



EPS

Escola Politècnica
Superior

Projecte/Treball Fi de Carrera

Estudi: Enginyeria Tècn. Ind. Electrònica Ind. Pla 2002

Títol: Millora experimental de les lleis de control predictiu, aplicades sobre la plataforma PRIM I, per el seguiment de trajectòries

Document: Resum

Alumne: Xavier Serra Serinya

Director/Tutor: Lluís Pacheco Valls

Departament: Arquitectura i Tecnologia de Computadors

Àrea: ATC

Convocatòria (mes/any): febrer / 2009

ÍNDIX

1. INTRODUCCIÓ	2
2. DESCRIPCIÓ DEL SISTEMA A CONTROLAR.....	3
2.1. Característiques del robot.....	3
2.2. Sistema odomètric	3
3. CÀLCUL DEL MODEL DE POSICIONAMENT I ORIENTACIÓ	5
4. LA ZONA MORTA	7
5. EL CONTROL PREDICTIU	8
5.1. Fonaments teòrics del control predictiu	8
5.2. Optimització convexa.....	9
6. IMPLEMENTACIÓ DEL CONTROL PREDICTIU	10
6.1. Programa de simulació del MPC	10
6.2. Programa d'horitzó	10
6.3. Programa de cerca de l'error	11
6.3.1. Mètode valor òptim	11
6.3.2. Mètode gradient.....	11
6.4. Funcions de costos.....	12
6.4.1. Funció de costos distància a la trajectòria	12
6.4.2. Funció de costos diferència d'orientació.....	13
6.4.3. Funció de costos error acumulat de la distància a la trajectòria	13
6.4.4. Funció de costos error acumulat de la diferència d'orientació.....	14
6.4.5. Funció de costos de gir i rotació	14
6.5. Condicions contractives.....	14
7. SIMULACIÓ DE TRAJECTÒRIES AMB EL MÈTODE VALOR ÒPTIM I EL GRADIENT .	15
8. MÈTODE DISSENY FACTORIAL	16
9. CONCLUSIONS	18

1. INTRODUCCIÓ

Aquest projecte s'aplica sobre el robot PRIM (Plataforma Robotitzada d'Informació Multimèdia), un robot autònom no humanoide creat el 2004 per Ateneu Informàtic (AI) que permet realitzar trajectòries 2D gràcies a un sistema de tracció format per dues rodes motrius propulsades independentment. La plataforma PRIM és controlada a partir del control predictiu, aquest control es va implementar en un projecte anterior, creat per l'Alexandre Blasco. El que es pretén en aquest projecte és millorar els resultats obtinguts en el passat projecte reformulant la llei de control i analitzar les discrepàncies obtingudes en les metodologies que s'utilitzen per minimitzar la funció de costos a partir de simulacions de trajectòries.

2. DESCRIPCIÓ DEL SISTEMA A CONTROLAR

Aquest projecte està basat en base al robot PRIM (Plataforma Robotitzada d'Informació Multimèdia) que l'Ateneu informàtic (AI) va presentar al Fòrum de les Cultures de Barcelona 2004.

2.1. Característiques del robot

El robot PRIM és un robot autònom no humanoide de 1,50 metres d'alçada i 30 kg de pes, amb una autonomia de 4 hores. Les mides d'aquest robot són de 60 x 45 cm. El seu sistema de tracció es compon de dues rodes motrius propulsades per dos motors elèctrics de corrent continu disposats de forma independent, permetent-li assolir una velocitat de 3,6 km/h amb 40 kg de càrrega. El robot consta també d'una roda central posterior, no motriu, que és l'encarregada de que mantingui l'equilibri. Aquesta roda és omnidireccional i té una rotació de 360° en l'eix z.

2.2. Sistema odomètric

L'odometria és el sistema de posicionament i orientació que ens permet trobar en cada instant les coordenades (x, y, θ) a partir del valor inicial d'aquestes i del recorregut que hagin realitzat les rodes del robot. El recorregut de les rodes seran conegudes per la lectura dels encoders que porten acoblades.

