

Vodenje invertiranega sferičnega nihala z robotom tipa SCARA

Žiga Petrič¹, Igor Škrjanc²

¹Tehna d.o.o., Tehnološki park 19, 1000 Ljubljana

²Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška cesta 25, 1000 Ljubljana
ziga.petric@tehna.si, igor.skrjanc@fe.uni-lj.si

Control of Inverted Pendulum with SCARA type Robot

This article considers possibilities of system simulation and modelling using a PLC (producer *Allen-Bradley*) and complementing software packages. Resulting platform aims for user training and commercial demonstration purposes. Selected control task involves SCARA type robot with spherical (2 DOF) inverted pendulum, pivoting freely on its effector point. Robot will swing up the pendulum and maintain upright position, while being able to move inside its working envelope and execute secondary tasks. Work includes development of spherical pendulum governing equations, swing-up and control procedures, simulation and 3D visualization of complete system. Graphical interface enables user to command the experiment and set different variables.

Kratek pregled prispevka

Prispevek raziskuje možnosti modeliranja in simulacije sistemov na PLC krmilniku proizvajalca *Allen-Bradley* in pripadajoči programski opremi. Za nalogo smo izbrali robot tipa SCARA, ki ima na prijemalo pritrjeno invertirano sferično nihalo (to je nihalo, ki je prosto gibljivo v dveh prostostnih stopnjah). Naloga robota je, da zaniha nihalo v navpično lego in jo vzdržuje, medtem pa se lahko giblje v delovnem območju in izvaja druge operacije (npr. pobiranje in odlaganje predmetov). Delo vključuje izdelavo matematičnega modela nihala, algoritem za zanihanje nihala v navpično lego, regulator za vodenje nihala, simulacijo in 3D vizualizacijo celotnega sistema. Narejen je tudi grafični vmesnik (HMI) za upravljanje s poskusom in nastavljanje različnih parametrov.

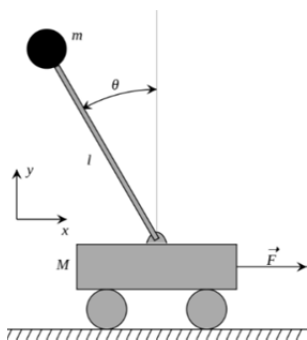
1 Uvod

Cilj naloge je preučiti, uporabiti in ovrednotiti nekatere napredne funkcije, ki jih ponuja oprema proizvajalca Rockwell Automation (Allen-Bradley), vendar so pri realizaciji praktičnih projektov večinoma neizkoriščene. Te funkcije so mehatronika oz. robotika, napredno vodenje, možnosti simulacije in modeliranja sistemov. Nastala naj bi demonstracijska in učna platforma, uporabna za izobraževalne namene in za prikaz zmožnosti krmilnega sistema potencialnim uporabnikom.

Fizična naprava je za takšen namen nepraktična, ker ni prenosljiva, poleg tega pa potrebujemo delovno mesto za več udeležencev šolanja ali predstavitve hkrati.

Demonstracijska naj bo torej modelirana in simulirana, vendar do te mere, da so rezultati hitro in z minimalnimi posegi prenosljivi na realno, fizično napravo. V tem smislu je izbrana platforma za simulacijo programabilni krmilnik (proizvajalca Allen-Bradley, družina CompactLogix), za vizualizacijo pa smo izbrali program RSTestStand (Rockwell Automation).

Kot delovno nalogo smo izbrali invertirano nihalo na vozičku, ki je klasičen referenčni sistem za preizkušanje različnih sistemov vodenja (slika 1.)



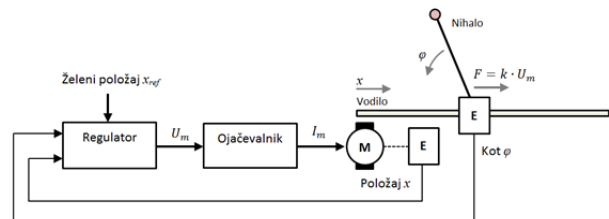
Slika 1: Shema invertiranega nihala na vozičku.

