### UNIVERSITATEA "POLITEHNICA" DIN BUCURESTI FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII ȘI TEHNOLOGIA INFORMAȚIEI

## CONTROLUL UNUI PENDUL INVERSAT CU ROATĂ INERȚIALĂ

# **PROIECT DE DIPLOMĂ**

Prezentată ca cerință parțială pentru obținerea titlului de Inginer în domeniul Electronică și Telecomunicații programul de studii: Microelectronică, Optoelectronică și Nanotehnlogii

**Conducător științific:** Prof. Dr. Ing. Corneliu BURILEANU Student: Vlad NICULESCU

București 2017

Anexa 1

Universitatea "Politehnica" din București Facultatea de Electronică, Telecomunicații și Tehnologia Informației Departamentul de Dispozitive, Circuite și Arhitecturi Electronice

Aprobat Director de Departament:

Prof. Dr. Ing. Claudius Dan

#### TEMA PROIECTULUI DE DIPLOMĂ a studentului Niculescu T. Vlad (441E)

1. Titlul temei: Controlul unui pendul inversat cu roată inerțială

2. Contribuția practică, originală a studentului va consta în (*în afara* părții de documentare):

Proiectul va consta în realizarea unui tip particular de pendul inversat ce va fi controlat de o roată inerțială.

Montajul practic va conține un cadru de aluminiu pe care va fi montat un motor Maxon EC 45 flat, controllerul acestuia ESCON 36/3 EC, placa de dezvoltare cu microcontroller ce va comanda controllerul și un giroscop BNO055. Cadrul de aluminiu va fi prins de suprafața orizontală prin intermediul unei balamale, lucru ce ii va permite să oscileze față de poziția verticală. Prin acționarea corespunzătoare a roții inerțiale, întreg sistemul poate fi menținut în echilibru. Giroscopul comunică în permanență poziția curentă a cadrului.

Principalul scop al proiectului este investigarea mai multor algoritmi de control și observarea rezultatelor obținute pentru fiecare. Criteriile de evaluare pentru fiecare dintre acești algoritmi vor fi: stabilitatea în poziția de echilibru, intervalul unghiular pentru care sistemul poate funcționa, timpul necesar pentru atingerea stării de echilibru.

Microcontrollerul va primi informații de la giroscop și în funcție de acestea va lua o decizie asupra comenzii driverului. Controlul motorului se va face în curent, iar valoarea curentului dorit va fi comunicată de microcontroller prin modificărea factorului de umplere a unui semnal PWM.

**3.** Proiectul se bazează pe cunoștințe dobândite în principal la următoarele 3-4 discipline: Microcontrolere, AMP, Programarea Calculatoarelor.

4. Proprietatea intelectuală asupra proiectului aparține: UPB

5. Locul de desfășurare a activității: UPB

6. Realizarea practică rămâne în proprietatea: studentului

7. Data eliberării temei: 13.12.2016

**CONDUCĂTOR LUCRARE:** 

**STUDENT:** 

Corneliu Burileanu

Niculescu Vlad

### DECLARAȚIE DE ONESTITATE ACADEMICĂ

Prin prezenta declar că lucrarea cu titlul "Controlul unui pendul inversat cu roată inerțială", prezentată în cadrul Facultății de Electronică, Telecomunicații și Tehnologia Informației a Universității "Politehnica" din București ca cerință parțială pentru obținerea titlului de Inginer în domeniul Inginerie Electronică și Telecomunicații, programul de studii Microelectronică Optoelectronică și Nanotehnologii, este scrisă de mine și nu a mai fost prezentată niciodată la o facultate sau instituție de învățământ superior din țară sau străinătate.

Declar că toate sursele utilizate, inclusiv cele de pe Internet, sunt indicate în lucrare, ca referințe bibliografice. Fragmentele de text din alte surse, reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și fac referință la sursă. Reformularea în cuvinte proprii a textelor scrise de către alți autori face referință la sursă. Înțeleg că plagiatul constituie infracțiune și se sancționează conform legilor în vigoare.

Declar că toate rezultatele simulărilor, experimentelor și măsurătorilor pe care le prezint ca fiind făcute de mine, precum și metodele prin care au fost obținute, sunt reale și provin din respectivele simulări, experimente și măsurători. Înțeleg că falsificarea datelor și rezultatelor constituie fraudă și se sancționează conform regulamentelor în vigoare.

București, 26.06.2017

Vlad NICULESCU

# CUPRINS

Lista de figuri	9
Lista de tabele	11
Lista de acronime	13
Introducere	15
CAPITOLUL 1 Noțiuni introductive	17
1.1 Pendulul inversat	17
1.2 Roata inerțială	18
1.3 Ansamblul pendul inversat – roată inerțială	18
1.4 Scopul proiectului	19
CAPITOLUL 2 Noțiuni teoretice	21
2.1 Sistemele fără reglaj	21
2.2 Sistemele cu reglaj automat	21
2.3 Algoritmii de control	23
2.3.1 Algoritmul de control PID	23
2.3.2 Control în spațiul stărilor	24
2.3.3 Algoritmul de control LQR	27
CAPITOLUL 3 Proiectarea sistemului fizic	29
3.1 Structura mecanică	29
3.2 Structura circuitului electronic	31
3.2.1 Sursa în comutație Mean Well LRS-150-24	33
3.2.2 Convertorul în comutație D24C10F5	33
3.2.3 Placa de dezvoltare Arduino Uno	33
3.2.4 Senzorul de mișcare BNO055	35
3.2.5 Controlerul de motor Maxon ESCON 36/3 EC	35
CAPITOLUL 4 Analiza procesului	40
4.1 Analiza sistemului folosind modele teoretice	40
4.2 Identificarea sistemului	43
CAPITOLUL 5 Proiectarea și testarea algoritmilor	47
5.1 Algoritmul PID	47
5.2 Algoritmul LQR	50
5.3 Proiectarea unui algoritm adaptiv	52
CAPITOLUL 6 Concluzii	55
Bibliografie	57
Anexa 1	59
Anexa 2	64

# LISTA DE FIGURI

Figure 1.1 Pendulul inversat clasic	
Figure 1.2 Structura mecanică a proiectului	19
Figure 2.1 Sistem fără reglaj	21
Figure 2.2 Sistem cu reacție	22
Figure 2.3 Răspunsul la semnal treaptă pentru regulatorul P, respectiv PD	24
Figure 2.4 Structura unui model de stare	25
Figure 2.5 Sensul semnalelor	25
Figure 2.6 Modelul de stare cu reacție	
Figure 2.7 Sistemul echivalent	26
Figure 3.1 Montajul practic	
Figure 3.2 Schema bloc a părții electronice	
Figure 3.3 Susra în comutație ce alimentează montajul	
Figure 3.4 Convertorul în comutație de 5V	
Figure 3.5 Placa de dezvoltare Arduino Uno	
Figure 3.6 Senzorul de mișcare	35
Figure 3.7 Rezultatul operației de reglare a driverului de motor	36
Figure 3.8 Controlerul motorului	
Figure 3.9 Sumatorul cu două rezistențe	37

Figure 3.10 Interfața de test a motorului	
Figure 4.1 Identificarea variabilelor din modelul teoretic	41
Figure 4.2 Analiza stabilității procesului	42
Figure 4.3 Procesul privit ca o cutie neagră	43
Figure 4.4 Achiziția setului de date experimentale	44
Figure 4.5 Răspunsul la impuls al modelului găsit	45
Figure 5.1 Etapele execuției algoritmului	47
Figure 5.2 Răspunsul la semnal treaptă al sistemului cu reacție	48
Figure 5.3 Diagrama poli - zerouri	49
Figure 5.4 Etapele execuției algoritmului LQR	50
Figure 5.5 Răspunsul la impuls al algoritmului LQR	51
Figure 5.6 Diagrama poli – zerouri pentru LQR	52
Figure 5.7 Schema bloc a algoritmului adaptiv	53

# LISTA DE TABELE

Table 3.1 Dimensionile componentelor mecanice	30
Table 3.2 Pinii folosiți și funcția lor	34
Table 3.3 Parametrii configurați	36
Table 3.4 Pinii microcontrolerului ce interacționează cu driverul de motor	37
Table 4.1 Dimensiunile pendulului	41
Table 4.2 Setul de date achiziționte	44
Table 5.1 Constantele algoritmului PID	49
Table 5.2 Constantele algoritmului LQR	50
Table 5.3 Constantele algoritmului PD - PD	54
Table 5.5 Constantele algorithmului PD - PD	

# LISTA DE ACRONIME

**PID** - Proportional, Integrative, Derivative

LQR - Linear Quadratic Regulator

MPC - Model Predictive Control

UART - Universal Asynchronous Receiver/Transmitter

**USB** - Universal Serial Bus

## Introducere

Lucrarea constă in analiza unor algoritmi de control ce servesc la stabilizarea unui pendul inversat bazat pe o roata inerțiala. Roata este acționată de un motor fără perii ce transmite o mișcare de rotație întregului pendul prin variația turației motorului. In dezvoltarea algoritmilor de control se va determina mai întâi comportamentul sistemului ce trebuie stabilizat, prin obținerea unor ecuații matematice ce îl caracterizează. Având acest model, se va putea proiecta și simula regulatorul. In urma simulării, soluția va fi portată pe sistemul fizic, unde se vor face ajustări in vederea maximizării performantelor. Programul folosit pentru a realiza simulările este MATLAB, deoarece acesta prezintă funcții si pachete ce simplifică mult scrierea codului, in comparație cu alte medii de programare. Mărimile ce oferă feedback algoritmilor de control sunt oferite de un senzor de mișcare si un senzor Hall incorporat in motor. Criteriile de performanță pentru fiecare algoritm sunt: domeniul unghiular de funcționare, abaterea față de poziția de echilibru in regim staționar si sensibilitatea la perturbații mecanice exterioare. Tema a fost aleasă datorită multiplelor aplicații ale algoritmilor de control în industria automobilelor și a ingineriei aerospațiale.

# CAPITOLUL 1

# NOȚIUNI INTRODUCTIVE

#### 1.1 PENDULUL INVERSAT

Un pendul inversat este un pendul care are centrul de masă deasupra punctului său de pivotare. Implementarea clasică a pendulului inversat constă într-un cărucior ce are un pivot fixat pe suprafața superioară. În jurul pivotului se rotește o tijă de greutate neglijabilă, ce are atașat un corp fix in capătul celălalt. Prin acționarea căruciorului cu o forța întru-un sens, va apărea o alta forță de inerție de sens opus ce va acționa asupra corpului și va genera o mișcare de rotație a tijei in jurul pivotului, ca in Figura 1.1 . Problema controlului constă în acționarea corespunzătoare a căruciorului cu o forță pe axa orizontală, pentru a menține tija in echilibru, adică la un unghi de zero grade cu axa verticală. Astfel, ansamblul format din cărucior si pendul reprezintă procesul, iar mecanismul care acționează căruciorul cu o forță proporțională cu unghiul pendulului reprezintă reacția negativă. Pendulul inversat reprezintă o problemă clasică în teoria a controlului și este folosit ca punct de referință pentru evaluarea algoritmilor de control [10].



Figura 1.1 Pendulul inversat clasic [11]

Pendulul inversat are aplicații in ghidarea rachetelor, unde centrul de greutate se află mai sus decât punctului de tracțiune, determinând instabilitate aerodinamică.