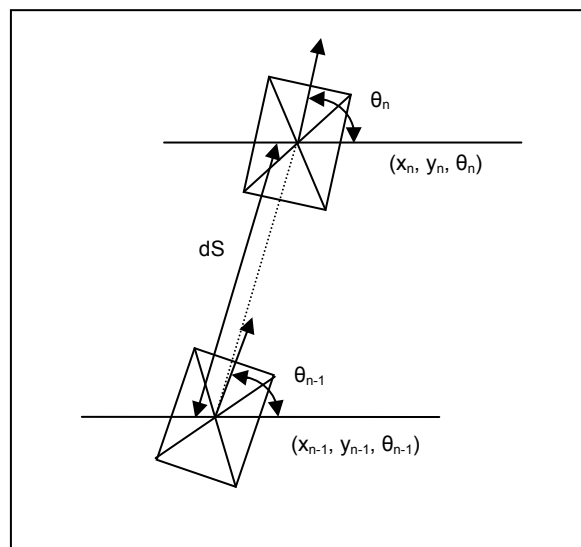


Figura 1. Determinació de les coordenades

Trigonomètricament obtindrem una expressió que ens doni el valor de x , y i θ actuals a partir de les coordenades de l'estat anterior i de la lectura dels polsos dels encoders. La constant dS indica el que ha recorregut el robot, la constant p indica el nombre de polsos que marca l'encoder per cada volta que realitza la roda, les variables n_e i n_d són el nombre de polsos que marquen els encoders, la constant L és l'amplada del robot, i R_e i R_d són els radis de les rodes.

$$x_n = x_{n-1} + \frac{\pi \cdot (R_e \cdot n_e + R_d \cdot n_d)}{p} \cdot \cos\left(\theta_{n-1} + \frac{2\pi \cdot (R_d \cdot n_d - R_e \cdot n_e)}{p \cdot L}\right) \quad (\text{Eq. 1})$$

$$y_n = y_{n-1} + \frac{\pi \cdot (R_e \cdot n_e + R_d \cdot n_d)}{p} \cdot \sin\left(\theta_{n-1} + \frac{2\pi \cdot (R_d \cdot n_d - R_e \cdot n_e)}{p \cdot L}\right) \quad (\text{Eq. 2})$$

$$\theta_n = \theta_{n-1} + \frac{2\pi \cdot (R_d \cdot n_d - R_e \cdot n_e)}{p \cdot L} \quad (\text{Eq. 3})$$

3. CÀLCUL DEL MODEL DE POSICIONAMENT I ORIENTACIÓ

El model de posició ens determina en cada instant les coordenades x i y del robot i la seva orientació θ . Per obtenir-lo ens basem en el model de velocitats i en la odometria del sistema.

Les següents funcions són els models continu de velocitat dels motors pels tres models de velocitat. El sistema de control del robot escollirà un model de velocitat segons la consigna que li arribi al motor.

$$V_{Dbaix}(s) = \frac{0,826}{0,45s + 1} \cdot U_{Dbaix}(s) \quad (\text{Eq. 4})$$

$$V_{Ebaix}(s) = \frac{0,877}{0,45s + 1} \cdot U_{Ebaix}(s) \quad (\text{Eq. 5})$$

$$V_{Dmig}(s) = \frac{0,924}{0,405s + 1} \cdot U_{Dmig}(s) \quad (\text{Eq. 6})$$

$$V_{Emig}(s) = \frac{0,919}{0,405s + 1} \cdot U_{Emig}(s) \quad (\text{Eq. 7})$$

$$V_{Dalt}(s) = \frac{0,953}{0,423s + 1} \cdot U_{Dalt}(s) \quad (\text{Eq. 8})$$

$$V_{Ealt}(s) = \frac{0,908}{0,417s + 1} \cdot U_{Ealt}(s) \quad (\text{Eq. 9})$$

Aplicant un retenidor d'ordre zero al model de velocitats i aplicant la transformada z a la funció de transferència, obtenim el model discret del sistema. Discretitzant aquestes funcions obtenim les equacions en diferències en funció de k . Aquestes equacions permeten uns càlculs més senzills i ràpids. Obtindrem les equacions que determinen els increments de recorregut de cada roda durant un període multiplicant les equacions de velocitats pel període T .

A partir de l'odometria i dels models anteriors es pot trobar les equacions que determinen $x(k)$, $y(k)$ i $\theta(k)$. L'Equació 10 mostra el model global del sistema.

$$\left. \begin{aligned} \Delta r_{\text{Dbaix}}(k) &= e^{-2,222T} \cdot \Delta r_{\text{Dbaix}}(k-1) + 0,826 \cdot (1 - e^{-2,222T}) \cdot T \cdot U_{\text{Dbaix}}(k-1) \\ \Delta r_{\text{Ebaix}}(k) &= e^{-2,222T} \cdot \Delta r_{\text{Ebaix}}(k-1) + 0,877 \cdot (1 - e^{-2,222T}) \cdot T \cdot U_{\text{Ebaix}}(k-1) \\ \Delta r_{\text{Dmig}}(k) &= e^{-2,469T} \cdot \Delta r_{\text{Dmig}}(k-1) + 0,924 \cdot (1 - e^{-2,469T}) \cdot T \cdot U_{\text{Dmig}}(k-1) \\ \Delta r_{\text{Emig}}(k) &= e^{-2,469T} \cdot \Delta r_{\text{Emig}}(k-1) + 0,919 \cdot (1 - e^{-2,469T}) \cdot T \cdot U_{\text{Emig}}(k-1) \\ \Delta r_{\text{Dalt}}(k) &= e^{-2,364T} \cdot \Delta r_{\text{Dalt}}(k-1) + 0,953 \cdot (1 - e^{-2,364T}) \cdot T \cdot U_{\text{Dalt}}(k-1) \\ \Delta r_{\text{Ealt}}(k) &= e^{-2,398T} \cdot \Delta r_{\text{Ealt}}(k-1) + 0,908 \cdot (1 - e^{-2,398T}) \cdot T \cdot U_{\text{Ealt}}(k-1) \\ \\ x(k) &= x(k-1) + \left(\frac{1}{2} \Delta r_{\text{D}}(k) + \frac{1}{2} \Delta r_{\text{E}}(k) \right) \cdot \cos\theta(k-1) \\ y(k) &= y(k-1) + \left(\frac{1}{2} \Delta r_{\text{D}}(k) + \frac{1}{2} \Delta r_{\text{E}}(k) \right) \cdot \sin\theta(k-1) \\ \theta(k) &= \theta(k-1) + \frac{1}{L} \Delta r_{\text{D}}(k) - \frac{1}{L} \Delta r_{\text{E}}(k) \end{aligned} \right\} \quad (\text{Eq. 10})$$

4. LA ZONA MORTA

La zona morta és un fenomen que ens anirem trobant quan el robot s'apropa al punt final. El robot realitza el seu recorregut però quan arriba al punt final no es decideix a acabar d'arribar-hi, tot i que es trobi a pocs centímetres. Això és degut a que no respon a les consignes que li arriben. La nostra solució és limitar les consignes baixes de manera que no intervinguin. Si la consigna esquerra i la dreta és inferior o igual a un rang de consignes determinat llavors les velocitats de la roda esquerra i dreta seran de 0 cm/s, prefixant les UE i UD a 0. Els rangs de consignes seran: si la consigna esquerra i la dreta és inferior o igual que 15; si la consigna esquerra és inferior o igual que 20 i la consigna dreta és inferior o igual que 5; i si la consigna esquerra és inferior o igual que 5 i la consigna dreta és inferior o igual que 20.

5. EL CONTROL PREDICTIU

El control predictiu, basant-se en el model del procés, calcula els valors de control necessaris per obtenir una sortida adient d'acord amb uns valors coneguts, unes restriccions i condicions d'operació.

5.1. Fonaments teòrics del control predictiu

Els controladors predictius calculen els valors de les variables que es tracten a cada període de mostreig tenint en compte els valors de consigna desitjats per les variables controlades, les restriccions i les condicions d'operació del procés. D'aquesta forma, el control predictiu basat en el model és una estratègia de control que calcula les consignes basant-se en un model explícit del procés, que s'utilitza per preveure el seu comportament futur sobre un horitzó determinat.

El control predictiu ha de tenir un model intern que és utilitzat per predir el comportament de la planta, començant en l'instant actual i fins a un futur horitzó de predicció. La idea del control és veure quina consigna donarà el millor comportament previst. El model pot dependre de consignes anteriors $u(k-1)$, $u(k-2)$,..., però no de la consigna actual $u(k)$.