2 Invertirano nihalo kot eksperimentalna platforma

Praktično je tak sistem navadno izdelan kot laboratorijska naprava za izobraževalne namene, ki jo sestavlja:

- linearno vodilo (dolžine 1 do 2 m);
- voziček s pritrjenim nihalom (0,2 do 1 m);
- DC elektromotor, ki žene voziček direktno ali preko jermena;
- napetostno krmiljen ojačevalnik (driver) za motor;
- dva optična linijska dajalnika (enkoder) za merjenje kota odmika nihala in pomika vozička, z vezjem za pretvorbo merjene veličine v analogno (napetost) ali digitalno vrednost;
- regulator, ki je lahko analogno vezje z operacijskimi ojačevalniki, programabilni (mikro)krmilnik ali PC programska oprema (npr. Matlab paket) z ustreznimi I/O vmesniki.

Splošno shemo naprave prikazuje slika 2.



Slika 2: Eksperimentalna naprava z invertiranim nihalom.

Vhodne veličine regulatorja so zeleni in dejanski položaj vozička ter dejanski položaj nihala. Zeleni položaj oz. kot nihala je 0 (navpična lega); če je ta različen od 0, se mora sistem pospešeno gibati v pozitivno ali negativno smer, to gibanje pa ima mehanske omejitve. Izhodna veličina regulatorja je napetost U_m , ojačena v krmilni tok motorja I_m . Ta generira moment M in preko jermenskega ali zobniškega prenosa silo F na voziček. Navadno se smatra, da so U_m , I_m , M in F linearno odvisne veličine:

$$F = k_1 \cdot M = k_2 \cdot I_m = k_3 \cdot U_m. \quad (1)$$

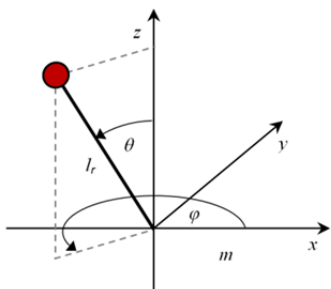
Regulator, ki vzdržuje navpično lego nihala, je lahko različnih izvedb; v literaturi lahko najdemo klasično rešitev z regulatorjem stanj (State Space Controller), PID regulator, uporabo mehke logike (Fuzzy Logic), nevronske mreže (Neuron Network Controller) in druge različne sisteme.

Alternativna oblika te naprave je »vozilo«, ki je v mirovanju nestabilno, vendar lahko lovi ravnotežje z vožnjo naprej in nazaj. Praktična izvedba je npr. Segway, pri katerem je »nihalo« voznik in krmilo, »voziček« pa sta dve kolesi, ki sta gnani neodvisno. Naprave takšnega tipa za izobraževalne namene lahko najdemo tudi na osnovi Fischertechnik in LEGO Mindstorms sestavljanek.

3 Dodatne zahteve

Problem invertiranega nihala je torej že mnogokrat teoretično in praktično obdelan, zato smo ga razširili z dodatnimi zahtevami:

- Nihalo je sferično nihalo, naj ima dve prostostni stopnji (slika 3.).



Slika 3: Sferično nihalo.

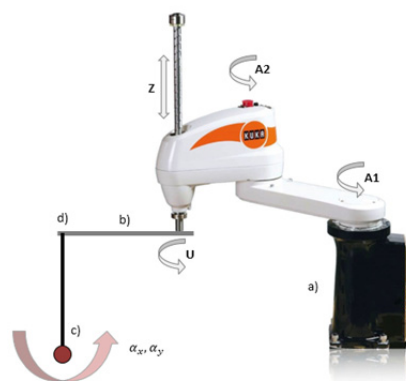
- V tem primeru je potreben sistem, ki lahko premika pivot nihala v smereh x in y . Analitično je najpreprostejša uporaba kartezičnega sistema, katerega prostostni stopnji gibanja sta x in y (lahko tudi z). Zdi se, da je regulator za takšen sistem preprosta razširitev že prikazane naprave (slika 2.) na dve neodvisni koordinati gibanja. Kot zahtevnejšo rešitev smo izbrali industrijski robot tipa SCARA, ki ravno tako omogoča gibanje v ravnini x - y , pri čemer pogoni za posamezne osi gibanja niso neodvisni. Kratica SCARA pomeni Selective Compliance Articulated Robot Arm in je pogosta oblika robota za preprostejša opravila (npr. "Pick-and-place").
- Sistem mora imeti možnost, mehansko in v smislu krmiljenja, da zaniha nihalo iz spodnje (stabilne) lege v zgornjo (nestabilno), kjer vodenje prevzame

regulator. Za ta namen se zdi SCARA robot primerna konstrukcija, ker minimalno ovira prosto gibanje nihala.

