

## Resum

El Projecte de Fi de Carrera desenvolupat consisteix en l'automatització d'un robot mèdic dissenyat per a operacions de laparoscòpia per tal d'ajudar al cirurgià davant tota una sèrie de limitacions amb les que es pot trobar, com per exemple la pèrdua de visió directa, la pèrdua de percepció tàctil i les restriccions de moviments. La robòtica i els sistemes de control i sensorials poden compensar en gran part aquestes carències.

Aquest treball es centra en el disseny d'un controlador robust amb l'objectiu de moure la càmera de vídeo durant l'operació per tal que el metge tingui una visió el més òptima i precisa possible de la zona intraabdominal en la que està realitzant la intervenció. S'ha realitzat la síntesi d'un control robust tenint en compte les incerteses del robot ja que aquest és un sistema no lineal i variant en el temps.

Per tal de resoldre el problema de control, prèviament esmentat, s'ha modelitzat el robot i s'ha desenvolupat un simulador que descriu de manera completa i global la dinàmica real del sistema. Després de realitzar un estudi i anàlisi del robot a partir d'un model de control o model simplificat, s'ha desenvolupat i programat un algoritme de control robust amb estructura PID ja que aquesta és fàcil d'implementar i la més emprada a la indústria. A continuació s'han presentat els resultats de simulació i s'ha observat i comprovat la "bondat" del control utilitzat, ja que es garanteix l'estabilitat robusta i les especificacions (precisió i rapidesa) per tota l'àrea de treball del robot.





## Sumari

<b>Resum</b> .....	<b>1</b>
<b>Sumari</b> .....	<b>3</b>
<b>1. Glossari</b> .....	<b>5</b>
<b>2. Prefaci</b> .....	<b>7</b>
<b>3. Introducció</b> .....	<b>9</b>
<b>3.1. Objectius del Projecte</b> .....	<b>9</b>
<b>3.2. Distribució de la Memòria</b> .....	<b>9</b>
<b>4. La robòtica en la medicina</b> .....	<b>11</b>
<b>4.1. Antecedents i evolució</b> .....	<b>11</b>
<b>4.2. La laparoscòpia</b> .....	<b>12</b>
<b>5. El Problema de Control</b> .....	<b>14</b>
<b>5.1. Introducció</b> .....	<b>14</b>
<b>5.2. Descripció de l'equip de laparoscòpia</b> .....	<b>14</b>
5.2.1. La planta: El robot.....	15
5.2.2. Unitat de control.....	16
5.2.3. Sensor de visió: L'algoritme de visió.....	17
<b>5.3. El problema de control</b> .....	<b>21</b>
<b>5.4. Solució adoptada: Control PID robust</b> .....	<b>22</b>
<b>6. Modelització del robot de laparoscòpia</b> .....	<b>23</b>
<b>6.1. Introducció</b> .....	<b>23</b>
<b>6.2. Model cinemàtic</b> .....	<b>23</b>
6.2.1. Model cinemàtic directe.....	23
6.2.2. Algorisme de Denavit-Hartenberg.....	24
6.2.3. Matriu de paràmetres de Denavit-Hartenberg.....	25
6.2.4. Matrius de Transformació.....	26
6.2.5. Model cinemàtic invers.....	28
<b>6.3. Model dinàmic</b> .....	<b>30</b>
6.3.1. Algorisme computacional de Lagrange-Euler.....	31
<b>6.4. Validació del simulador del robot</b> .....	<b>35</b>
6.4.1. Cas 1.....	36
6.4.2. Cas 2.....	37
6.4.3. Cas 3.....	39
6.4.4. Cas 4.....	41
<b>6.5. Linealització</b> .....	<b>43</b>
6.5.1. Linealització mitjançant Taylor.....	44
6.5.2. Model linealitzat en el punt nominal.....	45
6.5.3. Validació del model simplificat.....	46
<b>7. El controlador robust</b> .....	<b>50</b>
<b>7.1. Control convencional. Els controladors PID's</b> .....	<b>50</b>
7.1.1. Algoritme per trobar PID's que estableixen una planta lineal.....	51



7.1.2. Exemples .....	53
<b>7.2. Control robust .....</b>	<b>56</b>
7.2.1. Introducció .....	56
7.2.2. Objectius de control .....	56
7.2.3. Representació de les incerteses .....	57
<b>7.3. PID's robustos. Un compromís entre la simplicitat i el garantir l'estabilitat i la robustesa .....</b>	<b>59</b>
7.3.1. Algoritme per trobar PID's que estableixen una planta amb incertesa .....	60
7.3.2. Exemples .....	64
<b>8. Control del robot de laparoscòpia .....</b>	<b>65</b>
<b>8.1. Introducció .....</b>	<b>65</b>
<b>8.2. Problema de control .....</b>	<b>65</b>
<b>8.3. Model de control i de les incerteses .....</b>	<b>65</b>
<b>8.4. Disseny del controlador robust descentralitzat .....</b>	<b>70</b>
8.4.1. El problema de l'acoblament .....	70
8.4.2. Control descentralitzat .....	72
8.4.3. Aplicació de l'Algorisme .....	73
Articulació 1 .....	73
Articulació 2 .....	74
Articulació 3 .....	75
<b>8.5. Selecció de les constants del PID .....</b>	<b>76</b>
8.5.1. Overshoot (sobrepuig màxim) .....	78
8.5.2. Temps d'establiment .....	80
8.5.3. Valors elegits .....	82
<b>8.6. Comprovació de l'estabilitat del sistema global i de l'estabilitat robusta .....</b>	<b>82</b>
<b>8.7. Simulació del llaç tancat .....</b>	<b>84</b>
8.7.1. Simulació 1 .....	85
8.7.2. Simulació 2 .....	89
8.7.3. Simulació 3 .....	92
<b>8.8. Resultats de les simulacions .....</b>	<b>96</b>
<b>Conclusions .....</b>	<b>97</b>
<b>Agraïments .....</b>	<b>98</b>
<b>Bibliografia .....</b>	<b>99</b>
Referències bibliogràfiques .....	99
Bibliografia complementària .....	100



## 1. Glossari

- EDO: Equació Diferencial Ordinària.
- EDP: Equació en Derivades Parcial.
- HF (*High Frequencies*): altes freqüències
- MCD (*Model Cinemàtic Directe*): model que permet trobar la posició i orientació de l'últim element d'un robot a partir de les variables articulars.
- MCI (*Model Cinemàtic Invers*): model que permet trobar les variables articulars d'un robot a partir de la posició i orientació de l'últim element.
- MIMO (*Multiple Input/Multiple Output*): sistema multivariable, és a dir, amb moltes entrades i moltes sortides.
- PID (*Proporcional/Integral/Derivatiu*): controlador que es basa en tres accions, la Proporcional, la Integral i la Derivativa.
- SI: Sistema Internacional.
- SISO (Single Input/Single Output): sistema monovariable, és a dir, amb una entrada i una sortida.
- TCP (*Tool Center Point*): punt terminal del robot, a on es troba l'eina.





## 2. Prefaci

Aquest Projecte suposa la continuació de la feina realitzada al llarg de diversos Projectes de Fi de Carrera i estudis relacionats amb el Robot de Laparoscòpia construït en el Departament d'Automàtica i Informàtica Industrial (ESAI) de la Universitat Politècnica de Catalunya.

Principalment, aquest Projecte és la continuació i ampliació dels següents estudis i treballs:

- *Disseny d'un robot de quatre graus de llibertat*, de l'alumne José Miguel Ibáñez Borau, on es dissenya i s'implementa el maquinari de control que permet la interacció amb el robot. [J. Miguel Ibáñez, 1997]
- Patent de la construcció d'un braç de robot retràctil per part del Dept. d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAI). [Amat J., 2001]
- *Automatic Guidance of a Laparoscope Using Computer Vision*, desenvolupat per en Joakin Robert i presentat el novembre del 2002, en el qual es desenvolupa un programa de visió per computador per tal de fer la detecció i seguiment de les eines que s'usen durant la intervenció [Robert J., 2002].

El Projecte pretén doncs continuar la tasca començada anteriorment i té el següent propòsit: realitzar un sistema de control del robot de laparoscòpia per tal que aquest segueixi adequadament la consigna que rebí de l'Algorisme de Visió.







### 3. Introducció

#### 3.1. Objectius del Projecte

L'objectiu general d'aquest Projecte Final de Carrera és el de realitzar el servo control visual d'un robot mèdic especialment dissenyat per a ajudar en operacions quirúrgiques de laparoscòpia. El robot destinat a aquest tipus d'operació està pensat per subjectar i moure la càmera de vídeo durant l'operació de tal manera que la visió que tingui el cirurgià de la zona en la que s'està operant sigui l'òptima.

Per tal d'aconseguir l'anterior propòsit s'ha de resoldre un problema d'enginyeria de control (*'solving problem'*): el seguiment ràpid i precís de la posició de les eines de cirurgia laparoscòpica per tal de minimitzar el risc de que el pacient pateixi qualsevol tipus de lesió.

Aquest objectiu general requerirà assolir els següents objectius específics:

- Desenvolupar un model matemàtic que descriu la dinàmica completa del model real.
- Formular un model de control, adient pel disseny del controlador, que representi els aspectes essencials del comportament del robot.
- Desenvolupar un algorisme de control robust per a controladors amb estructura PID que permeti trobar el ventall de controladors que estableixen una determinada planta lineal, coneguda la incertesa existent.
- Aplicació d'aquest algorisme al model de control del robot per tal de trobar un controlador adient.
- Realitzar un simulador que permeti simular el comportament del sistema en llaç tancat i comprovar el bon funcionament del controlador trobat.

#### 3.2. Distribució de la Memòria

En el Capítol 4 es donarà una visió general de com ha influït la robòtica en el camp de la medicina i s'explicarà en què consisteix una operació de laparoscòpia.

En el Capítol 5 es formularà el Problema de Control que es pretén solucionar, realitzant-se una descripció de l'equip de laparoscòpia del qual es disposa i explicant quin és el objectiu general que es vol assolir.

En el Capítol 6 es formularà el model complet del robot a partir de lleis físiques, al qual se li aplicaran un seguit de simplificacions per tal de facilitar l'anàlisi del sistema i el disseny analític del control. A més, aquest model simplificat es linealitzarà al voltant del punt de treball nominal. Tot seguit es procedirà a la seva validació, comparant el comportament d'ambdós models (és a dir, el linealitzat amb el complet).

Després, en el Capítol 7, s'explicarà el control robust i es presentarà la implementació d'un algorisme que, donat un sistema de control, permet trobar un ventall de controladors PID's que garanteixen que aquest sigui estable.

Finalment en el Capítol 8, es trobaran les regions de controladors que estableixen el robot i es seguirà amb l'elecció dels valors de  $K_p$ ,  $K_i$  i  $K_d$  que facin que el comportament del sistema sigui el més adient, això és, que compleixi un seguit d'especificacions com són precisió o velocitat de resposta. Per últim, es desenvoluparà un Simulador (del model complet formulat en el Capítol 6) que permeti simular el sistema en llaç tancat i s'avaluarà el funcionament del sistema.





## 4. La robòtica en la medicina

### 4.1. Antecedents i evolució

La convergència dels descobriments informàtics i de comunicacions amb la medicina en general i la cirurgia en particular ha donat com a resultat un accelerat procés d'informatització de totes les àrees de la medicina.

De fet, dintre de les diverses aplicacions de la robòtica en la medicina destaca la cirurgia. Les primeres aplicacions d'aquest tipus daten de l'any 1982. En aquest any es va començar a Califòrnia un programa l'objectiu del qual era utilitzar un robot comercial (Puma 260) per realitzar operacions de neurocirurgia. Des d'aquell moment, s'han posat a punt diversos sistemes que, amb l'ajut d'un *scanner*, permeten registrar tota la informació necessària del cervell per tal que l'equip mèdic decideixi el punt exacte on s'ha de realitzar la incisió i que, posteriorment, aquesta sigui feta pel robot.

Dintre del camp de la cirurgia, també és molt important l'evolució de les tècniques de cirurgia laparoscòpia o cirurgia de mínima invasió. La primera intervenció d'aquesta naturalesa va tenir lloc a Alemanya l'any 1987, i donat el seu èxit va originar una revolució en les formes d'intervenir quirúrgicament en diverses especialitats, motivant una investigació intensa en tots els camps relacionats.

Els sistemes de laparoscòpia convencional continuaren la seva evolució fins a donar lloc a sistemes de tercera dimensió. Poc després es va començar a introduir l'ús de robots o braços mecànics que, sota les ordres del cirurgià, intervenien quirúrgicament a pacients en les àrees de cirurgia general, cirurgia cardiovascular, pediàtrica, ortopèdia, urologia i neurocirurgia.

Els robots poden ser controlats a distància per metges especialistes, donant lloc a la cirurgia robòtica, telepresència, telecirurgia o cirurgia assistida per computadors. Aquesta es basa en dos conceptes fonamentals, realitat virtual i cibernètica, i possiblement serà un dels avenços quirúrgics més importants del segle XXI.

La robòtica mèdica pretén comptabilitzar el cirurgià amb el robot per tal de millorar els procediments quirúrgics, compensant les deficiències i limitacions que pot tenir el cirurgià al realitzar determinades accions. El robot, entre d'altres funcions, pot ajudar en la percepció, proporcionant al cirurgià informació útil que no podria obtenir sense el seu ajut, i pot memoritzar posicions i accedir a un punt determinat amb una precisió superior a la d'una persona. És a dir, l'ús del robot connectat a un ordinador permet que la incisió quirúrgica es dugui a terme amb la màxima precisió i en un temps notablement inferior que si es fes amb el sistema habitual.

Per tant, l'ús de la robòtica i de la informàtica aplicada a les ciències de la salut ofereix, entre els seus principals beneficis, la possibilitat d'aconseguir una cirurgia mínimament invasiva, millorar l'exactitud d'algunes intervencions i permetre, fins i tot, la cirurgia a distància. A més a més, es descarrega al cirurgià de la mecànica de certes tasques com el correcte posicionament dels instruments de cirurgia, permetent una major concentració en el seguiment i control d la operació.

La robòtica quirúrgica té un seguit de particularitats que la diferencien de la robòtica industrial. El *software* és molt diferent, mentre que en la indústria es



programa un robot per tal que aquest tan sols realitzi una acció de forma repetitiva, en la robòtica quirúrgica el robot ha d'interactuar amb el cirurgià. En aquest sentit, és bàsic aconseguir que el metge senti la tasca que el robot està duent a terme, que ambdós actuïn com si fossin un de sol.

## 4.2. La laparoscòpia

La laparoscòpia és una tècnica de cirurgia intraabdominal mínimament invasiva que substitueix la visió directe de l'ull del cirurgià per un sistema òptic de visualització indirecta a distància, generalment a través d'un monitor de TV on es projecta la imatge de la zona que s'està operant. Es pot observar una fotografia d'una operació de laparoscòpia en la Figura 4.1:



Figura 4.1: Imatge d'una operació de laparoscòpia

El procediment és el següent (observar Figura 4.2): primer es realitza una petita incisió a la pell i a través d'ella s'introdueix a la cavitat abdominal un gas inofensiu com el Diòxid de Carboni, que s'expandeix creant una gran àrea de treball. Les operacions són les mateixes que es practiquen mitjançant la tècnica oberta, amb l'avantatge de realitzar-les a través d'uns petits orificis de 5 a 10mm que posteriorment es fan a la cavitat abdominal, pels quals s'introdueixen els instruments i les eines necessàries (veure Figura 4.3) i la càmera de vídeo, fet que permet operar amb un mínim de dany als teixits.

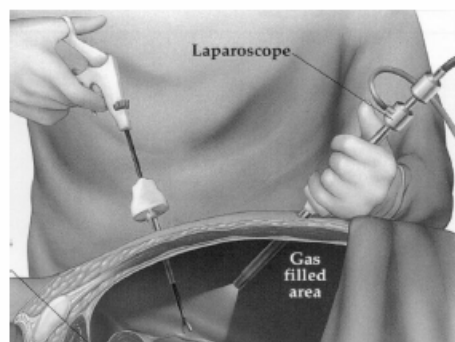
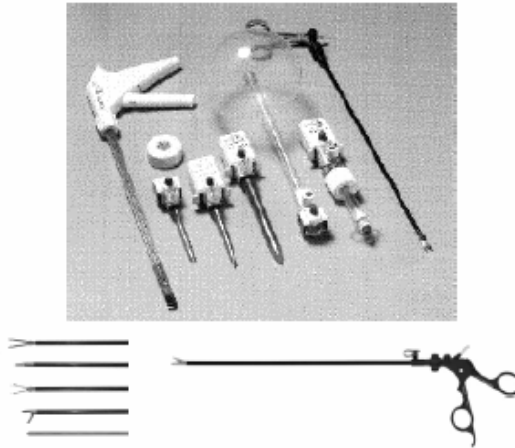


Figura 4.2: Laparoscòpia

És essencial, per tant, que les imatges que es vegin en el monitor de TV donin una visió clara de l'àrea on el cirurgià està treballant (veure Figura 4.4). En els orígens d'aquesta tècnica, era el mateix cirurgià qui movia la càmera, però això resultava molest i no permetia que es centrés en la tasca principal, l'operació. Una altra opció que soluciona aquest problema és que un ajudant s'encarregui de moure la càmera,

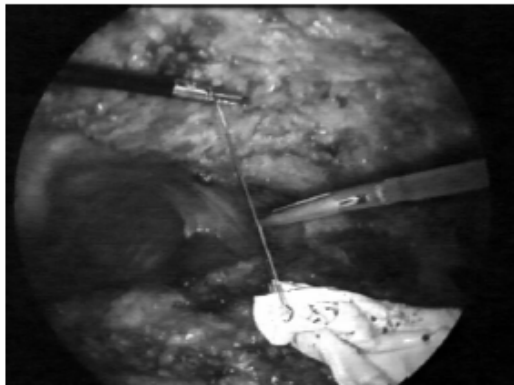


però això provoca també problemes degut a que la coordinació entre el cirurgià i l'ajudant no sempre és perfecte.



*Figura 4.3: Eines usades en una operació de laparoscòpia*

La idea és doncs usar un programa d'ordinador que analitzi la imatge rebuda per la càmera amb l'objectiu de saber quina és l'àrea en la qual el cirurgià vol treballar, i que el robot, mitjançant un algoritme de control adequat, s'encarregui de situar la càmera en la posició més adient. Per tal de saber quina és la zona de treball adequada, el programa de visió tindrà que saber on estan situades les eines per tal que el robot pugui centrar la imatge de la càmera.



*Figura 4.4: Visió de la càmera*



## 5. El Problema de Control

### 5.1. Introducció

Actualment cada vegada és major el nombre de robots que es van incorporant a la indústria, al sector de serveis, a la medicina, etc. No obstant això, l'aplicació d'un robot per a realitzar una determinada tasca depèn en un elevat percentatge del coneixement *a priori* que es té de l'espai de treball i de la localització dels objectes a manipular. Aquesta limitació és deguda a que els robots industrials comercials no integren sistemes sensorials, que lis permetin adaptar-se al seu entorn.

Dins dels sistemes sensorials adaptables als robots industrials comercials, els sistemes de visió artificial tenen com avantatge principal mimetitzar el sistema de visió humà i permetre, entre d'altres coses, obtenir mesures de l'espai de treball del robot sense necessitat de contacte físic.

Ja en 1973 [Shirai Y., 1973] es fa referència al control visual d'un manipulador descrivint com es pot utilitzar la retroalimentació visual (*visual feedback*) per tal de corregir la posició d'un robot incrementant amb això l'exactitud de les tasques d'ensamblatge. A finals dels 70 el SRI (*Stanford Research Institute*) International va realitzar algunes aplicacions en les que es va utilitzar retroalimentació visual per a la inserció de cargols i la recollida de peces d'una cinta transportadora. En l'actualitat es ja disposen de robots comercials amb controladors on s'integra la informació que prové d'un sistema de visió. Però, sobretot, on s'utilitzen més aquests sensors és en la robòtica mèdica, com per exemple en les intervencions de laparoscòpia.

Va ser en 1979 [Hill, J., 1979] quan es va introduir el concepte de controlar la posició d'un robot en llaç tancat mitjançant la informació visual, apareixent per primera vegada el terme "servo control visual" (*visual servoing*). S'entén per servo control visual el control a nivell dels servos del robot mitjançant la informació visual proporcionada pel sistema de visió.

En aquest capítol s'explica el plantejament del problema de control i una possible solució (que és la que nosaltres hem escollit) per el control de posició d'un robot laparoscòpic que rep informació d'un sensor de visió.

### 5.2. Descripció de l'equip de laparoscòpia

L'equip de laparoscòpia del que es disposa al laboratori de robòtica del Departament ESAII és un prototipus desenvolupat per aquest departament de la UPC a l'any 2001 [Amat, J., 2001]. Va ser dissenyat i implementat com a sistema de control per tal de poder aplicar nous algorismes de control i de visió i provar d'altres ja desenvolupats.

Aquest equip de laparoscòpia, que serveix per ajudar en operacions de laparoscòpia mitjançant el moviment d'una càmera, és el sistema que s'estudiarà, analitzarà i controlarà i està format pels següents elements (veure Figura 5.1):

- La planta, que és el robot de tres graus de llibertat.



- La unitat de control que està constituïda pel maquinari de control, el programari de control (*software* de l'algoritme de control) i el maquinari de potència.
- Sensor que és la càmera que necessita de l'algoritme de visió per tal de detectar les eines utilitzades pel cirurgià.

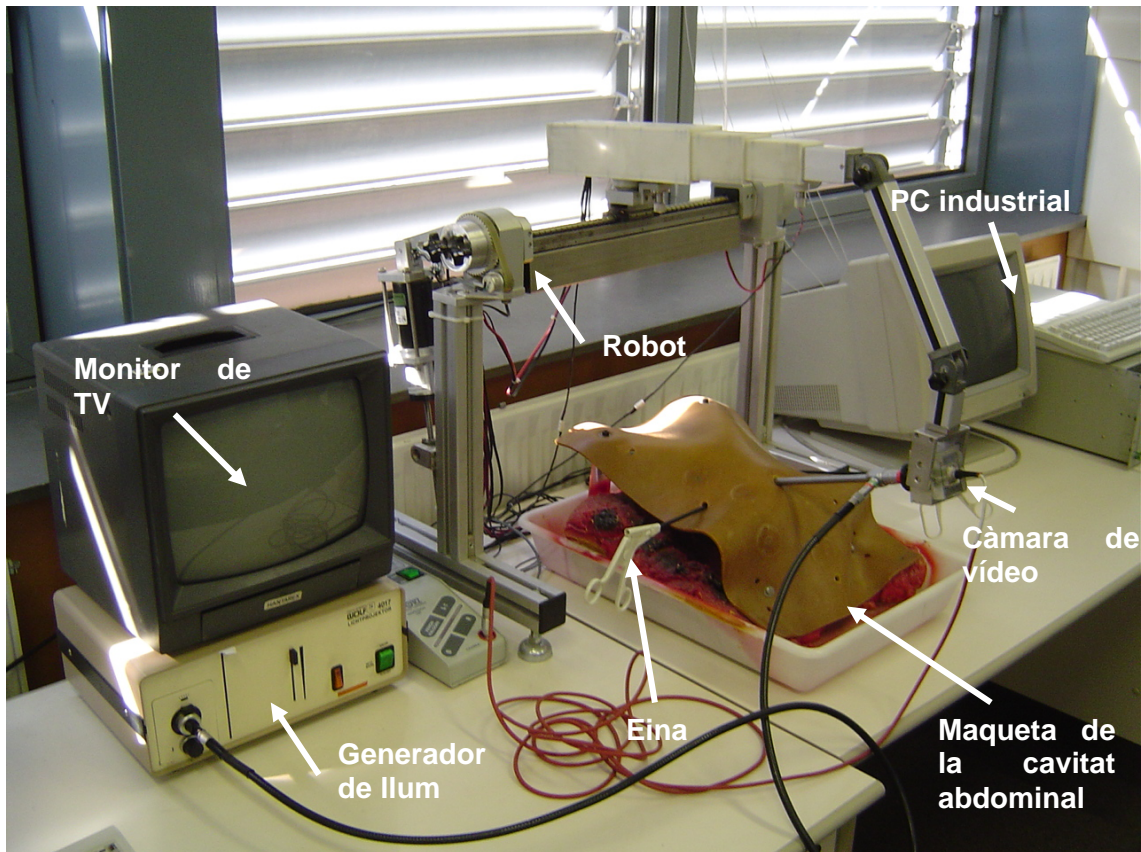


Figura 5.1. Equip de laparoscòpia del que es disposa.

### 5.2.1. La planta: El robot

S'entén per configuració mecànica del robot el conjunt d'elements fixes units els uns als altres mitjançant articulacions. Aquestes articulacions són les que donen capacitat de moviment al robot, i suporten els actuadors, els quals mouen físicament l'articulació. L'element terminal o TCP (*Tool Center Point*) és el que interactuarà amb l'entorn de treball del robot, i a tal efecte serà necessari situar-lo i orientar-lo en l'espai. Tot i que en teoria serien necessaris sis graus de llibertat (tres per obtenir la posició desitjada i tres més per orientar el TCP un cop assolit aquesta posició), la majoria de tasques es poden realitzar amb menys graus de llibertat i per tant es pot simplificar considerablement la complexitat del robot.

El robot de laparoscòpia té tres graus de llibertat, corresponents a les tres articulacions, tal i com es veu en la Figura 5.2. La primera articulació és una articulació de rotació, mentre que les dues següents són articulacions prismàtiques, i per tant produeixen un desplaçament lineal. Les dues articulacions prismàtiques han estat dissenyades amb un accionament telescòpic de tal manera que l'espai



ocupat no és constant, sinó que augmenta quan el robot vol abastar una posició més llunyana a l'origen de l'articulació en concret. La configuració del robot es pot observar en la següent fotografia:

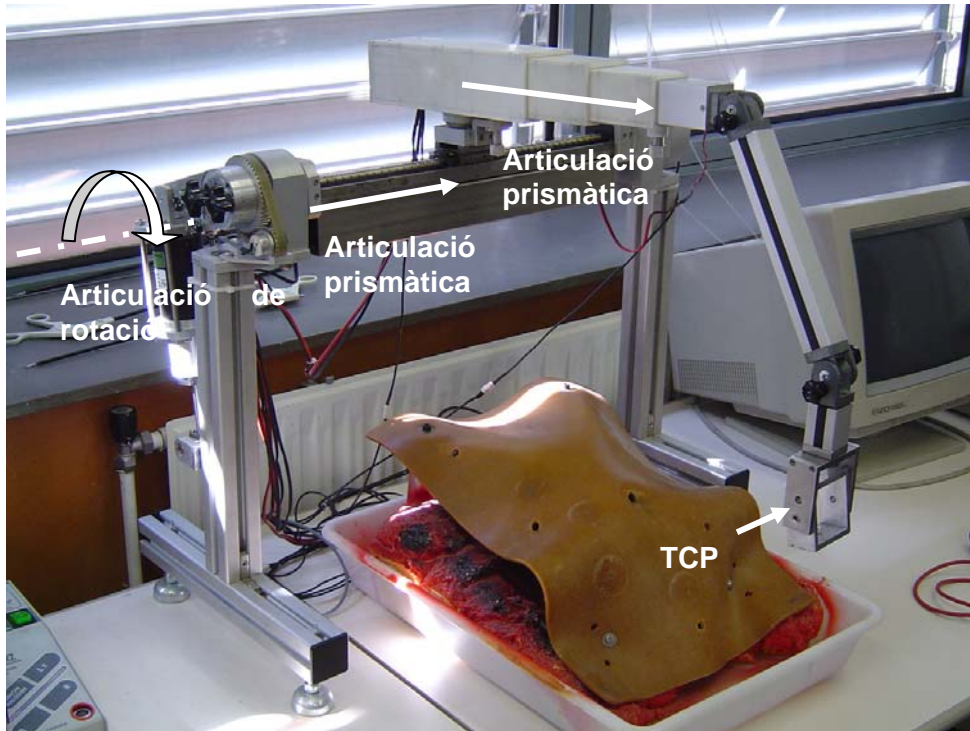


Figura 5.2: Els tres graus de llibertat del robot

### 5.2.2. Unitat de control

La unitat de control d'un robot és el conjunt de maquinari i programari destinat a controlar el robot a tots els nivells; des de la planificació de tasques en funció d'events externs, a la generació de senyals electrònics que després de passar per etapes de potència arribaran als motors provocant el moviment d'aquests i fent que el robot realitzi la tasca programada.

Dins la unitat de control hi ha tres blocs conceptualment diferents segons la tasca que es realitza, que són el maquinari de control, el programari de control, i el maquinari de potència. Tot això es complementa amb interfícies de cara a l'usuari i d'altres sistemes mitjançant dispositius de comunicacions específics, que permeten controlar el robot a tots els nivells.

El maquinari de control és tot el maquinari dins la unitat de control que suporta el programari de control, recull dades sensorials del robot i del seu entorn, i en funció d'aquestes genera els senyals de control que provoquen en la estructura mecànica del robot els moviments adients per que aquest realitzi el que es desitja i que estava prèviament programat i determinat. És a dir, el programari de control és el *software* de control (programa de l'algorisme de visió). El disseny del controlador a partir del model de control formulat és el punt principal per tal que el robot realitzi la tasca desitjada amb la rapidesa i la precisió necessàries, i sempre assegurant l'estabilitat del sistema de control.





Com a unitat de control del robot hi ha un ordinador tipus PC amb característiques industrials, al que s'han afegit les targetes adients que donen les funcionalitats específiques que requereix una unitat de control, i que el PC industrial no té.

Aquest ordinador és l'encarregat de suportar el programari de control i les aplicacions específiques per la tasca a realitzar pel robot. La utilització d'un PC ofereix totes les possibilitats d'aquest tipus de màquines, incorporant a més a més dispositius específics que augmenten el seu rendiment i que són de gran utilitat en aplicacions industrials.

L'equip té una velocitat d'execució que permet el control en temps real dels motors del robot, respon també en temps real a possibles eventualitats que puguin sorgir, i garanteix el bon funcionament de tot el programari de control. Habitualment les unitats de control de robots estan basades en processadors paral·lels que permeten la execució multi tasca, per tal de rentabilitzar al màxim el programari de control. Si no és així es requereix de un sistema operatiu en temps real que ofereixi aquest paral·lelisme. En aquest cas el sistema operatiu és MS-DOS 6.22; així doncs és el programari de control qui s'encarregui d'oferir la possibilitat d'execució en temps real i simular la multi tasca quan aquesta sigui necessària.

### 5.2.3. Sensor de visió: L'algoritme de visió

L'algoritme de visió consisteix en un complex algoritme amb subprogrames que mitjançant filtres d'imatge és capaç de detectar les dues eines utilitzades pel cirurgià, i determinar el punt mig d'ambdues. Aquest algoritme va ser desenvolupat amb anterioritat [Robert J, 2002] i en aquest Projecte es crearà un simulador d'aquest algoritme per tal de comprovar el funcionament del sistema de control implementat.

El simulador de l'algoritme de visió, a través de la posició del TCP del robot haurà de trobar el centre de la imatge (és a dir a on està apuntant la càmera) i amb la posició de les eines retornar els valors que retornaria l'algoritme de visió, que és la posició relativa al centre de la imatge, la posició de les eines ( $X_d$ ,  $Y_d$ ).

