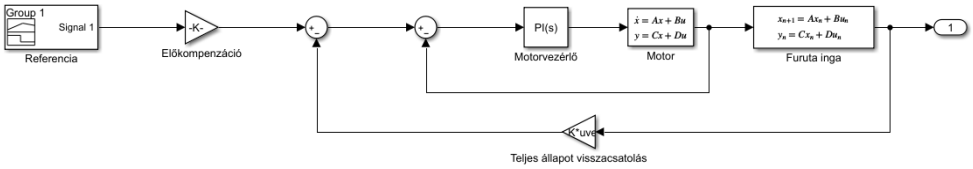


7-18. ábra. Zárt kör ugrásválasza

Látható, hogy a szabályozás hatására a második kartag kilengés után visszatér kezdeti állapotába, valamint, hogy az első kartag pozíciója is beáll egy állandósult értékre. Ez az állandósult érték azonban nem az előírt egységet veszi fel. Látható, hogy a teljes állapot visszacsatolás garantálja a stabilitást viszont a referencia követést nem. Az első kartag pozíciójának hibáját kiküszöbölhetjük, ha a referencia értéket előkompenzáción keresztül adjuk a rendszer bemenetére.Ennek az erősítésnek az értékét manuálisan addig változtattam amig az első kartag ugrásválaszának végértéke el nem éri az egyet. A meghatározott előkompenzáció értéke -800 lett.

7.3. Szimuláció

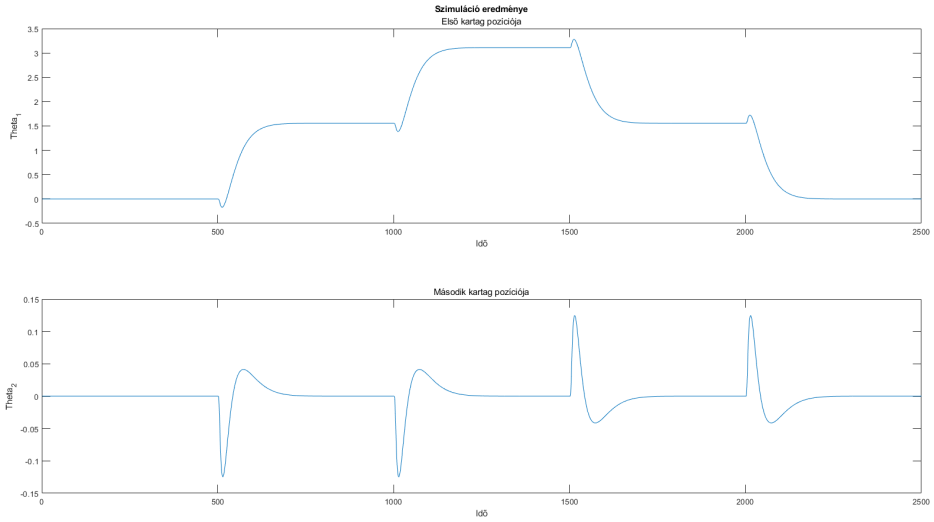
A teljes rendszer szimulációját a Simulink segítségével végeztem el. Az összeállított modell a következő képen néz ki:



7-19. ábra. Rendszer felépítése Simulinkben

A modell tartalmazza a motor áram szabályozási körét, a Furuta inga felső egyensúlyi állapota körüli linearizált diszkrét állapot teres modelljét és az fent leírtak alapján meghatározott visszacsatolási, valamint előkompenzációs értékeket.

A szimuláció során a referencia értéke nullától indulva 5 másodpercenként $\pi/2$ vel növekszik egészen π ig majd vissza csökken nullára.



