

Egyszabadságfokú mechanikai rendszer irányítása – nyílt hurkú vezérlés

A gyakorlat célja

Egyenáramú szervó motorral vezérelt egyszabadságfokú mechanikai rendszer meghajtó áramkörének és a NATIONAL INSTRUMENTS USB-6001 adatbegyűjtő kártyának programozásának megismerése, a vezérlőszoftver törzsének megvalósítása, a motor nyílt hurkú vezérlőprogramjának megírása.

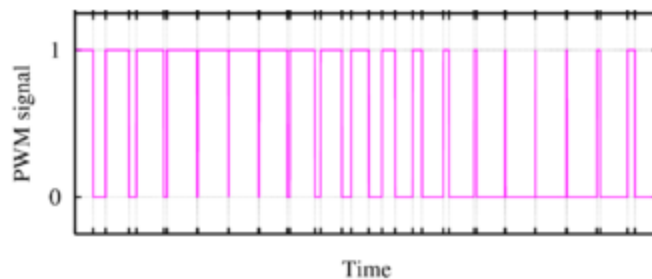
Elméleti bevezető

Robotikai rendszerek esetén a mechanikai rész mozgatása szervomotorral történik. A szervomotort elfordulás érékelővel szereljük fel (a szögelfordulás és szögpozíció mérésére). A robot szegmensét áttételen keresztül illesztjük a szervomotorhoz. A szervomotor az érékelővel és az áttétellel felelősek a robot csuklójának mozgatásáért. Szervomotorok jellegzetessége, hogy állandósult állapotban a motor sebessége arányosan változik a bemeneti feszültséggel.

Nyílt hurkú motorvezérlés esetén két megvalósítandó feladat van:

- a motor forgássebességének változtatása
- a motor forgásirányának változtatása

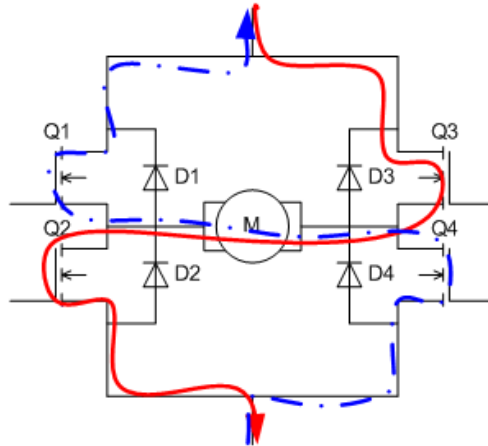
A szervomotorok forgási sebességét a motor feszültségének változtatásával vezérelhetjük. Ezt általában impulzus szélesség modulált jellel (PWM, Pulse Width Modulation) tesszük. Impulzus szélesség modulált jelek négyzetjelek, amelyekben az egyes négyzetimpulzusok szélességét változtatjuk (lásd 1. Ábra). Mivel a négyzetimpulzusok periódusa állandó a kitöltés változtatásával a jel átlagértéke egyenesen arányosan változik. Tehát a motor sebessége is egyenesen arányosan változik a kitöltéssel.



1 Ábra: PWM jel

Ha a PWM jel frekvenciája elégségesen nagy, akkor a rotor induktivitása miatt motor sebessége egyenletes lesz, nem fog változni a jel periódusa alatt. Ez abból adódik, hogy a

rotor inuktívitasából és ellenállásából álló szűrőáramkör a PWM jel nagyfrekvenciás komponenseit megszüri, az rotoráram állandósult állapotban egyenletes, a kitöltéssel arányos lesz. A PWM jeleket általában digitális jelként generáljuk, majd kapcsoló üzemű tranzisztorokkal felerősítjük.



