



FACULTAT
DE CIÈNCIES
I TECNOLOGIA

UVIC | UVIC·UCC

UNIVERSITAT DE VIC

TREBALL FINAL DE GRAU

GRAU EN ENGINYERIA MECATRÒNICA

Ball on Plate

Autor:

Òscar Allué Orriols

Tutors:

Dr. Moisès Serra-Serra

Dr. Pere Martí Puig

6 de juny de 2020

Agraïments

Dono gràcies als meus tutors per tot el suport que m'ha donat durant la realització d'aquest treball. Tot i haver estat una realització tan atípica a causa de la pandèmia, vull donar les gràcies pel grandíssim esforç que han fet per ajudar-me i guiar-me en tots els dubtes i dificultats que he tingut per realitzar aquest projecte. Moltes gràcies Moisès Serra i Pere Martí.

També vull donar les gràcies a tota la meva família i amics, que han estat al meu costat durant tota la meva estança a la Universitat, m'heu motivat i donat força per continuar. També vull agrair especialment a l'Alba Medina que m'ha aguantat tant a les bones, com a les no tan bones, ha estat un pilar fonamental, junt amb la família, per aconseguir realitzar els meus objectius.

Per últim, vull donar les gràcies a en Marc Masnou i a en Patrick Marchante, per tot el suport i ajudes que m'ha donat durant tota la carrera, ningú sap la quantitat d'hores que hem invertit als matins a la Universitat. En especial en aquest treball, a en Marc, per haver-me imprès les peces que necessitava per muntar aquest projecte.

Resum

En el present treball es descriu els passos per implementar un sistema de control LQR en un Arduino DUE, que controlarà un sistema de *Ball on Plate* (Bola sobre Plataforma).

La planta en la qual s'implementa el sistema de control és un sistema de *Ball on Plate*, el qual també ha estat dissenyat i muntat per l'autor. Aquesta planta consta d'una base on hi han muntats dos servomotors, la placa d'Arduino Due encarregada del control, l'electrònica necessària per alimentar correctament els servomotors i la làmina tàctil resistiva. A la base, hi ha muntat una junta Cardan que fa d'unió amb la plataforma pivotant on hi haurà la bola. Aquesta plataforma pivota gràcies a la junta Cardan, que a la vegada evita la rotació al voltant de l'eix perpendicular a la pròpia plataforma. El moviment de la plataforma és controlat per dos servomotors, connectats amb la plataforma a través de barres amb articulacions esfèriques en els seus extrems.

El control es realitza amb una tècnica de control LQR, que s'implementa a una placa d'Arduino DUE. El sensor d'aquest sistema és una làmina resistiva situada a la part superior de la plataforma i que detecta la posició de la bola quan està al damunt. Per altra banda, els actuadors són els dos servomotors. Cada un serà l'encarregat de moure un dels eixos de rotació lliures de la plataforma (X i Y).

Per realitzar el control final a la placa d'Arduino prèviament s'han realitzat una sèrie de simulacions i comprovacions amb el Matlab. Per una banda es comprova si aquest sistema es pot observar i controlar. A continuació, es realitza una petita simulació i s'observa la resposta del sistema i el lloc geomètric de les arrels, per comprovar la seva estabilitat. Es realitza una simulació comprovant que el sistema respon correctament per portar la bola a un punt de la plataforma.

Finalment es realitza la implementació del sistema de control al sistema real, realitzant ajustos per intentar que la resposta del sistema real sigui la més aproximada al sistema simulat amb el Matlab. La implementació es realitza programant la placa d'Arduino Due fent servir el llenguatge de programació C++ i les funcions pròpies d'Arduino.

Abstract

This work describes the steps to implement an LQR control system in an Arduino DUE, to control a Ball on Plate system.

The plant in which the control system is implemented is a Ball on Plate system, which has also been designed and assembled by the author. This plant consists of a base where two servomotors have been assembled, the Arduino Due board is responsible of the control, the electronics needed to properly power the servomotors and the resistive touch screen, and finally, there are a Cardan joint mounted above the base, this Cardan joins the pivoting platform where the ball will be. This platform pivots thanks to the Cardan joint, which in turn prevents rotation around the axis perpendicular to the platform itself. The movement of the platform is controlled through to the connection between it and the servomotors, connected both pieces by means of bars with spherical joints at their ends.

As already mentioned, the control is performed with an LQR control technique, which is implemented on an Arduino DUE board, the sensor of this system is a resistive touchscreen located at the top of the platform that detects the position of the ball on it. On the other hand, the actuators are the two servomotors. Each will be responsible for moving one of the free axes of rotation of the platform (X and Y). To carry out the final control on the Arduino board, a series of simulations and checks have previously been carried out with the Matlab. On the one hand it is checked if this system can be observed and controlled, then a simulation is performed and the response of the system and the geometric location of the roots is observed, to check its stability. Finally, a simulation checking that the system responds correctly in both cases, when the ball is going to the point on the platform and when the ball describes a trajectory.

Finally the implementation of the control system to the real system is realized, realizing adjustments to try that the answer of the real system is nearest the simulated system with the Matlab. The implementation is done by programming the Arduino Due board using the C++ programming language.

Índex

1	Introducció	10
1.1	Objectius	11
1.2	Estructura de la memòria	11
1.3	Restriccions	11
2	Estat de l'art	13
3	Modelització matemàtica	14
3.1	Equació de Langrange	15
3.2	Mètode Euler-Lagrange	16
3.3	Simulació	18
3.3.1	Matlab	19
3.3.1.1	Corba Posició-Temps	19
3.3.1.2	Estat de sortida en funció del temps	20
3.3.1.3	Representació del Lloc Geometric de les Arrels	21
3.3.2	Simulink	22
3.3.2.1	Simulació model teòric	23
3.3.2.2	Simulació del model real	24
4	Disseny mecatrònic del sistema	27
4.1	Arduino DUE	27
4.2	Làmina tàctil resistiva	28
4.3	Servomotors (MG996R)	29
4.4	Teclat Matricial	31
4.5	Part Mecànica	32
4.6	Base Inferior	32
4.7	Base superior	34
4.8	Suport metacrilat	36
4.9	Junta Cardan	38
4.10	Barres de connexió amb els Servomotors.	39
4.11	Placa de metacrilat	40
4.12	Esfera	41
5	Programació del Sistema	43
5.1	Làmina tàctil resistiva	43
5.2	Servomotors	43
5.3	Teclat matricial	43
5.4	Programa del sistema	44

6 Resultats	45
7 Conclusions	48
Bibliografia	49
A Simulink	50
A.1 Resultat de l'aplicació Control Tuner System	50
B Característiques de l'Arduino Due	51
B.1 Característiques de l'Arduino DUE	51
C Servomotor	53
C.1 servomotor	53
C.1.1 Comprovació del parell	55
C.1.2 Connexió amb l'Arduino	57
D Làmina tàctil resistiva	59
D.1 Làmina tàctil resistiva	59
D.1.1 Principi de funcionament	59
E Mida peces mecàniques	63
E.1 Mides de les peces mecàniques	63
E.1.1 Base Inferior	63
E.1.2 Base superior	64
E.1.3 Suport metacrilat	64
E.1.4 Placa de Metacrilat	65
F Codi Matlab	66
G Codi servomotor	71
H Codi làmina tàctil resistiva de 4 fils	72
I Codi del teclat matricial	74
J Codi <i>Ball on Plate</i> complert	76
Glossari	85
Acrònims	87

Índex de figures

Figura 3.1	Paràmetres que intervenen en el nostre sistema.	14
Figura 3.2	Representació de la posició de la bola en funció del temps.	20
Figura 3.3	Representació de l'estat de sortida en funció del temps.	21
Figura 3.4	Representació del Lloc Geometric de les Arrels.	22
Figura 3.5	Model teòric del sistema de control.	23
Figura 3.6	Modelització del sistema a partir del disseny CAD.	25
Figura 3.7	Model del sistema real a partir del disseny en CAD.	25
Figura 3.8	Relació entre la inclinació de la plataforma i l'angle d'inclinació dels Servomotors.	26
Figura 3.9	Visualització del sistema amb el Multibody.	26
Figura 4.1	Placa Arduino DUE	28
Figura 4.2	Làmina tàctil resistiva.	29
Figura 4.3	Servomotor MG996R.	30
Figura 4.4	Comprovació de la força que arriba al servomotor fent el símil amb una biga.	30
Figura 4.5	Connexió dels Servos amb la font d'alimentació i a l'Arduino a través de 2 transistors NPN en cascada.	31
Figura 4.6	Teclat matricial indicat les seves connexions.	31
Figura 4.7	Disseny CAD del sistema complert.	32
Figura 4.8	Base Inferior.	33
Figura 4.9	Unió del Servomotors.	33
Figura 4.10	Resultat Base Inferior 3D.	34
Figura 4.11	Disseny CAD base superior.	34
Figura 4.12	Unió Servo, part superior.	35
Figura 4.13	Pivot d'unió amb la Cardan.	35
Figura 4.14	Base superior impresa en 3D.	36
Figura 4.15	Suport metacrilat.	36
Figura 4.16	Pivot d'unió amb la Cardan.	37
Figura 4.17	Unió de la plataforma amb les barres.	37
Figura 4.18	Suport de metacrilat imprès en 3D.	38
Figura 4.19	Junta Cardan.	39
Figura 4.20	Cardan real.	39
Figura 4.21	Disseny de les barres de connexió entre servomotor i plataforma.	40
Figura 4.22	Barres de connexió reals.	40
Figura 4.23	Disseny CAD de la placa de metacrilat.	41
Figura 4.24	Placa de metacrilat real.	41
Figura 4.25	Disseny CAD de la bola.	42

Figura 6.1	Teclat matricial indicat les seves connexions.	45
Figura 6.2	Representació gràfica del comportament del control.	46
Figura 6.3	Resultat del control amb Simulink	46
Figura A.1	Resultat de l'aplicació Control System Tuner a cada senyal. . .	50
Figura C.1	Parts d'un servomotor.	53
Figura C.2	Diagrama de blocs d'un servomotor.	54
Figura C.3	Control PWM d'un servomotor.	55
Figura C.4	Comprovació de la força que arriba al servomotor fent el símil amb una viga.	56
Figura C.5	Comprovació parell que ha d'aguantar el servomotor.	57
Figura C.6	Connexió dels Servos amb la font d'alimentació i a l'Arduino a través de 2 transistors npn en cascada.	58
Figura D.1	Làmina tàctil resistiva.	60
Figura D.2	Funcionament de la làmina tàtil resistiva.	61
Figura D.3	Diagrama de flux d'obtenció de dades de la posició de l'esfera de la làmina tàctil resistiva.	62
Figura E.1	Mides de la base inferior.	63
Figura E.2	Mides de la base superior.	64
Figura E.3	Mides de la unió de la plataforam.	64
Figura E.4	Mides de la placa de metacrilat.	65

Índex de taules

Taula 3.1 Paràmetres i variables físiques del sistema *Ball on Plate* 18

Capítol 1

Introducció

El sistema "*Ball on Plate*" és un sistema que es fa servir actualment en algunes universitats per a realitzar investigacions sobre els sistemes de control. És un sistema inestable i no lineal, fet que provoca que sigui un sistema interessant per realitzar investigació sobre el control automàtic.

Aquest sistema el compon una plataforma plana amb un pivot central. El sistema d'actuació són 2 servomotors, que actuen en els dos eixos de llibertat i provoquen una inclinació de la plataforma. Una bola és col·locada sobre de la plataforma i es controla la seva posició mitjançant la inclinació que provoquen els dos servomotors sobre la plataforma.

Durant el pas per la Universitat es donen moltes eines per poder dur a terme un sistema Mecatrònic per realitzar una acció concreta. La Universitat ensenya de forma teòrica i pràctica en major o menor profunditat la part mecànica, amb assignatures com són Física Mecànica o Elasticitat i Resistència de Materials. També s'aprèn tot el necessari per a fer la part elèctrica i electrònica amb assignatures com Anàlisi de Circuits, Microcontroladors, etc. En totes aquestes assignatures es veu de forma teòrico-pràctica els continguts, es sap com hauries d'aplicar de manera pràctica els coneixements assolits. En canvi, a les assignatures de Regulació Automàtica o Control Discret es fa tota la part teòrica sobre el control, però la matèria és bastant abstracte. En aquestes assignatures es donen les eines de com realitzar el control automàtic d'un sistema, però no és realitzen tots els passos per fer-ho. Com que durant tota la carrera es diu que la regulació i el control automàtic són de les assignatures més importants de la carrera, es va voler realitzar aquest control sobre un sistema Mecatrònic real.

La motivació principal per a la realització d'aquest projecte és poder realitzar tots els passos per a construir un sistema Mecatrònic que es controli mitjançant un control automàtic, en aquest cas, un LQR (Linear Quadratic Regulator). Aquest tipus de control es caracteritza per la seva robustesa, tant en temps discret com en temps continu. El sistema LQR treballa amb un senyal de referència que ha d'assolir, realitzant comprovacions iteratives del sistema per comprovar la diferència entre l'estat actual i el desitjat. L'autor dona un criteri de penalització a cada variable per optimitzar el sistema de control. Es va escollir el problema de la bola sobre plataforma perquè és un bon exemple visual de com actua aquest tipus de control. Es realitza aquest projecte per comprendre les diferents parts d'un control automàtic, des de la modelització matemàtica, fins a la implementació final en el sistema real.

1.1 Objectius

Els objectius principals d'aquest TFG (Treball de Fi de Grau) són els següents:

- Realitzar el disseny i construcció d'una plataforma amb 2 graus de llibertat per dur a terme el sistema de *Ball on Plate*.
- Realitzar la simulació del sistema de *Ball on Plate* amb l'eina Matlab.
- Realitzar el disseny del control d'aquest sistema mitjançant LQR per a que la bola es quedi a una posició determinada sobre de la plataforma.

1.2 Estructura de la memòria

Aquest document segueix un ordre concret, comença per l'anàlisi teòric i acaba per la part pràctica. Cada capítol engloba una part del projecte:

- El Capítol 2 (2) mostra l'estat de l'art. La recerca es va realitzar abans d'iniciar el projecte per poder veure i comprovar quins treballs i investigacions hi havia sobre els diferents mètodes de control aplicats a aquest tipus de sistemes.
- El capítol 3 (3) explica com s'han obtingut les funcions matemàtiques per dur a terme la modelització del sistema *Ball on Plate* i com es realitza la simulació del control en Matlab.
- El capítol 4 (4) és una descripció del Disseny Mecatronic, de tots els components que formen aquest sistema de *Ball on Plate* i de perquè s'han escollit.
- El capítol 5 (5) és una explicació de com s'ha programat cada component per separat. Al final del capítol hi ha una descripció i un pseudocodi del programa principal que controla aquest sistema.
- En el capítol 6 (6) es mostren tots els resultats obtinguts al realitzar el present treball.
- El capítol 7 (7) descriu les conclusions que s'han extret de la realització del present treball.

1.3 Restriccions

Les restriccions són totes aquelles accions que un sistema no pot assolir per la pròpia naturalesa del sistema, com podria ser el pes màxim que pot suportar, la seva velocitat màxima, angle d'inclinació, etc. Aquestes restriccions provenen de toleràncies de les peces, del disseny del sistema, falta d'una velocitat d'algun actuator, entre d'altres. Les restriccions del sistema són:

- Per una banda, té unes restriccions mecàniques en quant a l'angle d'inclinació que pot tindre la plataforma, que és de 8° d'inclinació respecte a la base.

- Per una altra banda, en tema software, el sistema té una restricció en quant a velocitat del microprocessador, que en el nostre cas, al tindre un Arduino Due com a microprocessador, hi ha una limitació de velocitat del microprocessador de 84MHz.
- En quan a hardware, hi han diverses limitacions:
 - Per una banda, tenim les restriccions dels servomotors. Tenen una velocitat de $0.14\text{s}/60^\circ$, un parell màxim de $1,08\text{Nm}$.
 - Per altra banda, tenim unes restriccions a causa del error provocat per la làmina resistiva, ja que aquesta té un error de no linealitat de 1.5% .

Capítol 2

Estat de l'art

Aquest treball s'efectua seguint el Disseny Mecatronic, la qual cosa significa que tots els termes del disseny del sistema van lligats, és a dir, la part mecànica va lligada amb l'elèctrica i l'electrònica. Per poder realitzar un bon disseny adaptant totes les parts, primer s'ha de tindre una base de coneixement sobre el sistema que es vol realitzar.

Abans de començar a realitzar aquest treball s'ha investigat sobre els diferents tipus de sistemes de *Ball on Plate* que hi ha fins a la data. D'aquesta forma, s'ha pogut veure les diferències entre els diferents sistemes de *Ball on Plate*, i així, aprendre els aspectes tècnics i les dificultats reportades en els diferents treballs.

Fent investigació sobre els treballs que hi ha fets sobre aquest sistema, es pot observar que hi ha moltes formes de realitzar el control, parlant des del punt de vista de sensòrica, actuació o control. En el cas d'aquest treball, s'han pres les següents dedicions sobre el disseny:

- El control del sistema és mitjançant una làmina tàctil resistiva. És un sensor fàcil de programar, amb rapidesa d'adquisició de la posició de la bola sobre la plataforma i no gaire costós.
- La plataforma és una plataforma de 2 DoF moguda mitjançant dos servomotors digitals (MG996R). Aquests motors són els encarregats de moure la plataforma, que pivota sobre una junta Cardan. Un servomotor és el responsable de la rotació de la plataforma en l'eix X, i l'altre en l'eix Y. La junta Cardan a la cegada impedeix la rotació de la plataforma en l'eix perpendicular.
- Per altra banda, es fa servir un Arduino Due per al control del sistema. És una placa que treballa a una gran velocitat de rellotge (84MHz) i a més, la seva flexibilitat en quan a perifèrics permet poder ser més flexible per afegir objectius si se'n compleixen els escrits en el present treball. Es necessita aquesta velocitat del microcontrolador per a fer front a les pertorbacions del sistema. Amb aquest microcontrolador el sistema és més ràpid en reaccionar.

Tenint tots aquests conceptes ja es pot començar a fer el model matemàtic en MATLAB. D'aquesta manera es pot realitzar una simulació i veure com reaccionarà el sistema. Fem servir MATLAB com a eina de simulació ja que és una eina que s'ha fet servir durant la carrera, per tant és coneguda i permet fer les simulacions que es volen fer.

Capítol 3

Modelització matemàtica

En aquest capítol es realitza la modelització matemàtica d'aquest sistema per a poder realitzar la posterior simulació amb Matlab.

Per realitzar la modelització matemàtica es podien fer servir diversos mètodes, entre els quals els més rellevants per aquests casos són:

- Euler-Newton: El qual es basa en el sumatori de forces i moments per a determinar una relació entre totes les forces que intervenen en un sistema.
- Lagrange: Aquest mètode es basa en la diferència entre l'energia cinètica i l'energia potencial que actua en un sistema per a determinar la relació entre les velocitats del sistema.

Aquesta modelització es realitza mitjançant el mètode de Lagrange ($L(t)$, operador de Lagrange), que permet simplificar l'anàlisi matemàtic de la interacció dinàmica entre el plat i la bola. S'ha de realitzar la diferència entre l'energia cinètica $E_c(t)$ i l'energia potencial $E_p(t)$. En aquest cas, s'han importat els càlculs de Lagrange per trobar les equacions del moviment a través de diversos treballs reflectits a la bibliografia, tot i que la base dels càlculs s'ha obtingut del següent treball: Martínez Velasco et al.

$$L(t) = E_c(t) - E_p(t) \quad (3.1)$$

La següent il·lustració mostra els paràmetres que intervenen en el sistema per a un dels eixos de moviment:

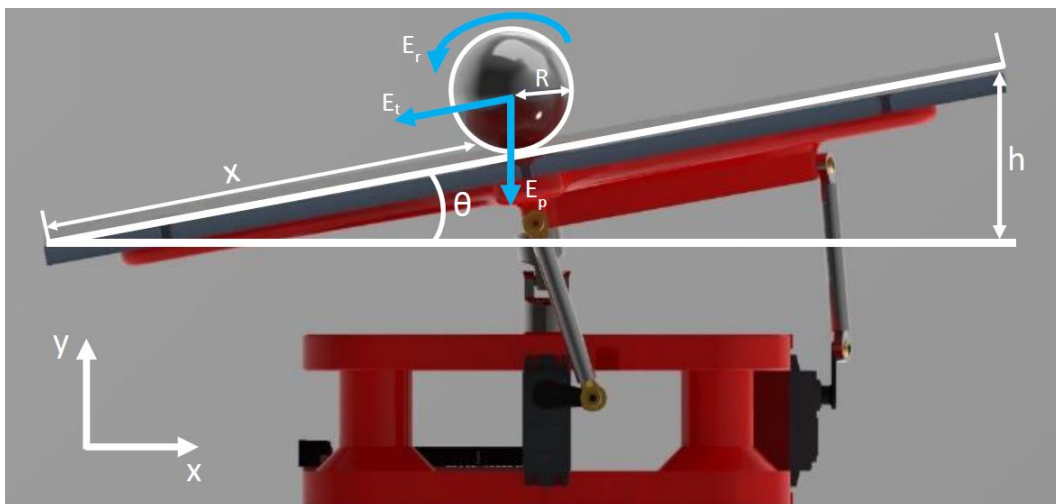


Figura 3.1: Paràmetres que intervenen en el nostre sistema.

3.1 Equació de Langrange

L'energia cinètica del sistema és pot dividir en moviment de translació i rotacional. A la vegada, aquesta translació i rotació es pot analitzar com dos components per separat. Un component és la plataforma ($E_{r2}(t)$ i $E_{t2}(t)$), i l'altre és l'esfera ($E_{r1}(t)$ i $E_{t1}(t)$) que hi ha a sobre. Això significa que hi ha quatre components que componen l'energia cinètica del sistema:

$$E_c(t) = E_{r1}(t) + E_{t1}(t) + E_{r2}(t) + E_{t2}(t) \quad (3.2)$$

Per a realitzar l'anàlisi del nostre sistema, s'han de fer algunes consideracions en quant al seu funcionament:

- La bola roda quan es desplaça per sobre de la plataforma.
- La bola és sòlida, homogènia i simètrica.
- La fricció entre el plat i la bola és negligible.
- Durant tota la dinàmica, l'esfera està en contacte constant amb la plataforma.

A continuació podem procedir a trobar cadascun dels elements que formen l'energia cinètica del nostre sistema.

$$\begin{cases} E_{r1}(t) = \frac{1}{2} \cdot I_e \cdot \omega_b^2(t) = \frac{1}{2} \cdot I_e \cdot \left(\frac{\dot{r}^2(t)}{R}\right)^2 \\ E_{t1}(t) = \frac{1}{2} \cdot m \cdot \dot{r}^2(t) \\ E_{r2}(t) = \frac{1}{2} \cdot I_e \cdot \dot{\theta}^2(t) \\ E_{t2}(t) = \frac{1}{2} \cdot m \cdot \dot{s}^2(t) \end{cases} \quad (3.3)$$

On $I_e = \frac{2}{5} \cdot m \cdot R^2$ és el moment d'inèrcia d'una esfera sòlida de radi R i massa m , ω és la seva velocitat angular. La \dot{r} és la velocitat lineal de l'esfera: $\dot{r}^2 = \dot{x}^2 + \dot{y}^2$. I velocitat angular de la bola (ω_e) és: $\omega_e = \frac{\dot{r}}{R}$

En el cas del plat, la velocitat angular $\dot{\theta}$ s'expressa com $\dot{\theta}^2 = \dot{\theta}_x^2 + \dot{\theta}_y^2$ i la seva velocitat lineal a causa de la rotació, està donada per $\dot{s}^2 = (x\dot{\theta}_x + y\dot{\theta}_y)^2$.