- Realiziranih naj bo več tipov regulatorjev, med katerimi lahko preklapljam (v realnem času) in primerjamo delovanje.
- Sistem bo sicer simuliran, vendar bo grafično prikazan v realnem času v 3D izrisu. Realiziran bo operatorski vmesnik za upravljanje s poskusno napravo.

4 Izvedba

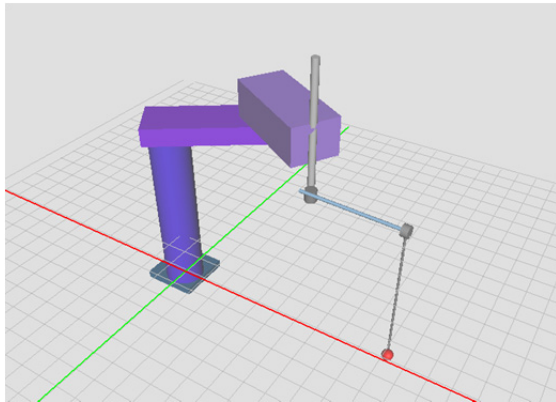
Idejno izvedbo naprave prikazuje slika 4.



Slika 4: Idejna shema SCARA robota s sferičnim nihalom na prijemalu.

Poskusna naprava je torej robot tipa SCARA (a). Na mesto za vpetje orodja je pritrjen tog nosilec (b), ter nanj nihalo (c) preko krogličnega zgloba (*universal joint*) (d). Nosilec omogoča, da se nihalo lahko prosto giblje v vsaj eni ravnini za polni kot 360° in pri tem ne zadane v druge dele robota. Prostostne stopnje robota so označene kot $A1$, $A2$, Z in U . $A1$ in $A2$ določata položaj roke v x - y ravnini, os Z ustreza kartezični koordinati z , os U pa opravlja vrtenje orodja okoli osi z . V pravilnem položaju roke robota je možno zanihati nihalo v navpično lego. Lego nihala določata kota α_x in α_y , ki ju dobimo s projekcijo nihala na ravnino x - z in y - z .

Idejno shemo smo nato realizirali kot 3D vizualizacijo v programu *RSTestStand* (slika 5.).



Slika 5: 3D vizualizacija robota s sferičnim nihalom v programu RSTestStand.

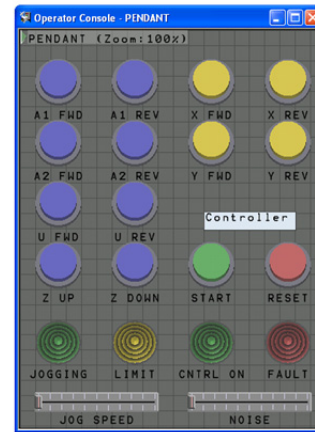
RSTestStand je namenjen interaktivnemu razvoju in preizkušanju krmilnega (PLC) programa na namizju – vhodne in izhodne veličine PLC zamenja s simuliranim procesom. Za to nalogo smo iz množice funkcij uporabili samo možnost 3D izrisa, virtualni operatorski vmesnik in generator šuma. Kvaliteta grafike 3D vizualizacije sicer zaostaja za sodobnimi sistemi, ki ponujajo 3D izris, je pa izdelava in animacija objektov dokaj preprosta in intuitivna.

Model robota in nihala je sestavljen iz osnovnih geometrijskih oblik (kvader, valj, krogl) v 3D editorju programa RSTestStand. Posamezni gradniki so hierarhično gnezdeni in povezani v skupine.

