

UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET – PODGORICA

Spec. Sci. Novica Daković

**UPRAVLJANJE INVERZNIM KLATNOM POMOĆU
DIFERENCIJALNOG FLATNESS KONTROLERA**
-MAGISTARSKI RAD -

Podgorica, 2016. godine

PODACI I INFORMACIJE O MAGISTRANDU

Ime i prezime	Novica Daković
Datum i mjesto rođenja	09.04.1991. u Nikšiću
Naziv završenog osnovnog studijskog i godina završetka studija	Energetika i automatika, 2012

INFORMACIJE O MAGISTARSKOM RADU

Naziv postdiplomskog studija	Automatika
Naslov rada	Upravljanje inverznim klatnom pomoću diferencijalnog flatness kontrolera
Fakultet na kojem je rad odbranjen	Elektrotehnički fakultet – Podgorica

UDK, OCJENA I ODBRANA MAGISTARSKOG RADA

Datum prijave magistarskog rada	05.12.2014
Datum sjednice vijeća na kojoj je prihvaćena tema	05.02.2015

Komisija za ocjenu teme i podobnosti magistranda	Prof. dr Zdravko Uskoković, ETF Podgorica Prof. dr Božo Krstajić, ETF Podgorica Doc. dr Milovan Radulović, ETF Podgorica
Mentor	Doc. dr. Milovan Radulović, ETF Podgorica
Komisija za ocjenu rada	Prof. dr Zdravko Uskoković, ETF Podgorica Doc. dr Milovan Radulović, ETF Podgorica Prof. dr Božo Krstajić, ETF Podgorica
Komisija za odbranu rada	Prof. dr Zdravko Uskoković, ETF Podgorica Doc. dr Milovan Radulović, ETF Podgorica Prof. dr Božo Krstajić, ETF Podgorica
Lektor	
Datum odbrane	21.01.2016
Datum promocije	

Predgovor

Tehnološki razvoj ljudske civilizacije praktično počinje od automatizacije svih životnih i radnih procesa. Još od kada je Džejms Vat konstruisao centrifugalni regulator brzine, postavlja se pitanje kako projektovati razne sisteme upravljanja i postići zadovoljavajući nivo stabilnosti i performansi pri djelovanju raznih unutrašnjih i spoljašnjih uticaja. U opštem slučaju, koncept povratne sprege je fundamentalan za rješavanje ovog problema. Ovaj rad će se baviti relativno novijom metodom upravljanja sistemima, flatness upravljanjem koje će se direktno porediti sa jednom od klasičnih, optimalnih metoda.

Rad može da posluži kao korisan materijal studentima osnovnih i postdiplomskih studija tehničkih i tehnoloških fakulteta, kao i svim onima koji se bave ili proučavaju sisteme automatskog upravljanja. Rad ima kako teorijski tako i praktični značaj, jer su rezultati potvrđeni eksperimentom. Autor se zahvaljuje svom mentoru profesoru Milovanu Raduloviću koji je bio značajna podrška tokom pisanja rada, kao i profesorici Nikol Gering sa Sarskog Univerziteta u Njemačkoj koja je pružila odgovore na mnoga pitanja i nedoumice.

Izvod rada

Rotaciono inverzno klatno ili Furuta klatno je mehanički uređaj koji se sastoji od klatna koje se slobodno kreće u vertikalnoj ravni i prikačeno je za rotirajuću platformu u horizontalnoj ravni. Furuta klatno je nelinearni i nestabilni dinamički sistem. Takođe, predstavlja veoma interesantan sistem upravljanja, sa stanovišta teorije i prakse. Zadatak upravljanja je projektovati algoritam koji će stabilizovati klatno u uspravnom položaju.

U radu su upoređena dva upravljačka algoritma na primjeru Furuta klatna: flatness i linearni kvadratni regulator. Flatness algoritam je vrlo efektivan za upravljanje nelinearnim „flat“ sistemima koji imaju za cilj da prate zadatu trajektoriju. Diferencijalni flat sistemi imaju zanimljive osobine, koje olakšavaju njihovo upravljanje. Linearni kvadratni regulator (LQR) predstavlja jednu od optimalnih metoda upravljanja linearnim sistemima, koja uključujući stanja dinamičkog sistema i upravljačke ulaze, a u skladu sa izabranim kriterijumom, generiše optimalno upravljanje. Suštinski, i jedna i druga metoda predstavljaju upravljanje po stanjima sistema, preko odgovarajućih koeficijenata pojačanja u povratnoj sprezi. Oba algoritma su implementirana na realnom sistemu u laboratorijskim uslovima i poređene su performanse, stabilnost i otpornost na spoljašnje smetnje u oba slučaja.

Abstract

Rotational inverted pendulum or Furuta pendulum is a mechanical device that consists of free vertical rotating pendulum with its pivot point mounted on a rotating base which is actuated in horizontal plane. Furuta pendulum is a nonlinear and unstable dynamic system. It is also a very interesting control system, both from a theoretical and practical point of view. The control task is to design an algorithm that will stabilize pendulum in the upright position.

This paper presents comparison of two control algorithms for Furuta pendulum control, flatness and linear quadratic regulator. Flatness is very effective for nonlinear flat systems in trajectory tracking tasks. Differential flat systems have useful features which can simplify control design. Linear quadratic regulator (LQR) is one of the optimal control techniques, which takes into account the states of the dynamical system and control input to make optimal control decisions. Essentially, both methods represent state feedback control using appropriate feedback gains. Both algorithms are implemented on the real system under laboratory conditions and comparative analysis of system performance, stability and disturbance rejection is made.

Lista skraćenica

ADC (*Analog-to-Digital Converter*) - analogno-digitalni konverter

DAC (*Digital-to-Analog Converter*) - digitalno-analogni konverter

DC (*Direct Current*) - jednosmjerna struja

DSP (*Digital Signal Processing*) - digitalna obrada signala

ECP (*Educational Control Products*) - proizvođač sistema upravljanja za edukativne svrhe

LQR (*Linear Quadratic Regulator*) - linearni kvadratni regulator

PC (*Personal Computer*) - personalni računar

SADRŽAJ

1. Uvod	1
2. Teorijski koncept diferencijalnih flat sistema	4
2.1. Definicija flat sistema	4
2.2. Definicija upravljačkog ulaza	5
2.3. Definicija stanja	6
2.4. Kanonski oblik za flat sisteme	7
2.5. Linearni flat sistemi	10
2.6. Planiranje trajektorije i feed-forward upravljanje	11
2.7. Projektovanje povratne sprege	14
3. Matematički model Furuta klatna	18
3.1. Nelinearne jednačine dinamike Furuta klatna	18
3.2. Linearizovani model Furuta klatna	21
3.3. Model u prostoru stanja	21
3.4. Laboratorijski model Furuta klatna	22
3.5. Efekti saturacije, diskretnog vremenskog odabiranja i senzorske kvantizacije	26
4. Linearni kvadratni regulator - LQR	28
4.1. Projektovanje LQR kontrolera	28
4.2. Osobine LQR upravljanja	30
4.3. Projektovanje LQR za Furuta klatno	30
5. Flatness upravljanje Furuta klatnom	32
5.1. Flat izlaz Furuta klatna	32
5.2. Brunovsky stanja Furuta klatna	34
5.3. Feed-forward upravljanje	35
5.4. Projektovanje povratne sprege	40
6. Eksperimentalna analiza Flatness i LQR metode	45
6.1. Praćenje referentne trajektorije	45
6.2. Robusnost na promjenu parametara	47

6.3. <i>Kompenzacija smetnje</i>	49
7. Zaključak	51
Literatura	53
Prilog 1 – Matlab kodovi	55
Prilog 2 – Kodovi za izvršni program	58

1. Uvod

Sistemi automatskog upravljanja su prisutni u skoro svim modernim tehnološkim sistemima. Princip povratne sprege je i dalje najmoćniji i najefektivniji način upravljanja sistemima, počevši od jednostavnih termostata koji regulišu sobnu temperaturu, pa sve do naprednih digitalnih kontrolera koji upravljaju procesima u industriji, robotici, avijaciji, satelitskoj navigaciji itd. Ubrzani razvoj računara, senzorskih i aktuatorskih elemenata omogućava primjenu veoma kompleksnih algoritama koji proizilaze iz teorije upravljanja sistemima.

Tipičan način primjene teorije upravljanja u mnogim modernim sistemima jeste da se invertuje dinamika sistema kako bi se proračunali upravljački ulazi potrebni za izvršenje specifičnog zadatka. Problem inverzije predstavlja pronalaženje odgovarajuće upravljačke akcije koja treba da prevede sistem iz jednog stacionarnog režima u drugi, ili da navede sistem da prati zadatu trajektoriju. U opštem slučaju, rješenje za ovakav problem nije jedinstveno i predstavlja kompromis između performansi i stabilnosti sistema, uz aktuatorska ograničenja. Često se ovaj kompromis opisuje kriterijumskom funkcijom koja treba da balansira zahtjeve performansi sistema sa zahtjevima stabilnosti, što rezultuje optimalnim upravljanjem. Problem inverzne dinamike je zasnovan na pretpostavci da je dinamika sistema poznata i fiksna. U praksi, greške pri modelovanju sistema, kao i prisustvo šuma često predstavljaju problem prilikom projektovanja regulatora. Koncept povratne sprege omogućava sistemu da lako prevaziđe ove probleme, kao i da značajno poveća efikasnost sistema tako što će ga stabilizovati.

Standardni način upravljanja u gotovo svim sistemima je da se definiše matematički model sistema, kako bi se dobilo rješenje inverznog problema i izvršila regulacija u povratnoj sprezi. Obično se sistem predstavlja linearnim modelom, tj. sistem se linearizuje u okolini radne tačke, a zatim se koriste tehnike upravljanja linearnim sistemima koje su veoma dobro istražene. Moguće je čak linearizovati sistem oko više radnih tačaka, pa u realnom vremenu mijenjati parametre regulatora u zavisnosti od radne tačke.

Međutim, ukoliko je sistem sam po sebi dovoljno kompleksan, upotreba linearne strukture često nije dovoljna da riješi problem upravljanja. To se pogotovo odnosi na probleme inverzne dinamike, gdje određeni zadatak može da se proširi u više radnih tačaka. Kako bi se riješio ovaj problem, teorija upravljanja pokušava da pored jednostavne linearne, izuči i mnoge nelinearne upravljačke strukture.

Ovaj rad će se bazirati na posebnoj klasi sistema upravljanja, tzv. diferencijalnih flat sistema. Flat sistemi su generalizacija linearnih sistema, u smislu da su svi linearni kontrolabilni sistemi flat sistemi, ali metode koje se koriste za upravljanje flat sistemima se znatno razlikuju od onih standardnih za linearne sisteme. Motivacija za razmatranje navedenog koncepta upravljanja nalazi se i u činjenici da je, osim brojnih radova u akademskoj sredini, veoma uspješno

implementiran u velikom broju industrijskih aplikacija. Upravljačka struktura, zasnovana na „flatness“ konceptu, može se primjeniti na mnoge elektro-mehaničke sisteme kao što su: letjelice, vozila, kranovi, mehatronički sistemi, alatne mašine, pozicioni sistemi [1]-[6]. Mogućnost eksperimentalne verifikacije rezultata dobijenih simulacijom predstavlja dodatni motivacioni faktor.

Koncept flatness upravljanja primjenjuje se na laboratorijskom modelu rotacionog inverznog klatno tj. Furuta klatna [7]-[10]. Od 1992. godine kada je nastalo kao izum Katsuhisa Furute na Tokijskom Tehnološkom Institutu, napisano je na hiljade naučnih radova koji opisuju primjenu raznih linearnih i nelinearnih zakona upravljanja na ovaj sistem. Ovaj tip klatna je specifičan jer predstavlja primjer kompleksnog nelinearnog oscilujućeg sistema. Sastoji se iz pokretnog diska koji pogonjen aktuatorom (DC motorom) rotira u horizontalnoj ravni i klatna koje je pričvršćeno za obod diska i slobodno rotira u vertikalnoj ravni. Furuta klatno je u suštini nestabilan sistem sa veoma izraženom nelinearnom dinamikom. Pripada klasi potpogonjenih („under-actuated“) mehaničkih sistema koje karakteriše manji broj upravljačkih ulaza u odnosu na broj stepeni slobode [11]-[12]. Potpogonjeni sistemi imaju široku primjenu u robotskim sistemima, posebno za robotske balansirajuće manipulatore ili „akrobote“. Dinamika Furuta klatna je slična dinamici akrobota. Za razliku od uobičajenih fabričkih robota koji su fiksirani za podlogu, ovi roboti moraju da se kreću u nekontrolisanom okruženju. Pri tome, oni moraju da održe vertikalnu stabilnost u tim uslovima, što uopšte nije lak zadatak.

U suštini, stabilizacija klatna se sastoji iz dva problema. Prvi problem je tzv. „swinging up“ problem koji zahtjeva da se klatno zamahivanjem iz neinverznog ili vertikalnog nadolje položaja prevede u inverzni, vertikalni nagore. Obično se pri tome želi minimizovati uložena energija za taj proces. Ovaj problem neće biti u fokusu ovog rada, pa je za potrebe eksperimenta primjenjena jednostavna metoda naglih pomjeraja diska u trenutku kada klatno pri zamahivanju prođe kroz tačku vertikalnog položaja nadolje, uzrokujući sve veće oscilovanje klatna oko ravnotežnog položaja. U trenutku kada se klatno pri sve većim zamahivanjima približi uspravnom položaju aktivira se odgovarajući algoritam za stabilizaciju klatna u tom položaju. To je ujedno i drugi problem, koji će biti tema ovog rada. Zadatak upravljanja će takođe biti i stabilizacija klatna kada se disk zarotira za određeni broj stepeni na način kako se definiše trajektorija ugaonog pomjeraja diska (npr. step, rampa ili polinom) što predstavlja problem praćenja trajektorije.

U radu će se na primjeru Furuta klatna, flatness porediti sa jednom od standardnih metoda upravljanja za linearne sisteme, optimalnim tj. LQR upravljanjem [13]-[15]. Pošto je LQR metoda za linearne sisteme, izvršiće se linearizacija klatna oko ravnotežne tačke (inverznom uspravnom položaju klatna). Osnovni cilj upravljanja za oba upravljačka algoritma je stabilizacija klatna u inverznom uspravnom položaju. Ovaj rad je podjeljen u sledeća poglavlja.

U poglavlju 2. opisan je teorijski koncept diferencijalnog flat sistema, kao i varijabli koje ga opisuju (stanja i ulazi). Takođe, predstavljen je kanonski oblik na koji se mogu svesti svi flat

sistemi. Opisana su dva koncepta flatness upravljanja koje treba da postignu željeno ponašanje sistema, upravljanje u otvorenoj (feedforward) i zatvorenoj sprezi (feedback).

U poglavlju 3. je izveden matematički model Furuta klatna na osnovu kojeg će se primjeniti flatness i LQR metode. Opisan je laboratorijski model klatna ECP A51 sa svojim fizičkim karakteristikama i ograničenjima [16].

Poglavlje 4. opisuje opšte principe LQR upravljanja za linearne sisteme što će u nastavku služiti kao podloga za primjenu ove metode na Furuta klatno.

U poglavlju 5. su principi flatness upravljanja iz poglavlja 2. primjenjeni na Furuta klatno i projektovana je povratna sprega koja stabilizuje sistem.

U poglavlju 6. su oba algoritma, flatness i LQR, primjenjena na laboratorijski model i izvršeno je njihovo poređenje u smislu praćenja zadate trajektorije, robusnosti, kompenzacije smetnje, kao i uložene energije.

Svi programske kodovi koji su sprovedeni u Matlab-u i softverskom paketu laboratorijskog modela Furuta klatna dati su u odgovarajućim prilogima na kraju rada.

2. Teorijski koncept diferencijalnih flat sistema

2.1. Definicija flat sistema

Teorija diferencijalnog Flatness-a je prvi put definisana od strane M. Fliess-a i saradnika [17]-[19], koji su u izučavanju nelinearnih sistema koristili formalizam diferencijalne algebre. Ovakav pristup zahtjeva poznavanje određenih algebarskih topoloških struktura kao što su diferencijalno glatke mnogostrukosti (eng. manifold) i ekvivalentnosti sistema. U ovom radu će se koristiti pregled osnovnih definicija i pojmova flat sistema koji proističu iz teorije diferencijalne algebre.

Dinamički sistem se definiše kao sistem od q jednačina:

$$S_i(z, \dot{z}, \dots, z^{(\sigma_i)}) = 0, \quad i = 1, \dots, q \quad (2.1)$$

gdje skup $z = (z_1, z_2, \dots, z_s)$ predstavlja varijable sistema. Sistem je diferencijalno ravan (eng. flat) ako postoji skup varijabli $y = (y_1, y_2, \dots, y_m)$ takvih da je:

$$y_i = \phi_i(z, \dot{z}, \dots, z^{(\alpha_i)}), \quad i = 1, \dots, m \quad (2.2)$$

i pri tome moraju biti zadovoljena dva uslova:

1. komponente vektora y su diferencijalno nezavisne, tj. ne postoji diferencijalna jednačina oblika:

$$R(y, \dot{y}, \dots, y^{(\beta)}) = 0 \quad (2.3)$$

2. sve varijable $z = (z_1, z_2, \dots, z_s)$ se mogu izraziti preko $y = (y_1, y_2, \dots, y_m)$ i konačnog broja njihovih izvoda:

$$z_i = \psi(y, \dot{y}, \dots, y^{(\gamma_i)}), \quad i = 1, \dots, s \quad (2.4)$$

Ako je sistem flat, bilo koji skup varijabli y koji zadovoljava uslove (2.3) i (2.4) predstavlja flat izlaz¹ sistema. Prvi uslov, diferencijalna nezavisnost, omogućava nam izbor proizvoljnih vremenskih trajektorija za sve komponente flat izlaza. Drugi uslov omogućava nam proračun vremenskih trajektorija svih varijabli sistema preko trajektorija flat izlaza. To se posebno odnosi na proračun upravljačkih ulaza.

¹ Izraz “flat izlaz” se ne odnosi na pojedinačnu varijablu, već na čitav skup varijabli, tj. vektor. Isto će se primjeniti prilikom definicije upravljačkog ulaza i stanja sistema.

Prema tome, flat sistem se često definiše kao sistem kojeg karakteriše kompletna, konačna i proizvoljna diferencijalna parametrizacija, sa y kao parametrom. Epitet kompletna znači da su sve varijable sistema definisane preko flat izlaza, a konačnost se odnosi na broj komponenti flat izlaza koji je konačan. Proizvoljna parametrizacija znači da su trajektorije komponenti flat izlaza nezavisne jedna od druge, pa ih je moguće proizvoljno izabrati. Na kraju, parametrizacija je diferencijalna zato što se koristi samo operacija diferenciranja, tj. nema integraljenja.

Još jedna osobina flat sistema je da flat izlaz nije jedinstven, tj. postoji beskonačan broj flat izlaza koje je moguće pronaći za dati sistem. Postavlja se pitanje kako izabrati pravi flat izlaz. To zavisi od specifičnosti datog sistema, pa se flat izlaz y bira u zavisnosti od fizičke interpretacije, simetrije modela, lakših proračuna, itd. Takođe, pri bilo kom izboru flat izlaza, broj komponenti je uvijek isti i jednak m . Ovaj broj ustvari predstavlja broj slobodnih varijabli za koje je moguće izvršiti parametrizaciju sistema. To znači da ako se sistem sastoji od q nezavisnih jednačina i s varijabli, broj komponenti flat izlaza predstavlja razliku ova dva broja:

$$m = s - q \quad (2.5)$$

Iz svih prethodno navedenih definicija se vidi da se upravljanje zasnovano na flatness osobini, može primjeniti samo na flat sisteme. Ipak, većina praktičnih sistema je flat, tako da postoji veliki broj primjena za ovu klasu sistema.

2.2. Definicija upravljačkog ulaza

Kako bi uticali na ponašanje sistema potrebno je iz sistemskih varijabli izabrati upravljački ulaz. Dakle, upravljački ulaz je moguće predstaviti vektorom $u = (u_1, \dots, u_m)$ koji zadovoljava sledeću relaciju:

$$u = \psi(z, \dot{z}, \ddot{z}, \dots, z^{(\gamma)}) \quad (2.6)$$

Takođe, upravljački ulaz treba da zadovoljava sledeće uslove:

1. da je diferencijalno nezavisan, tj. ne postoji diferencijalna relacija oblika:

$$R(u, \dot{u}, \dots, u^{(\beta)}) = 0 \quad (2.7)$$

2. da svaka varijabla sistema z_i , $i = 1, \dots, s$ zadovoljava implicitnu diferencijalnu jednačinu oblika:

$$E_i(z_i, \dot{z}_i, \dots, z_i^{(\delta_i)}, u, \dot{u}, \dots, u^{(\varepsilon_i)}) = 0 \quad (2.8)$$

Uslov (2.7), koji važi i za flat izlaz, znači da trajektorije komponenti ulaza mogu biti izabrane proizvoljno i nezavisno jedna od druge. Uslov (2.8) znači da kada izaberemo trajektorije za u , ponašanje sistema zavisi od konačnog broja izvoda z i početnih uslova. Ove uslove je generalno teško provjeriti. Međutim, ako se sistem predstavi u obliku $\dot{x} = f(x, u)$, može se uočiti da dati sistem uvijek zadovoljava ove uslove.

Skup varijabli sistema $z = (z_1, z_2, \dots, z_s)$ se može razdvojiti na dva podskupa $z = (\zeta, u)$, gdje je $\zeta = (\zeta_1, \dots, \zeta_p)$ i $u = (u_1, \dots, u_m)$, što znači da je $s = p + m$. Prema tome, sistem (2.1) se može preformulisati:

$$\tilde{S}_i(\zeta, \dot{\zeta}, \dots, \zeta^{(\beta_i)}, u, \dot{u}, \dots, u^{(\gamma_i)}) = 0, \quad i = 1, \dots, q \quad (2.9)$$

Kod flat sistema broj komponenti upravljačkog ulaza je jednak broju komponenti flat izlaza, tj. predstavlja razliku nezavisnih jednačina i varijabli sistema.

2.3. Definicija stanja

U slučaju linearnog sistema možemo uvijek uvesti stanje (vektor) x kako bi dobili model u prostoru stanja prvog reda:

$$\dot{x} = Ax + Bu \quad (2.10)$$

dok se to za slučaj nelinearnog sistema generalizuje:

$$\dot{x} = f(x, u) \quad (2.11)$$

Početni uslovi za x se mogu izabrati proizvoljno i rješenje se dobija integracijom već izabranih trajektorija. Međutim, u opštem slučaju, u izrazu (2.11), može se javiti i zavisnost od vremenskih izvoda upravljačkog ulaza u . Takva struktura se naziva generalizovana struktura stanja. Generalizovano stanje se definiše kao vektor $x = (x_1, \dots, x_n)$ koji je funkcija od:

$$x_i = \psi_i(\zeta, \dot{\zeta}, \dots, \zeta^{(\delta_i)}, u, \dot{u}, \dots, u^{(\varepsilon_i)}), \quad i = 1, \dots, n \quad (2.12)$$

i zadovoljava sledeće uslove:

1. komponente od x su nezavisne, tj. ne zadovoljavaju nijednu relaciju oblika:

$$\Phi(x, u, \dot{u}, \dots, u^{(\lambda)}) = 0 \quad (2.13)$$

2. Prvi izvod generalizovanog stanja zadovoljava relaciju oblika:

$$\dot{x}_i = f_i(x, u, \dot{u}, \dots, u^{(\alpha)}), \quad i = 1, \dots, n \quad (2.14)$$

U slučaju da se izvodi od u ne pojavljuju u ovim jednačinama, tj. da je $\alpha_i = 0, i = 1, \dots, n$, takva struktura stanja se naziva klasičnom. Broj komponenti stanja se često naziva dimenzijom stanja.

