

Témalaboratórium - BMEV8IAL02

Házi feladat dokumentáció

Csizy Ádám, Sipos Miklós

Feladat kiírás

Adott egy ágyúhoz hasonló kilövő szerkezet, melynek talajjal bezárt szögét egy motor állítja. A feladatban meg kell oldani a talajjal bezárt szög automatikus beállítását (a távolságtól függően). A szerkezet "elsütését" kézi beavatkozás indítja, az ágyú töltése is kézzel történik. A szabályozó programját az Arduino egységen valósítsa meg. A LabVIEW kezelőfelület segítségével lehessen manuálisan állítani a szöget, automatikus esetben pedig jelenítse meg a beavatkozó jelet.

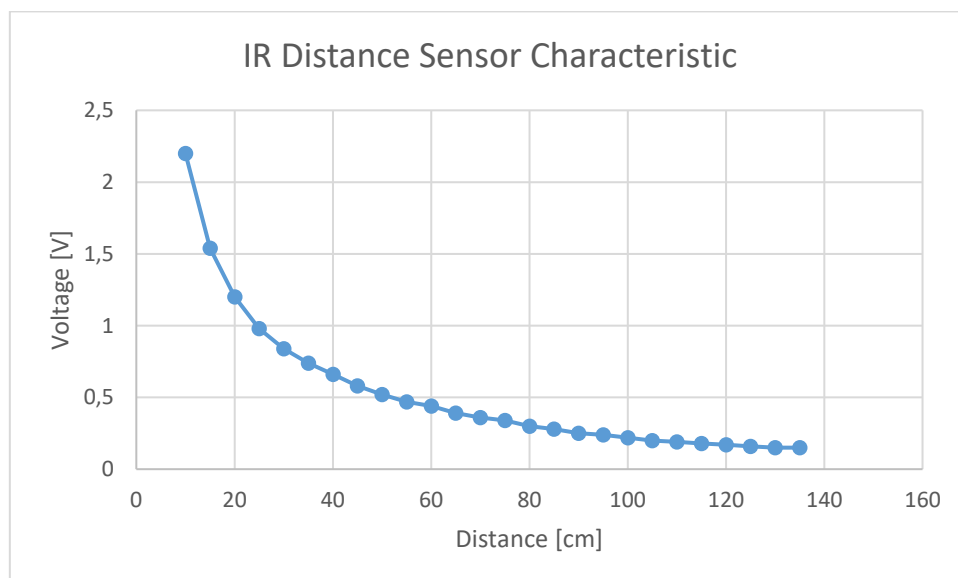
Használt fejlesztőkörnyezetek: Arduino IDE, LabVIEW

Használt fejlesztőkártya: Arduino MEGA + IIT-MEGA shield

Használt modulok, eszközök: motor illesztő shield (1 db), Sharp GP2Y0A60SZ0F IR távolságmérő (1 db), shield hosszabbító (1db), kilövő pozícionáló (1 db), USB A-B kábel (1 db), hálózati adapter (1 db)

Alkalmazott megoldások

A lövedék sikeres célba juttatásához nélkülözhetetlen a céltárgy ágyútól való pontos távolságának ismerete. Az infravörös szenzor kimenetén mérhető analóg feszültségértékekből analóg-digitális átalakító és nemlineáris feszültség-távolság karakterisztika segítségével a távolság megkapható. Mivel a szenzor gyártója által közzé tett karakterisztika a tesztelések során pontatlannak bizonyult, így egy saját, általunk kimért karakterisztika került felhasználásra lineáris interpoláció alkalmazásával.



A grafikonról jól leolvasható, hogy a 85 – 90 cm feletti távolságokhoz tartozó feszültségértékek esetén a parabola konvergál egy adott feszültség értékhez. Ezen tartományon a kvantálási zaj és a távolságegységek feszültségváltozása azonos nagyságrendbe esik, így a távolság pontos

meghatározása erősen bizonytalanná válik. Esetünkben az alkalmazott lövedék maximális hatótávolsága (~ 90,2 cm) miatt a torzítás hatása csak korlátozottan jelentkezik.

A távolság (IR szenzor), nehézségi gyorsulás (9.8 m/s^2) és a lövedék kezdősebessége ($2,9732 \text{ m/s}$) ismeretében az ágyú vízszintes talajjal bezárt szöge a ferde hajítás képletéből kiszámítható. Az így kapott szög érték képezi a PI szabályozó számára a referencia jelet automata üzemmód esetén. Az ágyú aktuális szögének detektálása inkrementális enkóder jeladókkal történik megszakításkezeléssel. Mivel az ágyú aktuális szögét tároló változó kezdeti értéke zérus, ezért a rendszer indítása csak az ágyú vízszintes helyzete mellett tehető meg. Az ágyút mozgató motor, mint beavatkozó szerv vezérlése H-híd és impulzus szélesség modulációval (8 bites felbontással) történik.