### 1.2 Roata Inerțială

O roată inerțială este un tip de volant utilizat în principal de navele spațiale pentru controlul altitudinii fără a folosi combustibil. Acesta este deosebit de utilă atunci când sunt necesare rotiri foarte fine, cum ar fi păstrarea unui telescop îndreptat către o stea. Mecanismul se realizează echipând un corp cu un motor electric atașat la un volant care, atunci când își schimbă viteza de rotație, determină o mișcare de rotație a corpului proporțională cu accelerația unghiulară a volantului, prin conservarea momentului unghiular. Roțile de reacție pot roti corpul de care sunt atașate doar in jurul centrului de masă. Ele nu sunt capabile să transfere aceste corpuri de la un loc la altul. Roțile de reacție funcționează în jurul unei viteze nominale de rotație zero. Cu cât volantul își schimba viteza de rotație mai rapid, cu atât mișcarea de rotație imprimată sistemului este mai puternică. Pentru a maximiza mișcarea de rotație ce este transmisă corpului, se folosește următoarea tehnică: mai întâi se duce volantul la o viteză de rotație constantă de valoare mare. Apoi acesta este supusă unui mecanism de frânare exterior, ce aduce viteza de rotație la zero într-un interval de timp foarte scurt. In acest fel se obține o accelerație negativă de valoare foarte mare, ce se traduce printr-o forță uriașă ce acționează asupra întregului ansamblu.

### 1.3 Ansamblul Pendul Inversat - Roată Inerțială

Prin fuzionarea celor doua concepte se obține unul nou, ce constituie subiectul de interes al acestei lucrări. Practic, în contextul acestui caz particular de pendul inversat, mecanismul reglare ce aduce pendulul înapoi în poziția de echilibru este roata inerțială [2].

Structura mecanică a sistemului aferent acestui proiect de diploma este prezentată in Figura 1.2. Pendulul inversat este reprezentat de cadrul metalic de forma pătrată. În centrul acestuia este atașat un motor ce acționează roata inerțială evidențiată cu verde in Figura 1.2. Întreg ansamblul se poate roti în jurul pivotului de culoare roșie cu frecări foarte mici. Momentul cinetic ce provoacă mișcarea cadrului este generat de cuplul produs de motor. In concluzie, motorul trebuie sa își ajusteze in permanență acest cuplu pentru a controla corespunzător modul in care cadrul se mișcă [7].



Figura 1.2 Structura mecanică a proiectului

### 1.4 Scopul Proiectului

Principalele focusuri ale proiectului sunt:

- 1. Dezvoltarea mai multor algoritmi capabili să controleze si sa mențină cadrul metalic din Figura 1.2 in poziție de echilibru. Poziția de echilibru se definește ca acea poziție în care dreapta ce trece prin vârful pătratului atașat pivotului si vârful opus este perpendiculara pe planul orizontal.
- 2. Analiza acestor algoritmi de reglare automata si observarea performanțelor fiecăruia. Montajul este echipat cu un giroscop și cu traductoare Hall ce furnizează feedback cu privire la starea curentă în care se află sistemul și oferă datele de intrare pentru blocul de control. Cu alte cuvinte, acestea ajută la realizarea reacției negative. În funcție de algoritmul folosit, se va alege care dintre acești senzori va fi plasat in bucla de reacție negativă.

Criteriile de performanță sunt:

- Domeniul unghiular de funcționare al sistemului
- Sensibilitatea la perturbații mecanice externe
- Amplitudinea oscilațiilor în poziția de echilibru când nu există un factor perturbator adițional.

Prima iterație a acestui proiect s-a numit "Cubli" și a fost dezvoltat în Institutul de Dinamică și Control din ETH. Din acest proiect au fost preluate ecuațiile matematice ale sistemului.

Componentele ce alcătuiesc pendulul le-am preluat de la Sorina Lupu, ce a realizat un proiect similar pentru proiectul ei de diplomă în 2015.

Deși montajul practic este foarte similar cu cel din proiectele menționate, scopul proiectului meu de diplomă este cu totul diferit. Acesta nu își propune doar stabilizarea pendulului, ci și compararea performanțelor algoritmilor folosiți.

# CAPITOLUL 2

# NOȚIUNI TEORETICE

#### 2.1 SISTEMELE FĂRĂ REGLAJ

Un sistem fără reglaj automat sau manual funcționează în circuit deschis, adică nu prezintă niciun fel de mecanism de reacție, după cum se vede în Figura 2.1. Acestea se mai numesc si circuite deschise. In cazul acestor sisteme, comportamentul ieșirii procesului trebuie bine cunoscut pentru a proiecta regulatorul, deoarece în timpul funcționarii ieșirea nu va fi observabilă.



Figura 2.1 Sistem fără reglaj

### 2.2 SISTEMELE CU REGLAJ AUTOMAT

Un sistem cu reacție este unul în care semnalul de ieșire este eșantionat, iar apoi furnizat înapoi intrării pentru a forma un semnal de eroare care este furnizat regulatorului pentru a ajusta corespunzător intrarea procesului. În acest fel, daca regulatorul este implementat corect, acesta poate controla procesul în vederea minimizării acelui semnal de eroare. În Figura 2.2, cu R s-a notat regulatorul, iar cu P procesul.

In contextul unui sistem cu reglare automată, senzorii au rolul de a furniza informații referitoare la starea curentă în care se afla sistemul. Aceștia măsoară si convertesc mărimile fizice din proces în mărimi electrice si furnizează datele de intrare pentru regulatorul ce controlează procesul. Pentru a alege corespunzător un traductor, se iau in calcul următorii parametrii: rezoluția senzorului, coeficientul de rejecție al zgomotului, frecvența maximă de lucru.



Figura 2.2 Sistem cu reacție

Pentru implementarea algoritmului de reglaj este mai întâi necesară cunoașterea modelului matematic ce caracterizează procesul. Modelarea matematică a proceselor și fenomenelor reprezintă determinarea unui set de relații între variabile fizice, ca structuri matematice de tipul ecuațiilor algebrice, pentru o caracterizare cât mai apropiată de realitate a funcționarii procesului respectiv.

Pentru determinarea comportamentului procesului, aceste ecuații se pot obține fie utilizând legi și teoreme ale fizicii pentru a obține modelul teoretic, fie identificând sistemul pe cale experimentală.

Prin identificarea unui sistem se înțelege modelarea unui proces folosind date experimentale ce au fost achiziționate pe parcursul funcționarii acestuia. În urma acestei etape se obține modelul experimental, adică o reprezentare abstractă a unei entități reale. Modelele pot fi liniare sau neliniare.

Obținerea modelului pe cale experimentală presupune următoarele etape:

- Achiziția unui set de eșantioane atât pentru perturbația ce a fost aplicată la intrarea sistemului, cât și pentru răspunsul măsurat la ieșirea acestuia;
- Stabilirea unei structuri matematice pentru modelul respectiv;
- Găsirea coeficienților acelei structuri ce caracterizează cel mai bine setul de date achiziționat.

Parcurgerea acestor etape implică următoarea presupunere: modelul matematic ales este suficient de complex pentru a putea caracteriza setul de date într-o proporție suficient de mare. În caz contrar, modelul procesului va fi unul eronat, lucru care va conduce la o proiectare eronată a regulatorului. De exemplu, o funcție de gradul doi, nu poate fi estimată cu ajutorul unei funcții liniare.

În general, pentru estimarea matematica a comportamentului procesului se folosește fie o funcție de transfer de ordin unu, fie una de ordin doi [3].

$$H_1(s) = \frac{K_P}{T_p \cdot s + 1} \cdot e^{-\tau s}$$
(2.1)

Unde  $K_P$  este coeficientul de amplificare,  $T_p$  reprezintă constanta de timp a procesului și  $\tau$  timpul mort.

$$H_2(s) = \frac{\omega_n^2}{s^2 + 2 \cdot \zeta \cdot \omega_n + \omega_n^2}$$
(2.2)

Unde  $\omega_n$  este pulsația naturală, iar  $\zeta$  factorul de amortizare. Pentru acest tip de sistem, forma răspunsului la impuls depinde de factorul de amortizare. Pentru orice valoare pozitivă a acestuia, sistemul este stabil.

Desigur, funcția de transfer obținută pentru proces caracterizează un sistem instabil. Scopul regulatorului este de a fi pus in buclă cu procesul si a forma un sistem global stabil.

Forma generală a unei funcții de transfer este [8]:

$$H(s) = \frac{b_0 + b_1 \cdot s + \dots + b_m \cdot s^m}{a_0 + a_1 \cdot s + \dots + a_n \cdot s^n}$$
(2.3)

Fie  $H_P(s)$  funcția de transfer a procesului, iar  $H_R(s)$  funcția de transfer a regulatorului. Dacă  $\varepsilon$  este semnalul eroare rezultat în urma diferenței între semnalul de referință și cel provenit din bucla de reacție, atunci:

$$out = \varepsilon \cdot H_P(s) \cdot H_R(s) \tag{2.4}$$

Deoarece regulatorul si procesul sunt legate în serie, iar  $\varepsilon$  reprezintă intrarea regulatorului.

$$\varepsilon = r - out \tag{2.5}$$

Unde r este semnalul de referință. Așadar:

$$out = (r - out) \cdot H_P(s) \cdot H_R(s)$$
(2.6)

$$out \cdot (1 + H_P(s) \cdot H_R(s)) = r \cdot H_P(s) \cdot H_R(s)$$
(2.7)

$$\frac{out}{r} = \frac{H_P(s) \cdot H_R(s)}{1 + H_P(s) \cdot H_R(s)}$$
(2.8)

#### 2.3 Algoritmii De Control

#### 2.3.1 Algoritmul de control PID

Un regulator PID calculează continuu o valoare de eroare e(t) ca diferența între o valoare de referință dorită și o variabilă de proces măsurată, iar apoi aplică o corecție bazată pe valoarea instantanee a erorii, pe derivata și integrala acesteia.

$$u(t) = K_P \cdot e(t) + K_I \cdot \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}$$
(2.9)

Aplicând transformata Laplace obținem:

$$U(s) = K_P + K_D \cdot \frac{1}{s} + K_D \cdot s \tag{2.10}$$

Unde  $K_P$ ,  $K_I$ ,  $K_D$ , sunt constante ce ponderează termenul proporțional, pe cel integral si respectiv derivativ. Proiectarea regulatorului presupune găsirea valorilor corespunzătoare ale acestor constante.

Semnificația fizica a fiecărui termen:

- Termenul proporțional compensează valoarea curentă a erorii. Daca de exemplu eroarea este de valoare mare și pozitivă, atunci si termenul proporțional va fi de valoare mare si pozitiv.
- Termenul ce însoțește integrala acționează în funcție de suma valorilor precedente ale erorii. De exemplu, dacă ieșirea curentă nu este suficient de puternică, integrala erorii se va acumula în timp, iar controlerul va răspunde aplicând o acțiune mai puternică.

• Termenul derivativ compensează posibile tendințe viitoare ale erorii, pe baza ratei actuale de variație. De exemplu, când termenul proporțional aduce eroarea spre o valoare apropiată de zero, acesta duce eroarea către o valoare negativă, apărând oscilații amortizate ce conduc la un timp de stabilizare mai mare. Termenul derivativ atenuează acest "overshoot" si asigura o convergenta mai rapidă.

În proiectarea unui regulator nu este obligatorie introducerea tuturor termenilor. Se pot realiza cu succes regulatoare de tip PI sau PD. Regulatoarele PI sunt preferate atunci când mărimea eșantionată de bucla de reacție este foarte zgomotoasă, iar termenul derivativ ar amplifica aceste perturbații.



Figura 2.3 Răspunsul la semnal treaptă pentru regulatorul P, respectiv PD

Un exemplu al performanței unui algoritm PD în raport cu un simplu P este ilustrat in Figura 2.3. Se poate vedea că prin adăugarea unui termen derivativ sistemul converge către poziția de echilibru mai rapid si mai lin.