L'horitzó de predicció tindrà una seqüència de consignes, d'aquesta només el seu primer element serà aplicat com a consigna real a la planta. Assignarem el valor de $\hat{u}(k|k)$ a $u(k)$. Llavors el cicle; mesura de la sortida, predicció i determinació de la seqüència de consignes serà repetit en el període de mostreig següent. Assignarem a la variable k el valor $k+1$ i tornarem a realitzar el cicle. Això és el que s'anomena, concepte d'horitzó de predicció mòbil ('receding horizon' idea). Aquest concepte d'horitzó pretén simbolitzar l'horitzó terrestre, on a mesura que un s'hi aproxima l'horitzó es va allunyant, de manera que sempre es veu però mai s'hi pot arribar.

La seqüència de consignes condicionarà la funció de costos (funció error). Haurem de buscar les consignes que donin un mínim error, i aquesta seqüència de consignes serà la que portarà a la planta a la millor posició possible en funció de la posició desitjada.

L'algorisme de control, veure Figura 2, determina els passos que ha de seguir el sistema per tal de portar a terme el control predictiu.

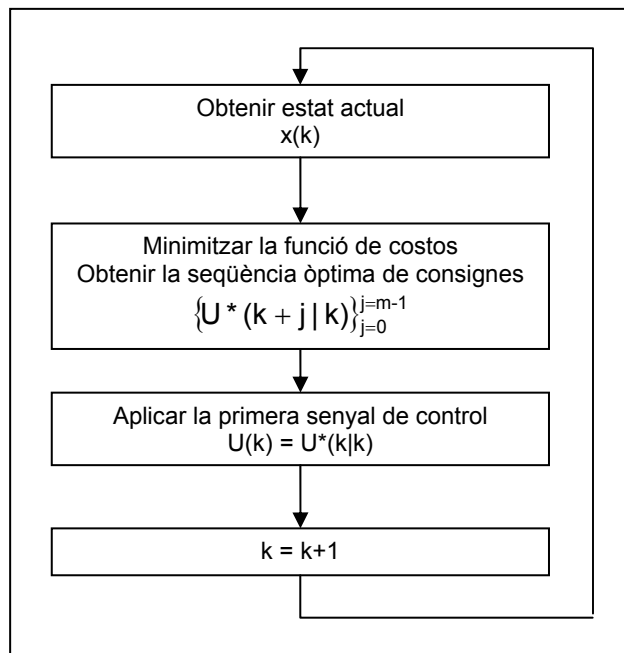


Figura 2. Algorisme de control predictiu

5.2. Optimització convexa

Matemàticament, el problema de control predictiu es redueix a la minimització de la funció de costos (funció de l'error), que està lligat a unes restriccions en la funció de les consignes a donar. Aquest és un problema matemàtic de difícil solució, no és possible solucionar amb mètodes analítics i requereix una gran potència de càlcul, ja que és necessari treballar a la inversa del model per cercar les consignes òptimes. Amb la tècnica de l'optimització convexa podrem resoldre problemes de minimització del tipus control predictiu a partir de mètodes numèrics. El seu estudi permetrà trobar alguns sistemes per poder aplicar al control predictiu, per poder-lo fer eficaç i ràpid. En el present projecte s'han utilitzat dos mètodes per cercar les consignes òptimes, el mètode valor òptim i el mètode gradient.

6. IMPLEMENTACIÓ DEL CONTROL PREDICTIU

La implementació del control es realitzarà a través de dues plataformes: Microsoft Visual C++ per poder realitzar simulacions des d'un PC personal i Linux que és la plataforma integrada dins la pròpia CPU de treball del robot. Es portaran a terme rutines, que representaran diferents mètodes de minimització de la funció de costos. Aquestes rutines actuaran com a simulació de les trajectòries que finalment podran ser realitzades pel robot.

