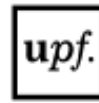




TecnoCampus
Escola Superior
Politécnica

Centre adscrit a:



Universitat
Pompeu Fabra
Barcelona

Grau en Enginyeria Mecànica

**ESTUDI I SIMULACIÓ DE LA CINEMÀTICA I DINÀMICA D'UN ROBOT
COL-LABORATIU DE 6 GRAUS DE LLIBERTAT**

Memòria

**ÀLEX PASCUAL MONROIG
PONENT: JOAN TRIADÓ AYMERICH**

ESTIU 2022



TecnoCampus
Mataró-Maresme

Resum

Aquest informe consisteix en el estudi i disseny d'un codi capaç de calcular la cinemàtica i la dinàmica del robot TM14X utilitzant l'aplicació de Matlab.

Per dissenyar les funcions s'han fet ús de les funcions internes bàsiques de Matlab i les funcions proporcionades per les llibreries *Robotic System Toolbox* i *Symbolic Toolbox*. Utilitzant aquestes funcions hem pogut dissenyar una sèrie de funcions capaces de representar el comportament cinemàtic del nostre robot. Per a calcular el comportament dinàmic hem utilitzat les metodologies de Newton-Euler i Walker-Orin conjuntament amb les llibreries mencionades anteriorment.

Resumen

Este informe consiste en el estudio y diseño de un código capaz de calcular la cinemática y la dinámica del robot TM14X utilizando la aplicación de Matlab.

Para diseñar las funciones se han hecho uso de las funciones internas básicas de Matlab y las funciones proporcionadas por las librerías *Robotic System Toolbox* y *Symbolic Toolbox*. Utilizando estas funciones hemos podido diseñar una serie de funciones capaces de representar el comportamiento cinemático de nuestro robot. Para calcular el comportamiento dinámico hemos utilizado las metodologías de Newton-Euler y Walker-Orin juntamente con las librerías mencionadas anteriormente.

Abstract

This report consists of the study and design of a code capable of calculating the kinematics and dynamics of the TM14X robot using the Matlab application.

To design functions, the basic internal Matlab functions and functions provided by the *Robotic System Toolbox* and *Symbolic Toolbox* libraries have been used. Using these functions we've been able to design a number of functions capable of representing the kinematic behavior of our robot. To calculate the dynamic behavior we have used Newton-Euler and Walker-Orin methodologies together with the libraries mentioned above.

Índex.

Índex.....	I
Índex de figures.....	IV
Índex de taules.....	VII
1. Objectius	1
1.1. Propòsit.....	1
1.2. Finalitat	1
1.3. Objecte.....	1
1.4. Abast	1
1.5. Context en les línies de recerca i transferència de coneixement del Tecnocampus	2
2. Marc teòric	3
3. Metodologia del projecte	7
3.1. Etapa d'Anàlisi:	7
3.2. Etapa d'Avaluació d'alternatives:.....	7
3.3. Etapa de Disseny de la solució:	7
3.4. Etapa de Verificació de la solució:	7
4. Objectius tècnics	9
5. Especificacions tècniques	11
6. Plantejament inicial.....	13

II

6.1.	Cinemàtica	15
6.1.1.	Possibilitats de càlcul	15
6.1.2.	Selecció i desenvolupament de la solució	16
6.2.	Dinàmica.....	18
6.2.1.	Possibilitats de càlcul	18
6.2.2.	Selecció i desenvolupament de la solució	21
6.3.	Simulació	24
6.3.1.	1era Simulació.....	24
6.3.2.	2na Simulació.....	25
7.	Desenvolupament de la solució	27
7.1.	Paràmetres Denavit-Hartenberg	27
7.2.	Cinemàtica inversa	33
7.3.	Dinàmica inversa	44
7.3.1.	Euler-Lagrange.....	44
7.3.2.	Newton-Euler	50
7.4.	Dinàmica directa.....	55
7.5.	Resultats Simulació	57
8.	Planificació.....	61
8.1.	Pla d'execució (Diagrama de Gantt)	61
8.2.	Pla d'execució (Diagrama de Gantt)	61

8.2.1.	Planificació projecte	61
8.2.2.	Planificació avantprojecte.....	63
8.3.	Anàlisi i Gestió de Riscos	65
8.3.1.	Taula FMEA	66
8.3.2.	Gestió dels riscos	67
8.4.	Eines de Seguiment.....	69
8.5.	Replantejament memòria intermèdia	71
9.	Anàlisi Econòmica	73
10.	Anàlisi Mediambiental.....	75
11.	Conclusions.....	77
12.	Referències.....	79

Índex de figures.

Figura 2.1: Quatrinomi robot amb capacitat	3
Figura 2.2 Músculs robòtics	5
Figura 2.3 Previsió del ús de robots col·laboratius	6
Figura 5.1 Robot del tipus universal	11
Figura 6.1 Articulació cargol de 2 GDL	13
Figura 6.2 Exemple simulació.....	25
Figura 7.1 Robot TM14X en la posició inicial.....	28
Figura 7.2 Exemple del procediment per situar trames.....	29
Figura 7.3 Obtenció paràmetres DH	30
Figura 7.4 Paràmetres DH hipòtesis 1	31
Figura 7.5 Paràmetres DH hipòtesis 2.....	32
Figura 7.6 Resum dels detalls del robot	37
Figura 7.7 Representació esquelet del robot configurat.....	38
Figura 7.8 Possibles solucions al problema de la cinemàtica inversa.....	42
Figura 7.9 Finestra del robot interactiva	43
Figura 7.10 Nova posició i orientació del robot interactiu.....	43
Figura 7.11 Articulació 1 amb PID - θ destitjada = 55°	57
Figura 7.12 Articulació 2 amb PID - θ destitjada = 50°	58
Figura 7.13 Articulació 3 amb PID - θ destitjada = 20°	58

Figura 7.14 Articulació 5 amb PID - θ destitjada = 70°	59
Figura 7.15 Articulació 4 amb PID - θ destitjada = 30°	59
Figura 7.16 Articulació 6 amb PID - θ destitjada = 60°	60
Figura 8.1 Diagrama de Gantt projecte	63
Figura 8.2 Diagrama de Gantt avantprojecte.....	65
Figura 8.3 Jira llistat de tasques	69
Figura 8.4 Jira Log Work	70
Figura 8.5 Diagrama de Gantt replantejament.....	72

Índex de taules.

Taula 5.1 Limitacions robot TM14X.....	11
Taula 7.1 Paràmetres DH hipòtesis 1	31
Taula 7.2 Paràmetres DH hipòtesis 2	32
Taula 7.3 Configuració de les posicions de les articulacions	41
Taula 7.4 Característiques enllaços	48
Taula 8.1 Anàlisi de riscos FMEA	66
Taula 9.1 Preus llicències	73

1. Objectius

1.1. Propòsit

El propòsit d'aquest projecte consisteix en el estudi de la cinemàtica i la dinàmica del robot col·laboratiu TM14X.

1.2. Finalitat

La finalitat del projecte consisteix en l'obtenció de les equacions de moviment que defineixin l'acceleració angular dels motors i el disseny d'una funció que permeti obtenir les coordenades angulars que defineixin el gir del motor donada una sèrie de coordenades articulars i l'orientació de l'eina en la posició destí.

1.3. Objecte

L'objecte d'aquest informe serà la presentació dels resultats obtinguts de l'estudi de la dinàmica i cinemàtica del robot col·laboratiu. El projecte en qüestió tindrà com a objectiu el disseny d'un controlador de posició per a un robot industrial de 6 eixos.

Per realitzar aquest disseny es dividirà el projecte en dos informes. Aquest informe consistirà en la component mecànica del projecte, mentre que la component electrònica del projecte es desenvoluparà en l'informe complementari anomenat "*Disseny i simulació d'un controlador PID pel seguiment de trajectòries d'un robot de 6 graus de llibertat*".

1.4. Abast

En aquest projecte es presentarà les funcions dissenyades per durant l'estudi de la cinemàtica inversa i la dinàmica així com els resultats obtinguts de les simulacions creades per a representar els resultats de forma visual utilitzant el Matlab.

El projecte està dividit en dues parts, la primera corresponent a la component mecànica del projecte i la segona corresponent a la component electrònica. Aquest informe inclou la part mecànica del projecte i, per tant, s'inclouran les parts de l'informe complementari que es creguin necessàries per presentar una explicació clara.

1.5. Context en les línies de recerca i transferència de coneixement del Tecnocampus

Aquest projecte ens servirà per aplicar varis dels coneixements adquirits durant els graus d'Enginyeria Mecànica i d'Enginyeria Electrònica Industrial i Automatització i ampliar amb els coneixements que considerem necessaris per a complir l'objectiu.

Els coneixements adquirits durant les assignatures de Robòtica, Modelització i Simulació, Control i Simulació de Processos Industrials i Control Digital de Sistemes seran especialment útils per al disseny i sintonització del controlador PID, l'obtenció de la cinemàtica inversa i la generació de trajectòries.

2. Marc teòric

En aquest projecte buscaré crear el sistema de control per un robot industrial però, abans de començar a elaborar la part tècnica de la solució seria oportú tindre certes nocions del estat del art actual i futur de la robòtica. Això ens pot aportar idees alhora de elaborar un millor sistema de control utilitzant tècniques actualitzades.

Parlar del futur de la robòtica és una tasca summament complicada degut al ràpid i divers avanç d'aquesta branca d'estudi. Tot i així, si que es possible observar certes tendències que agafaran potència durant els propers anys:

Incorporació de robots amb IA

Fins fa relativament poc els robots eren definits com a sistemes autònoms programables capaços de realitzar determinades accions i tasques per les quals han estat dissenyats. Aquestes tasques eren realitzades seguint el trinomi de funcionament percepció – decisió – acció segons el qual el robot era capaç de comprendre el seu entorn mitjançant sensors, decidir l'acció que s'ha de prendre segons la seva programació i executar-la. Això permet executar tasques senzilles en entorns inestables cosa que les màquines més senzilles no serien capaces [1] [2].

Però, degut al desenvolupament accelerat de la intel·ligència artificial durant els últims anys es podria considerar que aquest trinomi està cada cop més obsolet. Això es degut a la capacitat d'aprenentatge que proporciona la intel·ligència artificial el qual ens permet poder influenciar en l'etapa de decisió

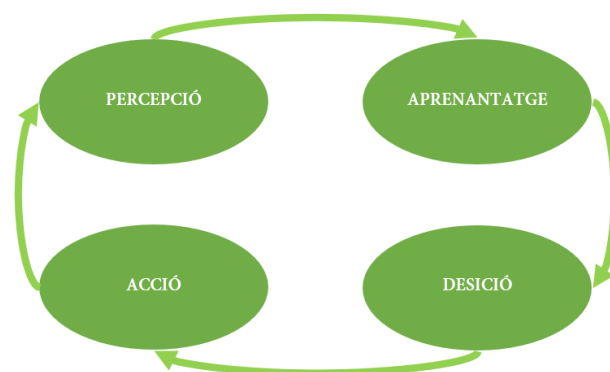


Figura 2.1: Quatrinomi robot amb capacitat

a partir de la informació reunida en l'etapa de percepció al llarg de les diferents iteracions i per la informació proporcionada externament de manera totalment autònoma.

Aquest nou polinomi de funcionament permet la realització de tasques cada cop més complexes i en entorns cada cop més inestables. Un exemple de robots on aquest sistema de funcionament està àmpliament estès són els bots virtuals com, per exemple, els chatbots o els bots encarregats de reconeixement facial o d'imatges. En el cas de, per exemple, els chatbots s'utilitzen les dades que reuneixen durant totes les conversacions online que mantenen amb els diferents usuaris i les utilitzen per a crear millors i més coherents respostes [3].

Aquest sistema de funcionament també pot ser introduït en el àmbit dels robots industrials. L'aplicació de la intel·ligència artificial en el àmbit industrial podrà aportar els següents efectes:

- + Millora en el manteniment
- + Realització de tasques més complexes
- + Millora en el sistema productiu (+ **Ràpid, + Precís**)
- + Menys productes defectuosos
- Major necessitat de computació
- Disseny exponencialment més complicat

Però, tot i que aquesta tecnologia ja s'està aplicant en un seguit de àmbits, el estat de la tecnologia no està suficientment desenvolupat per a poder ser utilitzat a gran escala i sobretot, en àmbits que requereixen gran precisió com és la indústria o en l'autoconducció.

Robots col·laboratius

Dins del món industrial hem pogut observar cada cop més freqüentment l'aparició de robots col·laboratius. Aquests consisteixen en robot capaços de treballar en harmonia juntament amb els operaris. Això crea una cadena de producció ininterrompuda en que el robot realitza les tasques repetitives que requereixin força i precisió i el operari aporta destresa i Know-How per identificar i resoldre problemes.

Els robots col·laboratius han permès crear cadenes de producció cada cop més compactes, eficients i automatitzades, relegant la feina dels operaris a feines que requereixen més creativitat o a feines de supervisió. Això ha permès augmentar la velocitat de producció, augmentar l'eficiència, automatitzar grans parts de la producció i reduir els accidents laborals a l'empresa.

La ràpida propagació dels cobots ha creat un nou entorn de treball on els operaris treballen conjuntament amb els robots. Aquesta convivència però, pot ser nociva pels treballadors en els casos de que la gestió de la càrrega del treball no es gestioni degudament. En el cas de que es porti una gestió inadequada es pot donar el cas en que ens deixem portar per el augment de la productivitat proporcionada pels cobots i es comenci a tractar els treballadors com si poguessin mantenir la mateix rendiment que els robots. Aquest cas no és hipotètic ja que es pot observar un exemple d'ell en els escàndols recents que s'han pogut veure a Amazon. En aquests escàndols s'ha pogut observar com s'imposava als treballadors de magatzem i conductors d'Amazon ratis de treball que només podria assolir un robot. Això ha tingut com a conseqüència condicions insalubres i perilloses amb constants accidents amb els robots que circulen pels magatzems o sent obligats a no agafar temps de descans ni tan sols per evacuar [5] [7] [8].

Tendències futures

Degut a la amplia varietat de robots que existeixen les tendències de disseny de robots són igualment de nombroses. Un exemple molt curiós de una d'aquestes tendències és el disseny de robots cada cop més *suaus*. Aquests robots *suaus* són robots que s'inspiren en sers biològics, ja siguin animals, plantes o propis humans, per a dissenyar estructures deformables, suaus i a la vegada resistents i capaces de resistir grans carregues. Utilitzant aquesta base d'inspiració s'estan dissenyant nous tipus d'estructures, articulacions i fins i tot sistemes motrius que simulen el funcionament de músculs [4] [6].

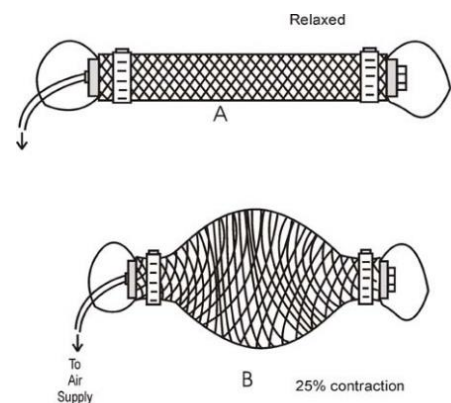


Figura 2.2 Músculs robòtics sistemes motrius que simulen el

Tot i que existeixen moltes tendències en l'etapa de disseny la tendència més interessant la trobem quan observem la robòtica des d'un punt de vista global. Observant diferents estadístiques podem observar que la venda i ús de robots està augmentant exponencialment especialment entre els robots col·laboratius. Segons el informe de World Robotics 2021 els robots industrials utilitzats en les empreses ha augmentat en un 10 % i segons les previsions de l'evolució de l'ús de robots en els propers 5 anys el augment serà de 37.5 % [9] [11].

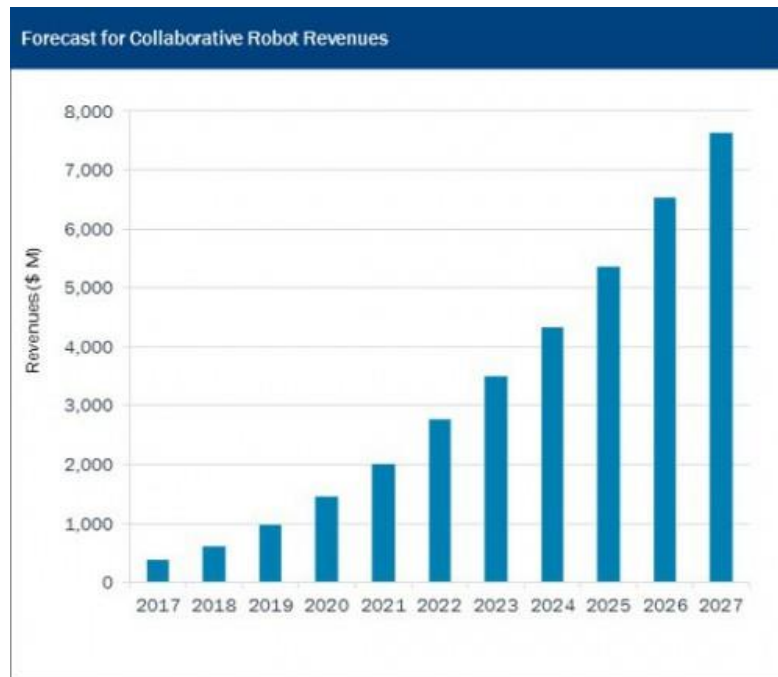


Figura 2.3 Previsió del ús de robots col·laboratius

Aquest augment implica que cada cop hi ha un major nombre de tasques que s'estan automatitzant el qual pot implicar un risc molt real de que cada cop més gent es trobi sense feina [5]. Això passarà especialment en el cas dels treballadors que realitzin tasques que no requereixin creativitat com, per exemple els operaris. Aquest problema ja està sent discutit en varis àmbits polítics i des de diferents punts de vista. Per una banda, es poden observar opinions de polítics com Donald Trump el qual considerava que s'havia de limitar l'automatització en les fàbriques per tal de mantenir el treballs pocs qualificats que en cas contrari es perdrien [12]. Per l'altre banda, trobem l'opinió de persones a favor de l'automatització i que la solució no resta en decidir si s'ha de limitar o no aquesta sinó trobar solucions a nivell estatal per facilitar la vida en un possible món del futur on els treballs poc qualificats siguin obsolets. Un exemple molt recent el trobem en una entrevista d'Elon Musk el qual sostenia que en un futur la gran majoria de les tasques repetitives serien completament automatitzades i que per tant molts treballs es perdrien. Degut això, Elon Musk proposava la implementació d'una Renta Bàsica Universal la qual provocaria un canvi en la nostra relació amb el treball ja que aquest passaria a ser opcional [10].

3. Metodologia del projecte

La metodologia que s'ha seguit per desenvolupar aquest projecte ha consistit en les següents etapes:

3.1. Etapa d'Anàlisi:

Durant l'etapa d'anàlisi es realitzarà un estudi de l'objecte del projecte i com s'assolirà l'objectiu d'aquest. Durant aquesta etapa s'elaborarà un Roadmap en el qual representarem les tasques que s'hauran de dur a terme per tal d'aconseguir el nostre fi. Aquesta etapa correspondrà als apartats del marc teòric i els objectius tècnics.

3.2. Etapa d'Avaluació d'alternatives:

Durant l'etapa d'avaluació d'alternatives es recercherà la informació necessària per a poder executar les diferents tasques del Roadmap. En aquesta etapa no es buscarà una única solució ni s'intentarà trobar la millor solució possible. En lloc d'això, es buscarà totes les alternatives possibles per a solucionar les diferents tasques que s'han dissenyat en el Roadmap i s'estudiarà el procés que es seguirà per cada una de les possibilitats. Els resultats d'aquesta etapa es poden observar en els subapartats de Possibilitats de càlcul a la viabilitat tècnica.

3.3. Etapa de Disseny de la solució:

Durant l'etapa de disseny de la solució es consideraran totes les alternatives trobades en l'etapa d'avaluació d'alternatives i se seleccionaran les opcions més adequades per a la nostra situació. Aquesta etapa la podem observar en els subapartats de *Selecció i desenvolupament de la solució* en l'apartat de la viabilitat tècnica.

3.4. Etapa de Verificació de la solució:

Durant l'etapa de verificació de la solució es realitzaran les comprovacions corresponents per a verificar que la solució escollida és adient i presenta una resposta acceptable. Els resultats d'aquesta etapa es podran observar en part en els subapartats de *Selecció i desenvolupament de la solució* en l'apartat de la viabilitat tècnica i en les simulacions que es duran terme per comprovar el correcte funcionament del sistema.