Având în vedere că sistemul abordat este unul digital, se va folosi forma discretă a regulatorului PID. În aceasta, integrala se înlocuiește cu suma eșantioanelor precedente, iar derivata se scrie ca diferența a două eșantioane consecutive:

$$u(n) = K_P \cdot e(n) + K_D \cdot \sum_{0}^{N} u(n) + \frac{K_D}{T} \cdot (u(n) - u(n-1))$$
(2.11)

Unde T este perioada de eşantionare.

#### 2.3.2 Control în spațiul stărilor

Spațiul stărilor este o metodă foarte comună de analiză și control al sistemelor. Stările reprezintă un set de variabile ale sistemului care prezic în mod clar comportamentul sistemului în timp. Cu alte cuvinte, pentru a avea un model al unui sistem care să prezică comportamentul său, trebuie, în general, condiții inițiale. În plus, știind valoarea fiecărei variabile de stare la un moment de timp, avem toată informația ce caracterizează sistemul în acel moment [6].

În contextul acestui tip de control, ecuațiile matematice ce caracterizează sistemul sunt scrise sub forma matriceală. Putem exprima relațiile din interiorul sistemului cu ajutorul a doua ecuații numite ecuațiile spațiului de stări, unde A, B, C și D sunt matrici, iar x este vectorul de stare, u este intrarea sistemului si y este ieșire [9].

$$x[n] = A \cdot x[n-1] + B \cdot u[n-1]$$
(2.12)

$$y[n] = C \cdot x[n] + D \cdot u[n] \tag{2.13}$$

Aceste ecuații pot fi reprezentate cu ajutorul diagramei din Figura 2.4, unde blocurile galbene reprezintă matricile sistemului, însă semnalele ce pot circula prin fire pot fi vectori sau matrici.

Figura 2.5 evidențiază drumul informației. Matriciile A si B indică în ce măsură stările curente ale sistemului le vor influenta pe cele viitoare.



Figura 2.4 Structura unui model de stare

Matricea A descrie modul în care stările sistemului se influențează reciproc de la un moment de timp la următorul. În această matrice se găsesc de obicei relații ce provin din legile fizicii. Matricea va avea dimensiunea de m  $\times$  m, unde m este numărul de stări. Matricea B indică cum sunt influențate stările sistemului de către intrarea acestuia.



Figura 2.5 Sensul semnalelor

Bucla evidențiată cu roșu este reprezentarea grafica a termenului  $A \cdot x[n-1]$ , iar linia punctată galbenă reprezintă termenul  $B \cdot u[n-1]$ .

Matricea C indică în ce măsură stările influențează direct ieșirea, iar matricea D modelează influența directă a intrării la ieșire. În majoritatea cazurilor, aceasta matrice este nulă, neexistând o cale directă de la intrare spre ieșire. Ca urmare, blocul aferent matricii D poate fi exclus din schemă, precum și legăturile acestuia.

Toată analiza de mai sus a fost pentru sistemul în buclă deschisă, deci pentru proces. În continuare va fi tratat cazul adăugării unui regulator în buclă cu acesta. Considerăm un vector K, pe care îl numim vectorul ponderilor de reacție. Pentru un sistem cu m stări, K va fi un vector de dimensiune 1 x m. Astfel, când acesta este înmulțit cu vectorul stărilor de dimensiune m x 1, va rezulta un scalar egal cu:

$$Kx = [K_1, K_2, \dots, K_m] \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} = \sum_{i=1}^m K_i x_i$$
(2.14)

Neglijând Matricea D și introducând vectorul de reacție obținem diagrama din Figura 2.6:



Figura 2.6 Modelul de stare cu reacție

In acest fel, la ieșirea blocului K se obține un scalar ce trebuie scăzut din valoarea referinței pentru a obține un semnal de eroare. Introducând semnalul de referința u' in ecuație, obținem:

$$x[n+1] = A \cdot x[n] + B \cdot (u'[n] - K \cdot x[n])$$
(2.15)

Desfacem paranteza și obținem:

$$x[n+1] = A \cdot x[n] + B \cdot u'[n] - BK \cdot x[n]$$
(2.16)

Înlocuind pe n cu n-1 se obține:

$$x[n] = A \cdot x[n-1] + B \cdot u'[n-1] - BK \cdot x[n-1]$$
(2.17)

Restructurând ecuația, aceasta devine:

$$x[n] = (A - BK) \cdot x[n-1] + B \cdot u'[n-1]$$
(2.18)

A cărei formă omogenă este:

$$x[n] = (A - BK)x[n - 1]$$
(2.19)

Înlocuind (A - BK) cu M, obținem:

$$x[n] = Mx[n-1]$$
(2.20)



Figura 2.7 Sistemul echivalent

Se observă că în lipsa reacției  $M \rightarrow A$ , ceea ce confirmă că ecuațiile sunt corecte.

Figura 2.7 ilustrează schema sistemului echivalent, ce include si reacția în componența sa. Practic am redus sistemul cu reacție la unul de aceeași formă cu sistemul în buclă deschisă, dar alți parametrii.

#### 2.3.3 Algoritmulde control LQR

Teoria controlului optimal se referă la operarea unui sistem dinamic la un cost minim. Cazul în care dinamica sistemului este descrisă de un set de ecuații diferențiale liniare, iar costul este descris de o funcție pătratică se numește problema LQ. Soluția acestor ecuații este furnizată de regulatorul liniar - cvadratic (LQR), un regulator cu reacție.

Fie un sistem analizat în intervalul de timp  $[t_0, t_1]$ .

$$\dot{x} = Ax + Bu \tag{2.21}$$

Funcția de cost se definește ca:

$$J = x^{T}(t_{1})F(t_{1})x(t_{1}) + \int_{t_{0}}^{t_{1}} (x^{T}Qx + u^{T}Ru + 2x^{T}Nu)dt$$
(2.22)

Relația ce minimizează valoarea costului este:

$$u = -Kx \tag{2.23}$$

Unde:

$$K = R^{-1}(B^T P(t) + N^T)$$
(2.24)

Iar P este găsit rezolvând ecuația diferențiala Ricatti:

$$A^{T}P(t) + P(t)A - (P(t)B + N)R^{-1}(B^{T}P(t) + N^{T}) + Q = -\dot{P}(t)$$
(2.25)

# CAPITOLUL 3

# PROIECTAREA SISTEMULUI FIZIC

### 3.1 Structura Mecanică

Figura 3.1 ilustrează implementarea practică a conceptelor discutate in primele capitole. Întregul ansamblu a fost pus pe o placa de plexiglas ce are atașate niște picioare de cauciuc, pentru a oferi o buna aderentă în contact cu vibrațiile generate de motor.

Părțile componente ale structurii cu pendul inversat sunt:

- Pivot
- Cadru de aluminiu
- Motor fără perii
- Suport pentru motor
- Segmente de plastic ce realizează conexiunea între motorul cu suport si cadru
- Roata cu striații în exterior

Pivotul și segmentele de plastic au fost realizate la imprimanta 3D și sunt confecționate din material ABS. Roata de culoare verde este fabricată din sticlo-textolit.

Principalul dezavantaj al acestui montaj este pivotul, deoarece mecanismul de prindere între acesta și cadrul mecanic se face prin intermediul unui șurub. Acest lucru implică frecări neliniare foarte mari, ce duc la imposibilitatea găsirii unui model matematic exact ce caracterizează sistemul.

Mărime	Dimensiune
Diametru roată	11.5 cm
Latură cadru metalic	14.5 cm
Grosime roată	0.5 cm
Distantă ax motor – punct de pivotare	9 cm

**Tabelul 3.1 Dimensiunile componentelor mecanice** 

Motorul folosit este un Maxon EC 45 flat. Acesta este alimentat la o tensiune de 24V, si poate atinge pana la 6700 rpm. Acesta este motivul pentru care a fost ales un motor fără perii în loc de un motor clasic de curent continuu, deoarece poate atinge viteze de rotație foarte mari. Ansamblul motor – roată, imprimă o mișcare de rotație sistemului atâta timp cât există accelerație unghiulară în rotația axului motorului. Pentru ca mișcarea imprimată sa fie puternică trebuie ca accelerația să fie menținută cât mai mult timp. Dacă valoarea maximă a numărului de rotații pe minut ar fi mai mică, atunci viteza unghiulara s-ar satura mai rapid, și implicit și accelerația unghiulară ar dura mai puțin. De aici nevoia de a avea o valoare cât mai ridicată pentru turația maximă. Nu a fost ales un motor mai performant, deoarece la valori foarte mari de rotații pe minut, ansamblul începe sa vibreze din cauza imperfecțiunilor de simetrie și duce întreg sistemul în starea de instabilitate.



Figura 3.1 Montajul practic

### 3.2 Structura Circuitului Electronic

Principalele blocuri electronice sunt:

- Placa de dezvoltare Arduino Uno cu microcontroler ATmega328P
- Controler motor Maxon ESCON 36/3 EC
- Giroscop BNO055
- Sursa în comutație coborâtoare de tensiune Pololu D24V10F5
- Sursa în comutație Mean Well LRS-150-24

Toată logica de control a motorului și partea de interogare senzori este gestionată de placa cu microcontroler. În plus, algoritmul rulează tot în microcontroler, sistemul fiind independent de orice unitate de procesare externă.

Pentru că montajul nu prezintă foarte multe conexiuni între blocurile sale, legăturile au fost făcute cu fire, evitându-se astfel realizarea unui cablaj imprimat.



Figura 3.2 Schema bloc a părții electronice

### 3.2.1 Sursa în comutație Mean Well LRS-150-24

Întreg montajul este alimentat din această sursă în comutație, care nu se află pe standul de test, ca celelalte placi de dezvoltare. Ea se alimentează la 220V AC, și prezintă o singură ieșire de 24V. Curentul maxim pe care îl poate furniza sursa este de 6.5A. Deși teoretic motorul necesită un curent mai mic, pentru că puterea sa nominala este 50W, iar tensiunea de alimentare 24V, acesta va consuma curent în conformitate cu regimul de funcționare în care se află. De exemplu, dacă acesta e comandat să accelereze cu valoarea maximă posibilă, el va necesita un curent de peste 4A pentru un interval de timp foarte scurt. Cât timp sursa de alimentare nu poate furniza acest curent, motorul nu va funcționa conform așteptărilor.



Figura 3.3 Sursa în comutație ce alimentează montajul<sup>[12]</sup>

Singurul circuit ce necesita 24V este controlerul motorului. Celelalte au o tensiune de funcționare egală cu 5 volți. Pentru a face conversia de la 24V la 5V un alt circuit intermediar a fost introdus.

### 3.2.2 Convertorul în comutație Pololu D24C10F5

Pentru ca majoritatea regulatoarelor liniare nu permit o diferența atât de mare între tensiunea de intrare și cea de ieșire, a fost nevoie de introducerea unui alt convertor în comutație ca să furnizeze tensiunea de 5V. Acesta poate să ofere la ieșire pana la 1A pentru alimentarea plăcii cu microcontroler si a giroscopului. Acestea consumă împreună pană la 60mA, ceea ce asigură ca acest convertor Buck nu va fi deloc solicitat. Eficienta de conversie este de 93%.



Figura 3.4 Convertorul în comutație de 5V<sup>[13]</sup>

### 3.2.3 Placa de dezvoltare Arduino Uno

Această placă de dezvoltare este bazată pe microcontrolerul Atmel ATmega328P ce poate opera la maxim 16Mhz. Prezintă 14 intrări/ieșiri digitale, dintre care 6 pot fi folosite ca ieșiri PWM. În plus, are 6 intrări analogice și poate realiza comunicații digitale cu protocoale ca SPI, UART sau I2C. Placa are integrat un chip ce realizează conversia UART-USB si astfel poate comunica cu calculatorul prin intermediul consolei seriale. Acest lucru este foarte important când se doreșt e monitorizarea in timp real a valorilor unor variabile, deoarece placa poate fi programată să trimită in permanență valorile acelor variabile către calculator.