Koncept generalizovanog stanja se može razumjeti na sledeći način. Pod pretpostavkom da je ulaz u poznat, skup (x, u) koji se sastoji od n stanja i m ulaza je dovoljan da se opiše ponašanje sistema, tj. nije potrebna nijedna dodatna varijabla.

Postoji beskonačno mnogo različitih stanja za dati ulaz. Ako se definiše alternativno stanje sistema za isti ulaz:

$$\tilde{x} = (\tilde{x}_1, \dots, \tilde{x}_n) \quad (2.15)$$

onda komponente od $\tilde{x}$ zadovoljavaju transformaciju oblika:

$$\tilde{x}_i = \tau_i(x, u, \dot{u}, \dots, u^{(\gamma)}), \quad i = 1, \dots, n \quad (2.16)$$

Slično, za komponente od x važi:

$$x_i = \tilde{\tau}_i(\tilde{x}, u, \dot{u}, \dots, u^{(\gamma)}), \quad i = 1, \dots, n \quad (2.17)$$

što je ustvari inverzna transformacija. Ako se u relacijama (2.16) i (2.17) ne pojavljuje ulaz u i njegovi izvodi onda je takva transformacija klasična.

2.4. Kanonski oblik za flat sisteme

Za sistem u obliku generalizovanih stanja:

$$\dot{x}_i = f_i(x, u, \dot{u}, \dots, u^{(\alpha)}), \quad i = 1, \dots, n \quad (2.18)$$

definicija flatness-a se svodi na postojanje funkcija ψ_u i ψ_x :

$$u = \psi_u(y, \dot{y}, \dots, y^{(\gamma)}) \quad (2.19)$$

$$x = \psi_x(y, \dot{y}, \dots, y^{(\delta)}) \quad (2.20)$$

Takođe, ukoliko je sistem flat, onda komponente flat izlaza zadovoljavaju:

$$y_i = h_i(x, u, \dot{u}, \dots, u^{(\beta)}), \quad i = 1, \dots, m \quad (2.21)$$

Pošto flat sistem opisuje trajektorija flat izlaza, prirodno je izabrati komponente stanja od komponenti flat izlaza i njihovih izvoda. Ovim će se postići posebna struktura upravljanja koja se sastoji od lanaca integratora, koja generalizuje poznati kanonski oblik za linearne sisteme [20]-[23]. Postavlja se pitanje na koji način izabrati komponente stanja. To se može riješiti upotrebom dijagrama na slici 2.1.

korak 0 :	y_1	y_2	$\dots$	y_{r_k}	$\dots$	y_{r_1}	$\dots$	y_{r_0}	$\dots$	y_m
korak 1 :	$\dot{y}_1$	$\dot{y}_2$	$\dots$	$\dot{y}_{r_k}$	$\dots$	$\dot{y}_{r_1}$	$\dots$	$\dot{y}_{r_0}$	$\dots$	$\dot{y}_m$
$\vdots$	$\vdots$								$\dots$	$\vdots$
korak k :	$y_1^{(k)}$	$y_2^{(k)}$	$\dots$	$y_{r_k}^{(k)}$	$\dots$	$y_{r_1}^{(k)}$	$\dots$	$y_{r_0}^{(k)}$	$\dots$	$y_m^{(k)}$
korak k + 1 :	$y_1^{(k+1)}$	$y_2^{(k+1)}$	$\dots$	$y_{r_k}^{(k+1)}$	$\dots$	$y_{r_1}^{(k+1)}$	$\dots$	$y_{r_0}^{(k+1)}$	$\dots$	$y_m^{(k+1)}$
$\vdots$	$\vdots$									$\vdots$
	$y_1^{(n-1)}$	$y_2^{(n-1)}$	$\dots$	$y_{r_k}^{(n-1)}$	$\dots$	$y_{r_1}^{(n-1)}$	$\dots$	$y_{r_0}^{(n-1)}$	$\dots$	$y_m^{(n-1)}$

Slika 2.1 Procedura formiranja stanja selektovanjem flat izlaza

Na slici 2.1 varijable koje su „boldovane“ predstavljaju stanje sistema, što znači:

1. da u koraku „0“ biramo maksimalni broj (r_0) komponenti flat izlaza $\tilde{y}^0$:

$$\tilde{y}^0 = (y_1, \dots, y_{r_0}) \in (y_1, \dots, y_m) \quad (2.22)$$

koji zadovoljavaju uslov (2.13), tj. da mogu biti stanje sistema ukoliko ne zadovoljavaju:

$$\Phi(\tilde{y}^0, u, \dot{u}, \dots, u^{(\lambda)}) = 0 \quad (2.23)$$

2. da u koraku „k“ biramo maksimalni broj (r_k) k-tog izvoda komponenti flat izlaza $\tilde{y}^k$:

$$\tilde{y}^k = (y_1^{(k)}, \dots, y_{r_k}^{(k)}) \in (y_1^{(k)}, \dots, y_{r_{k-1}}^{(k)}) \quad (2.24)$$

koji ne zadovoljavaju relaciju oblika:

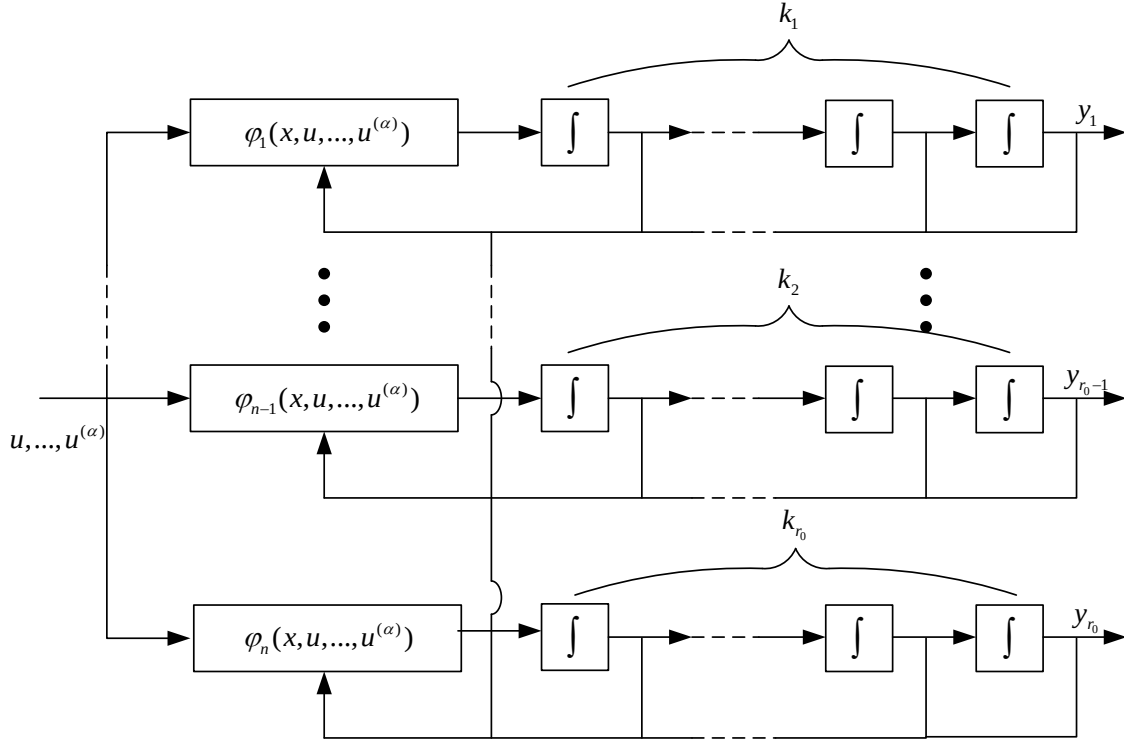
$$\Phi(\tilde{y}^k, \tilde{y}^{k-1}, \dots, \tilde{y}^0, u, \dot{u}, \dots, u^{(\lambda)}) = 0 \quad (2.25)$$

Prema tome, stanje sistema se sastoji od r_0 ($r_0 \leq m$) komponenti od y , zatim r_1 ($r_1 \leq r_0$) komponenti od $\dot{y}$, r_k ($r_k \leq r_{k-1}$) komponenti od $y^{(k)}$, itd. To znači da u svakom novom koraku biramo nove komponente na osnovu izvoda predhodnih. Ovakav sukcesivni postupak traje sve dokle god je moguće naći nove komponente stanja, a da relacija (2.25) ne bude zadovoljena. Dobijeno stanje se naziva *Brunovsky* stanje, zbog svoje sličnosti sa istoimenim stanjima kod kanonskog oblika linearnog kontrolabilnog sistema. Pod pretpostavkom da se komponenta y_1

zajedno sa svojim izvodima pojavljuje k_1 puta, komponenta y_2 sa svojim izvodima k_2 puta, i tako dalje do komponente y_{r_0} koja se pojavljuje k_{r_0} puta, broj stanja je $n = k_1 + k_2 + \dots + k_{r_0}$. Prema tome, *Brunovsky* stanje se može zapisati u obliku:

$$(x_1, x_2, \dots, x_n) = (y_1, \dot{y}_1, \dots, y_1^{(k_1-1)}, y_2, \dot{y}_2, \dots, y_2^{(k_2-1)}, \dots, y_{r_0}, \dot{y}_{r_0}, \dots, y_{r_0}^{(k_{r_0}-1)}) \quad (2.26)$$

Ovakav model stanja ima oblik kao na slici 2.2.



Slika 2.2 Kanoski blok dijagram za flat sisteme

Slika 2.2 predstavlja strukturu od n lanaca integratora, dužina $k_1, k_2, \dots, k_{r_0}$ respektivno, na čijim se ulaznim stranama nalaze nelinearne funkcije $\varphi_1, \varphi_2, \dots, \varphi_{r_0}$, koje zavise od stanja x , ulaza u i njegovih izvoda, što je posledica zadovoljenja relacije (2.25). Može se primjetiti da u slučaju linearnih sistema funkcije $\varphi_1, \varphi_2, \dots, \varphi_{r_0}$ predstavljaju linearnu kombinaciju stanja i ulaza, što odgovara kanonskom obliku za linearne sisteme. Na izlaznim stranama se nalaze flat izlazi $y_1, y_2, \dots, y_{r_0}$. Analitički se ovaj model može zapisati:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
&\vdots \\
\dot{x}_{k_1-1} &= x_{k_1} \\
\dot{x}_{k_1} &= \varphi_1(x, u, \dots, u^{(\alpha)}) \\
\dot{x}_{k_1+1} &= x_{k_1+2} \\
&\vdots \\
\dot{x}_{k_1+k_2-1} &= x_{k_1+k_2} \\
\dot{x}_{k_1+k_2} &= \varphi_2(x, u, \dots, u^{(\alpha)}) \\
&\vdots \\
&\vdots \\
\dot{x}_{k_1+k_2+\dots+k_{r_0-1}+1} &= x_{k_1+k_2+\dots+k_{r_0-1}+2} \\
&\vdots \\
\dot{x}_{n-1} &= x_n \\
\dot{x}_n &= \varphi_{r_0}(x, u, \dots, u^{(\alpha)})
\end{aligned} \tag{2.27}$$

2.5. Linearni flat sistemi

Ako je sistem linearan, može se opisati sistemom običnih diferencijalnih jednačina:

$$\sum_{i=1}^s \sum_{k=0}^{\sigma} p_{j,i,k} z_i^{(k)} = 0, \quad j = 1, \dots, q \tag{2.28}$$

gdje je $z = (z_1, \dots, z_s)$ skup sistemskih varijabli, a $p_{j,i,k}$ realni koeficijenti, tj. sistem je vremenski invarijantan. Relacija (2.28) se može zapisati na drugačiji način:

$$P\left(\frac{d}{dt}\right)z = 0 \tag{2.29}$$

gdje je $P\left(\frac{d}{dt}\right)$ matrica sa koeficijentima $p_{j,i,k}$ i operatorom izvoda $\frac{d}{dt}$. Sistem (2.29) je flat ako postoji $y = M\left(\frac{d}{dt}\right)z$, tako da su zadovoljeni sledeći uslovi:

$$1) \text{ ne postoji relacija oblika } r\left(\frac{d}{dt}\right)y = 0$$

2) postoji matrica $Q\left(\frac{d}{dt}\right)$ tako da je $z = Q\left(\frac{d}{dt}\right)y$

Ako su zadovoljeni gore navedeni uslovi, može se uočiti ekvivalentnost ovako definisanog sistema sa kontrolabilnim sistemom. Postojanjem diferencijalne zavisnosti flat izlaza, sistem bi postao autonoman, što znači da ne bi mogli uticati nijednim upravljačkim ulazom, a to je i odlika nekontrolabilnog sistema. Takođe, postojanje kanonske forme za linearne sisteme je potreban i dovoljan uslov za kontrolabilnost sistema, dok je izlaz na kraju lanaca integratora ustvari flat izlaz. Prema tome, dolazi se do veoma bitnog zaključka: linearni sistem je flat ako i samo ako je kontrolabilan [24].

Nelinearni flat sistem je moguće linearizovati oko referentne trajektorije. Pri tome, mora se voditi računa da se linearizacija vrši oko neregularne trajektorije tako da se dobije linearni vremensko-varijantni kontrolabilni sistem.

2.6. Planiranje trajektorije i feed-forward upravljanje

U sistemima upravljanja, često se želi postići da sistem na željeni način, pređe iz jednog stacionarnog režima u drugi. To su tzv. „rest to rest“ tranzicije. Primjeri ovakvog upravljanja su kretanje autonomnog vozila od početne do krajnje tačke po prethodno definisanoj putanji za zadato vrijeme, prenos tereta kod kranskih sistema, ubrzavanje DC motora do nove željene brzine, itd.

U opštem slučaju, u okolini različitih radnih tačaka, ponašanje sistema može biti značajno drugačije. Kao posledica toga, upotrebom aproksimativnog linearnog modela dobijenog pomoću lokalne linearizacije često je nedovoljno, pa nelinearne flatness tehnike dobijaju na prednosti.

U ustaljenom stanju svi izvodi su jednaki nuli, tj. za flat sistem stanje i ulaz se može zapisati:

$$x_s = \Psi_x(y, 0, \dots, 0) \quad (2.30)$$

$$u_s = \Psi_u(y, 0, \dots, 0) \quad (2.31)$$

Da bi se postiglo željeno ponašanje sistema, potrebno je definisati referentnu trajektoriju za flat izlaz, tj. :

$$y_r(t) = (y_{r,1}(t), \dots, y_{r,m}(t)) \quad (2.32)$$

pa se na upravljački ulazi mogu proračunati na osnovu relacije:

$$u_r(t) = \psi_u(y_r(t), \dot{y}_r(t), \dots, y_r^{(\gamma)}(t)) \quad (2.33)$$

Postavlja se pitanje kako izabrati referentnu trajektoriju za flat izlaz $y_r(t)$. Iz relacije (2.33) se može uočiti prvi uslov za takvu trajektoriju, a to je da komponente flat izlaza $y_{r,i}(t)$ moraju biti najmanje γ_i puta diferencijabilne, kako bi trajektorija upravljačkog ulaza bila kontinualna. Dodatna ograničenja za $y_r(t)$ mogu proisteći iz realnih problema pri upravljanju, kao što su fizička ograničenja, singulariteti, minimizacija energije, itd.

Za rest to rest trajektoriju flat izlaza, od značaja su početna i krajnja radna tačka u tranziciji, tj. potrebno je definisati vrijednosti flat izlaza u početnom ($t = 0$) i krajnjem trenutku ($t = t_f$). U tim trenucima svi izvodi flat izlaza su jednaki nuli. Diferencijalna nezavisnost flat izlaza sada dolazi do izražaja, jer trajektorije komponenti flat izlaza $y_{r,i}(t)$ mogu biti izabrane proizvoljno i nezavisno jedna od druge.

Bilo koja glatka (dovoljno diferencijabilna) funkcija može se izabrati za flat izlaz. Proračun je jednostavan ako se izabere polinom. U tom slučaju, određivanje koeficijenata iz početnih i krajnjih uslova za $y_r(t)$ se svodi na rješavanje linearnog sistema jednačina. Pod pretpostavkom da komponente flat izlaza $y_{r,i}(t)$ moraju biti diferencijabilne γ_i puta u trenucima $t = 0$ i $t = t_f$, onda se početni i krajnji uslovi mogu zapisati:

$$y_{r,i}(0) \quad \dot{y}_{r,i}(0) = \dots = y_{r,i}^{(\gamma_i)}(0) = 0 \quad (2.34)$$

$$y_{r,i}(t_f) \quad \dot{y}_{r,i}(t_f) = \dots = y_{r,i}^{(\gamma_i)}(t_f) = 0 \quad (2.35)$$

Prema tome, ukupno postoji $2\gamma_i + 2$ uslova koje moraju da ispune trajektorije za $y_{r,i}(t)$. To je ujedno i broj nepoznatih koeficijenata potrebnih za polinomijalne trajektorije, što znači da su to polinomi stepena $2\gamma_i + 1$:

$$y_{r,i}(t) = c_{i,0} + c_{i,1}t + c_{i,2}t^2 + \dots + c_{i,2\gamma_i+1}t^{2\gamma_i+1} \quad (2.36)$$

Uvrštavanjem uslova (2.34) u (2.36) dobijaju se nepoznati koeficijenti $c_{i,0}, \dots, c_{i,\gamma_i}$:

$$c_{i,0} = y_{r,i}(0) \quad c_{i,1} = \dots = c_{i,\gamma_i} = 0 \quad (2.37)$$

pa relacija (2.36) dobija oblik:

$$y_{r,i}(t) = y_{r,i}(0) + c_{i,\gamma_i+1}t^{\gamma_i+1} + \dots + c_{i,2\gamma_i+1}t^{2\gamma_i+1} \quad (2.38)$$

Uvrštavanjem uslova (2.35) u (2.38) dobija se linearni sistem od $\gamma_i + 1$ jednačina:

$$\begin{aligned}
y_{r,i}(t_f) - y_{r,i}(0) &= c_{i,\gamma_i+1}t_f^{\gamma_i+1} + \dots + c_{i,2\gamma_i+1}t_f^{2\gamma_i+1} \\
0 &= (\gamma_i+1)c_{i,\gamma_i+1}t_f^{\gamma_i} + \dots + (2\gamma_i+1)c_{i,2\gamma_i+1}t_f^{2\gamma_i} \\
&\vdots \\
0 &= (\gamma_i+1)!c_{i,\gamma_i+1}t_f + \dots + \frac{(2\gamma_i+1)!}{(\gamma_i+1)!}c_{i,2\gamma_i+1}t_f^{\gamma_i+1}
\end{aligned} \tag{2.39}$$

odakle se proračunavaju nepoznati koeficijenti $c_{i,\gamma_i+1}, \dots, c_{i,2\gamma_i+1}$. Ako se uvedu novi parametri:

$$a_{j-(\gamma_i+1)} = \frac{c_{i,j}t_f^j}{y_{r,i}(t_f) - y_{r,i}(0)}, \quad j = \gamma_i+1, \dots, 2\gamma_i+1 \tag{2.40}$$

relacija (2.38) dobija pogodniji oblik:

$$y_{r,i}(t) = y_{r,i}(0) + (y_{r,i}(t_f) - y_{r,i}(0)) \frac{t^{\gamma_i+1}}{t_f^{\gamma_i+1}} (a_0 + a_1 \frac{t}{t_f} \dots + a_{\gamma_i} \frac{t^{\gamma_i}}{t_f^{\gamma_i}}) \tag{2.41}$$

odakle se koeficijenti $a_0, \dots, a_{\gamma_i}$ dobijaju rješavanjem sistema jednačina:

$$\begin{aligned}
1 &= a_0 + a_1 + \dots + a_{\gamma_i} \\
0 &= (\gamma_i+1)a_0 + \dots + (2\gamma_i+1)a_{\gamma_i} \\
&\vdots \\
0 &= (\gamma_i+1)!a_0 + \dots + \frac{(2\gamma_i+1)!}{(\gamma_i+1)!}a_{\gamma_i}
\end{aligned} \tag{2.42}$$

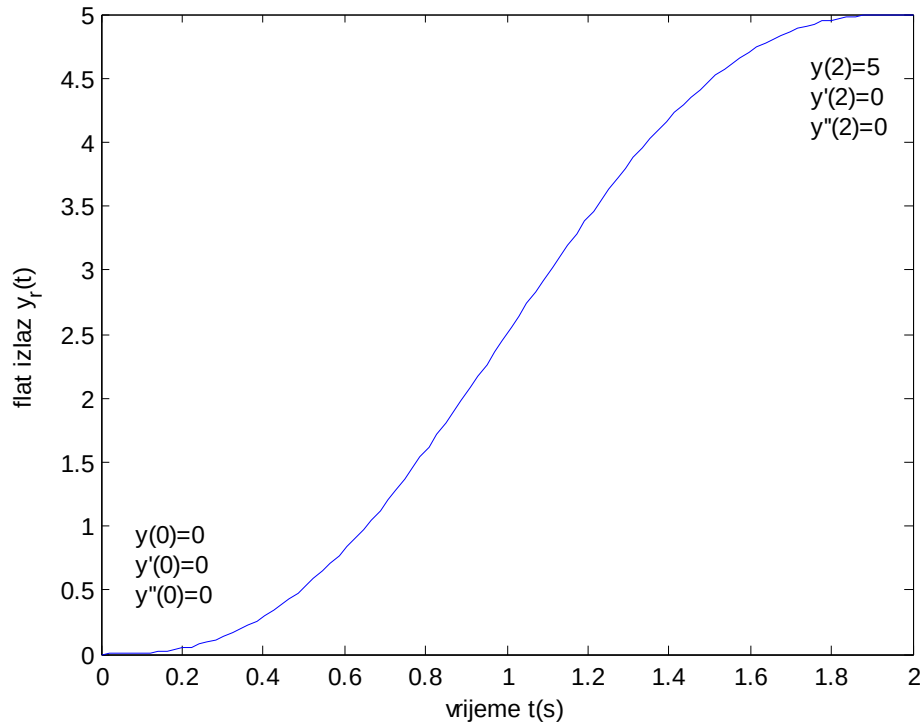
Prema tome, poznavajući početnu i krajnju vrijednost flat izlaza $(y_{r,i}(0), y_{r,i}(t_f))$, pomoću relacija (2.41) i (2.42) dobija se referentna trajektorija za bilo koji γ_i puta diferencijabilni flat izlaz. Time je određena i referentna trajektorija za upravljački ulaz preko relacije (2.33). Na primjer, referentna trajektorija flat izlaza za sistem sa jednim ulazom $u = f(y, \dot{y}, \ddot{y})$ mora biti najmanje dva puta diferencijabilna ($\gamma = 2$), pa bi se koeficijenti iz (2.40) dobili na osnovu (2.42) rješavanjem sistema:

$$\begin{aligned}
1 &= a_0 + a_1 + a_2 \\
0 &= 3a_0 + 4a_1 + 5a_2 \\
0 &= 6a_0 + 12a_1 + 20a_2
\end{aligned} \tag{2.43}$$

odakle se dobija $a_0 = 10, a_1 = -15, a_2 = 6$. Ako se pretpostavi da će tranzicija trajati dvije sekunde ($t_f = 2$) i da je u početnom i krajnjem trenutku $y(0) = 0$ i $y(2) = 5$, na osnovu (2.41) se dobija referentna trajektorija flat izlaza:

$$y_r(t) = 5 \frac{t^3}{t_f^3} \left(10 - 15 \frac{t}{t_f} + 6 \frac{t^2}{t_f^2} \right) \quad (2.44)$$

Na slici je prikazan oblik trajektorije flat izlaza iz (2.44).



