

# **Vodenje laboratorijske naprave z regulatorjem stanj na osnovi opazovalnika**

**Beti Fajdiga in Katja Pretnar**

**Mentorja: izr. prof. dr. S. Blažič in izr. prof. dr. G. Klančar**

**Univerza v Ljubljani, Fakulteta za elektrotehniko**

**Tržaška 25, 1000 Ljubljana**

**E-mail: beti.fajdiga@gmail.com; ajtak.pretnar@gmail.com**

## ***The control of a laboratory pilot plant using observer based state feedback***

This paper discusses state – space methods for designing a control system. We present an example of controlling a laboratory pilot plant with a spinning wheel by determining its velocity or its angle. A basic control – law is used and upgraded with introduction of the reference input and with an integral behavior. The regulator is always combined with an observer. This solves the problem of an optimal tracking without the error in steady state. Control system is practically implemented in Simulink using the NI-PCI 6014 interface and on microcontroller system Arduino Uno.

## ***Kratek pregled prispevka***

V članku se bomo posvetili vodenju v prostoru stanj. Načrtali bomo tako hitrostno kot tudi pozicijsko vodenje laboratorijske naprave z vrtečim obročem. Uporabili bomo osnovni regulator stanj z ustreznimi nadgradnjami (predkrmilno ojačenje reference, integrirni značaj), kar zagotavlja sledilno delovanje brez pogreška v ustaljenem stanju. Regulator bomo vedno dopolnili še z opazovalnikom stanj. Praktična izvedba bo vključevala vodenje v okolju Simulink preko vmesnika NI-PCI 6014 ter regulacijo s pomočjo mikrokrmilnika Arduino Uno.

## 1 Uvod

Zapis sistema v prostoru stanj podaja enačba (1). Regulator stanj izračunava regulirno veličino  $u(k)$  kot linearno kombinacijo vseh stanj (2).

$$x(k+1) = Ax(k) + bu(k) \quad (1)$$

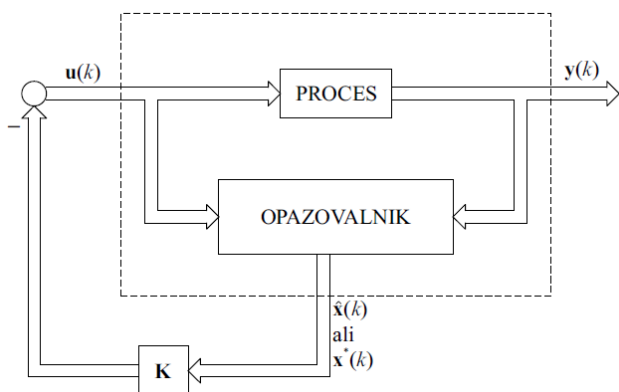
$$u(k) = -Kx(k) \quad (2)$$

Ob združitvi enačb (1) in (2) dobimo sistem (3), kjer opazimo, da s primernim izborom uteži  $K$  posameznih stanj lahko dosežemo ustrezno dinamiko povratnozančnega sistema.

$$x(k+1) = (A - bK)x(k) \quad (3)$$

Slabost regulatorja stanj je zahteva po informaciji o vseh stanjih sistema, ki pa jih običajno ne merimo. Te podatke je mogoče rekonstruirati iz merjenih izhodov s pomočjo opazovalnika. Luenbergerjev observator ocenjuje stanje na podlagi vhodnih in izhodnih signalov ter s tem vpliva na dinamiko spreminjanja pogreška ocene stanj  $\tilde{x}(k)$  – enačba (4). Problem iskanja vektorja uteži  $h$  je dualen določevanju  $K$  pri regulatorju stanj – enačbi (3) in (4).

$$\tilde{x}(k+1) = (A - hc)\tilde{x}(k) \quad (4)$$



Slika 1. Modelna shema vodenja z regulatorjem stanj na osnovi opazovalnika.