Figura 3.5 Placa de dezvoltare Arduino Uno<sup>[14]</sup>

Deși această placă nu e destinată aplicațiilor profesionale, datorită frecvenței de lucru relativ scăzută si a posibilităților limitate de a configura perifericele, fost aleasă pentru că ceea ce poate oferi satisface necesitățile acestui proiect. Partea dificilă a utilizării algoritmilor de control tratați în lucrare nu este etapa de rulare, ci etapa de implementare care presupune găsirea unor coeficienți corespunzători. Bineînțeles, etapa ce are loc in MATLAB si presupune niște simulări. Placa interoghează fiecare senzor și folosește aceste date pentru a recalcula anumiți parametrii. În funcție de parametrii calculați, această unitate de procesare decide cum să comande motorul. Comanda motorului se face indirect, prin intermediul driverului de motor.

Pinii utilizați sunt evidențiați în Tabelul 3.2.

Numele pinului	Funcția
9	Ieșire PWM
10	Ieșire digitala
11	Ieșire digitala
A1	Intrare analogica
A4	SDA (I2C)
A5	SCL (I2C)

Tabelul 3.2 Pinii folosiți și funcția lor

Pinul 11 este folosit pentru a transmite controlerului curentul cu care trebuie comandat motorul. Acest lucru se face prin intermediul modificării factorului de umplere al semnalului PWM. Gama de valori pe care o poate lua acest curent de comandă este limitată de rezoluția perifericului de PWM, adică 8 biți. Pinul 9 indică direcția de rotație a roții, iar pinul 10 activează sau dezactivează comanda motorului. Cât timp valoarea tensiunii pe acest pin este 0, motorul nu se va mișca.

### 3.2.4 Senzorul de mișcare BNO055

Acest circuit integrat este un senzor de mișcare cu nouă grade de libertate, ce înglobează un giroscop, un magnetometru si un accelerometru. A fost ales datorită performantelor sale foarte bune. Poate furniza date cu o frecventă de maxim 100Hz și datorită procesorului ARM Cortex M0 ce rulează în interiorul chipului, oferă o rejecție foarte bună a zgomotului. Acest procesor trece datele prin anumite filtre digitale înainte de a le furniza microcontrolerului. Așadar, nevoie de a filtra informația în microcontroler dispare.



Figura 3.6 Senzorul de mișcare

Mărimile ce pot fi furnizate sunt:

- Unghiurile Euler
- Cei patru cuaternioni
- Viteza unghiulară pe 3 axe
- Intensitatea câmpului magnetic pe 3 axe
- Accelerația liniară fără accelerația gravitațională
- Accelerația gravitațională
- Temperatura

Cu toate acestea, în rularea algoritmilor se va folosi doar unghiul de pe o anumita axă. Acesta va fi calculat cu ajutorul cuaternionilor.

### 3.2.5 Controlerul de motor Maxon ESCON 36/3 EC

Avantajul acestui controler constă în faptul că poate fi configurat foarte ușor, folosind interfața grafică oferită de producător. De asemenea, se pot alege și modifica funcțiile pinilor, dar și modul de control al motorului (de exemplu control în curent sau in turație). În urma selectării tipului de comandă, controlerul își calculează singur parametrii buclei de reglaj.

Pentru această aplicație a fost preferat un control al motorului în curent. Astfel, accelerația rotii este modificată prin variația curentului de comandă. Utilizatorul setează doar tipul controlerului și parametrii acestuia, iar întreaga proiectare este realizată de aplicația ESCON Studio iar apoi transferată în placa electronica.



Figura 3.7 Rezultatul operației de reglare a driverului de motor

Valorile impuse, după care a fost configurat driverul, sunt regăsiți în Tabelul 3.3.

Tabelul 3.3 Parametrii configurați

Mărime	Valoare
Constanta de viteza	271 rpm/V
Numărul de perechi de poli	8
Viteza de rotație maximă	6300 rpm
Curentul nominal	2.32 A
Curentul maxim	4 A

Figura 3.7 ilustrează rezultatul procesului de auto reglare a driverului, iar graficul arată în ce măsura controlerul proiectat satisface așteptările.



Figura 3.8 Controlerul motrului<sup>[15]</sup>

Driverul prezintă protecții foarte eficiente ce împiedică motorul să se ardă accidental. De asemenea, printre parametrii configurați se află și curentul maxim de funcționare. Așadar, de îndată ce curentul maxim este configurat corect, motorul nu se va arde niciodată.

Intrările și ieșirile ce controlează motorul sunt prezentate în Tabelul 3.4

Tabelul 3.4 Pinii microcontrolerului ce interacționează cu driverul de motor

Intrare/Ieșire	Funcție
Intrarea digitală 1	Curentul de comandă, transmis sub forma de PWM
Intrarea digitală 2	Activare: 1 – activat. 0 – dezactivat
Intrarea digitală 3	Sens: 1 – orar. 0 – anti orar
Ieșire analogică 1	Furnizează o tensiune proporțională cu viteza de rotație

Astfel microcontrolerul îi transmite driverului un semnal PWM cu un anumit factor de umplere, iar acesta interpretează valoarea și deduce curentul aferent acesteia. Factorul de umplere poate varia între 10% si 90%, valori corespunzătoare pentru 0A, respectiv 4A. În concluzie, legea de conversie este:

$$I = 5 \cdot FU - 0.5 \tag{3.1}$$

Unde FU este valoarea factorului de umplere.

Controlerul Maxon oferă și valoarea curentă a vitezei de rotație, oferind o tensiune proporțională cu aceasta, ce este eșantionată și interpretată de microcontroler. Acesta este configurat să furnizeze valori între 0V și 2.5V pentru viteze de rotație cuprinse intre 0rpm, respectiv 6300rpm. Acest lucru este valabil doar pentru sensul orar. Pentru celălalt sens, valoarea absolută a tensiunii furnizată va fi aceeași, dar semnul va fi opus. Aici apare nevoia unei translatări a semnalului analogic, deoarece convertorul analogic digital al microcontrolerului ATmega328P este unul unipolar, deci nu va fi capabil să citească semnale aflate la un potențial mai mic decât masa.



Figura 3.9 Sumatorul cu două rezistențe

Translația de nivel se face folosind un sumator simplu cu doua rezistențe. Din teorema superpoziției rezultă:

$$V_{Arduino\_In} = \frac{1}{2} (5 + V_{ESCON_{AnalogOut}})$$
(3.2)

Unde  $V_{ESCON_{AnalogOut}}$  poate lua valori între -2.5V si 2.5V. Așadar,  $V_{Arduino_In}$  va lua valori între 1.25V și 3.75V, interval cuprins în gama de citire a convertorului.

Pentru testarea funcționalității driverului de motor, au fost generate semnale de comandă din microcontroler, iar valoarea parametrilor interni ai driverului a fost observată cu ajutorul interfeței grafice a programului ESCON Studio, ca în Figura 3.9.



Figura 3.10 Interfața de test a motorului

# CAPITOLUL 4

# ANALIZA PROCESULUI

### 4.1 ANALIZA SISTEMULUI FOLOSIND MODELE TEORETICE

Primul pas în proiectarea regulatorului este determinarea modelului matematic pentru proces. Acesta se va determina prin două metode. Prima implica determinarea funcției de transfer a procesului pe cale teoretică, folosind ecuații ce rezultă din legile mecanicii. A doua metodă presupune identificarea sistemului folosind un set de măsurători înregistrate în timpul funcționarii acestuia. După determinarea acestui model se va putea trece la implementarea propriu zisă a regulatorului.

În continuare va fi tratată metoda teoretică. Ecuațiile sistemului vor fi preluate din [1] și vor fi adaptate la parametrii sistemului tratat în lucrare. Acestea sunt:

$$\ddot{\theta_c} = \frac{(m_c l_c + m_r l)g\sin\theta_c - T_m - C_c\theta_c + C_r\theta_r}{I_b + m_r l^2}$$
(4.1)

.

$$\ddot{\theta_r} = \frac{(I_c + I_r + m_r l^2)(T_m - C_r \dot{\theta_r})}{I_r (I_c + m_r l^2)} - \frac{(m_c l_c + m_r l)g\sin\theta_c - C_c \dot{\theta_c}}{I_c + m_r l^2}$$
(4.2)

$$T_m = K_m \cdot u \tag{4.3}$$

Unde  $m_c$  reprezintă masa cadrului metalic, iar  $m_r$  masa roții.  $C_r$  și  $C_c$  sunt coeficienții de frecare ai roții, respectiv cadrului metalic cu punctul de pivotare.u este intrarea în curent a sistemului.  $T_m$  reprezintă cuplul produs de motor,  $K_m$  constanta de cuplu, iar lungimile l si  $l_r$  sunt evidențiate în Figura 4.1.



Figura 4.1 Identificarea variabilelor din modelul teoretic

Așadar, l reprezintă distanța între punctul de pivotare și centrul de masă, iar  $l_r$  distanța între punctul de pivotare și centrul de masa. Numeric, acești parametrii se regăsesc in tabelul:

l	$9.5 \cdot 10^{-2} m$
l <sub>r</sub>	$0.85 \cdot 10^{-2} m$
m <sub>r</sub>	0.115 kg
m <sub>c</sub>	0.17 kg
I <sub>c</sub>	$10^{-3} kg \cdot m^2$
$I_r$	$0.27 \cdot 10^{-3} kg \cdot m^2$
C <sub>r</sub>	$0.05 \cdot 10^{-3}  kg \cdot m^2 \cdot s^{-1}$
C <sub>c</sub>	$3\cdot 10^{-3} kg \cdot m^2 \cdot s^{-1}$
K <sub>m</sub>	$33.5 \cdot 10^{-3} Nm \cdot A^{-1}$

**Tabelul 4.1 Dimensiunile pendulului** 

În modelul descris mai sus, se observă că există o neliniaritate introdusa de acel sinus. Prin urmare, principiile teoriei sistemelor liniare nu vor putea fi aplicate în acest caz. Așadar, analiza sistemului se va restrânge pentru o gata redusa de unghiuri. Conform aproximației unghiurilor mici,  $\sin \theta \cong \theta$ , pentru  $\theta \in (-15^\circ, 15^\circ)$ . Calculele și analizele ce vor urma adoptă aceasta aproximație. În continuare se va trece la modelul de stare, scriind matricile sistemului, în conformitate cu [1]:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{(m_c l_c + m_r l)g}{(l_c + m_r l^2)} & -\frac{C_c}{(l_c + m_r l^2)} & \frac{C_r}{(l_c + m_r l^2)} \\ -\frac{(m_c l_c + m_r l)g}{(l_c + m_r l^2)} & \frac{C_c}{(l_c + m_r l^2)} & -\frac{C_r (l_c + l_r + m_r l^2)}{l_r (l_c + m_r l^2)} \end{bmatrix}$$
(4.4)

$$B = \begin{bmatrix} 0 \\ -\frac{K_m}{I_c + m_r l^2} \\ \frac{K_m (I_c + I_r + m_r l^2)}{I_r (I_c + m_r l^2)} \end{bmatrix}$$
(4.5)

$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{4.6}$$

$$D = 0 \tag{4.7}$$

Răspunsul la impuls al sistemului este prezentat în Figura 4.2. Așadar, instabilitatea sistemului este evidentă în ambele diagrame din Figura 22. Polul din semiplanul drept provoacă această instabilitate.