Objekti ali skupine objektov imajo možnost povezave atributov rotacije, translacije in velikosti (po koordinatah x , y ali z) s spremenljivkami, katerih vir je krmilnik ali interni modeli oz. generatorji RSTestStand okolja. Tako lahko animiramo sklepe robota (A1, A2, Z, U), oba kota nihala (α_x , α_y) in dolžino nihala (l_r).

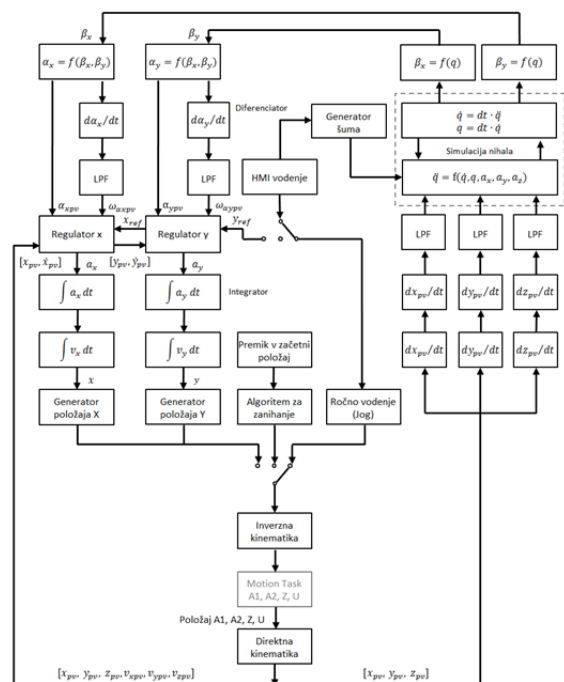
RSTestStand uporablja za povezavo s krmilnikom (ali drugimi napravami) standard OPC, kot fizično vodilo pa je uporabljen protokol Ethernet/IP (industrijski ethernet).

Operatorski vmesnik (HMI) za upravljanje s poskusom prikazuje slika 6.



Slika 6: HMI vmesnik v programu RSTestStand.

Programski del simulacije je v celoti realiziran na platformi PLC krmilnika CompactLogix in vključuje ročno in avtomatsko vodenje robota (inverzna kinematika za tip SCARA), model nihala, regulator (dve varianti), algoritem za zanihanje nihala in pomožne programe. Bločna shema je prikazana na sliki 7., posamezne sklope pa bomo podrobneje opisali v nadaljevanju.



Slika 7: Bločna shema simulacije na PLC platformi.

5 Simulacija nihala

Simulacija nihala povezuje položaj pivota nihala x_p , y_p in z_p (ki je funkcija položajev,

hitrosti in pospeškov osi A1, A2, Z in U) s položajem kotov α_x in α_y .

Za realizacijo regulatorja zadostuje model dveh nesklopljenih nihala, ki sta projecirani na ravnini x - z in y - z , želeli pa smo razviti kompletan model.

Razvoj simulacije naj bi potekal takole:

- razvoj matematičnega modela nihala z diferencialnimi enačbami (Lagrangeova funkcija);
- pretvorba enačb v eksplicitno obliko, kjer je najvišji odvod odvisne spremenljivke enak (nelinearni) funkciji nižjih odvodov, torej:

$$\ddot{q} = f(\dot{q}, q, \dots) \quad (2)$$

kjer je q posplošena koordinata;

- numerično reševanje diferencialnih enačb z Eulerjevo metodo, pri kateri najprej izračunamo vrednost najvišjih odvodov z enačbo (2), nato s (približno) integracijo izračunamo vse nižje odvode, jih vstavimo nazaj v izraz (2) in to zanko ponavljamo.

Spremenljivke, ki jih potrebujemo za opis nihala, so:

$x, y, z \dots$ kartezični koordinatni sistem,

$\varphi, \theta \dots$ sferična kota nihala,

$l_r \dots$ dolžina nihala,

$m \dots$ točkasta masa,

$g \dots$ gravitacija.