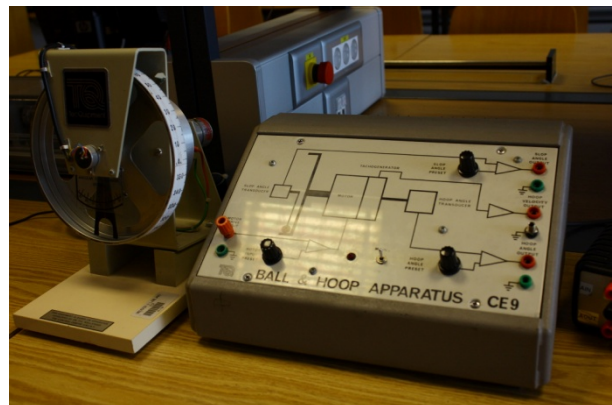
S principom ločitve lahko pokažemo, da je kombinirana uporaba regulatorja in observatorja (slika 1) povsem upravičena. Z neodvisnim načrtovanjem regulatorja in opazovalnika se poli sistema ob združitvi ne spremenijo.

## 2 Predstavitev uporabljenih naprav

Pri eksperimentu smo vodili laboratorijsko napravo z obročem. Vodenje smo izvedli najprej s pomočjo okolja Matlab – Simulink in NI-PCI 6014-vmesnika, nato pa smo to implementirali še na razvojni plošči Arduino Uno.

### 2.1 Laboratorijska naprava

Eksperiment je potekal na laboratorijski napravi, ki jo prikazuje slika 2. Naprava je sestavljena iz vrtečega se obroča in ločene čelne plošče z vhodno – izhodnimi priključki. Vhodna veličina procesa je napetost vzbujanja enosmernega motorja  $u$ , ki je preko osi povezan z obročem ter povzroča momentno vzbujanje  $T$ , s katerim delujemo na obroč. Merimo lahko kotno hitrost vrtenja obroča oz. motorja  $\omega$  in kotni položaj obroča  $\alpha$ . Napetostno območje je od 0V do 10V.



Slika 2. Laboratorijska naprava z vrtečim se obročem in čelno ploščo.

### 2.2 NI-PCI 6014-vmesnik

Napravo lahko enostavno vodimo z uporabo NI-PCI 6014 (A/D in D/A) vmesnika. Vmesnik vsebuje številne izhodne pine, s katerimi uporabnik lahko generira poljubne krmilne signale ter vhodne pine, ki omogočajo zajem procesnih veličin. Nastavljanje vmesnika je mogoče v okolju Simulink preko vključitve ustreznega bloka.

### 2.3 Arduino Uno

Arduino Uno je univerzalna razvojna tiskanina z ATmega328 mikrokrmilnikom. Zaradi različnih napetostnih območij naprave in mikrokrmilnika so dodana ustrezna vhodna in izhodna prilagoditvena vezja. Na vhode

mikrokrmilnika lahko priključimo signale v območju 0-5 V, zato je potrebno dodati vhodni napetostni delilnik. Izhod krmilimo s pomočjo PWM-signala frekvenca 33 kHz, ki ga filtriramo z RC-členom in dvakrat ojačimo z operacijskim ojačevalnikom. Lomna frekvenca filtra prvega reda (5) mora biti primerno izbrana, da izluščimo ustrezen zvezni signal.

$$f_{FILT} = \frac{1}{2\pi RC} \quad (5)$$

V našem primeru uporabimo vrednosti  $R=10\text{ k}\Omega$ ,  $C=100\text{ nF}$ , kar pomeni nemoteno opazovanje procesov s frekvenčnim pasom nekje do 100 Hz. Kadar uporabljamo napetostne izhode, moramo razvojni ploščici poleg standardne USB-povezave dodati še napajanje. Tiskanina s prilagoditvenimi vezji je zapakirana v ohišju, kot je prikazano na sliki 3.



Slika 3. Razvojna plošča Arduino Uno z vhodno – izhodnimi prilagoditvenimi vezji v ohišju.