4. Objectius tècnics

L'objectiu final d'aquest projecte és el disseny d'un sistema de control ràpid i el més estable possible per a un robot manipulador de 6 eixos. El projecte no contempla el disseny del robot manipulador de 6 eixos, en el seu lloc s'escollirà un robot industrial de 6 eixos i es trobarà les seves característiques mecàniques a partir d'un estudi de les seves articulacions i les característiques proporcionades per la fitxa tècnica del manipulador.

Els passos que s'hauran de complir per assolir l'objectiu final projecte són els següents els quals es dividiran en els dos informes del projecte:

Mecànica

- Trobar la cinemàtica del manipulador robòtic.
- Trobar la dinàmica del manipulador robòtic.

Electrònica

- Dissenyar un bloc de generació de trajectòries.
- Dissenyar un controlador PID amb un “% Overshoot” menor a 5 % i un temps d'estabilització el més proper al marcat en la trajectòria.

Adicionalment, es realitzaran una sèrie de simulacions per a verificar el correcte funcionament del sistema controlador executarem un seguit de simulacions. En aquestes simulacions es comprovarà el correcte disseny dels blocs de dinàmica i cinemàtica i el correcte funcionament del sistema de control global del manipulador robòtic.

5. Especificacions tècniques

El robot que s'utilitzarà serà un TM14X del fabricant OMRON. El TM14X és la variant sense càmera del robot col·laboratiu TM14 el qual li permet més rang de moviment en algunes de les articulacions. El robot col·laboratiu té una estructura de robot universal com es pot observar en la figura 5.1. El TM14X té un rang operatiu d'1,1 metres i permet aixecar fins a 14 kg de càrrega.



Figura 5.1 Robot del tipus universal

El rang d'operació i velocitats màximes de cada articulació varia segons l'articulació i podem veure les limitacions en la següent taula:

Articulació	Rang d'operació	Velocitat màxima
1	$\pm 360^\circ$	120°/s
2	$\pm 360^\circ$	120°/s
3	$\pm 163^\circ$	180°/s
4	$\pm 360^\circ$	150°/s
5	$\pm 360^\circ$	150°/s
6	$\pm 360^\circ$	180°/s

Taula 5.1 Limitacions robot TM14X

La resta de dades tècniques així com les seves dimensions es podran consultar en la fitxa tècnica de l'Annex.

6. Plantejament inicial

Termes importants

Robot col·laboratiu: Robot dissenyat amb la finalitat d'automatitzar processos per tal de millorar l'eficiència de la producció, millorar l'eficàcia dels procediments i substituir els operaris en les tasques o en les parts dels projectes que puguin ser perilloses o nocives per l'operari. Addicionalment, també disposa de la capacitat de treballar conjuntament amb un operari humà o complementar el treball que aquest realitza.

Manipulador: Robot dissenyat per, principalment, realitzar tasques de desplaçament i classificar.

Trames: Les trames consisteixen en un sistema de coordenades que utilitzarem per definir la posició i orientació de cada articulació.

Articulació: Les articulacions són els elements que uneixen els diferents enllaços. Generalment, seran de 1 o més graus de llibertat, però, també és possible tenir casos on l'articulació és fixa i per tant no té cap grau de llibertat.

Enllaç: Quan ens referim a enllaços ens referirem als cossos que componen els diferents elements sense cap grau de llibertat que uneixen les diferents articulacions del robot.

Graus de llibertat (GDL): Els graus de llibertat consisteixen en cada un de les coordenades independents de cada articulació que cal definir per tal de descriure la posició i orientació dels seus elements. En el cas d'un braç robòtic cada articulació generalment permetrà un tipus de moviment, rotació o desplaçament, tot i que alguns tipus d'articulació poden permetre ambdós com és el cas d'una articulació cargol com es veu a continuació.

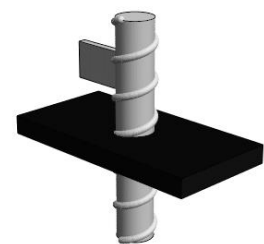


Figura 6.1 Articulació cargol de 2 GDL

Matriu de rotació bàsica: La matriu de rotació bàsica consisteix en la obtenció de una matriu per representar els canvis en l'orientació degut al gir sobre un eix.

Matriu de transformació: La matriu de transformació consisteix en la multiplicació de varies matrius de rotació bàsiques per tal d'obtenir una trama amb l'orientació desitjada.

Matriu de transformació homogènia: La matriu de transformació homogènia consisteix en una matriu que conté el vector de desplaçament i la matriu de transformació el qual ens permetrà definir el desplaçament i rotació que es vol aplicar a l'eina.

Variables tipus struct: Les variables struct són el tipus de variable que utilitzarà el solver que implementarem per trobar la cinemàtica inversa. Aquest tipus de variables consisteixen en una variable que genera una estructura la qual conté diferents llistes classificades segons temes. Addicionalment es poden crear subtemes però, en aquest cas no s'utilitzaran.

En aquest apartat estudiarem el procés que haurem de seguir per dissenyar el sistema de control. El sistema de control consistirà en 4 blocs principals:

- Bloc de cinemàtica (Directa o inversa)
- Bloc de la dinàmica
- Bloc de Generació de trajectòries
- Bloc del controlador

En aquest informe es tractaran els blocs que corresponen a la cinemàtica i a la dinàmica ja que són els blocs més relacionats amb la mecànica. En el projecte complementari el qual tracta la part electrònica del sistema de control es tractaran els blocs de la generació de trajectòries, el bloc del controlador i la realització de una simulació per comprovar el funcionament del controlador que hem creat.

6.1. Cinemàtica

El càlcul de la cinemàtica d'un robot consisteix en una representació matemàtica dels moviments del robot sense tenir en compte les forces que hi poden actuar i causar el moviment.

6.1.1. Possibilitats de càlcul

Per a calcular el bloc corresponent a la cinemàtica tenim les possibilitats de prendre la via de la cinemàtica directa o la via de la cinemàtica inversa. Depenent de la opció que triem haurem de modificar el sistema de control que dissenyem ja que les dades que necessita i el que retorna és diferent en cada cas.

6.1.1.1. Cinemàtica Directa

La cinemàtica directa consisteix en l'obtenció de la posició i orientació de l'eina del robot a partir del coneixement de les posicions angulars de cada un dels enllaços així com el coneixement d'altres paràmetres fixos.

Per trobar la cinemàtica inversa primer haurem de calcular les diferents transformacions d'enllaç i trobarem la transformació del sistema multiplicant matricialment les diferents transformacions d'enllaç. D'aquesta manera aconseguirem les equacions cinemàtiques que ens permetran calcular la posició i orientació de l'eina del robot respecte a la posició de la base del robot el qual representa la trama 0 [14] [19].

6.1.1.2. Cinemàtica Inversa

La cinemàtica inversa consisteix en l'obtenció de les diferents paràmetres que ens permetin identificar una possible configuració de les rotacions dels enllaços per que l'eina sigui capaç de arribar a una posició i orientació especificada. Poder trobar la cinemàtica directa sol ser relativament més complicat que obtenir la cinemàtica directa, ja que pot presentar diverses solucions acceptables o, fins i tot, no presentar cap solució possible i que sigui irresoluble. . Existeixen diverses maneres de calcular la cinemàtica inversa com l'aplicació de mètodes geomètrics o mètodes algebraics, però, en el cas que optem per utilitzar la cinemàtica inversa en lloc de la directa el mètode més senzill d'aplicar és aplicant matrius de transformació homogènies. Per aplicar aquest sistema de càlcul de la cinemàtica inversa primer s'haurà d'aconseguir el sistema cinemàtic directe de manera que s'hagi obtingut les relacions entre la posició i orientació de l'eina segons les posicions angulars de les articulacions del robot. Un cop s'hagi aconseguit les equacions cinemàtiques directes podrem obtenir les equacions del sistema cinemàtic invers aplicant les matrius de transformació homogènies [13] [14] [19].

6.1.2. Selecció i desenvolupament de la solució

Per calcular el bloc de la cinemàtica ens hem decantat per utilitzar la cinemàtica inversa utilitzant el mètode de les matrius de transformació homogènies, ja que a l'hora de dissenyar el sistema de control del robot ens permetrà dissenyar un sistema més clar i ordenat.

Per calcular la cinemàtica directa primer haurem de tenir l'orientació de les trames en cada un dels enllaços. Ja que treballarem amb un robot ja existent les diferents trames solen estar representades a la fitxa tècnica proporcionada pel fabricant.

Un cop tenim les diferents trames haurem de crear una matriu per representar els paràmetres de vincle. Aquesta matriu tindrà els següents 4 paràmetres:

$$a_i = \text{distància entre les trames } \hat{Z}_i \text{ i } \hat{Z}_{i+1} \text{ mesurada sobre } \hat{X}_i$$

$$\alpha_i = \text{distància entre les trames } \hat{Z}_i \text{ i } \hat{Z}_{i+1} \text{ mesurada sobre } \hat{X}_i$$

$$d_i = \text{distància entre les trames } \hat{X}_i \text{ i } \hat{X}_{i+1} \text{ mesurada sobre } \hat{Z}_i$$

$\theta_i = \text{distància entre les trames } \hat{X}_i \text{ i } \hat{X}_{i+1} \text{ mesurada sobre } \hat{Z}_i$

Un cop tinguem la matriu amb tots els paràmetres de vincle utilitzarem la següent fórmula per trobar les transformacions individuals homogènies de vincle:

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_i - 1 \\ \sin\theta_i * \cos\alpha_{i-1} & \cos\theta_i * \cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1} * d_i \\ \sin\theta_i * \sin\alpha_{i-1} & \cos\theta_i * \sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1} * d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.1)$$

Finalment, en el moment en què haguem calculat totes les transformacions individuals les multiplicarem matricialment per tal d'obtenir la transformació que relacioni la trama 0 amb la trama corresponent a la posició de l'eina del robot. Un cop haguem seguit aquests passos ja hauréem aconseguit les equacions de la cinemàtica directa.

Tal com hem comentat abans, utilitzarem la cinemàtica inversa la qual obtindrem utilitzant les matrius de transformació homogènies. Per obtenir les equacions de la cinemàtica inversa seguint aquest mètode hauréem de realitzar els següents passos:

- 1. Partim de la següent expressió.** Partim de l'operació realitzada anteriorment en que realitzàvem la multiplicació matricial de les diferents trames individuals.

$${}^0T = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad (6.2)$$

Sent T igual a:

$$T = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^0R \\ {}^0p \\ 0 & 1 \end{bmatrix} \quad (6.3)$$

Sent 0R la matriu de rotació des de la posició inicial de l'eina fins la posició desitjada i 0p les coordenades a la que es vol moure l'eina.

2. **Aïllem θ_1 .** Aïllem 0T de la següent manera:

$$({}^0T)^{-1} T = {}^1T {}^2T {}^3T {}^4T {}^5T \quad (6.4)$$

$$({}^0T)^{-1} \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^1T {}^2T {}^3T {}^4T {}^5T \quad (6.5)$$

Un cop haguem realitzat el canvi en la fórmula aïllem θ_1 segons els vectors n , o , a i p per trobar la primera equació de la cinemàtica inversa. Aquest pas pot tenir més d'un resultat així que en cas que existeixin més d'un resultat escollirem el que es consideri més adequat.

3. **Repetim el pas 2 per la resta de casos excepte per la trama 5T .** Repetirem el pas anterior aïllant la trama i trobant la θ_i en qüestió fins que quan aïllem una trama només ens quedi una trama sola a la dreta. Durant aquest pas trobarem θ_2 , θ_3 i θ_4 .
4. **Aïllem θ_5 i θ_6 .** Finalment seguirem el procediment que hem seguit en el pas dos però en aquest cas calcularem els dos últims valors.

Després d'aïllar totes les variables θ_i haurem obtingut totes les diferents equacions de la cinemàtica inversa el qual ens permet calcular els valors d'aquestes variables introduint les coordenades a la que es vol moure l'eina. En el cas que també es volgués rotar l'eina es donaria valor a la matriu de rotació que formen els vectors \vec{n} , \vec{s} i \vec{a} .

La metodologia i les fórmules que s'han utilitzat s'han adaptat a partir de les següents fonts d'informació: [14] [19]

6.2. Dinàmica

El càlcul de la dinàmica de un robot consisteix en la representació matemàtica de la relació entre les forces i el moviment del robot.

6.2.1. Possibilitats de càlcul

Per calcular el bloc de la dinàmica del sistema tenim la possibilitat de tirar per dues vies de càlcul:

6.2.1.1. Newton-Euler

Primerament, tenim la metodologia de càlcul més coneguda de Newton-Euler. Utilitzant aquesta metodologia té el desavantatge de calcular els moviments lineals i els rotacionals de manera separada, però els càlculs realitzats són molt més senzills, ja que la complexitat de computació no escala segons el nombre de graus de llibertat. Per tant, si desitgèssim utilitzar la metodologia Newton-Euler partiríem de les següents fórmules per començar a calcular tant els moviments lineals com els rotacionals.:

$$F_i = m_i * a_i = m_i * \dot{v}_i = m_i * \ddot{x}_i \quad (6.6)$$

Sent F la força que actua des del centre de masses de l'enllaç i , sent m_i la massa total de l'enllaç i i sent a_i , \dot{v}_i i \ddot{x}_i l'acceleració lineal de l'enllaç i expressat de diferents maneres.

$$\begin{aligned} M_{xi} &= I_{xi} * \dot{w}_{xi} + (I_{zi} - I_{yi})w_{zi} * w_{yi} \\ M_{yi} &= I_{yi} * \dot{w}_{yi} + (I_{xi} - I_{zi})w_{xi} * w_{zi} \\ M_{zi} &= I_{zi} * \dot{w}_{zi} + (I_{yi} - I_{xi})w_{yi} * w_{xi} \end{aligned} \quad (6.7)$$

Sent M el moment de l'enllaç, sent I el tensor d'inèrcia de l'enllaç en qüestió i sent \dot{w}_i i w_i l'acceleració i velocitat de rotació respectivament.

Resumint, el càlcul de la dinàmica utilitzant Newton-Euler ens permetrà calcular la dinàmica amb una menor càrrega computacional, ja que treballa amb operacions vectorials. Tot i això, a l'haver de calcular els moviments lineals i rotacionals de manera separada pot comportar que tinguem dificultats quan hagi de calcular els 6 enllaços [19]. Això és degut al fet que, mentre és veritat que la càrrega computacional és menor, perquè estic treballant amb un objecte de 6 enllaços tindrà una gran quantitat d'equacions utilitzant aquest mètode el qual fa que sigui més complex d'organitzar.

Per tant, es pot assumir que a causa del nombre d'equacions amb les quals haurem de treballar i que funcionaran de manera independent d'una amb l'altra **possibilitat de que aparegui error a causa de l'error humà augmentin.**

6.2.1.2. Lagrange

Per l'altra banda tenim **la formulació lagrangiana** la qual permet obtenir la dinàmica del robot a partir del càlcul de l'energia cinètica i potencial. Això ens permet poder calcular els moviments lineals i rotacionals conjuntament a diferència de la formulació segons Newton-Euler. Però, això té l'inconvenient de què la complexitat de l'operació augmentarà exponencialment a la quarta. Això implicarà un augment en les operacions de càlcul, però seran operacions molt més clares i més fàcils de tractar.

Per a poder calcular la dinàmica mitjançant Lagrange primer haurem de trobar el Lagrangiana el qual obtenim quan restem l'energia potencial a l'energia cinètica.

$$L_i = Ec_i - Ep_i \quad (6.8)$$

I, a continuació trobaré les equacions del moviment de la següent manera:

$$\frac{d}{dt} \frac{\partial L_i}{\partial \dot{\theta}_i} - \frac{\partial L_i}{\partial \theta_i} = \tau_i \quad (6.9)$$

El qual podem simplificar de la següent manera:

$$\frac{d}{dt} \frac{\partial Ec_i}{\partial \dot{\theta}_i} - \frac{\partial Ec_i}{\partial \theta_i} + \frac{\partial Ep_i}{\partial \theta_i} = \tau_i \quad (6.10)$$

Sent τ_i el moment de torsió de l'enllaç i sent $\dot{\theta}_i$ i θ_i la velocitat d'angular i la posició angular respectivament de l'enllaç.

Per tant tenim que la formulació lagrangiana ens permetrà organitzar millor les diferents fórmules però, en canvi s'haurà de fer un major nombre de càlculs. Tot i això, ja que només estarem treballant amb 3 graus de llibertat la diferència entre el nombre d'operacions amb les que haurem de treballar en cada cas és molt petita [16] [17] [19].

6.2.2. Selecció i desenvolupament de la solució

Finalment, ens hem decantat per la formulació lagrangiana, ja que en treballar en un sistema de 6 graus de llibertat l'increment exponencial de la complexitat del sistema de càlcul no ens afecta en gran manera. En el cas que estiguem calculant un sistema amb més graus de llibertat com pot ser el cas d'un dron aeri que pot arribar a treballar amb més de 6 graus de llibertat sí que seria convenient treballar amb la formulació de Newton-Euler.

El procediment que seguirem per a calcular la dinàmica en cada enllaç serà el següent:

- 1. Càlcul de la energia cinètica.** Per a generalitzar el càlcul de l'energia cinètica utilitzarem la següent equació general de la cinètica separant les parts corresponents a la translació i a la rotació.

$$Ec_i = Ec_{tra} + Ec_{rot} = \frac{1}{2} m * v^T * v + \frac{1}{2} m * w^T * I * w \quad (6.11)$$

Sent $v^T * v$ la multiplicació del vector de velocitat lineal transversal amb el vector velocitat per obtenir la velocitat al quadrat, sent w la velocitat angular i sent I la matriu d'Inèrcia el qual calculem de la següent manera:

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \quad (6.12)$$

Podent calcular els elements de la matriu de la següent manera:

$$\begin{aligned}
 I_{xx} &= \int_0^x \int_0^y \int_0^z (y^2 + z^2) \rho dv \\
 I_{yy} &= \int_0^x \int_0^y \int_0^z (x^2 + z^2) \rho dv \\
 I_{zz} &= \int_0^x \int_0^y \int_0^z (x^2 + y^2) \rho dv \\
 I_{xy} &= \int_0^x \int_0^y \int_0^z xy \rho dv \\
 I_{xz} &= \int_0^x \int_0^y \int_0^z xz \rho dv \\
 I_{yz} &= \int_0^x \int_0^y \int_0^z yz \rho dv
 \end{aligned} \tag{6.13}$$

En el cas que l'origen de coordenades s'ubiqués en el centre de masses de l'objecte, tots els termes excepte la diagonal serien igual a 0.

2. Càlcul energia potencial. Per a calcular l'energia potencial utilitzarem la següent fórmula:

$$E_{p_i} = m * g * \Delta h \tag{6.14}$$

Sent m la massa de l'enllaç, g la gravetat (per la constant gravitacional utilitzarem un valor de 9,81), Δh la diferència entre l'alçada a la qual es troba el centre de massa amb una altura de referència arbitrària. En aquest cas, l'altura de referència que agafarem serà la base de l'espai de treball on està situat el robot.

3. Obtenim el Lagrangia. Per obtenir el lagrangia simplement haurem de restar l'energia potencial a l'energia cinètica. Perquè organitzar millor els diferents càlculs dels lagrangians sumarem les energies cinètiques i les energies potencials en dos vectors, un per la suma de totes les energies cinètiques i un per la suma de totes les energies potencials.

$$Ec = \sum_{i=1}^n Ec_i$$

$$Ep = \sum_{i=1}^n Ep_i \quad (6.16)$$

$$L = Ec - Ep \quad (6.17)$$

4. Apliquem la formulació de Lagrange. A continuació aplicarem la fórmula mencionada anteriorment per a trobar les equacions del moviment:

$$\frac{d}{dt} \frac{\partial L_i}{\partial \dot{\theta}_i} - \frac{\partial L_i}{\partial \theta_i} = \tau_i \quad \text{o} \quad \frac{d}{dt} \frac{\partial Ec_i}{\partial \dot{\theta}_i} - \frac{\partial Ec_i}{\partial \theta_i} + \frac{\partial Ep_i}{\partial \theta_i} = \tau_i \quad (6.18)$$

En el cas que es volgués aplicar una força de fregament o es volgués aplicar una càrrega es restarà al moment de torsió com volem veure a continuació:

$$\frac{d}{dt} \frac{\partial L_i}{\partial \dot{\theta}_i} - \frac{\partial L_i}{\partial \theta_i} = \tau_i - \vec{F} * \frac{\partial \vec{r}_i}{\partial \theta_i} \quad (6.19)$$

Sent \vec{F} el vector força i r_i el vector posició.

Un cop haguem calculat les diferents derivades ja haurem obtingut les equacions del moviment del nostre robot.

5. Obtenir la matriu de masses. Per obtenir la matriu de masses s'utilitzaran tots els termes que multipliquen a $\ddot{\theta}$ i els termes que multipliquen a θ .

6. Obtenir el terme de velocitat. Per obtenir el terme de velocitat utilitzarem tots els elements que siguin dependents de la velocitat ($\dot{\theta}$).