Figura 4.2 Analiza stabilității procesului

Codul folosit pentru a obține graficele din Figura 4.2 este:

```
1 = 9.5 * 10^{-2};
1b = 0.85 \times 10^{-2};
mb = 0.17;
mw = 0.115;
Ib = 10^{-3};
Iw = 0.27 \times 10^{-3};
Cw = 0.05 \times 10^{-3};
Cb = 3 \times 10^{-3};
q = 9.81;
Km = 33.5 \times 10^{-3};
A = [0, 1, 0; ...
     g*(mb*lb + mw*l)/(Ib+mw*l^2), -Cb/(Ib+mw*l^2),
     Cw/(Ib+mw*l^2);...
      -g*(mb*lb + mw*l)/(Ib+mw*l^2), Cb/(Ib+mw*l^2),
     -Cw*(Ib+Iw+mw*l^2)/(Iw*(Ib+mw*l^2))];
B = [0; -Km/(lb+mw*l^2);
     Km*(Ib+Iw+mw*l^2)/(Iw*(Ib+mw*l^2))];
C = [1, 0, 0];
D = 0;
SYS = ss(A, B, C, D)
impulse(SYS)
pzplot(SYS)
```

### 4.2 IDENTIFICAREA SISTEMULUI

Această metodă folosește date înregistrate experimental pentru a determina funcția de transfer a sistemului. Deși modelul rezultat înglobează erorile cu o precizie mai mare, deoarece determinarea se face chiar pe modelul practic, acesta va fi unul simplificat în comparație cu cel rezultat din metoda teoretică.



Figura 4.3 Procesul privit ca o cutie neagră

Pentru a obține setul de măsurători se va aplica o treaptă de curent la intrarea sistemului și se va înregistra cum variază ieșirea. Înregistrarea datelor a fost făcută cu ajutorul consolei seriale, ca în Figura 4.4.

După ce fișierul a fost introdus sub formă de vector, s-a folosit instrumentul de identificare a sistemelor din MATLAB. Acestuia i s-a specificat că acel vector reprezintă răspunsul sistemului în domeniul timp la semnal treaptă, iar acesta a calculat funcția de transfer ce caracterizează cel mai bine acel set de date. Numărul de poli și zerouri pentru aceasta funcție de transfer a fost setat de utilizator.

Determinarea funcției de transfer din răspunsul la semnal treaptă folosește principiile sistemelor liniare. Așadar, folosirea acestui set de date pentru a estima funcția de transfer se poate face doar dacă sistemul este liniar.

1.10         0.00         94.02         0.00 <td< th=""><th>COM14</th><th>(Arduino,</th><th>Genuino Uno)</th><th>- c</th><th>□ ×</th></td<>	COM14	(Arduino,	Genuino Uno)	- c	□ ×
Out         First         Part         Part         Part         Part           33         -0.00         84.83         200 - 1.40         -0.00 </th <th></th> <th></th> <th></th> <th></th> <th>Send</th>					Send
	0.00	291.26	avairs 6 - Vu		1 Linderson
Al -0.0 4.42 100 -2.40 Al -0.0 4.42 100 -2.40 Al -0.0 4.42 100 -2.40 Al -0.0 4.42 100 -2.40 Al -0.0 4.43 100 -2.40 Al -0.0 4.40 4.40 100 -2.40 Al -0.0 4.40 4.40 100 -2.40 Al -0.0 4.40 4.40 1	.38 0.00	0 94.92	LQR: -2.68		
48         6.8         9.4.8         208         2.4.8         208         2.4.8           38         0.00         9.4.3         100         -2.4.7         2.4.8	.38 -0.0	00 94.92	LQR: -2.68		
1.3       0.4       1.3       1.0       1.4.       0.4       1.0       1.4.       0.4       1.0       1.0.       1.	.38 0.00	94.92	LQR: -2.68		
Al 0.0 4.3 100 4.3 100 -2.40 100 4.3 100 4.3 100 -2.40 100 4.43 100 -2.40 100 4.44 10	.38 0.00	94.53	LQR: -2.67		
1.31 0.6 9.4.5 100 - 2.40 1.40 0.0 9.4.5 100 - 2.40 1.40 0.0 9.4.5 100 - 2.40 1.40 0.0 9.4.5 100 - 2.46 1.40 0.0 9.4.5 100 - 2.46 1.41 0.0 9.4.5 100 - 2.45 1.41 0.0 9.4.5 100 - 2.15 1.41 0.0 9.4.5 100 - 2.15 1.41 0.0 9.4.5 100 - 2.13 1.41 0.0 9.4.5 100 - 2.14 0.0 9.4.5 100 - 2.14 1.41 0.0 9.4.5 100 - 2.14 0.0 9.4.5 100 -	1.38 0.00	94.53	LQR: -2.67		
3.3       0.0       9.4.9       100       100       9.4.9       100	.38 0.00	94.53	LQR: -2.67		
1.40         0.10         1.43         0.01         1.43         0.01         1.43         0.01         1.43         0.01         1.43         0.01         1.44         0.01         1.43         0.02         1.43         0.02         1.43         0.02         1.43         0.02         1.43         0.02         1.43         0.02         1.43         0.03         1.43         0.04         <	1.38 0.00	94.92	LQR: -2.68		
A.4 0.01 8.42 0.01 8.43 0.08 - 2.46 0.07 8.43 0.01 8.43 0.08 - 2.46 0.08 0.03 9.43 0.00 - 2.40 0.08 9.43 0.00 - 2.40 0.09 9.43 0.00 - 2.40 0.09 9.43 0.00 - 2.40 0.01 9.43 0.00 - 2.43 0.01 9.43 0.00 - 2.44 0.01 9.44 0.00 9.45 0.00 - 2.44 0.01 9.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.00 9.45 0.00 - 2.44 0.	1.40 0.01	1 94.53	LQR: -2.66		
And 0.01 P.122 (200 - 2.46) 1.48 0.01 P.122 (200 - 2.46) 1.48 0.03 P.122 (200 - 2.46) 1.48 0.03 P.122 (200 - 2.46) 1.48 0.03 P.122 (200 - 2.47) 1.47 0.03 P.122 (200 - 2.47) 1.48 0.03 P.122 (200 - 2.47) 1.48 0.03 P.122 (200 - 2.48) 1.48 0.03 P.124 (200 - 2.48) 1.48 0	1.43 0.01	1 94.92	LQR: -2.66		
hile i col vice of vi	1.46 0.01	1 94.92	LQR: -2.65		
1.38         0.19         1.3.1         0.01         -4.1.5           1.0         0.4.3         1.0.2         -4.1.5           1.0         0.4.4         -4.1.5         -4.1.5           1.0         0.4.4         -4.1.5         -4.1.5           1.0         0.4.4         -4.1.5         -4.1.5           1.0         0.4.4         -4.1.5         -4.1.5           1.0         0.4.4         -1.0.5         -4.1.6           1.1.5         0.02         +4.1.5         -4.1.6           1.1.5         0.02         +4.1.5         -4.1.6           1.1.5         0.02         +4.1.5         -4.1.6           1.1.5         0.02         +4.1.5         -4.1.6           1.1.6         0.1.7         +1.1.6         -4.1.6           1.1.6         0.1.7         +1.1.6         -4.1.6           1.1.2         1.1.6         1.1.6         -1.1.6           1.1.2         1.1.6         1.1.6         -1.1.6           1.1.2         1.1.6         1.1.6         -1.1.6           1.1.2         1.1.6         1.1.6         1.1.6           1.1.2         1.1.6         1.1.6         1.1.6	1.51 0.02	2 94.92	LQR: -2.64		
140 0.04 0.05 0.05 0.05 0.05 0.05 0.05 0.	1.58 0.03	3 94.53	LQR: -2.61		
1.10 0.03 0.04 0.05 0.00 0.05 0.05 0.05 0.05 0.05	1.68 0.04	4 94.92	LOR: -2.59		
Li 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1.78 0.04	4 94.92	LUR: -2.5/		
1.13 0.26 14.15 200 - 2.16 2.16 0.27 0.17 0.17 0.17 0.17 0.17 0.17 0.17 0.1	1.87 0.03	3 94.92	LURI -2.55		
1.10         0.13         0.14         0.15         0.14         0.15         0.14         0.15         0.14         0.15         0.14         0.15 <td< td=""><td>1.95 0.03</td><td>3 94.92</td><td>LQR: -2.53</td><td></td><td></td></td<>	1.95 0.03	3 94.92	LQR: -2.53		
1.15 0.23 0.42 0.05 0.42 0.05 0.42 0.05 0.42 0.05 0.42 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.0	2.15 0.00	94.03	Luk: -2.43		
A 10 0.3 0.4 0.2 0.0 - 1.45 5.4 0.0 0.4 0.4 0.2 0.0 - 1.45 5.5 0.4 0.4 0.4 0.2 0.0 - 1.45 5.5 0.4 0.4 0.4 0.2 0.0 - 1.45 5.3 0.4 0.4 0.4 0.2 0.0 - 1.0 5.3 0.4 0.4 0.4 0.2 0.0 - 1.0 5.3 0.4 0.4 0.4 0.0 0.1 0.0 5.4 0.4 0.4 0.4 0.0 0.1 0.0 5.4 0.4 0.4 0.0 0.0 0.0 5.4 0.4 0.4 0.0 0.0 0.0 5.4 0.4 0.4 0.0 0.0 0.0 5.4 0.4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 5.4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 5.4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0	2.62 0.19	9 94.92	LQR1 -2.32		
1.14         0.20         1.40         0.20         1.40         0.20         1.40         0.20         1.40         0.20         1.40         0.20         1.40         0.20         1.40         0.20         1.40         0.20         1.40         0.20         1.40         0.20         1.40         0.20         1.40         0.20         0.20         1.40         1.40         1.40         1.40         1.40         1.40         1.40         1.40         1.40         1.40         1.40         1.40         1.40 <td< td=""><td>4 31 0.29</td><td>0 04 02</td><td>101-1 07</td><td></td><td></td></td<>	4 31 0.29	0 04 02	101-1 07		
1.21 0.40 14.30 2021 - 1.40 1.51 0.52 1.52 1021 - 0.51 1.52 1.54 9.43 2021 0.57 0.50 1.53 2.1 0.9 4.52 1021 0.70 1.54 0.57 9.43 2021 0.57 0.9 2.64 0.57 9.43 2021 0.50 1.0 1.54 0.5 1.54 9.43 2021 0.50 1.0 1.55 0.5 1.54 0.50 1.0 1.55 0.5 1.54 0.50 1.0 1.55 0.5 0.51 1021 0.52 0.0 1.55 0.5 0.51 1021 0.51 0.0 1.55 0.5 0.51 1021 0.51 0.0 1.55 0.5 0.51 1021 0.52 0.0 1.55 0.51 1021 0.50 0.0 1.55	4.31 U.33	3 34.32	Jun: 1.03		
A.19 0.38 44.20 208 - 0.30 J.23 1.68 45.20 208 - 0.30 J.32 7.58 45.20 208 - 0.30 J.32 7.58 45.20 208 - 0.30 J.32 7.58 45.20 208 - 0.34 J.32 7.58 45.20 208 - 0.34 J.33 7.58 45.20 208 - 0.34 J.34 45.20 208 -	7 25 0 69	0 04 02	10R: -1.45		
13.32 1.43 94.32 128 0.70 13.22 1.49 94.22 128 0.70 13.2 1.59 94.22 128 1.50 13.2 1.59 94.22 128 1.50 13.2 7 94.22 128 1.50 13.2 7 94.2 128 1.25 1.50 13.2 7 94.2 128 1.25 1.50 13.2 7 94.2 128 1.25 1.50 10.2 64 7.54 95.2 128 1.25 1.51 102.54 7.54 95.2 128 1.25 1.55 1.55 1.55 1.55 1.55 1.55 1.55	0.60 0.00	04.04	200. 0 30		
19:21 2:19 94:22 128: 2.18 2:64 3:76 94:22 128: 2.08 2:64 3:76 94:22 128: 2.08 2:72 7:89 94:22 128: 12.68 2:72 7:89 94:22 128: 12.68 2:73 7:89 94:22 128: 12.68 2:74 7:80 94:22 128: 12.68 2:74 7:74 7:74 7:74 7:74 7:74 7:74 7:74	19 92 1 4	10 04 07	10P: 0.30		
24.45 5.15 94.32 126:5.60 43.13 5.46 4.43 126:5.60 43.28 7.94 44.92 126:1.4.65 43.28 7.94 94.92 126:1.4.65 100.46 7.54 95.01 126:2.5.4 100.46 7.54 95.01 126:2.5.4 100.105 7.54 95.01 126:2.5.4 101.105 7.54 95.01 126:2.5.4 101.105 7.54 95.01 126:2.5.4 101.105 7.54 95.01 126:1.5.4 101.105 7.54 95.0	10.21 2.2	25 04 03	109-3-28		
10.11 5.46 54.52 108: 54.26 10.21 5.46 54.52 108: 54.26 10.29 7.59 44.29 108: 14.66 14.75 6.16 94.92 108: 12.66 10.54 10.66 7.56 94.92 108: 12.66 10.54 10.66 7.56 94.92 108: 12.68 10.54 10.66 7.56 94.92 108: 12.68 10.54 10.66 7.56 94.92 108: 12.68 10.55 10.66 7.56 94.92 108: 12.68 10.55 10.66 7.56 95.31 108: 12.68 10.55 10.46 -3.59 95.31 108: 12.68 10.55 10.45 -3.59 95.58 10.55 10.45 -3.59 95.31 108: 12.68 10.55 10.45 -3.59 95.55 10.45 -3.59 95.55 10.45 -3.59 95.55 10.45 -3.59 95.55 10.45 -3.59 95.55 10.45 -3.55 95.55 10.45 -3.	20 55 2 3	10 04 01	Age, 2.00		
51.39 7.39 44.32 L02: 14.86 41.37 5.69 44.32 L02: 14.86 103.56 7.56 45.31 L02: 25.26 103.56 7.56 45.31 L02: 25.26 104.53 7.27 94.52 L02: 26.43 105.50 -0.11 94.52 L02: 26.43 105.50 -0.11 94.52 L02: 25.43 105.51 -0.11 94.52 L02: 25.53 105.54 -0.15 95.31 L02: 15.65 105.16 -0.15 95.31 L02: 15.65	12 21 5 8	06 94 92	100-9.28		
14.13         5.16         44.32         5.01         5.01           10.16         7.64         5.01         5.01         5.01           10.16         7.64         5.01         5.01         5.01           10.16         7.64         5.01         5.01         5.01           10.16         7.64         5.01         5.01         5.01           10.16         7.64         5.01         5.01         5.01           10.17         7.10         5.01         5.01         5.01           10.16         7.64         5.01         5.01         5.01           11.46         5.01         5.01         5.01         5.01           11.46         5.01         5.01         5.01         5.01	63 29 2 9	00 04 01	TOP 11 02		
103.64 7.54 95.31 024 95.26 103.64 7.54 95.31 024 95.26 103.50 7.54 95.31 024 95.26 103.50 7.54 95.01 95.12 024 95.51 103.50 7.54 95.01 024 95.2 024 95.51 105.54 95.31 024 95.12 024 95.45 105.16 024 95.11 024 105.65 105.16 024 95.11 024 105.65	R4 75 8 5	58 94.92	IOR: 20.54		
111.163 1.27 94.52 (26) 24.63 102.164 - 2.40 94.52 (26) 12.16 102.164 - 2.40 94.52 (26) 12.16 102.164 - 2.40 94.52 (26) 12.06 102.164 - 2.40 94.52 (26) 12.06 10	103.66 7	56 95 1	T/DP- 25-26		
199-10 - 0-41 94-32 509 25.41 102-56 - 2.36 94.52 509 25.43 32-41 -4.06 94-52 509 25.45 32-41 -4.06 94-52 509 55.1 508:10.65 50.46 -3.59 95.1 508:10.65	111.83 3.	.27 94.9	LOB: 26.63		
102.55 - 2.48 94.52 (201: 22.63 92.41 - 4.06 94.52 (201: 23.59 93.465.8 95.31 (201: 23.65	109.80 -0	0.81 94.	12 LOR: 25.51		
22-41 -4.06 99-52 L208 20.99 10.46 -5.06 95.1 L208 18.65 Channeed International Control Internatione Control International Control International Control I	102.56 -2	2.90 94.	12 LOB: 23.63		1
83.46 -3.58 -95.31 L08: 18.65	92.41 -4.	.06 94.9	LOB: 20.99		
Distance la la action y 20000 has	83.46 -3.	.58 95.3	LOR: 18.65		
Autocrall No line and/on V 200000 has					
	Autoscro	al l		No line anding v 7	50000 haud