Per altra banda l'energia potencial segueix la següent equació:

$$E_p(t) = -m \cdot g \cdot h \quad (3.4)$$

L'altura h s'aconsegueix a partir de la següent equació: $h = x \cdot \sin(\theta_x) + y \cdot \sin(\theta_y)$. Tenint les equacions que descriuen les energies que afecten al sistema, l'equació de Lagrange queda de la següent forma:

$$L(t) = \frac{1}{2} \cdot I_e \cdot \left(\frac{\dot{r}^2(t)}{R}\right)^2 + \frac{1}{2} \cdot m \cdot \dot{r}^2(t) + \frac{1}{2} \cdot I_e \cdot \dot{\theta}^2(t) + \frac{1}{2} \cdot m \cdot \dot{s}^2(t) + m \cdot g \cdot (x \cdot \sin(\theta_x) + y \cdot \sin(\theta_y)) \quad (3.5)$$

Es reescriu l'equació anterior per a que tots els termes depenguin de X i de Y:

$$L(t) = \frac{1}{2} \cdot I_e \cdot (\dot{\theta}_x^2(t) + \dot{\theta}_y^2(t) + \frac{\dot{x}^2(t)}{R^2} + \frac{\dot{y}^2(t)}{R^2}) + \frac{1}{2} \cdot m \cdot (\dot{x}^2(t) + \dot{y}^2(t) + \dots \dots + (x\dot{\theta}_x(t) + y\dot{\theta}_y(t))^2) + m \cdot g \cdot (x \cdot \sin(\theta_x) + y \cdot \sin(\theta_y)) \quad (3.6)$$

3.2 Mètode Euler-Lagrange

Un cop s'aconsegueix l'equació amb el mètode de Lagrange, s'aplica el mètode de Euler-Lagrange per trobar les equacions que descriuen el moviment de sistema. Per aplicar aquest mètode, es parteix de la seva equació bàsica, on L és l'equació de Lagrange trobada anteriorment, q_i és la variable d'estat i Q_i és la component de força per a determinar les equacions diferencial del sistema i així obtindre una representació de l'estat.:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \left(\frac{\partial L}{\partial q_j}\right) = Q_j \quad (3.7)$$

En aquest sistema sistema, les incògnites que ens determinen els estats són el moviment de la bola sobre de la plataforma, i el moviment d'inclinació de la plataforma. El moviment de la inclinació de la plataforma és el moviment que controlem directament, i en conseqüència, modificarem el moviment de la bola. Seguint el treball de Garzón Melo et al. [2016], s'aplica l'equació anterior per a les coordenades de moviment x i y , que descriuen la posició de la bola i l'angle d'inclinació del plat:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \left(\frac{\partial L}{\partial x}\right) = 0 \quad (3.8)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}}\right) - \left(\frac{\partial L}{\partial y}\right) = 0 \quad (3.9)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_x}\right) - \left(\frac{\partial L}{\partial \theta_x}\right) = T_x \quad (3.10)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_y}\right) - \left(\frac{\partial L}{\partial \theta_y}\right) = T_y \quad (3.11)$$

Avaluant l'equació (3.8) s'obté l'acceleració lineal de la bola sobre la coordenada X , que permet, obtindre la posició sobre el mateix eix:

$$\begin{aligned} \frac{\partial L(t)}{\partial x(t)} &= \frac{1}{2}m[2x\dot{\theta}_x^2 + 2y\dot{\theta}_x\dot{\theta}_y^2] + mgsin\theta_x \\ \frac{\partial L(t)}{\partial \dot{x}(t)} &= \frac{1}{2}I_e \frac{2\dot{x}}{R^2} + \frac{1}{2}m2\dot{x} \\ \frac{d}{dt} \frac{\partial L(t)}{\partial \dot{x}(t)} &= \frac{I_e}{R^2}\ddot{x} + m\ddot{x} \\ \ddot{x}\left[\frac{I_e}{R^2} + m\right] - mx\dot{\theta}_x^2 - my\dot{\theta}_x\dot{\theta}_y - mgsin\theta_x &= 0 \end{aligned} \quad (3.12)$$

Avaluant l'equació (3.9) s'obté l'acceleració lineal de la bola sobre la coordenada Y , que permet, obtindre la posició sobre el mateix eix:

$$\begin{aligned} \frac{\partial L(t)}{\partial y(t)} &= \frac{1}{2}m[2y\dot{\theta}_y^2 + 2x\dot{\theta}_x\dot{\theta}_y^2] + mgsin\theta_y \\ \frac{\partial L(t)}{\partial \dot{y}(t)} &= \frac{1}{2}I_e \frac{2\dot{y}}{R^2} + \frac{1}{2}m2\dot{y} \\ \frac{d}{dt} \frac{\partial L(t)}{\partial \dot{y}(t)} &= \frac{I_e}{R^2}\ddot{y} + m\ddot{y} \\ \ddot{y}\left[\frac{I_e}{R^2} + m\right] - my\dot{\theta}_y^2 - mx\dot{\theta}_x\dot{\theta}_y - mgsin\theta_y &= 0 \end{aligned} \quad (3.13)$$

Avaluant l'equació (3.10) s'obté l'acceleració angular de la plataforma sobre l'eix X, que permet, obtenir l'angle de gir sobre aquest mateix eix, en aquest cas hi ha un parell (T_x) perquè la plataforma té un força deguda a un servomotor:

$$\begin{aligned}\frac{\partial L(t)}{\partial \theta_x(t)} &= mgx \cos \theta_x \\ \frac{\partial L(t)}{\partial \dot{\theta}_x(t)} &= \frac{1}{2} I_e 2 \dot{\theta}_x + \frac{1}{2} m [2x^2 \dot{\theta}_x + 2xy \dot{\theta}_y] = I_e \dot{\theta}_x + mx^2 \dot{\theta}_x + mxy \dot{\theta}_y \\ \frac{d}{dt} \frac{\partial L(t)}{\partial \dot{\theta}_x(t)} &= I_e \ddot{\theta}_x + m(x^2 \ddot{\theta}_x + 2x \dot{x} \dot{\theta}_x) + m(\dot{x} y \dot{\theta}_y + x \dot{y} \dot{\theta}_y + xy \ddot{\theta}_y) \\ I_e \ddot{\theta}_x + m(x^2 \ddot{\theta}_x + 2x \dot{x} \dot{\theta}_x) + m(\dot{x} y \dot{\theta}_y + x \dot{y} \dot{\theta}_y + xy \ddot{\theta}_y) - mgx \cos \theta_x &= T_x\end{aligned}\tag{3.14}$$

Avaluant l'equació (3.11) s'obté l'acceleració angular del plat sobre l'eix Y, que permet, obtenir l'angle de gir sobre aquest mateix eix, en aquest cas hi ha un parell (T_x) perquè la plataforma té un força deguda a un servomotor, en aquest cas hi ha un parell (T_y) perquè la plataforma té un força deguda a un servomotor:

$$\begin{aligned}\frac{\partial L(t)}{\partial \theta_y(t)} &= mgy \cos \theta_y \\ \frac{\partial L(t)}{\partial \dot{\theta}_y(t)} &= \frac{1}{2} I_e 2 \dot{\theta}_y + \frac{1}{2} m [2y^2 \dot{\theta}_y + 2xy \dot{\theta}_x] = I_e \dot{\theta}_y + my^2 \dot{\theta}_y + mxy \dot{\theta}_x \\ \frac{d}{dt} \frac{\partial L(t)}{\partial \dot{\theta}_y(t)} &= I_e \ddot{\theta}_y + m(y^2 \ddot{\theta}_y + 2y \dot{y} \dot{\theta}_y) + m(\dot{x} y \dot{\theta}_x + x \dot{y} \dot{\theta}_x + xy \ddot{\theta}_x) \\ I_e \ddot{\theta}_y + m(y^2 \ddot{\theta}_y + 2y \dot{y} \dot{\theta}_y) + m(\dot{x} y \dot{\theta}_x + x \dot{y} \dot{\theta}_x + xy \ddot{\theta}_x) - mgy \cos \theta_y &= T_y\end{aligned}\tag{3.15}$$

De les equacions 3.12,3.13,3.14 i 3.15 es poden extreure les acceleracions lineals de la bola i les acceleracions angulars del plat en els eixos de rotació, d'aquesta forma, s'obtenen el conjunt d'equacions 3.16,3.17,3.18 i 3.19 que descriuen el comportament dinàmic del sistema:

$$\ddot{x} = \frac{m}{\frac{I_e}{R^2} + m} \cdot [x \dot{\theta}_x^2 + y \dot{\theta}_x \dot{\theta}_y + g \sin \theta_x]\tag{3.16}$$

$$\ddot{y} = \frac{m}{\frac{I_e}{R^2} + m} \cdot [y \dot{\theta}_y^2 + x \dot{\theta}_x \dot{\theta}_y + g \sin \theta_y]\tag{3.17}$$

$$\ddot{\theta}_x = \frac{1}{I_e + mx^2} \cdot [T_x - m(2x \dot{x} \dot{\theta}_x + \dot{x} y \dot{\theta}_y + x \dot{y} \dot{\theta}_y + xy \ddot{\theta}_y - gx \cos \theta_x)]\tag{3.18}$$

$$\ddot{\theta}_y = \frac{1}{I_e + my^2} \cdot [T_y - m(2y \dot{y} \dot{\theta}_y + \dot{x} y \dot{\theta}_x + x \dot{y} \dot{\theta}_x + xy \ddot{\theta}_x - gy \cos \theta_y)]\tag{3.19}$$

D'aquesta forma, es prenen com a variables d'estats:

$$\begin{array}{llll} x_1 = x & x_2 = \dot{x} & x_3 = \theta_x & x_4 = \dot{\theta}_x \\ x_5 = y & x_6 = \dot{y} & x_7 = \theta_y & x_8 = \dot{\theta}_y \end{array}$$

El resultat és un sistema d'equacions no lineals. Es linealitzava i es resol el sistema per a un punt d'operació específic. S'agafa el punt d'equilibri d'estabilització de la

plataforma. Es defineix un vector d'estats \mathbf{q}_i ($x_1, x_2, x_3, x_4, x_5, x_6, x_7$ i x_8) a partir de les coordenades, les seves derivades respecte al temps en un punt específic de valor 0 per a totes les variables del vector d'estat \mathbf{x} . Tenint en compte que les entrades al sistema corresponen als parells externs exercits pels servos i que les sortides del model es trien que coincideixin amb els estats del sistema $i=\mathbf{x}$ es pot representar el sistema d'equacions.

Taula 3.1: Paràmetres i variables físiques del sistema *Ball on Plate*

Nom	Símbol	Valor	Unitats
Gravetat	g	9,80665	m/s^2
Massa de la bola	m	0.175	kg
Radi de la bola	R	0.0175	m
Inèrcia de la bola	I_e	$I_e = \frac{2}{5}mR^2$	$kg * m^2$
Radi de la bola	R	0.0175	m
Desplaçament de la bola en l'eix "x"	x	—	m
Velocitat de la bola en l'eix "x"	\dot{x}	—	m/s
Acceleració de la bola en l'eix "x"	\ddot{x}	—	m/s^2
Desplaçament de la bola en l'eix "y"	y	—	m
Velocitat de la bola en l'eix "y"	\dot{y}	—	m/s
Acceleració de la bola en l'eix "y"	\ddot{y}	—	m/s^2
Angle entre l'eix "x" i el pla horitzontal	θ_x	—	rad
Angle entre l'eix "y" i el pla horitzontal	θ_y	—	rad
Velocitat lineal degut a la rotació de la plataforma	\dot{s}	$\dot{s}^2 = (x\dot{\theta}_x + y\dot{\theta}_y)^2$	m/s
Velocitat lineal de la bola	\dot{r}	$\dot{r}^2 = \dot{x}^2 + \dot{y}^2$	m/s
Velocitat angular de la bola	ω_b	$\omega = \frac{\dot{r}}{R}$	rad/s
Velocitat angular de la plataforma	$\dot{\theta}$	$\dot{\omega} = \dot{\theta}_x^2 + \dot{\theta}_y^2$	rad/s
Altura de la bola	h	$h = x\sin\theta_x + y\sin\theta_y$	m
Parell dels servomotors	T_x, T_y	—	N/m

3.3 Simulació

Quan es tenen les equacions d'estat, abans de procedir amb les simulacions amb Matlab i representar així el sistema d'equacions, és va fer una petita anàlisi cinemàtica i dinàmica amb el PTC CREO. Es va definir en quines peces hi havia motors, se'ls hi va donar un moviment oscil·lant i finalment, se li va afegir l'acció de la gravetat al conjunt per tindre una primera visualització dels possibles problemes que podia tindre el disseny mecànic. En aquesta anàlisi prèvia es buscava contacte entre peces mecàniques, moviments erronis de les peces, etc. Aquests problemes haurien provocat errors en la posterior simulació amb el Multibody.

Un cop verificat que el disseny era correcte, es va procedir a realitzar la simulació amb Matlab per trobar les matrius de control i després, es va simular el sistema complet amb l'ajuda de l'aplicació Simulink.

3.3.1 Matlab

Com es pot veure a les equacions d'estat, aquest sistema no és lineal. El primer que s'ha de fer és linealitzar el sistema per poder-lo resoldre. Per fer això es fa servir el Matlab per linealitzar-lo trobant les equacions d'estat, que tenen la següent forma:

$$\begin{aligned} \dot{q} &= Aq + Bu \\ y &= Cq + Du \end{aligned} \tag{3.20}$$

Un cop trobades les matrius A, B, C i D, s'escolleix el controlador que es vol fer servir per controlar aquest sistema, com ja s'ha comentat, i sent un dels objectius d'aquest treball, aquest sistema es vol controlar a través d'un control LQR.

Abans de tot, mitjançant dues funcions de l'aplicació Matlab es comprova que el sistema és controlable i observable:

- **Observable:** La comprovació de l'observabilitat en aquest sistema ha donat com a resultat, la màxima observabilitat possible, això significa que l'estat d'aquest sistema es pot visualitzar mitjançant els sensors.
- **Controlable:** La comprovació de la controlabilitat també ha donat com a resultat la màxima controlabilitat possible. Citant el treball Garzón Melo et al. [2016], defineix que un sistema és completament controlable si és possible transferir el sistema d'un estat inicial arbitrari fins a qualsevol estat desitjat en un temps finit.

Un cop comprovats els aspectes d'observabilitat i controlabilitat. Es pot procedir a trobar la matriu K que ens dóna el control LQR. La matriu K és la matriu que dóna la sortida als servomotors en funció de l'entrada per a controlar el sistema.

Per trobar aquesta matriu K, es fa servir una eina del Matlab, a la qual, se li han passat les matrius d'estats, i les importàncies que es vol donar a cada variable (posició de la bola, velocitat, posició de la plataforma, etc.). Aquesta K, en aquest cas s'ha de trobar en el domini discret, ja que la implementació d'aquest control es realitza mitjançant un microcontrolador que treballa en el domini discret.

Un cop s'ha calculat la K, gràcies a l'eina Matlab es troba la funció de transferència en el domini discret.

Finalment es determina un temps límit de la funció i un temps de mostreig, d'aquesta manera ja es pot tindre una sèrie de corbes que donen informació de com es comporta el sistema i, si és necessari, es canvien els paràmetres d'importància de cadascuna de les variables per a modificar la reacció que es vol que tingui aquest sistema.

3.3.1.1 Corba Posició-Temps

La primera corba que interessa saber és la que indica quant temps tarda la bola en assolir la posició desitjada partint d'una posició arbitrària:

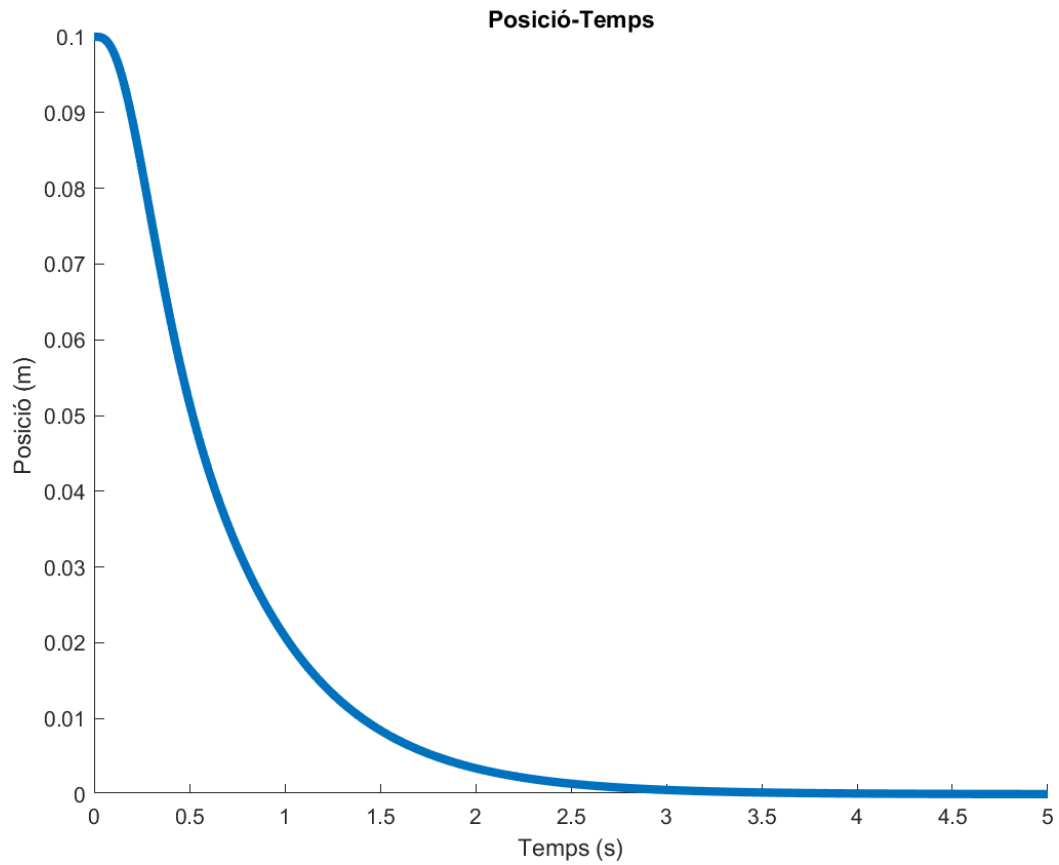


Figura 3.2: Representació de la posició de la bola en funció del temps.
Font: Matlab

3.3.1.2 Estat de sortida en funció del temps

En la següent corba que es mostra, es pot apreciar la reacció que tindrà el sistema davant d'una pertorbació, aquest gràfic ens mostra l'estat de sortida en funció del temps.

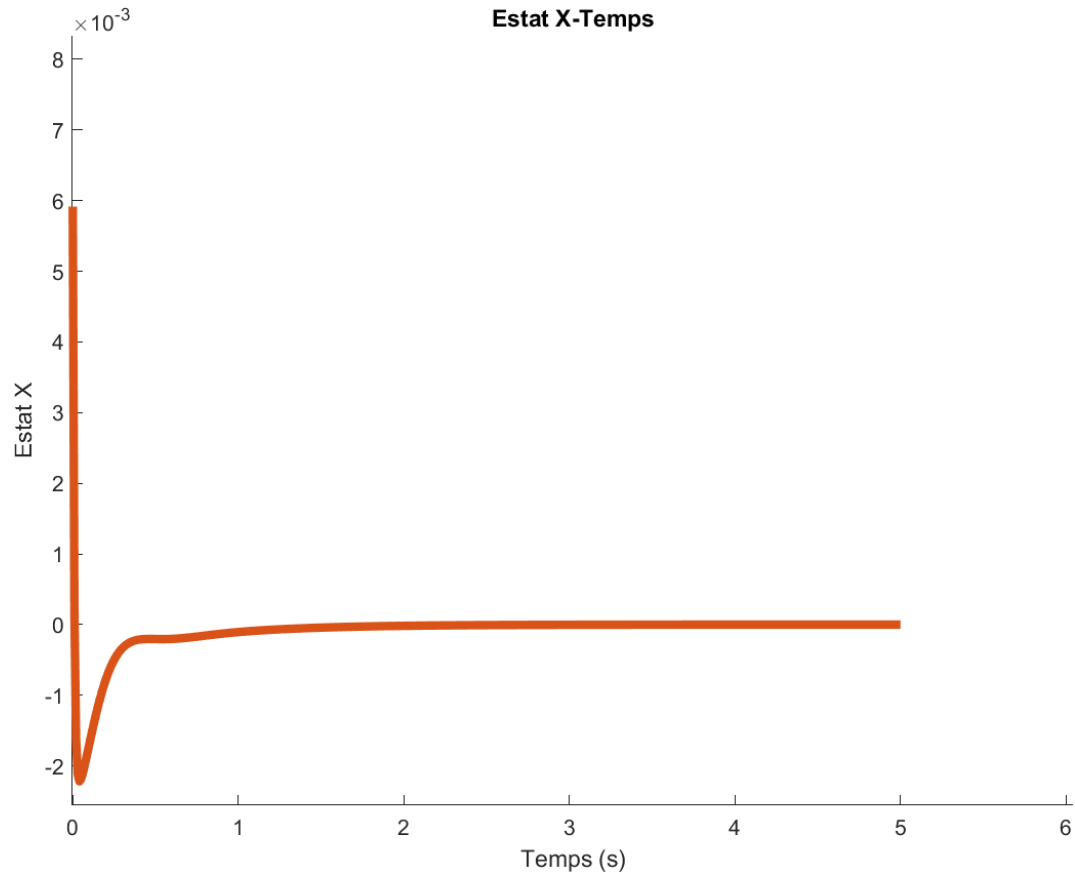


Figura 3.3: Representació de l'estat de sortida en funció del temps.
Font: Matlab

3.3.1.3 Representació del Lloc Geometric de les Arrels

Per últim, la següent representació és el Lloc Geometric de les Arrels, que ens permet conèixer d'una forma qualitativa si el sistema es tornarà inestable o no.