Lagrangeova funkcija sistema je razlika med kinetično (T) in potencialno energijo (V) sistema (mehansko trenje, upor in izgube v tej izpeljavi zanemarimo):

$$L = T - V \quad (3)$$

Po izpeljavi dobimo diferencialne enačbe za sferično nihalo:

$$\ddot{\theta} = (\dot{\varphi}^2 \sin(\theta) \cos(\theta) + g \sin(\theta) / l_r) \quad (4)$$

$$\ddot{\varphi} = -2\dot{\varphi}\dot{\theta} \cos(\theta) / \sin(\theta) \quad (5)$$

Vidimo, da izraz (5) predstavlja težavo za reševanje z Eulerjevo numerično metodo, ker

vsebuje singularnost v točki $\sin(\theta) = 0$, torej v ravnovesni ($\theta = 0$) in navpični legi ($\theta = \pi$). Opisana metoda reševanja je zato tu neuporabna.

Po več poskusih smo razvili alternativno metodo, kjer problem rešujemo v x - y - z koordinatnem sistemu kot gibanje masnega delca m po površini sfere z radijem l_r (slika 8.)

- V nekem trenutku ima masa m položaj $P_0(x, y, z)$, ki je na površini sfere z radijem l_r , in hitrost V , ki je pravokotna na normalo sfere oz. P_0 .

- Pod vplivom sile $F(x, y, z)$ naredi masa m v času dt pot $S(x, y, z)$ do novega položaja $P_1(x, y, z)$.

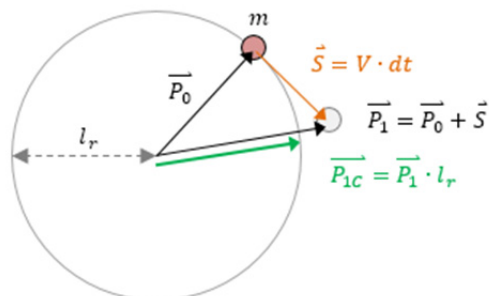
- P_1 ni več na površini sfere, zato izvedemo korekcijo:

$$P_{1c} = P_1 \cdot l_r / |P_1| \quad (6)$$

- Nov trenutni položaj P_0 postane enak P_{1c} .

- Zanko ponavljamo s periodo dt .

(Tak model ne upošteva rotacije nihala okoli lastne osi.)



Slika 8: Izpeljava rešitve za invertirano nihalo.

Vhodi v sistem so pospeški a_x , a_y in a_z . Dobimo jih z dvojnimi odvaji položaja pivota nihala. Izhod sistema je položaj mase nihala P_{x0} , P_{y0} , P_{z0} , ki je nato pretvorjen v kota α_x in α_y .

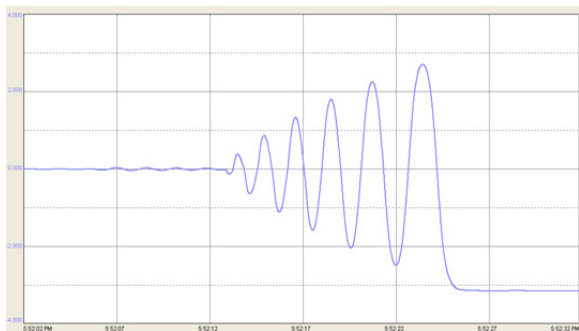
6 Strategija za zanihanje

Strategija za zanihanje temelji na naslednjih izhodiščih:

- Smer zanihanja je v ravnini y - z , nekako na sredini dosega robota v koordinati x , kjer ima ta največji doseg v smeri y .

- Temu ustreza tudi izbrani tip robotske roke in izvedba pritrdilnega mehanizma (kot α_y , ki predstavlja odklik v ravnini y - z , nima omejitve gibanja, medtem ko je kot α_x (v ravnini x - z).
- Pri vzburjanju se sme nihalo gibati samo v izbrani ravnini, sicer pride do kroženja.
- Energijo za vzburjanje je potrebno dovajati v intervalih, ki so enaki naravni frekvenci sistema (ki pa ni konstantna).
- Postopek je uspešen, ko privede nihalo v bližino navpične lege pri dovolj majhni krožni hitrosti, oba parametra sta nastavljiva v programski proceduri in sta bila ugotovljena s poskušanjem. Vgrajen evaluator nato preklopi fazo zanihanja v regulacijo.