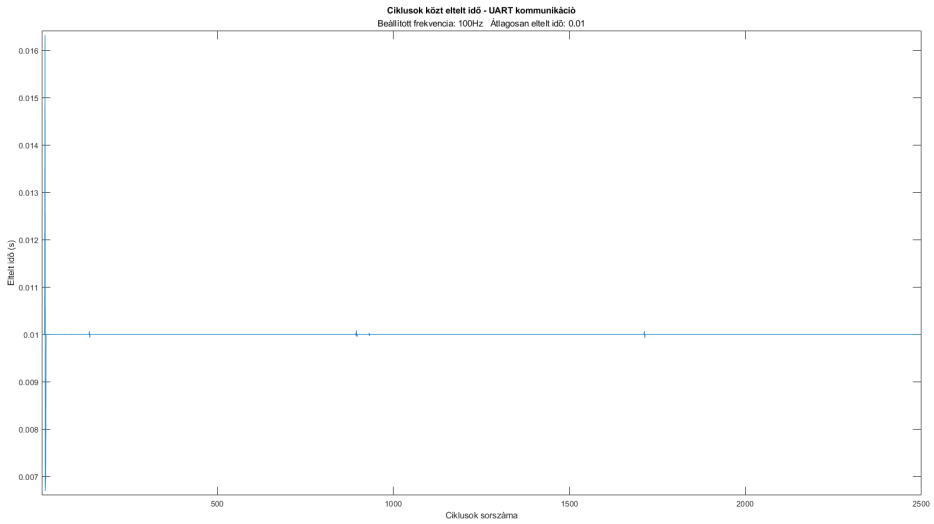
7-20. ábra. Szimuláció eredménye

Látható, hogy az első kartag jól követi a referencia értékeket és a második kartag is a kilengés után visszaáll az egyensúlyi állapotába.

8. Valós rendszeren elvégzett kisérletek eredményei

A megtervezett a szabályzó működését a valós rendszeren is teszteltem. Matlabban megkapott K vektort és az előkompenzációt a szabályzást végző programba átemelve kiszámítható a szükséges beavatkozó jel.

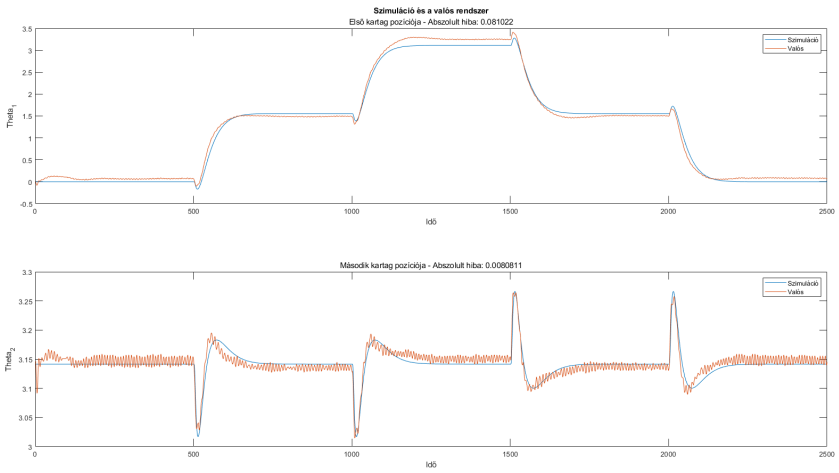
Először az inga pozícióját az Arduino soros kommunikáció útján küldi el a PC felé. Az Arduino úgy lett beállítva hogy 0.005 másodpercenként küldje el a kiolvasott enkóder értéket. A szabályozást végző PC egy I5-5300U processzorral és Windows 10 operációs rendszerrel rendelkezik. A szabályozást 100Hz-en végeztem el igazodva az előző fejezethez. Ahhoz, hogy megbizonyosodhassak arról, hogy a program képes a megfelelő frekvenciával végezni a szabályozást, ezért a szabályzási ciklusban logoltam az aktuális rendszeridőt. Kiszámolva a ciklusok közt eltelt időt a következő eredményt kaptam.