Figura 4.4 Achiziția setului de date experimentale

Din metoda precedentă, rezultă clar intervalul în care sistemul poate fi aproximat cu unul liniar. În concluzie se poate face estimarea pe setul de date, în intervalul  $(0^\circ, 15^\circ)$ . Limitând setul achiziționat la acest interval, obținem următoarele valori ale unghiului cadrului exprimat în radiani:

Fabelul	4.2	Setul	de	date	achiziționat
---------	-----	-------	----	------	--------------

0 0.02 0.05 0.10 0.16 0.25
----------------------------

Funcția de transfer ce estimează cel mai bine acest set de date, ce are un zero si doi poli este:

$$H(s) = \frac{24}{s^2 - 34.8s} \tag{4.8}$$

Codul ce calculează răspunsul la impuls pentru sistemul găsit este:

```
num = [24];
den = [1, -34.8, 0];
G = tf(num,den)
[y,t]=impulse(G);
plot(t, y)
figure
pzplot(G)
```



Figura 4.5 Răspunsul la impuls al modelului găsit

# CAPITOLUL 5

# PROIECTAREA ȘI TESTAREA ALGORITMILOR

#### 5.1 Algoritmul PID

Pentru a-și calcula ieșirea, acest algoritm folosește valoarea instantanee a unghiului făcut de cadrul metalic cu perpendiculara ridicata din punctul de pivotare. Cu alte cuvinte, acest unghi este singura mărime eșantionata de bucla de reacție.



Giroscopul furnizează cei patru cuaternioni, care mai apoi sunt convertiți în radiani de către microcontroler după formula:

$$theta = \sin^{-1}[2 \cdot (q_0 q_2 - q_3 q_1)] \tag{5.1}$$

Aşadar, valoarea ieşirii este:

$$out = K_p \cdot theta + K_d \cdot thetaPrec + K_i \cdot thetaSum$$
 (5.2)

Unde *thetaPrec* reprezintă diferența a doua eșantioane consecutive raportată la perioada de eșantionare, iar *thetaSum* reprezintă suma tuturor eșantioanelor.

In procesul de implementare al acestui algoritm a fost ignorat coeficientul de frecare al roții. Așadar funcția de transfer este:

$$\ddot{\theta_c} = \frac{(m_c l_c + m_r l)g\theta_c - u \cdot K_m - C_c \dot{\theta_c}}{I_b + m_r l^2}$$
(5.3)

Aplicând transformata Laplace se obține:

$$\theta_c s^2 = \frac{(m_c l_c + m_r l)g\theta_c - U \cdot K_m - C_c \theta_c s}{I_b + m_r l^2}$$
(5.4)

Deci:

$$(I_b + m_r l^2)\theta_c s^2 = (m_c l_c + m_r l)g\theta_c - U \cdot K_m - C_c \theta_c s$$
(5.5)

Rezultă funcția de transfer:

$$H(s) = \frac{\theta_c(s)}{U(s)} = \frac{K_m}{-(I_b + m_r l^2)s^2 - C_c s + (m_c l_c + m_r l)g}$$
(5.6)

Pentru simularea sistemului cu reacție și determinarea constantelor a fost folosit utilitarul "pidtool" din MATLAB, ce primește ca parametru funcția de transfer a procesului. În urma calibrării constantelor, rezultă următorul răspuns la semnal treaptă.



Figura 5.2 Răspunsul la semnal treaptă al sistemului cu reacție



Figura 5.3 Diaglama poli - zerouri

Iar constantele sunt găsite în Tabelul 5.1

**Tabelul 5.1 Constantele algoritmului PID** 

Kp	-34
K <sub>i</sub>	-61
K <sub>d</sub>	-5

Programul folosit este:

```
1=9.5 \times 10^{-2};
lb=0.85 *10^-2;
mb = 0.17;
mw = 0.115;
Ib=10^-3;
Iw=0.27*10^-3;
Cw=0.05*10^{-3};
Cb=3*10^-3;
q = 9.81;
Km = 33.5 * 10^{-3};
g=9.81;
%P=Km/(-(Ib+mw*l^2)*s^2 - Cb*s + (mb*lb + mw*l)*g)
num=[0, 0, Km]
den= [-(Ib+mw*l^2), -Cb, (mb*lb + mw*l)*g]
G = tf(num, den)
pidtool(G)
```

Pentru vizualizarea polilor și zerourilor a fost adaugată următoarea secvență de cod:

```
P = pid(-34,-61,-5)
C=tf(P)
F=feedback(G,C)
pzplot(F)
figure
impulse(F)
```

În cazul acestui sistem, instabilitatea poate fi produsă nu doar de unghiul cadrului metalic, dar și de viteza unghiulara a roții. Chiar dacă unghiul este menținut la valoarea corespunzătoare, viteza de rotație a roții poate atinge viteze mai mari decât limita superioară, și poate satura ieșirea motorului la o valoare constantă, unde nu va mai exista accelerație.

### 5.2 ALGORITMUL LQR

Acest algoritm de control este unul care se bazează pe modelul de stare. Spre deosebire de implementarea clasica a PID-ului, acesta poate opera cu multiple variabile de reacție, ce se regăsesc printre variabilele de stare [5]. Acest lucru înseamnă ca există mai multă informație despre sistem la un anumit moment de timp, ceea ce rezultă într-un algoritm de control mai robust. Nevoia mai multor mărimi de reacție este evidențiată de cazul precedent, unde din cauza lipsei regulatorului pentru viteza unghiulara a roții, sistemul intră cu ușurință în instabilitate [4].



Figura 5.4 Etapele execuției algoritmului LQR

Variabilele de stare folosite sunt:

- Unghiul de înclinație al cadrului metalic:  $\theta_c$
- Viteza unghiulară a cadrului metalic:  $\dot{\theta_c}$
- Viteza unghiulară a roții:  $\dot{\theta_r}$

Etapele proiectării regulatorului LQR în MATLAB:

- Se folosesc matricile de stare pentru a se construi un obiect de tip sistem
- Se introduc matricile Q,R si N în funcție de specificații
- Se obține matricea de control F

Matricea R, de dimensiune  $m \times m$  se numește matrice de cost. Aceasta ponderează comanda, și oferă proiectantului posibilitatea de a optimiza consumul de energie în execuția comenzii, devenind utilă în situația existenței anumitor restricții de tip protectiv asupra elementelor de execuție ce sunt sensibile la salturi de comandă sau variații bruște. Pentru sistemele SISO sau SIMO, R este egal cu 1, deci comanda este un scalar [3].