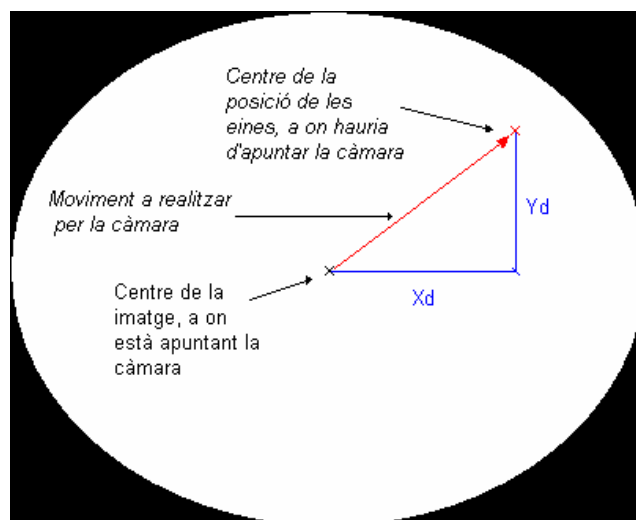


Figura 5.3: Algoritme de visió



## Simulació i funcionament de l'algorisme de visió

El funcionament de l'algorisme consisteix en retornar el vector de posició del centre de les eines respecte el centre de la imatge ( $X_d$ ,  $Y_d$ ). Això és possible degut al fet que la càmera es troba subjectada al robot en el TCP a través d'una eina que li permet tenir dos graus de llibertat respecte del robot, que són dos angles d'Euler: el primer, una rotació lliure respecte la vertical, i el segon, després d'haver realitzat la primera, una rotació respecte l'eix horitzontal ortogonal a la càmera. Això permet mantenir la horitzontalitat i verticalitat de les imatges mostrades per l'algorisme. Operant de forma semblant a [P Berkelman et al., 2003], a la Figura 5.4 es poden observar els dos angles d'Euler  $\psi_1$  i  $\psi_2$  corresponents al primer i al segon respectivament:

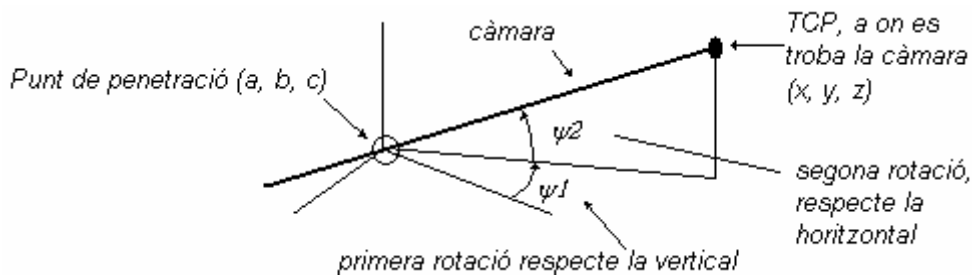


Figura 5.4. Graus de llibertat de la càmera

Com s'ha comentat anteriorment, l'algorisme de visió no funciona correctament, i s'ha realitzat una simulació del mateix. Per a la simulació es necessari el coneixement de la posició de tres punts:

- Posició actual del TCP ( $x, y, z$ ).
- Punt de penetració i pivotament de la càmera,  $(a, b, c)$ .
- Punt a on es troba actualment el centre de les eines.

Prenent un pla per tal de projectar els punts, que correspon amb el pla del fons d'imatge de la càmera, es troben les projeccions de les eines en el pla fix ( $X_e, Y_e$ ), i construint la recta que passa pels punts  $(x, y, z)$  del TCP i  $(a, b, c)$  del punt de penetració, i realitzant la intersecció amb el pla fix es troba el punt  $(X, Y)$ , corresponent al punt a on està enfocada la càmera. Ara disposem dels dos punts:

- $(X_e, Y_e)$  projecció de la posició de les eines sobre el pla fix.
- $(X, Y)$  el centre de la imatge projectat en el pla fix (a on apunta la càmera).

El pla utilitzat per a realitzar les projeccions és el pla ortogonal a la càmera en la posició inicial a una distància de 10 [cm] en l'eix de la càmera respecte el punt de penetració, tal com mostra la projecció de la Figura 5.5:



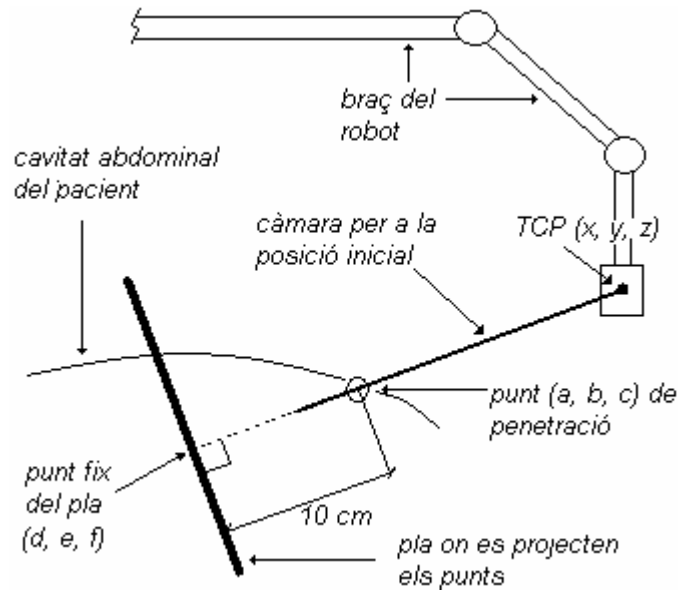


Figura 5.5. Representació de la càmera i el pla fix

Els punts (a, b, c) i les coordenades del TCP per a la posició inicial són els mostrats a continuació:

$$(a, b, c) = (0.0385, 0.1552, 0.3000) \text{ [cm]}$$

$$(x, y, z)_{\text{inicial}} = (0.115, 0.34, 0.3) \text{ [cm]}$$

Amb aquests valors, es pot trobar la equació del pla fix:

$$0.0765x + 0.1848y - 0.011625 = 0, \text{ x i y en [cm]}$$

El punt fix del pla que correspon amb el centre de la imatge de la càmera en la posició inicial és:

$$(d, e, f) = (2.5000e-004, 0.0628, 0.3000) \text{ [cm]}$$

Ara només cal trobar la diferència entre  $(X_e, Y_e)$  i  $(X, Y)$  per tal d'obtenir la posició relativa de les eines respecte del centre de la imatge, i de tenir simulat l'algoritme de visió, tal com mostra la figura:

$$(X_d, Y_d) = (X_e, Y_e) - (X, Y)$$



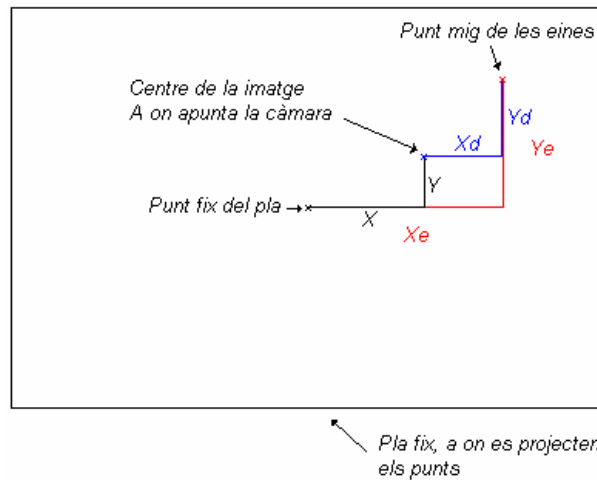


Figura 5.6: Senyal d'error en el pla fix

El resultat mostrat per l'algorisme, és a dir el que retornaria l'algorisme de visió, és pot observar a la Figura 5.3. Per a simular aquest algorisme s'ha dissenyat un bloc de SIMULINK que s'explica a l'Annex A.

### Determinació de la nova posició del TCP, transformació de 2D a 3D

Per a trobar la nova posició del TCP, és necessari extreure la informació de les coordenades (Xd, Yd) retornada amb l'algorisme de visió. Un primer punt a tenir en compte és el fet que el TCP he d'estar ubicat a una distància constant respecte del punt de penetració (a, b, c), per tal d'evitar que la canya de la càmera entri i surti constantment del pacient, en el cas tractat aquesta distància és anomenada  $l_2$  i li correspon una longitud de 20 [cm].

El funcionament consisteix en transformar les distàncies retornades per l'algorisme de visió (Xd, Yd) en dos angles que es mesuraren des del punt de penetració (a, b, c). Amb aquests dos angles, només cal rotar el vector posició des del punt (a, b, c) fins al punt actual (x, y, z) per a trobar la nova posició (x, y, z)' del TCP. Es pot veure esquematitzat en la Figura 5.9:

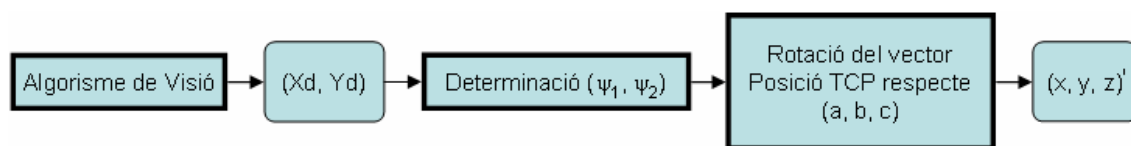


Figura 5.7: Transformació 2D-3D

El moviment de la càmera amb les posicions actual i futura es mostra a continuació:



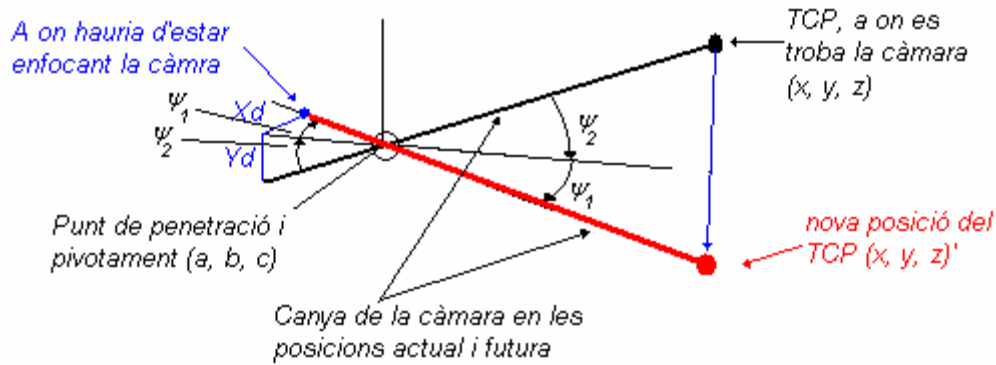


Figura 5.8. Representació del moviment de la càmera

Seguint aquest procés s'ha dissenyat un bloc anomenat 2D-3D que té per funció retornar la nova posició (x, y, z) a on hauria d'estar el TCP a partir de la senyal (Xd, Yd) i de la posició actual del TCP (x, y, z). El bloc es descriu en l'Annex A.

### 5.3. El problema de control

El problema de control que es planteja en aquest Projecte és el de guiar de manera efectiva el robot durant una intervenció de laparoscòpia de tal manera que la posició de la càmera sigui en tot moment la que permeti una millor visió al cirurgià de la zona que està operant. De fet, l'opció que es planteja en aquest Projecte és que un robot s'encarregui de moure la càmera. Aquesta és la millor solució, donat que s'eviten els possibles errors humans derivats de realitzar una mateixa tasca durant molt de temps.

El funcionament del sistema de control del Robot es pot esquematitzar a partir del següent diagrama de flux (Figura 5.9):

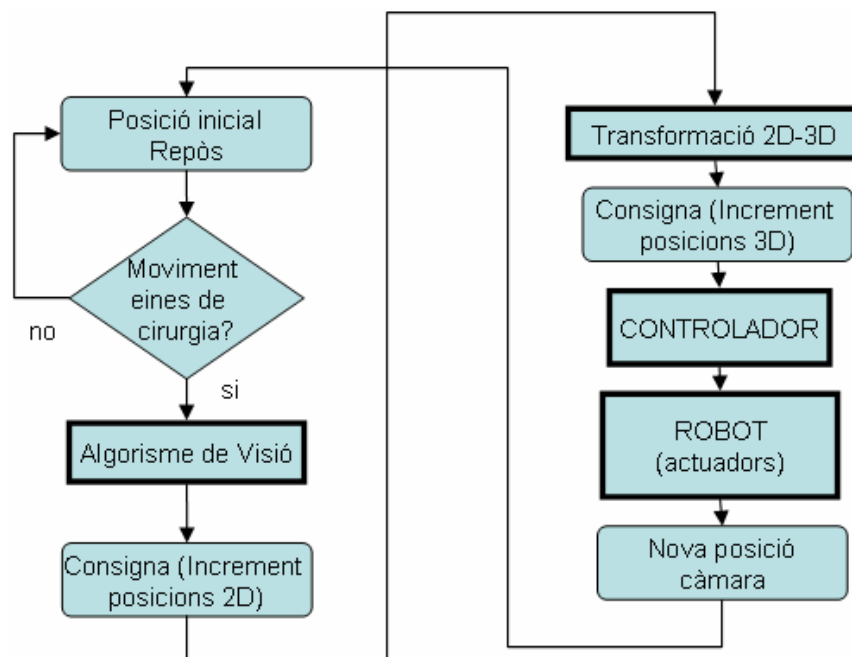


Figura 5.9: Diagrama de flux de les tasques que realitza el sistema de control del robot



El robot, i per tant la càmera, parteix d'una posició de repòs. En el moment en què hi ha un moviment de les eines usades pel cirurgià, l'Algorisme de visió, a través de les imatges captades per la càmera, calcula la nova posició que hauria de tenir el TCP per tal que la visió de les eines a través del monitor de televisió sigui la òptima. L'Algorisme de visió retorna aquesta posició en forma de dues coordenades, que representen el nou punt on hauria d'enfocar la càmera, això és, cap on s'hauria de moure. Aquestes senyals són una projecció 2D sobre el pla de visió, i per tant de cara a controlar el robot mitjançant el moviment de les seves tres articulacions s'han d'expressar en l'espai. Això és el que fa la Transformació 2D-3D.

Les tres senyals obtingudes (que són l'increment de les tres variables articulars necessaris) es controlaran mitjançant el Controlador, per tal de que el moviment del robot compleixi un seguit d'especificacions (com són el compliment de l'estabilitat, temps d'establiments adequats, *overshoots* petits...). Les senyals controlades són les que actuaran sobre els actuadors del Robot possessionant la càmera en el punt desitjat.

#### **5.4. Solució adoptada: Control PID robust**

Cal destacar que existeixen moltes fonts d'incertesa que influeixen en la dinàmica d'un robot com ara la variació de la càrrega i de la fricció, la dinàmica no modelada, les pertorbacions exteriors, etc. Per a tenir un comportament robust i el més insensible possible front la incertesa hem d'utilitzar el que s'anomena control robust. Com aquesta metodologia de control proporciona controladors d'alt ordre i difícils d'implementar, s'ha escollit com a controladors a aplicar, els PID's robustos, la síntesi dels quals està explicada al Capítol 7 que es basa en l'algorisme de control proposat en [Ming-Tzu Ho, 2003].



## 6. Modelització del robot de laparoscòpia

### 6.1. Introducció

Per realitzar el control d'un sistema de qualsevol tipus (mecànic, hidràulic, elèctric, etc.) es necessita conèixer el seu comportament i, per tant, és necessari comptar amb models que ens donin l'anterior informació. Per obtenir una descripció detallada i completa del sistema a controlar normalment s'utilitzen models complexos representats mitjançant EDP<sup>1</sup>, EDO<sup>2</sup>, etc., que es poden resoldre per mètodes analítics o bé mètodes numèrics. Aquests models complexos que descriuen fidelment i amb precisió el comportament del sistema físic s'utilitzen com a simuladors, com si fossin el sistema real (substituint-lo). Però, per fer la síntesi del controlador ens calen models matemàtics que siguin precisos però a la vegada senzills per tal de poder emprar les eines d'anàlisi i síntesi. Per tant, voldrem obtenir un model de control senzill, precís i robust a les incerteses mitjançant el qual es calcularà el controlador pel control del nostre sistema. Els models de control es poden obtenir de dues maneres: 1) teòricament, és a dir, a partir de les lleis físiques (modelació teòrica), o bé 2) experimentalment, és a dir, a partir d'experiments (identificació). En aquest Projecte el model es trobarà per modelació teòrica.

Per a l'estudi, anàlisi i posterior del robot de laparoscòpia descrit en el Capítol 5 es requereix l'obtenció de dos models de diferents tipus: 1) el model cinemàtic (directe i invers) i, 2) el model dinàmic.

El **model cinemàtic** dona la informació de la posició i orientació de tots els elements del robot, és a dir, es fa un estudi del moviment del robot sense considerar les forces que produeixen aquest moviment. Per tant, es tracta d'estudiar tant les propietats geomètriques com temporals del moviment [A. Ollero, 2001].

El **model dinàmic** té per objectiu trobar la relació entre els parells o forces i les variables articulars del robot. Les velocitats lineals i angulars venen donades per les forces i parells que s'apliquen a l'estructura mecànica encara que també depenen de les magnituds de les masses i la seva distribució [A. Ollero, 2001].

### 6.2. Model cinemàtic

Mitjançant el model cinemàtic invers s'obtenen les variables articulars que fan que la posició i l'orientació del punt de referència del terminal (TCP<sup>3</sup>)

#### 6.2.1. Model cinemàtic directe

En aquest apartat es pretén trobar una funció que permeti relacionar la posició i orientació del sistema de referència objectiu en l'espai cartesià,  $p$ , a partir de les variables articulars  $q$  del robot:

$$p = f(q)$$

---

<sup>1</sup> EDP: Equacions en derivades parcials.

<sup>2</sup> EDO: Equacions diferencials ordinàries.

<sup>3</sup> TCP (*Tool Center Point*): Punt real o virtual representatiu del terminal per a una aplicació donada, fixat el sistema de coordenades del punt.



on  $f$  és un conjunt de funcions no lineals.

Un robot articular es pot representar mitjançant quatre magnituds associades a cada articulació, una de les quals serà la variable d'articulació i les restants paràmetres fixes del robot.

Per construir el model directe és necessari definir així mateix un sistema de referència solidari a cada segment, i descriure les cadenes cinemàtiques, és a dir, la relació entre el sistema de referència  $\{j\}$  i l'anterior  $\{j-1\}$ . Aquestes relacions s'expressen mitjançant les matrius de transformació de cada articulació:

$$p = f(q) = T_1^0 * T_2^1 * \dots * T_n^{n-1} = T_n^0 \quad (\text{Eq. 6.1})$$

La matriu de transformació  $T_n^0$  relaciona la localització  $\{n\}$  amb la  $\{0\}$ , en funció de les variables d'articulació  $q_1 \dots q_n$ . Per trobar els sistemes de referència, els paràmetres i les matrius de transformació s'empra l'algorisme de Denavit-Hartenberg.

## 6.2.2. Algorisme de Denavit-Hartenberg

L'algorisme de Denavit-Hartenberg consta dels següents passos [Barrientos, 1997, pag.97-103]:

**D-H1.** Numerar els elements començant per 1 (primer element mòbil de la cadena) i acabar amb  $n$  (últim element mòbil). Es numerarà com element 0 a la base fixa del robot.

**D-H2.** Numerar cada articulació començant per 1 (la corresponent al primer grau de llibertat) i acabant en  $n$ .

**D-H3.** Localitzar l'eix de cada articulació. Si es de rotació, l'eix serà el seu propi eix de gir. Si és prismàtica, serà al llarg del qual es produeix el desplaçament.

**D-H4.** Per  $i$  a  $n-1$  situar l'eix  $z_n$  sobre l'eix de l'articulació  $i+1$ .

**D-H5.** Situar l'origen del sistema de la base  $\{S_0\}$  en qualsevol punt del eix  $z_0$ . Els eixos  $x_0$  i  $y_0$  es localitzaran de forma que formin un sistema dextroàgil<sup>4</sup> amb  $z_0$ .

**D-H6.** Per  $i$  de 1 a  $n-1$ , situar el sistema  $\{S_i\}$  (solidari a l'element  $i$ ) en la intersecció de l'eix  $z_i$  amb la línia normal comú a  $z_{i-1}$  i  $z_i$ . Si ambdós eixos es tallen es situaria  $\{S_i\}$  en el punt de tall. Si fossin paral·lels  $\{S_i\}$  es situaria en la articulació  $i+1$ .

**D-H7.** Situar  $x_i$  en la línia normal comuna a  $z_{i-1}$  i  $z_i$ .

**D-H8.** Situar  $y_i$  de manera que formi un sistema dextroàgil amb  $x_i$  i  $z_i$ .

**D-H9.** Situar el sistema  $\{S_j\}$  en l'extrem del robot de manera que  $z_n$  coincideixi amb la direcció de  $z_{n-1}$  i  $x_n$  sigui normal a  $z_{n-1}$  i  $z_n$ .

<sup>4</sup> Dextroàgil: que gira en el mateix sentit que les agulles del rellotge.





**D-H10.** Obtenir  $\theta_i(t_i)$  com l'angle que s'ha de girar al voltant de  $z_{i-1}$  per que  $x_{i-1}$  i  $x_i$  quedin paral·lels.

**D-H11.** Obtenir  $d_i$  com la distancia, mesurada al llarg de  $z_{i-1}$ , que s'ha de desplaçar  $\{S_i\}$  per tal que  $x_i$  i  $x_{i-1}$  quedin alineats.

**D-H12.** Obtenir  $a_i$  com la distancia mesurada al llarg de  $x_i$  (que ara coincidiria amb  $x_{i-1}$ ) que cal desplaçar el nou  $\{S_{i-1}\}$  per que el seu origen coincideixi amb  $\{S_i\}$ .

**D-H13.** Obtenir  $\alpha_i$  com l'angle que cal girar al voltant de  $x_i$  (que ara coincidiria amb  $x_{i-1}$ ), per que el nou sistema  $\{S_{i-1}\}$  coincidís totalment amb  $\{S_i\}$ .

**D-H14.** Obtenir les matrius de transformació  ${}^{i-1}A_i$  ( $T_i^{i-1}$ ) definides com:

$${}^{i-1}A_i = T_i^{i-1} = \begin{pmatrix} \cos(\theta_i) & -\cos(\alpha_i) \cdot \sin(\theta_i) & \sin(\alpha_i) \cdot \sin(\theta_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i) \cdot \cos(\theta_i) & -\sin(\alpha_i) \cdot \cos(\theta_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{Eq. 6.2})$$

**D-H15.** Obtenir la matriu de transformació que relaciona el sistema de la base amb l'extrem del robot (TCP):

$$T = {}^0A_1 * {}^1A_2 * \dots * {}^{n-1}A_n = T_1^0 * T_2^1 * \dots * T_n^{n-1} = T_n^0.$$

**D-H16.** La matriu  $T$  defineix l'orientació (submatriu de rotació) i posició (submatriu de translació) de l'extrem referit a la base en funció de les  $n$  coordenades articulars.

### 6.2.3. Matriu de paràmetres de Denavit-Hartenberg

En la *Figura 6.1* es pot veure una representació esquemàtica simplificada del robot amb els sistemes de referència i paràmetres tal com indica l'algorisme de Denavit-Hartenberg.

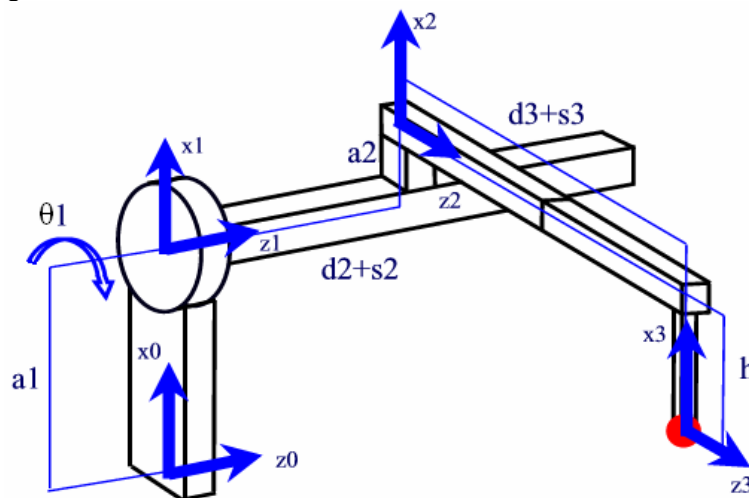


Figura 6.1: Esquema simplificat del robot de laparoscòpia



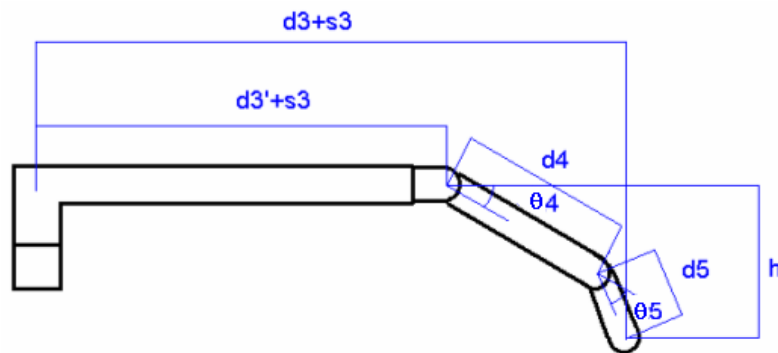
El robot té tres graus de llibertat corresponents a les tres articulacions. Es requeriran per tant tres variables articulars:

- $\theta_1$  : articulació de revolució
- $s_2$  : articulació de translació
- $s_3$  : articulació de translació

A més és necessari definir els següents paràmetres:

$$\begin{aligned}
 a_1 &= 42 \text{ [cm]} \\
 a_2 &= 9,5 \text{ [cm]} \\
 d_2 &= 30 \text{ [cm]} \\
 d_3 &= 34 + d_4 \cdot \cos(\theta_4) + d_5 \cdot \cos(\theta_4 + \theta_5) \text{ [cm]} \\
 h &= d_4 \cdot \sin(\theta_4) + d_5 \cdot \sin(\theta_4 + \theta_5) \text{ [cm]} \\
 d_4 &= 25 \text{ [cm]} \\
 d_5 &= 13,5 \text{ [cm]}
 \end{aligned}$$

on  $\theta_4$  i  $\theta_5$  son uns paràmetres que es poden ajustar tal com mostra la *Figura 6.2*.



*Figura 6.2: Esquema representació de la coordenada d3*

L'aplicació de l'algorisme té com a resultat la següent matriu de paràmetres:

$i$	$\alpha_i$	$a_i$	$\theta_i$	$d_i$
1	0	$a_1$	$\theta_1$	0
2	-90°	$a_2$	0	$d_2+s_2$
3	0	h	0	$d_3+s_3$

*Taula 6.1: Matriu de paràmetres del robot.*

### 6.2.4. Matrius de Transformació

Les matrius obtingudes després de l'algorisme de Denavit-Hartenberg són les següents:

*Matriu de rotació entre les referències {0} i {1}:*



$$T_1^0 = \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & a_1 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matriu de translació entre les referències {1} i {2}:

$$T_2^1 = \begin{pmatrix} 1 & 0 & 0 & a_2 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 + s_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matriu de translació entre les referències {2} i {3}:

$$T_3^2 = \begin{pmatrix} 1 & 0 & 0 & -h \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 + s_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Si es multipliquen les tres s'obté la matriu de transformació entre les referències {0} i {3}:

$$T_3^0 = T_1^0 * T_2^1 * T_3^2$$

$$T_3^0 = \begin{pmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & -\cos(\theta_1) * h - \sin(\theta_1) * (d_3 + s_3) + \cos(\theta_1) * a_2 + a_1 \\ \sin(\theta_1) & 0 & \cos(\theta_1) & -\sin(\theta_1) * h + \cos(\theta_1) * (d_3 + s_3) + \sin(\theta_1) * a_2 \\ 0 & -1 & 0 & d_2 + s_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La obtenció d'aquestes matrius es pot observar en l'arxiu adjunt 'model.m' de MATLAB (Annex B).

La matriu  $T_3^0$  mostra la informació de l'orientació de l'últim element {3} (submatriu 3x3) i la posició del TCP o element terminal en coordenades cartesianes respecte la referència {0}. Com només es necessita la posició del TCP, s'extreu la següent informació, que determina el model cinemàtic directe:

$$\begin{aligned} x &= -\cos(\theta_1) * h - \sin(\theta_1) * (d_3 + s_3) + \cos(\theta_1) * a_2 + a_1 \\ y &= -\sin(\theta_1) * h + \cos(\theta_1) * (d_3 + s_3) + \sin(\theta_1) * a_2 \\ z &= d_2 + s_2 \end{aligned}$$

(Eq. 6.3)



### 6.2.5. Model cinemàtic invers

El què interessa, però, no és el model directe, sinó que es necessita saber el valor de les variables d'articulació per tal que el robot es desplaci fins a la posició desitjada (que és el punt que resultarà de l'aplicació de l'algorisme de visió). Es tracta de resoldre, doncs, el problema invers a l'anterior.

A tal efecte, es defineix una matriu de transformació genèrica de la forma en què es mostra a continuació:

$$T_3^0 = \begin{pmatrix} nx & sx & ax & px \\ ny & sy & ay & py \\ nz & sz & az & pz \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Com s'ha vist anteriorment es té a més la relació donada per l'equació (Eq.6.1):

$$T_3^0 = T_1^0 * T_2^1 * T_3^2$$

Si es premultipliquen ambdós membres per  $(T_1^0)^{-1}$  s'obté:

$$(T_1^0)^{-1} * T_3^0 = T_2^1 * T_3^2$$

A partir de les matrius resultants, es pot igualar termes per tal de trobar una relació entre les variables d'estat (variables articulars) i les coordenades del TCP. En aquestes relacions no poden aparèixer els vectors n, s i a donat que es desconeix l'orientació de l'element terminal.

Procedint com s'ha comentat, s'arriba a les equacions següents:

$$h - a2 + (px - a1) * \cos(\theta1) + \sin(\theta1) * py = 0 \quad (\text{Eq. 6.4})$$

$$s2 = pz - d2 \quad (\text{Eq. 6.5})$$

$$s3 = -d3 - \sin(\theta1) * px + \cos(\theta1) * py + \sin(\theta1) * a1 \quad (\text{Eq. 6.6})$$

Per aïllar  $\theta1$  de l'equació (Eq.6.4) es procedeix:

Substituir el  $\cos(\theta1)$  per  $\sqrt{1 - \sin^2(\theta1)}$ :

$$h - a2 + (px - a1) * \sqrt{1 - \sin^2(\theta1)} + \sin(\theta1) * py = 0$$

Aïllar el terme amb  $\sqrt{1 - \sin^2(\theta1)}$ :



$$(px - a1) * \sqrt{1 - \sin^2(\theta1)} = (-h + a2) - \sin(\theta1) * py$$

Elevant al quadrat ambdós membres de la igualtat:

$$(px - a1)^2 * (1 - \sin^2(\theta1)) = (-h + a2)^2 - \sin^2(\theta1) * py^2$$

Resoldre l'equació de segon grau amb incògnita el  $\sin(\theta1)$ :

$$\sin(\theta1) = \frac{(-h + a2) * py \pm (px - a1) * \sqrt{py^2 + (px - a1)^2 - (a2 - h)^2}}{(px - a1)^2 + py^2}$$

S'aïlla  $\theta1$ :

$$\theta1 = \arcsin \frac{(-h + a2) * py \pm (px - a1) * \sqrt{py^2 + (px - a1)^2 - (a2 - h)^2}}{(px - a1)^2 + py^2}$$

Amb el valor de  $\theta1$  aïllat s'obté el model cinemàtic invers:

$$\theta1 = \arcsin \frac{(-h + a2) * py \pm (px - a1) * \sqrt{py^2 + (px - a1)^2 - (a2 - h)^2}}{(px - a1)^2 + py^2}$$

$$s2 = pz - d2$$

$$s3 = -d3 - \sin(\theta1) * px + \cos(\theta1) * py + \sin(\theta1) * a1 \text{ (amb } \theta1 \text{ ja conegut)}$$

(Eq. 6.7)

També es pot obtenir el model cinemàtic invers a través del següent esquema:

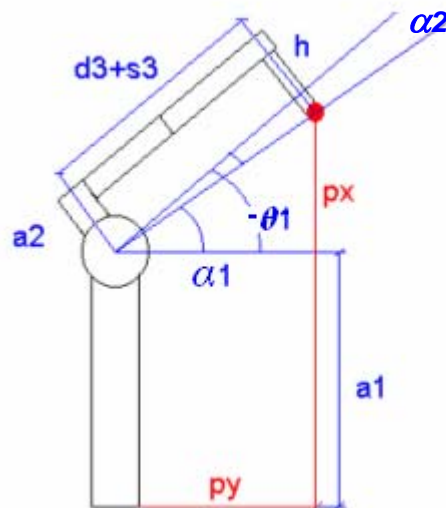


Figura 6.3: Representació dels angles  $\alpha1$  i  $\alpha2$ .