Slika 2.3 Referentna trajektorija flat izlaza

2.7. Projektovanje povratne sprege

Feed-forward upravljanje neće biti uvijek dovoljno da se ispuni zadovoljavajuće praćenje putanje, tj. trajektorije. To se posebno odnosi na primjene za sisteme koji su nestabilni i podložni varijacijama parametara i smetnjama. Prema tome, potrebno je projektovati odgovarajuću povratnu spregu kako bi se postiglo da se ponašanje stvarnog sistema u vremenu podudara sa željenim modelovanim sistemom, tj. da stvarne trajektorije ne odstupaju od zadatih. Povratna sprega trebala bi da omogući stabilizaciju sistema i efikasno praćenje trajektorije, čak i u slučaju kada početni uslovi razlikuju od planiranih, kao i kada se model sistema ne poklapa sa stvarnim sistemom. Projektovanje povratne sprege zahtjeva informaciju o stanju, tj. potrebno je mjeriti ili estimirati stanje sistema, kako bi se proračunao upravljački ulaz.

Ukoliko je sistem flat, onda je za stanje sistema moguće izabrati Brunovsky stanja, tj. pojedine komponente flat izlaza i njihove izvode. Jedan od načina da se projektuje povratna sprega po stanjima jeste da se izabere željena dinamika greške praćenja referentne trajektorije, $e = y - y_r$. Na primjer, za sistem sa ulazom $u = f(y, \dot{y}, \ddot{y})$ dinamika greške se može zapisati:

$$\ddot{e} + K_1 \dot{e} + K_0 e = 0 \quad (2.45)$$

ili

$$(\ddot{y} - \ddot{y}_r) + K_1(\dot{y} - \dot{y}_r) + K_0(y - y_r) = 0 \quad (2.46)$$

odakle je potrebno izabrati koeficijente K_0 i K_1 tako da greška asimptotski konvergira ka nuli.

U robotici postoji sličan metod za upravljanje momentom (computed torque control). Za robotski manipulator sa generalizovanim koordinatama $q = (q_1, \dots, q_m)$ koje opisuju mehaničke stepene slobode kretanja zajedno sa motorom za svaki stepen slobode, sistem je „potpuno pogonjen“ i jednačine modela imaju standardnu formu:

$$M(q)\ddot{q} + g(q, \dot{q}) = N(q)\tau \quad (2.47)$$

U relaciji (2.47) $M(q)$ predstavlja matricu inercija, a u vektoru $g(q, \dot{q})$ su uračunate sve sile koje zavise od brzina i gravitacije. Eksterni upravljački momenti su sadržani u vektoru τ . Radi jednostavnosti za upravljački ulaz može se uzeti $u = N(q)\tau$. Sistem je flat sa flat izlazom $y = q$. Feed-forward upravljanje opisuje sledeća relacija:

$$u_r = M(y_r)\ddot{y}_r + g(y_r, \dot{y}_r) \quad (2.48)$$

Upravljački zakon (2.48) ne samo da zahtjeva savršeno modelovani sistem, već i da se početne pozicije i brzine poklapaju sa planiranim. Može se definisati povratna sprega sa greškom praćenja $e = q_i - q_{r,i}, i = 1, \dots, m$ tako da dinamika greške predstavlja asimptotski stabilni linearni oscilator drugog reda sa pogodnim izborom koeficijenata K_1 i K_2 :

$$\ddot{e} + K_1 \dot{e} + K_0 e = 0 \quad (2.49)$$

što se može preformulisati koristeći $y = q, y_r = q_r$:

$$\ddot{y} = \ddot{y}_r - K_1(\dot{y} - \dot{y}_r) - K_0(y - y_r) \quad (2.50)$$

Ako se uvede pomoćna varijabla v koja jednaka desnoj strani relacije (2.50), dobija se upravljanje:

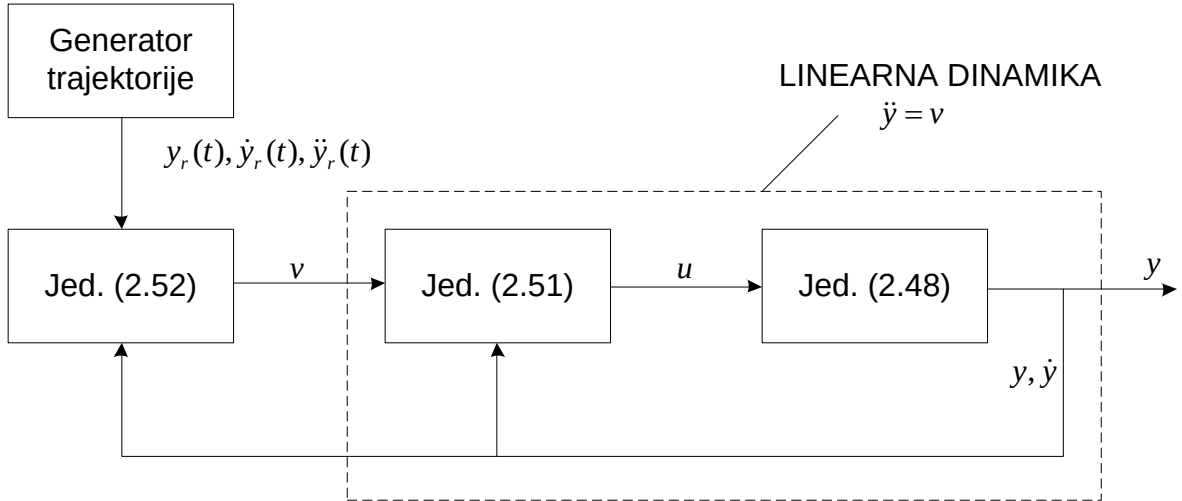
$$u = M(y)v + g(y, \dot{y}) \quad (2.51)$$

$$v = y_r - K_1(\dot{y} - \dot{y}_r) - K_0(y - y_r) \quad (2.52)$$

Varijabla v se naziva „novi ulaz“. Prvi dio upravljanja, relacija (2.51), kompenzuje originalnu nelinearnu dinamiku sistema, pri čemu dinamika sistema između flat izlaza i novog ulaza postaje linearna:

$$\ddot{y} = v \quad (2.53)$$

Na ovaj način je izvršena tzv. egzaktna linearizacija, tj. nelinearni sistem u zatvorenoj petlji je postao linearan. Relacija (2.53) može se predstaviti u obliku dva integratora u kaskadnoj vezi. Ovakav linearizovani sistem je modifikovan drugim dijelom upravljanja, relacija (2.52), kako bi se postigla željena dinamika greške. Ona je takođe linearna i sistem u zatvorenoj petlji će da konvergira prema referentnoj trajektoriji. Na slici 2.4 je prikazan blok dijagram sistema, gdje je isprekidanom linijom označen linearizovani sistem sa novim ulazom.



Slika 2.4 Flat sistem u povratnoj sprezi sa zahtjevom praćenja trajektorije

Upravljanje u zatvorenoj petlji se može generalizovati za sve flat sisteme. Prvo je potrebno parametrizovati upravljački ulaz $u = \Psi_u(y, \dot{y}, \dots, y^{(\nu)})$, a zatim izvršiti egzaktnu linearizaciju tako što je potrebno izabrati nove ulaze $v_i = y_i^{(\lambda_i)}, i=1, \dots, m$. Vrijednosti λ_i se najčešće biraju na osnovu (2.26), tako da je $\lambda_i = k_i$. Na taj način se dobija „kvazi-statična“ povratna sprega [25].

Drugi zadatak upravljanja je stabilizacija sistema oko referentne trajektorije i izbor dinamike greške praćenja. Pri tome se mogu koristiti razne metode iz teorije linearnih sistema upravljanja. Slično kao na primjeru manipulatora, potrebno je uvesti:

$$v_i = y_{r,i}^{(\lambda_i)} - \sum_{j=0}^{\lambda_i-1} a_{i,j} (y_i^{(j)} - y_{r,i}^{(j)}), \quad i=1, \dots, m \quad (2.54)$$

Relacijom (2.54) zadovoljava željenu dinamiku za m komponenti greške praćenja:

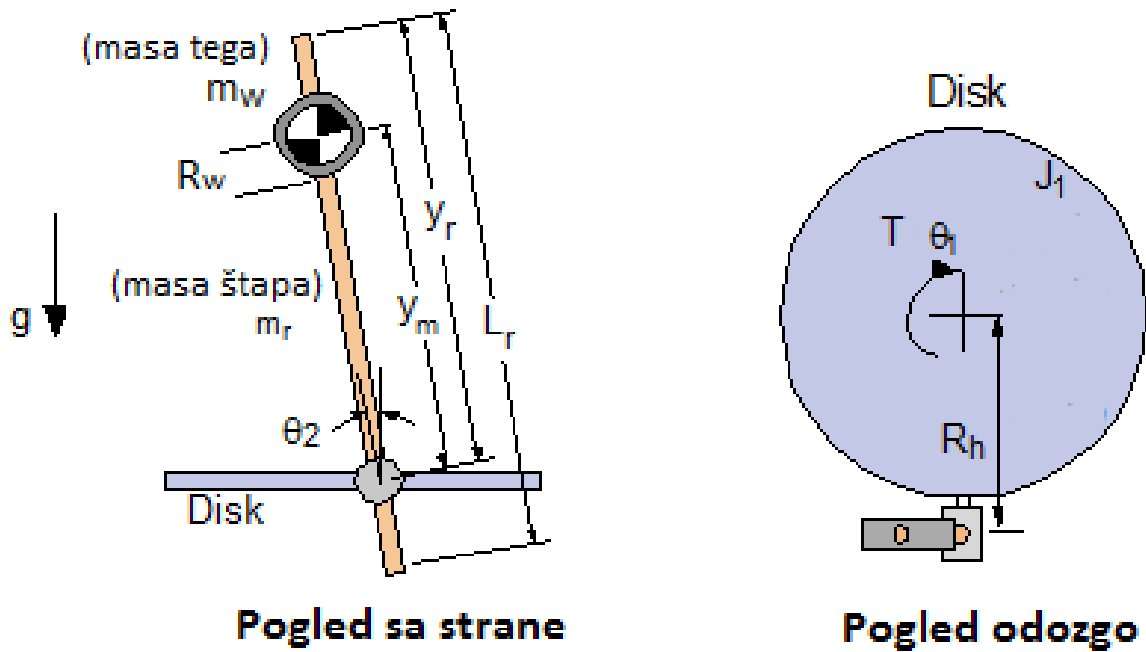
$$\sum_{j=0}^{\lambda_i} a_{i,j} e_i^{(j)} = 0, \quad i = 1, \dots, m \quad (2.55)$$

Dinamika greške je asimptotski stabilna ako se parametri $a_{i,j}$ izaberu tako da karakteristični polinom (sa koeficijentima $a_{i,j}$) sadrži korijene sa negativnim realnim djelom. Dodatno, izbor ovih koeficijenata često može da zavisi od dinamike sistema u otvorenoj petlji kao i ograničenja upravljačkog ulaza (aktuatora).

3. Matematički model Furuta klatna

3.1. Nelinearne jednačine dinamike Furuta klatna

Na slici 3.1 je prikazana skica Furuta klatna, na kojoj su označeni parametri neophodni za opisivanje dinamike sistema diferencijalnim jednačinama. Ugaone pozicije diska i klatna su θ_1 i θ_2 , respektivno. Klatno se sastoji od štapa mase m_r i tega mase m_w . Pomjeranjem tega dužinom štapa mijenja se težište klatna. Ukupna dužina štapa je L_r , dok su odgovarajuće dužine y_m i y_r , rastojanje od spoja na disku do tega i kraja štapa, respektivno. Rastojanje od centra diska do klatna je R_h , dok je J_1 moment inercije diska zajedno sa svim elementima koji se uniformno kreću sa diskom. Eksterni moment koji djeluje na disk je T .



Slika 3.1 Šema klatna na osnovu koje se izvodi model

Na osnovu ovih parametara, za lakši zapis modela biće izvedene i pomoćne varijable:

$$l_r = y_r - \frac{1}{2}L_r \quad (3.1)$$

$$l_w = y_m \quad (3.2)$$

$$m = m_r + m_w \quad (3.3)$$

$$l_{cg} = \frac{m_r l_r + m_w l_w}{m_r + m_w} \quad (3.4)$$

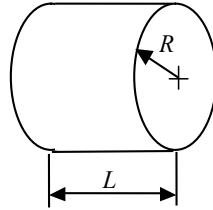
Takođe, izvešće se varijable J_x, J_y, J_z , koje predstavljaju odgovarajuće momente inercije klatna relativne prema njegovom centru mase (z – duž ose rotacije klatna, y – duž štapa klatna, x – normalan na y i z):

$$J_x = \frac{1}{12} m_r L_r^2 + \frac{1}{4} m_w R_w^2 \quad (3.5)$$

$$J_y = \frac{1}{4} m_w R_w^2 \quad (3.6)$$

$$J_z = \frac{1}{12} m_r L_r^2 + \frac{1}{2} m_w R_w^2 \quad (3.7)$$

gdje se indeksi “r” i “w” odnose na štap klatna i teg, respektivno, dok se parametri “L” i “R” odnose na dužinu i prečnik odgovarajućih cilindara, što se može uočiti naslici 3.2. Parametri R_r i L_w su veoma mali u poređenju sa ostalim članovima gore navedenih relacija, pa su izostavljeni.



Slika 3.2 Dužina i prečnik cilindričnih objekata

Matematički model izvodi se primjenom Euler-Lagrange-ovih jednačina kretanja za mehaničke sisteme. Analizirajući sliku 3.2, kinetička energija sistema sastoji se od:

1. kinetičke energije rotirajućeg diska:

$$E_{k1} = \frac{1}{2} J_1 \ddot{\theta}_1 \quad (3.8)$$

2. kinetičke energije centra mase klatna:

$$\begin{aligned} E_{k2} &= \frac{1}{2} m \left[(\dot{\theta}_1 R_h + \dot{\theta}_2 l_{cg} \cos \theta_2)^2 + (\dot{\theta}_2 l_{cg} \sin \theta_2)^2 + (\dot{\theta}_1 l_{cg} \sin \theta_2)^2 \right] \\ &= \frac{1}{2} m \left[(R_h^2 + l_{cg}^2 \sin^2 \theta_2) \dot{\theta}_1^2 + 2 R_h l_{cg} \dot{\theta}_1 \dot{\theta}_2 \cos \theta_2 + l_{cg}^2 \dot{\theta}_2^2 \right] \end{aligned} \quad (3.9)$$

3. kinetičke energije rotacije klatna relativne u odnosu na centar mase klatna:

$$E_{k3} = \frac{1}{2} J_z \dot{\theta}_2^2 + \frac{1}{2} J_x (\dot{\theta}_1 \sin \theta_2)^2 + \frac{1}{2} J_y (\dot{\theta}_1 \cos \theta_2)^2 \quad (3.10)$$

Potencijalna energija sistema je :

$$U = mgl \cos \theta_2 \quad (3.11)$$

Lagrange-ova funkcija je:

$$L = E_{k1} + E_{k2} + E_{k3} - U \quad (3.12)$$

pa se jednačine kretanja dobijaju na osnovu Euler-Lagrange-ove jednačine:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} = q_i, \quad i=1,2 \quad (3.13)$$

gdje q_i predstavljaju tzv. virtuelne sile, s tim da je $q_2 = 0$. Na osnovu (3.13) jednačine kretanja koje se dobijaju su:

$$\begin{aligned} & \ddot{\theta}_1 \left[J_1 + mR_h^2 + (J_x + ml_{cg}^2) \sin^2 \theta_2 + J_y \cos^2 \theta_2 \right] + \ddot{\theta}_2 mR_h l_{cg} \cos \theta_2 \\ & + \dot{\theta}_1 \dot{\theta}_2 (J_x + ml_{cg}^2 - J_y) \sin 2\theta_2 - \dot{\theta}_2^2 mR_h l_{cg}^2 \sin 2\theta_2 = T \end{aligned} \quad (3.14)$$

$$\begin{aligned} & \ddot{\theta}_1 mR_h l_{cg} \cos \theta_2 + \ddot{\theta}_2 (J_z + ml_{cg}^2) - \dot{\theta}_1^2 (J_x + ml_{cg}^2 - J_y) \sin 2\theta_2 \\ & + \dot{\theta}_1 \dot{\theta}_2 mR_h l_{cg} \sin \theta_2 - mgl_{cg} \sin \theta_2 = 0 \end{aligned} \quad (3.15)$$

Ukoliko se uvedu odgovarajući momenti:

$$\bar{J}_1 = J_1 + mR_h^2 \quad (3.16)$$

$$\bar{J}_x = J_x + ml_{cg}^2 \quad (3.17)$$

$$\bar{J}_z = J_z + ml_{cg}^2 \quad (3.18)$$

kao i modeluje viskozno trenje na disku sa koeficijentom c_1 , relacije (3.14) i (3.15) se zapisuju:

$$\begin{aligned} & \ddot{\theta}_1 \left[J_1 + mR_h^2 + \bar{J}_x \sin^2 \theta_2 + J_y \cos^2 \theta_2 \right] + \ddot{\theta}_2 mR_h l_{cg} \cos \theta_2 \\ & + \dot{\theta}_1 \dot{\theta}_2 (\bar{J}_x - J_y) \sin 2\theta_2 - \dot{\theta}_2^2 mR_h l_{cg}^2 \sin 2\theta_2 + c_1 \dot{\theta}_1 = T \end{aligned} \quad (3.19)$$

$$\begin{aligned} & \ddot{\theta}_1 mR_h l_{cg} \cos \theta_2 + \ddot{\theta}_2 \bar{J}_z - \dot{\theta}_1^2 (\bar{J}_x - J_y) \sin 2\theta_2 \\ & + \dot{\theta}_1 \dot{\theta}_2 mR_h l_{cg} \sin \theta_2 - mgl_{cg} \sin \theta_2 = 0 \end{aligned} \quad (3.20)$$

3.2. Linearizovani model Furuta klatna

Pod pretpostavkom da će se upravljanje vršiti za male pomjeraje diska i klatna, sistem se linearizuje oko radne tačke $\theta_1 = 0$ i $\theta_2 = 0$ i glasi:

$$\ddot{\theta}_1[\bar{J}_1 + J_y] + \ddot{\theta}_2 m R_h l_{cg} + c_1 \dot{\theta}_1 = T \quad (3.21)$$

$$\ddot{\theta}_1 m R_h l_{cg} + \ddot{\theta}_2 \bar{J}_z - m g l_{cg} \theta_2 = 0 \quad (3.22)$$

Parametar l_{cg} postaje negativan ukoliko y_m ili y_r postanu negativni. Pri tome se centar mase klatna premješta ispod tačke spoja klatna i diska, tj. klatno je u neinvertovanom položaju. Prema tome, jednačine (3.21) i (3.22) su uopštene za invertovani i neinvertovani slučaj.

Iz relacija (3.21) i (3.22), upotrebom Laplace-ove transformacije, dobijaju se prenosne funkcije $\theta_1(s)/T(s)$ i $\theta_2(s)/T(s)$:

$$\frac{\theta_1(s)}{T(s)} = \frac{\bar{J}_z s^2 - m g l_{cg}}{[\bar{J}_z(\bar{J}_1 + J_y) - (m R_h l_{cg})^2] s^4 + c_1 \bar{J}_z s^3 - [m g l_{cg}(\bar{J}_1 + J_y)] s^2 - m g l_{cg} c_1 s} \quad (3.23)$$

$$\frac{\theta_2(s)}{T(s)} = \frac{-m R_h l_{cg} s}{[\bar{J}_z(\bar{J}_1 + J_y) - (m R_h l_{cg})^2] s^3 + c_1 \bar{J}_z s^2 - [m g l_{cg}(\bar{J}_1 + J_y)] s - m g l_{cg} c_1} \quad (3.24)$$

Imenioci u relacijama (3.23) i (3.24) su identični osim što prva ima pol u nuli. Imaju jedan pol u lijevoj kompleksnoj poluravni za pozitivno l_{cg} , dok su za negativno l_{cg} svi polovi u lijevoj kompleksnoj poluravni. Za $l_{cg} = 0$, imenioci tj. karakteristična jednačina je ista kao u slučaju kratanja krutog tijela u viskoznoj sredini. Ukoliko bi koeficijent viskoznog trenja bio zanemaren ($c_1 = 0$), sistem bi imao imaginarne polove za l_{cg} negativno, tj. ponašao bi se kao oscilator, dok bi za l_{cg} pozitivno imao sve polove u nuli, tj. ponašao bi se kao kruto tijelo.