Tot els termes que continguin el producte de velocitats representaran la força de Coriolis, mentre que tots els termes que continguin la velocitat al quadrat d'un enllaç correspondran a la força centrífuga.

7. Obtenir el terme gravetat. Finalment, per obtenir el terme de gravetat utilitzarem tots els termes que continguin la constant de gravitacional.

La metodologia i les fórmules emprades en aquest apartat s'han adaptat a partir de les següents fonts d'informació: [15] [17] [18] [19].

6.3. Simulació

Per observar el funcionament del robot segons la cinemàtica i la dinàmica trobades es realitzaran dues simulacions utilitzant el programa de Matlab.

6.3.1. 1era Simulació

La primera simulació consistirà en una simulació utilitzant el Simulink per observar l'evolució les variables θ , $\dot{\theta}$, $\ddot{\theta}$ i τ les quals obtindrem a l'implementar els blocs de cinemàtica i dinàmica. Mitjançant aquestes gràfiques podrem observar el comportament dels diferents paràmetres i podrem observar la diferència amb els resultats que obtindrem un cop apliquem el control de posició (PID).

Per introduir els conjunts d'equacions que representin la cinemàtica i la dinàmica obtingudes utilitzarem dos blocs de *Function Matlab* els quals programarem perquè funcionin segons les equacions trobades.

6.3.2. 2na Simulació

En la segona simulació utilitzarem la llibreria de Matlab Robotic System Toolbox per representar d'una manera simplificada el robot en una vista 3D de manera que a partir de la cinemàtica obtinguda podrem observar el comportament d'aquest. Tot i que la informació que ens aporta és menor que en la 1a simulació, ja que només es té en compte la cinemàtica, la representació en 3D pot suposar un ajut visual per entendre els diferents moviments que aquest pot realitzar. Posteriorment, aquest model creat ens servirà per a calcular altres paràmetres del robot.

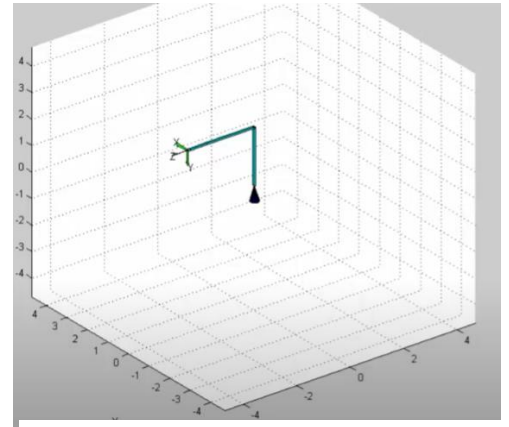


Figura 6.2 Exemple simulació

Per realitzar aquesta simulació també s'haurà d'introduir límits a la rotació que poden efectuar les diferents articulacions. Aquesta informació normalment es pot trobar en la fitxa tècnica del robot, però, en el cas que no es disposés de la informació s'introduirien uns límits escollits segons el nostre criteri personal. També s'hauran d'establir unes toleràncies les quals les haurem d'escollir segons el nostre criteri.

7. Desenvolupament de la solució

En aquest apartat utilitzarem la informació i els procediments recercats i proposats en l'avantprojecte i els aplicarem en la realització dels diferents estudis necessaris per a la simulació i disseny del controlador del robot.

Tal com s'ha mencionat anteriorment el sistema de simulació del controlador contindrà quatre blocs principals. Els blocs de la cinemàtica inversa i la generació de trajectòries seran els primers blocs que s'hauran de dissenyar i realitzaran la funció de traducció de les coordenades de la posició i orientació desitjades de l'eina i dissenyar la trajectòria que els diferents enllaços hauran de seguir. A continuació es dissenyarà el bloc de la dinàmica el qual farà la funció de simular la relació entre les forces que s'apliquen en el robot, tant internes com externes, i la posició i velocitat reals de les diferents articulacions. Un cop tenim la trajectòria dissenyada i un bloc perquè simularà el robot físic i proporcionarà la posició i velocitat reals podrem començar a dissenyar els controladors que s'utilitzaran de manera independent en cada una de les articulacions.

7.1. Paràmetres Denavit-Hartenberg

Tal com s'ha mencionat anteriorment en l'avantprojecte la cinemàtica inversa consisteix en l'obtenció de la posició angular de les articulacions a partir de la posició i orientació de l'eina proporcionades. Això ens permetrà obtenir els angles que cada un de les articulacions hauran d'adoptar per arribar a la posició desitjada.

Per començar el disseny del controlador primer hem establert una posició base per al robot col·laboratiu TM14X. Aquesta posició inicial s'ha establert per facilitar la posterior obtenció dels paràmetres de Denavit-Hartenberg (a partir d'aquest punt ens referirem a paràmetres DH quan els vulguem mencionar). Aquesta posició serà la posició original des d'on partirà el robot durant el primer moviment.

Abans de començar a obtenir els paràmetres de DH haurem d'establir els tres eixos en cada una de les trames de les diferents articulacions.

El primer eix que establirem serà l'eix Z_i , ja que aquest són els més fàcils de deduir, ja que coincidirà amb l'eix de rotació o de desplaçament de cada articulació.

A continuació, situarem els eixos X_i . Per deduir aquests eixos ens podem trobar en tres casos. El primer cas que ens podem aconseguir és quan els vectors Z_i i Z_{i-1} no siguin paral·lels en el qual l'eix X_i el definirem segons la recta normal comuna entre els dos eixos. En el segon cas tindrem el cas en què els vectors Z_i i Z_{i-1} son paral·lels en el qual l'eix X_i serà donat també per la normal comuna, però, ja que en ser paral·lels el nombre de normals comunes és infinit escollirem la que més ens convingui.

Finalment, ens quedarà situar l'eix Y_i el qual trobarem fen ús de la regla de la mà dreta col·locant el polze en el sentit de l'eix Z_i , l'índex en el sentit de X_i i el dit del mig o cor correspondrà al sentit de l'eix Y_i .

Per situar la primera trama o trama base es podrà utilitzar el criteri que es consideri oportú per a obtenir la disposició dels enllaços que permeti una obtenció dels paràmetres DH més sencilla.

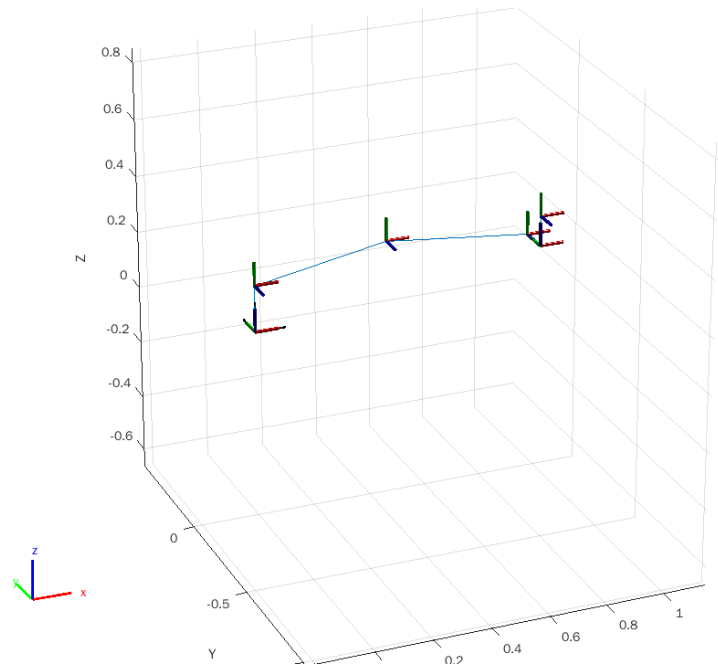


Figura 7.1 Robot TM14X en la posició inicial

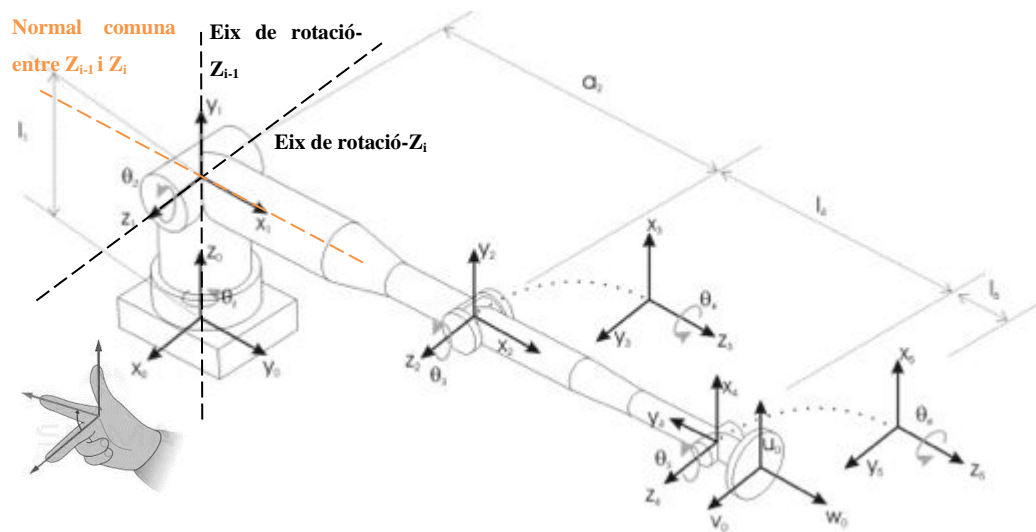


Figura 7.2 Exemple del procediment per situar trames

Un cop hem establert els diferents eixos de cada una de les articulacions obtindrem les trames que es poden observar en la figura 7.1. Com es pot veure, seguint els passos anteriors obtenim un total de 7 trames, 1 trama per cada articulació més la trama base la qual podem observar en la part inferior esquerra de la figura 7.2. Tal com mencionarem més endavant, es considerarà afegir una trama addicional amb una articulació fixa per tenir en compte la distància final entre la trama final i la punta de l'eina la qual no s'ha tingut en compte inicialment. Inicialment, no s'ha tingut en compte aquesta distància com que el solver que s'utilitzarà per trobar la cinemàtica inversa requereix que els paràmetres DH siguin introduïts amb un format que no té en compte els offsets de theta en les articulacions giratòries.

A partir d'aquestes trames podrem obtenir els paràmetres DH (a partir d'aquest punt ens referirem a paràmetres DH quan els vulguem mencionar). Els paràmetres DH s'utilitzen per definir la geometria i dimensions dels diferents enllaços del braç robòtic a partir de només 4 paràmetres. Aquests paràmetres són els següents:

- Paràmetre a : El paràmetre a correspon a la distància entre la trama i i la trama $i-1$, mesurada al llarg de l'eix x_i fins la intersecció amb l'eix z_{i-1} .
- Paràmetre α : El paràmetre α correspon al angle entre els eixos z_{i-1} i z_i mesurat des del eix x_i .
- Paràmetre d : El paràmetre d correspon a la distància entre la trama i i la trama $i-1$, mesurada al llarg de l'eix z_{i-1} fins la intersecció amb l'eix x_i .
- Paràmetre θ : El paràmetre θ correspon al angle entre els eixos x_{i-1} i x_i mesurat des de l'eix z_i .

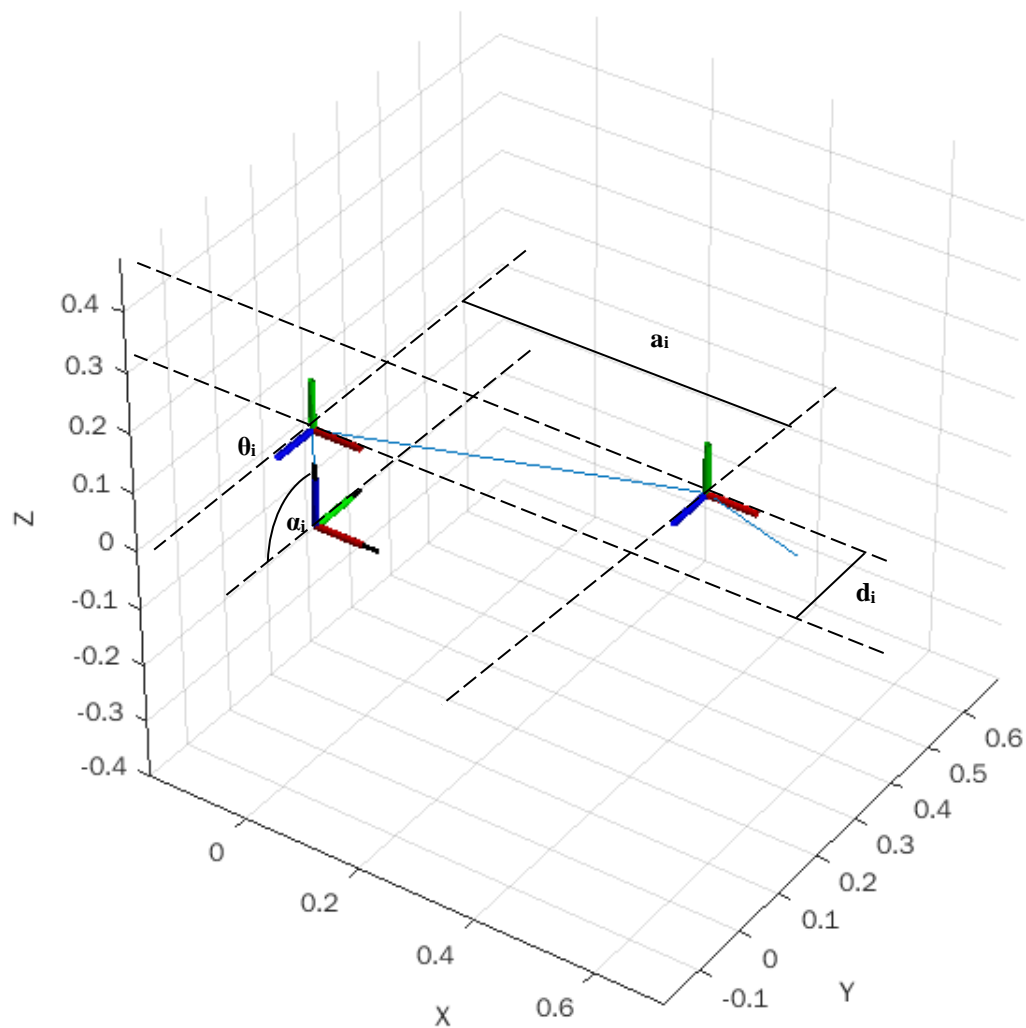


Figura 7.3 Obtenció paràmetres DH

Seguint el procediment mostrat en la figura 7.3 serà possible l'obtenció dels paràmetres DH. Existeixen múltiples possibilitats a l'hora d'obtenir els paràmetres DH depenent de la posició inicial que s'hagi emprat. Inicialment, es van obtenir els paràmetres DH seguint el procediment après en l'assignatura de Robòtica del Grau d'Enginyeria Electrònica Industrial i Automatització en el qual es tenia en consideració l' en theta. Aquesta hipòtesi inicial, així com els paràmetres obtinguts amb ella, es poden observar a continuació:

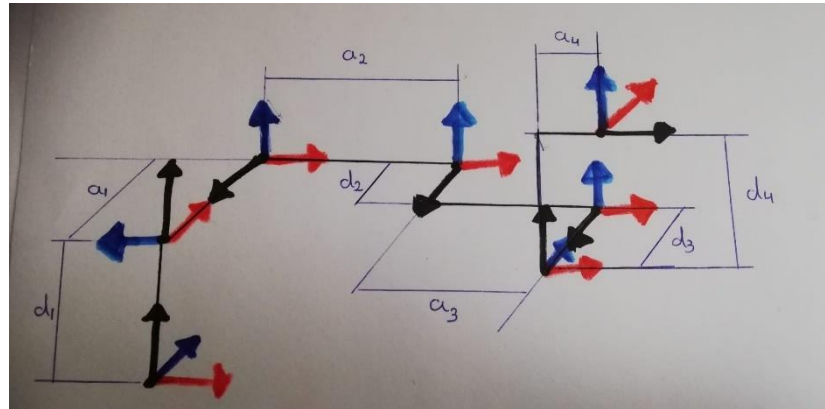


Figura 7.4 Paràmetres DH hipòtesis 1

	a_i	α_i	d_i	θ_i
1	0	0	d_1	$\theta_1 + \pi/2$
2	a_1	$\pi/2$	0	$\theta_2 - \pi/2$
3	a_2	0	0	θ_3
4	a_3	0	d_2	θ_4
5	0	$-\pi/2$	d_3	θ_5
6	a_4	$\pi/2$	d_4	$\theta_6 + \pi/2$

Taula 7.1 Paràmetres DH hipòtesis 1

Les distàncies a i d que s'utilitzaran en aquesta primera hipòtesis per obtenir els paràmetres DH són:

$$a_1 = 0,18$$

$$d_1 = 0,1652$$

$$a_2 = 0,5361$$

$$d_2 = 0,1297$$

$$a_3 = 0,4579$$

$$d_3 = 0,106$$

$$a_4 = 0,11315$$

$$d_4 = 0,106$$

Però, ja que per obtenir la cinemàtica inversa utilitzarem un solver de Matlab Online, aquest procediment no és vàlid, ja que aquest solver no té en compte l'offset en theta. Degut això hem hagut de tornar a trobar els paràmetres DH, però sense tenir en compte l'offset el qual ens donarà un resultat lleugerament diferent. En la figura i matriu a continuació podem observar els paràmetres DH i trames obtinguts a partir d'aquesta nova hipòtesis.

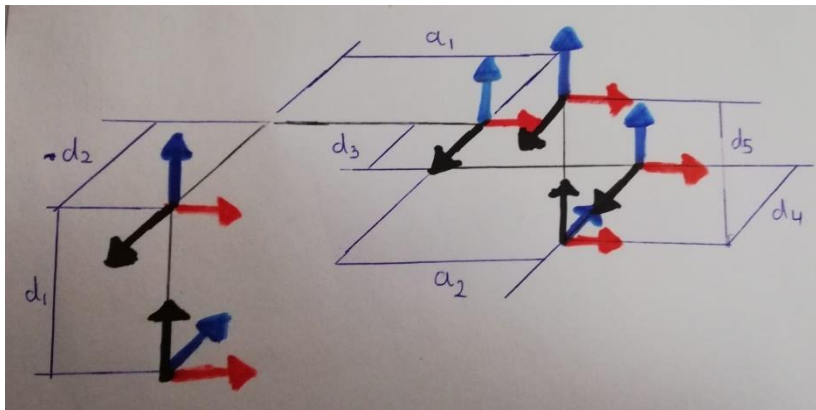


Figura 7.5 Paràmetres DH hipòtesis 2

	a_i	α_i	d_i	θ_i
1	0	0	0	θ ₁
2	0	π/2	d ₁	θ ₂
3	a ₁	0	- d ₂	θ ₃
4	a ₂	0	d ₃	θ ₄
5	0	-π/2	d ₄	θ ₅
6	0	π/2	d ₅	θ ₆

Taula 7.2 Paràmetres DH hipòtesis 2

Les distàncies a i d que s'utilitzaran en aquesta primera hipòtesis per obtenir els paràmetres DH són:

$$\mathbf{a_1} = 0,5361$$

$$\mathbf{d_1} = 0,1652$$

$$\mathbf{a_2} = 0,4579$$

$$\mathbf{d_2} = 0,18$$

$$\mathbf{d_3} = 0,1297$$

$$\mathbf{d_4} = 0,106$$

$$\mathbf{d_5} = 0,106$$

Un cop hem obtingut els paràmetres DH ja podem procedir a resoldre la cinemàtica inversa.

7.2. Cinemàtica inversa

Durant l'avantprojecte es va considerar resoldre la cinemàtica inversa mitjançant el mètode de les transformacions homogènies o el mètode geomètric. Inicialment, es va intentar aplicar els mètodes mencionats, però, a causa de la complexitat de treballar amb un robot de 6 graus de llibertat s'ha considerat utilitzar un mètode diferent.

Per obtenir la cinemàtica inversa d'una manera més senzilla hem utilitzat, tal com hem mencionat anteriorment, un solver del Matlab Online [27]. El solver en qüestió es crida usant la funció de *inverseKinematics* [29] el qual ens permetrà simular i visualitzar el robot utilitzant una representació de diferents enllaços units mitjançant les articulacions definides. Aquesta representació del robot es tracta d'un robot simulat en estructura d'arbre el qual es pot crear utilitzant la funció de *rigidBodyTree*. [28]

El primer pas consistirà a crear una variable la qual anomenarem "robot" la qual definirem com a la representació del robot cridant la funció *rigidBodyTree*. D'aquesta manera quan vulguem visualitzar el robot ho podrem fer cridant la funció *show(robot)*.