8-21. ábra. Átlagos futási idő

Látható, hogy ugyan vannak kisebb kiugrások az eltelt idők között de többnyire a program képes stabilan, 0.01 másodpercenként biztosítani a szabályzást.

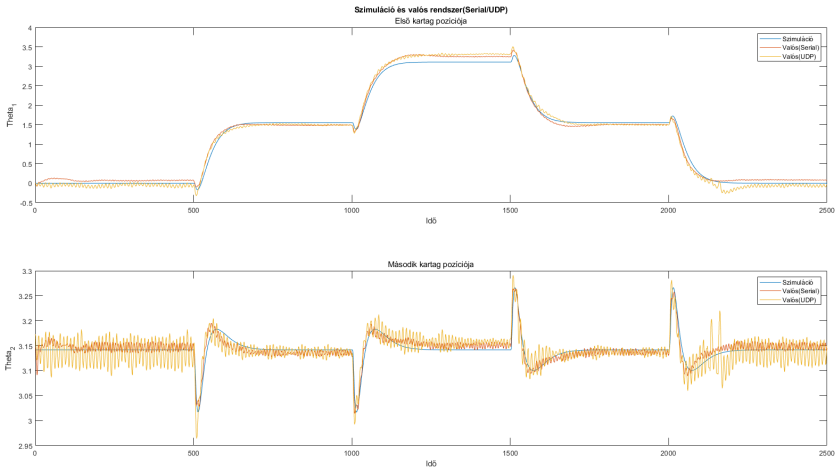
A szabályozás során a használt referencia érték azonos a szimulációnál bemutatottal így könnyen összehasonlítható a két eredmény.



8-22. ábra. Szimuláció összehasonlítása a valós rendszerrel

Látható, hogy az ingát a szabályzás sikeresen képes stabilizálni, valamint az is, hogy a valós rendszer mozgása követi a szimuláció alapján várt viselkedést.

Következő mérés során a második kartag pozícióját az Arduino soros kommunikáció útján az M5Stick-C-nek továbbítja, innen pedig UDP csomagok formájában továbbítódnak a PC felé. A mérés során a számítógép és a mikrovezérlő is WiFi-n keresztül csatlakozik a hálózathoz.



8-23. ábra. Kommunikációs megoldások összehasonlítása

Mint látható a vezeték nélküli kommunikáció nagyobb kitéréseket eredményez, de az inga felső állapotba tartásához elegendően gyors.

9. Konkluzió

Diplomamunkám célkitűzése az volt, hogy megtervezzek és megépítsek egy Furuta ingát, melynek segítségével a későbbiekben egy szenzor fúziós könyvtár működését fogom tesztelni. Kiemelt szempont volt a tervezés során, hogy az inga állapotát többféle szenzor által legyünk képesek méri. A rendszer működőképességét a rajta végzett szabályozással kívántam ellenőrizni, ami az inga fordítót pozícióban való megtatására irányult.

A szabályozás tervezés alapját a rendszer matematikai modellje nyújtotta, melynek levezetését az Euler-Lagrange egyenlet segítségével mutattam be a rendszer geometriája alapján. A mozgásegyenletet az inga fordított egyensúlyi állapota körül linearizálva kaptam meg a szabályozás tervezéséhez szükséges modellt.

Áttekintésre kerültek a lehetséges eszközök melyek képesek mérni a rendszer állapotát. Ezek alapján az inkrementális enkóder, a szögsebesség, a gyorsulás és a szöghelyzet mérésére alkalmas IMU lett kiválasztva.

Mivel a Furuta inga forgó mozgásai megnehezítik a rajta elhelyezett ezközök közötti vezetékes kommunikációt, ezért a rendszert úgy tervezem meg hogy lehetőség legyen a szenzor adatokat vezeték nélküli kommunikáció segítségével begyűjteni. Ezen ismeretek alapján megépítésre került a tényleges rendszer.

A rendszer paramétereit mérési úton a Matlab System Identification Toolbox segítségével határoztam meg. Az így megkapott paraméterek segítségével előált modell alapján valósult meg a szabályzó tervezése. Az inga szabályozását teljes állapot visszacsatolással valósítottam meg. Az állapot visszacsatolást LQ szabályozás segítségével határoztam meg. A megtervezett szabályzó és a modell alapján szimulációt készítettem, hogy lássam a rendszer várható viselkedését.

Végezetül a megtervezett szabályozást a valós rendszeren is teszteltem. A rendszer viselkedését összehasonlítva a szimulációk eredményeivel az tapasztaltam, hogy a viselkedése a szimulációk alapján vártaknak megfelelő. A rendszer működését vezetékes és vezeték nélküli kommunikáció használatával is teszteltem. Ebből az látható, hogy vezetékes kommunikáció hatására a rendszer jobban közelít az elvárt viselkedéshez, azonban vezeték nélküli kommunikációval is megvalósítható az inga fordított pozícióban tartása.

9.1. További tervek

Most, hogy megépült a rendszer, valamint tesztelve lett a struktúra működése, készen áll a környezet a szenzorfúziós könyvtár tesztelésére. A továbbiakban a rendszer szabályozását a könyvtár felhasználásával kapott állapotbecslés alapján szeretném megvalósítani. A tesztek során szeretném megvizsgálni, a különböző szenzoradatok felhasználásával végzett állapotbecslés alapján végzett szabályzás pontosságát.

A további kísérletek során a szabályzást végző programot valós idejű Linux operációs rendszeren szeretném futtatni. A valós idejű operációs rendszerek nagy pontosságú ütemezést és célfeladatok gyorsabb futtatását teszik lehetővé, nem úgy mint az álltalános felhasználású operációs rendszerek. Ennek köszönhetően lehetővé válhatna a rendszer szabályozása magasabb frekvencián. A programot oly módon szeretném tovább fejleszteni, hogy minél jobban kihasználja a valós idejű operációs rendszer nyújtotta előnyöket, felépítését pedig úgy bővíteni és/vagy átalakítani, hogy könnyen kialakíthatóak legyenek különböző szenzor értékek alapján megvalósított szabályzási esetek. Továbbá szeretném implementálni az inga fellendítését végző szabályozást is.

Hivatkozások

[1] W. Elmenreich, „An introduction to sensor fusion," *Vienna University of Technology, Austria*, vol. 502, pp. 1–28, 2002.

[2] K. Furuta, M. Yamakita, and S. Kobayashi, „Swing up control of inverted pendulum," in *IECON*, vol. 91, pp. 2193–2198, 1991.

[3] B. S. Cazzolato and Z. Prime, „On the dynamics of the furuta pendulum," *Journal of Control Science and Engineering*, vol. 2011, 2011.

[4] J. E. Marsden and J. Scheurle, „The reduced euler-lagrange equations," 1993.

[5] D. Gruber, „The mathematics of the 3d rotation matrix," in *Xtreme Game Developers Conference*, pp. 1–14, 2000.

[6] T. Atanacković, S. Konjik, and S. Pilipović, „Variational problems with fractional derivatives: Euler–lagrange equations," *Journal of Physics A: Mathematical and Theoretical*, vol. 41, no. 9, p. 095201, 2008.

[7] D. S. Z. Dr. Bokor József, Dr. Gáspár Péter, „Irányításelmélet," 2014.

[8] H. SOMOGYI and A. SOUMELIDIS, „Inerciális érzékelők tesztelése szimulációs környezetben és gnss/imu szenzorfúzióban való alkalmazhatóságának vizsgálata: Testing of inertial sensors in a simulation environment and investigation of the applicability in gnss/imu sensor fusion," *Nemzetközi Gépészeti Konferencia–OGÉT*, pp. 342–345, 2020.