Rezultat vidimo na sliki 9., prikazan je dosežen kot α_y , ki je v spodnji legi enak 0 in se v zgornji navpični legi približa $-\pi$. Celotno trajanje procesa zanihanja je 15 sekund.



Slika 9: Časovni diagram zanihanja od mirovne lege do vklopa regulatorja.

7 Regulator

Invertirano nihalo je klasična platforma za preizkušanje različnih regulatorjev. V literaturi so opisani uspešni primeri vodenja s PID, regulatorjem stanj (*State Space Controller*), mehko logiko (*Fuzzy Logic*), adaptivnimi nevronskimi mrežami (*ANN*) in drugim.

Najpogostejši regulator je *regulator stanj*, načrtan po metodi LQR. Ta tip regulatorja je bil prvi, ki smo ga uporabili na razvitem modelu.

Izhodišče za načrtovanje regulatorja je predpostavka, da sferično nihalo v okolici navpične (in ravnovesne) lege ($\alpha_x, \alpha_y \rightarrow 0$) lahko aproksimiramo z dvema nesklopljenima (neodvisnima) nihaloma – projekcijama na

ravnini x - z in y - z . Načrtovanje regulatorja torej lahko opravimo na osnovi modela za nihalo z eno prostostno stopnjo (7), kot je to v primeru klasičnega poskusa z vozičkom.

$$\ddot{\alpha}_y = \left(g \cdot \sin \alpha_y + \ddot{y} \cdot \cos \alpha_y - \frac{f \cdot \dot{\alpha}_y}{m \cdot l_r} \right) \cdot \frac{1}{l_r} \quad (7)$$

Uporabljene spremenljivke in njihove privzete vrednosti so:

$\alpha_x, \alpha_y \dots$ kot nihala;

$g \dots$ gravitacijski pospešek ($9,8 \text{ m/s}^2$);

$x, y, z \dots$ kartezični koordinatni sistem;

$f \dots$ konstanta viskoznega trenja ($0,02 \text{ Nm/s}$);

$l_r \dots$ dolžina nihala ($0,6 \text{ m}$);

$m \dots$ masa nihala ($0,3 \text{ kg}$).

Sistem lahko lineariziramo s pretpostavko (8):

$$\sin(\alpha_y) \approx \alpha, \quad \cos(\alpha_y) \approx 1, \quad \dot{\alpha}_y \approx 0 \quad (8)$$

Dobimo izraz (9), ki je primeren za pretvorbo v matrično obliko - prostor stanj (10):

$$\ddot{\alpha}_y = \left(g \cdot \alpha_y + \ddot{y} \right) \cdot \frac{1}{l_r} \quad (9)$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & g/l_r & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1/l_r \end{bmatrix}$$

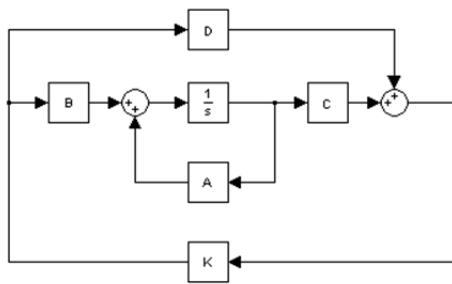
$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad D = 0 \quad (10)$$

Odpertozančni sistem je inherentno nestabilen, brez ustreznega zunanje vzburjanja bo nihalo padlo iz navpične v spodnjo ravnovesno lego.

Regulator stanj dobimo, če sistem zaključimo z (negativno) povratno zanko preko ojačanja K (slika 10).

Vhod sistema, u je sedaj povezan z vektorjem stanj X preko matrike ojačanja K :

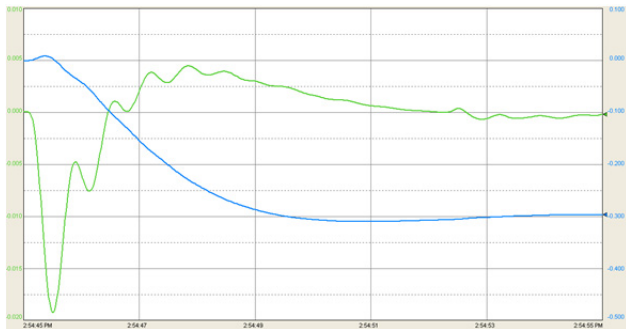
$$u = -K \cdot X \quad (11)$$



Slika 10: Shema zaprtozančnega sistema v obliki prostora stanj.

Parametre vektorja K lahko dobimo s pomočjo paketa *Matlab* in vgrajene funkcije *lqr*. Dobljen rezultat je $k_1 = -1,00$, $k_2 = -1,91$, $k_3 = -27,70$ in $k_4 = -6,85$.

Odziv regulatorja na stopničasto vzbujanje (0,3 m) prikazuje slika 11.

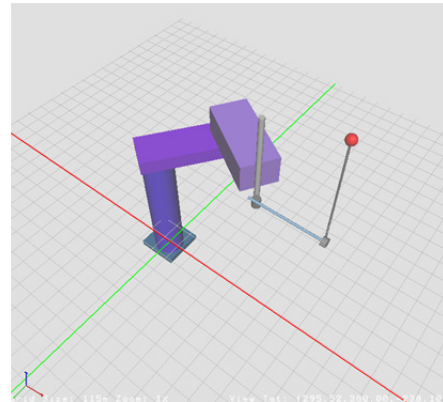


Slika 11: Odziv LQR regulatorja.

Regulator vklopimo, ko algoritem za zanihanje uspešno privede nihalo v bližino navpične lege pri majhni kotni hitrosti. Regulator prevzame gibanje robota v ravnini x - y in s tem vzdržuje navpično lego nihala (slika 12.). V model nihala pripeljemo tudi generator šuma (slika 7.), ki simulira različne motnje v sistemu. Referenčni položaj pivota nihala v ravnini x - y seveda lahko spreminjamo. Neodvisno od regulatorja lahko vodimo osi Z (koordinatna os z) in U (vrtenje orodja), kar predstavlja motnjo za sistem.

Izveden je tudi PID regulator, ki je po pričakovanju manj stabilen od regulatorja stanj, saj ima kot vhodni veličini na voljo samo dve spremenljivki, x in α_x (oz. y in α_y).

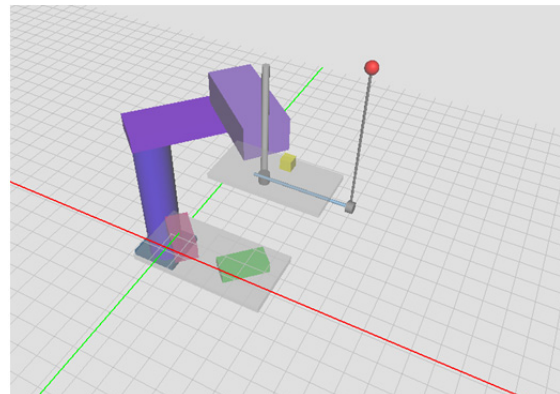
Natančnost odlaganja je seveda neposredno odvisna od motnje, ki jo nihalo vnaša v gibanje prijemala.



Slika 12: Regulator v delovanju.

8 Sekundarna opravila

Robot se lahko, medtem ko vzdržuje navpično lego nihala, giblje v delovnem območju in izvaja sekundarne operacije (*Nullspace Motion*), npr. pobiranje in odlaganje predmetov (*pick-and-place*). V ta namen smo v okolju *RSTestStand* dodali virtualno delovno površino z nekaj predmeti, ki jih lahko robot z orodjem prime, prenese in odloži (slika 13.).



Slika 13: Robot in virtualno delovno okolje.

9 Zaključki

Opisano platformo smo v podjetju uspešno uporabili v demonstracijske in učne namene. Vsi rezultati, matematične izpeljave, programska koda in drugo so prosto dostopni pri avtorju prispevka.

Platforma je odprta za množico nadaljnjih poskusov in upamo, da jih bomo v sodelovanju s Fakulteto in bodočimi študenti tudi uresničili.