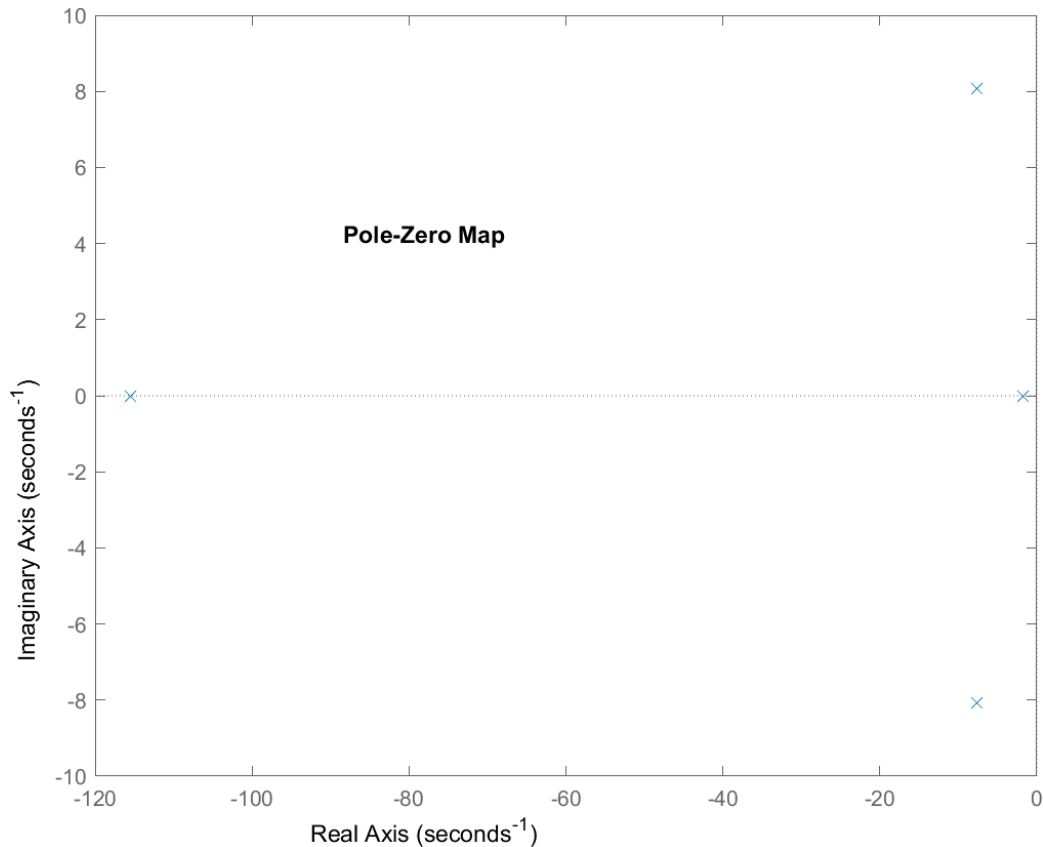


Figura 3.4: Representació del Lloc Geometric de les Arrels.

Font: Matlab

El codi de Matlab que és fa servir es pot observar en l'annex F.

Posteriorment, quan s'ha acabat de comprovar el sistema de control amb el Matlab, es pot procedir a realitzar la simulació del sistema complet amb el Simulink, aquesta aplicació permet fer la simulació del sistema de control trobat anteriorment junt amb el disseny mecànic dissenyat amb el programa PTC CREO.

3.3.2 Simulink

Simulink és una eina que hi ha a dins de la pròpia aplicació de Matlab, que permet fer el modelatge i simulació d'aquest sistema complet, d'aquesta forma, es pot tindre una idea de com reacciona el sistema a les diferents pertorbacions. Amb aquesta aplicació és possible simular el sistema per a poder visualitzar com reacciona el sistema quan ha de portar la bola a un punt de la plataforma, o bé, si la bola hagués de seguir una trajectòria determinada.

Per realitzar el modelatge d'un sistema de control, una de les moltes variables que s'ha de tindre en compte és el tipus de realimentacions que hi ha, i per tant aquesta va ser una decisió que també s'ha pres en aquest sistema. S'ha escollit realitzar un tipus de sistema de control seguidor:

- Sistema seguidor: Aquest sistema es caracteritza per tindre una doble realimentació. Un sistema amb aquest tipus de control intenta seguir una trajectòria definida a l'entrada.

Es fan dues simulacions del sistema, la primera, es tracta del modelatge del sistema teòric, que significa que la resolució del moviment dels actuadors i la dels sensors és

mitjançant les equacions d'estat, trobades gràcies a l'equació de Lagrange trobada amb el Matlab:

3.3.2.1 Simulació model teòric

Com s'ha comentat, per a aquesta simulació s'ha realitzat el modelatge del sistema de forma teòrica. Per realitzar aquest modelatge es necessiten les parts que es veuen a la imatge:

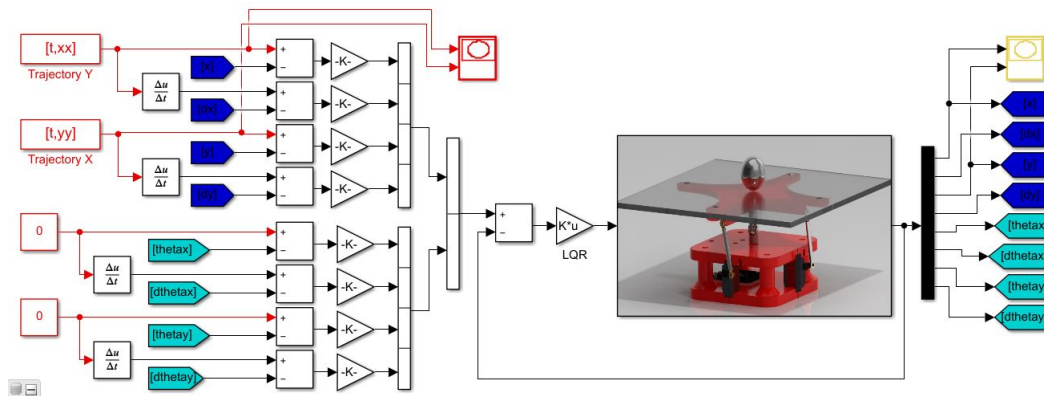


Figura 3.5: Model teòric del sistema de control.

Font: Matlab

Com es pot veure a la imatge 3.5 es poden apreciar diferents parts que a continuació es descriuen d'esquerra a dreta:

- Senyals d'entrada: A l'esquerra de tot del sistema es poden veure els senyals d'entrada del sistema. Aquests senyals són els senyals de referència o objectius, els quals el sistema ha d'assolir gràcies al control, ja siguin punts sobre la plataforma o trajectòries. Com es pot veure, els únics senyals diferents de zero són la posició X i Y de la bola sobre de la plataforma, els quals es deriven per aconseguir la velocitat d'aquesta. Es pot apreciar que en el cas de l'angle de la plataforma el valor és zero (valor quan la plataforma està anivellada). L'objectiu és que la bola estigui a la posició desitjada amb la plataforma anivellada.
- Primera realimentació: Aquesta primera realimentació es pot apreciar a la dreta de les derivades dels senyals d'entrada, amb marques blaves (blau fosc és la posició de la bola i blau clar fa referència a la inclinació de la plataforma). Es resten els senyals d'entrada esmentats en el punt anterior amb aquests senyals per obtenir el primer senyal d'error, fruit de la primera realimentació.
- Guany dels senyals: Després de realitzar aquest error, amb l'ajuda d'una aplicació del Simulink anomenada Control System Tuner (modifica els guanys escollits en un sistema per millorar la resposta) es modifiquen els senyals d'errors a través dels guanys per aconseguir tindre una millor resposta en el sistema, es pot veure els senyals modificats en l'annex A, figura A.1.
- Segona realimentació: El segon senyal d'error és la diferència entre els senyals que surten dels guanys i els senyals de sortida. La diferència d'aquests dos

senyals ens dona el segon senyal d'error, que llavors passa per la matriu LQR de control, que dona valor als dos senyals d'actuació (servomotors).

- Equacions d'estats: Les equacions d'estat estan situades al bloc central on hi ha una fotografia del sistema. L'entrada de les equacions d'estat són els dos senyals d'actuació esmentats anteriorment. En funció de quin valor tinguin, en passar per les equacions d'estat, donaran un valor de sortida o un altre, en funció d'en quin estat de control estigui el sistema. Les sortides de les equacions d'estat seran les que s'agafaran per fer les realimentacions, i es tornarà a començar el cicle.

D'aquesta forma el sistema anirà fent iteracions per controlar l'entrada a les equacions d'estat. Això provocarà que els senyals de sortida siguin cada cop més semblants als d'entrada, l'error es farà cada cop més petit. En l'apartat de resultat es pot veure com reacciona el sistema de control per seguir un senyal de referència, a la figura 6.3.

Quan s'ha verificat que el resultat del model de Simulink, que donat un senyal d'entrada al sistema el sistema es capaç de seguir-lo amb un error petit, es procedeix a fer el modelatge del sistema real.

3.3.2.2 Simulació del model real

Evidentment no es pot fer el modelatge del sistema 100% real, ja que hi intervenen massa factors que no es poden controlar, com les toleràncies de les peces, medi ambient, etc. Però s'ha intentat modelar el sistema més real possible. Per realitzar-ho, el primer que s'ha fet és importar el disseny CAD realitzat amb el PTC CREO al Simulink. El Simulink és capaç de modelar el sistema gràcies als paràmetres que proporciona el PTC CREO, com podria ser la unió rígida entre peces, els contactes que hi ha, on van situats els eixos dels motors i quina peça mouen, etc. Gràcies a aquesta importació es va aconseguir el següent model del sistema real a partir del disseny CAD:

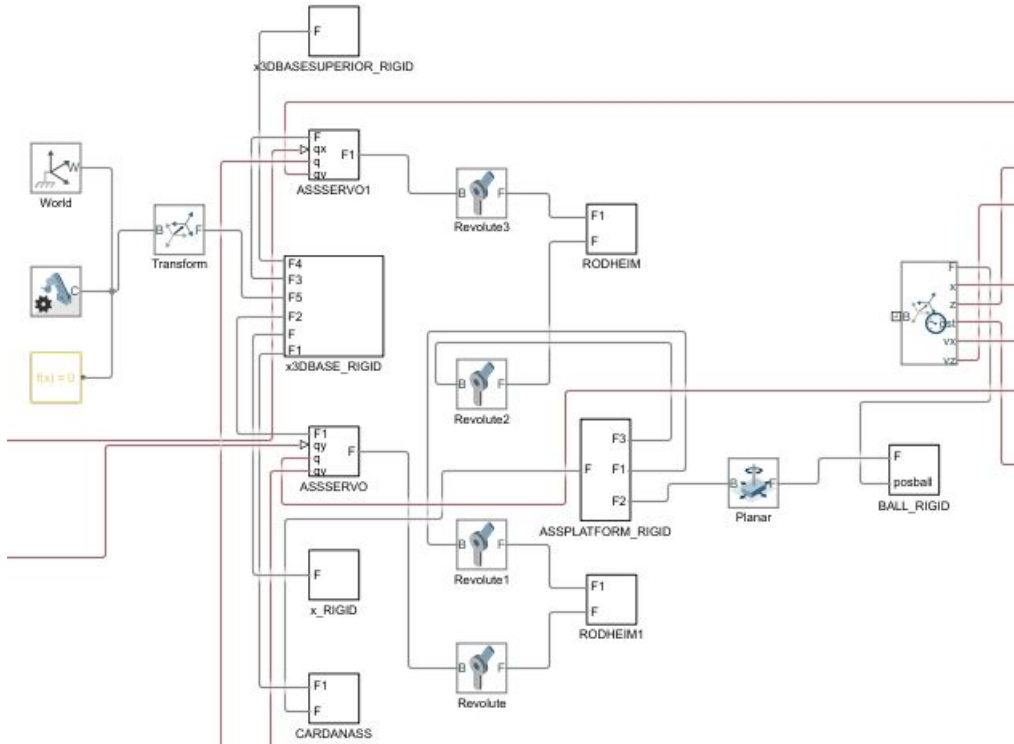


Figura 3.6: Modelització del sistema a partir del disseny CAD.
 Font: Matlab

El model de la imatge 3.6 és el que substitueix a les equacions d'estat en la simulació teòrica (a partir d'ara anomenat "model realista"), de forma que les entrades d'aquest model són el senyal que van als servomotors i les sortides són els senyals dels sensors. Amb aquest canvi, el model real del sistema queda de la següent forma:

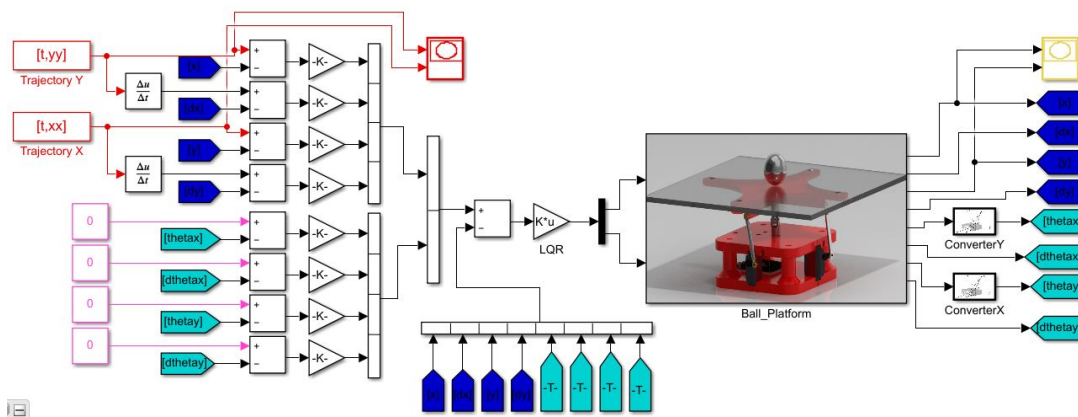


Figura 3.7: Model del sistema real a partir del disseny en CAD.
 Font: Matlab

Com es pot observar en la imatge 3.7, el model del sistema és molt semblant al cas teòric que es pot observar en la imatge 3.5 amb algunes diferències.

- Bloc central: Com s'ha comentat, el bloc central s'ha substituït pel model més realista importat del PTC CREO.

- L'altra diferència que hi ha entre aquest model i l'altre són els convertidors d'angle que hi ha a la sortida del bloc del "model realista". Aquests dos convertidors són necessaris perquè, en el sistema real, no hi ha forma de poder saber l'angle de la plataforma, per tant, aquest angle s'extreu indirectament a partir de l'angle de cada servomotor, gràcies a la cinemàtica inversa.

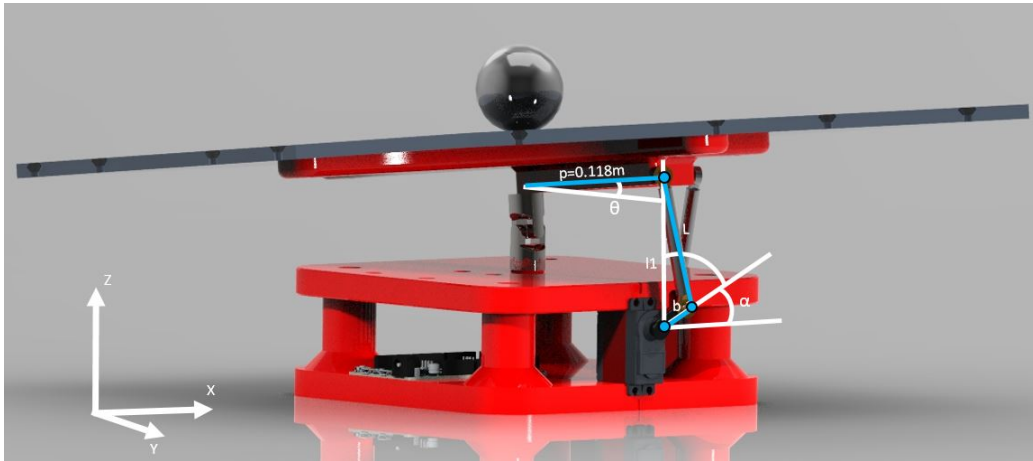


Figura 3.8: Relació entre la inclinació de la plataforma i l'angle d'inclinació dels Servomotors.

Font: PTC CREO

Quan es té realitzat el sistema i s'han definit els valors del guany, amb aquest model realista del sistema i amb ajuda d'una altra aplicació del Matlab anomenada Multibody. Amb aquesta aplicació és possible visualitzar el sistema en format 3D quan s'executa el control. Gràcies a aquesta aplicació es pot visualitzar el sistema de la següent manera quan fa el control de la bola, amb aquesta visualització es veu de forma molt més clara com es comporta el sistema, i es pot intuir si funcionarà correctament al sistema real.

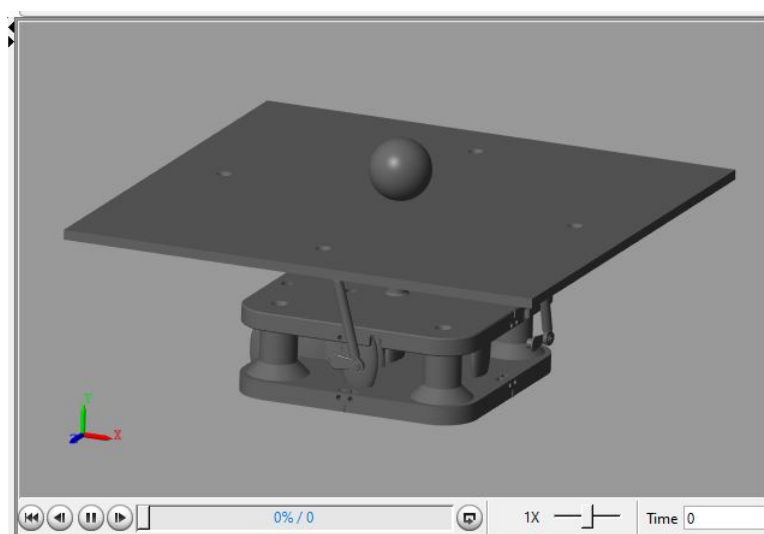


Figura 3.9: Visualització del sistema amb el Multibody.

Font: Matlab

Capítol 4

Disseny mecatrònic del sistema

En aquest capítol es descriu tot el Disseny Mecatronic. Es fa una descripció de tots els components que formen part del sistema. En concret es fa una descripció del component, perquè s'ha escollit i com repercuteix a la resta del sistema.

Un cop s'ha decidit quines característiques es vol que tingui el sistema, i ja s'hagi validat la simulació, llavors es pot començar a decidir els components que el formaran. A l'hora de fer l'elecció de tots els components s'ha tingut en compte que alguns components poden afectar en altres de forma indirecta.

4.1 Arduino DUE

Arduino és una plataforma electrònica de codi obert basada en un tipus de hardware i software fàcil d'utilitzar i amb el qual es poden realitzar ràpidament prototips de sistemes. Algunes dels grans avantatges de fer servir Arduino són:

- Baix cost: Les plaques d'Arduino són relativament barates en comparació a altres microcontroladors.
- Entorn de programació simple i clar.
- Software de codi obert i extensible.
- Codi obert i hardware extensible.
- Un altre dels avantatges que té, és la gran quantitat d'informació que hi ha de forma oberta i gratuïta sobre aquesta plataforma i sistemes que han desenvolupat els usuaris, pel que hi ha nombrosos exemples i suports.

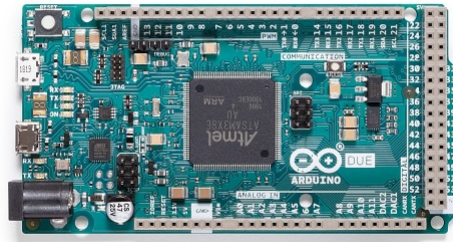


Figura 4.1: Placa Arduino DUE

Font: <https://store.arduino.cc/arduino-due>

La placa que es va escollir per dur a terme aquest projecte és un Arduino DUE. Aquesta placa va ser escollida principalment per la seva rapidesa de processador, ja que treballa a una velocitat de 84 MHz de clock de rellotge. En aquest projecte es necessita aquesta velocitat de rellotge per a poder reaccionar ràpidament davant de les pertorbacions. També es va escollir perquè aquesta placa dóna molta flexibilitat per afegir elements gràcies a la seva quantitat de pins que incorpora, tant pins digitals, sortides PWM, analògics, etc.

Seguint amb el mètode del Disseny Mecatronic, a l'escollir aquesta placa, s'ha tingut en compte que s'han hagut de fer algunes modificacions pel que fa a la connexió dels actuadors com els servomotors, ja que els pins d'aquesta placa en particular subministren 3,3V, a part que no és aconsellable connectar directament actuadors de potència directament a la placa, per tant, s'ha hagut de realitzar un circuit de connexió per solucionar-ho.

4.2 Làmina tàctil resistiva

Aquest element és el que fa servir el sistema per saber la posició de la bola sobre de la plataforma. La posició de la bola permet la realimentació del sistema, ja que gràcies a aquesta dada, el sistema pot saber la posició i velocitat de la bola, i fer les correccions pertinents per controlar-la.

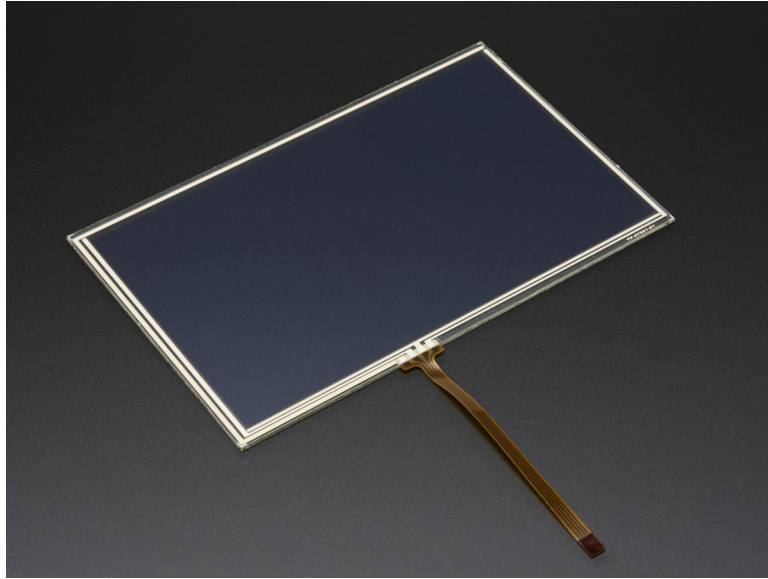


Figura 4.2: Làmina tàtil resistiva.

Font: <https://www.adafruit.com/product/1676>

Es va escollir aquest tipus de sensor per saber la posició de la bola per la seva rapidesa, ja que els altres mètodes de sensòrica per aquesta aplicació són més costosos. Un altre avantatge d'aquest sistema és que és relativament senzill d'implementar, comparat amb altres com podria ser la visió artificial.

Les pantalles tàctils resistives són pantalles més econòmiques de fabricar i més fàcils de fer servir en entorns difícils en comparació a altres tipus de pantalles tàctils, com les capacitives.

En concret, la làmina que s'ha escollit és de 19 polzades (323x396mm). S'ha escollit aquesta mida perquè es vol que sigui molt visual el moviment de control del sistema, per tant, el posterior suport s'ha fet a mida per a aquest sensor. Aquesta mida de la pantalla provoca que hagin de realitzar un disseny mecànic que pugui aguantar el seu pes, i uns actuadors que puguin moure-la.

4.3 Servomotors (MG996R)

En aquest projecte es va escollir realitzar el moviment de la plataforma amb dos servomotors. Es va escollir aquest tipus d'actuadors perquè són actuadors que són fàcil d'utilitzar, ràpids, amb gran precisió i amb suficient parell motor (el nostre model té un parell motor de 1,08Nm) per a fer moure la plataforma junt amb el seu suport i una bola d'un kilogram, un cas que no es contempla, ja que la bola utilitzada té un pes menor.



Figura 4.3: Servomotor MG996R.

Font: http://www.hobbyandyou.com/content/images/thumbs/0002238_towerpro-mg996r-metal-gear-servo-motor.jpeg

Per poder haver seleccionat aquest servomotor i seguint amb el mètode del Disseny Mecatronic, primer s'ha d'haver comprovat quin era el parell necessari per a poder fer moure el conjunt de la plataforma més la bola en el cas que féssim servir una bola d'1kg (pitjor cas). Per fer-ho, s'ha hagut de fer un símil amb una biga. S'ha pogut comprovar que el parell mínim que es necessitava era de 0,323 Nm, per tant, els 1,08 Nm del nostre actuador és més que suficient.