Si es parteix l'angle  $\theta1$  com la suma de  $\alpha1$  i  $\alpha2$  podem obtenir les equacions següents per  $\alpha1$ :

$$\sin(\alpha1) = \frac{(px - a1)}{\sqrt{(px - a1)^2 + py^2}} \quad (\text{Eq. 6.8})$$



$$\cos(\alpha_1) = \frac{py}{\sqrt{(px-a_1)^2 + py^2}} \quad (\text{Eq. 6.9})$$

I per  $\alpha_2$ :

$$\sin(\alpha_2) = \frac{(h-a_2)}{\sqrt{(px-a_1)^2 + py^2}} \quad (\text{Eq. 6.10})$$

$$\cos(\alpha_2) = \frac{d_3 + s_3}{\sqrt{(px-a_1)^2 + py^2}} \quad (\text{Eq. 6.11})$$

Si s'aïlla  $\alpha_1$  i  $\alpha_2$  de les equacions (Eq.6.8) i (Eq.6.10) respectivament s'obté:

$$\alpha_1 = \arcsin \frac{(px-a_1)}{\sqrt{(px-a_1)^2 + py^2}}$$

$$\alpha_2 = \arcsin \frac{(h-a_2)}{\sqrt{(px-a_1)^2 + py^2}}$$

Ara es pot trobar  $\theta_1$  con la suma dels angles  $\alpha_1$  i  $\alpha_2$ , i  $s_3$  es pot aïllar de l'equació (Eq.6.11):

$$\theta_1 = -\alpha_1 - \alpha_2$$

$$s_3 = \sqrt{(px-a_1)^2 + py^2} \cdot \cos(\alpha_2) - d_3$$

### 6.3. Model dinàmic

La dinàmica s'ocupa de la relació entre les forces que s'apliquen al robot i el moviment que aquestes forces originen en ell. Aquesta relació s'obté mitjançant el denominat model dinàmic del robot, el qual relaciona matemàticament:

- La localització del robot definida per les seves variables articulars o per les coordenades de localització del seu extrem i les seves derivades: velocitat i acceleració.
- Les forces (parells) aplicats a les articulacions.
- Els paràmetres del robot, com ara són les longituds, masses i inèrcies dels seus elements.

El model dinàmic, tot i que ens molts casos no es calcula degut a la seva complexitat, és imprescindible per aconseguir, per exemple:

- Simular el moviment del robot.
- Dissenyar i avaluar el control dinàmic del robot.

Sovint es realitzen un seguit d'hipòtesis per simplificar. Es aquest cas es considerarà la massa de cada element concentrada en un punt, com es pot veure en la Figura 6.4:



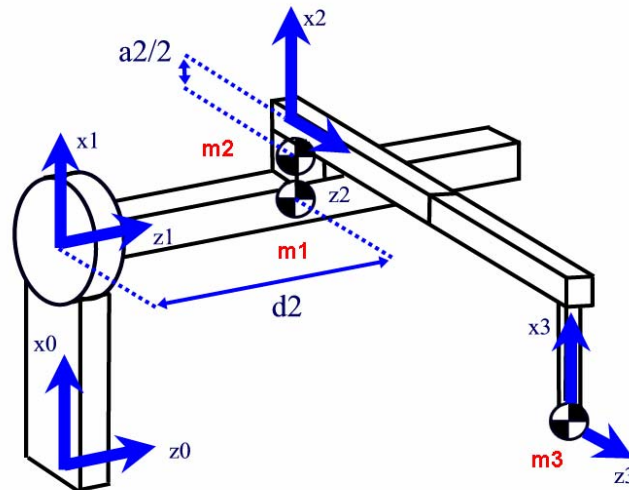


Figura 6.4. Situació de les masses concentrades

El model dinàmic del robot es pot escriure de forma general com:

$$\tau = M(q) \cdot \ddot{q} + V(q, \dot{q}) + G(q) + F(\dot{q}) \quad (\text{Eq. 6.12})$$

on:

$\tau$ : vector dels parells que s'apliquen a cada articulació (dimensió 3x1)

$M(q)$ : matriu de masses o inèrcies del manipulador (dimensió 3x3)

$G(q)$ : vector de termes gravitatoris (dimensió 3x1)

$V(q, \dot{q})$ : vector de les forces centrípetes i de Coriolis (dimensió 3x1)

$F(q, \dot{q})$ : vector de friccions (dimensió 3x1)

$$q = \begin{pmatrix} \theta_1 \\ s_2 \\ s_3 \end{pmatrix} \quad \dot{q} = \frac{\partial q}{\partial t} = \begin{pmatrix} d\theta_1 \\ ds_2 \\ ds_3 \end{pmatrix} \quad \ddot{q} = \frac{\partial \dot{q}}{\partial t} = \begin{pmatrix} d^2\theta_1 \\ d^2s_2 \\ d^2s_3 \end{pmatrix}$$

### 6.3.1. Algorisme computacional de Lagrange-Euler

Per trobar el model dinàmic del manipulador s'emprarà l'algorisme de Lagrange-Euler, realitzant els càlculs amb l'ajuda del fitxer de MATLAB *dinamic\_l\_e.m*, que es pot consultar en l'Annex B. Els passos ha dur a terme són els següents [A.Ollero, 2001]:

**L-E1: Assignar a cada articulació un sistema de referència d'acord a les normes de Denavit-Hartenberg**

L'assignació dels sistemes de referència es fa seguint la *Figura 6.1*.

**L-E2: Obtenir les matrius de transformació  ${}^0T_i$  per cada element  $i$**

Les matrius de transformació  ${}^0T_i$  són les calculades en l'apartat 6.2.4.



**L-E3: Obtenir les matrius  $U_{ij}$  definides per:**

$$U_{ij} = \frac{\partial^0 A_i}{\partial q_j} \quad (\text{Eq. 6.13})$$

Fent els càlculs corresponents s'obtenen les matrius:

$$U_{11} = \begin{pmatrix} -\sin(\theta_1) & -\cos(\theta_1) & 0 & 0 \\ \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; U_{12} = (0); \quad U_{13} = (0)$$

$$U_{21} = \begin{pmatrix} -\sin(\theta_1) & 0 & -\cos(\theta_1) & -\sin(\theta_1) * a_2 \\ \cos(\theta_1) & 0 & -\sin(\theta_1) & \cos(\theta_1) * a_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; U_{22} = (0); \quad U_{23} = (0)$$

$$U_{31} = \begin{pmatrix} -\sin(\theta_1) & 0 & -\cos(\theta_1) & \sin(\theta_1) * h - \cos(\theta_1) * (d_3 + s_3) - \sin(\theta_1) * a_2 \\ \cos(\theta_1) & 0 & -\sin(\theta_1) & -\cos(\theta_1) * h - \sin(\theta_1) * (d_3 + s_3) + \cos(\theta_1) * a_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$U_{32} = (0); \quad U_{21} = \begin{pmatrix} 0 & 0 & 0 & -\sin(\theta_1) \\ 0 & 0 & 0 & \cos(\theta_1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

**L-E4: Obtenir les matrius  $U_{ijk}$  definides per:**

$$U_{ijk} = \frac{\partial U_{ij}}{\partial q_k} \quad (\text{Eq. 6.14})$$

Les matrius queden de la següent forma:

$$U_{111} = \begin{pmatrix} -\cos(\theta_1) & \sin(\theta_1) & 0 & 0 \\ -\sin(\theta_1) & -\cos(\theta_1) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; U_{112} = (0); \quad U_{113} = (0)$$

$$U_{121} = (0); \quad U_{122} = (0); \quad U_{123} = (0)$$

$$U_{131} = (0); \quad U_{132} = (0); \quad U_{133} = (0)$$





$$U_{211} = \begin{pmatrix} -\cos(\theta_1) & 0 & \sin(\theta_1) & -\cos(\theta_1) * a_2 \\ -\sin(\theta_1) & 0 & -\cos(\theta_1) & -\sin(\theta_1) * a_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; U_{212} = (0); U_{213} = (0)$$

$$U_{221} = (0); U_{222} = (0); U_{223} = (0)$$

$$U_{311} = \begin{pmatrix} -\cos(\theta_1) & 0 & \sin(\theta_1) & \cos(\theta_1) * h + \sin(\theta_1) * (d_3 + s_3) - \cos(\theta_1) * a_2 \\ -\sin(\theta_1) & 0 & -\cos(\theta_1) & \sin(\theta_1) * h - \cos(\theta_1) * (d_3 + s_3) - \sin(\theta_1) * a_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$U_{312} = (0); U_{313} = \begin{pmatrix} 0 & 0 & 0 & -\cos(\theta_1) \\ 0 & 0 & 0 & -\sin(\theta_1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$U_{321} = (0); U_{322} = (0); U_{323} = (0)$$

$$U_{331} = \begin{pmatrix} 0 & 0 & 0 & -\cos(\theta_1) \\ 0 & 0 & 0 & -\sin(\theta_1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; U_{332} = (0); U_{333} = (0)$$

**L-E5: Obtenir les matrius de pseudo-inèrcia  $J_i$  per a cada element, que venen definides per:**

$$J_i = \begin{pmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int y_i x_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int z_i x_i dm & \int z_i y_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{pmatrix} \quad (\text{Eq. 6.15})$$

Operant es troba:

$$J_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & m_1 * d_2^2 & m_1 * d_2 \\ 0 & 0 & m_1 * d_2 & m_1 \end{pmatrix} \quad J_2 = \begin{pmatrix} \frac{1}{4} * a_2^2 * m_2 & 0 & 0 & -\frac{1}{2} * m_2 * a_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2} * m_2 * a_2 & 0 & 0 & m_2 \end{pmatrix}$$

$$J_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_3 \end{pmatrix}$$



**L-E6: Obtenir la matriu d'inèrcies  $M=[m_{ij}]$ , els elements de la qual venen definits per:**

$$m_{ij} = \sum_{k=\max(i,j)}^n \text{Traza}(U_{kj} J_k U_{ki}^T) \text{ amb } i,j=1,2,\dots,n ; n: \text{ graus de llibertat} \quad (\text{Eq. 6.16})$$

$$\begin{aligned} m_{11} &= \frac{1}{4} * a^2 * m_2 + m_3 * a^2 + m_3 * s^2 + m_3 * d^2 + m_3 * h^2 - m_3 * a * h - m_3 * h * a + 2 * m_3 * d * s \\ m_{12} &= 0 \\ m_{13} &= a * m_3 - m_3 * h \\ m_{21} &= 0 \quad m_{22} = m_3 + m_2 \\ m_{23} &= 0 \\ m_{31} &= -h * m_3 + a * m_3 \\ m_{32} &= 0 \\ m_{33} &= m_3 \end{aligned}$$

Per tant la matriu de masses resultant és:

$$M = \begin{pmatrix} \frac{1}{4} * a^2 * m_2 + m_3 * a^2 + m_3 * s^2 + m_3 * d^2 + m_3 * h^2 - m_3 * a * h - m_3 * h * a + 2 * m_3 * d * s & 0 & a * m_3 - m_3 * h \\ 0 & m_3 + m_2 & 0 \\ -h * m_3 + a * m_3 & 0 & m_3 \end{pmatrix} \quad (\text{Eq. 6.17})$$

**L-E7: Obtenir els termes  $h_{ikm}$  definits per:**

$$h_{ikm} = \sum_{j=\max(i,k,m)}^n \text{Traza}(U_{jkm} J_j U_{ji}^T) \text{ amb } i,j=1,2,\dots,n ; n: \text{ graus de llibertat} \quad (\text{Eq. 6.18})$$

$h_{111} = 0$	$h_{112} = 0$	$h_{113} = m_3 * d + m_3 * s$
$h_{121} = 0$	$h_{122} = 0$	$h_{123} = 0$
$h_{131} = m_3 * d + m_3 * s$	$h_{132} = 0$	$h_{133} = 0$
$h_{211} = 0$	$h_{212} = 0$	$h_{213} = 0$
$h_{221} = 0$	$h_{222} = 0$	$h_{223} = 0$
$h_{231} = 0$	$h_{232} = 0$	$h_{233} = 0$
$h_{311} = -m_3 * d - m_3 * s$	$h_{312} = 0$	$h_{313} = 0$
$h_{321} = 0$	$h_{322} = 0$	$h_{323} = 0$
$h_{331} = 0$	$h_{332} = 0$	$h_{333} = 0$

**L-E8: Obtenir el vector de forces centrífugues i de Coriolis  $V=[v_i]^T$ , els elements del qual venen definits per:**

$$v_i = \sum_{k=1}^n \sum_{m=1}^n h_{ikm} \dot{q}_k \dot{q}_m \quad (\text{Eq. 6.19})$$

El vector resultant és:



$$V = \begin{pmatrix} 2 * m_3 * s_3 * d\theta_1 * ds_3 + 2 * m_3 * d_3 * d\theta_1 * ds_3 \\ 0 \\ - m_3 * d\theta_1^2 * d_3 - m_3 * d\theta_1^2 * s_3 \end{pmatrix} \quad (\text{Eq. 6.20})$$

**L-E9: Obtenir el vector de forces de gravetat  $G=[g_i]^T$  els elements del qual estan definits per:**

$$g_i = \sum_{j=1}^n (-m_j g U_{ji}^i r_j) \quad (\text{Eq. 6.21})$$

amb:

$\{0\}$  g: vector de la gravetat expressat en el sistema de la base del robot

$$g = (-g, 0, 0, 0)$$

Es troba el vector:

(Eq. 6.22)

$$G = \begin{pmatrix} -\frac{1}{2} * a_2 * m_2 * \sin(\theta_1) * g + m_3 * g * (\sin(\theta_1) * h - \cos(\theta_1) * (d_3 + s_3) - \sin(\theta_1) * a_2) \\ 0 \\ - m_3 * \sin(\theta_1) * g \end{pmatrix}$$

**L-E10: L'equació de la dinàmica del sistema serà doncs:**

$$\tau = M(q) \cdot \ddot{q} + V(q, \dot{q}) + G(q) \quad (\text{Eq. 6.23})$$

amb les matrius M, V i G definides respectivament per les equacions (Eq.6.17), (Eq.6.20) i (Eq.6.22). El terme de friccions que apareixia a l'equació (Eq.6.12) en un principi no es tindrà en compte.

## 6.4. Validació del simulador del robot

A continuació es realitzen un seguit de simulacions i experiments per tal de validar el model complet del robot formulat anteriorment. Aquest, representa la dinàmica "real" del robot ja que descriu el comportament d'aquest d'una manera detallada, completa, complexa i precisa.

La descripció del comportament del robot real es simula mitjançant un simulador desenvolupat en SIMULINK i utilitzant diverses *Toolboxes* de MATLAB.

Per a realitzar aquesta verificació del model simplificat front al model complet (simulador), s'estudia, s'analitza i es valida el significat físic de la dinàmica del robot (simulador) en diferents escenaris de treball. Es considera que totes les unitats de les magnituds ( $\theta_1, s_2, s_3$ ) de les següents gràfiques estan en el SI.



### 6.4.1. Cas 1

En aquesta primera simulació es parteix de la posició inicial i no es transmet cap parell o força a les articulacions. Es vol comprovar quina serà l'evolució en el temps de les variables articulars, això és, com es mourà el robot si no està excitat. La representació i esquema en SIMULINK seria la que mostra la figura següent:

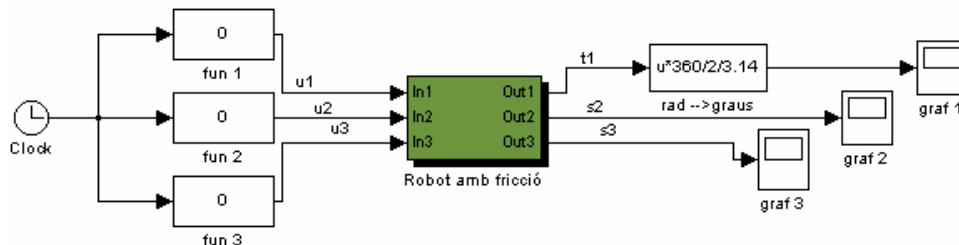


Figura 6.5: Representació del model complet del robot en SIMULINK pel cas 1.

Les gràfiques mostren el valor de les coordenades articulars en funció del temps. Les unitats són en el sistema internacional.

Per a la coordenada  $\theta_1$  el resultat es mostra a continuació:

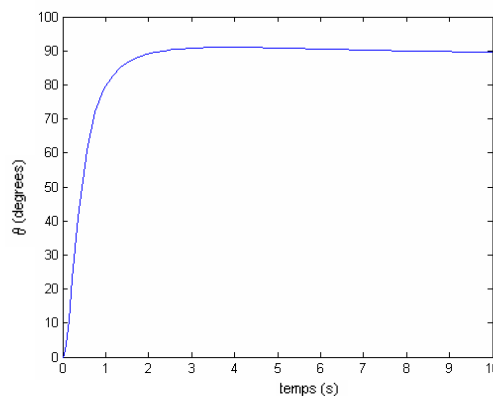


Figura 6.6: Coordenada  $\theta_1$  en funció del temps

Per a la coordenada  $s_2$ :

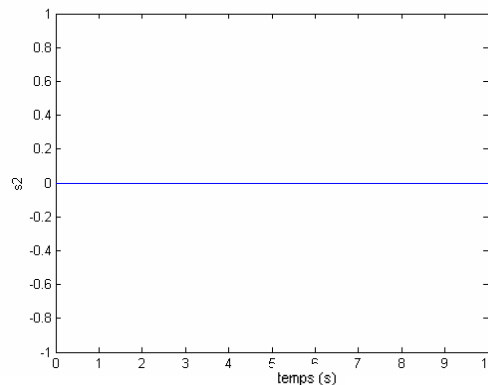


Figura 6.7: Coordenada  $s_2$  en funció del temps



Per la coordenada  $s_3$ :

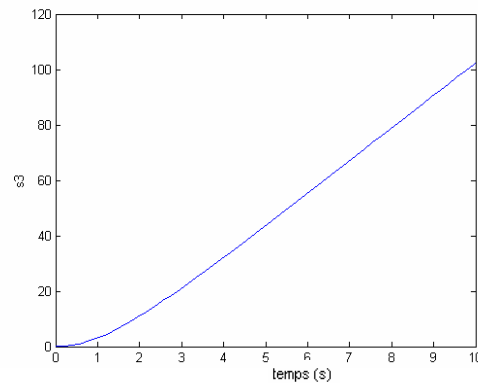


Figura 6.8: Coordenada  $s_3$  en funció del temps

A partir de l'observació de les tres gràfiques es pot deduir que el robot, al no tenir aplicat cap parell, cau des de la seva configuració inicial de forma que la coordenada  $s_2$  no varia, el braç cau fins als  $90^\circ$  i, en conseqüència, s'allarga fins a l'infinit amb velocitat constant, doncs hi ha frec sec.

#### 6.4.2. Cas 2

Si s'estudia la posició inicial, es poden trobar els valors de les tensions a aplicar per tal que el robot es mantingui en repòs. En aquesta situació de repòs, només cal aplicar parell en l'articulació 1 per evitar que, com s'ha vist en el Cas 1, el robot caigui.

Teòricament, aquesta tensió es pot trobar a partir de l'equació de la dinàmica del sistema (Eq. 6.23):

$$\tau = M(q) \cdot \ddot{q} + V(q, \dot{q}) + G(q)$$

Si es vol que el robot estigui en repòs, es complirà que:

$$q = \dot{q} = \ddot{q} = 0$$

Per tant, el parell que es necessitarà serà el suficient per compensar les forces gravitatòries:

$$\tau = G(q) = \begin{pmatrix} -4.0151 \\ 0 \\ 0 \end{pmatrix}$$

És a dir, que cal aplicar un parell en l'articulació 1 de 4.0151 [Nm] en l'eix  $z_1$  negatiu. Aquest és el parell necessari per mantenir el braç horitzontal. En la segona i la tercera articulació no cal aplicar cap tipus de força, ja que es troben en una posició singular.

Aquest cas es pot simular amb SIMULINK a partir del següent esquema:



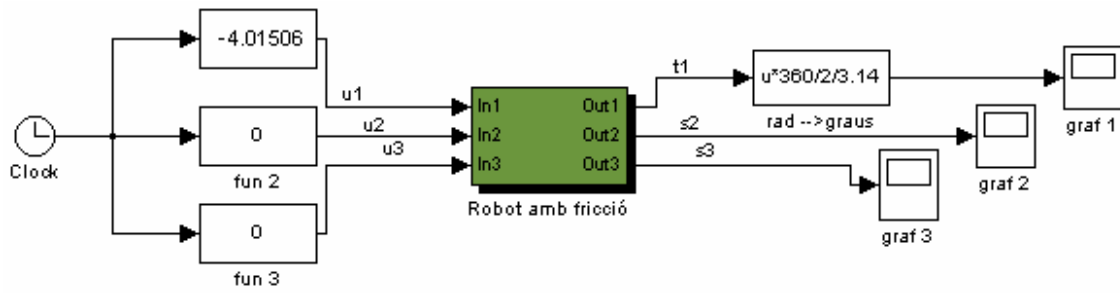


Figura 6.9: Representació en SIMULINK del cas 2

S'obtenen els següents resultats per a la coordenada  $\theta_1$ :

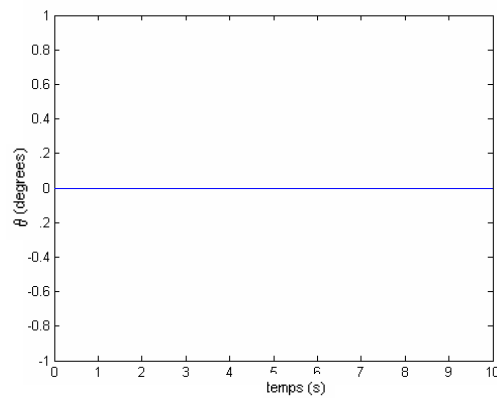


Figura 6.10: Coordenada  $\theta_1$  en funció del temps

Per a la coordenada  $s_2$ :

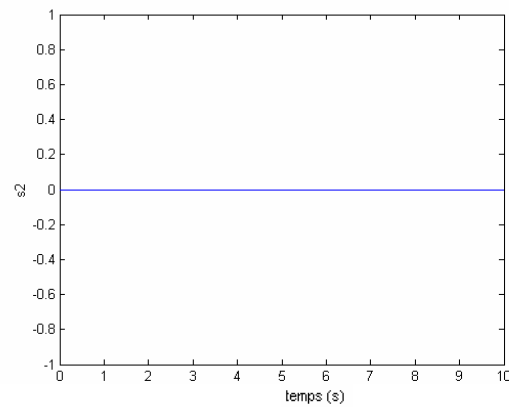


Figura 6.11: Coordenada  $s_2$  en funció del temps



Per a  $s_3$ :

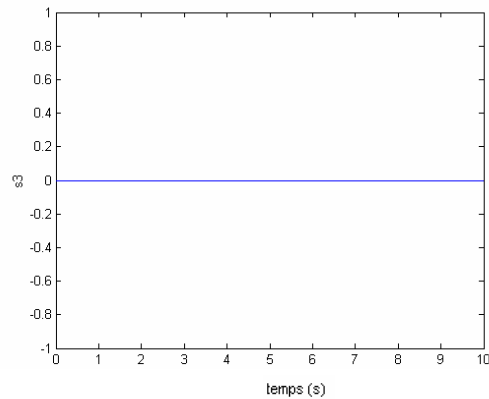


Figura 6.12: Coordenada  $s_3$  en funció del temps

S'observa que si s'aplica el parell que s'ha trobat teòricament el braç del robot es manté horitzontal, en repòs.

### 6.4.3. Cas 3

En aquesta simulació s'aplica a l'articulació 2 un parell de la forma  $F_2=0.1*\cos(t)$ , tot mantenint el parell de l'articulació 1 i sense cap altre pertorbació, tal com mostra la Figura 6.13:

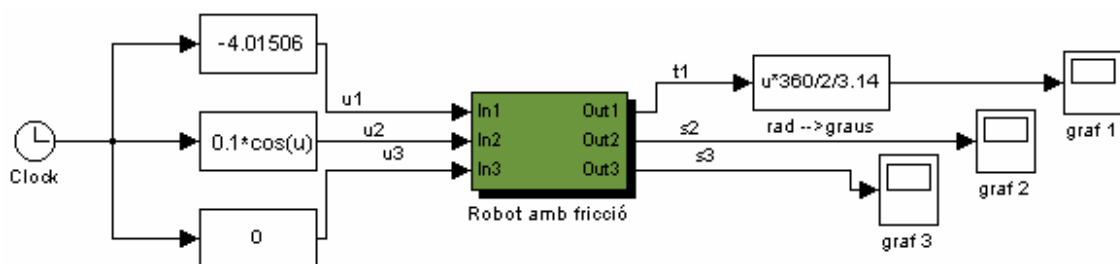


Figura 6.13: Representació en SIMULINK del cas 3

- El resultat per a la coordenada  $\theta_1$ :

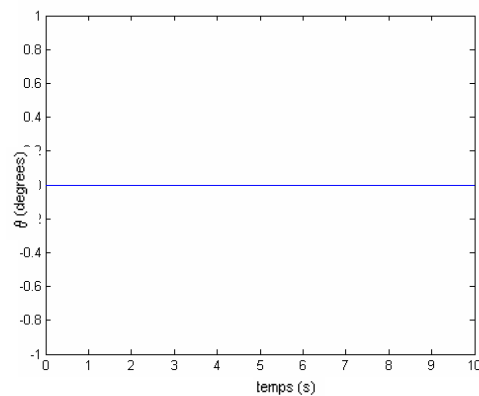


Figura 6.14: Coordenada  $\theta_1$  en funció del temps



Per a  $s_2$ :

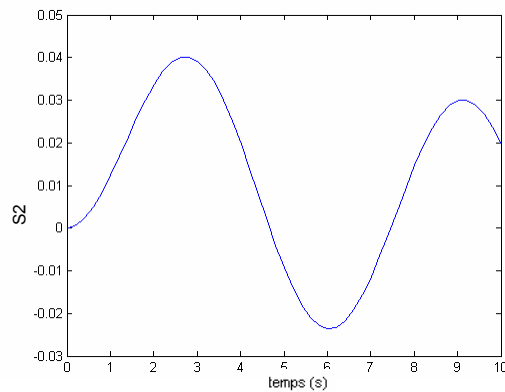


Figura 6.15: Coordenada  $s_2$  en funció del temps

Per a  $s_3$ :

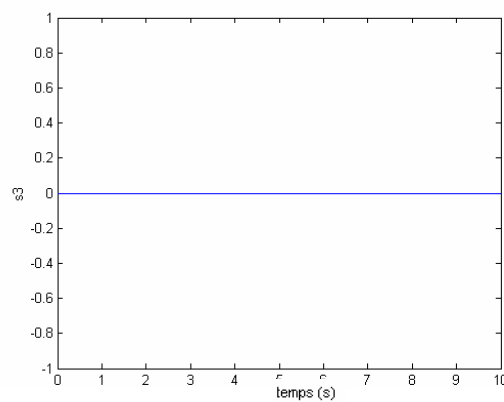


Figura 6.16: Coordenada  $s_3$  en funció del temps

Tal i com era d'esperar, el braç es desplaça fent una sinusoide en el pla horitzontal. A més, es pot observar en la gràfica de la coordenada  $s_2$  el fet que el braç es desplaça lleugerament més en un sentit que en l'altre. Aquest comportament és degut a l'existència de fricció. Si s'elimina la fricció s'observa el següent resultat per a la coordenada  $s_2$ :

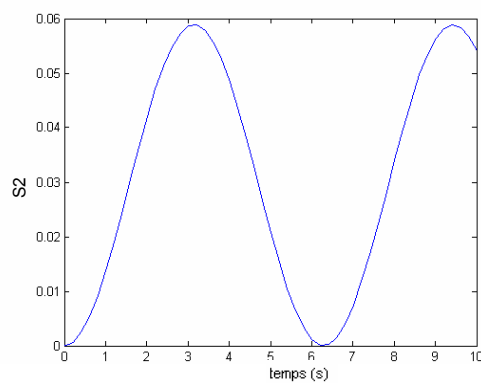


Figura 6.17: Coordenada  $s_2$  en funció del temps

Ara el braç retorna sempre al mateix punt, donat que la força total ha estat la mateixa en els dos sentits, no com quan hi ha fricció, que aconsegueix desplaçar-se més en un sentit que en l'altre.





### 6.4.4. Cas 4

Si en les mateixes condicions que en el Cas 3 s'aplica a més una força en la tercera articulació, el resultat previsible és que es perdés l'estabilitat de la posició ja que el parell en l'articulació 1 no seria suficient per compensar l'allargament del braç. Així, si la força aplicada a l'articulació 3 és positiva (+0.1 [N]), el braç s'allarga i cau cap endavant. L'esquema per realitzar la simulació d'aquest cas és el següent:

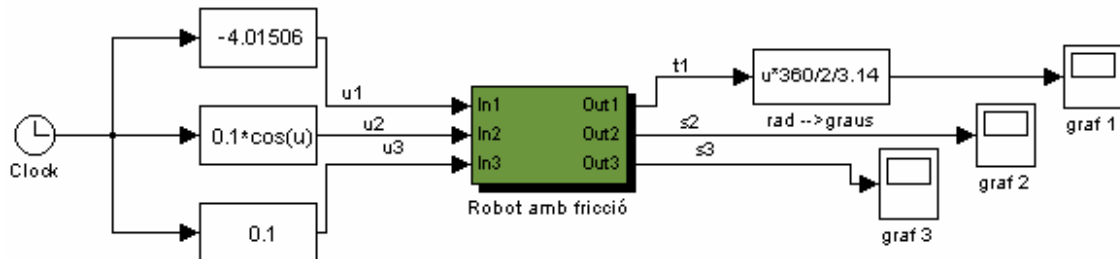


Figura .6.18: Representació en SIMULINK del cas 4

Els resultats per a les coordenades és el mostrat a les figures:

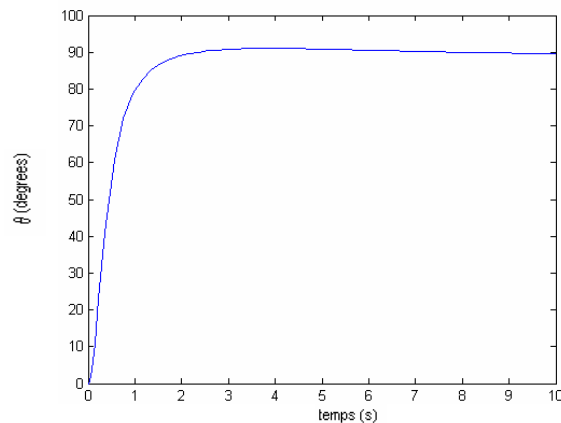


Figura6.19: Coordenada  $\theta_1$  en funció del temps

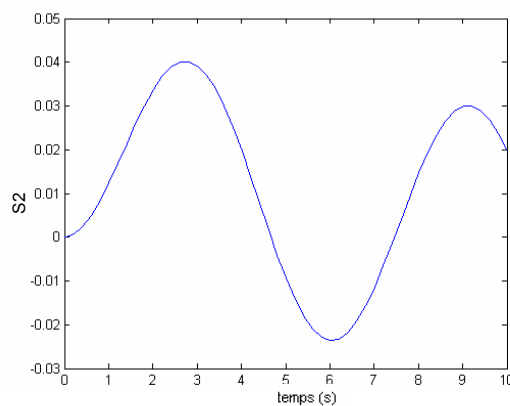


Figura 6.20: Coordenada  $s_2$  en funció del temps



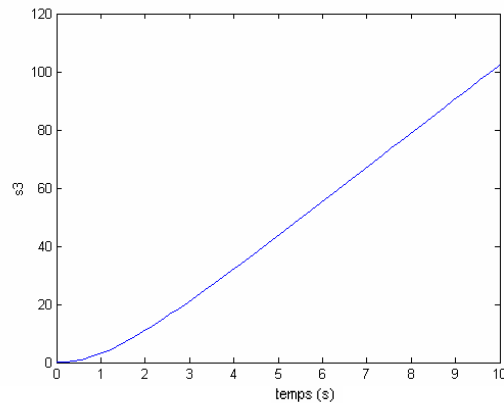


Figura 6.21: Coordenada  $s_3$  en funció del temps

EL resultat obtingut és l'esperat; el robot es cau fins als  $90^\circ$  i el braç s'allarga fins l'infinit, mentre l'articulació 2 intenta realitzar una sinusoide.