[9] A. Szántó, G. Á. Szíki, S. Hajdu, and A. Gábora, „Soros gerjesztésű egyenáramú motor szimulációja matlab környezetben," 2017.

[10] S. István and H. Barnabás, „Mechatronikai rendszerek speciális érzékelői és aktuátorai," *Pannon Egyetem, Keszthely*, 2014.

[11] y. Dr. Kránicz Balázs Dr. Halas János, „Mikrovezérlők mechatronikai alkalmazásai,"

[12] A. S. Tanenbaum, C. András, and C. Andrásné, *Számítógép-hálózatok*. Novotrade, 1992.

[13] „Epos4 50/5 adatlap." Elérhető: https://www.maxongroup.com/medias/sys_master/root/8834325250078/EPOS4-50-5-Hardware-Reference-En.pdf (2021/05/10).

[14] „Lrs-150 adatlap.” Elérhető: https://www.meanwell.com/Upload/PDF/LRS-150/LRS-150-SPEC.PDF (2021/05/10).

[15] „Heds-9040/9140 adatlap.” Elérhető: http://www.farnell.com/datasheets/1816988.pdf (2021/05/10).

[16] „Arduino nano adatlap.” Elérhető: http://www.farnell.com/datasheets/1682238.pdf (2021/05/10).

[17] „M5stickc esp32-pico mini iot development board adatlap.” Elérhető: https://m5stack.hackster.io/products/m5stickc-esp32-pico-mini-iot-development-board (2021/05/10).

[18] L. Ljung, „System identification toolbox,” *The Matlab user's guide*, 1988.

[19] L.-C. Hsu and F.-R. Chang, „The generalized ackermann's formula for singular systems,” *Systems & control letters*, vol. 27, no. 2, pp. 117–123, 1996.

[20] L. Dávid, K. György, and A. Kelemen, „Modell alapú prediktív irányítási algoritmus, állapotfüggő riccati egyenlet illetve véges horizontú dlqr algoritmusok összehasonlítása,” 2015.

[21] V. Kučera, „The discrete riccati equation of optimal control,” *Kybernetika*, vol. 8, no. 5, pp. 430–447, 1972.

[22] I. Fantoni and R. Lozano, „Stabilization of the furuta pendulum around its homoclinic orbit,” *IFAC Proceedings Volumes*, vol. 34, no. 6, pp. 807–812, 2001.

[23] S. Mori, H. Nishihara, and K. Furuta, „Control of unstable mechanical system control of pendulum,” *International Journal of Control*, vol. 23, no. 5, pp. 673–692, 1976.

[24] J. Acosta, „Furuta's pendulum: A conservative nonlinear model for theory and practise,” *Mathematical Problems in Engineering*, vol. 2010, 2010.

[25] A. Szántó, A. Szántó, and G. Á. Sziki, „Soros gerjesztésű egyenáramú motor modellezési eljárásainak áttekintése,” 2020.

[26] Y.-y. Fang and X.-j. Chen, „Design and simulation of uart serial communication module based on vhdl,” in *2011 3rd International Workshop on Intelligent Systems and Applications*, pp. 1–4, IEEE, 2011.

[27] O. Garćıa-Alarćon, S. Puga-Guzan, and J. Moreno-Valenzuela, „On parameter identification of the furuta pendulum," *Procedia Engineering*, vol. 35, pp. 77–84, 2012.

[28] K. F. D. A. H. J. D. V. J. D. L. B. Dr. Korondi Péter, Décsei-Paróczi Annamária, „Robotirányítások," 2014.

[29] E. V. Kumar and J. Jerome, „Robust lqr controller design for stabilizing and trajectory tracking of inverted pendulum," *Procedia Engineering*, vol. 64, pp. 169–178, 2013.

[30] L. Ljung, „Design and implementation of a furuta pendulum device for benchmarking non-linear control methods," *The Matlab user's guide*, 1988.

Ábrajegyzék