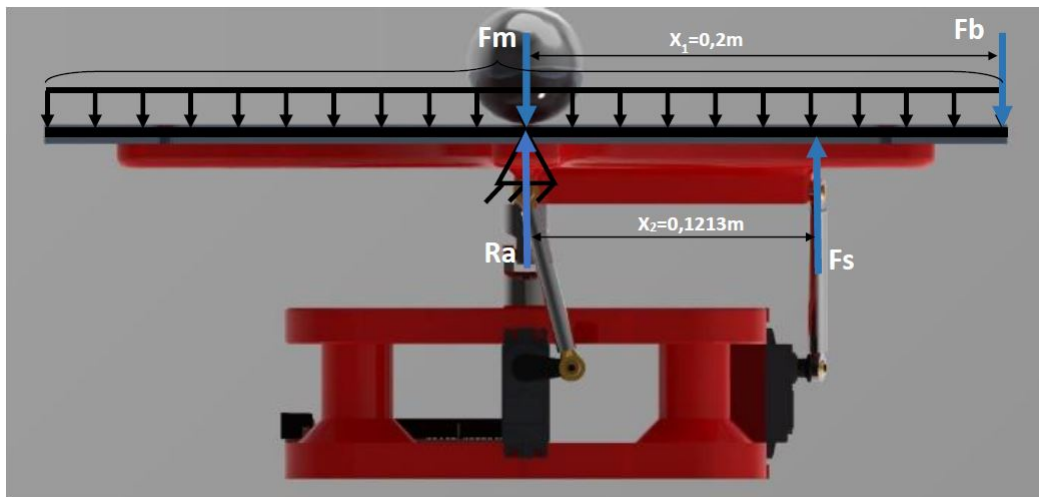


Figura 4.4: Comprovació de la força que arriba al servomotor fent el símil amb una biga.

Per últim, aquest servomotor no es podia connectar directament al PWM de l'Arduino com s'ha comentat anteriorment a causa de la tensió (l'Arduino dona 3,3V màxims en els seus pins, mentre que els servomotors en necessiten 4,8 de mínima). Així que s'ha hagut de connectar a través d'un circuit amplificador, el qual també s'ha comprovat per tal que tots els elements estiguessin dintre dels seus paràmetres de funcionament:

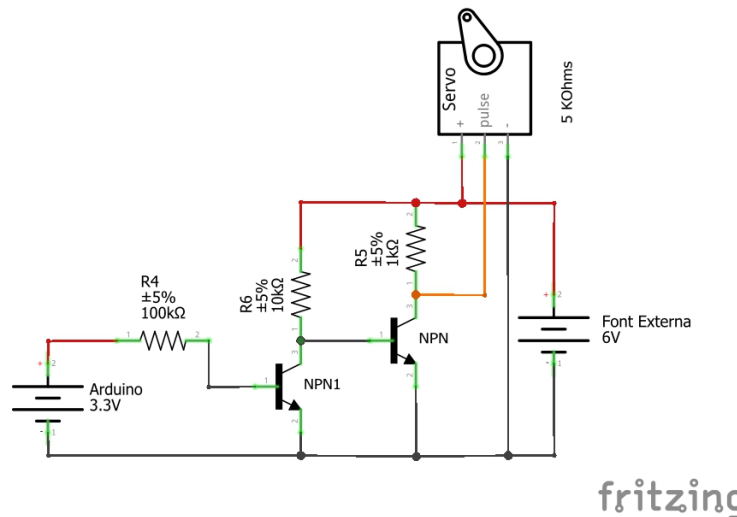


Figura 4.5: Connexió dels Servos amb la font d'alimentació i a l'Arduino a través de 2 transistors NPN en cascada.

Font: Fritzing program.

Aquest circuit agafa el senyal PWM de l'Arduino i el copia amplificat, el senyal que arriba als servomotors va des de 0 fins a 5,2V.

4.4 Teclat Matricial

Un teclat matricial és un dispositiu que agrupa diversos polsadors i, que permet controlar-los amb un nombre inferior de cables que si fossin per separat. Els polsadors estan disposats en files i columnes formant una matriu, de forma que, quan premem un polsador, es connecta una fila amb una columna i es pot saber el polsador que s'ha premut.

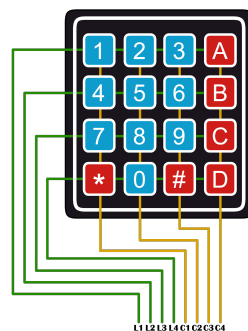


Figura 4.6: Teclat matricial indicat les seves connexions.
Font: <http://2.bp.blogspot.com/-4dsuRjAJyDQ/VcpG-1sB-GI/AAAAAAAAAJHU/DxM7i1U34Po/s1600/keypad.png>

Com es pot observar la connexió d'aquest teclat és molt senzilla, ja que simplement s'ha de connectar cada pin del teclat, amb un pin digital de l'Arduino.

4.5 Part Mecànica

La part mecànica del projecte és la responsable del suport de tots els components elèctrics, la làmina tàctil, els dos servomotors, l'Arduino i finalment, de totes les connexions. A més, la part mecànica s'encarrega de transformar el moviment rotacional dels servomotors, en rotació en l'eix X i Y de la plataforma. La major part de les parts mecàniques van ser de disseny de l'autor i van ser impreses amb una impressora 3D. A continuació es mostra el disseny definitiu que es va fer amb el programa PTC CREO, el qual conté totes les peces mecàniques, tant les dissenyades per l'autor, com les comprades:

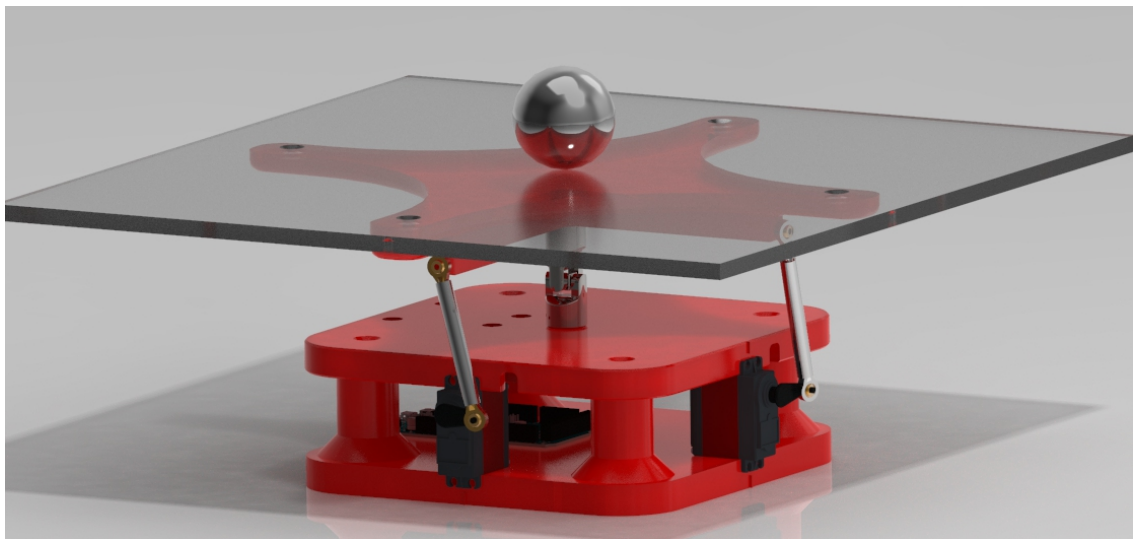


Figura 4.7: Disseny CAD del sistema complert.
Font: Programa PTC CREO.

A continuació es mostra cadascuna de les peces, junt amb una explicació de la seva tasca en el sistema.

4.6 Base Inferior

La base inferior és la peça encarregada del suport del pes del nostre sistema i del seu suport amb el terra. La superfície ha de ser completament llisa i anivellada perquè quan es col·loqui el sistema sobre una superfície, aquesta es mogui el menys possible. Si el terra de suport està anivellat, en conseqüència tot el sistema estarà anivellat. Aquesta base també és l'encarregada de la subjecció de diverses peces:

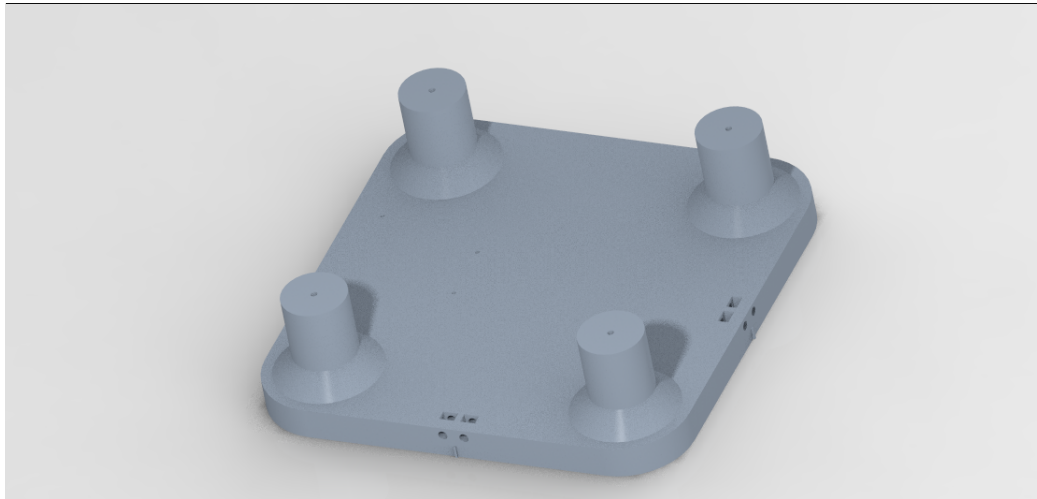


Figura 4.8: Base Inferior.
Font: Programa PTC CREO.

- Placa d'Arduino: Aquesta es col·loca amb cargols roscats de M2 a la superfície d'aquesta base.
- Servomotors: Aquesta base s'encarrega de la subjecció inferior dels dos Servomotors. Aquesta subjecció es fa mitjançant unes cavitats amb la forma d'una femella hexagonal i es cargola amb cargols de M4 tal com es veu a la imatge:

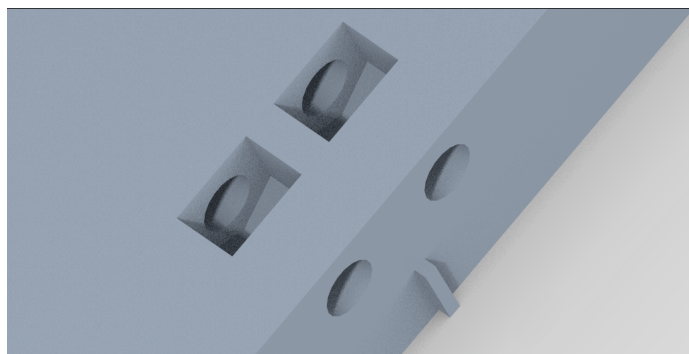


Figura 4.9: Unió del Servomotors.
Font: Programa PTC CREO.

- Subjecció de la base superior mitjançant 4 cargols de M4 de fusta. Aquesta subjecció es realitza a través de les quatre columnes cilíndriques que hi ha a la base.

Aquesta peça, en ser una peça de cert grau de complexitat per a ser mecanitzada, es va realitzar amb una impressora 3D.

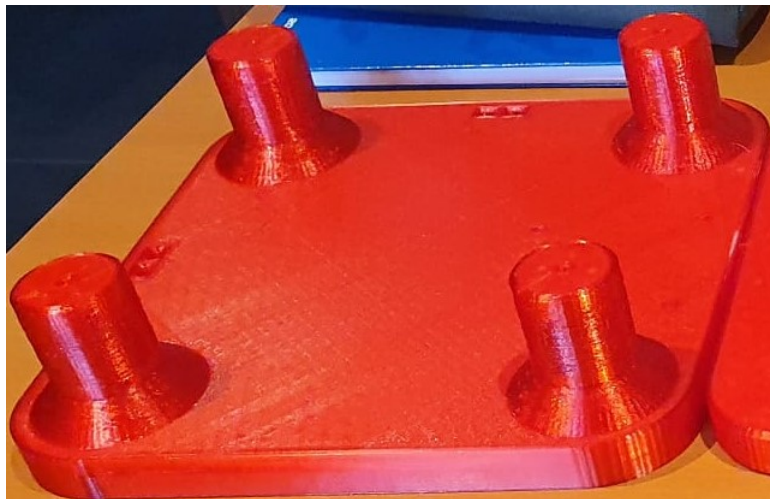


Figura 4.10: Resultat Base Inferior 3D.

4.7 Base superior

La base superior és la peça que s'encarrega de suportar el pes de la làmina tàctil, una placa de metacrilat i una peça de subjecció d'aquestes dues peces. La base superior suporta totes aquestes peces a través d'una junta articulada de tipus Cardan que va unida a un pivot que sobresurt de la base.



Figura 4.11: Disseny CAD base superior.
Font: Programa PTC CREO.

A continuació es poden veure quines peces subjecta aquesta peça, i com ho fa:

- Subjecció superior dels servomotors: Aquesta base s'encarrega de fer la subjecció superior dels dos servomotors. El sistema és el mateix que en el cas de la base inferior invertit, però amb la peculiaritat que se li han hagut d'afegir dos túnels perquè es puguin acollir el cables dels servomotors. En el disseny també s'aprecia una guia per centrar el servomotor. El disseny es mostra en la imatge següent:

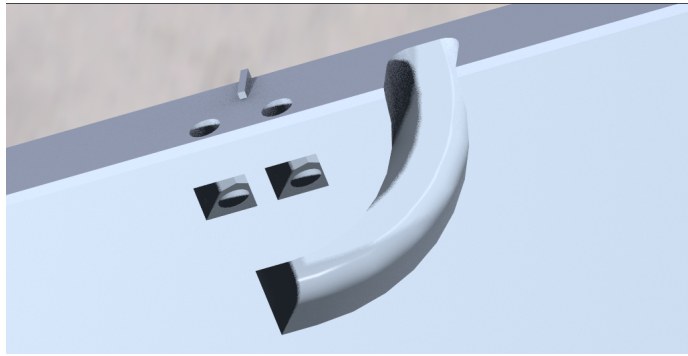


Figura 4.12: Unió Servo, part superior.
Font: Programa PTC CREO.

- L'altra subjecció que s'ha de destacar és el pivot que s'uneix amb la Cardan. Aquest pivot és de 9,7mm de diàmetre, ja que la Cardan és de 10mm, i es volia tindre tolerància. Aquest pivot té un orifici central perpendicular a l'eix, ja que la unió d'aquesta Cardan és realitzar mitjançant passadors. En la següent imatge es pot veure el detall del pivot:

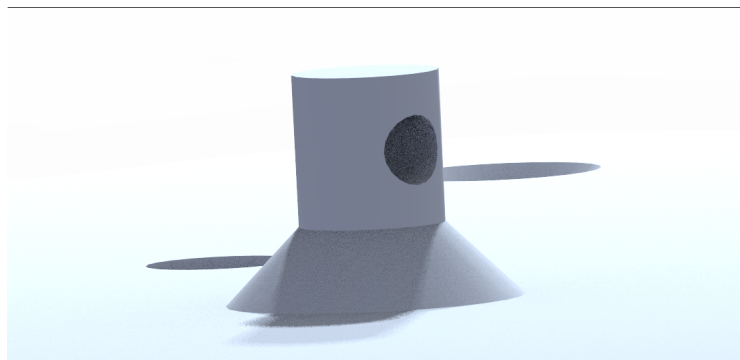


Figura 4.13: Pivot d'unió amb la Cardan.
Font: Programa PTC CREO.

- En aquesta base superior també s'han fet els forats i els avellanats perquè es pugui carregar a la base inferior, com es pot veure a la imatge general, 4.11.
- Finalment també es van afegir forats per a poder muntar la placa d'Arduino tot i estar muntada la base superior. Aquests forats també es poden apreciar a la imatge general, 4.11.

Igual que la peça anterior, aquesta peça va ser impresa en 3D pel seu grau de complexitat en la seva mecanització.



Figura 4.14: Base superior impresa en 3D.

4.8 Suport metacrilat

Aquesta és potser la peça més complexa del treball. Aquesta peça s'encarrega de fer la unió entre una làmina de metacrilat (unida a la làmina tàctil resistiva) i la junta Cardan per una banda. Per altra banda aquesta peça també s'uneix amb les barres que mouen els servomotors, cosa significa que aquesta peça es mou juntament amb la làmina tàctil.

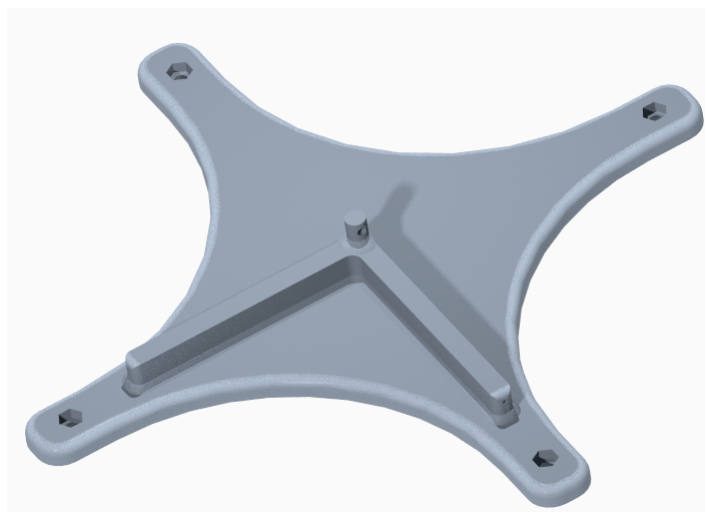


Figura 4.15: Suport metacrilat.
Font: Programa PTC CREO.

Aquesta peça és la responsable de la unió amb les següents peces:

- Cardan: Per una banda, té un pivot igual al de la base superior però amb l'orifici rotat 90 graus en l'eix del cilindre, ha de poder unir-se amb la Cardan. En la següent imatge es pot veure el detall:

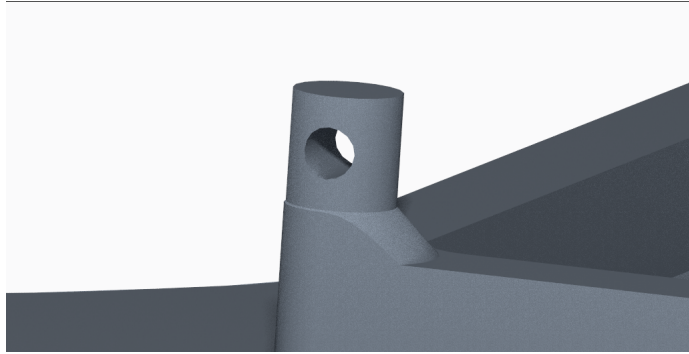


Figura 4.16: Pivot d'unió amb la Cardan.
Font: Programa PTC CREO.

- Un altra de les unions importants que fa aquesta peça és la unió amb el metacrilat. Aquesta unió és una unió cargolada amb cargols de M6, que es cargolen a unes femelles col·locades en unes cavitats a mida, aquest detall es pot observar clarament a la imatge de la peça general.
- Finalment, la unió que té aquesta peça és una unió amb les barres que transmeten el moviment dels servomotors a la plataforma. Aquesta unió es realitza mitjançant cargols de M2 cargolats directament sobre el plàstic. La unió està situada als extrems de l'extrusió en forma de L que es pot veure en la següent imatge, 4.17:

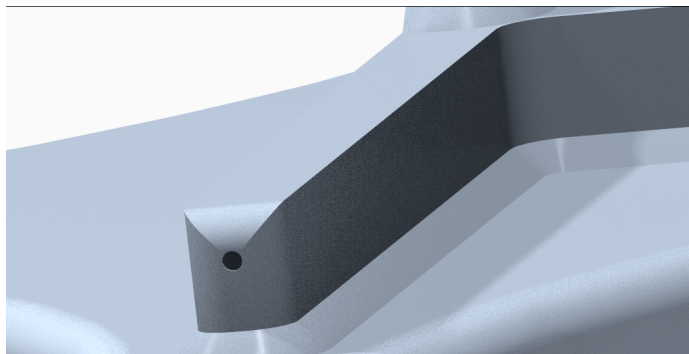


Figura 4.17: Unió de la plataforma amb les barres.
Font: Programa PTC CREO.

Com s'ha comentat anteriorment, aquesta és potser la peça més complexa de mecanitzar, per tant, igual que es va fer amb les dues peces anteriors, es va imprimir amb una impressora 3D.



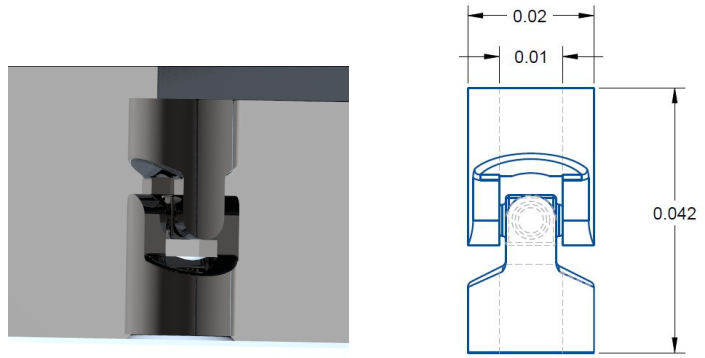
Figura 4.18: Suport de metacrilat imprès en 3D.

4.9 Junta Cardan

Una junta Cardan és una peça que originalment es fabrica per a transmetre el moviment entre dos eixos que tenen els angles de rotació desviats uns certs graus. Però, en aquest projecte es va agafar aquesta propietat de transmetre el moviment precisament per restringir-lo. Aquesta peça s'encarrega de donar llibertat de moviment als eixos X i Y de la plataforma, restringint l'eix Z, d'aquesta forma aquest sistema queda com un sistema de 2 DoF.

Aquesta peça va muntada entre el pivot de la base superior i el pivot del suport de la plataforma.

Com que aquesta peça va ser comprada a través d'Internet, es va adaptar el disseny de les peces que anaven a ser impreses en 3D perquè es pogués muntar correctament. La Cardan que es va seleccionar pot transmetre una força de rotació de 45Nm, i està fabricada en aleació d'acer. La nostra plataforma amb la bola, màxim tindrem una massa de 2,8kg comptant amb la bola, per tant, junta suporta el pes. Té un diàmetre exterior de 20mm, un diàmetre interior de 10mm, i una llargada total de 42mm.



(a) Junta Cardan entre la base superior i el suport de la plataforma.

(b) Mides de la Cardan.

Figura 4.19: Junta Cardan.
Font: Programa PTC CREO.

A continuació es mostra una imatge de la junta Cardan real:



Figura 4.20: Cardan real.

Font https://www.amazon.es/gp/product/B07KPHL51L/ref=ppx_yo_dt_b_asin_title_o05_s00?ie=UTF8&psc=1

4.10 Barres de connexió amb els Servomotors.

Aquestes barres són les encarregades de transformar el moviment de rotació del braç del servomotor en un moviment de rotació de la plataforma a través de la unió al suport del metacrilat.

Aquestes es poden ajustar en llargada, amb una llargada mínima d'uns 65 mm i de màxim, uns 85mm. Cada extrem de la barra té incorporada una articulació esfèrica que proporciona la mobilitat necessària per adaptar-se a l'orientació dels dos eixos als quals està unida (l'eix del braç del servomotor, i el forat on es cargolarà al suport de la placa de metacrilat). Aquestes articulacions tenen un forat al centre de l'esfera de 3mm de diàmetre, per on passarà el cargol per cargolar la barra al suport corresponent.