6.1. Programa de simulació del MPC

Aquest programa s'anomena `mpc.cpp` i serà la base de totes les rutines. El que farà serà calcular en l'entorn d'un horitzó de predicció de nombre de passos determinat, el valor de les variables de posició (x , y i θ), de consignes (UE i UD) i de velocitats (Δv_e i Δv_d) en cada instant de temps a partir dels valors inicials i dels valors procedents de les lectures del robot. També demanarà a l'usuari els valors desitjats per les variables (x , y) i ens retornarà els valors de posicionament i l'error. Tal com està configurat ara el programa, el robot pot realitzar trajectòries de 4 punts.

La consigna serà donada amb una estructura determinada segons l'horitzó de predicció. Hi ha tres tipus d'horitzons: el curt, el mig i el llarg.

El programa està sotmès a tres variables: `SimularRobot`, en funció de si la tenim activada o no podem simular o actuar directament sobre el propi robot. `Rotacio`, ens permetrà que a l'hora de girar, el robot roti sobre el seu propi eix z utilitzant consignes positives i negatives, o ve efectuari un gir de forma normal utilitzant consignes positives per orientar-se. I `ErrorMinim` que permet escollir entre aplicar el mètode valor òptim o el mètode gradient.

Finalment es representen tots els valors de posició, consignes i velocitat de cada iteració (k) per pantalla i també es guarden en un fitxer que podem visualitzar i representar gràficament amb el Matlab com a un arxiu tabulat.

6.2. Programa d'horitzó

Aquest programa afecta a la subrutina que genera les consignes, ja que la manera que tenim de variar el nostre horitzó de predicció és actuant sobre les consignes que podem aplicar al robot. Disposem de tres models d'horitzó: l'horitzó curt que simula una visió del

robot més pròxima, l'horitzó mig que simula una visió del robot més llunyana i l'horitzó llarg que simula una visió del robot molt més llunyana. El model curt consta de cinc períodes: tres d'increments i dos de constants. El model mig consta de vuit períodes: quatre d'increments i quatre de constants. I l'horitzó llarg consta de deu períodes, cinc d'increments i cinc de constants.

6.3. Programa de cerca de l'error

Aquest programa afecta a la subrutina que calcula el mínim error a partir de la funció de costos. Aquesta subrutina calcula les consignes òptimes per que, amb l'horitzó de predicció que s'ha determinat, l'error entre les variables desitjades i les variables predites pel programa de simulació siguin el mínim. Disposem de dos mètodes per trobar l'error: mètode valor òptim i mètode gradient.

6.3.1. Mètode valor òptim

Aquest mètode calcula l'error de totes les combinacions de consigna que hi hagi en una iteració. Aquest error es calcularà a partir de la funció de costos que hi hagi a la subrutina i es guardarà en una taula d'errors. De tots els errors que hi hagi en aquesta taula, el programa trobarà el valor més petit, que serà l'error mínim. Aquest mètode podria ser relativament lent, segons l'horitzó de predicció que utilitzem, haurà de fer 1681, 961 o 625 simulacions si utilitzem l'horitzó curt, mig o llarg, respectivament. I podria ser possible que excedís el període del sistema. Tot i així, en les proves de simulació i en els experiments amb el robot, aquests problemes no ens han sortit. Per tant considerem que pot ser un mètode pràctic per tal de trobar les consignes òptimes.

6.3.2. Mètode gradient

Aquest mètode té el mateix objectiu que el mètode valor òptim, trobar la seqüència de consignes òptima que donin un error mínim per realitzar el seguiment de la trajectòria. Però utilitzant la tècnica del gradient per tal d'agilitar els càlculs i hagi de realitzar menys simulacions.

A la taula d'errors, es genera una taula de x per x , segons l'horitzó, amb els valors de les possibles consignes. El programa es situa en el punt mig de la taula i llavors calcula l'error dels vuit valors que té al voltant, que serien els punts veïns, realitzant la simulació amb les

consignes a cada punt. El punt mig, juntament amb els altres 8 punts veïns formen un quadrat, que es podria dividir en quatre quadrants de quatre valors cada un. En cada quadrant apareixeria el punt del mig. Llavors suma els errors de cada punt de cada quadrant, sense sumar-hi el punt del mig. El quadrant que tingui l'error més petit serà la direcció desplaçament, el programa entén que cap a aquella direcció es troben els valors més petits, la funció error té un pendent descendent cap a aquella direcció. Cap a aquella direcció es formarà un altra quadrat, i es posarà en el seu centre i tornarà a realitzar el mateix procés, tantes vegades com facin falta fins que la taula resultant sigui prou petita.