3.3. Model u prostoru stanja

Linearizovani model (3.21) i (3.22) se može prevesti u model u prostoru stanja. Iz (3.22), $\ddot{\theta}_2$ se zapisuje:

$$\ddot{\theta}_2 = \frac{m g l_{cg}}{\bar{J}_z} \theta_2 - \frac{m R_h l_{cg}}{\bar{J}_z} \ddot{\theta}_1 \quad (3.25)$$

Uvrštavanjem (3.25) u (3.21) dobija se izraz za $\ddot{\theta}_1$:

$$\ddot{\theta}_1 = \frac{1}{p} [-c_1 \dot{\theta}_1 \bar{J}_z - m^2 l_{cg}^2 R_h g \theta_2 + \bar{J}_z T_1] \quad (3.26)$$

gdje je:

$$p = \bar{J}_z (\bar{J}_1 + J_y) - (m R_h l_{cg})^2 \quad (3.27)$$

Zatim, uvrštavanjem (3.26) u (3.22) izraz za $\ddot{\theta}_2$:

$$\ddot{\theta}_2 = \frac{1}{p} [c_1 m R_h l_{cg} \dot{\theta}_1 - m g l_{cg} (\bar{J}_1 + J_y) \theta_2 - T_1 m R_h l_{cg}] \quad (3.28)$$

Ako se sada definiše vektor stanja:

$$\mathbf{x} = \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix} \quad (3.29)$$

relacije (3.26) i (3.28) se mogu predstaviti u obliku modela u prostoru stanja:

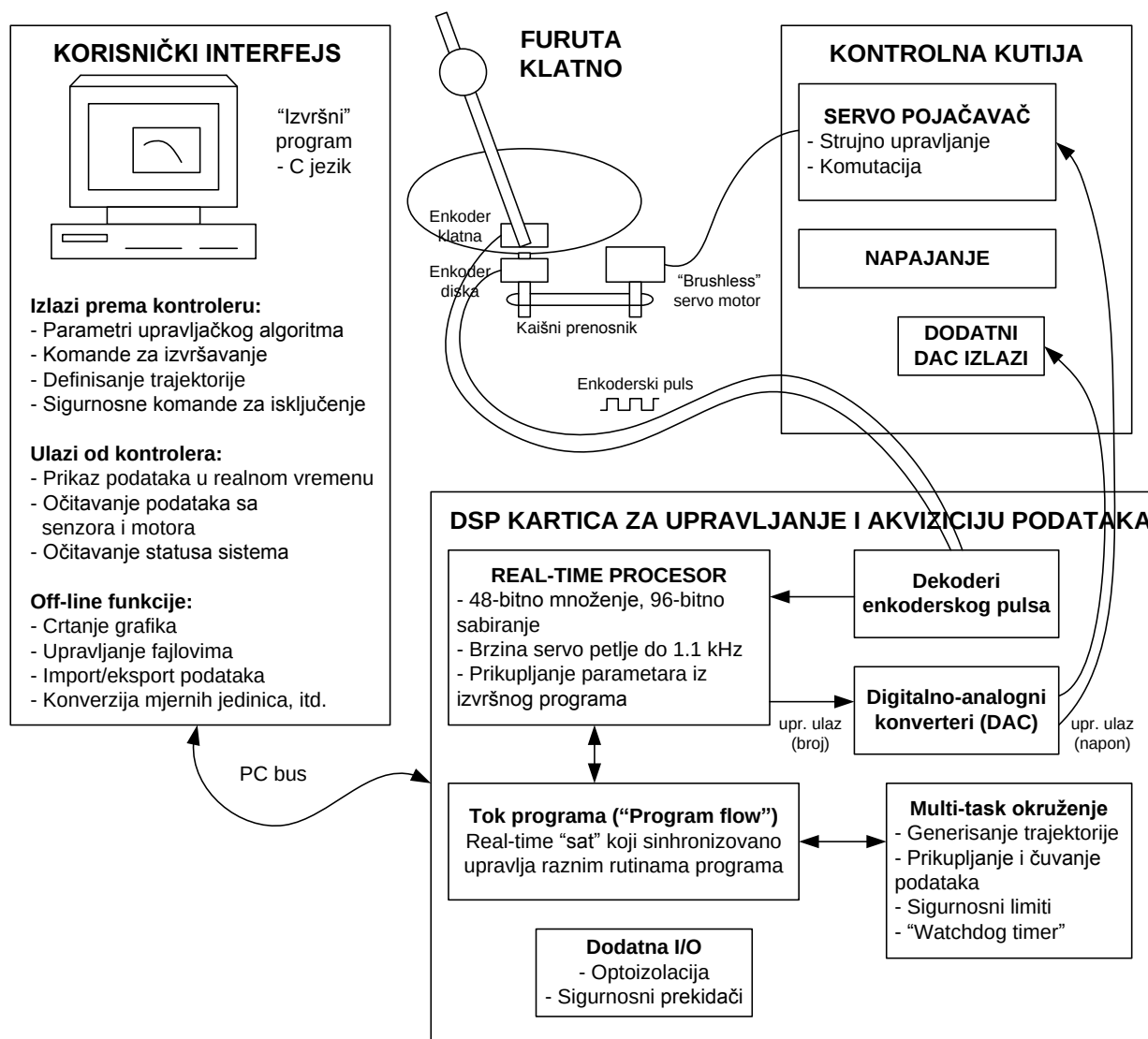
$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}T \quad (3.30)$$

gdje su:

$$\mathbf{A} = \frac{1}{p} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -c_1 \bar{J}_z & -m^2 l_{cg}^2 R_h g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & c_1 m R_h l_{cg} & m l_{cg} g (\bar{J}_1 + J_y) & 0 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ \bar{J}_z \\ 0 \\ -m R_h l_{cg} \end{bmatrix} \quad (3.31)$$

3.4. Laboratorijski model Furuta klatna

Laboratorijski model Furuta klatna ECP A51 na kojem će se izvesti eksperimenti sastoji se od tri podsistema (slika 3.3): elektromehaničkog (Furuta klatno, DC motor koji pogoni disk i enkoderi koji mjere ugaone pozicije diska i klatna), elektronskog (Real-Time kontrolera) i softverskog (izvršni program).



Slika 3.3 Struktura modela ECP A51

Arhitektura sistema na slici 3.3 se može objasniti tako što se prati tok podataka kada je sistem u operativnom stanju. Najprije, korisnik specificira upravljački algoritam u izvršnom programu i učitava ga u karticu za digitalnu obradu signala – DSP, preko komande „Implement Algorithm“. DSP kartica komunicira preko PC bus-a sa računarom. Real-time procesor unutar DSP kartice izvršava algoritam prema zadatom vremenu odabiranja. Ovo uključuje očitavanje referentnog ulaznog signala i vrijednosti senzora (optičkih enkodera), na osnovu kojih se onda izvršava algoritam rezultujući digitalnim upravljačkim signalom koji se prosleđuje digitalno-analognom konvertoru – DAC.

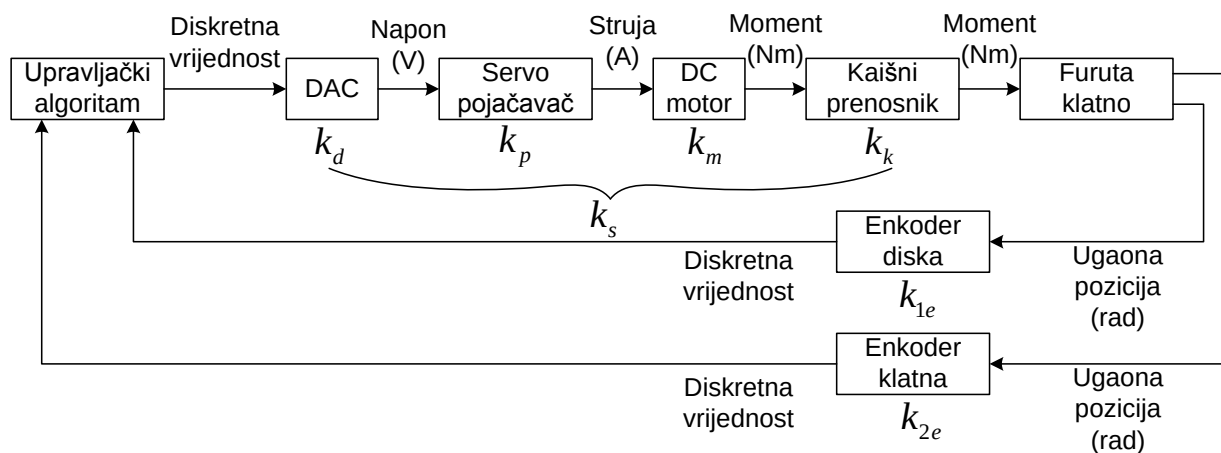
DAC konvertuje digitalne vrijednosti u analogni napon, koji se dalje transformiše u struju posredstvom servo pojačavača, koji se nalazi u posebnoj elektronskoj („kontrolnoj“) kutiji zajedno sa napojnim jedinicama i dodatnim DAC analognim izlazima. Servo pojačavač upravlja

motorom i daje momenat na osovini motora, koja je preko kaišnog prenosnika povezana sa osovinom diska. Kretanje diska i klatna se mjeri pomoću dva optička enkodera, koji se nalaze na osovini diska i neposredno uz klatno, respektivno. Ovi enkoderi generišu podatke u vidu pulseva. Pulsevi se dekodiraju u dekoderu na DSP kartici, tj. prevode u digitalnu riječ koja predstavlja poziciju.

Kada korisnik definiše trajektoriju i pokrene komandu „Execute“, parametri te trajektorije se upisuju u DSP kartici, koja u realnom vremenu koristi te vrijednosti za proračune algoritma. Za vrijeme ovog postupka, sva mjerenja i podaci koje je korisnik želio da snimi se čuvaju u internoj memoriji na DSP kartici. Kada se trajektorija izvrši ovi podaci se prosleđuju računaru i pogodni su za crtanje grafika i druge operacije.

Proračun upravljačkog algoritma se vrši za vrijeme odabiranja T_s . Ovo vrijeme može biti podešeno najkraće 0.000884 sekunde (1.1 kHz) ili kao umnožak ovog broja. Proračunati upravljački ulaz je ograničen u softveru na ± 16384 opseg koji predstavlja $\pm 5V$ na izlazu DAC. Dodatni DAC analogni izlazi omogućavaju korisniku mjerenja raznih signala upotrebom osciloskopa ili druge analogne opreme. Za izvršavanje svih programskih operacija u okviru Real-time procesora postoji hijerarhijska struktura. Za to je zaduženo multi-tasking okruženje, koje preko real-time sata izvršava operacije prema prioritetima. Najveći priortet ima izvršavanje upravljačkog algoritma, a zatim slijedi planiranje trajektorije, korisnički interfejs, očitavanje dodatnih DAC izlaza, itd.

Dinamička šema sistema u zatvorenoj petlji sa elektronskim i aktuatorskim komponentama prikazana je na slici 3.4. Sve ove komponente karakteriše određena dinamika koja je višestruko brža od mehaničkog sistema Furuta klatna, tako da će prenosna funkcija svake komponente biti modelovana sa određenim koeficijentom pojačanja. U tabeli 1 prikazani su parametri Furuta klatna, a u tabeli 2 parametri elektronskih i aktuatorskih komponenti laboratorijskog modela.



Slika 3.4 Blok dijagram modela ECP A51

Tabela 1. Parametri mehaničkog sistema ECP A51

Simbol	Vrijednost	Jedinica
m_r	0.069	kg
m_w	0.089	kg
L_r	0.43	m
y_m	0.32	m
y_r	0.42	m
R_w	0.05	m
R_h	0.25	m
J_1	0.0166	kgm ²
c_1	0.02	kgm ² /s

Tabela 2. Parametri elektronskih i aktuatorskih komponenti ECP A51

Simbol	Vrijednost	Jedinica
k_d	10/32768	diskr. vr.
k_p	2	A/V
k_m	0.1	Nm/A
k_k	3	Nm/Nm
k_{1e}	2546	Nm/diskr. vr.
k_{2e}	2608	Nm/diskr. vr.

Prošireni model Furuta klatna u otvorenoj petlji se dobija na osnovu relacija (3.21) i (3.22) tako što se varijable θ_1, θ_2, T prevode u $\theta_{1e}, \theta_{2e}, u$ sa transformacijom:

$$\theta_1 = \frac{\theta_{1e}}{k_{1e}} \quad (3.32)$$

$$\theta_2 = \frac{\theta_{2e}}{k_{2e}} \quad (3.33)$$

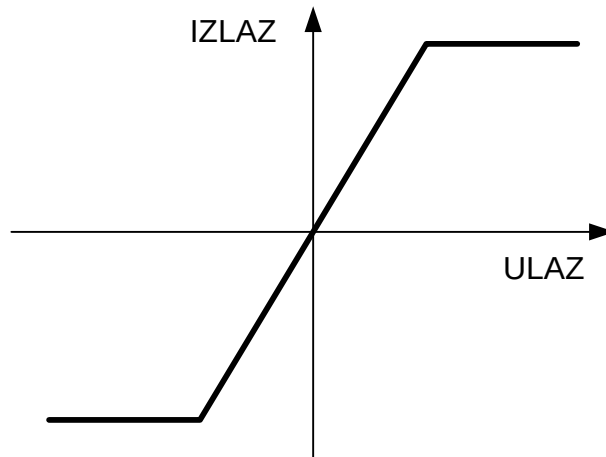
$$T = k_d k_p k_m k_k = k_s u \quad (3.34)$$

gdje su varijable $\theta_{1e}, \theta_{2e}, u$ izražene u obliku diskretnih vrijednosti (“counts”). Isti postupak se primjenjuje na model u prostoru stanja (3.30) i (3.31).

3.5. Efekti saturacije, diskretnog vremenskog odabiranja i senzorske kvantizacije

U ovoj sekciji biće opisano neidealno ponašanje sistema koje se javlja u raznim primjenama automatskog upravljanja, kao i načini kako ublažiti te efekte. Tu spadaju: saturacija pogona i aktuatora, odabiranje diskretnog vremena i senzorska kvantizacija.

Saturacija (zasićenje) uređaja predstavlja zadržavanje iste vrijednosti izlaza, pozitivne ili negativne, za sve vrijednosti ulaza ispod ili iznad nekog limita. To je predstavljeno na slici 3.5 koja pokazuje ulazno-izlaznu zavisnost za linearni sistem.



Slika 3.5 Primjer saturacije u zavisnosti od ulaza

Uzrok saturacije može biti prekoračenje upravljačkog signala iz DAC, servo pojačavača, kao i magnetnog polja aktuatora. U praksi su pojačavač i aktuator često izabrani da podnesu čitav opseg napona iz DAC ili je kontrolna elektronika programirana tako da izlaz iz DAC uđe u zasićenje prije nego upravljačke veličine drugih komponenti i time spriječe njihovu saturaciju ili oštećenje.

Sa modernim procesorskim jedinicama visoke frekvencije, vremensko odabiranje signala je toliko brzo da upravljačke strukture mogu biti analizirane i implementirane koristeći metode u kontinualnom vremenu. Često se kao zadatak upravljanja postavlja brzina odziva što može da preoptereći upravljački procesor, ili su zbog ekonomskih razloga izabrani „sporiji“ procesori tako da efekti diskretnog vremenskog odabiranja signala mogu postati značajni.

Od posebnog interesa je fazni pomjeraj (u frekventnom domenu) uzrokovao vremenskim kašnjenjem koje se javlja kod diskretizovanih sistema. Vremensko kašnjenje nastaje prilikom odabiranja signala iz senzora i zadržavanja njihove trenutne vrijednosti u ADC (analogno-digitalnom konverteru). Takođe, kašnjenja se javljaju i prilikom DAC konverzije upravljačkog signala, kao i u analognom servo pojačavaču. U Laplace-ovom domenu kašnjenje se zapisuje:

$$\phi(s) = e^{-sT_s} \quad (3.35)$$

tako da je fazni pomjeraj:

$$\arg\{\phi(j\omega)\} = -T_s\omega \quad (3.36)$$

Efekti numeričke kvantizacije postoje u svim upravljačkim sistemima koji sadrže digitalne elemente. Ovi efekti, poznati kao efekti konačne dužine riječi, uključuju kvantizaciju parametara i koeficijenata, DAC i ADC rezoluciju, dinamički opseg signala i efekat zaokruživanja. Oni mogu da izazovu duboke posledice na stabilnost i performanse sistema.

Kod laboratorijskog modela Furuta klatna efekat kvantizacije nastaje kod optičkih enkodera. Diskretne vrijednosti izmjerene pozicije često se koriste za proračun brzine na sledeći način:

$$\omega_n = \frac{\theta_n - \theta_{n-1}}{T_s} \quad (3.37)$$

gdje θ_n predstavlja trenutno, a θ_{n-1} prethodno mjerenje pozicije. Ukoliko se sa $\Delta\theta$ označi rezolucija mjerenja enkodera (0.0225° za enkoder diska i 0.0222° za enkoder klatna) onda je inkrement brzine (ili kvant), u oznaci $\Delta\omega$, jednak:

$$\Delta\omega = \frac{\Delta\theta}{T_s} \quad (3.38)$$

Inkrement upravljačkog momenta $T_{\Delta\omega}$ je proporcionalan inkrementu brzine:

$$T_{\Delta\omega} = k \frac{\Delta\theta}{T_s} \quad (3.39)$$

Za velike vrijednosti $T_{\Delta\omega}$, čak najmanji enkoderski puls može uzrokovati velike vrijednosti upravljačkog momenta u određenom trenutku. Ovo dalje može izazvati oscilacije mjerene pozicije, tj. sistem se ponaša kao da na njega djeluje šum. Na osnovu (3.39) kvantizacija momenta se povećava sa lošijom rezolucijom enkodera. Takođe, kvantizacija momenta i šum se povećavaju sa povećanjem brzinskog koeficijenta i smanjenjem perioda odabiranja.

4. Linearni kvadratni regulator - LQR

Teorija optimalnog upravljanja teži da dinamiku sistema minimizuje prema odabranom kriterijumu. Linearni kvadratni regulator (LQR) je dobro ustanovljena tehnika za upravljanje i stabilizaciju multivarijabilnih sistema u zatvorenoj petlji. Ovdje će biti ukratko opisan postupak za rješavanje optimalnog problema [15], kao i neke od karakteristika ovakvog tipa upravljanja.

4.1. Projektovanje LQR kontrolera

Za izvođenje linearnog kvadratnog regulatora smatra se da je sistem linearan ili predstavljen u linearizovanoj formi:

$$\dot{x} = Ax + Bu \quad (4.1)$$

zajedno sa početnim uslovom $x(0) = x_0$. Uopštena procedura za rješavanje optimalnog problema koristi varijacioni račun. Formulacija problema se sastoji u minimizaciji indeksa performanse:

$$J = \frac{1}{2} \int_0^\infty (x^T Q x + u^T R u) dt \quad (4.2)$$

gdje će množenje sa 1/2 dati pogodniji oblik rješenja optimalnog problema. Matrice Q i R su po pretpostavci pozitivno definitne i predstavljaju težinske faktore kojima se opisuju željena brzina odziva sistema, kao i upravljačka akcija. Ove matrice se često biraju kao dijagonalne.

Indeks performanse, tj. kriterijumska funkcija se može preformulisati:

$$J = \frac{1}{2} \int_0^{t_f} (x^T Q x + u^T R u + \lambda^T (Ax + Bu - \dot{x})) dt \quad (4.3)$$

gdje $\lambda(t)$ može biti proizvoljno izabran jer je $Ax + Bu - \dot{x} = 0$ prema (4.1). Prva varijacija od integrala J je:

$$\delta J = \int_0^{t_f} (x^T Q \delta x + u^T R \delta u + \lambda^T A \delta x + \lambda^T B \delta u - \lambda^T \delta \dot{x}) dt \quad (4.4)$$

pri čemu se poslednji član integrala, koristeći parcijalnu integraciju, može zapisati:

$$-\int_0^{t_f} \lambda^T \delta \dot{x} dt = -\lambda^T(t_f) \delta x(t_f) + \lambda^T(0) \delta x(0) + -\int_0^{t_f} \dot{\lambda}^T \delta x dt \quad (4.5)$$

Uvrštavanjem (4.5) u (4.4) dobija se izraz:

$$\delta J = \int_0^{t_f} (u^T R + \lambda^T B) \delta u dt + \int_0^{t_f} (x^T Q + \lambda^T A + \dot{\lambda}^T) \delta x dt - \lambda^T(t_f) \delta x(t_f) + \lambda^T(0) \delta x(0) \quad (4.6)$$

Uslov za minimum kriterijumske funkcije jeste da njena prva varijacija bude jednaka nuli. Zadnji član u relaciji (4.6) je jednak nuli zato što se ne može varirati početni uslov koji je fiksiran. Prema tome, izjednačavanjem (4.6) sa nulom dobijaju se sledeći uslovi optimizacije:

$$u^T R + \lambda^T B = 0 \quad (4.7)$$

$$\dot{\lambda} = -Qx - A\lambda^T \quad (4.8)$$

$$\lambda^T(t_f) = 0 \quad (4.9)$$

Postavlja se pitanje izbora $\lambda(t)$ koje se često naziva koštanje sistema. Pošto je sistem linearan, može se pokušati sa:

$$\lambda(t) = P(t)x(t) \quad (4.10)$$

gdje je P pozitivno definitna i simetrična matrica. Uvrštavanjem (4.10) u (4.8) dobija se diferencijalna matricna Rikatijska jednačina:

$$\dot{P} + PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (4.11)$$

Kada $t_f \rightarrow \infty$ onda $\dot{P}(t)$ teži nuli, tj. $P(t)$ konvergira ka ustaljenom stanju, pa samim tim postaje konstantna matrica. U tom slučaju relacija (4.11) glasi:

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (4.12)$$

i predstavlja algebarsku matricnu Rikatijsku jednačinu. Potrebno je proračunati Rikatijsku matricu P kako bi se na osnovu (4.7) i (4.10) dobila relacija za upravljanje:

$$u = -R^{-1}B^T P x = -Kx \quad (4.13)$$

što predstavlja povratnu spregu po stanjima. Pošto je matrica $K = -R^{-1}B^T P$ konstantna, povratna sprega je statična, tj. dimenzija sistema u zatvorenoj petlji je ista kao kod originalnog sistema. Dinamika sistema u zatvorenoj petlji je:

$$\dot{x} = Ax + Bu = Ax - BKx = (A - BK)x \quad (4.14)$$

Rikatijsku matricu P je teško analitički proračunati na osnovu (4.12), ali se zato numeričkim metodama koji su softverski implementirani lako proračunava.

4.2. Osobine LQR upravljanja

LQR upravljanje ima sledeće karakteristike:

1. Stabilnost: Dokazuje se upotrebom Ljapunov-ove funkcije $V(x)$. Ova funkcija mora biti skalarna, kontinualna i pozitivno definitna. Ukoliko se pronade ova funkcija i pokaže da ona uvijek opada, onda je sistem asimptotski stabilan. Ako se izabere da je $V(x) = x^T P x$, potrebno je pokazati da je $\dot{V}(x) < 0$, tj. diferenciranjem $V(x)$ dobija se:

$$\begin{aligned}\dot{V} &= \dot{X}^T P X + X^T P \dot{X} = [(A - BK)X]^T P X + X^T P [(A - BK)X] = \\ &= X^T [(A - BK)^T P + P(A - BK)]X = \\ &= X^T [(A - BR^{-1}B^T P)^T P + P(A - BR^{-1}B^T P)]X = \\ &= X^T [(PA + A^T P - PBR^{-1}B^T P + Q) - Q - PBR^{-1}B^T P]X = \\ &= -X^T [Q + PBR^{-1}B^T P]X\end{aligned}\tag{4.15}$$

Pošto su matrice Q, R i P pozitivno definitne, onda je i izraz $Q + PBR^{-1}B^T P$ takođe pozitivno definitna matrica, čime je dokazano da je $\dot{V}(x) < 0$, a samim tim i da je sistem u zatvorenoj petlji asimptotski stabilan.