A continuació es mostren els dissenys de les barres que es fan servir en el disseny CAD:

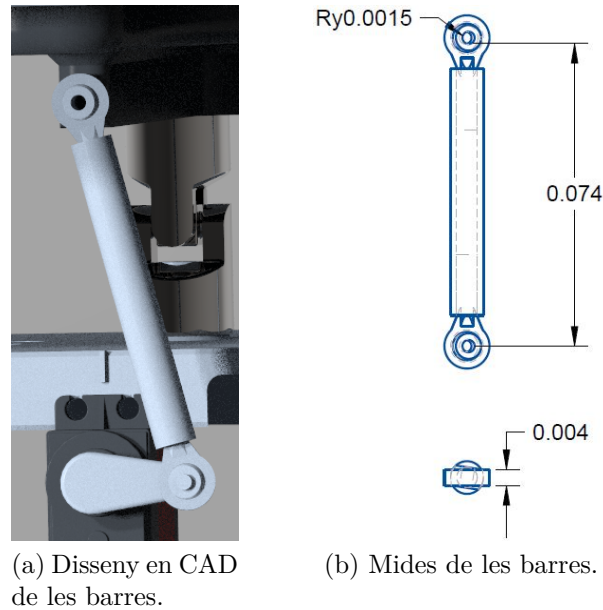


Figura 4.21: Disseny de les barres de connexió entre servomotor i plataforma.
Font: Programa PTC CREO.

Aquestes barres van ser comprades a través d'Internet, en la següent imatge es pot veure les barres reals que es van comprar:



Figura 4.22: Barres de connexió reals.

Font https://www.amazon.es/gp/product/B07QDYZZVN/ref=ppx_yo_dt_b_asin_title_o09_s00?ie=UTF8&psc=1

4.11 Placa de metacrilat

Aquesta peça és una peça que fa de suport de la làmina tàctil resistiva, és més, aquesta làmina tàctil fa de plataforma. Aquesta placa es col·loca de suport per diverses raons:

- Proporcionar a la làmina tàctil d'un gran suport pla i resistent, cosa que provocarà que el conjunt plataforma sigui més resistent a deformacions elàstiques, cops, etc, per part de la bola.
- És la unió amb el suport de la plataforma. El metacrilat és un tipus de plàstic que es pot mecanitzar sense gaires dificultats.
- Degut al seu grau de transferència, dona un caràcter més estètic al conjunt.

La placa de metacrilat que es va escollir és una placa de metacrilat de 400x400x8 mm, ja que la làmina tàctil resistiva és de 323x396 mm. A continuació es mostra el disseny de la placa de metacrilat que es va fer per a la part del disseny CAD:

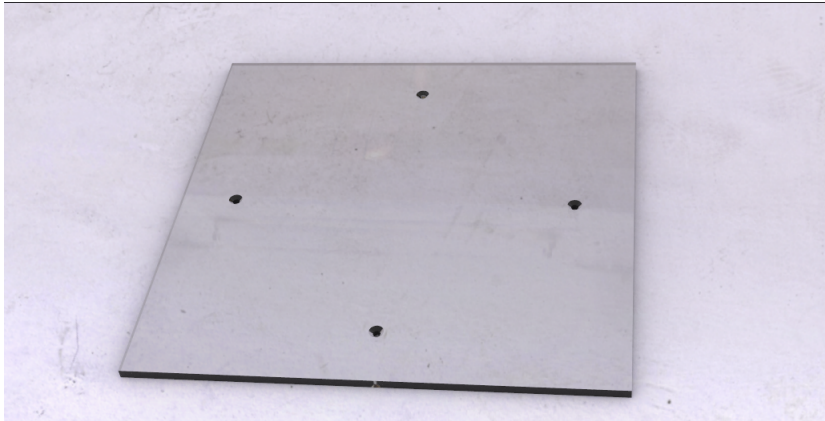


Figura 4.23: Disseny CAD de la placa de metacrilat.
Font: Programa PTC CREO.

Finalment aquesta última imatge mostra el muntatge de la placa de metacrilat sobre el suport i amb la làmina tàctil resistiva muntada al damunt:

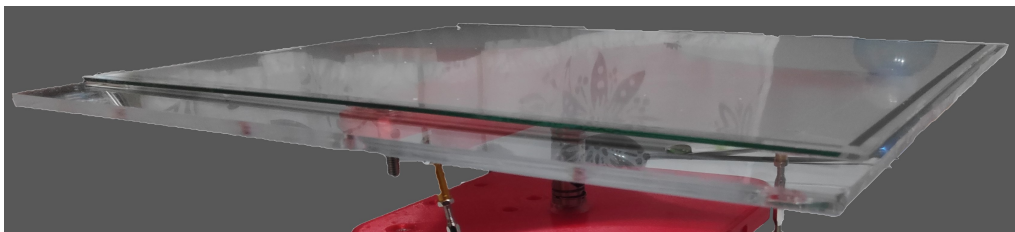


Figura 4.24: Placa de metacrilat real.

4.12 Esfera

Per finalitzar amb aquest apartat sobre el Disseny Mecatronic es mostra una peça que té la peculiaritat que simplement es col·loca de forma lliure en el sistema, però que és clau perquè el sistema funcioni. Aquesta peça és la bola que va sobre la plataforma. És la bola que el nostre sistema ha de controlar sobre de la làmina tàctil resistiva.

Les úniques restriccions que hi ha per a escollir aquesta peça són, que ha d'estar entre els 20g i 1 kg, que és la massa que s'ha calculat que podrà suportar el motor en l'extrem de la plataforma. El pes i mida de l'esfera es tenen en compte als càlculs per realitzar el control. També s'ha de procurar que l'esfera tingui un cert grau de tolerància geomètrica.

La bola es va haver d'afegir també al disseny CAD per poder fer posteriorment la simulació de tot el sistema. La bola del disseny CAD és de 50 mm de diàmetre:

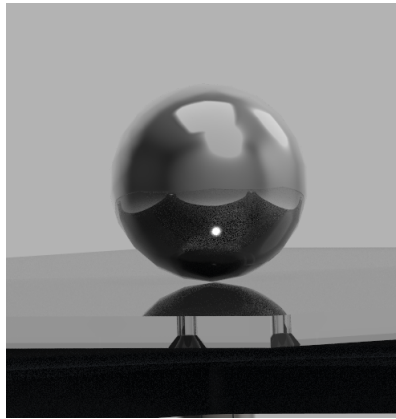


Figura 4.25: Disseny CAD de la bola.
Font: Programa PTC CREO.

Capítol 5

Programació del Sistema

En aquest apartat s'explica com es dur a terme la programació d'aquest sistema per a controlar el sistema real i també, com es programen cada un dels elements que el formen.

El microcontrolador que s'ha fet servir per a controlar aquest sistema és l'Arduino Due que, com passa en tots els Arduino, el llenguatge que fa servir no és un llenguatge gaire eficient, ja que està pensat per poder compilar-lo en nombrosos microcontroladors. Però té l'avantatge que és un llenguatge de programació bastant entenedor, ja que la majoria d'accions en fan mitjançant funcions.

5.1 Làmina tàctil resistiva

La programació de la làmina tàctil per poder reconèixer la posició de la bola es basa en un bucle a on es llegeix la posició de la bola cada cop. Primer, es llegeix la posició de la bola en l'eix X i després, es llegeix la posició de la bola en l'eix Y. En la figura de l'apèndix E, figura D.3 es mostra el diagrama de flux per poder programar la làmina tàctil resistiva i obtenir així la posició de la bola. Es pot observar el programa que es fa servir per a la làmina tàctil en l'annex H.

5.2 Servomotors

Per que els servomotors assoleixin una posició desitjada se'ls hi ha d'enviar un senyal PWM d'un determinat Duty cycle. S'ha de fer un programa que en funció de l'angle, generi el senyal PWM adequat. Arduino disposa d'una llibreria que precisament fa aquesta funció, per tant, a l'hora de realitzar la programació dels servomotors s'ha fet servir la seva llibreria i les seves funcions. Aquest senyal PWM passa pel circuit amplificador abans d'anar als servomotors. Es pot observar el programa que es fa servir per programar els servomotors en l'annex G.

5.3 Teclat matricial

El teclat matricial és una disposició de botons en forma matricial que comparteixen els conductors de cada fila i cada columna amb els polsadors de la mateixa fila o columna. Per programar el teclat s'hauria de fer un escombrat per fila i després per columna per saber quin polsador s'ha pres. Com en el cas de la programació dels

servomotors, es fa servir una llibreria que facilita molt la feina, ja que gràcies a les seves funcions, permet saber quin botó s'ha pres amb un nombre de línies de codi molt més baix. Es pot observar el programa que es fa servir per programar el teclat matricial en l'annex I.

5.4 Programa del sistema

A continuació s'exposa un pseudocodi que representa al codi que s'ha realitzat a la placa d'Arduino:

```
Es defineixen les variables;
Es defineixen els pins d'entrada i sortida;
Es defineix una posició inicial a la que haurà d'anar la bola (X=0, Y=0);
while Teclat Matricial != '*' do
  | El programa s'atura.
end
while (1) do
  | Comprovació d'entrada al teclat matricial;
  | if Si s'ha premut algun polsador del teclat Matricial. then
  |   | S'actualitzen els valors de posició de la bola (X, Y);
  | end
  | Es llegeixen els sensor (s'obté la velocitat i la posició, tant de la bola
  |   com dels servomotors);
  | Es comprova el temps del bucle;
  | if (Temps del bucle >= 7ms) then
  |   | S'actualitza el control per donar una nova sortida als servomotors en
  |     funció dels valors dels sensors;
  | end
end
```

Algorithm 1: Programa principal del *Ball on Plate*

El codi d'Arduino complert es pot trobar a l'annex J.

Capítol 6

Resultats

En el present treball es van definir uns objectius a complir, els quals s'analitzen en aquest capítol per determinar si s'han assolit:

- Disseny i construcció de la plataforma amb 2 graus de llibertat: en fer servir la metodologia de Disseny Mecatronic, s'ha efectuat un disseny on els seus components, tant elèctrics com mecànics, estan molt adaptats entre si. Amb aquesta metodologia s'ha adaptat el disseny CAD i s'han comprat uns servomotors adequats perquè suportessin el pes de la plataforma. Els servomotors també s'encarreguen del moviment de la plataforma en els eixos X i Y. La junta Cardan permet restringir la rotació de la plataforma en el seu eix perpendicular, quedant un sistema de 2 DoF. El disseny CAD també s'ha adaptat per allotjar als dos servomotors, el microcontrolador, la junta Cardan i per poder subjectar la plataforma. Seguint amb el Disseny Mecatronic també s'ha hagut d'adaptar l'alimentació dels dos servomotors, ja que aquests necessiten més de 4,8V d'alimentació, i l'Arduino només pot subministrar-ne 3.3V. En aquest treball s'ha pogut veure el disseny final que es va fer amb CAD en la imatge 4.7. La següent imatge mostra el disseny ja muntat:



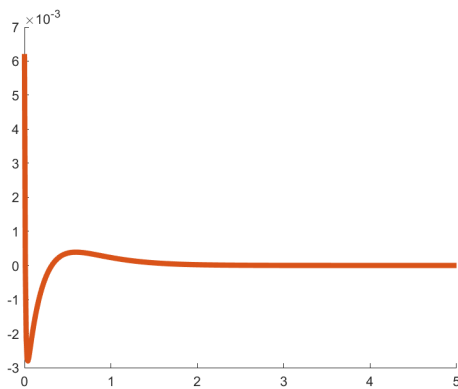
Figura 6.1: Teclat matricial indicat les seves connexions.

Al realitzar el muntatge de la Cardan es va tindre que afegir un reforç de silicona per evitar l'excés de joc respecte al pivot de la base superior i al del suport de la plataforma. Per tant es considera que s'ha assolit aquest objectiu.

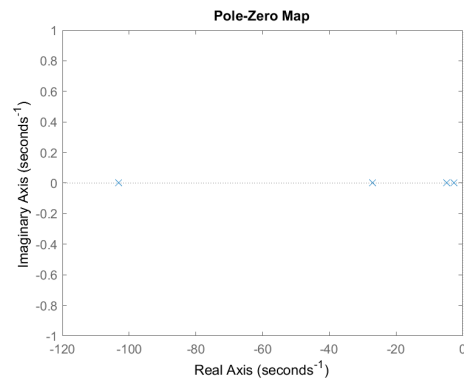
- Realitzar la simulació del sistema de *Ball on Plate* amb l'eina Matlab. La part de simulació ha estat molt important en la realització del control d'aquest sistema, ja que gràcies a aquesta simulació, s'ha pogut comprovar que el sistema

es podia controlar. S'ha pogut preveure com reaccionaria el sistema i si hi havien errors en el control abans d'implementar-ho en el sistema real.

- Comprovació Matlab: Amb aquesta comprovació s'ha pogut veure que el sistema de *Ball on Plate* dissenyat és controlable i observable. S'ha pogut observar la reacció que tenia el sistema davant d'una pertorbació i s'ha calculat el Lloc Geometric de les Arrels. Amb aquesta informació s'ha pogut ajustar aquesta reacció del sistema gràcies als guanys afegits. La millor reacció que ha tingut el sistema s'ha aconseguit amb les següents representacions:



(a) Gràfic Estat-Temps escollit.

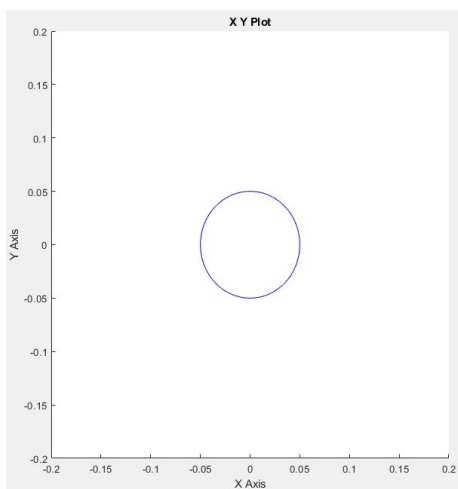


(b) Lloc Geometric de les Arrels escollit.

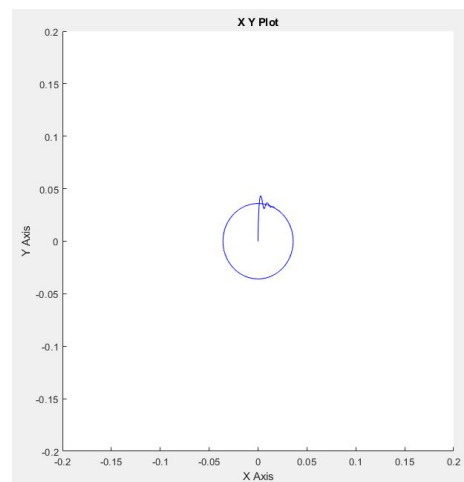
Figura 6.2: Representació gràfica del comportament del control.

Font: Programa PTC CREO.

- Un cop aconseguides les representacions desitjades, s'extreu la matriu de control LQR, es simula amb el model de Simulink i es comprova el resultat:



(a) Trajectòria definida.



(b) Resultat de la Simulació del control.

Figura 6.3: Resultat del control amb Simulink

Font: Programa PTC CREO.

Aquesta simulació havia estat modificada amb l'aplicació Control System Tuner del Matlab. Es pot veure a les imatges anteriors com el sistema segueix la trajectòria definida a l'entrada correctament.

- Finalment es comprova el resultat de la simulació en el Multibody, que ens permet observar la reacció del disseny CAD amb els paràmetres de control del Simulink. El resultat d'aquesta simulació també és correcte. Ja només quedaria implementar els paràmetres de control al sistema real i comprovar el funcionament.

Crec que aquest objectiu s'ha assolit, ja que s'ha aconseguit modelitzar el sistema i simular-lo amb el control que es faria servir.

- Un cop es finalitza la simulació amb el Matlab, es procedeix a ficar els paràmetres de la matriu de control del LQR al programa d'Arduino perquè es pugui controlar el sistema real. Quan s'afegeix la matriu de control al sistema real, aquest mai es comporta igual a la simulació, això és perquè, per molt acurada que sigui la simulació, la realitat sempre varia degut a paràmetres que no es poden controlar. Tot i això, la simulació ens acosta molt al control desitjat i, per tant, l'única cosa que hem de fer, és realitzar petits ajustos als paràmetres per aconseguir la reacció desitjada. Aquest objectiu crec que s'ha aconseguit. Si l'adquisició de dades de la làmina tàctil resistiva pogués ser més ràpida i els servomotors també actuessin amb més velocitat i precisió, el control hagués millorat. Tot i això s'ha assolit l'objectiu. Es pot observar el control del sistema real en el següent enllaç: <https://www.youtube.com/watch?v=P-cminIHnYw>

Capítol 7

Conclusions

Durant la realització d'aquest projecte he pogut endinsar-me en el món del control automàtic, cosa que em feia bastant por, ja que trobo que és un branca d'estudi molt abstracte però que pot fer coses increïbles. Realitzant aquest projecte he après molt sobre aquest camp d'estudi, però també m'he adonat del grau de dificultat que té realitzar un bon sistema de control per a qualsevol sistema. Ara cada cop que observo un aparell que incorpora un sistema de control, puc imaginar-me tota la gent i hores de treball que hi ha al darrere perquè aquell sistema tingui un funcionament adequat i sobretot, segur.

Els sistemes de control necessiten una gran quantitat de potència computacional i també un model matemàtic del sistema suficientment acurat perquè funcioni adequadament. En realitzar el present treball m'he adonat de la importància que tenen tots els elements en el control. Per exemple, en el cas d'aquest projecte, un dels elements que han fet que el control no fos tan bo com s'hagués desitjat ha estat la làmina tàctil resistiva que, el temps d'adquisició de dades és molt lent i això perjudica molt al control. Per altra banda, els servomotors utilitzats no són gaire robustos i potser s'haurien d'escollir uns amb una velocitat major.

En realitzar aquest treball he pogut millorar les meves competències d'organització, metodologia, entre altres competències que crec que m'ajudaran molt en el meu futur com a professional, i en l'àmbit personal.

Per acabar, jo estic bastant satisfet del treball realitzat, tot i haver assolit els objectius, m'hauria agradat poder avançar més, ja que el projecte sempre es podrà millorar, en uns aspecte o en uns altres.

Bibliografia

- Arduino Due, (Data d'accés: 2020-03-13). URL <https://store.arduino.cc/arduino-due>.
- Fer servir un teclat Matricial en Arduino, (Data d'accés: 2020-05-05). URL <https://www.luisllamas.es/arduino-teclado-matricial/>.
- G. Andrews, C. Colasuonno, and A. Herrmann. Ball on plate balancing system progress report for ecse-4962 control systems design. 2004.
- O. H. Durán, I. O. Leal, E. A. G. Vidal, J. O. T. Salgado, R. V. Segura, C. C. Casillo, J. P. O. Oliver, et al. Diseño, construcción y control de una plataforma bola-plano. *Pàdi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, 7 (Especial):88–95, 2019.
- Y. Garzón Melo, O. L. Garzón Mancera, et al. Diseño e implementación de una plataforma bola y plato, para la aplicación de técnicas de control pid y lqr. 2016.
- A. Kassem, H. Haddad, and C. ALBITAR. Comparison between different methods of control of ball and plate system with 6dof stewart platform. *IFAC-PapersOnLine*, 48(11):47–52, 2015.
- J. D. Martínez Velasco, L. T. Poveda Galvis, et al. Diseño e implementación de un control óptimo lqr con la tarjeta raspberry pi.
- J. J. Penco and M. R. Modesti. Control difuso versus pid para un sistema de bola y plataforma. *Revista Tecnología y Ciencia*, (36):155–168, 2019.
- L. C. Rodríguez Cifuentes. Diseño de un sistema estabilizador de una bola sobre una plataforma móvil mediante microcontrolador. B.S. thesis, Universitat Politècnica de Catalunya, 2018.
- L. Spacek, J. Vojtesek, F. Gazdos, and T. Kadavy. Ball & plate model for robotic system. In *ECMS*, pages 226–231, 2018.

Apèndix A

Simulink

A.1 Resultat de l'aplicació Control Tuner System

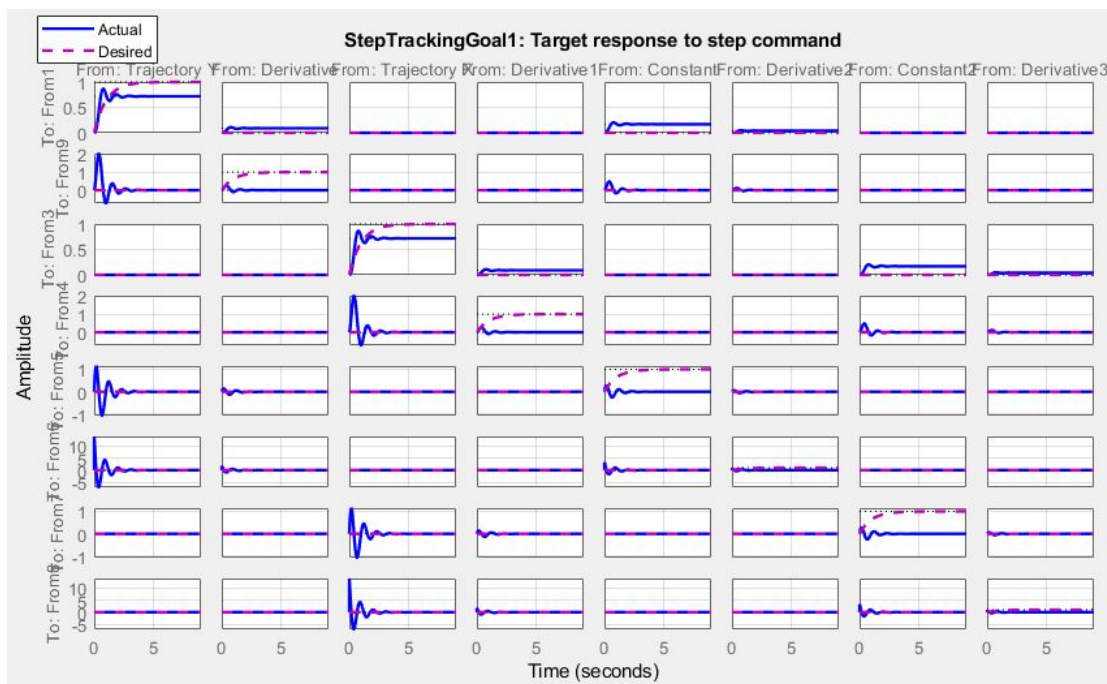


Figura A.1: Resultat de l'aplicació Control System Tuner a cada senyal.
Font: Matlab

Apèndix B

Característiques de l'Arduino Due

B.1 Característiques de l'Arduino DUE

En el present treball s'ha explicat les raons d'escollir aquesta placa per a la nostra aplicació i algunes peculiaritats que té.