6.4. Funcions de costos

La funció de costos serà la funció error, aquesta busca les consignes que donen un error mínim. Amb aquesta funció el control predictiu cerca la seqüència de consignes que portarà a la planta a la millor situació possible en funció de la situació desitjada al principi del plantejament del recorregut de la trajectòria que es vulgui fer.

6.4.1. Funció de costos distància a la trajectòria

Aquesta funció estarà formada per dos factors: l'aproximació al punt destí (dif) i la distància a la trajectòria rectilínia (distTrj). Tal com podem veure a l'Equació 11.

$$\text{difTot}^2 = \text{dif}^2 + \text{dif}^2 \cdot \text{distTrj}^2 \quad (\text{Eq. 11})$$

El factor aproximació al punt destí, dif^2 , compara el valor predit al final de l'horitzó de predicció (x, y) amb els valors desitjats (x_d , y_d) i calcula la suma dels quadrats de l'error total de cada variables.

El factor distància a la trajectòria rectilínia calcula la distància que hi ha entre una recta rectilínia que representa la trajectòria que ha de seguir la simulació del robot i el punt on es trobi en cada període. Per calcular aquest factor generem una trajectòria rectilínia que va del punt inicial fins al punt desitjat, això permetrà minimitzar la distància del valor predit al valor desitjat i a més minimitzar la distància d'aquest valor a la recta que conté la trajectòria generada.

El factor aproximació al punt destí serà utilitzat com a factor de ponderació. Així es podrà plantejar una suma entre el factor aproximació al punt destí (dif) i el producte entre aquest

factor (dif) i el factor de la distància a la trajectòria rectilínia, tal com es pot veure a l'equació 70. D'aquesta manera es normalitza la funció de costos, ja que li donem un pes al factor de la distància a la trajectòria rectilínia per no restar-li importància.

6.4.2. Funció de costos diferència d'orientació

Aquesta funció de costos estarà formada pels factors aproximació al punt destí (dif) i el factor diferència d'orientació (odif2). La funció de costos diferència d'orientació s'expressa de la següent forma.

$$\text{difTot}^2 = \text{dif}^2 + \text{dif}^2 \cdot \frac{\text{odif2}^2}{0,025} \quad (\text{Eq. 12})$$

El factor diferència d'orientació (odif2) compara el valor de l'angle predit al final de l'horitzó de predicció (o) amb el valor de l'angle desitjat (od2) en cada període. Amb l'angle desitjat sabrem quin hauria de ser l'angle correcta per tal d'orientar-se correctament cap al punt desitjat.

El factor diferència d'orientació serà utilitzat com a factor de ponderació. Li donem més pes al factor diferència d'orientació dividint-lo per 0,025. D'aquesta manera li donem un pes determinant a l'orientació del robot, primant per sobre del factor aproximació al punt destí.

6.4.3. Funció de costos error acumulat de la distància a la trajectòria

Aquesta funció de costos té la mateixa filosofia que la funció de costos distància a la trajectòria, però en comptes de mirar l'error mínim en l'últim període de l'horitzó de predicció, aquesta mira el valor quadràtic de l'error acumulat del factor de la distància a la trajectòria rectilínia (distTrj) per tots els períodes de l'horitzó de predicció.

$$\text{difTot}^2 = \text{dif}^2 + \text{dif}^2 \cdot \text{errorAcDistTraj} \quad (\text{Eq. 13})$$

6.4.4. Funció de costos error acumulat de la diferència d'orientació

Aquesta funció de costos té la mateixa filosofia que la funció de costos diferència d'orientació. Però es mira el valor quadràtic de l'error acumulat de la diferència d'orientació ($odif^2$) per tots els períodes de l'horitzó de predicció.

$$difTot^2 = dif^2 + dif^2 \cdot \frac{errorAcOrientacio}{0,025} \quad (Eq. 14)$$

6.4.5. Funció de costos de gir i rotació

Aquesta funció de costos buscarà el valor quadràtic de la diferència d'angles, entre l'angle predit al final de l'horitzó de predicció (o) i l'angle desitjat (od).

$$odif^2 = (od - o)^2 \quad (Eq. 15)$$

6.5. Condicions contractives

Les condicions contractives permeten que quan el robot arribi en el punt desitjat, aquest s'orienti de manera que pugui recórrer la trajectòria amb la millor precisió possible i permeti passar per espais reduïts. Per aturar-se en aquests punts, el programa té unes condicions de convergència cap al punt objectiu basades en les pròpies condicions contractives, que consideren un apropament al punt final tant pel que fa a les coordenades mètriques com a la desviació angular.