10 Literatura

- [1] Control Tutorials for MATLAB and Simulink (CTMS).
<http://ctms.engin.umich.edu/CTMS/index.php?aux=Home>. [Dostopano: 12. 8. 2014].
- [2] System Dynamics.
http://www.profjrwhite.com/system_dynamics/sdyn/s0/outl509/outl509.html. [Dostopano: 12. 8. 2014].
- [3] R. Yang, Geometric Techniques for Control of a 2-DOF Spherical Inverted Pendulum. The Hong Kong University of Science and Technology, Department of Electrical and Electronic Engineering, 2000.
- [4] G. Schreiber, C. Ott in G. Hirzinger, Interactive Redundant Robotics: Control of the Inverted Pendulum with Nullspace Motion. Wessling: German Aerospace Center – DLR, Institute for Robotics and Mechatronics, b. 1.
- [5] B. Sprenger, L. Kucera in S. Mourad, Balancing of an Inverted Pendulum with a SCARA Robot. Zurich: Swiss Federal Institute of Technology Zurich (ETHZ), Institute of Robotics.
- [6] K. J. Astrom in K. Furuta, “Swinging up a Pendulum by Energy Control”, *Automatica*, vol. 36, 2000.
- [7] Control System Design. Dosegljivo: <http://csd.newcastle.edu.au/control/simulations/pendulum.html>. [Dostopano: 12. 8. 2014].
- [8] J. P. Hespanha, Undergraduate Lecture Notes on LQG/LQR controller design, April 1, 2007.
- [9] Wikipedia, The Free Encyclopedia, »Inverted_pendulum«. Dosegljivo: http://en.wikipedia.org/wiki/Inverted_pendulum. [Dostopano: 12. 8. 2014].
- [10] Wikipedia, The Free Encyclopedia, »Euler–Lagrange equation«. Dosegljivo: http://en.wikipedia.org/wiki/Euler_lagrange. [Dostopano: 12. 8. 2014].
- [11] Pendulum Project From Scswiki. Dosegljivo: <http://www5.in.tum.de/wiki/index.php/PendulumProject>. [Dostopano: 12. 8. 2014].
- [12] Y. Beceriklia in K. Celik, Fuzzy control of inverted pendulum and concept of stability using Java application. *Mathematical and Computer Modelling*, 46, str. 24–37, 2007.
- [13] Damped Spherical Pendulum – Wolfram Demonstrations Project Dosegljivo: <http://demonstrations.wolfram.com/DampedSphericalPendulum>. [Dostopano: 12. 8. 2014].
- [14] A. Zaidan, B. M. Y. Nouri in B. Alsayid, Swing Up a Pendulum by Energy Control, *International Journal of Engineering and Technology*, vol. 2, no. 3, 2012.
- [15] K. J. Åström in K. Furuta, Swinging up a pendulum by energy control. Paper presented at IFAC 13th World Congress, San Francisco, California, 1996.
- [16] Googol Technology, ”Inverted Pendulum Experimental Manual suitable for GLIP series,” 2nd edition, 2006.
- [17] M. E. M. Janthong. Doktorska disertacija: Design and Implementation of Control Concepts for Image-Guided Object Movement. Von der Fakultät für Maschinenbau der Gottfried Wilhelm Leibniz Universität Hannover zur Erlangung des akademischen Grades, 2006.
- [18] B. Zupančič, Simulacija dinamičnih sistemov. Ljubljana: Fakulteta za elektrotehniko in računalništvo, 1995.
- [19] B. Zupančič, Zvezni regulacijski sistemi (2. del, 2. popravljena in dopolnjena izdaja), Ljubljana: Fakulteta za elektrotehniko in računalništvo, 1995.
- [20] I. Škrjanc, Regulacije I, Ljubljana: Fakulteta za elektrotehniko in računalništvo, 2006.
- [21] I. Škrjanc, Seminar: vodenje sistemov II. Praktikum (1. izdaja). Ljubljana: Fakulteta za elektrotehniko in računalništvo, 2006.
- [22] T. Bajd, Osnove robotike (7. popravljena in dopolnjena izdaja). Ljubljana: Fakulteta za elektrotehniko, 2006.
- [23] B. Jurčič in B. Orel, Numerične metode. Ljubljana: Fakulteta za elektrotehniko in računalništvo, 2004.