A continuació s'exposen les característiques de la placa que han estat extretes directament de la Web oficial d'Arduino:

- Vin: És el pin pel qual podem subministrar la tensió a la placa fent servir una font d'energia externa. Es pot subministrar energia per aquest pin, o bé pel connector jack (7-12V), que accedeix al mateix pin. També es pot alimentar a través del USB (5V). No és aconsellable connectar la placa a través dels pins 5V o 3.3V, saltant-se el regulador.
- 5V: Per aquest pin surt una tensió regulada de sortida de 5V de la placa.
- 3V3: Per aquest pin surt una tensió regulada de sortida de 3.3V, que suporta un corrent màxim de 800mA, aquest regulador també és el que proveeix la potència al microcontrolador.
- : GND: Pins de terra.
- IOREF: Aquest pin proveeix un voltatge de referència amb el que opera el microcontrolador.
- Memoria: El SAM3X té 512KB (2 blocs de 256 KB) de memòria flash per guardar el codi. La SRAM disponible és de 96 KB en dos parts contigües de 64KB i 32 KB.
- Digital I/O pins de 0 fins a 53: Tots es poden fer servir com a Inputs o Outputs. Funcionen a 3.3V. Depenent del pin, poden proporcionar 3 o 15 mA, i rebre 6 o 9 mA. Tenen incorporat una resistència pull-up de 100 kOhms. A més alguns pins tenen funcions especials:
 - Serial: 0(RX) i 1(TX)
 - Serial 1: 19(RX) i 18(TX)
 - Serial 2: 17(RX) i 16(TX)
 - Serial 3: 15(RX) i 14(TX). Tots els Serial funcionen a 3.3V.

- PWM: del pin 2 al 13: Són sortides PWM de 8 bits, la resolució d'aquests pins es pot canviar gràcies a la funció `analogWriteResolution()`.
 - SPI: Aquests pins suporten la comunicació SPI, nomès es poden fer servir per comunicar-se amb altres dispositius que suportin aquests tipus de comunicació.
 - CAN: CANRX i CANTX: Aquests pins suporten el protocol de comunicació CAN pero aquest protocol no es suportat per l'API de l'Arduino.
 - LED: El pin 13 està connectat a un LED de la pròpia placa.
 - TWI 1: 20 (SDA) i 21 (SCL).
 - TWI 2: SDA1 i SCL1. Aquests pins i els anteriors suporten la comunicació TWI fent servir una llibreria d'arduino.
- Entrades analògiques: dels pins A0 fins al A11. Cadascún d'aquests pins té una resolució de 12 bits (4096 valors diferents). Per defecte, la resolució de lectura està configurada a 10 bits, per la compatibilitat entre arduinos. Es possible canviar aquesta resolució mitjançant la funció `withanalogReadResolution()`. Poden mesurar de 0V fins a un màxim de 3.3V.
 - DAC1 i DAC2: Aquests pins proveeixen de dos sortides analògiques a la placa amb 12 bits de resolució. Aquests pins es poden fer servir com a sortida de audio. Funcionen de 0.55V fins a 2.75V

Apèndix C

Servomotor

C.1 servomotor

Un servomotor és un tipus de motor elèctric al que li podem controlar la seva posició i velocitat. Quan parlem de servomotor en realitat ens referim a un sistema compost d'una part electromecànica i un control electrònic.

Hi han molts tipus de Servomotors diferents, i per a usos molt diferents, aquests tipus de motors es fan servir tan a la indústria, en màquines CNC, robots, etc. com en petits projectes electrònics o modelisme. Com es pot imaginar, aquests varien molt de preu segons l'aplicació. La següent imatge mostra les parts d'un servomotor com el que es fa servir en aquest projecte, en aquest cas és un motor per a modelisme.

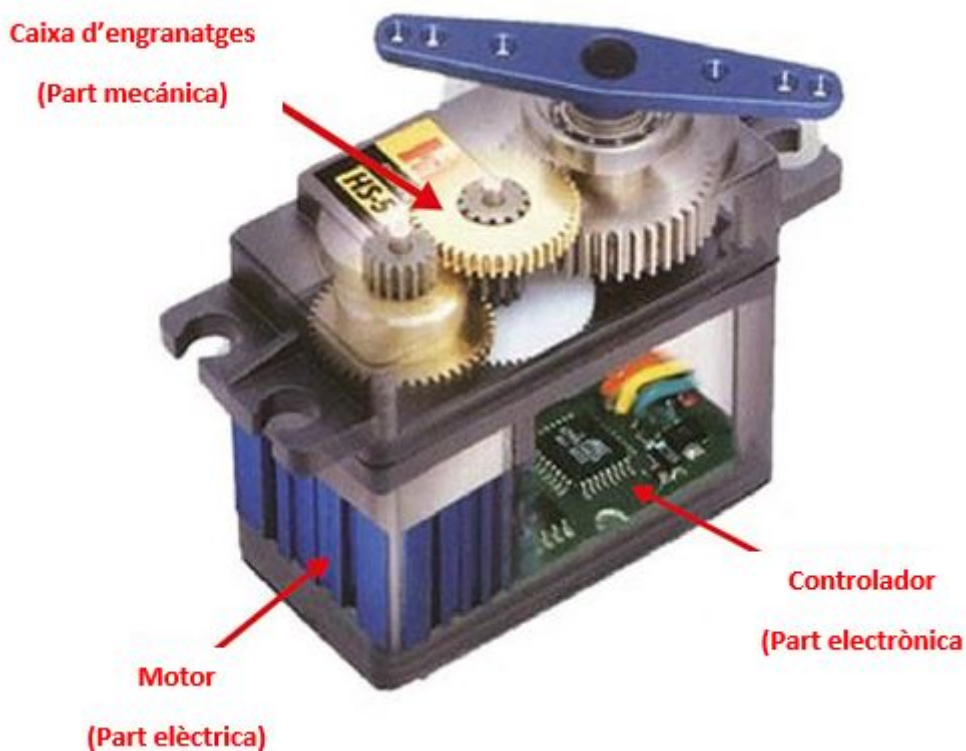


Figura C.1: Parts d'un servomotor.

Font : [http:](http://panamahitek.com/que-es-y-como-funciona-un-servomotor/)

[//panamahitek.com/que-es-y-como-funciona-un-servomotor/](http://panamahitek.com/que-es-y-como-funciona-un-servomotor/)

Com és pot observar en la imatge anterior, les parts d'un servomotor es poden separar en tres grups principals:

- Part elèctrica: Motor elèctric DC, es el responsable de generar el moviment al sistema a través del seu eix.
- Part Mecànica: La part mecànica d'un servomotor es compon d'un tren d'engranatges que es mogut pel motor i que provoca un augment del parell i una reducció de la velocitat del motor a la sortida.
- Part electrònica: La part electrònica és la part que s'encarrega de detectar la posició del servomotor en tot moment (ja sigui amb un potenciòmetre o bé amb un encoder), de tractar el senyal PWM que prové de l'aparell que controla el servo, i que indica a quina posició vol que és col·loqui el servomotor, i de traduir aquest senyal a moviment del motor DC. En la següent imatge es pot veure el diagrama de blocs del control d'un servomotor:

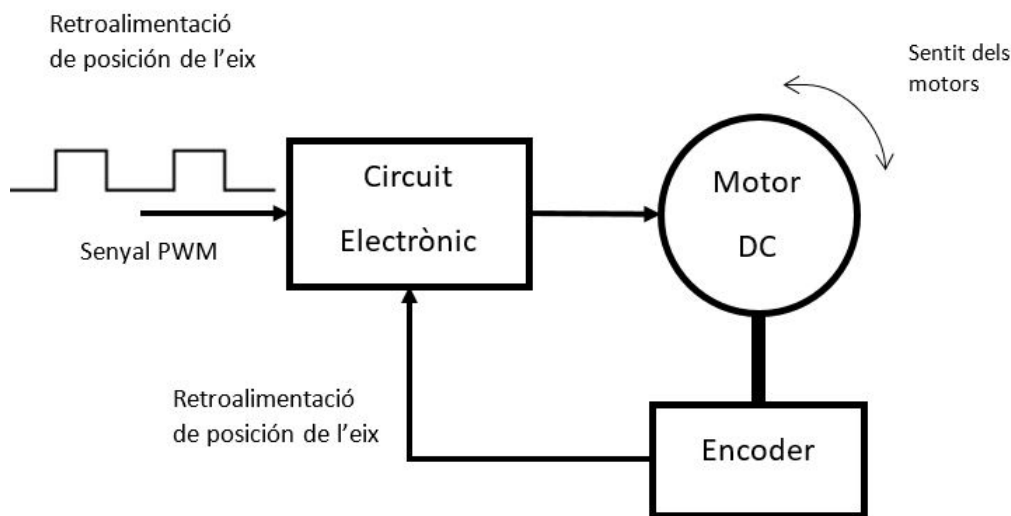


Figura C.2: Diagrama de blocs d'un servomotor.

Font: <https://panamahitek.com/que-es-y-como-funciona-un-servomotor/>

Com he comentat abans, un servomotor es controla a través d'un senyal de tipus PWM, que no deixa de ser una senyal quadrada d'una determinada freqüència en la que es fa variar el temps del senyal a la màxima tensió, que provoca la variació del senyal a la mínima tensió, en altres paraules, variant el Duty cycle com es veu a la següent imatge:

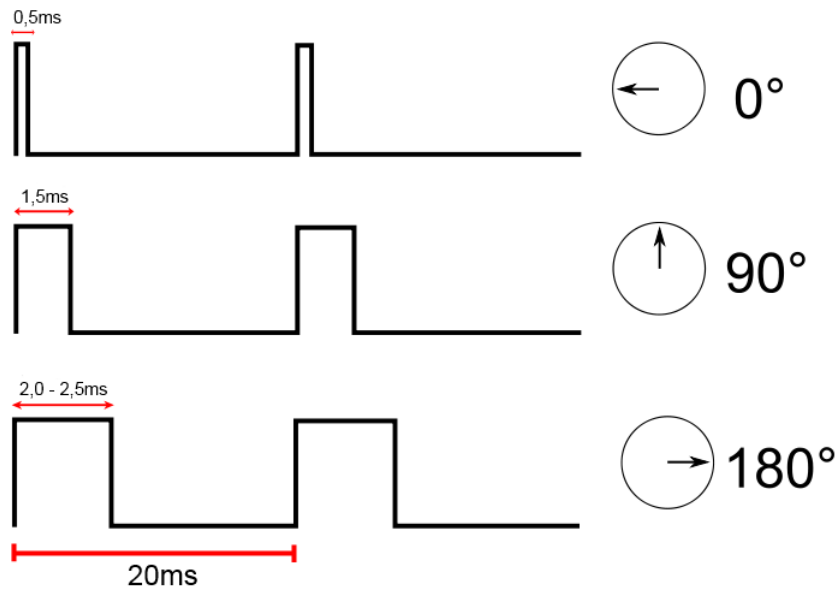


Figura C.3: Control PWM d'un servomotor.

Font: <https://upload.wikimedia.org/wikipedia/commons/thumb/f/f6/TiemposServo.svg/800px-TiemposServo.svg.png>

El motor escollit per a la nostra aplicació és un MG996r, que té les següents especificacions:

- Massa: 55g
- Dimensions: 40,7 x 19,7 x 42,9 mm aproximadament.
- Parell (6V): 1,08 Nm.
- Velocitat d'operació(6V): 0,14s/60°.
- Voltatge de funcionament: 4,8 a 7,2V.
- Intensitat (6V): 500 mA - 900 mA.
- Rang de temperatura: 0°C-55°C.

C.1.1 Comprovació del parell

Es comprova amb una bola d'un kilogram, tot i que el pes màxim que en realitat posarem serà de molt menor per limitació de la làmina tàctil resistiva. Els subíndex m fan referència al metacrilat, el subíndex b fa referència a la bola, R_a fa referència al recolzament i finalment, el subíndex s fa referència al servomotor.

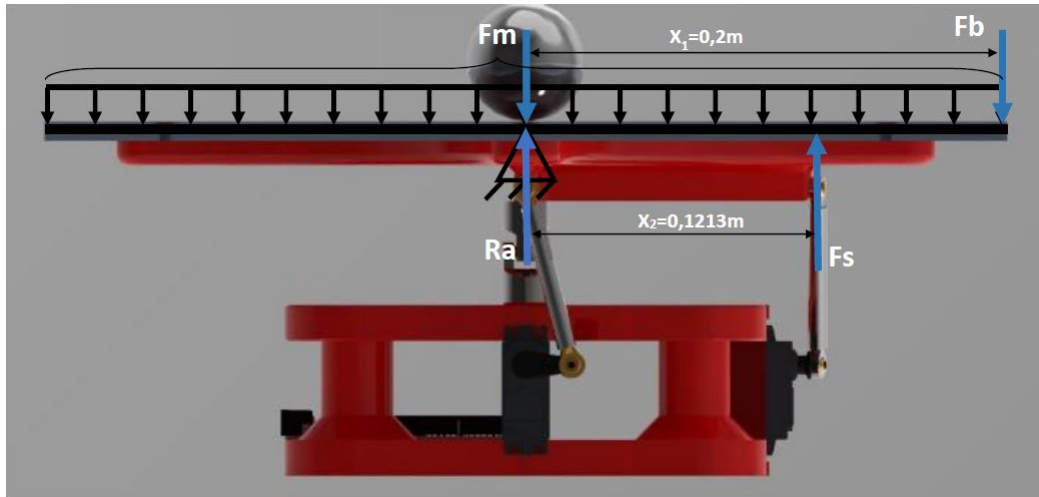


Figura C.4: Comprovació de la força que arriba al servomotor fent el símil amb una viga.

$$\begin{aligned}
 \rho_m &= 1190 \text{kg/m}^3 \\
 V_m &= 0,0013 \text{m}^3 \\
 F_m &= 1190 \text{kg/m}^3 * 0,0013 \text{m}^3 * 9,8 \text{m/s}^2 = 14,93 \text{N} \\
 F_b &= 1 \text{kg} * 9,8 \text{m/s}^2 = 9,8 \text{N}
 \end{aligned}
 \tag{C.1}$$

Forces verticals:

$$F_b + F_m = F_s + R_a
 \tag{C.2}$$

Moments en Ra:

$$\begin{aligned}
 F_b * 0,2 \text{m} &= F_s * 0,1213 \text{m} \\
 F_s &= \frac{9,8 \text{N} * 0,2 \text{m}}{0,1213 \text{m}} = 16,16 \text{N} \\
 R_a &= F_b + F_m - F_s = 8,57 \text{N}
 \end{aligned}
 \tag{C.3}$$

Aquesta és la força que es transmet per la varilla des de la plataforma fins al braça del servomotor, ara s'ha de comprovar el parell que ha d'aguantar:

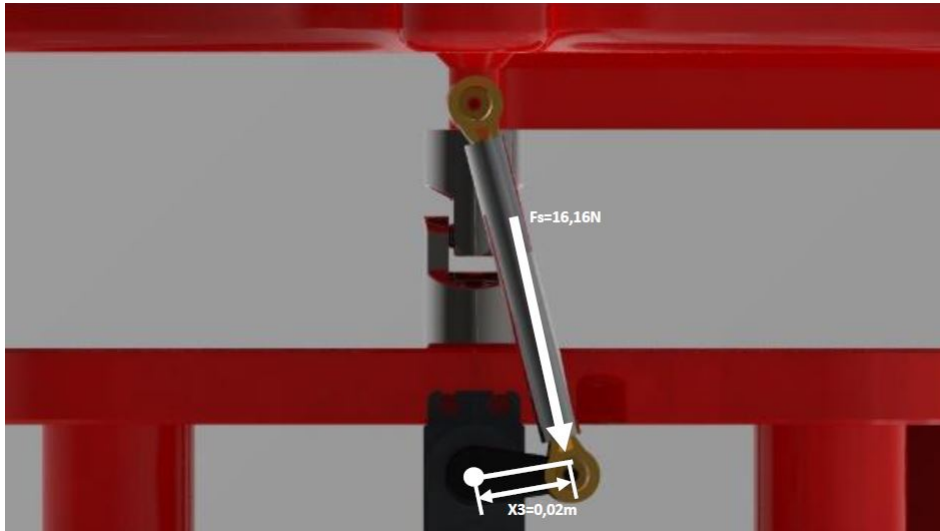


Figura C.5: Comprovació parell que ha d'aguantar el servomotor.

$$M_s = 16,16N * 0,02m = 0,323Nm \quad (C.4)$$

Com és pot observar, es 0,323 Nm és més petit que el parell màxim que pot transmetre el nostre servomotor, que són 1,08 Nm.

C.1.2 Connexió amb l'Arduino

Per realitzar la connexió amb la placa d'Arduino s'ha de fer a través de transistors, ja que la tensió de sortida dels pins PWM és de 3.3V, i l'alimentació que necessita un servomotor és d'entre 4,8 i 7,2 V, per tant, és necessari augmentar la tensió d'entrada al servomotor. Per fer aquest augment de tensió fem servir la mateixa font d'alimentació que alimenta al motor a 6V, que reproduïx el senyal PWM que prové de l'arduino per amb una tensió màxima de 6V gràcies al següent circuit:

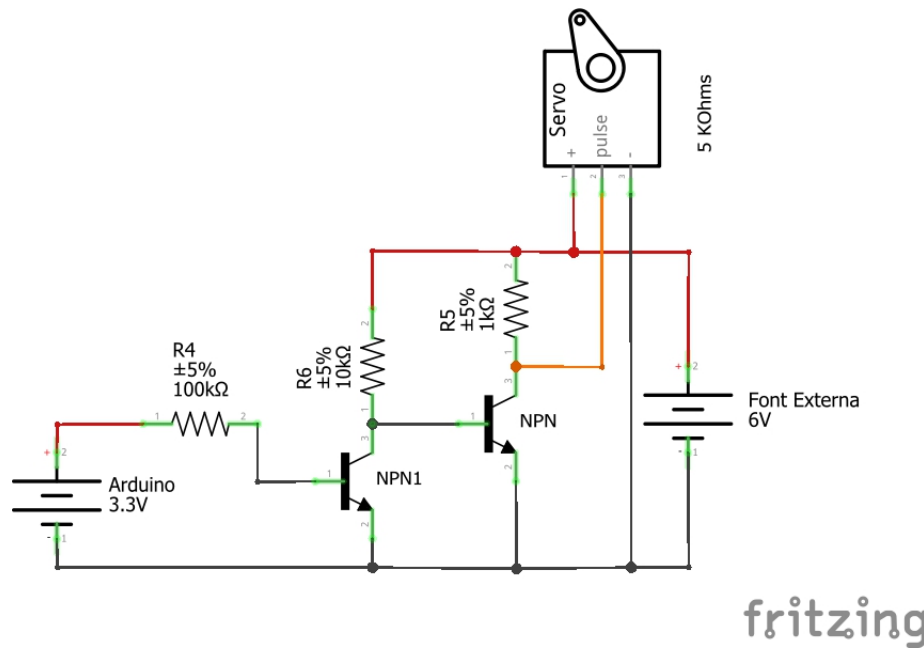


Figura C.6: Connexió dels Servos amb la font d'alimentació i a l'Arduino a través de 2 transistors npn en cascada.

Font: Fritzing program.

Aquest circuit és un circuit que reproduïx la sortida PWM de l'Arduino gràcies a dos transistors PWM connectats tal i com es veu a la imatge i, que treballen en zona de saturació. Quan la sortida del PWM és baixa (0V), el NPN1 està en tall, per tant, el NPN està conduït per la resistència R6 i la sortida serà 0V, ja que està saturat. Quan l'entrada és alta (3,3V), el transistor NPN1 entrarà en saturació, i per tant, el transistor NPN entrarà en tall i la sortida serà directament els 6V. D'aquest circuit s'ha comprovat que el corrent que passa pels transistors no excedeixi el màxim corrent que pot passar per aquests. Els transistors utilitzats són transistors 2N2222, que pot passar una corrent màxim pel col·lector de 0,8A.

$$I_1 = \frac{6V}{10000\text{ohms}} = 0,65mA$$

$$I_2 = \frac{6V}{1000} = 6,5mA$$
(C.5)

Ja és pot veure que les dues intensitats són molt menys a la màxima suportada.

Apèndix D

Làmina tàctil resistiva

D.1 Làmina tàctil resistiva

La làmina que és va escollir és una làmina resistiva de 19 polzades de 4 fils (ja que és volia una gran superfície per a que el moviment sigui molt més visual) amb les següents característiques tècniques més importants per a la nostra aplicació:

- Voltatge operatiu: $\leq 10V$
- Resposta lineal amb error ≤ 1.5
- Força d'accionament de 20 a 100 grams.
- Precisió tàctil 1.15mm a 2.5mm.
- Temps de resposta: $< 10ms$

D.1.1 Principi de funcionament

És un dispositiu construït de dos capes de material separat per uns espaiadors molt fins. Cadascuna de les capes contenen una substància resistiva, normalment composta d'un metall anomenat Indium Tin Oxide (ITO). Aquest metall es col·locat en les dues capes, la capa superior és una capa de Polietilè (PET), i la capa inferior és normalment de vidre, les dues separades pels espaiadors.

Quan es presiona la lamina superior, les dos superfícies resistives entren en contacte, la posició de la zona on es presiona és localitza gràcies a que aquestes dos superfícies estan connectades a 4 fils tal i com es veu a la imatge següent:

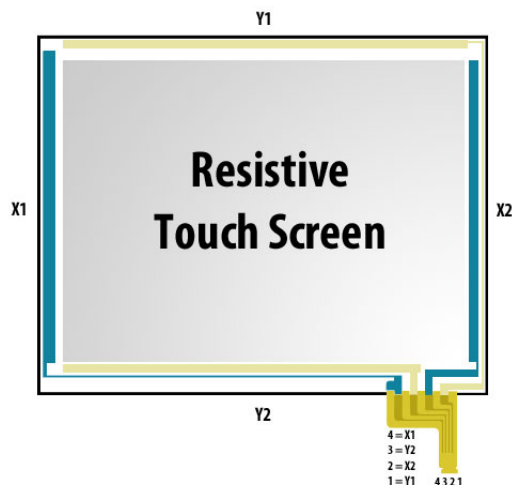


Figura D.1: Làmina tàctil resistiva.

Font: <https://www.pocketmagic.net/wp-content/uploads/2015/08/touchscreen.jpg>

Com es pot veure a la imatge anterior cada làmina crea una resistència per a cada eix (X i Y) i va connectada a dos dels 4 connectors. Per saber quins connectors corresponen a cada làmina, és tan fàcil com comprovar la seva resistència, ja que com les làmines en repòs (sense ser presionades) tenen resistència infinita. Quan la lamina és presionada, les lamines entren en contacte i actuen en conjunt com una resistència variable. Si els connectors de l'eix Y estàn connectats a positiu i negatiu, i llavors medim la tensió entre un del connectors de l'eix X i negatiu, es podrà veure com la tensió varia en funció d'on presionem la lamina tàctil, tal i com es pot apreciar en la següent imatge:

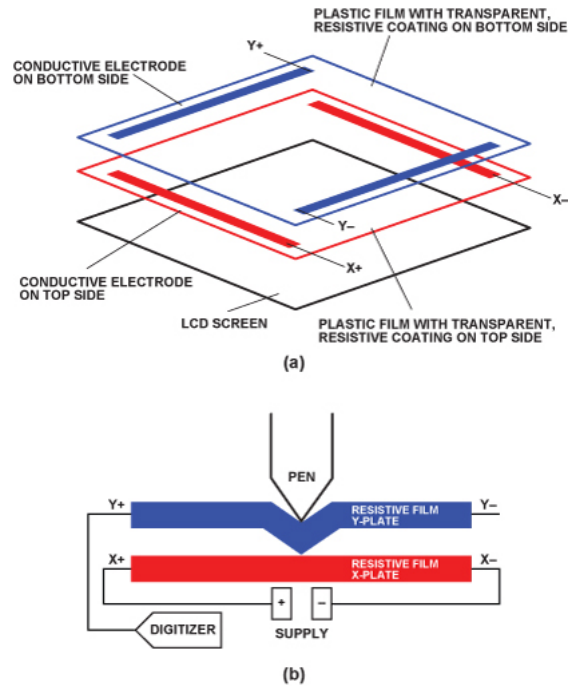


Figura D.2: Funcionament de la làmina tàtil resistiva.