Les condicions per que el robot s'aturi en un punt i s'orienti serà que el factor aproximació al punt destí sigui més petit o igual que la variable `ERROR_MAXIM`, que serà un valor prefixat a 20.

7. SIMULACIÓ DE TRAJECTÒRIES AMB EL MÈTODE VALOR ÒPTIM I EL GRADIENT

Per poder veure les diferències que ens podem trobar aplicant un mètode o un altra, hem realitzat una sèrie de simulacions de trajectòries amb totes les funcions de costos i per cada un dels mètodes. Les trajectòries que hem simulat han estat; una trajectòria rectilínia amb les coordenades $(0, 0)$ i $(0, 200)$, una trajectòria de dos punts amb les coordenades $(0, 0)$, $(0, 100)$ i $(-50, 200)$, i una de tres punts amb les coordenades $(0, 0)$, $(25, 50)$, $(25, 100)$, $(0, 150)$ i $(0, 200)$. Amb els resultats de les simulacions hem vist que obtenim diferents resultats en cada trajectòria. No podem dir que hi hagi una funció que ens doni un millor resultat que una altra d'una manera global. Depèn de la trajectòria ens surt una funció amb millors resultats que una altra. Per això ens decidim a fer experiments factorials per diverses trajectòries, aplicant-les a les lleis de control, per trobar una millora en les lleis de control predictiu.

8. MÈTODE DISSENY FACTORIAL

Vists els resultats de les simulacions, millorarem el control del robot aplicant-li el disseny d'experiments factorials per tal d'analitzar els diferents paràmetres de les funcions de costos. A partir d'anar fent proves, variant el nivell fixat dels factors per totes les seves possibles combinacions analitzarem el resultat dels factors que volem millorar.

Els factors que analitzarem seran la desviació de la distància trajectòria (TDD), la desviació de la orientació (OD) i l'aproximació al punt destí (APD). El factor APD es refereix al factor de les funcions de costos amb nomenclatura dif^2 que es troba sol a la funció, el factor TDD es refereix al producte entre dif^2 i $distTraj^2$, o $errorAcDistTraj$, i el factor OD es refereix al producte entre dif^2 i $Odif^2/0,025$, o $errorAcOrientacio/0,025$, segons la funció de costos que utilitzem.

Els paràmetres estudiats són el temps, la mitjana de l'error distància a la trajectòria i la mitjana de la velocitat del robot (Ar). Amb el paràmetre temps estudiarem el temps que tarda el robot en arribar al punt final, amb el paràmetre mitjana error distància a la trajectòria estudiarem la desviació de la trajectòria i amb el paràmetre mitjana velocitat estudiarem l'esforç de control. Aquests paràmetres seran estudiats donant un pes determinat a cada factor de la funció de costos. Els pesos que li donarem a la funció tindran dos nivells. I per cada combinació es faran 2 o 3 proves del recorregut de la trajectòria amb el robot. La mitjana de les 2 o 3 proves dels paràmetres a estudiar permetrà un anàlisi estadístic més apurat per cada combinació de factors. L'estudi s'ha fet a partir de cinc trajectòries diferents, en les que el seu punt final està dins del marge de visió que ens proporcionarà la càmera que s'encarrega de la visió. A partir de la desviació estàndard dels paràmetres es determinarà la importància dels efectes. Si la desviació estàndard de l'efecte principal o lateral, relatiu a TDD, OD i APD, és similar a 2 o 3 vegades més gran que la desviació estàndard dels paràmetres llavors tindrem en consideració la importància d'aquests efectes i podrem considerar que hi ha hagut millores segons els paràmetres que estiguin implicats.