Però, abans de poder visualitzar res haurem de crear els enllaços i articulacions del robot amb el qual treballarem. Per crear els enllaços i articulacions el procediment a seguir serà el següent:

1. El primer pas consistirà en crear la variable que s'utilitzarà per definir el robot de la següent manera:

```
robot = rigidBodyTree;
```

2. A continuació haurem de crear el primer enllaç i li assignarem un nom. De manera opcional en aquest punt podem decidir si assignar la massa del cos o no depenen del nivell de precisió que busquem en la simulació:

```
body1 = rigidBody('body1');
```

```
body1.Mass = 0;
```

3. Seguidament, crearem l'articulació del robot que va connectada a l'enllaç que hem creat prèviament. Quan creem l'articulació li haurem d'assignar un nom en aquesta així com el

tipus d'articulació que és. La funció admet 3 tipus d'articulacions. La primera és l'articulació del tipus *fixed* la qual es tracta d'una articulació fixa que no permet el desplaçament ni la rotació envers un eix el qual implicat que aquesta articulació no té cap grau de llibertat. El segon tipus d'articulació és del tipus *revolute* la qual consisteix en una articulació d'un grau de llibertat que permet el moviment rotatiu segons un eix donat. Finalment, tenim l'articulació del tipus *prismatic* la qual consisteix en una articulació d'un grau de llibertat que permet el desplaçament al llarg d'un eix. En aquest cas, les articulacions del robot amb el qual treballarem tindrà totes les articulacions del tipus *revolute*.

```
jnt1 = rigidBodyJoint('jnt1', 'revolute');
```

4. El següent pas consistirà a fixar els límits de les articulacions rotatives i prismàtiques. En el cas de les articulacions rotatives els límits que s'establiran consistiran en el rang d'angles que l'articulació pot assolir. Pel que fa en el cas de les articulacions prismàtiques les limitacions vindran donades en forma del desplaçament màxim que l'articulació permet. Els límits que s'establiran correspondran a la versió del robot TM14X (el qual és el model que no té càmera) que podrem observar en la fitxa tècnica de l'annex.

```
jnt1.PositionLimits = [-2*pi 2*pi];
```

5. A continuació, repetirem els passos 2, 3 i 4 fins que haguem creat tots els elements del robot.

```
body2 = rigidBody('body2');
```

```
body2.Mass = 6.337;
```

```
jnt2 = rigidBodyJoint('jnt2', 'revolute');
```

```
jnt2.PositionLimits = [-2*pi 2*pi];
```

```
body3 = rigidBody('body3');
```

```
body3.Mass = 16.156;
```

```
jnt3 = rigidBodyJoint('jnt3', 'revolute');
```

```
jnt3.PositionLimits = [deg2rad(-163) deg2rad(163)];
```

```
body4 = rigidBody('body4');
```

```
body4.Mass = 6.289;

jnt4 = rigidBodyJoint('jnt4', 'revolute');
jnt4.PositionLimits = [-2*pi 2*pi];

body5 = rigidBody('body5');

body5.Mass = 0.933;

jnt5 = rigidBodyJoint('jnt5', 'revolute');
jnt5.PositionLimits = [-2*pi 2*pi];

body6 = rigidBody('body6');

body6.Mass = 0.933;

jnt6 = rigidBodyJoint('jnt6', 'revolute');
jnt6.PositionLimits = [-2*pi 2*pi];
```

6. Un cop haguem creat tots els elements del robot passarem a assignar els paràmetres DH que hem obtingut prèviament a les articulacions corresponents el qual servirà per establir la relació entre articulacions. Aquest pas consisteix a assignar transformades fixes a cada una de les trames de les articulacions a partir dels paràmetres DH de manera que les articulacions sempre hauran de mantenir les relacions establertes per aquests paràmetres.

```
setFixedTransform(jnt1, dhparams(1, :), 'dh');
setFixedTransform(jnt2, dhparams(2, :), 'dh');
setFixedTransform(jnt3, dhparams(3, :), 'dh');
setFixedTransform(jnt4, dhparams(4, :), 'dh');
setFixedTransform(jnt5, dhparams(5, :), 'dh');
setFixedTransform(jnt6, dhparams(6, :), 'dh');
```

Com es pot observar en el codi anterior la funció que ens permet establir les transformades fixes de les articulacions ens demana tres inputs. El primer input que hem d'establir és la variable on hem creat l'articulació a la qual volem assignar la transformada. El segon input que haurem d'afegir és la fila sencera de la matriu dels paràmetres DH que correspon a l'articulació en qüestió. Els paràmetres DH han de ser donats en el següent ordre: [a, alpha,

d, theta]. El tercer input que haurem d'aportar consisteix a indicar quina relació s'estableix. Si la relació és entre l'articulació i l'enllaç previ s'indicarà com 'dh', en canvi, si la relació és entre l'articulació i l'enllaç posterior s'indicarà com 'mdh'.

7. Un cop haguem assignat els paràmetres DH a les seves respectives articulacions passarem a enllaçar les articulacions amb els seus enllaços respectius de la següent manera:

```
body1.Joint = jnt1;
```

```
body2.Joint = jnt2;
```

```
body3.Joint = jnt3;
```

```
body4.Joint = jnt4;
```

```
body5.Joint = jnt5;
```

```
body6.Joint = jnt6;
```

8. Finalment, construirem el robot afegint tots els enllaços creats. Per afegir els cossos s'hauran d'introduir tres paràmetres. El primer paràmetre a introduir consistirà en la variable on hem creat el robot. La segona variable consistirà en la variable on hem creat el cos que volem afegir al robot. La tercera variable consistirà a indicar el nom del cos previ al que volem afegir. Atenció, ja que en aquesta tercera variable s'ha d'indicar el nom que se li ha donat al cos en els passos 2 i 5 i no el nom de la variable on s'ha creat el nom. També cal mencionar que quan s'afegeixi el primer cos en aquesta tercera variable s'haurà d'indicar que el cos previ és 'base', és a dir, la base del robot.

```
addBody(robot, body1, 'base')
```

```
addBody(robot, body2, 'body1')
```

```
addBody(robot, body3, 'body2')
```

```
addBody(robot, body4, 'body3')
```

```
addBody(robot, body5, 'body4')
```

```
addBody(robot, body6, 'body5')
```


Un cop haguem afegit l'últim cos ja haurem creat la simulació de l'esquelet del nostre robot. El robot simulat serà una estructura en forma d'arbre, és a dir, els cossos de l'estructura estaran connectats a altres mitjançant una relació de 'pare' o 'fill'. Les relacions de 'pare' consisteix en la relació entre un cos, el qual li direm cos i , i el cos $i-1$, sent i el cos que s'estigui estudiant en aquell. Aquests dos cossos estaran connectats per l'articulació vinculada al cos i . Per l'altra banda la relació 'fill' consisteix en la relació entre el cos i i el cos $i+1$ els quals estan units per l'articulació vinculada al cos $i+1$.

D'aquesta manera, tots els cossos tindran ambdues relacions, la de 'pare' i la de 'fill', excepte pel primer cos que correspondrà a la base del robot i només tindrà relacions de cossos 'fills' i l'últim enllaç de cada branca o braç robòtic el només tindrà relacions de cossos 'pare'.

Per a comprovar que tot el robot s'ha creat correctament escriure'm la següent funció i executarem el programa:

```
-----
Robot: (6 bodies)

Idx   Body Name   Joint Name   Joint Type   Parent Name(Idx)   Children Name(s)
----   -
  1     body1       jnt1        revolute     base(0)            body2(2)
  2     body2       jnt2        revolute     body1(1)           body3(3)
  3     body3       jnt3        revolute     body2(2)           body4(4)
  4     body4       jnt4        revolute     body3(3)           body5(5)
  5     body5       jnt5        revolute     body4(4)           body6(6)
  6     body6       jnt6        revolute     body5(5)
-----
```

Figura 7.6 Resum dels detalls del robot

`showdetails(robot)`

En la figura 7.6 podem observar una llista amb tots els cossos que hem afegit en el nostre robot així com les diferents característiques que tenen. Com podem observar ens indica quin cos està vinculat a quina articulació i de quin tipus és aquesta, i ens indica les diferents relacions 'pare' i 'fill' que té el cos. Com que la informació obtinguda concorda amb el mencionat anteriorment podem assegurar que el robot s'ha construït correctament. El robot que haurem simulat un cop haguem comprovat que el robot s'ha configurat correctament executarem la funció `show(robot)` per visualitzar l'esquelet del robot que hem configurat.

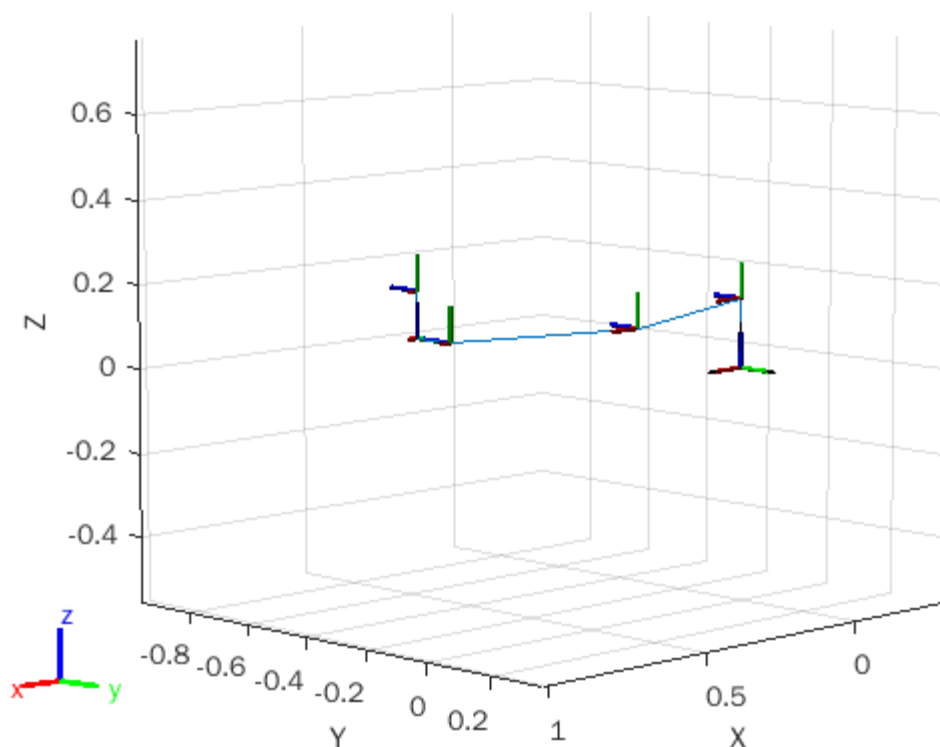


Figura 7.7 Representació esquelet del robot configurat

El codi complet de la funció per dissenyar el model `RigidBodyTree` del robot TM14X el podem observar a l'Annex II.

El següent pas per obtenir la cinemàtica inversa consistirà a començar a configurar el solver. El primer cas consistirà a crear el solver. Per fer-ho utilitzarem la funció *inverseKinematics*. Aquesta funció ens demana introduir dos valors, el primer sent el nom de la propietat entre cometes i el segon el nom de la variable on hem guardat l'estructura del robot.

```
ik=inverseKinematics('RigidBodyTree',robot);
```

La variable on hem creat el solver actuarà com a una funció que podem cridar quan desitgem executar el solver de la cinemàtica inversa. Aquesta funció requerirà que introduïm 4 paràmetres.

El primer paràmetre que haurem d'obtenir abans d'executar el solver és el nom del cos final de l'estructura del robot configurat, és a dir, el nom de l'últim enllaç afegit en el robot. L'articulació d'aquest últim enllaç correspondrà a la posició en el robot on s'instal·la l'eina.

El segon paràmetre a definir consistirà a aconseguir la transformada homogènia que relaciona la posició inicial de l'eina amb la posició final d'aquesta. En altres paraules, és la transformació que

defineix la translació i rotació que sofreix la posició de l'eina per arribar a la posició desitjada. Com ja s'ha mencionat anteriorment, les transformades homogènies consisteixen en una matriu on tenim una part corresponent a la matriu de rotació i una segona part corresponent al vector de desplaçament.

$$T = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^0R & {}^0p \\ 0 & 1 \end{bmatrix} \quad (7.1)$$

El vector de desplaçament és el més senzill d'obtenir, ja que només cal trobar la posició relativa de la posició desitjada respecte a la posició inicial. Un cop haguem trobat les posicions relatives en els tres eixos crearem un vector 1X3 que serà el nostre vector de desplaçament.

Per aconseguir la matriu de rotació primer caldrà definir en quina orientació voldrem col·locar l'eina. Un cop haguem decidit l'orientació en què col·locarem l'eina haurem de crear tres matrius de rotació bàsiques. La primera matriu de rotació bàsica serà un gir al voltant de l'eix Z, la segona matriu de rotació serà un gir al voltant de l'eix X i finalment la tercera matriu de rotació tornarà a ser un gir al voltant de l'eix Z. Realitzant aquests tres girs serà possible col·locar l'eina en qualsevol orientació que desitgem. Finalment, multiplicarem les tres matrius de rotació bàsiques creades en l'ordre que s'han fet els girs de manera que obtindrem una sola matriu de rotació de 3 X 3.

$$R_z(\alpha_1) = \begin{bmatrix} \cos \alpha_1 & -\sin \alpha_1 & 0 \\ \sin \alpha_1 & \cos \alpha_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.2)$$

$$R_x(\alpha_2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_2 & -\sin \alpha_2 \\ 0 & \sin \alpha_2 & \cos \alpha_2 \end{bmatrix} \quad (7.3)$$

$$R_z(\alpha_3) = \begin{bmatrix} \cos \alpha_3 & -\sin \alpha_3 & 0 \\ \sin \alpha_3 & \cos \alpha_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.4)$$

$${}^0R = R_z(\alpha_1) \times R_x(\alpha_2) \times R_z(\alpha_3) \quad (7.5)$$

Un cop haguem obtingut la matriu de rotació i el vector de desplaçament formarem la transformada que relaciona la posició inicial i final de l'eina el qual ens donarà el segon paràmetre que haurem d'introduir en el solver.

A continuació, haurem de definir el tercer paràmetre necessari per executar el solver. Aquest tercer paràmetre consistirà a establir les toleràncies que haurà de complir la solució donada pel solver. Els primers tres paràmetres del vector correspondran a les toleràncies del desplaçament en xyz donades en m. Els tres últims paràmetres del vector correspondran a les toleràncies de la rotació envers els eixos xyz donades en rad.

```
weights = [0.001 0.001 0.001 0.01 0.01 0.01];
```

Finalment, afegirem l'últim paràmetre abans de calcular la cinemàtica inversa el qual correspondrà a la configuració d'angles de les articulacions en la posició inicial o base. Per trobar aquestes configuracions simplement executarem la següent funció:

```
initform = robot.homeConfiguration;
```

Un cop haguem trobat els quatre paràmetres ja podem executar el solver. Quan l'executem ens hem d'assegurar de guardar els resultats d'aquest en dues variables diferents com es pot observar a continuació:

```
[configSol, solInfo] = ik('body6', tform, weights, initform);
```

En la variable anomenada *solInfo* tindrem la informació diversa sobre l'execució del solver per exemple el nombre d'iteracions, l'error o l'estatus de la solució el qual ens indicarà si és possible trobar una solució amb els paràmetres i/o toleràncies aportades. En la variable *configSol* la torbarem la configuració d'angles necessària per a assolir la posició de l'eina desitjada. Ja que la variable serà del tipus struct per poder visualitzar la configuració d'angles haurem d'executar la següent funció.

```
configSol.JointPosition
```

Per a verificar el funcionament d'aquest solver hem realitzat una prova volent desplaçar l'eina segons el següent vector desplaçament i matrius de rotació:

$${}^0_6p = \begin{bmatrix} 0.5 \\ -0.5 \\ 0.3 \end{bmatrix} \quad (7.6)$$

$$R_z(30) = \begin{bmatrix} \cos 30 & -\sin 30 & 0 \\ \sin 30 & \cos 30 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.7)$$

$$R_x(90) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 90 & -\sin 90 \\ 0 & \sin 90 & \cos 90 \end{bmatrix} \quad (7.8)$$

$$R_z(20) = \begin{bmatrix} \cos 20 & -\sin 20 & 0 \\ \sin 20 & \cos 20 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.9)$$

Les toleràncies que hem establert per aquest cas són d'1 mm per les toleràncies de desplaçament i de 0,01 rad o aproximadament 0,60 graus pel moviment l'orientació de l'eina.

Introduint tots aquests paràmetres en el solver hem obtingut la següent configuració d'angles.

Articulació	Solució 1 (rad)	Solució 2 (rad)
Articulació 1	0,7931	-3,3657
Articulació 2	-4,1588	-6,2832
Articulació 3	-2,7139	1,7762
Articulació 4	-2,0510	-4,2322
Articulació 5	-4,9946	-1,0203
Articulació 6	-6,0008	-6,0008

Taula 7.3 Configuració de les posicions de les articulacions

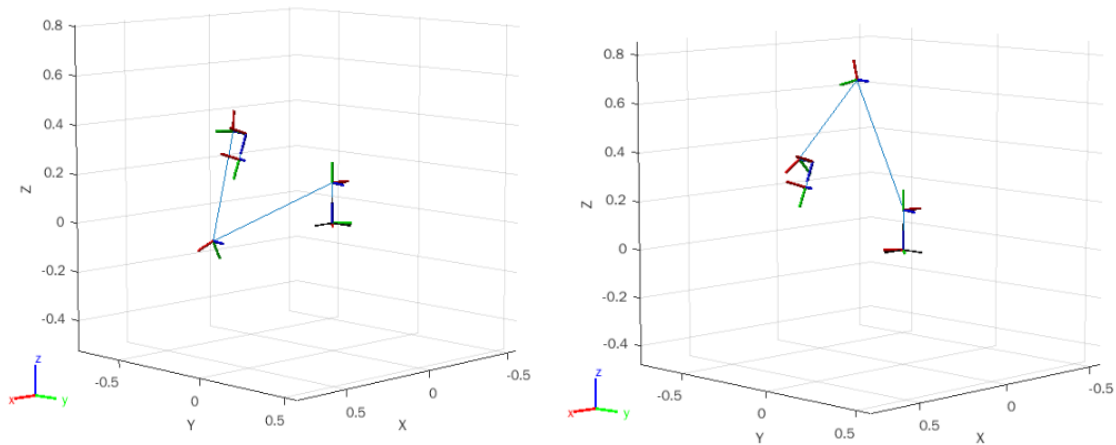


Figura 7.8 Possibles solucions al problema de la cinemàtica inversa

Com es pot observar en la figura 7.8 i la taula 7.3 existeixen múltiples solucions pel problema de la cinemàtica inversa. Com que la cinemàtica inversa pot donar múltiples resultats diferents cada cop que executem el solver ens retornarà una configuració d'angles diferents com es pot observar en els dos casos representats anteriorment.

Alternativament, a aquesta solució hem estudiat la possibilitat d'obtenir la configuració dels angles i la generació de trajectòries utilitzant una interfase interactiva. Aquesta forma d'aconseguir la configuració dels angles ens permetrà moure el robot manualment utilitzant una simulació i marcar la posició exacta que volem que adopti el robot. Aquesta manera d'introduir la posició simularà el procediment d'un operari movent l'eina del robot col·laboratiu fins a una posició per programar-la de manera ràpida i senzilla.

Per utilitzar aquest mètode per obtenir la configuració dels angles utilitzarem el *RigidBodyTree* que hem creat anteriorment i utilitzarem la següent funció per a habilitar una figura que ens permetrà interactuar manualment amb l'esquelet del robot.

```
gui = interactiveRigidBodyTree(robot, MarkerScaleFactor=0.4);
```

Executant la funció anterior se'ns obrirà la següent finestra.

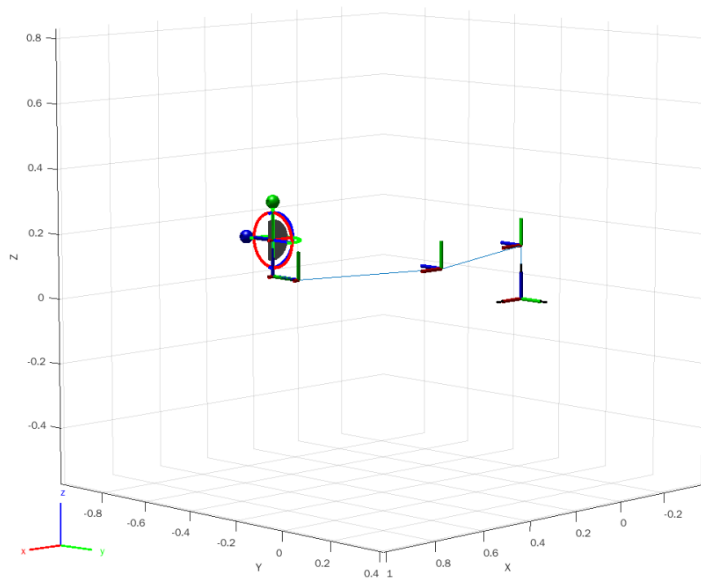


Figura 7.9 Finestra del robot interactiva

Per defecte, l'articulació interactiva que es podrà moure serà l'articulació més propera a l'extrem. Com es pot observar en la figura 7.9 en l'última articulació del braç robòtic tenim uns marcadors que anteriorment no apareixien. Aquests marcadors serveixen per modificar la posició i l'orientació del tram interactiva.

Com es pot observar tenim tres cercles amb una esfera en un dels seus extrems. Aquests cercles serveixen per modificar l'orientació de la trama interactiva. Cada un dels tres cercles ens permetrà girar la trama envers un dels eixos movent l'esfera que al costat del cercle.

Per modificar la posició de la trama, utilitzarem l'àrea negra que trobem a l'interior dels cercles i l'arrossegarem per tal de moure la trama en l'espai XYZ.

Un cop haguem col·locat la trama final en la posició i orientació que desitgem executarem la següent funció per guardar la configuració actual dels angles de les articulacions.

```
gui.addConfiguration
```

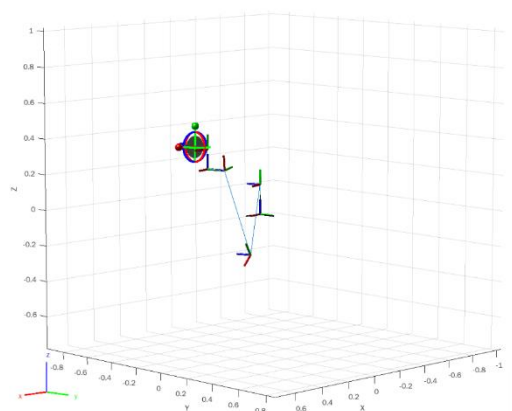


Figura 7.10 Nova posició i orientació del robot interactiu

Executant aquesta funció ens permetrà guardar la configuració dels angles actual la qual podrem visualitzar executant el següent el qual ens permetrà visualitzar totes les configuracions guardades.

gui.StoredConfigurations

Per poder consultar el codi complet dissenyat pel càlcul de la cinemàtica inversa es pot consultar en el Annex III;

7.3. Dinàmica inversa

Com s'ha comentat anteriorment, existeixen dues metodologies principals pel càlcul del parell d'un robot. El primer mètode consisteix en la metodologia d'Euler-Lagrange i la metodologia Newton-Euler. El resultat que obtindrem del càlcul de la dinàmica inversa utilitzant qualsevol dels dos mètodes mencionats serà l'obtenció del parell de cada un dels motors a partir de les posicions, velocitats i acceleracions angulars del manipulador.

7.3.1. Euler-Lagrange

Per resoldre la dinàmica Lagrangiana hem utilitzat el procediment explicat en la viabilitat tècnica desenvolupada durant l'avantprojecte. Durant el càlcul de la dinàmica Lagrangiana utilitzarem els paràmetres DH trobats en l'apartat 7.1.

Ja que la dinàmica Lagrangiana es calcula a partir de les energies cinètiques i potencials, el primer pas consistirà en el disseny d'una funció capaç de calcular les velocitats lineals i angulars en el centre de masses de cada enllaç i en l'obtenció dels tensors d'inèrcia en el centre de masses de cada enllaç.

Per calcular les velocitats lineals i angulars buscarem la matriu Jacobiana que relacioni la velocitat angular del motor amb les velocitats lineals i angulars de cada enllaç en el seu centre de masses.

El primer pas per calcular la matriu Jacobiana consistirà en l'obtenció de les matrius homogènies entre els diferents enllaços i les matrius de transformació homogènies entre la base i cada un dels enllaços. D'aquesta manera obtindrem 2 matrius de transformació homogènia per cada enllaç.

Per calcular les transformacions homogènies entre dos enllaços contigus programarem l'equació 6.1 i per obtenir les transformacions homogènies entre la base i els enllaços programarem l'equació

6.2. A continuació presentem el codi per trobar les matrius de transformacions homogènies entre enllaços i les matrius homogènies entre la base i l'enllaç.

```

i=1;
for q = [q1 q2 q3 q4 q5 q6]
%Càlcul transformació homogènia entre enllaços (Eq. 6.1)
A(:, :, i)=[cosd(rad2deg(q)) -sind(rad2deg(q)) 0 dhparams(i,1);
sind(rad2deg(q))*cosd(rad2deg(dhparams(i,2)))
cosd(rad2deg(q))*cosd(rad2deg(dhparams(i,2))) -sind(rad2deg(dhparams(i,2))) -
sind(rad2deg(dhparams(i,2)))*dhparams(i,3);
sind(rad2deg(q))*sind(rad2deg(dhparams(i,2)))
cosd(rad2deg(q))*sind(rad2deg(dhparams(i,2))) cosd(rad2deg(dhparams(i,2)))
cosd(rad2deg(dhparams(i,2)))*dhparams(i,3);0 0 0 1];

    i=i+1;
end
%Càlcul transformació homogènia base-enllaç (Eq. 6.2)
T(:, :, 1)=A(:, :, 1);
T(:, :, 2)=A(:, :, 1)*A(:, :, 2);
T(:, :, 3)=A(:, :, 1)*A(:, :, 2)*A(:, :, 3);
T(:, :, 4)=A(:, :, 1)*A(:, :, 2)*A(:, :, 3)*A(:, :, 4);
T(:, :, 5)=A(:, :, 1)*A(:, :, 2)*A(:, :, 3)*A(:, :, 4)*A(:, :, 5);
T(:, :, 6)=A(:, :, 1)*A(:, :, 2)*A(:, :, 3)*A(:, :, 4)*A(:, :, 5)*A(:, :, 6);

```

A continuació, separarem el vector Z el qual correspon a la component Z de la matriu de rotació. És a dir, guardarem en una variable la tercera columna (excepte el 4t valor) de la matriu de rotació de la transformada homogènia entre la base i els enllaços. Per aïllar els vectors Z executarem el següent codi canviant la i pel nombre de l'enllaç:

```
Z(:, :, i)=T(1:3, 3, i);
```

També haurem de guardar el vector O el qual consisteix en el vector posició situat en la 4a columna (excepte el 4t valor) de les transformades homogènies entre la base i l'enllaç. En aquest cas, haurem de guardar un vector O inicial igual a 0. Per tant, la programació del vector O es realitzarà d'una manera similar a l'obtenció del vector Z excepte per la diferència que tindrem $n+1$ vectors sent n el nombre de trames del nostre manipulador.

```
O(:, :, 1) = sym([0; 0; 0]);
```

```
O(:, :, i) = sym(T(1:3, 4, i));
```

El càlcul de la matriu Jacobiana està compost per dues parts. La primera correspon a la matriu que ens relaciona la velocitat lineal en el centre de masses amb la velocitat angular del motor. La segona, consisteix en la matriu que relaciona la velocitat angular del centre de masses amb la velocitat angular del motor. L'equació a seguir per a calcular la matriu Jacobiana completa serà la següent:

$$J = \begin{bmatrix} Z_i x (O_{i+1} - O_i)^1 \\ Z_i \end{bmatrix} \quad (7.10)$$

A continuació presentem l'equació 7.1 programada utilitzant el Matlab:

```
for x=1:6
```

```
J(:, :, x) = sym(zeros(6));
```

```
    for i=1:x
```

```
        J(:, i, x) = [(cross(Z(:, :, i), (O(:, :, x+1) - O(:, :, i))))]; Z(:, :, i)];
```

```
    end
```

```
end
```

Un cop hem obtingut la matriu Jacobiana multiplicarem la velocitat angular del motor per la part corresponent i podrem obtenir la velocitat lineal o angular en el centre de masses.

```
function w = angvel(J, qd)
```

```
    for i=1:6
```

```
        w(:, :, i) = J(4:6, :, i) * qd;
```

¹ Cal destacar que l'operador \times correspon al producte vectorial entre dos valors, i no a una multiplicació, que en la funció Matlab la calculem amb la funció `cross()`

```

end

end

function v=Linel(J,qd)

for i=1:6

v(:, :, i)=J(1:3, :, i)*qd;

end

end

```

Un cop calculades les velocitats lineals i angulars en els centres de masses procedirem a calcular el tensor d'inèrcia de cada un dels enllaços en el seu centre de masses. Pel fet que el tensor d'inèrcia l'estudiem respecte al centre de masses, només haurem de calcular la diagonal de l'equació 6.12. Per calcular els elements de la diagonal utilitzarem l'equació 6.13 per a calcular la inèrcia segons cada un dels eixos de rotació.

Per simplificar el càlcul del tensor d'inèrcia i mantenir-lo el més proper a la realitat farem servir una geometria cilíndrica per descriure els diferents enllaços. A partir de les dades proporcionades per la fitxa tècnica deduirem l'alçada i radi de cada un d'aquests cilindres i calcularem el seu volum. Un cop tenim el volum de cada enllaç i , ja que només coneixem la massa total del manipulador repartirem la massa total proporcionalment entre els diferents enllaços segons el seu volum.

Enllaç	Altura [H] (m)	Radi [R] (m)	Massa [M] (kg)
Enllaç 1	0	0	0
Enllaç 2	0.2652	0.08	6.337
Enllaç 3	0.6761	0.08	16.156
Enllaç 4	0.4679	0.06	6.289
Enllaç 5	0.1	0.05	0.933
Enllaç 6	0.1	0.05	0.933

Taula 7.4 Característiques enllaços

Un cop establertes les característiques de cada enllaç procedirem a programar el càlcul la diagonal dels tensors d'inèrcia.

```
for i=1:6
    I(:, :, i)=[(M(i)*(3*R(i)^2+H(i)^2))/12 0 0; 0 (M(i)*(3*R(i)^2+H(i)^2))/12 0;
0 0 M(i)*R(i)^2/2];
end
```

A continuació ja haurem calculat totes les dades necessàries per poder calcular l'equació 6.11 i obtenir l'energia cinètica en cada enllaç. El codi Matlab que hem programat per a poder calcular l'energia cinètica és el següent:

```
for i=1:6
    Ki(i)= M(i)*v(:, :, i).'*v(:, :, i)/2+w(:, :, i).'*I(:, :, i)*w(:, :, i)/2;
end
```

Per calcular l'energia potencial només ens resta trobar la distància segons l'eix Z entre la trama de la base i l'enllaç que volem calcular. Per calcular aquesta distància primer haurem de trobar les transformades homogènies entre la base i l'enllaç en qüestió tal com hem fet anteriorment però donant-li un valor negatiu a l'angle *tetha*. Un cop tenim aquestes transformades procedirem a multiplicar la matriu de rotació d'aquestes transformades pel vector de posició que uneix una trama amb l'enllaç al que està connectat a continuació. A aquest resultat li haurem de sumar el vector posició que uneix la trama base amb l'enllaç que volem connectar de manera que el resultat ens donarà el vector posició que uneix la base amb el centre de masses d'aquest enllaç.

Un cop haguem obtingut aquest vector posició procedirem a calcular l'energia potencial de cada enllaç utilitzant l'equació 6.14.

```
for i=1:6
    Pi(i)= M(i)*[0; 0; 9.81].'*Zcm(:, :, i);
end
```

Per simplificar el càlcul i reduir la càrrega computacional d'aquest unirem les dues equacions de les energies en el mateix *for*.

El següent pas consistirà a obtenir el Lagrangià del manipulador el qual obtenim de restar la suma de totes les energies cinètiques menys la suma de totes les energies potencials tal com es veu en l'equació 6.8.

$$K=K_i(1)+K_i(2)+K_i(3)+K_i(4)+K_i(5)+K_i(6);$$

$$P=P_i(1)+P_i(2)+P_i(3)+P_i(4)+P_i(5)+P_i(6);$$

$$L=K-P;$$

Finalment, aplicarem l'equació 6.18 per a trobar el parell de cada motor. Per aplicar aquesta equació es podran resoldre les derivades parcials mitjançant el Matlab, però per la derivada temporal s'haurà de realitzar el càlcul manualment.

```
for i = 1:6
    dpP(i)=simplify(diff(P,q(i)));
    dpK(i)=simplify(diff(K,q(i)));
    dvK(i)=simplify(diff(K,qd(i)));
end
```

Un cop hem calculat les derivades parcials i la derivada de manera manual haurem obtingut finalment el par de cada motor.

7.3.1.1. Problemes amb Euler-Lagrange

Durant el disseny del codi de càlcul de la dinàmica inversa d'Euler-Lagrange ens hem trobat amb una sèrie de problemes que han provocat que no utilitzem aquesta metodologia per la solució final.

El primer problema residia en què la complexitat del càlcul mitjançant Euler-Lagrange augmentava de manera exponencial. Això no seria un problema si es pogues calcular la dinàmica de Lagrange utilitzant un mètode tancat. Però, ja que el càlcul conté una part que s'ha de calcular manualment no és possible el disseny d'una funció que calculi de forma tancada com a una funció del Simulink. Degut a això per obtenir equacions que ens permetin el càlcul de les acceleracions angulars s'hauria de calcular els parells utilitzant variables simbòliques i posteriorment aïllar les acceleracions angulars utilitzant la funció solve del Matlab. Això, però, ens donaria 6 equacions excepcionalment llargues les quals no serien possibles de resoldre utilitzant les Matlab Functions del Simulink.

El segon problema amb el qual ens hem trobat és la resolució manual de les derivades temporals a causa de l'alta complexitat de les equacions simbòliques a resoldre. Si el cas d'estudi consistís en un robot col·laboratiu de 4 o menys graus de llibertat encara podria ser factible calcular les derivades temporals, però a partir dels 5 graus de llibertat la complexitat del problema a resoldre augmenta en gran manera i el càlcul de la derivada temporal passa a ser inviable de resoldre d'una manera manual.

Degut a aquests possibles es va decidir variar del procediment inicial dissenyat i aplicar el mètode recursiu de Newton-Euler el qual ens permetrà especialment evitar els problemes esmentats anteriorment.

Per observar el codi creat per obtenir la dinàmica inversa mitjançant Euler-Lagrange fins el punt de les derivades parcials consultar el Annex IV.

7.3.2. Newton-Euler

A causa dels inconvenients del càlcul de la dinàmica inversa utilitzant Euler-Lagrange mencionats en l'apartat anterior hem decidit desviar-nos del plantejament i aplicar el mètode de càlcul recursiu de Newton-Euler per resoldre la dinàmica inversa. Usant aquesta metodologia serem capaços de crear una funció capaç de calcular els parells en cada una dels diferents motors.

El mètode recursiu de Newton-Euler consisteix en dues sèries d'equacions. La primera sèrie consistirà en un càlcul progressiu de les velocitats angulars i les acceleracions lineals i angulars començant per l'enllaç 1 fins l'enllaç n (sent n el nombre de enllaços que té el manipulador). Un cop calculada la primera sèrie de càlculs procedirem a calcular la segona sèrie d'equacions. La segona sèrie d'equacions consistirà en un càlcul regressiu de les forces i el moment començant per l'enllaç n fins al enllaç 1. Les equacions que utilitzarem i que presentarem a continuació s'han obtingut principalment del llibre *Robòtica* per John J. Craig i el llibre *Introducción a la Robótica* per Subir Kumar Saha.

El càlcul de la sèrie d'equacions progressives ens permetrà calcular la propagació de les velocitats angulars i les acceleracions lineals i angulars. A continuació podem observar les equacions i l'ordre en què haurem de programar les equacions de la primera sèrie [19] [20].

$$w_i = {}_{i-1}^i R * w_{i-1} + \dot{\theta}_i * z \quad (7.11)$$

$$(7.12)$$

$$\dot{w}_i = {}_{i-1}^i R * \dot{w}_{i-1} + {}_{i-1}^i R * w_{i-1} \times \dot{\theta}_i * z + \ddot{\theta}_i * z$$

$$\dot{v}_i = {}_{i-1}^i R * (\dot{w}_{i-1} \times {}^{i-1}_i P + w_{i-1} \times (w_{i-1} \times {}^{i-1}_i P)) + \dot{v}_{i-1} \quad (7.13)$$

$$\dot{v}_{C_i} = \dot{w}_i \times {}_{C_i}^i P + w_i \times (w_i \times {}_{C_i}^i P) + \dot{v}_i \quad (7.14)$$

$$F_i = m_i * \dot{v}_{C_i} \quad (7.15)$$

$$N_i = I_{C_i} * \dot{w}_i + w_i \times I_{C_i} * w_i \quad (7.16)$$

Sent:

w_i : Velocitat angular de la trama i

\dot{w}_i : Acceleració angular de la trama i

\dot{v}_i : Acceleració lineal de la trama i

\dot{v}_{C_i} : Acceleració lineal en el centre de masses del enllaç i

F_i : Força en el centre de masses del enllaç i

N_i : Moment de torsió en el centre de masses del enllaç i

${}^{i-1}_i P$: Vector posició que uneix la trama $i - 1$ amb la trama i

${}_{C_i}^i P$: Vector posició que uneix la trama i amb el centre de masses del enllaç i

z : Vector igual a $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

$\theta, \dot{\theta}, \ddot{\theta}$: Posició angular, velocitat angular i acceleració angular

Quan hem creat el codi en qüestió ens hem trobat que la primera iteració del codi és particular, ja que no tenim dades de les velocitats i acceleracions d'una iteració anterior. Degut això primer de

tot s'ha d'establir unes condicions inicials per a la base. Les característiques inicials a la base seran les següents:

$$w_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \dot{w}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \dot{v}_i = \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix}$$

Un cop establertes les condicions inicials programarem les equacions en un bucle *for* que anirà des d'1 fins a 6.

```
for i=1:6
```

```
% Distància entre trama i-1 i La trama i (dhparams correspon a La matriu de Denavit-Hartenberg)
```

```
    p = [dhparams(i,1); dhparams(i,3)*sind(dhparams(i,2));  
        dhparams(i,3)*cosd(dhparams(i,2))];
```

```
% Distància entre trama i i el centre de masses del enllaç i
```

```
    r = -A(1:3,1:3,i)*[dhparams(i,1)/2; dhparams(i,3)*sind(dhparams(i,2))/2;  
        dhparams(i,3)*cosd(dhparams(i,2))/2];
```

```
    if i==1
```

```
% Equació 7.2
```

```
        wi(:,i)=ei*qd(i);
```

```
% Equació 7.3
```

```
        wdi(:,i)=ei*qdd(i);
```

```
% Equació 7.4
```

```
        vdi(:,i)= A(1:3,1:3,i)*(g);
```

```
% Equació 7.5
```

```
        vdci(:,i)=vdi(:,i)+cross(wdi(:,i),r)+cross(wi(:,i),cross(wi(:,i),r));
```

```
    else
```

```
% Equació 7.2
```

```
        wi(:,i)= A(1:3,1:3,i)*(wi(:,i-1))+ei*qd(i);
```

```
% Equació 7.3
```

```
        wdi(:,i)= A(1:3,1:3,i)*wdi(:,i-1)+cross(A(1:3,1:3,i)*wi(:, i-1),ei*qd(i))+ei*qdd(i);
```


% Equació 7.4

```
vdi(:,i)= A(1:3,1:3,i)*(vdi(:,i-1)+cross(wdi(:,i-1),p)+cross(wi(:,i-1),cross(wi(:,i-1),p))));
```

% Equació 7.5

```
vdci(:,i)=vdi(:,i)+cross(wdi(:,i),r)+cross(wi(:,i),cross(wi(:,i),r));
```

```
end
```

% Equació 7.6

```
Fi(:,i)=M(i)*vdci(:,i);
```

% Equació 7.7

```
Ni(:,i)=I(:, :, i)*wdi(:,i)+cross(wi(:,i),I(:, :, i)*wi(:,i));
```

```
end
```

Un cop haguem programat el bucle que calculi la sèrie d'equacions progressives començarem a estudiar la sèrie d'equacions regressiva la qual ens permet calcular les forces i el parell de torsió que actuen en les articulacions i el parell dels motors. Les equacions de la sèrie regressiva que haurem de programar són les següents [19]:

(7.17)

$$f_i = {}_{i+1}^i R * f_{i+1} + F_i$$

(7.18)

$$n_i = N_i + {}_{i+1}^i R * n_{i+1} + c_i^i P \times F_i + {}_{i+1}^i P \times {}_{i+1}^i R * f_{i+1}$$

(7.19)

$$\tau_i = n_i^T * z$$

En aquest cas tornarem a programar les equacions de la sèrie regressiva en un bucle *for*, però, en aquest cas el bucle anirà des de n (sent n el nombre d'enllaços del manipulador) fins a 1.

```
for i=6:-1:1
```

```
    p = [dhparams(i,1); dhparams(i,3)*sind(dhparams(i,2));
dhparams(i,3)*cosd(dhparams(i,2))];
```

```
    r = A(1:3,1:3,i)*[dhparams(i,1)/2; dhparams(i,3)*sind(dhparams(i,2))/2;
dhparams(i,3)*cosd(dhparams(i,2))/2];
```

```
    if i==6
```

% Equació 7.8

```
        fi(:,i)=Fi(:,i);
```

% Equació 7.9

```
        ni(:,i)=Ni(:,i)+cross(r,Fi(:,i));
```

```
% Equació 7.10
```

```
    tau(i)=ni(:,i).'*ei;
```

```
else
```

```
% Equació 7.8
```

```
    fi(:,i)=Fi(:,i)+A(1:3,1:3,i).'*fi(:,i+1);
```

```
% Equació 7.9
```

```
    ni(:,i)=Ni(:,i)+cross(r,Fi(:,i))+cross(p,A(1:3,1:3,i).'*fi(:,i+1))+A(1:3,1:3,i).'*ni(:,i+1);
```

```
% Equació 7.10
```

```
    tau(i)=ni(:,i).'*ei;
```

```
end
```

```
end
```

En el codi presentat pel càlcul de la sèrie regressiva de Newton-Euler presentem un cas on el robot no està agafant cap càrrega i , per tant, no tenim cap força aplicada a l'enllaç final del robot. En el cas que el manipulador aixequés un objecte i , per tant, s'hagués de tenir en compte la força donada pel pes d'aquest es substituiria les fórmules 7.8 i 7.9 dins del *if* inicial en el que es calcula les forces, moments i parell en l'articulació final. Les fórmules 7.8 i 7.9 d'aquest *if* eren fórmules simplificades de l'original la qual podem observar dins del *else* just després del condicional. Per tenir en compte el pes que aixeca el manipulador haurem de substituir les fórmules simplificades pels completes i substituir $fi(:,i+1)$ i $ni(:,i+1)$ per els vectors de la força i el moment de torsió que aplica la càrrega sobre el robot.

Utilitzant aquestes funcions, serem capaços de calcular el parell dels diferents motors del nostre manipulador a partir de qualsevol combinació de posicions, velocitats i acceleracions angulars.

Tal com, hem mencionat, utilitzant aquestes fórmules haurem resolt el problema de la dinàmica inversa, però per poder realitzar la simulació necessitem obtenir les equacions de moviment que ens donin les acceleracions angulars dels motors a partir de les posicions i velocitats angulars i el parell dels motors. Per a poder trobar això, haurem de realitzar un últim pas el qual és trobar la dinàmica directa.

El codi complet que hem dissenyat per a calcular la dinàmica inversa utilitzant el mètode recursiu de Newton-Euler el podem consultar en l'Annex V.

7.4. Dinàmica directa

La dinàmica directa del manipulador consisteix en l'obtenció de les acceleracions angulars de cada una de les articulacions a partir del parell dels motors i la posició i velocitats angulars. Per resoldre la dinàmica directa hem decidit utilitzar el mètode de Walker-Orin el qual ens permet calcular les acceleracions angulars utilitzant la dinàmica inversa de Newton-Euler que hem dissenyat prèviament.

Si desenvolupéssim de forma simbòlica i de forma genèrica les equacions de Newton-Euler obtindríem la següent equació:

$$\tau = M(\theta) * \ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) \quad (7.20)$$

Sent:

$M(\theta)$: Matriu de masses de dimensions $n \times n$

$V(\theta, \dot{\theta})$: Suma dels vector amb els paràmetres centrífugs i el vector amb els paràmetres de Coriolis $n \times 1$

$G(\theta)$: Vector corresponent als termes corresponents a la gravetat $n \times 1$

El primer pas per resoldre la dinàmica directa utilitzant el mètode de Walker-Orin parteix de l'equació 7.11 i consisteix en sumar els vectors $V(\theta, \dot{\theta})$ i $G(\theta)$ en un vector el qual anomenarem $B(\theta, \dot{\theta})$. Aquest vector es pot calcular resolent la funció que hem creat del mètode recursiu de Newton-Euler substituint les acceleracions per un vector de 6×1 amb tots els seus paràmetres igual a 0. D'aquesta manera estarem calculant el valor del parell, però eliminant l'efecte que té la matriu de masses de manera que en aquest cas el parell serà igual al vector B . Substituint B en la fórmula 7.11 i aïllant la matriu de masses i l'acceleració ens queda:

$$M(\theta) * \ddot{\theta} = (\tau - B) \quad (7.21)$$

A continuació aïllem l'acceleració angular de manera que només ens restarà trobar la inversa de la matriu de masses.

$$(7.22)$$

$$\ddot{\theta} = M(\theta)^{-1} * (\tau - B)$$

Fins aquest punt el mètode de Walker-Orin es calcula utilitzant formes pràcticament idèntiques, però, el càlcul de la matriu de masses és el punt on el mètode de Walker-Orin presenta diversos mètodes per a calcular la matriu de masses. Els mètodes de Walker-Orin que utilitzarem és el primer que, tot i que es el mètode que requereix una càrrega computacional més gran ens permetrà utilitzar la funció del càlcul regressiu de Newton-Euler per calcular la matriu [30].

Executant la matriu d'Inercia substituint les velocitats angulars, la gravetat i les forces externes per 0 i l'acceleració per un vector de $n \times 1$ amb tots els seus paràmetres iguals a 0 excepte per una acceleració angular a la qual igualarem a 1 ens retornarà una de les columnes de la matriu de masses. Per tant, executant la funció de Newton-Euler n vegades amb els paràmetres explicats anteriors i igualant a 1 cada cop una acceleració angular diferent obtindrem n columnes les quals formaran la nostra matriu de masses.

Programant aquest procediment obtindrem el següent codi:

```
H1=dinamica_newton(q,[0 0 0 0 0 0],[1 0 0 0 0 0],[0;0;0]).';
H2=dinamica_newton(q,[0 0 0 0 0 0],[0 1 0 0 0 0],[0;0;0]).';
H3=dinamica_newton(q,[0 0 0 0 0 0],[0 0 1 0 0 0],[0;0;0]).';
H4=dinamica_newton(q,[0 0 0 0 0 0],[0 0 0 1 0 0],[0;0;0]).';
H5=dinamica_newton(q,[0 0 0 0 0 0],[0 0 0 0 1 0],[0;0;0]).';
H6=dinamica_newton(q,[0 0 0 0 0 0],[0 0 0 0 0 1],[0;0;0]).';
H=cat(2,H1,H2,H3,H4,H5,H6);    %Funció per juntar les diferents columnes
```

Un cop hem calculat la matriu de masses programarem l'equació 7.22 i finalment haurem resolt la dinàmica directa i obtingut les equacions de moviment necessàries per calcular les acceleracions angulars.

Per a poder consultar la funció completa dissenyada per calcular la dinàmica directa es pot consultar el codi en l'Annex VI.

7.5. Resultats Simulació

El objectiu final d'aquest projecte compost per dos informes era l'obtenció i disseny d'un model de simulació amb controlador PID que tingués una resposta el més similar possible a la trajectòria dissenyada.

A continuació, presentarem els resultats obtinguts durant el projecte de la component electrònica on s'han tractat la simulació del sistema robòtic aplicant el controlador PID que hem sintonitzat. El model del Simulink utilitzat per realitzar les simulacions utilitza blocs de la llibreria *Robotic Systems Toolbox* per calcular la dinàmica inversa i la matriu de masses. Els blocs que hem utilitzat d'aquesta llibreria utilitzen els mateixos mètodes explicats, però executen el càlcul d'una manera molt més optimitzada i eficient reduint així el temps que tarda en fer-se la simulació. Per consultar el model Simulink creat per realitzar la simulació consultar els Annexos VII, VIII, IX.

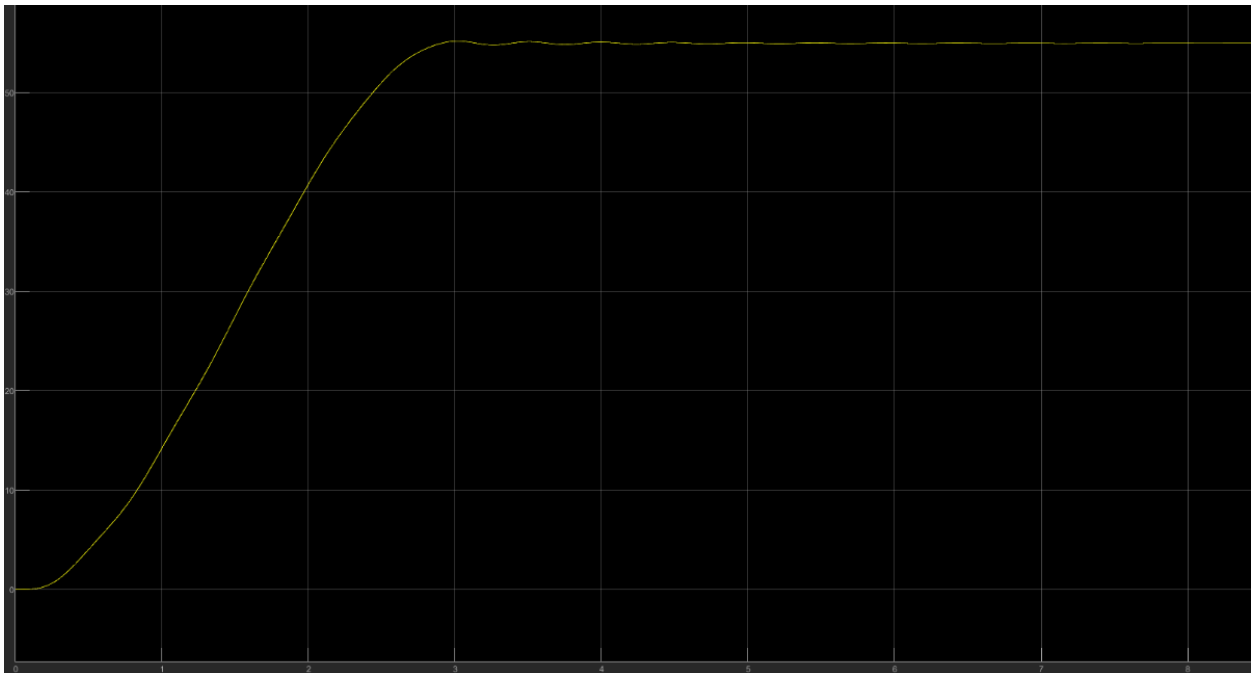


Figura 7.11 Articulació 1 amb PID - θ destitjada = 55°

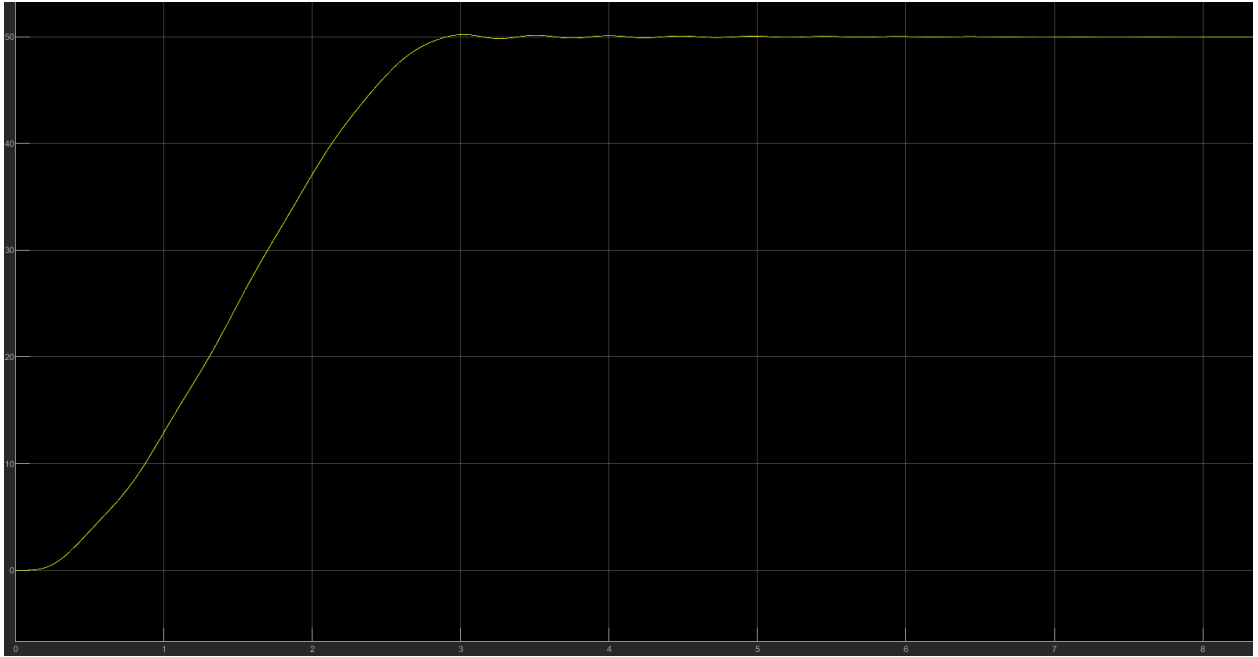


Figura 7.12 Articulació 2 amb PID - θ destitjada = 50°

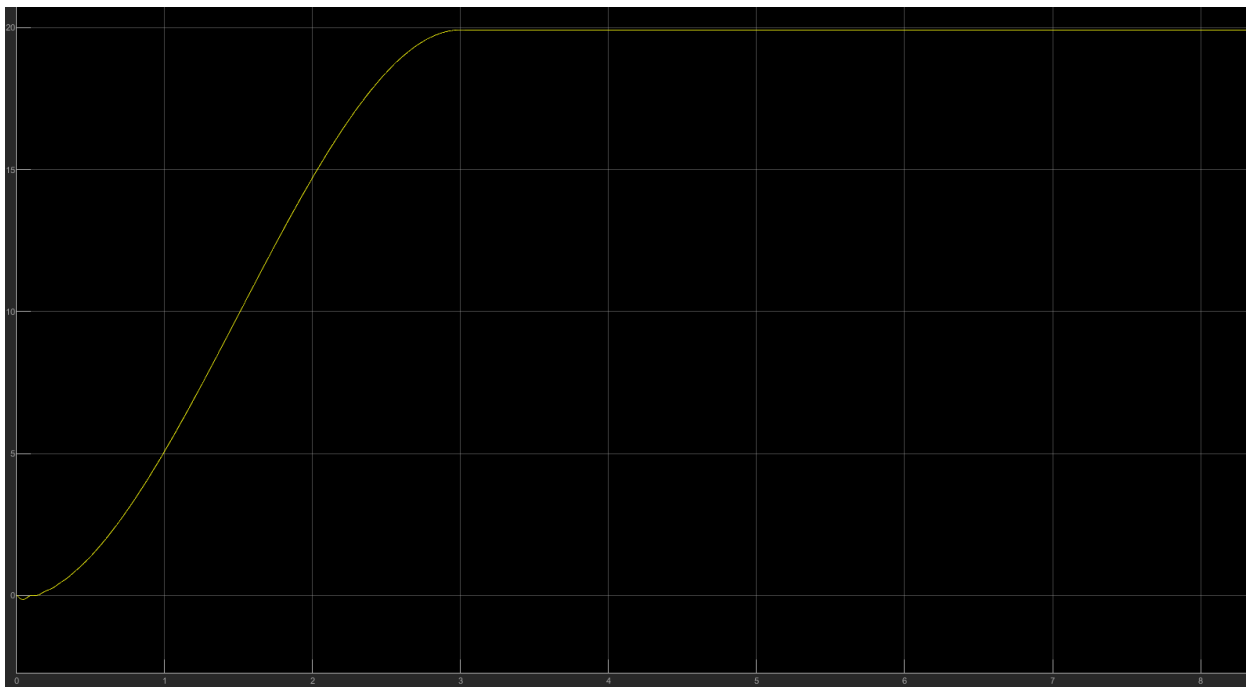


Figura 7.13 Articulació 3 amb PID - θ destitjada = 20°

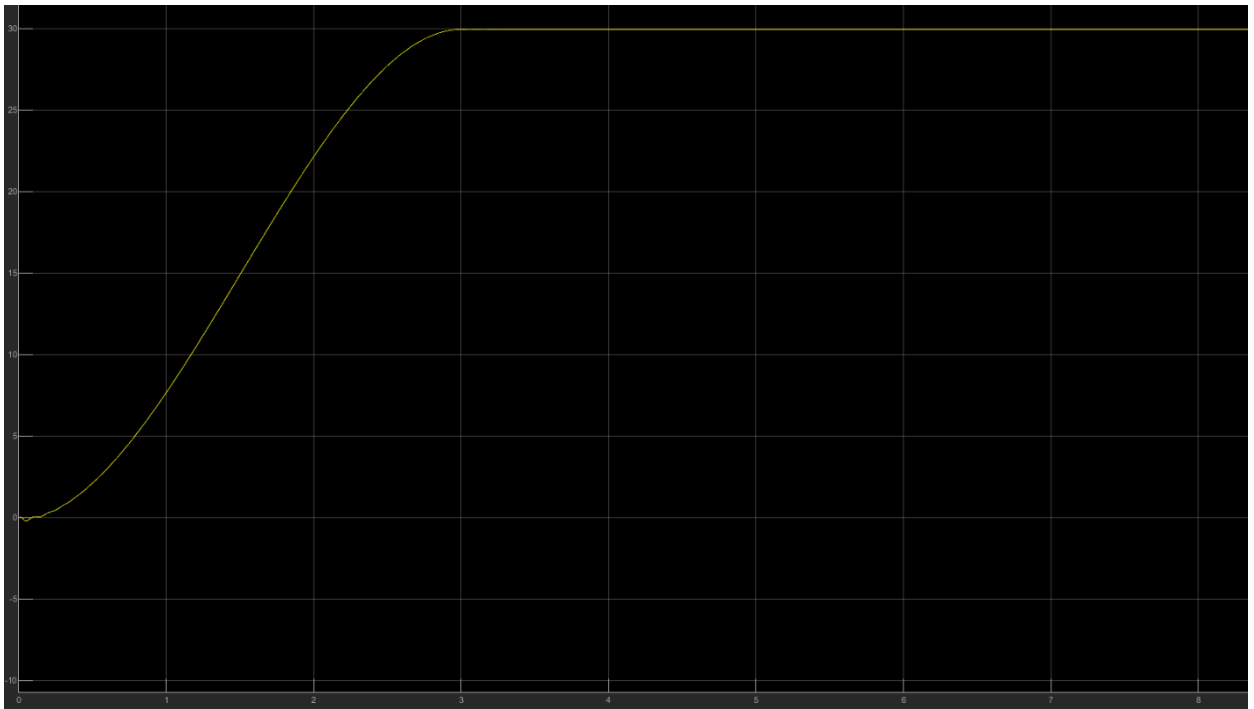


Figura 7.15 Articulació 4 amb PID - θ destitjada = 30°

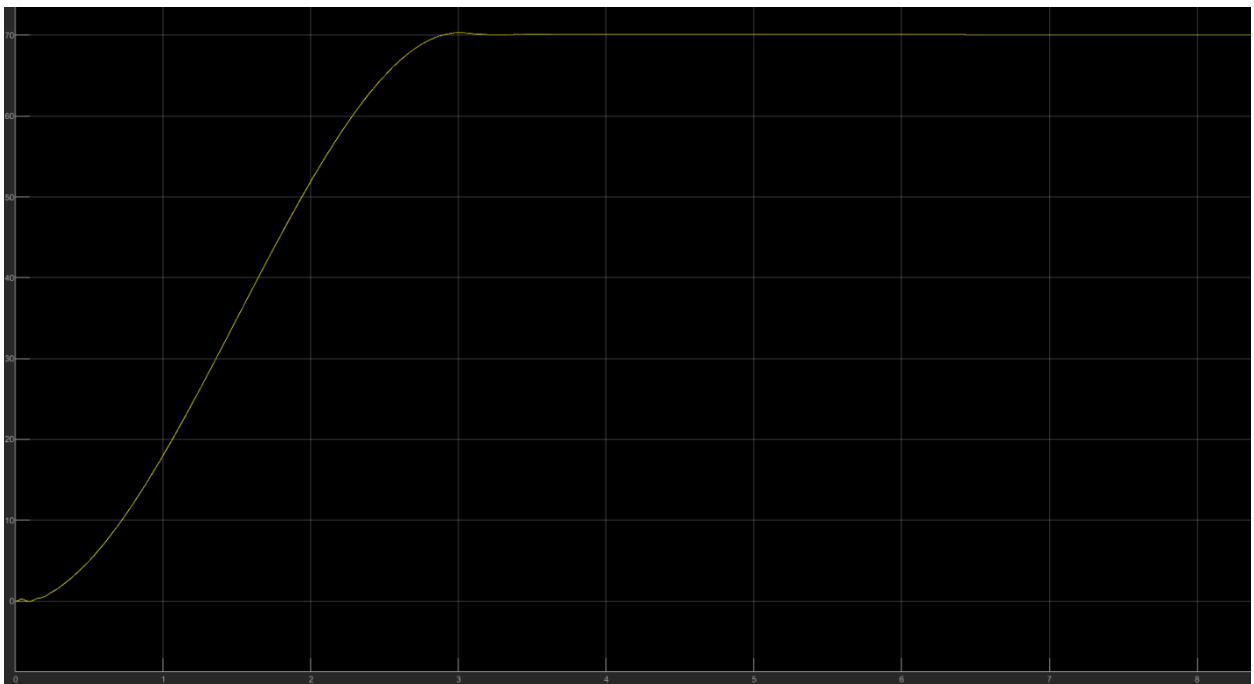


Figura 7.14 Articulació 5 amb PID - θ destitjada = 70°

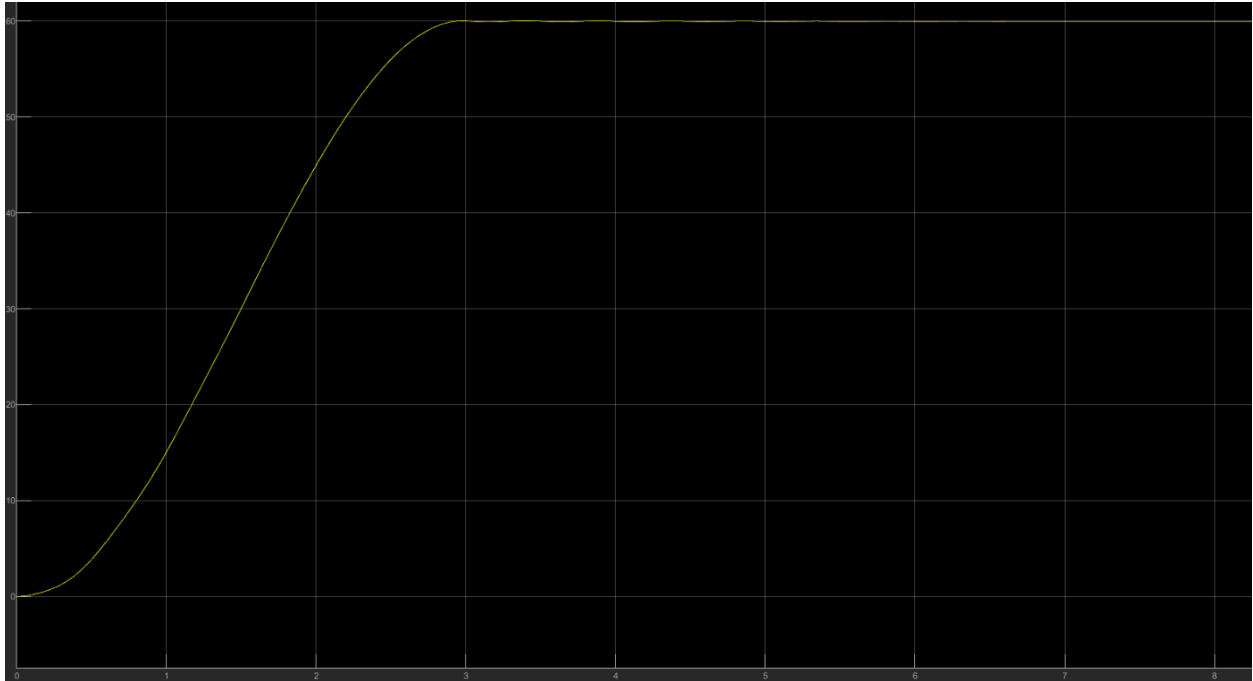


Figura 7.16 Articulació 6 amb PID - θ desitjada = 60°

Com podem observar en les figures 7.11-7.16, la resposta és molt propera a la resposta desitjada. La resposta obtinguda es realitza en el temps programat (3 segons) i amb un “overshoot” percentual inferior al 2 %. En el cas més desfavorable, el qual és l’articulació 3, l’error relatiu és menor al 0.4 % de manera que tenim una resposta amb una precisió molt alta.

8. Planificació

En el següent apartat parlarem de la planificació que es seguirà per gestionar el projecte. Aquesta planificació inclourà una planificació des de la presentació del avantprojecte fins l'entrega final d'ambdós projectes, així com el anàlisi i gestió de riscos i l'aplicació d'eines de seguiment en el projecte.

8.1. Pla d'execució (Diagrama de Gantt)

En el següent apartat parlarem de la planificació que se seguirà per gestionar el projecte. Aquesta planificació inclourà una planificació des de la presentació de l'avantprojecte fins a l'entrega final d'ambdós projectes, així com l'anàlisi i gestió de riscos i l'aplicació d'eines de seguiment en el projecte.

8.2. Pla d'execució (Diagrama de Gantt)

8.2.1. Planificació projecte

En el següent apartat presentarem el Diagrama de Gantt de l'execució del projecte realitzat utilitzant el Microsoft Project i s'explicaran les diferents tasques que el conformen.

Les tasques a realitzar durant l'etapa d'execució són:

Replantejament avantprojecte

El replantejament de l'avantprojecte consisteix en la implementació dels canvis necessaris que s'hagin de realitzar en el projecte després de la retroacció per part dels responsables de projecte. Aquests canvis preveiem que no s'allarguin més de 3 dies, però es pot donar el cas que s'hagin de fer canvis majors el qual es tindrà en compte durant l'anàlisi i gestió de riscos.

Estudis

Aquesta tasca consistirà en la realització dels diferents càlculs de manera seqüencial. Els càlculs es realitzaran seguint la metodologia indicada en l'apartat de viabilitat tècnica i formarà la part principal de l'entrega de la memòria intermèdia. En el cas que alguna tasca dels estudis s'hagi

allargat més del previst i no es pugui entregar en un estat acceptable es farà la corresponent menció en la memòria intermèdia i es presentarà juntament amb la memòria final del projecte.

Simulacions

Les simulacions consisteixen en l'elaboració de un model de simulació per a poder representar els resultats obtinguts en els estudis. Degut això es realitzaran posteriorment als estudis i es preveu que es presentaran durant l'entrega final.

Redacció de la memòria

La redacció de la memòria l'hem separat en dues parts. La primera consistirà en la redacció del gruix de l'informe i es començarà de manera paral·lela als estudis i les simulacions. Ja que s'hi dedicarà 5 hores diàries al projecte, s'empraran tres hores per avançar en els estudis i la resta del temps sobrant per redactar el que s'ha fet prèviament. Això ens permetrà poder portar un informe actualitzat dia a dia el qual ens atorgarà certa flexibilitat si sorgeix qualsevol imprevist que allargui pugui perjudicar el projecte.

La segona part del projecte consistirà en el tancament de l'informe el qual consistirà en l'adequació de l'informe a un format uniforme i que s'adeqüi als criteris demanats. També es farà un repàs ortogràfic, s'actualitzaran els índexs d'apartats, imatges i taules i s'afegiran els reforços visuals que es creguin necessaris per reforçar l'explicació del projecte.

Millores/ Recuperació d'hores

Finalment, afegim una última tasca abans de cada entrega la qual consistirà en un període de dues setmanes en les quals s'executaran les tasques que s'hagin endarrerit a causa de imprevistos o s'utilitzaran per implementar possibles millores constructives que s'hagin considerat posteriorment a l'entrega de l'avantprojecte.

En total el projecte tindrà una duració de **125 dies**.

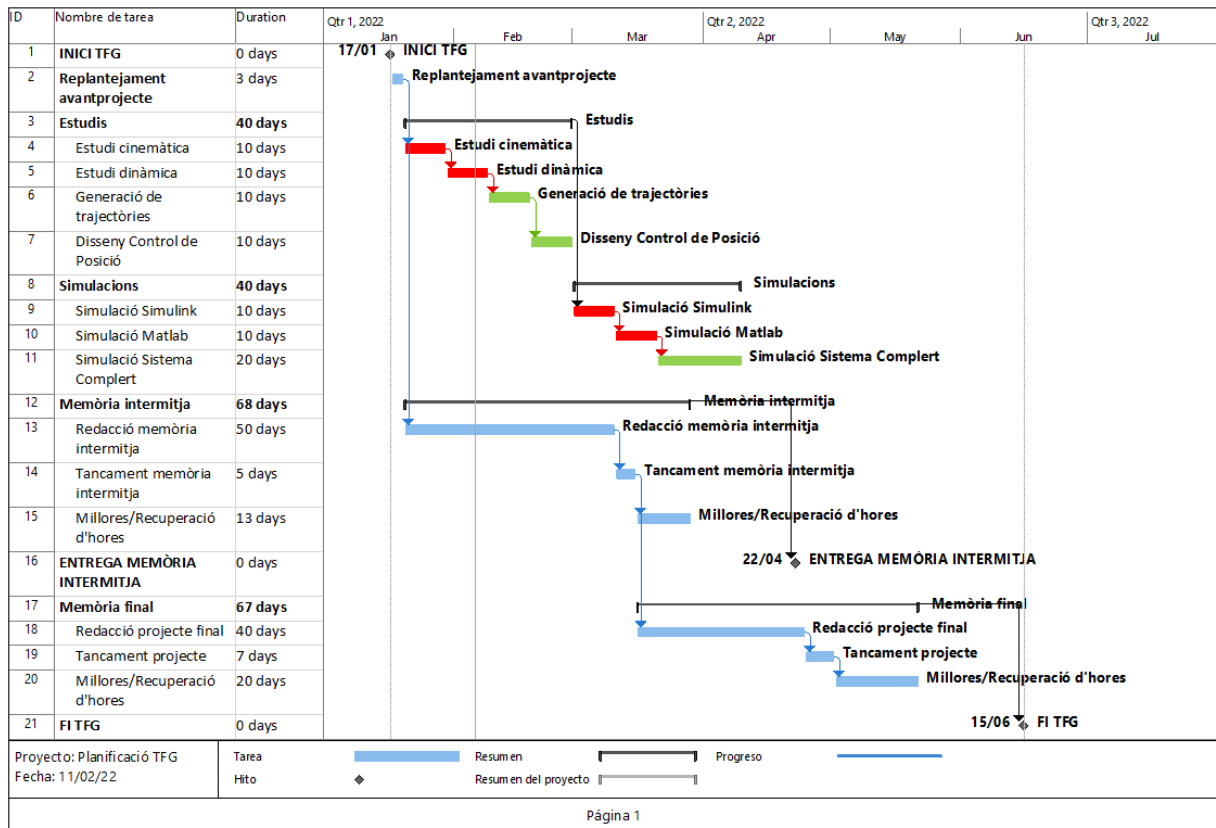


Figura 8.1 Diagrama de Gantt projecte

En aquest Diagrama de Gantt podem observar tasques de tres tipus. Les tasques de color blau corresponen a tasques que es realitzen tant pel projecte de mecànica com al projecte d'electrònica, les tasques de color vermell corresponen a les tasques que es realitzen només en el projecte de mecànica i les tasques de color verd corresponen a les tasques que es desenvolupen en el projecte electrònic.

8.2.2. Planificació avantprojecte

A continuació presentarem el Diagrama de Gantt de l'execució de l'avantprojecte realitzat utilitzant el Microsoft Project i s'explicaran les diferents tasques que el conformen.

Les tasques de la redacció de l'avantprojecte són molt similars a les que s'han explicat en la planificació de l'avantprojecte i s'ha estructurat de la mateixa manera. Pel que fa a la resta de tasques les explicarem a continuació:

Reunió plantejament TFG

Reunió amb el tutor encarregat del TFG per aclarir dubtes i posar en marxa el projecte. En aquesta reunió es van aclarir els objectius que tindria el projecte i com encarar-los de cara a l'avantprojecte i al projecte.

Anàlisi problema

En aquesta tasca s'ha analitzat el problema que es tractarà durant el projecte i s'han identificat les parts clau d'aquest per organitzar i posteriorment executar d'una manera clara i ordenada.

Cerca d'informació

Aquesta tasca ha consistit en la recerca d'antecedents i l'estudi del marc teòric del projecte en qüestió. En aquesta cerca de la informació també s'ha estudiat els procediments o procediments que s'hauran de seguir per a poder assolir l'objectiu final del projecte.

Planificació projecte

En aquesta etapa es realitzarà una planificació de l'execució del projecte utilitzant el Microsoft Project. Durant la planificació es tindran en consideració les dues entregues corresponents a la memòria intermèdia i la memòria final.

En total l'avantprojecte tindrà una duració de **112 dies**.

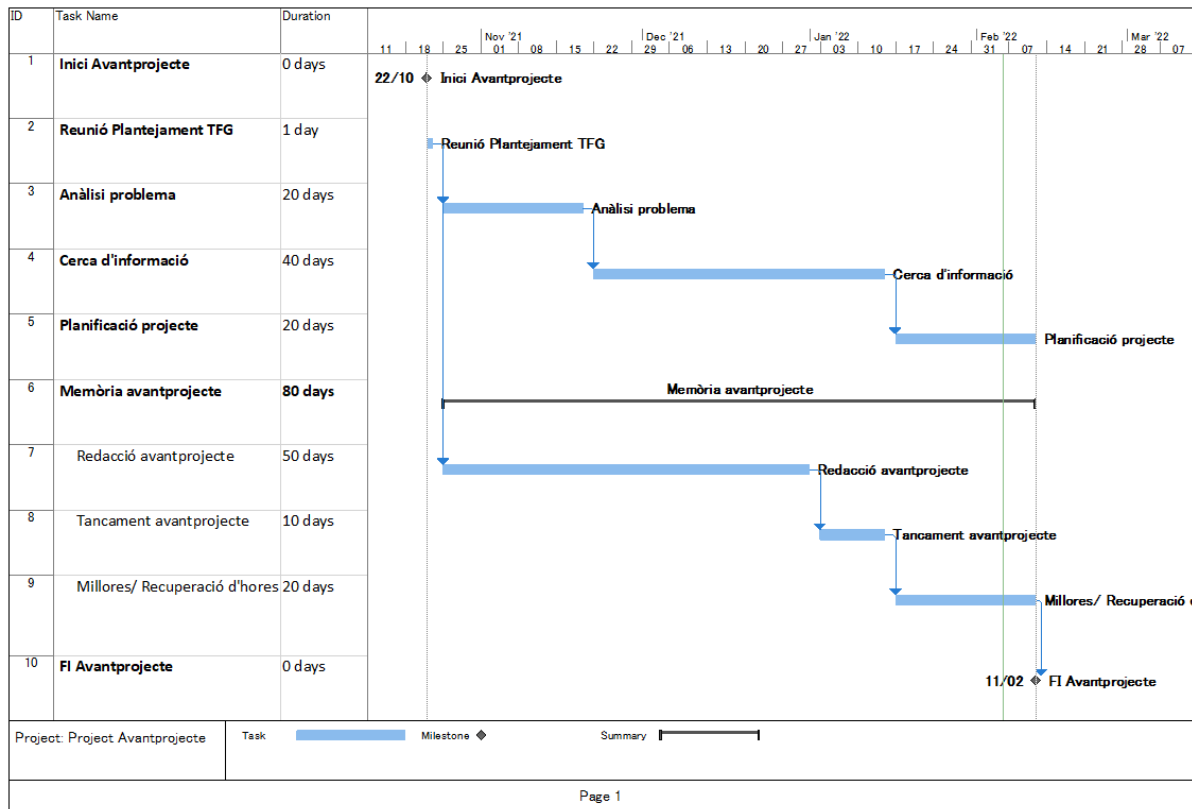


Figura 8.2 Diagrama de Gantt avantprojecte

8.3. Anàlisi i Gestió de Riscos

Per analitzar els riscos utilitzarem la metodologia FMEA (Failure Mode and Effective Analysis) amb la qual comencem identificant els possibles riscos i els efectes que poden tenir. A continuació classificarem els diferents riscos segons una puntuació donada pels següents criteris:

- **Freqüència:** Els diferents riscos es classificaran segons la freqüència puntuant d'1 fins a 10 sent 1 un risc molt poc probable i sent 10 un risc molt probable.
- **Gravetat:** Els diferents riscos es classificaran segons la gravetat de les conseqüències puntuant d'1 fins a 10 sent 1 un risc amb unes conseqüències menyspreables i sent 10 un risc que pugui comprometre la finalització del projecte.
- **Detecció:** Els diferents riscos es classificaran segons la seva fàcil detecció puntuant d'1 a 10 sent 1 un risc el qual molt probablement es detectarà i sent 10 un risc el qual es molt probable que ocorri sense que ens assabentem.

Un cop tinguem la classificació realitzada i amb les puntuacions escollides se sumaran els diferents punts atorgats i es farà una última classificació per determinar la criticitat del risc.

8.3.1. Taula FMEA

A continuació presentem la taula obtinguda amb la metodologia FMEA [20] on podem observar la classificació realitzada dels diferents riscos considerats.

Risc	Freqüència	Gravetat	Detecció	Criticitat
Incidències meteorològiques	2	6	1	9
Incidències mèdiques	7	5	4	16
Aparells informàtics malmesos	4	8	7	19
Planificació poc realista	4	8	5	18
Falta de claredat	5	7	8	20
Rendiment menor de l'esperat	5	6	4	15

Taula 8.1 Anàlisi de riscos FMEA

La columna corresponent a la *Criticitat* es el resultat de la suma dels paràmetres de la *Freqüència*, *Gravetat* i la *Detecció* i anirà des de un valor mínim de 3 fins a un valor màxim de 30. La criticitat la classificarem de la següent manera:

- Valors entre 3 i 10: Els riscos amb una puntuació entre 3 i 10 els considerarem riscos amb baixa importància. Per a aquests riscos es portarà un control trimestral i establir unes mesures preventives i d'actuació bàsiques.
- Valors entre 11 i 20: Els riscos amb una puntuació entre 11 i 20 els considerarem amb una importància moderada. Per aquests riscos es portarà un control mensual i establir mesures preventives i d'actuació.
- Valors entre 21 i 30: Els riscos amb una puntuació superior als 20 punts seran riscos crítics els quals s'haurà de portar un control setmanal i establir mesures de prevenció i actuació amb una àmplia dedicació de recursos per contrarestar els riscos.

8.3.2. Gestió dels riscos

Els riscos ha considerar en aquest projecte són molt limitats ja que el projecte en qüestió es purament teòric. Tenint això en consideració els riscos que s'han tingut en consideració han sigut els següents:

- **Incidències meteorològiques:** El risc d'incidències meteorològiques el tenim en consideració, no per quan hi hagi mal temporal o pluges, sinó per quan es produeixin borrasques com el Glòria o altres incidents meteorològics que puguin parar el moviment interurbà o inclús produir parades en el subministrament elèctric. Aquests incidents, tot i que són poc freqüents i fàcilment detectables consultant els informatius meteorològics, poden tenir greus conseqüències en el projecte en el cas que es produeixin. En els casos en què es produeixin talls de subministrament elèctrics s'avançarà el projecte amb mitjans analògics i es passarà a format digital quan el temporal hagi amainat.
- **Incidències mèdiques:** A causa de l'actual situació de la pandèmia hem considerat necessari incloure les incidències mèdiques com a un risc clau del projecte. Degut a l'estat actual de la pandèmia de la COVID-19 relativament probable contraure la malaltia el qual pot endarrerir l'execució del projecte. Un cop un s'ha contagiat no es poden prendre moltes mesures addicionals en un projecte individual, però sí que es poden prendre mesures preventives per tal de reduir les probabilitats d'infecció com l'ús de EPIs i mantenir el distanciament social sempre que sigui oportú i possible.
- **Aparells informàtics malmesos:** Existeix la possibilitat que els aparells electrònics que s'estiguin utilitzant per elaborar el projecte, com per exemple el ordinador, s'avariïn. Això pot suposar una parada en el projecte durant un parell de dies en el millor dels casos o una pèrdua de tot el projecte en format digital en el pitjor dels casos. Per això, serà convenient guardar els avenços en el projecte amb un Backup al núvol per a no perdre el treball realitzat fins aquell moment i poder continuar desenvolupant-lo des de un ordinador alternatiu de manera temporal. Addicionalment, quan el risc es produeixi una altra conseqüència en que es produirà en el projecte és el augment del cost del projecte. Degut a tot això, serà imperatiu portar un bon manteniment dels aparells que s'utilitzin per reduir la probabilitat d'averies.
- **Planificació poc realista:** Es pot donar el cas que la planificació realitzada durant l'avantprojecte hagi estat massa optimista i, per tant, no es pugui executar el projecte tal com estava previst. Aquest problema pot tenir greus conseqüències, ja que una mala planificació pot provocar un gran nombre d'endarreriments que impedeixin la correcta finalització del projecte. Per evitar aquest cas, a part d'intentar fer la planificació més realista possible durant l'avantprojecte, s'aplicaran les eines de seguiment que s'explicaran en el següent apartat les quals ens permetran portar un control sobre la planificació inicial i modificar-la sempre que ens sigui necessari.
- **Falta de claredat:** Un altre problema que sorgeix del plantejament de l'avantprojecte és la falta de claredat en la metodologia o procediments que s'hauran de seguir. Això pot ser

donat a una mala o insuficient recerca d'informació o que durant l'execució del projecte s'hagi pivotat en una direcció que no s'havia previst durant l'avantprojecte. Sense importar l'origen de la falta d'informació, el risc en qüestió provocarà que s'hagi de dedicar vaires hores a la cerca d'informació el qual pot endarrerir la planificació realitzada.

- **Rendiment menor al esperat:** Aquest risc va molt lligat al risc de la planificació poc realista mencionat anteriorment. Es pot donar el cas que s'hagi subestimat la complexitat de les tasques que s'hagin de realitzar o que s'hagi sobreestimat les capacitats d'un mateix el qual pot provocar que el ritme de treball sigui menor al previst el qual pot provocar endarreriments. Degut això s'hauran d'aplicar les eines de seguiment per portar un control de si la feina diària feta és la prevista o si és menor al previst realitzant un ràpid informe del treball realitzat en el qual mencionarem la feina realitzada, el temps emprat i les dificultats que s'han trobat durant aquest.

8.4. Eines de Seguiment

Per realitzar el seguiment del projecte s'ha considerat la utilització dels programes de Notion o Jira. Al final s'ha optat per utilitzar Jira perquè permet un millor seguiment i control del projecte i els diferents problemes que vagin sorgint durant el projecte [21] [22].

A l'hora de crear el projecte en el programa hem seleccionat el *template* d'un projecte de negocis, ja que era la base que més s'adequa a les nostres necessitats. Un cop hem creat el projecte hem introduït les tasques representades en el Diagrama de Gantt i hem definit les mateixes dates d'inici i final que hem definit.

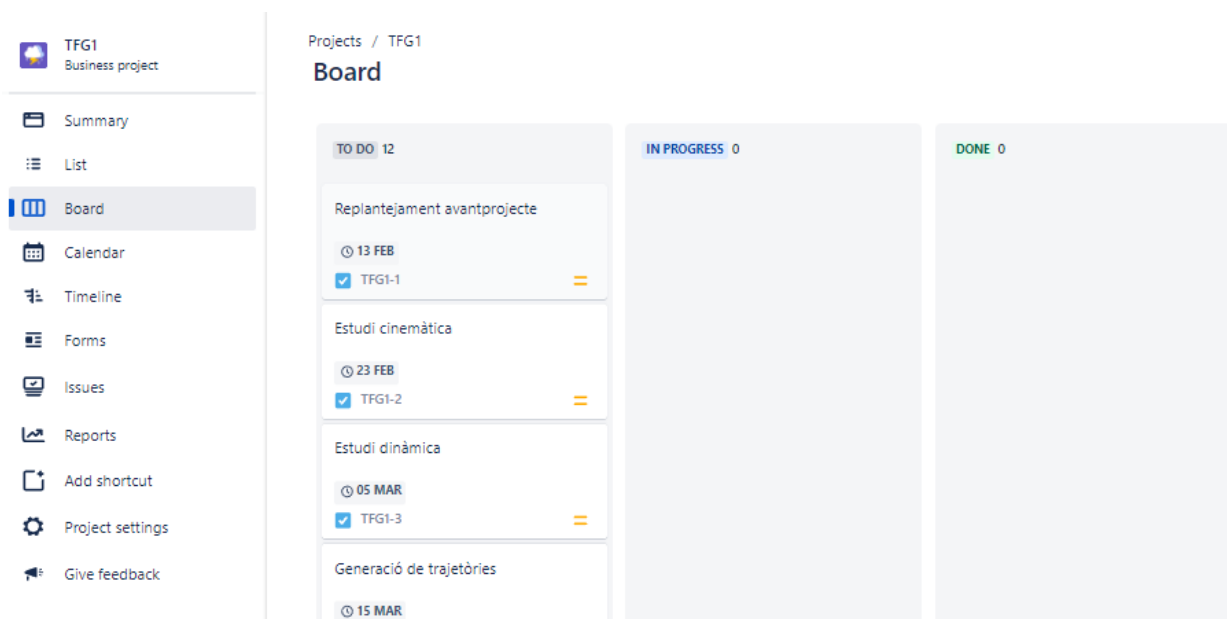


Figura 8.3 Jira llistat de tasques

En la imatge anterior podem observar les diferents funcionalitats que tenim disponibles. Les pestanyes que utilitzarem amb més freqüència seran el calendari, el *timeline* i la pestanya de problemes.

El *timeline* consisteix en una representació simplificada i dinàmica del Diagrama de Gantt creat el qual ens permetrà tenir una idea clara de l'estat del projecte. Addicionalment, es poden crear subtasques el qual ens permetrà tenir un control més precís de l'estat i saber més fàcilment si s'han produït retard en la planificació.

La pestanya del calendari ens permetrà poder visualitzar les dates límits de les diferents tasques del projecte així com crear dates d'entrega pròpies que no s'hagin plantejat en la planificació ideal.

Finalment, la pestanya de problemes (*issues*) ens permetrà crear incidències durant el projecte i gestionar-les al llarg del temps. Aquesta pestanya serà l'eina principal que utilitzarem per a portar el seguiment i resoldre els diferents problemes que es vagin detectant durant el seguiment del projecte.

Per a portar el seguiment del projecte, al final de cada dia treballat anirem a la pestanya de problemes en el programa Jira i seleccionarem les tasques que s'hagin avançat en aquell dia. A continuació clicarem els tres punts a la dreta de la tasca i seleccionarem l'opció de *Log Work* i se'ns obrirà la següent finestra.

Log Work: TFG1-12

Time Spent ^{*}

 (eg. 3w 4d 12h) [?]

An estimate of how much time you have spent working.

Date Started ^{*}

 [?]

Remaining Estimate






Adjust automatically
the estimate will be reduced by the amount of work done, but never below 0.

Leave estimate unset

Set to (eg. 3w 4d 12h)

Reduce by (eg. 3w 4d 12h)

Work Description

Style ^v | B | I | U | A ^v | ~~A~~ ^v |  ^v |  |  |  ^v | + ^v | 



 [?]  Viewable by All Users

Figura 8.4 Jira Log Work

En aquesta anirem introduint les hores que s'han dedicat a la tasca i una descripció de la feina que s'ha realitzat durant el dia així com els diferents problemes que hagin sorgit. En el cas que durant el dia l'avanç hagi sigut menor a l'esperat, s'haurà de seleccionar l'opció de *Reduce by* i indicar la reducció real estimada en la duració del projecte.

En el cas que hagi sorgit un problema important es crearà una subtasca en la tasca afectada, se li assignarà la duració estimada i s'allargarà la tasca afectada en aquesta duració.

8.5. Replantejament memòria intermèdia

L'execució del projecte fins a la memòria intermèdia no ha sigut completament sense imprevistos. Per aquest motiu s'ha considerat adient realitzar una segona planificació del projecte per a poder portar una planificació actualitzada i així poder evitar possibles problemes deguts aquests retard.

Dins dels estudis les tasques corresponents a l'estudi de la cinemàtica inversa i la generació de trajectòries s'ha pogut executar segons la planificació plantejada durant l'avantprojecte. Però, pel que fa a l'estudi de la dinàmica, l'execució d'aquest s'ha endarrerit més del que s'havia plantejat inicialment el qual ha provocat que no es pugui presentar els resultats de l'estudi de la dinàmica del robot durant la memòria intermèdia. Aquest endarreriment provocarà que totes les tasques que depenguin de l'estudi de la dinàmica s'endarrereixi. Entre aquestes trobem el disseny de la posició el qual no es pot iniciar fins que no s'hagi dissenyat la resta de blocs així com les simulacions del sistema les quals tenien com a precedent els estudis. La simulació de la cinemàtica és l'única excepció, ja que s'ha realitzat la simulació de la cinemàtica com a part de l'estudi d'aquesta.

A continuació, presentarem el nou Diagrama de Gantt amb la informació de l'estat d'execució del projecte actualitzada:

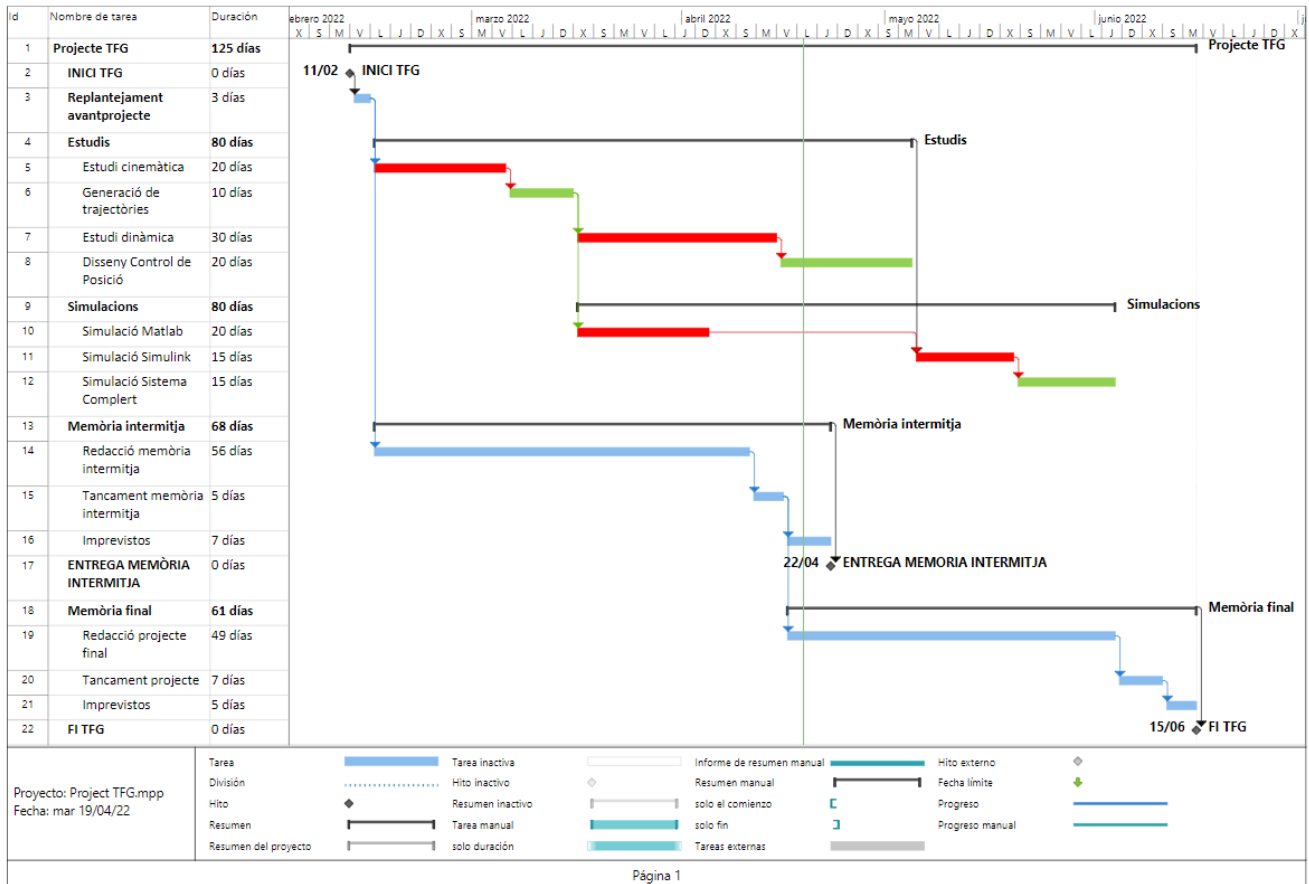


Figura 8.5 Diagrama de Gantt replantejament

Com es pot observar, l'estudi de la cinemàtica inversa i la dinàmica ha durat més del previst per culpa de certs imprevistos puntuals. Això ha provocat que no es puguí presentar l'estudi de la dinàmica en la memòria intermèdia tal com estava previst en la planificació de l'avantprojecte. Per culpa de això s'ha post posat l'inici del disseny del control de posició així com de les simulacions. Tot i això es podrà minimitzar l'efecte d'aquest retard reduint el temps dedicat a la tasca dels imprevistos per a poder complir les entregues establertes pel projecte.

9. Anàlisi Econòmica

En aquest apartat realitzarem un estudi dels costos aplicables en el nostre projecte. Ja que aquest projecte que estem desenvolupant és un estudi teòric i, per tant, no es durà a terme un estudi en profunditat dels costos i la rendibilitat que donarà el projecte. En el seu lloc es presentarà un resum del cost de les hores personals dedicades així com el cost de les llicències dels programes emprats.

El cost de les hores l'hem calculat a partir de les hores dedicades a l'estudi i el desenvolupament dels avantprojectes en qüestió. Estimem que les hores que s'han dedicat estan a prop de les 400 h des de l'inici del projecte. El cost hora que aplicarem l'hem obtingut de les dades aportades per l'ECCAT en el qual podem observar que el sou mitjà d'un enginyer treballant a Catalunya està al voltant dels 40.000 € l'any [21]. D'aquí podem deduir que el cost hora rondarà al voltant dels 20 €/h el qual ens donaria un cost de les hores dedicades de 8.000 €.

A continuació podem veure una taula on es poden observar els diferents softwares que s'utilitzaran i els preus de la compra de les llicències anuals. A causa del fet que el projecte no dura el mateix que la llicència anual també hem considerat quin serà el cost proporcional de les llicències anuals durant els 237 dies del projecte [22] [23].

Software	Preu/any	Preu proporcional
Matlab (licència perpetua)	800 €	519,45 €
Microsoft Word	69 €	44,80 €
Microsoft Project	39,90 €	25,91 €

Taula 9.1 Preus llicències

A causa de la possibilitat que es produeixin incidents que augmentin el cost s'ha considerat un cost addicional corresponent a imprevistos igual a un 10% del valor del projecte, és a dir, 859,02 €.

Per tant, el cost final del projecte teòric serà de **9.449,18 €** el qual com es pot observar és un preu molt petit, ja que el projecte només inclou l'estudi teòric del sistema de control d'un braç robòtic i, en conseqüència, no es tindrà en compte els costos pertinents a la fabricació ni a la compra de materials. Per visualitzar el desglossament dels costos veure l'Annex I.

A continuació exposem el desglossament dels costos de l'execució i planificació del projecte:

Resum Costos del Projecte

Cost hores	Unitats	Cost unitari	Cost total
Hores emprades	400 h	20,00 €	8.000,00 €
Cost llicències			
Matlab	237 dies	2,19 €	519,45 €
Microsoft Word	237 dies	0,19 €	44,80 €
Microsoft Project	237 dies	0,11 €	25,91 €
			8.590,16 €
10% Imprevistos ¹			859,02 €
		TOTAL	9.449,18 €

¹ Entre els imprevistos es consideren els costos de papereria, estris d'escriptura i el augment en costos considerat en el risc de Aparells informàtics malmesos tractat en el apartat d'Anàlisi de riscos.

10. Anàlisi Mediambiental

Per a realitzar l'estudi de la viabilitat mediambiental ens trobem amb un problema similar al que ens trobem amb la viabilitat econòmica. Ja que el projecte consisteix en un estudi purament teòric l'impacte mediambiental és pràcticament nul. Tot i això, s'han pres mesures com fer les reunions amb el supervisor del projecte (tutor del TFG) via Zoom i en el cas que ens haguéssim de desplaçar utilitzar el transport públic sempre que sigui possible. D'aquesta manera es minimitza encara més l'impacte que pugui tindre el desenvolupament del projecte.

Les taules de l'estudi mediambiental que s'han dut a terme per realitzar l'estudi mediambiental les podem consultar a l'Annex I.

11. Conclusions

Un cop executat el projecte complert hem pogut comprovar que s'ha pogut completar els objectius planejats al inici d'aquest per ambdues parts del projecte conjunt. Pel que fa la component mecànica del projecte, la qual és la que tractem en aquest informe, s'ha pogut dissenyar satisfactòriament. S'ha pogut dissenyar funcions que calculin adequadament la cinemàtica inversa i la dinàmica directa del nostre robot les quals es podran utilitzar posteriorment en el informe *Disseny i simulació d'un controlador PID pel seguiment de trajectòries d'un robot de 6 graus de llibertat* per a dissenyar la simulació del robot amb o sense PID.

Tot i així, encara existeixen certs punts els quals podrien ser objecte de futurs estudis de cara a un futur ja que no s'ha tingut temps a desenvolupar i incloure'ls en el projecte. Durant el projecte vàrem haver de fer compromisos alhora de calcular la funció de la dinàmica directa ja que el 1er mètode de càlcul de Walker-Orin implica la major càrrega computacional dels diferents mètodes. Durant el projecte vàrem escollir aquest mètode degut a la seva simplicitat i perquè ja teníem dissenyada la funció de Newton-Euler, però, ja que aquest mètode requereix de un major nombre de càlculs es podria considerar un futur estudiar els diferents mètodes de càlcul de Walker-Orin com el *Composite-Rigid-Body Method*.

Com a segona millora i tal i com s'ha mencionat en el informe corresponent a la part electrònica hem considerat realitzar el disseny d'un controlador de la força de manera que obtindrem un controlador híbrid de posició/força.

12. Referències

Marc teòric

- [1] <https://blogs.20minutos.es/ciencia-para-llevar-csic/2016/04/22/en-que-se-diferencia-un-robot-de-otros-tipos-de-maquinas/>
- [2] <https://www.areatecnologia.com/electronica/tipos-de-robots.html>
- [3] <https://revistabyte.es/actualidad-it/el-futuro-del-mercado-de-la-robotica-inteligencia-y-movilidad/>
- [4] <https://www.bbvaopenmind.com/articulos/la-robotica-los-materiales-inteligentes-y-su-impacto-futuro-para-la-humanidad/>
- [5] <https://www.sciencedirect.com/science/article/pii/S0921889016306285>
- [6] Article “*Soft Robotic*”: *Una nueva generación de robots* per José Medina H. i Paulina Vélez N.
- [7] <https://blog.infaimon.com/automatizacion-y-robotica-industrial/>
- [8] <https://www.tecnicaindustrial.es/la-robotica-industrial-en-el-ambito-de-la-aut/>
- [9] <https://www.interempresas.net/Robotica/Articulos/370816-Las-ventas-de-robots-suben-de-nuevo-segun-el-ultimo-informe-de-la-IFR.html>
- [10] <https://www.businessinsider.com/elon-musk-universal-basic-income-physical-work-choice-2021-8>
- [11] <https://www.interactanalysis.com/collaborative-robot-market-to-grow-by-60-in-2018/>
- [12] <https://wjla.com/news/nation-world/trump-vs-the-robots-inside-the-white-house-plan-to-save-us-jobs-from-automation>

Cinemàtica

- [13] Video per César Chávez -Ingeniería-: Cinemática inversa por método geométrico. Ejemplo sencillo y comprobación en Matlab.
- [14] Presentació per Jon Legarreta i Raquel Martinez: *Modelado Geométrico y Cinemático del Robot*

Dinàmica

- [15] Video per ET 4860 Robotic: *Robot Manipulato Dynamics*

- [16] Video per Javier Garcia: *2 – Mecánica Teórica [Euler-Lagrange 1]*
- [17] Video per Northwestern Robotics: *Modern Robotics, Chapter 8.1: Lagrangian Formulation of Dynamics (Part 1 of 2)*
- [18] Video per Ali Raza: *Dynamics of Robotic Manipulators – Part 1*

Llibres

- [19] Tercera Edició de *Robotica* per John J. Craig
- [20] Primera Edició de *Introducción a la Robótica* per Subir Kumar Saha

Viabilitat econòmica

- [21] <https://www.engineerscivils.cat/ca/node/4878>
- [22] <https://es.mathworks.com/pricing-licensing.html>
- [23] <https://www.microsoft.com/es-es/microsoft-365/p/microsoft-365-personal/cfq7ttc0k5bf?activetab=pivot%3aoverviewtab>

Planificació

- [24] <https://blog.infraspeak.com/es/analisis-fmea-amfe/>
- [25] <https://apalm.atlassian.net/jira/core/projects/TFG1/board>
- [26] <https://www.notion.so/>

Informació Matlab Online

- [27] <https://matlab.mathworks.com/>
- [28] <https://es.mathworks.com/help/robotics/ref/rigidbodytree.html>
- [29] <https://es.mathworks.com/help/robotics/ref/inversekinematics-system-object.html>

Dinàmica directa

- [30] Article “*Efficient Dynamic Computer Simulation of Robotic Mechanisms*”
per M. W. Walker i D. E. Orin