Font: https://sticktech.files.wordpress.com/2011/09/ad45-06_fig_011.jpg

//sticktech.files.wordpress.com/2011/09/ad45-06_fig_011.jpg

Seguint la descripció de funcionament de la làmina tàtil resistiva, per connectar la làmina, el que s'ha de fer es connectar els 4 pins que provenen de la làmina tàtil a 4 pins analògics de l'Arduino sabent quins pins van connectats a la part X i quins van connectats a la part Y del sensor.

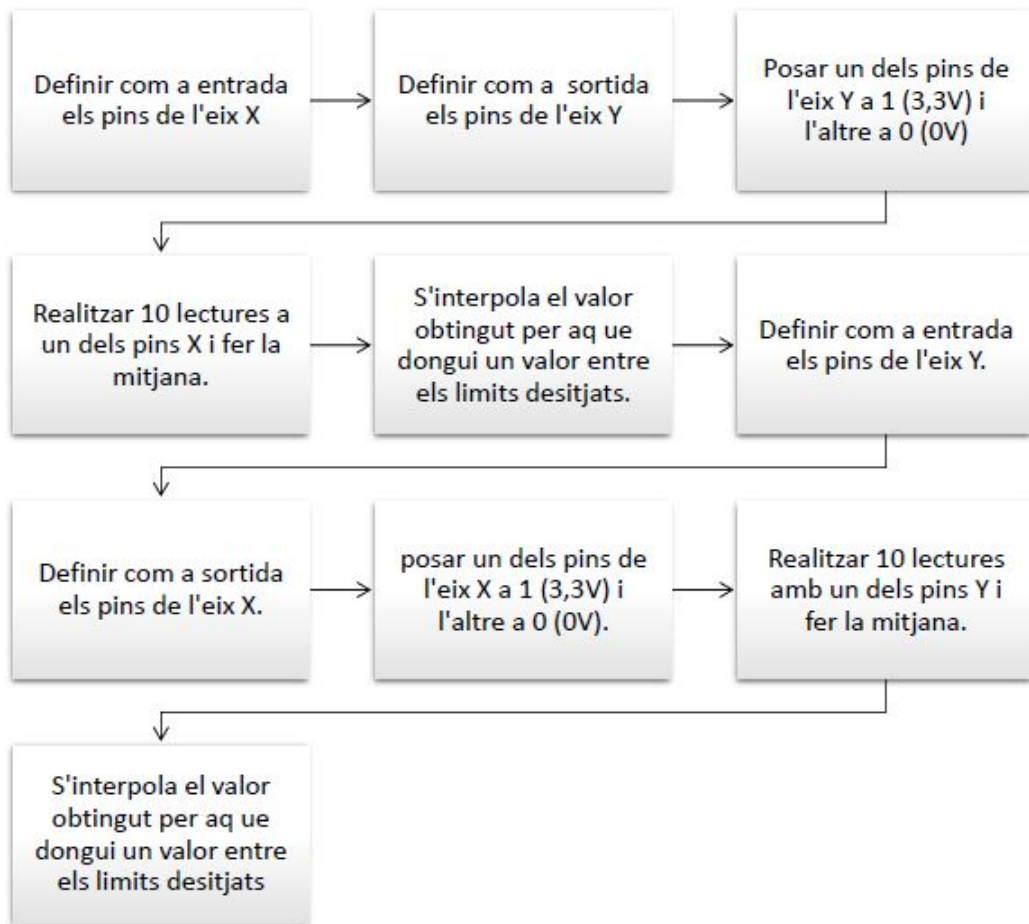


Figura D.3: Diagrama de flux d'obtenció de dades de la posició de l'esfera de la làmina tàctil resistiva.

Apèndix E

Mida peces mecàniques

E.1 Mides de les peces mecàniques

E.1.1 Base Inferior

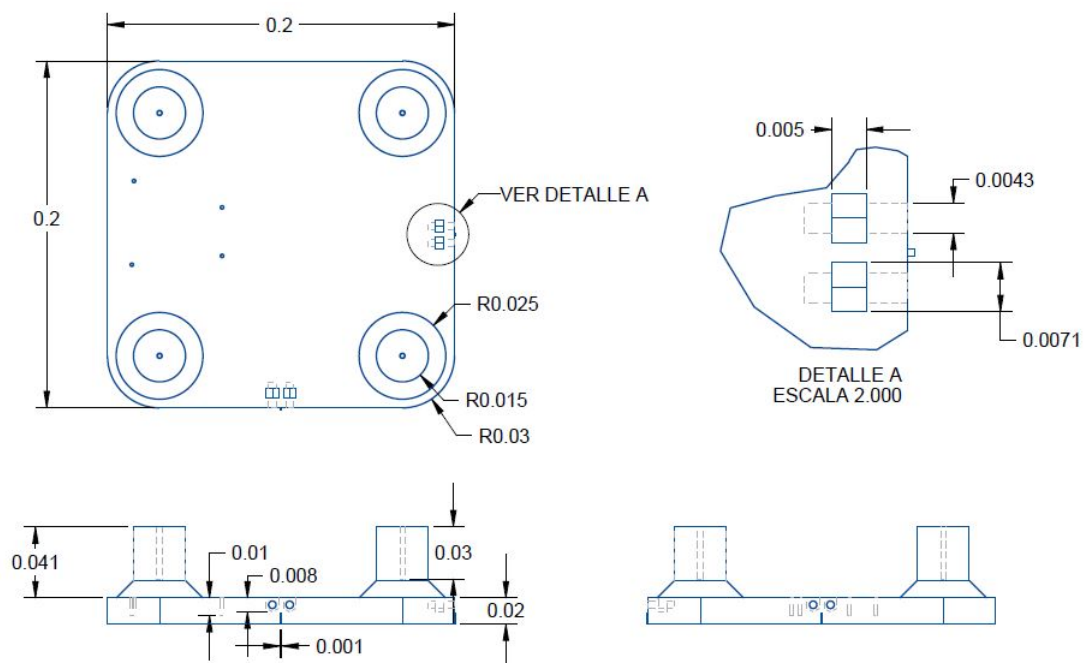


Figura E.1: Mides de la base inferior.
Font: Programa PTC CREO.

E.1.2 Base superior

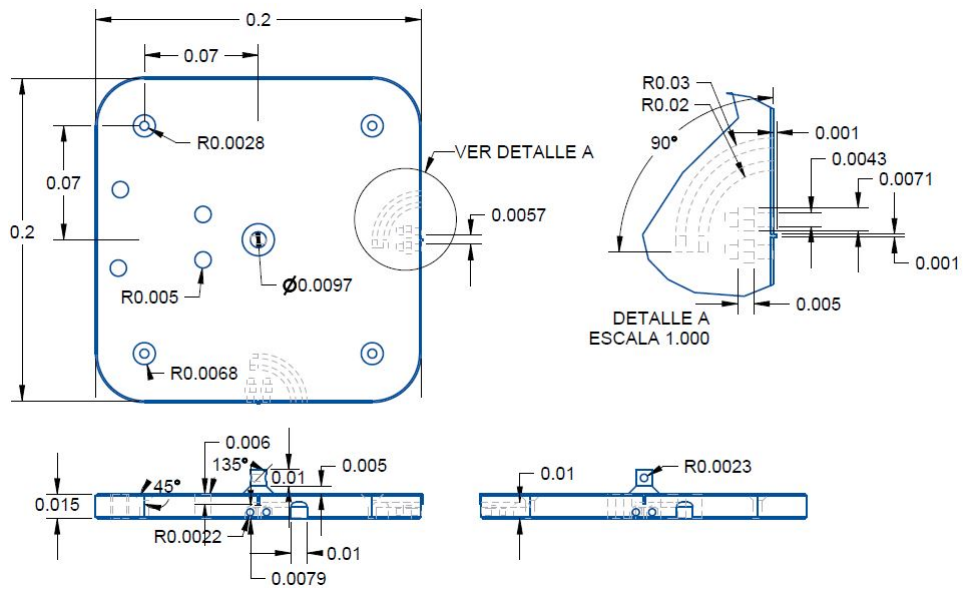


Figura E.2: Mides de la base superior.
Font: Programa PTC CREO.

E.1.3 Suport metacrilat

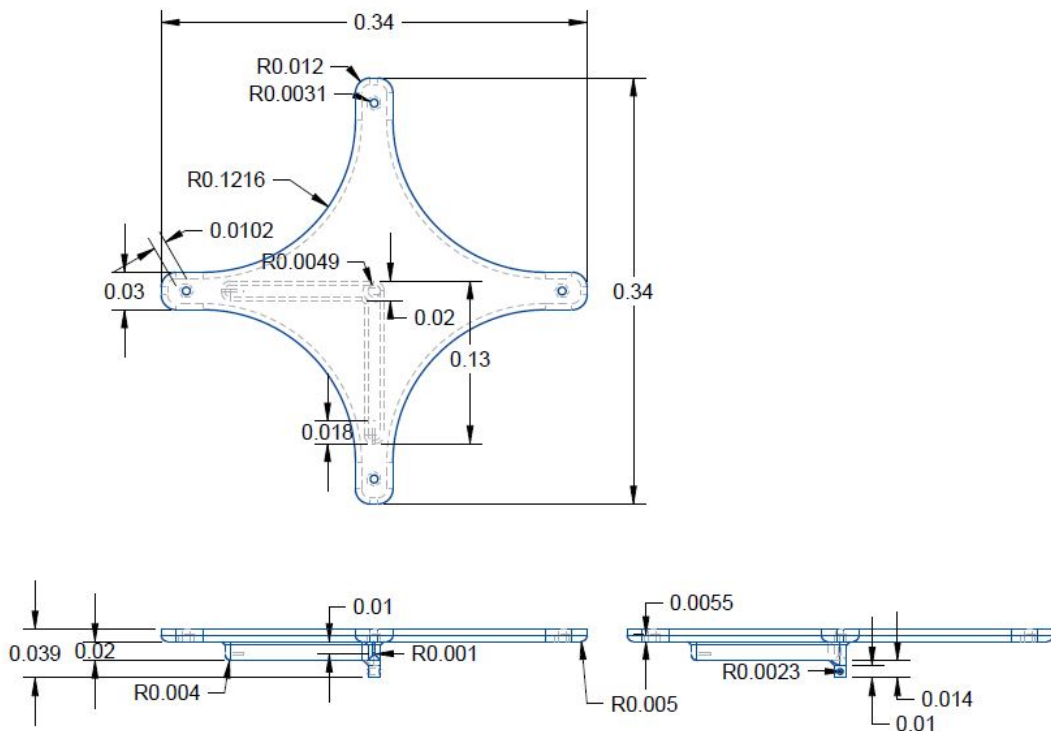


Figura E.3: Mides de la unió de la plataform.
Font: Programa PTC CREO.

E.1.4 Placa de Metacrilat

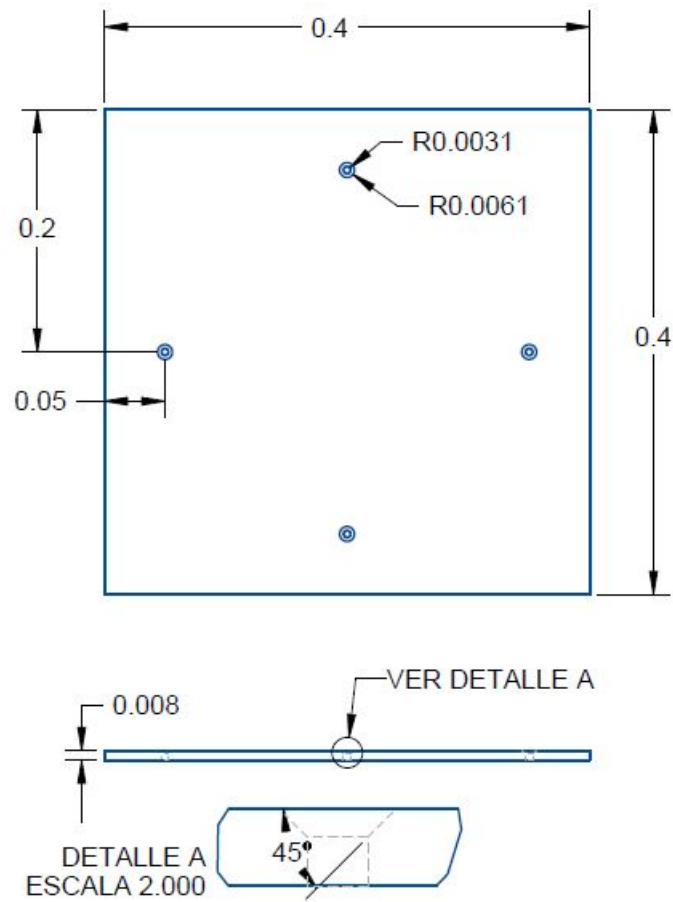


Figura E.4: Mides de la placa de metacrilat.
Font: Programa PTC CREO.

Apèndix F

Codi Matlab

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Codi Matlab %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4  clear all; close all; clc;
5  %Model del sistema
6  syms ddx ddy ddthetax ddthetay
7  m=0.175; % [kg]
8  R=0.0175; % [m]
9  Ie=(2/5)*m*R^2; % [kg*m^2]
10 g=9.81; % [m/s^2]
11 ddq=[ddx ddy ddthetax ddthetay];
12 Kpp=1;
13 Kpv=1;
14 Kep=1;
15 Kev=1;
16
17 syms l l1 r alpha theta xa ya za xb yb zb xc yc zc
18
19 %Equació de Lagrange
20 syms x y thetax thetay dx dy dthetax dthetay
21 L = simplify((1/2)*Ie*(dthetax*dthetax+dthetay*dthetay+...
22 + (dx*dx/R^2)+(dy*dy/R^2))+(1/2)*m*(dx*dx+dy*dy+...
23 +(x*dthetax+y*dthetay)^2)+m*g*(x*sin(thetax)+y*sin(thetay)));
24
25 %Derivació de les equacions de moviment.
26 dLq= simplify([diff(L,x),diff(L,y),diff(L,thetax),
27 diff(L,thetay)]);
28 dLdq=simplify([diff(L,dx),diff(L,dy),diff(L,dthetax),
29 diff(L,dthetay)]);
30 syms dx(t) dy(t) dthetax(t) dthetay(t) x(t) y(t) thetax(t)
   ↪ thetay(t) t
31
32 dLdq2=subs(dLdq,[dx, dy, dthetax, dthetay],[dx(t), dy(t),
   ↪ dthetax(t), dthetay(t)]);
33 dLdqt = simplify(diff(dLdq2,t));
```

```

34
35 syms ddx ddy ddthetax ddthetay T1 T2
36
37 %dq = [dx dy dthetax dthetay]
38 dLdqqt = simplify(subs(dLdqqt,[diff(dx(t),t), diff(dy(t),t),
  ↪ diff(dthetax(t),t), diff(dthetay(t),t)], [ddx, ddy,
  ↪ ddthetax, ddthetay]));
39 dLdqq = simplify(subs(dLdqqt,[dx(t) dy(t) dthetax(t)
  ↪ dthetay(t)], [dx dy dthetax dthetay]));
40 EQ = simplify(dLdqq - dLq)
41
42 EQ1 = simplify(EQ(1))
43 EQ2 = simplify(EQ(2))
44 EQ3 = simplify(EQ(3)-T1)
45 EQ4 = simplify(EQ(4)-T2)
46
47 RQ1=simplify(EQ1-simplify((m/((Ie/R^2)+m))*(x*(dthetax*dthetax)+...
48 +y*dthetax*dthetay+g*sin(thetax))-ddx))
49
50 %T1=Tx (Parell motor en l'eix X) T2=Ty (Parell motor en
  ↪ l'eix Y)
51
52
53 syms dthetax dthetay dthetaz dphix dphiy thetax thetay
  ↪ thetaz phix phiy
54
55 [r1,o1]=(polynomialReduce(EQ1,ddq,ddq));
56 [r2,o2]=(polynomialReduce(EQ2,ddq,ddq));
57 [r3,o3]=(polynomialReduce(EQ3,ddq,ddq));
58 [r4,o4]=(polynomialReduce(EQ4,ddq,ddq));
59
60
61 syms q11S q12S q13S q14S q21S q22S q23S q24S q31S q32S q33S
  ↪ q34S q41S q42S q43S q44S
62 q1S = [q11S, q12S, q13S, q14S];
63 q2S = [q21S, q22S, q23S, q24S];
64 q3S = [q31S, q32S, q33S, q34S];
65 q4S = [q41S, q42S, q43S, q44S];
66
67
68 syms r1S r2S r3S r4S
69
70 varSolS = simplify(linsolve([q1S; q2S; q3S; q4S],[-r1S;
  ↪ -r2S; -r3S; -r4S]));
71 varSol=subs(varSolS,[q11S,q12S,q13S,q14S,q21S,q22S,q23S,q24S
72 ,q31S,q32S,q33S,q34S,q41S,q42S,q43S,q44S,r1S,r2S,r3S,r4S],...
73 [o1(1),o1(2),o1(3),o1(4),o2(1),o2(2),o2(3),o2(4),o3(1)

```

```

74         ↪ , o3(2), o3(3), o3(4), o4(1), o4(2), o4(3), o4(4), r1, r2, r3, r4]);
75
76
77 syms x y thetax thetay dx dy dthetax dthetay
78 X = [x, dx, y, dy, thetax, dthetax, thetay, dthetay].';
79 U = [T1, T2].';
80 h = [X(1), X(2), X(3), X(4), X(5), X(6), X(7), X(8)];
81
82 dq=[dx, dy, dthetax, dthetay].';
83
84 f=vpa([X(2), varSol(1), X(4), varSol(2), X(6), varSol(3), X(8),
85 varSol(4)], 3);
86
87 dq=[dx dy dthetax dthetay].';
88 A0=eval(vpa(subs(jacobian(f,X), [X(1), X(2), X(3), X(4), X(5), X(6),
89 X(7), X(8), U(1), U(2)], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]), 3))
90 B0=eval(vpa(subs(jacobian(f,U), [X(1), X(2), X(3), X(4), X(5), X(6),
91 X(7), X(8), U(1), U(2)], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]), 3))
92 C0 =
93     ↪ eval(vpa(subs(jacobian(h,X), [X(1), X(2), X(3), X(4), X(5), X(6), X(7), X(8),
94 D0 =
95     ↪ eval(vpa(subs(jacobian(h,U), [X(1), X(2), X(3), X(4), X(5), X(6),
96 X(7), X(8)], [0, 0, 0, 0, 0, 0, 0, 0]), 3))
97 controlable = rank(ctrb(A0,B0))           %Huarria de sortir 8
98 observable = rank(observ(A0,C0))         %Hauria de sortir 8
99 %Voldria dir que les variables d'estat es poden veure i es
100     ↪ poden
101 %controlables directament pel sistema.
102
103 syms ox oy othetax othetay odx ody odthetax odthetay
104 states = {'x' 'y' 'thetax' 'thetay' 'dx' 'dy' 'dthetax'
105     ↪ 'dthetay'};
106 inputs = {'IT1' 'IT2'};
107 outputs = {'ox' 'oy' 'othetax' 'othetay' 'odx' 'ody'
108     ↪ 'odthetax' 'odthetay'};
109
110 sys_mimo = ss(A0,B0,C0,D0, 'statename', states, ...
111     'inputname', inputs, ...
112     'outputname', outputs);
113 %Converteix el sistema trobat en múltiples entrades i
114     ↪ múltiples sortides
115
116 %close loop
117
118 %Càlcul del control i de les matrius LQR.
119
120 clear figure;

```

```

115 x0=[0.1 0 0.1 0 0 0 0 0].'; %Estat inicial del sistema
116 Ts=0.01;
117 v = [1500 250 1500 250 750 1 750 1];           %Pesos
      ↪ d'importància de presició.
118 Q1=diag(v);
119 v2=[200 200];           %Pesos d'importància dels motors.
120 R1=diag(v2);
121 K1=lqr(A0,B0,Q1,R1);           %Matriu LQR que busquem per
      ↪ controlar el sistema. EQUACIO RICCATI?
122 [K1dd,S,E]=lqrd(A0,B0,Q1,R1,Ts);           %El mateix k pero
      ↪ en domini discret.
123 sys_loop=ss((A0-B0*K1dd),B0,C0,D0,'statename',
124 states,'inputname',inputs,'outputname', outputs); %Troba la
      ↪ funció de transferència en llaç tancat, tenint en
      ↪ compte que la K és la realimentació.
125 [K1d,S,E]=lqrd((A0-B0*K1dd),B0,Q1,R1,Ts);           %K1d -> La
      ↪ fem servir a l'arduino.
126 DiscBot= c2d(sys_loop,0.01);           %Trobar la funció de
      ↪ transferència en discret.
127 t=0:0.01:5;
128 [y,t,x]=initial(DiscBot,x0,t);           %t=Temps de simulació
      ↪ initial=u(z)
129 %[y,t,x]=step(DiscBot,t);
130 %[y,t,x]=step(sys_loop,t)
131 figure(1)
132 hold on;
133 p1=plot(t,y(:,1), 'LineWidth', 4)
134 hold off
135 figure(2)
136 hold on;
137
138
139 p2=plot(t,K1d*x.', 'LineWidth', 4)
140 hold off;
141 figure(3);
142 pzmap(sys_loop) %Mostra el lloc d les arrels en el temps
      ↪ discret
143 %construim les trayectories
144 %construim les trayectories
145 radi=0.050;
146 t=[0.01:0.01:72].'
147 %t=ones(1,7201). '
148 angle=[0.1:0.1:720]. '
149
150 %Trajectoria per a fer una espiral
151 % [xx]=radi*sin(angle.*pi/180).*t
152 % [yy]=radi*cos(angle.*pi/180).*t
153

```

```
154 %Trajectoria per a fer un cercle
155 [xx]=radi*sin(angle.*pi/180)
156 [yy]=radi*cos(angle.*pi/180)
157
158 %Trajectòria El·lipse
159 % [xx]=0.10*sin(angle.*pi/180)
160 % [yy]=radi*cos(angle.*pi/180)
161
162 %Trajectòria Infinit
163 % d=0.1
164 % [xx]=(d.*sqrt(2).*cos(t))/(sin(t).^2+1)
165 % [yy]=(d.*sqrt(2).*cos(t).*sin(t))/(sin(t).^2+1)
166 % plot(x,y)
167 % [xx]=normalize(x,'range',[-0.1,0.1])
168 % [yy]=normalize(y,'range',[-0.1,0.1]) %Normalitza la X i
  ↪ la Y per a que vagi d'un rang de -0.1 a 0.1 per que no
  ↪ superi 0.1
169 % t=[0:0.01:50].';
170 % xx=[0:0.00004:0.200].';
171 % [xx]=0.02*(0.5.*xx).*t;
172 % [yy]=0.02*(0.5.*xx).*t;
173 % radi=0.025;
174 %t=ones(1,3601).';
175 % t=[0:0.01:144].';
176 % angle=[0:0.1:1440].';
177 % [xx]=radi*sin(angle.*pi/180).*t
178 % [yy]=radi*cos(angle.*pi/180).*t
179 % [xx,dx,dxx]=tpoly(0.01,0.100,7200)
180 % [yy,dy,dyy]=tpoly(0.01,0.0500,7200)
```

Apèndix G

Codi servomotor