A feladat megoldása során egy PI szabályozó szoftveres implementálására került sor, Arduino környezetben C programozási nyelv felhasználásával. A hibajel a referencia szög és az ágyú aktuális szögének különbségeként áll elő, ez a szabályozót megvalósító függvény bemeneti paramétere. A szabályozó implementálása során figyelembe kell venni a beavatkozó szerv véges képességeit, így esetünkben a beavatkozó jelet (PWM jel kitöltési tényezője) korlátok közé szorítjuk (szaturáljuk), az integrátor értékét pedig nem növeljük tovább az új értékkel, elkerülve ezzel az elintegrálódás kedvezőtlen hatásait. Az integrátor alkalmazásának hasznossága a tesztelések során akkor mutatkozik meg, amikor az arányos tag által előállított beavatkozó jel nagysága az alacsony hibajel és a kilövő pozícionáló elemeinek súrlódása, tehetetlensége miatt nem elegendő az ágyú elforgatásához. Ekkor az integrátor értékének fokozatos növelésével a beavatkozó jel egy idő után elég nagyra nő ahhoz, hogy a villanymotor nyomatéka legyőzze a mechanikai ellenállást, így a referencia szög apró léptetésekkel elérhető, így garantálva a közel zérus maradandó hibát. A szabályozónál használt konstans együtthatók értéke a szabályozó hangolása során került meghatározásra.

Az automata irányzék üzemmódjainak működése állapotgép alapú. Az állapotok közötti váltás a LabVIEW grafikus felületén keresztül lehetséges a felhasználó által. A három üzemmód: automata, manuális és zárolt. Automata üzemmódban az ágyú talajjal bezárt szögének állítása a már említett PI szabályozó segítségével történik. A szabályozást megvalósító, a beavatkozó szerveket működtető és a rendszer bemeneti jeleit (szögelfordulás, távolság) feldolgozó függvények meghívása meghatározott sorrendben, periodikusan történik egy végtelen ciklusban, megvalósítva ezzel az autonóm működést. A célpont hatótávolságon kívül kerülését a motor vezérlés letiltása és egy piros állapotjelző LED kigyulladás jelzi a felhasználó számára. Manuális üzemmódban a felhasználó által a LabVIEW felületen megadott, soros kommunikációs porton keresztül továbbított szög képezi a referencia szöget, az ágyú vezérlése is ezen értéknek megfelelően zajlik. Zárolt üzemmódban az ágyú irányzását végző motor letiltásra kerül, továbbá egy zöld állapotjelző LED is kigyullad, jelezve ezzel a felhasználónak, hogy az ágyú töltése és elsütése biztonságosan megkezdhető.

A szabályozó mechanizmussal és az állapotgéppel a LabVIEW grafikus felhasználói felületen keresztül léphetünk interakcióba. A használt kommunikációs port egy legördülő menüsorból választható ki, a sikeres kapcsolódást az állapotjelző csík zöld színre váltása jelzi. Az üzemmódok közötti váltás a panel megfelelő gombjainak egyszeri megnyomásával lehetséges. Manuális üzemmód esetén a referencia szög egy csúszka segítségével állítható. A fentebb leírt interfészek a blokk diagramban egy sorosan kapcsolódó, a beavatkozó gombok igaz-hamis állapotától (megnyomásától) függő switch struktúrából állnak, melyek interakció során a kiválasztott állapotnak megfelelő és szükséges adatokat (referencia szög manuális üzemmód esetén) tartalmazó karaktersort küldenek ki a soros porton VISA blokkok felhasználásával. Ezen karaktersorok dekódolása (állapot kód és referencia szög értékének kiolvasása) Arduino oldalon történik. Az Arduino felől érkező beavatkozó jel folyamatos kiolvasása ciklikusan (20 ns) történik, az adatvizualizáció a LabVIEW chart elemének segítségével került megvalósításra.

A feladat megoldásához felhasznált források

- A tanszéki weboldalon fellelhető segédanyagok
(<https://www.iit.bme.hu/targyak/BMEVIAL02/modulok>)
- Az Arduino hivatalos weboldalán közzétett programnyelvi referenciák
(<https://www.arduino.cc/reference/en/>)
- Az Arduino hivatalos weboldalán közzétett mikroprocesszor lábkiosztás referencia
(<https://www.arduino.cc/en/Hacking/PinMapping2560>)
- A National Instruments weboldalán közzétett LabVIEW VI referenciák
(<https://zone.ni.com/reference/en-XX/help/371361R-01/TOC120.htm>)

A dokumentáció fejlécében szereplő hallgatók kijelentik, hogy a kiadott feladatot meg nem engedett segítség nélkül, saját maguk oldották meg, az elkészített forráskód valamint a hozzá tartozó dokumentáció a saját szellemi termékük.