2. Robusnost: Pretek pojačanja je $k_p = \infty$, a pretek faze je $\gamma \geq 60^\circ$, što inače predstavlja vrijednosti pogodne za praktični sistem upravljanja [26].
3. Ponašanje sistema u zatvorenoj petlji: U slučaju da je $R \gg Q$, u indeksu performanse je naglasak na energiji upravljačkog signala, tj. vrši se minimizacija uložene energije, dok u slučaju $Q \gg R$ sistem može imati veoma brz odziv i time značajno, nekada i kritično, povećati trenutne vrijednosti upravljačkog ulaza.

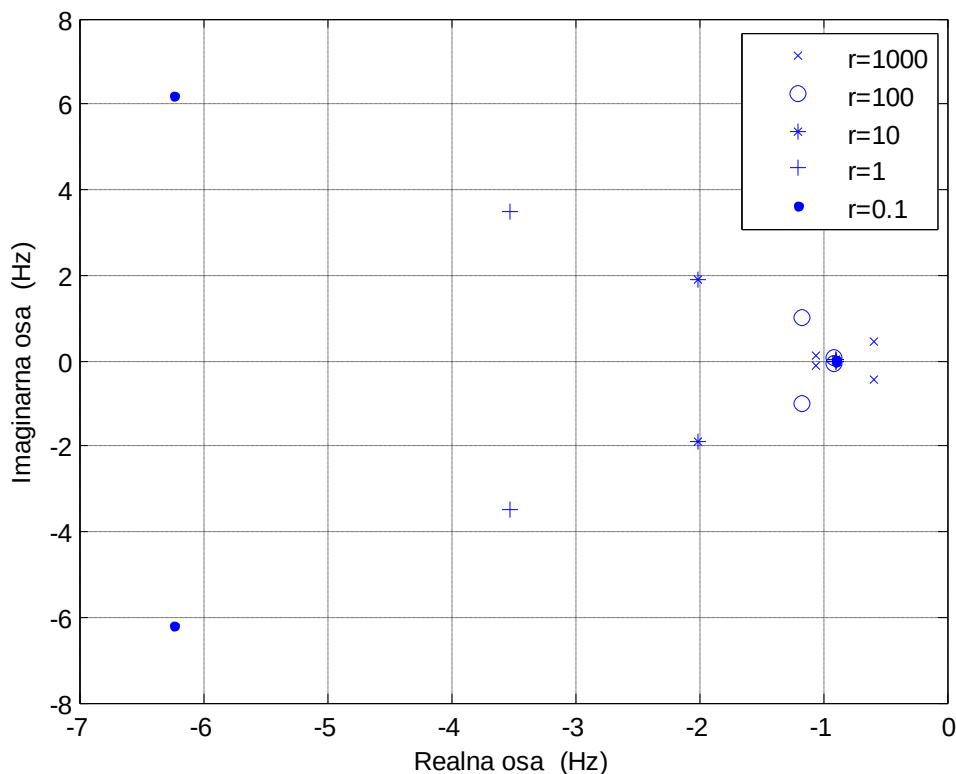
4.3. Projektovanje LQR za Furuta klatno

Za potrebe eksperimentu projektovanja linearnog kvadratnog regulatora za Furuta klatno minimizovaće se kriterijumska funkcija:

$$J = \int (x^T Q x + u^2 r) dt\tag{4.16}$$

gdje je $x = (\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2)^T$ i $u = T$, a potrebno je odrediti težinske koeficijente Q i r , da bi se izvršila optimizacija datog problema. Izabraće se $Q = C^T C$ gdje je $C = [1 \ 0 \ 0 \ 0]^T$, kako bi prvi integralni član iz (4.16) dobio oblik θ_1^2 i na taj način se minimizovala greška pozicije diska u odnosu na referentni položaj.

Sa ovako izabranom matricom Q , potrebno je izabrati skalar r . Treba naglasiti da nije bitna veličina ovih težinskih koeficijenata, već kako se odnose. Prema tome, pošto je Q fiksirano, izvršiće se variranje koeficijenta r u opsegu $\{1000,100,10,1,0.1\}$. Izabraće se ona vrijednost za koju je propusni opseg sistema u zatvorenoj petlji najvširi, ali tako da učestanost polova bude manja ili jednaka od 3 Hz [16]. Ova vrijednost predstavlja praktični limit za ovakav sistem, dok širi propusni opseg može prouzrokovati šum (zbog efekta numeričke diferencijacije diskretnih signala enkodera) ili nestabilnost zbog faznog kašnjenja signala. Ovo je zadovoljeno za slučaj $r = 10$, što se može uočiti na slici 4.1.



Slika 4.1 LQR regulator – polovi sistema za različite vrijednosti r

Upravljački zakon za sistem u zatvorenoj petlji ima oblik $u = Kx$, odakle se dobija vektor $K = [-0.316 \quad -0.139 \quad -1.211 \quad -0.211]$ za prethodno izabrano Q i r .

5. Flatness upravljanje Furuta klatnom

5.1. Flat izlaz Furuta klatna

Principe i tehnike za projektovanje flatness upravljanja iz poglavlja 2. moguće je primjeniti na Furuta klatno. Iako se ovakav tip upravljanja može primjeniti za veliku klasu nelinearnih sistema, u ovom radu će se zbog jednostavnosti koristiti linearizovani matematički model Furuta klatna. To se prije svega odnosi na jednostavni oblik flat izlaza kao što će biti pokazano u nastavku. Prema tome, posmatra se prošireni (sa elektronskim i aktuatorskim komponentama) dinamički model opisan sa relacijama:

$$\frac{\bar{J}_1}{k_{1e}k_s}\ddot{\theta}_{1e} + \frac{mR_h l_{cg}}{k_{2e}k_s}\ddot{\theta}_{2e} + \frac{c_1}{k_{1e}k_s}\dot{\theta}_{1e} = u \quad (5.1)$$

$$\frac{mR_h l_{cg}}{k_{1e}}\ddot{\theta}_{1e} + \frac{\bar{J}_z}{k_{2e}}\ddot{\theta}_{2e} - \frac{mgl_{cg}}{k_{2e}}\theta_{2e} = 0 \quad (5.2)$$

Radi jednostavnosti relacije (5.1) i (5.2) je moguće zapisati:

$$a\ddot{\theta}_{1e} + b\ddot{\theta}_{2e} + c'_1\dot{\theta}_{1e} = u \quad (5.3)$$

$$c\ddot{\theta}_{1e} + d\ddot{\theta}_{2e} - f\theta_{2e} = 0 \quad (5.4)$$

gdje su:

$$a = \frac{\bar{J}_1}{k_{1e}k_s} \quad (5.5)$$

$$b = \frac{mR_h l_{cg}}{k_{2e}k_s} \quad (5.6)$$

$$c'_1 = \frac{c_1}{k_{1e}k_s} \quad (5.7)$$

$$c = \frac{mR_h l_{cg}}{k_{1e}} \quad (5.8)$$

$$d = \frac{\bar{J}_z}{k_{2e}} \quad (5.9)$$

$$f = \frac{mgl_{cg}}{k_{2e}} \quad (5.10)$$

Model Furuta klatna sadrži dvije nezavisne jednačine (5.3) i (5.4). Takođe, sadrži tri varijable θ_{1e}, θ_{2e} i u . Razlika broja varijabli i nezavisnih jednačina je jedan, što znači da ovakav sistem sadrži jedan flat izlaz.

Pri pronalaženju flat izlaza, u početnom koraku treba eliminisati onu jednačinu gdje se određene varijable pojavljuju samo jednom. Dakle, možemo ukloniti (5.3) zbog upravljačke varijable u , koja je prisutna samo u toj relaciji. Prema tome, flat izlaz treba tražiti u relaciji (5.4). Na prvi pogled, za flat izlaz bi logičan izbor bila varijabla θ_{1e} ili θ_{2e} . Ako se izabere θ_{1e} za flat izlaz, relacija (5.4) poprima oblik:

$$c\ddot{y} + d\ddot{\theta}_{2e} - f\theta_{2e} = 0 \quad (5.11)$$

Posmatrajući (5.11) ne bi bilo moguće izvršiti parametrizaciju $\theta_{2e} = f(y, \dot{y}, \dots)$, a da se pri tome ne izvrši integracija što nije dozvoljeno pri izboru flat izlaza. Za slučaj θ_{2e} kao flat izlaza relacija (5.4) se zapisuje:

$$c\ddot{\theta}_{1e} + d\ddot{y} - fy = 0 \quad (5.12)$$

i slijedi isti zaključak kao i u prethodnom slučaju.

Kako ni θ_{1e} ni θ_{2e} ne mogu biti izabrani za flat izlaz, može se doći do pogrešnog zaključka da ovakav sistem nije flat. U ovom slučaju flat izlaz nije tako očigledan. Za Furuta klatno, flat izlaz je linearna kombinacija θ_{1e} i θ_{2e} :

$$y = k_1\theta_{1e} + k_2\theta_{2e} \quad (5.13)$$

pri čemu je potrebno pronaći konstante k_1 i k_2 . Ako se (5.13) dva puta diferencira:

$$\ddot{y} = k_1\ddot{\theta}_{1e} + k_2\ddot{\theta}_{2e} \quad (5.14)$$

i uporedi sa (5.4) koja se može preformulisati:

$$\theta_{2e} = \frac{c}{f}\ddot{\theta}_{1e} + \frac{d}{f}\ddot{\theta}_{2e} \quad (5.15)$$

dobijaju se konstante $k_1 = c/f$ i $k_2 = d/f$ tako da flat izlaz glasi:

$$y = \frac{c}{f}\theta_{1e} + \frac{d}{f}\theta_{2e} \quad (5.16)$$

Relacija za θ_{2e} je:

$$\theta_{2e} = \ddot{y} \quad (5.17)$$

dok se uvrštavajući (5.17) u (5.13) dobija relacija za θ_{1e} :

$$\theta_{1e} = \frac{1}{k_1} y - \frac{k_2}{k_1} \ddot{y} = \frac{f}{c} y - \frac{d}{c} \ddot{y} \quad (5.18)$$

Na kraju, uvrštavajući (5.17) i (5.18) u (5.3) dobija se relacija za u :

$$u = \frac{c'_1 f}{c} \dot{y} + \frac{af}{c} \ddot{y} - \frac{c'_1 d}{c} y^{(3)} + (b - \frac{ad}{c}) y^{(4)} \quad (5.19)$$

Dakle, na osnovu relacija (5.17)-(5.19), sve varijable sistema $(\theta_{1e}, \theta_{2e}, u)$ su izražene preko flat izlaza y kao i njegovih izvoda što dovodi do jasnog zaključka da je sistem (5.3)-(5.4) flat sa (5.16) kao flat izlazom.

5.2. Brunovsky stanja Furuta klatna

Model Furuta klatna kao flat sistema moguće je predstaviti u kanonskom obliku modela stanja. Pošto ovakav sistem sadrži jedan flat izlaz, on predstavlja stanje sistema, kao i konačan broj njegovih izvoda. Preciznije, Brunovsky stanje sistema je $x = (y, \dot{y}, \ddot{y}, y^{(3)})$. Relacija (5.19) pokazuje zavisnost:

$$y^{(4)} = \Phi(y, \dot{y}, \ddot{y}, y^{(3)}, u) \quad (5.20)$$

i kada se uporedi sa uslovom (2.13) jasno je da ne $y^{(4)}$ ne može da predstavlja stanje sistema, tako da je broj komponenti stanja četiri. Sa ovakvim stanjem i relacijom (5.19) sistem u ovom obliku se može zapsati:

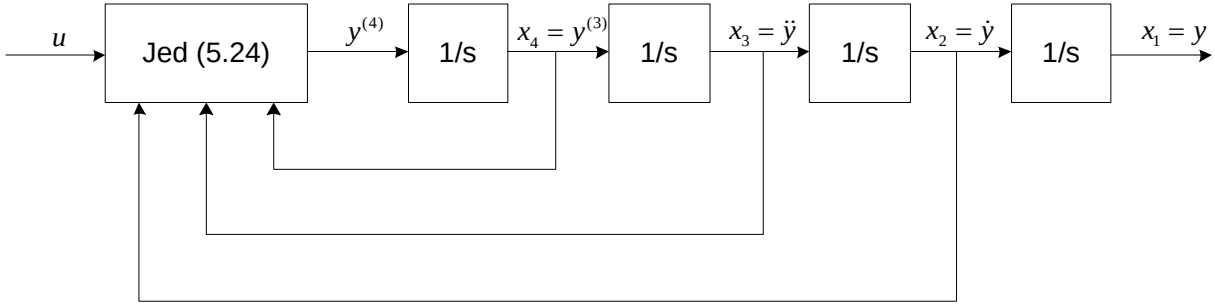
$$\dot{x}_1 = x_2 \quad (5.21)$$

$$\dot{x}_2 = x_3 \quad (5.22)$$

$$\dot{x}_3 = x_4 \quad (5.23)$$

$$\dot{x}_4 = \frac{b}{cb - ad} [-c'_1 f x_2 - a f x_3 + c'_1 d x_4 + cu] \quad (5.24)$$

što se može predstaviti i šematski (slika 5.1).



Slika 5.1 Kanonski oblik blok dijagrama Furuta klatna

5.3. Feed-forward upravljanje

Izborom pogodne trajektorije za flat izlaz i inverzijom modela dobija se upravljački ulaz koji treba da djeluje na sistem. To je u suštini relacija (5.19). Trajektorija za flat izlaz će se izabrati u skladu sa “rest to rest” režimom rada [27]. To znači da će se izvršiti pomjeranje diska za određeni ugao, ali tako da ne postoje oscilacije diska i klatna u početnom i krajnjem trenutku. Ovaj uslov se odražava na njihove ugaone brzine i ubrzanja, koji moraju biti jednaki nuli u tim trenucima. Pod pretpostavkom da će se pomjeranje diska izvršiti za ugao θ_{1ef} i trajati određeno vrijeme t_f , ovi početni, kao i krajnji uslovi se mogu zapisati:

$$\theta_{1e}(0) = 0 \quad \theta_{1e}(t_f) = \theta_{1ef} \quad (5.25)$$

$$\dot{\theta}_{1e}(0) = 0 \quad \dot{\theta}_{1e}(t_f) = 0 \quad (5.26)$$

$$\ddot{\theta}_{1e}(0) = 0 \quad \ddot{\theta}_{1e}(t_f) = 0 \quad (5.27)$$

$$\theta_{2e}(0) = 0 \quad \theta_{2e}(t_f) = 0 \quad (5.28)$$

$$\ddot{\theta}_{2e}(0) = 0 \quad \ddot{\theta}_{2e}(t_f) = 0 \quad (5.29)$$

Na osnovu relacije (5.19) se može zaključiti da trajektorija flat izlaza mora biti najmanje četiri puta diferencijabilna. Takođe, poštujući principe projektovanja iz poglavlja 2, tj. relaciju (2.36), referentna trajektorija dobija oblik polinoma devetog stepena:

$$y_r(t) = \sum_{i=0}^9 c_i t^i \quad (5.30)$$

Nepoznati koeficijenti $c_i, i = 0, \dots, 9$ se dobijaju na osnovu pet početnih i pet krajnjih uslova, tj. $y_r(0), \dot{y}_r(0), \ddot{y}_r(0), y_r^{(3)}(0), y_r^{(4)}(0)$ i $y_r(t_f), \dot{y}_r(t_f), \ddot{y}_r(t_f), y_r^{(3)}(t_f), y_r^{(4)}(t_f)$. Kako ove vrijednosti nijesu poznate potrebno ih je indirektno dobiti preko uslova (5.25)-(5.29) na sledeći način:

$$y_r(0) = \frac{c}{f} \theta_{1e}(0) + \frac{d}{f} \theta_{2e}(0) = 0 \quad (5.31)$$

$$\dot{y}_r(0) = \frac{c}{f} \dot{\theta}_{1e}(0) + \frac{d}{f} \dot{\theta}_{2e}(0) = 0 \quad (5.32)$$

$$\ddot{y}_r(0) = \ddot{\theta}_{2e}(0) = 0 \quad (5.33)$$

$$y_r^{(3)}(0) = \dot{\theta}_{2e}(0) = 0 \quad (5.34)$$

$$y_r^{(4)}(0) = \ddot{\theta}_{2e}(0) = 0 \quad (5.35)$$

$$y_r(t_f) = \frac{c}{f} \theta_{1e}(t_f) + \frac{d}{f} \theta_{2e}(t_f) = \frac{c}{f} \theta_{1ef} \quad (5.36)$$

$$\dot{y}_r(t_f) = \frac{c}{f} \dot{\theta}_{1e}(t_f) + \frac{d}{f} \dot{\theta}_{2e}(t_f) = 0 \quad (5.37)$$

$$\ddot{y}_r(t_f) = \ddot{\theta}_{2e}(t_f) = 0 \quad (5.38)$$

$$y_r^{(3)}(t_f) = \dot{\theta}_{2e}(t_f) = 0 \quad (5.39)$$

$$y_r^{(4)}(t_f) = \ddot{\theta}_{2e}(t_f) = 0 \quad (5.40)$$

Na osnovu uopštene relacije (2.41), referentna trajektorija flat izlaza u ovom slučaju je:

$$y_r(t) = \frac{c}{f} \theta_{1ef} \frac{t^5}{t_f^5} (a_0 + a_1 \frac{t}{t_f} + a_2 \frac{t^2}{t_f^2} + a_1 \frac{t^3}{t_f^3} + a_4 \frac{t^4}{t_f^4}) \quad (5.41)$$

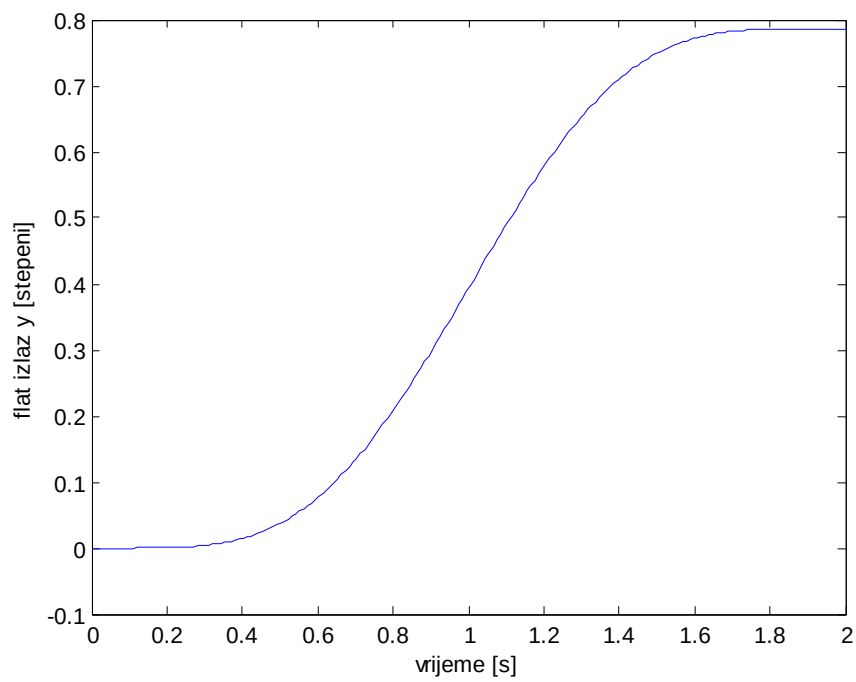
pa se primjenom (2.42) koeficijenti a_0, a_1, a_2, a_3, a_4 dobijaju rješavanjem sledećeg linearnog sistema jednačina:

$$\begin{aligned} 1 &= a_0 + a_1 + a_2 + a_3 + a_4 \\ 0 &= 5a_0 + 6a_1 + 7a_2 + 8a_3 + 9a_4 \\ 0 &= 20a_0 + 30a_1 + 42a_2 + 56a_3 + 72a_4 \\ 0 &= 60a_0 + 120a_1 + 210a_2 + 336a_3 + 504a_4 \\ 0 &= 120a_0 + 360a_1 + 840a_2 + 1680a_3 + 3024a_4 \end{aligned} \quad (5.42)$$

odakle su koeficijenti $a_0 = 126, a_1 = -420, a_2 = 540, a_3 = -315, a_4 = 70$. Konačno, $y_r(t)$ je:

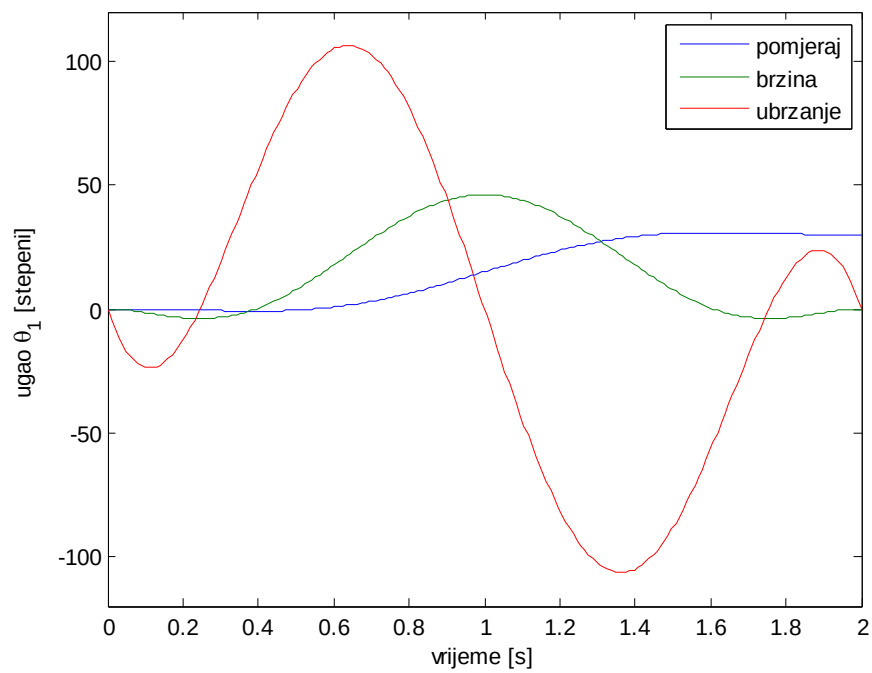
$$y_r(t) = \frac{c}{f} \theta_{1ef} \frac{t^5}{t_f^5} (126 - 420 \frac{t}{t_f} + 540 \frac{t^2}{t_f^2} - 315 \frac{t^3}{t_f^3} + 70 \frac{t^4}{t_f^4}) \quad (5.43)$$

Uvrštavanjem (5.43) u (5.17)-(5.19) dobijaju se referentne trajektorije svih varijabli. Referentna trajektorija flat izlaza opisana relacijom (5.43) za pomjeraj diska od 30 stepeni u trajanju od 2 sekunde prikazana je na slici 5.2.