```
1 #include <Servo.h>
2
3 Servo myservo1; // create servo object to control a servo
4 Servo myservo2; // create servo object to control a servo
5
6 int potpin = 0; // analog pin used to connect the
   potentiometer
7 int val; // variable to read the value from the analog
   pin
8
9 void setup() {
10  myservo1.attach(2); // attaches the servo on pin 9 to
   the servo object
11  myservo2.attach(4); // attaches the servo on pin 9 to
   the servo object
12 }
13
14 void loop() {
15  val = analogRead(potpin); // reads the value
   of the potentiometer (value between 0 and 1023)
16  val1 = map(val, 0, 1023, 0, 180); // scale it to use
   it with the servo (value between 0 and 180)
17  val2 = map(val, 0, 1023, 180, 0);
18  myservo1.write(val1); // sets the servo
   position according to the scaled value
19  myservo2.write(val2);
20  delay(15); // waits for the
   servo to get there
```

Apèndix H

Codi làmina tàctil resistiva de 4 fils

```
1 int Y1=A0;
2 int X1=A1;
3 int Y2=A2;
4 int X2=A3;
5 #define Xresolution 500
6 #define Yresolution 500
7 float timeStar=0, temps=0;
8
9 void setup()
10 {
11     Serial.begin(115200);
12 }
13
14 void loop()
15 {
16     temps=millis()-timeStar;
17     timeStar=millis();
18     Serial.println(temps);
19     Touchscreen();
20 }
21
22 void Touchscreen()
23 {
24     int Xmes, Ymes; //Touch Coordinates are stored in X,Y
        variable
25     float X,Y;
26     pinMode(X1,OUTPUT);
27     pinMode(X2,OUTPUT);
28     pinMode(Y1,INPUT);
29     pinMode(Y2,INPUT);
30     digitalWrite(X1,HIGH);
31     digitalWrite(X2,LOW);
32     digitalWrite(Y2,LOW);
```



```
33  delay(2);
34 //  Xmes=(analogRead(Y1))/(4023/Xresolution);
35  Xmes=0;
36  for (int i=0; i<10; i++){
37    Xmes=Xmes +(analogRead(Y1))/(4023/Xresolution);
38  }
39  Xmes=Xmes/10;
40
41  pinMode(Y1,OUTPUT);
42  pinMode(Y2,OUTPUT);
43  pinMode(X1,INPUT);
44  pinMode(X2,INPUT);
45  digitalWrite(Y1,HIGH);
46  digitalWrite(Y2,LOW);
47  digitalWrite(X2,LOW);
48  delay(2);
49 //  Ymes=(analogRead(X1))/(4023/Yresolution);
50  Ymes=0;
51  for (int i=0; i<10; i++){
52    Ymes=Ymes +(analogRead(X1))/(4023/Yresolution);
53  }
54  Ymes=Ymes/10;
55
56  //Display X and Y on Serial Monitor
57  if (Xmes>=35){
58    X=map(Xmes,66,431,-189,189); //Reads X axis touch
    position
59  }
60  else if (Xmes<35){
61    X=0;
62  }
63  if (Ymes>=30){
64    Y=map(Ymes,70,390,-150,150);
65  }
66  else if (Ymes<30){
67    Y=0;
68  }
69
70  Serial.print("X_=_");
71  Serial.print(X);
72  Serial.print("\t_Y_=_");
73  Serial.println(Y);
74 }
```

Apèndix I

Codi del teclat matricial

```
1 #include <Keypad.h>
2
3 const byte ROWS = 4; //four rows
4 const byte COLS = 4; //four columns
5 //define the cymbols on the buttons of the keypads
6 char hexaKeys[ROWS][COLS] = {
7   { '1','4','7', '*' },
8   { '2','5','8', '0' },
9   { '3','6','9', '*' },
10  { 'A','B','C', 'D' }
11 };
12 byte rowPins[ROWS] = {25, 24, 23, 22}; //connect to the row
    pinouts of the keypad
13 byte colPins[COLS] = {29, 28, 27, 26}; //connect to the
    column pinouts of the keypad
14 volatile char customKey;
15 int interruptOUT = 32;
16 int interruptIN = 33;
17
18 //initialize an instance of class NewKeypad
19 Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins
    , colPins, ROWS, COLS);
20
21 void setup() {
22   Serial.begin(9600);
23   pinMode(interruptOUT, OUTPUT);
24   pinMode(interruptIN, INPUT_PULLUP);
25   attachInterrupt(digitalPinToInterrupt(interruptIN),
    Keypads, CHANGE);
26 }
27
28 void loop() {
29   customKey = customKeypad.getKey();
30   if (customKey){
31     digitalWrite(interruptOUT, HIGH);
```

```
32  }
33  else{
34      digitalWrite(interruptOUT, LOW);
35  }
36 }
37
38 void Keypads() {
39     Serial.println(customKey);
40 }
```

Apèndix J

Codi *Ball on Plate* complet

```
1 #include <DueTimer.h>
2 #include <Servo.h>
3 #include <Keypad.h>
4
5 //===== TECLAT MATRICIAL =====
6 const byte ROWS = 4; //quatre files
7 const byte COLS = 4; //quatre columnes
8 //es defineixen els símbols del teclat matricial
9 char hexaKeys[ROWS][COLS] = {
10   { '1', '4', '7', '*' },
11   { '2', '5', '8', '0' },
12   { '3', '6', '9', '#' },
13   { 'A', 'B', 'C', 'D' }
14 };
15 byte rowPins[ROWS] = {25, 24, 23, 22}; //es defineixen els
    pins de les files
16 byte colPins[COLS] = {29, 28, 27, 26}; //es defineixen els
    pins de les columnes
17 volatile char customKey;
18 int intTECLATOUT = 32;
19 int intTECLATIN = 33; //S'inicialitza una instància de la
    classe teclat matricial
20 Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins
    , colPins, ROWS, COLS);
21
22 //===== SERVO =====
23 Servo myservo1; // Es crea l'objecte servo per a controlar
    el servo
24 Servo myservo2;
25 double motorX, motorY, val; //Variables de per a fer
    funcionar el Servo
26
27 //===== TOUCHSCREEN =====
28 int Y1=A0;
29 int X1=A1;
```

```

30 int Y2=A2;
31 int X2=A3;
32 #define Xresolution 500 //Aquesta resolucio es aproximada i
    s'ajusta de forma experimental
33 #define Yresolution 500
34
35 //    CALCULS
36 double Ke,Kep, Kev, Kp, Kpp, Kpv;
37 float X,Y, thetaX, thetaY, dX, dY, dthetaX, dthetaY, alphaX
    , alphaY; //Posicio i velocitat objectius
38 float Xant=0, Yant=0, thetaXant=0, thetaYant=0; //
    Valors variables estats anteriors
39 double Xmesurada,Ymesurada,Zmesurada, dXmesurada,
    dYmesurada; //
40 double thetaXX, thetaYY;
41 double timer, TimerStar=0, dt_Vel, TimerStar2=0;
42 int sampleRate=7, C_angle=0;
43 double Klqr[8][1];
44 float EstatReal[8][1];
45 float Error[8][1];
46 float Error2[8][1];
47 float EstatRef[8][1];
48 long double K[2][8]={0.0621, 0.0402, 0,0 ,0.0661, 0.0021,
    0, 0},{0, 0, 0.0621, 0.0402, 0, 0, 0.0661, 0.0021}};
49 int i,k;
50
51 void setup() {
52 // put your setup code here, to run once:
53 X=0;
54 Y=0;
55 dX=0;
56 dY=0;
57 dthetaX=0; //Definim variables inicials
58 dthetaY=0;
59 Serial.begin(115200);
60 myservo1.attach(4); // attaches the servo on pin 9 to
    the servo object
61 myservo2.attach(2); // attaches the servo on pin 9 to
    the servo object
62 pinMode(intTECLATOUT, OUTPUT);
63 pinMode(intTECLATIN, INPUT_PULLUP);
64 attachInterrupt(digitalPinToInterrupt(intTECLATIN),
    KeyPads, CHANGE); //Interrupcio teclat
65 customKey = customKeypad.getKey();
66 while (customKey!='*'){
67     customKey = customKeypad.getKey();
68 }
69 Serial.println(customKey);

```

```
70 delay(100);
71 Serial.print("Entra_Kep_(guany_posicio_esfera):");
72 while (!Serial.available());
73 Kep = Serial.parseFloat();
74 Serial.println(Kep);
75 Serial.print("Entra_Kev_(guany_velocitat_esfera):");
76 while (!Serial.available());
77 Kev = Serial.parseFloat();
78 Serial.println(Kev);
79 Serial.print("Entra_Kpp_(guany_posicio_pivotatge_(
    inclinacio_plataforma):");
80 while (!Serial.available());
81 Kpp = Serial.parseFloat();
82 Serial.println(Kpp);
83 Serial.print("Entra_Kpv_(guany_velocitat_pivotatge_(
    inclinacio_plataforma):");
84 while (!Serial.available());
85 Kpv = Serial.parseFloat();
86 Serial.println(Kpv);
87 }
88
89 void loop() {
90   customKey = customKeypad.getKey();
91   if (customKey) {
92     digitalWrite(intTECLATOUT, HIGH);
93   }
94   else{
95     digitalWrite(intTECLATOUT, LOW);
96   }
97   dt_Vel=millis()-TimerStar2; //Comprovem el temps de cicle
    (loop)
98   TimerStar2=millis();
99   LlegirSensor();
100  thetaXX=(double)thetaX;
101  thetaYY=(double)thetaY;
102  timer=millis()-TimerStar;
103  if (timer>=sampleRate){
104    TimerStar=millis();
105    EstatRef[0][0]=X;
106    EstatRef[1][0]=0;
107    EstatRef[2][0]=Y;
108    EstatRef[3][0]=0;
109    EstatRef[4][0]=0;
110    EstatRef[5][0]=0;
111    EstatRef[6][0]=0;
112    EstatRef[7][0]=0;
113
114    EstatReal[0][0]=Xmesurada/1000;
```

```

115     EstatReal[1][0]=dXmesurada;
116     EstatReal[2][0]=Ymesurada/1000;
117     EstatReal[3][0]=dYmesurada;
118     EstatReal[4][0]=thetaXX;
119     EstatReal[5][0]=dthetaX;
120     EstatReal[6][0]=thetaYY;
121     EstatReal[7][0]=dthetaY;
122     CalculError();
123     UpdateMotors();
124 }
125 }
126
127 //===== CALCUL ERROR =====
128 void CalculError() {
129     for(k=0;k<4;k++) {
130         if(k==0 | k==2)Ke=Kep;
131         else Ke=Kev;
132         Error[k][0] = (EstatRef[k][0] - EstatReal[k][0])*Ke;
133     }
134     for(k=4;k<8;k++) {
135         if(k==4 | k==6)Kp=Kpp;
136         else Kp=Kpv;
137         Error[k][0] = (EstatRef[k][0] - EstatReal[k][0])*Kp;
138     }
139     for(k=0;k<8;k++) {
140         Error2[k][0] = (Error[k][0] - EstatReal[k][0])*Ke;
141     }
142 // //Saturacio Error
143 // int E=20;
144 // if (Error2[0][0]>=E) Error2[0][0]=E;
145 // if (Error2[2][0]>=E) Error2[2][0]=E;
146 // if (Error2[0][0]<=-E) Error2[0][0]=-E;
147 // if (Error2[2][0]<=-E) Error2[2][0]=-E;
148
149     for (i=0; i<2; i++) {          //2 = Nombre de files de K
150         for (k=0; k<8; k++) {      //8 = Nombre de columnes de A
151             i nombre de files de error
152             val=val + K[i][k]*Error2[k][0];    // K -> Matriu i
153             ->fila k-> columna
154         }
155         Klqr[i][0]=val;           //Sumatori d'errors que entrara en
156                                     el motor
157         val=0;
158     }
159     Serial.println(Klqr[0][0]);
160     Serial.println(Klqr[1][0]);
161 }

```

```
160 //===== UPDATE MOTORS =====
161 void UpdateMotors() {
162   motorX=Klqr[0][0]*(180/PI)+90; //Aixo em donaria l'angle
      que s'hauria de moure els motors
163   motorY=Klqr[1][0]*(180/PI)+90;
164   if (motorX>=180) motorX=175;
165   if (motorX<=0) motorX=0; //Saturacio
166   if (motorY>=180) motorY=175;
167   if (motorY<=0) motorY=0;
168   if (Zmesurada==0) {
169     motorX=90;
170     motorY=90;
171   }
172   myservol.write(motorX); // sets the
      servo position according to the scaled value
173   myservo2.write(motorY);
174 }
175
176 //===== LLEGIR SENSORS =====
177 void LlegirSensor() {
178   Touchscreen();
179   CalculX();
180   CalculY();
181   //   velocitat
182   dXmesurada=( (Xmesurada-Xant)/dt_Vel); //m/s
183   dYmesurada=( (Ymesurada-Yant)/dt_Vel);
184   dthetaX=(thetaXX-thetaXant)/(dt_Vel/1000); //graus/s
185   dthetaY=(thetaYY-thetaYant)/(dt_Vel/1000);
186   Xant=Xmesurada;
187   Yant=Ymesurada;
188   thetaXant=thetaXX;
189   thetaYant=thetaYY;
190 //   Serial.println(dt_Vel);
191   if (Zmesurada==1) {
192 //     Serial.print("X = ");
193 //     Serial.print(Xmesurada);
194 //     Serial.print("\t Xant = ");
195 //     Serial.print(Xant);
196 //     Serial.print("\t Y = ");
197 //     Serial.print(Ymesurada);
      //=====
198 //     Serial.print("\t Yant = ");
199 //     Serial.print(Yant);
200 //     Serial.print("\t dX = ");
201 //     Serial.print(dXmesurada);
202 //     Serial.print("\t dY = ");
203 //     Serial.println(dYmesurada);
204   }
```



```

205 }
206
207 //===== TRANSFORMACIO DELS ANGLES PLATAFORMA-SERVO X
    ====
208 void CalculX(){
209     float za=0, l=0.0754321, r=0.019, theta=0, xb=0, ya
        =0.200, yc=0.200, xa=0, xc=0; //calcul angle
        plataforma
210     float yb, zb, zc, l1, N, M, K;
211 //     Theta X mesurada
212     yb=r*sin(alphaX)*cos(theta)+ya;
213     zb=r*cos(alphaX)*sin(theta)+za;
214     zc=sqrt(pow(r,2)+pow(l,2)+2*r*l*sin(alphaX));
215     l1=sqrt(pow((xc-xa),2)+pow((yc-ya),2)+pow((zc-za),2));
216     N=2*(za-zc);
217     M=pow(l,2)-pow(l1,2)-pow(r,2);
218     K=2*(xa-xc)*r*cos(theta)+xa+2*(ya-yc)*r*sin(theta)+ya;
219     thetaX = acos(M/(sqrt(pow(K,2)+pow(N,2))))-PI/2;
220 }
221 //===== TRANSFORMACIO DELS ANGLES PLATAFORMA-SERVO Y
    ====
222 void CalculY(){
223     float za=0, l=0.0754321, r=0.019, theta=0, xb=0, xa
        =0.200, xc=0.200, ya=0, yc=0; //calcul angle
        plataforma
224     float yb, zb, zc, l1, N, M, K;
225 //     theta Y mesurada
226     alphaY=myservol.read()*(PI/180);
227     yb=r*sin(alphaY)*cos(theta)+xa;
228     zb=r*cos(alphaY)*sin(theta)+za;
229     zc=sqrt(pow(r,2)+pow(l,2)+2*r*l*sin(alphaY));
230     l1=sqrt(pow((xc-xa),2)+pow((yc-ya),2)+pow((zc-za),2));
231     N=2*(za-zc);
232     M=pow(l,2)-pow(l1,2)-pow(r,2);
233     K=2*(xa-xc)*r*cos(theta)+xa+2*(ya-yc)*r*sin(theta)+ya;
234     thetaY = acos(M/(sqrt(pow(K,2)+pow(N,2))))-PI/2; //
        Velocitat real
235 }
236
237 //===== POSICIO BOLA SOBRE LA PLATAFORMA
    =====
238 void Touchscreen()
239 {
240     int Xmes, Ymes; //Touch Coordinates are stored in X,Y
        variable
241     pinMode(X1,OUTPUT);
242     pinMode(X2,OUTPUT);
243     pinMode(Y1,INPUT);

```

```
244  pinMode(Y2, INPUT);
245  digitalWrite(X1, HIGH);
246  digitalWrite(X2, LOW);
247  digitalWrite(Y2, LOW);
248  delay(3);
249  Xmes=0;
250  for (int i=0; i<10; i++){
251    Xmes=Xmes +(analogRead(Y1))/(1024/Xresolution);
252  }
253  Xmes=Xmes/10;
254
255  pinMode(Y1, OUTPUT);
256  pinMode(Y2, OUTPUT);
257  pinMode(X1, INPUT);
258  pinMode(X2, INPUT);
259  digitalWrite(Y1, HIGH);
260  digitalWrite(Y2, LOW);
261  digitalWrite(X2, LOW);
262  delay(2);
263  Ymes=0;
264  for (int i=0; i<10; i++){
265    Ymes=Ymes +(analogRead(X1))/(1024/Yresolution);
266  }
267  Ymes=Ymes/10;
268
269  //Display X and Y on Serial Monitor
270  if (Xmes>=35){
271    Xmesurada=map(Xmes, 66, 429, -189, 189); //Reads X axis
272    touch position
273  }
274  else if (Xmes<35){
275    Xmesurada=0;
276  }
277  if (Ymes>=35){
278    Ymesurada=map(Ymes, 75, 390, -150, 150);
279  }
280  else if (Ymes<35){
281    Ymesurada=0;
282  }
283  if (Xmes<35 and Ymes<35){
284    Zmesurada=0; //No pressing on touchscreen
285  }
286  else{
287    Zmesurada=1;
288  }
289
290 //===== KEYPAD =====
```

```
291 void KeyPads() {
292     switch (customKey) {
293     case '1':
294         Serial.println("Case_1: X=0, Y=0 (m)");
295         X=0;
296         Y=0;
297         break;
298     case '2':
299         Serial.println("Case_2: X=0.1, Y=0.1 (m)");
300         X=0.100;
301         Y=0.100;
302         break;
303     case '3':
304         Serial.println("Case_3: X=-0.1, Y=-0.1 (m)");
305         X=-0.100;
306         Y=-0.100;
307         break;
308     case '4':
309         Serial.println("Case_4: X=0.1, Y=-0.1 (m)");
310         X=0.100;
311         Y=-0.100;
312         break;
313     case '5':
314         Serial.println("Case_5: X=-0.1, Y=0.1 (m)");
315         X=-0.100;
316         Y=0.100;
317         break;
318     case '6':
319         Serial.println("Case_6: X=0.05, Y=0.05 (m)");
320         X=0.050;
321         Y=0.050;
322     case '7':
323         Serial.println("Case_7: X=-0.05, Y=-0.05 (m)");
324         X=-0.050;
325         Y=-0.050;
326         break;
327     case '8':
328         Serial.println("Case_8: X=-0.05, Y=0.05 (m)");
329         X=-0.050;
330         Y=0.050;
331         break;
332     case '9':
333         Serial.println("Case_9: X=0.05, Y=-0.05 (m)");
334         X=0.050;
335         Y=-0.050;
336         break;
337     default:
338     // X=0; // if nothing else matches, do the default QUE LA
```

```
        PILOTA ES QUEDI AL CENTRE
339 //      Y=0; // default is optional
340  break;
341 }
342 }
```

Glossari

- analogWriteResolution()** Funció que s'utilitza per canviar la resolució dels pins de sortida PWM.. 52
- Anàlisi de Circuits** És l'assignatura que ens va ensenyar les tècniques per obtenir paràmetres sobre els diferents circuits elèctrics.. 10
- Arduino** Plataforma electrònica de codi obert basat en hardware i software fàcil d'utilitzar amb una gran quantitat d'informació oberta i gratuïta per als usuaris. 3, 4, 6, 7, 13, 27, 28, 30–33, 35, 43–45, 47, 51, 57, 58, 61
- Ball on Plate** Sistema compost per una bola sobre una plataforma mòvil, controlada per motors, que s'utilitza en experimentació per estudiar els diferents tipus de control automàtic.. 3, 4, 6, 9–11, 13, 18, 44–46, 76
- C++** Llenguatge de programació d'alguns microcontroladors, que deriva del llenguatge de programació C.. 3, 4
- Cardan** Unió articulada netre dos eixos que permet transmetre un moviment de rotació a diversos angles.. 3, 4, 7, 13, 34–36, 38, 39, 45
- clock** En aquest treball es refereix a la freqüència a la que treballa un microcontrolador.. 28
- Control Discret** Camp interdisciplinari entre l'enginyeria i les matemàtiques que estudia el comportament de sistemes en el temps discret.. 10
- Control System Tuner** Aplicació que ajuda a modificar guanyos d'un sistema de control per a millorar les seves característiques.. 23, 47
- Disseny Mecatronic** Disseny en el que és té en compte tots els aspectes del sistema, la part mecànica, la part elèctrica i la part electrònica.. 11, 13, 27, 28, 30, 41, 45
- Duty cicle** Part del període en què un senyal està actiu (està al valor superior d'un senyal quadrat).. 43, 54
- Elasticitat i Resistència de Materials** És considerada com aquella part de la mecànica que estudia els sòlids deformables elàstics d'interès en la enginyeria.. 10
- Euler-Lagrange** Aquestes equacions ens permeten comptar amb un sistema analític per arribar a les equacions que descriuen el comportament físic de les partícules.. 16

- Física Mecànica** La mecànica es la branca de la física que estudia el moviment dels cossos, i la seva evolució en el temps, sota l'acció de forces.. 10
- Lagrange** Mètode matemàtic que descriu la dinàmica d'un sistema mecànic a través de la diferència entre l'energia cinètica i la potencial.. 14–16, 23
- Lloc Geometric de les Arrels** Representació dels pols i zeros d'un sistema de control en un gràfic bidimensional que dóna informació sobre la resposta transitòria i l'estabilitat d'un sistema de control.. 7, 21, 22, 46
- Matlab** Programa informàtic per càlcul matemàtic. 3, 4, 11, 14, 18, 19, 22, 23, 26, 47, 86
- Microcontroladors** És l'assignatura que ens va ensenyar els diferents tipus de microcontroladors i com programar-los.. 10
- Multibody** Aplicació de Matlab que permet unir un sistema de control amb el sistema físic per poder simular i visualitzar en format 3D el comportament del sistema.. 18, 26, 47
- NPN** Tipus de transistor NPN.. 7, 31
- PTC CREO** Programa de disseny assistit per ordinador que ajuda a la realització del disseny mecànic.. 18, 22, 24, 25, 32
- Regulació Automàtica** Camp interdisciplinari entre l'enginyeria i les matemàtiques que estudia el comportament de sistemes dinàmics. 10
- servomotor** Tipus de motor elèctric al que podem controlar la seva velocitat i posició.. 3, 4, 6–8, 10, 12, 13, 17, 19, 24–26, 28–32, 34, 36, 37, 39, 40, 43–45, 47, 48, 53, 54, 56, 57
- Simulink** Programa de Matlab per simular sistemes. 18, 22–24, 47
- withanalogReadResolution()** Funció que s'utilitza per canviar la resolució dels pins Analògics d'entrada, que per defecte tenen 10 bits.. 52

Acrònims

3D 3 Dimensions. 32, 33, 35, 37, 38

CAD Computer-Aided Design. 7, 24, 32, 34, 39–41, 45, 47

DC Direct current. 54

DoF Degrees of Freedom. 13, 38, 45

LQR Linear Quadratic Regulator. 3, 4, 10, 11, 19, 24, 46, 47

PWM Pulse-Width Modulation. 28, 30, 31, 43, 54, 57, 58

TFG Treball de Fi de Grau. 11