Programsko kodo pišemo znotraj priloženega programskega paketa. Za periodično vzorčenje procesnih signalov in generiranje PWM-pulzov uporabimo dva časovnika. Branje analognih vrednosti preko A/D-pretvorbe in nastavljanje širine izhodnih PWM-pulzov je možno s preprostimi ukazi.

### 3 Modeliranje laboratorijske naprave

Za napravo želimo načrtati sistem vodenja, zato potrebujemo informacijo o njeni dinamiki. Prvi korak je določitev ustreznega modela naprave v obliki prenosne funkcije. Glede na poznavanje fizikalnih zakonitosti in oblike odziva lahko ocenimo, da gre v primeru hitrostnega vodenja za napravo prvega reda, v primeru pozicijskega vodenja pa uporabimo integrirni model drugega reda. Ker bomo

napravo vodili z regulatorjem stanj, prenosno funkcijo enostavno transformiramo v zapis v prostoru stanj.

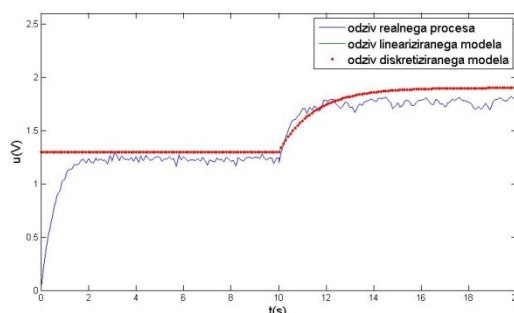
#### 3.1 Ocena modela pri hitrostnem vodenju

V primeru vodenja naprave preko določanja hitrosti vrtenja obroča dobimo enostaven proporcionalen model prvega reda, ki ga opišemo z uporabo stopničnega preizkusa. Za izbrano delovno točko  $(U_{DT}, Y_{DT}) = (2V, 1,3V)$  je lineariziran model podan z enačbo (6).

$$G(s) = \frac{1,2}{1,5s+1} \quad (6)$$

Ker bomo napravo vodili digitalno, je smiselno določiti še diskretiziran model. Kot najprimernejšo metodo diskretizacije smo izbrali metodo zadrževalnika ničtega reda s časom vzorčenja  $T_{VZ}=0,05\text{ s}$  oziroma s frekvenco  $f_{VZ}=20\text{ Hz}$ . Ker uporabljamo zadosti zmogljivo opremo, lahko čas vzorčenja izberemo poljubno majhen. Po Shannonovem teoremu mora biti frekvenca vzorčenja vsaj dvakrat večja od najvišje frekvenca signalov v sistemu, najoptimalnejša pa je 10- do 30-krat večja od najvišje frekvenca signala sistema.

Dobljena linearna modela (zveznega in diskretnega) ustrezno ovrednotimo; odziv izhodnih signalov na vzbujanje s stopnico prikazuje slika 4.



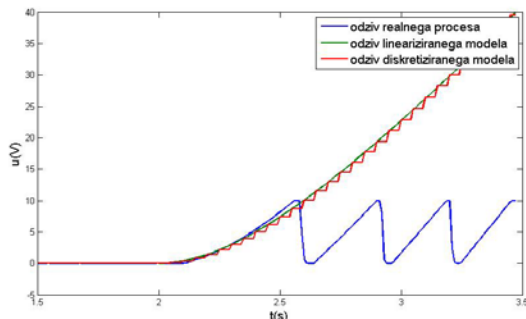
Slika 4. Primerjava odzivov realnega procesa in dveh lineariziranih modelov pri isti stopničasti spremembi hitrosti.

#### 3.2 Ocena modela pri pozicijskem vodenju

Laboratorijsko napravo lahko vodimo tudi preko želene pozicije oz. kota. Ker vemo, da je kot pravzaprav integral kotne hitrosti, pričakujemo model integrirnega značaja. Ocenimo ga z ustreznim pristopom za

modeliranje  $I_n$  sistemov. Dobimo lineariziran model (7).

$$G(s) = \frac{40}{s(0,5s+1)} = \frac{40}{0,5s^2+s} \quad (7)$$



Slika 5. Primerjava izhoda senzorja kota z odzivom zveznega in diskretiziranega modela pri enakem vzburjanju.

Posebnost naprave pri pozicijskem vodenju je izhodni žagast signal pri konstantnem vzburjanju (stopnici), ki ima pri vsakem polnem obratu nezvezen preskok. Zaradi tega je problematično vodenje v okolici kotov  $0^\circ$  ali  $360^\circ$ . Poleg tega je potrebno upoštevati tudi mrtvo cono naprave. Ovrednotenje linearnih modelov je prikazano na sliki 5. Opazimo, da modela ne vključujeta cikličnosti na območju od  $0^\circ$  do  $360^\circ$ , temveč njun izhod pri konstantnem vzburjanju le narašča.

#### 4 Hitrostno vodenje

Zagotoviti želimo sledilno delovanje in izregulacijo motenj. Osnovni regulator (2) dopolnimo z dodatnim vhodom, ki ga množimo s predkrmilnim ojačenjem  $N$ , (8) in (9).

$$N = KN_x + N_u \quad (8)$$

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (9)$$

Če si želimo izogniti pogrešku v ustaljenem stanju, moramo shemo vodenja razširiti še z dodatnim stanjem, ki predstavlja vsoto pogreška (trenutne in pretekle vrednosti). To stanje predstavlja integrirni značaj, ki odpravi vpliv netočnega modela in morebitnih motenj. Iz zapisa razširjenega regulatorja (10), (11), (12) in vhoda v proces (13) lahko preko Matlab-ovega ukaza določimo ojačenje  $K_N$ .

$$x_N(k+1) = A_N x_N(k) + B_{Nu} u(k) + B_{Nr} r(k) \quad (10)$$

$$\begin{bmatrix} x(k+1) \\ x_e(k+1) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ x_e(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(k) \quad (11)$$

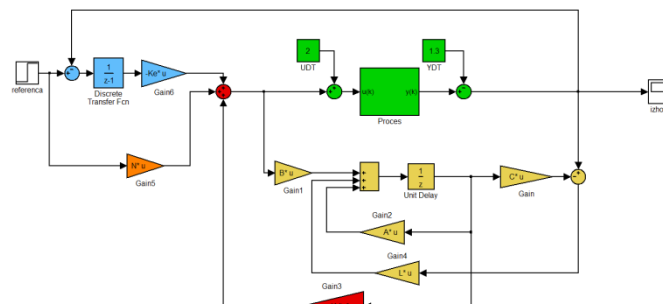
$$y(k+1) = C_N x_N(k) = [C \ 0] \begin{bmatrix} x(k) \\ x_e(k) \end{bmatrix} \quad (12)$$

$$u(k) = -K_N x_N(k) = -[K \ K_e] \begin{bmatrix} x(k) \\ x_e(k) \end{bmatrix} \quad (13)$$

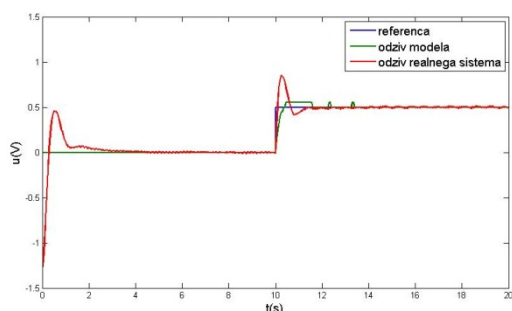
Želene pole regulatorja določimo tako, da bo zaprtzoančni sistem deloval dvakrat hitreje kot odprta zanka. Kljub temu, da imamo le eno stanje (ki je enako merjenemu izhodu), uporabimo opazovalnik, saj nam ta služi kot filter in zmanjšuje šum meritve. Želeni poli observatorja naj bodo dvakrat hitrejši od polov osnovnega zaprtzoančnega sistema (z regulatorjem, brez dodatnega stanja  $x_e(k)$ ).

#### 4.1 Hitrostno vodenje preko Simulink-a

Shemo vodenja v okolju Simulink prikazuje slika 6. Zeleno obarvan del predstavlja realen proces ali njegov model z ustrezno kompenzacijo delovne točke. Spodaj na sliki je prikazano ojačenje  $K$  regulatorja (rdeča barva), ki na podlagi ocenjenih stanj (izhod opazovalnika, ki je obarvan rumeno) prispeva del regulirne veličine. Preostali del  $u(k)$  izračunava razširjen regulator (integral pogreška med referenco in merjenim izhodom v modri barvi ter predkrmilno ojačenje oranžne barve). Rezultat regulacije je prikazan na sliki 7. Pri odzivu realnega sistema opazimo večji prenihaj, kar lahko pripisemo vztrajnosti vrtečega se obroča in motorja.



Slika 6. Razširjen regulator stanj z opazovalnikom.



Slika 7. Odziv na stopničasto spremembo reference hitrosti.

## 4.2 Hitrostno vodenje z mikrokrmilnikom

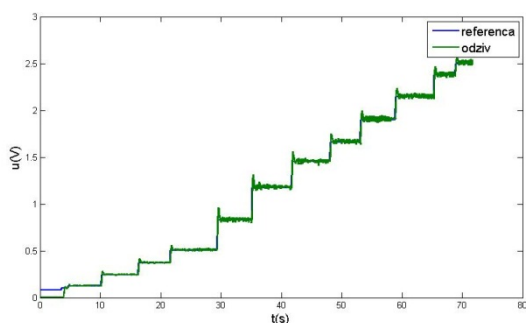
Regulator želimo realizirati še s pomočjo razvojne plošče Arduino. Koncepti vodenje ostajajo enaki kot zgoraj, le zapisati jih moramo v obliki, ki bo primerna za zapis v programski kodi – enačbe (14), (15) in (16) veljajo le za deviacijske signale. Potrebno je ustrezno kompenzirati delovno točko, pozabiti pa ne smemo na definiranje ustreznih konstant.

$$u(k) = -Kx(k) - K_e x_e(k) + Nr(k) \quad (14)$$

$$x_e(k+1) = x_e(k) + (r(k) - y(k)) \quad (15)$$

$$x(k+1) = (A - LC)x(k) + Bu(k) + Ly(k) \quad (16)$$

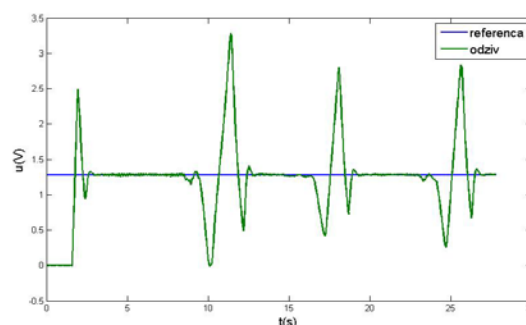
Referenco posredujemo preko vmesnika NI-PCI 6014 na dodatni analogni vhod, možno bi jo bilo nastavljanje tudi s pomočjo serijske komunikacije. Izhod senzorja prav tako opazujemo znotraj okolja Simulink. Rezultati so prikazani na slikah 8 in 9.



Slika 8. Sledenje spremenljivi hitrostni referenci – vodenje z mikrokrmilnikom Arduino.

Slika 8 prikazuje sledenje referenci (odstopanje od delovne točke izhoda), ki je pričakovano podobna razmeram na sliki 7. Slika 9 prikazuje izregulacijo motenj. Motnje generira

uporabnik, ki z roko rahlo upočasni vrtenje obroča.



Slika 9. Izregulacija motenj pri hitrostnem vodenju z uporabo Arduino-a.

## 5 Pozicijsko vodenje

Model pri pozicijskem vodenju je že po naravi integriren, zato se pogrešku v ustaljenem stanju izognemo z uporabo osnovnega regulatorja iz enačbe (2), ki pa mu moramo dodati tudi opazovalnik (4). Sledilno delovanje zagotovimo tako, da ustreznemu stanju preprosto odštevamo referenco.

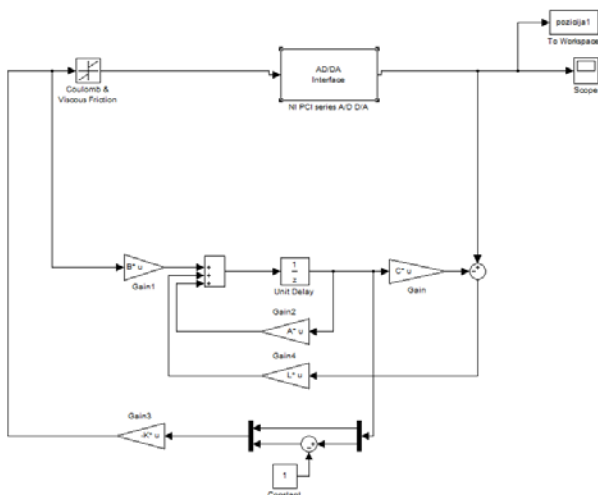
Pri načrtovanju regulatorja izberemo dinamiko zaprtozančnega sistema z dušenjem  $\zeta=0.8$  in lastno frekvenco  $\omega_N=4$  rad/s, za observator določimo dušenje  $\zeta=0.95$  in lastno frekvenco  $\omega_N=8$  rad/s. Pola preračunamo v skladu z enačbama (17) in (18):

$$s_{1,2} = -\zeta\omega_N \pm i\omega_N\sqrt{1-\zeta^2} \quad (17)$$

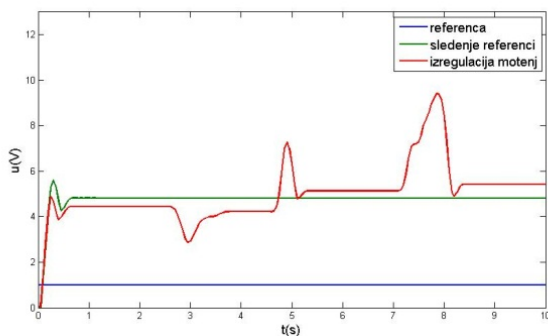
$$z_{1,2} = e^{s_{1,2}T} \quad (18)$$

### 5.1 Pozicijsko vodenje v okolju Simulink

Iz rezultatov na sliki 11 opazimo, da sledenje ni točno. Že na začetku (pred nastopom kakršnekoli motnje) je neko odstopanje, to se ponovi tudi po vsaki izregulaciji motnje. V praksi to pomeni, da obroč ne zavzame vedno natančno iste pozicije. Razlogi izvirajo iz samega procesa; naprava ima neko mrtvo cono, lastno vztrajnost, pojavlja se trenje in podobno. Motnje generira uporabnik in sicer preko ročnega izmika obroča iz ustaljene lege.



Slika 10. Shema pozicijskega vodenja v programu Simulink.



Slika 11. Rezultati pozicijskega vodenja pri konstantni referenci.

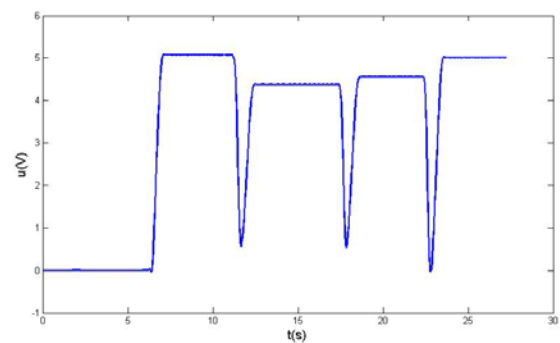
## 5.2 Pozicijsko vodenje z mikrokrmilnikom

V razvojnem okolju Arduino zapišemo ustrezne enačbe vodenja (19) in (20). Poskrbeti moramo tudi za mrtvo cono naprave. Referenca je v tem primeru konstantna, definiramo jo z  $x_{SS}$ .

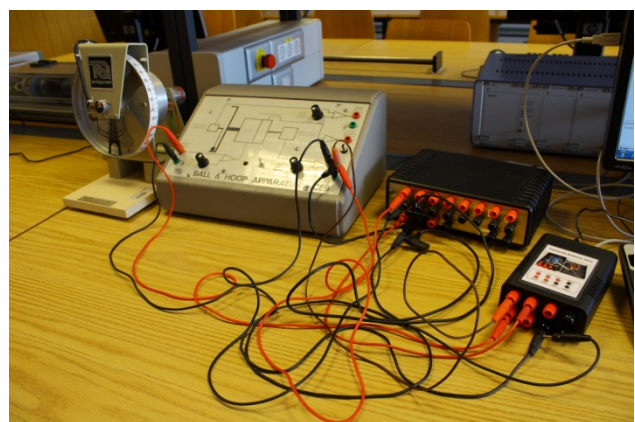
$$u(k) = -[K_1 \quad K_2] \begin{bmatrix} x_1(k) \\ x_2(k) - x_{SS} \end{bmatrix} \quad (19)$$

$$x(k+1) = (A - LC)x(k) + Bu(k) + Ly(k) \quad (20)$$

Izhod sensorja opazujemo v Simulink-u. Rezultat, ki prikazuje izravnavo motenj (uporabnik ročno izmakne obroč) je prikazan na sliki 12 in je podoben rezultatom slike 11.



Slika 12. Izregulacija motenj pri konstantni referenci – pozicijsko vodenje z mikrokrmilnikom.



Slika 13. Delovna naprava z vmesnikom NI-PCI 6014 in mikrokrmilnikom Arduino Uno.

## 6 Zaključki

Na digitalno pozicijsko in hitrostno vodenje izvedeno s pomočjo regulatorja stanj na osnovi opazovalnika vpliva več dejavnikov. Ti so izbrani čas diskretizacije zveznega sistema v diskretni sistem, način določitve modela s prenosno funkcijo (določitev ojačenja in časovne konstante) ter pravilno izbrani želeni poli.

Naš cilj je bil implementacija sistema vodenja na popularnem nizkocenovnem razvojnem mikrokrmilniku Arduino Uno. Prepričali smo se o njegovi enostavni uporabi in programiranju, hkrati pa se je to razvojno okolje izkazalo za dovolj zmogljivo pri izvedbi kompleksnejših regulacijskih algoritmov.

## 7 Literatura

- [1] S. Blažič, Digitalno vodenje, Fakulteta za elektrotehniko v Ljubljani, Ljubljana 2013
- [2] G. Klančar, Digitalno vodenje: Laboratorijske vaje, Fakulteta za elektrotehniko v Ljubljani, Ljubljana 2013
- [3] G. F. Franklin, J. D. Powell, M. L. Workman, Digital Control of Dynamic Systems Second Edition, Addison-Wesley Publishing Company, New York 1990
- [4] D. Matko, Računalniško vodenje procesov, Založba FER, Ljubljana 1995
- [5] B. Zupančič, Računalniška simulacija, Fakulteta za elektrotehniko v Ljubljani, Ljubljana 2012
- [6] D. Matko, Diskretni regulacijski sistemi, Založba ZAFER, Ljubljana 1991
- [7] <http://www.arduino.cc/>
- [8] Matlab, The Language of Technical Computing, Version 7.9.0, The Math Works Inc., 2009.