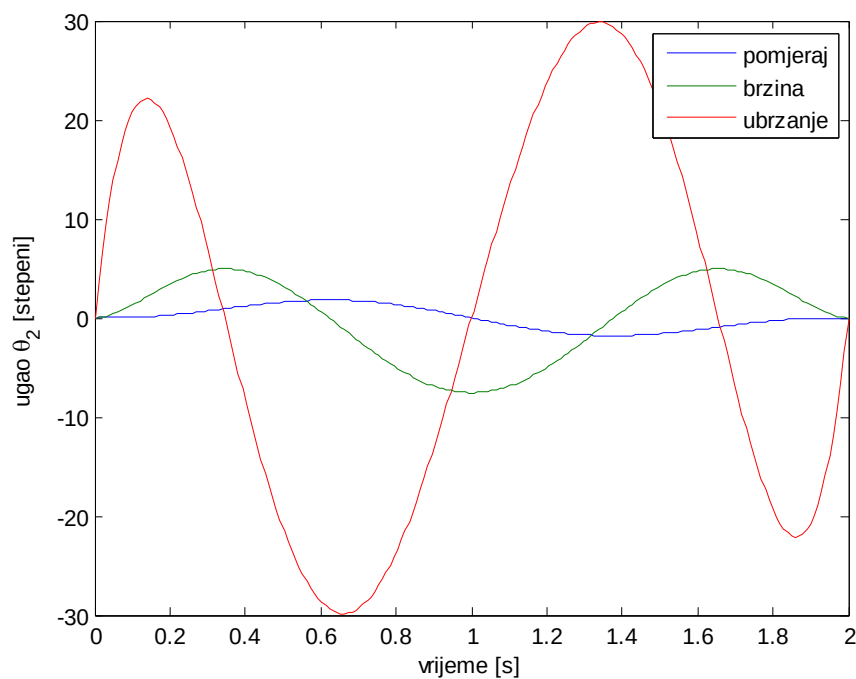
Slika 5.2 Referentna trajektorija flat izlaza

Na slici 5.3 prikazane su referentne trajektorije ugaonog pomjeraja, brzine i ubrzanja diska.



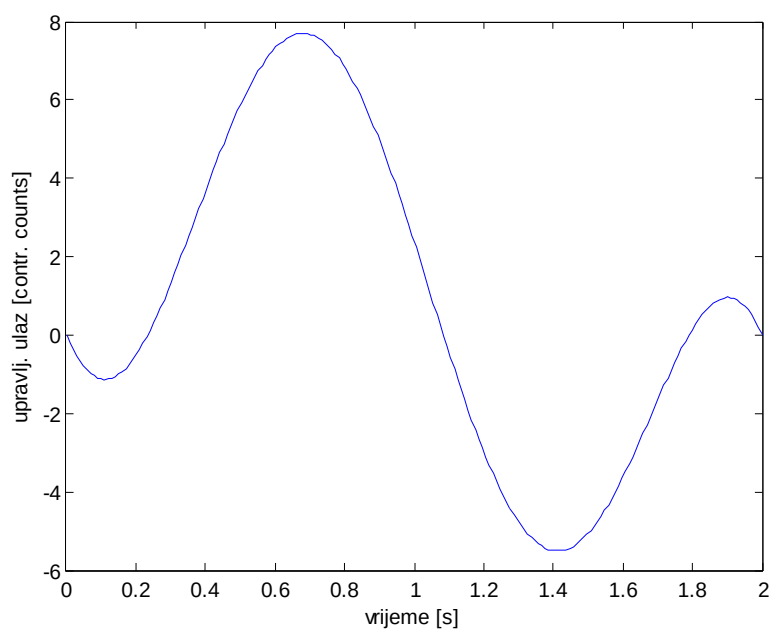
Slika 5.3 Referentna trajektorija ugaonog pomjeraja, brzine i ubrzanja diska

Na slici 5.4 prikazane su referentne trajektorije ugaonog pomjeraja, brzine i ubrzanja klatna.



Slika 5.4 Referentna trajektorija ugaonog pomjeraja, brzine i ubrzanja klatna

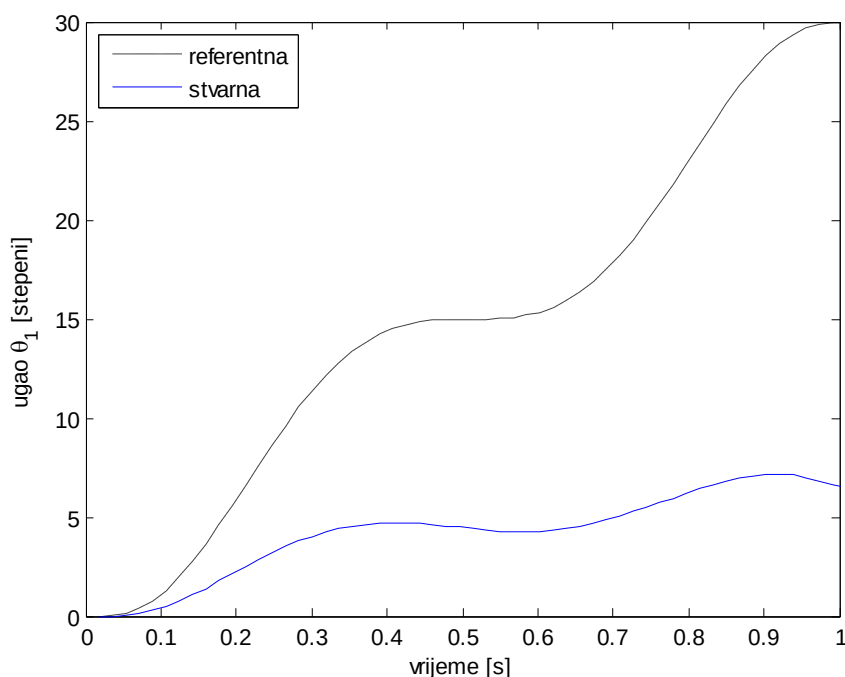
Na slici 5.5 prikazana je referentna trajektorija upravljačkog ulaza.



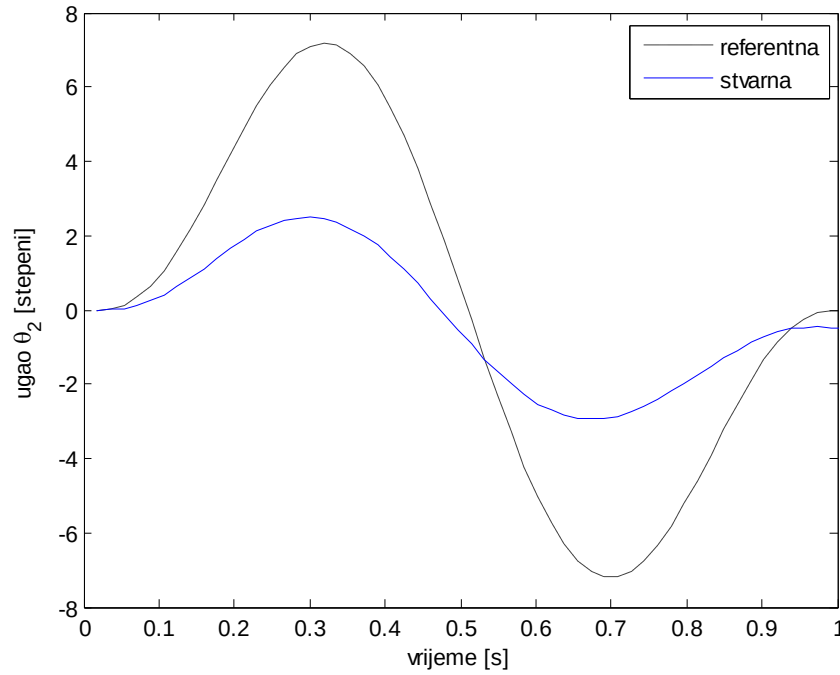
Slika 5.5 Referentna trajektorija upravljačkog ulaza

Sa slika 5.3 i 5.4 se može primjetiti da su u početnom ($t=0$) i krajnjem ($t=2$) trenutku brzine i ubrzanja jednake nuli, čime će sistem na zadovoljavajući način preći iz jednog u drugi stacionarni režim.

Primjenom upravljačkog ulaza kao na slici 5.5, sve varijable sistema bi trebalo da prate svoje referentne trajektorije. Na ovaj način vrši se tzv. feed-forward upravljanje i u suštini predstavlja upravljanje u otvorenoj petlji sa isplaniranim kretanjem. Međutim, ovakav tip upravljanja je pogodan samo u idealnom slučaju, kada se matematički model poklapa sa realnim sistemom i kada na sistem ne djeluju smetnje. Međutim, u stvarnosti to nikad nije slučaj, jer na sistem uvijek djeluju razne spoljašnje smetnje, dok je kod senzora skoro uvijek prisutan mjerni šum. Takođe, prilikom opisivanja modela sistema određena dinamika može biti zanemarena. To se radi zbog jednostavnosti modela ili slabog uticaja te dinamike, npr. za disk je modelovano viskozno trenje, ali ne i Kolumbovo (statičko) trenje. Isto tako zanemarena je dinamika senzora i motora. Uzrok lošeg modela mogu biti i varijacije parametara sistema. U većini slučajeva, nemodelovana dinamika je nepoznata i nije je moguće uključiti u model. Za Furuta klatno primjenjen je linearizovani model, što ponovo može dovesti do pitanja opravdanosti ovakvog modela. Na slikama 5.6 i 5.7 može se vidjeti primjer feed-forward upravljanja klatnom za neinvertovani slučaj. Jasno se vidi da se kretanje diska i klatna ne poklapa sa svojim trajektorijama, što nas može navesti na zaključak da model Furuta klatna nije dovoljno precizno opisan.



Slika 5.6 Feedforward upravljanje - ugaoni pomjeraj diska



Slka 5.7 Feedforward upravljanje - ugaoni pomjeraj klatna

Ipak, projektovanjem povratne sprege, u praksi se ovakvi nedostaci mogu prevazići i učiniti sistem vrlo robusnim na interne i eksterne poremećaje. To će biti tema sledećeg poglavlja.

5.4. Projektovanje povratne sprege

U poglavlju 2. opisana je procedura projektovanja povratne sprege za flat sisteme. Pri tome je uvedena pomoćna varijabla, tzv. novi ulaz v :

$$v_i = y_i^{(\lambda_i)}, \quad i = 1, \dots, m \quad (5.44)$$

tako da se dobija sistem koji se sastoji od jednog ili više lanaca (kaskadnih veza) integratora. Na ovaj način je izvršena egzaktna linearizacija sistema. Slično kao u poglavlju 2. stanje sistema se može zapisati preko flat izlaza:

$$x = (y_1, \dot{y}_1, \dots, y_1^{(k_1-1)}, y_2, \dots, y_m^{(k_m-1)}) \quad (5.45)$$

Ako se izabere:

$$\lambda_i = k_i, \quad i = 1, \dots, m \quad (5.46)$$

dobija se sistem u povratnoj sprezi za koji se lako vrši stabilizacija dinamike greške na osnovu relacije (2.54) i pogodnim izborom koeficijenata.

Potrebno je primjeniti ovu proceduru na Furuta klatno. Stanje klatna je:

$$x = (y, \dot{y}, \ddot{y}, y^{(3)}) \quad (5.47)$$

pa na osnovu (5.46) slijedi da je $\lambda = 4$, što znači da se novi ulaz zapisuje:

$$v = y^{(4)} \quad (5.48)$$

Dinamika greške praćenja $e = y - y_r$ je četvrtog reda:

$$e^{(4)} + k_3 e^{(3)} + k_2 \ddot{e} + k_1 \dot{e} + k_0 e = 0 \quad (5.49)$$

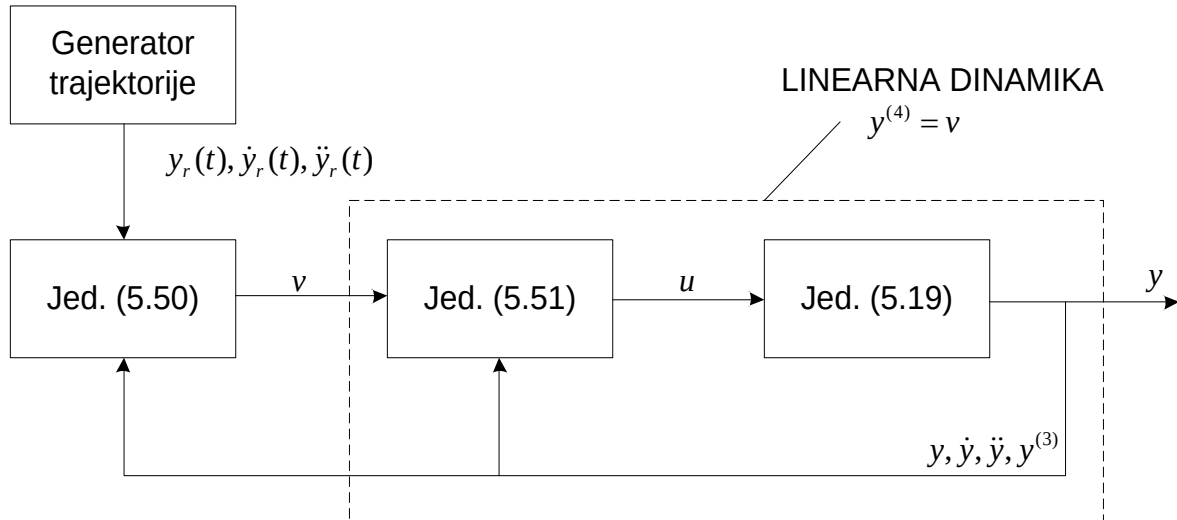
što se može preformulisati:

$$v = y_r^{(4)} - k_3 (y^{(3)} - y_r^{(3)}) - k_2 (\ddot{y} - \ddot{y}_r) - k_1 (\dot{y} - \dot{y}_r) - k_0 (y - y_r) \quad (5.50)$$

dok relacija za upravljački ulaz dobija oblik:

$$u = \frac{c'_1 f}{c} \dot{y} + \frac{af}{c} \ddot{y} - \frac{c'_1 d}{c} y^{(3)} + (b - \frac{ad}{c})v \quad (5.51)$$

Upravljačka struktura sistema u povratnoj sprezi je prikazana na slici 5.8.



Slika 5.8 Furuta klatno sa projektovanom povratnom spregom

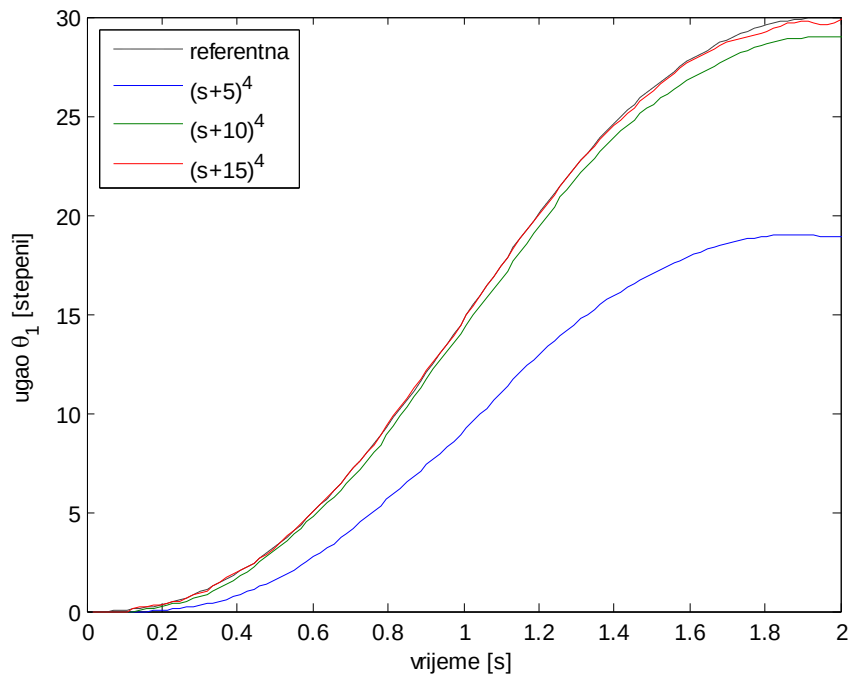
Sa slike 5.8 se uočava da je potrebno mjeriti stanje $x = (y, \dot{y}, \ddot{y}, y^{(3)})$. To podrazumjeva mjerenja brzina $\dot{\theta}_1$ i $\dot{\theta}_2$. Optički inkrementalni enkodери obezbjeđuju samo mjerenja ugla θ_1 i θ_2 .

Jedan od načina da se dobije informacija o brzinama, bez prisustva senzora za brzinu, jeste da se projektuje observer koji bi vršio njihovu estimaciju. U ovom radu će se koristiti jednostavnija ali manje pouzdana metoda numeričkog diferenciranja:

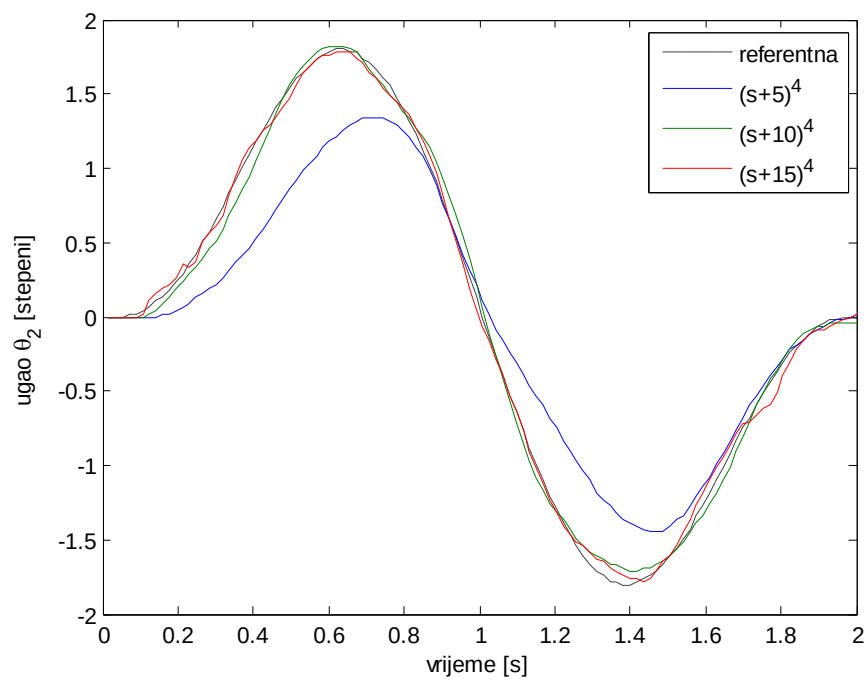
$$\omega_i = \frac{\theta_i - \theta_{i-1}}{T_s} \quad (5.52)$$

Ovakva metoda može da uzrokuje nepoželjni mjerni šum, o čemu je bilo riječi u poglavlju 3. Ipak, pošto je vrijeme odabiranja $T_s = 0.00884$ veoma malo, vrijednosti proračunate brzine u određenom trenutku imaće zadovoljavajuću tačnost.

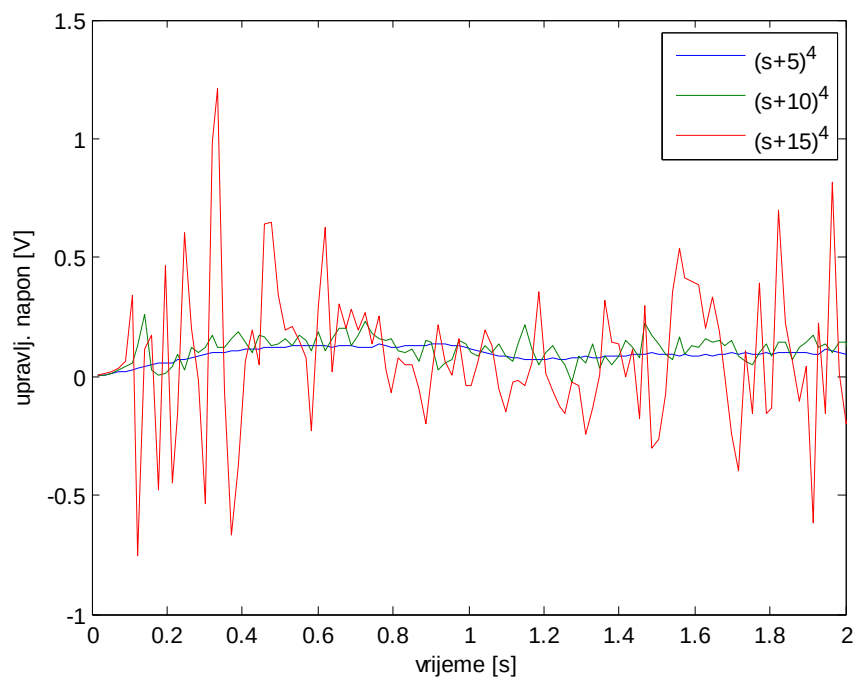
Koeficijenti iz relacije (5.49) koji opisuju dinamiku greške su određeni eksperimentalno, ponovo na primjeru neinvertovanog klatna. Testirana su tri slučaja za četiri pola greške u -5, -10 i -15, na osnovu kojih su zatim analizirane performanse sistema. Na slikama 5.9 i 5.10 su prikazani odzivi uglova θ_1 i θ_2 za sva tri slučaja i poređeni su sa svojim referentnim trajektorijama. Slično, na slici 5.11 prikazane su vrijednosti upravljačkog ulaza.



Slika 5.9 Ugaoni pomjeraj diska za različite dinamike greške



Slika 5.10 Ugaoni pomjeraj klatna za različite dinamike greške



Slika 5.11 Upravljački napon za različite dinamike greške

Sa slika 5.9 i 5.10 se može vidjeti da je najbolje praćenje trajektorije postignuto za polove greške u -15. Postavljanjem polova što dalje ulijevo od imaginarne ose u kompleksnoj ravni postiže se brži odziv sistema ali to utiče na veća upravljačka naprezanja, što se može vidjeti na slici 5.11. To se odražava na ulazni napon DC motora koji pogoni disk, koji može ući u zasićenje i prouzrokovati nestabilni rad. Iz tog razloga izabrana dinamika greške će biti sa karakterističnim polinomom $(s + 12)^4$ koji kada se proširi dobija oblik:

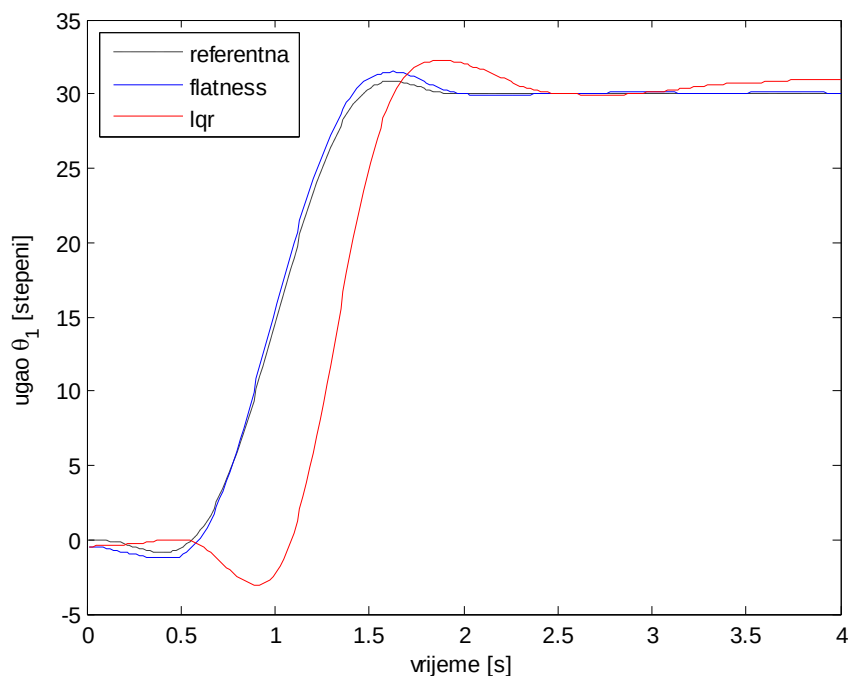
$$s^4 + 48s^3 + 864s^2 + 6912s + 20736 \quad (5.53)$$

odakle se dobijaju koeficijenti $k_3 = 48, k_2 = 864, k_1 = 6912, k_0 = 20736$.