Matricea N este de fapt un cost ce ponderează inter influențele dintre stare și comanda. În aceasta aplicație, se va considera nulă.

Matricea F este chiar soluția ecuației Ricatti. În cazul de fată, aceasta este de forma:

$$F = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}$$
(5.7)

Din aceasta se obține valoarea de control astfel:

$$Out_{lqr} = F \cdot X = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \cdot \begin{bmatrix} \theta_c \\ \dot{\theta_c} \\ \dot{\theta_r} \end{bmatrix}$$
(5.8)

In urma execuției programului au fost obținute următoarele constante LQR:

Tabelul 5.2 Constantele algoritmului LQR

$k_1$	-144.29
<i>k</i> <sub>2</sub>	-18.42
k <sub>3</sub>	-1

#### Răspunsul la impuls al sistemului cu reacție este evidențiat în Figura 5.5.



Figura 5.5 Răspunsul la impuls al algoritmului LQR

#### Diagrama poli – zerouri este:



Codul pentru generarea celor trei constante este:

```
1=9.5 * 10^-2;
lb=0.85 *10^-2;
mb = 0.17;
mw=0.115;
Ib=10^-3;
Iw=0.27*10^-3;
Cw=0.05*10^-3;
Cb=3*10^-3;
g = 9.81;
Km = 33.5 \times 10^{-3};
A = [0, 1, 0; ...
    g*(mb*lb
                  +
                         mw*l)/(Ib+mw*l^2), -Cb/(Ib+mw*l^2),
Cw/(Ib+mw*l^2);...
    -q*(mb*lb + mw*l)/(Ib+mw*l^2), Cb/(Ib+mw*l^2),
Cw*(Ib+Iw+mw*l^2)/(Iw*(Ib+mw*l^2))];
B = [0; -Km/(Ib+mw*l^2); Km*(Ib+Iw+mw*l^2)/(Iw*(Ib+mw*l^2))];
C = [1, 0, 0];
D = 0;
SYS = ss(A, B, C, D);
q11 = 10;
q22 = 10;
q33 = 1;
Q = [q11, 0, 0; 0, q22, 0; 0, 0, q33];
R = 1;
N = 0;
F = lqr(SYS, Q, R, N);
```

#### 5.3 PROIECTAREA UNUI ALGORITM ADAPTIV

Principalul punct slab al primei metode este lipsa informației asupra evoluției în timp a vitezei roții. De aceea, principiul acelei metode a fost extins în scopul realizării unui algoritm mai complex. Daca în cazurile precedente sistemul era unul controlat în curent și a cărui ieșire era unghiul de inclinație, acum s-a produs o divizare a acestuia în două subsisteme mai mici. Acest lucru duce la o mai bună observabilitate a proceselor interne.

Ansamblul din Figura este compus din doua regulatoare de tip PD:

- Primul regulator controlează accelerația unghiulara a roții, folosind ca referință ieșirea celuilalt regulator
- Al doilea regulator folosește unghiul curent al cadrului ca si referință si calculează accelerația necesară reechilibrării acesteia.

Logica acestui algoritm se bazează pe faptul că unghiul de înclinație nu este controlat direct de curentul de comandă, ci de accelerația rotii. Al doilea regulator decide cât este accelerația necesară, iar primul se ocupă de furnizarea acelei accelerații.



Figura 5.6 Schema bloc a algoritmului adaptiv

Deoarece acest algoritm folosește două procese, înseamnă ca e necesară cunoașterea a două modele matematice. În mod normal, acest lucru s-ar face prin separarea ecuațiilor (), extrăgând informația necesară realizării modelului pentru fiecare proces. Deoarece acest lucru este destul de complicat și pentru că cele doua procese prezintă o complexitate mai scăzută, se va încerca o calibrare a algoritmilor fără a se cunoaște modelele proceselor.

În schema din Figura 5.6 este evidențiata eșantionarea accelerației unghiulare a rotii. Sistemul nu este echipat cu senzori care să ofere direct aceasta mărime, însă s-a folosit senzorul Hall al motorului pentru a obține viteza unghiulara. Folosind două valori consecutive ale acestei viteze se calculează accelerația.

$$PD1_{out} = K_{P1} \cdot \varepsilon_R(n) + K_{D1} \cdot (\varepsilon_R(n) - \varepsilon_R(n-1))$$
(5.9)

$$PD2_{out} = K_{P2} \cdot \varepsilon_u(n) + K_{D2} \cdot (\varepsilon_u(n) - \varepsilon_u(n-1))$$
(5.10)

Etapele proiectării algoritmului sunt:

- Se începe cu bucla interioara, si se creste constanta de proporționalitate a lui PD<sub>1</sub> pană când ieșirea acestuia începe să conveargă la valoarea de referință
- Se ajustează constanta termenului derivativ, începând de la o valoare foarte mica. Aceasta este de obicei mai mică decât cea de proporționalitate și ajută sistemul să conveargă mai rapid și mai puțin brutal spre valoarea de referință
- Se reiau pașii de mai sus pentru celălalt regulator

Algoritmul este adaptiv, deoarece accelerația de referință se modifică la fiecare iterație.

Constantele obținute sunt găsite în Tabelul 5.3.

K <sub>P1</sub>	-14
K <sub>D1</sub>	-2
K <sub>P2</sub>	-60
K <sub>D2</sub>	-12

Tabelul 5.3 Constantele algoritmului PD - PD

# CAPITOLUL 6

# CONCLUZII

Motivul alegerii acestei teme ca proiect de diplomă s-a datorat pasiunii mele pentru roboți, iar controlul electronic al sistemelor mecanice este reprezentativ pentru robotică. În plus, controlul sistemelor este un domeniu cu aplicații nenumărate, de la sateliți sau rachete pană la automobile sau alte sisteme mecanice. De altfel, principiile discutate în această lucrare pot fi generalizate, acestea fiind valabile nu doar pentru dispozitive mecanice, ci pentru orice alt tip de sistem ce se bazează pe feedback.

Lucrarea prezintă demersul normal pentru abordarea unei probleme de control și modul în care această problemă este structurată pentru a fi rezolvată. În plus, sunt enumerate și criteriile de evaluare a soluțiilor ce rezolvă acea problemă. Aceste criterii sunt foarte importante, deoarece, atunci când vine vorba de o problemă legată de control, nu există o soluție perfectă care să o rezolve. De exemplu un anumit algoritm poate oferi o stabilitate mai bună la perturbații, însă doar într-un domeniu restrâns al mărimii de intrare, în timp ce alt algoritm poate opera într-o gamă mai largă, dar va oferi o stabilitate cu performanțe reduse. Algoritmul ideal este cel care oferă cea mai bună performanță în raport cu condițiile de funcționare stabilite.

Etapele urmate în această lucrare sunt:

- Proiectarea sistemului mecanic si asamblarea sa
- Determinarea modelului matematic ce caracterizează procesul pe cale teoretică
- Identificarea sistemului folosind date experimentale
- Stabilirea modelului ce va fi folosit, alegând între cel determinat teoretic și cel determinat practic în acest caz, a fost ales cel teoretic
- Proiectarea și testarea algoritmilor de control
- Observarea performanțelor

În urma testării celor trei algoritmi, s-au constatat diferențe notabile de performanță.

- 1. Algoritmul PID este singurul dintre cei trei ce folosește o singură mărime de feedback, adică unghiul de înclinație al cadrului. Acesta menține ansamblul în stabilitate, dacă nu există factori perturbați. În momentul apariției unei perturbații mecanice, sistemul reușește să revină în echilibru, însă viteza roții nu scade, ci rămâne la o valoare constantă. Acest lucru nu provoacă mișcare, însă pe măsură pe sunt aplicate perturbații, acesta continuă să crească până în punctul în care cadrul începe să vibreze si intră în instabilitate.
- 2. Algoritmul LQR ține sub control și viteza roții, prin includerea acesteia între variabilele de stare. Din acest motiv sistemul se menține în echilibru și în urma aplicării unei perturbații. În schimb, daca inițial unghiul la care ajunge sistemul este mai mare de 15°, atunci el nu mai converge către poziția de echilibru, ieșind din zona de sistem liniar. Din punct de vedere at structurii formulei de calcul, acesta este de fapt un algoritm de tip PD, ce are un regulator proporțional pentru viteza unghiulară.
- 3. Algoritmul format din cele doua regulatoare PD are performanțe similare cu LQR, cele două folosind aceleași mărimi de intrare. Singura diferență este că, în acest caz, viteza unghiulară este transformată în accelerație unghiulară. Dezavantajul acestei metode este timpul relativ lung necesar pentru calibrarea algoritmului, necunoscându-se nimic despre cele două sisteme.

Au mai existat și alte limitări în reglarea sistemului. Cea mai mare ar fi pivotul în jurul căruia se învârte cadrul metalic. Acesta are frecări ce influențează modul de funcționare. Efectul acestora este cel mai sesizabil pentru valori mici ale unghiului de înclinație, unde și turația motorului este mică și nu reușește să învingă forța de frecare. Cea mai bună rezolvare este un rulment, cu care s-ar obține rezultate mai bune și mai apropiate de cele anticipate.

Un alt algoritm de control pe care mi-am propus să îl dezvolt în contextul acestui proiect este MPC, ce se bazează pe predicție în momentul în care actualizează valoarea ieșirii. În viitor voi studia și implementa acest algoritm.

### BIBLIOGRAFIE

- [1] M. Gajamohan, M. Merz, I. Thommen, și R. D'Andrea, "The Cubli: A cube that can jump up and balance," în Conferința Internațională a Roboților și Sistemelor Inteligente (IROS), 2012, pp. 3722–3727.
- [2] A. Beznos, A. Grishin, A. Lenskij, D. Okhotsimskij, şi A. Formal'skij, "A flywheel use-based control for a pendulum with a fixed suspension point", no. 1, pp. 27–38, 2004.
- [3] Monica Pătraşcu, Cursul de "Sisteme de Conducere în Robotică", Facultatea de Inginerie Mecanică și Mecatronică.
- [4] P. Reist și R. Tedrake, "Simulation-based lqr-trees with input and state constraints," în Robotică și Automatizare (ICRA), Mayi 2010, pp. 5504 –5510.
- [5] B. Andrievsky, "Global stabilization of the unstable reactionwheel pendulum," Automatizare şi Control la Distanţă, vol. 72, pp. 1981–1993, 2011. http://dx.doi.org/10.1134/ S0005117911090189
- [6] D. Alonso, E. Paolini, şi J. Moiola, "Controlling an inverted pendulum with bounded controls," în Dinamică, Bifurcări şi Control, Berlin / Heidelberg, 2002, vol. 273, pp. 3–16.
- [7] A. Stephenson, "On a new type of dynamical stability," Indexare: "Manchester Literary and Philosophical Society", vol. 52, pp. pp. 1–10, 1908
- [8] Cursul "Introduction to Control System Design a First Look", MITx 6.302.0x
- [9] Cursul "Introduction to Control System Design Computational State Space Approaches", MITx 6.302.1x
- [10] http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=ControlPID
- [11] https://en.wikipedia.org/wiki/Inverted\_pendulum
- [12] http://www.pulsar.pl/EN\_1300\_LRS-150-24\_lrs-24v,150w,6,5a-enclosed-power-supply-unit
- [13] ww.optimusdigital.ro/surse-coboratoare-de-5-v/2735-sursa-coboratoare-pololu-5-v-500-ma-d24v5f5.html
- [14] http://marcusjenkins.com/wp-content/uploads/2014/06/ARDUINO\_V2.png
- [15] http://circuitcellar.com/wp-content/uploads/2013/03/Maxon-300x229.jpg