L'efecte principal del factor APD es calcularà utilitzant la següent funció:

$$ME_{APD} = \frac{Y_3 + Y_2}{2} - \frac{Y_1 + Y_0}{2} \quad (\text{Eq. 16})$$

L'efecte principal del factor TDD o OD podrà ser calculat per l'equació següent.

$$ME_{\text{TDD or OD}} = \frac{Y_3 + Y_1}{2} - \frac{Y_2 + Y_0}{2} \quad (\text{Eq. 17})$$

Els efectes laterals seran calculats per l'equació següent.

$$LE_{\text{APD}_{\text{(TDD or OD)}}} = Y_3 - Y_0 \quad (\text{Eq. 18})$$

On Y_n fa referència a la mitjana de la combinació que li correspongui segons el número que tingui indexat per cada paràmetre que analitzem; temps, mitjana error distància a la trajectòria i mitjana velocitat. Y_0 correspon a la combinació 00, Y_1 a la 01, Y_2 a la 10 i Y_3 a la 11.

A partir dels resultats dels experiments veiem que per cada trajectòria una combinació determinada ens dóna uns resultats significatius que milloren els paràmetres que hem estudiat. Amb aquests resultats, modifiquem la llei de control del programa de manera que segons el tipus de trajectòria, el càlcul de la predicció de la trajectòria vindrà precedida per una combinació de la funció de costos determinada.

Amb la modificació de la llei de control realitzem una sèrie d'experiments on el robot es desplaça per una sèrie de punts, aquests experiments s'han fet per cada una de les funcions de costos en que hem experimentat per millorar la llei de control a partir dels experiments factorials amb trajectòries d'un sol punt. I tornem a analitzar el temps, la precisió de la trajectòria, i la velocitat del robot. Per aquests experiments compararem la llei fixa i la llei variable. En la llei fixa la funció de costos no varia, els factors APD, TDD i OD es mantenen en valor alt, en canvi en la llei variable la funció de costos anirà canviant segons el tipus de trajectòria que el robot es tingui que desplaçar.

Un cop hem analitzat els resultats dels experiments factorials de trajectòries de varis punts, hem vist que obtenim resultats diversos per cada funció de costos i tipus de trajectòria.

9. CONCLUSIONS

Com a objectius, al ser la continuació del projecte de l'Alexandre Blasco Gutiérrez, ens hem marcat estudiar la teoria de la llei del model control predictiu aplicat a la plataforma robòtica PRIM, així com el programari desenvolupat en el projecte esmentat anteriorment.

S'ha inclòs els tres models experimentals de velocitat obtenint un control més precís en el seguiment de les trajectòries.

S'ha intentat evitar el fenomen de la zona morta restringint les consignes amb un valor baix, de manera que aquestes no seran utilitzades.

S'ha simulat i analitzat les discrepàncies dels resultats obtinguts de diverses trajectòries utilitzant el mètode valor òptim i el mètode gradient. Hem comprovat que depèn de la trajectòria que simulem obtenim una funció amb un mètode específic que ens donen uns resultats millors. Però vists els resultats, no ens podem quedar amb una sola funció i amb un sol mètode.

Per millorar la llei de control predictiu hem fet un disseny d'experiments factorials per trajectòries d'un sol punt per trobar els valors adients en el seguiment de la trajectòria, aquests experiments han estat realitzats per els tres horitzons. A partir dels resultats dels experiments factorials, hem reformulat la llei de control. I hem fet nous experiments factorials amb trajectòries de varis punts amb les noves lleis amb l'horitzó curt, ja que amb els altres dos horitzons no hem obtingut els resultats esperats. Els resultats que hem obtingut en els experiments factorials de les trajectòries de varis punts han estat diversos per cada trajectòria i tipus de funció de costos. Per tant, al utilitzar funcions de costos variables respecte les funcions fixes utilitzades anteriorment, podem afirmar que hem obtingut millores en el control del robot, i seria interessant seguir investigant el mètode del disseny factorial per millorar els resultats obtinguts.

I per últim s'ha fet un estudi i una implementació de les condicions contractives que considerin un apropament al punt final tant per les coordenades mètriques com per a la desviació angular.

Com a treballs futurs es proposa un estudi de la zona morta per millorar el control del robot quan arriba al punt final, implementar el present projecte a la visió per computador per que realitzi la part de control.