Si la força aplicada a l'articulació 3 és negativa ( $-0.1$  [N]), el braç es recull i per tant el parell sobre l'articulació 1 és més gran que el necessari per mantenir-lo en repòs. Conseqüentment, el braç pujarà fins a situar-se a  $90^\circ$ , i aquest fet provocarà que el braç s'allargui fins l'infinit, però ara en sentit contrari. Això es pot observar en les figures següents:

Coordenada  $\theta_1$ :

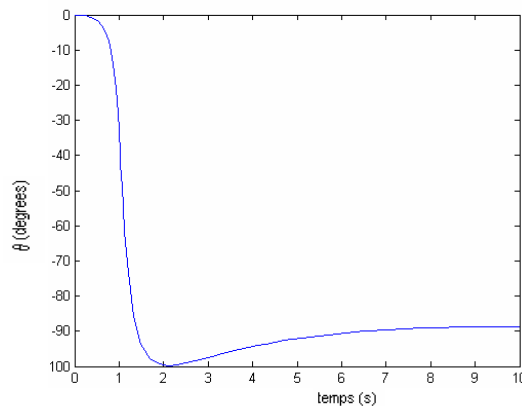


Figura 6.22: Coordenada  $\theta_1$  en funció del temps

Coordenada  $s_2$ :

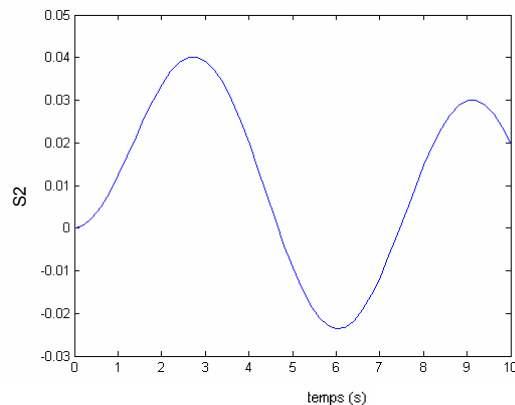


Figura 6.23: Coordenada  $s_2$  en funció del temps



Coordenada  $s_3$ :

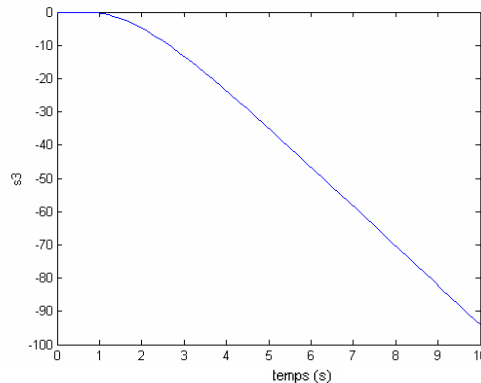


Figura 6.24: Coordenada  $s_3$  en funció del temps

A partir de les simulacions realitzades es pot afirmar que el model complert del robot és correcte.

## 6.5. Linealització

Com es pot veure a partir de les equacions de la dinàmica del moviment (Eq. 6.23), el model que descriu la dinàmica del robot és un model molt complex (existeixen un seguit de no-linealitats i de variacions dels paràmetres amb el temps). Per tant, fer l'anàlisi d'aquest per tal de després procedir al disseny del control de manera analítica amb un model d'aquestes característiques resulta d'una gran complexitat, i en molts casos, és impossible i inviable. Llavors, amb el propòsit de simplificar el procés de disseny del controlador es procedirà a simplificar el model eliminant-ne les seves no-linealitats. Però, un model lineal només pot descriure de manera local el comportament no lineal d'un sistema real. És a dir, la dinàmica d'un sistema no lineal només es pot aproximar manera correcta amb un ventall de model lineals vàlids a cadascun dels punts d'operació del sistema. Per representar de manera adequada el comportament global del sistema aquests models no lineals s'han d'interpol·lar [Shamma J., 1995].

En el nostre cas, a partir de les equacions 6.9 s'observa que el model dinàmic complet és un sistema de tres equacions diferencials de segon ordre no lineals i acoblades (les dues articulacions  $\theta_1$  i  $s_2$ ). Abans de linealitzar el model, es fa una altra simplificació, es desprècia la força de la gravetat (es farà  $g=0$ ). És correcte fer aquesta simplificació donat a que en absència de parell aplicat a les articulacions aquest no es cau, roman en equilibri gràcies a la configuració mecànica de les articulacions.

Després de realitzar aquesta simplificació i linealització del model "real" s'ha de comprovar que aquest model més senzill sigui correcte, això és, que approximi adequadament, amb "bondat" la dinàmica del sistema real. Per tant, es realitzarà una comparativa del comportament del model linealitzat (de control) amb el model complet que descriu de manera acurada la dinàmica del sistema real, i així d'aquesta forma s'assegura la validesa de les hipòtesis simplifícatives.



### 6.5.1. Linealització mitjançant Taylor

Per linealitzar el model al voltant d'un punt d'operació s'emprarà l'expansió de Taylor, tal i com es mostra en [Shamma J, 1995, pàg.388-399].

Donada una funció

$$\dot{x}(t) = f(x(t), u(t)) = \begin{pmatrix} f_1(x_1 \dots x_n, u_1 \dots u_m) \\ \vdots \\ f_p(x_1 \dots x_n, u_1 \dots u_m) \end{pmatrix}$$

i un punt d'operació definit per  $\dot{x} = f(x_0, u_0)$ , l'expansió de Taylor despreciant els termes de grau superior a 1 és:

$$\dot{x} \approx f(x_0, u_0) + D_1 f(x_0, u_0) * (x(t) - x_0) + D_2 f(x_0, u_0) * (u(t) - u_0) \quad (\text{Eq. 6.24})$$

$$\text{on } D_1 f = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_p}{\partial x_1} & \dots & \frac{\partial f_p}{\partial x_n} \end{pmatrix} \text{ i } D_2 f = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \dots & \frac{\partial f_1}{\partial u_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_p}{\partial u_1} & \dots & \frac{\partial f_p}{\partial u_n} \end{pmatrix} \text{ són els jacobians respecte } x$$

i  $u$  respectivament.

Si es realitza el canvi de variable:

$$\tilde{x}(t) = x(t) - x_0$$

$$\tilde{u}(t) = u(t) - u_0$$

resulta:

$$\dot{\tilde{x}} \approx f(x_0, u_0) + D_1 f(x_0, u_0) * \tilde{x}(t) + D_2 f(x_0, u_0) * \tilde{u}(t) \quad (\text{Eq. 6.25})$$

Aquí, s'observa que les matrius  $D_1 f$  i  $D_2 f$  es corresponen a les matrius  $A$  i  $B$  del sistema en representació d'espai d'estats:

$$\begin{aligned} \dot{\tilde{x}} &\approx f(x_0, u_0) + A * \tilde{x}(t) + B * \tilde{u}(t) \\ y &= C * \tilde{x}(t) + D * \tilde{u}(t) \end{aligned}$$

Per tant, es pot linealitzar a partir de l'equació 6.23 la dinàmica del robot i trobar la seva representació en espai d'estats.

Retornant al cas del robot, es sap que l'equació de la dinàmica és (Eq. 6.23):

$$M(q) * \ddot{q} + C(q, \dot{q}) + G(q) = u$$

Si es posa en la forma general es tindrà:

$$\dot{x} = f(x, u) = E^{-1}(q) * F(q, \dot{q}) + E^{-1}(q) * B * u \quad (\text{Eq. 6.26})$$



on:

$$E(q) = \begin{pmatrix} I & 0 \\ 0 & M(q) \end{pmatrix}$$

$$F(q, \dot{q}) = \begin{pmatrix} \dot{q} \\ -C(q, \dot{q}) - G(q) \end{pmatrix}$$

$$B = \begin{pmatrix} 0 \\ I \end{pmatrix}$$

A partir de l'equació (Eq.6.26) es pot trobar el model linealitzat en qualsevol punt d'operació del sistema seguint la metodologia d'expansió de Taylor a dalt explicada. Tot seguit es procedirà a linealitzar-lo en el punt de treball nominal definit per  $f(x_0, u_0) = (0,0)$ , és a dir, es linealitzat al voltant de la posició inicial i sense aplicar parell a les articulacions.

### 6.5.2. Model linealitzat en el punt nominal

El fitxer de MATLAB *lineal\_taylor.m* (consultar a l'Annex B) és un programa que realitza la linealització en qualsevol punt i troba les matrius del sistema linealitzat a partir de l'equació de la dinàmica del sistema. Si s'executa s'obté:

*Matrius de la dinàmica:*

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (\text{Eq. 6.27})$$

$$B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 6.9326 & 0 & 2.1144 \\ 0 & 0.2943 & 0 \\ 2.1144 & 0 & 1.4748 \end{pmatrix} \quad (\text{Eq. 6.28})$$

*Matrius de la sortida:*

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (\text{Eq. 6.29})$$



$$D = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (\text{Eq. 6.30})$$

Aquest sistema representat en espai d'estats també es pot descriure mitjançant una matriu de funcions de transferència:

$$G(s) = \begin{pmatrix} \frac{6.933}{s^2} & 0 & \frac{2.114}{s^2} \\ 0 & \frac{0.2943}{s^2} & 0 \\ \frac{2.114}{s^2} & 0 & \frac{1.475}{s^2} \end{pmatrix}$$

Aquests resultats poden ser confirmats mitjançant la funció de MATLAB *'linmod'*:

```
>>[A,B,C,D]=linmod('robot_no_lineal',x0,u0)
```

on 'robot\_no\_lineal' és el model del robot creat amb SIMULINK i  $x_0$ ,  $u_0$  és el punt en el qual es vol linealitzar, és a dir, ( 0 0 0) i (0 0 0).

### 6.5.3. Validació del model simplificat

A continuació es realitzen un seguit de simulacions i experiments per tal de validar el model simplificat o model de control amb el model complet (model "real", que es considera que és el simulador desenvolupat en SIMULINK, i explicat anteriorment).

Amb la validació del model de control el que es pretén és comprovar si el model simplificat s'ajusta el màxim possible a la realitat, és a dir al comportament real del robot, per tal que els resultats de l'anàlisi del sistema i del disseny del controlador puguin ser adequats i considerats correctes.

A continuació es pretén comprovar que el comportament del sistema descrit per les matrius A i B de la dinàmica del sistema, que es poden observar en les equacions (Eq.6.27) i (Eq.6.28), aproxima correctament al comportament del sistema no-lineal. A tal efecte, es simularan la resposta d'ambdós models en llaç tancat davant una entrada graó de magnitud 0.1 (sense aplicar-hi cap control) i es compararan les respostes.

Per tal de simular el comportament del sistema linealitzat es crearà un bloc de SIMULINK, una *S-Function*, anomenada *'linealització\_taylor'* que modelitza el sistema no-lineal:



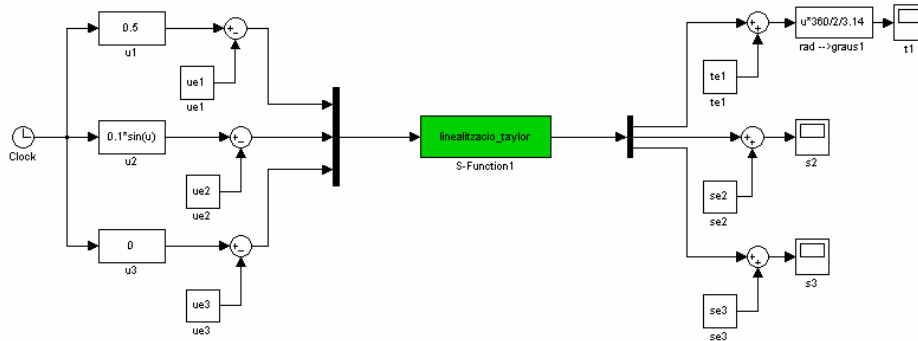


Figura 6.25: Bloc SIMULINK 'lineal\_taylor'

on  $(u_1, u_2, u_3)$  i  $(\theta_{e1}, s_{e2}, s_{e3})$  és el punt en el qual es linealitzava

Per a la primera articulació les dues respostes són:

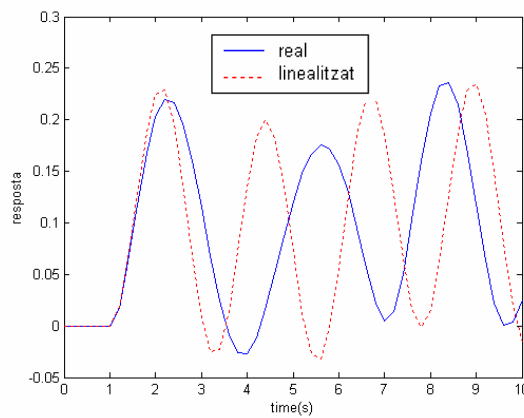


Figura 6.26: Comparació de la coordenada  $\theta_1$ .

Es pot comprovar que a l'inici de la simulació la resposta és la mateixa, però que a mesura que el braç va caient (és a dir que  $s_3$  i  $s_1$  creixen) la resposta deixa de ser igual, donat que el sistema s'està allunyant del punt en el qual s'ha linealitzat i per tant el model deixa de ser vàlid.

Per a l'articulació 3 les dues respostes són:

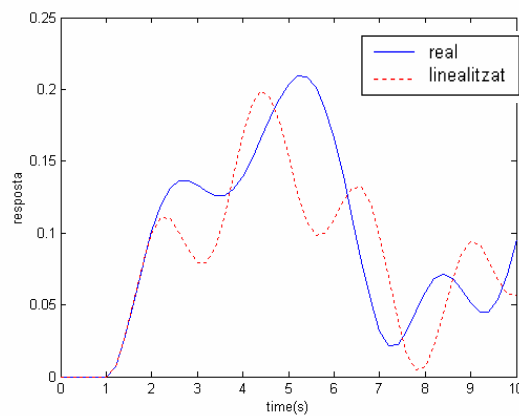


Figura 6.27: Comparació de la coordenada  $s_3$



S'observa el mateix que abans; al principi les dues respostes són semblants, però a mesura que s'allunyen de la posició en la qual s'ha linealitzat el sistema les respostes deixen de ser iguals.

Per a la segona articulació, la resposta per a ambdós sistemes és exactament igual:

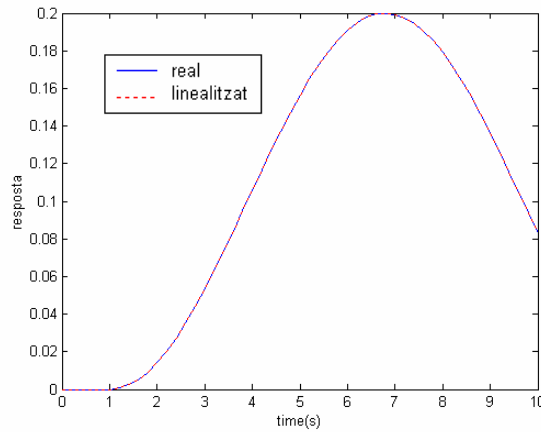


Figura 6.28: Comparació de la coordenada  $s_2$

Degut al comportament oscil·latori que presenta la planta es veu la necessitat de tancar el llaç (retroalimentació o *feedback*) per aconseguir un sistema més estable. Si es realitza la retroalimentació del robot utilitzant un controlador senzill, com per exemple un PID es pot observar l'estabilització de la planta:

Per a la primera articulació:

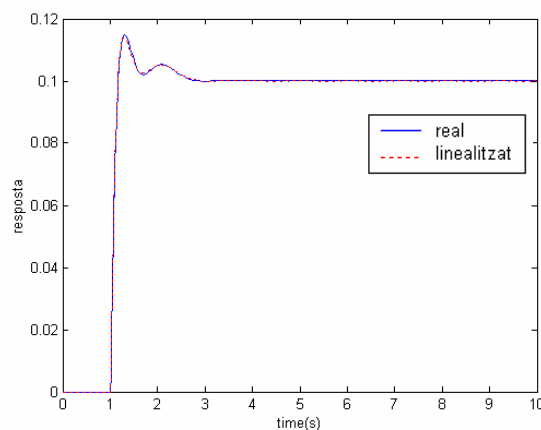


Figura 6.29: Comparació de la coordenada  $\theta_1$





Per a la segona:

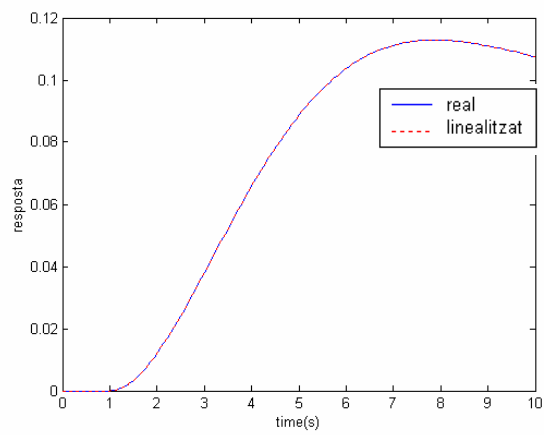


Figura 6.30: Comparació de la coordenada  $s_2$

Finalment, per a la tercera:

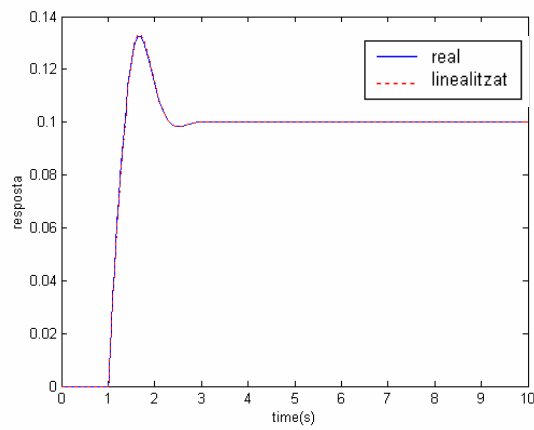


Figura 6.31: Comparació de la coordenada  $s_3$



## 7. El controlador robust

En aquest capítol es tractarà i revisarà el control Proporcional/Integral/Derivatiu (PID) que és el tipus de control dominant a la pràctica ja que és molt senzill d'implementar. També, s'explicarà la necessitat de l'ús de controladors robustos per a la solució d'alguns problemes de control, que per les seves característiques necessiten una metodologia de control que contempli les incerteses o errors del model deguts a diferents factors: no linealitats, no contemplació de les altes freqüències, del soroll, de les toleràncies, de l'envelliment, de pertorbacions exteriors (velocitat del vent, pluja, etc.). Exemples típics d'aquests sistemes poden ser els sistemes mecànics (principalment degut a les seves no linealitats) i processos químics (sobretot degut als grans retards a més de a la no linealitat).

Es donaran per sabuts coneixements bàsics de control, pel que es comentarà breument l'evolució que han sofert els controladors PID i els aspectes fonamentals d'aquests, i s'explicarà més exhaustivament un mètode de control robust no estructurat que és el control robust  $H_{\infty}$ .

### 7.1. Control convencional. Els controladors PID's

Encara que en els darrers anys els investigadors de l'àrea de control han desenvolupat molts algorismes de control avançat que en alguns casos han donat molt bons resultats en simulació, la indústria és un sector que es resisteix a emprar metodologies de control modern complexes ja que ens proporcionen controladors de ordre molt gran difícils d'implementar i de resintonitzar de manera *on-line*, i, en moltes ocasions, fràgils. Actualment el 98% dels sistemes automatitzats (mecànics, hidràulics, químics, etc.) empren controls PID's degut a la seva simplicitat, al clar efecte de cadascun dels seus tres paràmetres que s'han d'ajustar (P, I, i D), i a la flexibilitat suficient de la seva estructura per tal d'assegurar un bon comportament en la majoria d'aplicacions. Existeixen molts mètodes clàssics per a la sintonització d'aquest tipus de controladors: Ziegler-Nichols, Coon-Cohen, etc. [Aström K.J. i Hägglund T., 1995].

Dintre de la família dels controladors PID's existeixen tres accions: l'acció Proporcional (P) l'acció Integral (I) i l'acció Derivativa (D). Com a conseqüència d'aquestes, el controlador PID presenta les següents característiques:

- El mode proporcional genera una resposta ràpida però no elimina l'error en estat estacionari i genera una sortida del controlador proporcional a l'error, i per uns valors d'aquest guany molt grans pot arribar a inestabilitzar el sistema.
- El mode integral genera una correcció proporcional a l'error acumulat, és a dir, a la integral de l'error, cosa que permet poder eliminar l'error en règim permanent. Però proporciona una compensació retroactiva relativament lenta, i encara que ofereix precisió pot provocar inestabilitat.
- El mode derivatiu actua proporcionalment a la velocitat de canvi de la sortida o de l'error estacionari però no té cap efecte sobre aquest, i encara produeix una acció de control al canvi del rang de l'error, que en general tendeix a produir un efecte estabilitzant, com amplifica el soroll a altes freqüències pot provocar pics no desitjats en el transitori.



L'estructura o configuració general més habitual d'un controlador PID és la següent:

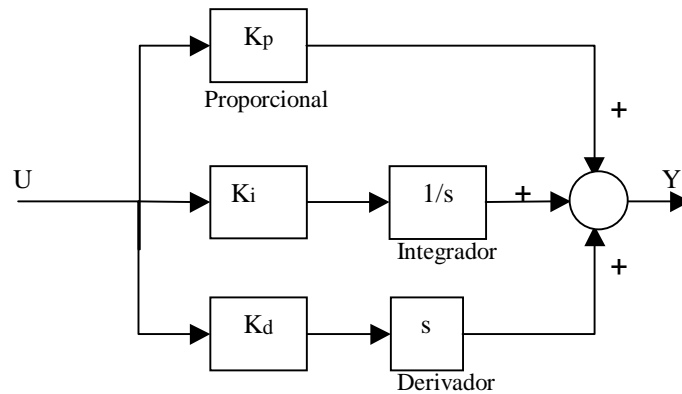


Figura 7.1: Estructura d'un controlador PID

La funció de transferència del controlador és per tant:

$$C(s) = \frac{U(s)}{Y(s)} = K_p + \frac{K_i}{s} + K_d s = \frac{K_i + K_p s + K_d s^2}{s} \quad (\text{Eq. 7.1})$$

Per tal de seleccionar els valors de les constants  $K_p$ ,  $K_i$ ,  $K_d$  es tindran en compte diferents factors segons els objectius de control desitjats: 1) garantir l'estabilitat per tots els punts d'operació del sistema de control, i 2) complir les especificacions desitjades (rapidesa, precisió, mínim consum d'energia, etc.) per tots els punts d'operació del sistema de control. A continuació es presenta un possible algorisme per trobar un ventall de PID's que garantiran un sistema de control estable quan la planta és lineal.

### 7.1.1. Algorisme per trobar PID's que estableixen una planta lineal

Per tal de trobar els valors d'un controlador PID que fan estable una planta general, s'ha programat una funció per l'aplicació MATLAB, que retorna un gràfic en tres dimensions d'eixos  $K_i$ ,  $K_d$  i  $K_p$  amb la regió a on és estable la planta considerada. Aquesta funció es basa en l'article [Ming-Tzu Ho, A. Datta i S.P. Bhattacharyya, 1997a].

El sistema a controlar és:

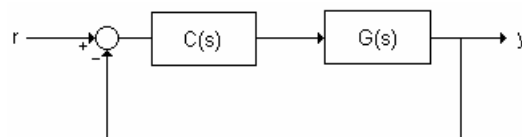


Figura 7.2: Esquema de control

on  $C(s)$  és la funció de transferència del controlador PID i  $G(s)$  la de la planta a controlar:



$$C(s) = Kp + \frac{Ki}{s} + Kd \cdot s = \frac{Kd \cdot s^2 + Kp \cdot s + Ki}{s} \quad (\text{Eq. 7.2})$$

$$G(s) = \frac{N(s)}{D(s)} \quad (\text{Eq. 7.3})$$

Amb aquestes expressions, el polinomi característic de la planta en llaç tancat és el següent:

$$\delta(s) = sD(s) + (k_i + k_d s^2)N(s) + k_p N(s) \quad (\text{Eq. 7.4})$$

El problema d'estabilitzar la planta usant un PID consisteix en determinar el valors de  $k_i$ ,  $k_d$  i  $k_p$  pels quals el polinomi característic en llaç tancat  $\delta(s)$  és Hurwitz. Això es, determinar els valors pels quals es compleix que totes les arrels del polinomi característic tenen part real negativa, és a dir, estan situades en el semiplà esquerre.

El funcionament de l'algoritme és el següent:

Per a un cert valor determinat de valors de  $Kp \in [Kpmin, Kpmax]$ , pels quals s'ha trobat prèviament que el sistema és estable, i amb un cert nombre d'iteracions  $h$ , es procedeix a realitzar els següents passos:

Pas 1) Fixar  $Kp$  amb el valor de  $Kpmin$ .

Pas 2) Determinar els valors de  $Kd$  i  $Ki$  que fan Hurwitz el polinomi característic (Eq.7.4), que es donen en forma de inequacions.

Pas 3) Trobar la intersecció de totes les inequacions per tal de trobar els vèrtexs de la regió.

Pas 4) Dibuixar la regió en un gràfic en tres dimensions, amb els eixos  $Kd$  i  $Ki$  en el pla i  $Kp$  l'alçada.

Pas 5) Trobar el següent valor de  $Kp$  amb el nombre  $h$  i anar a 2) i repetir fins al valor  $Kp = Kpmax$ , de forma que es tindrà el dibuix de  $h$  plans per a  $h$  valors diferents de  $Kp$ .

Per imposar que un polinomi és Hurwitz, si aquest és de la forma següent,

$$L(s) + (Kd \cdot s^2 + Kp \cdot s + Ki) \cdot M(s) \quad (\text{Eq. 7.5})$$

amb un  $Kp$  fixat, s'utilitza l'algoritme extret del Teorema Hermite-Biehler per a polinomis, que retorna un cert nombre d'inequacions amb variables  $Ki$  i  $Kd$ , tant si els coeficients de  $L(s)$  i  $M(s)$  són reals o si són imaginaris, amb  $M(s)=N(s)$  i  $L(s)=s \cdot D(s)$ .

La funció de MATLAB que executa aquest algoritme té la següent capçalera:

**PID\_W(N, D, Kpmin, Kpmax, h)**

on les variables d'entrada són:

- N i D són els polinomis que representen el numerador i denominador de la planta  $G(s)$  respectivament.



- $K_{pmin}$  i  $K_{pmax}$  són els valors màxim i mínim de la constant  $K_p$  del controlador PID.
- $h$  és el nombre de plans i iteracions a realitzar entre els valors  $K_{pmin}$  i  $K_{pmax}$ , ambdós inclosos.

El diagrama de flux de l'algorisme, de forma simplificada, vindria a ser:

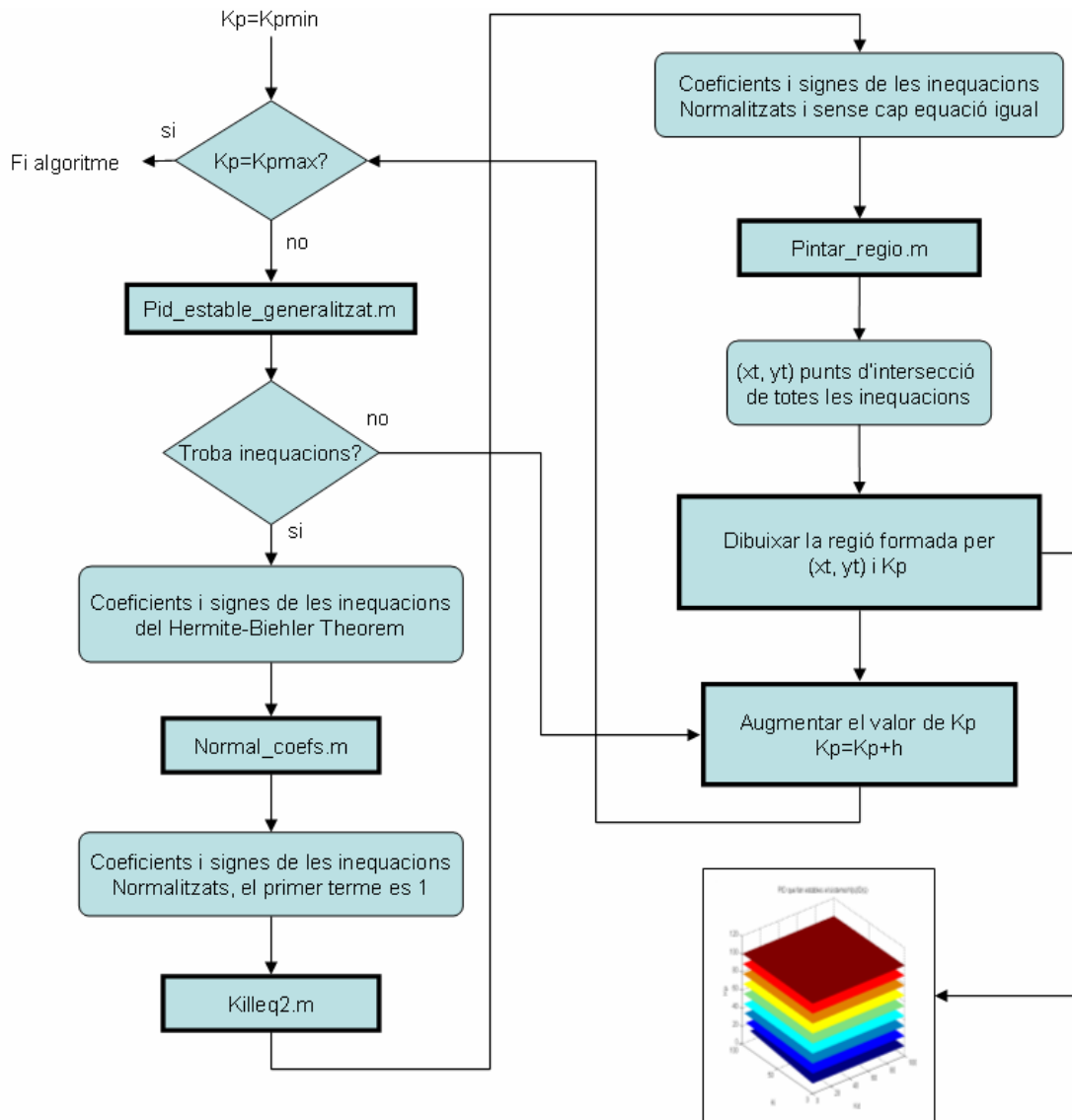


Figura 7.3: Diagrama de flux de l'algorisme

On *Pid\_estable\_generalitzat.m*, *Normal\_coefs.m*, *Killeq2.m* i *pintar\_regió.m* són funcions creades en MATLAB.

### 7.1.2. Exemples

A continuació, es procedirà a comprovar el bon funcionament de l'algorisme implantat, mitjançant la realització d'un seguit d'exemples extrets de [Ming-Tzu Ho, A. Datta i S.P. Bhattacharyya, 2000] i [Ming-Tzu Ho, A. Datta i S.P. Bhattacharyya, 1997b].



- 1) Considerar la planta  $G(s) = \frac{s^3 - 4s^2 + s + 2}{s^5 + 8s^4 + 32s^3 + 46s^2 + 46s + 17}$  amb  $K_p \in [-10, 5]$  i  $h=25$ .

El gràfic resultant és el següent:

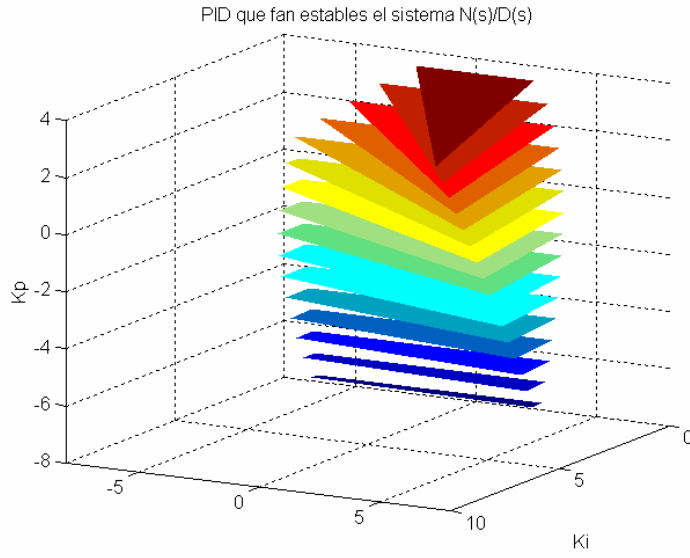


Figura 7.4: Regions (PID) que estabilitzen la planta

- 2) Considerar la planta  $G(s) = \frac{s^3 + 3s^2 + s + 8}{s^4 + 2s^3 + 3s^2 + 7s + 14}$  amb  $K_p \in [-3.5, 1.6]$  i  $h=55$ .

El gràfic resultant és el següent:

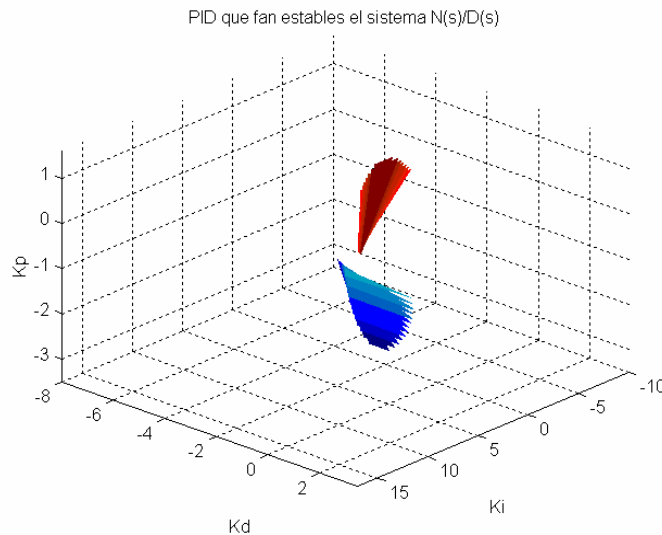


Figura 7.5: Regions (PID) que estabilitzen la planta



- 3) Considerar la planta  $G(s) = \frac{s-100}{s^4 + 5s^3 + 10s^2 - s + 1}$  amb  $K_p \in [-0.16, 0.02]$  i  $h=25$ .

El gràfic resultant és el següent:

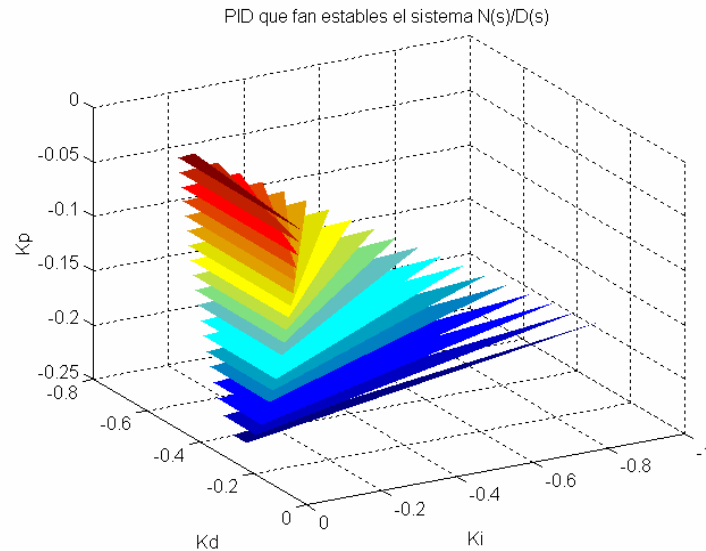


Figura 7.6: Regions que estabilitzen la planta

- 4) Considerar la planta  $G(s) = \frac{1}{(s+1)^8}$  amb  $K_p \in [-0.16, 0.02]$  i  $h = 25$ .

El gràfic resultant és el següent:

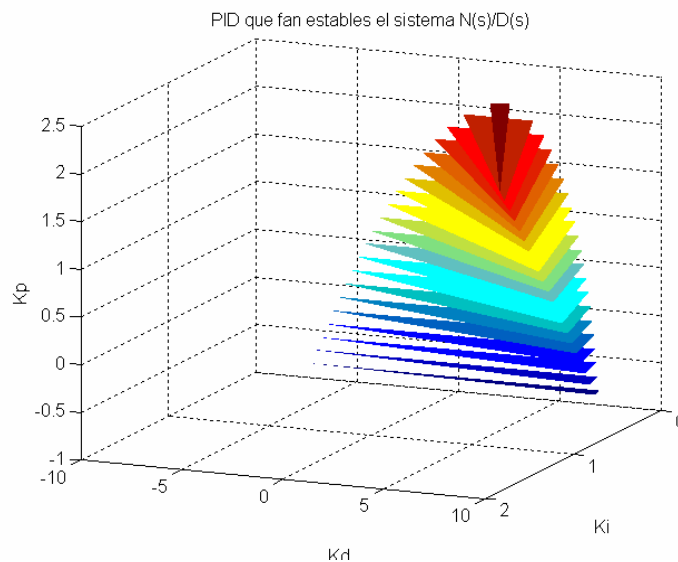


Figura 7.7: Regions que estabilitzen la planta

Els resultats obtinguts són els mateixos presentats en [Ming-Tzu Ho, A. Datta i S.P. Bhattacharyya, 2000], i per tant es pot considerar que la implementació de l'algorisme és correcta.



## 7.2. Control robust

### 7.2.1. Introducció

L'últim objectiu que té el disseny d'un sistema de control és el de construir un sistema que treballi en el medi real realitzant unes tasques determinades de manera eficaç, amb el mínim cost i de manera respectuosa amb el medi ambient. Degut a que aquest sistema pot canviar amb el temps (per exemple, els components poden envellir o els paràmetres poden variar amb la temperatura), o ho faran les condicions d'operació (canvis en la càrrega, pertorbacions), el sistema de control ha de ser capaç de suportar aquestes variacions sense canviar notablement el seu funcionament.

De fet, la majoria de dissenys de control es basen en l'ús d'un model de control i les característiques de disseny dependran en gran mesura de la fidelitat amb la que el model de control descriu el comportament del sistema. Un dels principis de modelització de sistemes per a control és el de la simplificació, que consisteix en que de la forma més senzilla possible el model capti els aspectes fonamentals del procés real.

Un procés real pot ser extremadament complex mai serà descrit d'una manera molt precisa pel model de control ja que aquest necessita de senzillesa per tal d'aplicar els mètodes d'anàlisi i de disseny analític de controladors lineals. Per tant, el model emprat per fer el disseny del controlador tindrà errors, el que s'anomena els errors de modelat. Unes de les principals hipòtesis simplificadores que contribueixen a incrementar els errors en el model són el descriure el sistema com un model lineal i invariant en el temps. Per tant, qualsevol model matemàtic de control que representi a un sistema real serà en més o menys grau imprecís, és a dir, comptarà amb imprecisions o errors de modelització. Llavors, és possible doncs que el sistema de control dissenyat usant models molt simplificats no funcioni com s'espera al provar-lo en sistemes reals.

### 7.2.2. Objectius de control

Per tant, si es desitja controlar eficientment un sistema real s'hauran de tenir en compte les possibles fonts d'incertesa existents i avaluar-ne l'efecte.

Quan un sistema de control té la propietat d'operar satisfactòriament en situacions reals es diu que és *robust*. Això significa que el controlador ha de comportar-se correctament no només per una sola planta, sinó per una família de plantes. El problema de dissenyar controladors que satisfacin els requisits d'estabilitat robusta s'anomena control robust.

Els objectius del control són, doncs, que el controlador que es dissenyi funcioni acceptablement quan s'implanti en el procés real. Aquest objectiu es pot descomposar en un seguit de subobjectius:

Primer de tot es tracta d'aconseguir estabilitzar el sistema en llaç tancat operant sota unes condicions de treball donades, les condicions nominals. Això és el que s'anomena Estabilitat Nominal.





Un cop es té que el sistema és estable sota les condicions nominals, es tracte d'assolir un seguit de prestacions, és a dir, que el sistema tingui un comportament òptim en quant a aspectes com ara la velocitat de reacció, el temps d'establiment, etc. Això és el que s'anomena com a Comportament Nominal.

Si a més el sistema és estable no només pel model nominal sinó també pel conjunt de totes les plantes possibles, llavors es diu que s'ha assolit l'Estabilitat Robusta.

Finalment, si es considera que per totes les plantes possibles no és suficient que el sistema sigui estable sinó que es pretén optimitzar-ne el funcionament, es parla de Comportament Robust.

El problema de control es pot resumir a partir del següent esquema:

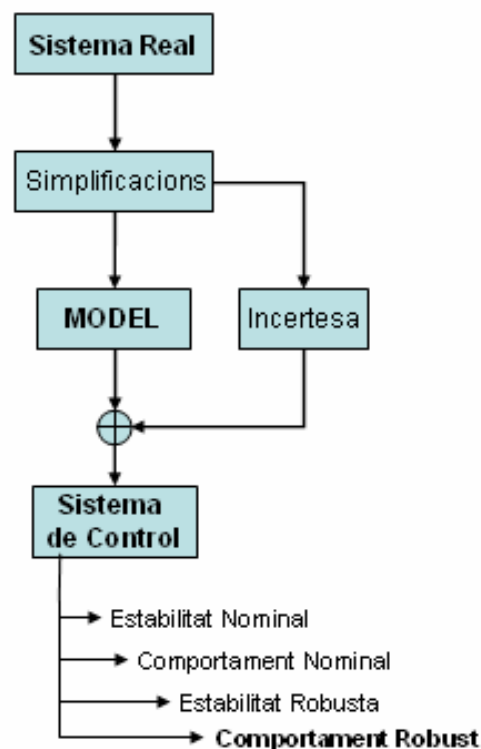


Figura 7.8. Problema del control robust

### 7.2.3. Representació de les incerteses

Com s'ha dit anteriorment, la majoria de dissenys de control es basen en l'ús d'un model de disseny o simplificat. La relació entre aquest model i la realitat és subtil i complexa. Un model matemàtic de control ens dona la relació entre les entrades (excitacions) i les sortides (respostes) del sistema. La qualitat i "bondat" del model de control depèn de quant de properes estiguin les seves respostes respecte a les de la planta real.

El terme incertesa es refereix a les diferències o errors entre els models i la realitat. Les principals fonts d'incertesa, és a dir, les principals raons de la imperfecció d'un model són: 1) La no contemplació de les no linealitats (friccions, histèresis, etc.). Els



processos reals són no lineals, mentre que en la majoria dels casos el model de control s'ha obtingut mitjançant linealització. 2) Un sistema real no pot ser identificat de manera exacta degut a la mancança de precisió dels dispositius de mesura. 3) Un model teòric no pot ser implementat de manera exacta degut a les toleràncies de la fabricació dels components. 4) El comportament d'un sistema real varia respecte al nominal de manera imprevista per causes ambientals, per les propietats dels materials, per l'envelliment i els canvis del mode d'operació que provoquen una variació dels paràmetres del procés. 5) Sovint el model matemàtic és una simplificació dels processos reals per a permetre i facilitar el tractament analític. Per tant, fonts importants d'incertesa són també la mancança de coneixements teòrics, la reducció deliberada de l'ordre (no tenint en compte les dinàmiques ràpides per tal d'aconseguir un model més senzill), la no modelització dels fenòmens d'alta freqüència.

Existeixen diverses tècniques matemàtiques d'anàlisi de sistemes de control robust. Aquestes tècniques es corresponen amb el tipus d'incertesa del model. En general, es pot dir que n'hi ha dues grans maneres de tractar amb sistemes incerts que són l'enfocament freqüencial i el polinòmic. La incertesa del model es divideix generalment en dues categories: incertesa estructurada i incertesa no estructurada.

La incertesa estructurada és aquella de la que es coneix l'origen de la seva estructura amb molta precisió. Es disposa dels models d'aquestes i dels marges i cotes de certs paràmetres del sistema (és a dir, podem tenir una funció de transferència vàlida del model d'un sistema, però no obstant això tenir certa incertesa de la localització exacta dels zeros, pols o el guany del sistema). Exemples:

La incertesa no estructurada suposa un menor coneixement del sistema. És a dir, són les incerteses de naturalesa més desconeguda, com les associades a les dinàmiques no conegudes, truncament dels modes d'altres freqüències, etc. Únicament es té en compte que la resposta freqüencial del sistema estigui entre dues cotes. S'admet que l'estructura del model no sigui exacta. Per tant, és indicadora de la nostra capacitat per a representar la màxima informació sobre la planta en un model el suficientment simple com per a poder treballar amb ell. La inexactitud del model pot ser deguda a canvis en els paràmetres, dinàmiques desestimades i reduccions d'ordre.

La seva representació típica és per mitja de funcions de transferència que es sumen o multipliquen a la funció de transferència del model de control. Quan una família de sistemes incerts es descriu en el domini de la freqüència, la funció de transferència està deteriorada per termes additius o multiplicatius, llavors parlarem d'una funció de transferència pertorbada,  $G_0 + \Delta G = G_0 + I_a$  o  $G_0(1 + \Delta G) = G_0(1 + I_m)$ .

Aquesta descripció té l'avantatge de ser útil per a treballar amb sistemes relativament complexes permetent, a més, caracteritzar fàcilment les especificacions de funcionament del sistema de control i, per tant, possibilitant el disseny del sistema de control. Pel contrari, la interpretació física de les pertorbacions es perd, ja que es treballa amb pertorbacions matemàtiques. Això fa que el mètode d'anàlisi i disseny quedi molt llunyà del sistema físic. A més, quan es coneixen les incerteses estructurades al utilitzar el model freqüencial es perd informació, sobredimensionant el model de les pertorbacions.



Dintre de les incerteses no estructurades existeixen de dos tipus: les incerteses additives i les multiplicatives (veure Figura 7.9). Les incerteses multiplicatives són les més freqüentment emprades degut a que satisfan les propietats intuïtives de ser petites a baixes freqüències (on el model de la planta nominal és generalment conegut), i per un altre costat són elevades a altes freqüències (on el model és sempre més imprecís). Habitualment el nivell d'incertesa augmenta amb la freqüència, degut principalment a la dinàmica no modelada d'alta freqüència (sensors, actuadors, models d'ordre reduït de la planta). Aquest tipus d'incertesa es caracteritza de la següent manera:  $G=G_0(1+\Delta G)=G_0(1+I_m)$ , on  $G$  representa a la planta real,  $G_0$  és el model nominal de la planta real i  $I_m$  és el model d'incertesa multiplicativa.

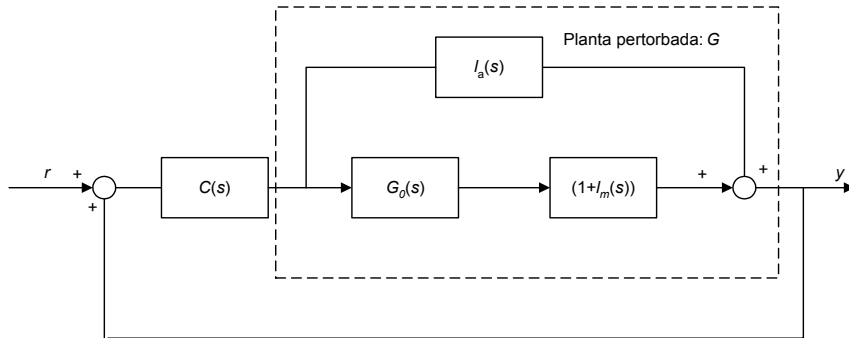


Figura 7.9: Sistema amb incertesa additiva i incertesa multiplicativa de sortida.

Ambdues classes d'incertesa estan presents en moltes aplicacions. En el nostre cas, com el propòsit és controlar un robot de més d'un grau de llibertat, que és un sistema no lineal, variable amb el temps i amb acoblaments, és molt adequat emprar una metodologia de control robust. Com no tenim suficient informació del sistema per modelitzar una incertesa estructurada, hem d'optar per modelitzar la incertesa del model de control del robot mitjançant una incertesa per fer la síntesi d'un control robust no estructurat, i per tant serà adequat emprar una metodologia de control no estructurat, com el control  $H_\infty$ , tal i com s'explica en [Zhou, Doyle J. i Glover K., 1996] i [Rodríguez Rubio F., López Sánchez M.J., 1996].

Vist això, a continuació es procedirà a ampliar l'algorisme anterior (per a sistemes lineals) per tal que pugui treballar amb incerteses i poder obtenir un controlador que funcioni correctament si es produeixen variacions en la planta.

### 7.3. PID's robustos. Un compromís entre la simplicitat i el garantir l'estabilitat i la robustesa

La necessitat de garantir l'estabilitat malgrat els errors del model causats per diversos factors, anteriorment citats, com les no linealitats, soroll fe mesura, etc., ha generat la necessitat en els darrers anys de desenvolupar tècniques de control robustes, que garanteixen l'estabilitat i el 'performance' (comportament desitjat) per a totes les plantes (model nominal amb incertesa) que representen el comportament del sistema real. No obstant a aquesta avantatge, els controladors robustos presenten alguns inconvenients com el que són d'un ordre molt gran i per tant difícils d'implementar i resintonitzar. Degut a aquest aspecte, i a que l'aplicació d'aquest Projecte és merament pràctica, s'ha vist la necessitat d'emprar un controlador amb estructura de baix ordre, però que a la vegada contempli les



incerteses de la planta. Per tant, s'ha cregut necessari combinar la simplicitat dels controladors PID amb les garanties d'estabilitat que el control robust ens proporciona quan s'ha de controlar una planta no lineal, s'ha optat per fer la síntesi de controladors PID's robustos. A continuació, es presenta la modificació de l'algorisme explicat a l'apartat 7.1.1 incorporant la incertesa.

### 7.3.1. Algorisme per trobar PID's que estableitzen una planta amb incertesa

L'ampliació de l'algorisme anterior per tal de poder treballar amb funcions d'incertesa es basa en la implementació de l'Algorisme de [Ming-Tzu Ho, 2003].

Les bases d'aquest algorisme són les mateixes que l'anterior: es té el mateix esquema de control (Figura 7.2), amb les funcions de transferència de la planta i el controlador definides igualment per les equacions (Eq.7.2) i (Eq.7.3):

$$G(s) = \frac{N(s)}{D(s)}$$

$$C(s) = Kp + \frac{Ki}{s} + Kd \cdot s = \frac{Kd \cdot s^2 + Kp \cdot s + Ki}{s}$$

D'aquesta forma, la planta total del sistema en llaç tancat és la següent:

$$T(s, Kp, Ki, Kd) = \frac{A(s) + (Kd \cdot s^2 + kp \cdot s + Ki) \cdot B(s)}{s \cdot D(s) + (Kd \cdot s^2 + Kp \cdot s + ki) \cdot N(s)} \quad (\text{Eq. 7.6})$$

Ara cal imposar que el Denominador del sistema, és a dir, el polinomi característic, sigui Hurwitz (que tingui totes les arrels amb part real negativa). Cal remarcar que aquest denominador depèn dels valors del controlador  $Kd$ ,  $Kp$  i  $Ki$ , com ja s'ha vist en l'equació (Eq.7.4):

$$\delta(s) = sD(s) + (k_i + k_d s^2)N(s) + k_p N(s)$$

Per tal d'imposar que també sigui estable tenint en compte la funció d'incertesa  $W(s)$  cal imposar el criteri d'optimització  $H_\infty$  amb un nombre donat  $\gamma$  de forma que es compleixi l'estabilitat robusta:

$$\|W(s) \cdot T(s, Kp, Ki, Kd)\|_\infty < \gamma \quad (\text{Eq. 7.7})$$

$$\text{amb } W(s) = \frac{W_n(s)}{W_d(s)}$$

que equival a les dues condicions següents:

$$\left[ sW_d(s) \cdot D(s) + \frac{1}{\gamma} e^{j\theta} W_n(s) A(s) \right] + (Kd \cdot s^2 + Kp \cdot s + Ki) \cdot \left[ W_d(s) \cdot N(s) + \frac{1}{\gamma} e^{j\theta} W_n(s) B(s) \right]$$

$$\text{sigui Hurwitz amb } \theta \in [0, 2\pi] \quad (\text{Eq. 7.8})$$



$$|W(\infty)T(\infty, Kp, Ki, Kd)| < \gamma \quad (\text{Eq. 7.9})$$

Per tant, el funcionament de l'algorisme de forma simplificada vindria a ser:

Per a un cert valor determinat de valors de  $Kp \in [Kpmin, Kpmax]$  i amb un cert nombre d'iteracions  $h$ , es procedeix amb el següent algoritme:

Pas 1) Fixar  $Kp$  amb el valor de  $Kpmin$ .

Pas 2) Determinar els valors de  $Kd$  i  $Ki$  que fan Hurwitz el polinomi descrit per (Eq.7.4), que es donen en forma de inequacions.

Pas 3) Determinar els valors de  $Kd$  i  $Ki$  que fan Hurwitz el polinomi (Eq.7.8) per a una discretització de  $\theta$  dins l'interval  $[0, 2\pi]$ , que es donen en forma de inequacions.

Pas 4) Trobar l'inequació resultant de (Eq.7.9) amb el valor de  $Kp$  fixat en el pas 1.

Pas 5) Reunir totes les inequacions dels passos 1), 2) i 3) i dibuixar la regió en un gràfic en tres dimensions, amb els eixos  $Kd$  i  $Ki$  en el pla i  $Kp$  l'alçada.

Pas 6) Trobar el següent valor de  $Kp$  amb el nombre  $h$  i anar a 2) i repetir fins al valor  $Kp = Kpmax$ , de forma que es tindrà el dibuix de  $h$  plans per a  $h$  valors diferents de  $Kp$ .

Per imposar que un polinomi és Hurwitz, si aquest és de la forma següent,

$$L(s) + (Kd \cdot s^2 + Kp \cdot s + Ki) \cdot M(s) \quad (\text{Eq. 7.10})$$

amb un  $Kp$  fixat, s'utilitza l'algoritme extret del Teorema Hermite-Biehler per a polinomis complexos, que retorna un cert nombre d'inequacions amb variables  $Ki$  i  $Kd$ , tant si els coeficients de  $L(s)$  i  $M(s)$  són reals com passa en el pas 2) o si són imaginaris com en el 3). Per anular la funció d'incertesa cal aplicar l'algorisme amb

$$W(s) = \frac{0}{1}.$$

El funcionament de la funció per a l'aplicació MATLAB és el que es mostra a continuació, la funció és cridada amb la següent capçalera:

#### **PID\_W(N, D, Wn, Wd, Kpmin, Kpmax, gamma, h)**

- N i D són els polinomis que representen el numerador i denominador de la planta  $G(s)$  respectivament.
- Wn i Wd són els polinomis que representen el numerador i denominador de la funció d'incertesa.
- Kpmin i Kpmax són els valors màxim i mínim de la constant  $Kp$  del controlador PID.
- Gamma és el valor  $\gamma$  de l'equació (Eq. 7.9).
- h és el nombre de plans i iteracions a realitzar entre els valors Kpmin i Kpmax, ambdós inclosos.

A continuació es mostren les subfuncions més importants que crida la funció principal:

**[signe,coeficients]=PID\_general(N,D,Wn,Wd,Kp,gamma)**, equival a l'anterior amb només un valor de  $k_p$ . Dins d'aquest es troben:



**[signe,coeficients]=pid\_estable\_generalitzat(L,M,Kp)**, retorna les inequacions del pas 1) amb un vector de signes i un vector de coeficients. Consisteix en l'algoritme extret del Teorema Hermite-Biehler.

**[signe,coeficients]=pid\_theta(N,D,Wn,Wd,Kp,gamma)**, retorna les inequacions del pas 2) amb un vector de signes i un vector de coeficients.

**[coeficients]=normal\_coefs(coeficients)**, normalitza els valors del vector coeficients resultant de arregar totes les inequacions, de forma que el primer valor serà un 1.

**[signe,coeficients]=killeq2(signe,coeficients)**, compara totes les inequacions entre si i elimina les que siguin equivalents.

**[vectorw]=stepone(qf)**, retorna el vector  $\omega$  de les arrels d'un polinomi.

**[sigma]=sigma(deltas)**, retorna el nombre d'arrels en el semiplà esquerra menys el nombre d'arrels en el semiplà dret d'un polinomi delta.

**[xt,yt]=pintar\_regió(signe, coeficients)**, retorna un vector  $x_t$  i un  $y_t$  amb les coordenades  $x$  i  $y$  respectivament dels vèrtexs ordenats de la intersecció de totes les inequacions. Dins d'aquest es poden trobar les següents funcions:

**[xi,yi]=intersecció(recta1, recta2)**, retorna el vector intersecció de dues rectes.

**[xt, yt]=killiguals(xt, yt)**, elimina els punts que apareixen dos cops dins dels vectors  $(x_t, y_t)$ .

Totes aquestes funcions es poden consultar en l'Annex B.



El seu diagrama de flux simplificat és el mostrat a continuació:

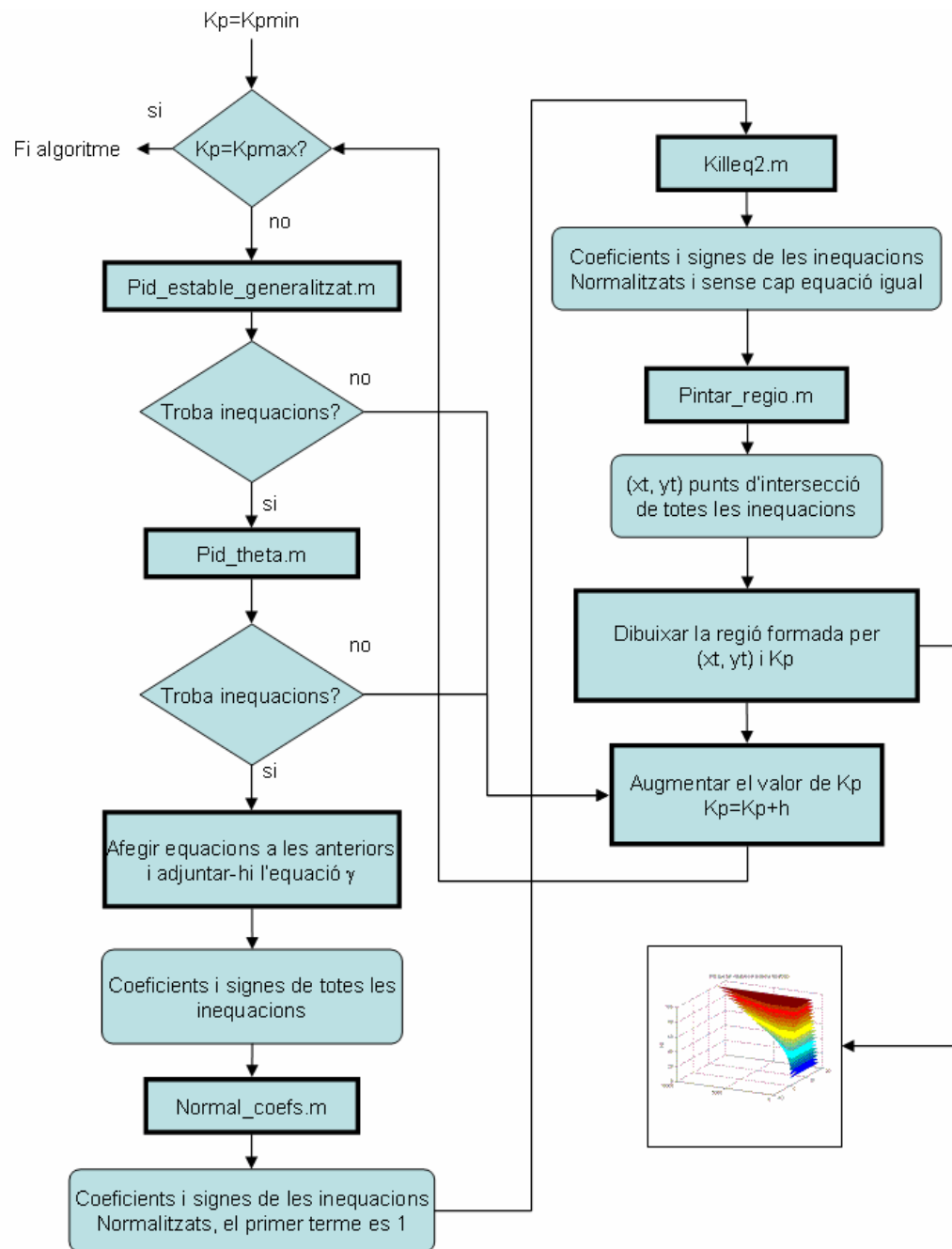


Figura 7.10: Diagrama de flux de l'algoritme amb incertesa



### 7.3.2. Exemples

Considerar la planta  $G(s)$  i la funció d'incertesa  $W(s)$ :

$$G(s) = \frac{s-1}{s^2 + 0.8s - 0.2} \quad W(s) = \frac{s+0.1}{s+1}$$

Prenent els valors  $K_{pmin}=-1.8$ ,  $K_{pmax}=-0.2$ ,  $\gamma=1$  i  $h=20$  el resultat es mostra a continuació:

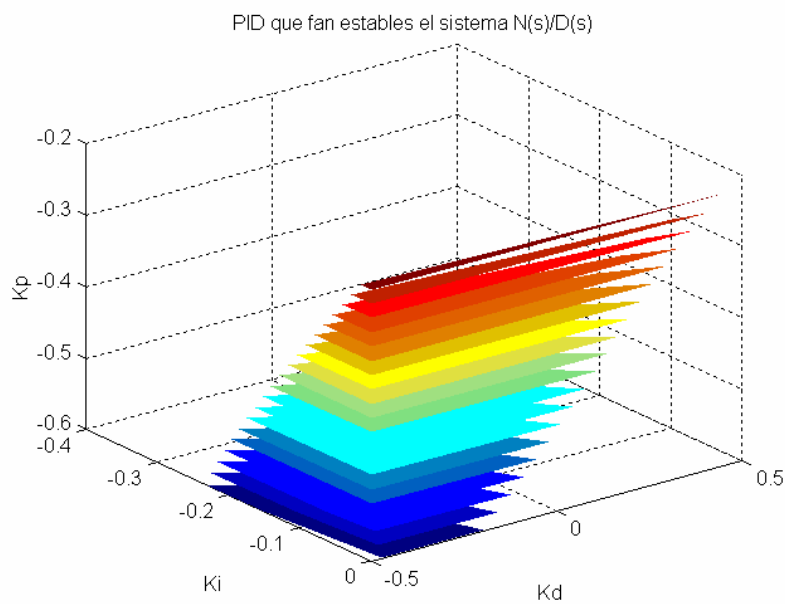


Figura 7.11: Regions que estabilitzen el sistema

El resultat és exactament el mateix que es presenta a [Ming-Tzu Ho, 2003] i per tant es pot considerar que la implementació de l'algorisme és correcte.





## 8. Control del robot de laparoscòpia

### 8.1. Introducció

Pel sistema robòtic mèdic descrit en el Capítol 5 resulta apropiat utilitzar un control robust degut principalment al següent motiu: el robot és un sistema fortament no lineal i si s'escull un mètode de control lineal és necessari que aquest contempli els possibles errors del model de control, ja que aquest és un model lineal i simplificat, només vàlid pel mode d'operació nominal i que no contempla la dinàmica no lineal, els fenòmens d'alta freqüència (HF), la dinàmica dels actuadors, etc.

El fet d'incloure incertesa en el model fa que es tinguin les no linealitats i els possibles intervals de variació dels paràmetres del model simplificat obtingut en una zona de treball determinada i, per tant, garantirà l'estabilitat i possibilitarà que es compleixin les especificacions desitjades (*performance*) en aquesta zona d'operació del sistema. Encara que mitjançant controladors robustos la solució de control serà bastant "conservadora", ja per garantir estabilitat aquests mètodes de control, i en concret la metodologia  $H_\infty$  (control robust no estructurat), consideren el pitjor cas (*worst case*), s'ha considerat interessant i innovador aplicar i provar aquestes metodologies de control avançat a aquest sistema, ja que aquest té unes característiques apropiades per aquest tipus de control.

A continuació, es resoldrà un problema d'enginyeria de sistemes (*solving problem*) en el que s'ha de plantejar i resoldre el problema de control següent: realització d'un control visual (*visual servoing*) per tal de fer un seguiment precís, acurat i ràpid de la trajectòria del centre de les eines sempre garantint l'estabilitat del sistema de control.

### 8.2. Problema de control

Tal com es va explicar en el Capítol 5, i s'ha citat anteriorment, el problema de control que es planteja en aquest Projecte és el de guiar de manera efectiva durant una intervenció de laparoscòpia la posició de la càmera per tal que aquesta sigui en tot moment la que permeti una millor visió (seguiment precís, acurat i ràpid de les eines mitjançant un *feedback* 'visual') al cirurgià de la zona que està operant.

### 8.3. Model de control i de les incerteses

Des del punt de vista de control d'un robot es pot dir que existeixen dues condicions de moviment ben diferenciades. La primera es tracta del cas en que el robot necessita realitzar una determinada tasca on necessita moure's amb esforços grans i prolongats de les variables de control (motors que proporcionen més o menys tensió - i en conseqüència parell - a les articulacions), on els efectes no lineals dominen el comportament del sistema. L'altra es dona en situacions de manteniment de la posició, o quan el procés de regulació davant un canvi de consigna requereix unes desviacions de les articulacions del robot que no són excessives ni en temps ni magnitud. Sota aquestes condicions el sistema es pot aproximar per un model lineal.



Concretament, com el model de control del robot ha de ser un model simple però a la vegada precís, ja que interessa que descriu els aspectes més rellevants de la dinàmica del robot, s'ha realitzat una simplificació del model no lineal, complet i complex (simulador, que s'empra com a model "real") representat per les equacions del moviment (Eq.6.23) mitjançant la seva linealització (veure Capítol 6.5) al voltant d'una solució estacionària per a unes condicions nominals de funcionament, i també no tenint en compte la seva dinàmica a altes freqüències (es desprecia el terme de la gravetat).

Llavors, en principi, el model linealitzat que s'ha trobat tan sols és correcte si el robot es mou en posicions al voltant del punt nominal, mentre que al allunyar-se d'aquesta configuració el model deixaria de ser vàlid. Això és degut a que les equacions de la dinàmica depenen de les variables articulars.

Per tal d'assegurar que el controlador que es dissenyi sigui estable per a totes les possibles configuracions del robot (és a dir, en l'àrea de treball) es linealitzarà el model no lineal en diferents punts dins de l'àrea de treball i es veurà quina és la variació dels paràmetres del sistema que es produeix. Per tant, en alguns punts d'aquesta possible àrea de funcionament existiran errors del model de control, que posteriorment s'utilitza per fer la síntesi del controlador. Els errors deguts a la no linealitat, al no tenir en compte les altes freqüències, el soroll i possibles perturbacions s'han de considerar d'alguna manera per tal de garantir un control amb estabilitat robusta, és a dir per qualsevol dels possibles models de la planta. Això es pot mitjançant la modelització de les incerteses del model. Com es va explicar en el Capítol 4, existeixen dos tipus d'incerteses (les estructurades i les no estructurades). En aquest cas, s'utilitzaran les no estructurades ja no tenim la suficient informació per tal de que l'algorisme de control robust  $H_\infty$  utilitza aquest tipus d'incerteses.

Per simplicitat es linealitzarà el model (sense tenir en compte la gravetat) en els vèrtexs de l'àrea de treball del robot:  $\theta_1 \in (-0.1745, 0.1745)$  [rad],  $s_2 \in (-0.1, 0.1)$  [m],  $s_3 \in (-0.1, 0.1)$  [m].

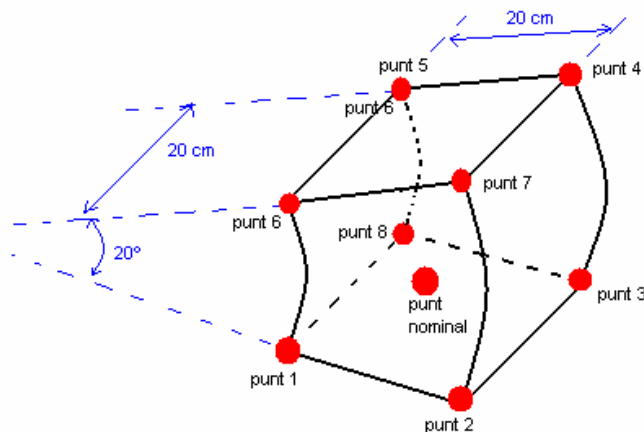


Figura 8.1. Àrea de treball del robot

Fent ús de la funció *lineal\_taylor.m* es linealitza el model complet en el punt nominal  $(\theta_1, s_2, s_3) = (0, 0, 0)$  [rad, m, m] i s'obté la següent matriu de funcions de transferència:



$$G(s) = \begin{pmatrix} \frac{6.933}{s^2} & 0 & \frac{2.114}{s^2} \\ 0 & \frac{0.2943}{s^2} & 0 \\ \frac{2.114}{s^2} & 0 & \frac{1.475}{s^2} \end{pmatrix}$$

A més, realitzant la linealització en els diferents punts de l'àrea de treball del robot s'arriba a la conclusió que els guanys de cadascuna de les funcions de transferència de les articulacions estan entre els següents intervals:

Articulació 1:  $k_1 \in (4.198, 13.45)$

Articulació 2:  $k_2 = 0.2943$

Articulació 3:  $k_3 \in (1.22, 2.081)$

Degut a que el sistema pot moure's en tota l'anterior àrea de treball (veure Figura 8.1), les seves diferents condicions d'operació, la no contemplació de la seva no linealitat, de la variació dels seus paràmetres (masses, moments d'inèrcia, etc.) amb el temps, de la dinàmica dels actuadors..., el model de control proposat presenta errors. Aquests errors es poden modelitzar mitjançant una funció d'incertesa no estructurada. Aquesta incertesa no estructurada es modelitzarà com una incertesa multiplicativa [Sigur Skogestad i I. Postlethwaite, 1997]:

Es defineix:

$$l_m(\omega) = \max_{G \in \Pi} \left| \frac{G(j\omega) - G_0(j\omega)}{G_0(j\omega)} \right| \quad (\text{Eq. 8.1})$$

on  $G_0(j\omega)$  és la funció de transferència de la planta nominal i  $G(j\omega)$  indica una de les plantes del conjunt  $\Pi$  de plantes possibles.

Llavors la funció pes s'haurà d'escollir de forma que:

$$|W(j\omega)| \geq l_m(\omega), \forall \omega \quad (\text{Eq. 8.2})$$

és a dir, que la magnitud de la resposta freqüencial de la funció de pes  $W(j\omega)$  sigui major o igual que la de la funció  $l_m(\omega)$ .

Per tant, es procedirà a calcular i dibuixar les respectives funcions pes per cadascun dels models linealitzats anteriorment en els diferents punts de l'àrea de treball, i també la funció de pes de les incerteses produïdes per factors tal com les no linealitats, la no contemplació de la dinàmica d'alta freqüència, etc. A continuació, es presentaran els diagrames de Bode d'aquestes funcions de ponderació per cadascuna de les articulacions.

S'emprarà la funció de SIMULINK 'pes.m' per a trobar les funcions  $l_i(\omega)$  i posteriorment es dibuixarà el diagrama de Bode:



Per a la primera articulació:

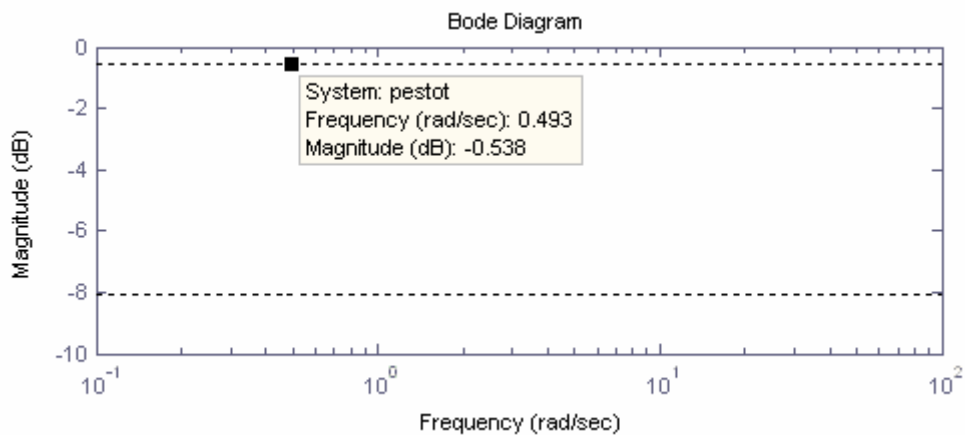


Figura 8.2. Diagrama de Bode de la funció de pes degut a a la variació del punt de treball

A partir del diagrama de Bode anterior es pot trobar una funció d'incertesa adequada pel model.

$$-20 \log(K) = -0.538 \Rightarrow K = 0.9339$$

Per tant, la funció de pes hauria de ser:

$$W_1'(s) = 0.9339 \quad (\text{Eq. 8.3})$$

Si es té en compte els altres paràmetres, com són el funcionament dels actuadors, les variacions en les masses i l'efecte no lineal que introdueix el pes s'obté la funció:

$$W_1''(s) = \frac{0.1s^2 - 4s - 4}{0.025s^2 + 4s + 4} \quad (\text{Eq. 8.4})$$

Per aquest motiu, s'escull una funció d'incertesa que englobi a les dues. La funció pes escollida és:

$$W_1(s) = \frac{0.1s + 1}{0.025s + 1} \quad (\text{Eq. 8.6})$$

En la Figura 8.3 es poden observar les tres funcions descrites, i com la funció  $W_1(s)$  té una magnitud sempre superior a les dues primeres.



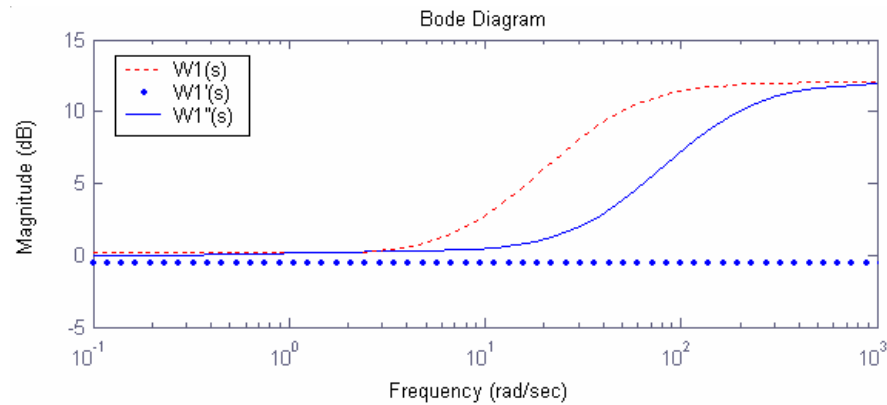


Figura 8.3. Diagrama de Bode de la funció pes

Per a la segona articulació:

Tal i com s'observa en la matriu de funcions de transferència anterior i en la variació del guany en la funció de transferència de l'articulació 2, aquesta és independent del punt on es linealitzava i no es produeix cap variació de paràmetres, no es produeix cap variació de paràmetres, a més a més, la segona articulació no es veu afectada per l'efecte de la força pes. És per això que és correcte dir que per aquesta articulació no hi ha incertesa deguda a la linealització i simplificació del model, i per tant es pot considerar que la funció pes és zero:

$$W_2(s) = 0 \quad (\text{Eq. 8.7})$$

Per a la tercera articulació es té:

Seguint els mateixos passos i operant d'igual manera que en les altres dues articulacions es pot trobar la funció de pes per a l'articulació 3:

$$-20\log(K) = -7.72 \Rightarrow K = 0.411$$

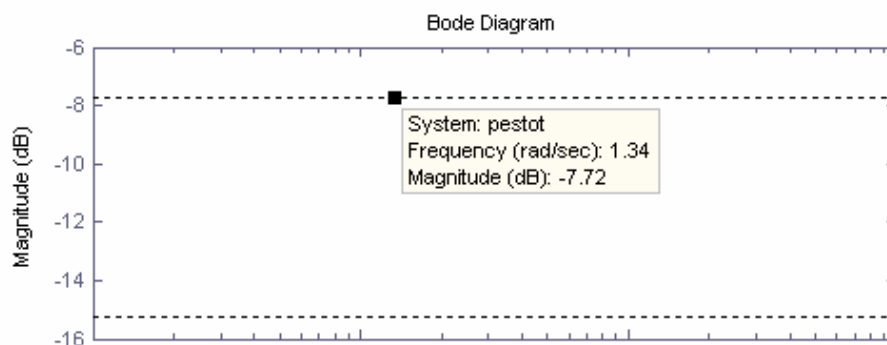


Figura 8.4. Diagrama de Bode de la funció pes degut a la variació del punt de treball

Per tant, inicialment la funció de pes és:

$$W_3'(s) = 0.411 \quad (\text{Eq. 8.8})$$

Però, si es té en compte l'efecte no lineal de la força pes, així com l'efecte de la variació de les masses s'obté la següent funció:



$$W_3''(s) = \frac{0.1s^2 - 4s - 4}{0.01s^2 + 4s + 4} \quad (\text{Eq. 8.9})$$

Cal trobar una funció pes que amb un guany superior a les dues, i la funció escollida és la següent:

$$W_3(s) = \frac{0.1s + 1}{0.01s + 1} \quad (\text{Eq. 8.10})$$

La Figura 8.5 mostra el diagrama de Bode per a les tres funcions, i es pot observar com la funció pes global escollida sempre es situa amb un guany superior a les anteriors.

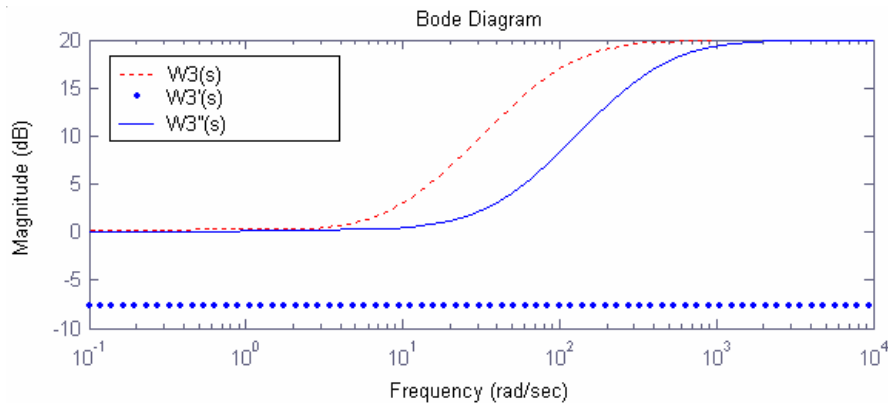


Figura 8.5. Diagrama de Bode per trobar la funció pes

## 8.4. Disseny del controlador robust descentralitzat

### 8.4.1. El problema de l'acoblament

El robot sotmès a estudi és un sistema MIMO i acoblat.

Si s'observa el comportament del sistema (veure Capítol 6) per a diferents entrades (parells) es pot veure clarament que la resposta de l'articulació 2 ( $s_2$ ) és independent dels parells aplicats a les altres dues articulacions. D'igual forma es pot observar que existeix un acoblament entre la primera i la tercera articulació. Això té lògica donat que el parell que s'haurà d'aplicar a l'articulació 1 serà diferent dependent de quina sigui la posició de l'articulació 3.

Aquest fet es pot corroborar a partir de la funció de transferència del sistema linealitzat. Usant MATLAB:

A partir de les matrius A, B, C, D trobades anteriorment, es crea el sistema i se'n troba la funció de transferència:

```
>> sys=ss(A,B,C,D)
>> G=tf(sys)
```

El resultat és la matriu de funcions de transferència següents:



$$G(s) = \begin{pmatrix} \frac{6.933}{s^2} & 0 & \frac{2.114}{s^2} \\ 0 & \frac{0.2943}{s^2} & 0 \\ \frac{2.114}{s^2} & 0 & \frac{1.475}{s^2} \end{pmatrix} \quad (\text{Eq. 8.11})$$

Les equacions del sistema serien doncs:

$$\begin{aligned} \theta 1(s) &= G_{11}(s) * u_1(s) + G_{12}(s) * u_2(s) \\ s2(s) &= G_{22}(s) * u_2(s) \\ s3(s) &= G_{31}(s) * u_1(s) + G_{33}(s) * u_3(s) \end{aligned} \quad (\text{Eq. 8.12})$$

i en forma de matriu:

$$\begin{pmatrix} \theta 1(s) \\ s2(s) \\ s3(s) \end{pmatrix} = \begin{pmatrix} G_{11}(s) & 0 & G_{13}(s) \\ 0 & G_{22}(s) & 0 \\ G_{31}(s) & 0 & G_{33}(s) \end{pmatrix} \bullet \begin{pmatrix} u_1(s) \\ u_2(s) \\ u_3(s) \end{pmatrix}$$

Si s'observa la funció de transferència G:

- El terme  $G_{13}$  és diferent de zero, per tant l'articulació 3 influeix en el moviment de l'articulació 1.
- D'igual manera, al ser  $G_{31}$  diferent de zero es pot dir que l'articulació 1 influeix en el moviment de l'articulació 3. Ambdues articulacions estan doncs acoblades.
- Els termes  $G_{12}$ ,  $G_{21}$ ,  $G_{23}$  i  $G_{32}$  són zero, i per tant el moviment de l'articulació 2 només dependrà del parell que se li aplica a ella mateixa. No té acoblament amb les altres dues.

L'acoblament existent es pot representar a partir d'un diagrama de blocs:

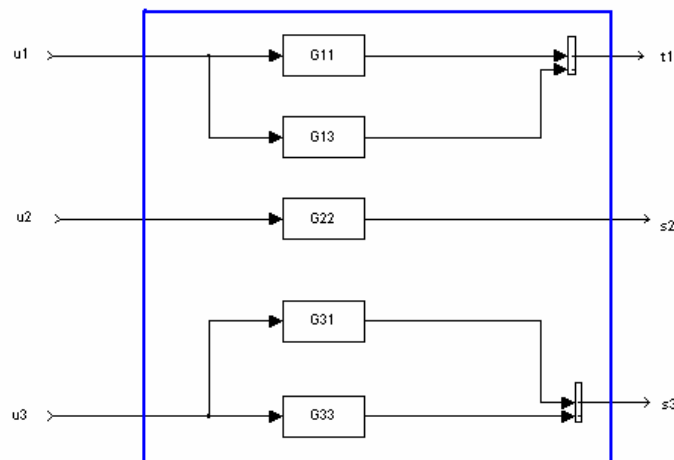


Figura 8.6: Esquema de l'acoblament existent



### 8.4.2. Control descentralitzat

És sabut que els sistemes MIMO presenten algunes característiques que requereixen un anàlisi especial. Quan les entrades i les sortides d'aquests sistemes estan acoblades, un canvi en qualsevol de les entrades tindrà efecte en més d'una de les sortides. En aquestes condicions, el problema de dissenyar un controlador MIMO pot esdevenir una tasca realment complexa [L. Acosta et al., 2002].

Si es tingués que  $G_{ij}(s)=0$  ( $i \neq j$ ), està clar que es podrien dissenyar els tres controladors de forma independent, com si el sistema fos la suma de tres subsistemes SISO totalment independents, fet que faria notablement més senzill i simple el disseny.

La majoria de sistemes MIMO presenten acoblaments i el control descentralitzat és una solució adequada quan es té un sistema en el qual  $G_{ij}(s) \neq 0$  ( $i \neq j$ ) i es decideix dissenyar  $m$ -controladors independents, essent  $m$  l'ordre del sistema, com si es tractés de subsistemes SISO totalment independents, sense cap mena d'acoblament. El problema consisteix en poder garantir que el sistema complet, amb els controladors SISO dissenyats, continuï sent estable, i per això s'hauran de complir un seguit de condicions.

Seguint el procediment exposat en [T.C.Yang, 2001] i [T.C.Yang, 1998]:

La funció de transferència  $G(s)=[g_{ij}(s)]$  ( $i,j=1,2,..m$ ) d'un sistema MIMO  $m \times m$  es pot descomposar en:

$$G(s) = \tilde{G}(s) + \hat{G}(s) \quad (\text{Eq. 8.13})$$

on  $\tilde{G}(s) = \text{diag}[g_{ii}(s)]_{i=1,2,..m}$  és una matriu diagonal formada pels elements de la diagonal de  $G(s)$  mentre que  $\hat{G}(s)$  té els elements de la diagonal zero i els elements fora la diagonal són iguals als de  $G(s)$ .

A més, es defineix:

$$E(s) = \hat{G}(s)\tilde{G}(s)^{-1} \quad (\text{Eq. 8.14})$$

$$\tilde{H}(s) = \tilde{G}(s)F(s)(I + \tilde{G}(s)F(s))^{-1} = \text{diag}[h_i(s)] \quad (\text{Eq. 8.15})$$

$$H(s) = G(s)F(s)(I + G(s)F(s))^{-1} \quad (\text{Eq. 8.16})$$

on:

- $F(s) = \text{diag}[f_i(s)]_{i=1,2,..m}$  és una matriu de funcions de transferència els elements de la diagonal de la qual són els  $m$ -controladors SISO dissenyats per estabilitzar independentment cada un dels llaços independents.
- $\tilde{H}(s)$  i  $H(s)$  són les matrius de funcions de transferència en llaç tancat dels sistemes realimentats formats per  $F(s)$  i  $\tilde{G}(s)$ , i  $F(s)$  i  $G(s)$  respectivament.





Llavors, el sistema en llaç tancat  $H(s)$  és estable si:

- Condició 1:  $G(s)$  i  $\tilde{G}(s)$  tenen el mateix nombre de pols en el semiplà dret
- Condició 2:  $\tilde{H}(s)$  és estable, és a dir, cada un dels subsistemes és estable amb els controladors dissenyats
- Condició 3:  $|h_i(j\omega)| < \mu^{-1}(E(j\omega)) \forall \omega (i = 1, 2 \dots m)$ , on  $\mu$  és el valor singular estructurat o ssv.

Per tant, el procediment per trobar els controladors que estabilitzin el sistema acoblat del robot consistirà en dissenyar per a cada articulació un controlador SISO que l'estabilitzi a partir del l'algorisme de control explicat en el Capítol 7 i després comprovar que amb aquests controladors el sistema global també és estable aplicant les tres condicions anteriors.

### 8.4.3. Aplicació de l'Algorisme

Amb les funcions d'incertesa trobades en l'apartat anterior i usant l'algorisme explicat en l'apartat 7.3 es pot trobar ja les regions de valors de  $K_i$ ,  $K_d$  i  $K_p$  que estabilitzen cada una de les tres articulacions del robot.

#### Articulació 1

Com es pot observar en l'equació (Eq.8.1), la funció de transferència per a la articulació 1 és:

$$G_{11} = \frac{6.933}{s^2} \quad (\text{Eq. 8.17})$$

Si s'aplica l'algorisme sense funció d'incertesa, s'obté la regió mostrada a continuació:

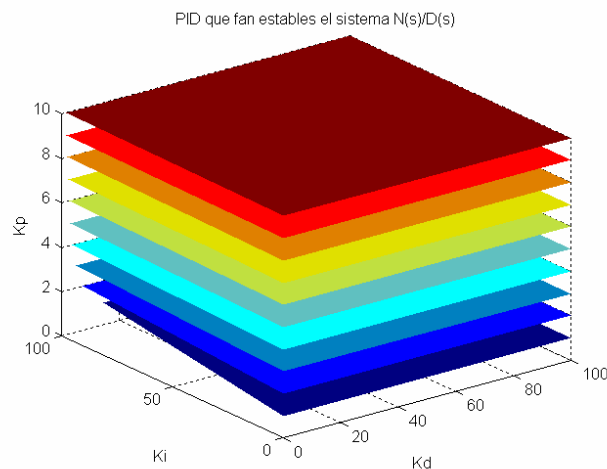


Figura 8.7:Regions que estabilitzen la primera articulació sense funció d'incertesa



D'aquest gràfic s'obtenen els rangs seguidament mostrats per als valors del controlador:

$$K_p > 0 \quad K_i > 0 \quad K_d > 0$$

Si s'utilitza la funció d'incertesa trobada amb anterioritat (Eq.8.6)  $W_1(s) = \frac{0.1s + 1}{0.025s + 1}$  s'obté la següent regió amb l'algorisme amb funció d'incertesa:

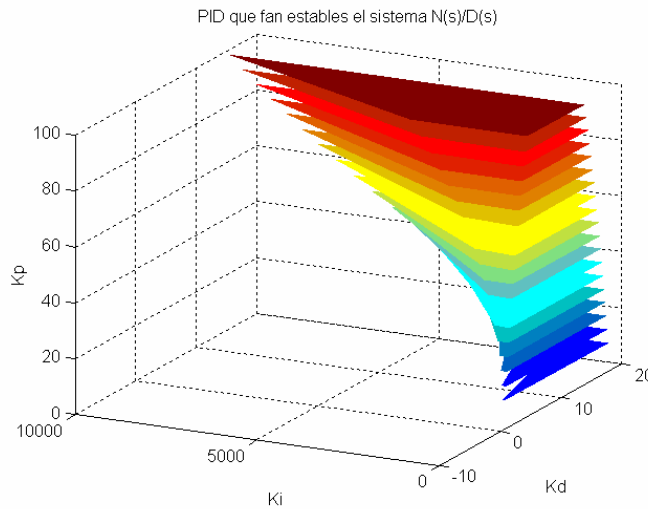


Figura 8.8:Regions que estableixen la primera articulació amb funció d'incertesa

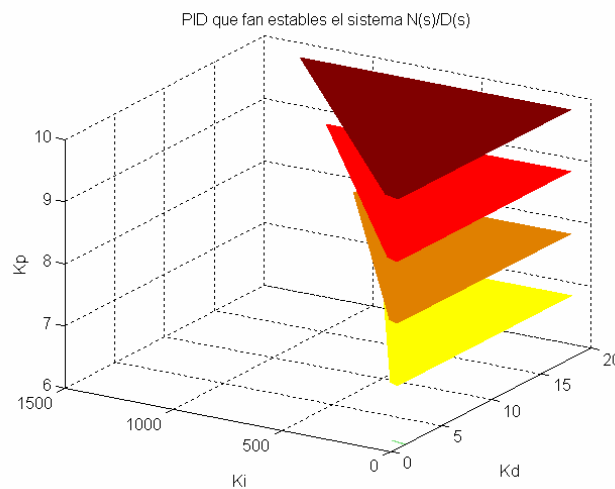


Figura 8.9:Detall de les regions que estableixen l'articulació amb incertesa

## Articulació 2

La funció de transferència per a la articulació 2 és:

$$G_{22} = \frac{0.2943}{s^2} \tag{Eq. 8.18}$$



Per a la segona articulació s'ha obtingut la funció d'incertesa  $W_2(s)=0$ , és a dir, no hi ha funció d'incertesa. Si es realitza l'algoritme s'obté la regió mostrada a continuació:

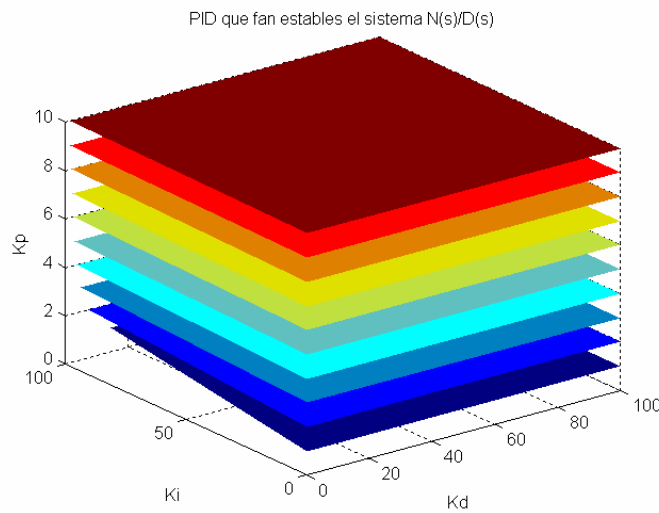


Figura 8.10:Regions que estabilitzen la segona articulació

S'obté per tant, al igual que en el primer cas, les següents limitacions:

$$Kp > 0 \quad Ki > 0 \quad Kd > 0$$

### Articulació 3

La funció de transferència per a la articulació 3 és (apartat 6.5.2):

$$G_{33} = \frac{1.475}{s^2} \quad (\text{Eq. 8.19})$$

Si es realitza l'algoritme sense funció d'incertesa, s'obté la regió mostrada a continuació:

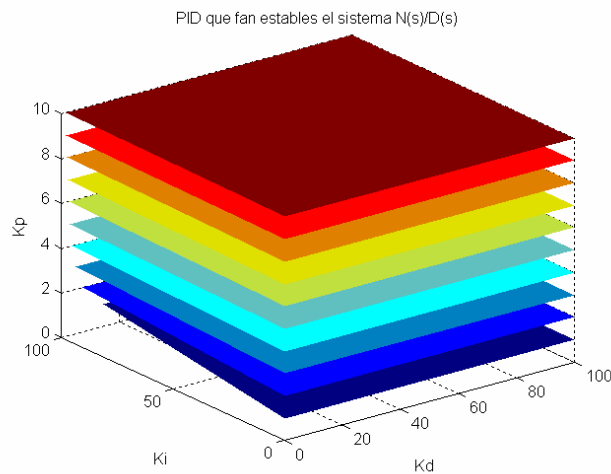


Figura 8.11:Regions que estabilitzen la tercera articulació



D'aquest gràfic s'obtenen els rangs seguidament mostrats per als valors del controlador:

$$K_p > 0 \quad K_i > 0 \quad K_d > 0$$

Si s'utilitza la funció d'incertesa trobada (Eq.8.10),  $W_3(s) = \frac{0.1s + 1}{0.01s + 1}$ , el resultat obtingut és el mostrat a continuació:

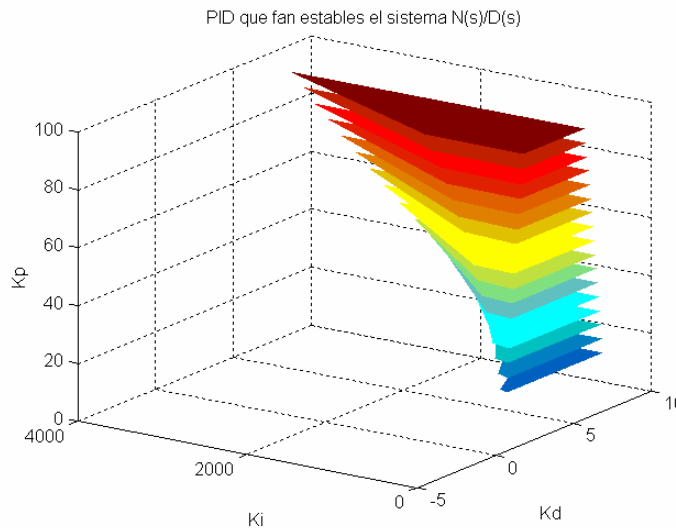


Figura 8.12:Regions que estabilitzen l'articulació 3 tenint en compte l'incertesa

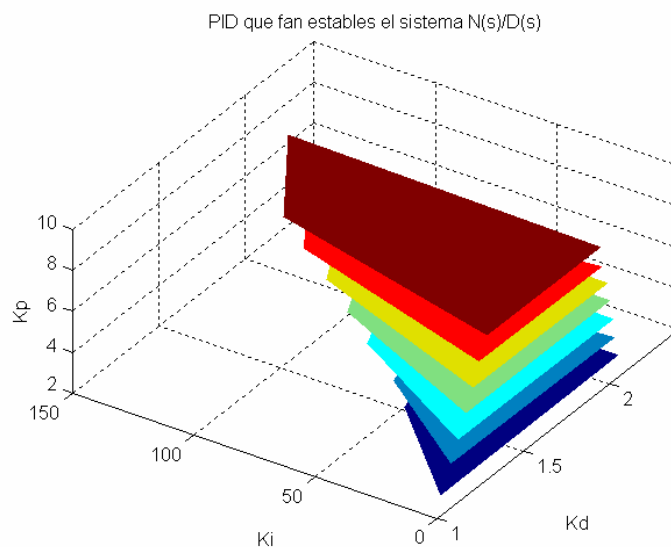


Figura 8.13:Detall de les regions que estabilitzen l'articulació 3 amb incertesa

### 8.5. Selecció de les constants del PID

A partir de l'Algorisme s'han determinat les regions de  $K_p$ ,  $K_i$  i  $K_d$  que fan que el sistema sigui estable. A continuació es seleccionaran uns valors d'aquestes constants per tal que el sistema es comporti de la manera desitjada, és a dir, que compleixi un seguit d'especificacions.



Es tindran en compte bàsicament dos aspectes a l'hora de seleccionar les constants del PID: l'*overshoot* o sobrepuig màxim i el temps d'establiment. El sobrepuig màxim d'un sistema es defineix com la quantitat màxima en què la sortida sobrepassa el seu valor final, mentre que el temps d'establiment es defineix com el temps necessari per poder assegurar que la sortida es manté dins d'una certa franja a l'entorn del seu valor final. Es prendrà com a franja un 2% del valor final [Ricard Villà, 1999].

En la següent gràfica es pot veure el significat d'aquests conceptes:

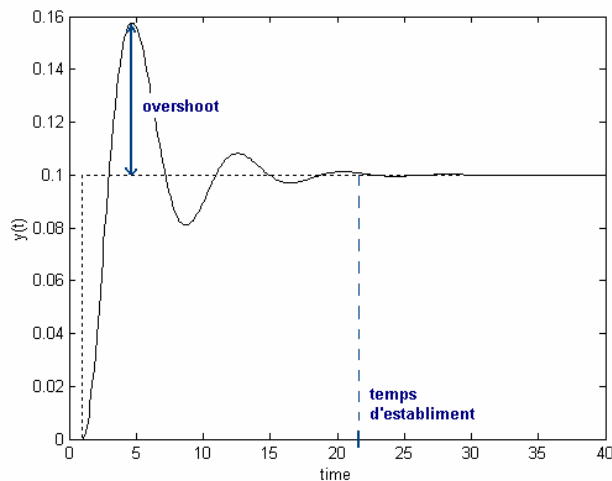


Figura 8.14: Representació de l'*overshoot* i el temps de pic

En el cas de l'aplicació de laparoscòpia, es vol que el sobrepuig màxim sigui inferior al 20%, mentre que el temps d'establiment hauria de ser de com a molt de 5 segons.

És per això que per la selecció dels valors es farà per experimentació, simulant el sistema en llaç tancat i veient com evolucionen el sobrepuig i el temps d'establiment.

La simulació es farà mitjançant la següent funció creada per MATLAB:

*pic\_establiment(kp, kimin, kimax, kadmin, kdmax, sys, n)*

que fa  $n$ -simulacions del sistema 'sys' per a una entrada graó unitari, amb un valor de  $k_p$  determinat i entre els intervals de valors  $K_i \in (k_{i\min}, k_{i\max})$ ,  $K_d \in (k_{d\min}, k_{d\max})$ , retornant dos gràfiques, una per l'*overshoot* i l'altre pel temps d'establiment.

A continuació es mostren els resultats per la primera articulació (Eq.8.11):



### 8.5.1. Overshoot (sobreuig màxim)

$K_p=0.1$

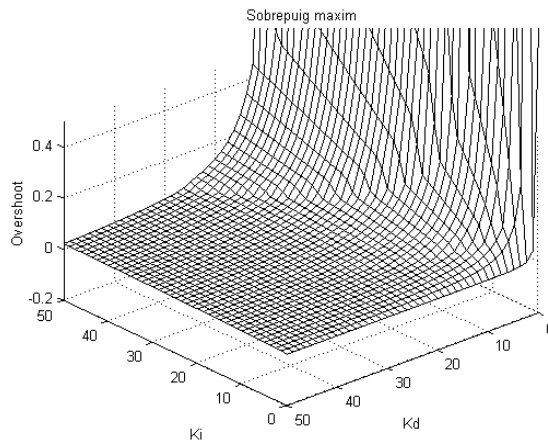


Figura 8.15: Sobreuig per a  $K_p=0.1$

$K_p=1$

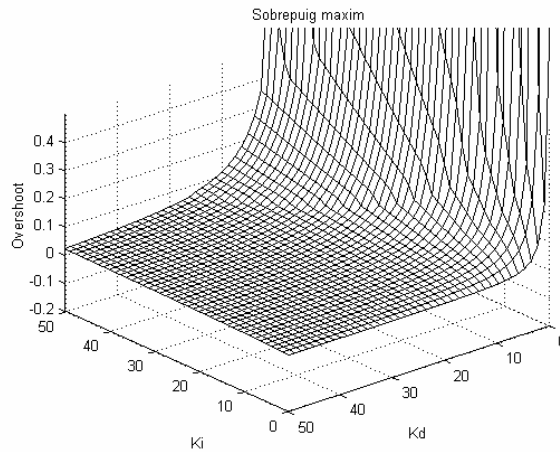


Figura 8.16: Sobreuig per a  $K_p=1$

$K_p=10$

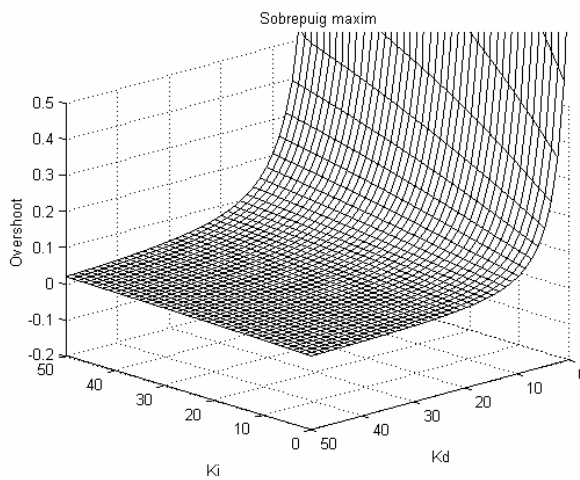


Figura 8.17: Sobreuig per a  $K_p=10$



$$K_p=100$$

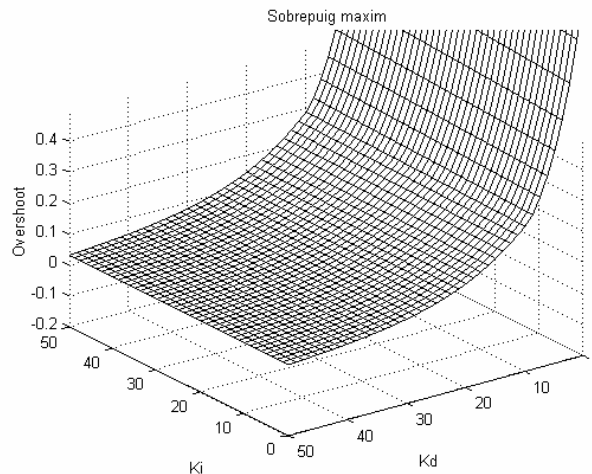


Figura 8.18: Sobrepuig per a  $K_p=100$

Si s'observen les gràfiques anteriors, és fàcil deduir que:

- pel que fa a la constant  $K_p$ , a mesura que s'augmenta el seu valor el sobrepuig va augmentant.
- es veu també que si s'augmenta el valor de  $K_i$  l'overshoot de la resposta creix lleugerament
- per últim, es pot observar que quan el valor de  $K_d$  s'aproxima a zero el sobrepuig del sistema creix exponencialment, mentre que a mesura que s'augmenta el seu valor el sobrepuig disminueix.

Per tant, si només es tingués en compte com a especificació el sobrepuig màxim, s'hauria d'escollir els valors de les constants del PID de la següent forma:

- la  $K_p$  hauria de ser la mínima possible.
- la  $K_i$  hauria de ser com més petita millor.
- la  $K_d$  hauria de ser el més gran possible.



### 8.5.2. Temps d'establiment

$K_p=0.1$

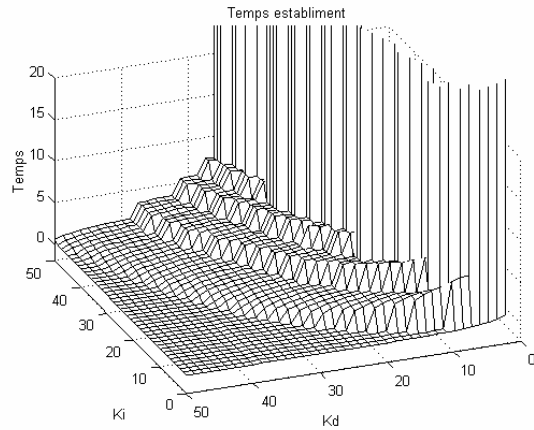


Figura 8.19: Temps d'establiment per a  $K_p=0.1$

$K_p=1$

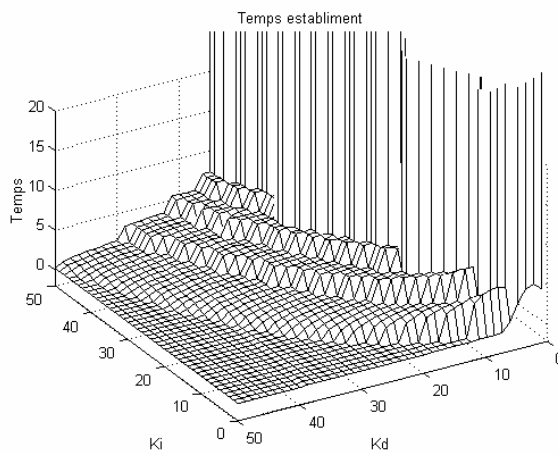


Figura 8.20: Temps d'establiment per a  $K_p=1$

$K_p=10$

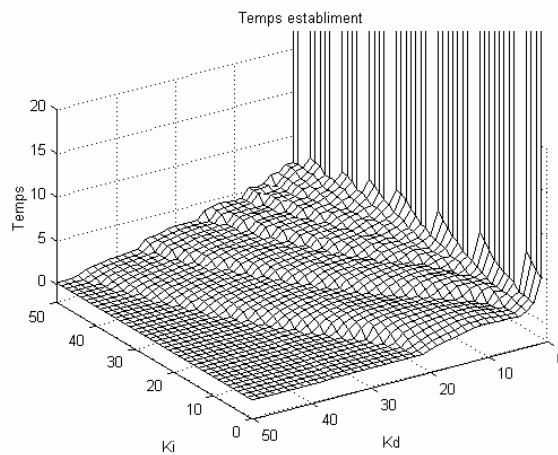


Figura 8.21: Temps d'establiment per a  $K_p=10$





$$K_p=100$$

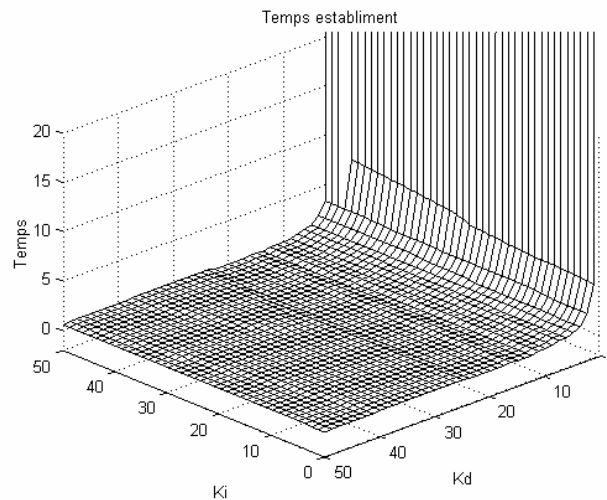


Figura 8.22: Temps d'establiment per a  $K_p=100$

Observant les gràfiques anteriors es poden extreure les següents conclusions:

- pel que fa a la constant  $K_p$ , a mesura que s'augmenta el seu valor el temps d'establiment va disminuint.
- es veu també que si s'augmenta el valor de  $K_i$  el temps d'assentament creix lleugerament
- per últim, s'observa clarament que quan el valor de  $K_d$  s'aproxima a zero el temps d'establiment del sistema creix, mentre que a mesura que s'augmenta el seu valor disminueix.

Per tant, si només es tinguéssim en compte com a especificació el sobrepuig màxim, s'hauria d'escollir els valors de les constants del PID de la següent forma:

- la  $K_p$  hauria de ser la màxima possible.
- la  $K_i$  hauria de ser com més petita millor.
- la  $K_d$  hauria de ser el més gran possible.

De forma anàloga es podria repetir el procés per a les altres dues articulacions, però donat que en el model simplificat tan sols varia el guany de les tres funcions de transferència, els fenòmens que es produiran seran els mateixos que per la primera articulació i les lleis per escollir els valors de les constants dels altres dos controladors seran les mateixes que les comentades anteriorment.

A més, també s'ha de tenir en compte que el senyal de control (és a dir, la senyal que entra en el robot) no pot sobrepassar uns certs límits, donat que la tensió que han de rebre les articulacions del robot real han d'estar compreses entre -6V i 6V.

Per tant, un cop seleccionats els valors adients per aconseguir l'*overshoot* i el temps d'establiment desitjats s'haurà de comprovar que amb aquests valors la senyal de control estigui acotada entre -6 i 6V. Com a norma general, com més gran és la  $K_p$  més gran tendeix a ser el senyal de control i per això s'intentarà agafar la  $K_p$  més petita possible.



### 8.5.3. Valors elegits

Tenint en compte totes les consideracions anteriors, i provant-ho sobre el sistema real, s'han elegit els següents controladors per a cada una de les articulacions:

- Articulació 1

$$\begin{aligned} K_p &= 10 \\ K_i &= 10 \\ K_d &= 2 \end{aligned}$$

- Articulació 2

$$\begin{aligned} K_p &= 10 \\ K_i &= 0 \\ K_d &= 20 \end{aligned}$$

- Articulació 3

$$\begin{aligned} K_p &= 10 \\ K_i &= 10 \\ K_d &= 2 \end{aligned}$$

### 8.6. Comprovació de l'estabilitat del sistema global i de l'estabilitat robusta

Com s'ha comentat en l'apartat 8.4.2, s'ha procedit a dissenyar tres PID's com si el sistema fossin tres llaços de control independents, però com també s'ha vist existeix un acoblament entre les articulacions, i per tant de cara a assegurar l'estabilitat del sistema global amb aquests tres controladors SISO s'ha de comprovar el compliment de tres condicions:

- Condició 1:  $G(s)$  i  $\tilde{G}(s)$  tenen el mateix nombre de pols en el semiplà dret
- Condició 2:  $\tilde{H}(s)$  és estable, és a dir, cada un dels subsistemes és estable amb els controladors dissenyats
- Condició 3:  $|h_i(j\omega)| < \mu^{-1}(E(j\omega)) \forall \omega (i = 1, 2, \dots, m)$ , on  $\mu$  és el valor singular estructurat o ssv.

Tenint en compte els conceptes explicats en **¡Error! No se encuentra el origen de la referencia.**, a continuació es veurà si es compleixen o no les condicions:

- Condició 1:

Es compleix donat que

$$G(s) = \begin{pmatrix} \frac{6.933}{s^2} & 0 & \frac{2.114}{s^2} \\ 0 & \frac{0.2943}{s^2} & 0 \\ \frac{2.114}{s^2} & 0 & \frac{1.475}{s^2} \end{pmatrix} \text{ i } \tilde{G}(s) = \begin{pmatrix} \frac{6.933}{s^2} & 0 & 0 \\ 0 & \frac{0.2943}{s^2} & 0 \\ 0 & 0 & \frac{1.475}{s^2} \end{pmatrix}$$

no tenen cap pol en el semiplà dret.

- Condició 2:



També es compleix donat que els tres PID's s'han dissenyat de tal forma que estableixen cada articulació per separat.

- Condició 3:

$|h_i(j\omega)| < \mu^{-1}(E(j\omega)) \forall \omega (i = 1, 2, \dots, m)$ , on  $\mu$  és el valor singular estructurat o ssv.

Es tenia que  $E(s) = \hat{G}(s)\tilde{G}(s)^{-1}$ , per tant  $E(s) = \begin{pmatrix} 0 & 0 & 0.3049 \\ 0 & 0 & 0 \\ 0.3049 & 0 & 0 \end{pmatrix}$

Llavors, amb l'ajuda de la funció de MATLAB 'ssv' es troba que

$$\mu^{-1}(E) = \frac{1}{0.663} = 1.5083 \forall \omega$$

i els diagrames de Bode de la matriu  $\tilde{H}$  amb els seus corresponents valors màxims són:

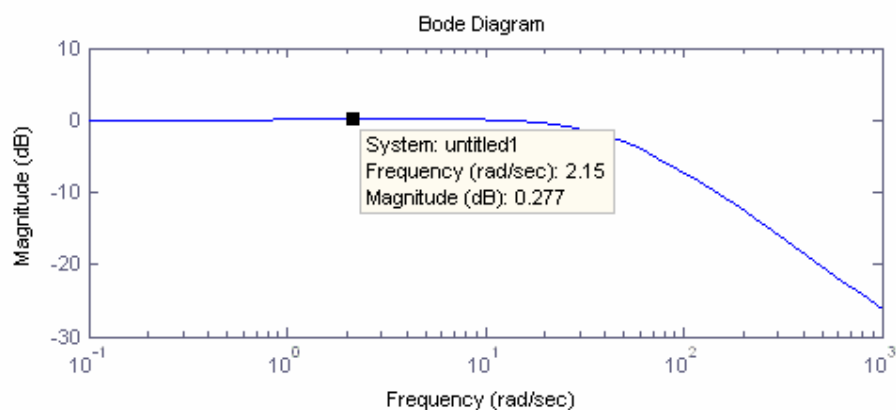


Figura 8.23: Diagrama de Bode de  $h_1$

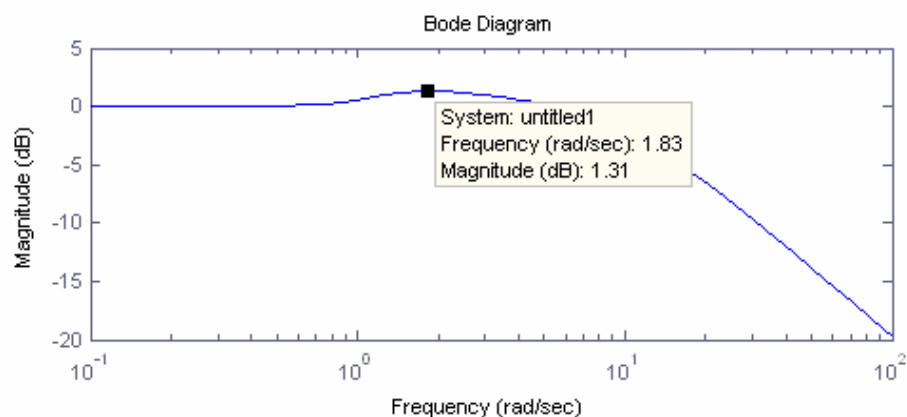


Figura 8.24: Diagrama de Bode de  $h_2$



Per tant, la condició 3 també es compleix, i es pot assegurar que el sistema global en llaç tancat amb els controladors dissenyats serà estable.

Per a comprovar l'estabilitat robusta de cadascun dels controls, s'ha de complir la condició  $\|WT\|_{\infty} < 1$  a cadascuna de les articulacions.

Això es verifica observant la figura següent:

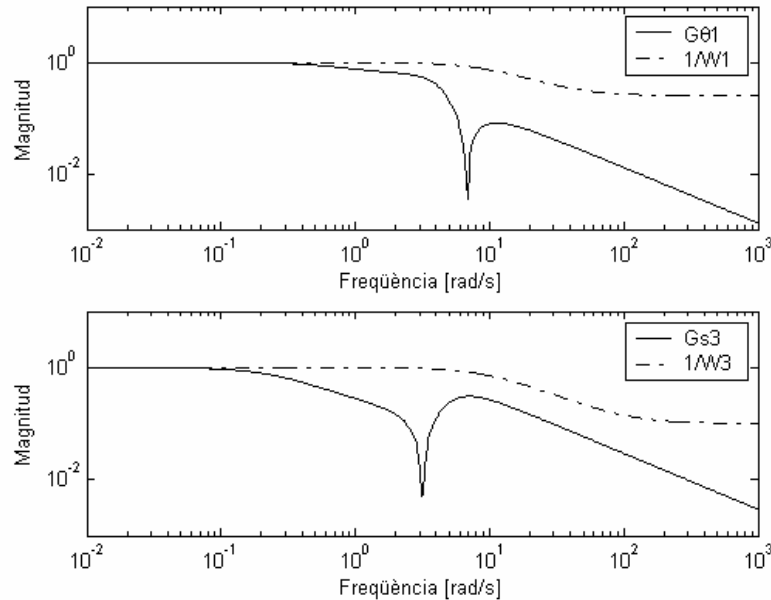


Figura 8.25: Comprovació de l'estabilitat robusta

### 8.7. Simulació del llaç tancat

A continuació es realitzaran un seguit de simulacions del sistema complet en llaç tancat a partir dels blocs creats en SIMULINK que es poden consultar en l'Apèndix A. Aquests blocs representen el Controlador, el Model Cinemàtic Directe, el Model Cinemàtic Invers, el Robot, l'Algorisme de Visió i la Transformació 2D->3D.

L'esquema de control a partir dels blocs és:

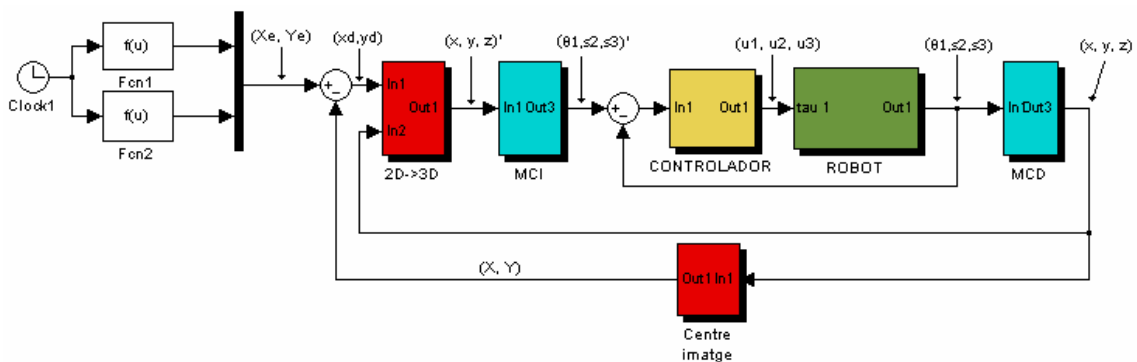


Figura 8.26: Esquema complet de control



En aquest esquema es poden observar els blocs Fcn1 i Fcn2, que representen les coordenades del centre de les eines en funció del temps, i tots els blocs anomenats anteriorment.

També s'observen les senyals següents:

- el senyal (Xe, Ye) que representa la trajectòria del centre de les eines en funció del temps
- el senyal (Xd, Yd) que és el senyal d'error que retornaria l'algorisme de visió
- posteriorment, després de passar pel bloc 2D->3D apareix la senyal (x, y, z)' que representa la posició a on hauria d'estar el TCP per fer l'error de l'algorisme de visió zero.
- a continuació després del bloc MCI s'obtenen els valors corresponents a (x, y, z)' de les variables articulars ( $\theta_1, s_2, s_3$ ).
- el valor de la senyal de control ( $u_1, u_2, u_3$ ).
- aquesta senyal arriba al model del robot i a la sortida es tenen les variables articulars actuals ( $\theta_1, s_2, s_3$ ).
- al passar pel MCD s'obtenen els corresponents valors en variables cartesianes del TCP (x, y, z).

### 8.7.1. Simulació 1

Primer de tot es suposarà que el punt mig de les eines es mou seguint una trajectòria circular dins el pla de visió de radi 0.1 [m]. L'entrada o senyal de consigna serà doncs:

$$\begin{aligned} X_e &= 0.1 \cdot \cos(t/10) \\ Y_e &= 0.1 \cdot \sin(t/10) \end{aligned}$$

Si s'executa la simulació, es poden extreure els següents resultats:

#### Comparació dels senyals (x, y, z)' i (x, y, z) de la trajectòria del TCP

És a dir, s'està comparant la trajectòria que hauria de seguir el TCP amb la que realment segueix:

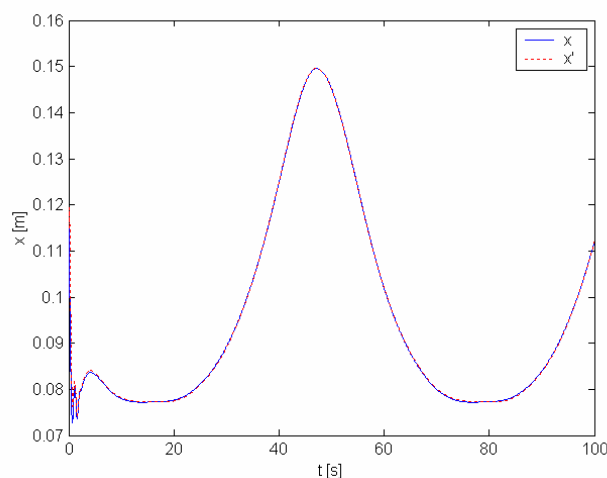


Figura 8.27: Comparació dels senyals  $x'$  i  $x$



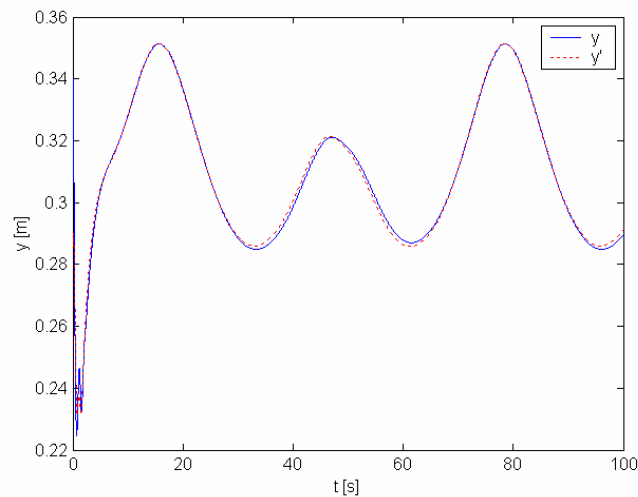


Figura 8.28: Comparació dels senyals  $y'$  i  $y$

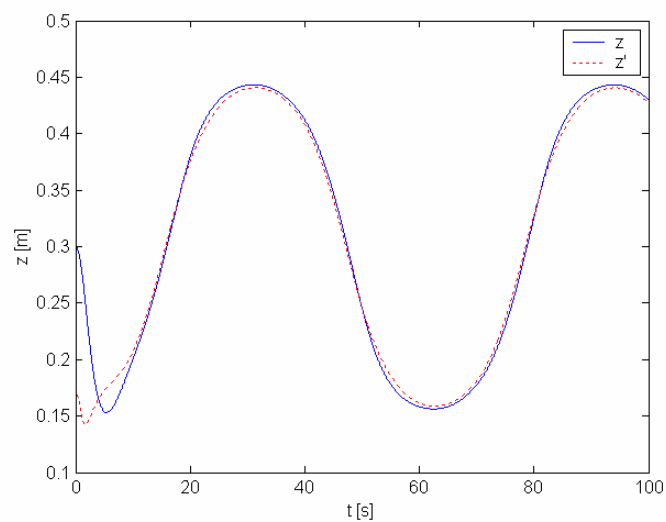


Figura 8.29: Comparació de les senyals  $z'$  i  $z$

S'observa que la càmera tarda uns 4 [s] aproximadament en començar a seguir la consigna amb error zero.

### Comparació dels senyals $(\theta_1, s_2, s_3)'$ i $(\theta_1, s_2, s_3)$ variables articulars

És a dir, s'està comparant les variables articulars que retorna el robot amb les variables articulars de consigna:



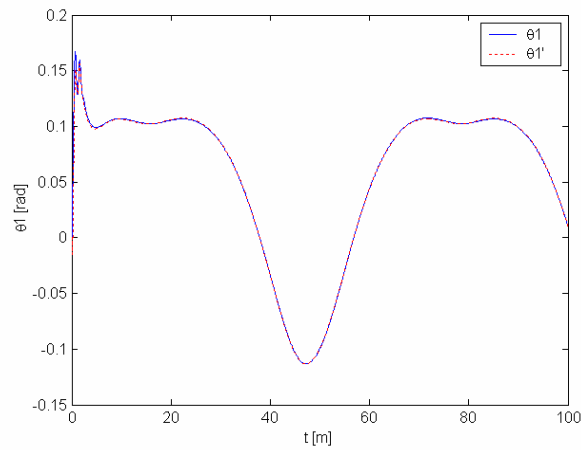


Figura 8.30: Comparació de les senyals  $\theta_1'$  i  $\theta_1$

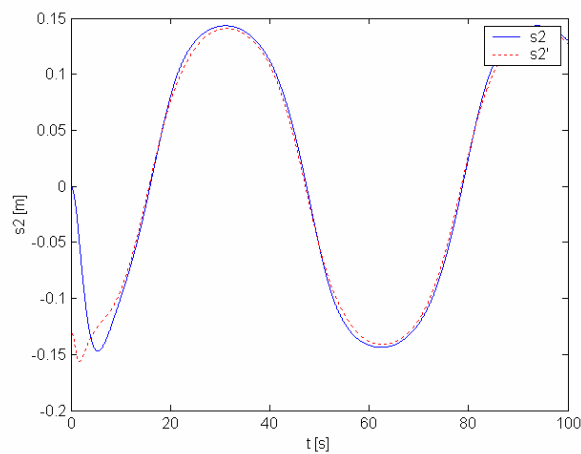


Figura 8.31: Comparació de les senyals  $s_2'$  i  $s_2$

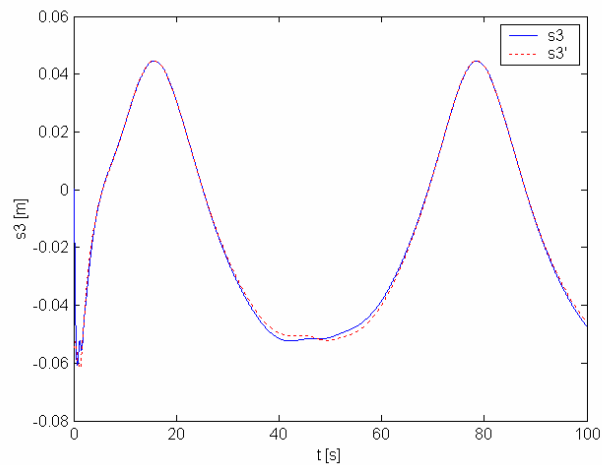


Figura 8.32: Comparació de les senyals  $s_3'$  i  $s_3$

### Comparació de les senyal ( $X_e, Y_e$ ) i ( $X, Y$ )

Amb això el que es compara és la trajectòria que segueix l'eina dins el pla de visió amb el punt on està enfocant la càmera.



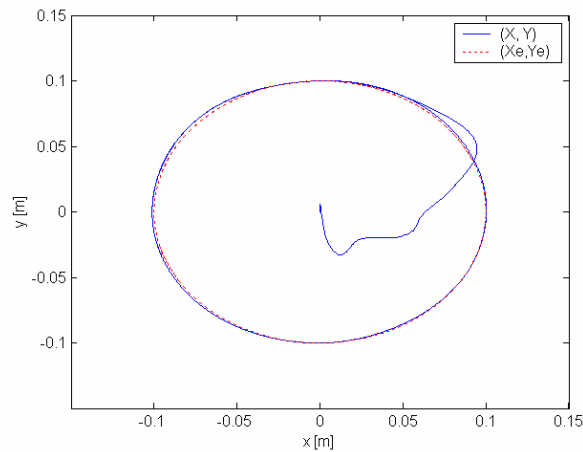


Figura 8.33: Comparació dels senyals  $(X_e, Y_e)$  i  $(X, Y)$

Es pot veure com l'eina, que comença des de la posició  $(0,0)$ , va a buscar la trajectòria circular fins a seguir-la correctament.

### Senyal retornat per l'algorisme de visió ( $X_d, Y_d$ )

A continuació es mostra l'evolució en el temps de la senyal retornada per l'algorisme de visió, que representa l'error existent i la posició de l'eina.

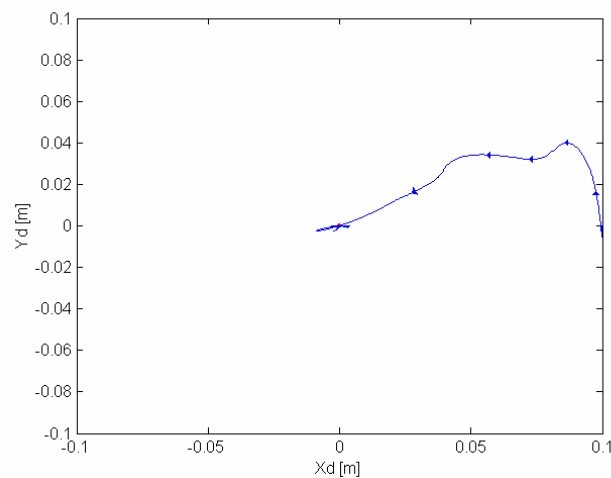


Figura 8.34: Senyal d'error

A l'inici la senyal d'error és gran perquè la càmera està apuntant al centre  $(0,0)$  i ha d'anar a buscar la trajectòria circular. Després però l'error es va reduint fins que al final la senyal d'error és pràcticament zero.

Per últim, es representa la trajectòria del TCP en l'espai per comprovar que sempre es troba a la mateixa distància del punt de pivotament de la càmera  $(a,b,c)$ .





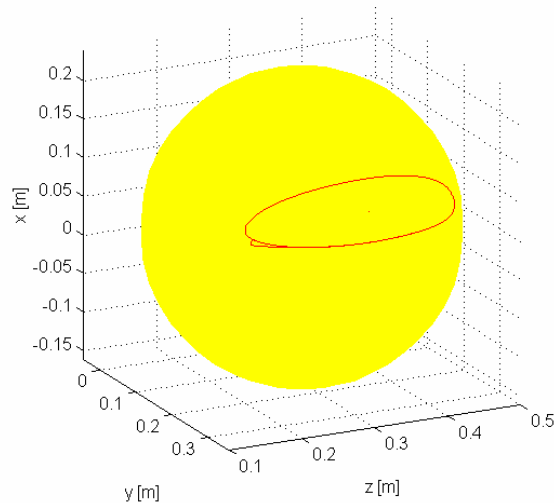


Figura 8.35: Trajectòria en l'espai

S'observa que el TCP sempre es troba a sobre de l'esfera, i per tant es troba sempre a una distància  $l_2$  del punt de pivotament.

### 8.7.2. Simulació 2

Ara es simularà la mateixa trajectòria circular que en la Simulació 1 però augmentant-ne la velocitat:

$$X_e = 0.1 * \cos(t/5)$$

$$Y_e = 0.1 * \sin(t/5)$$

#### Comparació de les senyals $(x, y, z)$ i $(x, y, z)$ '

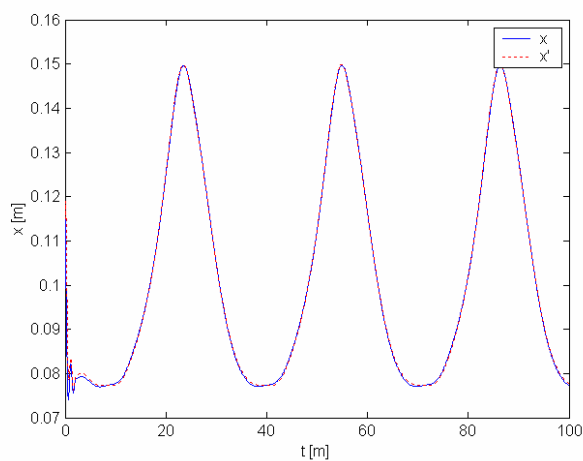


Figura 8.36. Comparació dels senyals  $x'$  i  $x$



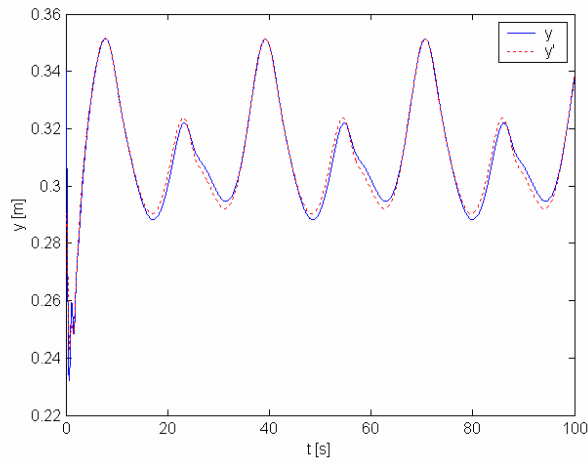


Figura 8.37. Comparació de les senyals  $y'$  i  $y$

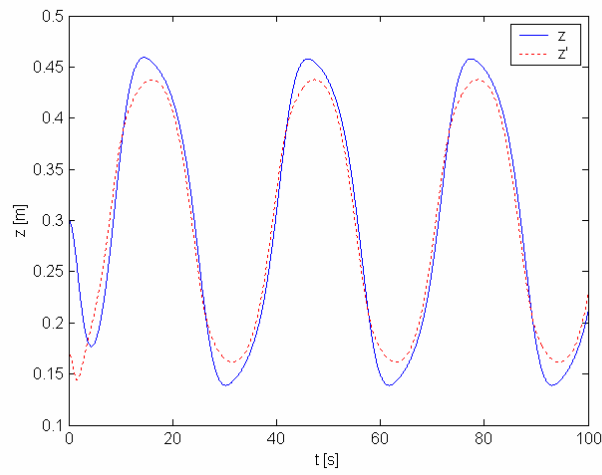


Figura 8.38: Comparació de les senyals  $z'$  i  $z$

**Comparació dels senyals  $(\theta_1, s_2, s_3)'$  i  $(\theta_1, s_2, s_3)$  variables articulars**

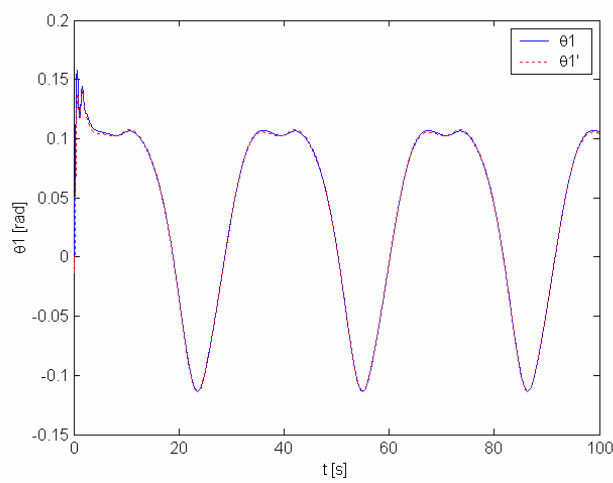


Figura 8.39: Comparació de les senyals  $\theta_1'$  i  $\theta_1$



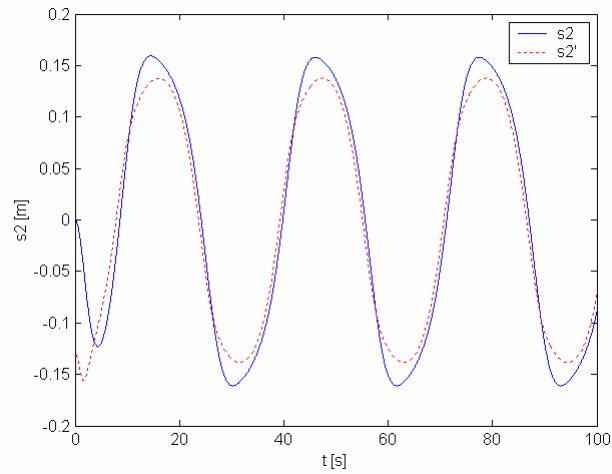


Figura 8.40: Comparació de les senyals  $s_2'$  i  $s_2$

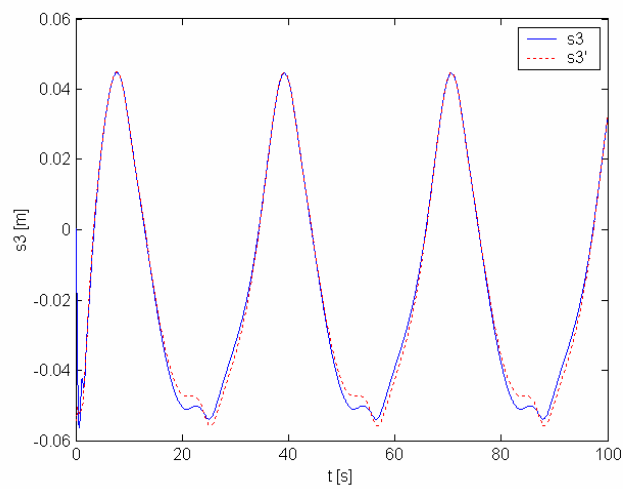


Figura 8.41: Comparació de les senyals  $s_3'$  i  $s_3$

### Comparació de les senyal $(X_e, Y_e)$ i $(X, Y)$

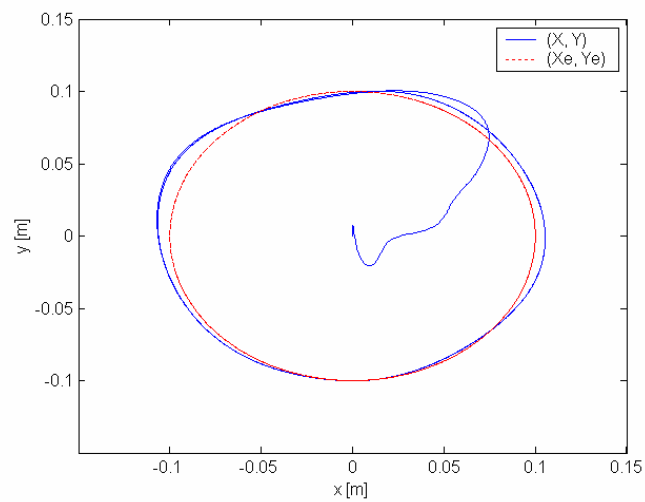


Figura 8.42: Comparació de les senyal  $(X_e, Y_e)$  i  $(X, Y)$



### Senyal retornat per l'algorisme de visió ( Xd, Yd )

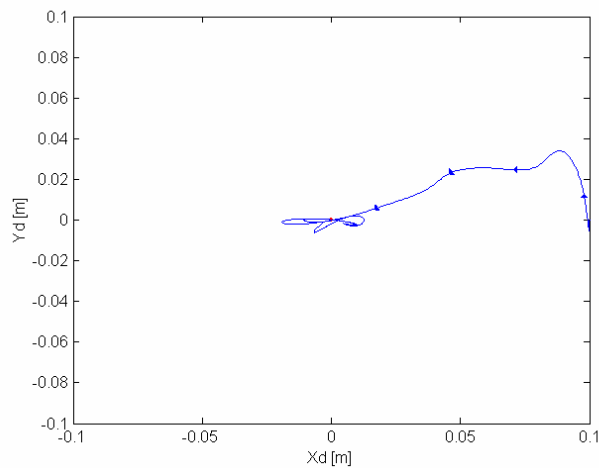


Figura 8.43: Senyal d'error

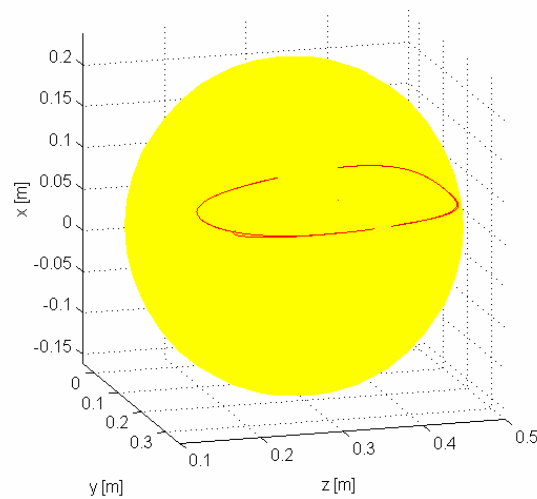


Figura 8.44: Trajectòria en l'espai

Si es comparen aquestes gràfiques amb les de la Simulació 1 s'observa que la resposta és la mateixa però els errors són majors, degut a que al augmentar la velocitat li costa més seguir la trajectòria.

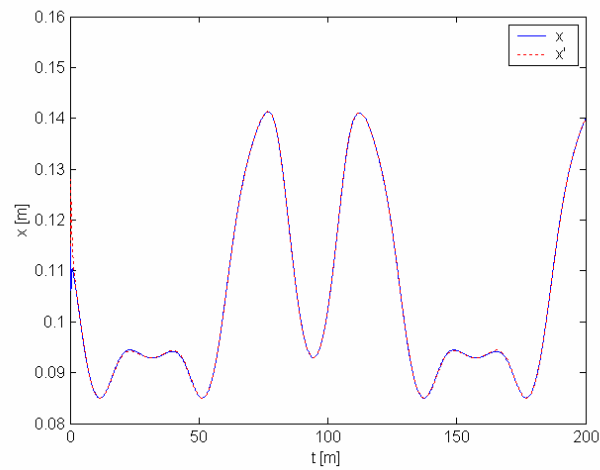
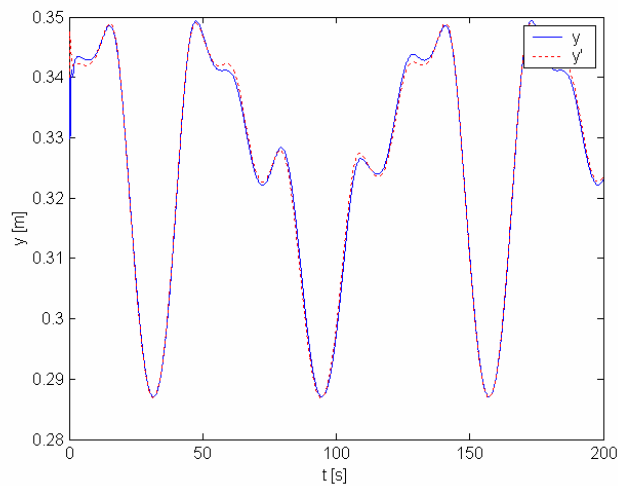
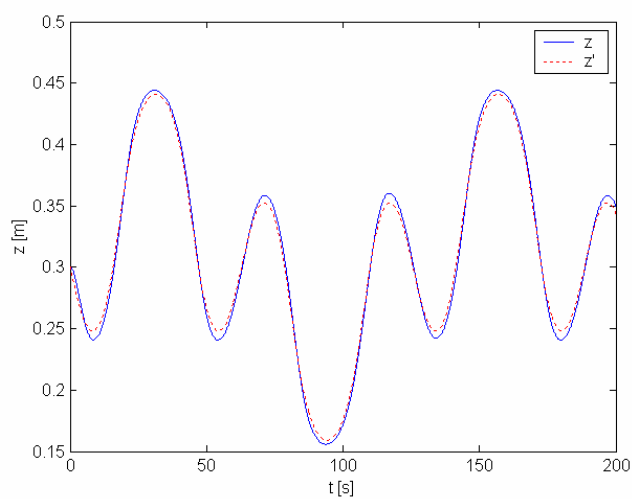
### 8.7.3. Simulació 3

Per últim, es simularà un moviment arbitrari de les eines:

$$X_e = 0.1 \cdot \cos(u/10) \cdot \sin(t/20)$$

$$Y_e = 0.1 \cdot \sin(u/10) \cdot \cos(t/20)$$



**Comparació de les senyals  $(x, y, z)$  i  $(x', y', z')$** *Figura 8.45: Comparació de les senyals  $x'$  i  $x$* *Figura 8.46: Comparació de les senyals  $y'$  i  $y$* *Figura 8.47: Comparació de les senyals  $z'$  i  $z$* 

### Comparació dels senyals $(\theta_1, s_2, s_3)'$ i $(\theta_1, s_2, s_3)$ variables articulars

És a dir, s'està comparant les variables articulars que retorna el robot amb les variables articulars de consigna:

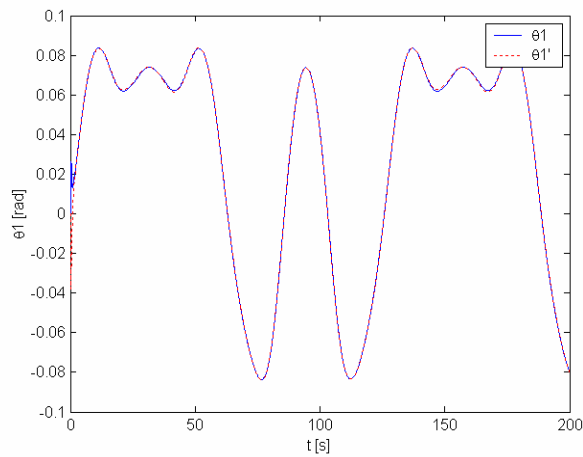


Figura 8.48: Comparació de les senyals  $\theta_1'$  i  $\theta_1$

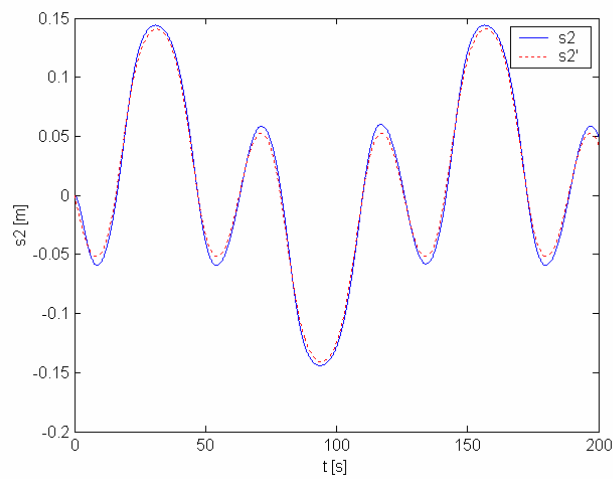


Figura 8.49: Comparació de les senyals  $s_2'$  i  $s_2$

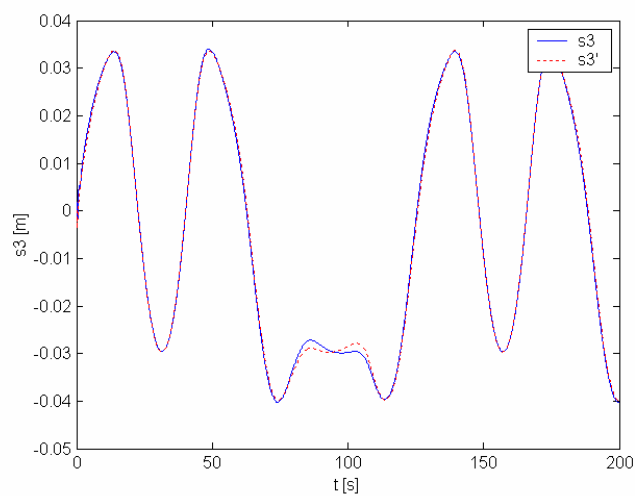
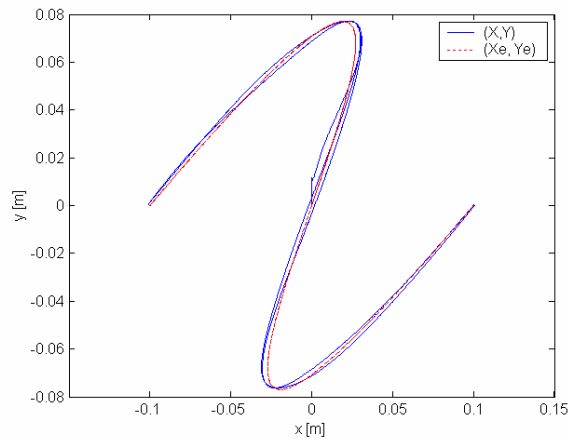
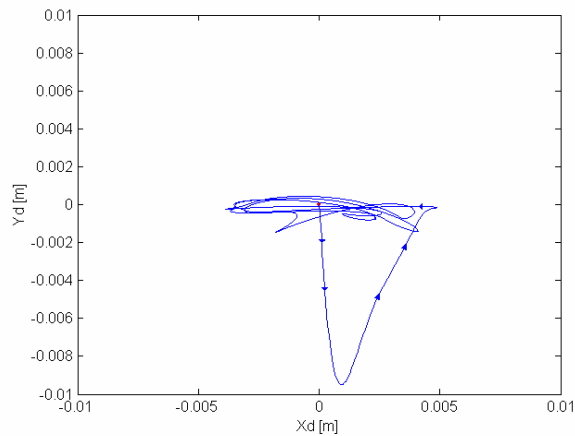
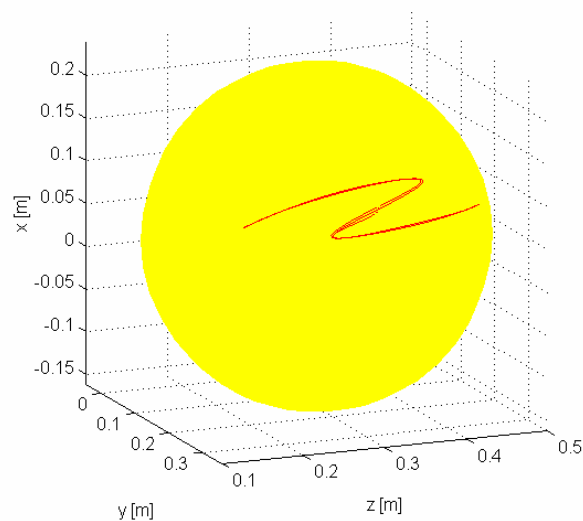


Figura 8.50: Comparació de les senyals  $s_3'$  i  $s_3$



**Comparació dels senyals  $(X_e, Y_e)$  i  $(X, Y)$** *Figura 8.51. Comparació dels senyals  $(X_e, Y_e)$  i  $(X, Y)$* **Senyal retornat per l'algorisme de visió  $(X_d, Y_d)$** *Figura 8.52. Senyal d'error**Figura 8.53: Trajectòria en l'espai*

## 8.8. Resultats de les simulacions

En les gràfiques mostrades anteriorment, es pot apreciar el funcionament correcte del sistema i el compliment de les especificacions requerides. Primerament es comparen els senyals de consigna  $(x, y, z)$  amb la resposta simulada del robot  $(x, y, z)$ , es pot apreciar com a partir dels 4 segons el sistema es troba controlat i comença a seguir el senyal de consigna amb molta precisió, en aquests quatre primers segons el robot cau i ràpidament es controlat pels actuadors i comença a seguir el senyal de consigna.

Si s'augmenta la velocitat, en la simulació 2, s'observa com l'error en alguns punts augmenta però segueix correctament la trajectòria de les eines. El mateix es pot apreciar en les gràfiques que mostren els senyals de les variables articulars, la consigna  $(\theta_1, s_2, s_3)$  i la resposta simulada del robot  $(\theta_1, s_2, s_3)$ , on es constaten els mateixos resultats.

Posteriorment, es comparen els senyals  $(X, Y)$  amb  $(X_e, Y_e)$  que ens mostren el punt corresponent al centre de la imatge i la trajectòria realitzada per les eines sobre el pla fix. En la gràfica que representa l'anterior, es pot observar com el centre de la imatge segueix la trajectòria de les eines amb un petit error. L'error entre ambdues senyals, mostrat a la gràfica següent, correspon amb el senyal retornat per l'algorisme de visió  $(X_d, Y_d)$ . En ell s'aprecia com aquest error va disminuint fins a quedar-se dins d'una bola de radi 0.01 metres al voltant del punt  $(0,0)$ .

La última gràfica mostra la trajectòria corresponent al TCP  $(x, y, z)$  representada en 3D. Això permet comprovar com el TCP es situa sempre a sobre d'una l'esfera de radi  $l_2=20$  centímetres i centrada en el punt de pivotament  $(a, b, c)$ . Per tant es comprova que la trajectòria del TCP sempre es situa a la mateixa distància del punt de pivotament i la canya de la càmera no llisca en aquest punt, per garantir la seguretat del pacient.





## Conclusions

Amb aquest estudi teòric s'ha demostrat que el sistema utilitzat és útil per a realitzar la tasca encomanada, i que la utilització d'un controlador PID és suficient per a realitzar el control i garantir les especificacions.

En aquest projecte s'han desenvolupat els següents estudis:

- Estudi de les operacions en regions abdominals, amb la tècnica de la laparoscòpia. Tècnica no agressiva pel pacient, però que comporta una dificultat afegida pel cirurgià. Per intentar millorar els inconvenients d'aquesta operació, es va desenvolupar un robot amb una sèrie de graus de llibertat adequats per a facilitar el moviment de la càmera, de forma senzilla, i millorar l'espai de treball del cirurgià.
- Estudi de la cinemàtica i la dinàmica del robot especialment dissenyat per a realitzar aquesta operació. Per fer-ho consta de tres graus de llibertat, una articulació de rotació i dos de translació, encarregades de moure la càmera, que donarà una visió de l'espai de treball del cirurgià, mostrant en tot moment les eines en pantalla.
- Simplificació i linealització dels models cinemàtics i dinàmics del robot per tal de poder treballar amb algorismes de control lineals i analítics i poder deduir el funcionament i les respostes en diferents situacions i punts de treball.
- Estudi dels controladors adequats pel correcte funcionament del robot, que presenta no linealitats i acoblaments entre dues de les articulacions els i diferents punts de treball.
- Desenvolupament de dos algorismes de control basats en una estructura PID, un primer per a sistemes lineals, i un segon tenint en compte funcions d'incertesa per contemplar les no linealitats, acoblaments, i el funcionament del sistema en diferents punts de treball, que es basa en el control robust.
- Aplicació dels algorismes de control en el model simplificat, i estudi dels diferents controladors possibles per tal d'escollir un que funcioni correctament i compleixi les especificacions recomanables en les operacions de laparoscòpia, com són una resposta no necessàriament molt ràpida, inferior a quatre segons, i amb errors acceptables, el centre de les dues eines es troba a no més d'un centímetre del centre de la imatge.
- Estudi teòric del sistema a través de diverses simulacions mitjançant el desenvolupament d'un simulador que corroboren el bon funcionament del robot complint les especificacions recomanades.

Després d'aquest estudi s'ha pogut comprovar el bon funcionament d'un robot dedicat a la recerca en l'àmbit mèdic, i com poc a poc es van fent petits avenços en aquest món. S'espera que aquest estudi sigui un principi per a altres estudis posteriors, que continguin altres idees innovadores pel que fa a configuracions dels robots, algorismes de detecció i algorismes de control entre d'altres.



Actualment s'està escrivint un article referent a aquest treball que pròximament s'enviarà a un Congrés de Control i Robòtica.

### **Possibles millores i recomanacions per treballs futurs**

- Caldria millorar l'estructura mecànica del robot de laparoscòpia donat a que el freqüència existent a les articulacions és massa elevada. A més de cara a treballar en un entorn quirúrgic, caldria revisar les normes referents a la seguretat per tal que el robot sigui apte per treballar-hi.
- És necessària la revisió de l'algorisme de visió donat que la detecció de les eines no és l'adequada en tot instant. A més, sorgeix el problema de la comunicació entre les dades obtingudes de l'algorisme de visió i el controlador implantat en MATLAB, donat que la visió està implementada en Visual C++ i la comunicació entre ambdós programes no és fàcil d'aconseguir en temps real. Si es decideix refer l'algorisme de visió, es proposa programar-lo a partir de les llibreries de MATLAB per evitar aquest problema.
- La comunicació entre el PC industrial i el PC sobre el qual s'executa el control (algorisme de visió i controlador) es va aconseguir a l'inici del Projecte, però caldria revisar-la per tal que s'adaptés a les condicions de treball existents un cop s'aconsegueixi tancar el llaç amb l'algorisme de visió.

## **Agraïments**

Volem donar les gràcies especialment a la Yolanda Bolea, la directora d'aquest Projecte, que sempre ha estat en tot moment disposada a ajudar-nos.

Agrair la col·laboració de l'Antoni Grau en les tasques de comunicació i en l'inici del projecte.

La inestimable ajuda dels amics i de les nostres famílies que ens han fet costat en tot moment.



## Bibliografia

### Referències bibliogràfiques

- OLLERO, *Robótica: Manipuladores y robots móviles*, MARCOMBO BOIXAREU EDITORES, 2001.
- AMAT, J. *Brazo robot retráctil*, Patent estatal 982526. Duració:1/10/2001 a 30/9/2001.
- ASTRÖM K.J. i HÄGGLUND T., *PID Controllers: Theory, Design and Tuning*, INSTRUMENT SOCIETY OF AMERICA, 1995.
- BARRIENTOS A., *Fundamentos de Robótica*, Capítulo 4, MCGRAW HILL, 1997.
- HILL, J., PARK, W.T., *Real time control of a robot with a mobile camera.*, Proc. 9<sup>th</sup> ISIR, Washington, 1979, pp.223-246.
- J. MIGUEL IBÁÑEZ, *Disseny d'un robot de quatre graus de llibertat*, Projecte de Fi de Carrera, Desembre 1997.
- L.ACOSTA, M.SIGUT, S.TORRES, A. HAMILTON i J.J. RODRIGO, *A decentralized control strategy for a large-scale system including uncertainties in the dynamic model*, Proceedings of the 41st IEEE, Conference on Decision and Control, December 2002.
- MING-TZU HO, *Synthesis of  $H_{\infty}$  PID controllers: A parametric approach*, Automatica 39 1069-1075, February 2003.
- MING-TZU HO, ANIRUDDHA DATTA i S.P.BHATTACHARYYA, *Structure and synthesis of PID controllers*, SPRINGER, 2000.
- MING-TZU HO, ANIRUDDHA DATTA i S.P.BHATTACHARYYA, *A linear Programming Characterization of All Stabilizing PID Controllers*, Proceedings of the American Control Conference, 1997a.
- MING-TZU HO, ANIRUDDHA DATTA i S.P.BHATTACHARYYA, *Control System Design Using Low Order Controllers: Constant Gain, PI and PID*, Proceedings of the American Control Conference, 1997b.
- P BERKELMAN, P CINQUIN, E BOIDARD, J TROCCAZ, C LÉTOUBLON i J-M AYOUBI, *Design, control and testing of a novel compact laparoscopic endoscope manipulator*, Proc. Instn Mech. Engrs Vol.217 Part I: J. Systems and Control Engineering, 2003.
- RICARD VILLÀ, *Dinàmica de sistemes*, UPC-CPDA, 1999.
- ROBERT J., *Automatic Guidance of a Laparoscope using Computer Vision*, Projecte de Fi de Carrera, Dept. ESAI (ETSEIB), Novembre 2002.
- RODRÍGUEZ RUBIO F., LÓPEZ SÁNCHEZ M.J., *Control adaptativo y robusto*, Universidad de Sevilla, 1996.
- SIGURD SKOGESTAD, IAN POSTLETHWAITE, *Multivariable Feedback Control (Analysis and Design)*, WILEY, 1997.
- SHAMMA J., *The Control Handbook, Capítol 20: Linealization and Gain Scheduling*, CRC PRESS LLC, 1995.
- SHIRAI, Y., INOUE, H., *Guiding a robot by visual feedback assembling tasks. Pattern Recognition*, vol.5, 1973, pp.99-108.
- T.C. YANG, Z.T. DING, H. YU, *Decentralised power system load frequency control beyond the limit of diagonal dominance*, Electrical Power and Energy Systems 24 173-184, March 2001.
- T.C. YANG, J.H. ZHANG, H. YU, *A new decentralised controller design method with application to power-system stabiliser design*, Control Engineering Practice 7 537-545, October 1998.
- ZHOU, DOYLE J. i GLOVER K., *Robust and Optimal Control*, PRENTICE-HALL, 1996.



## Bibliografia complementaria

A continuació es citen altres referències consultades que no es citen en la Memòria:

- A.CASALS, J.AMAT i R.SARRATE, *Robot control strategy for camera guidance in laparoscopic surgery*, ISRAM 1996.
- DAE RYOOK YANG, *Control of Multi-Input Multi-Output Processes*, CBE495 Process Control Application, 2003.
- GOODWIN G., GRAEBA S.F., SALGADO M.E., *Control System Design*, Chapter 26, PRENTICE-HALL, 2001. (per desacoplador)
- HEIDAR A.MALKI, DAVE MISIR, DENNY FEIGENSPAN, *Fuzzy PID control of a flexible-joint robot arm with uncertainties from time-varying loads*, IEEE Transactions on Control Systems Technology vol.5 n°3, May 1997.
- HONG CUI, ELLING W.JACOBSEN, *Journal of Process Control, Performance limitations in decentralized control*, Journal of Process Control 12 485-494, 2002.
- J. E.NORMEY-RICO, ISMAEL ALCALÁ, JUAN GÓMEZ-ORTEGA, EDUARDO F.CAMACHO, *Mobile robot path tracking using a robust PID controller*, Control Engineering Practice 9 1209-1214, April 2001.
- KAZEMIAN, *The sof-PID controller fort he control of a MIMO robot arm*, IEEE Transactions on Fuzzy Systems Vol.10 n°4, August 2002.
- KIM J., *Thesis: Underactuated control of two-link manipulator*, Purdue University, Agost 1998.
- K.J.ASTRÖM, KARL HENRIK JOHANSSON, QING-QUO WANG, *Design of Decoupled PID Controllers for MIMO Systems*, Proceedings of the American Control Conference, June 2001.
- L.ACOSTA, M.SIGUT, J.A. MÉNDEZ, A. HAMILTON, G.N. MARICHAL i L.MORENO, *Decoupling of the 10m grantecan telescope's primary mirror dynamics and design of a controller for noise rejection*, Proceedings of the American Control Conference, 2000.
- M.T. THAM, *Multivariable control: An Introduction to Decoupling Control/ MTT/*, IEE Control Engineering Series 37, 1999.
- TECH KIANG LEE, JINGXUE SHEN, MIN-SHEN CHIU, *Independent design of robust partially decentralized controllers*, Journal of Process Control 11 419-428, 2001
- Y.HALEVI, Z.J.PALMOR, T.EFRATI, *Automatic tuning of decentralized PID controllers for MIMO processes*, Journal of Process Control Vol.7 n°2, 1997.
- YONGHO LEE, JEONGSEOK LEE, SUNWON PARK, *PID controller tuning for integrating and unstable processes with time delay*, Chemical Engineering Science 55 3481-3493, October 1999.
- YUDI SAMYDUIA, KUSMAYANTO KADIMAN, *Control design for recycled, multi unit processes*, Journal of Process Control 13 311-324, 2003.
- ZIDONG WANG, HANQING ZENG, DANIEL W.C.HO, *Multiobjective Control of a Four-Link Flexible Manipulator: A Robust Approach*, IEEE Transactions on Control Systems Technology Vol.10 n°6, November 2002.
- [[http://www.cpr2valladolid.com/tecno/cyr\\_01/robotica/aplicaciones/medicina.htm](http://www.cpr2valladolid.com/tecno/cyr_01/robotica/aplicaciones/medicina.htm), 10 de febrer de 2004].
- [<http://ciberhabitat.com/hospital/robotica.htm>, 10 de febrer de 2004].
- [<http://www.diariomedico.com/grandeshist/informat00/cinco.html>, 10 de febrer de 2004].