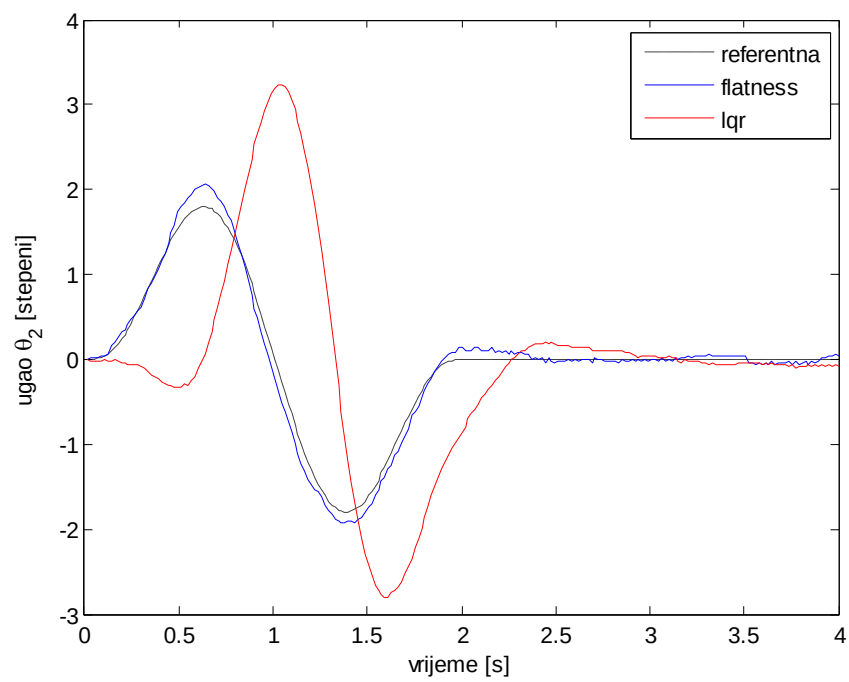
6. Eksperimentalna analiza Flatness i LQR metode

6.1. Praćenje referentne trajektorije

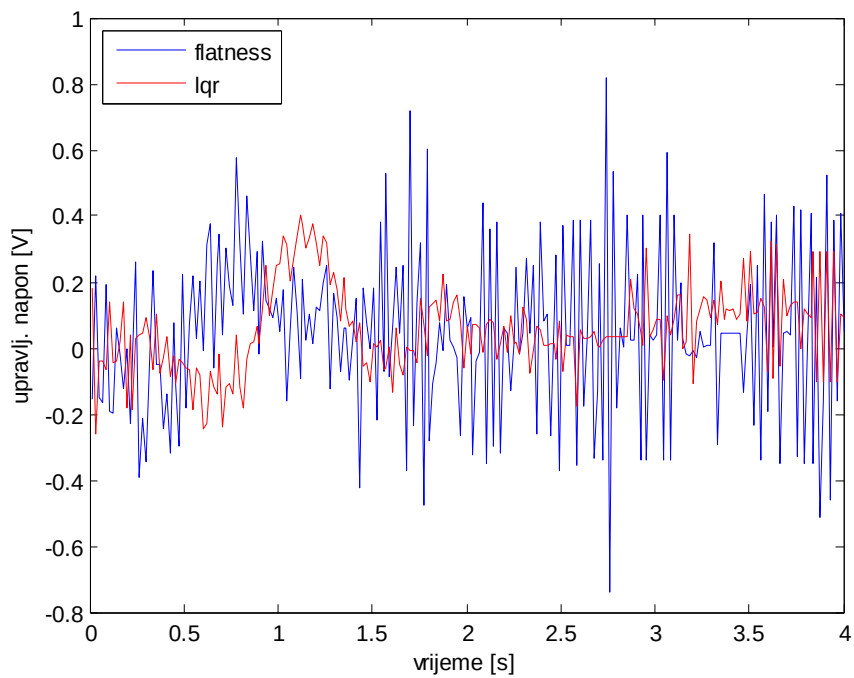
Eksperiment je urađen za pomjeraj diska od 30° za 2 sekunde, tako što prati referentnu trajektoriju $\theta_{1er}(t)$. Nakon toga disk treba da stabilizuje klatno dodatne 2 sekunde. Trajektorija za $\theta_{1er}(t)$ se dobija tako što se u (5.43) uvrsti $\theta_{1ef} = \theta_{1f}(\circ) \cdot \pi / 180 \cdot k_{1e} = 30 \cdot \pi / 180 \cdot 2546$ i $t_f = 2$, čime se dobija trajektorija $y_r(t)$ koja se zatim uvrštava u (5.18). Slično se dobija i referentna trajektorija za $\theta_{2er}(t)$, gdje se $y_r(t)$ uvrštava u (5.17). Na slikama 6.1, 6.2 i 6.3 su prikazani eksperimentalni rezultati za zadati pomjeraj diska. Početni položaj klatna je inverzni položaj, koji se postiže pomoću samo-invertujućeg algoritma kada se disku zadaju naizmjenični step pomjeraji koji zamahuju klatno sve dok se klatno ne uspravi. U trenutku kada klatno postigne taj položaj aktivira se upravljački algoritam za stabilizaciju. Kod Flatness upravljanja disk skoro idealno prati zadatu trajektoriju, dok LQR značajno kasni i odstupa. Takođe, u slučaju LQR, veće su i oscilacije klatna. U slučaju primjene Flatness algoritma varijacije upravljačkog napona izraženije su u odnosu na LQR. Kod LQR algoritma se vrši optimizacija utroška energije, dok je kod Flatness algoritma dominantan zahtjev tačno praćenje trajektorije.



Slika 6.1 Poređenje praćenja referentne trajektorije diska



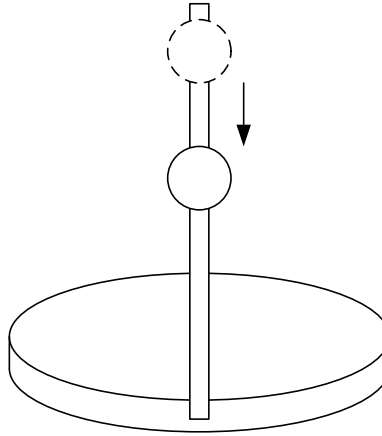
Slika 6.2 Poređenje praćenja referentne trajektorije klatna



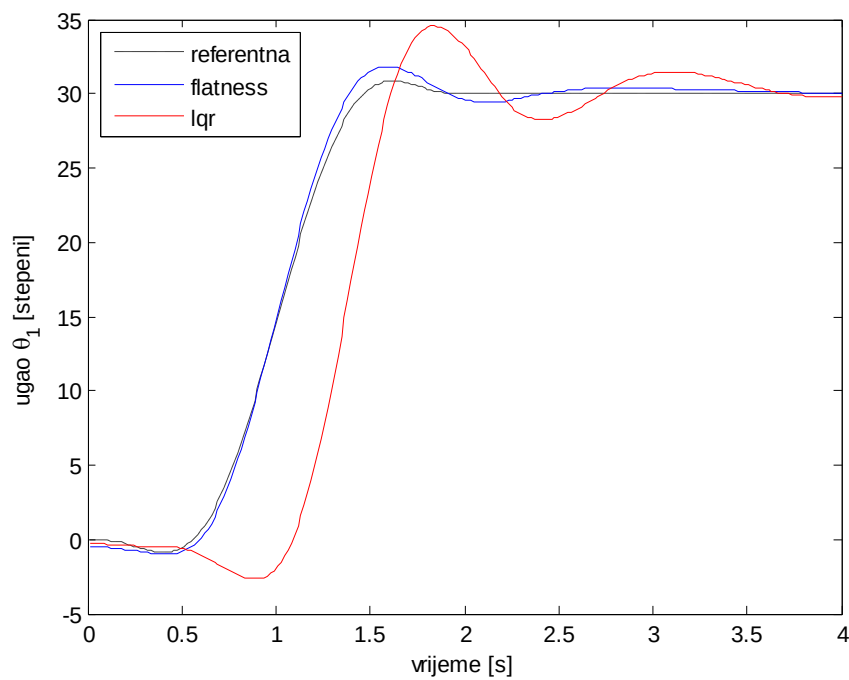
Slika 6.3 Poređenje upravljačkog napona

6.2. Robusnost na promjenu parametara

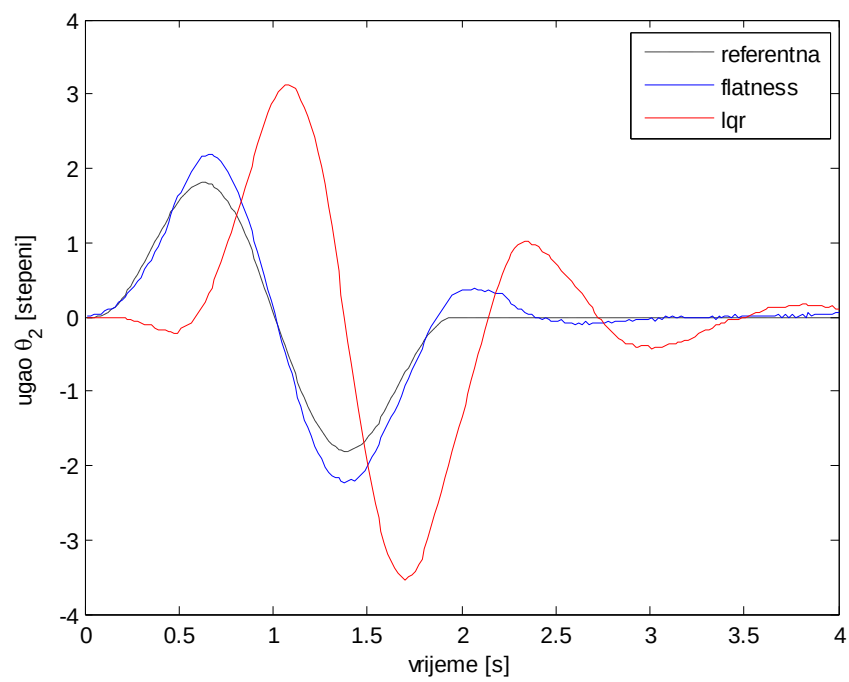
Eksperiment koji je opisan u 6.1 je ponovljen za slučaj kada se teg na disku pomjeri za 10 cm u smjeru prema disku, kao na slici 6.4. Na taj način testira se robusnost sistema u zatvorenoj petlji, tj. stabilnost sistema za slučaj varijacije parametara. Na slikama 6.5, 6.6 i 6.7 se jasno uočava da flatness upravljanje i u ovom slučaju pokazuje bolje performanse u odnosu na LQR, ali sa mnogo većim naponskim oscilacijama.



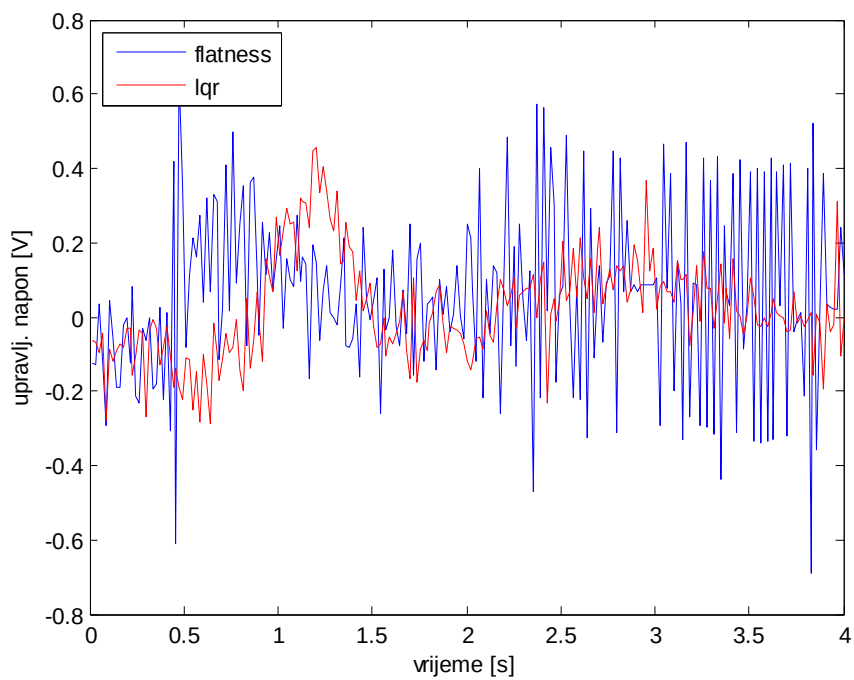
Slika 6.4 Pomjeranje težišta klatna



Slika 6.5 Poređenje prećenja referentne trajektorije diska pri promjeni parametara



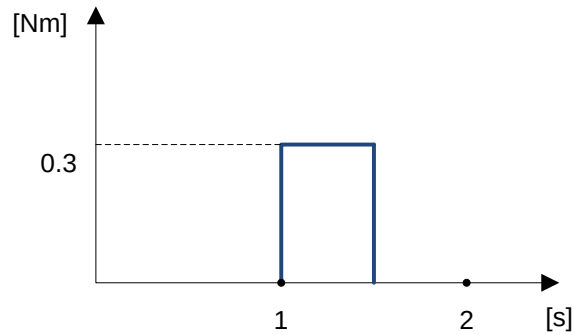
Slika 6.6 Poređenje prećenja referentne trajektorije klatna pri promjeni parametara



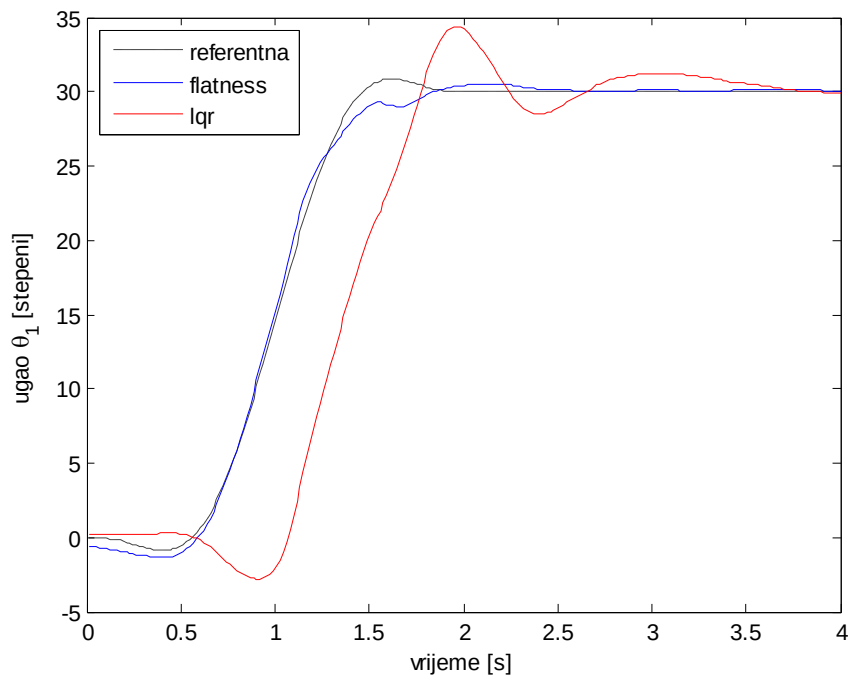
Slika 6.7 Poređenje upravljačkog napona pri promjeni parametara

6.3. Kompenzacija smetnje

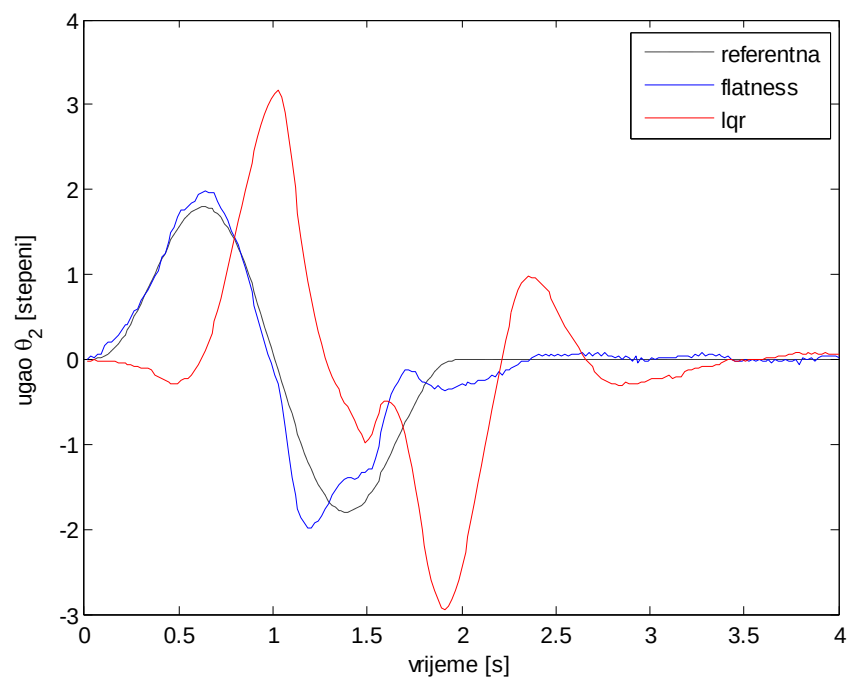
Testiranje sistema u zatvorenoj petlji u slučaju djelovanja spoljašnjeg poremećaja sprovedeno je tako što je u izvršnom kodu na upravljački ulaz dodat pravougaoni impuls u trajanju od 0.5 sekundi (od 1 do 1.5) veličine 0.3 Nm, kao na slici 6.8. Kao i u prethodna dva eksperimenta, i u ovom slučaju flatness mnogo brže i bolje stabilizuje klatno što se vidi na slikama 6.9, 6.10 i 6.11, po cijenu većeg naponskog naprezanja motora a time i veće uložene energije.



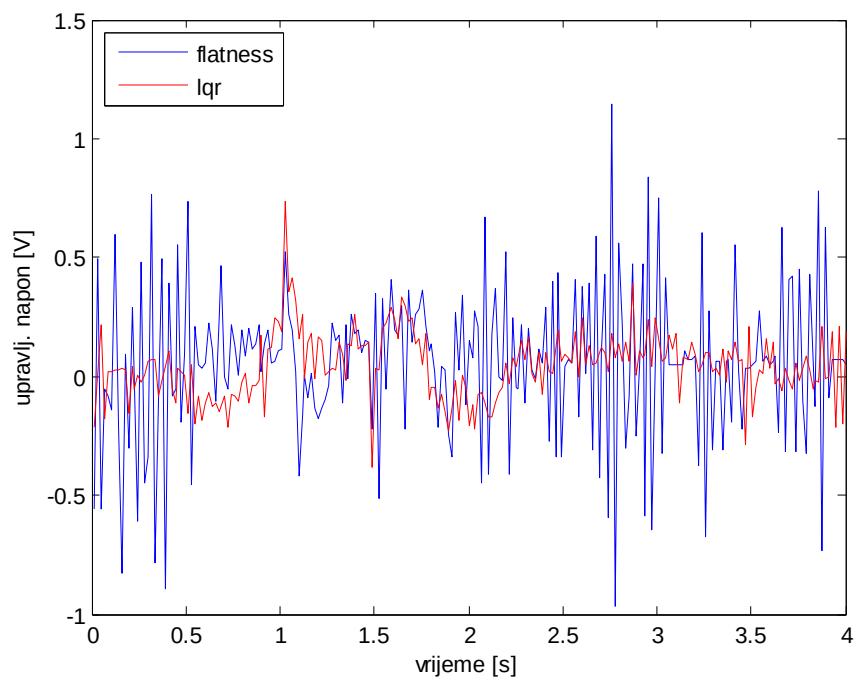
Slika 6.8 Smetnja koja djeluje na upravljačkom ulazu (momentu motora)



Slika 6.9 Poređenje kompenzacije smetnje pri praćenju trajektorije diska



Slika 6.10 Poređenje kompenczacije smetnje pri praćenju trajektorije klatna



Slika 6.11 Poređenje upravljačkog napona pri kompenczaciji smetnje

7. Zaključak

U ovom radu izvršena je uporedna analiza dva algoritma za upravljanje Furuta klatnom. Klatno se nalazi na rotirajućoj platformi i predstavlja nelinearni, mehanički i potpogonjeni sistem, tj. sistem koji ima veći broj stepeni slobode od broja aktuatora. Dinamika klatna je opisana diferencijalnim jednačinama pomoću Euler-Lagrange-ove jednačina. Takođe, zbog jednostavnosti, dinamički model je linearizovan oko ravnotežne tačke. Matematički model je poslužio kao polazište za eksperimentalnu analizu laboratorijskog modela klatna ECP A51. Ovaj model, pored osnovnog mehaničkog sistema, čine razne elektronske i aktuatorске komponente čiji je uticaj pojednostavljen, ali ne i zanemaren. Prema tome, ovakav model predstavlja realni sistem, sa svim hardverskim i softverskim rješenjima koja omogućavaju testiranje i primjenu teorijskih rezultata. Na modelu je sprovedeno flatness i LQR upravljanje. Oba algoritma se zasnivaju na moćnom konceptu povratne sprege, tj. na upravljačkom ulazu koji je u obliku povratne sprege po stanjima sistema.

Flatness upravljanje je jedna od savremenih metoda upravljanja nelinearnim sistemima koji pokazuju posebne karakteristike pa se shodno tome nazivaju flat sistemi. Ova metoda se zasniva na parametrizaciji svih sistemskih varijabli pomoću vektora diferencijalno nezavisnih varijabli kojeg nazivamo flat izlaz. Flat izlaz za slučaj Furuta klatna je u obliku linearne kombinacije ugaonog pomjeraja rotirajućeg diska i klatna. Konvencionalne metode su teško rješavale problem praćenja izabrane trajektorije zbog potrebe rješavanja diferencijalnih jednačina iz početnih uslova kako bi se dobilo krajnje rješenje. Za slučaj flat sistema nije potrebno rješavati diferencijalne jednačine već samo projektovati trajektoriju flat izlaza, a zatim upotrebom povratne sprege definisati dinamiku greške praćenja trajektorije tako da ona konvergira što je brže moguće ka nuli. Dakle, ovakav pristup se može podijeliti u dva koraka. Prvi korak je izbor referentne trajektorije, a drugi izbor dinamike greške praćenja trajektorije flat izlaza i proračun zakona upravljanja. Dinamika greške praćenja za Furuta klatno je određena prema rezultatima koji su dobijeni na osnovu eksperimentalnog testiranja.

LQR algoritam upravlja sistemom tako da se minimizuje zadati kriterijum performanse. Postoji veliki broj naučnih radova koji opisuju primjenu ove metode na razne sisteme. LQR pokazuje značajne nedostatke prilikom praćenja zadate trajektorije. Flatness upravljanje, kao tehnika za upravljanje nelinearnim sistemima, prevazilazi ove nedostatke kako za nelinearne, tako i za linearne kontrolabilne sisteme. Komparativna prednost flatness upravljanja se vidi i u slučaju promjene parametara modela, kako bi se simulirala relativna stabilnost sistema u zatvorenoj petlji, kao i u slučaju kompenzacije eksterne smetnje. Eksperimentalni rezultati potvrđuju navedene karakteristike oba algoritma. Flatness metoda je jednostavna i efikasna. Međutim, to ne bi bila veoma važna metoda da postoji samo nekoliko akademskih primjera flat sistema. Činjenica je da postoji veliki broj tehnoloških upravljačkih sistemima koji su flat sistemi.

Dalji pravac istraživanja odnosiće se na poređenje flatness metode sa drugim sličnim i efektivnim metodama za nelinearne sisteme, kao i primjena metode na razne druge praktične sisteme. Takođe, u radu je primjenjen samo jedan od mogućih načina za projektovanje povratne sprege flat sistema, što ostavlja dodatni prostor i za teorijska proširenja same metode.

Literatura

- [1] F. Rotella, F. Carrillo and M. Ayadi, Digital flatness-based robust controller applied to a thermal process, IEEE international Conference on Control application, pp. 936- 941, Mexico 2001.
- [2] R. Rothfuss, J. Rudolph and M. Zeitz, Flatness based control of chemical reactor model, European Control Conference, pp. 637-642, Rome, September 1995.
- [3] A. Chelouah, E. Delaleau, P. Martin and P. Rouchon Differential flatness and control of induction motors, symposium on Control, Optimization and Supervision, Computational engineering in system applications, IMACS Multiconference, pp. 80-85, Lille, 9-12 July 1996.
- [4] Souilem, Haifa, Hassen Mekki, and Nabil Derbel. "Crane control by flatness." Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on. IEEE, 2012.
- [5] Aguilar-Ibáñez, Carlos, Hebertt Sira-Ramírez, and Miguel S. Suárez-Castañón. "A flatness-based controller for the stabilization of the inverted pendulum." Mathematical Problems in Engineering 2011 .
- [6] Mansour, Ali, and Housseem Jerbi. "An exact differential flatness control for a non minimum phase model of an inverted pendulum." Modelling & Simulation, 2009. AMS'09. Third Asia International Conference on. IEEE, 2009.
- [7] Furuta, K., Yamakita, M. and Kobayashi, S. (1992) Swing-up control of inverted pendulum using pseudo-state feedback, Journal of Systems and Control Engineering, 206(6), 263-269.
- [8] Xu, Y., Iwase, M. and Furuta, K. (2001) Time optimal swing-up control of single pendulum, Journal of Dynamic Systems, Measurement, and Control, 123(3), 518-527.
- [9] Furuta, K., Iwase, M. (2004) Swing-up time analysis of pendulum, Bulletin of the Polish Academy of Sciences: Technical Sciences, 52(3), 153-163.
- [10] Iwase, M., Åström, K. J., Furuta, K. and Åkesson, J. (2006) Analysis of safe manual control by using Furuta pendulum, Proceedings of the IEEE International Conference on Control Applications, 568-572.
- [11] Tedrake, Russ (2008). Underactuated Robotics: Learning, Planning, and Control for Efficient and Agile Machines
- [12] Yan, Qiguo. "Output tracking of underactuated rotary inverted pendulum by nonlinear controller." Decision and Control, 2003. Proceedings. 42nd IEEE Conference on. Vol. 3. IEEE, 2003.

- [13] B. Friedland, Control System Design: An Introduction to State Space Methods. Dover, 2004.
- [14] F. L. Lewis and V. L. Syrmos. Optimal Control. Wiley, second edition, 1995.
- [15] Kwakernaak, Huibert and Sivan, Raphael (1972). Linear Optimal Control Systems. First Edition. Wiley-Interscience.
- [16] Educational Control Products, Manual for A-51 Inverted Pendulum Accessory, 2002.
- [17] M. Fliess, J. Lévine, Ph. Martin and P. Rouchon, On differentially flat nonlinear systems, IFAC-Synposium, NOLCOS'92 pp. 408-412, Bordeaux, 1992.
- [18] M. Fliess, J. Lévine, and P. Rouchon. Generalized statevariable representation for a simplified crane description. Int. J. Control, 58:277–283, 1993.
- [19] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. Int. J. Control, 61:1327–1361, 1995.
- [20] Levine, Jean. Analysis and control of nonlinear systems: A flatness-based approach. Springer, 2009.
- [21] Sira-Ramirez, Hebertt, and Sunil K. Agrawal. Differentially flat systems. Vol. 17. CRC Press, 2004.
- [23] J. Rudolph, Flatness Based Control - Introduction, on October 7-11, 2013.
- [24] Quadrat, A., Robertz, D. Controllability and differential flatness of linear analytic ordinary differential systems, 2004.
- [25] Delaleau E., Rudolph J., Control of flat systems by quasi-static feedback of generalized states, International Journal of Control Volume 71, Issue 5, 1998.
- [26] M. G. Safonov and M. Athans, Gain and phase margins for multiloop LQG regulators, IEEE Trans. Automat. Contr., Apr. 1977.
- [27] De Luca A., Di Giovanni G., Rest-to-rest motion of a two-link robot with a flexible forearm, IEEE/ASME International Conference On Advanced Intelligent Mechatronics Proceedings, 929 - 935 vol.2, 2001

Prilog 1 – Matlab kodovi

Trajektorije.m – iscrtava referentne trajektorije ugaonih pomjeraja, brzina i ubrzanja diska i klatna, kao i upravljačkog napona i flat izlaza

```
function [] = Trajektorije(theta1f,tf)
%Parametri linearizovanog modela
inv=1; %1 za invertovani, -1 za neinvertovani model
k1=0.0265; %k1=J1_+Jy
k2=inv*0.0107; %k2=m*Rh*lcg
k3=0.0131; %k3=Jz_
k4=inv*0.4182; %k4=m*grav*lcg
k5=0.02; %k5-viskozno trenje diska
ksys=0.00675; %hardversko pojačanje sistema
ke1=2546; %pojačanje enkodera diska
ke2=2608; %pojačanje enkodera klatna
a=k1/ksys/ke1;
b=k2/ksys/ke2;
c=k2/ke1;
d=k3/ke2;
f=k4/ke2;
c1=k5/ksys/ke1;

%Definisanje trajektorije flat izlaza
t=linspace(0,tf,200); %vremenski opseg
yf=c/f*theta1f*44.44; %krajnji uslov y(tf), svi ostali pocetni i krajnji jednaki nuli
y=yf*t.^5/tf.^5.*(126-420*t/tf+540*t.^2/tf.^2-315*t.^3/tf.^3+70*t.^4/tf.^4); %trajektorija flat izlaza
dy=yf*t.^4/tf.^5.*(630-2520*t/tf+3780*t.^2/tf.^2-2520*t.^3/tf.^3+630*t.^4/tf.^4); %prvi izvod flat izlaza
d2y=yf*t.^3/tf.^5.*(2520-12600*t/tf+22680*t.^2/tf.^2-17640*t.^3/tf.^3+5040*t.^4/tf.^4); %drugi izvod flat izlaza
d3y=yf*t.^2/tf.^5.*(7560-50400*t/tf+113400*t.^2/tf.^2-105840*t.^3/tf.^3+35280*t.^4/tf.^4); %treći izvod flat izlaza
d4y=yf*t./tf.^5.*(15120-151200*t/tf+453600*t.^2/tf.^2-529200*t.^3/tf.^3+211680*t.^4/tf.^4); %četvrti izvod flat izlaza

%Parametrizacija varijabli sistema
theta2=d2y/45.51; %trajektorija ugla klatna
theta1=(-d/c*d2y+f/c*y)/44.44; %trajektorija ugla diska
T=c1/c*(f*dy-d*d3y)+a/c*(f*d2y-d*d4y)+b*d4y; %trajektorija momenta motora
yn=c/f*theta1+d/f*theta2;

%Brzine i ubrzanja diska i klatna
theta2d=d3y/45.51;
theta2d2=d4y/45.51;
theta1d=(-d/c*d3y+f/c*dy)/44.44;
theta1d2=(-d/c*d4y+f/c*d2y)/44.44;

%Crtanje grafika
plot(t,theta1,t,theta1d,t,theta1d2);
xlabel('vrijeme [s]);
```

```

ylabel('ugao \theta_1 [stepeni]');
xlim([0 tf]);
figure
plot(t,theta2,t,theta2d,t,theta2d2);
xlabel('vrijeme [s]');
ylabel('ugao \theta_2 [stepeni]');
xlim([0 tf]);
figure
plot(t,T);
xlabel('vrijeme [s]');
ylabel('upravlj. ulaz [contr. counts]');
xlim([0 tf]);
figure
plot(t,y_n);
xlabel('vrijeme [s]');
ylabel('flat izlaz y [stepeni]');
xlim([0 tf]);

```

LQRModel.m – model u prostoru stanja za LQR upravljanje

```

INV=1 % 1 za invertovani, -1 za neinvertovani model
yw=INV*.32; %udaljenost od diska do tega
yr=INV*.42; %udaljenost od diska do kraja stapa klatna

```

```

%Fiksni parametri
mw=0.089; %masa tega (kg)
mr=.069;% %masa stapa (kg)
Lr=INV*.43; %ukupna duzina stapa (m)
c1=.02; %viskozno trenje na disku
grav=9.81; %gravitaciono ubrzanje (m/s^2)
J1=.0166; %Moment inercije diska
Rh=0.25; %Udaljenost centra diska do spoja sa klatnom (m)
Ksys =.00675; %hardversko pojacanje sistema
Kenc1=2546; % pojacanje enkodera diska (counts/rad)
Kenc4=2608; % pojacanje enkodera klatna (counts/rad)

```

```

% Izvedene velicine
m=mr+mw;
lcgr=yr-Lr/2;
lcg=(mr*lcgr+mw*yw)/m;

```

```

%Izvedene inercijalne velicine
J1_=J1+m*Rh^2;
Jz_=mr*(Lr^2/12+lcgr^2)+mw*yw^2;
Jy=0;

```

```

%Prenosne funkcije
N1=Kenc1*Ksys*[Jz_ 0 -m*grav*lcg];
N2=-[Kenc4*Ksys*m*Rh*lcg 0];
imenilac=([J1_+Jy]*Jz_-(m*Rh*lcg)^2 c1*Jz_ -(J1_+Jy)*m*grav*lcg -c1*m*grav*lcg];

```



```

N1=N1/imenilac(1),N2=N2/imenilac(1),imenilac=imenilac/imenilac(1); %Imenilacpostaje monicni
polinom, tj. koef. najveceg stepena je 1
D1=[imenilac 0]
D2=imenilac

```

%Model u prostoru stanja

```

param=1/((J1_+Jy)*Jz_-(m*Rh*lcg)^2);
A1=[0 1 0 0];
A3=[0 0 0 1];
A2=param*[0 -c1*Jz_ -m^2*lcg^2*Rh*grav*Kenc1/Kenc4 0];
A4=param*[0 c1*m*Rh*lcg*Kenc4/Kenc1 (J1_+Jy)*m*grav*lcg 0];
A=[A1;A2;A3;A4]
B=Ksys*param*[0;Kenc1*Jz_;0;-Kenc4*m*Rh*lcg]

```

LQRpolovi.m- iscrtavanje polova sistema u zatvorenoj petlji za različite vrijednosti tezninskog koeficijenta r. Prije pokretanja ovog fajla pokrenuti LQRModel.m

```

C=[1 0 0 0]
Q=C'*C;

```

%Određivanje matrice pojačanja K upotrebom komande lqr

```

Klqr1=lqr(A,B,Q,1000)
Klqr2=lqr(A,B,Q,100)
Klqr3=lqr(A,B,Q,10)
Klqr4=lqr(A,B,Q,1)
Klqr5=lqr(A,B,Q,.1)

```

%Polovi sistema u zatvorenoj petlji

```

P1=eig(A-B*Klqr1)/2/pi
P2=eig(A-B*Klqr2)/2/pi
P3=eig(A-B*Klqr3)/2/pi
P4=eig(A-B*Klqr4)/2/pi
P5=eig(A-B*Klqr5)/2/pi

```

% Crtanje polova

```

plot(P1,'x')
hold;
plot(P2,'o');
plot(P3,'*');
plot(P4,'+');
plot(P5,'.');
xlabel('Realna osa (Hz)')
ylabel('Imaginarna osa (Hz)')
legend('r=1000','r=100','r=10','r=1','r=0.1')
grid;
axisnormal

```

Prilog 2 – Kodovi za izvršni program

Feedforward algoritam za neinverzni slučaj

```
;Definicija varijabli
#define yr q1
#define yr1 q2
#define yr2 q3
#define yr3 q4
#define yr4 q5
#define T q6
#define Tf q7
#define TTf q8 ;T/Tf
#define Tf2 q9 ;Tf^2
#define TTf2 q14 ;(T/Tf)^2
#define Ts q15
#define a q16
#define b q17
#define c q18
#define d q19
#define c1 q20
#define f q21
#define theta1f q22
#define yf q23
#define var1 q10
#define var2 q11

;Inicijalizacija definisanih varijabli
yr=0
yr1=0
yr2=0
yr3=0
yr4=0
Ts=0.00884
T=0
Tf=1
Tf2=Tf*Tf
TTf=T/Tf
TTf2=TTf*TTf
a=0.0265
b=-0.0107
c=b
d=0.0131
f=-0.4182
```

```

c1=0.02
theta1f=30*3.14/180
yf=c/f*theta1f

;Servo petlja
begin
    if(T!>Tf AND cmd_pos>0)
        yr=yf*TTf2*TTf2*TTf*(126-420*TTf+540*TTf*TTf-315*TTf2*TTf+70*TTf2*TTf2)
        yr1=yf*TTf2*TTf2/Tf*(630-2520*TTf+3780*TTf*TTf-
2520*TTf2*TTf+630*TTf2*TTf2)
        yr2=yf*TTf2*TTf/Tf2*(2520-12600*TTf+22680*TTf*TTf-
17640*TTf2*TTf+5040*TTf2*TTf2)
        yr3=yf*TTf2/Tf2/Tf*(7560-50400*TTf+113400*TTf*TTf-
105840*TTf2*TTf+35280*TTf2*TTf2)
        yr4=yf*TTf/Tf2/Tf2*(15120-151200*TTf+453600*TTf*TTf-
529200*TTf2*TTf+211680*TTf2*TTf2)
        control_effort=(c1/c*(f*yr1-d*yr3)+a/c*(f*yr2-d*yr4)+b*yr4)*4740
        T=T+Ts
        TTf=T/Tf
        TTf2=TTf*TTf
        var1=(f/c*yr-d/c*yr2)*2546
        var2=yr2*2608
    endif
    if(T>Tf OR abs(control_effort)>10000)
        control_effort=0
    endif
end
end

```

LQR algoritam

```

#define kp_se q1
#define kd_se q2
#define kd_se_d q3
#define k1 q4
#define k2 q5
#define k3 q6
#define k4 q7
#define k2d q8
#define k4d q9
#define kpf q14
#define past_pos1 q15
#define past_pos4 q16
#define uval q17
#define Ts q18

```

```

#define flag q20
#define se_cmd_pos q21
#define enc4_delta q22
#define sign q23
#define enc4_offset q24
#define enc4_inv q25
#define se_step_ampl q26
#define dist q27

;Inicijalizacija varijabli
Ts=0.00884
kp_se=2
kd_se=.12
kd_se_d=kd_se/Ts
flag=0
se_step_ampl=280
se_cmd_pos=0
k1=-.316
k2=-.107
k3=-1.01
k4=-0.174
k2d=k2/Ts
k4d=k4/Ts
kpf=k1
q10=0
dist=0

;Servo petlja
begin
enc4_delta=enc4_pos-past_pos4
if (abs(enc4_pos)/32<7600)
    if (flag=0 and abs(enc4_pos)/32>300)
        flag=1
        sign=enc4_delta/sqrt(enc4_delta*enc4_delta)
        se_cmd_pos=sign*se_step_ampl*32
    endif
    if (flag=1 and abs(enc4_pos)/32<300)
        sign=enc4_delta/abs(enc4_delta)
        se_cmd_pos=sign*se_step_ampl*32
    endif
    control_effort=kp_se*(se_cmd_pos-enc1_pos)-kd_se_d*(enc1_pos-past_pos1)
endif
if (abs(enc4_pos)/32 > 7400 and flag=1)
    sign=enc4_pos/abs(enc4_pos)

```

```

        enc4_offset=sign*8192*32
        flag=2
        control_effort=0
endif
if (flag=2)
    enc4_inv=enc4_pos-enc4_offset
    q12=enc4_offset
    if(cmd_pos>666*32 and cmd_pos<1332*32)
        dist=0.3
    else
        dist=0
    endif
    uval=kpf*cmd_pos-k1*enc1_pos-k3*enc4_inv-k2d*(enc1_pos-past_pos1)-k4d*(enc4_pos-
past_pos4)+dist*4740
    control_effort=uval
endif
    past_pos1=enc1_pos
    past_pos4=enc4_pos
q10=enc4_inv/32
q11=dist
end

```

Flatness algoritam

```

#define yr q1
#define yr1 q2
#define yr2 q3
#define yr3 q4
#define yr4 q5
#define T q6
#define Tf q7
#define TTf q8 ;T/Tf
#define Tf2 q9 ;Tf^2
#define TTf2 q14 ;(T/Tf)^2
#define Ts q15
#define a q16
#define b q17
#define c q18
#define d q19
#define c1 q20
#define f q21
#define theta1f q22
#define yf q23
#define y q24

```

```

#define y1 q25
#define y2 q26
#define y3 q27
#define v q28
#define past_pos1 q29
#define past_pos4 q30
#define var1 q10
#define var2 q11
#define var3 q12
#define var4 q13
#define enc4_delta q31
#define sign q32
#define se_cmd_pos q33
#define se_step_ampl q34
#define kp_se q35
#define kd_se q36
#define enc4_offset q37
#define enc4_inv q38
#define flag q39
#define kd_se_d q40
#define uval q41
#define kpf q43
#define k1 q44
#define k2 q45
#define k3 q46
#define k4 q47
#define dist q48

```

;Inicijalizacija definisanih varijabli

Ts=0.00884

T=0

Tf=2

Tf2=Tf*Tf

TTf=T/Tf

TTf2=TTf*TTf

a=0.0265

b=0.0107

c=b

d=0.0131

f=0.4182

c1=0.02

theta1f=30*3.14/180

yf=c/f*theta1f

flag=0

```

kp_se=2
kd_se=0.12
se_step_ampl=280
se_cmd_pos=0
kd_se_d=kd_se/Ts
k1=-0.316
k2=-0.139
k3=-1.22
k4=-0.216
kpf=k1
dist=0

;Servo petlja
begin
enc4_delta=enc4_pos-past_pos4
if (abs(enc4_pos)/32<7600)
    if (flag=0 and abs(enc4_pos)/32>300)
        flag=1
        sign=enc4_delta/sqrt(enc4_delta*enc4_delta)
        se_cmd_pos=sign*se_step_ampl*32
    endif
    if (flag=1 and abs(enc4_pos)/32<300)
        sign=enc4_delta/abs(enc4_delta)
        se_cmd_pos=sign*se_step_ampl*32
    endif
    control_effort=kp_se*(se_cmd_pos-enc1_pos)-kd_se_d*(enc1_pos-past_pos1)
endif
if(abs(enc4_pos)/32> 7400 and flag=1)
    sign=enc4_pos/abs(enc4_pos)
    enc4_offset=sign*8192*32
    flag=2
    control_effort=0
endif
if (flag=2)
    enc4_inv=enc4_pos-enc4_offset
    uval=kpf*cmd_pos-k1*enc1_pos-k3*enc4_inv-k2*(enc1_pos-past_pos1)/Ts-k4*(enc4_pos-
past_pos4)/Ts
    control_effort=uval
endif
if (cmd_pos>0)
    flag=3
endif
if(flag=3 and T<Tf)
    if(T>1 and T<1.5)

```

```

dist=0 ; 0.3 za smetnju
else
dist=0
endif
yr=yf*TTf2*TTf2*TTf*(126-420*TTf+540*TTf*TTf-315*TTf2*TTf+70*TTf2*TTf2)
yr1=yf*TTf2*TTf2/Tf*(630-2520*TTf+3780*TTf*TTf-2520*TTf2*TTf+630*TTf2*TTf2)
yr2=yf*TTf2*TTf/Tf2*(2520-12600*TTf+22680*TTf*TTf-
17640*TTf2*TTf+5040*TTf2*TTf2)
yr3=yf*TTf2/Tf2/Tf*(7560-50400*TTf+113400*TTf*TTf-
105840*TTf2*TTf+35280*TTf2*TTf2)
yr4=yf*TTf/Tf2/Tf2*(15120-151200*TTf+453600*TTf*TTf-
529200*TTf2*TTf+211680*TTf2*TTf2)
enc4_inv=enc4_pos-enc4_offset
y=c/f*enc1_pos/2546/32+d/f*enc4_inv/2608/32
y1=c/f*(enc1_pos-past_pos1)/Ts/2546/32+d/f*(enc4_pos-past_pos4)/Ts/2608/32
y2=enc4_inv/2608/32
y3=(enc4_pos-past_pos4)/Ts/2608/32
v=yr4-48*(y3-yr3)-864*(y2-yr2)-6912*(y1-yr1)-20736*(y-yr); polovi u -12
T=T+Ts
TTf=T/Tf
TTf2=TTf*TTf
var1=(f/c*yr-d/c*yr2)*2546
var2=yr2*2608
control_effort=(c1/c*(f*y1-d*y3)+a/c*(f*y2-d*v)+b*v+dist)*4740
var3=enc4_inv/32
var4=dist
endif
if(T!<Tf)
flag=4
yr=c/f*30*3.14/180
endif
if (flag=4)
enc4_inv=enc4_pos-enc4_offset
y=c/f*enc1_pos/2546/32+d/f*enc4_inv/2608/32
y1=c/f*(enc1_pos-past_pos1)/Ts/2546/32+d/f*(enc4_pos-past_pos4)/Ts/2608/32
y2=enc4_inv/2608/32
y3=(enc4_pos-past_pos4)/Ts/2608/32
v=-48*y3-864*y2-6912*y1-20736*(y-yr); polovi u -12
control_effort=(c1/c*(f*y1-d*y3)+a/c*(f*y2-d*v)+b*v)*4740
var3=enc4_inv/32
endif
past_pos1=enc1_pos
past_pos4=enc4_pos
end

```