

Pozíció és sebességmérés robotirányítási rendszerekben

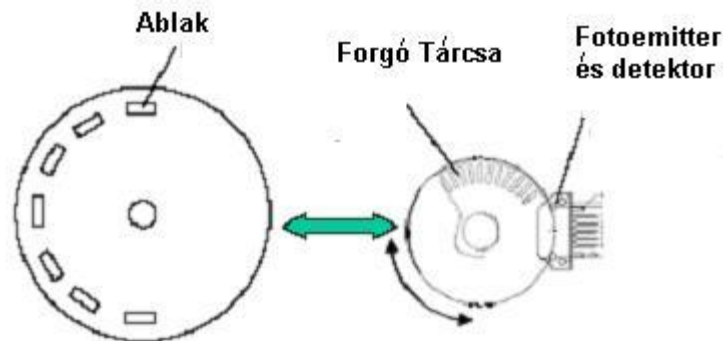
A gyakorlat célja

Inkrementális adóval felszerelt egy szabadságfokú mechanikai rendszer számítógépes interfészének megismerése, Az inkrementális adó jeleinek beolvasása számítógépbe, a jelek kalibrálása.

Elméleti bevezető

A robotikai rendszerekben a csuklók pozícióját és sebességét általában inkrementális adóval mérjük.

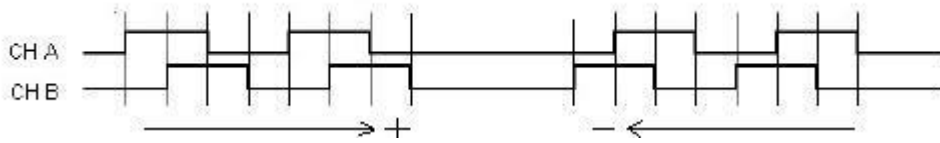
Az inkrementális adó szögelfordulás és szögsebesség mérésre szolgáló szenzorok. Az elfordulással arányos számú impulzust adnak a kimeneten. Az impulzusokat átlátszó ablakokat tartalmazó tárcsa és egy fotóemitter- fotódetektor pár segítségével lehet előállítani. Fotóemitterként LED –et fotódetektorként fotótranszisztort használhatunk. Ha forgó tárcsa egy ablaka kerül a fotóemitter és a fotódetektor közé a szenzor egy impulzust szolgáltat a kimenetén. (lásd 1. Ábra)



1. Ábra: Az inkrementális adó elvi működése

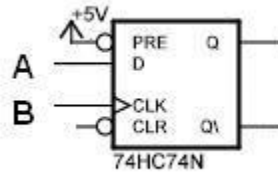
Ha a sebesség megváltozik az impulzussorozat frekvenciája nő meg. Így az inkrementális adó pozíciómérés impulzusszámlálásra, a sebességmérés frekvenciamérésre vezethető vissza.

Mivel a forgás irányát is célszerű meghatározni, két fotóemitter -fotódetektor párt használnak, amelyek egymáshoz képest el vannak tolva. Ennek megfelelően két, 90 fokkal eltolt impulzussorozatot kapunk egy helyett. (lásd 2. Ábra). Mind a két impulzussorozat mérhető az inkrementális adón.



2. Ábra: Az inkrementális adó kimeneti jelei

Az irány meghatározásához azt használhatjuk ki, hogy amikor az inkrementális adó tárcsája pozitív irányba forog, az *A* csatorna pozitív frontján a *B* csatorna értéke 1. Ha az inkrementális adó tárcsája negatív irányba forog, az *A* csatorna pozitív frontján a *B* csatorna értéke 0. Egy D bistabillal az irányt meghatározhatjuk. A CLK bemenetre az *A* csatornát a D bemenetre a *B* csatornát kötjük. Ha a CLK bemeneten pozitív front van, a D bemenetet kiírja a Q kimenetre. Tehát pozitív irány esetén a Q 0, negatív irány esetén a Q 1 lesz.



3. Ábra: D bistabil a forgásirány meghatározására

Sebesség mérése: Feltételezzük, hogy az inkrementális adó fizikai felbontása N , vagyis az inkrementális adó 2π radián alatt N impulzust ad (N az inkrementális adó felbontása). Egy fix, T időegység alatt beérkezett impulzusok számát határozzuk meg. Ha T idő alatt n impulzus érkezett be akkor a szögsebesség:

$$\omega_k = \frac{2\pi n}{NT} = \frac{2\pi f}{N} \left[\frac{\text{rad}}{\text{mp}} \right] \quad (1)$$

A szögpozíciót (α_k) a sebesség integráljaként kapjuk. A numerikus integráláshoz a trapéz módszert alkalmazzuk. A módszer rekurzív formája:

$$\alpha_k = \alpha_{k-1} + \frac{(\omega_k + \omega_{k-1})T}{2} \frac{\pi}{180} \text{ [fok]} \quad (2)$$

ω_k a sebességet jelöli a k -ik mintavételben, ω_{k-1} a sebesség a $k-1$ -ik mintavételben, T a mintavételi periódus. A $\pi/180$ -as értékkel azért szorzunk, mert a pozíciót fokkal szeretnénk kiírni, számolni.

A gyakorlat menete

A *MotorControl* C# tervben az alábbi feladatokat kell elvégezni:

- Az NI USB-6001 kártya számláló bemenetének konfigurálása, a jelek periodikus beolvasása
- Sebesség kiszámítása a beolvasott frekvenciaérték alapján
- Pozíció kiszámítása a kiszámított sebességérték alapján

Impulzusok beolvasása: Az alkalmazott adatbegyűjtő kártya számlálóval és időzítővel van felszerelve. Ezen számláló a bemenetre kapcsolt jel fel- vagy lefutó éleire növeli a regiszter értékét.

```
NationalInstruments.DAQmx.Task CInTask = new NationalInstruments.DAQmx.Task();  
CounterSingleChannelReader getCIn;
```

```
CInTask.CIChannels.CreateCountEdgesChannel("Dev0/ctr0", "CIn",  
CICountEdgesActiveEdge.Rising, 0, CICountEdgesCountDirection.Up);  
getCIn = new CounterSingleChannelReader(CInTask.Stream);  
CInTask.Start();
```

A frekvencia beolvasását periódikusan, az időzítő esemény metódusban végezzük. Ennek használatához az alábbi függvényt alkalmazzuk.

```
UInt32 Counter_In = getCIn.ReadSingleSampleUInt32();
```

Mivel nekünk frekvencia kell, ezért két beolvasás között eltelt ütemre vagyunk kíváncsiak (figyelembe kell venni a pozitív 32 bites túlcsoordulást). Ha tudjuk az időzítő periódusát, és az események között beérkezett jel élek számát, a frekvenciát egyszerű osztással kapjuk meg.

A *sebesség méréséhez* először az irány bitet kell beolvasni. Ez egy digitális bement 0-ik bitjén. A digitális beolvasásra alkalmazott függvény:

```
Int32 DigitalIn_Val = getDIn.ReadSingleSamplePortByte();  
Int32 Dir_Encoder = DigitalIn_Val & 0x01;
```

Az iránybit segítségével meghatározzuk a sebesség előjelét. Ha nulla, akkor az irány negatív. A sebesség értékét az (1) összefüggéssel számíthatjuk. Az alkalmazott enkóder felbontása $N=500$.

A pozíció értékét a (2) összefüggéssel számíthatjuk. Látszik, hogy a pozíció értéke függ az *időzítő* metódus T periódusától, amit meg tudunk mérni (a Windows pontossága miatt az 50ms nem mindig 50ms).

```

mmTimer_Tick
{
double T = 0;
DateTime t_k = DateTime.Now;
if (t_k_1 != DateTime.MinValue)
{
    TimeSpan period = t_k - t_k_1;
    T = period.TotalSeconds;
}
t_k_1 = t_k; .....
}

```

A mért és számított értékeket (T periódus, beolvasott frekvencia, pozíció, sebesség) a `setTextboxVal("Sampling_Period_textBox", T.ToString());` függvénnyel írhatjuk ki. Mivel az *időzítő* periódusa ezredmásodperc nagyságrendű és a kiírt értékek gyorsan változnak, a folytonos kijelzés nem lesz élvezhető az kiírt értékek vibrálni fognak a képernyőn. Ezért a kiírt értékeket csak minden $K=10$ -ik mintavételben írjuk ki. Ezt egy számláló segítségével oldhatjuk meg. A számláló értékét folyamatosan növeljük, és ha elérte a 10-et, akkor végrehajtjuk a kiírást, és az értékét nullázzuk.

Kérdések, feladatok

1. Oldja meg más Advantech függvénnyel (csak a számláló értékének kiolvasásával) a frekvenciamérést, ismerve az *OnTimer* periódusát.
2. Mennyi a pozíciómérés pontossága?
3. Mennyi a sebességmérés pontossága?