2. Ábra: H híd

A motor forgásirányát a rotoron átfolyó áram irányával módosíthatjuk. Ha az áram iránya megváltozik a mozgás iránya is megváltozik. Ezt H hiddal oldjuk meg (lásd 2. Ábra). Ha a hídban vagy az Q1 és Q4 tranzisztor van nyitva vagy a Q2 és Q3 tranzisztor van nyitva. Ennek függvényében a motor forgásiránya megváltoztatható.

Motorvezérlő áramkör

A laboratóriumi gyakorlat során használt vezérlőáramkör kapcsolási rajza a 4. Ábrán látható.

A lapon található dsPIC30F2010 DSC integrált áramkör segítségével állítható elő a motorvezérléshez szükséges impulzus szélesség modulált jel. A mikrovezérlő egy analóg csatornán (CMD láb) beolvassa a kitöltés tényező értékét, majd egy beépített hardveres PWM jelgenerátor előállítja a megfelelő jelet.

A H híd vezérlésére LMD18200 áramkört alkalmazunk. Az integrált áramkört MOS tranzisztorokból kialakított H hidak vezérlésére alkalmazzák. Az áramkörnek 2 BOOTSTRAP kimenete van, amire kondenzátorokat kell kötni a megfelelő motorvezérlő kimenetekről. A DIRECTION bemenet megadja a forgás irányát. A BRAKE bemenettel rövidre zárjuk a motort, így a rendszer fékezni fog. A PWM bemeneten kell megadni a kitöltés tényezőt. Az integrált áramkör biztosít egy árammérő kimenetet is ($377\mu\text{A/A}$), illetve egy hőmérsékletjelző kimenetet, amelyik 170°C fölött kapcsol be.

A hídvezérlő IC biztosítja azt is, hogy a hídon ne keletkezzen rövidzárlat. Ugyanis előfordulhat, hogy a kapcsolás pillanatában egy nagyon rövid ideig a két azonos oldalon található tranzisztor nyitva lehet. Ennek a kivédésére a hídvezérlő áramkör egy nagyon rövid késleltetéssel iktat be a kikapcsolás és bekapcsolás között (tehát ha például a Q1

kikapcsol, akkor a Q2 csak egy előre meghatározott, mikro szekundumos nagyságrendű időintervallum után fog bekapcsolni)

Az irányt egy digitális jellel határozzuk meg (DIR_MOTOR). A PWM jel kombinálva a DIR_MOTOR jellel állíthatjuk egy időben a motor sebességét és irányát.

A hídhoz tartozik egy engedélyező láb ami szintén egy digitális jel. A hídvezérlő IC csak akkor aktív (küld jeleket a tranzisztorra), ha BRK lábán digitális 1 jelenik meg. Ez lesz az engedélyező bit (EN), amelynek ha az értéke 0 a motor nem forog

A szervomotorok vezérlésénél ugyancsak probléma a túláram megjelenése. Ha a motoron megnő a terhelés (például a motor rotorjára szerelt mechanika ütközik egy, a környezetében levő tárggyal) a motor árama megengedett érték felé nőhet, ami a motor károsodásához vezethet. Ezért a motor áramot szükséges mérni, és ha azt tapasztaljuk, hogy értéke nagyobb, mint egy előre meghatározott érték akkor a motorra kapcsolt feszültséget le kell venni, a vezérlést meg kell állítani. Motor áramát az R14 és R15 ellenállásokkal mérjük. Ezeken az ellenállásokon mindig átfolyik a motoráram függetlenül attól, hogy milyen irányba forog a motor. Az ellenállásokon megjelenő feszültség arányos lesz a motoráram abszolút értékével. A jelen áramkör esetében a motorárammal arányos feszültség (I_MAX_LIMIT) a PWM jel generáló IC működését tiltja le, ha az értéke nagyobb lesz egy előre meghatározható feszültség szintnél.

A motor szabályozását a számítógépen kell elvégezni, ezért egy interfészre lesz szükség ami a motor standot összeköti a számítógéppel. Ezt a célt szolgálja a NATIONAL INSTRUMENTS által gyártott USB-6001 kártya. Ezen adatbegyűjtő kártya fel van szerelve 8 14b-es analóg bemenettel, 2 14b-es analóg kimenettel, 13 digitális ki/bemenettel, és egy 32b-es számlálóval.



3. Ábra: NATIONAL INSTRUMENTS USB-6001

A kártya elérhető ANSIC, C#, .NET, VB.NET, LabVIEW, LabWindows/CVI, és MeasurementStudio környezetekből.

A mérés menete

A motorvezérlő C# alapú legyen, és teljesítse az alábbi feladatokat:

- Az alkalmazás indításakor inicializáljuk a kártyát és az ablakelemeket
- Az ciklikusan meghívható függvényben indítsuk el a motort a gy *EditBox* – ban megadott feszültséggel.
- Az alkalmazás bezárásakor állítsuk meg a motort.

A program sikeres működtetéséhez szükséges lesz alkalmazni a kártya meghajtó állományokat (*NationalInstruments.Common.dll*, *NationalInstruments.DAQmx.dll*), illetve egy időzítő állományt is (*MmTimer.dll*). Az első két állomány biztosítja az adatbegyűjtő kártyakezelő függvényeket, az utolsó állomány az alpertelmezett időzítő helyett nyújt pontosabb megoldást (*Multimedia Timer* – MIDI eldolgozás, audio/video kódolás, dekódolás).

A motor vezérléséhez az alábbi kimeneteket kell használni:

COMMANDVOLTAGE – A motor vezérlőfeszültsége - Analóg kimenet – 0-10 V
ENABLE – A motor engedélyezése – digitális kimenet - 1 aktív
DIRECTION – A motor forgásiránya – digitális kimenet

A kártya egyes funkcióit külön feladat biztosítással kell ellátni (tasks):

```
NationalInstruments.DAQmx.Task AOutTask = new NationalInstruments.DAQmx.Task();  
AnalogSingleChannelWriter setAOut;  
NationalInstruments.DAQmx.Task DOutTask = new NationalInstruments.DAQmx.Task();  
DigitalSingleChannelWriter setDOut;
```

A digitális kimenet tartalmazza mind az engedélyező bitet, mind az irány bitet.

Az *AOutTask.AOChannels.CreateVoltageChannel* metódus segítségével az Ao0 kimenetet unipoláris 0-10V-osnak kell beállítani.

Az *AnalogSingleChannelWriter* metódussal *setAOut*-on keresztül állíthatjuk be az analóg csatorna értékét:

Pl.: *setAOut.WriteSingleSample(true, 3.0);* -- 3V-ot ad ki az analóg csatornán

- A ciklikus időzítő függvény paraméterezése és elindítása:

.Mode – metódussal megmondjuk az időzítő típusát (egyszeri *TimerMode.OneShot*, vagy ciklikus *TimerMode.Periodic*)

.Resolution – megadjuk a felbontást milliszekundumban

.Period – megadjuk a periódust milliszekundumban

.Tick – eseményt rendelünk az időzítő szálunkhoz

Pl.: *timer1.Tick += new EventHandler(tickHandler);*
private void tickHandler (object sender, EventArgs e)

Az időzítő függvényben engedélzezzük a motorvezérlést beállítjuk a motor irányát (digitális kimenetek). Az EditTextból kiolvasott feszültségértéket kiküldjük az analóg kimeneten.

- EditTextból beolvasáshoz az alábbi függvényt alkalmazzuk:

```
string temp;  
getTextboxVal("Command_textBox", out temp);
```

Az értéket voltban olvassuk be, majd voltban küldjük ki, de lebegőpontos adatokkal dolgozunk, nem karakterláncokkal.

- A digitális kimenetek használata:

Az ENABLE jel a nullás bitnek, a DIRECTION jel az egyes bitnek felel meg. Az összesített jelet pedig az alábbi paranccsal küldjük ki:

```
setDOut.WriteSingleSamplePort(true, digitalVal);
```

- Az analóg jel kiküldése:

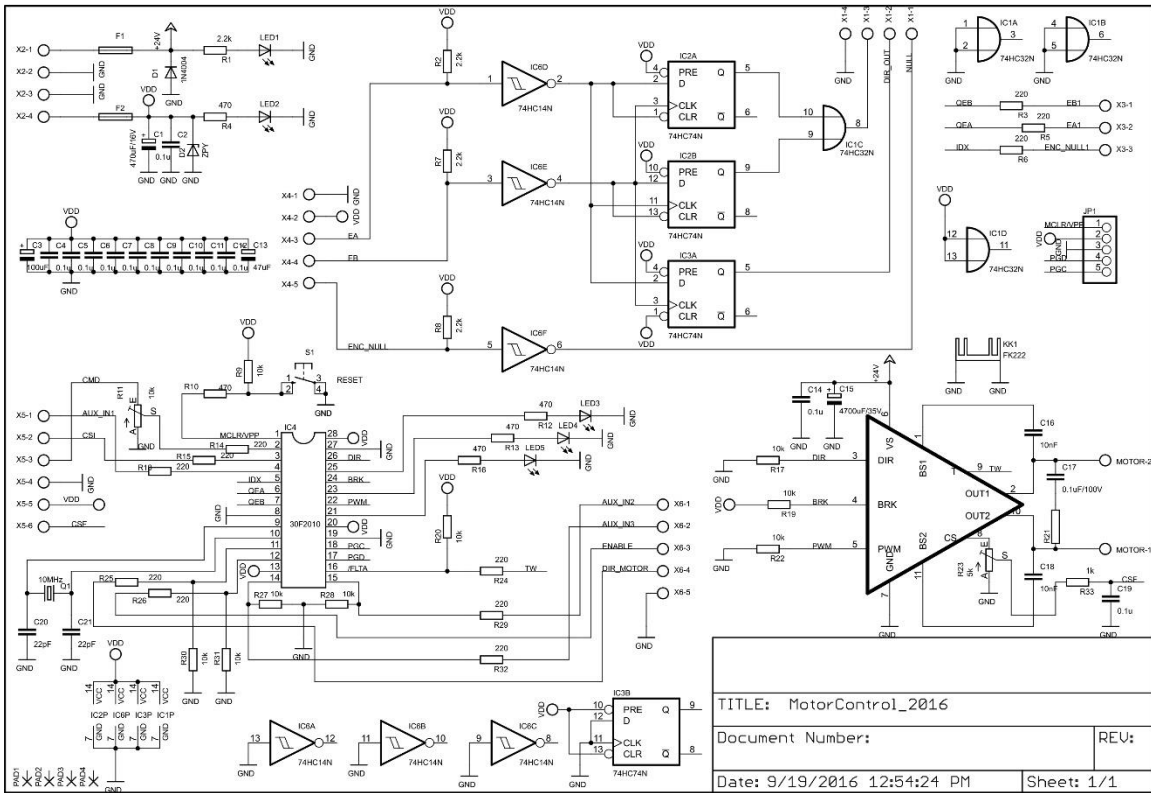
Az analóg jel pozitív kell legyen és nem szabad meghaladja az analóg kimenet beállításakor megadott határt.

```
setAOut.WriteSingleSample(true, analogVal);
```

Az alkalmazás bezárásakor az ENABLE kimenetre nullát, az analóg kimenetre szintén nullát küldünk.

Kérések, feladatok

1. Keressen más áramkörü megoldásokat a H híd lecserélésére
2. Módosítsa úgy a programot, hogy a motorra kiküldendő feszültséget egy csuszkaról olvassa le.
3. Alkalmazzon más Advantech függvényt az analóg jel kiküldéséhez.



4. Ábra: A vezérlőáramkör szematikus rajza