# Anexa 1

### **Algoritmul LQR:**

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BNO055.h>
#include <utility/imumaths.h>
#include <math.h>
#define BNO055_SAMPLERATE_DELAY_MS
(100)
#include "TimerOne.h"
Adafruit BNO055 bno = Adafruit BNO055();
double q0,q1,q2,q3;
double r1, r2, r3;
double beta, theta, thetaD;
double K1,K2,K3,K4;
int i=0:
double beta prev, v prev=0;
int t=0;
int start=0:
double rpm;
double lgr;
double v=0;
int state = 0;
void setup() {
//Constantele
 K1=6.5;
 K2=4.5;
 K3=0.8;
 K4=1;
 pinMode(9,OUTPUT);
 pinMode(10,OUTPUT);
 pinMode(11,OUTPUT);
 pinMode(13,OUTPUT);
 setPwmFrequency(11, 64);
 Serial.begin(250000);
 /*Initializare senzor de miscare*/
 if(!bno.begin())
  Serial.print("Nu poate comunica cu BNO055");
  while(1);
 delay(1000);
 int8_t temp = bno.getTemp();
 bno.setExtCrystalUse(true);
 Serial.println("Calibrat");
 /*Initializare timer*/
 Timer1.initialize(20000);
 Timer1.attachInterrupt(callback); ]
 digitalWrite(9,HIGH);
}
void callback()
 i=1;
```

}

```
void loop() {
 char c;
 if(Serial.available())
  {
   c=Serial.read();
   if(c=='s')
   {
Serial.println("start");
   digitalWrite(10,HIGH);
}
   if(c=='o')
   digitalWrite(10,LOW);
   start=0;
   }
  }
 c='g';
 if(i==1) //au trecut inca 20ms
 {
  i=0:
  imu::Quaternion quat = bno.getQuat();
  q0=quat.x();
  q1=quat.y();
  q2=quat.z();
  q3=quat.w();
  r2=2*(q0*q2-q3*q1);
  beta=asin(r2); //calculeaza unghi in radiani
  rpm = analogRead(A1); //obtinere viteza roata
  rpm = (rpm-512)*2;
  v = (double) rpm * (100.0/512);
  theta = beta * (100.0/0.8); //scalare unghi
  thetaD = (beta - beta_prev)/0.02; //calcul viteza
unghiulara
  thetaD = thetaD * (100.0/100); //scalare viteza
unghiulara
  lgr = K4*(K1*theta + K2*thetaD - K3*v); //calcul
iesire
 lqr = lqr^{*}(4.0/(100.0));
  Serial.print(theta); Serial.print(" ");
Serial.print(thetaD); Serial.print(" ");
Serial.print(v);Serial.print(" LQR: ");
Serial.println(lqr);
 beta_prev = beta;
  v_prev=v;
 currentToPWM(lqr);
 if(lqr > 0)
  {
  digitalWrite(9,HIGH);
  }
 else
  digitalWrite(9,LOW);
  }
 }
}
```

void currentToPWM(double c) { c=abs(c); if(c>4) c=4;double pwm=c \*51 + 25; if(pwm<255) analogWrite(11, pwm); else analogWrite(11, 255); void setPwmFrequency(int pin, int divisor) { byte mode; if(pin == 5 || pin == 6 || pin == 9 || pin == 10)switch(divisor) { case 1: mode = 0x01; break; case 8: mode = 0x02; break; case 64: mode = 0x03; break; case 256: mode = 0x04; break; case 1024: mode = 0x05; break; default: return: if(pin == 5 || pin == 6)TCCR0B = TCCR0B & 0b11111000 | mode; } else { TCCR1B = TCCR1B & 0b11111000 | mode; } else if(pin == 3 || pin == 11)switch(divisor) { case 1: mode = 0x01: break: case 8: mode = 0x02; break; case 32: mode = 0x03; break; case 64: mode = 0x04; break; case 128: mode = 0x05; break; case 256: mode = 0x06; break; case 1024: mode = 0x7; break; default: return; TCCR2B = TCCR2B & 0b11111000 | mode; }

#### Algoritmul LQR:

int state = 0;

#include <Wire.h> #include <Adafruit\_Sensor.h> #include <Adafruit\_BNO055.h> #include <utility/imumaths.h> #include <math.h> #define BNO055\_SAMPLERATE\_DELAY\_MS (100)#include "TimerOne.h" Adafruit\_BNO055 bno = Adafruit\_BNO055(); double q0,q1,q2,q3; double r1, r2, r3; double beta, theta, thetaD; double Kp,Ki,Kd; int i=0; double beta\_prev; int t=0: int start=0; double rpm; double pid; double sum=0;

void setup() { //Constantele Kp=7; Ki=2; Kd=0.4: pinMode(9,OUTPUT); pinMode(10,OUTPUT); pinMode(11,OUTPUT); pinMode(13,OUTPUT); setPwmFrequency(11, 64); Serial.begin(250000); /\*Initializare senzor de miscare\*/ if(!bno.begin()) ł Serial.print("Nu poate comunica cu BNO055"); while(1); delay(1000); int8\_t temp = bno.getTemp(); bno.setExtCrystalUse(true); Serial.println("Calibrat"); /\*Initializare timer\*/ Timer1.initialize(20000); Timer1.attachInterrupt(callback); digitalWrite(9,HIGH); } void callback() { i=1; } void loop() { char c; if(Serial.available()) ł c=Serial.read(); if(c=='s')Serial.println("start"); digitalWrite(10,HIGH); if(c=='o')digitalWrite(10,LOW); start=0; } } c='g'; if(i==1) //au trecut inca 20ms i=0; imu::Ouaternion quat = bno.getOuat(); q0=quat.x();q1=quat.y(); q2=quat.z(); q3=quat.w(); r2=2\*(q0\*q2-q3\*q1); beta=asin(r2); //calculeaza unghi in radiani

```
theta = beta * (100.0/0.8); //scalare unghi
  thetaD = (beta - beta_prev)/0.02; //calcul viteza
unghiulara
  thetaD = thetaD * (100.0/100); //scalare viteza
unghiulara
  sum+=theta:
  pid = Kp*theta + Kd*thetaD + Ki*sum; //calcul
iesire
  pid = pid*(4.0/(100.0));
  beta_prev = beta;
  currentToPWM(pid);
  if(pid > 0)
  digitalWrite(9,HIGH);
  ł
  else
  digitalWrite(9,LOW);
  }
 }
}
void currentToPWM(double c)
{
 c=abs(c);
 if(c>4) c=4;
 double pwm=c * 51 + 25;
if(pwm<255) analogWrite(11, pwm);
 else analogWrite(11, 255);
}
void setPwmFrequency(int pin, int divisor) {
 byte mode;
 if(pin == 5 || pin == 6 || pin == 9 || pin == 10) 
  switch(divisor) {
   case 1: mode = 0x01: break:
   case 8: mode = 0x02; break;
   case 64: mode = 0x03; break;
   case 256: mode = 0x04; break;
   case 1024: mode = 0x05; break;
   default: return;
  if(pin == 5 || pin == 6) 
   TCCR0B = TCCR0B & 0b11111000 | mode;
  } else {
   TCCR1B = TCCR1B & 0b11111000 | mode;
 else if(pin == 3 || pin == 11) 
  switch(divisor) {
   case 1: mode = 0x01; break;
   case 8: mode = 0x02; break;
   case 32: mode = 0x03; break;
   case 64: mode = 0x04; break;
   case 128: mode = 0x05; break:
   case 256: mode = 0x06; break;
   case 1024: mode = 0x7; break;
   default: return:
  TCCR2B = TCCR2B & 0b11111000 | mode;
 }
```

}

#### Algoritmul PD – PD:

#include <Wire.h> #include <Adafruit Sensor.h> #include <Adafruit BNO055.h> #include <utility/imumaths.h> #include <math.h> #define BNO055\_SAMPLERATE\_DELAY\_MS (100)#include "TimerOne.h" Adafruit\_BNO055 bno = Adafruit\_BNO055(); double q0,q1,q2,q3; double r1, r2, r3; double beta, theta, thetaD; double Kp1,Kd1,Kp2, Kd2; int i=0: double beta prev; int t=0; int start=0; double rpm; double pdpd; double pd1, pd2; double sum=0; int state = 0; double v=0; //viteza unghiulara double v\_prev; double a; //acceleratia unghiulara double a\_prev; void setup() { //Constantele Kp1=8.6; Kd1=1; Kp2=5.4; Kd2=0.1; pinMode(9,OUTPUT); pinMode(10,OUTPUT); pinMode(11,OUTPUT); pinMode(13,OUTPUT); setPwmFrequency(11, 64); Serial.begin(250000); /\*Initializare senzor de miscare\*/ if(!bno.begin()) Serial.print("Nu poate comunica cu BNO055"); while(1); delay(1000); int8\_t temp = bno.getTemp(); bno.setExtCrystalUse(true); Serial.println("Calibrat"); /\*Initializare timer\*/ Timer1.initialize(20000); Timer1.attachInterrupt(callback): digitalWrite(9,HIGH); } void callback() ł i=1;

}

void loop() { char c; if(Serial.available()) ł c=Serial.read(); if(c=='s') Serial.println("start"); digitalWrite(10,HIGH); if(c=='o')digitalWrite(10,LOW); start=0; } } c='g'; if(i==1) //au trecut inca 20ms { i=0; imu::Quaternion quat = bno.getQuat(); q0=quat.x();q1=quat.y(); q2=quat.z(); q3=quat.w(); r2=2\*(q0\*q2-q3\*q1); beta=asin(r2); //calculeaza unghi in radiani //obtinere viteza roata rpm = analogRead(A1);rpm = (rpm-512)\*2; v = (double) rpm \* (100.0/512); //scalaretheta = beta \* (100.0/0.8); //scalare unghi thetaD = (beta - beta\_prev)/0.02; //calcul viteza unghiulara thetaD = thetaD \* (100.0/100); //scalare viteza unghiulara sum+=theta;  $a = (v-v_prev)/0.02;$ pd2 = Kp2\*theta + Kd2\*thetaD; $pd1 = Kp1*(pd1-a) + Kd1*(pd1 - a - a_prev)/0.02;$ pdpd = pd1\*(4.0/(100.0));beta\_prev = beta;  $v_prev = v;$  $a_prev = a - pd1;$ currentToPWM(pdpd); if(pdpd > 0)digitalWrite(9,HIGH); } else ł digitalWrite(9,LOW); }

} } void currentToPWM(double c) { c=abs(c); if(c>4) c=4;double pwm=c \* 51 + 25; if(pwm<255) analogWrite(11, pwm); else analogWrite(11, 255); } void setPwmFrequency(int pin, int divisor) { byte mode; if(pin == 5 || pin == 6 || pin == 9 || pin == 10)switch(divisor) { case 1: mode = 0x01; break; case 8: mode = 0x02: break: case 64: mode = 0x03; break; case 256: mode = 0x04; break; case 1024: mode = 0x05; break; default: return; if(pin == 5 || pin == 6)TCCR0B = TCCR0B & 0b11111000 | mode; } else { TCCR1B = TCCR1B & 0b11111000 | mode; } else if(pin == 3 || pin == 11)switch(divisor) { case 1: mode = 0x01; break; case 8: mode = 0x02; break; case 32: mode = 0x03; break; case 64: mode = 0x04; break; case 128: mode = 0x05; break; case 256: mode = 0x06; break; case 1024: mode = 0x7; break; default: return: ł TCCR2B = TCCR2B & 0b11111000 | mode; }

}

# Anexa 2

#### Foaia de catalog a motorului:

