



Universitat de Girona

DISSENY D'AGENTS FÍSICS: INCLUSIÓ DE CAPACITATS ESPECÍFIQUES PER A L'AVALUACIÓ DE L'EFICIÈNCIA D'ACCIONS

Albert OLLER PUJOL

ISBN: 84-688-6471-4

Dipòsit legal: Gi.385-2004

<http://hdl.handle.net/10803/7723>

ADVERTIMENT. L'accés als continguts d'aquesta tesi doctoral i la seva utilització ha de respectar els drets de la persona autora. Pot ser utilitzada per a consulta o estudi personal, així com en activitats o materials d'investigació i docència en els termes establerts a l'art. 32 del Text Refós de la Llei de Propietat Intel·lectual (RDL 1/1996). Per altres utilitzacions es requereix l'autorització prèvia i expressa de la persona autora. En qualsevol cas, en la utilització dels seus continguts caldrà indicar de forma clara el nom i cognoms de la persona autora i el títol de la tesi doctoral. No s'autoritza la seva reproducció o altres formes d'explotació efectuades amb finalitats de lucre ni la seva comunicació pública des d'un lloc aliè al servei TDX. Tampoc s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant als continguts de la tesi com als seus resums i índexs.

ADVERTENCIA. El acceso a los contenidos de esta tesis doctoral y su utilización debe respetar los derechos de la persona autora. Puede ser utilizada para consulta o estudio personal, así como en actividades o materiales de investigación y docencia en los términos establecidos en el art. 32 del Texto Refundido de la Ley de Propiedad Intelectual (RDL 1/1996). Para otros usos se requiere la autorización previa y expresa de la persona autora. En cualquier caso, en la utilización de sus contenidos se deberá indicar de forma clara el nombre y apellidos de la persona autora y el título de la tesis doctoral. No se autoriza su reproducción u otras formas de explotación efectuadas con fines lucrativos ni su comunicación pública desde un sitio ajeno al servicio TDR. Tampoco se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al contenido de la tesis como a sus resúmenes e índices.

WARNING. Access to the contents of this doctoral thesis and its use must respect the rights of the author. It can be used for reference or private study, as well as research and learning activities or materials in the terms established by the 32nd article of the Spanish Consolidated Copyright Act (RDL 1/1996). Express and previous authorization of the author is required for any other uses. In any case, when using its content, full name of the author and title of the thesis must be clearly indicated. Reproduction or other forms of for profit use or public communication from outside TDX service is not allowed. Presentation of its content in a window or frame external to TDX (framing) is not authorized either. These rights affect both the content of the thesis and its abstracts and indexes.



UNIVERSITAT DE GIRONA

DEPARTAMENT D'ELECTRÒNICA INFORMÀTICA I AUTOMÀTICA

DISSENY D'AGENTS FÍSICS: INCLUSIÓ DE
CAPACITATS ESPECÍFIQUES PER A L'AVALUACIÓ
DE L'EFICIÈNCIA D'ACCIONS

MEMÒRIA REDACTADA PER OPTAR AL GRAU DE DOCTOR ENGINYER

INDUSTRIAL, PER LA UNIVERSITAT DE GIRONA, PER ALBERT OLLER I PUJOL

VIST-I PLAU DEL DIRECTOR DE LA TESI

DR. JOSEP LLUÍS DE LA ROSA I ESTEVA

TITULAR D'UNIVERSITAT

D'ENGINYERIA DE SISTEMES I AUTOMÀTICA

GIRONA, JULIOL DE 2002



UNIVERSITAT DE GIRONA

DEPARTAMENT D'ELECTRÒNICA INFORMÀTICA I AUTOMÀTICA

DISSENY D'AGENTS FÍSICS: INCLUSIÓ DE
CAPACITATS ESPECÍFIQUES PER A L'AVALUACIÓ
DE L'EFICIÈNCIA D'ACCIONS

ALBERT OLLER I PUJOL

DIRECTOR:

JOSEP LLUÍS DE LA ROSA I ESTEVA

GIRONA, JULIOL DE 2002

*A la Carme,
i als nostres fills, en Pere i la Neus.*

Agraïments

En primer lloc vull donar les gràcies a en Josep Lluís de la Rosa, director i supervisor d'aquest treball, per la seva experiència i suport.

Vull també agrair especialment l'ajuda de na Bianca Innocenti per la feina feta en l'obtenció de dades experimentals dels robots futbolistes.

Als membres del ROGI-Team dels primers anys: Josep Sitjà, Santi Esteve, Albert Figueras, Josep Anton Ramon i en Rafael García, que sense el seu treball els robots futbolistes no existirien.

Als Drs. Ulises Cortés (Universitat Politècnica de Catalunya), Beatriz López (Universitat de Girona), Armin Schmilovici (Ben Gurion University, Israel), Josep Vehí (Universitat de Girona), Javier Serrano (Universitat Autònoma de Barcelona), Andreu Català (Universitat Politècnica de Catalunya), Rita Planas (Universitat Politècnica de Catalunya), Joaquim Meléndez (Universitat de Girona), Josep Puyol (Institut d'Intel·ligència Artificial) i Enric Vidal (Universitat Rovira i Virgili), els hi dono les gràcies per haver revisat els continguts d'aquest text i per la seva contribució a millorar-ne algunes parts.

Finalment, haig d'agrair a tot el grup eXiT de la Universitat de Girona el seu recolzament, el temps i la paciència invertits, i, en especial, el bon ambient respirat durant els anys d'estada a la Universitat de Girona.

INDEX

INTRODUCCIÓ GENERAL

PART I: INTRODUCCIÓ ALS AGENTS FÍSICS

| | |
|--|-----------|
| 1. ELS SISTEMES MULTI-AGENT | 1 |
| <i>1.1. Introducció: l'evolució de la I.A. cap a sistemes basats en agents</i> | <i>1</i> |
| <i>1.2. L'estat actual dels sistemes multi-agent</i> | <i>6</i> |
| 1.2.1. La tecnologia agent | 6 |
| 1.2.1.1. Camps d'aplicació..... | 7 |
| 1.2.1.2. Plataformes de disseny d'agents..... | 10 |
| 1.2.2. Els sistemes multi-agent | 12 |
| 1.2.2.1. Comportament cooperatiu per a la resolució de problemes..... | 14 |
| 1.2.2.2. Avaluació dels sistemes multi-agent..... | 16 |
| 1.2.2.3. Situació Actual..... | 17 |
| 1.2.2.3.1 Teories formals | 17 |
| 1.2.2.3.2. Arquitectures d'agent | 18 |
| 1.2.2.3.3. Llenguatges d'agents | 19 |
| 1.2.2.3.4. Aplicacions | 19 |
| 1.2.2.3.5. Plataformes | 20 |
| 1.2.3. Programació orientada a agents | 22 |
| 1.2.3.1. Llenguatge formal..... | 23 |
| 1.2.3.2. Programa intèrpret | 24 |
| 1.2.3.3. Agentificador | 24 |
| 1.2.3.4. L'AGENT-0: general | 25 |
| 1.2.3.5. L'AGENT-0 : llenguatge i intèrpret | 26 |
| 1.2.4. FIPA: un intent d'estandarització | 27 |
| 1.2.4.1. Gestió de l'agent | 28 |
| 1.2.4.2. Llenguatge de comunicació per agents | 30 |
| 1.2.4.3. Protocols d'interacció entre agents | 32 |
| 1.2.4.3. Plataformes compatibles amb FIPA | 33 |
| | |
| 2. LA ROBÒTICA MÒBIL I ELS SISTEMES MULTI-ROBOT | 35 |
| <i>2.1. La robòtica mòbil: robots autònoms</i> | <i>36</i> |
| <i>2.2. El control del moviment del robot mòbil</i> | <i>37</i> |
| <i>2.3. La cooperació en sistemes multi-robot</i> | <i>39</i> |
| 2.3.1. Arquitectures de sistemes multi-robot | 41 |
| 2.3.2. Línies generals de recerca | 42 |
| <i>2.4. Resolució de problemes de móns dinàmics amb planificació</i> | <i>46</i> |
| 2.4.1. Planificació en temps real | 48 |
| <i>2.5. El problema dels sistemes de temps real</i> | <i>48</i> |
| 2.5.1. Arquitectures RTAIS | 50 |

| | |
|---|-----------|
| 3. ROBOT + IA = AGENT FÍSIC | 53 |
| 3.1. <i>Del paradigma GOFAIR al situated agent</i> | 55 |
| 3.2. <i>L'agent físic dinàmic</i> | 57 |
| 3.2.1. Exemple il·lustratiu: comboi de dos vehicles | 58 |
| 3.2.2. Extensió del concepte d'agent software al concepte d'agent físic dinàmic | 60 |
| 3.2.3. L'arquitectura InteRRaP per a agents físics | 63 |

PART II: DISSENY D'AGENTS FÍSICS

| | |
|---|-----------|
| 4. DESCRIPCIÓ DE L'ARQUITECTURA DPAA | 1 |
| 4.1. <i>Funcionament d'un mòdul: primera aproximació</i> | 2 |
| 4.2. <i>Arquitectura de Control Global</i> | 5 |
| 4.2.1. Funcionament intern dels mòduls | 7 |
| 4.2.1.1. Mode d'Introspecció (I-mode) | 8 |
| 4.2.1.2. Mode de Control (C-mode) | 8 |
| 4.2.2. Missatges i protocols de diàleg | 9 |
| 4.2.2.1. Protocols de diàleg entre mòduls | 10 |
| 4.2.2.2. Tipus de missatges | 11 |
| 4.2.2.3. Protocol de diàleg entre agents | 12 |
| 4.2.2.4. Estructura dels missatges | 13 |
| 4.2.3. Cicles de funcionament | 13 |
| 4.2.3.1. Activació top-down: procés d'introspecció | 14 |
| 4.2.3.2. Activació bottom-up: procés d'actuació | 16 |
| 4.2.3.3. Comentaris addicionals | 17 |
| 4.3. <i>Definició de les capacitats modulars</i> | 19 |
| 4.3.1. Les capacitats en el mòdul de control | 19 |
| 4.3.1.1. Exemples | 21 |
| 4.3.1.2. Exemple real: comboi de dos robots (I) | 24 |
| 4.3.2. Capacitats en el mòdul de supervisió | 29 |
| 4.3.2.1. Exemples | 30 |
| 4.3.2.2. Exemple real: comboi de dos robots (II) | 37 |
| 4.3.3. Capacitats en el mòdul agent | 38 |
| 4.3.3.1. Exemple | 40 |
| 4.3.3.2. Exemple real: comboi de dos robots (III) | 42 |
| 4.3.4. Especificacions de les capacitats col·lectives: escenes | 46 |
| 4.3.4.1. Exemple | 47 |
| 5. EXEMPLE D'APLICACIÓ: EL FUTBOL ROBÒTIC | 53 |
| 5.1. <i>El domini del futbol robòtic</i> | 54 |
| 5.1.1. Breu història sobre les competicions de robots futbolistes | 54 |
| 5.1.2. La complexitat del domini dels robots futbolistes | 58 |
| 5.2. <i>Disseny dels agents amb DPAA</i> | 60 |
| 5.2.1. Anàlisi prèvia del problema | 61 |

| | |
|---|-----------|
| 5.2.2. Especificacions dels controladors: capacitats atòmiques | 63 |
| 5.2.2.1. "Anar a punt proper" | 65 |
| 5.2.2.2. Capacitats atòmiques | 68 |
| 5.2.2.3. Comentaris sobre la dinàmica del robot | 69 |
| 5.2.2.4. "Anar a punt llunyà" | 70 |
| 5.2.2.5. Capacitats atòmiques | 73 |
| 5.2.3. Especificacions de les tasques: capacitats bàsiques | 74 |
| 5.2.3.1. "Xutar la pilota cap a porteria" | 74 |
| 5.2.3.2. Capacitats bàsiques | 79 |
| 5.2.3.3. Comentaris | 80 |
| 5.2.3.4. "Anar cap a pilota" | 82 |
| 5.2.4. Especificacions dels rols: capacitats simbòliques | 84 |
| 5.2.4.1. 'Rebre la pilota' | 85 |
| 5.2.4.2. Maniobres d'apropament i de xut a porteria..... | 86 |
| 5.2.4.3. Exemples..... | 88 |
| 5.2.5. Especificacions de les escenes: capacitats col·lectives | 92 |
| 5.2.5.1. Escena 'Atac' | 93 |
| 5.3. Conclusions | 94 |

CONCLUSIONS

| | |
|----------------------|----------|
| CONCLUSIONS | 1 |
| <i>Treball futur</i> | 3 |

REFERÈNCIES

| | |
|---|-----------|
| BIBLIOGRAFIA | 1 |
| APÈNDIXS | 15 |
| <i>Apèndix I: Aspectes tècnics dels robots mòbils utilitzats</i> | 17 |
| <i>Apèndix II: Limitacions dinàmiques del robot mòbil</i> | 23 |
| La dinàmica del model del robot | 24 |
| Les restriccions dinàmiques del robot | 25 |
| <i>Apèndix III: Descripció del simulador PHAGSIM</i> | 27 |
| <i>Apèndix IV: Control de moviment d'un robot mòbil amb tracció diferencial</i> | 33 |
| TERMES ESPECÍFICS I ACRÒNIMS | 41 |
| ADRECES WEB INTERESSANTS | 43 |

Introducció general

L'interès científic de l'autor és el control de comunitats de sistemes autònoms, és a dir, grups o conjunts de sistemes que prenen les seves decisions sense la intervenció d'un controlador humà i que tenen recursos per a poder-les executar. Un bon exemple d'un d'aquests sistemes consisteix en ser el robot mòbil. Des d'un punt de vista tecnològic es podria entendre que l'autonomia fa referència a la mobilitat del robot, a la capacitat de poder-se traslladar d'un punt a un altre. En aquesta tesi l'autonomia fa referència a la capacitat que ha de tenir el robot d'observar i raonar sobre l'entorn, i de poder decidir què fer en funció dels objectius que té i de la representació interna que té del món. En aquest sentit, qualsevol robot manipulador podria ser programat com a autònom encara que no pogués traslladar la seva base. En aquest cas la dinàmica pròpia faria referència a les característiques que s'infereixen del moviment de les seves articulacions.

L'interès sobre el control d'aquest tipus de sistemes està motivat per la seva aplicabilitat al control de sistemes complexos. En els darrers anys estan apareixent organitzacions, sistemes i processos amb uns graus de complexitat que dificulten el disseny de sistemes de control centralitzats i, en canvi, afavoreixen l'aparició de tècniques relacionades amb el disseny de sistemes de control distribuït. Aquesta distribució afecta tant a la informació com als recursos i això fa que l'avaluació de la qualitat d'aquests sistemes sigui un problema. De fet, l'optimalitat com a criteri de mesura de qualitat perd sentit i pot substituir-se pel criteri de robustesa. Un punt a tenir

en compte en aquests sistemes és la propietat de la interoperabilitat, és a dir, la capacitat de poder intercanviar informació amb d'altres sistemes generalment diferents i treballar amb ells en entorns que canvien ràpidament.

Les tecnologies basades en agents són les que actualment prometen més alhora de poder dissenyar sistemes de control distribuït, sobretot perquè inclouen tot el simbolisme de la intel·ligència artificial. Els agents són sistemes *hardware* o *software* amb un grau d'autonomia que varia segons els casos, i que poden realitzar tasques amb un grau de concreció variable, també segons els casos. Aquests agents autònoms interactuen amb l'entorn recollint-ne informació i modificant-lo parcialment a partir de l'execució d'accions. Un sistema multi-agent (SMA) consisteix en ser un grup d'agents que poden prendre diferents rols segons quina sigui l'estructura organitzativa del grup. Una qualitat important d'aquests sistemes és el fet que la capacitat executiva del col·lectiu d'agents és més gran que la suma de les capacitats dels membres; és l'anomenat comportament *emergent*.

Hi ha molts treballs que mostren l'encert de les tecnologies basades en agents i, avui dia, és un fet acceptat per tota la comunitat científica que l'ús de sistemes multi-agent en el control de sistemes complexos és la millor opció. Per a la descripció de l'estructura general d'aquests sistemes, cal fer esment a dos aspectes fonamentals: d'una banda els aspectes relacionats amb el llenguatge de comunicació que permet el traspàs d'informació d'un agent als altres i viceversa, i els aspectes de comportament del col·lectiu, és a dir, com queden definides les normes o regles que regulen l'evolució temporal tant dels components individuals com del conjunt global. S'han definit moltes arquitectures i s'han construït aplicacions per a sistemes complexos però encara hi ha molts problemes associats en l'anàlisi, el disseny, la implementació i la validació (avaluació) d'aquests sistemes. Es fa evident, doncs, la necessitat d'eines específiques per a la construcció de SMA.

En el camp d'aplicació dels SMA en entorns robotitzats, els algorismes de decisió individual i col·lectiva de les arquitectures actuals no tenen en compte la dinàmica del moviment dels robots. En aquesta tesi es presenten alguns resultats encaminats a introduir aquesta dinàmica en la presa de decisions individuals i col·lectives d'aquests agents. Per fer això cal ampliar la interpretació de l'agent no ja com a agent físic sinó com a agent físic amb una dinàmica pròpia la qual fa que certes accions siguin realitzables i d'altres no. Aquesta ampliació passa per especificar quins són els elements

a introduir alhora de dissenyar un agent que sigui capaç de tenir en compte la dinàmica del propi “cos”, entenent com a “cos” l’objecte físic el moviment del qual és descriptible amb equacions diferencials.

Objectius

En aquesta tesi, la proposta d’aquestes especificacions queda emmarcada en una arquitectura multi-nivell i modular anomenada DPAA, sigles angleses que responen a “Arquitectura per a Agents Físics Dinàmics”. Es planteja com a objectiu que l’arquitectura respongui a diferents necessitats: independència del maquinari, robustesa, tolerància a fallades, fiabilitat, flexibilitat, etc. Per comprovar que els objectius s’han assolit s’ha plantejat com a requisit que l’arquitectura i la implementació puguin funcionar en dos dominis diferents utilitzant una única metodologia de disseny basada en l’estructura de l’arquitectura. La idea central d'utilitzar aquests dominis és justificar la necessitat d’una metodologia de disseny basada en una estructura jeràrquica per nivells que permeti treballar amb agents en l’àrea del control de sistemes dinàmics distribuïts. Concretament, es tracta de trobar la manera d’establir adequadament quines són les estructures de dades i llurs relacions que fan falta per connectar el món físic (real) amb el món agent (*software*), en aquest cas dedicat a les tasques de decisió cooperativa. En el camp de recerca dels sistemes multi-agent, aquest problema queda concretat en com definir la metodologia i l’estructura necessàries per poder treballar amb agents físics amb dinàmica pròpia.

El primer domini consisteix en un sistema aparentment senzill format per dos robots mòbils que es desplacen junts l’un darrera l’altre: és un comboi de dos robots. El segon exemple consisteix en un sistema multi-robot en què diferents robots mòbils es mouen en un entorn poc estructurat, els quals tenen assignada la tasca d'emular el joc de futbol. Amb el primer exemple es vol mostrar que les diferències dinàmiques entre els mòbils poden afectar a la consecució d’algunes accions i que, per tant, la dinàmica del mòbil és una component important a tenir en compte en l’etapa de decisió a nivell agent. En el segon exemple s’hi troben tots els elements necessaris per exemplificar el problema de la definició de l’estructura dels agents físics:

- El sistema necessita d'una estructura de decisió per tal de coordinar el moviment col·lectiu dels robots. Aquesta decisió s'ha de fonamentar en algorismes de decisió coordinada. L'important en aquest punt està en el fet que la decisió assignada a cada robot ha de dependre tant de les necessitats del grup com de les possibilitats d'execució pròpies del robot.
- Cada robot necessita, de forma individual, algorismes de control de moviment d'acord amb la decisió que s'ha pres. L'algorisme de control ha de ser l'adequat a cada situació o tipus de moviment. Cal remarcar que el moviment controlat que s'aconsegueixi tindrà una sèrie de paràmetres que definiran la bondat del moviment i/o les prestacions del controlador.
- La presència d'altres robots i de diversos objectes en el mateix entorn introdueix una evolució al sistema completament impredecible. Aquest fet obliga a introduir un bloc planificador on s'hi especifiquen tant limitacions d'espai com temporals. Les primeres es poden resoldre a partir d'algorismes planificadors de trajectòries lliures de col·lisió, i les temporals de forma indirecte a partir de la dinàmica dels robots.
- Cada robot es troba constantment canviant d'estat en el sentit purament cinemàtic. De tota manera, cal tenir present que un comportament determinat pot estar assignat a un robot durant un cert període de temps i que això impliqui que executi tota una sèrie d'accions diferents. Aquest fet justifica la presència d'un bloc que integri tot el coneixement necessari per gestionar la informació útil per tal de dur a terme els diferents comportaments definits prèviament.

Aquest domini és un sistema bastant complex i la programació d'un sistema multi-agent adient per aquest domini no és una tasca a realitzar per una sola persona. L'objectiu que es planteja en aquesta tesi és mostrar com es pot utilitzar la metodologia de disseny per analitzar el problema de la passada de la pilota i poder avaluar si l'acció de passar-se la pilota entre dos robots és factible o no.

Organització

Aquest treball està organitzat en cinc capítols distribuïts en dues parts ben diferenciades. La primera part consta de tres capítols i es fa una introducció als agents físics i s'hi descriuen els principals antecedents relacionats amb les aportacions del

treball. La segona part consta de dos capítols on s'hi estableixen les bases de l'aportació d'aquest treball i s'assoleixen els objectius descrits en la secció anterior.

Quant a la primera part, en el primer capítol s'introdueixen els principals conceptes relacionats amb els sistemes multi-agent i es mostra quin és l'estat actual. El capítol segon descriu els principals antecedents relacionats amb la robòtica mòbil i s'estableixen les problemàtiques associades a la cooperació entre robots, la planificació de moviments i les restriccions dels sistemes de temps real. A continuació, en el tercer capítol s'introdueix el concepte d'agent físic i les bases de la motivació d'aquesta tesi.

En el quart capítol, a la segona part, es fa una descripció de l'arquitectura DPAA. En un primer apartat es presenta la formalització de l'arquitectura i s'explica com funciona internament i quins són els cicles de funcionament. En el segon apartat es proposen les definicions de les capacitats i s'estableix formalment com s'hi inclou la dinàmica del cos físic. Consta de diversos exemples que il·lustren com aplicar el formalisme a sistemes simulats i reals. En concret, s'estudia en detall el sistema del comboi de dos robots i es mostren resultats reals.

En el cinquè capítol es mostra com s'aplica la metodologia de disseny al domini del futbol robòtic i es mostren diversos resultats obtinguts mitjançant simulació. Els resultats fan referència a la situació en què un robot ha de rebre la pilota i ha de xutar-la cap a porteria, i es mostra l'eficàcia d'utilitzar índexs per avaluar l'eficiència d'aquesta acció en concret. Al final es presenten les conclusions i la proposta de treballs futurs.

PART I:

INTRODUCCIÓ ALS AGENTS FÍSICS

1. Els sistemes multi-agent

En aquest capítol s'explica quin és l'estat de l'art en la temàtica dels sistemes multi-agent. S'expliquen els orígens de la terminologia agent, la situació actual en diversos aspectes d'aquesta tecnologia i al final es comenta la implementació AGENT-0 i l'estàndard FIPA.

1.1. Introducció: l'evolució de la I.A. cap a sistemes basats en agents

L'àrea de les aplicacions de la Intel·ligència Artificial (IA) té una història de més de 25 anys des dels primers treballs considerats com veritablement de la IA (tot i que hi ha autors que postulen que al 1996 la IA complia 40 anys des dels primers jocs basats en l'heurística implementats en computador —jocs d'escacs, entre altres). A la dècada dels 70 la recerca es va centrar pràcticament en definir el poder i les limitacions dels mètodes de la intel·ligència artificial. Així, tot i que el treball estava motivat pels problemes reals, els punts bàsics de recerca se centraven en temes com ara la representació del coneixement, l'adquisició de coneixements, el raonament causal, els mètodes de resolució de problemes, el raonament temporal, els models d'incertesa, etc. Finalment, l'àrea de les aplicacions de la IA va donar lloc a quatre departaments de computació de universitats nord-americanes: Massachussets, Rutgers, Pittsburgh i Stanford. En aquestes universitats, els metges i els informàtics varen treballar en els quatre projectes pioners de l'àrea: el "*Present Illness Program*" [Pauker, 76], CASNET [Weihmayer, 94], Internist [Miller, 82] i MYCIN

[Shoham,76]. Des d'aleshores hi ha hagut una activitat sostinguda mitjançant congressos específics, programes de recerca i revistes especialitzades. Així, en els últims anys han aparegut propostes d'arquitectures generals adequades a la construcció de sistemes intel·ligents en medicina, com KADS [Wielinga, 92], MILORD II [Sierra, 93] i *The Componential Framework* [Stels, 89]. I també es pot citar ARCHON [Jennings, 95] en aplicacions de distribució d'energia elèctrica, executat com un sistema expert distribuït que supervisa l'operació de la xarxa elèctrica a nivell de subestació (mitja-alta tensió), amb finançament ESPRIT juntament amb l'empresa IBERDROLA.

Des del principi, la IA ha tingut l'ambició de trobar mecanismes generals de raonament assolint alguns èxits tan espectaculars com el mètode GPS aplicat a la demostració de teoremes. De tota manera, la IA clàssica s'ha caracteritzat pel desenvolupament de mètodes de resolució específics segons la classe de problema a resoldre: jocs, planificació, diagnosi, etc. En dominis on la IA no ha funcionat han anat apareixent diferents maneres de simular la intel·ligència: Xarxes Neurals (NN, de l'anglès), Algoritmes Genètics (GA, de l'anglès)¹ i la Intel·ligència Artificial Distribuïda (IAD), tots basats en resoldre els problemes de forma col·lectiva.

La IAD s'interessa per dominis que consisteixen en ser situacions on la presència d'una única entitat de resolució o d'una sola màquina no han estat apropiades, i ha motivat l'aparició d'arquitectures en què l'objectiu era la resolució distribuïda de problemes basant-se en la distribució del coneixement en diferents entitats físiques o lògiques. Termes com autonomia, conflicte i negociació, i ambigüitat semàntica, apareixen en aquests tipus de dominis pel fet que aquestes entitats individuals tenen la capacitat de posseir i raonar sobre les seves creences. Per tant, la IAD estudia simultàniament tant el comportament col·lectiu com els objectius individuals de cada una d'aquestes entitats que s'anomenen **agents**. De fet, la terminologia IAD té encara en un darrer pla una concepció centralitzada de la descomposició d'una activitat col·lectiva de forma que els treballs s'orienten cada cop més en l'aspecte de l'autonomia de l'agent mantenint la cohesió de la societat mitjançant les interaccions

¹ Tant les NN com els GA estudien les condicions "d'emergència" del comportament intel·ligent que apareix en l'agregació d'entitats més simples, i que no estan directament programades per això. Ambdós mètodes es basen en tenir un coneixement bastant complet del conjunt i en un control estricte de l'organització de la col·lectivitat. El que els diferencia està en el fet que les NN es basen en interconnectar tots els membres de la col·lectivitat, mentre que en els GA tot el funcionament està basat en un control centralitzat.

agent-agent. Així, en l'actualitat, la recerca es centra tant en les capacitats d'interacció dels agents com en l'emergència de comportaments col·lectius: compatibilitat entre objectius globals i locals, aprenentatge davant noves situacions, adaptabilitat a objectius no previstos, etc.

En aquests col·lectius o sistemes multi-agent (SMA), la col·lectivitat complexa és la resultant de processos individuals simples realitzats de manera autònoma en els agents. El grau d'intel·ligència que apareix sorgeix de l'aptitud individual de l'agent de raonar amb les creences (i per tant, amb les interaccions) per tal d'adquirir nous coneixements i transformar l'organització del conjunt d'agents. D'aquesta manera, el comportament d'un SMA es pot entendre com a una cerca d'objectius distribuïts en un espai d'estats. Des d'aquest punt de vista, els algorismes centralitzats clàssics no es poden aplicar integralment sense incloure-hi una dimensió que correspongui a la comunicació entre els agents. Igualment, les eines de IA relacionades amb el manteniment de la coherència també es poden incloure si es fan algunes transformacions. L'important dels SMA roman en el fet que pressuposen que la intel·ligència sorgeix de l'agregació de competències simples i que la tasca assignada a cada individu és tant important com la tasca col·lectiva². Per tant, **el procés de desenvolupament d'una aplicació multi-agent s'ha de centrar en la modelització i en la validació de les competències d'un agent particular, i s'ha de centrar en aspectes més socials com ara els mètodes d'interacció agent-entorn i agent-agent**: cal seleccionar diversos tipus d'actituds socials.

El problema que hi ha en les aplicacions que utilitzen SMA és que hi ha un gran salt entre els treballs teòrics en què s'estudien els models de col·laboració i les implementacions en sistemes reals en què se solen utilitzar models *ad-hoc* d'interacció social. Tot plegat fa que apareguin moltes aplicacions en àmbits cada cop més diversos i ha provocat una gran diversitat de noms i tipus d'agents. Sigui com sigui, es pot dir que les aplicacions basades en agents són precursors d'un nou paradigma per a desenvolupar i implementar Sistemes Complexos (COSY, en anglès). Aquest paradigma implica la construcció de components sofisticats i auto-continguts, i que poden interaccionar flexiblement amb un nombre indeterminat de components similars. Aquesta interacció no pot fer-se a través d'interfícies rígides i predeterminades, sinó que cal fer-la a través de negociació i persuasió, i seguit per

² A tall d'exemple, en les NN cap neurona té una tasca precisa assignada i la tasca que realitza només té sentit dintre de la col·lectivitat

compromisos i agraïments entre les parts. El potencial de cadascun d'aquests components fa que les propietats del sistema global i del seu comportament no puguin ser pre-programades sinó que emergiran soles a mida que les interaccions entre els agents es vagin desenvolupant. El problema que tant el comportament com les propietats del sistema no puguin ser fixades alhora de dissenyar els agents, afegit amb el fet que no es pot garantir que les dependències entre els agents es puguin gestionar efectivament, posa un límit en els tipus d'aplicacions en què es puguin usar els agents.

Malgrat ja s'han desenvolupat alguns sistemes intel·ligents basats en agents i s'han fet prototipus avançats d'aplicació real, encara hi ha problemes a resoldre. Des del punt de vista de disseny de sistemes en podem destacar els següents:

◆ Desenvolupar una metodologia per dissenyar agents i sistemes basats en agents

Manca una metodologia per dissenyar els agents de forma que siguin suficientment robustos i flexibles, de forma que es puguin explotar al màxim i que donin lloc a un sistema global igualment robust i flexible. El problema d'aquesta metodologia està en el fet que alguna de les propietats de l'agent es veuen un cop el sistema està funcionant i els mètodes proposats esdevenen complexos i difícils d'implementar.

◆ Desenvolupar *benchmarks* per tal d'avaluar els paràmetres del disseny

La utilització dels agents en sistemes complexos és de moment la sortida més raonable. El problema que es presenta està en que són sistemes amb grans quantitats de dades i això significa que el dissenyador té poc coneixement sobre quins mecanismes funcionen bé i en quines condicions: no té prou coneixement amb què basar-se per tal de dissenyar-lo bé. El que faria que el disseny obtingués agents ben informats serien tot un seguit de *benchmarks* per tal de testar el disseny, i es podrien comparar diferents alternatives o, fins i tot, fer-les competir per comparar-les de forma objectiva.

◆ Desenvolupar eines reutilitzables

Els SMA requereixen un alt nivell d'infraestructura bàsica abans de poder estar operatius: els mecanismes i protocols de comunicació per intercanviar missatges, la capacitat per treballar sobre diferents plataformes, i les eines de monitorització i *debugging*. Normalment aquestes eines s'han de desenvolupar de nou per a cada nova aplicació i això suposa una pèrdua de temps i un cost econòmic que podria obviar-se amb la disponibilitat d'eines reutilitzables. D'aquesta manera, els temps dedicats a la implementació es podria dedicar íntegrament al disseny dels agents.

Igualment, des del punt de vista de disseny de l'agent en si, no hi ha encara una metodologia per obtenir-ne el màxim aprofitament així com tampoc per assolir tot el potencial que pot oferir el paradigma agent en els següents aspectes:

✓ Heterogeneïtat

La recerca sobre agents comença tot just a tractar amb els problemes associats amb la inevitable heterogeneïtat dels seus components. El problema bàsic que es planteja està en com fer que interaccionin els agents de forma efectiva i escalable davant diferents dominis, o utilitzant diferents esquemes de representació de coneixement, o utilitzant diferents paradigmes de resolució, o fins i tot, treballant amb diferents esquemes del seu món o dels altres agents.

✓ Raonament amb informació incerta, incompleta i contradictòria

Pel fet que els agents tenen una visió parcial del seu món i com que el domini del problema que tenen és obert i distribuït, necessiten mecanismes sofisticats de raonament a partir d'informació incerta, imprecisa, incompleta i contradictòria. Aquest tipus d'informació pot aparèixer per la pròpia naturalesa del problema a resoldre com de la pròpia interacció entre els diferents agents.

✓ Operabilitat en temps real

La tecnologia agent s'està aplicant cada cop més en dominis on el temps és crític i adquireix una gran importància el fet que el sistema pugui respondre en temps real davant situacions que poden ser claus. A més, fa falta que el resultat de les interaccions socials entre agents autònoms sigui més previsible o, fins i tot, més influenciable a través d'algun agent amb més jerarquia o través de tasques de prioritat alta.

✓ Adaptabilitat

Finalment, és important que en entorns on l'agent està típicament situat pugui adquirir més informació i pugui entendre millor el seu paper dintre del procés de resolució del problema. D'aquesta manera l'agent podria adaptar-se (en definitiva, aprendre) i modificar el seu comportament per tal de resoldre el seu problema de forma més efectiva.

Avui dia encara resulta difícil la implantació de les tècniques de IAD en els entorns de control automàtic i sistemes autònoms (robots). Una explicació d'aquesta dificultat d'implantació és la constatació de la dificultat de realitzar recerca en

intel·ligència artificial en la robòtica mòbil i en sistemes multi-robot. En l'actualitat els treballs s'entenen com a elements de l'àrea més ampla del control automàtic i IA. Per tant, els investigadors en IAD coneixen una àrea més ampla i estan disposats a treballar en la seva integració amb altres sistemes informàtics. La tendència està en la definició d'eines o components de programa, que permetin desenvolupar aplicacions robòtiques a través d'entorns informàtics agradables.

1.2. L'estat actual dels sistemes multi-agent

1.2.1. La tecnologia agent

El terme agent s'està utilitzant cada cop més en les tecnologies de la informació aplicant-lo per a descriure un marge molt ampli d'entitats computacionals. Aquestes entitats van des de sistemes relativament simples com ara sistemes que filtren el correu electrònic [Maes, 94a] fins a grans sistemes que interconnecten sistemes experts o bases de dades amb grans quantitats de codi [Jennings, 95]. D'una forma poc formal, es poden identificar tres tipus d'agents. En un nivell simple, hi ha agents que poden executar tasques molt concretes i basades en un seguit de regles i suposicions pre-establertes. En un segon nivell de sofisticació, tenim els agents que executen una sèrie de serveis, és a dir, executen tasques ben definides i d'alt nivell quan l'usuari ho demana. Finalment, hi ha els agents que voluntàriament donen informació i servei a l'usuari sense que hi hagi hagut cap petició, però en el moment on pot ésser apropiat.

Per tant, donat que hi ha aquesta diversitat, és bo concretar quins són els trets fonamentals que identifiquen aquests sistemes. Tot i que no existeix cap caracterització universalment acceptada, es pot dir que **"un agent consisteix en una entitat autocontinguda que sap resoldre problemes, i que gaudeix de les següents propietats"** [Wooldridge, 95]:

- **Autonomia** : els agents han de ser capaços de realitzar les tasques relacionades amb la resolució del seu problema sense la intervenció directe de cap altre tipus d'agent (humà?), i han de tenir algun grau de control tant de les seves accions com del seu estat intern.

- **Capacitat Social** : han de tenir la capacitat d'interaccionar amb altres tipus d'agents o humans per tal de poder assolir la resolució del seu problema i per ajudar als altres en llurs activitats. Això requereix per part seva que, com a mínim, disposin de mecanismes de comunicació amb els altres per tal de demanar-los-hi requeriments, i de mecanismes interns per a decidir quan és que cal interaccionar. Ambdós mecanismes tenen la funció de generar requeriments i de qüestionar els dels altres.
- **Responsabilitat** : han de saber percebre el seu entorn per tal de reaccionar a temps davant dels canvis que puguin produir-s'hi. Aquest entorn pot ser tant el món real, com un usuari d'ordinador, com un col·lectiu d'agents, Internet, etc.
- **Capacitat d'Acció** : han de saber reaccionar no només com a acte reflex sinó també tenir un comportament amb iniciativa i persistència davant d'un objectiu prèviament establert.

A part d'aquestes propietats necessàries, n'hi ha d'altres que poden afegir-se però només són importants en dominis d'aplicació específics si bé també poden afegir-se a les anteriorment esmentades ja que també poden ser útils. Per exemple, es pot parlar de les capacitats d'adaptabilitat, mobilitat, veracitat, i racionalitat proposades per Wooldridge i Jennings [Wooldridge, 95] o per Franklin i Graesser [Franklin, 96]. Tot aquest conjunt de característiques fan que la tecnologia basada en agents es pugui enfrontar davant problemes on altres tecnologies han fracassat per motius relacionats tant a la complexitat del problema com a la falta d'una solució apropiada.

1.2.1.1. Camps d'aplicació

La tecnologia basada en agents té un paper molt important a jugar en el desenvolupament d'aplicacions informàtiques, però cal tenir present que no tot problema difícil de resoldre pot basar-se en agents per a resoldre'l. Per exemple, moltes aplicacions que surten ara basades en agents no són la millor solució per a resoldre el problema que plantegen, però el simple fet que un sistema sigui obert, o complex, o distribuït (tant per les dades com pels recursos) fa que es pugui plantejar basat en agents.

- **Agents als problemes oberts**

Quan els components d'un sistema no són coneguts a priori, poden variar en el temps, i són extremadament heterogenis, diem que és un problema obert. Com a exemple clar podem anomenar la xarxa Internet que contínuament està expandint-se i introduint eines cada cop més complexes. En un entorn com aquest, com que els components han d'interaccionar amb d'altres que no són coneguts a priori, els agents actuen com a elements d'interfície i les fan flexibles i robustes.

La introducció de la tecnologia agent a Internet és bastant profunda i ja es parla d'agents web [Petrie, 96] i fins i tot la versió 3.0 del protocol estàndard HTTP inclou funcions més avançades útils per a la comunicació entre agents. Tot i així, la introducció d'agents a la *web* encara no està resolta a causa de l'estructura de la informació: el llenguatge HTML descriu formats mentre que els agents necessiten d'una semàntica basada en tasques. Una altra gran diferència està en el fet que la *web* està basada en una estructura client-servidor, que no deixa de ser una trivialització del procés de col·laboració entre agents, i això fa que la *web* esdevingui una tecnologia per sota de les possibilitats de la tecnologia agent. De tota manera hi ha treballs que intenten apropar les dues tecnologies com pot ser l'estructura *Java Agent Template* (JAT) de Robert Frost que ha definit agents en Java dotant-los de capacitat de comunicació en llenguatge KQML [Finin, 93,94].

En sistemes oberts, un agent ha de tenir moltes capacitats socials a través de mecanismes de comunicació que li permetin transmetre informació de manera comprensible. A un altre nivell d'importància, un agent també necessita monitoritzar el que està passant tot i que normalment no serà capaç de predir què pot passar a curt termini per tal d'adaptar-s'hi amb anterioritat.

- **Agents i la complexitat**

La complexitat apareix en dominis grans i sofisticats, o impredecibles, on la única forma de tractar el problema és desenvolupant un nombre determinat de components modulars especialitzats ja sigui a la resolució d'una part del problema com a la seva representació. En aquest cas, quan apareixen problemes interdependents, els agents necessiten interaccionar entre ells i assegurar-se de que les dependències que tenen estan ben gestionades.

Exemples reals d'aquest tipus de dominis en què s'han desenvolupat sistemes basats en agents són : el control d'un accelerador de partícules [Jennings, 93a], la

gestió d'una xarxa de telecomunicacions [Weihmayer, 94], o la manufactura íntegrament computeritzada [Parunak, 95]. En aquests dominis la visió agent implica que tot el problema queda subdividit en parts petites i simples, que són fàcils tant de desenvolupar com de mantenir, i que són especialistes en resoldre subproblemes de forma molt efectiva. Aquesta descomposició permet que cada agent utilitzi el paradigma més apropiat per resoldre el seu problema particular.

- **Sistemes amb distribució de dades, del control, i dels recursos**

La tecnologia agent s'aplica en dominis on els recursos necessaris pel càlcul estan físicament o lògica distribuïts, o en dominis on diferents entitats necessiten comunicar-se però tenen una representació diferent de l'entorn com, per exemple, en la comunicació home-màquina [Suzuki, 95]. En tots aquests casos la resolució basada en agents proporciona una forma natural de modelar el problema perquè es tracta d'entitats reals que resolen problemes localment i que necessiten interaccionar, és a dir, transmetre informació, ordres i resultats.

Un tipus d'aplicació real consisteix en la gestió dels recursos d'un hospital on es presenta un problema de gran complexitat: es tracta de gestionar infermeres, practicants i especialistes, amb pacients i habitacions, tot desenvolupat en un entorn on el factor temps és important. En aquest sentit un dels darrers treballs importants [Huang, 95] resol el problema contemplant-lo des del punt de vista de distribució de dades, control, coneixement i recursos. Distribució de dades en el sentit que tant les infermeres com els practicants com els especialistes tenen les seves agendes de treball i que s'han de coordinar amb pacients diferents o no. Distribució de control en el sentit que cada treballador és responsable de la realització de les tasques que té assignades. Distribució de coneixement en el sentit que la informació que s'intercanvien aquests treballadors varia en funció de la especialització que tenen, i distribució de recursos en el sentit que un especialista és responsable dels llits dels seus pacients, el practicant és responsable del material que utilitza, etc.

Des d'un punt de vista més industrial, un bon exemple és la gestió de dades subministrades per una xarxa de sensors [Lesser, 83] o en el control del moviment d'un robot amb un centenar de sensors, 19 graus de llibertat i 8 processadors interconnectats [Ferrel, 96]. En aquests casos la utilització d'agents significa que el processament es pot realitzar localment allà on es generen les dades només si hi ha un intercanvi d'informació a alt nivell. D'aquesta manera la tasca d'enviar moltes

dades cap un processador central distant queda alleugerida i pot optimitzar-se l'ús de l'ampla de banda de què es disposa per a la comunicació.

- **Presència d'agents en eines de programari**

En els darrers anys ha aparegut la noció d'agents autònoms relacionada amb tot aquell tipus de programari que dóna servei a l'usuari d'acord amb unes especificacions personalitzades que reflexen les preferències de cada usuari. En aquest sentit, un programa que filtra el correu electrònic es pot presentar com a agent autònom assistent [Maes, 94] igual com un programa que planifica una agenda [Jennings, 95a], o inclòs en sistemes de realitat virtual on els personatges consisteixen en ser entitats individuals autocontingudes que resolen problemes [Bates, 94].

En un altre aspecte, també hi ha molts programes que necessiten restaurar-se periòdicament a mida que apareixen noves utilitats, i una manera de possibilitar que aquests canvis no resultin costosos és incorporant mòduls que s'hi connectin de manera que funcionin en forma d'interfície o inclòs que permetin la connectivitat amb altres programes. En aquest cas estariem parlant de comunitats d'agents que cooperen per tal de traspasar-se informació [Papazoglou, 92] de forma que els agents "recobreixen" individualment els seus programes i es comuniquen entre ells per tal de fer funcionar els programes correctament.

1.2.1.2. Plataformes de disseny d'agents

El procés anomenat de *disseny* ha de passar per les següents etapes:
(aspectes metodològics)

Anàlisi → Descripció del problema i delimitació del domini. Identificació del domini d'aplicació i de la clau del problema

Disseny → Procés on es defineix l'arquitectura solució del problema, d'una forma declarativa. Consisteix en especificar el principi de la solució del problema.

(aspectes tècnics)

Desenvolupament → Procés de construcció d'una solució funcional. Consisteix en obtenir el codi de la solució emprant un llenguatge determinat.

Implementació → Posta en marxa de la solució al problema real. Consisteix en instal·lar el software a la xarxa i mantenir-ne la funcionalitat i/o estendre-la.

A més, el procés de disseny no és senzill donat que hi ha molts elements que hi intervenen. En concret, es poden enumerar els següents:

- (i) el model de l'agent: comportament, aprenentatge, aplicacions
- (ii) el tipus d'arquitectura: reactiva, deliberativa, horitzontal, vertical
- (iii) els models computacionals: concurrència, comunicació, control, modularitat
- (iv) la base teòrica: llenguatge simbòlic, la base temporal, raonament
- (v) els mecanismes d'inferència, de decisió, i de planificació
- (vi) la implementació: plataforma del *host*, sincronisme dels processos

Actualment resulta complicat seleccionar l'eina de disseny d'agents perquè cada eina es crea seguint un paradigma i/o arquitectura específica, i s'implementen amb solucions pragmàtiques diferents. Amb tot, és difícil saber quina és la millor solució. En aquesta secció es mostraran els resultats d'una anàlisi basada en els següents criteris de qualitat i que serviran per avaluar totes les etapes que apareixen durant la creació d'un agent:

- a) **Completitud** → grau de cobertura que dona la plataforma: quantitat i qualitat de documentació, i eines que incorpora.
- b) **Aplicabilitat** → possibilitats i restriccions ofertes per la plataforma.
- c) **Complexitat** → dificultat per finalitzar el procés associat a l'etapa: quantitat de feina a fer i la seva dificultat.
- d) **Reutilitzable** → quantitat de feina estalviada alhora de reutilitzar treballs anteriors.
- e) **Suport** → possibles versions futures de la plataforma, s'utilitza a gran escala.

Les plataformes que s'han escollit són AgentBuilder®, Jack™, MadKid i Zeus, i l'elecció es basa en el fet que són populars, utilitzen models acadèmics ben coneguts, estan desenvolupades amb qualitat industrial, i són senzilles de fer funcionar:

AgentBuilder® : <http://www.agentbuilder.com>

Eina per a construir agents software intel·ligents. Es basa en els models Agent0 [Shoham, 90] i Placa [Thomas, 93]. Desenvolupada per Reticular Systems Inc.

Jack™ : <http://www.agent-software.com.au>

Entorn per a construir, executar i integrar SMAs comercials basats en JAVA, utilitzant *component-based approach*. Es basa en el model dMARS [d'Inverno, 98]. Desenvolupada per *Agent Oriented Programming Pty. Ltd.*

MadKid : <http://www.madkit.org>

Runtime engine plataforma per a multi-agents JAVA. Es basa en un *organizational model*. Desenvolupada per O.Gutknecht i J.Ferber del LIRMM (*Lab d'Informatique, de Robotique et de Microélectronique* de Montpellier).

Zeus : <http://www.lab.bt.com/projects/agents/zeus>

Entorn integrat per a la construcció d'aplicacions amb agents col·laboratius. Es basa amb agents basats en regles. Desenvolupat per el *Agent Research Programme* del *British Telecom Intelligent system Research Lab*.

La següent taula mostra les valoracions que s'han fet des de diferents surveys [Khalil, 00]. És de remarcar l'alta puntuació que es dona a Zeus, una plataforma de recent creació, i que està a l'alçada de la popular AgentBuilder:

| Etapa → | Anàlisi | | | | Disseny | | | | Desenvolupament | | | | Implementació | | | | Su |
|--------------|---------|---|----|---|---------|---|----|---|-----------------|---|----|---|---------------|---|----|---|----|
| | Ct | A | Cx | R | Ct | A | Cx | R | Ct | A | Cx | R | Ct | A | Cx | R | |
| AgentBuilder | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Jack | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| MadKit | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Zeus | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |

Ct: Completitud; A: Aplicabilitat; Cx: Complexitat; R: Reutilitzable; Su: Suport
Codi Valoració: ■ Bona ■ Mitja □ Cap

Taula 1.2.1.2: Valoracions de diverses plataformes de disseny d'agents

1.2.2. Els sistemes multi-agent

Per a la descripció de l'estructura dels SMA, cal fer esment a dos aspectes fonamentals:

- Aspectes de comportament del col·lectiu, és a dir, com queden definides les normes o regles que regulen l'evolució temporal tant dels components individuals com del conjunt global.

- Aspectes relacionats amb el llenguatge de comunicació que permet el traspàs d'informació d'un agent als altres, i viceversa.

Quan es desenvolupa una aplicació multi-agent l'atenció cal centrar-la en la modelització i en la validació de les competències d'un agent particular. Així, és d'esperar que quan aquest agent estigui en actiu faci que el seu entorn immediat es vegi modificat pels seus "moviments". Ara bé, alhora, aquests moviments no poden realitzar-se de forma independent al comportament de la col·lectivitat. Per tant, hi ha una sèrie de fets inherents al fet de treballar amb col·lectius d'agents (generació de conflictes, coneixements contradictoris entre agents,...) que fan que l'atenció en la fase de disseny també s'hagi de centrar en aspectes socials per tal de garantir la integritat del col·lectiu. En realitat, aquest aspecte social de l'agent que el lliga estructuralment al col·lectiu pot fer que les creences (coneixements particulars) que té es vegin afectades en un grau més o menys elevat. Per tant una altra qüestió que cal plantejar-se en l'etapa de disseny està relacionada amb quina ha de ser la necessitat que pot tenir l'agent de revisar les seves creences i en quin grau ha de fer-ho [MARCIA, 93]. Així, tant les creences com el coneixement segueixen camins diferents però co-dependents: l'agent ha de ser capaç de distingir i percebre els senyals de l'entorn i saber reconèixer d'on venen.

La forma de resoldre aquesta qüestió és estudiar primer les teories lògiques d'intenció i d'acció per saber com s'articulen els coneixements amb les creences [Hintikka, 62] i després cal estendre-les a sistemes de diversos agents per veure les propietats que apareixen [Wooldridge, 94a]. La dificultat d'aplicabilitat directe en aplicacions reals fa que una forma més pragmàtica consisteixi en separar físicament les creences del coneixement per tal de raonar de forma separada. Així, una creença pot ésser avaluada en termes de coneixement i de veritat, i pot passar a convertir-se en coneixement sempre i quan sigui coherent amb l'estat actual [SMI, 94]. D'aquesta manera l'agent es pot construir una teoria de l'entorn a partir d'accions reactives, que són les que produeixen les creences— aquestes accions també s'anomenen actes reflexes [Cattoni, 93] en el sentit que no disposen de representació simbòlica ni contenen cap funció de reflexió. Per tant, quan l'agent revisa les creences i les compara amb el coneixement que disposa pot eliminar situacions ambigües i tractar els conflictes de forma que globalment augmenta la qualitat de la resolució [SMI, 94]. Ara bé, aquesta revisió pot tenir molts graus de complexitat i fa que hi hagi tota una gradació des dels agents reactius purs, que només es mouen

davant d'estímuls, fins els agents totalment cognitius [Decker, 94], que disposen d'un model actualitzat simbòlic de l'entorn que els permet planificar les seves accions.

En aquest punt hi ha una gran diversitat d'opinions relatives al grau de necessitat de la comunicació entre els agents d'una col·lectivitat, però en línies generals pot concloure's que aquestes necessitats de comunicació estan en funció del problema concret a resoldre i de les propietats dels agents que constitueixen el col·lectiu. Ara bé, des del moment que apareix la paraula creença en la terminologia, es fa palesa la necessitat que té un agent de treballar amb els altres, és a dir, de comunicar-se o cooperar.

1.2.2.1. Comportament cooperatiu per a la resolució de problemes

El comportament cooperatiu és present en tot el regne animal la qual cosa indica que la generació de comportament cooperant pot ser relativament simple, és a dir, que no fa falta més intel·ligència a nivell local per tal d'actuar de forma cooperativa. Per tant, no cal considerar-lo com a un comportament complex, com pot ser l'humà, alhora de trobar solucions viables per realitzar models cooperatius sobre agents autònoms.

La cooperació consisteix en la integració de les accions de diferents agents autònoms per tal d'assolir un objectiu comú. Aquesta cooperació pot ser útil per a problemes on la resolució comporta dificultats relacionades amb la simultaneïtat tant espacial com temporal, i amb dificultats intrínseques a la complexitat del problema [Franklin, 87]. La simultaneïtat fa referència al fet que un agent no pot estar fent més coses de les que pot, o al fet que dos agents no poden estar fent la mateixa feina, i la complexitat es refereix al fet que un problema pot subdividir-se en diverses parts i repartir-se en diversos agents especialistes en una sola tasca o distribuïts de forma jeràrquica.

Tot i la gran diversitat de tipus de problemes aptes per resoldre a través de la cooperació, hi ha una sèrie d'elements comuns en qualsevol sistema cooperant:

Estratègia: El conjunt de regles d'operació juguen un paper important perquè s'estableixen abans de que el sistema comenci a funcionar i pre-defineixen el tipus de comportament cooperant. Per tant, s'ha de procurar que les regles siguin independents del context (almenys parcialment).

Evolució en el temps i en l'espai: La cooperació es desenvolupa amb accions correlacionades, és a dir, de forma seqüencial i en un interval de temps (o espacial). Aquesta seqüencialització determina una finestra de coordinació [Steels, 84] que ve determinada tant per les característiques dels agents com pels requeriments d'una tasca específica.

Flux d'informació: La forma de transmetre informació pot ser a través de comunicació o d'inferència, però ambdues categories poden estar presents alhora. Amb la primera la informació es transmet o es rep a partir d'un procés de diàleg, mentre que amb processos d'inferència és l'agent qui obté la informació mitjançant observació dels altres i/o de si mateix.

Per tant, la forma apropiada de resoldre un problema mitjançant la cooperació vindrà determinada tant pel grau com pel tipus d'especialització dels agents cooperants. El tipus de cooperació pot classificar-se tenint en compte l'estructura del col·lectiu, de forma que es pot parlar d'estructures anàrquiques, heteràrquiques, col·laboratives, jeràrquiques flexibles o fixades. La forma d'escollir el model de cooperació dependrà tant del grau de centralització necessària per a controlar el grup com de les capacitats de què disposin els agents, tal i com es mostra a la figura 1.2.2.1.

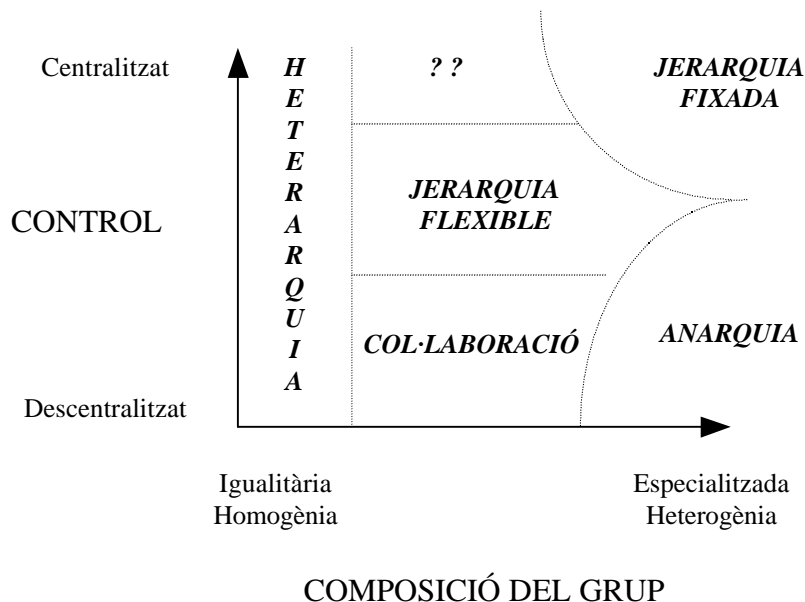


Fig.1.2.2.1.: Tipus de cooperació en diferents condicions de control i en funció del tipus de població [Franklin, 87]

La recerca de principis i tècniques per a coordinar sistemes multi-agent ha donat lloc a models formals que es comentaran a la següent secció, i també han sorgit mètodes basats en l'estructura i les característiques de l'entorn [Decker, 95], o centrats en la resolució de problemes on la solució consisteix en un conjunt de solucions que compleixen uns requisits mínims— *constraint satisfaction problems* [Jyishane, 94].

Gràcies a la cooperació, el coneixement global del sistema és més elevat i pot utilitzar-se aquest fet en aplicacions on la seguretat és important: en sistemes de control i supervisió (sistemes potencialment perillosos), en sistemes detectors de falles, i per a sistemes que ofereixen serveis segurs (per exemple, la gestió de sistemes informàtics o al control de trànsit aeri).

1.2.2.2. *Avaluació dels sistemes multi-agent*

Actualment no existeix cap eina o metodologia per avaluar si un SMA evolucionarà de forma correcta. Hi ha alguns intents com [Müller, 96] en què associa una component estocàstica al comportament de l'agent la qual cosa permet fer una anàlisi matemàtica utilitzant cadenes de Markov. Els resultats són molt prematurs i manquen de generalitat ja que fan l'anàlisi en situacions molt específiques. La recerca actual es centra fonamentalment en usar eines potents de simulació i en proposar diverses mesures *a posteriori* sobre el sistema. Fan diverses simulacions i es comparen els resultats fent diversos canvis en els paràmetres que configuren el sistema: l'amplada de banda del canal de comunicació [Suzuki, 95][Chan, 90], la quantitat d'agents individuals i el grau d'homogeneïtat/heterogeneïtat dels components del col·lectiu [Arkin, 93], o el grau de dificultat del problema [Wang, 93][Ueyama, 93][Ota, 95]. Aquests tipus de treballs tenen el problema de la manca de generalitat i mostren resultats en dominis molt concrets. Aquesta manca de generalitat ve causada per un problema intrínsec dels SMA, i és el seu alt grau de complexitat.

Un altre fet que dificulta l'anàlisi dels SMA és el grau d'autonomia dels propis agents. En general poden distingir-se dos tipus d'agents: reactius (poc o cap procés de raonament, són instintius), deliberatius (amb un alt grau de raonament). Quan la col·lectivitat està formada per individus reactius els intents com el de [Müller, 96]

poden ser adequats, però quan el grau de raonament és alt el comportament és poc previsible i cal recórrer sempre a l'ús dels simuladors.

1.2.2.3. Situació Actual

Per donar una visió clara de quin és l'estat actual de la tecnologia agent es farà un repàs de les contribucions més destacades en els cinc nivells en què es pot fer referència a l'hora de treballar amb agents, és a dir: a nivell de teories formals, a nivell d'arquitectures, a nivell de llenguatges de programació, a nivell d'aplicacions, i a nivell de plataformes.

1.2.2.3.1 Teories formals

Són bàsicament especificacions formals, és a dir, estudis formals sobre les propietats dels agents, la seva representació i com raonar sobre elles. La principal línia d'investigació que s'ha seguit és la de considerar als agents com a *sistemes intencionals* [Seel, 89], dotant-los de creences, desigs i comportament racional [Dennett, 87] [Rosenschein, 86] [Shoham, 90]. A partir de les propostes inicials de representar coneixement i creença basades en la semàntica de móns possibles [Hintikka, 62] [Kripke, 63], han aparegut diverses alternatives per a resoldre, d'entre altres problemes, les dificultats de l'anomenada "omnisciència lògica" [Levesque, 84] [Konolige, 86]. Una alternativa més radical és la basada en l'ús de metallenguatges [Perlis, 85,88] [Turner, 90] que, tot i que des del punt de vista formal pot arribar a ser problemàtic, és molt atractiu des del punt de vista de l'enginyeria. La representació d'objectius i desigs ha tingut enfocaments semblants, també basats en la semàntica de móns possibles. En aquesta línia, un dels treballs més importants és el de Cohen i Levesque [Cohen, 90] sobre la formalització lògica d'una teoria sobre intencions, definides a partir de dues primitives bàsiques: la creença i els objectius. Aquesta lògica ha estat també usada per a l'anàlisi de conflictes i la cooperació en sistemes multi-agent així com en la resolució cooperativa de problemes. El treball de Rao i Georgeff [Rao, 91,93], a diferència de l'anterior, defineix una teoria basada en tres modalitats primitives (anomenades categories mentals): creences, desigs i intencions (formalisme BDI). El formalisme es basa en un model temporal ramificat cap el futur. La definició de lògiques

deontiques ha estat una altra àrea molt activa en els darrers anys [Belnap, 93, 88]. De totes maneres, les teories més rellevants neixen de l'ús combinat de les anteriors propostes, és a dir, sense centrar-se únicament en un sol aspecte de les capacitats de l'agent. Entre altres treballs que segueixin aquesta línia poden destacar-se els treballs [Moore, 90] [Jennings, 92] [Singh, 90] [Wooldridge, 92].

1.2.2.3.2. Arquitectures d'agent

Representen el pas de les especificacions del punt anterior a les implementacions, és a dir, com construir sistemes computacionals reals que tinguin en compte les teories o els models particulars d'agents. Pel que fa a aquesta qüestió, hi ha hagut tres grans línies de treball:

1. *Arquitectures Deliberatives*: són arquitectures que contenen un model simbòlic, representat de forma explícita, de l'entorn en què les decisions es prenen mitjançant raonament (pseudo-) lògic, basat en manipulacions simbòliques. El principal problema es troba en la dificultat de poder fer deducció de forma automàtica, encara que sigui en lògiques relativament simples. Algunes arquitectures en aquesta línia són IRMA [Bratman, 88], HOMER [Vere, 90] i GRATE [Jennings, 93]. Aquesta última és una arquitectura estratificada on els agents es divideixen en dues parts: un sistema a nivell de domini i un sistema de cooperació i control. El sistema de control s'encarrega de coordinar les activitats de l'agent a nivell de domini amb les dels altres agents de la comunitat. En el sistema de cooperació s'especifica com ha d'interactuar l'agent dintre un marc de resolució cooperativa de tasques, implementant un model teòric particular.
2. *Arquitectures Reactives*: són arquitectures que no inclouen cap tipus de representació simbòlica del món i no s'usa cap tipus de raonament simbòlic, i han estat proposades com a alternativa per evitar els seriosos problemes que planteja la IA simbòlica [Brooks, 86][Steels, 90][Chapman, 86][Maes, 89]. Per exemple, Maes ha desenvolupat una arquitectura en què un agent es defineix com a un conjunt de mòduls, i on cada mòdul ve determinat per una sèrie de *pre* i *post* condicions i per un nivell d'activació que indica el grau de rellevància del mòdul en una situació particular i, per tant, el grau en què aquest mòdul influirà en el comportament de

l'agent. Finalment, aquests mòduls s'integren en una xarxa d'activació on els mòduls s'interconnecten d'acord a les seves *pre* i *post* condicions.

3. Arquitectures Híbrides: intenten recollir trets dels dos tipus anteriors. Exemples d'aquest enfoc poden ser PRS [Georgeff, 87], TouringMachines [Ferguson, 92], Interrap [Müller, 94] i COSY [Burmeister, 92]. Les arquitectures híbrides constitueixen una àrea molt activa de desenvolupament, encara que la seva potencial dificultat està en què tendeixen a ser massa *ad hoc*.

1.2.2.3.3. Llenguatges d'agents

Un llenguatge d'agents consisteix en ser un sistema que permet programar sistemes computacionals, *software* o *hardware*, en termes d'alguns conceptes desenvolupats en teories o models formals d'agents. Els predecessors d'aquests llenguatges són els llenguatges concurrents orientats a objecte, com ara *Actor model* [Hewitt, 77] [Agha, 86]. El primer que va proposar un nou paradigma per a la programació d'agents fou Shoham [Shoham, 90]. La idea principal ja u en el fet que cal programar agents directament en termes de nocions mentalistes i intencionals. Shoham proposa un paradigma de programació orientat a agent (AOP, *Agent Oriented Programming*) basat en tres components: un sistema lògic per a definir estats mentals, un llenguatge de programació interpretat, i un procés "d'agentificació" per compilar programes d'agent en sistemes executables de baix nivell. Alguns intents de desenvolupar llenguatges per a agents són Agent0 [Shoham, 90], Concurrent METATEM [Fisher, 94], APRIL [McCabe, 94], OASIS [Cheong, 92], MAIL [Haugeland, 94] i AgentSpeak(L) [Rao, 96].

1.2.2.3.4. Aplicacions

La implementació d'aquestes aplicacions és relativament recent tal i com ho demostra el fet que el primer congrés internacional d'aplicacions pràctiques basades en agents (*Practical Applications of Intelligent Agents and Multi-Agents: PAAM'96*) es va celebrar l'Abril del 1996. Les principals àrees en què s'han aplicat les tecnologies de sistemes multi-agent són:

1. Resolució Cooperativa de Problemes i Intel·ligència Artificial Distribuïda: existeixen diverses aplicacions en gestió de plantes elèctriques, control de trànsit aeri, recuperació intel·ligent de documents i gestió de xarxes de telecomunicació, treballs presentats tots ells a l'edició especial sobre agents intel·ligents a la revista *IEEE Intelligent Systems* el Desembre del 1996.
2. Agents Interfície: sistemes que utilitzen tècniques d'intel·ligència artificial per proporcionar ajuda a usuaris d'una aplicació informàtica determinada. Existeixen moltes aplicacions tant a nivell de prototipus com comercials, com per exemple el sistema NEWT [Maes, 94], o un kernel de seguretat anomenat Scheme 48 [Rees, 96].
3. Agents especialitzats en Informació i Sistemes d'Informació Cooperatius: sistemes que tenen accés a fonts d'informació i que són capaços de sintetitzar/filtrar i manipular la informació obtinguda d'aquestes fonts per donar resposta als requeriments d'usuaris i d'altres agents. Exemples d'aquests sistemes poden ser IRA [Voorhees, 94] i CARNOT [Hunhs, 92].

1.2.2.3.5. Plataformes

Les plataformes consisteixen en entorns experimentals (simulats o reals) que permeten experimentar de forma controlada. Les plataformes de recerca per IAD estan orientades o bé per a proveir d'entorns de treball per construir agents o bé per fer interaccionar agents en móns simulats o reals.

| Sistema | Objectiu de Recerca | Sensors/Actuadors | Monitorització |
|------------|---|---|---|
| DVMT | Problemes generals de IA distribuïda (IAD) | L'exactitud i rang dels sensors és variable | Fa estadístiques temporals i numèriques |
| Phoenix | Agents en móns complexos | La sensitivitat dels sensors és ajustable | Mostra la posició dels agents |
| Truckworld | Comportament d'agents reactius en móns dinàmics | Exactitud variable. Els sensors recullen tipus d'informació variada | Mostra la posició dels agents |
| MAGES | Interaccions entre agents heterogenis | Simulat amb missatges de l'entorn | Mostra la posició dels agents |
| RATMAN | Test del comportament d'agents racionals en entorns multi-agent | Simula l'entorn a través d'un escenari (<i>current world scene</i>) | Recull seqüències o intervals temporals del que passa |
| MICE | Coordinació entre agents heterogenis | Sensors i actuadors molt realistes | Recompte d'accions fetes i d'accions refusades |
| MYWORLD | Agents programats en llenguatge AOP | El rang dels sensors es pot especificar | Facilitats per veure les traces dels agents |

Taula 1.2.2.3.5: Plataformes d'experimentació de tècniques de IAD en sistemes multi-agent

En aquest cas, els experiments consisteixen en modificar alguns aspectes del món i veure com canvia el comportament dels agents. Per tant, l'objectiu d'aquest tipus d'experimentació és interpretar i generalitzar els resultats per tal de fer-los aplicables a la construcció d'aplicacions basades en agents. A la taula 1.2.2.3.5 es detalla una llista d'algunes d'aquestes plataformes, amb els trets essencials que les caracteritzen.

La plataforma DVMT (*Distributed Vehicle Monitoring Testbed*) [Durfee, 87][Lesser, 83] consisteix en simular un món amb mòbils amb sensorització acústica limitada espacialment i l'objectiu és obtenir mapes globals a partir de les informacions parcials de què disposen els mòbils. En aquesta plataforma els mètodes d'intercanvi de coneixement són molt elaborats i es basen en una estructura *blackboard*.

A la plataforma Phoenix [Cohen, 89][Greenberg, 90] es simula un món realista que consisteix en una batalla contra el foc: els bombers (en forma de *bulldozers*) són agents. Hi ha canvis meteorològics i topogràfics, i el terreny està subdividit en cel·les de forma que el comportament dels agents varia en funció de que hi ha a cada cel·la. El comportament dels agents pot ser tant reactiu (en forma de reflexos) com deliberatiu (generació i execució de plans) però la comunicació està sempre dirigida cap a la central (el cap de bombers) la qual s'encarrega de dirigir als agents.

TRUCKWORLD [Nguyen, 93][Hanks, 93] consisteix en un simulador de vies de circulació on hi ha punts de destí, material dispers i camions de transport—disposen de dos braços per poder carregar material i diversos sensors (cameres, sonars, raigs-X, balances). Els camions són agents i no tenen estratègies de cooperació predefinides. Es dediquen a recollir i transportar el material dispers i els casos de conflicte (espacials o de material) es resolen amb un procés de negociació relativament simple.

Una plataforma més elaborada és MAGES (*Multi-Agent System*) [Bouron, 90] on els agents estan definits per classes i en una jerarquia determinada. La comunicació entre ells és per pas de missatges i mitjançant diversos protocols que permeten seleccionar qui o quins són els destinataris de la informació. Això permet establir mecanismes de cooperació més elaborats perquè la comunitat pot estar molt organitzada, és a dir, poden definir-se grups d'agents amb un intèrpret particular dels missatges que hi ha entre els membres del grup.

Altres plataformes són [Burxhert, 90][Montgomery, 92][Wooldridge, 93]. La primera està pensada per agents racionals: *The Rational Testbed for Multi Agent Networks*, i permet treballar amb bases de coneixement molt estructurades i l'usuari pot participar en el sistema directament. La segona permet treballar amb agents tant reactius com deliberatius. Finalment, a MYWORLD els agents tenen un caràcter més deliberatiu donat que es programen amb el llenguatge AOP de [Shoham, 90] (veure següent secció) i no hi ha cap tipus d'organització pre-establerta ni cap possibilitat de fer-ho, és a dir, els agents operen de forma completament independent i sense dependències jeràrquiques.

1.2.3. Programació orientada a agents

La Programació Orientada a Agents (*Agent-Oriented Programming*, AOP) [Shoham, 90] es pot veure com a una especialització de la programació orientada a objectes (OOP) la qual proposa que les aplicacions informàtiques es poden plantejar a partir de mòduls o estructures d'informació anomenades objectes. Aquests objectes tenen la capacitat d'intercanviar-se informació i d'interpretar-la de diferents formes a través dels anomenats mètodes. A partir d'aquí, l'AOP aprofita aquests elements i estén la noció d'estat d'aquests objectes (anomenats agents) i els considera com a estats mentals dels agents, els quals estan formats per diferents components: creences, decisions, intencions, capacitats,... A més, l'AOP inclou mètodes avançats per passar missatges entre els agents especialitzats per a diferents funcions: per informar (*inform*), per demanar (*query*), fer requeriments (*request*), oferir, acceptar, rebutjar, assistir als altres, etc.

Els agents es programen perquè actuïn a partir d'una sèrie de regles de compromisos (*commitment rules*) amb què poden demanar o ser demanats per tal de realitzar una acció concreta. Aquestes regles s'executen amb *forward-chaining* i permeten relacionar els missatges amb els estats interns dels agents i amb llurs accions.

Per tant, l'estructura AOP inclou tres components principals: un llenguatge formal que descriu uns estats mentals ben definits, un llenguatge de programació per programar als agents, i un "agentificador" per convertir els processos de baix nivell de la màquina al nivell intencional que necessiten els programes dels agents.

1.2.3.1. Llenguatge formal

Les especificacions del llenguatge necessiten una prèvia definició de les categories mentals i de les seves propietats [Thomas, 92]. En l'AOP hi ha definides tres categories mentals : la creences (*belief*), la noció d'obligació o de compromís (*commitment*) i les capacitats (*capabilities*), i la de decisió (*decision* o *choice*) també està inclosa implícitament en forma d'obligació a un mateix.

- BELIEF : un agent té creences del seu entorn situades en un temps concrets i durant un cert interval de temps. La forma de indicar un estat d'aquest tipus és $B_a^t \varphi$ on a indica l'agent, t l' instant de temps, i φ és una sentència.
- OBLIGATION : és un operador modal amb dos arguments. La sintaxi és $OBL_{a,b}^t \varphi$ on s'indica que al temps t l'agent a està obligat (*commitment*) per l'agent b a realitzar φ . La diferència entre accions i obligacions roman en el fet que les accions es representen com a fets, mentre que una obligació indica a un agent que ha de realitzar una acció.
- DECISION : La llibertat de decisió entre diferents accions consisteix en ser una característica primordial de la tecnologia agent. La decisió es defineix com a una obligació a un mateix de la forma : $DEC_a^t \varphi \equiv OBL_{a,a}^t \varphi$.
- CAPABILITY : les capacitats d'un agent també constitueixen una part molt important i queden definides com a el conjunt d'accions que un agent és realment capaç de realitzar. La sentència és $CAN_a^t \varphi$ on queda indicat que l'agent a pot realitzar φ en t.

En aquest llenguatge el temps és bàsic en el sentit que les creences estan limitades en el temps i poden canviar d'un temps t_1 a un temps t_2 , és a dir, tot el procés es desenvolupa en una escala de temps (*grain time*) la qual cosa permet incloure el temps com a un paràmetre. Les accions s'executen en el temps i, en funció de les circumstàncies actuals, tenen efectes sobre els altres. És a dir, mentre en altres llenguatges els fets i les accions queden diferenciats, en l'AOP una acció sempre és certa. Aquesta certesa constitueix una de les limitacions del AOP tot i que queda suavitzada per la introducció de la decisió.

Finalment, respecte a les propietats dels estats mentals, s'assumeix que hi ha d'haver una consistència interna entre obligacions i creences, que els agents

s'obliguen a capacitats reals, que tenen un sentit de responsabilitat per tal de garantir les obligacions a què s'han compromès, i que els estats mentals han de tenir un mecanisme de persistència.

1.2.3.2. Programa intèrpret

El paper del programa intèrpret és controlar l'evolució de l'estat mental de l'agent i ha de consistir en sentències lògiques referides a l'estat mental i al seu processament controlant que, per exemple, creences contradictòries no quedin actualitzades al mateix temps. D'aquesta manera, el comportament dels agents queda estructurat en un procés de dos intervals : lectura de missatges amb actualització de l'estat mental, creences i requeriments, i execució dels requeriments corresponents al temps actual. El diagrama de flux d'un intèrpret pot veure's a la figura 1.2.3.2.

1.2.3.3. Agentificador

Per si mateix, l'intèrpret ja fa de pont entre el procés a baix nivell de la màquina i el nivell intencional del programa d'agents, però necessita una relació directa entre les estructures que representen els estats mentals i la representació a nivell de màquina. En altres paraules, l'agentificació consisteix en el procés mitjançant el qual un dispositiu (màquina, camera, sensor,...) queda equipat amb creences i regles de requeriments.

Per tant, aquest "traductor" ha de ser capaç d'agafar la descripció d'una màquina en llenguatge de processament i convertir-la en un programa intencional. L'estructura d'aquest traductor només està descrita a nivell teòric a [Shoham, 90] per què és un problema de caire tecnològic i només s'especifiquen algunes propietats (execució de processos de forma asíncrona, control del temps, inclusió de diversos nivells d'abstracció) de forma que es planteja com a un problema obert.

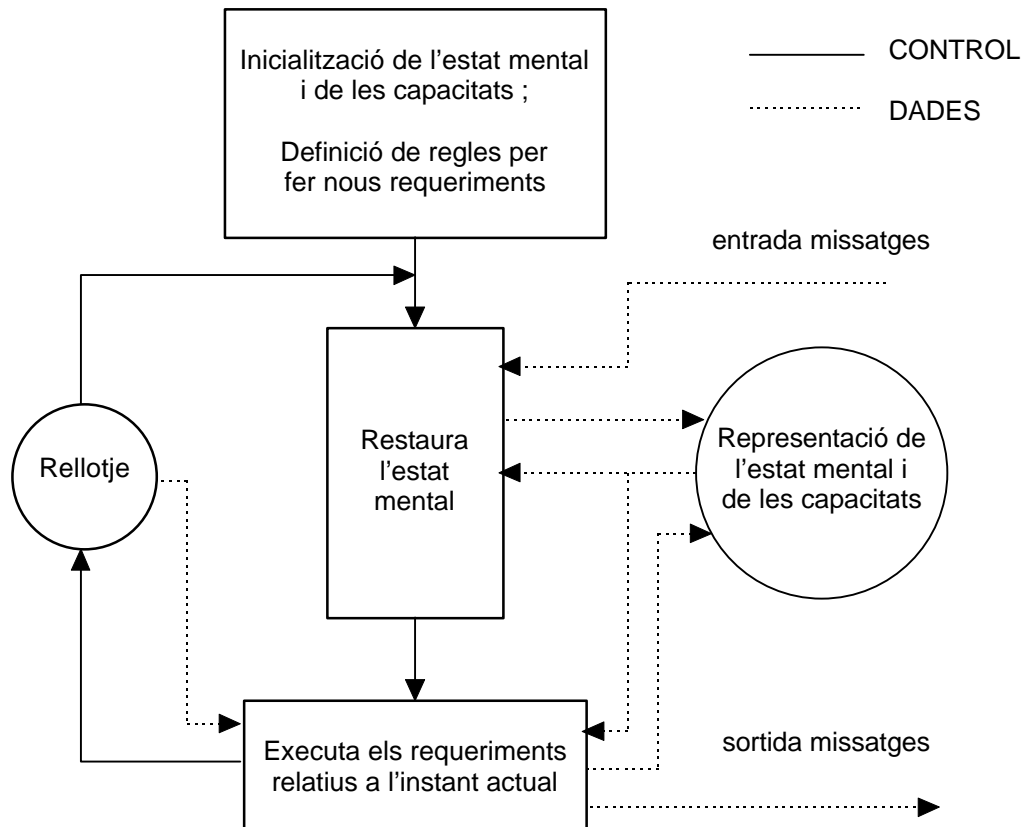


Figura 1.2.3.2 - Diagrama de flux de l'interpret d'un agent genèric

1.2.3.4. L'AGENT-0: general

L'Agent-0 és una implementació basada en els principis establerts per l'AOP i que suporta tres estats mentals: *Belief*, *Capability* i *Commitment* :

- Els *believes* són sentències lògiques relatives al món que l'agent creu com a certes o falses.
- Les *capabilities* són les accions que l'agent és capaç de realitzar.
- Els *commitments* són una garantia de que l'agent portarà a terme una acció en un temps determinat.

Respecte al pas de missatges, s'utilitzen tres tipus bàsics : *inform*, *request* i *unrequest*.

- *Inform* permet a un agent informar dels seus estats mentals als altres.
- *Request* permet que puguin demanar als altres que facin una acció.
- *Unrequest* cancel·la l'acció *request*.

A grans trets, el mètode de funcionament es basa en les regles de compromís (*commitment rules*), que consisteixen en un seguit de regles que contenen condicions sobre antecedents. Aquestes regles utilitzen la informació que disposa l'agent tant a partir dels missatges que van arribant com dels estats mentals que té (per exemple, utilitza la informació que li donen les seves creences). Quan una d'aquestes regles s'activa, l'agent genera un *commitment* per tal d'executar una acció interna (*private action*), o aturar una acció en execució (*refrain*), o senzillament enviar un missatge a un altre agent (*communicative action*). Finalment, els *commitments* que es generen s'executen en el temps especificat i per l'agent que ha fet la petició (*request*).

Amb aquest conjunt de característiques que proporciona l'AGENT-0 resulta que un agent no pot ser intencional. Aquesta limitació en les capacitats ha estat eliminada en una nova versió d'aquest intèrpret anomenada PLACA [Thomas, 93] la qual inclou la intenció com a un *commitment* per assolir un estat en un temps posterior. D'acord amb aquesta nova capacitat, resulta que les accions poden estar planificades i, per tant, un conjunt de múltiples accions (accions privades, accions comunicatives, etc.) poden formar part de les especificacions d'un pla. Així, quan un agent ha determinat un pla genera un *commitment* per tal d'executar-lo, la qual cosa implica que a partir d'una sola regla s'activaran diversos *commitments* (s'activaran a mida que les accions del pla vagin executant-se).

1.2.3.5. L'AGENT-0 : llenguatge i intèrpret

Aquest llenguatge ha de proporcionar les eines per descriure les regles de compromís, les accions, les capacitats i les creences d'un agent. Per fer la descripció del llenguatge anirem introduint definicions o components del llenguatge que són necessaris per a construir les regles.

- **Fact statements** : especifiquen el contingut de les accions així com les condicions per a la seva execució. Exemple:

```
(t (doctorand oller universitat)), o bé  
(NOT (t (doctorand oller universitat)))
```

- **Private and communicative action statements** : especifiquen quines accions s'han d'executar. Les accions es subdivideixen en dos tipus incompatibles entre ells: les privades, que són invisibles als altres agents, i les comunicatives.

Exemples:

```
(DO t accio_privada)
(INFORM t agent fact)
(REQUEST t agent action)
(UNREQUEST t agent action)
```

- **Conditional action statements** : fan referència a les condicions referides als estats mentals (*mental condition*), per exemple, si una *mental condition* és :

```
(B (t (doctorand oller universitat)))
```

la sintaxi d'una acció condicional és :

```
(IF mentalcond action)
```

- **Commitment rules** : es construeixen a partir de condicions mentals i a partir de les condicions que fixen els missatges rebuts (*message conditions*). Exemples:

```
(COMMIT msgcond mntlcond (agent action))
(COMMIT      (agent REQUEST action)
              (B (now (amic agent))))
              (agent action))
```

A partir d'aquestes definicions es pot escriure un programa especificant en primer lloc les capacitats i les creences dels agents, i després construir la seqüència de regles i especificant l'escala de temps (*time grain*). L'interpret executa el programa repetint una seqüència en què actualitza els *beliefs* i els *commitments*, i després s'encarrega d'executar totes les accions que toca fer en el temps actual, és a dir, enviar missatges si l'acció és comunicativa, o consultar la base de dades d'estats mentals i de *commitments* per comprovar si una condició determinada és certa i executar l'acció.

1.2.4. FIPA: un intent d'estandarització

La Fundació per als Agents Físics Intel·ligents (FIPA, *Foundation for Intelligent Physical Agents*) és una associació sense ànim de lucre registrada a Ginebra, i té el propòsit d'incentivar i promoure l'èxit de les aplicacions, serveis i equipaments basats en agents. Amb la participació activa dels seus membres³ es genera un document anual que recull les especificacions bàsiques de la tecnologia agent per tal que els desenvolupadors de sistemes complexos les tinguin en compte. De fet, FIPA

³ A l'octubre del 1998 hi havia 37 membres repartits en 12 països d'arreu del món. Cada membre representa una institució (industrial o universitària); a part hi ha un gran nombre de membres associats.

especifica les interfícies dels diferents components de l'entorn amb què l'agent pot interactuar, és a dir, humans, altres agents, programari no-agent i el món físic⁴.

Aquest document anual consta de dues parts:

- ✓ Especificacions sobre normatives, és a dir, les que regeixen el comportament extern d'un agent, i assegurin la interoperabilitat amb altres sistemes FIPA.
- ✓ Especificacions informatives, són una guia de disseny destinades al món industrial, per què puguin usar les especificacions de la FIPA.

Les especificacions que es comentaran en aquesta secció corresponen a les normatives, les quals contenen les següent parts⁵:

1. *Agent Management* : Gestió de l'agent
2. *Agent Communication Language (ACL)* : Llenguatge de comunicació per agents
3. *Agent/Software Integration*
4. *Personal Travel Assistance*
5. *Personal Assistance*
6. *Audio/Video Entertainment & Broadcasting*
7. *Network Management & Provisioning*

Com es pot veure, el document abasta un ampli ventall de problemàtiques en què la tecnologia agent és susceptible de ser-hi aplicada. En relació a la temàtica d'aquesta tesi, les especificacions més rellevant són les dels dos primers punts, és a dir, la gestió del propi agent i la definició del llenguatge de comunicació entre agents i els protocols associats. Les dues sub-seccions següents detallen algunes d'aquestes especificacions i, més concretament, aquelles que seran d'utilitat en capítols posteriors.

1.2.4.1. Gestió de l'agent

Proporciona la normativa que descriu l'entorn de treball (*framework*) on ha d'existir un agent, on pot operar i on pot ser gestionat. En concret es defineix l'acció de registre d'un agent, i s'especifica com ha de ser el pas de missatges o com ha de

⁴ Els detalls d'implementació dels agents i de les plataformes no es comenten enlloc; són opcions de disseny de cada desenvolupador de sistemes basats en agents.

⁵ El detall complet de les set parts és de domini públic i és completament accessible a través d' Internet a <http://www.fipa.org>.

funcionar el cicle de vida de l'agent en l'anomenada Plataforma Agent (AP, *Agent Platform*). Aquesta plataforma consisteix en ser la infraestructura física en què resideixen els agents, la qual pot contenir diversos *dominis*⁶. Les especificacions FIPA expliquen com ha de funcionar el sistema de comunicació entre agents d'un mateix domini, independentment de que algun d'ells estigui en una AP diferent. Així, defineix l'agent AMS (*Agent Management System*) que s'encarrega de gestionar tot el que fa referència als agents d'una mateixa AP⁷, i defineix un ACC (*Agent Communication Channel*) que serveix per gestionar les comunicacions entre agents de diferents AP's o de diferents dominis. La filosofia d'aquesta estructura pot entendre's millor a partir de l'esquema de la figura 1.2.4.1.1 on pot observar-se que un agent també pot pertànyer a diversos dominis.

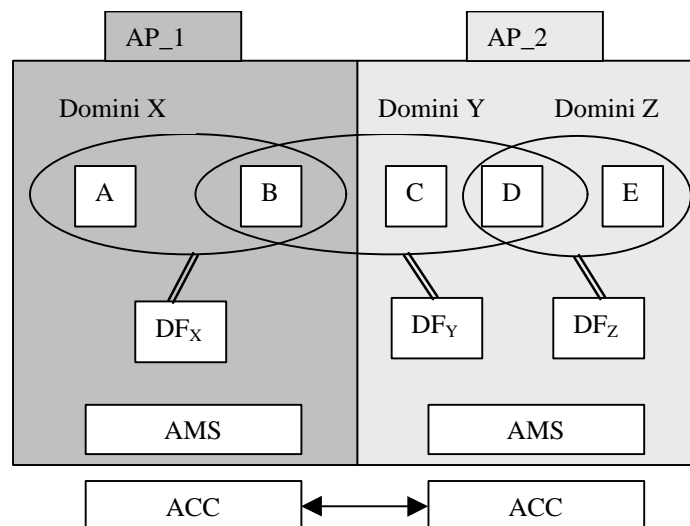


Figura 1.2.4.1.1: Fragment del model de referència de la Plataforma Agent (AP) proposada per la FIPA

A partir d'aquesta figura es veu com els agents A i B són residents de la AP_1 (a part del DF_X i del AMS), i que B pertany també a un altre domini. Quant a l'AP_2, els residents són C, D i E i estan repartits en dos dominis. A part del cicle d'evolució d'una AP, és important ressaltar el fet que els dominis són dinàmics i que, per tant, responen a un model de cicle de vida en què l'agent evoluciona per diferents estats (actiu, suspès, retirat) amb l'activació de diverses transicions (registrament,

⁶ Un domini consisteix en un grup d'agents els quals tenen un agent associat que s'encarrega de donar-los serveis. Cada domini té un agent associat (DF, *Directory Facilitator*) que proporciona servei de "pàgines grogues" només als membres del domini.

⁷ Per exemple, un agent és resident en una AP si l'agent AMS el té inscrit com a tal.

suspendre, invocar, retirar, esborrar). La relació entre aquests estats i transicions respon al següent model:

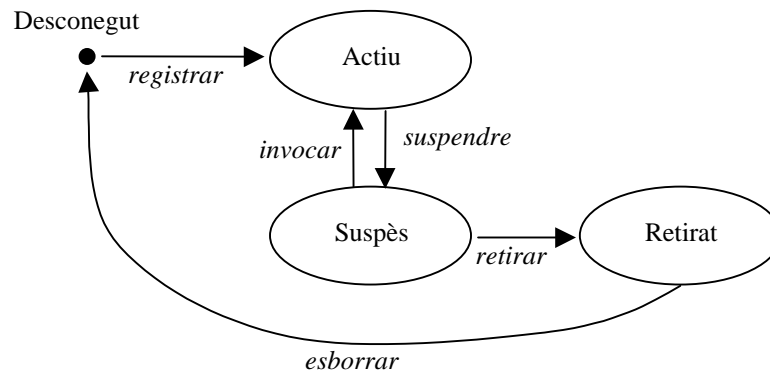


Figura 1.2.4.1.2: Model del cicle de vida d'un agent en un domini

Segons aquest model, un agent és actiu en un domini un cop està registrat a la DF corresponent i es manté en aquest estat a no ser que l'agent passi a ser inoperatiu temporalment. L'estat actiu es pot recuperar amb l'operació invocar—de fet consisteix en una acció de modificació. En cas de que la inoperabilitat deixi de ser temporal l'estat de l'agent és el de retirat. Finalment, quan l'agent demana al DF que l'esborri del domini és quan s'acaba qualsevol tipus d'acció de l'agent en el domini.

1.2.4.2. Llenguatge de comunicació per agents

Les especificacions dels llenguatges de comunicació per agents (ACL, en anglès) descriuen els missatges, els protocols d'interacció i el semàntica formal. És a dir, es detalla l'estructura i sintaxi dels missatges, i s'enumeren tots els tipus possibles, es defineixen fins a sis protocols diferents i, finalment, es defineix un llenguatge semàntic (SL, *Semantic Language*) que permet formalitzar la semàntica del llenguatge. L'objectiu final de l'ACL és cobrir un ampli ventall de situacions de comunicació amb una definició de redundància mínima i sense ambigüitats. És a dir, una especificació complerta, simple i concisa.

Els missatges en ACL són actes de comunicació, i responen a la següent estructura:

```
(inform
  :sender nom_agent
  :receiver id_recipient
  :content
    ..(objecte de l'acció, en una semàntica)
  :in-reply-to en resposta a una accióprèvia
  :in-reply-with identifica_el_missatge que es respon
  :language identifica la semàntica de content
  :ontology ontologia que dona significat a l'acció
)
```

Les dades d'aquesta estructura permeten facilitar el direccionament del missatge (**sender**, **receiver**), la recepció i la resposta cooperativa (**in-reply-to**, **in-reply-with**) i la interpretació del contingut del missatge (**language**, **ontology**).

Seguint aquesta estructura, es defineixen 20 actes de comunicació. Els significats d'aquests actes fan referència a les actituds mentals del model d'agent concret. Aquestes actituds són, per exemple, les creences, els compromisos, les decisions i les capacitats en el model de [Shoham, 90], o les creences, les intencions i els desigs en el model BDI [Rao, 91,93]. Els actes definits són els de la taula 1.2.4.2.

Tots aquests missatges s'utilitzen per a la comunicació entre agents. Aquesta comunicació permet l'intercanvi d'informació, el control d'execució d'accions, i la negociació entre diversos agents.

| Acció comunicativa | Pas d'informació | Demanda d'informació | Negociació | Execució d'una acció | Gestió d'errors |
|--------------------|------------------|----------------------|------------|----------------------|-----------------|
| Accept-proposal | | | √ | | |
| Agree | | | | √ | |
| Cancel | | | | √ | |
| Cfp | | | √ | | |
| Confirm | √ | | | | |
| Disconfirm | √ | | | | |
| Failure | | | | | √ |
| Inform | √ | | | | |
| Inform-if | √ | | | | |
| Inform-ref | √ | | | | |
| Not-understood | | | | | √ |
| Propose | | | √ | | |
| Query-if | | √ | | | |
| Query-ref | | √ | | | |
| Refuse | | | | √ | |
| Reject-proposal | | | √ | | |
| Request | | | | √ | |
| Request-when | | | | √ | |
| Request-whenever | | | | √ | |
| Subscribe | | | √ | | |

Taula 1.2.4.2: Categories d'accions comunicatives definides per la FIPA

1.2.4.3. Protocols d'interacció entre agents

Un protocol de comunicació és una estructura relacional que permet descriure la seqüència de missatges que hi ha d'haver entre agents. Per assolir objectius diferents, els agents hauran de seguir protocols diferents i específics per a cada situació. De tota manera cal tenir en compte que un agent pot estar involucrat en diversos diàlegs i per això la FIPA utilitza el mot conversa. En aquest sentit, els protocols especificats per la FIPA fan referència al protocol de cada conversa. Aquests protocols estan pensats per facilitar l'interoperabilitat entre agents simples i agents més complexos. Tot i que no són el resultat d'un estudi exhaustiu dels protocols ja existents, intenten recollir un ampli ventall de situacions.

Els vuit protocols que defineix la FIPA són els següents:

1. FIPA-Request protocol: permet que un agent demani a un altre que executi una acció.
2. FIPA-Query protocol: permet que un agent demani a un altre que faci una acció informativa. És una demanda d'informació.
3. FIPA-Request-When protocol: permet que un agent demani a un altre que executi en el futur una acció quan una condició determinada sigui certa.
4. FIPA-Contract-Net protocol⁸: permet que un agent pugui buscar l'agent que millor pot executar una acció.
5. FIPA-Iterated-Contract-Net protocol: com l'anterior, però amb la possibilitat de reintentar la negociació.
6. FIPA-Auction-English protocol: específic per a situacions venedor/comprador. El venedor va augmentant el valor de venda fins que un comprador ho atura.
7. FIPA-Auction-Dutch protocol: específic per a situacions venedor/comprador. El venedor va disminuint el valor de venda fins que un comprador ho atura.

A part d'aquesta llista, la FIPA també enumera una sèrie de convencions per tal que l'agent pugui estar en diverses converses, o que un agent faci d'intermediari en la conversa d'altres dos, o que s'iniciïn sub-converses dintre d'un protocol.

1.2.4.3. Plataformes compatibles amb FIPA

Actualment és difícil seleccionar l'entorn informàtic d'ajuda al desenvolupament de sistemes multi-agent donat que hi ha molts aspectes a tenir en compte. Fent referència a la conformitat amb les especificacions de la FIPA, hi ha diverses plataformes que presenten aquest tipus de característica. La taula 1.2.4.3. mostra les conformitats amb les normes que proposen FIPA i KQML [Finin, 93,94] de diferents plataformes així com la transparència que proporciona la plataforma per a desenvolupar aplicacions distribuïdes.

Les plataformes que consten a la taula són les següents:

AgentBuilder: <http://www.agentbuilder.com/>

April Agent Platform (AAP): <http://www.nar.fujitsulabs.com/aap/index.html>

⁸ És una versió modificada del protocol definit per Smith i Davis [Smith, 80]

JAFMAS: <http://www.ececs.edu/~abaker/JAFMAS/>

JATLite: http://java.stanford.edu/java_agent/html/

IBM-ABE: <http://networking.ibm.com/iag/iagsoft.htm/>

LALO: <http://www.crim.ca/sbc/english/lalo/>

Agent X: <http://www.iks.com/agentx.htm/>

Gypsy: <http://www.infosys.tuwien.ac.at/Staff/lux/Gypsy/>

FIPAOS: <http://www.fipa.org>

Jack: <http://www.agent-software.com.au/jach.html>

Kafka: <http://fujitsu.co.jp/hypertext/free/kafka/>

Via: <http://www.kinetoscope.com/via/default.htm>

Voyager: <http://www.objectspace.com/>

| Plataforma | Sistema de comunicacions | KQML | FIPA |
|-------------------|---------------------------------|-------------|-------------|
| AgentBuilder | √ | √ | √ |
| AAP | √ | | √ |
| JAFMAS | √ | | |
| JATLite | √ | √ | √ |
| IBM-ABE | | | |
| LALO | | √ | √ |
| Agent X | √ | | √ |
| Gypsy | √ | | |
| FIPAOS | √ | | √ |
| Jack | √ | | |
| Kafka | √ | | |
| Via | | | |
| Voyager | √ | √ | |

Taula 1.2.4.3.: Conformitat de diverses plataformes amb FIPA i KQML.

Nota: Cal fer notar que la compatibilitat amb FIPA significa que la plataforma ha de complir, almenys, les normes que estableix FIPA a les parts 1 i 2, és a dir, les normes relacionades amb la gestió de l'agent (*Agent Management*) i amb les del llenguatge de comunicació (*Agent Communication Language*).

2. La robòtica mòbil i els sistemes multi-robot

El mot “robot” deriva de la paraula txeca *robota* que significa treballador, servent i va estar introduïda per l’escriptor Karel Capek l’any 1921 a l’obra RUR quan intentava descriure unes màquines que havien de substituir l’home en treballs repetitius. Deixant apart aquest tipus de fet històric, el significat actual de robot ha evolucionat bastant gràcies a l’evolució de la tècnica, sobretot a partir dels anys 70. Els robots actuals (s’entén en un entorn industrial) substitueixen l’home en alguns aspectes per tal d’augmentar la productivitat en el procés de fabricació, i en d’altres situacions realitzen tasques que de cap manera podrien ser realitzables per l’home. A nivell de recerca o en aplicacions molt especials els robots van equipats amb un instrumental de sensorització bastant sofisticat la qual cosa els permet reproduir moviments humans i realitzar tasques exclusives, fins ara, per l’home: visió artificial estereoscòpica, articulacions antropomòrfiques (mans de cinc dits, dues cames), sensors tàctils en forma de pell, etc.

No s’entrarà a discutir la relació entre l’evolució de la robòtica i la repercussió social que se’n deriva. En tot cas direm que els robots ja estan presents en moltes situacions que ni el mateix Capek hagués pogut imaginar. Com a exemple constatarem el fet que en els darrers anys han aparegut conceptes com el de la

robòtica lúdica (fent referència a aquella robòtica destinada als aspectes de l'oci, també anomenada *edutainment robotics*), robòtica assistencial (destinada a ajudar la gent amb discapacitats físiques), robòtica mèdica (mecanismes autònoms d'alta precisió d'ús quirúrgic), robòtica submarina, tele-robòtica, etc.

2.1. La robòtica mòbil: robots autònoms

La robòtica mòbil fa referència al camp d'aplicació de la robòtica on la característica essencial dels robots és la seva capacitat de motricitat autònoma. Aquesta motricitat permet al robot la realització de desplaçaments en entorns més o menys estructurats i l'obliga a equipar-se d'una sensorització essencial per a captar l'estat de l'entorn. Quant a l'aspecte de la sensorització, hi ha robots tele-operats, és a dir, controlats a distància per un operador humà com, per exemple, robots desactivadors de bombes. Ara bé, l'aplicabilitat industrial actual de la robòtica mòbil està en entorns industrials altament estructurats (centres logístics com aeroports, ports marítims, grans hangars, etc.) on el robot és l'encarregat de portar càrregues d'un lloc a un altre i l'espai és suficientment ampli i segur per evitar problemes imprevistos al robot.

És fàcil suposar que l'optimització de la producció industrial obliga a usar més d'un robot en aquells casos que l'entorn així ho permet, és a dir, quan l'ús de més robots permet fer més feina. De tota manera, cal puntualitzar el fet que hi ha tasques que poden ser realitzades per un sol robot i tasques que necessiten més d'un robot per a ser portades a terme. És a dir, l'estudi de sistemes multi-robot és una extensió natural dels sistemes mono-robot, però també és una àrea de recerca per si mateixa quan el treball en equip de diversos robots és la única via de solució d'un problema (moure objectes de grans dimensions, aixecar bigues, etc.).

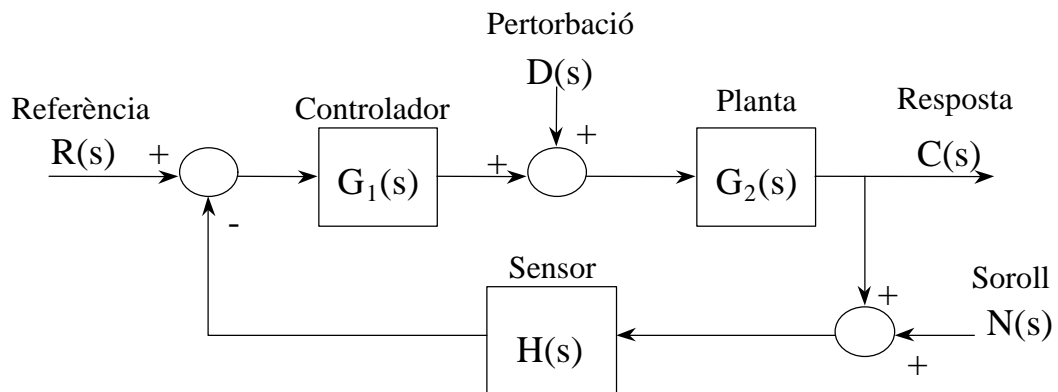
En l'àrea de la robòtica mòbil hi ha tres problemàtiques generals que són:

- ✓ El **control del moviment** del robot mòbil (velocitat i posició) com a entitat individual
- ✓ El control d'un sistema compost per diversos robots: la **cooperació**
- ✓ La **planificació de les accions** a fer en funció de restriccions temporals i espaials

Aquestes problemàtiques es desenvolupen a continuació, i al final s'inclou una discussió sobre la problemàtica de la imposició de restriccions temporals.

2.2. El control del moviment del robot mòbil

Suposem que volem que un robot mòbil segueixi una línia recta. A partir de la posició inicial del robot i del seu model podem calcular les velocitats lineal i angular que necessitem per fer-ho. Ara bé, una implementació real sempre presenta problemes de soroll (normalment de l'entorn) i d'imperficcions (normalment del model del robot o planta) fet que obliga a tancar el llaç, és a dir, a recalculer les velocitats en funció dels valors mesurats en el temps. Això fa que el sistema de control pugui ser menys sensible a diversos factors d'error: simplificacions del model, perturbacions externes, variació de paràmetres del model i incerteses de les mesures. D'altra banda, apareix el problema de la inestabilitat, és a dir, el problema de com calcular els paràmetres de control per tal que el sistema sigui estable.



Tot i que la llei de control en llaç tancat permet fer el sistema menys sensible als factors d'error, quan les perturbacions són massa grans no és possible assegurar el comportament del sistema. Aquest fet connecta amb la problemàtica que planteja la teoria de control robust, és a dir:

- ✓ Tenir en compte les incerteses tant de les mesures com del model, i de les perturbacions.
- ✓ Garantir el bon funcionament del sistema a partir d'un conjunt de criteris.
- ✓ Realitzar una tasca determinada amb una certa precisió.

En el cas de la robòtica mòbil hi ha molts factors que fan necessari l'ús de lleis robustes de control. Aquests factors van des de la modelització imprecisa del robots, el terra que no és exactament planer, els motors són no lineals o amb paràmetres

imprecisos (saturacions, zones mortes, fregaments, etc.), i sobretot, l'error en la localització del robot. Tots aquests factors són més o menys importants segons l'entorn on es trobi el robot i segons el tipus d'instrumentació de què es disposi. Per exemple, pel problema de la localització del robot (posició i orientació) es poden utilitzar tècniques de relocalització continuada [Leonard, 91], es poden combinar sensors interns amb sensors externs [Preciado, 91], etc. En canvi, pel que fa al problema de l'exactitud del model cal tenir en compte que la dinàmica dels actuadors no se sol modelar, és a dir, que es treballa només amb un model cinemàtic que, de fet, facilita el disseny de la llei de control [Aguilar, 97].

A part de la complexitat que suposa dissenyar un bon controlador, el robot mòbil força la introducció d'eines de control no lineal. A part de les no linealitats que apareixen en qualsevol sistema real, la geometria del robot dóna lloc a la propietat de no-holonomia⁹. Aquesta propietat introdueix un element nou de complicació. La relativa dificultat del problema de control no depèn només del tipus de no holonomia que presenti el robot, sinó que també depèn de l'objectiu de control. Hi ha algunes situacions on les tècniques clàssiques de control no lineal poden solucionar el problema de l'estabilitat [Bloch, 92][Campion, 91] o del seguiment de certs tipus de trajectòries [Sarkar, 93], però el problema més important a resoldre és la planificació de moviments del robot, tema que actualment encara està obert. En el cas d'un sistema holònom, sempre és possible calcular el conjunt de coordenades generalitzades independents i, amb elles, qualsevol moviment arbitrari sempre és realitzable. En el cas d'un sistema no holònom (veure figura 2.2), aquest conjunt de coordenades generalitzades independents no existeix i, per tant, no sempre serà possible executar qualsevol moviment: només es podran executar aquells que satisfacin instantàniament les restriccions no holònomes.

⁹ Un sistema mecànic és no holonòmic quan les restriccions que s'imposen en el moviment del sistema són no integrables, és a dir, les restriccions no són expressables com a derivades d'una funció expressada en coordenades generalitzades. Així, un sistema de control no holonòmic és aquell que s'aplica en sistemes no holonòmics que inclouen actuadors. Els exemples clàssics de sistemes no holonòmics són l'esfera o el disc en posició vertical damunt d'una superfície, és a dir, qualsevol tipus de roda d'un robot mòbil dóna lloc a la propietat de no holonomia. Més informació a [Kolmanovsky, 95]

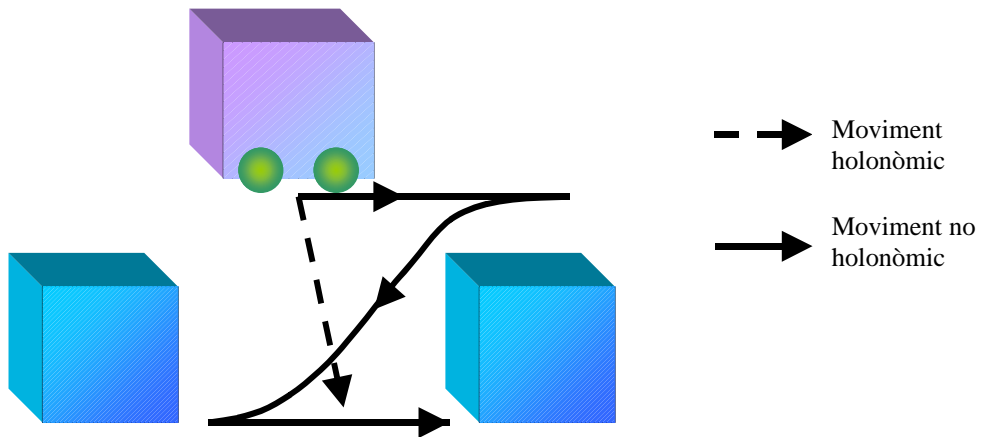


Figura 2.2: Exemple d'aparcament en paral·lel per a un robot mòbil amb les rodes davanteres giratòries. El moviment holònom correspon al no realitzable

Aquest fet provoca que els algorismes de planificació de trajectòries lliures de col·lisió han de tenir-ho en compte, mentre que els que no ho tenen en compte deixen de ser vàlids.

Finalment, també cal comentar que els desplaçaments dels robots mòbils amb rodes es realitzen damunt de superfícies típicament planes i homogènies, és a dir, amb un coeficient de fregament constant. Això indica que si s'exigeix que les rodes no patinin, les acceleracions lineals i angulars han de complir algunes restriccions (veure apèndix 2).

2.3. La cooperació en sistemes multi-robot

Els sistemes formats per grups de robots mòbils amb comportament col·lectiu són de gran interès en diverses situacions:

- ✓ Quan la tasca a realitzar és inherentment tan complexa que no és realitzable per un de sol
- ✓ Quan amb l'ús de diversos robots s'incrementa la productivitat
- ✓ Quan l'ús de diversos robots senzills (més barats) és més flexible i tolerant a fallades
- ✓ Si interessa derivar-hi conceptes de les ciències socials (teoria de l'organització, economia), de les ciències de la vida (biologia teòrica, etologia animal) i de les ciències del coneixement (psicologia, aprenentatge, intel·ligència artificial)

La paraula *comportament col·lectiu* fa referència al comportament dels agents en sistemes on hi ha més d'un agent, però el terme *comportament cooperatiu* és aquell en què hi ha associacions amb altres agents per tal d'obtenir un benefici mutu. Aquest benefici pot ser l'interès per assolir un objectiu, aconseguir algun objecte, guanyar temps, millorar algun procediment, etc. El problemes relacionats amb la cooperació fan referència a la tasca en si mateixa, o bé al mecanisme de cooperació, o bé també al grau de bon funcionament del sistema. En qualsevol dels casos, cal comentar que segons quin sigui el mecanisme de cooperació apareixen altres problemes com són el de la comunicació entre els robots per tal de compartir informació.

El fet que el robot mòbil sigui autònom i físicament independent ha conduït a que la utilització de la tecnologia agent s'hagi estès de forma natural¹⁰. I el mateix ha passat amb els sistemes multi-robot. De fet, els mecanismes de coordinació i d'interacció entre agents *software* s'estudiava en el camp de la IAD ja als anys 70 [Cao, 95] i no ha estat fins els anys 80 que la recerca en el camp de la robòtica cooperativa ha començat a ser molt activa. Els primers treballs eren simulacions com ara CEBOT [Fukuda, 89] o ACTRESS [Asama, 89] amb presència de fins a només tres robots, i SWARM [Beni, 88] amb un alt número de robots. A part del treball de G.Walter a mitjants dels 40 amb robots tortuga (segons es comenta a [Dorf, 90]), cal anar a treballs molt més recents per trobar implementacions de sistemes robòtics cooperatius [Mataric, 92][Parunak, 92]. De fet, el progrés de la tecnologia dels darrers 20 anys (sensors, cameres, sistemes electromecànics, processadors, etc.) ha conduït a la situació actual en què els sistemes multi-robot mòbils es dissenyen, es simulen i es construeixen en funció del problema plantejat. Així, han anat apareixent exemples cada cop més típics ja que estan sent utilitzats en molts treballs: control del trànsit (quan diversos agents/robots es mouen en un mateix espai i han d'evitar les col·lisions), recollida de material (simulant neteges de dipòsits tòxics, rescats i recerques) o arrossegament del material (desplaçament d'objectes o mercaderies en una nau industrial).

¹⁰ En aquesta secció les paraules 'robot' i 'agent' són completament equivalents

2.3.1. Arquitectures de sistemes multi-robot

L'obtenció d'un comportament cooperatiu s'ha de basar en una arquitectura, és a dir, una estructura determinada que proporciona una infraestructura amb què es poden avaluar les limitacions i les capacitats del sistema. Les arquitectures més dominants són de caràcter distribuït (descentralitzat) perquè són més tolerants a fallades, són escalables, i usen el paral·lelisme. Un tret important en les arquitectures és la definició del grau d'homogeneïtat/heterogeneïtat dels agents perquè afecta al mètode de distribució de tasques entre els agents; en un col·lectiu heterogeni els mètodes de repartiment de recursos són més complexos. El comportament d'aquests sistemes descentralitzats sol etiquetar-se amb el nom d'*emergent* o *auto-organitzatiu* gràcies al proveïment de mètodes que permeten la comunicació entre els agents. La forma de comunicació pot ser implícita quan dos agents interaccionen localment (es detecten sensorialment a poca distància), o pot ser explícita a través de missatges intencionats. En aquest segon cas la cooperació pot ser molt més efectiva o elaborada perquè permet que un agent obtingui un model de les intencions, creences, accions, capacitats i estats dels altres agents.

A continuació es detallen algunes de les arquitectures més referenciades a la literatura:

- **SWARM:** [Beni, 88] consisteix en un sistema distribuït amb un gran número de robots autònoms i poc intel·ligents. El col·lectiu té un caràcter homogeni i la interacció entre robots té lloc només entre els més propers i dintre d'una mateixa cel·la. El sistema està pensat per a dissenyar experiments de moviment en formació. Els experiments demostren que el col·lectiu presenta un grau d'intel·ligència superior a la suma de la intel·ligència dels components: consisteix en un comportament auto-organitzatiu. Aquest sistema és el primer en usar el concepte de sistema robòtic cel·lular (CRS).
- **CEBOT:** [Fukuda, 89] consisteix en una estructura jeràrquica completament descentralitzada inspirada en les organitzacions cel·lulars pròpies d'entitats biològiques. És un sistema reconfigurable en què les cel·les (robots) poden acoblar-se entre elles en funció dels canvis de l'entorn. CEBOT respon a les sigles en anglès de *Cellular Robot System*. La jerarquia consisteix en la definició de la cel·la (el robot individual), grups de cel·les (grups de robots), teixits (robots acoblats físicament), i òrgans (grups de robots acoblats). Un dels requeriments del sistema és

que les cel·les es controlin de forma distribuïda i coordinada i de forma que hi hagi poca acumulació d'intercanvis d'informació entre cel·les i una alta flexibilitat del sistema. Aquesta arquitectura permet estudiar quins són els requeriments mínims de comunicació per obtenir un comportament intel·ligent.

- **ALLIANCE/L-ALLIANCE:** [Parunak, 92] està pensat per estudiar la cooperació en equips no molt nombrosos de robots heterogenis i independents. Els robots poden captar l'efecte de les seves accions o dels demés a través de comunicació explícita i de sensors. El control dels robots està basat en comportament, és a dir, disposen de conjunts de comportaments associats a tasques que s'activen quan la tasca associada està activada. L-ALLIANCE consisteix en una ampliació posterior que inclou capacitats d'aprenentatge amb la tècnica de l'aprenentatge reforçat (*reinforcement learning*). Aquesta arquitectura s'ha implementat en robots reals i en simulació en entorns de desplaçament de material, marxa en formació, i neteja de tancs.
- **GOFER:** [Calou, 90] consisteix en un sistema central de planificació de tasques i de *scheduling* (CTPS) que té una visió global del sistema tant de les tasques a fer com de les disponibilitats dels robots. Es pot comunicar amb els robots per enviar-los ordres a partir de la generació d'un pla. Els robots accedeixen a les tasques a partir de l'algorisme d'assignació de tasques (el *Contract Net Protocol* definit a [Smith, 80]). Un cop tenen una tasca assignada, els robots poden utilitzar tècniques de planificació de IA. De fet cal comentar que la motivació original d'aquesta arquitectura era l'estudi de la resolució distribuïda de problemes a partir de tècniques clàssiques de IA. Aquesta arquitectura s'ha implementat en un sistema de tres robots en tasques de moviment de material (per arrossegament) o de moviment dels propis robots en entorns de passadissos.

2.3.2. Línies generals de recerca

Els treballs en robòtica cooperativa sempre giren al voltant de la mateixa pregunta: "Donat un grup de robots, un entorn i una tasca, com es pot obtenir un comportament cooperatiu?" [Cao, 95]. La forma de respondre aquesta pregunta és investigant els diversos punts que es detallen a continuació:

- **Fonts de conflictes**

El conflicte apareix quan diversos processos intenten accedir a un mateix recurs. Aquest tema no és exclusiu dels sistemes multi-robot ja que ha estat estudiat com a

problema d'exclusió mútua en algorismes distribuïts o com a problema de multi-accés en xarxes d'ordinadors. En la robòtica mòbil el conflicte fa referència a la compartició d'una zona de l'espai, o d'un objecte manipulable o d'un medi de comunicació. El conflicte més estudiat (i més freqüent) és el que fa referència al conflicte espacial; quan diversos robots planifiquen trajectòries en un entorn comú, poden aparèixer col·lisions entre ells o amb objectes de l'entorn (parets, columnes, etc.). En entorns reals la planificació *a priori* de camins lliures de col·lisió no sol ser una bona opció [Cao, 95] de forma que cal usar tècniques alternatives: planificació recursiva [Laumond, 94], inserció de plans [Aguilar, 97] o simplement la comunicació [Calou, 90].

- **Problemes geomètrics**

L'aplicació de la tecnologia agent està tenint molts èxits tant en els sistemes multi-robot com en xarxes d'ordinadors. Hi ha autors que discuteixen si té sentit aplicar aquestes tècniques a la robòtica mòbil en el sentit de que no aporta res nou que no es pugui fer amb xarxes d'ordinadors [Etzioni, 93]¹¹. La robòtica mòbil presenta problemes específics per a resoldre com són els problemes geomètrics, és a dir, els derivats del fet que els robots es mouen en un món físic real i que han d'interaccionar físicament amb altres. Un d'aquests problemes és el de la planificació de trajectòries, ja comentat anteriorment com a generador de conflictes. Un altre problema és el del moviment en formació perquè planteja el problema de com gestionar informació global amb coneixement local [Parker, 93][Balch, 95]. En relació a aquest problema, hi ha treballs que utilitzen sistemes robòtics cel·lulars (els clàssics CEBOT [Fukuda, 89] i SWARM [Beni, 88]) en què l'aspecte important del treball és donar la capacitat espacial d'auto-organització al sistema, és a dir, que el sistema es reconfiguri sol sense cap tipus d'intervenció externa.

- **L'origen de la cooperació**

L'aparició de la cooperació és deguda al fet que està explicitada en el mateix disseny del sistema¹². Una forma d'entendre quin tipus de cooperació sorgeix en un sistema és a través de l'anàlisi de l'estructura de control del sistema i de l'anàlisi del

¹¹ Contràriament al que opina Etzioni, l'autor d'aquesta tesi és de l'opinió que la recerca en IA treballant amb robots planteja problemes que no apareixen treballant amb xarxes d'ordinadors. Un d'aquests problemes són les limitacions que imposen les lleis de control que regeixen els moviments del robot. Etzioni, tot i que no discuteix la importància de la recerca en robòtica, basa la seva opinió en el fet que el problema de la percepció en robòtica força tant la recerca que els resultats són perfectament reproduïbles amb *softbots*—agents que interaccionen en móns *software* reals com ara Internet.

grau d'igualtat que pugui haver entre els individus (veure l'esquema proposat per S. Franklin l'any 1987, figura 1.2.2.1), però entendre com apareix el comportament cooperatiu és un problema que estudien més les ciències socials, l'etologia animal, i fins i tot des de la economia o les teories de jocs. Quant als treballs més específicament de la robòtica, hi ha alguns grups de recerca que treballen amb micro-robots molt simples amb la idea de dissenyar sistemes suficientment simples que permetin fer una anàlisi a fons del que passa mentre la cooperació apareix, com el grup d'Intel·ligència Artificial del MIT que emulen el comportament de colònies de formigues. En línies generals, la recerca en aquest camp de la robòtica mòbil es limita a classificar el comportament cooperatiu i en veure els efectes que es produeixen quan es modifica la configuració inicial del sistema.

La dificultat per entendre la cooperació i el comportament col·lectiu està en el fet que els sistemes multi-robot poden considerar-se com a sistemes complexos adaptatius (CAS, de l'anglès)(veure figura 2.3.2.), és a dir, sistemes amb diversos agents heterogenis que poden tenir interaccions entre ells i amb l'entorn de forma no lineal en el temps, i que són capaços d'adaptar el seu comportament basant-se amb la seva experiència. La barreja d'aquestes dinàmiques dóna lloc a les següents característiques emergents dels CAS: auto-organització, trajectòries evolutives, co-evolució, i equilibri puntual [Rosenau, 97].

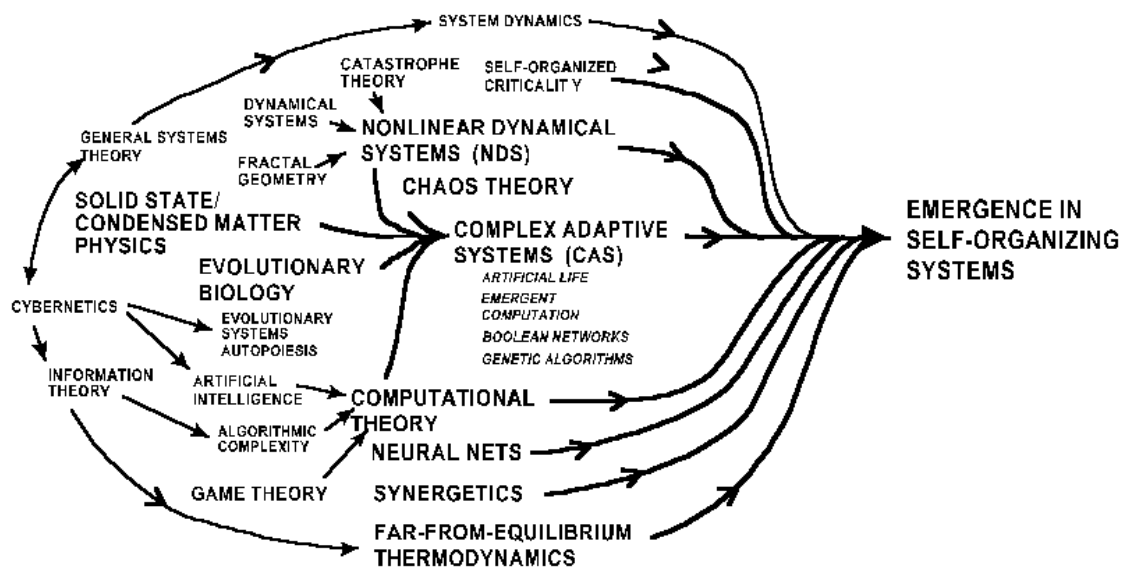


Figura 2.3.2: Àrees de la ciència que analitzen el fenomen de l'emergència.

¹² L'autor d'aquesta tesi no té coneixement de treballs en què la cooperació entre els robots aparegui sense motivació humana explícita, és a dir, del mateix dissenyador del sistema.

Aquestes característiques són emergents en el sentit que el coneixement complet dels agents de forma individual no és suficient per inferir els detalls o la cronologia d'aquestes característiques.

- **L'aprenentatge**

L'aprenentatge fa referència a la possibilitat que un robot o un col·lectiu de robots puguin optimitzar els seus resultats i adaptar-se als canvis de l'entorn. Els robots construeixen models del món per tal d'usar-lo com a base del comportament que respondrà de forma efectiva a l'entorn. Diferentment a l'aprenentatge humà, quan els robots intel·ligents utilitzen l'aprenentatge *off-line* tenen la opció de bolcar el coneixement après als altres robots. És a dir, un sol robot pot fer per si sol tant l'aprenentatge com la transferència de coneixement. D'altra banda, l'aprenentatge *on-line* es pot compartir per tal d'ampliar-lo. Mentre els robots puguin aprendre a través de les seves diferents experiències en temps d'execució, també podran compartir les seves experiències quan la sincronització periòdica així ho permeti.

L'ús de les tècniques d'aprenentatge en dominis complexos sol consistir en comprovar les relacions directes (*mapping*) des dels sensors cap els actuadors. De tota manera, quan els dominis són molt complexos —amb comunicació limitada, temps real, entorn sorollós —aleshores s'utilitzen les arquitectures d'aprenentatge per nivells [Mataric, 95] [Stone, 98], i els dissenys d'aprenentatge jeràrquic reforçat [Mahadevan, 91] [Maes, 90] [Lin, 93]. L'aprenentatge per nivells té reminiscències de l'arquitectura de subsumció de Brooks [Brooks, 86]; Mataric usa aquest tipus d'aprenentatge en dominis multi-robot construint controladors que regulen els comportaments base apresos, i Stone proposa un paradigma d'aprenentatge per nivells per a dominis complexos més generals. Així, a partir d'una descomposició jeràrquica de tasques permet l'aprenentatge a cada nivell jeràrquic i que afecta a l'aprenentatge del nivell immediatament superior. D'altra banda, l'aprenentatge reforçat es concentra en diversos comportaments (*gated behaviours*): Mahadevan treballa a partir de comportaments apresos prèviament, i Maes dirigeix l'aprenentatge cap a les funcions que modulen cada comportament i aplica l'aprenentatge tant a les funcions com als propis comportaments.

La robòtica cooperativa és un camp de recerca que està creixent molt en els darrers anys i planteja moltes qüestions noves. Segurament els pocs anys

d'existència d'aquesta robòtica marquen la falta de formalisme en molts aspectes. De tota manera cal remarcar el fet que aquests sistemes són complexos en disseny, implementació, funcionament i, sobretot, en l'anàlisi. De fet l'avaluació del bon funcionament d'aquests sistemes és encara un tema obert de recerca sobretot pel fet que encara no es comprenen bé els mecanismes de la cooperació. Una forma d'obtenir resultats experimentals més rigorosos és la definició de problemes estàndards, és a dir, problemes *benchmark*. Això és una dificultat difícil de superar donat que els sistemes reals amb molts (o pocs) robots presenten un alt grau de perturbacions: interferències de l'entorn i del propi robot, no-linealitats, imprecisions de les mesures, el caràcter continu (en contraposició del discret) dels canvis de l'entorn i del mateix robot, etc. Com ja s'ha comentat anteriorment, aquest fet ha portat a considerar com a més important la robustesa del sistema en lloc del caràcter òptim del seu comportament.

En referència a la definició de problemes estàndards o *benchmark*, en els darrers anys han aparegut diversos tipus d'events científics encaminats a treballar en aquesta direcció: són trobades científiques/tecnològiques en què la discussió de treballs es fa paral·lelament a competicions per a robots. En aquest sentit cal remarcar l'aparició de la iniciativa RoboCup l'any 1997 [Kitano, 97,97a] en què la competició consisteix en emular el joc de futbol, és a dir, es planteja que un col·lectiu de diversos robots funcioni en un entorn molt dinàmic i en situació de competència respecte d'un altre col·lectiu.

2.4. Resolució de problemes de móns dinàmics amb planificació

Un robot o agent podrà planificar explícitament o implícita per tal d'assolir finalment els seus objectius, és a dir, seleccionarà les seves accions en funció dels resultats que ell espera que se'n derivi o a partir de les conseqüències racionals que provocarà en ell mateix. Les tècniques de planificació també poden ser útils en dominis multi-robot en què hi hagi jerarquia social entre els robots. Així, un robot planificador pot enviar tasques als altres d'acord amb els resultats d'un algorisme. Tanmateix, qualsevol robot podria contenir també algorismes de planificació intern per tal d'obtenir més oportunitats i assolir així les tasques que se li han encomanat, tal com es proposa a [Matellán, 98].

En dominis estàtics, el món no canvia entre l'acabament d'una acció i l'inici de la següent. Això permet representar en un estat tot el que és cert del món en un instant de temps entre una acció i una altra. En canvi, en un domini dinàmic, això no és suficient ja que el món canvia de forma contínua. De fet, si alguna cosa és certa en un instant de temps concret no és indicatiu de que encara ho sigui en un instant de temps immediatament posterior. Alguns treballs suggereixen que alguns problemes de planificació de móns dinàmics formulats com a problemes multi-agent, es poden reformular com a problemes de móns estàtics i amb un sol agent. Per tant, això permetria resoldre'ls amb les tècniques clàssiques de planificació [Pednault, 90] [Georgeff, 86]. Sota aquest punt de vista, els problemes de móns dinàmics es poden modelar de la mateixa manera com si fossin móns estàtics i on les accions simultànies es podrien modelar en forma de seqüències d'accions. Segons aquests treballs, un element important a tenir en compte és la visió del temps com a paràmetre descriptor dels móns dinàmics. La clau per modelar accions en entorns dinàmics és fer-ho amb conjunts de parells del tipus estat_actual/ estat_següent ja que s'aconsegueix catalogar què és cert en cada punt del temps i per a tots els punts temporals [Pednault, 90]. Així, d'acord amb aquesta nova terminologia, un estat és una *crònica*, amb tot allò que ha estat, és i serà cert en el món [MacDermott, 82]. Com que els estats es poden interpretar com a cròniques, les transicions entre estats es poden interpretar com a les inicialitzacions d'accions o com als estats en què es troba l'execució d'una acció en un instant de temps particular.

En algunes situacions, l'efecte d'executar accions de forma simultània és el mateix que el de fer-ho de forma seqüencial, independentment de l'ordre de l'execució. Això passa, per exemple, quan les accions afecten parts independents del món i, per tant, no interactuen. Si aquest és el cas, el problema es pot resoldre assumint aquelles accions que s'han d'executar de forma seqüencial i posteriorment transformant el pla seqüencial que en surt en un pla paral·lel per coordinar les seves execucions simultànies. De tota manera, no totes les accions es comporten igual: algunes d'elles tenen efectes *sinèrgics* quan s'executen de forma simultània, és a dir, els seus efectes són diferents si es comparen els que s'obtenen d'una execució seqüencial dels que s'obtenen d'una execució simultània. Pernault il·lustra aquest procés de seqüenciació en el problema de l'aterratge lunar tant en situacions d'un sol agent com en situacions multi-agent [Pednault, 90]. De tota manera, tal i com diu ell

mateix en el seu treball, aquest exemple consisteix en ser només una il·lustració i no representa cap tipus de problemàtica comuna en els problemes multi-agent.

2.4.1. Planificació en temps real

A causa de la limitació de recursos en entorn reals multi-robot, els robots sovint han de dedicar temps a les tasques d'obtenció dels recursos. En el cas d'un sistema de temps real, el recurs més crític consisteix en ser el temps. Per tant, els mètodes de planificació per a distribuir el temps entre els robots/agents són molt importants, com també ho és l'avaluació del grau de precisió de la solució a què s'arriba tenint en compte les restriccions de temps reals.

A causa d'aquesta limitació en el temps, hi ha dos tipus de tècniques de planificació d'agents: les basades en algorismes *anytime* i les basades en *design-to-time*. Les primeres [Dean, 88] es basen en un algorisme iteratiu que refina la solució en cada iteració per la qual cosa és possible interrompre l'execució en qualsevol moment. Així, a partir d'una corba que proporciona la qualitat de la solució en funció del temps (perfil d'execució) es pot saber la qualitat de la resposta en el moment que es produeix la interrupció del procediment. Els algorismes *design-to-time* [Garvey, 93] es basen en tenir diversos algorismes per a un mateix problema i que produeixen diferents solucions a diferents temps i amb diferents qualitats. En aquestes condicions, la tècnica es basa en intentar trobar solucions satisfactòries (no òptimes) a un problema donat intentant utilitzar tot el temps disponible.

2.5. El problema dels sistemes de temps real

Un sistema de temps real es defineix com aquell en què la correcció del sistema depèn no només del resultat lògic del càlcul sinó també de l'instant de temps en què es produeix aquest resultat [Stankovic, 88]. En un sistema d'aquest tipus algunes tasques poden tenir *dead-lines*, és a dir, han de subministrar una resposta abans d'un interval de temps definit. Si la resposta s'obté més tard és probable que ja no sigui servible i que, fins i tot, produeixi efectes nefastos¹³. Una altra característica

¹³ Per entendre millor el terme de "sistema de temps real" es pot utilitzar un exemple. En l'accés a una base de dades, un sistema que sigui capaç de respondre en una escala humana de temps ràpida (alguns segons), es considera que és un sistema de temps real. Però en una situació amb condicions adverses és

temporal d'aquests sistemes és el seu comportament periòdic: el període determina la freqüència d'activació de les tasques en el sistema, i és necessari a causa de la naturalesa dinàmica del propi entorn.

Els sistemes de control de robots mòbils autònoms no només han de prendre decisions correctes sinó que també les han de prendre en el moment adequat, sobretot si es mouen en entorns dinàmics. La recerca en els sistemes de temps real es dirigeix precisament cap a aquest objectiu: el desenvolupament de mètodes que garanteixin que la velocitat de reacció del sistema es correspongui amb la velocitat de canvi de l'entorn. En aquest sentit, el terme "sistemes de temps real" no es refereix a sistemes ràpids, sinó a sistemes suficientment ràpids i predictibles.

Per assolir aquests objectius, en el camp dels sistemes de temps real s'han desenvolupat diverses tècniques i eines per a caracteritzar les tasques i per generar planificadors adequats. Aquests sistemes solen desenvolupar-se en sistemes poc complexos i en entorns molt estructurats, i això ha dificultat la seva extensió cap a tasques més complexes [Botti, 99]. No ha estat fins els anys noranta que els sistemes d'intel·ligència artificial en temps real (RTAIS, de l'anglès) estan esdevenint una bona opció per a la resolució de problemes complexos que necessiten intel·ligència i temps de resposta en temps real, com poden ser GARDIAN [Hayes-Roth, 92], PRS [Howe, 90], PRS [Ingrand, 92], CIRCA [Musliner, 93], i CELLO [Occhetto, 98]. En aquest sentit, un entorn de temps real té alguns trets que afecten al disseny de les arquitectures de control. Els dos trets més importants són:

No determinisme: l'estat següent de l'entorn no es pot determinar a partir de l'estat actual i de l'acció que s'executa.

Dinamisme: l'entorn pot canviar mentre el sistema està deliberant.

Qualsevol sistema que operi en un entorn moderadament complex i impredecible ha de ser *reactiu*¹⁴, és a dir, ha de poder respondre de forma dinàmica als canvis del seu entorn. Una de les formes més utilitzades per obtenir un bon temps de resposta és oferir una arquitectura amb capacitats reactives. En aquest sentit, l'arquitectura de Kaelbling [Kaelbling, 89,90] inclou un llenguatge declaratiu

possible que aquest mateix sistema tardi algunes desenes de segons, és a dir, que aleshores es consideraria no acceptable. Aquest és precisament el comportament totalment oposat al d'un sistema de temps real on el que es tracta és de garantir que les diferents tasques compleixen amb els seus terminis màxims.

anomenat REX dissenyat per implementar sistemes de temps real que s'assembla a un llenguatge de descripció de *hardware*.

2.5.1. Arquitectures RTAIS

S'han construït grans aplicacions o arquitectures que contempen les restriccions que imposa el temps real [Garvey, 94], i en aquesta secció se'n comentarà una selecció d'acord als criteris utilitzats a [Vivancos, 98]:

- (i) Adaptació a entorns dinàmics (DIN)
- (ii) Resposta reflexa (REF)
- (iii) Interrupció del processament (INT)
- (iv) Sincronia amb l'entorn (SIN)
- (v) Dinamisme de l'entorn (DIN)
- (vi) Execució en temps real (EXE)
- (vii) Coherència en el comportament → fa esment a si es resolen conflictes quan hi ha objectius contradictoris (COH)
- (viii) Tasques robòtiques (ROB)
- (ix) Deliberació respecte de operació → fa esment a si es compara el temps que es dedica a la planificació respecte del de l'operació (TEM)

La taula 2.5.1. mostra les valoracions que s'han fet de les següents arquitectures:

- **AIS** [Hayes-Roth, 95]: El nom respon a *Adaptive Intelligent Systems* desenvolupat al *Knowledge Systems Lab.* (CS Dept-Stanford Univ). Proporciona una arquitectura per a agents intel·ligents que funciona amb eficàcia en diferents entorns d'alta variabilitat. GARDIAN [Hayes-Roth, 92] consisteix en un agent experimental que vigila una unitat de cures intensives.
- **PHOENIX** [Cohen, 89][Greenberg, 90][Howe, 90]: Permet experimentar amb agents autònoms per a entorns complexos. El camp d'aplicació és el control d'incendis de boscos, és a dir, es dedica a planificar el treball d'equips humans, excavadores, avionetes, etc.(també comentada a la secció 1.2.2.3.5.)
- **PRS** [Ingrand, 92]: El nom respon a *Procedural Reasoning System* i consisteix en un sistema de raonament general per a aplicacions en temps real. Les aplicacions

¹⁴ Un robot és reactiu si segueix cegament les operacions o el pla i verifica que les seves accions provoquen els efectes esperats.

reals que s'han fet consisteixen en el *Space Shuttle's Reaction Control System*, o el sistema IRTNMS (control i monitorització de Telecom Australia).

- **REAKT** [Kersual, 94]: El nom respon a *Environment and Methodology for Real-Time Knowledge-Based Systems* i ha estat desenvolupada en el marc del programa europeu ESPRIT-II. Consisteix en un sistema multi-agent basat en una pissarra temporal i que serveix per controlar i gestionar diferents components: extensió del sistema operatiu en temps real, un gestor del coneixement, tasques periòdiques (comunicacions), tasques esporàdiques (events externs), tasques dels actuadors i un servidor expert que gestiona les tasques d'alt nivell.
- **ARTIS** [García-Fornés, 95]: Desenvolupada al Dept. de Sistemes Informàtics i Computació de la Univ. Politècnica de València. Consisteix en un SMA construït a partir de l'evolució de REAKT en què s'han realitzat extensions per tractar les restriccions de temps real.

| <i>Criteri</i> | AIS | REAKT | ARTIS | PRS | PHOENIX |
|----------------|------------|--------------|--------------|------------|----------------|
| DIN | * | * | * | * | * |
| REF | * | * | * | * | * |
| INT | * | | | * | * |
| SIN | * | * | * | * | * |
| DIN | * | * | * | * | * |
| EXE | * | * | * | * | * |
| COH | * | | | | * |
| ROB | * | | * | * | * |
| TEM | | | * | * | |

Taula 2.5.1: Propietats d'algunes arquitectures RTAIS [Vivancos, 98]

Respecte a aquestes arquitectures, s'observa que hi ha tres tipologies per integrar la IA i el temps real. En primer lloc, hi ha la tipologia que integra el temps real en el sistema IA en què el sistema es comporta com un sistema intel·ligent convencional, però en algunes situacions s'inhibeixen les tasques de raonament per poder executar una acció en temps real. En segon lloc, hi ha la tipologia en què hi ha una cooperació entre el sistema IA i el sistema de temps real, intentant sempre mantenir la potència de cada sistema i garantint que el sistema de IA no interfereixi les operacions del sistema de

temps real. Finalment, la tercera tipologia consisteix en integrar els sistemes de IA en sistemes de temps real però modificant-les per tal de facilitar el compliment dels terminis màxims d'execució del sistema de temps real.

En totes les arquitectures s'observa la separació en mòduls independents de les components encarregades de la percepció, el raonament i l'actuació, separació que reflexa la tendència de l'ús d'agents independents per a cascuna d'aquestes tasques [Vivancos, 98].

3. Robot + IA = agent físic

Les operacions d'un robot estan molt lligades a les dinàmiques de les interaccions entre el sistema de control del robot, el robot físic i l'entorn. Les tècniques convencionals de programació de robots¹⁵ solen ignorar o solen fer suposicions implícites sobre la dinàmica del controlador. Aquest fet ha donat lloc a un corrent de pensament que suggereix que els models computacionals clàssics s'han de deixar de banda i buscar un punt de vista proper als sistemes síncrons, és a dir, aquells en què el programador pot oblidar-se dels problemes relacionats amb la temporalització [Westhead, 95] i pot disposar d'algun tipus de llenguatge de control formalment definit com el LUSTRE [Halbwachs, 91] o el ESTEREL [Bérry, 92]. La diferència entre els dos punts de vista està en el fet que les tècniques clàssiques programen accions "per a un entorn" o "en un entorn", mentre que en els sistemes síncrons es busca que el robot sigui més reactiu i pensi més en quines seran les seves reaccions "respecte dels canvis de l'entorn".

Un altre corrent de pensament és el que busca una visió més propera a la dinàmica de sistemes [Smithers, 94][van Gelder, 94][Bergener, 98][Randall, 00], en què es tenen en compte quines són les "interaccions amb l'entorn", és a dir, és una visió més

¹⁵ El llenguatge de programació BL (*Behaviour Language*) [Brooks, 90] pot ser l'exemple més clar. S'ha utilitzat per a la programació de l'hexàpod Ghengis: consta de 56 màquines d'estats finits, cadascuna amb un màxim de sis estats. El resultat és impressionant per la seva robustesa però les tasques que realitza no són complexes. Si l'interès fos fer de Ghengis un robot realment autònom, faria falta un controlador de diversos ordres de magnitud superior a l'actual [Westhead, 95].

integradora del robot amb l'entorn (veure figura 3.1): tant el cos físic del robot/agent, com la seva intel·ligència, com l'entorn mateix estan contínuament interaccionant. De fet, els darrers deu anys hi ha hagut una gran activitat de recerca [Smithers, 92][Beer, 92][Cliff, 93][Schöner, 95][Tani, 99] per a modelar els agents com a "sistemes nerviosos"¹⁶: aquesta tècnica possibilita l'anàlisi de la dinàmica de l'evolució dels agents i permet estudiar com afecta el comportament *cognitiu* individual al comportament *cognitiu* col·lectiu [Beer, 97].

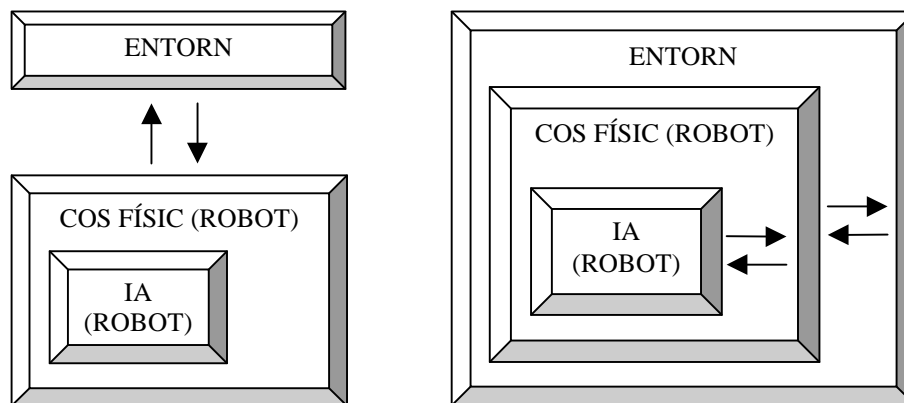


Figura 3.1: A l'esquerra un esquema de la relació entre un agent/robot i l'entorn segons una concepció convencional i/o de sistemes síncrons. A la dreta el mateix esquema segons una concepció de dinàmica de sistemes.

Sota aquests punts de vista, no és important l'estudi dels efectes (en el comportament) de les restriccions del món real com poden ser la imprecisió o la incertesa. Aquests efectes poden ser més o menys importants en funció de quina sigui la realització física de l'agent: en el cas que la realització sigui el robot mòbil aquests efectes són molt importants donat que influeixen en la qualitat de l'execució dels moviments/accions i, per tant, cal tenir-los en compte a les etapes de decisió. Per analitzar-los cal incorporar als agents *software* la problemàtica de la interfície numèrica-simbòlica i simbòlica-numèrica que caracteritza els sistemes físics [Asada,97].

¹⁶ Típicament els comportaments s'implementen en forma de xarxes neurals o amb programació evolutiva, i sempre amb capacitats d'aprenentatge, és a dir, amb capacitat d'adaptació a l'entorn.

3.1. Del paradigma GOFAIR al *situated agent*

Un dels problemes més importants que s'ha trobat la IA en treballar amb sistemes robòtics ha estat el que s'ha anomenat GOFAI (*Good Old Fashion Artificial Intelligence*). Aquesta frase va estar introduïda per Haugeland [Haugeland, 85] per referir-se a la metodologia de la IA basada en la manipulació de símbols: la intel·ligència es correspon amb el raonament i el raonament està basat en la manipulació de regles i amb estructures simbòliques. La forma com un sistema GOFAI percep i actua sobre el món té una importància secundària i roman delegada a mòduls específics. Aplicat a la robòtica, GOFAIR (*Good Old Fashion Artificial Intelligence and Robotics*) caracteritza la idea de construir sistemes robòtics fent una descomposició funcional en tres mòduls: percepció, raonament i actuació. En línies generals, els supòsits amb què es treballa en sistemes GOFAIR són [Mackworth, 93]:

- ✓ Qualsevol cosa que sigui útil per un agent és descriptible.
- ✓ Les creences que té un agent sobre el món són certes i justificades.
- ✓ El coneixement que té l'agent del món està definit i és positiu.
- ✓ El coneixement que té l'agent del món és complet. L'agent adquireix el coneixement de qualsevol cosa rellevant que passa en el món. Això permet a l'agent inferir que quelcom és fals si no pot deduir que és cert.
- ✓ L'entorn és estàtic a no ser que l'agent el modifiqui.
- ✓ Només hi ha un agent en el món.
- ✓ L'agent pot predir tots els efectes en el món de les seves accions.
- ✓ Les accions són discretes i s'executen de forma seqüenciada.

Amb aquests supòsits un agent no pot cooperar ni competir amb altres, ni necessita treballar en temps real i, ni molt menys, reaccionar a successos imprevistos. Tampoc permet treballar amb execucions concurrents d'accions ni amb events continus (processos). De fet, tots aquests supòsits venen de postular que tots els efectes d'una acció són perfectament coneguts abans que l'acció es porti a terme. En aquestes condicions, un pla consisteix en ser una llista d'accions que si s'executen portaran el món a l'estat desitjat perquè existeix un model del món que es correspon amb la realitat i perquè no hi ha més factors que intervinguin en

l'evolució del món. Evidentment un pla d'aquest tipus en un entorn real no funcionaria de cap de les maneres: un dels elements dels entorns reals més contraris als supòsits dels sistemes GOFAIR és la presència d'incerteses.

La metodologia de programar la percepció, el raonament i l'actuació en mòduls diferents no és escalable [Brooks, 88] i cal buscar estratègies reduccionistes en forma d'estructures jeràrquiques per tal d'obtenir una *integració cognitiva*. Aquesta busca ha motivat l'aparició de l'anomenada *nouvelle IA* en què el disseny dels agents es fa de manera *ad hoc*, és a dir, l'agent consisteix en ser un sistema físic real desenvolupat per treballar en un món real i que actua i reacciona en temps real. Són els anomenats *Situated Agents*¹⁷ (SA). Aquests agents poden treballar en entorns multi-agent i en móns dinàmics gràcies a una percepció dinàmica [Mackworth, 93]. En aquest sentit, es franquegen tots els supòsits dels sistemes GOFAIR.

Aquesta *integració cognitiva* obliga a dissenyar i a implementar els sistemes robòtics en un mateix entorn informàtic de treball en què hi hagi eines per a treballar amb metodologies de la IA i amb metodologies de la teoria de control. Ara bé, el problema de lligar la IA amb la teoria de control està en què no hi ha interfícies entre l'alt nivell en què es treballa a la IA i el baix nivell de la teoria de control¹⁸. La coordinació entre aquests dos nivells no s'entén del tot i això provoca que el comportament d'un sistema d'aquest tipus sigui difícil d'analitzar [Randall, 00]. Per resoldre aquest problema s'ha proposat la creació de teories unificades amb fonaments matemàtics i amb formalismes computacionals adequats com la proposada a [Zhang, 94]. Una altra opció és intentar barrejar en una mateixa arquitectura el coneixement numèric (propi del processat de senyal, de la teoria de control, i de la dinàmica de sistemes) amb el coneixement simbòlic (propi del raonament dialèctic, dels autòmats, de les estructures d'arguments, i de la planificació), com són les arquitectures proposades per Asada [Asada, 97] i l'anomenada InteRRaP [Müller, 96]. Aquesta tesi s'emmarca en aquesta segona opció: **el disseny intern de l'agent consisteix en una estructura modular amb diferents nivells d'abstracció de forma que el funcionament intern es basa en el flux d'informació a través dels diferents mòduls.**

¹⁷ Una possible traducció al català podria ser *Agents Situats*.

¹⁸ A part de la diferència del grau d'abstracció, cal tenir en compte que la IA es basa en el càlcul *off-line* i el control es basa en models computacionals *on-line*.

3.2. L'agent físic dinàmic

Els agents que porten a terme tasques en entorns dinàmics i multi-agent (anomenats **agents físics**) han de complir els requisits que imposa el funcionament en temps real i en un món real. Alguns d'aquests requeriments són els següents [Asada, 97]:

- Saber-se comportar d'acord amb les circumstàncies (*situated behavior*), i saber reaccionar davant d'esdeveniments no previstos.
- Tenir un comportament dirigit cap a un objectiu (*goal-directed behavior*) i saber seleccionar les accions en funció d'aquest objectiu.
- Les tasques s'han de realitzar amb eficiència tot i tenint en compte les imposicions restrictives del treball en temps real.
- Han de tenir present la presència d'altres agents i saber-se comportar de forma cooperativa.

Un problema important a resoldre està en com es poden integrar aquests requeriments en una arquitectura agent. Tot i que hem vist que hi ha diversos treballs en aquesta direcció, encara hi ha un problema bàsic que no ha estat resolt ni plantejat: **tenint en compte que el mot 'dinàmica' es refereix tant a rapidesa del canvi de l'entorn com a la dinàmica del propi cos de l'agent, la qüestió a plantejar és com es pot especificar la dinàmica del cos físic d'un agent, o en altres paraules, quines són les especificacions mínimes necessàries per englobar tots els requeriments abans esmentats en una sola arquitectura?** Per a la resolució d'aquesta qüestió, els cossos físics tenen un paper important: el de conduir el sistema agent a una interacció significativa amb l'entorn físic, complex, incert, però que es regeix per un conjunt de restriccions. El significat de "tenir un cos físic" es pot resumir així:

- Les capacitats sensorials i les d'actuació estan molt lligades.
- Per a l'execució de les tasques encomanades, els espais sensorials i d'actuació han de ser abstrerts dintre dels recursos de l'agent (memòria, potència de processament, controladors, etc.).
- L'abstracció depèn de les interaccions de l'agent amb l'entorn.
- Fruit de l'abstracció, cada agent té un model de representació de l'entorn.

- En el món real, tant les interaccions agent-agent com agent-entorn són asíncrones, paral·leles i arbitràriament complexes. Els agents físics evolucionen de forma continuada a temps continu.
- La complexitat natural de les interaccions físiques genera distribucions fiables de dades per als algorismes d'aprenentatge dels agents físics, en comparació als agents software.

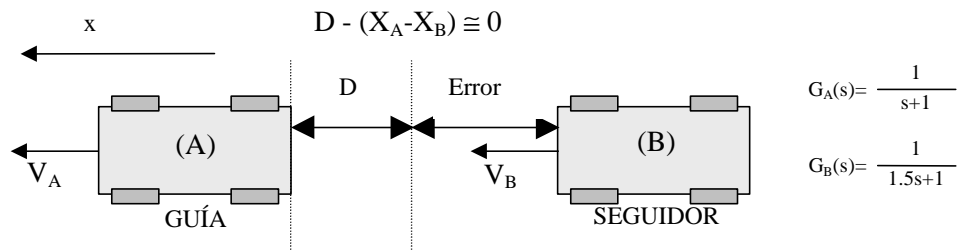
Per tant, el que cal resoldre són les següents problemàtiques:

- **Percepció:** Els agents físics han de tenir la capacitat d'utilitzar una percepció àmplia, discriminar altres agents físics, estimar les seves posicions i la seva pròpia. La percepció és una aplicació bàsica.
- **Accions:** L'agent físic ha de ser capaç de controlar el seu cos físic. En el cas de robots mòbils, ha de ser capaç d'executar trajectòries, seguir objectius, els quals no han de ser necessàriament estàtics.
- **Situació i comportament:** Encara que les tasques a realitzar siguin simples, casi apareixen infinits situacions degut als continus canvis de l'entorn, com per exemple els objectius mòbils o que altres agents físics es desplacin en el mateix entorn, etc.
- **Temps real:** Com que la situació canvia de forma continuada, hi ha restriccions de temps en la presa de noves decisions, re-planificació de trajectòries, i adequació dels controls.
- **Plataforma:** Cal decidir quina és la plataforma d'experimentació dels agents físics que faciliti la recerca sobre aquests agents.

3.2.1. Exemple il·lustratiu: comboi de dos vehicles

Quan dos vehicles han de mantenir una distància de separació en una cua poden aparèixer problemes de col·lisió, sobretot quan apareixen canvis en les velocitats dels vehicles. Normalment, la responsabilitat d'evitar la col·lisió recau en el vehicle del darrera, el qual ho aconsegueix modificant el seu comportament dinàmic accelerant o frenant tenint en compte la variació de la distància de separació amb el del davant i el llum vermell indicatiu d'inici de frenada. Tot i així, hi ha situacions en què la col·lisió és inevitable, sobretot a causa del desconeixement de la dinàmica de l'altre vehicle.

Amb una simulació numèrica es pot veure que es pot preveure *a priori* si hi haurà col·lisió en una frenada a partir del coneixement de les dinàmiques dels dos vehicles [Oller, 99b]. En una primera aproximació es considerarà un moviment rectilini i amb velocitats dels vehicles amb comportament completament lineal (sense canvis de marxa ni patinades a la frenada) i representable per funcions de transferència de primer ordre. Per a poder treballar amb agents autònoms haurem de suposar que cada vehicle disposa d'un sistema de control de velocitat intern; suposarem que tenen el mateix sistema de control i que les funcions de transferència $G_A(s)$ i $G_B(s)$ que representen les velocitats en llaç tancat són de primer ordre (les velocitats tenen un comportament exponencial). Què passa quan posem els vehicles l'un darrere de l'altre? El vehicle del darrera (sistema B) ha d'aconseguir que es compleixi la següent especificació:



, on X_A i X_B són les distàncies recorregudes pels dos mòbils i D la distància a mantenir entre ells. Com ho aconsegueix? Sense perdre generalitat, usarem un control proporcional, és a dir, la consigna de velocitat per a B vindrà donada per l'expressió següent: $K_P \cdot (D - (X_A - X_B))$, on K_P és una constant (veure figura 3.2.1.1).

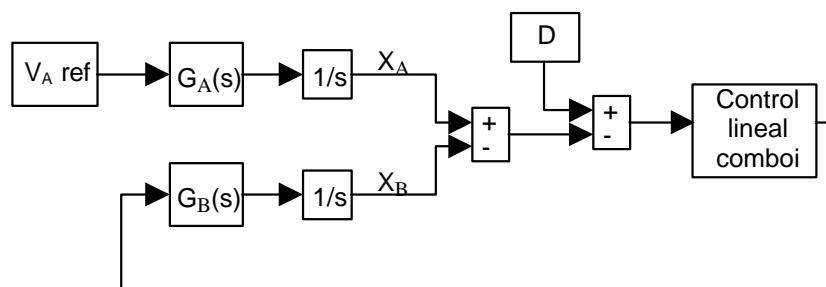


Fig. 3.2.1.1: Diagrama de blocs del sistema de control.

Amb una simulació d'aquest sistema s'observa que la velocitat del sistema B modifica la dinàmica i passa a ser de segon ordre. Suposem les condicions inicials $D=5\text{m}$, $V_A(t=0)=8\text{m/s}$ i $V_B(t=0)=8\text{m/s}$, $X_A=5\text{m}$. (els vehicles mantenen inicialment la distància D en un règim de 8 m/s). Quan el vehicle A redueix la velocitat a $V_A=4\text{m/s}$ s'observa un comportament de 2ⁿ ordre per al sistema B. També pot

observar-se que en aquestes condicions es produeix una col·lisió posterior (veure figura 3.2.1.2).

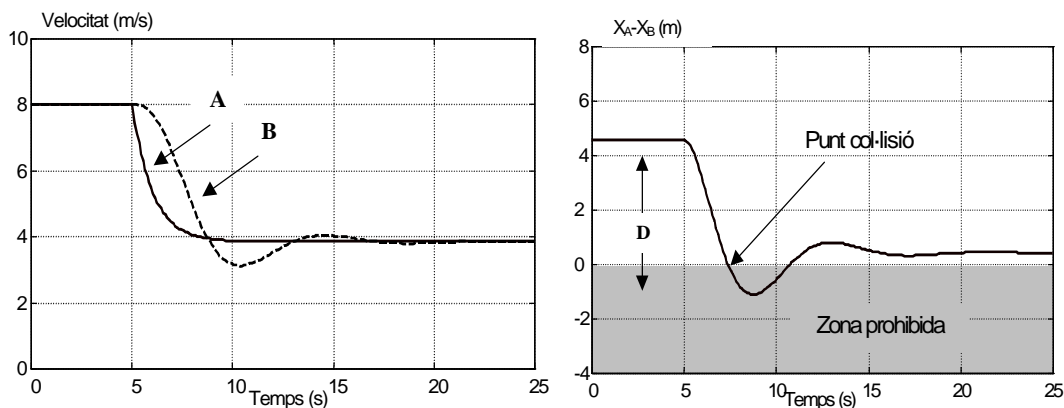


Figura 3.2.1.2: (a) Velocitats dels sistemes A i B, (b) Visualització de l'instant de la col·lisió.

El que pot deduir-se d'aquest exemple¹⁹ és que encara que el sistema A tingués la possibilitat de comunicar al B la intenció de frenar fins a 4 m/s, hauria de dir-li de quina manera ho fa. Concretament, hauria de subministrar-li informació sobre la seva dinàmica de frenada: temps de frenada (temps de resposta del sistema A), possibles oscil·lacions (ordre del sistema), velocitat final real (error en estat estacionari), etc. De fet, per fer una frenada sense problemes caldria una negociació entre els dos sistemes amb l'ús d'informació que inclogui tant paràmetres estàtics com dinàmics del procés.

3.2.2. Extensió del concepte d'agent software al concepte d'agent físic dinàmic

D'acord amb les especificacions de la FIPA, “*un agent és l'actor fonamental en una plataforma agent (AP) la qual combina diverses capacitats de servei en un model d'execució integrada i unificada. L'agent existeix físicament en una AP i usa els serveis que li ofereix. Cal que l'agent estigui registrat en una AP per tal de poder interactuar amb els agents de la mateixa plataforma o d'una altra*”. Aquestes són les especificacions que donen lloc al model de referència en què els agents *software* treballen i existeixen. Cal observar que el terme ‘físic’ es refereix a la infraestructura física²⁰ en què els agents es despleguen.

¹⁹ En el següent capítol s'insistirà en aquest exemple i es donarà una solució al problema de la col·lisió.

²⁰ Típicament, una infraestructura informàtica.

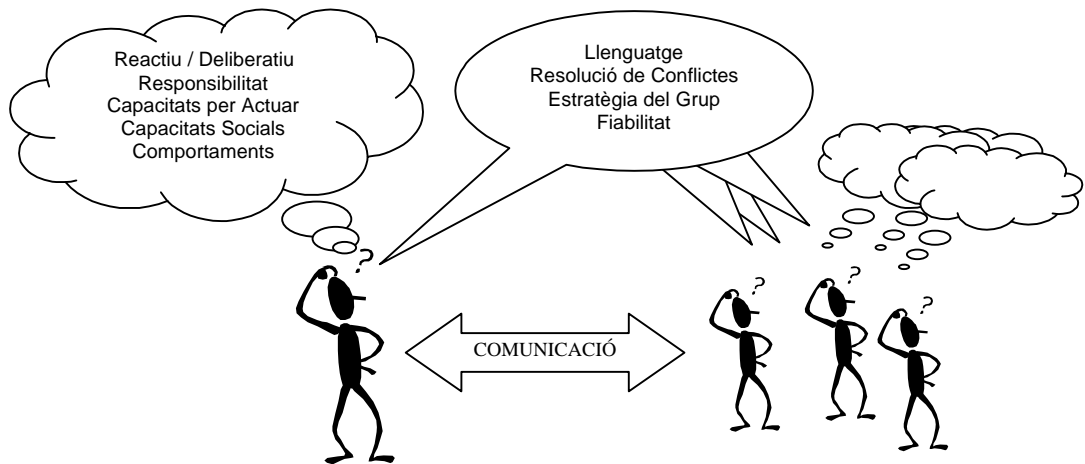


Figura 3.2.2.1: Esquema d'un sistema multi-agent amb agents software.

Com a explicació informal del funcionament d'un agent *software* (figura 3.2.2.1) podríem donar la següent:

Definició 1: funcionament dels agents software. Suposant que un agent és incapaç de portar a terme una tasca tot sol, obtenim que cada agent ha de recórrer als altres per obtenir-ne ajuda. Aleshores, cadascun d'ells ha de decidir el següent: “A quin altre agent puc ajudar?”, “Fins on arribarà el meu compromís envers ell?”, “Sota quines condicions puc fer-ho?”. Tot i que la decisió final podrà ser reactiva o deliberativa, quan l'agent es compromet ha de saber el que impliquen els seus compromisos. En altres paraules, ha de saber si pot fer-ho o no.

Feta aquesta definició, la pregunta que ens plantejem és la següent: on es pot introduir el concepte del cos físic de l'agent? La resposta està en la definició de l'**agent físic**.

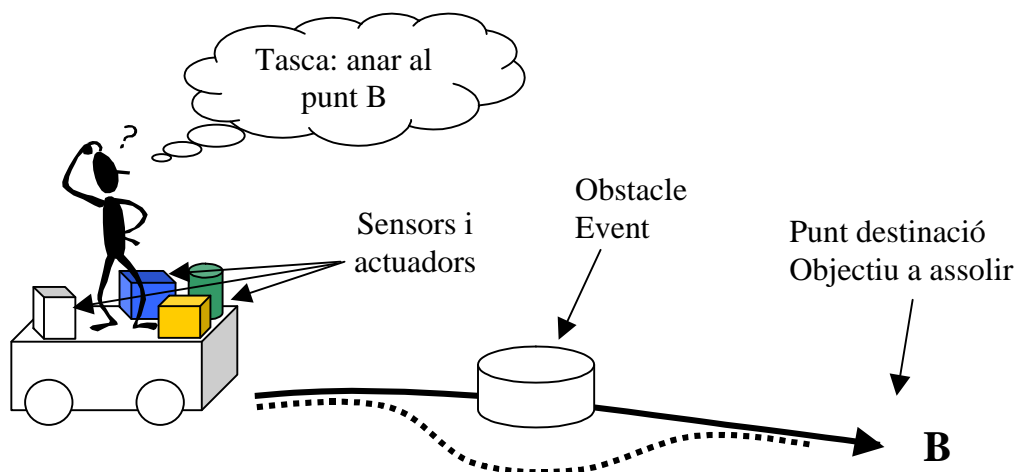


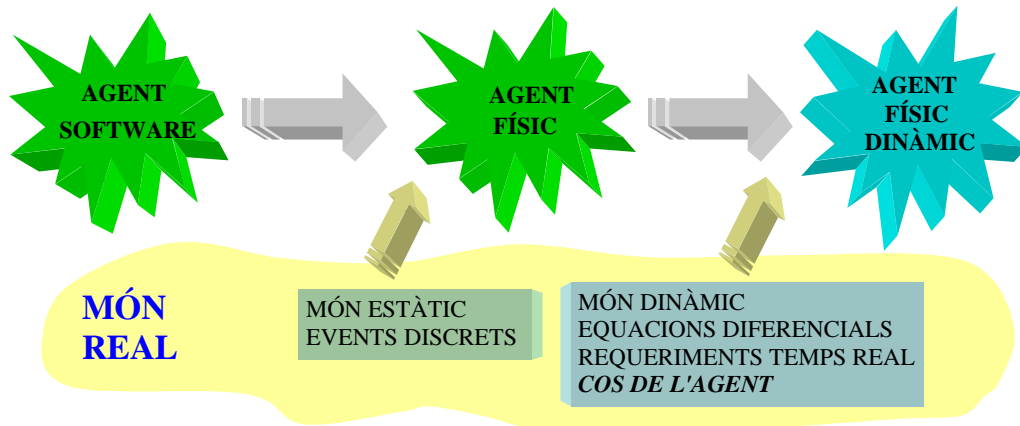
Figura 3.2.2.2: Esquema d'una problemàtica típica d'un agent físic.

Com a explicació informal del funcionament d'un agent físic podríem donar la següent:

Definició 2: Extensió del concepte d'agent software a agent físic. La definició 1 i el que segueix: per saber què és possible fer i què no, cal tenir en compte algun coneixement físic del sistema (robot i entorn). Això significa que les entrades i les sortides físiques cap i des de l'entorn han d'estar integrades en la base de coneixement de l'agent. Això és així perquè l'agent està contingut en un cos físic que ha de controlar i moure mitjançant les decisions.

Encara que el funcionament de l'agent físic pugui semblar adequat, no té en consideració el fet que l'agent ha de ser capaç de controlar el seu propi cos físic, és a dir, ha de poder seguir correctament trajectòries predefinides o perseguir i evitar obstacles, etc. En definitiva, ha de poder realitzar qualsevol tipus de tasca que pot contenir infinites situacions a causa de l'aparició continuada de canvis de l'entorn (canvis de naturalesa discreta i/o contínua) i a causa dels seus propis moviments (accions) a l'entorn. Els treballs actuals amb robots reals no avaluen la problemàtica associada a la dinàmica del propi robot i l'únic que fan és interpretar els problemes que apareixen des del punt de vista de la supervisió o de la detecció de falles [Beckers, 94][Cao, 95][Steels, 94]. Per tant, si tenim en compte que els moviments dels robots es poden descriure amb equacions diferencials, podem concloure que la teoria del control automàtic és necessària per a l'anàlisi de la dinàmica dels agents físics i dels seus efectes a les etapes de decisió.

Definició 3: Extensió del concepte d'agent físic a agent físic dinàmic. Definició 2 i el que segueix: el coneixement físic s'obté a partir de la dinàmica del cos físic, i es representa de forma declarativa a través de les capacitats. En concret, **una capacitat a nivell de control consisteix en ser la percepció de les qualitats d'un controlador**. Per tant, les capacitats integren tant la identificació del cos físic com la percepció de l'entorn a través del cos i, en última instància, poden integrar també l'adaptació de l'agent físic a l'entorn, és a dir, l'habilitat de poder aprendre i adaptar-se als canvis.



3.2.3. L'arquitectura InteRRaP per a agents físics

Aquesta arquitectura parteix de la constatació que els estudis amb agents es centren o bé en els aspectes de com combinar la reactivitat amb la deliberació (aspectes individuals), o bé en modelar la interacció/cooperació entre agents (aspectes col·lectius). La idea de InteRRaP és poder treballar amb agents que tinguin comportament situat, dirigit cap a un objectiu, amb eficiència i amb capacitats de coordinar-se, és a dir, una arquitectura que integri tant els aspectes individuals de l'agent com els col·lectius abans esmentats.

La definició de l'agent es fa basant-se en tres nivells de coneixement: *behavior-based layer* (BBL), *local planning layer* (LPL) i el *cooperative planning layer* (CPL). La tècnica basada en nivells permet integrar diferents nivells d'abstracció i, per tant, permet combinar el raonament reactiu amb el deliberatiu. L'agent respon a una arquitectura derivada del model BDI de Rao i Georgeff [Rao, 91,93] en què l'estat d'un agent es descriu a partir de creences, objectius, intencions i plans, i on la percepció es relaciona amb l'actuació mitjançant un conjunt de funcions que expliquen les relacions entre les categories mentals de l'agent. En molts aspectes el model BDI respon a la metodologia de la programació orientada a agents proposat per Shoham [Shoham, 90]: les diferències estan relacionades amb la nomenclatura de les categories mentals i en la forma interna d'obtenir les decisions. El model conceptual de l'agent està basat en el model de Bratman [Bratman, 87] en què l'agent té una estructura per nivells. Defineix les categories mentals que determinen l'estat intern de l'agent (percepció, creences, situacions, objectius, intencions i primitives operacionals), les funcions bàsiques que modifiquen els estats mentals (reconeixement de la situació, activació d'objectius,

planificació, el *scheduling* i l'execució. Cada nivell de control pot accedir a una part de la base de coneixement. Aquest accés està organitzat de forma incremental de forma que cada nivell de la unitat de control pot accedir a la part corresponent de la base de coneixement i a les parts corresponents als nivells immediatament inferiors. El funcionament de cada nivell també té en compte la comunicació amb els altres nivells, és a dir, els processos SG i PS canvien l'estat intern de l'agent tenint en compte la possibilitat que el nivell immediatament inferior hagi enviat algun missatge demanant una decisió a més alt nivell. Així, cada nivell pot decidir si és competent per decidir una nova acció i deixar-la activada.

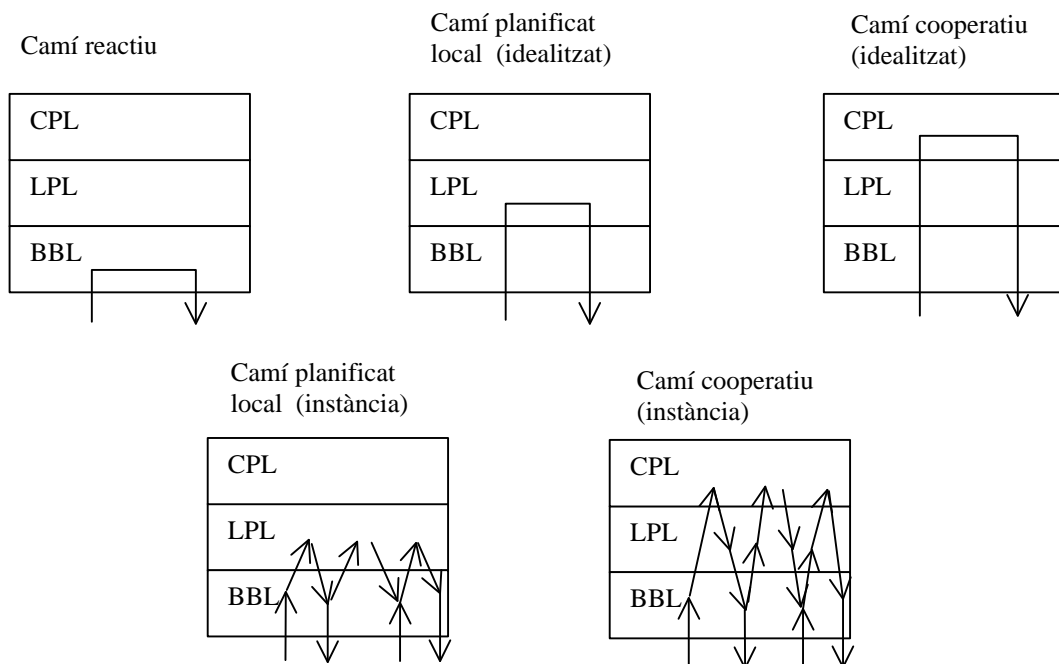


Figura 3.2.3.2: Fluxos de control entre els nivells de InteRRaP

El funcionament d'un agent InteRRaP és el resultat de la interacció entre els diferents nivells de control a partir de dos mecanismes d'activació: la demanda d'activació ascendent, la comanda de compromís descendent. La primera es produeix quan en un nivell el procés SG_i reconeix una nova situació i el procés PS_i es reconeix com a no competent per a decidir cap acció. Aleshores, s'envia un missatge al procés SG_{i+1} del nivell superior. L'avantatge d'aquest mecanisme està en el fet que els nivells inferiors no necessiten saber res sobre les capacitats dels nivells superiors: només necessiten saber si poden resoldre o no una determinada situació. Respecte l'altre mecanisme d'activació, es produeix quan un determinat nivell determina l'acció a realitzar i vol posar-la en marxa. En aquest sentit, el PS_i envia un missatge de

comandament al PS_{i-1} i es queda a l'espera d'una confirmació per saber si l'acció és realitzable pel nivell immediatament inferior o si necessita decidir una nova acció.

A partir d'aquests dos mecanismes s'obtenen cinc possibles fluxos de control entre els nivells (veure figura 3.2.3.2). Es pot observar que a causa del primer mecanisme es generen tres possibles fluxos o camins a través dels nivells: el camí reactiu (per a situacions d'emergència o sense necessitat de planificació), el camí amb planificació local (per a situacions on cal planificar una tasca concreta), i el camí cooperatiu (situacions on la resolució requereix la cooperació amb altres agents). Els altres dos camins possibles apareixen quan la planificació i l'execució es processen en diferents nivells en diferents instàncies.

Aquesta arquitectura s'ha utilitzat en diversos exemples robòtics reals i simulats on s'ha demostrat la seva capacitat de funcionament. S'han realitzat diversos estudis comparatius amb altres arquitectures conegudes com la horitzontal de subsumpció [Brooks, 86], AGENT-0 [Shoham, 90], dMars [Georgeff, 87], ALP [Kowalski et al, 96] i altres. S'ha vist que InteRRaP és una bona metodologia de disseny d'agents físics híbrids que integra diferents funcionalitats en mòduls separats, i que modela bastant bé la transició de la reactivitat cap a la deliberació a través d'una estructura per nivells i un control jeràrquic.

PART II:

DISSENY D'AGENTS FÍSICS

4. Descripció de l'arquitectura DPAA

Hem vist que els agents que treballen en entorns dinàmics i multi-agent han de complir certs requeriments per a poder treballar en un món real i en temps real, els quals han d'estar implícits en la seva pròpia arquitectura. Hem vist també que les arquitectures híbrides més importants que treballen amb agents físics necessiten concentrar-se en dos aspectes: d'una banda la necessitat de barrejar el comportament reactiu (actes reflexos) amb el deliberatiu (actes cognitius amb representació simbòlica del món), i de l'altra la necessitat de modelar tant la interacció com la cooperació entre els agents. L'arquitectura DPAA (*Dynamical Physical Agents Architecture*) està basada en l'arquitectura híbrida InteRRap [Müller, 96], i la seva estructura està orientada a complir els següents requeriments:

- Comportament situat → Els agents han de saber reconèixer els events asíncrons i reaccionar a temps i de forma apropiada tenint sempre en compte les possibilitats dinàmiques del cos físic.
- Comportament vers un objectiu → Els agents han de seleccionar les accions basant-se amb el que volen assolir a nivell individual i després d'haver adquirit el compromís de fer-ho respecte a la col·lectivitat.
- Eficiència → Les tasques s'han d'executar eficientment, tenint en compte les possibilitats físiques reals que tenen per aconseguir-ho, tant les temporals

(restriccions de temps real) com les espaials (limitacions cinemàtiques i dinàmiques).

- Coordinació → Els agents han de tenir presents les interaccions amb altres agents.

4.1. Funcionament d'un mòdul: primera aproximació

En totes les arquitectures RTAIS que hem vist, s'observa la presència d'una estructuració o separació en mòduls independents que s'encarreguen de gestionar la percepció, el raonament i l'actuació, separació que reflexa la tendència de l'ús d'agents independents per a cascuna d'aquestes tasques. En referència a InteRRap [Müller, 96], hem vist que utilitza una arquitectura amb una jerarquia vertical indicativa del grau d'abstracció de la informació que es processa: un nivell és superior a un altre si processa informació més abstracte. Quant al flux d'informació entre nivells, cada nivell té la possibilitat de gestionar informació de quatre tipus:

- Pot **rebre ordres** d'un nivell superior que indiquen quina és l'acció a portar a terme.
- Pot **rebre informació** de nivell inferiors que l'ajuda a entendre (percebre) el món.
- Pot **enviar ordres** de comandament a nivells inferiors.
- Pot **informar** a nivells superiors sobre l'estat del sistema.

Per tant, cada nivell té quatre canals d'informació: dos d'entrada i dos de sortida tal com es mostra a la figura 4.1.1:

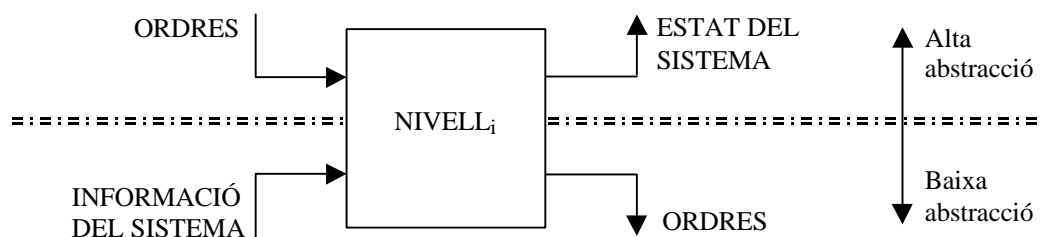


Figura 4.1.1: Canals d'informació associats a un nivell: primera aproximació cap a DPAA

El que es proposa utilitzar per a l'arquitectura DPAA és també una estructura jeràrquica vertical per nivells inspirada en InteRRap. Els canvis que s'incorporen intenten donar una concepció més modular a l'arquitectura global partint d'una concepció interna del *layer* diferent: els *layers* es reconfiguren en forma de mòduls i

tots ells responen a una mateixa estructura interna i de funcionament. Aquesta concepció dóna un caràcter més modular que, com es veurà, facilita el procés de disseny de l'agent físic.

La forma com s'ha decidit estructurar la informació i el seu flux a través dels mòduls està basada en un punt vista de disseny *bottom-up*, és a dir, considerant que el bon funcionament a alt nivell està condicionat per les bones condicions de funcionament a baix nivell. Així, el mòdul prioritari serà aquell relacionat amb els aspectes del control del moviment de l'agent físic: aquest mòdul s'anomenarà *mòdul de control*.

Els mòduls tindran quatre canals d'informació tal com es mostra a la figura 4.1.2 i per saber quines peticions d'informació pot haver-hi damunt d'un mòdul qualsevol, anem a veure en primer lloc com ha de treballar el *mòdul de control*, és a dir, el mòdul amb menor abstracció de dades i al nivell més inferior:

Recepció d'ordres. Un mòdul superior pot encomanar al de control el seguiment d'una consigna (o d'un conjunt de consignes) amb l'ús d'un controlador concret (o d'un conjunt de controladors). Ara bé, aquest mòdul superior ha de poder saber amb anterioritat si aquest seguiment és factible i amb quin grau de precisió s'executarà. Per tant, el mòdul de control també ha de poder rebre preguntes relacionades amb la qualitat de l'execució de la següent acció.

Percepció. La percepció és necessària per a poder portar a terme les ordres comentades anteriorment. La percepció inclou l'adquisició de les dades de sensors analògics i digitals, la qual cosa pot permetre avaluar variables contínues i events discrets. Aquesta percepció no inclou cap tipus d'interpretació, és a dir, són els senyals directes de l'entorn que recull el mòdul de control.

Actuació. Sempre hi haurà actuació del mòdul de control mentre l'estat actual i el de referència no coincideixin, és a dir, mentre l'objectiu de control no s'hagi assolit. Tant la percepció com l'actuació poden estar en taules o ser simulades amb models interns del món, és a dir, el mòdul de control també podria ser capaç de simular el resultat d'una acció de control a partir d'un estat inicial i així analitzar les característiques de la dinàmica de l'acció.

Informació del sistema. El mòdul ha de ser capaç d'informar instantàniament al mòdul superior quan s'ha assolit l'estat final desitjat. Tanmateix, també ha d'informar dels events o problemes imprevistos que poden aparèixer durant l'execució de l'acció de control. Com ja s'ha comentat a l'apartat de 'Recepció

d'ordres', la possibilitat de rebre preguntes va acompanyada de la capacitat de contestar-les. Per tant, el subministrament automàtic d'informació sobre l'estat del sistema ha d'anar precedit per una sol·licitud del mòdul superior. Aquesta informació subministrada s'elabora a partir dels resultats simulats o en taules, i consisteix en un seguit de paràmetres que donen característiques tant del règim permanent com del transitori de l'acció simulada. Cal fer notar que aquesta simulació no serà un procés a executar cada vegada que hi hagi la petició sinó només en alguns casos, és a dir, normalment aquest subministrament d'informació serà immediat gràcies a la disposició de taules.

El problema que es presenta ara està en com gestionar tota la informació que arriba a un mòdul. Com que el mòdul podria treballar concurrentment amb altres mòduls, caldrà establir un mecanisme de activació/desactivació dels canals d'informació del mòdul en funció de les peticions externes que hi hagi, és a dir, un mecanisme que permeti mantenir la coherència de la informació.

Un cop vistes les possibles peticions a mòdul que poden haver-hi i vista la necessitat d'un mecanisme de activació/desactivació dels canals d'informació, pot deduir-se que el mòdul de control ha de poder treballar de dues maneres bastant diferenciades: una manera consisteix en aplicar una llei de control utilitzant la percepció com a mecanisme de realimentació, i l'altra consisteix en utilitzar informació interna per a subministrar informació al mòdul superior. Donat que els mòduls tenen tots la mateixa estructura, la forma de funcionament dels demés mòduls pot obtenir-se fent un procés d'escalat d'aquest mòdul de control.

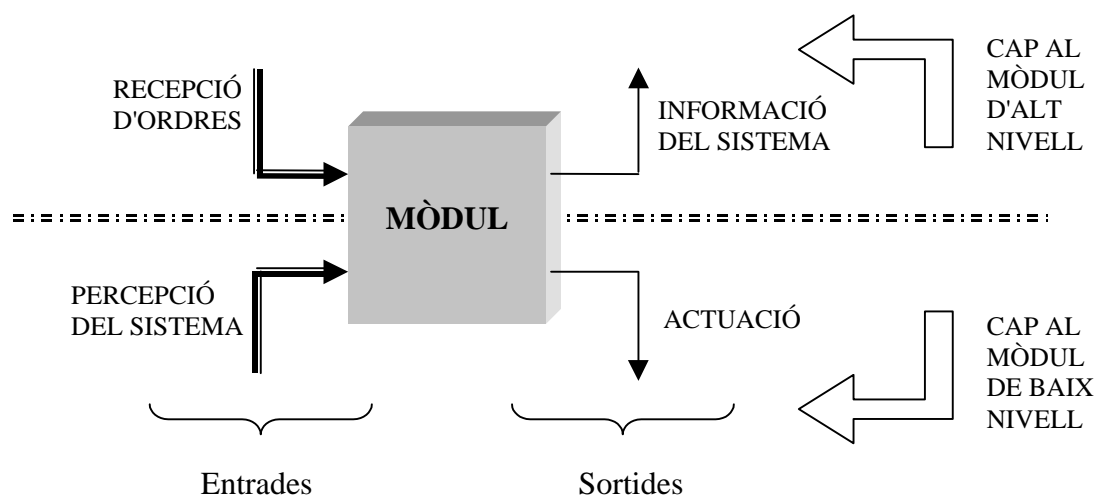


Figura 4.1.2: Tipus d'informació que ha de gestionar un mòdul de la DPAA

4.2. Arquitectura de Control Global

L'arquitectura DPAA (*Dynamical Physical Agents Architecture*) [Oller, 99a] descompon el sistema de control global en diferents nivells d'abstracció proporcionant accés als senyals externs no interpretats al nivell més baix i amb sofisticats models del món accessibles al nivell més alt. Aquesta arquitectura utilitza un conjunt de mòduls similars entre ells i que van interconnectats per diversos tipus de canals d'informació: el canal de **percepció**, el canal de **comportament**, el canal **d'actuació** i el canal **d'introspecció** tal com mostra la figura 4.2.1:

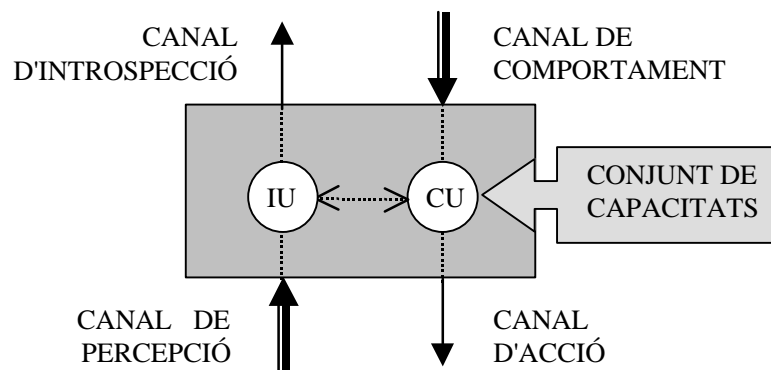


Figura 4.2.1: Esquema d'un mòdul de la DPAA

D'aquesta manera, el sistema de control global consisteix en una cadena de mòduls¹ cadascun d'ells usant informació amb diferent grau d'abstracció respecte dels altres. Els canals d'informació serveixen per a transmetre missatges els continguts dels quals es processen internament en el mòdul fent ús de dues unitats de procés intern (figura 4.2.1): la **unitat d'introspecció (IU)**, i la **unitat de control (CU)**. Amb l'ús de diversos protocols de conversa es possibilita que mòduls amb alt nivell d'abstracció puguin treballar amb informació que s'ha preprocessat prèviament en mòduls de grau d'abstracció més baix. Cal remarcar que només està permesa la comunicació entre mòduls veïns, és a dir, el pas d'informació sempre es produeix entre mòduls enllaçats directament.

Encara que els mòduls siguin estructuralment iguals i que, com es veurà més endavant, la forma de funcionament sigui la mateixa, els mòduls tenen un nucli

¹ Cal explicitar el fet que la modularitat de l'arquitectura permet interpretar-la des d'un punt de vista diferent: enlloc de parlar de mòduls encadenats amb estructures equivalents es pot parlar d'un conjunt d'agents homogenis integrats en una estructura jeràrquica. Aquesta equivalència serà important més endavant quan es parli dels protocols de comunicació pel pas de missatges entre els mòduls.

específic que els diferencia dels demés: un **conjunt específic de capacitats** predefinides. Aquestes capacitats són les que permeten establir la relació inherent entre els comportaments i les corresponents accions en el mateix mòdul.

L'arquitectura que es proposa en aquest treball intenta encabir els requeriments de les arquitectures dels sistemes de control amb aquelles de les arquitectures per a sistemes multi-agent. En concret, DPAA proposa l'ús de tres mòduls (figura 4.2.2):

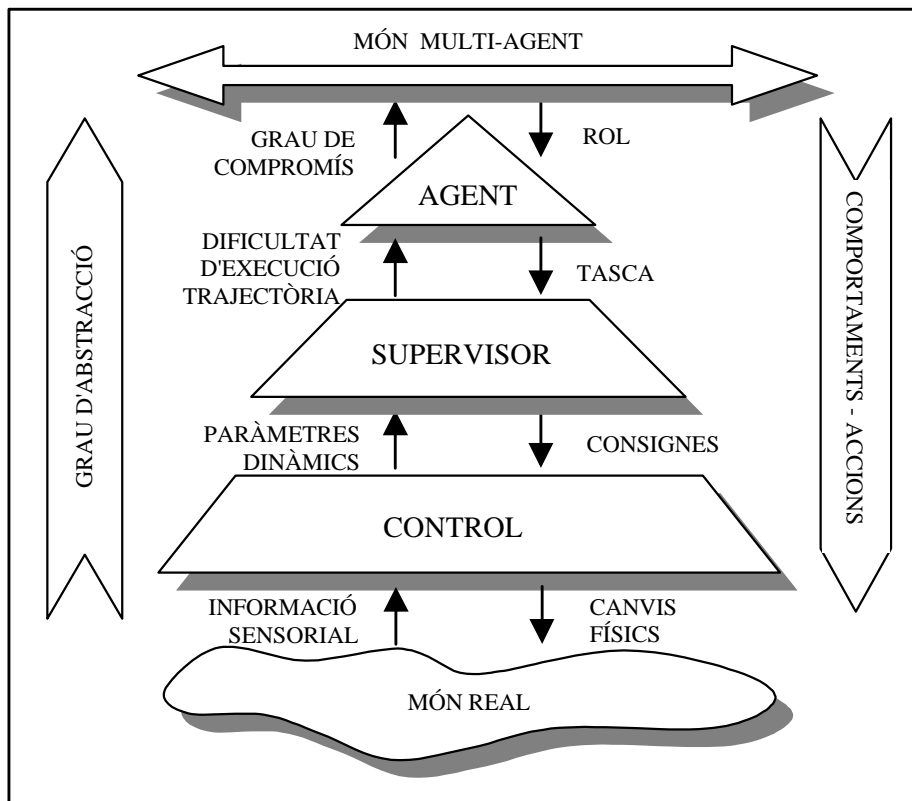


Figura 4.2.2: Esquema de la jerarquia modular de la DPAA

- I. Mòdul de Control: és la connexió directa de l'agent físic al món real. Conté els sensors i els actuadors del cos físic i les lleis de control que fan que el cos físic segueixi les consignes. Proporciona informació relativa al comportament dels controladors.
- II. Mòdul Supervisor: és la interfície entre els paràmetres del món real i el món lògic de l'agent. Conté la informació necessària per a seleccionar els controladors i llurs paràmetres que són necessaris per a executar diferents comportaments predeterminats. Proporciona informació simbòlica a partir de paràmetres numèrics del mòdul de control.

- III. Mòdul Agent: consisteix en ser la connexió de l'agent físic cap el món multi-agent (la col·lectivitat). Conté les funcions necessàries per a poder identificar les diferents situacions en què es troba, i conté la lògica de decisió i de negociació associada a cada situació.

Com es pot veure a la figura 4.2.2, a mida que el grau d'abstracció augmenta, la naturalesa dels comportaments esdevé més sofisticada i tendeix a ser directament independent dels condicionants que imposa el món real. De fet, la informació que es transmet entre mòduls és cada vegada més simbòlica. De forma paral·lela, a mida que el grau d'abstracció disminueix les accions es degraden fins a consistir en simples canvis físics en els actuadors del cos físic de l'agent. En definitiva **es tracta d'una estructura que permet traduir la informació numèrica a simbòlica a través d'un procés d'introspecció, i que també permet traduir la informació simbòlica a numèrica a través d'un procés d'actuació.**

4.2.1. Funcionament intern dels mòduls

Quan el mòdul agent està decidit a fer una tasca després d'haver-la negociat amb els altres agents és quan el sistema de control global s'activa i comença a funcionar. Aquesta decisió depèn tant d'informació externa a l'agent (de l'entorn o d'altres agents) com dels estats interns de l'agent. La informació externa la pot obtenir de dues maneres: o bé intercanviant-la a través dels mecanismes de comunicació que hi ha entre els agents, o bé a través d'un mecanisme intern que interpreti el que està passant a baix nivell. D'altra banda, els estats interns han de poder descriure estats del cos físic de l'agent des d'un punt de vista tant d'alt com de baix nivell— per exemple, paràmetres sobre la dinàmica d'un robot mòbil quan està accelerant o un paràmetre que indiqui que l'execució d'una acció determinada té un grau de dificultat del 30%. Per tant, el mode de funcionament ha de complir el següent requisit: **els mòduls amb alt grau d'abstracció han de poder obtenir informació dels mòduls de baix grau d'abstracció per tal de poder incloure-la en els processos de decisió. Aquesta informació ha de servir per poder determinar la dificultat de l'execució d'una acció.**

Per altra banda, hi ha un altre procés a tenir en compte i és el que s'encarrega de convertir les decisions d'alt nivell en un llenguatge simbòlic a especificacions numèriques dels controladors a baix nivell. Aquesta conversió s'ha de fer de forma

esglaonada ja que les ordres passen a través de diversos mòduls fins arribar al de control on esdevenen consignes de velocitat de rodes motrius. Per tant, el mode de funcionament ha de complir el següent requisit: **les ordres encomanades des de mòduls d'alt nivell d'abstracció s'han d'anar convertint a ordres amb formes més numèriques que simbòliques a mida que van passant cap a mòduls amb baix grau d'abstracció.**

És per aquests dos requisits que s'han definit dos modes de treball modular. Els mòduls estaran treballant en una modalitat o en l'altra en funció del tipus de missatge que rebin a través del canal de comportament: missatge de demanda ('*query*') o de comanda ('*request*').

4.2.1.1 Mode d'Introspecció (*I-mode*)

El mòdul es posa en aquest mode de funcionament quan rep un missatge de demanda '*query*' (veure la figura 4.2.1.1). En aquest mode de funcionament, el mòdul pot ajudar al mòdul amb grau d'abstracció immediatament superior a obtenir dades perceptuals de nivell més baix. El mòdul emissor del missatge demana quin és el grau de dificultat d'executar una acció determinada en un temps específic i el mòdul receptor ha de consultar el conjunt de capacitats per tal de poder respondre a la pregunta. El mòdul contesta a través del canal d'instrospecció i es posa en estat d'espera. Quan el conjunt de capacitats no doni prou informació aleshores el mòdul passa a funcionar a mode de control per tal d'obtenir més informació. Tot aquest procés intern es realitza amb la unitat d'instrospecció.

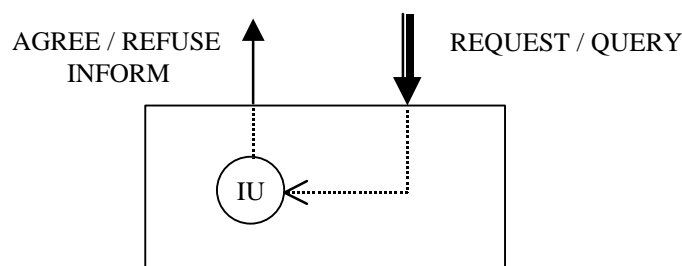


Figura 4.2.1.1: Flux de dades en mode d'instrospecció

4.2.1.2. Mode de Control (*C-mode*)

El pas a mode de control es produeix per causes externes quan el mòdul rep un missatge de comanda '*request*' (veure la figura 4.2.1.2). En canvi, el pas de *I-mode* a *C-mode* és automàtic quan al mòdul se li ha demanat una informació relativa a un

comportament i no ha pogut trobar la informació suficient dintre del conjunt de capacitats: en aquest cas envia un missatge de demanda 'query' al mòdul immediatament inferior. D'aquesta manera, el mòdul pot obtenir dades perceptuals i, tot seguit, integrar-les per tal de completar el seu conjunt de capacitats. Un cop integrades pot passar a *I-mode* i continuar el diàleg amb el mòdul immediatament superior.

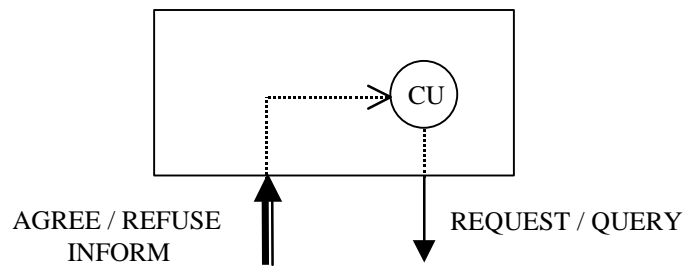


Figura 4.2.1.2: Flux de dades en mode de control

Per tant, quan el mòdul està en aquest mode pot ordenar al mòdul immediatament inferior que iniciï l'execució d'una acció determinada o pot obtenir-ne informació i actualitzar les seves capacitats.

A part d'aquests dos modes de funcionament cal comentar que quan un mòdul no està en cap diàleg, es considera que està en estat d'espera, és a dir, les unitats de control i d'introspecció estan a l'espera de l'arribada de missatges.

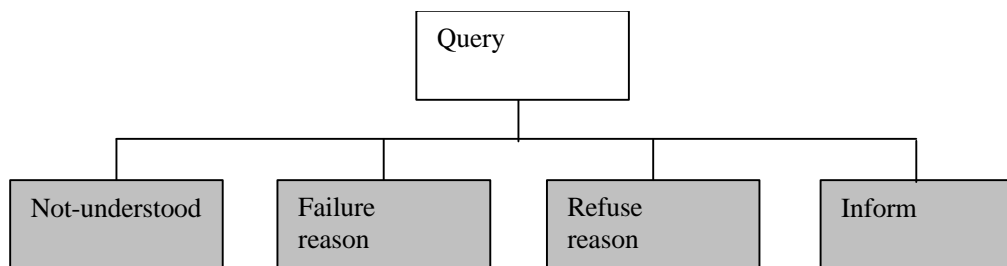
4.2.2. Missatges i protocols de diàleg

L'intercanvi d'informació i la transmissió d'ordres entre mòduls es realitza a través de pas de missatges. Hem vist que els mòduls poden funcionar de dues formes diferents i la manera com passen d'una modalitat a una altra està marcat pels tipus de missatges que s'intercanviarà o que li arribaran dels mòduls veïns. En altres paraules, el funcionament de cada mòdul dependrà de les necessitats del sistema de control global. Les seqüències de missatges que s'intercanvien dos mòduls en un diàleg segueixen diversos patrons anomenats protocols. L'equivalència que s'ha comentat anteriorment del conjunt de mòduls de l'arquitectura DPAA amb un grup d'agents ordenats en una estructura jeràrquica permet apreciar la importància d'especificar quins són els protocols de diàleg intermodular.

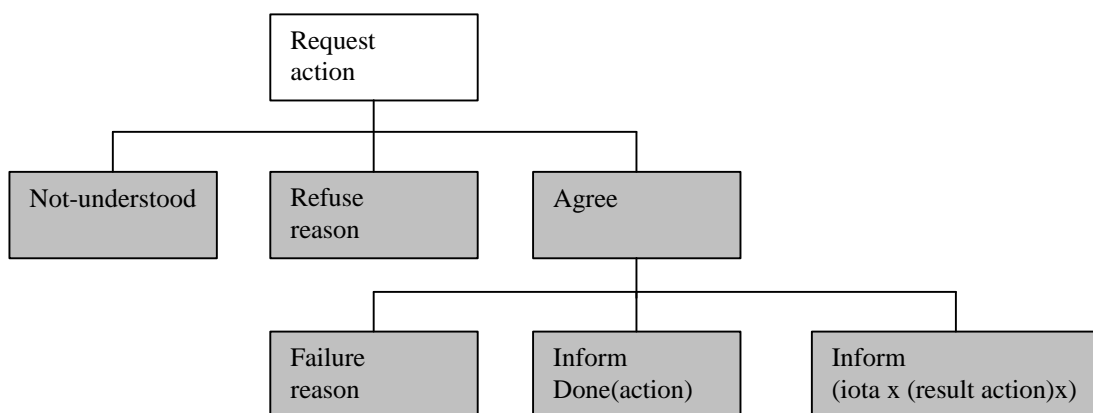
4.2.2.1. *Protocols de diàleg entre mòduls*

Hem vist que cada mòdul té dos modes de funcionament i que en qualsevol dels dos sempre hi ha o una comanda o una demanda, és a dir, un mòdul pot portar a terme quatre tipus de papers dialectals: ordenant de demanda, receptor de demanda, ordenant de comanda i receptor de comanda. Per tant, només s'han definit dos protocols²: un pel diàleg de demanda (*QUERY*) i un altre pel de comanda (*REQUEST*), els quals estan basats en els protocols *FIPA-query* i el *FIPA-request* establerts per la FIPA [FIPA, 00]:

- Protocol *FIPA-query*: el mòdul receptor rep un missatge en què se li demana que faci una acció informativa. En el nostre cas l'acció informativa serà contestar a la pregunta formulada per l'emissor³.



- Protocol *FIPA-request*: permet que un mòdul demani a un altre que executi una acció. En cas d'acceptació també avisarà quan acabi o si hi ha hagut alguna fallada. El mòdul receptor pot refusar l'ordre d'execució de l'acció i donar els motius del perquè. Aquest refús és indicatiu que té algun problema amb les seves capacitats i, per tant, no pot executar les accions relacionades.



² En color blanc es mostra l'acció comunicativa iniciada pel mòdul emissor (ordenant). En gris les accions realitzades pel receptor.

³ Per exemple, la resposta del mòdul supervisor al mòdul agent inclou un coeficient que indica el grau de dificultat que tindria l'execució de l'acció sol·licitada calculat a partir de les capacitats que té.

Un detall a tenir en compte és el missatge de confirmació que envia el mòdul contractat al gestor: aquesta confirmació és important quan la fa el mòdul de control perquè indica l'instant en què comença la nova acció de control.

4.2.2.2. Tipus de missatges

Hi ha definits dos tipus generals de missatges: els d'**interrupció** i els de **diàleg**. Els primers provoquen un canvi immediat en la modalitat de funcionament del mòdul que el rep, mentre que els altres es regeixen per les regles de diàleg o protocols.

- Missatges d'interrupció (amb prefix INT): Aquests missatges provoquen un canvi en el mode de funcionament del mòdul receptor. Més concretament, quan arriben pel canal de percepció provoca el pas a funcionament en *C-mode* mentre que quan arriben pel canal de comportament provoca el pas a funcionament en *I-mode*. En el primer cas aquest tipus de missatge requereix una atenció immediata ja que una percepció pot fer canviar una decisió presa en un mòdul amb grau d'abstracció més alt, mentre que en el segon cas aquesta atenció ha de tenir prioritats diferents per a mòduls diferents. Això es deu al fet que el pas de *C-mode* a *I-mode* pot conduir a conseqüències diferents —una aturada perillosa al mòdul de control, o un canvi de pla al mòdul supervisor.

| Canal de Comportament [<i>C-mode</i> → <i>I-mode</i>] | Canal de Percepció [<i>I-mode</i> → <i>C-mode</i>] |
|--|---|
| INT Request (Action, time) | INT Inform(Action,END) |
| | INT Inform(Action,EVENT, 'Description') |
| | INT Inform(Action,PROBLEM, 'Description') |

Taula 4.1: Llista de missatges d'interrupció: la primera fila indica el canal d'entrada del missatge i el canvi que s'indueix en el mode de funcionament del mòdul receptor.

- Missatges de diàleg: Aquests són els missatges que poden intercanviar-se entre dos mòduls veïns i que provoquen els canvis en els modes de funcionament d'ambdós fins que la conversa ha finalitzat (veure taula 4.2). La funció d'aquests diàlegs és permetre gestionar el control de les accions i el pas del valor d'algunes variables.

| Canal de Comportament | Canal de Percepció |
|---|--|
| Inform (Action, DO) | Confirm (Action, time, μ) |
| Inform (Action, STOP) | Refuse (Action, 'Reason') |
| Request (Action, 'Variable', time, value) | Inform (Action, START, time, μ) |
| | Inform (Action, 'Variable', time, value) |

Taula 4.2 : Llista de missatges de diàleg. La primera fila indica el canal d'entrada del missatge

4.2.2.3. Protocol de diàleg entre agents

El protocol de diàleg entre agents que s'ha usat en aquesta arquitectura està basat en el FIPA-Contract-Net Protocol (veure figura 4.2.2.3), especificat per la FIPA [FIPA, 00]. Aquest protocol no afecta a l'estructura modular de la DPAA donat que és específic del mòdul agent. En aquest protocol l'agent gestor sol·licita propostes d'altres agents per tal d'optimitzar la solució a un problema concret que cal solucionar. Aquesta sol·licitud ve precedida d'un missatge CFP (*Call For Proposals*) enviat pel mateix agent gestor, en què hi fa constar una precondició determinada. Per tant, en cas de no refusar la proposta, les propostes dels altres agents hauran de contenir el valor que obtenen per aquesta precondició. Un cop el gestor ha avaluat les propostes, envia un missatge d'acceptació a l'agent seleccionat i missatges de rebuig als altres. A partir d'aquest moment l'agent contractat adquireix el compromís de resoldre el subproblema, i un cop solucionat ha d'enviar el missatge corresponent al gestor:

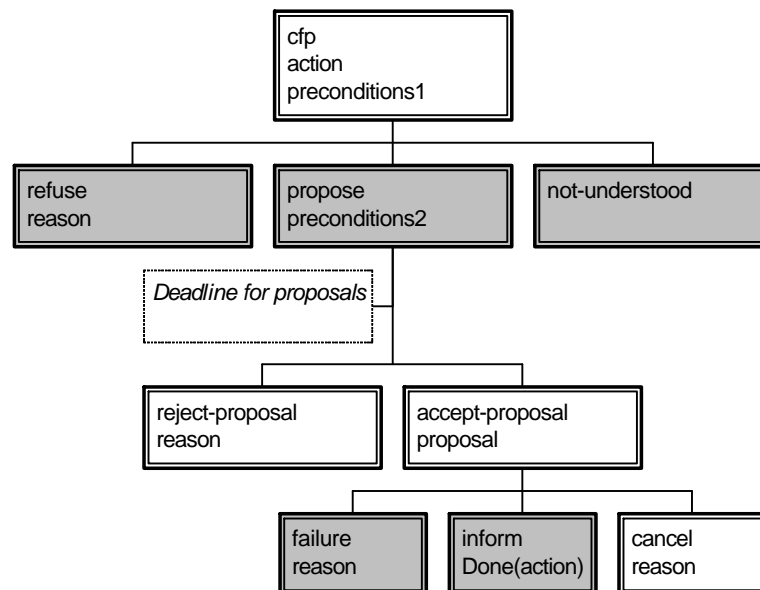


Figura 4.2.2.3: Diagrama del FIPA-Contract-Net Protocol.

En color gris s'indica què respon l'agent contractat.

D'acord amb el contingut de les taules 4.1 i 4.2, els missatges en cada bloc són els següents:

| FIPA-Contract-Net | DPAA |
|---------------------------|---|
| CFP Action Preconditions1 | Request(Action, time) & INT Request(Action, time) |
| Not-Understood | <i>No usat</i> |
| Refuse Reason | Refuse (Action, 'reason') |
| Propose Preconditions2 | Inform (Action, time, μ) |
| Reject-Proposal Reason | <i>No usat</i> |
| Accept-Proposal Proposal | Inform (Action, DO) & Confirm (Action, START) |
| Failure Reason | Inform (Action, PROBLEM, 'Description') |
| Inform Done(action) | INT Inform (Action,END) |
| Cancel Reason | INT Inform (Action, STOP) |

4.2.2.4. Estructura dels missatges

Finalment cal comentar que l'estructura dels missatges ha de seguir la de la FIPA. En concret, ha respondre a la següent estructura:

```
(tipus_missatge
  :sender mòdul_i
  :receiver mòdul_j , (j=i±1)
  :content
    (ID_ACTION, [...])
  :in-reply-to ID_ACTION
  :in-reply-with ID_MESSAGE
  :language AOP
  :ontology ..
)
```

Com es pot veure, sempre ha de constar que l'emissor i el receptor són dos mòduls veïns i que el llenguatge que s'utilitza és el proposat al paradigma del *Agent Oriented Programming* (AOP) de Shoham. Al camp **content** del missatge hi pot figurar qualsevol dels missatges que s'han tabulat anteriorment a les seccions 4.2.2.2 i 4.2.2.3 tenint en compte que tant l'acció com el missatge estan codificats amb un número identificatiu.

4.2.3. Cicles de funcionament

A partir de l'estructura de l'arquitectura s'observa que la comunicació entre els mòduls pot iniciar-se o bé des d'alt nivell quan l'agent físic es compromet a executar un

rol (figura 4.2.2), o bé des de baix nivell quan el mòdul de control detecta un event a través del canal perceptual. Seguint la nomenclatura de l'arquitectura InteRRap, en el primer cas es tracta d'una activació *top-down*, i el segon cas d'una activació *bottom-up*.

Cal comentar que per descriure el cicle de funcionament de la DPAA s'ha posat un quart mòdul corresponent a la col·lectivitat (els altres agents). Com es pot veure, aquest quart mòdul pot afegir-se a l'arquitectura DPAA de forma natural (veure figura 4.2.2): **la percepció que té la col·lectivitat sobre un agent és el grau de compromís que adquireix aquest agent en acceptar un rol.**

Els esquemes que es mostren a continuació mostren quin és el flux d'informació entre els mòduls de la DPAA. Observant aquests fluxos es pot tenir una idea de quin és el temps de cicle en qualsevol dels dos casos: els esquemes indiquen en diferents columnes quants i quins són els missatges a passar entre els mòduls.

4.2.3.1. Activació *top-down*: *procés d'introspecció*

L'activació *top-down* es produeix quan un agent físic ha d'adquirir un compromís en una comunitat d'agents. L'objectiu d'aquest cicle de funcionament és obtenir un índex que proporcioni informació sobre la capacitat que té l'agent d'executar un rol determinat, és a dir, que avaluï si la dinàmica del cos físic ho permet. Per tant, el flux d'informació ha de permetre que el mòdul agent pugui adquirir algun tipus de coneixement de baix nivell: el procés d'adquisició d'aquest coneixement l'anomenarem **procés d'introspecció**. El funcionament d'aquest cicle és el següent (figura A.2.3.1):

1. S'activa el flux d'informació quan la col·lectivitat sol·licita a l'agent físic si pot realitzar un rol determinat.
2. L'agent inicia el procés d'introspecció. Fa una demanda al supervisor per saber el grau de dificultat d'execució del rol.
3. El supervisor determina la tasca associada al rol en qüestió. Si el supervisor no disposa de prou informació per avaluar el grau de dificultat d'execució d'alguna part de la tasca, fa una demanda de paràmetres dinàmics al mòdul de control.
4. Si la demanda arriba a aquest mòdul significa que el supervisor necessita informació addicional. En aquest cas el mòdul de control consulta el conjunt de capacitats per tal d'obtenir els paràmetres dinàmics que descriuen l'execució .

5. Un cop el supervisor disposa de tota la informació relativa a l'execució de la tasca, obté un índex del grau de dificultat d'execució de la tasca i ho comunica al mòdul agent.
6. L'agent dona per finalitzat el procés d'introspecció i contesta a la col·lectivitat. La contesta pot ser afirmativa i proporcionar l'índex del supervisor com a grau de compromís. També pot ser negativa i refusar el rol sol·licitat. En aquest cas suposarem una contesta afirmativa.
7. Interval de temps indeterminat que depèn del altres agents.
8. (8 ..12) Ídem passos (1..5)
13. Suposem ara que l'agent proporciona un grau de compromís suficient.
14. La col·lectivitat està d'acord amb el compromís de l'agent i li adjudica el rol.
15. El mòdul agent fa la comanda de la tasca al mòdul supervisor. Es queda en estat d'espera d'events tant de la col·lectivitat com del mòdul supervisor.
16. El mòdul supervisor determina les consignes a seguir i fa la comanda corresponent al mòdul de control. Es queda en estat d'espera d'events tant del mòdul agent com del mòdul de control.
17. El mòdul de control inicia la nova acció de control ordenada pel mòdul supervisor i es queda en *C-mode*. Quan es produeixin events, problemes o si l'acció de control finalitza, aquest mòdul de control enviarà el missatge informatiu corresponent.

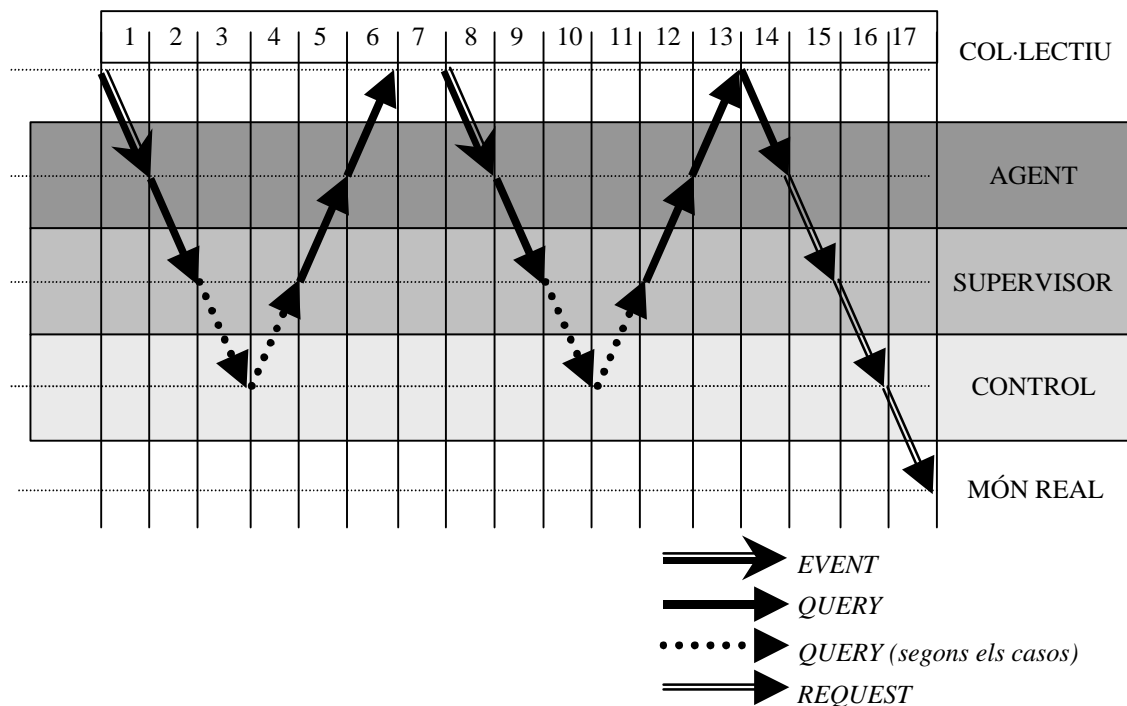


Figura 4.2.3.1: Esquema del flux d'informació en una activació top-down.

4.2.3.2. Activació *bottom-up*: procés d'actuació

L'activació *bottom-up* es produeix quan un agent físic detecta un event en el món real i necessita prendre una decisió. La nova decisió ha de tenir en compte quina és la importància de l'event i com pot afectar als demés agents de la col·lectivitat. Per tant, el flux d'informació ha de fer que el mòdul de control executi comandes ordenades des del mòdul agent: el procés d'adjudicació de les noves comandes l'anomenarem **procés d'actuació**. El funcionament d'aquest cicle és el següent (figura A.2.3.2):

1. Es produeix un event en el món real a través de detecció directa mitjançant un sensor, o bé el mateix mòdul de control detecta que es compleix alguna de les condicions pre-especificades (pas per zero d'alguna variable, fi de la llei de control, etc.) o bé detecta algun tipus de problema dinàmic (oscil·lacions mantingudes, sobrepics excessius, saturacions, etc.).
2. El mòdul de control envia un missatge d'interrupció per tal d'atreure l'atenció del mòdul supervisor. Es queda en estat d'espera en cas que l'event estigui causat per un problema dinàmic o per la finalització d'una acció. En el cas que l'event el generi l'estat d'un sensor, l'estat del controlador dependrà dels casos.
3. El supervisor intenta identificar l'event dintre de l'estructura de la tasca. Si no l'identifica o si és informació d'un problema dinàmic interrompeix l'execució de la tasca i ho notifica al mòdul agent amb un missatge d'interrupció. Es queda en estat d'espera.
4. La informació del que passa a baix nivell arribarà fins el mòdul agent si hi ha algun tipus de problema dinàmic o bé quan la tasca hagi finalitzat. Si la realització de la tasca presenta problemes el mòdul agent haurà de decidir què fer. Si la decisió representa un canvi de rol es cancel·larà l'execució del rol actual i es passarà a la negociació d'un nou rol seguin el cicle de funcionament per activació *top-down*. Si la decisió no representa cap canvi de rol, el mòdul no informa a la col·lectivitat i fa la comanda de les noves ordres.
5. El mòdul supervisor determina la tasca a fer i fa la comanda corresponent al mòdul de control. Es queda en estat d'espera d'events futurs tant del mòdul agent com del mòdul de control.

6. El mòdul de control cancel·la l'última acció de control i inicia la nova acció de control ordenada pel mòdul supervisor. Quan es tornin a produir events el mòdul de control enviarà el missatge informatiu corresponent i torna a començar el cicle.
7. ..
8. El mòdul de control detecta que està en marxa la nova acció.
9. El mòdul de control informa al supervisor que l'acció està fent-se la nova acció, és a dir, que la tasca està activa. En mòdul de control es queda en *C-mode*.
10. El mòdul supervisor informa al mòdul agent que la tasca està en marxa, és a dir, que el rol està actiu. En mòdul supervisor es queda en estat d'espera.

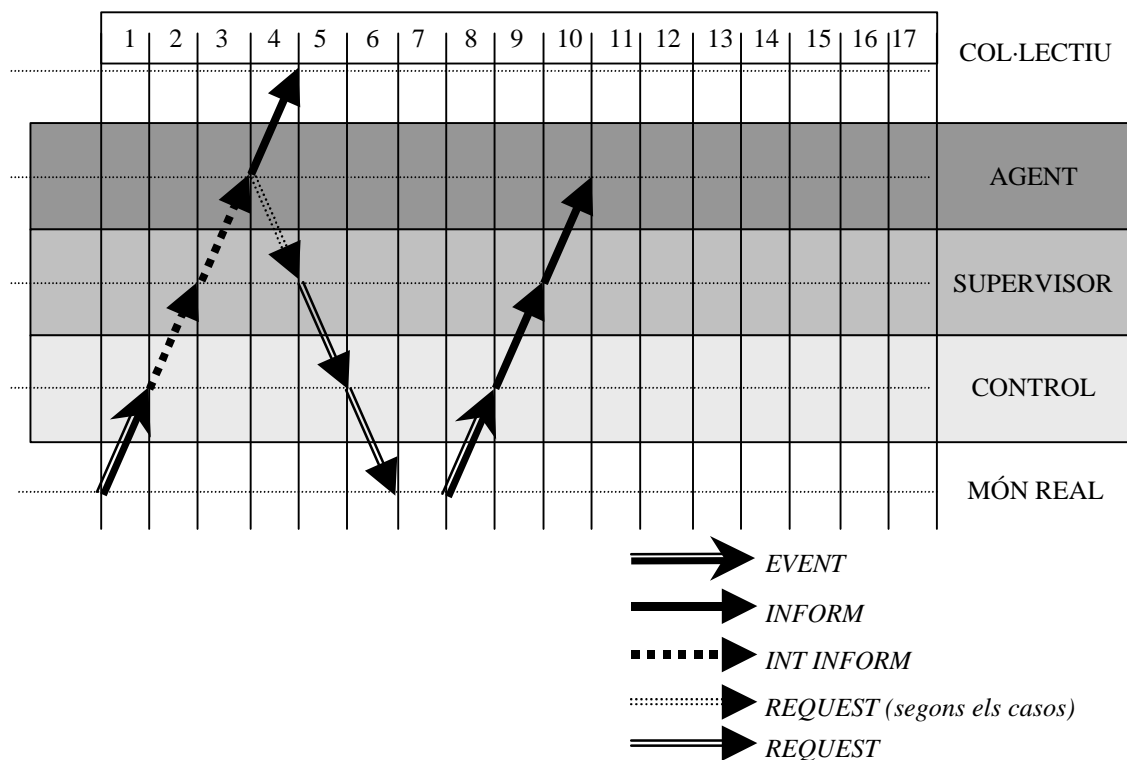


Figura A.2.3.2: Esquema del flux d'informació en una activació bottom-up.

4.2.3.3. Comentaris addicionals

Hem vist que hi ha dos possibles modes de funcionament els quals podrien estar actius de forma simultània, és a dir, podria ser que es donés el cas que els dos fluxos d'informació poguessin ser coexistents. La solució depèn de si el flux d'activació arriba a l'agent o si es queda només fins el supervisor. En el primer cas significaria que hi ha una situació que cal analitzar. A causa de la prioritat de la informació perceptual, el mòdul agent deixaria temporalment de banda qualsevol acció d'introspecció i prendria una decisió al respecte. En posterioritat, reactivaria el procés d'introspecció. En el segon

cas la simultaneïtat no presentaria problemes. En aquest sentit, cal observar que sempre es prioritza la informació perceptual per tal de fer minvar el flux d'informació en una activació *bottom-up*, és a dir, es prioritzen els problemes de baix nivell.

A la següent secció veurem que el mòdul de supervisió conté diversos supervisors que gestionen el control de les tasques. També veurem que els controladors del mòdul de control no disposen de supervisors específics. Aquest fet no suposa un problema de funcionament donat que els supervisors de les tasques són qui detecten els problemes dels controladors: a la definició de les tasques sempre cal incloure un procediment on analitzar els problemes que es detecten en els controladors a partir de la recepció dels missatges `INT Inform(Action, EVENT, 'Description')` i `INT Inform(Action, PROBLEM, 'Description')`. Aquest procediment inclourà tota la lògica necessària per la supervisió del controlador i de la tasca.

El procés de control del nivell més baix (funcionament en *C-mode*) té la prioritat més alta a no ser que arribi missatge `INT Request (Action, time)` del supervisor. L'opció de permetre això dependrà del dissenyador, però hi ha molts riscos a assumir si es permet la cancel·lació d'una acció. En aquests casos la programació del supervisor haurà de tenir en compte la possibilitat de decidir si és lícit cancel·lar una acció que està en marxa per tal d'iniciar-ne una altra.

4.3. Definició de les capacitats modulars

La forma de definir les capacitats depèn del nivell d'abstracció en què es treballi. Per obtenir una definició sense ambigüitat cal basar-la en un llenguatge descriptiu numèric. Caldrà començar per descriure-les pels nivells baixos d'abstracció (mòduls de control i supervisió) i anar augmentant el grau d'abstracció per anar passant a descripcions més simbòliques (mòdul agent). Una raó de pes per començar a definir les capacitats pel mòdul de control està en el fet que tot agent físic disposa d'uns controladors que el limiten i capaciten per executar una certa quantitat d'accions o moviments d'una forma més o menys correcte. Així, quan els paràmetres que descriuen la dinàmica i l'estàtica d'una acció o moviment indiquin que pot haver-hi problemes, el mòdul immediatament superior s'ha de fer responsable d'inhibir o no la seva execució. Així, el món lògic de l'agent podrà saber si és capaç de controlar el seu cos per tal d'assumir un rol concret en el col·lectiu d'agents fent ús del procés d'introspecció. En aquells casos en què no hi hagi suficient capacitat per portar a terme una acció, caldrà concloure que cal inhibir-la i comunicar a la resta d'agents que el rol no es pot assumir amb prou garanties.

4.3.1. Les capacitats en el mòdul de control

La forma de definir les capacitats corresponents al mòdul de control és la següent:

"Una capacitat del mòdul de control consisteix en la percepció de les qualitats d'un controlador".

Per tant, les capacitats a nivell de control integren tant la identificació del cos físic com la percepció de l'entorn a través del cos i, en última instància, integren també l'adaptació de l'agent físic a l'entorn, és a dir, l'habilitat de poder aprendre.

Per tal de formalitzar la definició de les capacitats, anomenem les capacitats del mòdul de control *capacitats atòmiques* (C_A), que són aquelles que l'acció associada pot executar-se directament amb la percepció i l'actuació del nivell més baix, sense interpretar les dades adquirides. Per tant, l'acció es portarà a terme a partir de:

- (i) La mesura de les coordenades generalitzades del propi cos $\{ p, \dot{p}, \ddot{p}, t \}$.
- (ii) La detecció dels events amb un conjunt de detectors.
- (iii) La mesura de magnituds externes $\{ q, \dot{q}, \ddot{q}, t \}$ mitjançant sensors.

Així, la definició de les capacitats atòmiques s'ha de fer observant la qualitat d'execució d'aquestes accions amb els diferents controladors implementats en el cos físic. El mòdul de control té la possibilitat d'actuar sobre el món real a través d'un conjunt de controladors $\chi = \{C_1, C_2, \dots, C_n\}$, on cada controlador possibilita l'execució d'una acció ordenada pel mòdul supervisor i queda especificat amb la següent terna:

$$C_i = \langle \mathbf{IN}_i, \mathbf{IC}_i, \mathbf{FC}_i, \mathbf{AA}_i, \mathbf{CL}_i, \mathbf{LB}_i \rangle, \forall i=1, \dots, n$$

\mathbf{IN}_i : conjunt d'entrades que necessita, tant contínues com discretes. Està fixat.

\mathbf{IC}_i : estat o condició inicial, tant temporal com espacial. És variable.

\mathbf{FC}_i : estat o condició final, tant temporal com espacial. És variable.

\mathbf{AA}_i : conjunt d'actuadors associats. Està fixat.

\mathbf{CL}_i : llei de control i conjunt de paràmetres que el defineixen. És variable.

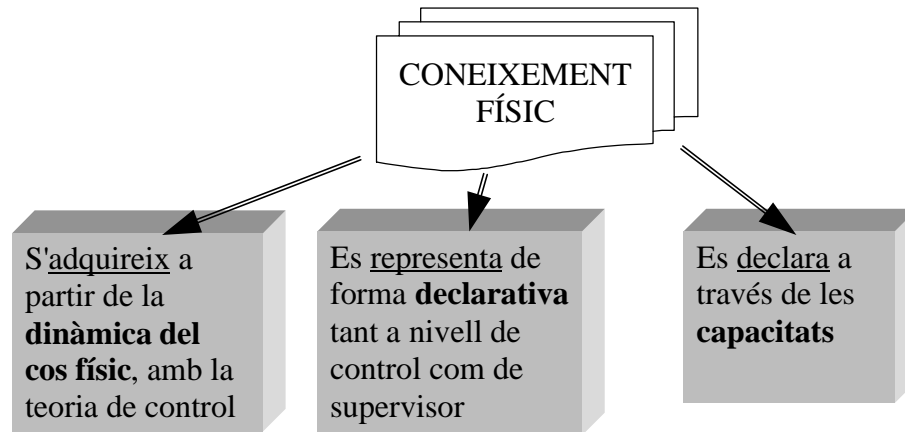
\mathbf{LB}_i : etiqueta o descripció de l'acció. Està fixat.

Per tant, si s'especifiquen unes condicions inicials i finals per a un controlador concret, el resultat de l'execució de l'acció serà diferent si varien els paràmetres de la llei de control. Així, per cada parell de dades $\langle \mathbf{IC}_i, \mathbf{FC}_i \rangle$, es pot crear un conjunt de N capacitats a partir d'un conjunt $\langle \mathbf{CL}_i^1 \dots \mathbf{CL}_i^N \rangle$ de combinacions diferents de dades de \mathbf{CL}_i . Aquest conjunt de capacitats queda definit de la següent forma:

$$\mathbf{CAN}^t_{\mathbf{A} \mathbf{J}_i}, \quad \text{on } \mathbf{J}_i = \{ \mathbf{LB}_i, \langle \mathbf{IC}_i, \mathbf{FC}_i \rangle, \langle \mathbf{J}_i^1 \dots \mathbf{J}_i^N \rangle \}$$

on \mathbf{J}_i^k consisteix en un conjunt de dades que permet descriure quines són les característiques dinàmiques i estàtiques de la resposta del sistema durant l'execució de l'acció amb el controlador C_i . Aquestes característiques corresponen als atributs de la resposta temporal i freqüencial com, per exemple, la mesura de l'error en estat estacionari, el temps de resposta, l'amplitud d'oscil·lació, l'avaluació d'una funció de cost, i altres.

En teoria de control clàssica és possible conèixer les característiques dinàmiques i estàtiques de la resposta d'un sistema lineal a partir del coneixement del model de la planta i de la llei de control. Aquest coneixement s'estableix en termes de funcions de transferència o variables de l'espai d'estat, i es basa fonamentalment en les característiques dels pols i els zeros. Per tant, la correspondència que hi ha entre els conjunts $\langle \mathbf{CL}_i^1 \dots \mathbf{CL}_i^N \rangle$ i $\langle \mathbf{J}_i^1 \dots \mathbf{J}_i^N \rangle$ s'estableix a partir de la teoria de control i es pot descriure de forma numèrica.



En molts casos la correspondència no es pot establir de forma analítica a causa de les característiques de la planta, del propi controlador o del mateix entorn. Ens referim a plantes de dinàmica d'ordre elevat o complexes (multi-variables, no-lineals,..), a controladors complexos (no-lineals, *fuzzy*, xarxes neurals,..) i a entorns molt sorollosos o amb presència elevada de perturbacions. En aquests casos la correspondència es pot establir a partir de taules, és a dir, construint taules que continguin els resultats d'assaigs previs simulats o experimentals.

4.3.1.1. Exemples

Anem a suposar que tenim un robot mòbil equipat amb un conjunt de N detectors de proximitat <detector_1, .., detector_N> que tenen dos estats lògics (1/0), i amb sensors odomètrics⁴ que permeten la mesura de les coordenades generalitzades del propi cos $\{p, \dot{p}, \ddot{p}, t\}$, on $p=(x, y, \mathbf{q})$. En aquestes condicions, un possible controlador que pot utilitzar aquest robot seria el següent:

$$C_1 = \langle \mathbf{IN}_1, \mathbf{IC}_1, \mathbf{FC}_1, \mathbf{AA}_1, \mathbf{CL}_1, \mathbf{LB}_1 \rangle$$

$\mathbf{IN}_1 : \{ \langle \text{detector}_1, \dots, \text{detector}_N \rangle, \text{encoder dret}, \text{encoder esquerra} \}$

$\mathbf{IC}_1 : \{ p_o, \dot{p}_o, \ddot{p}_o, t_o \}$, amb $p_o = (x_o, y_o, \mathbf{q}_o)$

$\mathbf{FC}_1 : \{ p_f, \dot{p}_f, \ddot{p}_f, t_f \}$, amb $p_f = (x_f, y_f, \mathbf{q}_f)$

$\mathbf{AA}_1 : \{ \text{Motor dret}, \text{Motor esquerra} \}$

$\mathbf{CL}_1 : \{ \text{"Controlador fuzzy"}, \langle \text{guany encoder dret}, \text{guany encoder esquerra} \rangle \}$

$\mathbf{LB}_1 : \text{"Perseguir un objecte detectat"}$

L'acció consisteix en utilitzar l'estat dels detectors frontals per tal d'activar amb més o menys magnitud l'actuació sobre els dos motors. D'aquesta manera el robot mòbil podria perseguir un objecte mòbil que estigués al davant. En aquest cas s'utilitza un controlador *fuzzy* i, per tant, caldria fer diversos assaigs per tal de trobar la correspondència entre el controlador \mathbf{C}_1 i el conjunt associat de capacitats. Els assaigs consistirien en variar els paràmetres de \mathbf{IC}_1 , \mathbf{FC}_1 i els guanys especificats a \mathbf{CL}_1 .

Anem a suposar ara que tenim el mateix robot mòbil. Un altre possible controlador podria ser el següent:

$$\mathbf{C}_2 = \langle \mathbf{IN}_2, \mathbf{IC}_2, \mathbf{FC}_2, \mathbf{AA}_2, \mathbf{CL}_2, \mathbf{LB}_2 \rangle$$

\mathbf{IN}_2 : {encoder dret, encoder esquerra}

\mathbf{IC}_2 : { \dot{p}_o, t_o }, amb $p_o = (x_o, y_o, \mathbf{q}_o)$

\mathbf{FC}_2 : { \dot{p}_f, t_f }, amb $p_f = (x_f, y_f, \mathbf{q}_f)$

\mathbf{AA}_2 : {Motor dret, Motor esquerra}

\mathbf{CL}_2 : {"Controlador P", $\langle K_p \rangle$ }

\mathbf{LB}_2 : "Mantenir una velocitat lineal constant "

En aquest cas es tracta d'un moviment en què el robot intenta mantenir una velocitat lineal constant a partir només de la mesura de la velocitat de les rodes. Les característiques de la resposta estan condicionades a la dinàmica dels motors. Si suposem que els motors són iguals i que tenen un comportament de primer ordre, les diferents respostes que observariem són del tipus que es mostra a la figura 4.3.1.1. Per a 5 valors diferents del paràmetre K_p la resposta temporal queda completament caracteritzada amb dos paràmetres: el temps de resposta (τ) i l'error en estat estacionari (e_{ss}). En el gràfic s'observa que a mida que el valor de K_p augmenta la resposta del sistema és més ràpida (τ disminueix) i té menys error (e_{ss} disminueix).

En aquest segon exemple la correspondència entre les cinc diferents parametritzacions $\langle \mathbf{CL}_2^1 \dots \mathbf{CL}_2^5 \rangle$ del controlador \mathbf{C}_2 i les cinc capacitats associades $\langle \mathbf{J}_2^1 \dots \mathbf{J}_2^5 \rangle$ s'ha establert a partir dels assaigs realitzats mitjançant simulació. El model que s'ha emprat pels dos motors respon a la següent funció de transferència:

$$G(s) = \frac{A}{s + a} \quad , \quad \text{on } a = 1, A = 1$$

⁴ La odometria fa referència a la mesura del propi moviment a partir del recompte de voltes que fan les rodes motrius.

La taula de capacitats $\mathbf{J}_2 = \langle \mathbf{J}_2^1 \mathbf{J}_2^2 \mathbf{J}_2^3 \mathbf{J}_2^4 \mathbf{J}_2^5 \rangle$ associada al controlador \mathbf{C}_2 és la següent:

| \mathbf{C}_2 | \mathbf{CL}_2^k | | \mathbf{J}_2^k | |
|----------------|-------------------|-------|------------------|---------|
| | k | K_p | e_{ss} (m/s) | t (s) |
| | 1 | 10 | 0.09 | 0.04 |
| | 2 | 5 | 0.17 | 0.07 |
| | 3 | 2 | 0.34 | 0.13 |
| | 4 | 1 | 0.5 | 0.2 |
| | 5 | 0.5 | 0.67 | 0.26 |

En aquest cas, la correspondència entre els paràmetres \mathbf{CL}_2^k del controlador i les capacitats associades \mathbf{J}_2^k també es pot establir de forma analítica a partir de resultats teòrics de la teoria de control. En concret tenim les següents relacions:

$$3.7 \cdot \tau \approx \frac{1}{a + A \cdot K_p} \qquad e_{ss} = \frac{1}{1 + K_p \cdot G(0)}$$

Per tant, a partir del coneixement físic del sistema (en aquest cas un model de primer ordre en forma de funció de transferència) es pot saber quines seran les capacitats \mathbf{J}_2^k per a qualsevol parametrització \mathbf{CL}_2^k del controlador. En casos com aquest no caldrà construir la taula sinó que amb un conjunt de relacions matemàtiques i llurs limitacions és possible avaluar el conjunt de capacitats.

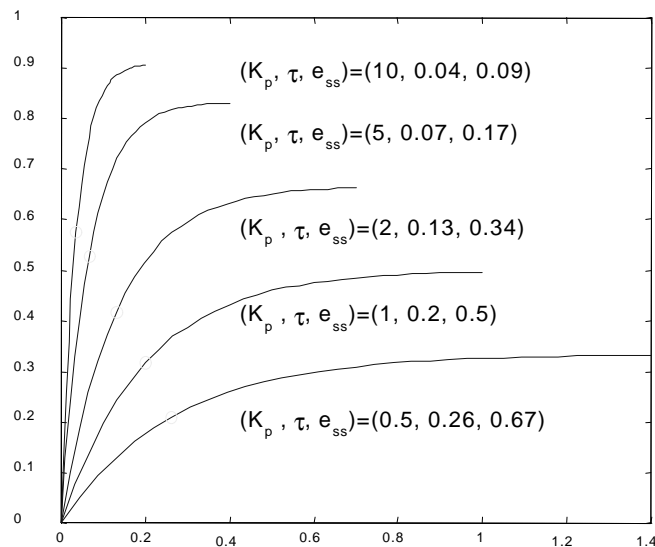


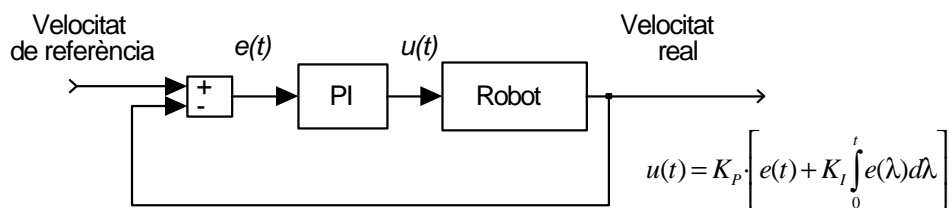
Figura 4.3.1.1: Diferents respostes temporals de la velocitat lineal d'un robot mòbil amb un control proporcional. La consigna de velocitat està normalitzada a 1.

4.3.1.2. Exemple real: comboi de dos robots (I)

Seguint amb l'exemple exposat a l'apartat 3.2.1 del tercer capítol, hem vist que quan dos vehicles estan en formació de comboi pot haver-hi problemes de col·lisió entre ells: quan el vehicle del davant decideix frenar s'observa que la dinàmica del vehicle del darrera es modifica i es produeixen oscil·lacions en la distància de separació. Per tant, el problema apareix a causa de que el guia canvia la seva velocitat sense tenir en compte la dinàmica del seguidor.

El problema del comboi s'ha implementat utilitzant dos robots mòbils al laboratori del ROGI-Team de la Universitat de Girona⁵ i durant tot aquest capítol utilitzarem aquest problema per il·lustrar com es dissenyen els agents físics associats a cada vehicle. És a dir, suposarem que tant el guia com el seguidor són autònoms, que disposen controladors propis idèntics i que han de coordinar-se entre ells per tal d'evitar possibles col·lisions.

El controlador dels robots reals és de tipus PI, és a dir, depèn de dos paràmetres: K_P i K_I .



Utilitzant aquest controlador definirem quatre accions diferents de frenar/accelerar, és a dir, que facin que els robots passin bruscament d'una velocitat inicial a una altra de final. A la figura 4.3.1.2.1 es mostren diversos resultats on s'observa la velocitat real dels robots per a dos parametritzacions del controlador intern i per als següents canvis de la velocitat de referència: $0 \rightarrow 20\text{cm/s}$, $0 \rightarrow 25\text{cm/s}$, $0 \rightarrow 30\text{cm/s}$, i $0 \rightarrow 35\text{cm/s}$. Tot i que aquests canvis corresponen a accelerar, caldria comentar que s'ha vist experimentalment que la dinàmica de la frenada és molt similar tal i com era d'esperar en un sistema d'aquestes característiques.

A partir d'aquests gràfics es pot observar que el controlador amb traç continu ($K_P=0.45$, $K_I=1.0$) és més ràpid i presenta oscil·lacions, mentre que l'altre ($K_P=0.45$, $K_I=0.3$) és més lent però sense oscil·lar. Per tant, la manera d'assolir la velocitat de referència és diferent en funció de la parametrització del controlador que s'esculli. A partir d'aquestes dades es poden obtenir diferents taules de capacitats associades als dos

controladors. Concretament tenim quatre accions (C_i , $i=1,2,3,4$) i per a cada acció dues capacitats $J_i^{k=1,2}$: en aquest exemple del comboi ens interessa obtenir només el valor del temps de resposta o desacceleració (T_D) que aproximem al valor de 3τ .

Així, les quatre accions queden definides de la manera següent:

$C_i = \langle IN_i, IC_i, FC_i, AA_i, CL_i, LB_i \rangle$

IN_i : {encoder dret, encoder esquerra}

IC_i : { \dot{p}_o }, amb $p_o = (x_o, y_o)$

FC_i : { \dot{p}_f }, amb $p_f = (x_f, y_f)$

AA_i : {Motor dret, Motor esquerra}

CL_i : {"Controlador PI", $\langle K_P, K_I \rangle$ }

LB_i : "Canviar bruscament de velocitat"

Nota: Considerarem que $i=1$ correspon al canvi $0 \leftrightarrow 20$ cm/s, $i=2$ a $0 \leftrightarrow 25$ cm/s, $i=3$ a $0 \leftrightarrow 30$ cm/s, $i=4$ a $0 \leftrightarrow 35$ cm/s.

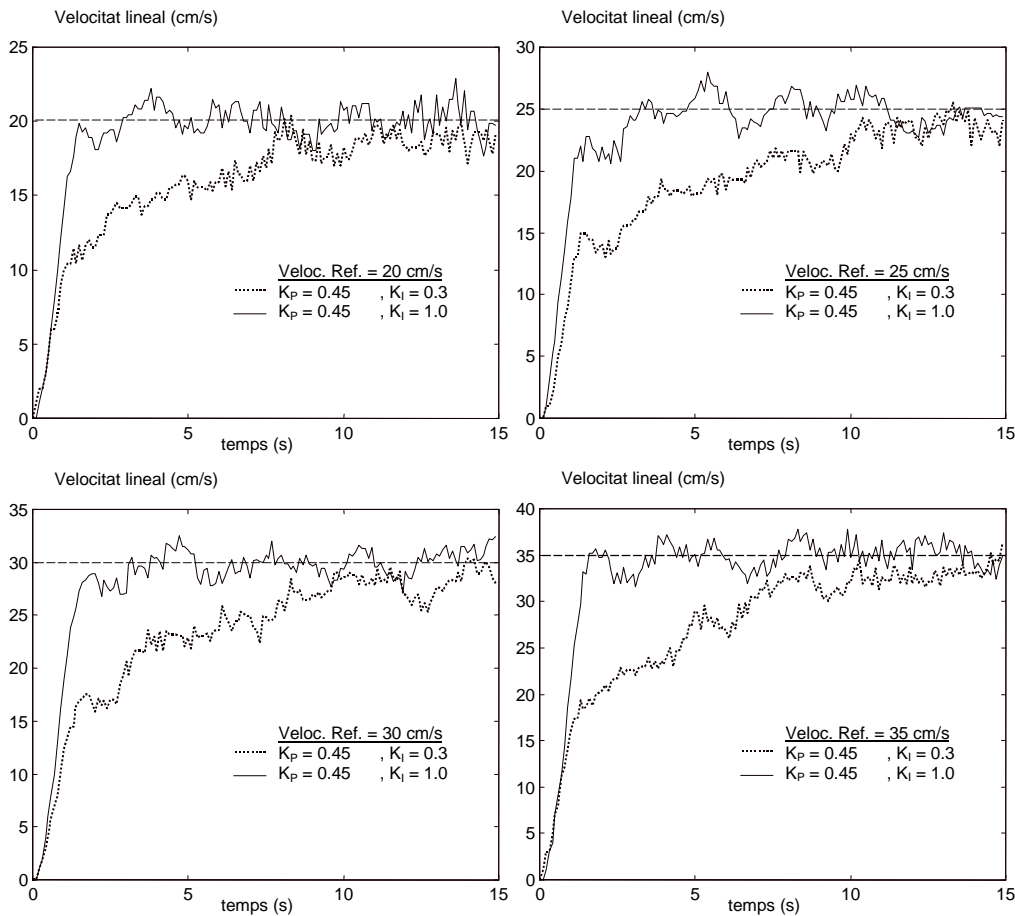


Figura 4.3.1.2.1: Respostes de la velocitat del robot per a diferents canvis de la velocitat de referència i per a dues parametritzacions del controlador PI. Període de mostreig $T = 0,06s$.

⁵ Les característiques tècniques d'aquests robots estan detallades a l'apèndix II.

A partir dels gràfics anteriors, les taules de capacitats queden de la següent manera:

| C_1 | CL_1^k | | | J_1^k |
|-------|----------|-------|-------|----------|
| | k | K_P | K_I | $T_D(s)$ |
| | 1 | 0.45 | 1 | 1.6 |
| | 2 | 0.45 | 0.3 | 9.4 |

| C_2 | CL_2^k | | | J_2^k |
|-------|----------|-------|-------|----------|
| | k | K_P | K_I | $T_D(s)$ |
| | 1 | 0.45 | 1 | 1.55 |
| | 2 | 0.45 | 0.3 | 8.5 |

| C_3 | CL_3^k | | | J_3^k |
|-------|----------|-------|-------|----------|
| | k | K_P | K_I | $T_D(s)$ |
| | 1 | 0.45 | 1 | 1.5 |
| | 2 | 0.45 | 0.3 | 7.5 |

| C_4 | CL_4^k | | | J_4^k |
|-------|----------|-------|-------|----------|
| | k | K_P | K_I | $T_D(s)$ |
| | 1 | 0.45 | 1 | 1.45 |
| | 2 | 0.45 | 0.3 | 6.0 |

Taula 4.3.1.2: Capacitats atòmiques associades a les accions $C_{i=1,2,3,4}$.

Per fer cada acció s'han proposat dues parametritzacions diferents del controlador:
 CL_i^1 : {"Controlador PI", $\langle K_p = 0.45, K_I = 1 \rangle$ }, CL_i^2 : {"Controlador PI", $\langle K_p = 0.45, K_I = 0.3 \rangle$ }.

Així, les capacitats d'aquests robots per dur a terme l'acció d'accelerar amb els controladors $C_{i=1,2,3,4}$ s'escriuen de la següent manera:

CAN^t_A {" Canviar bruscament de velocitat" , $\langle (0, 0), (0, 35) \rangle$, $\langle 1.45 \rangle$ }

CAN^t_A {" Canviar bruscament de velocitat" , $\langle (0, 0), (0, 30) \rangle$, $\langle 1.5 \rangle$ }

CAN^t_A {" Canviar bruscament de velocitat" , $\langle (0, 0), (0, 25) \rangle$, $\langle 1.55 \rangle$ }

CAN^t_A {" Canviar bruscament de velocitat" , $\langle (0, 0), (0, 20) \rangle$, $\langle 1.6 \rangle$ }

CAN^t_A {" Canviar bruscament de velocitat" , $\langle (0, 0), (0, 35) \rangle$, $\langle 6.0 \rangle$ }

CAN^t_A {" Canviar bruscament de velocitat" , $\langle (0, 0), (0, 30) \rangle$, $\langle 7.5 \rangle$ }

De la mateixa manera, algunes capacitats per l'acció de la frenada són les següents:

CAN^t_A {" Canviar bruscament de velocitat" , $\langle (0, 0), (20,0) \rangle$, $\langle 1.6 \rangle$ }

CAN^t_A {" Canviar bruscament de velocitat" , $\langle (0, 0), (35,0) \rangle$, $\langle 6.0 \rangle$ }

CAN^t_A {" Canviar bruscament de velocitat" , $\langle (0, 0), (30,0) \rangle$, $\langle 7.5 \rangle$ }

CAN^t_A {" Canviar bruscament de velocitat" , $\langle (0, 0), (25,0) \rangle$, $\langle 8.5 \rangle$ }

CAN^t_A {" Canviar bruscament de velocitat" , $\langle (0, 0), (20,0) \rangle$, $\langle 9.4 \rangle$ }

4.3.2. Capacitats en el mòdul de supervisió

La forma de definir les capacitats corresponents al mòdul de supervisió és la següent:

"Les capacitats del mòdul de supervisió s'anomenen *capacitats bàsiques* (C_B), i són aquelles que poden obtenir-se combinant les atòmiques".

El mètode de combinació pot consistir en utilitzar controladors com a estats en una màquina d'estats finits, i en utilitzar les condicions que obliguen a abandonar un controlador i escollir-ne un altre com a transicions entre estats. Si el comportament dels controladors és conegut i es coneix la seqüència de controladors per passar de l'estat inicial al final, serà possible avaluar algunes característiques de la resposta del sistema. Altres mètodes de combinació a utilitzar poden basar-se en sistemes de control híbrid si el model del sistema és canviant, en xarxes de Petri si el procés presenta concurrència o en mètodes més heurístics com el basat en casos.

El mòdul de supervisió té la feina d'executar les tasques encomanades pel mòdul agent, i cadascuna d'aquestes tasques consisteix en anar activant diferents controladors del mòdul de control i ordenant-los diferents consignes. La seqüència d'activació d'aquests controladors ve determinada per les característiques de la tasca que ha ordenat executar el mòdul agent. Aquesta actuació sobre el mòdul de control és a través d'un conjunt de supervisors $\Sigma = \{S_1, S_2, \dots, S_m\}$, on cada supervisor S_i s'encarrega de controlar el funcionament d'una tasca específica. Un supervisor queda especificat amb la següent terna:

$$S_i = \langle \mathbf{IN}_i, \mathbf{IC}_i, \mathbf{FC}_i, \mathbf{AA}_i, \mathbf{CL}_i, \mathbf{LB}_i \rangle, \forall i=1, \dots, m$$

\mathbf{IN}_i : conjunt de senyals que defineixen les transicions. Està fixat.

\mathbf{IC}_i : estat inicial i transicions que el desactiven. Paràmetres de l'estat variable.

\mathbf{FC}_i : estat final i transicions que l'activen. Paràmetres de l'estat variable.

\mathbf{AA}_i : conjunt de controladors que s'utilitzen en els estats. Està fixat.

\mathbf{CL}_i : màquina d'estats associada a l'acció. Paràmetres dels estats variables.

\mathbf{LB}_i : etiqueta o descripció de la tasca. Està fixada.

Per tant, si s'especificuen els paràmetres dels estats inicial i final per a una tasca concreta, el resultat de l'execució de la tasca serà diferent si varien les condicions de

funcionament dels controladors. Així, per a cada parell de dades $\langle \mathbf{IC}_i, \mathbf{FC}_i \rangle$, es pot crear un conjunt de M capacitats a partir d'un conjunt $\langle \mathbf{CL}_i^1 \dots \mathbf{CL}_i^M \rangle$ de parametritzacions diferents de les dades de \mathbf{CL}_i . Aquest conjunt de capacitats queda definit de la següent forma:

$$\mathbf{CAN}^t_{\mathbf{B}j_i}, \quad \text{on } j_i = \{\mathbf{LB}_i, \langle \mathbf{IC}_i, \mathbf{FC}_i \rangle, \langle \mathbf{J}_i^1 \dots \mathbf{J}_i^M \rangle\}$$

on cada \mathbf{J}_i^k consisteix en un conjunt de dades que permet descriure quines són les característiques de la resposta del sistema al final de l'execució de la tasca associada. Aquestes característiques poden ser, per exemple, un índex de la dificultat o de la precisió en assolir l'estat final, o el temps de durada de la tasca.

Quant al conjunt de controladors que defineixen les dades \mathbf{AA}_i , és possible que en alguns casos els estats siguin processos de càlcul necessaris per efectuar feines de planificació *on-line*, tal i com es mostrarà en un exemple.

Finalment, cal fer notar que el mòdul supervisor utilitza un llenguatge menys numèric que el mòdul de control i això fa obligatori l'ús d'etiquetes. Com es veurà en un exemple a la secció següent, això provoca que pugui haver-hi accions bàsiques que siguin coincidents en la combinació d'accions atòmiques, però no en l'etiquetat que utilitzen.

4.3.2.1. Exemples

Exemple 1: "Xutar la pilota endavant"

En aquest exemple agafarem el mateix robot dels exemples anteriors. Suposarem que el robot té la tasca de colpejar un objecte mòbil (pilota) en una certa direcció⁶ i que disposa de tres controladors: dos definits anteriorment (\mathbf{LB}_1 : "Perseguir un objecte detectat", \mathbf{LB}_2 : "Mantenir una velocitat lineal constant ") i d'un tercer amb l'etiqueta \mathbf{LB}_3 : "Anar a buscar un objecte". El disseny del supervisor d'aquesta tasca es basa en una màquina d'estats amb els tres estats corresponents als tres controladors, un estat inicial i un de final, i l'estat "Resoldre problema" on s'analitzen els problemes que apareguin durant el funcionament dels diferents controladors i es decideix què fer amb l'execució de la tasca. Les transicions (T_1, T_2, T_3) cap a aquest estat es produeix amb la recepció del missatge INT Inform(Action, PROBLEM, 'Description') mentre que les altres transicions són l'estat ON/OFF d'un detector de la pilota (missatge INT Inform(Action, EVENT,

'Description')) i el valor de la variable $\Delta x = x_{\text{robot}} - x_{\text{pilota}}$ (missatge Inform (Action, "Variable", time, value)). La màquina d'estats corresponent al supervisor d'aquesta tasca es mostra a la següent figura:

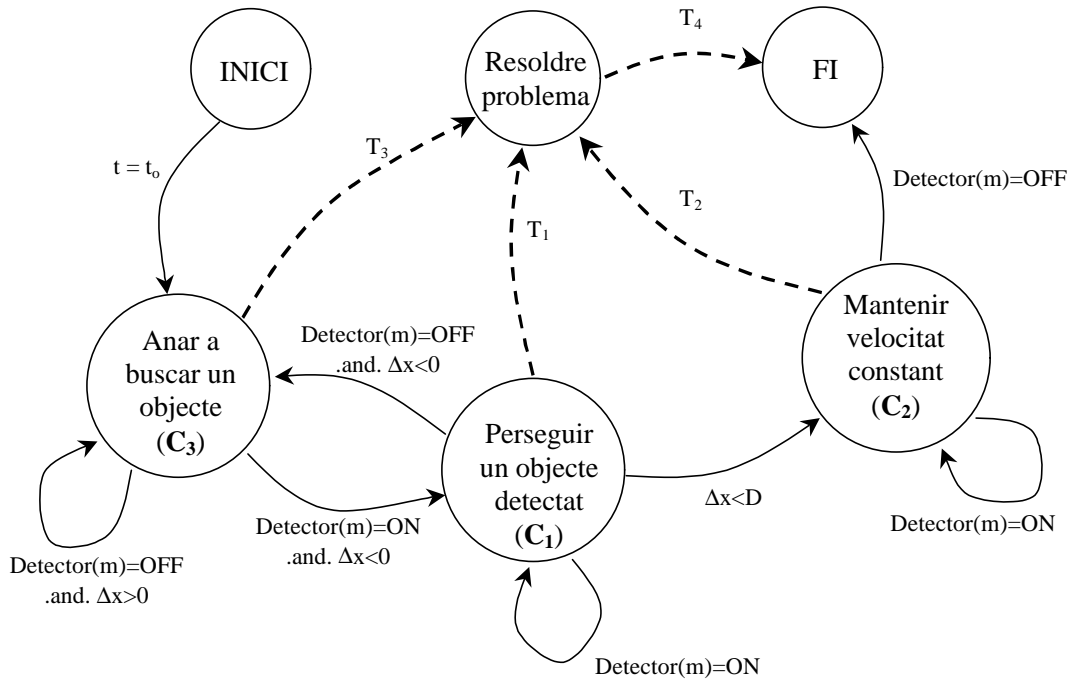


Figura 4.3.2.1.1: Màquina d'estats corresponent al supervisor de la tasca "Xutar la pilota endavant"

Per tant, aquest supervisor S_1 queda especificat de la següent manera:

- $IN_1 : \{ \text{Detector}(m), \Delta x = x_{\text{robot}} - x_{\text{pilota}} \}$
- $IC_1 : \{ \langle \text{INICI}: (x, y, \theta)_{\text{punt inicial}}, t = t_0 \rangle \}$
- $FC_1 : \{ \langle \text{FI}: (x, y, \theta)_{\text{punt final}}, \langle \text{Detector}(m)=\text{OFF}, T_4 \rangle \}$
- $AA_1 : \langle C_1, C_2, C_3 \rangle$
- $CL_1 : \text{Màquina d'estats de la figura 4.3.2.1.1.}$
- $LB_1 : \text{"Xutar la pilota endavant"}.}$

Quant a les capacitats bàsiques que caracteritzen la resposta del sistema al final de l'execució d'aquesta tasca donarem com a exemple el càlcul del temps de durada de la tasca i de la possibilitat d'incloure diversos índexs.

Quant al càlcul del temps de durada, resulta evident que el temps de durada total de la tasca serà la suma dels temps de resposta de cada estat. En general, l'ús d'una mesura física com a capacitat bàsica fa necessari que les capacitats atòmiques de tots els

⁶ Aquest exemple s'estudiarà en detall a la tercera part d'aquesta tesi.

controladors que intervenen en el supervisor de la tasca incloguin aquesta magnitud en el conjunt de les seves capacitats. En aquest exemple caldrà almenys que $\mathbf{J}_i^{k=1,2,3} = \langle \mathbf{t} \rangle$.

Finalment observar que és possible avaluar la dificultat d'executar una tasca utilitzant les funcions de cost. Ja s'ha comentat en la definició de les capacitats atòmiques que l'avaluació d'una funció de cost també es podria utilitzar com a paràmetre que caracteritza de la resposta d'un sistema. Una funció de cost típica és la que s'obté a partir de la integral del valor absolut de la derivada de la variable de control $u(t)$, l'anomenat esforç de control:

$$IADU = \frac{\int_{t_1}^{t_2} \left\| \frac{du(t)}{dt} \right\| dt}{u_{\max} - u_{\min}}$$

Tenint en compte el cas anterior caldrà que $\mathbf{J}_i^{k=1,2,3} = \langle \mathbf{t}, IADU \rangle$. Així, l'esforç de control de tota la tasca es podrà calcular realitzant de la suma dels esforços de control de cada estat.

Exemple 2: Coincidència amb la combinació d'accions atòmiques

Per concretar més aquest fet anem a utilitzar un conjunt de cinc accions atòmiques per definir un altre conjunt de capacitats bàsiques. A partir de la mesura de les coordenades del propi cos i de les magnituds externes al cos, podríem construir un conjunt $\langle \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \mathbf{C}_4, \mathbf{C}_5 \rangle$ cinc controladors per poder executar les següents accions atòmiques:

- LB₁**: "Anar a una posició"
- LB₂**: "Mantenir una velocitat"
- LB₃**: "Perseguir un objecte"
- LB₄**: "Evitar el xoc amb un objecte"
- LB₅**: "Colpejar un objecte"

Per a les accions bàsiques podríem definir el següent conjunt de supervisors $\langle \mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, \mathbf{S}_4, \mathbf{S}_5 \rangle$ definits parcialment de la següent forma:

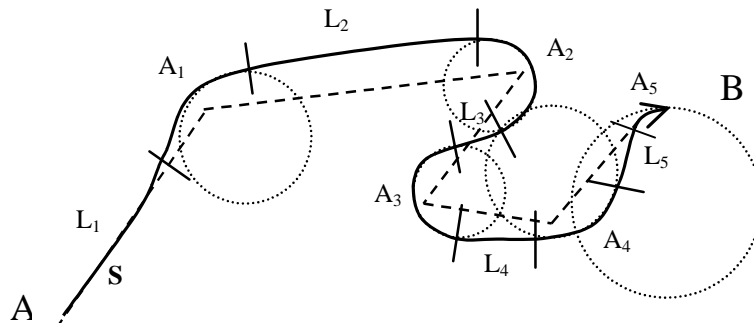
- LB₁**: "Anar de A a B seguint una trajectòria"; **AA₁**: $\langle \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_4 \rangle$
- LB₂**: "Xutar la pilota"; **AA₂**: $\langle \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_5 \rangle$
- LB₃**: "Portar la pilota"; **AA₃**: $\langle \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3 \rangle$
- LB₄**: "Perseguir un contrari"; **AA₄**: $\langle \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3 \rangle$
- LB₅**: "Moure's seguint la paret"; **AA₅**: $\langle \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \mathbf{C}_4 \rangle$

Respecte a les accions \mathbf{LB}_3 i \mathbf{LB}_4 es pot veure que són coincidents en el conjunt de controladors que utilitzen (dades \mathbf{AA}_i) però no en l'etiquetat que utilitzen. Concretament, a \mathbf{S}_3 l'etiqueta de l'objecte del controlador \mathbf{C}_3 correspon a 'pilota' mentre que \mathbf{S}_4 és 'contrari'.

Exemple 3: "Anar a una posició seguint una trajectòria lliure de col·lisió"

L'estructura de control d'aquesta tasca pot obeir a diverses estratègies en funció del tipus de trajectòria que es vulgui realitzar. Una possibilitat és executar una trajectòria amb restriccions temporals, una altra és executar-la amb restriccions espacials, i una altra amb tots dos tipus de restriccions. Tot i que hi ha molts mètodes de càlcul de trajectòries lliures de col·lisió, aquí es mostrarà com es pot construir una taula de capacitats bàsiques en dues possibilitats: l'una consisteix en el mètode de fragmentació de la trajectòria en trams rectes i arcs (aquest exemple), i l'altra en el mètode basat en la idea dels camps de potencial artificial (el següent exemple).

La següent figura mostra un exemple d'una trajectòria fragmentada en arcs i segments lineals:



La trajectòria S està formada per cinc trams lineals (L_i) i per cinc arcs (A_i), i la tasca que permet realitzar-la consisteix en una màquina d'estats que activa dos controladors diferents. El primer s'encarrega de mantenir una velocitat lineal constant sense gir ($\mathbf{LB}_1 = \text{"Anar en línia recta"}$) i el segon s'encarrega de mantenir la mateixa velocitat lineal anterior però amb un velocitat angular constant ($\mathbf{LB}_2 = \text{"Traçar un arc"}$). Una capacitat bàsica C_B relacionada amb l'execució d'aquesta trajectòria ha de tenir en compte les capacitats atòmiques \mathbf{J}_1^k i \mathbf{J}_2^k per a realitzar els diferents trams lineals L_i i els arcs A_i . Suposem que les capacitats atòmiques són el valor mig de l'acceleració angular en tot el tram: $\mathbf{J}_i^k = \alpha_m$. Donat que hi ha una relació directe entre l'acceleració

angular i la curvatura de la trajectòria ($1/\rho$), és possible avaluar la dificultat d'execució de tota la trajectòria a partir del càlcul de la curvatura mitjana, és a dir:

$$C_B = \frac{\sum_{i=1}^5 |L_i| J_1^k + \sum_{i=1}^5 |A_i| J_2^k}{|S|}$$

Exemple 4: "Anar a una posició seguint una trajectòria lliure de col·lisió"

En el mòdul supervisor l'avaluació de la qualitat d'execució de les tasques és més complexa ja que el llenguatge descriptiu no és tant numèric com en mòdul de control. De tota manera, amb mètodes heurístics es pot avaluar la precisió i/o la dificultat de les execucions de les accions. En aquest exemple es proposa avaluar l'execució de la tasca "Anar a una posició seguint una trajectòria lliure de col·lisió" calculant un índex del grau de dificultat en funció de les oscil·lacions de la trajectòria calculada a través de la tècnica dels camps de potencial artificial [Oller, 00].

La idea dels camps de potencial consisteix en assignar càrregues positives als obstacles i càrregues negatives als objectius. D'aquesta manera, si al mòbil se li assigna una càrrega positiva es pot calcular quina és la trajectòria lliure de col·lisió del mòbil en presència d'obstacles calculant la trajectòria de la càrrega positiva en presència de les altres. Per tant, la resolució del problema passa per construir el camp de potencial electrostàtic associat a la distribució dels obstacles de l'entorn i calcular⁷ el gradient en direcció a l'objectiu, tal com s'il·lustra a la següent figura:

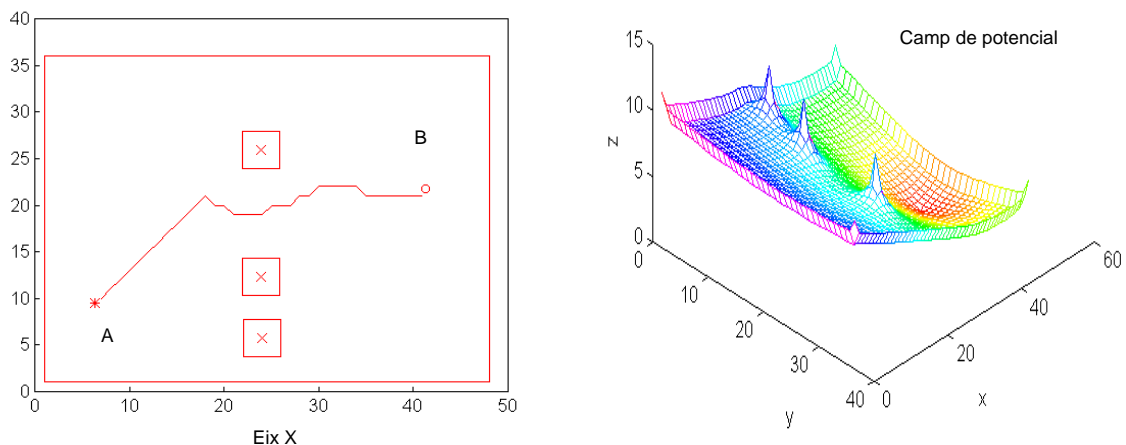


Figura 4.3.2.1.2: Exemple on es mostra quina és la trajectòria (esquerra) que ha de seguir un robot per anar de 'A' a 'B' calculada a partir del camp de potencial artificial (dreta)

⁷ Tot i que aquest mètode presenta problemes de mínims locals en el potencial, hi ha molts treballs que en proposen solucions.

En aquest exemple es presenta un mètode basat en la idea original de O.Kathib [Kathib, 85] i en les modificacions proposades per N.Clavel [Clavel, 93]. En concret, la totalitat del camp de joc s'associa a una matriu de forma que cadascun dels seus elements es correspon a una petita quadrícula real (cel·la). S'utilitza un sistema cartesià de coordenades per definir les posicions dels centres de les cel·les les quals contenen el valor del potencial associat. Una vegada la matriu està calculada, l'algorisme cerca sistemàticament quina de les cel·les veïnes conté un valor de potencial inferior al de la cel·la ocupada pel robot: la cel·la seleccionada també dependrà de la direcció relativa en què es trobi l'objectiu. Així, l'algorisme segueix un mètode de busca orientat a trobar la cel·la objectiu la qual tindrà associat un valor de potencial nul. Amb aquest mètode el camí que es troba presenta oscil·lacions quan hi ha presència d'obstacles en la direcció cap a l'objectiu.

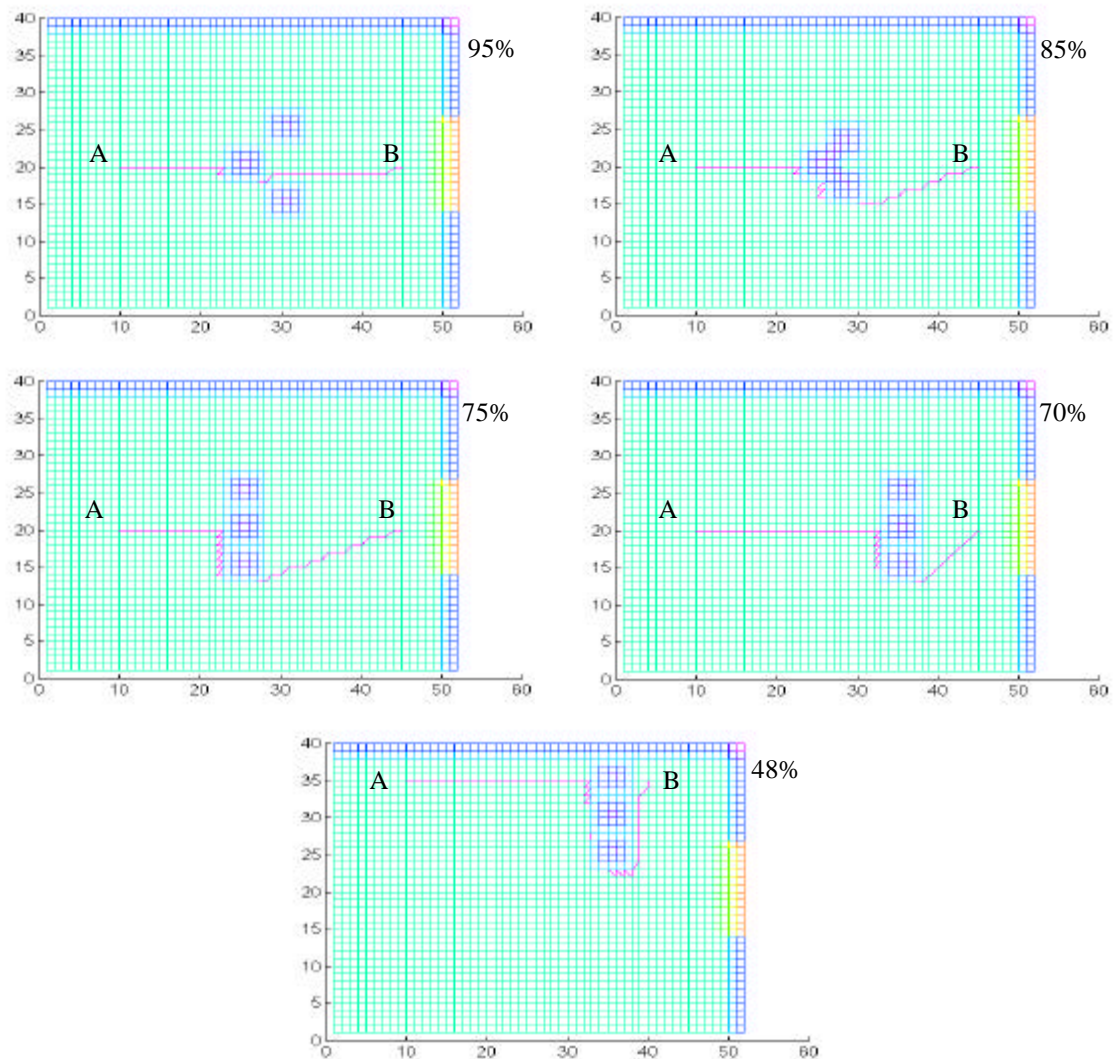


Figura 4.3.2.1.3: Diversos exemples de càlcul de l'índex m_{ami_lliure} per a un robot que va del punt 'A' al punt 'B'. La línia contínua indica la trajectòria que s'obté a partir del camp de potencial.

Amb l'objectiu de poder calcular la capacitat que té el mòbil per a executar la trajectòria lliure de col·lisió, l'algorisme comptabilitza el nombre de cel·les que formen el camí (P) i el nombre de cel·les que s'han rebutjat a causa del valor del potencial (L). Tenint en compte el valor d'aquests dos paràmetres es pot calcular un coeficient m que dóna idea de la dificultat del camí a causa de la presència d'obstacles. Aquest coeficient es calcula a partir de la següent expressió⁸:

$$m_{\text{camí_lliure}} = 1 - 2 * L / P$$

A la figura 4.3.2.1.3 es mostren diversos exemples de càlcul d'aquest índex per a un robot mòbil que ha d'anar d'un punt inicial 'A' a un de final 'B' evitant els xocs amb tres obstacles i amb unes parets que delimiten l'entorn. Les mides dels obstacles estan dimensionades de manera que si la trajectòria passa entremig de dos d'ells significa que el robot mòbil té prou espai per passar.

L'important d'aquest mètode és que permet calcular un índex sobre la dificultat del camí i, per tant, permet avaluar la capacitat que té el robot d'executar la tasca d'anar d'un punt 'A' a un punt 'B'. En aquest exemple, la capacitat bàsica és un índex que es calcula a través d'un algorisme propi del procés de supervisió i que no depèn dels valors de capacitats atòmiques. Aquest supervisor S_2 queda especificat de la següent manera:

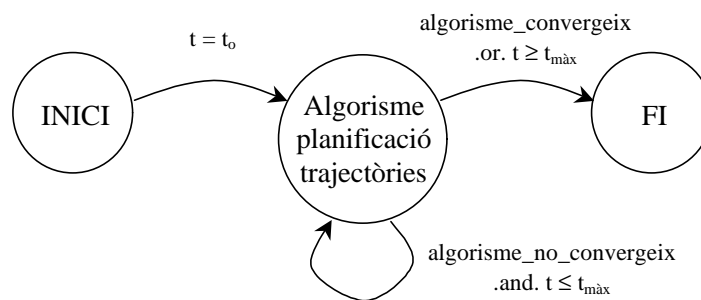
IN₂ : { t }

IC₂ : { <INICI: (x,y)_{A, B, OBSTACLES}>, t = t₀ }

FC₂ : { <FI: >, <t ≥ t_{màx} .or. algorisme_convergeix > }

AA₂ : <"Algorisme de planificació de trajectòries">

CL₂ : Màquina d'estats de la figura:



LB₂ : "Anar a una posició seguint una trajectòria lliure de col·lisió".

⁸ El factor 2 fa referència al fet que l'algorisme no accepta camins en què hi hagi un 50% de cel·les rebutjades: en aquests casos l'algorisme s'atura.

4.3.2.2. Exemple real: comboi de dos robots (II)

A l'apartat 4.3.1.2 hem vist quines són les capacitats individuals que tenen tant el robot guia com el seguidor. De fet són les mateixes perquè els assaigs s'han fet amb robots idèntics. Així, per exemple, hem vist que tant l'un com l'altre poden dur a terme l'acció de frenar de 20cm/s a 0cm/s en almenys 1.6 segons.

Aquest temps de frenada s'ha calculat definint-lo com a tres vegades el valor de la constant de temps del sistema τ . Donat que proposem una solució on els dos vehicles/agents han de acordar el temps de frenada, utilitzarem aquest valor per obtenir un índex de la certesa de l'acció de la frenada. D'aquesta manera, els agents podran utilitzar-lo com a coeficient per poder acceptar o descartar un temps de frenada durant la negociació. El coeficient de certesa el definim de la següent manera [de la Rosa, 00]:

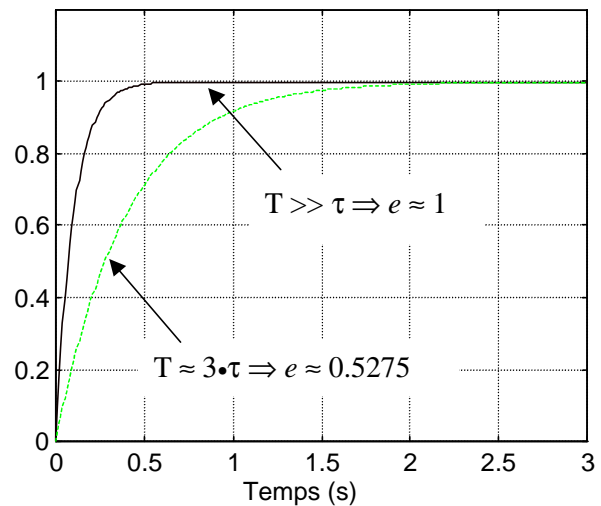
$$e = 1 - \frac{V_{FINAL} - V_T}{V_{FINAL} - V_{INICIAL}}$$

, on:

V_{FINAL} = velocitat final

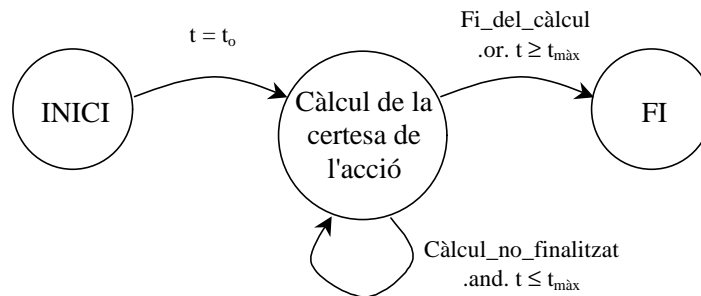
$V_{INICIAL}$ = velocitat inicial

V_T = velocitat a l'instant de temps T



A la figura es mostra la resposta temporal teòrica dels dos vehicles per a un canvi bruscat de velocitat de 0cm/s a 1cm/s. Si el valor de T s'escull de forma que $T \gg \tau$ el valor de certesa e és proper a 1 i l'acció s'executarà exactament: direm que la certesa de dur a terme l'acció és molt alta. En canvi, si el valor de T és menor, e tindrà valors inferiors, és a dir, nivells de certesa inferiors.

Per tant, en aquest exemple, la capacitat bàsica és un índex que es calcula a través d'una expressió matemàtica durant el procés de supervisió i que depèn del valor de les capacitats atòmiques. El corresponent supervisor S_3 queda especificat de la següent manera:

$\mathbf{IN}_3 : \{ t \}$
 $\mathbf{IC}_3 : \{ \langle \text{INICI} : (\dot{p}_o, T_D) \rangle, t = t_0 \}$
 $\mathbf{FC}_3 : \{ \langle \text{FI} : \rangle, \langle t \geq t_{\text{màx}} \text{ .or. } \text{Fi_del_càlcul} \rangle \}$
 $\mathbf{AA}_3 : \langle \text{"Càlcul de la certesa de l'acció"} \rangle$
 \mathbf{CL}_3 : Màquina d'estats de la figura:

 \mathbf{LB}_3 : "Aturar-se".

Amb tot, el conjunt de capacitats bàsiques $\mathbf{J}_3 = \langle e \rangle$ associades a aquest supervisor queda així:

| Velocitat inicial cm/s | Velocitat final cm/s | Temps desitjat T_D (s) | Índex de certesa e | K_P | K_I |
|------------------------|----------------------|--------------------------|----------------------|-------|-------|
| 20 | 0 | 1.6 | 0.96 | 0.45 | 1.0 |
| 20 | 0 | 9.4 | 0.91 | 0.45 | 0.3 |
| 25 | 0 | 1.55 | 0.93 | 0.45 | 1.0 |
| 25 | 0 | 8.5 | 0.93 | 0.45 | 0.3 |
| 30 | 0 | 1.5 | 0.92 | 0.45 | 1.0 |
| 30 | 0 | 7.5 | 0.96 | 0.45 | 0.3 |
| 35 | 0 | 1.45 | 0.91 | 0.45 | 1.0 |
| 35 | 0 | 6.0 | 0.99 | 0.45 | 0.3 |

Taula 4.3.2: Taula de capacitats bàsiques dels vehicles. La tasca associada en tots els casos correspon a la d'aturar el mòbil.

4.3.3. Capacitats en el mòdul agent

La forma de definir les capacitats corresponents al mòdul agent és la següent:

"Les capacitats al mòdul agent s'anomenen *capacitats simbòliques* (C_S) i es defineixen a partir de les capacitats bàsiques."

El mètode de definició és completament escalable al que s'utilitza en el supervisor, és a dir, les capacitats en el nivell d'abstracció del mòdul agent es construeixen combinant diferents capacitats bàsiques i en funció de l'aplicació. El mètode de combinació potser una màquina d'estats finits en què els estats són supervisors de

tasques i les transicions són les condicions que afavoreixen o no el manteniment de l'execució d'una tasca concreta.

El mòdul agent té la feina d'executar els rols adquirits amb la col·lectivitat i cadascun d'aquests rols consisteix en anar activant diferents tasques del mòdul de supervisió i imposant-los diferents restriccions. La seqüència d'activació d'aquestes tasques ve determinada per les característiques del rol que ha adquirit l'agent amb la col·lectivitat. Aquesta actuació sobre el mòdul de supervisió és a través d'un conjunt de conductors $\Delta = \{D_1, D_2, \dots, D_p\}$, on cada conductor D_i s'encarrega de gestionar el funcionament d'un rol específic. Un conductor queda especificat amb la següent terna:

$$D_i = \langle IN_i, IC_i, FC_i, AA_i, CL_i, LB_i \rangle, \forall i=1, \dots, p$$

- IN_i : conjunt de variables que defineixen les transicions. Està fixat.
- IC_i : estat inicial i transicions que el desactiven. Paràmetres de l'estat variable.
- FC_i : estat final i transicions que l'activen. Paràmetres de l'estat variable.
- AA_i : conjunt de supervisors que s'activen. Està fixat.
- CL_i : màquina d'estats associada a l'acció. Paràmetres dels estats variables.
- LB_i : etiqueta o descripció del rol. Està fixada.

Per tant, si s'especifiquen els paràmetres dels estats inicial i final per a un rol concret, el resultat de l'execució del rol serà diferent si varien les condicions de funcionament dels supervisors. Així, per a cada parell de dades $\langle IC_i, FC_i \rangle$, es pot crear un conjunt de P capacitats simbòliques a partir d'un conjunt $\langle CL_i^1 \dots CL_i^P \rangle$ de parametritzacions diferents de les dades de CL_i . Aquest conjunt de capacitats queda definit de la següent forma:

$$CAN^t_{Sj_i}, \quad \text{on } j_i = \{LB_i, \langle IC_i, FC_i \rangle, \langle J_i^1 \dots J_i^P \rangle\}$$

on cada J_i^k consisteix en una quantitat numèrica que indica quina és la viabilitat d'executar el rol associat. Aquesta viabilitat és el paràmetre que utilitza l'agent físic en els processos de negociació amb els altres agents, és a dir, el paràmetre μ del missatge Confirm (Action, time, μ).

Les característiques dels supervisors que defineixen les dades AA_i poden ser molt variades: en alguns casos les tasques poden consistir en executar accions comunicatives amb altres agents per tal de coordinar o millorar l'execució d'altres tasques del rol. Finalment també cal observar que el mòdul agent utilitza un llenguatge més simbòlic.

Per tant, les variables que s'utilitzen per definir les transicions han de ser de caràcter qualitatiu o *fuzzy*.

4.3.3.1. Exemple

Com a exemple escollirem el sistema multi-robot que s'estudiarà en detall al capítol 5 d'aquesta tesi. Aquest sistema consisteix en el futbol robòtic, és a dir, els robots han de ser capaços de moure's en un entorn rectangular tancat, evitar el xoc amb altres robots, comportar-se de forma cooperant amb els robots del mateix equip, etc. Una funcionalitat molt important en aquest sistema és la d'interaccionar correctament amb la pilota donat que és un objectiu mòbil i de difícil accés. Anem a suposar que disposem d'un conjunt de supervisors, alguns dels quals ja han estat definits en exemples anteriors:

| | |
|--|---|
| LB₁ : "Anar cap a la pilota" | LB₈ : "Portar la pilota" |
| LB₂ : "Anar cap la porteria local" | LB₉ : "Evitar xocs" |
| LB₃ : "Anar cap la porteria contrària" | LB₁₀ : "Evitar el xoc amb la pilota" |
| LB₄ : "Perseguir un contrari" | LB₁₁ : "Desmarcar-se d'un contrari" |
| LB₅ : "Moure's seguint la paret" | LB₁₂ : "Buscar direcció sense contrari" |
| LB₆ : "Xutar la pilota cap a porteria" | LB₁₃ : "Buscar direcció àrea contrària" |
| LB₇ : "Xutar la pilota cap a un punt" | LB₁₄ : "Buscar direcció cap a company" |

Alguns dels rols que cal definir en aquest exemple han d'incloure moviments coordinats amb altres agents que s'han de negociar mitjançant accions comunicatives. En aquests casos l'eficiència amb què l'agent es compromet a executar el seu rol s'obté a partir de les capacitats simbòliques J_i^k . Per aquest sistema robòtic definirem el següent conjunt de conductors de rols:

| | |
|--|---|
| D₁ : Protegir porteria | D₂ : Passar la pilota |
| D₃ : Rebre la pilota | D₄ : Demanar la pilota |
| D₅ : Interceptar la pilota | D₆ : Obrir joc |
| D₇ : Desmarcar-se | D₈ : Perseguir contrari |

Alguns d'aquests rols consisteixen en moviments que han de coordinar-se amb els moviments d'altres agents. Així, els algorismes propis de la negociació que permetin

avaluar quina és la viabilitat del rol s'han de basar en els valors dels graus de compromís que s'intercanvien els diferents mòduls agents: la negociació és un procés que es realitza mitjançant diàleg entre agents amb el protocol FIPA-Contract-Net (veure secció 4.2.2.3). En aquest exemple ens fixarem amb el rol D_2 , és a dir, el rol que fa que un robot pugui passar la pilota a un altre robot. És un rol que necessita d'un pas previ de negociació de les condicions finals del rol: posició final desitjada per a la pilota a un instant de temps concret. El conductor D_2 d'aquest rol es podria definir de la següent manera:

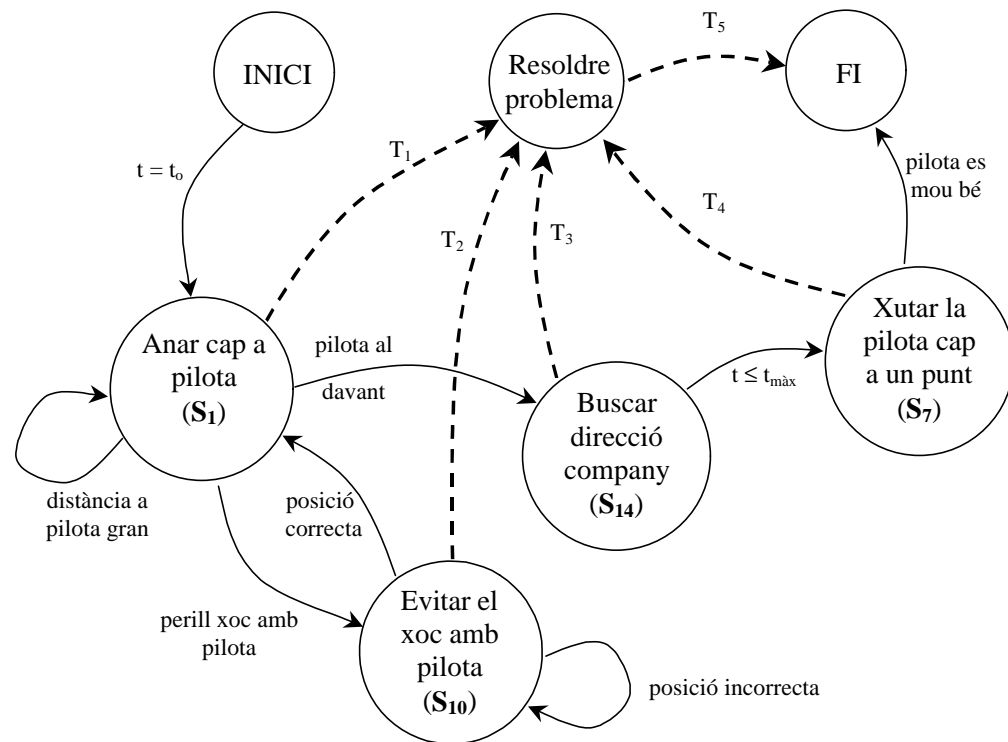
$IN_2 : \{ t \}$

$IC_2 : \{ \langle INICI: (x,y)_{PILOTA}, (v_x,v_y)_{PILOTA}, \rangle, t = t_0 \}$

$FC_2 : \{ \langle FI: (x,y)_{DESTI PILOTA}, t_{FINAL} \rangle, \langle "pilota es mou bé", T_5 \rangle \}$

$AA_2 : \langle S_1, S_{10}, S_{14}, S_7 \rangle$

CL_2 : Màquina d'estats de la figura:



LB_2 : "Passar la pilota".

Les capacitats associades a aquest rol cal obtenir-les només a partir dels índexs proporcionats pels supervisors S_1 i S_7 donat que són els únics que estan relacionats amb els moviments del robot. Així, el grau de compromís d'execució del rol serà una combinació de les capacitats bàsiques J_1^k i J_7^k corresponents als supervisors S_1 i S_7 .

4.3.3.2. Exemple real: comboi de dos robots (III)

En l'exemple del comboi hem vist que cada robot només disposa d'un únic controlador que s'utilitza tant per frenar com per anar més ràpid. Per tant, no cal un conductor que gestioni el funcionament de diferents supervisors donat que només en cal un de sol que hem anomenat S_3 (veure l'apartat 4.3.2.2). Ara bé, si en el model dels vehicles es volgués tenir en compte que l'augment o disminució de la velocitat implica canvis en el model (per exemple, canvis de marxa en el vehicle) aleshores faria falta afegir nous controladors i els corresponents supervisors, i després caldria dissenyar un conductor.

Per finalitzar aquest exemple del comboi que hem desenvolupat a través de diversos apartats, ara dissenyarem el mètode que permeti negociar les condicions d'aturada entre els dos vehicles. Per la simplicitat del problema no ens cal dissenyar cap conductor i podríem considerar que és innecessari. Així, l'índex que s'obté a partir de la capacitat bàsica J_3 el podem considerar directament com a l'índex de viabilitat d'executar el rol de frenada, és a dir, el paràmetre μ del missatge Confirm (Action, time, μ). L'algorisme de negociació que proposem en cinc passos és el següent:

- Pas 1. L'agent A decideix canviar la velocitat i envia un missatge a l'agent B:
request ("Aturar-se", temps);
- Pas 2. L'agent B rep el missatge i busca alguna capacitat que compleixi els requeriments.
IF CAPACITY exist
THEN
 inform ("Aturar-se", μ);
ELSE
 inform ("Aturar-se", $\mu=0.0$);
END
- Pas 3. L'agent A rep la resposta:
IF $\mu >$ threshold
THEN
 inform ("Aturar-se", DO);

L'agent B rep l'ordre i inicia l'execució de l'acció enviant la comanda al mòdul supervisor i confirmant-ho a l'agent A:

```
    INT request ("Aturar-se", temps);  
    Confirm ("Aturar-se", START);  
ELSE
```

- Pas 4. L'agent A busca alguna altra capacitat relativa als mateixos paràmetres de velocitats inicial i final, però amb un altre controlador, és a dir, canvia l'especificació del temps de frenada i l'algorisme torna al pas 2:

```

IF CAPACITY exist
  THEN
    request ("Aturar-se", temps);

```

- Pas 5. L'agent A decideix no executar l'acció i s'acaba la negociació:

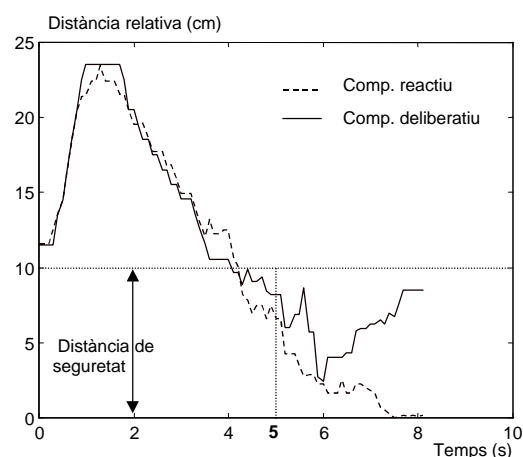
```

END

```

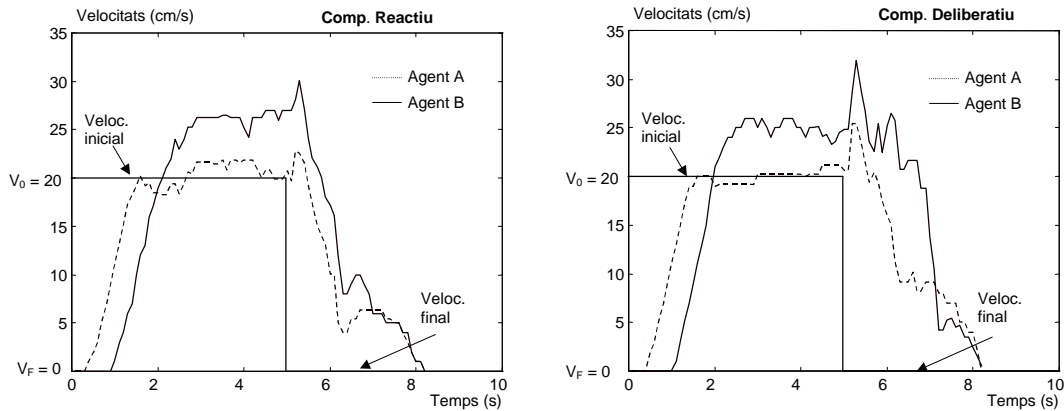
La idea que es busca amb l'algorisme de negociació és aconseguir que l'agent A canviï els paràmetres del seu controlador PI i tingui una dinàmica que eviti la col·lisió. A continuació es mostren diversos resultats amb els robots reals en què es compara l'evolució de la distància de separació entre ells en dos supòsits: suposant que hi ha negociació usant l'algorisme que s'ha exposat abans (comportament deliberatiu), i suposant que no n'hi ha de cap tipus (comportament reactiu). En el primer supòsit s'ha posat un valor de $\text{threshold} = 0.8$. En tots els casos el controlador de l'agent B té inicialment un valor de $K_I = 0.3$ mentre que l'agent A té un valor de $K_I = 1.0$, és a dir, inicialment l'agent B és més lent durant la seva frenada i la col·lisió és segura. Cal comentar que en els gràfics es mostra l'evolució de les velocitats dels dos mòbils considerant que inicialment estan aturats i que després de 5 segons comença l'acció de l'aturada. També es mostren els gràfics de l'evolució de la distància entre els mòbils per tal d'apreciar si es produeix la col·lisió.

Cas 1: velocitat inicial 20 cm/s

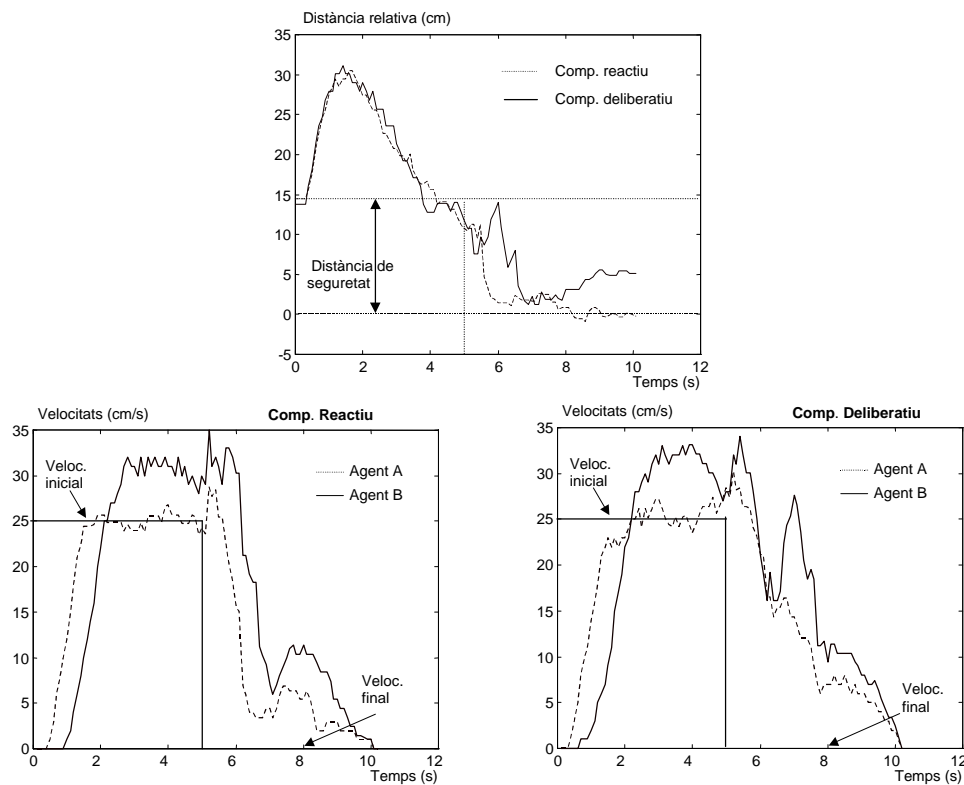


El gràfic mostra la distància de separació entre els agents amb un comportament reactiu (línia puntejada) i per a un comportament deliberatiu (línia contínua). Com es pot veure, les dues corbes coincideixen els primers 5 segons, que és l'instant de temps

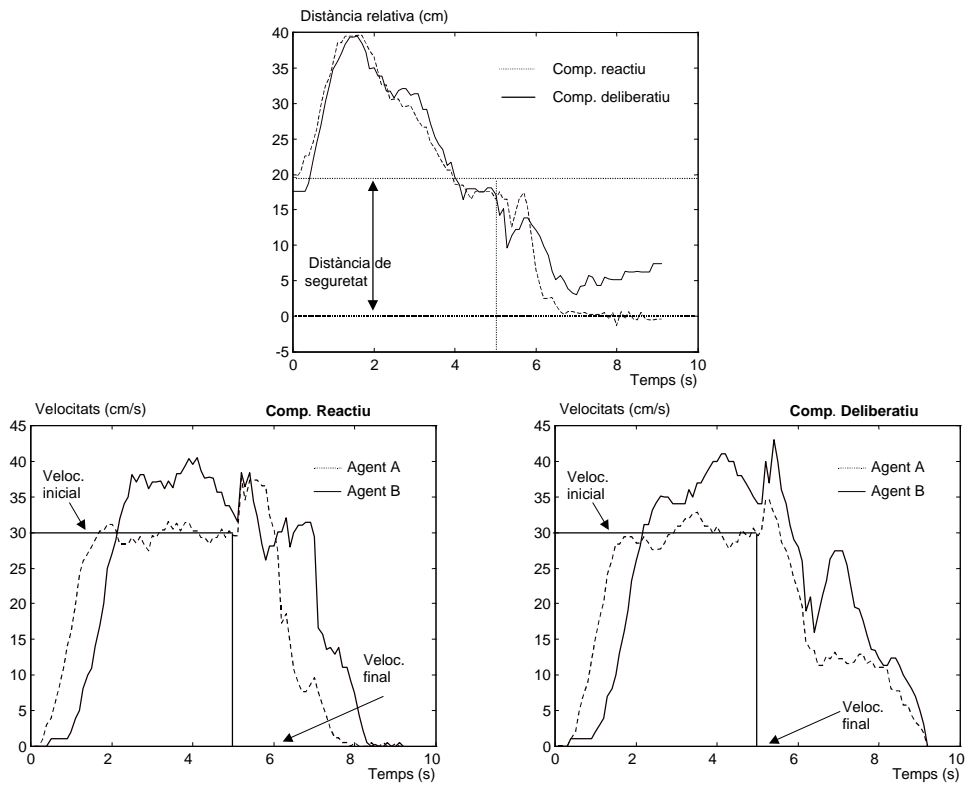
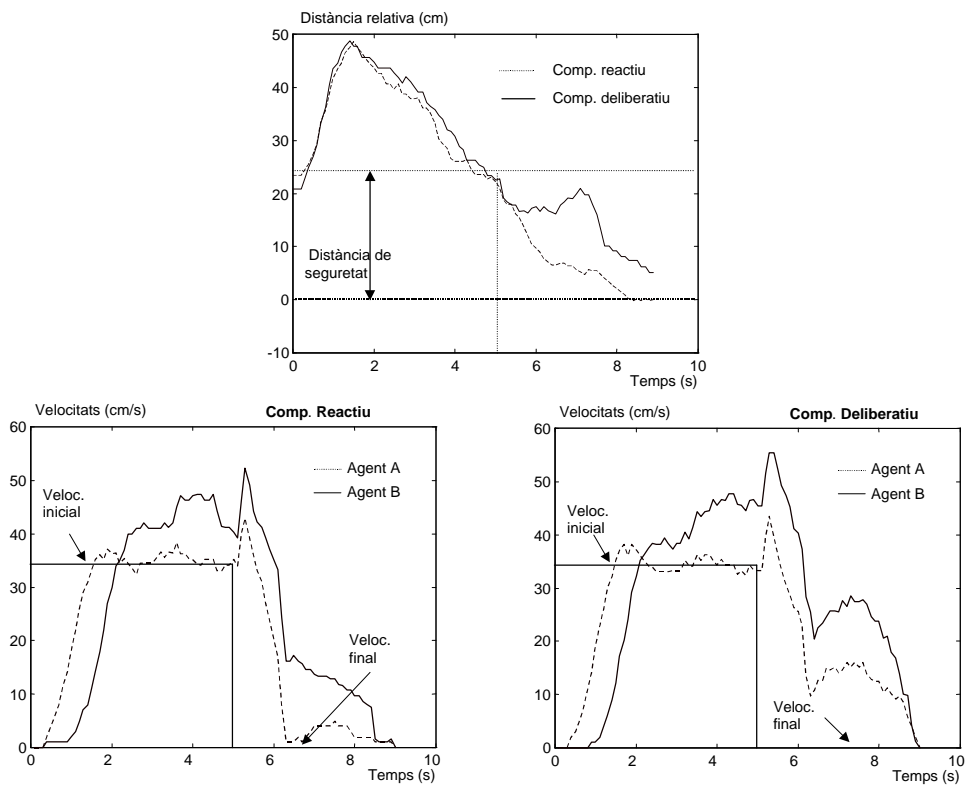
en què l'agent A decideix aturar-se. Pocs segons després s'observa que en el cas reactiu es produeix la col·lisió (la línia puntejada evoluciona cap a zero) mentre que en el cas deliberatiu la distància disminueix però no s'anul·la. Els següents gràfics mostren l'evolució de les velocitats dels dos mòbils en els dos comportaments: reactiu a l'esquerra i deliberatiu a la dreta:



Cas 2: velocitat inicial 25 cm/s



En aquest segon cas també s'observa que la distància de separació entre els robots no arriba a anul·lar-se en el cas de comportament deliberatiu gràcies a l'algorisme de negociació.

Cas 3: velocitat inicial 30 cm/s**Cas 4: velocitat inicial 35 cm/s**

En aquests dos darrers casos també s'observa com el control de la distància de separació durant l'acció d'aturada és millor en el cas deliberatiu.

4.3.4. Especificacions de les capacitats col·lectives: escenes

La necessitat d'especificar capacitats col·lectives està lligada al fet que en sistemes multi-agent sempre apareixeran relacions de comportament entre els individus que en formen part. Aquestes interaccions dels agents estan articulades al voltant de trobades de grups d'agents, que anomenarem **escenes**, amb un protocol de comunicació ben definit. Entenem per protocol els possibles diàlegs que els agents poden tenir-hi. En altres paraules, una escena consisteix en un esquema predefinit per a repartir rols entre els robots que en formen part a partir d'unes regles de comportament. Les accions dialògiques dels agents tenen unes conseqüències en forma de compromisos que imposen restriccions en els rols adquirits. Aquests rols han de ser comportaments cooperatius planificats prèviament pel dissenyador del sistema. La definició formal que utilitzarem per a l'escena és la donada a [Sierra, 98]:

Una escena és una tupla $s = \langle A, R, W, w_0, W_f, r \rangle$, on

A és un conjunt finit de variables d'agent.

$R : A \rightarrow \mathbf{Rols}$ assigna un rol a cada variable d'agent en l'escena.

W és un conjunt d'estats. $w_0 \in W$ és l'estat inicial de l'escena i $W_f \subseteq W$ els estats finals de l'escena.

r és una funció de transició entre estats, on les transicions estan etiquetades amb una il·locució.

Una escena es pot activar quan hi ha uns agents assignats per a totes i cadascuna de les variables del conjunt A . Un cop aquesta assignació està feta l'escena s'activa a partir de l'estat inicial i evoluciona cap a algun dels estats finals. Un punt important a tenir en compte és determinar quins agents abandonen quina escena, en quin estat l'abandonen, i a quina escena s'incorporen. En aquest sentit, el comportament d'un sistema multi-agent consistirà en un conjunt d'escenes específiques que poden presentar característiques concurrents o excloents. En el cas que les escenes es defineixin amb caràcter no excloent caldrà definir algun mètode de selecció d'escenes. Un treball en aquesta direcció és proposat a l'arquitectura ABC² [Matellán, 98] on el problema de la diversitat d'opcions es soluciona seguint un esquema oportunista.

Finalment cal comentar que independentment de la definició formal d'escena adoptada en aquest treball, qualsevol intent d'implementació hauria de buscar eines que

permetin la concurrència. Això es deu al fet que l'adjudicació de rols entre els agents es fa de forma completament asíncrona i que l'evolució dels agents és completament paral·lela. Un dels mètodes més usats en aquests casos és la metodologia de les xarxes de Petri en qualsevol de les seves variants.

4.3.4.1. Exemple

En robòtica mòbil els moviments en formació o els coordinats per a moure objectes grans són bons exemples d'escenes. Tornant a l'exemple del futbol robòtic, les escenes serviran per descriure com es realitza una passada de pilota, com es realitza un contraatac, o com es realitza una defensa en formació. En concret, podrien definir-se les següents escenes:

- Atac → situació general ofensiva, quan no hi ha perill a la porteria local.
- Contraatac → situació específica d'atac quan hi ha pocs contraris a la zona propera a la porteria contrària.
- Defensa zonal → situació general defensiva amb contraris ràpids.
- Defensa a l'home → situació general defensiva amb contraris no ràpids.
- Altres escenes possibles: defensa urgent, joc individual, protecció de la porteria.

Cal remarcar que la protecció explícita de la porteria no és un comportament col·lectiu com a tal, sinó que consisteix en ser un rol adjudicat a un agent concret i que mentre evoluciona el sistema global aquesta adjudicació no canvia en cap moment. Ara bé, hi ha situacions en competició real en què es produeixen bloqueigs i es fa necessari que aquesta adjudicació sigui dinàmica. És des d'aquest punt de vista que aquest rol ha de tenir un sentit col·lectiu, no com a comportament cooperatiu sinó com a comportament d'ajuda o de suport.

Hi ha un punt a considerar i és el fet relacionat amb l'emergència del comportament col·lectiu. En una situació nominal cada agent avalua la seva pertinença a diferents escenes, i la pertinença continuada a unes escenes en detriment d'altres significa que el sistema es veu forçat a comportar-se d'una manera i no d'una altra. En el cas que la pertinença sigui variant o fins i tot múltiple en diferents escenes significa que el sistema global (col·lectivitat d'agents + entorn) té una dinàmica oscil·lant o, el que és equivalent, aquest sistema no és molt estable. En qualsevol dels casos no es pot

concloure que sempre hi hagi un comportament emergent de la col·lectivitat. Per exemple, l'activació continuada d'escenes de caire defensiu pot formar part d'una estratègia col·lectiva, és a dir, pot haver-hi una intencionalitat en aquest fet. Per tant, un comportament col·lectiu ofensiu o defensiu pot correspondre a una intenció programada pel dissenyador del sistema, i haurà de incentivar l'activació d'escenes d'atac si la intenció és ofensiva o de defensa en el cas contrari. Des d'un punt de vista de disseny, es pot optar per programar aquesta intencionalitat de forma centralitzada en un agent *software*⁹ o de forma local en cada agent. En el primer cas es pot interpretar l'agent com a un supervisor de l'evolució del comportament de la col·lectivitat mentre que en el segon cas és necessari introduir alguna eina de correcció *on-line* com per exemple alguna tècnica d'aprenentatge a partir de la dinàmica dels rols.

4.4. Arquitectures híbrides per a SMA

Les arquitectures híbrides estan orientades a entorns dinàmics i impredecibles per què incorporen les propietats tant de les arquitectures deliberatives com de les reactives. En aquests entorns la capacitat de reacció, la resposta en temps real i la representació simbòlica són els requeriments més importants dels agents [Busmann, 94]. La capacitat de reacció fa referència a la supervisió contínua, és a dir, la que assegura que l'agent pugui reaccionar davant d'events inesperats. La resposta en temps real fa referència a la planificació de la seqüència d'accions per tal d'assegurar l'adaptació de l'agent a l'evolució de la velocitat de canvi de l'entorn. Finalment, la representació simbòlica està íntimament relacionada amb la capacitat d'interpretar l'entorn d'acord amb el model del món de què disposa l'agent. De fet, aquesta interpretació més abstracta és la que permet que es puguin prendre decisions més meditades per part l'agent.

La forma d'assolir aquests requeriments és dissenyant arquitectures multi-nivell on cada nivell té un paper diferent als demés tant per la seva posició a la jerarquia com per la connexió amb els altres nivells. Hi ha arquitectures que enlloc de nivells parlen de mòduls en el sentit que eliminen qualsevol tipus d'ordre jeràrquic i proposen connectar els diferents mòduls amb funcionalitats complementàries.

Les arquitectures híbrides normalment s'estructuren en tres nivells que inclouen comportaments referint-se als estats interns de l'agent. Els tres nivells són el

⁹ La primera opció és la que es va utilitzar a [Rosenau, 97] on s'utilitzava el concepte de *coach-agent* com a agent que marcava les intencions en situacions poc definides.

controlador (també sol anomenar-se *reactiu*), el qual executa comportaments purament reactius; el *seqüenciador* (també conegut com el de *control d'execució*), el qual utilitza estats de móns passats per tal de seleccionar quins comportaments assignar en un temps determinat; i el *deliberador* (també sol anomenar-se *planificador*), el qual fa prediccions dels estats futurs i executa càlculs costosos en temps com els associats als algoritmes de planificació. En aquest nivell, el més alt, el planificador transforma la missió en un conjunt de tasques que ha d'executar el sistema de control. El raonament simbòlic i altres eines relacionades amb IA també es produeixen en aquest nivell. A part del procés de control d'execució, el nivell intermedi pot entendre's com un procés de refinament o d'adaptació de les tasques en funció del context actual. Sovint l'execució d'un pla consisteix en moure el sistema a través d'una seqüència de fases o estats que es caracteritzen per un conjunt de comportaments actius. Finalment, el nivell més baix és funcionalment reactiu i s'executa de forma asíncrona respecte dels altres nivells. Sovint s'estructura en forma d'un conjunt de mòduls funcionals enllaçats entre ells.

Interrap [Müller, 96] és un exemple d'aquest tipus d'arquitectura especialment dissenyada per a sistemes multi-agent¹⁰. A Interrap, el nivell més alt (*Cooperative Planning Layer*) inclou raonament col·laboratiu usant el model BDI [Rao, 95] per tal de tenir en compte tant les intencions com els objectius dels altres agents, i per resoldre conflictes. A un segon nivell (*Local Planning Layer*) hi ha tots els mòduls funcionals relacionats amb la planificació: generació, interpretació i avaluació de plans, *scheduling*, i execució. El nivell més baix (*Behavior Based Layer*) té dues finalitats: d'una banda s'encarrega d'executar les tasques rutinàries de forma reactiva i en temps real, i per altra obtenir el coneixement procedural necessari per executar les tasques de forma eficient.

A GRATE¹¹ [Jennings, 92] l'agent es dissenya en dos nivells: un nivell anomenat de domini el qual conté el coneixement per resoldre els problemes en dominis específics, i un nivell de cooperació i control el qual avalua situacions locals i globals, i controla les interaccions socials. En aquesta arquitectura s'utilitza la idea de la *responsabilitat conjunta*: un cop un equip d'agents està d'acord en resoldre un problema aquests obeeixen un codi de conducta, és a dir, es comprometen entre ells per cooperar fins que resolen la situació. En aquest sentit, es defineixen tres tipus de cooperació: el

¹⁰ Aquesta arquitectura està explicada amb detall al tercer capítol de la primera part.

¹¹ Cal fer notar que aquesta arquitectura és deliberativa, però s'inclou per ajudar a la comprensió de l'arquitectura *ARCHON*.

task-sharing, en què un agent busca ajuda quan no pot completar una tasca, el *result-sharing*, en què els agents comparteixen resultats que poden ser útils per a altres, i el *joint-action*, en què un agent executarà certes accions si altres agents n'executen d'altres de complementàries.

A ARCHON (*Architecture for Cooperative Heterogeneous On-line Systems*) [Jennings, 95] l'agent continua tenint dues components però funcionalment no es corresponen al mateix que a GRATE: una component és el *Intelligent System* que consisteix en un entitat pre-existent que sap resoldre problemes, i l'altra és l'*ARCHON Layer*, que es dedica a mantenir els models i a gestionar la comunicació amb altres agents. Aquesta component disposa d'un model de si mateixa i decideix les accions a fer a partir del mòdul PCM (*Planning & Coordination Module*). En aquesta arquitectura el rol d'un agent és el que determina la seva actitud per a cooperar. Així, pot dedicar-se a resoldre problemes propis (rol *serves-self*), pot actuar com a servidor cap als altres com a base de dades (rol *serves-others*) o pot actuar d'acord amb les seves necessitats i per ajudar als altres (rol *mixed*). Per tant, ja no es parla de tipus de cooperació, sinó d'actituds de cooperació.

A MECCA (*Multi-agent Environment for Constructing Cooperative Applications*) [Steiner, 90] l'agent està format per tres parts o mòduls: *body*, *head*, i *communicator*. El primer mòdul conté el conjunt de capacitats, el segon fa l'*scheduling* i la cooperació, i el darrer es dedica a gestionar el pas de missatges. Aquesta arquitectura pot funcionar tant de forma reactiva com de forma deliberativa gràcies als diferents tipus de protocols de comunicació que incorpora: el *contract-net* [Smith, 80], el mestre/esclau i un mètode de negociació específic. De fet cal esmentar que també possibilita la definició de mètodes de cooperació, la qual cosa permet que altres sistemes intel·ligents basats en agents puguin acomodar-se a MECCA.

Una altra arquitectura és la proposada a [Mackworth, 93] que utilitza el model dels *Constraint-Nets* (CN) com a plataforma de treball unitària per desenvolupar sistemes intel·ligents híbrids. La idea és obtenir un model abstracte per a sistemes dinàmics en general, en què s'hi barregen el temps discret amb el temps continu i els events asíncrons amb les estructures síncrones. Aquesta barreja fa referència al fet que un sistema robòtic està compost de tres entitats: el controlador (sensors, actuadors i algorismes de control), la planta (braç robòtic o un vehicle amb rodes motrius) i l'entorn (obstacles o objectes, el terreny). Tant la dinàmica de l'entorn com la dinàmica del robot es modelen amb estructures de CN jeràrquiques, i a partir d'estructures

algebraiques i de topologia analitza el sistema a diferents nivells d'abstracció. Aquesta formalització proporciona una semàntica per desenvolupar un llenguatge de programació de temps real amb què es poden especificar/dissenyar les propietats del robot. Aquesta arquitectura s'ha aplicat a un robot amb dos braços, a un robot mòbil que es mou en un laberint i a un grup de robots que juguen a futbol.

Finalment, comentar l'arquitectura HUGHES [Payton, 86] que té dos requeriments fonamentals: la *celeritat* (reactivitat) en relació a la incertesa de la informació sensorial i l'*assimilació* (deliberació). En aquesta arquitectura es proposen quatre nivells: planificació de la missió, planificació basada en mapes, planificació local, i planificació reflexiva. Cada nivell està fragmentat en un conjunt de mòduls més petits: són agents experts capaços de rebre combinacions de dades sensorials processades i de comunicar-se entre ells en un mateix nivell a través d'una pissarra (*blackboard*). Cada agent es compon d'un sensor virtual i d'un comportament reflexiu. El sensor es pot interpretar com un o més processos associats a un o més sensors físics, mentre que un comportament és un procediment parametrizat que proporciona ordres al controlador del robot en funció de les dades rebudes pels sensors. El mecanisme per resoldre conflictes es basa en la relació de prioritats entre les ordres enviades pels comportaments i les prioritats entre els comportaments del nivell. Finalment, els errors es gestionen a través d'un processos paral·lels que poden aturar o reiniciar una activitat segons com evolucioni el sistema.

5. Exemple d'aplicació: el futbol robòtic

En aquest capítol es presenta un exemple d'aplicació de la DPAA basat en un domini de recent aparició en el món científic: consisteix en un equip de robots que emulen el joc del futbol. En les següents seccions s'explica com es pot realitzar el disseny d'un agent robòtic capaç de funcionar en un equip de robots d'aquestes característiques. L'explicació es centra fonamentalment en mostrar detalladament com es calculen les capacitats atòmiques en el mòdul de control i quin és el seu efecte en les capacitats en els mòduls superiors. Mostrar resultats reals no resulta senzill si es tenen en compte les característiques de l'entorn i, per tant, s'ha optat per realitzar un simulador molt realista de robots mòbils. Aquest simulador permet analitzar la dinàmica dels moviments dels robots amb molta precisió i s'ha preparat per a escenificar una situació típica d'aquest domini que necessita que la cooperació entre dos robots es realitzi de forma molt eficient: la passada de la pilota.

5.1. El domini del futbol robòtic

A les àrees de recerca aplicada de la robòtica mòbil i de la intel·ligència artificial hi ha tot un seguit de competicions orientades a posar a prova diferents tècniques i algorismes. El denominador comú d'aquestes competicions és la presència de robots mòbils. Hi ha competicions destinades a introduir noves idees sobre el disseny de robots, com són les competicions de robots voladors, robots caminadors amb diverses potes, micro-robots que surten de laberints, etc. D'altres competicions més noves com la AIPS'98 (*4th Int'l Conf. on Artificial Intelligence Planning Systems*, celebrada a la *Carnegie Mellon University*) estan destinades a trobar els millors algorismes de planificació de tasques en entorns més o menys estructurats utilitzant el robot com a objecte mòbil autònom. De tota manera, hi ha molt poques competicions destinades a provar el funcionament de sistemes formats per diversos robots.

En els darrers anys ha aparegut la idea de fer competicions de futbol, és a dir, enfrontar dos equips de robots emulant el joc del futbol. Aquest tipus de competició permet testejar les tecnologies relacionades amb els sistemes de visió artificial (captació, processat, anàlisi i seguiment d'objectes mòbils), els sistemes de comunicació (senyal de ràdio i de televisió), els sistemes mecatrònics per la motricitat de robots (rodes omnidireccionals, sistemes d'aprehensió), i sistemes sensorials diversos (ultrasons, infrarojos, tàctils). Deixant de banda els aspectes tecnològics, la vessant científica és la que aporta més empenta a aquest tipus de competició donat que aquestes competicions sempre tenen lloc dintre d'events científics internacionals com la ICMAS (*Int'l Conf. on Multi-Agent Systems*), o la IJCAI (*Int'l Joint Conf. on Artificial Intelligence*), o els mateixos *workshops* associats a les pròpies competicions.

5.1.1. Breu història sobre les competicions de robots futbolistes

Diversos investigadors han proposat una plataforma comuna per a la prova de diferents tecnologies: integració de sensors, raonament en temps real, el disseny d'agents autònoms en forma de robots mòbils, la cooperació i el control en sistemes multi-agent, etc. Aquesta

plataforma consisteix en el desenvolupament d'un partit de futbol entre dos equips de robots completament autònoms.

La idea de robots jugant a futbol va estar proposada inicialment per Alan MacWorth l'any 1992 en un projecte anomenat *Dynamo* [MackWorth, 93][Zang, 94]. El juny del 1993 un grup d'investigadors japonesos també varen proposar-ho com a eina de promoció de la ciència i la tecnologia englobat en un projecte anomenat *J-League (Japan League)*. Posteriorment aquest projecte es va estendre com a projecte internacional amb el nom de *Robot World Cup Initiative*, abreujat RoboCup. El centre d'investigació japonès *Electro Technical Laboratory (ETL)* va començar a desenvolupar un simulador de partits de futbol i el començà a utilitzar com a eina de recerca en sistemes multi-agent.

El setembre del 1993 es va fer el primer anunci oficial de la iniciativa RoboCup i es varen començar a redactar les regles de joc i els aspectes tècnics relacionats. A la conferència nacional d'intel·ligència artificial d'Estats Units (AAAI-94) del 1994 ja es va presentar la versió 0 del simulador oficial de la RoboCup i investigadors d'arreu del món ja estaven involucrats en el projecte (Japó, Estats Units, i alguns països europeus, fonamentalment Alemanya). L'anunci de la primera competició RoboCup es va fer durant la *Internacional Joint Conference on Artificial Intelligence (IJCAI-95)* l'agost del 1995 a Montreal: s'anomenaria *Robot World Cup Soccer Games and Conferences*, i se celebraria en el marc del IJCAI-97 a Nagoya (Japó). També es va proposar celebrar una Pre-RoboCup l'any 1996 amb joc simulat amb una versió millorada del simulador presentada al IJCAI'95: *Soccer Server v.1.0*.

De forma paral·lela, a principis del 1996, Jong-Hwan Kim del *Korean Advanced Institut of Science and Technology (KAIST)* de Corea del Sud va organitzar una competició semblant per al Novembre del mateix any, és a dir, un any abans de la primera RoboCup. Per potenciar-la va organitzar un *summer camp* on van participar-hi investigadors de Universitats de Corea, Japó, Estats Units, Espanya (l'autor d'aquesta tesi), Itàlia, Holanda, França, Brasil i Austràlia. Aquesta competició s'anomenaria *MiRoSot (Micro-Robot Soccer Tournament)* i els robots serien lleugerament més petits (les mides màximes serien les d'un cub de 7.5cm de costat).

La Pre-RoboCup es va celebrar del 4 al 8 de Novembre del 1996 a Osaka (Japó) on es va fer una lliga simulada (*Simulation League*) amb 8 equips. També es va fer una exhibició

amb robots reals corresponents a la *Middle Size League* (50cm de diàmetre).

La primera competició internacional de robots jugant a futbol es va celebrar a la ciutat coreana de Taejon el Novembre del 1996 al KAIST. En aquesta competició tot el joc era amb robots reals i amb equips de només tres robots. També hi havia una modalitat de joc amb equips formats per un sol robot (modalitat S-MiroSot). Van participar-hi grups de recerca de Corea (diversos equips del mateix KAIST), Estats Units (MIT i CMU), Espanya (Univ. de Girona) i Suïssa (LAMI), i investigadors d'Austràlia i Japó varen participar-hi amb robots cedits pel mateix KAIST amb la interessant fórmula de fer IJT's (*International Joint Teams*).

Els primers *Soccer Games and Conferences* de la RoboCup es celebraren l'any 1997 amb una participació de quatre equips a la *Small-Size League* (centres universitaris de Japó, Estats Units, Espanya i França), 6 equips a la *Middle Size League*, i més de 30 a la *Simulation League*. Les mides dels robots de la categoria petita eren les necessàries per encabir-los en cilindres de 20cm de base i 20cm d'alçada, i els de la mida mitjana arribaven fins els 50cm. Tots els equips estaven formats per cinc robots. A la lliga simulada els equips estaven formats per onze jugadors, i les especificacions del simulador oficial obligaven a programar onze agents autònoms i comunicant-se entre ells a través de la xarxa.



Figura 5.1.1: Fotografia en un moment de la competició l'any 2000, a Melbourne.

Ambdues competicions s'han anat celebrant anualment de forma paral·lela. El juny del

1997 MiroSot va celebrar-se també a Corea del Sud amb un grau de participació bastant semblant al del primer any. Durant el mateix any es varen celebrar competicions continentals (Europa, Àsia i Estats Units) prèvies a la de Corea. L'any 1998 ambdues competicions varen coincidir a París amb la celebració del campionat mundial de futbol real, i se celebraren en conjunció amb la ICMAS-98 (*International Conference on Multi-Agent Systems'98*). Aquell any la MiroSot va presentar dues noves modalitats: la competició amb robots encara més petits (3.75×3.75×3.75 cm, anomenada NaroSot) però amb equips ja de cinc robots, i la competició amb robots Kephra. La RoboCup, en canvi, va experimentar un augment molt gran de participació en totes les categories. Posteriorment, les competicions s'han anat celebrant anualment en paral·lel i s'han anat consolidant fins enguany que s'han celebrat a Corea i al Japó coincidint amb la copa mundial de futbol FIFA-2002.

L'interès per aquesta plataforma està augmentant arreu del món i la competició s'està convertint en l'event de robòtica mòbil i intel·ligència artificial més important. La plataforma ha donat lloc a diversos treballs sobre diverses problemàtiques dels sistemes multi-agent: coordinació, planificació, control, i aprenentatge d'entre altres. També ha facilitat el desenvolupament de tècniques noves en comunicacions, en sistemes de visió, i en mecànica. Des d'un punt de vista europeu, a principis de l'any 1999 la xarxa europea d'excel·lència *AgentLink* va donar llum verda a la creació d'un *Special Interest Group* (SIG) en coordinació i control de sistemes multi-agent creat fonamentalment a partir de gent involucrada a la RoboCup. Tot això ha de donar idea sobre la importància que està adquirint el vessant científic d'aquesta plataforma.

Enguany, la RoboCup ha inclòs variants a la plataforma per a facilitar experiments relacionats amb la temàtica del rescat de víctimes mitjançant elements robòtics. Aquesta variant sobre la plataforma s'anomena *Rescue Robotics*, és a dir, té una clara visió d'aplicació de la Robòtica Mòbil col·lectiva en un entorn del món real, encara que sigui a petita escala. Cal esmentar que aquest nou camp d'aplicació de la robòtica mòbil és molt important al Japó i als Estats Units, on es pensa en aplicacions reals en inundacions, terratrèmols, incendis urbans, etc. A Europa també està adquirint certa importància, donant llum verda a diversos projectes de recerca en aquest sentit, com el projecte portuguès "RESCUE - Navegación Cooperativa para Robots de Salvamento" desenvolupat al ISR

(Inst. de Sistemas e Robótica, de Lisboa) amb un finançament de 100000 EURO. Citem aquest projecte perquè el ISR ha estat participant a la RoboCup des de la segona edició l'any 1998, i l'any 2000 va aconseguir proclamar-se com a campió a la categoria *Simulation League*.

5.1.2. La complexitat del domini dels robots futbolistes

El domini del futbol robòtic trenca amb moltes de les suposicions en què treballa la GOFAIR¹ i, en canvi, compleix les especificacions per a treballar amb *Situated Agents*. Aquest domini es caracteritza per les següents propietats [MackWorth, 93]:

- Hi ha agents neutrals, amics i hostils
- Hi ha cooperació entre agents
- Hi ha interacció en temps real
- L'entorn és dinàmic
- El món és real i impredecible
- Hi ha criteris objectius de rendiment
- Els experiments són repetitius.

Tot i que el futbol no és el món real, sí que és un bon exemple on hi apareixen problemes que sí són els mateixos que poden aparèixer en el món real. De fet, el futbol és una abstracció del món suficientment apropiada com per a fer-hi recerca en IA i per encaminar el desenvolupament de teories més abstractes [Macworth, 93][Asada, 97] [Kitano, 97]. A nivell mundial la repercussió científica ha estat molt forta i ha donat lloc a un gran nombre de publicacions científiques en àmbits de la Robòtica, la Visió per Computador, la Intel·ligència Artificial, i de forma molt més general, en totes les temàtiques relacionades amb l'àmbit dels sistemes multi-agent.

En referència als sistemes multi-agent, els treballs es centren en diversos temes:

- Plantejar i resoldre situacions de cooperació entre dos o més robots.
- Plantejar i resoldre situacions de joc que requereixen moviments coordinats, típicament moviments en formació (atac, defensa,...).
- Planificació *on-line* i *off-line* d'estratègies de joc: els recursos són l'espai, la pilota.

¹ Veure secció 3 del capítol 3 de la primera part.

- Resolució de conflictes en les decisions dels comportaments individuals dels robots.
- Algorismes de decisió i aprenentatge en temps real, sempre en funció del comportament de l'equip contrari.

Des del punt de vista d'aquesta tesi, la riquesa de possibilitats que ofereix aquest domini està bàsicament en el fet que incorpora problemàtiques pròpies de dos àrees de coneixement (robòtica mòbil i intel·ligència artificial). Aquesta coincidència ha provocat que els investigadors hagin de treballar amb equips multidisciplinars i moltes vegades amb eines de treball diferents a les que estan habituats. Així, la gent de intel·ligència artificial s'enfronten amb sistemes reals que introdueixen elements dinàmics, sorolls i amb perturbacions constants. Els de la robòtica mòbil es troben davant d'un sistema on moltes tècniques que resolen problemes per a un sol robot, no són vàlides per treballar amb més robots i, menys encara, amb un altre equip de robots en situació de competitivitat.

Les problemàtiques que es consideren importants en aquesta tesi estan relacionades amb el control i la coordinació dels moviments dels robots, i la via de resolució és a través del concepte d'agent físic. El que es mostrarà en aquest capítol és com es pot utilitzar l'arquitectura DPAA per dissenyar els robots que formen part d'un equip de futbol d'aquestes característiques. En concret, s'explicarà com dissenyar el comportament dels robots començant pel mòdul de control i amb l'objectiu de resoldre l'escena de la passada de la pilota.

Aquesta escena ha estat objecte d'estudi en altres treballs [Stone, 98][Diumé, 99] però amb una orientació completament diferent. El problema de la passada de la pilota és un problema que presenta un grau de complexitat molt alt, sobretot si en tenen en compte les dinàmiques dels robots i de la pilota. El treball de P.Stone de la Univ. de Carnegie Mellon analitza el problema utilitzant tècniques d'aprenentatge reforçat per nivells (*layered learning*): fa un aprenentatge *off-line* de diverses situacions de passada de pilota utilitzant el simulador RoboSoccer. El treball de X.Diumé en el sí del grup ROGI de la Univ. de Girona utilitza xarxes neurals com a eina d'aprenentatge *off-line*. L'aprenentatge es realitza simulant un conjunt de situacions canòniques amb el simulador *Javasoccer*². Aquests dos

² El simulador *Javasoccer* ha estat desenvolupat per T.Balch. És un simulador molt flexible realitzat en Java i que permet realitzar modificacions a mida. En concret, el grup de Girona ha introduït la dinàmica de la pilota. El problema és que, actualment, el codi no és portable a un sistema operatiu en temps real.

treballs no tenen en compte restriccions temporals en el xut de forma que els resultats que obtenen permeten decidir si el xut es pot fer bé basant-se en els resultats de l'entrenament anterior. Cal comentar que els aprenentatges es realitzen sense incloure dinàmica dels robots i, en tot cas, els errors que aprecien en els assaigs amb robots reals s'expliquen justificant-se en la dinàmica dels robots.

5.2. Disseny dels agents amb DPAA

A continuació s'especifiquen quins són els trets principals del sistema, tant el que fa referència a l'entorn en què es mouen els robots com el que fa referència al funcionament dels robots. Alguns detalls tècnics de funcionament es comenten a l'apèndix III perquè seran importants per a la comprensió dels resultats que s'exposaran al final d'aquest capítol.

L'experimentació es basa en la plataforma més comuna dels primers anys d'aquests esdeveniments, és a dir, el període 1996-99 (veure l'apèndix I). Les dimensions dels robots són equivalents a les d'un cub de 7.5cm de costat, i es desplacen en un entorn tancat de 120x90 cm (camp de joc). En aquest entorn hi ha definides dues zones 30cm de llarg que fan el paper de porteries i que estan situades als dos extrems del camp de joc. Hi ha una barrera física que limita els moviments tant dels robots com de la pilota. Els robots tenen completa autonomia motriu, és a dir, disposen de controladors propis que els permeten desplaçar-se a través del camp de joc. La motricitat està proporcionada per dos motors c.c. de forma que permeten que el robot pugui realitzar tot tipus de moviments. Tenint en compte que els motors no poden girar en sentit contrari, el moviment de rotació sobre sí mateixos no és possible. Per tant, qualsevol maniobra de desplaçament donarà lloc a trajectòries rectilínies a una velocitat màxima de 70 cm/s o en forma d'arc (a velocitat lineal màxima inferior a 70 cm/s). Una especificació mecànica de les primeres generacions del període 1996-99 a tenir en compte és el fet que els robots no disposen de cap mecanisme de xut per impulsar la pilota, és a dir, tant la velocitat com la direcció del xut vindrà determinada per les condicions prèvies al xut: velocitat lineal i direcció del robot. La velocitat de la pilota (mòdul i direcció) abans del xut no determina en cap manera la velocitat (mòdul i direcció) de la pilota després del xut. Això es deu al fet que el xoc entre el robot i la pilota és completament inelàstic. De fet, el xut es produeix empenyent la pilota

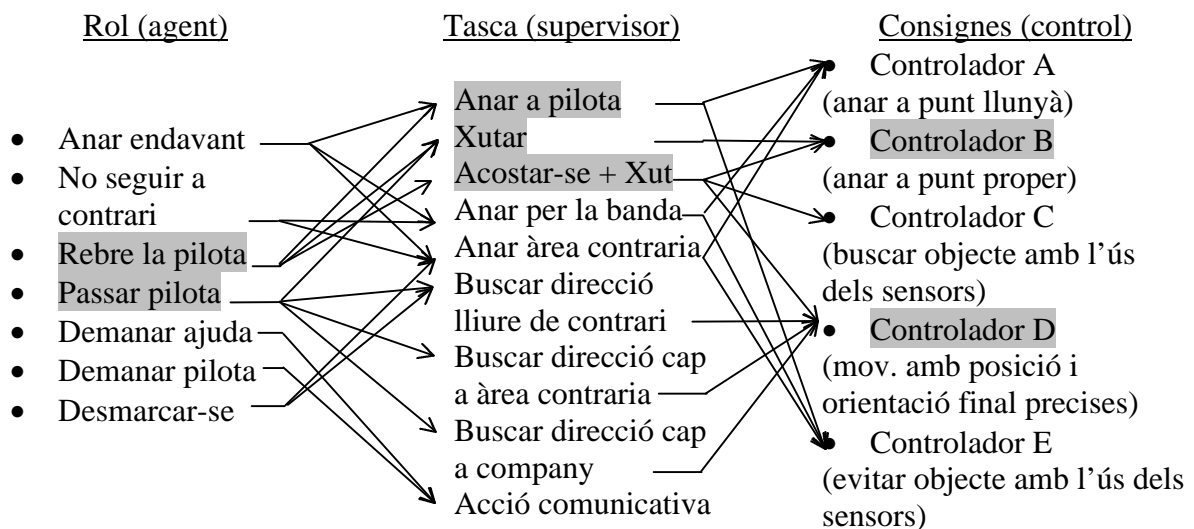
en lloc de colpejar-la. Per tant, la direcció del moviment de la pilota posterior al xut serà la del robot prèvia al xoc.

Des d'un punt de vista de funcionament, cal dir que els robots es mouen d'acord a les comandes enviades per un ordinador central. Aquestes comandes són consignes de velocitat per a cada roda. Aquest ordinador és el que disposa de tota la informació referent a l'estat del sistema (posicions i velocitats de tots els objectes en el camp de joc) i és on es duu a terme tota la negociació entre els agents. A efectes pràctics, es pot suposar que el temps de cicle d'aquest procés de negociació és de 50ms.

5.2.1. Anàlisi prèvia del problema

En aquesta anàlisi es considerarà que l'equip pot mostrar només dos tipus de comportament col·lectiu: l'ofensiu i el defensiu.

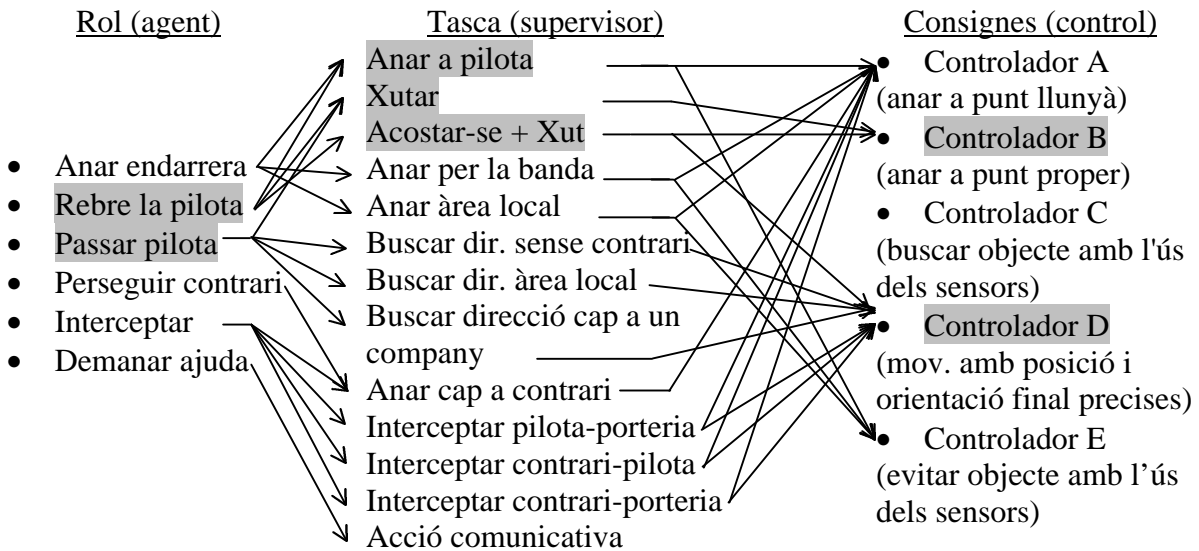
Les accions pròpies d'un comportament ofensiu poden ser les que es proposen a continuació:



En aquest diagrama es mostren quines són les relacions que es poden establir entre les accions, és a dir, pot observar-se per a cada tasca quins són els controladors que necessita i per a cada rol quins són els supervisors que necessita. Pot observar-se que en el mòdul de control cal especificar les característiques de cinc controladors diferents: "Anar a punt

llunyà", "Anar a punt proper", "Buscar un objecte amb l'ús dels sensors", "Moviment amb posició i orientació final precises" i "Evitar un objecte amb l'ús dels sensors".

Quant a les accions pròpies d'un comportament defensiu, es proposen les següents:



L'anàlisi s'ha resolt de manera que en aquest comportament defensiu també apareguin els mateixos cinc controladors al mòdul de control que en el cas del comportament ofensiu. En aquest segon esquema s'ha intentat contemplar molts casos per tal de mostrar que si es vol fer una anàlisi molt completa, el disseny de les accions en un mòdul i les relacions amb les accions del mòdul inferior es pot complicar molt: és una mostra més de la complexitat del domini del futbol robòtic.

A partir d'aquesta anàlisi es pot concloure que el disseny dels supervisors de tasques i dels conductors de rols es basaran en última instància en el bon funcionament de cinc controladors. A continuació es mostraran les característiques d'alguns d'aquests controladors i com afecten en el disseny dels mòduls superiors.

5.2.2. Especificacions dels controladors: capacitats atòmiques

L'estat d'un mòbil es pot establir a partir dels següents conjunts de variables o representacions (veure figura 5.2.2):

$$\left\{ \begin{array}{l} S_m(t) = \{ x, y, \theta, V_E, V_D \}, \text{ o bé} \\ S_m(t) = \{ x, y, \theta, V_{LIN}, V_{ANG} \}, \text{ o bé} \\ S_m(t) = \{ x, y, \theta, V_X, V_Y \} \end{array} \right.$$

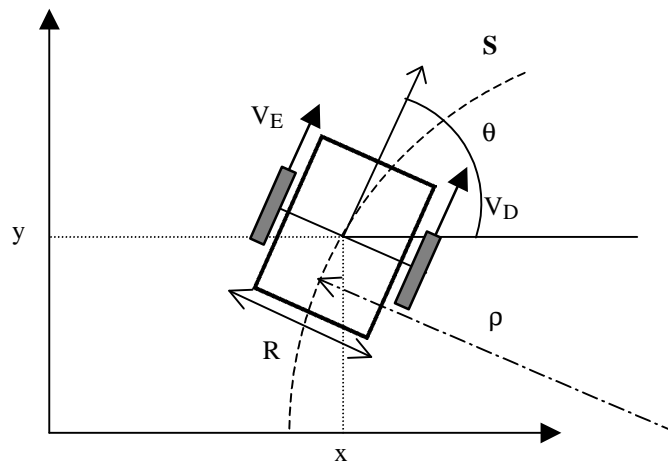


Figura 5.2.2: Esquema de les variables que defineixen el model d'un robot mòbil amb tracció diferencial.

Suposant que el robot disposa d'un sistema de tracció diferencial, és a dir, que el moviment de les rodes és independent l'una de l'altra, la relació entre les diferents representacions s'estableix a partir de les següents relacions cinemàtiques:

$$\left\{ \begin{array}{l} V_X = \frac{dx}{dt} = \cos \theta \cdot \frac{(V_E + V_D)}{2} \\ V_Y = \frac{dy}{dt} = \sin \theta \cdot \frac{(V_E + V_D)}{2} \end{array} \right. \quad \left\{ \begin{array}{l} V_{ANG} = \frac{d\theta}{dt} = \frac{(V_E - V_D)}{R} \\ V_{LIN} = \sqrt{V_X^2 + V_Y^2} \end{array} \right.$$

A partir d'aquestes relacions s'observa la necessitat de controlar les velocitats lineals de les rodes del robot (V_E , V_D) per tal de poder controlar el moviment del centre de masses del robot, és a dir, les coordenades (x , y).

L'objectiu de control del moviment del robot és calcular les consignes de velocitat de cada roda per tal de possibilitar que un mòbil pugui anar d'un estat inicial a un estat final. Per tant, es tracta d'un sistema MIMO (*Multi-Input Multi-Output*) el disseny del qual no

resulta senzill si es té en compte que cal incloure les especificacions de la resposta del sistema. Aquestes especificacions han de tenir en compte les limitacions dinàmiques del robot³ i les propietats geomètriques i cinemàtiques de la trajectòria. Una altra cosa a tenir en compte en el disseny dels controladors és el fet que el tipus de moviment d'aquests robots correspon a la d'un sistema no-holònom⁴ i el càlcul de trajectòries per aquests mòbils és encara un problema obert. Per tant, el control de la velocitat dels mòbils es planteja com un problema amb un alt grau de complicació.

Per tal d'il·lustrar com s'obtenen totes les capacitats atòmiques ens basaríem en el conjunt de controladors $\langle C_1, C_2, C_3, C_4, C_5 \rangle$ esmentats a la secció anterior:

LB₁: "Anar a punt proper"

LB₂: "Moviment amb posició i orientació final precises"

LB₃: "Buscar un objecte amb l'ús dels sensors"

LB₄: "Anar a punt llunyà"

LB₅: "Evitar un objecte amb l'ús dels sensors"

Els controladors que s'analitzaran a continuació són els que resulten útils per a les tasques relacionades amb la pilota tant en el comportament ofensiu com en defensiu (marcades en gris). Així, s'estudiaran les capacitats associades als controladors C_1 i C_2 : el primer s'ha aplicat amb èxit als robots de l'equip ROGI-Team de la Universitat de Girona a les competicions MiroSot i RoboCup en què ha participat [Oller, 01], i el segon està dissenyat a partir del model linealitzat que s'explica a l'apèndix IV [Oller, 02].

Aquesta delimitació a només dos controladors es deu al fet que, com ja s'ha comentat anteriorment, l'objectiu en aquest capítol és resoldre l'escena de la passada de la pilota en què l'agent receptor ha de decidir si pot complir les condicions del xut proposades pel passador.

³ Les limitacions dinàmiques del robot mòbil fan referència a les condicions per evitar que les rodes patinin, i als valors màxims de velocitats i acceleracions tangencials i normals. El detall d'aquestes limitacions es troben a l'apèndix II.

⁴ Cal tenir en compte que el robot només es pot moure en la direcció perpendicular a l'eix de les rodes motrius, és a dir, que apareix la següent condició: $dy/dx = \tan(\alpha)$. Aquesta condició no és integrable [Barraquand, 89] i, per tant, dona un caràcter no-holònom al sistema.

5.2.2.1. "Anar a punt proper"

L'objectiu d'aquest controlador és el de desplaçar el robot des d'una posició inicial a una de final relativament propera. Les variables d'entrada i l'expressió analítica són les següents:

- Distància relativa entre la posició inicial i la final (variable 'rr', en cm.)
- Angle relatiu entre l'angle d'orientació del mòbil i l'angle de la recta que uneix els punts inicial i final del centre del robot (variable 'bb', en radians)
- Velocitat lineal de la pilota (variable 'v_linb', en cm/s.)
- Velocitat actual del mòbil (variable 'vel', calculada a partir de les coordenades)
- Temps de mostreig (variable 'dt', amb valor constant de 50ms.)

```

factor → constant positiva inferior o igual a 3
KPV    → constant positiva adimensional
KPA    → constant positiva ((cm/s)/rad)
0.06   → 1/(2R)=0.06, on R és la distància entre les dues rodes(cm)
(x, y) → coordenades de posició del robot (cm)

dx=(x(k-1)-x(k));
dy=(y(k-1)-y(k));
vel=sqrt(dx*dx+dy*dy)/dt;
gir=a_rel*180/pi;
consigna_v=(d_rel+v_linb*dt);

v_motor_dret = 10.*(KPV*(consigna_v-vel)+KPA*gir)*factor;
v_motor_esquerra = 10.*(KPV*(consigna_v-vel)-KPA*gir)*factor;

```

Amb aquest controlador s'obtenen les consignes de velocitat lineal per a cada motor ('v_1' i 'v_2') en funció de la distància i l'orientació relatives a l'objectiu ('d_rel' i 'a_rel'), la velocitat de l'objectiu ('v_linb') i la velocitat actual del robot ('vel'). Com es veurà més endavant els paràmetres 'KPA' i 'KPV' d'aquest controlador estan dimensionats de forma que cap dels motors funcioni de forma saturada en les situacions en què ha de funcionar.

El següent gràfic mostra el diagrama de blocs corresponent a aquest controlador. S'observa que a partir de les variables d'entrada (blocs de l'esquerra) s'obtenen les velocitats necessàries pels dos motors: 'v_motor_dret' i 'v_motor_esquerra'. El diagrama també mostra com s'obtenen les velocitats lineals i angulars del robot.

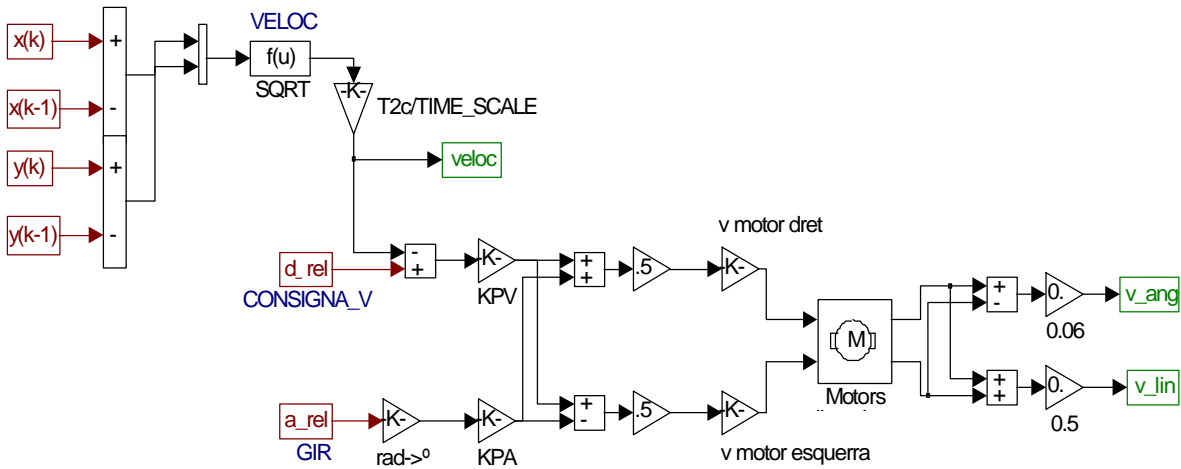


Figura 5.2.2.1: Diagrama de blocs del controlador: implementació amb Matlab/Simulink

Per tant, el controlador $C_1 = \langle \mathbf{IN}_1, \mathbf{IC}_1, \mathbf{FC}_1, \mathbf{AA}_1, \mathbf{CL}_1, \mathbf{LB}_1 \rangle$ queda definit de la següent manera:

```

IN1 : {v_lin_robot, ang_relatiu, dist_objectiu, v_lin_objectiu,
         temps_mostreig }

IC1 : { (x, y, q)robot }

FC1 : { (x, y)objectiu }

AA1 : {motor dret, motor esquerra}

CL1 :
    { "dx=(x(k-1)-x(k));
      dy=(y(k-1)-y(k));
      vel=sqrt(dx*dx+dy*dy)/dt;
      gir=bb*180/pi;
      consigna_v=(rr+v_linb*dt);
      v_motor_dret=10.*(KPv*(consigna_v-vel)+Kpa*gir)*factor;
      v_motor_esquerra=10.*(KPv*(consigna_v-vel)-Kpa*gir)*factor; ",
      <KPv, Kpa, factor>
    }

LB1 : "Anar a punt proper"
  
```

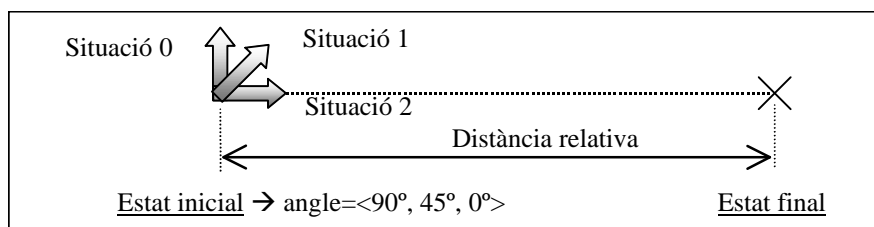
Per tal de poder obtenir el conjunt \mathbf{J}_1 de capacitats d'aquest controlador s'ha introduït el model del sistema en el simulador Matlab/Simulink (veure figura 5.2.2.1) i s'han realitzat assaigs per a diferents combinacions d'estats i per a diferents paràmetres de control. Les combinacions d'estats han de tenir en compte variacions de distància i

d'orientació relatives i per això s'han definit tres condicions inicials per l'angle i sis per la distància:

Distància relativa : { 25, 40, 55, 70, 85, 105 } cm.

Orientació relativa : { 90°, 45°, 0° }.

El següent gràfic il·lustra la geometria d'aquestes situacions:



S'han realitzat els assaigs corresponents a totes les combinacions possibles de distància i d'angle i per a diferents parametritzacions del controlador. A continuació es mostren les trajectòries que s'obtenen quan l'angle inicial és de 90° (situació 0) i amb una distància relativa de 50cm. Els diferents gràfics mostren com el valor del paràmetre 'factor' afecta tant a la forma de la trajectòria com a l'error de posició final.

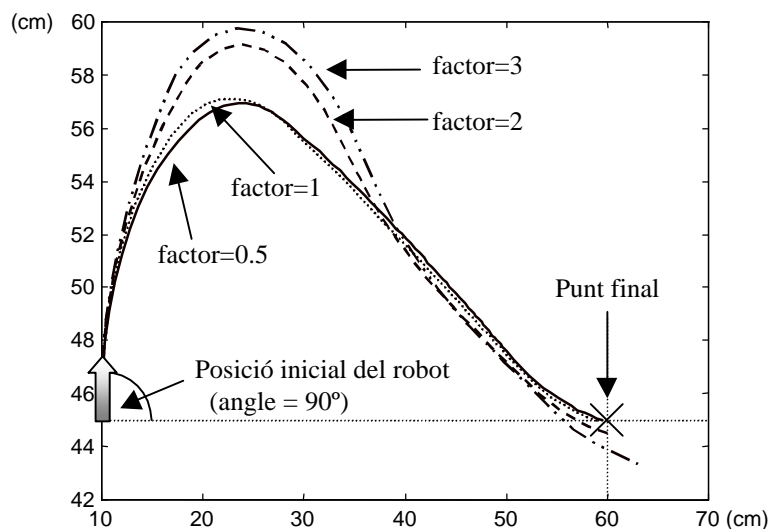


Figura 5.2.2.1.1: Trajectòries del robot amb $\text{factor} = \langle 0.5, 1, 2, 3 \rangle$

Una dada que no es veu en el gràfic anterior és el temps d'assoliment del punt final. La següent taula mostra el valor d'aquest temps corresponents a les trajectòries anteriors:

| factor → | 0.5 | 1 | 2 | 3 |
|-------------------------------|------------|----------|----------|----------|
| Temps d'assoliment (s) | 3.6 | 1.95 | 1.3 | 1.25 |

Taula 5.2.2.1.1: Temps d'assoliment per a diferents valors de factor. Dades corresponents a un angle inicial de 90°.

5.2.2.2. Capacitats atòmiques

Les capacitats \mathbf{J}_1^k que es poden obtenir en relació amb les parametritzacions \mathbf{CL}_1^k del controlador \mathbf{C}_1 poden ser indicadors que caracteritzin la geometria de la trajectòria. Com es pot veure a partir de la figura 5.2.2.1.1 aquestes característiques poden ser: el desviament màxim de la línia recta, el temps del desviament màxim, el temps d'assoliment, la distància/error en arribar al punt final, l'angle d'arribada, la velocitat lineal d'arribada, la velocitat lineal màxima, la velocitat angular màxima, etc. A continuació considerarem que el temps d'assoliment és una qualitat interessant d'aquest controlador, és a dir: $\mathbf{J}_1 = \langle T_s \rangle$. La següent taula mostra els temps d'assoliment obtinguts fent assaigs per tots els estats inicials possibles i per als quatre valors del paràmetre 'factor':

| Distància ↓ | Angle inicial = 90° | | | | Angle inicial = 45° | | | | Angle inicial = 0° | | | | |
|----------------|---------------------|------|------|------|---------------------|------|------|------|--------------------|------|------|------|---|
| | k→ | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| factor→ | 0.5 | 1 | 2 | 3 | 0.5 | 1 | 2 | 3 | 0.5 | 1 | 2 | 3 | |
| 25 | 2.6 | 1.25 | 0.75 | 0.7 | 2.4 | 1.15 | 0.65 | 0.55 | 2.3 | 1.1 | 0.45 | 0.25 | |
| 40 | | 1.4 | | | | 1.3 | | | | 1.3 | | | |
| 55 | 3.6 | 1.95 | 1.3 | 1.25 | 3.25 | 1.6 | 1.15 | 1.0 | 3.2 | 1.55 | 0.8 | 0.7 | |
| 70 | | 2.2 | | | | 1.85 | | | | 1.8 | | | |
| 85 | | 2.35 | | | | 2.1 | | | | 2.05 | | | |
| 105 | 4.8 | 2.6 | 1.9 | 1.75 | 5.1 | 2.45 | 1.75 | 1.6 | 4.85 | 2.35 | 1.6 | 1.5 | |

Taula 5.2.2.2.1: Taula de capacitats atòmiques del controlador \mathbf{C}_1 .

Per tant, d'aquest controlador obtenim en particular les següents capacitats d'anar a un punt proper:

$$\text{CAN}^t_A \{ \text{"Anar a punt proper"}, \langle (0, 0, 90^\circ), (0, 55) \rangle, \langle 3.6 \rangle \}$$

$$\text{CAN}^t_A \{ \text{"Anar a punt proper"}, \langle (0, 0, 90^\circ), (0, 55) \rangle, \langle 1.95 \rangle \}$$

$$\text{CAN}^t_A \{ \text{"Anar a punt proper"}, \langle (0, 0, 90^\circ), (0, 55) \rangle, \langle 1.3 \rangle \}$$

$$\text{CAN}^t_A \{ \text{"Anar a punt proper"}, \langle (0, 0, 90^\circ), (0, 55) \rangle, \langle 1.25 \rangle \}$$

A continuació ampliarem el conjunt de capacitats per incloure-hi paràmetres geomètrics com són l'error frontal (ΔW) i l'error d'orientació ($\Delta \alpha$) en el moment en què el

robot arriba al punt final: les capacitats ara seran $\mathbf{J}_1 = \langle T_s, \Delta\alpha, \Delta W \rangle$. La següent taula mostra aquests valors fent assaigs per els mateixos estats que abans per a una distància de 25 cm:

| Distància 25 cm | Angle inicial = 90° | | | | Angle inicial = 45° | | | | Angle inicial = 0° | | | | |
|--------------------|---------------------|------|------|------|---------------------|------|------|------|--------------------|-----|------|-----|---|
| | k→ | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| factor→ | 0.5 | 1 | 2 | 3 | 0.5 | 1 | 2 | 3 | 0.5 | 1 | 2 | 3 | |
| T_xut | 2.45 | 1.25 | 0.8 | 0.7 | 2.4 | 1.15 | 0.65 | 0.55 | 2.35 | 1.1 | 0.55 | 0.4 | |
| Da | 13.4 | 11.7 | 17.9 | 23.6 | 6.1 | 7.0 | 8.5 | 22.6 | 0 | 0 | 0 | 0 | |
| DW | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Taula 5.2.2.2.2: Taula ampliada de capacitats atòmiques del controlador C_1 .

Com abans, d'aquest controlador obtenim en particular les següents capacitats d'anar a un punt llunyà:

$$CAN^t_A \{ \text{"Anar a punt proper"}, \langle (0, 0, 90^\circ), (0, 25) \rangle, \langle 2.45, 13.4, 0 \rangle \}$$

$$CAN^t_A \{ \text{"Anar a punt proper"}, \langle (0, 0, 45^\circ), (0, 25) \rangle, \langle 2.4, 6.1, 0 \rangle \}$$

$$CAN^t_A \{ \text{"Anar a punt proper"}, \langle (0, 0, 45^\circ), (0, 25) \rangle, \langle 1.15, 7.0, 0 \rangle \}$$

$$CAN^t_A \{ \text{"Anar a punt proper"}, \langle (0, 0, 0^\circ), (0, 25) \rangle, \langle 0.55, 0, 0 \rangle \}$$

5.2.2.3. Comentaris sobre la dinàmica del robot

Les dades que s'han presentat corresponen a la simulació d'un robot amb dos motors amb una funció de transferència de primer ordre amb un temps de resposta i un guany tensió/velocitat equivalents als dels motors dels robots reals. També s'ha tingut en compte que la velocitat lineal màxima dels motors és de 70cm/s. La inclusió d'aquestes característiques dels motors és la que provoca que alguns girs no es puguin produir o que els temps d'assoliment siguin més alts que els que es tindrien si es considerés que els motors poguessin executar acceleracions infinites i sense valors màxims a les velocitats. Per tal de veure això, s'han comparat els temps d'assoliment amb un angle inicial de 45° a una distància de 55cm considerant tres models diferents per als motors:

- 1 **Model_1:** Motor amb acceleracions infinites.
- 2 **Model_2:** Motor amb acceleracions infinites i velocitat màxima de 70cm/s.
- 3 **Model_3:** Motor amb dinàmica de 1^r ordre i velocitat màxima de 70cm/s.

A continuació es mostra la taula amb els temps d'assoliment que s'obtenen amb aquests tres models:

| factor ↓ | Temps d'assoliment (s) | | |
|------------|--|---------|---------|
| | <i>menys realista</i> ← [model motors] → <i>més realista</i> | | |
| | Model_1 | Model_2 | Model_3 |
| 3 | 0.2 | 0.6 | 1.0 |
| 2 | 0.4 | 0.65 | 1.15 |
| 1 | 0.9 | 0.95 | 1.6 |
| 0.5 | 1.95 | 2.45 | 3.25 |

Taula 5.2.2.3: Valors del temps d'assoliment per a diferents models del motor.

Es pot comprovar fàcilment com el temps d'assoliment augmenta a mida que el model del motor és més realista, és a dir, la dinàmica dels motors imposa restriccions en l'execució dels moviments del robot mòbil. Per tant, les dinàmiques dels actuadors del robot condicionen les característiques temporals del moviment del robot.

5.2.2.4. "Moviment amb posició i orientació final precises "

Suposem que volem controlar el moviment del robot de forma que vagi seguint l'eix horitzontal $(x_r, 0)$ amb una velocitat lineal constant v , segons es mostra al següent gràfic:

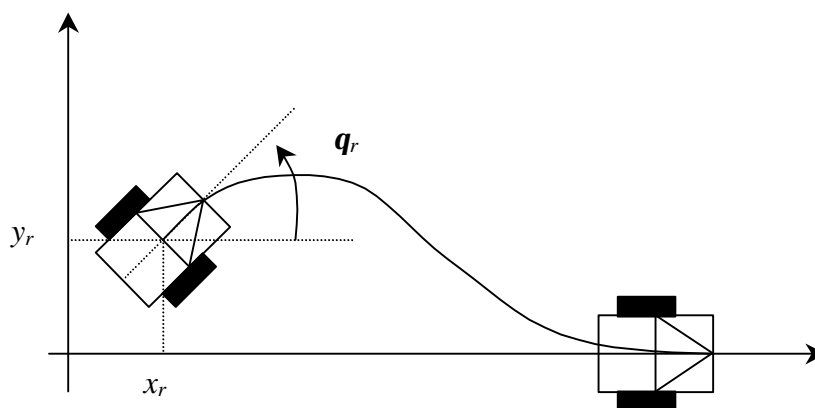


Figura 5.2.2.4.1: Trajectòria genèrica desitjada pel controlador "Moviment amb posició i orientació final precises"

Per fer aquest control cal trobar quina ha de ser la velocitat angular en funció de la posició del robot per tal que el moviment sigui el desitjat, és a dir, seguint l'eix horitzontal. Per tant, ens cal una llei de control $\omega(y_r, \theta_r)$ que a cada instant de temps faci que la terna de valors (y_r, θ_r) tendeixi a $(0,0)$. Aquesta llei té la següent expressió (veure apèndix IV):

$$\omega = -\left(\frac{\alpha_1 \alpha_2}{v}\right) \cdot y_r - (\alpha_1 + \alpha_2) \cdot \theta_r$$

Amb aquesta llei de control la funció de transferència de θ_r és la següent:

$$\theta_r(s) = \frac{\theta_r(0) \cdot (s + \alpha_1 \alpha_2) + \dot{\theta}_r(0)}{s^2 + (\alpha_1 + \alpha_2)s + \alpha_1 \alpha_2}$$

Es pot observar que els paràmetres α_1 i α_2 són els pols de θ_r i, per tant, indiquen que el comportament del sistema correspon a un sistema de segon ordre, és a dir:

$$\begin{cases} \zeta = \frac{1}{2} \left(\frac{1}{\alpha_1} + \frac{1}{\alpha_2} \right) \sqrt{\alpha_1 \alpha_2} \\ \omega_n^2 = \alpha_1 \alpha_2 \end{cases} \quad (5.1)$$

El paràmetre ζ s'anomena **coeficient d'esmoreïment** i el paràmetre ω_n s'anomena **frequència natural**. A partir d'aquests paràmetres podem calcular els valors dels paràmetres S_P i T_S amb les següents expressions:

$$S_P = q_r(0) \cdot \exp\left(-\frac{\pi \zeta}{\sqrt{1 - \zeta^2}}\right) \quad T_S \cong \frac{4}{\zeta \omega_n} \quad (5.2)$$

Per tant, els paràmetres de la llei de control α_1 i α_2 es poden escollir de forma apropiada per tal que el comportament de segon ordre de la variable θ_r sigui el desitjat, és a dir, podem imposar uns valors màxims per a T_S i S_P i calcular quins són els valors adients dels paràmetres de la llei de control α_1 i α_2 .

A continuació es mostren diversos resultats en relació a aquesta llei de control de posició del robot mòbil. En concret la figura 5.2.2.4.2 mostra les corbes de la resposta temporal de l'orientació relativa θ_r per a diferents valors de ζ i ω_n , és a dir, dels pols α_1 i α_2 . Aquestes corbes s'han obtingut a partir de la funció de transferència $\theta_r(s)$ i suposant $\dot{\theta}_r(0) = 0$ rad/s i $\theta_r(0) = 90^\circ$. Pot observar-se com l'augment del coeficient d'esmoreïment fa disminuir l'amplitud de l'oscil·lació màxima (S_P) i fa disminuir el temps d'establiment (T_S),

tal com era d'esperar. També es pot observar com l'augment del coeficient d'esmoreïment per sobre de la unitat fa augmentar el temps d'establiment.

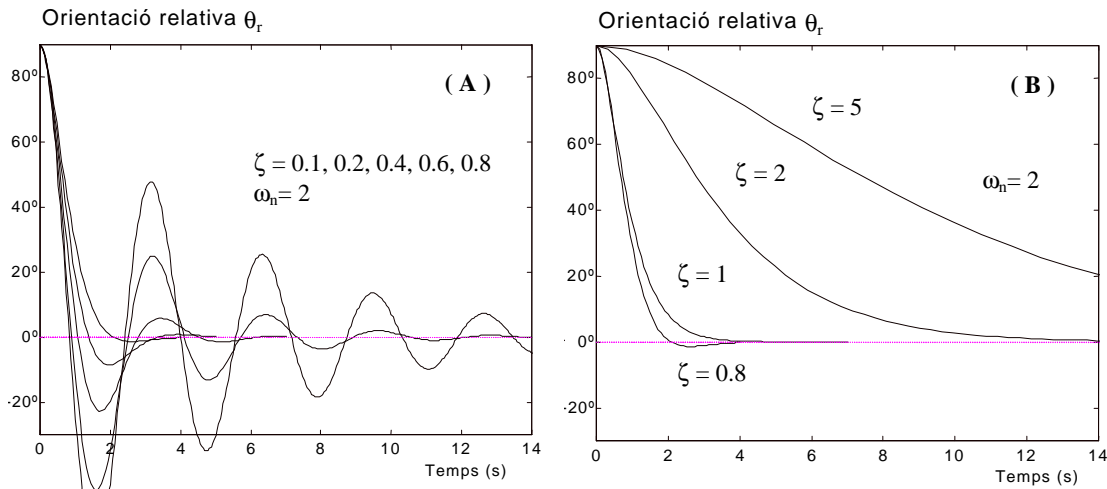


Figura 5.2.2.4.2: Resposta temporal de l'orientació relativa: (A) $\mathbf{q}_r(0)=90^\circ$, (B) $\mathbf{q}_r(0)=90^\circ$.

D'aquests resultats podem deduir que aquest controlador permet portar el robot des d'un estat inicial $(0, y, \theta)$ a un altre final $(x, 0, 0)$ en un temps T_S que es calcula a partir de l'expressió següent (combinant les equacions (5.1) i (5.2)):

$$T_S = \frac{4}{\zeta\omega_n} = \frac{8}{(\alpha_1 + \alpha_2)} \quad (5.3)$$

A més, tenint en compte que les dinàmiques de θ_r i de y_r són equivalents (veure apèndix IV) també és possible calcular el valor de S_p de la magnitud y_r , és a dir, el valor de la desviació màxima respecte de l'eix horitzontal ($S_p(y_r)$). Per tant, aquest controlador és adient per a l'acció atòmica "*Moviment amb posició i orientació final precises*" donat que garanteix una bona orientació final i un compliment de restriccions temporals i geomètriques de la trajectòria [Oller, 02]. Finalment, el controlador \mathbf{C}_2 queda definit de la següent manera:

$\mathbf{IN}_2 : \{v, y_r, \theta_r\}$

$\mathbf{IC}_2 : \{(0, y, \mathbf{q})_{robot}\}$

$\mathbf{FC}_2 : \{(x, 0, 0)_{objectiu}, t \leq T_S\}$

$\mathbf{AA}_2 : \{\text{motor dret, motor esquerra}\}$

$\mathbf{CL}_2 : \{ \omega = - \left(\frac{\alpha_1 \alpha_2}{v} \right) \cdot y_r - (\alpha_1 + \alpha_2) \cdot \theta_r, < \alpha_1, \alpha_2 > \}$

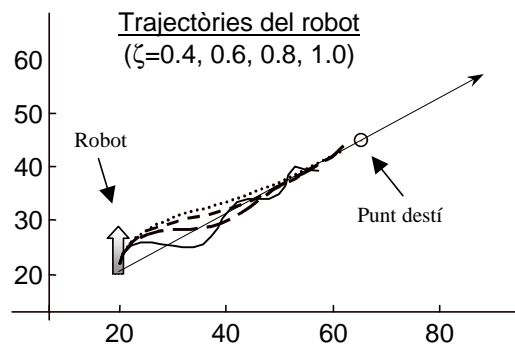
$\mathbf{LB}_2 : \text{"Moviment amb posició i orientació final precises"}$

5.2.2.5. Capacitats atòmiques

Hem vist que les qualitats que podem exigir al controlador C_2 corresponen a les restriccions temporals i geomètriques de la trajectòria. Donat que es disposa d'un controlador lineal és possible definir analíticament altres característiques com és el temps del desviament màxim de y_r ($S_p(y_r)$) o l'esforç de control. Sigui quin sigui el conjunt de característiques que es vulgui incloure al conjunt de capacitats J_2 , amb aquest controlador no caldrà construir una taula de capacitats donat que es poden calcular a partir d'expressions analítiques. Suposarem que les qualitats interessants d'aquest controlador són les següents: $J_2 = \langle T_s, S_p(\theta_r), S_p(y_r) \rangle$. Totes les tres estan relacionades directament amb els valors dels pols a través de les equacions 5.2, 5.3 i la següent:

$$S_p(y_r) = y_r(0) \cdot \exp\left(-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right) \quad (5.4)$$

Cal fet notar que tot i que sempre és possible exigir al controlador C_2 el compliment d'alguna de les qualitats, en molts casos no és possible el compliment simultani de les tres restriccions donat que la restricció temporal va en detriment de les restriccions geomètriques de la trajectòria.



Finalment, mostrem d'aquest controlador algunes capacitats d'anar a un punt precís. En concret mostrem els casos $\alpha_{1,2} = -0.8 \pm 1.833j$ ($\omega_n=2$, $\zeta=0.4$) i $\alpha_{1,2} = -1$ ($\omega_n=2$, $\zeta=1$):

CAN^t_A {"Moviment amb posició i orientació final precises",

$$\langle (0, 1, 90^\circ), (x, 0, 0) \rangle, \langle 5s, 19.8^\circ, 0.22m \rangle$$

CAN^t_A {" Moviment amb posició i orientació final precises",

$$\langle (0, 1, 90^\circ), (x, 0, 0) \rangle, \langle 4s, 0^\circ, 0m \rangle$$

5.2.3. Especificacions de les tasques: capacitats bàsiques

En aquesta secció es construeixen les taules de capacitats d'algunes tasques del mòdul supervisor, és a dir, són capacitats bàsiques que es construeixen a partir de les atòmiques i introduint-hi informació simbòlica. Les tasques que poden definir-se són les següents⁵:

| | |
|--|--|
| LB₁ : "Anar cap a la pilota" | LB₇ : "Interceptar la pilota" |
| LB₂ : "Anar cap a la porteria local" | LB₈ : "Marcar un contrari" |
| LB₃ : "Anar cap a la porteria contrària" | LB₉ : "Desmarcar-se d'un contrari" |
| LB₄ : "Evitar xocs" | LB₁₀ : "Portar la pilota" |
| LB₅ : "Xutar la pilota cap a porteria" | LB₁₁ : "Evitar el xoc amb la pilota" |
| LB₆ : "Xutar la pilota cap a un punt" | LB₁₂ : "Seguir la paret" |

Algunes d'aquestes tasques necessiten una estructura de control específica mentre que d'altres funcionen amb la mateixa. Així, les tasques 1, 2, 3 i 4 poden funcionar amb l'estructura de control equivalent a "Anar de *A* a *B* seguint una trajectòria lliure de col·lisió", mentre que les tasques 5, 6 i 7 funcionen igual que l'equivalent a "Anar a un punt amb direcció final específica". El que canvia en els diversos casos és l'etiquetat del punt final, és a dir, 'pilota', 'contrari', 'paret', 'porteria contrària', 'porteria local' i 'punt'. A les seccions següents es comenten en detall les estructures de control de les tasques **LB₅** i **LB₁**.

5.2.3.1. "Xutar la pilota cap a porteria"

Aquesta tasca s'ha de realitzar amb un supervisor estructurat de forma molt semblant a la dels supervisors d'altres tasques variants d'aquesta com són "Xutar la pilota cap a un punt" o "Interceptar la pilota": en tots aquests casos és molt important controlar l'orientació del robot quan s'arriba al punt final. En aquest apartat s'explicarà quin mètode s'ha seguit per calcular les capacitats bàsiques per realitzar la tasca "Xutar la pilota cap a porteria".

Les dues condicions per obtenir un xut de la pilota adient són estar al darrera de la pilota i estar correctament orientat per poder empènyer la pilota cap a la porteria. La màquina d'estats que s'ha construït per realitzar aquesta tasca s'encarrega d'anar assignant

⁵ Aquest conjunt de tasques és il·lustratiu i completament arbitrari. De tota manera resulta suficient per al funcionament del sistema.

consignes de posició a mida que el robot va desplaçant-se a través de les 8 zones definides en el següent gràfic:

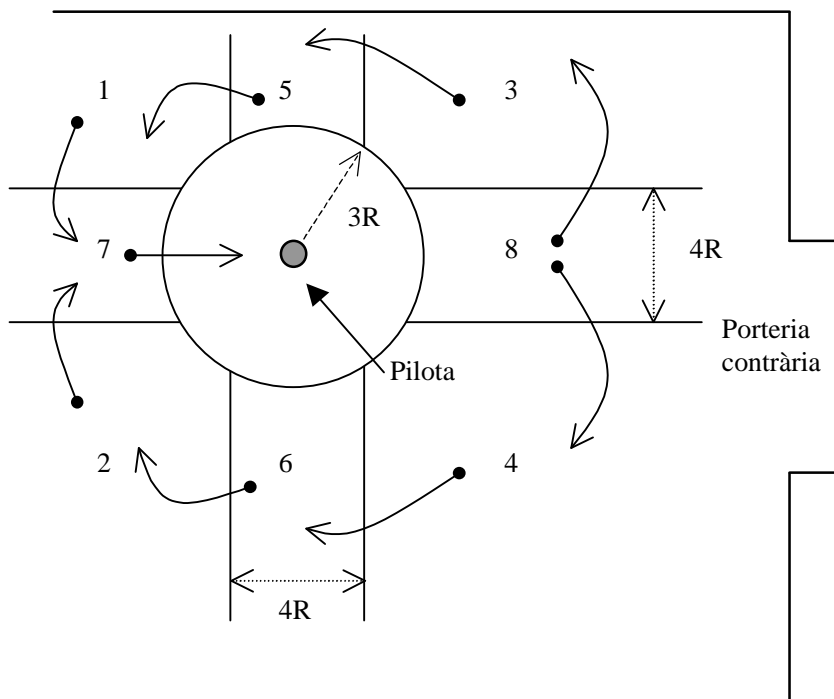
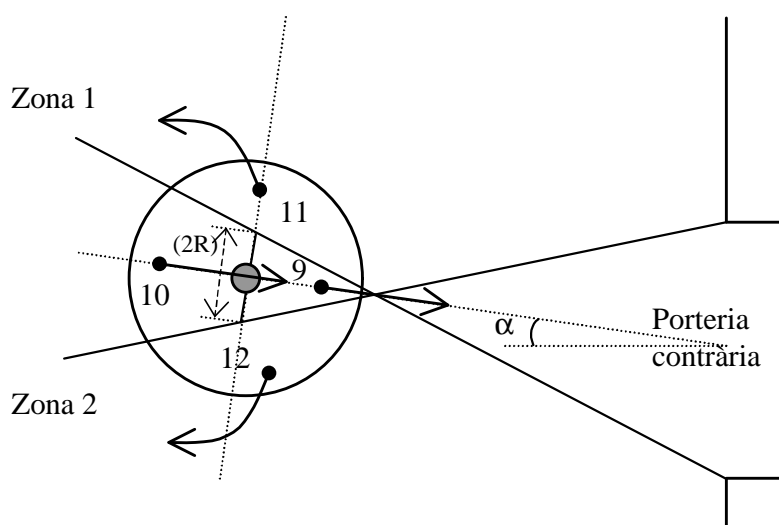


Figura 5.2.3.1.1: Esquema que il·lustra la variació de les consignes de posició.

Aquesta màquina d'estats s'encarrega de portar el robot cap al darrera de la pilota evitant al mateix temps que es produeixi una xut en direcció a la porteria local. Per complir amb la condició de l'orientació correcta, la zona circular més propera a la pilota ($<3R$) consta de quatre zones més:



A continuació es mostra una part del codi associat a la màquina d'estats finits. En

concret es veuen quines són les consignes de posició assignades al robot quan es troba a la zona 'zona' i quin és el controlador que s'activa en cada cas:

```

zona=1:
x=xb-2*D;
y=yb;
ctrl=c1;

zona=2:
x=xb-2*D;
y=yb;
ctrl=c1;

zona=3:
x=xb;
y=(3*ya+yb)/4;
if y<(yb+2*D)
    y=yb+2*D;
ctrl=c2;

zona=4:
x=xb;
y=(3*ya+yb)/4;
if y>(yb-2*D)
    y=yb-2*D;
ctrl=c2;

zona=5
x=(3*xa+xb)/4;
y=yb+(D-R);
if x>(xb-2*D)
    x=xb-2*D;
ctrl=c1;

zona=6:
x=(3*xa+xb)/4;
y=yb-(D-R);
if x>(xb-2*D)
    x=xb-2*D;
ctrl=c1;

zona=7:
x=xb-inc_x;
if yb<=45
    y=yb-abs(inc_y);
else
    y=yb+abs(inc_y);
ctrl=c1;

zona=8:
x=xb;
if ya>=yb
    y=yb+(D+R);
else
    y=yb-(D+R);
ctrl=c2;

zona=9:
x=xb+inc_x;
y=(xb+inc_x-xb)*(yb-45)/(xb-130)+yb;
ctrl=c1;

zona=10:
if xa>xb
    x=xb-abs(inc_x);
    if (yb>45 & ya>yb)|(yb<45 & ya>yb)
        y=yb+D;
    elseif (yb<45 & ya<yb)|(yb>45 & ya<yb)
        y=yb-D;
else
    x=xb-abs(inc_x);
    y=yb;
ctrl=c3;

zona=11:
xa=xb;
ya=yb-D;
ctrl=c1;

zona=12:
xa=xb;
ya=yb+D;
ctrl=c1;

```

Definició de variables:

(x,y) → Nova consigna de posició.
(xa,ya) → Coordenades del robot mòbil.
(xb,yb) → Coordenades de la pilota.
(R) → Amplada frontal del mòbil.
(D=2*R)
(inc_x,inc_y)= D*(cos α ,sin α)
(ctrl) → Flag selector del controlador.

S'han definit cinc localitzacions de test (**A**, **B**, **C**, **D** i **E**) representatives de moltes situacions possibles: aquestes posicions són independents de l'angle α . Per a cada posició també s'han definit diferents estats inicials de l'orientació del robot. La següent figura mostra quines són aquestes cinc posicions de test i llurs diferents angles inicials:

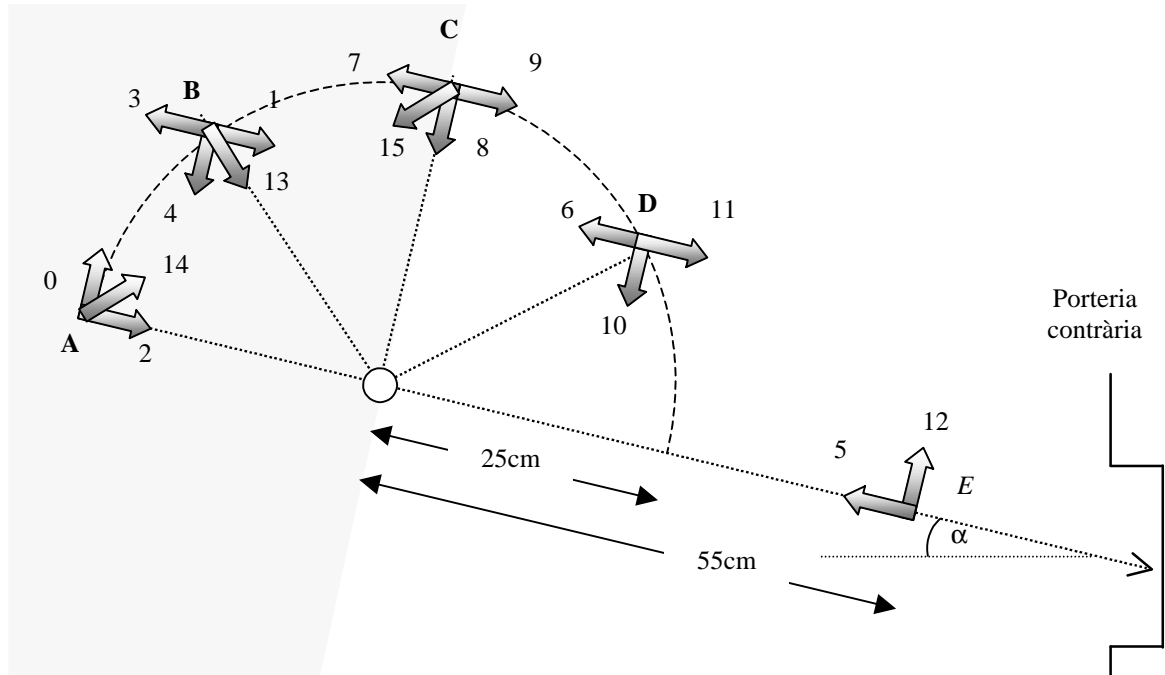


Fig. 5.2.3.1.2: Esquema de les cinc posicions de test per al xut de la pilota cap a la porteria

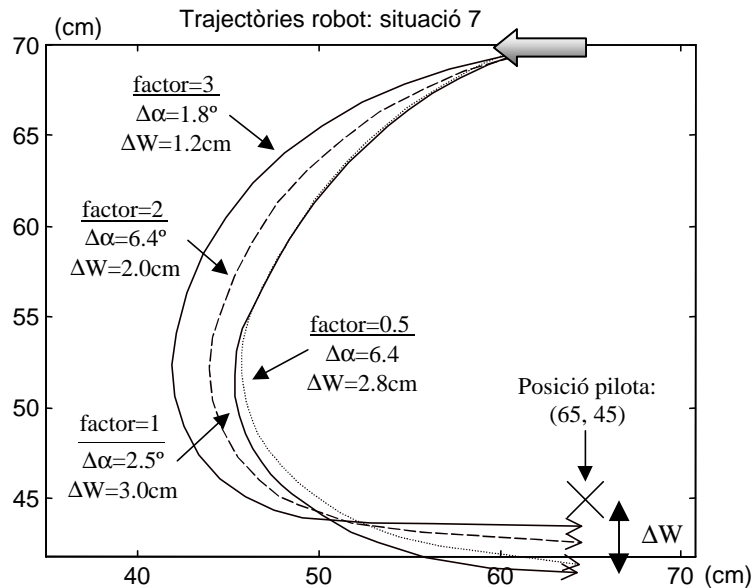
Per l'obtenció de la taula de capacitats s'ha tingut en compte la mesura dels següents paràmetres de la resposta del sistema: el temps d'arribada a la pilota (T_{xut} , en segons), l'error d'orientació del robot en el moment del xut ($\Delta\alpha$, en graus) i l'error frontal del xoc entre el robot i la pilota (ΔW , en centímetres). Cal observar que les posicions estan definides en relació a la posició relativa entre la pilota i la porteria contrària. La taula sencera de capacitats es mostra a la secció 5.2.3.2. i a continuació es mostra la taula de capacitats corresponents a la configuració 10 (situació **D**):

| factor=0.5 | | | factor=1 | | | factor=2 | | | factor=3 | | |
|------------|-----|-----|----------|-----|-----|----------|-----|-----|----------|-----|-----|
| T_xut | Da | DW | T_xut | Da | DW | T_xut | Da | DW | T_xut | Da | DW |
| 4.35 | 2.8 | 4.8 | 2.3 | 2.5 | 4.7 | 1.7 | 0.4 | 5.2 | 1.7 | 7.2 | 4.9 |

Per aquest estat inicial es pot veure com l'execució del moviment té moltes imperfeccions a causa de les restriccions no holònomes del mòbil. Apareix un error frontal

de xoc ΔW de l'ordre de 5cm, és a dir, la maniobra del robot no és adient per poder empènyer la pilota endavant ja que l'amplada frontal del robot és de 7.5cm. Per tant, la configuració 10 no és apropiada per executar la tasca de "Xutar la pilota cap a la porteria": el robot no tindrà capacitat per a fer-ho.

A continuació es mostra el gràfic de les trajectòries per a la configuració 7 (situació C). Cada trajectòria correspon a un valor diferent del paràmetre de control 'factor':



S'observa en aquest cas que els errors frontals del xut són inferiors a 3cm, és a dir, en tots els casos és possible realitzar el xut. Ara bé, en cada cas també apareix un error d'orientació en el moment del xut. Aquest error angular serà més o menys important en funció de quina sigui la distància de la pilota a la porteria: errors grans a petites distàncies o errors petits a grans distàncies faran que la pilota pugui entrar a porteria, mentre que en altres casos no serà així. Aquest fet s'haurà de tenir en compte en el càlcul de les capacitats simbòliques de rols que incloguin la tasca "Xutar la pilota cap a la porteria".

Per acabar, aquest supervisor S_5 queda especificat de la següent manera:

$IN_5 : \{(x,y)_{\text{robot}}, (x,y)_{\text{pilota}}, \alpha\}$

$IC_5 : \{<INICI: (x,y)_{\text{punt inicial}}, t = t_0\}$

$FC_5 : \{<FI: (x,y, \alpha)_{\text{punt final}}, <'zona' = 9 >\}$

$AA_5 : <C_1, C_2>$

$CL_5 : \text{Màquines d'estats de les figures 5.2.3.1.1. i 5.2.3.1.2}$

$LB_5 : \text{"Xutar la pilota cap a la porteria"}$.

Quant a les tasques **LB₆** i **LB₇**, cal comentar que llurs estructures de control es poden obtenir a partir de modificacions de l'estructura que s'ha presentat en aquesta secció. En el primer cas, l'estructura hauria de tenir en compte que la velocitat en el moment del xut és una qualitat molt important a tenir en compte, i més si es vol considerar l'efecte de la dinàmica del moviment lliure de la pilota. En el segon cas s'entén que la intercepció és una acció defensiva, o sigui, que la direcció del xut de la pilota no ha de ser tant precisa: només cal que sigui en direcció al camp contrari o en oposició a la porteria local.

5.2.3.2. Capacitats bàsiques

A continuació es mostra la taula de capacitats per al conjunt dels 16 estats inicials diferents que es mostren a la figura 5.2.3.1.2. Aquesta taula conté els valors de T_{xut} , $\Delta\alpha$ i ΔW per a cadascuna de les situacions i per a diferents valors del paràmetre 'factor'.

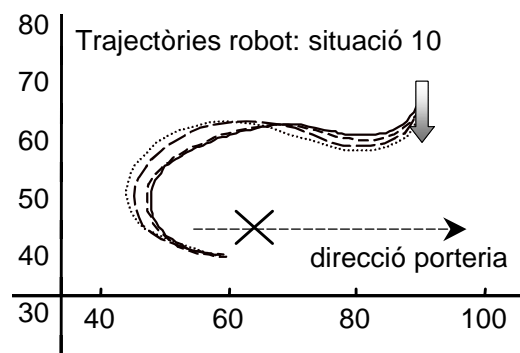
| estat inicial | factor=0.5 | | | factor=1 | | | factor=2 | | | factor=3 | | |
|---------------|------------|------|------------|----------|------|------------|----------|------|------------|----------|------|------------|
| | T_xut | Da | DW | T_xut | Da | DW | T_xut | Da | DW | T_xut | Da | DW |
| 0 | 2.45 | 13.4 | ≈0 | 1.25 | 11.7 | ≈0 | 0.8 | 17.9 | ≈0 | 0.7 | 23.6 | ≈0 |
| 1 | 2.2 | 13.4 | ≈0 | 1.2 | 6.8 | ≈0 | 0.95 | 8.6 | ≈0 | 0.85 | 24.4 | ≈0 |
| 2 | 2.35 | 0 | ≈0 | 1.1 | 0 | ≈0 | 0.55 | 0 | ≈0 | 0.4 | 0 | ≈0 |
| 3 | 3.6 | 3.6 | ≈0 | 1.8 | 3.5 | ≈0 | 1.15 | 1.9 | ≈0 | 1.1 | -5.7 | ≈0 |
| 4 | 3.0 | 7.6 | ≈0 | 1.45 | 7.3 | ≈0 | 0.75 | 10.0 | ≈0 | 0.65 | 10.8 | ≈0 |
| 5 | 5.35 | 2.9 | ≈0 | 2.7 | -0.1 | × | 1.85 | -3.2 | × | 1.85 | 0.3 | × |
| 6 | 4.35 | 2.8 | × | 2.2 | 0.2 | × | 1.45 | -2.0 | × | 1.35 | 8.1 | × |
| 7 | 3.15 | 6.4 | 2.8 | 1.6 | 2.5 | 3.0 | 1.0 | 6.4 | 2.0 | 1.05 | 1.8 | 1.2 |
| 8 | 2.2 | 17.3 | 3.9 | 1.1 | 16.5 | 3.8 | 0.8 | 23.7 | 4.0 | 0.85 | 0.6 | 7.2 |
| 9 | × | | | × | | | × | | | × | | |
| 10 | 4.35 | 2.8 | 4.8 | 2.3 | 2.5 | 4.7 | 1.7 | 0.4 | 5.2 | 1.7 | 7.2 | 4.9 |
| 11 | 4.2 | 18.6 | 5.7 | 2.6 | 3.2 | 7.3 | 2.2 | 13.5 | 5.9 | 1.95 | 6.2 | 4.6 |
| 12 | 5.6 | 4.3 | 4.2 | 3.05 | 2.6 | 4.3 | 3.15 | 5.7 | × | 5.25 | 3.5 | × |
| 13 | 2.75 | 11.0 | ≈0 | 2.65 | 3.2 | 7.3 | 2.2 | 13.5 | 5.9 | 1.95 | 6.2 | 4.6 |
| 14 | 2.4 | 6.1 | ≈0 | 1.15 | 7.0 | ≈0 | 0.65 | 8.4 | ≈0 | 0.55 | 22.6 | ≈0 |
| 15 | 2.65 | 5.1 | 4.2 | 1.35 | 1.7 | 4.8 | 1.0 | 7.5 | 6.8 | 1.0 | 18.0 | 7.9 |

Taula 5.2.3.2: Taula de capacitats J_5^k corresponents a la tasca "Xutar la pilota cap a porteria".
 Símbols: '≈0' indica que l'error és inferior a 1cm, i '×' indica que la maniobra no és possible.

S'observa que la maniobra corresponent a la situació número 9 no es pot completar amb èxit. El que passa en realitat és que durant la maniobra el robot colpeja la pilota des de la zona 11 i el xut es produeix en direcció a la porteria local. En aquest cas es considera que el robot no té capacitat per xutar la pilota correctament donat que colpeja la pilota des d'una situació errònia. En altres casos (valors en color gris) la maniobra no es pot completar no perquè presenti aquest problema sinó perquè el valor de ΔW és massa gran com per a que el robot pugui colpejar la pilota: s'ha considerat que un valor $\Delta W \geq 4$ no assegura que el robot colpegi la pilota endavant. Als demés casos la maniobra és realitzable i pot observar-se que corresponen a les posicions de la zona puntejada en gris a la figura 5.2.3.1.2. Per tant, considerarem que el robot té capacitat de xutar la pilota cap a porteria quan es troba a les posicions **A**, **B** i **C**.

5.2.3.3. Comentaris

A continuació es mostra el gràfic de les trajectòries corresponents a la situació 10. A través de la geometria de les trajectòries pot observar-se que a només 25cm de la pilota el robot ha de fer una maniobra molt forçada per poder xutar bé la pilota.

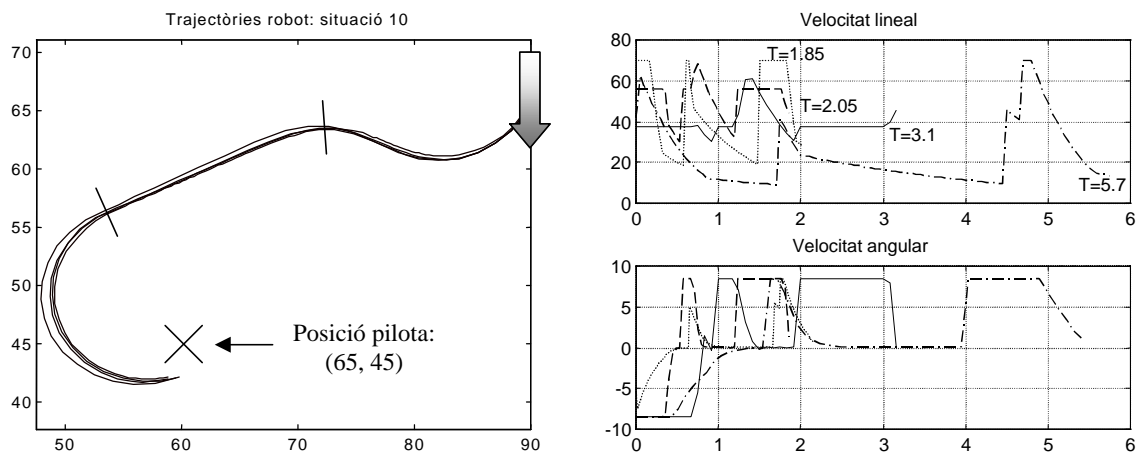


En tots els casos el valor de ΔW és superior a 4 però es pot reduir si es canvien els altres paràmetres del controlador. Per exemple, continuant amb la situació 10, es pot parametritzar el controlador de la manera següent:

$KPV = 1/2$ → constant adimensional (abans era 1)

$KPA = 2$ → constant dimensional (abans era $1(\text{cm/s})/\text{rad}$)

Amb aquests canvis s'observa que els temps de xut es mantenen però els valors de ΔW es redueixen sensiblement en tots els casos. Observant la trajectòria s'observa que té tres trams: dos arcs i una recta. En els trams en forma d'arc la velocitat angular està saturada, és a dir, el controlador dóna el valor màxim de velocitat a un dels motors. Això passa per als valors 0.5, 1, 2 i 3 per al paràmetre de control 'factor' i és per aquest motiu que les quatre trajectòries són molt coincidents:



En tots aquests casos el control entra en una dinàmica saturada i possiblement el robot real patinaria⁶, és a dir, es violarien les limitacions dinàmiques comentades a l'apèndix II. Aquesta dinàmica saturada també s'observa en altres casos semblants al que presenta la situació 10 i, es pot concloure que en general la variació dels paràmetres de control K_{PV} i K_{PA} no és aconsellable si l'objectiu és disminuir el valor de l'error frontal.

Respecte a l'efectivitat de la màquina d'estats que s'ha utilitzat, cal comentar que possiblement els resultats són millorables però, en tot cas, la influència de la dinàmica sempre és important. A la figura 5.2.3.2 es mostren diversos gràfics on es comparen les respostes del sistema per a la situació 1 considerant un model pels motors més ideal:

Model_2: Motor amb acceleracions infinites i velocitat màxima de 70cm/s.

Model_3: Motor amb dinàmica de 1^r ordre i velocitat màxima de 70cm/s.

⁶ Les dades de la simulació no modelen el fet que el robot pugui patinar.

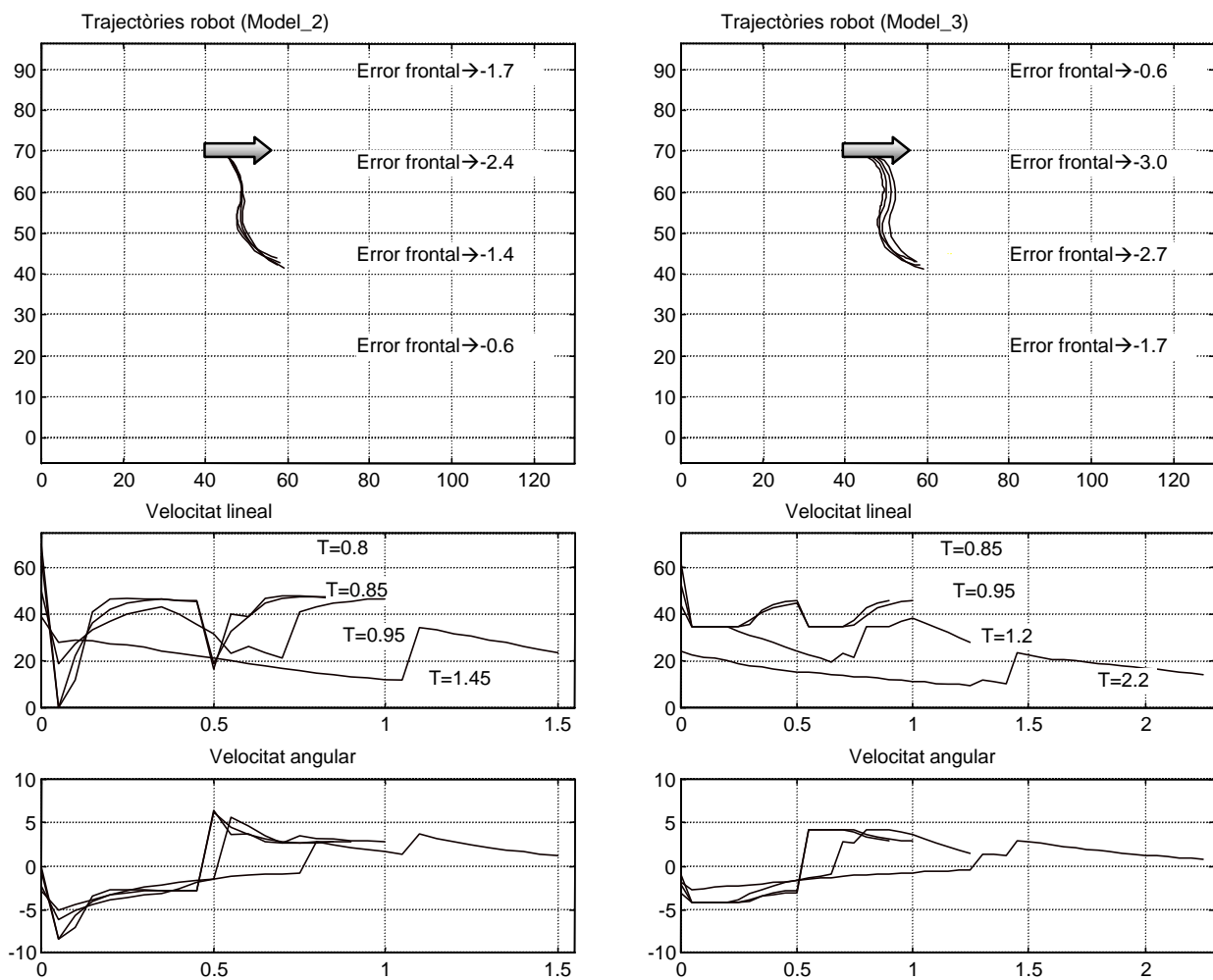


Figura 5.2.3.2: Comparació de les respostes del robot per a la situació de xut n° 1. A l'esquerra amb el Model_2, a la dreta amb el Model_3. Als gràfics de la fila superior es mostren els valors de l'error frontal de xoc: de dalt a baix els valors corresponen a valors de factor = 3, 2, 1 i 0.5. Ídem a la segona fila.

S'observa que s'obtenen resultats sensiblement diferents per a T_{xut} i per a ΔW . Si es comparen els dos models s'observa que el model més realista (a la dreta de la figura 5.2.3.2) presenta valors de T_{xut} i de ΔW superiors en tots els casos. Per tant, les limitacions dinàmiques dels motors tenen els seus efectes a les capacitats bàsiques d'aquesta tasca.

5.2.3.4. "Anar cap a la pilota"

L'objectiu d'aquesta tasca és fer que el robot vagi fins a una distància de 25cm. de la pilota, o millor dit, en algun punt damunt del cercle de radi 25cm centrat a la posició que ocupa la pilota i amb una orientació relativa a la pilota de 90° (tangent al cercle). El

supervisor utilitzarà el controlador C_2 ja analitzat (LB_2 "Moviment amb posició i orientació final precises") i haurà de calcular quins són els paràmetres del controlador que fan que el temps d'establiment sigui mínim.

A partir de l'anàlisi feta d'aquest controlador es pot deduir que el controlador que fa que aquest temps sigui mínim és aquell que té els dos pols idèntics ($\alpha_1=\alpha_2$) donat que es compleix que $\zeta=1$. Així, a partir de l'equació 5.3 obtenim la següent expressió:

$$T_s = \frac{4}{\zeta\omega_n} = \frac{4}{\omega_n} = \frac{4}{\frac{1}{2}(\alpha_1 + \alpha_2)} \Bigg|_{\zeta=1, \alpha_1=\alpha_2} = \frac{4}{\alpha_1} \quad (5.5)$$

A més, a causa de la tracció mecànica diferencial del robot i de la limitació de les velocitats de les rodes (70cm/s) obtenim la següent limitació del valor de la velocitat angular:

$$\omega_{max} = \max\{(V_D - V_E) / R\} = V_{MAX} / R \cong 4.5 \text{ rad / s}$$

Aleshores, si el robot té una velocitat lineal constant V i ha d'anar a una coordenada relativa X_R en un temps T_A obtenim que s'ha de complir que $T_s \leq T_A$, és a dir, $X_R \geq V \cdot T_s$. Aquest valor mínim de X_R és el que, inicialment, permet decidir si la tasca es pot executar però el supervisor també ha d'avaluar si $\omega(0) < \omega_{max}$, és a dir, ha d'avaluar si es pot produir una saturació del controlador. Per veure el significat d'aquesta restricció es mostren els següents gràfics:

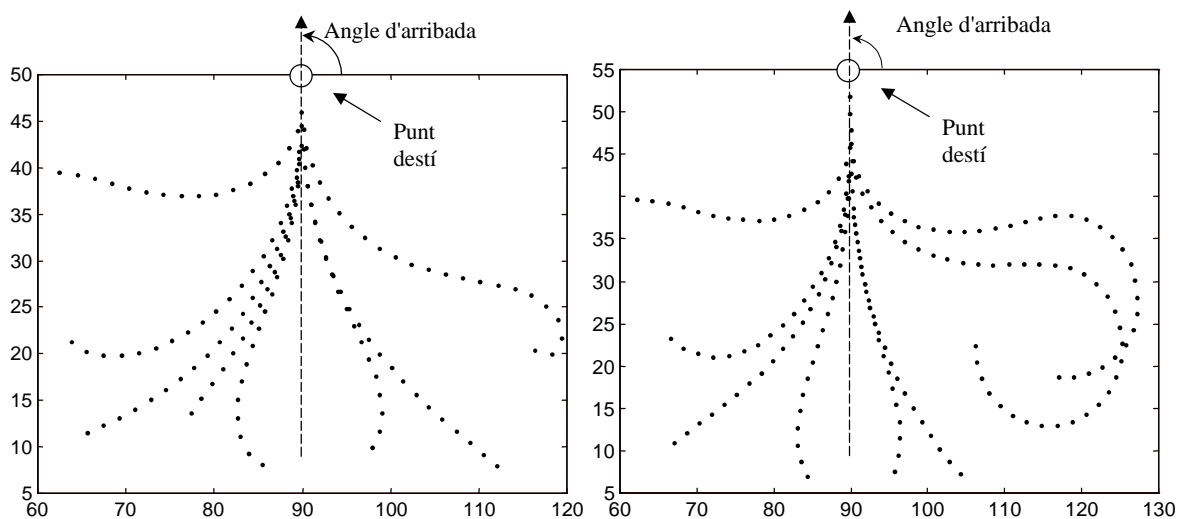


Figura 5.2.3.4: Trajectòries per a diferents posicions inicials del robot. L'angle d'arribada al punt de destí és de 90° . Esquerra: sense limitacions per a w . Dreta: $V=70\text{cm/s}$, $w_{max}=4.5\text{rad/s}$.

D'aquests gràfics es pot veure que si no es tenen en compte les limitacions esmentades els càlculs teòrics serien erronis donat que en realitat el que passaria seria que el robot patinaria. A l'esquerra es pot observar que les trajectòries tenen arcs molt acusats (ω elevada) quan en realitat les trajectòries han de tenir arcs més suaus com a la figura de la dreta. L'important d'aquesta qüestió és que el supervisor ha de garantir que el controlador no arribi a dinàmiques saturades donat que, en aquest cas, els càlculs teòrics no es correspondrien amb la realitat: els controladors perdrien la coherència.

Quant a les capacitats J_1^a d'aquest supervisor comentem que ara tenim que sempre es compleix $\zeta=1$ i que a partir de la condició $T_S \leq T_A$ i l'equació 5.5 es pot calcular fàcilment un valor mínim per α . Així, el supervisor S_1 queda així:

| |
|--|
| $IN_1 : \{(x,y)_{\text{relativa robot-pilota}}, T_S \leq T_A\}$ |
| $IC_1 : \{<INICI: (x,y)_{\text{punt inicial}}>, t = t_0\}$ |
| $FC_1 : \{<FI: <\omega \equiv 0>\}$ |
| $AA_1 : <C_2>$ |
| $CL_1 : \{\omega(0) < \omega_{\text{max}} \text{ .AND. } X_R \geq V \cdot T_S\}$ |
| $LB_1 : \text{"Anar cap a la pilota"}$ |

5.2.4. Especificacions dels rols: capacitats simbòliques

Les accions a nivell agent són rols que poden incloure moviments coordinats o cooperatius amb altres agents que s'han de negociar almenys entre dos, i l'eficiència amb què l'agent es compromet a executar el seu rol es calcula a partir de les capacitats simbòliques. La negociació no s'estableix aquest nivell sinó a nivell d'escena.

Per aquest sistema robòtic podem definir d'entre altres, els següents rols:

- ✓ Protegir porteria → Acció defensiva
- ✓ Interceptar → Acció defensiva
- ✓ Passar la pilota → Acció ofensiva/defensiva
- ✓ **Rebre la pilota** → Acció ofensiva/defensiva
- ✓ Demanar la pilota → Acció ofensiva
- ✓ Obrir joc → Acció ofensiva

Les característiques de cada rol són descriptibles de forma simbòlica i, en general, cal un mètode de disseny complex per saber com s'ha d'executar el rol donat l'alt grau de dinamisme del sistema. En tot cas, un mètode de disseny basat en casos és suficient per portar a bon terme el disseny de forma satisfactòria. En aquesta secció s'estudiarà en detall el rol corresponent a "Rebre la pilota" en el cas d'una escena ofensiva. L'objectiu final serà que el robot receptor de la pilota xuti per fer gol de forma correcta, és a dir, avaluant prèviament l'efectivitat del seu xut. L'anàlisi d'aquest rol s'ha fet tenint en compte les següents consideracions:

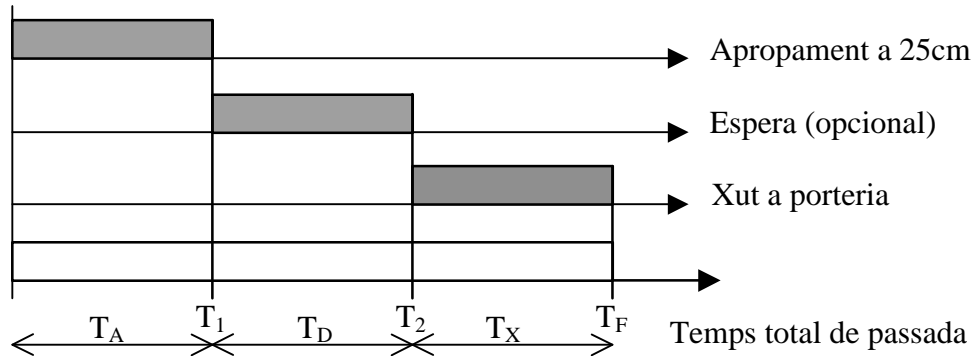
- ✓ La passada és una acció coordinada i cooperativa entre dos robots: el passador i el receptor.
- ✓ El passador fa un xut en una direcció i velocitat determinades.
- ✓ El receptor no es queda la pilota sinó que la xuta.
- ✓ L'objectiu de la passada és que el receptor pugui xutar la pilota cap a porteria contrària.
- ✓ El robot passador fa el xut de forma completament controlada.
- ✓ El receptor ha de maniobrar de forma que colpegi la pilota després de T_F segons d'haver-ho fet el passador.
- ✓ La pilota té una dinàmica de primer ordre.
- ✓ La simulació de la dinàmica del robot receptor té un model realista dels motors.

5.2.4.1. 'Rebre la pilota'

S'ha resolt que la maniobra del receptor consti de dos moviments: un primer moviment d'aproximació a una distància de 25 cm de la pilota, i un segon moviment de xut de la pilota cap a la porteria. Aquests dos moviments consisteixen en ser tasques que ja s'han analitzat anteriorment: el primer moviment es realitza amb la tasca "Anar cap a pilota" mentre que el segon es realitza amb la tasca "Xutar la pilota cap a porteria". Per tant, l'avaluació de les capacitats de xut per part del receptor haurà de tenir en compte les capacitats bàsiques ja estudiades en seccions anteriors.

El següent cronograma mostra un esquema temporal de quines són les maniobres que s'han definit per tal que el receptor pugui colpejar la pilota cap a la porteria contrària. El

cronograma mostra que hi ha una maniobra opcional d'espera que només està definida per aquells casos en què les altres dues maniobres no puguin ser més lentes, com ja es veurà en un exemple. A continuació s'explica com són aquestes dues maniobres.



5.2.4.2. Maniobres d'apropament i de xut a porteria

Hi ha dues coses que cal tenir en compte en especificar la maniobra d'apropament. En primer lloc, s'ha considerat que la maniobra de xut a porteria s'ha de començar a uns 25cm de la pilota. La raó d'aquesta elecció està en el fet que en els casos reals aquestes maniobres queden sovint bloquejades per la presència d'altres robots de l'equip contrari. Per tant, s'ha escollit una distància relativament petita. Així, la maniobra d'apropament consistirà en fer que el robot redueixi, si és el cas, la seva distància a la pilota. L'altre punt a tenir en compte és que aquesta maniobra també s'ha d'aprofitar per col·locar el robot en una posició favorable per iniciar posteriorment la maniobra del xut. Aquesta col·locació fa referència a la posició del robot relativa a la pilota, i no a la orientació del robot al final de la maniobra. De fet, la orientació final del robot sempre serà la mateixa que la de la línia que uneix els punts inicial i final del moviment. Tot plegat obliga a dissenyar un esquema de decisió basat en casos en què es calculi quin és el punt final de destí d'aquesta maniobra tenint en compte, a més, la velocitat de la pilota. El problema de la velocitat es corregeix de dues maneres: d'una banda la seva magnitud es té en compte en el mateix controlador, i d'altra banda s'utilitza una predicció lineal per a la posició futura de la pilota.

Analitzant els resultats de la maniobra del xut a la secció 5.2.3.1. pot observar-se que només en els casos **A**, **B** i **C** és totalment satisfactòria, és a dir, el robot té capacitat de xutar la pilota de forma correcta només quan es troba 'darrera' la pilota (la zona puntejada de la

figura 5.2.3.1.2⁷). La forma d'escollir quin dels punts **A**, **B** o **C** serà el de destí dependrà completament de que el temps de maniobra compleixi la següent restricció temporal:

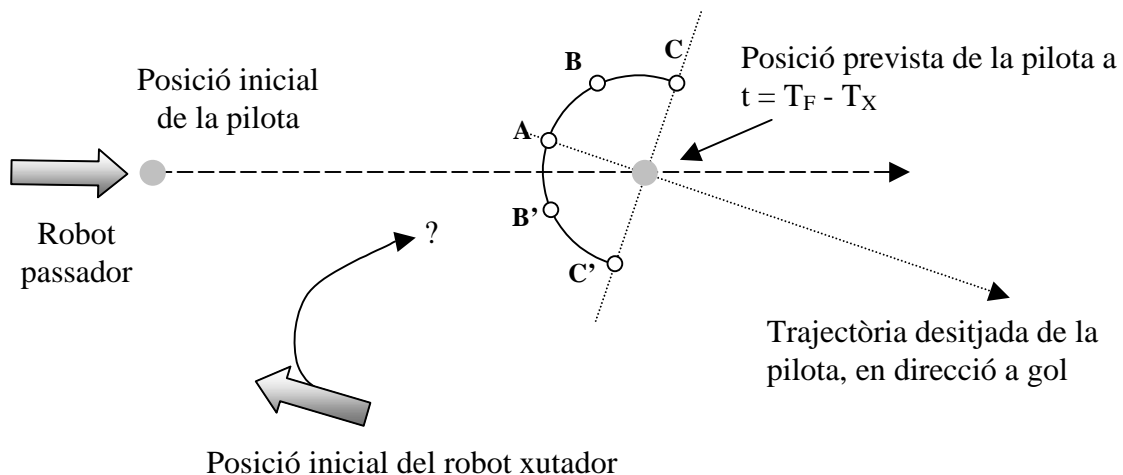
$$T_F \geq T_A + T_X$$

Ara bé, des del punt de vista d'optimització de la capacitat d'efectuar el xut amb el menor valor de $\Delta\alpha$, el millor punt sempre serà l'**A**. Per tant, abans d'escollir cap punt es calcula si la maniobra cap el punt **A** compleix la restricció temporal. En cas negatiu s'intenta el punt **B** i posteriorment el **C**. Aquest ordre de preferència es deu a dos fets: d'una banda s'intenta que la maniobra final del xut pugui ser el més ràpida possible i de l'altra s'intenta tenir en compte que la presència d'altres robots pugui dificultar l'accés a alguns dels punts **A**, **B** o **C**. Sigui quin sigui el punt final **P** que s'esculli, el raonament per casos sempre intentarà minimitzar la suma dels temps d'execució de les dues maniobres. Aquests temps d'execució s'obtenen a partir de les capacitats bàsiques relatives als temps dels supervisors **S₁** i **S₅**:

$$T_A = J_1^a, \text{ considerant diferents els punts } P = \{ A, B, C, B' \text{ o } C' \}$$

$$T_X = \min\{ \max(J_5^k, \text{situació A}), \max(J_5^k, \text{situació B}), \max(J_5^k, \text{situació C}) \}$$

Com es pot observar en el càlcul de T_X , s'ha optat per escollir sempre el cas pitjor en cada una de les situacions. La següent figura mostra quina és la situació inicial que ha de resoldre el raonament basat en casos:

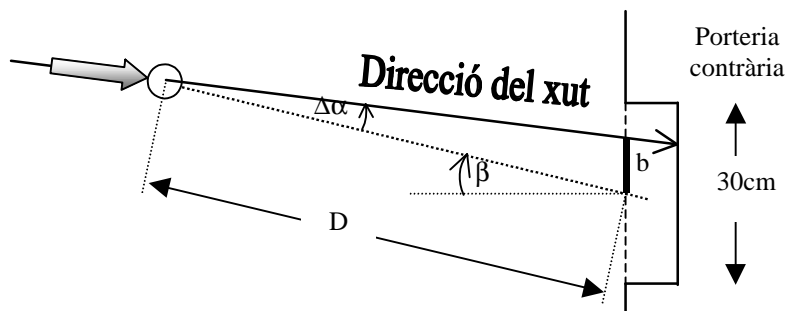


En aquest estat de coses, quan un robot vulgui passar la pilota haurà d'especificar quina és la trajectòria que seguirà la pilota, quina velocitat inicial tindrà i a quin instant de temps (T_F) s'ha de realitzar el xut. El receptor haurà d'avaluar els valors de T_A i T_X

⁷ Cal exceptuar la configuració corresponent a la situació 9, i el cas de `factor=3` per a les situacions 15 i 8.

consultant les capacitats \mathbf{J}_1^a (pels diferents punts \mathbf{P}) i \mathbf{J}_5^k (per les diferents situacions). En cas que no pugi complir la restricció temporal, els dos robots hauran de negociar algun dels paràmetres del moviment de la pilota: velocitat, trajectòria o T_F . Cal comentar que també pot haver-hi situacions en què no hi hagi restricció temporal: en aquest cas el paràmetre a negociar podria ser la velocitat inicial de la pilota. Un altre comentari a fer està relacionat amb la intersecció de trajectòries: els casos en què la maniobra d'apropament intersecció qualsevol punt de la trajectòria de la pilota seran descartats. Per simplificar el càlcul del punt d'intersecció es considera que la maniobra fins el punt \mathbf{P} s'executa en línia recta.

Com ja s'ha comentat anteriorment, la magnitud dels valors de $\Delta\alpha$ són els més importants per poder avaluar la capacitat que té el robot de xutar correctament la pilota cap a la porteria contrària. L'avaluació d'aquesta capacitat també depèn del valor de la distància de la pilota a la porteria. El següent gràfic mostra una situació de xut a una distància D de la porteria i amb un error positiu d'orientació ($\Delta\alpha > 0$). En aquest cas, la pilota entra a la porteria a b cm del centre: aquesta desviació respecte del centre permet fer el càlcul d'un índex de la precisió del xut.



Per tant, aquest índex de precisió (μ) es pot calcular a partir de la següent expressió⁸:

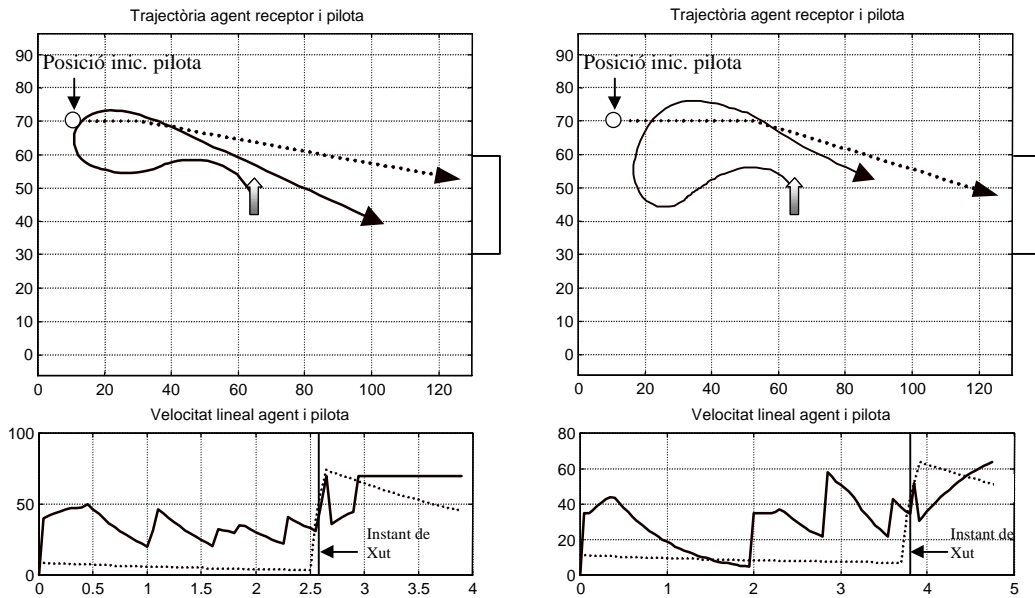
$$b = D \cdot \tan(\Delta\alpha) / \cos\beta$$

$$\mu_{\text{xut}} = 1 - b/15$$

5.2.4.3. Exemples

A continuació es mostren dos exemples: l'un sense restricció temporal (esquerra) i l'altra amb la restricció $T_F=4s$, ambdós amb una velocitat inicial de 10cm/s per a la pilota:

⁸ Si $\mu < 0$ significa que la pilota va fora de la porteria, és a dir, una capacitat negativa és equivalent a dir que el robot no té capacitat per a fer-ho.



En el cas sense restricció temporal el xut es produeix a 2.6s mentre que en el cas de restricció temporal es produeix a 3.9s, és a dir, un valor molt proper a $T_F = 4s$. En el primer cas $\Delta\alpha = 0.37^\circ$, mentre que en segon $\Delta\alpha = 0.8^\circ$, però les distàncies a la porteria en el moment del xut són diferents. Així, els coeficients d'eficiència dels xuts són:

$$\mu_{xut} = 1 - b/15 = 1 - 6/15 = \mathbf{0.6} \quad (\text{sense restricció temporal})$$

$$\mu_{xut} = 1 - b/15 = 1 - 1/15 = \mathbf{0.93} \quad (\text{amb restricció temporal})$$

Respecte al cas amb restricció temporal, s'observa que la maniobra cap el punt **A** presenta una intersecció amb la trajectòria de la pilota, és a dir, l'algoritme s'escull el punt **B'** com a punt **P**. El valor de T_X és doncs:

$$T_X = \max(\mathbf{J}_5^k, \text{situació } \mathbf{B}) = \max(1.35, 1.45, 1.45, 1.8) = 1.8s$$

La localització del punt **B'** es calcula a partir de la posició predita per a la pilota a l'instant de temps següent:

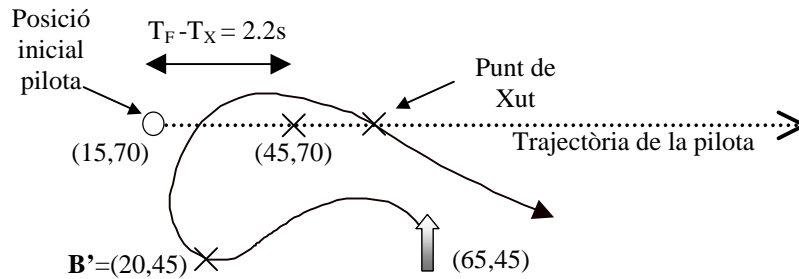
$$t = T_F - T_X = 4 - 1.8 = 2.2s ,$$

és a dir, a la posició

$$(x, y) = (45, 70), \text{ suposant una velocitat de la pilota de } 10\text{cm/s}.$$

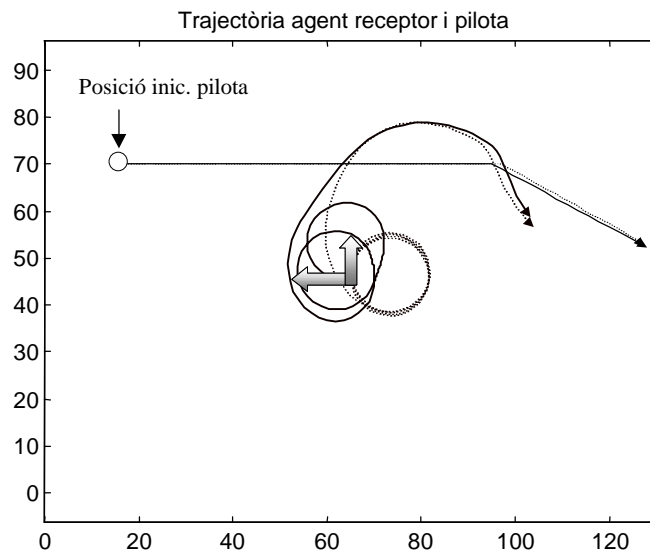
Per tant, el punt **B'** serà finalment el $(x, y) = (20, 45)$. Això permet calcular la distància des del punt inicial del robot fins al **B'** i consultant les equacions del supervisor \mathbf{S}_1 s'obté $T_A = 2s$ si $\alpha = 5.7143$. Així, el temps total acumulat per a les dues maniobres és:

$$T_F = T_A + T_X = 2.0 + .8 = \mathbf{3.8s}$$

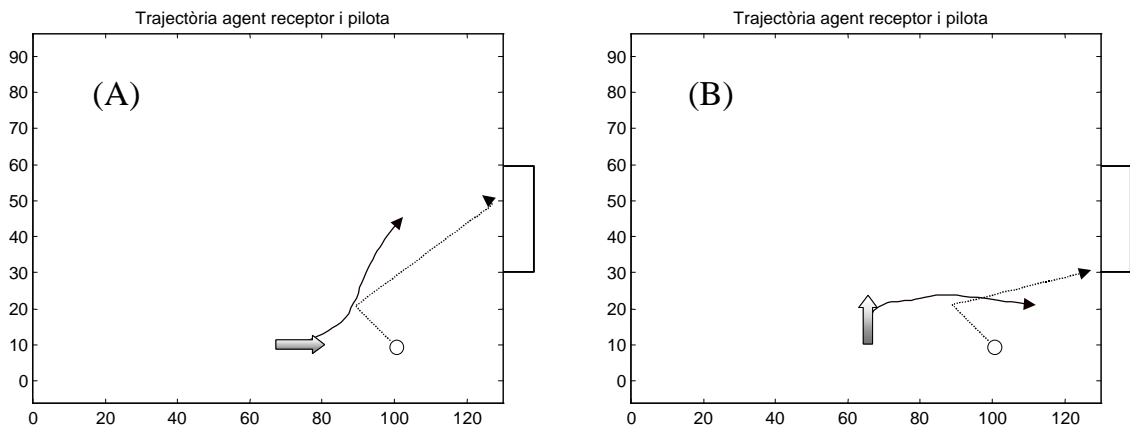


És a dir, a partir de les capacitats bàsiques J_1^a i J_5^{factor} és possible fer càlculs bastant aproximats del temps total T_F i, per tant, hi ha prou coneixement per saber si el robot receptor té temps per xutar la pilota correctament. Quant al valor de μ_{xut} presenta algunes imprecisions: els valors que s'han exposat abans són valors calculats gràficament a partir de la resposta del robot, però si es fa a partir de les taules els resultats són diferents. Per exemple, la maniobra de xut des de la posició B' té un valor $\Delta\alpha = \max\{6.8^\circ, 3.5^\circ, 7.3^\circ, 3.2^\circ\} \cong 7^\circ$ (taula 1, secció 2.2.3., situacions 1, 3, 4 i 13, factor=1), és a dir, s'obté un valor de $b = 9.6\text{cm}$ que, a una distància de 82.8cm , dóna un valor de $\mu_{\text{xut}} = \mathbf{0.36}$. Evidentment aquest valor és bastant inferior al mesurat a partir de la gràfica, però consisteix en ser una cota mínima.

A continuació es mostren dos exemples més en què la restricció temporal es compleix amb escreix. En concret s'ha definit que si la diferència relativa entre els temps és superior al 10% ($(T_F - (T_A + T_X)) / T_F > 0.1$) serà necessari incloure una maniobra d'espera, que consisteix en mantenir la consigna de posició actual. El resultat d'això és que el robot dóna voltes sobre si mateix esperant el temps que li sobra per fer les altres maniobres. L'exemple correspon a un cas amb velocitat inicial de la pilota de 10cm/s però amb $T_F = 8s$. Aquests dos casos intenten il·lustrar el fet que el robot pugui estar bloquejat per altres robots i això impedeix que pugui estar parat del tot.



En els casos en què la maniobra d'apropament no sigui necessària a causa de la proximitat inicial del robot al punt previst de xut, la trajectòria és molt més curta però també molt més forçada, és a dir, hi ha més possibilitats d'obtenir un valor de μ_{xut} baix. A continuació es mostren dos exemples molt semblants: les posicions inicials són les mateixes però les orientacions inicials del robot difereixen en 90° .



Com es pot apreciar, en el cas de l'esquerra (A) la maniobra de xut es pot fer completament i dona lloc a un xut correcte. En canvi, en el de la dreta (B) el xut no és tant correcte perquè la maniobra de xut no es pot finalitzar. Els valors de μ_{xut} que s'obtenen són els següents:

| | Cas (A) | Cas (B) |
|--|--------------------------------------|------------------------------------|
| A partir de les gràfiques | 0.57 | 0.25 |
| A partir de les taules | 0.73 | 0.45 |
| (taula 1, secció 2.2.3, situacions 0 i 14) | $\Delta\alpha=11.7^\circ$ $b=4/15$ | $\Delta\alpha=71^\circ$ $b=8/15$ |

També s'observen discrepàncies entre els valors mesurats gràficament i els calculats mitjançant taules. Els valors són relativament alts perquè s'ha escollit una posició de la pilota en què el xut es faci en una zona molt propera a la porteria contrària (valor de D petit).

En aquest punt cal tenir present que l'objectiu d'aquest paràmetre és conèixer la capacitat que té el robot receptor de xutar la pilota cap a porteria en el rol 'Rebre la pilota'. En concret, la utilitat de μ_{xut} està en el seu ús com a índex d'avaluació de l'eficiència de l'execució del rol. Per tant, si les taules proporcionen una diferència qualitativa entre els dos casos i aquesta diferència és certa, significa que el paràmetre continua essent vàlid.

5.2.5. Especificacions de les escenes: capacitats col·lectives

Recordant que en una escena sempre hi apareixeran relacions de comportament entre els agents que en formen part, en aquest sistema robòtic l'escena es correspon a un moviment tàctic a seguir per un grup de robots. És a dir, consisteix en ser un esquema predefinit per a repartir rols entre els robots de la mateixa escena. Per exemple, el moviment tàctic pot ser tant un moviment en formació (moviment defensiu en bloc) com moviments independents però coordinats (passada de la pilota). En tot cas, els rols d'una escena consistiran en ser comportaments cooperatius planificats prèviament pel dissenyador del sistema.

Els comportaments ofensiu i defensiu d'un equip no es corresponen a dues escenes específiques sinó que hi haurà un grup d'escenes de tipus ofensiu i un altre grup de tipus defensiu. Tot i que podria haver-hi escenes que podrien activar-se en qualsevol dels dos casos, en el disseny d'aquest sistema de robots s'ha optat per dissenyar escenes que siguin o bé ofensives o bé defensives. Així, les escenes serviran per a descriure com es realitza una passada de pilota, com es realitza un contraatac, o com es realitza una defensa per zones o

individual. Continuant amb l'exemple d'escena presentat a la secció 4.3.4.1, s'ha optat per definir les següents escenes:

- Atac → situació general ofensiva, quan no hi ha perill a la porteria local.
- Contraatac → situació específica d'atac quan hi ha pocs contraris a la zona propera a la porteria contrària.
- Defensa zonal → situació general defensiva amb contraris ràpids
- Defensa a l'home → situació general defensiva amb contraris no ràpids
- Altres escenes possibles: defensa urgent, joc individual, protecció de la porteria.

Tornem a remarcar que la protecció de la porteria (el porter) és un comportament que s'adjudica de forma definitiva a un agent. Experimentalment s'observa que moltes vegades el porter queda encallat o bloquejat i fa necessari que el rol de porter s'hagi d'adjudicar dinàmicament. És des d'aquest punt de vista que aquest rol ha de tenir un sentit col·lectiu, no com a comportament cooperatiu sinó com a comportament d'ajuda o de suport.

5.2.5.1. Escena 'Atac'

Aquesta escena correspon a una situació general ofensiva que s'activa quan no hi ha perill a la porteria local, és a dir, quan la pilota és fora d'una zona definida com de perill. Aquesta escena pot definir-se de moltes maneres donat que el comportament d'atac permet moltes interpretacions. S'ha optat per una definició concreta que no pretén ser generalista: l'atac consisteix en fer que un agent xuti la pilota cap el camp contrari i que algun altre faci el xut cap a la porteria contrària. Mentrestant, un tercer agent es col·loca en una situació defensiva per protegir un possible atac. Per tant, el nombre mínim d'agents per activar aquesta escena és 3, i les característiques dels tres rols són les següents:

Rol 1

Rol → 'Passar la pilota'

Precondició → La pilota està a distància petita

Acció comunicativa → Negociar condició del xut amb l'agent de rol 2.

Rol 2

Rol → 'Rebre la pilota'

Precondició → L'agent està per davant de la pilota

Acció comunicativa → Negociar condició de recepció amb l'agent amb rol 1.

Rol 3

Rol → 'Anar de A a B seguint una trajectòria lliure de col·lisió'

Precondició → .NOT. precond(rol 2) .AND NOT. precond(rol 1)

Acció comunicativa → cap

5.2.5.2. Escena 'Contraatac'

Aquesta escena correspon a una situació també d'atac que s'activa quan hi ha pocs contraris a la zona propera a la porteria contrària. En aquesta situació hi ha una alta probabilitat de que un xut cap a porteria contrària es converteixi en gol. Aquesta escena s'ha definit considerant que pot ésser executada amb dos agents: el contraatac consisteix en fer que l'agent més proper a la pilota la xuti cap a porteria contrària. En ocasions, un altre agent pot ser receptor de la pilota i executar el xut cap a la porteria contrària. Per tant, el nombre mínim d'agents per activar aquesta escena és 2, i els rols coincidarien amb els anteriors però eliminant el tercer.

5.3. Conclusions

L'objectiu d'aquest capítol és mostrar un exemple on l'aplicació del concepte d'introspecció sigui útil per tal d'avaluar l'eficiència d'execució d'accions. Per les raons que s'han comentat al principi del capítol, el domini del futbol robòtic proporciona diverses situacions en què la cooperació entre robots és necessària i on l'eficiència en l'execució dels moviments dels robots és un factor crític. L'exemple que s'ha triat es basa en la situació on un robot (passador) vol passar la pilota a un altre (xutador) que ha de rebre-la xutant-la al mateix temps cap a la porteria contrària. Per tant, és un exemple on la cooperació entre els robots serà bona si els moviments s'executen de forma eficient.

Agafant l'arquitectura DPAA com a marc de referència s'ha mostrat en detall els passos a seguir per dissenyar un agent físic capaç d'avaluar l'eficiència de l'acció de rebre la pilota xutant-la cap a la porteria contrària, és a dir, un agent físic capaç de fer introspecció.

En primer lloc s'han obtingut les capacitats atòmiques a partir de les característiques de dos controladors: un controlador no lineal per a moviments curts, i un altre de lineal per a desplaçaments llargs. Després s'ha mostrat com aquests controladors es poden utilitzar per definir l'estructura de supervisors de tasques de nivell més alt i quina és la forma de

calcular les capacitats bàsiques. Finalment, s'ha definit el rol de rebre la pilota a partir de la combinació de supervisors definits en el pas previ i s'ha mostrat com l'agent/robot xutador pot calcular un índex per avaluar l'eficiència en executar el rol.

En aquest darrer pas s'han mostrat diversos exemples simulats on el robot xutador executa el seu rol i s'han comparat els resultats finals amb els valors del l'índex esmentat. Es pot concloure que l'agent és capaç de fer introspecció si disposa de la informació adequada i estructurada d'acord amb les especificacions de l'arquitectura DPAA. El resultat de la introspecció és un índex que pot utilitzar-se per a negociar amb l'agent passador les condicions de la recepció de la pilota.

CONCLUSIONS

Conclusions

L'experiència de l'autor en la temàtica d'agents intel·ligents i la seva aplicació als robots que emulen el joc de futbol, tal com va proposar Allan Macworth el 1993, han donat el bagatge suficient per poder encetar i proposar la temàtica plantejada en aquesta tesi: com fer que un complicat robot pugui treure el màxim suc de l'autoconeixement de l'estructura de control inclosa al seu propi cos físic, i així poder cooperar millor amb d'altres agents per optimitzar el rendiment a l'hora de resoldre problemes de cooperació.

Per resoldre aquesta qüestió s'ha proposat d'incorporar la dinàmica del cos físic en les decisions cooperatives dels agents físics unificant els móns de l'automàtica, la robòtica i la intel·ligència artificial a través de la noció de **capacitat**: la capacitat vista com a entitat on els enginyers de control dipositen el seu coneixement, i a la vegada la capacitat vista com la utilitat on un agent hi diposita el seu autoconeixement del seu cos físic que ha obtingut per introspecció.

En aquesta tesi s'ha presentat l'arquitectura DPAA que s'organitza seguint una jerarquia vertical en tres nivells d'abstracció o mòduls —control, supervisor i agent, els quals presenten una estructura interna homogènia que facilita les tasques de disseny de l'agent. Aquests mòduls disposen d'un conjunt específic de capacitats que els permeten avaluar com seran les accions que s'executaran en un futur. En concret, al mòdul de control (baix nivell d'abstracció) les capacitats consisteixen en paràmetres que descriuen el comportament dinàmic i estàtic que resulta d'executar un controlador determinat, és a dir, encapsulen el coneixement de l'enginyer de control. Així, a través dels mecanismes

de comunicació entre mòduls aquest coneixement pot anar introduint-se als mecanismes de decisió dels mòduls superiors (supervisor i agent) de forma que quan els paràmetres dinàmics i estàtics indiquin que pot haver-hi problemes a baix nivell, els mòduls superiors es poden responsabilitzar d'inhibir o no l'execució d'algunes accions. Aquest procés *top-down* intern d'avaluació de la viabilitat d'executar una acció determinada s'anomena **procés d'introspecció**.

S'han presentat diversos exemples per tal d'il·lustrar com es pot dissenyar un agent físic amb dinàmica pròpia utilitzant l'arquitectura DPAA com a referent. En concret, s'ha mostrat tot el procés a seguir per dissenyar un sistema real format per dos robots en formació de comboi, i s'ha mostrat com es pot resoldre el problema de la col·lisió utilitzant les capacitats a partir de les especificacions de disseny de l'arquitectura DPAA.

Al cinquè capítol s'ha exposat el procés d'anàlisi i disseny en un domini més complex: un grup de robots que emulen el joc del futbol. Els resultats que s'han mostrat fan referència a l'avaluació de la validesa de l'arquitectura per resoldre el problema de la passada de la pilota. S'han mostrat diversos resultats on es veu que és possible avaluar si una passada de pilota és viable o no. Encara que aquesta possibilitat ja ha estat demostrada en altres treballs, l'aportació d'aquesta tesi està en el fet que és possible avaluar la viabilitat a partir de l'encapsulament de la dinàmica en unes capacitats específiques, és a dir, és possible saber quines seran les característiques de la passada: el temps del xut, la precisió o inclòs la geometria del moviment del robot xutador.

Els resultats mostren que la negociació de les condicions de la passada de la pilota és possible a partir de capacitats atòmiques, les quals inclouen informació sobre les característiques de la dinàmica dels controladors. La complexitat del domini proposat fa difícil comparar els resultats amb els altres treballs. Cal tenir present que els resultats mostrats s'han obtingut utilitzant un simulador fet a mida que incorpora les dinàmiques dels motors dels robots i de la pilota. En aquest sentit cal comentar que no existeixen treballs publicats sobre el problema de la passada en què es tingui en compte la dinàmica dels robots.

El present treball permet assegurar que la inclusió de paràmetres dinàmics en el conjunt de les capacitats de l'agent físic permet obtenir un millor comportament col·lectiu dels robots, i que aquesta millora es deu al fet que en les etapes de decisió els agents utilitzen informació relativa a la viabilitat sobre les seves accions: aquesta viabilitat es pot calcular a partir del comportament dinàmic dels controladors. De fet, la

definició de capacitats a partir de paràmetres dinàmics permet treballar fàcilment amb sistemes autònoms heterogenis: l'agent físic pot ser conscient de les seves capacitats d'actuació a través de mecanismes interns d'introspecció, i això permet que pugui prendre compromisos amb altres agents físics.

Treball futur

La idea d'encapsular la dinàmica en les capacitats de l'agent s'ha utilitzat amb èxit en el disseny dels comportaments dels robots de l'equip ROGI-Team a la competició RoboCup 1999 i 2000. Cal aclarir que l'arquitectura no és la DPAA ja que els agents programats en aquest equip són petits sistemes experts, és a dir, estan implementats en forma de sistema de decisió basat en regles *fuzzy*. Com a treball futur immediat es planteja construir una plataforma informàtica que mecanitzi tot el procés de disseny d'agents físics basats en l'arquitectura DPAA.

Les idees presentades en aquesta tesi poden tenir impacte en aplicacions d'àmbits científics i tecnològics on emprar coneixement explícit de la dinàmica sigui crític per evitar catàstrofes, o en entorns canviants on assolir una millor adaptabilitat es faci necessari, o bé en problemes complexos en què faci falta optimitzar la solució. Exemples concrets d'aquestes aplicacions són a l'àmbit de la navegació aeroespacial i l'autopista del futur, així com els futurs ecosistemes universals d'informació on va a parar l'actual societat de la informació.

És desitjable l'aplicació del mètode de disseny a altres sistemes autònoms reals que mostrin casuístiques relacionades amb dinàmiques d'ordre superior o amb fenòmens no lineals: s'haurà de veure quins són els mecanismes correctes per construir capacitats bàsiques que mantinguin la coherència. Possiblement la solució sigui passar per mètodes de disseny basats en controladors híbrids.

Els resultats de la discussió presentada en aquesta tesi mostren que la noció de capacitat és molt útil i que planteja noves qüestions: com se sintetitzen les capacitats, com s'aprenen? Sembla natural que el següent pas sigui analitzar si la definició estructural de la capacitat proposada en aquesta tesi és suficient per encabir la síntesi o l'aprenentatge.

Actualment, l'autor d'aquesta tesi està involucrat en un projecte de recerca que té per nom "Sistema multi-agente de observadores avanzados para análisis y

reconocimiento de escenas en entornos adversos" en què l'objectiu consisteix en dissenyar un sistema multi-robot capaç de obtenir un model 3D d'una escena poc estructurada a partir de la cooperació dels robots. El sistema consta d'un mínim de tres robots completament autònoms i heterogenis, és a dir, amb diferents instruments de mesura i amb capacitats de moviment també diferenciades. En aquest sistema la cooperació és necessària tant per compartir informació com per repartir les tasques relacionades amb els moviments dels robots a través de l'escena. És a dir, que la noció d'introspecció presentada en aquesta tesi serà necessària per a optimitzar el funcionament del sistema global.

REFERÈNCIES

Bibliografia

- [Agha, 86] G.Agha. ACTORS: A Model of Concurrent Computation in Distributed Systems. *The MIT Press. Cambridge, MA* (1986)
- [Aguilar, 97] L.E Aguilar Melo. Tesi doctoral al Laboratoire d'Analyse et Architecture des Systèmes du CNRS (LAAS). Report LAAS no. 97452 (1997)
- [Asada, 97] M.Asada, Y.Kuniyoshi, et al. The RoboCup Physical Agent Challenge. *First RoboCup Workshop in the XV IJCAI-97 International Joint Conference on Artificial Intelligence*, pp.51-56 (1997)
- [Asama, 89] H.Asama, A.Matsumoto and Y.Ishida. Design of an Autonomous and Distributed Robot System: ACTRESS. *IEEE/RSJ IROS*. pp. 283-2990 (1989)
- [Arkin, 93] R.C.Arkin, T.Balch and E.Nitz. Communication of Behavioral State in Multi-Agent Retrieval Tasks. *ICAR'93*. pp. 588-594 (1993)
- [Balch, 95] T.Balch, R.C.Arkin. Motor Schema-Based formation control for Multi-Agent Teams. *In Proc.1st Int'l Conference on Multi-Agent Systems (ICMAS'95)* (1995)
- [Barraquand, 89] J.Barraquand and J.C.Latombe. On non-holonomic mobile robots and optimal maneuvering. *Revue d'Intelligence Artificielle*, 3(2):77-103, (1989)
- [Barker, 95] H.A.Barker. Open Environments and Object-Oriented Methods for Computer-Aided Control System Design. *Control Engineering Practice*, 3(3): 347-356 (1995)
- [Bates, 94] J.Bates. The role of emotion in believable agents. *Comms. of the ACM*. 37(7): 122-125 (1994)
- [Beer, 92] R.D.Beer, and J.C.Gallagher. Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1: 91-112 (1992)
- [Belnap, 88] N.Belnap, M.Perloff. Seeing to it that: a canonical form for agentives. *Theoria*. 54: 175-199 (1988)
- [Belnap, 93] N.Belnap, and M.Perloff. In the realm of agents, *Annals of Mathematics and Artificial Intelligence*. 9: 25-48 (1993)
- [Beni, 88] G.Beni. The concept of cellular robotics system. *IEEE Int. Symposium on Intelligent Control*. pp. 57-62 (1988)
- [Bergener, 98] T.Bergener, and A.Steinlage. An Architecture for Behavioral Organization using Dynamical Systems. *C.Wilke, S.Altmeyer & T.Martínez (eds.), Abstrancting and Synthesizing the Principles of Living Systems. 3th German Workshop on Artificial Life (GWAL'98)*. pp. 31-42 (1998)

- [Bérry, 92] G.Bérry, and G.Gonthier. The ESTEREL synchronous programming language: Design, semantics, implementation. *Science of Computer Programming* 19(2): 87-152 (1992)
- [Bloch, 92] A.Bloch, M.Reyhanoglu and N.H.McClambroch. Control and stabilization of nonholonomic dynamic systems. *IEEE Transactions on Automatic Control*. 37(11): 1746-1757 (1992)
- [Botti, 99] V.Botti, C.Carrascosa et al. Modelling Agent in Hard Real-Time Environments. *Multi-Agent System Engineering*. LNAI 1647, pp.63-76. *Proc. of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'99*. Valencia, Spain. June-July (1999)
- [Bouron, 90] T.Bouron, J.Ferber, and F.Samuels. MAGES: A Multi-Agent Testbed for Heterogeneous Agents. In *Y.Demazeau and J-P Müller, editors, Decentralized A.I. 2. Proceedings of MAAMAW'90*. North-Holland (1990)
- [Bratman, 88] M.E.Bratman, D.J.Israel, M.E.Pollock. Plans and resource-bounded practical reasoning, *Computational Intelligence*. 4: 349-355 (1988)
- [Brooks, 86] R.A.Brooks. A Robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14-23 (1986)
- [Brooks, 88] R.A.Brooks. A robot that walks: emergent behaviour from a carefully evolved network. *Cambridge, MA: MIT* (1988)
- [Brooks, 90] R.A.Brooks. The behaviour language; users guide. *A.I. memo 1227, Massachusetts Institute of Technology* (1990)
- [Burmeister, 92] B.Burmeister, and K.Sundermeyer. Cooperative problem solving guided by intentions and perception. In Werner, E. and Demazeau, Y. editors, *Decentralized AI 3 - Proceedings of the Third European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91)*, pp. 77-92. Elsevier Science Publishers B. V.: Amsterdam, The Netherlands (1992)
- [Burxhert, 90] H.Burxhert and J.Müller. RATMAN: Rational Agents for Multi-Agent Networks. In *Decentralized A.I. 3. Proceedings of MAAMAW'91*. pp. 243-257. North-Holland (1990)
- [Busmann, 94] S.Bussmann and Y.Demazeau. An Agent Model Combining Reactive & Cognitive Capabilities. *IROS'94*. pp. 2095-2102 (1994)
- [Calou, 90] P.Calou, W.Choi et al. Indoor automation with many mobile robots. In *IEEE/RSJ IROS*, pp. 67-72 (1990)
- [Campion, 91] G.Campion, B.d'Andrea-Novel and G.Bastin. Controllability and state feedback stability of nonholonomic mechanical systems. *Advanced Robot Control. C.Canudas de Wit ed. LNCIS 1662, Springer*. pp. 106-124 (1991)

- [Cao, 95] Y.U. Cao, A.S. Fukunaga, A.B. Kahng. Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots*, vol. 4, pp. 1-23. Kluwer Academic Publishers, Boston. (1995)
- [Cattoni, 93] R.Cattoni, T.Coianiz, and B.Caprile. Reactivity and Planning in the Mobile Robot of MAIA. *Int'l Conf. On Robotics and Automation (ICRA'93)* (1993)
- [Chan, 90] R.B.Chan and D.S.Childress. On Information Transmission in Human-Machine Systems: Channel Capacity and Optimal Filtering. *IEEE Transactions on Systems, Man and Cybernetics*. Vol 90, no. 5, Sept/Oct (1990)
- [Chapman, 86] D.Chapman, and P.Agre. Abstract reasoning as emergent from concrete activity. In Georgeff, M. P. and Lansky, A. L., editors. *Reasoning About Actions and Plans - Proceedings of the 1986 Workshop*, pages 411-424. Morgan Kaufmann Publishers: San Mateo, CA (1986)
- [Cheong, 92] Fah-Chun Cheong. OASIS: An Agent-Oriented Programming Language for Heterogeneous Distributed Environment PhD. Thesis, Dept. of Computer Science and Engineering, The University of Michigan (1992)
- [Clavel, 93] N.Clavel, R.Zapata, and F.Sevilla. Non-holonomic Local Navigation by means of Time and Speed Dependent Artificial Potential Fields. IFAC Intelligent Autonomous Vehicles. Southampton, UK (1993)
- [Cliff, 93] D.Cliff, I.Harvey, and P.Husbands. Explorations in evolutionary robotics. *Adaptive Behavior*, 2: 73-110 (1993)
- [Cohen, 89] P.R.Cohen, M.L.Greenberg, D.M.Hart, and A.E.Howe. Trial by Fire: Requirements for Agents in Complex Environments. *AI Magazine*, 10(3):33-48 (1989)
- [Cohen, 90] P.R.Cohen, and H.J.Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213-261 (1990)
- [de la Rosa, 97] J.Ll.de la Rosa, A.Oller, J.Vehí and J.Puyol. Soccer Team Based on Agent-Oriented Programming. *Robotics and Autonomous Systems*. Ed. Elsevier Science B.V. 21, pp. 167-176. (1997)
- [de la Rosa, 00] J.Ll.de la Rosa, B.Innocenti, A.Oller, A.Figueras, J.A.Ramon, I.Muñoz, M.Montaner. An Example of Dynamical Physical Agents. *European Robotics Workshop*. Amsterdam - Holland. (2000)
- [Dean, 88] T.Dean and M.Boddy. An Analysis of Time-Dependent Planning. *In Proc. of the 7th Nat'l Conf. on Artificial Intelligence*. pp.49-54 (1988)
- [Decker, 94] K.S.Decker and V.R.Lesser. Communication in the service of Coordination. *Treball intern del projecte DARPA*. Dep. of Computer Science. Univ. of Massachusetts (1994)
- [Decker, 95] K.S.Decker. Environment Centered Analysis and Design of Coordination Mechanisms. *UMass CMPSCI Technical Report 95-XXX*. Dep. of Computer Science. Univ. of Massachusetts (1995)

[Dennett, 87] D.C.Dennett. The intentional stance. *The MIT Press: Cambridge, MA* (1987)

[d'Inverno, 98] M.d'Inverno, D.Kinny, M.Luck, and M.Wooldridge. A formal especificacion of dMARS. Singh et al, aditors, *In Proc. of the 4th Int'l Workshop on Agent, Theories, Architectures and Languages (ATAL'97)*, LNAI. 1365: 155-176, Springer, 1998.

[Dorf, 90] R.Dorf. Concise Int'l Encyclopedia of Robotics: Applications and Automation. *Wiley-Interscience* (1990)

[Durfee, 87] E.D.Durfee, V.R.Lesser, and D.E.Corkill. Cooperation Through Communication in Distributed Problem Solving Network. A *M.N. Huhns, editor, Distributed Artificial Intelligence. Pitman Publishers* (1987)

[Etzioni, 93] O.Etzioni. Intelligence without Robots (A Reply to Brooks). *AI Magazine, December* (1993)

[Feigenbaum, 88] E.A.Feigenbaum, P.McCorduck, H.P.Nii. The Rise of the Expert Company: How Visionary Companies are Using Artificial Intelligence to Achieve Higher Productivity and Products. *Times Books, New York* (1988)

[Ferguson, 92] I.A.Ferguson. Turing Machines: An Architecture for Dynamic, Rational, Mobile Agents. *PhD Thesis, Clare Hall, University of Cambridge, UK* (1992)

[Ferrel, 96] C.Ferrel. Robust Agent Control of an Autonomous Robot with many Sensors and Actuators. *Ph.D Thesis*, presentada al Novembre del 1996 al Massachusetts Institute of Technology (1996)

[Finin, 93] T.Finin,et. al. Draft Specifications of the KQML (The Knowledge Query & Manupulation Language) Unpublished draft (1993)

[Finin, 94] E.Finin, R.Fritzson, D.McKay, R.McEntire. KQML as an agent communication language. *Proceedings CIKM'94* (1994)

[FIPA, 98] FIPA'98 Draft Specifications: Part 1. Aquesta versió i altres posteriors es poden trobar a la pàgina web de la FIPA: <http://www.fipa.org>

[Fisher, 94] M.Fisher. A survey of Concurrent MetateM - The language and its applications. In Gabbay, D. M. and Ohlbach, H. J., editors, *Temporal Logic - Proceedings of the first International Conference (LNAI Volume 827)*, pp. 480-505. Springer-Verlag: Heidelberg, Germany (1994)

[Franklin, 87] R.F.Franklin and L.A.Harmon. Elements of Cooperative Behavior. *Internal Research and Development Final Report, 655404-1-F. ERIM* (1987)

[Franklin, 96] S.Franklin and A.Graesser. It is an Agent or just a Program?: A Taxonomy for Autonomous Agents. *Proc. of the 3th Int'l Workshop on Agents Theories, Architectures, and Languages (ATAL-96)*. Springer-Verlag (1996)

- [Fukuda, 89] T.Fukuda, S.Nakagawa et al. Structure Decision Method for Self Organizing Robots based on Cell Structures-CEBOT. *In Proc. IEEE Int'l Conference on Robotics and Automation*. Vol 2, pp. 695-700 (1989)
- [García-Fornés, 95] A.García-Fornés, A.Crespo, V.Botti. Adding hard real-time tasks to artificial intelligence environments. *Workshop on Real-Time Programming WRTP'95*. A Postprint Volumen from the 20th International Whorkshop on Real-Time Programming (ISBN 0-08-042590-9). Pergamon. Elsevier Scientific Fort Lauderdale, Florida (USA) (1995)
- [García, 98] R.García, J.Ll.de la Rosa, A. Oller, A.Figueras, J.A.Ramon. A Fast Vision System for Tracking in Robotic Soccer. *Proc of Workshop of MIROSOT '98 (3th IEEE Micro-Robot Soccer Tournament)*. pp. 73-83. París, França (1998)
- [Garvey, 93] A.Garvey, M.Humphrey and V.Lesser. Task interdependencies in design-to-time real-time scheduling. *In Proc. of the 11th Nat'l Conf. on Artificial Intelligence*. pp. 580-585 (1993)
- [Garvey, 94] A.Garvey and V.Lesser. A Survey of Research in deliberative Real-Time Artificial Intelligence. *Journal of Real-Times Systems*, 6(3):317-347 (1994)
- [Georgeff, 86]. M.Georgeff. A representation of events in multi-agent domains. *In Proc. of AAAI'86*. Morgan Kaufmann Publishers. pp. 70-75 (1986)
- [Georgeff, 87] M.P.Georgeff, and A.L.Lansky. Reactive Reasoning and Planning. *In Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, pp. 677-682, Seattle, WA (1987)
- [Greenberg, 87] M.Greenberg and L.Werbrook. The PHOENIX Testbed. *Technical Report COINS TR 90-19, Dept. of Computer and Information Science, Univ. of Massachussets* (1990)
- [Halbwachs, 91] N.Halbwachs, P.Caspi, P.Raymond, and D.Pilaud. The synchronous dataflow programming language lustre. *In Proc. of the IEEE*, 79(9):1305-1320 (1991)
- [Hanks, 93] S.Hanks, M.E.Pollack, and P.R.Cohen. Benchmarks, Testbeds, Controlled Experimentation, and the Design of Agent Architectures. *AI Magazine*, 14(3):17-42 (1993)
- [Haugeneder, 94] H.Haugeneder, D.Steiner, and F.G.McCabe. IMAGINE: A framework for building multi-agent systems. In Deen, S. M., editor, *Proceedings of the 1994 International Working Conference on Cooperating Knowledge Based Systems (CKBS-94)*, DAKE Centre, University of Keele, UK (1994)
- [Haugeland, 85] J.Haugeland, *Artificial Intelligence: The Very Idea*. Cambridge, MA: MIT Press (1985)
- [Hayes-Roth, 92] B.Hayes-Roth, R.Washington, et al. Guargian: A prototype for Intensive-care monitoring agent. *Artificial Intelligence in Medicine*. 4:165-185 (1992)

[Hayes-Roth, 95] B.Hayes-Roth. An Architecture for Adaptive Intelligent Systems. *Artificial Intelligence*, 72:329-365 (1995)

[Hewitt, 77] C.Hewitt. Viewing control structures as patterns of passing messages. *Artificial Intelligence*, 8(3):323-364 (1977)

[Hintikka, 62] J.Hintikka. Knowledge and Belief. *Cornell University Press*. Ithaca, NY (1962)

[Howe, 90] A.E.Howe, D.M.Hart, and P.R.Cohen. Addressing real-time constraints in the design of autonomous agents. *The Journal of Real-Time Systems*. 2:81-97 (1990)

[Huang, 95] J.Huang, N.R.Jennings, and J.Fox. An agent based approach to health care management. *Journal of Applied Artificial Intelligence*. Special Issue on Intelligent Agents- eds. M.J.Wooldridge & N.R.Jennings (1995)

[Hunhs, 92] M.N.Hunhs, N.Jacobs, T.Ksiezyk, W.M.Shen, M.P.Singh, P.E.Cannata. Integrating enterprise information models in Carnot. *In Proc. of the International Conference on Intelligent and Cooperative Information Systems*, pp. 32-42 (1992)

[Ingrand, 92] F.Ingrand, M.P.Georgeff and A.Rao. An Architecture for real-time reasoning and system control. *IEEE Expert*. 34-44, December (1992)

[Jennings, 92] N.R.Jennings. On being responsible. In E.Werner, and Y.Demazeau editors, *Decentralized AI Proceedings of the Third European Workshop on modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91)*, pp. 93-102. Elsevier Science Publishers B.V. Amsterdam, The Netherlands (1992)

[Jennings, 92a] N.R. Jennings. Joint Intentions as a Model of Multi-Agent Cooperation. *PhD thesis, Queen Mary and Westfield College, University of London* (1992)

[Jennings, 93] N.R.Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *Knowledge Engineering Review*, 8(3): 223-250 (1993)

[Jennings, 93a] N.R.Jennings, L.Z.Varga, R.P.Aarnts, J.Fuchs and P.Skarek. Transforming standalone expert systems into a community of cooperating agents. *Int. Journal of Engineering Applications of Artificial Intelligence* 6 (4): 317-331 (1993)

[Jennings, 95] N.R. Jennings, J.Corera, I.Laresgoiti, E.H.Mamdani, F.Perriolat, P.Skared and L.Z.Varga. Using ARCHON to develop real-world DAI applications for electricity transportation management and particle accelerator control. *IEEE Expert - Special Issue on Real World Applications of DAI* (1995)

[Jennings, 95a] N.R.Jennings and A.J.Jackson. Agent based Meeting Scheduling : A Design and Implementation. *IEE Electronics Letters Journal*. (1995)

[Jyishane, 94] L.Jyishane and K.P.Sycara. Distributed Problem Solving through Coordination in a Society of Agents. *In the Proceedings of the 13th International Workshop on Distributed Artificial Intelligence*. (1994)

- [Kaebling, 89] L.P.Kaebling, N.J.Wilson. Rex programmer's manual. *Technical report 381R, Artificial Intelligence Center, SRI Int'l, Menlo Park, California* (1988)
- [Kaebling, 90] L.P. Kaebling. An Architecture for Intelligent Reactive Systems. *Readings in Planning*. pp. 713-728 (1990)
- [Khalil, 00] C. Khalil. Multi-Agent Systems: A Review of Current Technologies. *IMPACT Research Group, Department of Computer Science, Loughborough University* (2000)
- [Kathib, 85] O.Kathib. "Real-Time Obstacle Avoidance for manipulators and Mobile Robots". IEEE St. Louis (USA). March, 25-26. 1985.
- [Kersual, 94] D.Kersual and A.Mensch. REAKT: A Real-Time Architecture for Knowledge Based Systems. *IFAC Artificial Intelligence in Real-Time control*. ISBN: 84-7721-274. pp. 513-518 (1994)
- [Kitano, 97] H.Kitano, M.Asada, Y.Kuniyoshi, I.Noda, E.Osawa, H.Matsubara. RoboCup: A Challenge AI Problem, *AI Magazine*, Spring. (1997)
- [Kitano, 97a] H. Kitano, M. Tambe, P. Stone, M. Veloso, S. Coradeschi, E. Osawa, H. Matsubara, I. Noda, M. Asada. The RoboCup Synthetic Agent Challenge 97. *In Proc. of IJCAI-97*. (1997)
- [Kolmanovsky, 95] I.Kolmanovsky and N.H.Haris. Developments in Nonholonomic Control Problems. *IEEE Control Systems*. 15(6): 20-36 (1995)
- [Konolige, 86] K.Konolige. A deduction Model of Belief. *Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA* (1986)
- [Kripke, 63] S.Kripke. Semantical analysis of modal logic. *Zeitschrift für Matematische Logik und Grudlagen der Mathematik*, 9:67-96 (1963)
- [Laumond, 94] J.P. Laumond, P.E.Jacobs et al. A Motion Planner for Nonholonomic Mobile Robots. *IEEE Transactions on Robotics & Automation*. 10(5): 577-593 (1994)
- [Leonard, 91] J.J.Leonard and H.H.Durrant-Whyte. Mobile Robot Localization by Tracking Geometric Beacons. *IEEE Transactions on Robotics & Automation*. 7(3): 376-382 (1991)
- [Lesser, 83] V.R.Lesser and D.D.Corkill. The distributed vehicle monitoring testbed: a tool for investigating distributed problem solving network. *AI Magazine*, pp. 15-33 (1983)
- [Levesque, 84] H.J.Levesque. A logic for implicit and explicit belief. *In Proceedings of the Fourth National Conference on Artificial Intelligence (AAAI-84)*, pp. 198-202, Austin, TX (1984)
- [Lin, 93] L.J.Lin. Scaling up reinforcement learning for robot control. *In Machine Learning: Proceedings of the Tenth International Conference*, pp. 182-189 (1993)

[Mackworth, 93] A.K.Mackworth. On Seeing Robots. *Computer Vision: Systems, Theory, and Application*. Pp 1-13, World Scientific Press, Singapore (1993)

[Maes, 89] P.Maes. The dynamics of action selection. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, pp. 991-997, Detroit, MI (1989)

[Maes 90] P. Maes. Situated Agents Can Have Goals. *Designing Autonomous Agents*. Ed: P.Maes. MIT-Bradford Press, 1991. També publicat com a *Special issue of the Journal for Robotics and Autonomous Systems*, 6(1) , North-Holland, June (1990)

[Maes, 94] P.Maes. Social interface agents: Acquiring competence by learning from users and other agents. In Etzoni O. Editor. *Software Agents - Papers from the 1994 Spring Symposium*, pp. 71-78, AAAI press (1994)

[Maes, 94a] P.Maes. Agents that reduce work and information overload. *Comms. of the ACM* 37 (7): 31-40 (1994).

[Mahadevan, 91] M.Mahadevan, and J.Connell. Scaling Reinforcing Learning to robotics by exploiting the subsumption architecture. In the 8th Intl Workshop on Machine Learning. S. 328-332 (1991)

[Matellán, 98] V.Matellán. ABC²: Un Modelo para el Control de Robots Autónomos. *Tesi Doctoral. Universidad Politécnica de Madrid. Facultad de Informática, Dep. de Inteligencia Artificial (1998)*

[Marcia, 93] MARCIA. Report d'avançament del grup de treball MARCIA. *Journée Multi-Agent du PRC-GDR IA*. Montpellier (1993)

[Mataric, 92] M.J.Mataric. Distributed approaches to behavior control. In *SPIE – Sensor Fusion V*. 1828: 373-382 (1992)

[Mataric, 95]. M.J.Mataric and D.Cliff. Challenges In Evolving Controllers for Physical Robots. *Brandeis University Computer Science Technical Report CS-95-184*, Nov. 1995. També amb el títol "Evolutional Robotics", special issue of *Robotics and Autonomous Systems*, 19(1), Oct 1996, 67-83.

[McCabe, 94] F.G.McCabe, and K.L.Clark. APRIL - agent process interaction language. In Wooldridge, M. and Jennings, N. R., editors, *Pre-proceedings of the 1994 Workshop on Agent Theories, Architectures, and Languages*, pages 280-296, Amsterdam, The Netherlands (1994)

[McDermott, 82] D.McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*. 6:101-155 (1982)

[Miller, 82] R.A.Miller, H.E.Pople, J.D.Myers. INTERNIST-1: An experimental computer-based diagnostic consultant for general internal medicine. *New. Eng. J. Med.* 307: 468-476 (1982)

[Moore, 90] R.C.Moore. A formal theory of knowledge and action. In Allen, J. F., Hendler, J., and Tate, A., editors. *Readings in Planning*, pp. 480-519. Morgan Kaufmann Publishers: San Maeto, CA (1990)

[Montgomery, 92] T.A.Montgomery, J.Less, D.J.Musliner, E.H.Durfee et al. MICE Users Guide. Distributed Intelligent Agents Group, Dept. of Electrical Engineering and Computer Science, Univ. of Michigan (1992)

[Müller, 96] J.P.Müller. The Design of Intelligent Agents. A Layered Approach. *Lecture Notes in Artificial Intelligence*, Vol. 1177. Ed. Springer-Verlag. 1996.

[Musliner, 93] D.J.Musliner, E.Durfee and K.Shin. CIRCA: a co-operative intelligent real-time control architecture. *IEEE Transactions of Systems, Man and Cybernetics*. 23(6) (1993)

[Nagendrapassad, 94] M.Nagendrapassad, V.Lesser, S.E.Lander. Learning organizational roles in a multi-agent system. *UMass Technical report*. (1994)

[Noda, 95] I.Noda. Tota la informació sobre el simulador "Soccer Server" pot trobar-se a <http://sourceforge.net/projects/sserver/>

[Nguyen, 93] D.Nguyen, S.Hanks, and C.Thomas. The TRUCKWORLD Manual. Technical Report TR-FIA-91-04, Dept. of Engineering, Univ. of Washington (1993)

[Ocelllo, 98] M.Ocelllo and Y.Demazeau. Modelling decision making systems using agents satisfying real time constraints.*IFAC Proc. of the 3th IFAC Symposium on Intelligent Autonomous Vehicles*. 1:51-56 (1998)

[Oller, 99a] A.Oller, J.Ll.de la Rosa. DPA2: Architecture for Co-operative Dynamical Physical Agents. *Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW'99)*. (1999)

[Oller, 99b] A.Oller, J.Ll.de la Rosa, B.Innocenti. Uso de parámetros dinámicos para mejoras la cooperación ente robots móviles. Aplicación en un convoy de dos robots. *Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI'99)*. 311-314 (1999)

[Oller, 00] A.Oller, J.Ll. de la Rosa, R. García, J.A. Ramon, A. Figueras. Micro-Robots Playing Soccer Games: a Real Implementation Based on a Multi-Agent Decision-Making Structure. *International Journal on Intelligent Control and Soft Computing. Special Issue on Soccer Robotics: Micro-Robot World Cup Soccer Tournament '97*. 6(1) (2000)

[Oller, 01] A.Oller, J.Ll. de la Rosa Inclusión de capacidades específicas para la evaluación de la eficiencia de acciones físicas. *2º Workshop Hispano-Luso en Agentes Físicos*. 157-170 (2001)

[Oller, 02] A.Oller, M.A.García. Using a Linear Model to Evaluate Performance of Non-Holonomic Paths: the Shooting-Ball as a Case Study. In Proc. of *FIRA Robot World Congress 2002*. 643-647 (2002)

[Ota, 95] J. Ota and N. Miyata and T. Arai and E. Yoshida and D. Kurabayashi and J. Sasaki. *Transferring and regrasping a large object by cooperation of multiple mobile robots*. In IEEE/RSJ IROS, pp. 543-548, (1995)

[Payton, 86] D.W.Payton. An Architecture for Reflexive Autonomous Vehicle Control. *Proceedings of the International Conference on Robotics and Automation*, pp 1838-1845. (1986)

[Papazoglou, 92] M.P. Papazoglou, S.C. Laufman, and T.K. Sellis. An organizational framework for cooperating intelligent information systems. *Journal of Intelligent and Cooperative Information Systems* 1(1):169-202 (1992)

[Parker, 92] L.E.Parker. Adaptive action selection for cooperative agent teams. *In Proc. Simulation of Adaptive Behavior* 442-450. MIT Press, December (1992)

[Parker, 93] L.E.Parker. Designing control laws for cooperative agent teams. *In Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 582-587 (1994)

[Parunak, 95] H.V.D.Parunak. Applications of distributed artificial intelligence in industry. *In Foundations of Distributed Artificial Intelligence* (eds. G.M.P.O'Hare and N.R.Jennings), Wiley (1995)

[Pauker, 76] S.G.Pauker, G.A.Gorry, W.B.Schwartz, J.P.Kassirer. Towards the simulation of clinical cognition, *American Journal of Medicine* 60: 981-996 (1976)

[Pednault, 90] E.P.D. Pednault. Formulating multiagent, dynamic-world problems in the classical planning framework. *In J. Allen and J. Hendler and A. Tate, editors, Readings in planning*, pp. 675-710, Morgan Kaufmann Publishers, San Mateo, (1990)

[Perlis, 85] D.Perlis. Languages with self reference I: Foundations. *Artificial Intelligence*, 25:301-322 (1985)

[Perlis, 88] D.Perlis. Languages with self reference II: Knowledge, Belief, and modality. *Artificial Intelligence*, 34:179-212 (1988)

[Petrie, 96] C.J.Petrie. Agent-Based Engineering, the Web, and Intelligence. *IEEE Expert*, Decembre (1996)

[Preciado, 91] A.Preciado, D.Meizel et al. Fusion of Multi-Sensor Data: a Geometric Approach. *In Proc. IEEE International Conference on Robotics and Automation*. pp. 2806-2811 (1991)

[Randall, 00] D.B.Randall. Dynamical Approaches to Cognitive Sciences. *Trends in cognitive Sciences*. 4(3): 91-99 (2000)

[Rao, 91] A.S.Rao, and M.P.Georgeff. Asymmetry thesis and side-effect problems in linear time and branching time intention logics. *In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pp. 498 -504, Sydney, Australia. (1991)

- [Rao, 93] A.S.Rao, and M.P.Georgeff. A model-theoretic approach to the verification of situated reasoning systems. *In Proceedings of IJCAI-93*, pp. 318-324, Chambéry, France. (1993)
- [Rao, 95] A.S.Rao, and M.P. Georgeff. BDI agents: From theory to practice. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pp. 312-319, Menlo Park, California, (1995)
- [Rao, 96] A.S.Rao. AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. *In W. van der Velde and J.W. Perram, 54 editors, Agents Breaking Away (LNAI 1038)*, pp. 42-55. SpringerVerlag. (1996)
- [Rees, 96] J.A.Rees. A Security Kernel Based on the Lambda-Calculus. *AI Memo n° 1564. Artificial Intelligence Laboratory. Massachusetts Institute of Technology.* (1996)
- [Rosenau, 97] J.N. Rosenau. Many Damn Things Simultaneously: Complexity Theory and World Affairs. *In Alberts, David S. and Thomas J. Czerwinski, eds. Complexity, Global Politics and National Security.* Washington, DC: National Defense University. pp. 73-100 (1997)
- [Rosenschein, 86] S.Rosenschein, and L.P.Kaelbling. The synthesis of digital machines with provable epistemic properties. In Halpern, J. Y., editor, *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, pp. 83-98. Morgan Kaufmann Publishers. San Mateo, CA. (1986)
- [Sarkar, 93] N.Sarkar, X.Yun and V.Kumar. Dynamic path following: A new control algorithm for mobile robots. *In Proc. 32nd IEEE Conference on Decision and Control*, pp. 2670-2675 (1993)
- [Scheuer, 97] A. Scheuer and Th. Fraichard. Continuous-curvature path planning for car-like vehicles. *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, 2:997-1003, (1997)
- [Schöner, 95] G.Schöner, M.Dose, and C.Engels. Dynamics of Behavior: theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16:213-245 (1995)
- [Seel, 89] N.Seel. Agent Theories and Architectures. *PhD Thesis, Surrey University, Guildford, UK.* (1989)
- [Sen, 94] S.Sen, M.Sekaran and J.Hale. Learning to coordinate without sharing information. *In Proc. AAAI.* pp. 426-431 (1994)
- [Shiller, 90] Z. Shiller and H-H. Lu. Robust computation of path constrained time-optimal motion. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 144-149 (1990)
- [Shoham, 90] Y.Shoham. 1990. Agent-oriented programming. Published at *Artificial Intelligence*, 60 (1):51-92, 1993. Previous citation: *Technical Report STAN-CS-1335-90*, Computer Science Department, Stanford University, Stanford, CA (1990)

[Shortliffe, 76] E.H.Shortliffe. Computer-Based Medical Consultations: MYCIN. Elsevier/North Holland, New York, (1976)

[Sierra, 93] C.Sierra, L.Godo. Specifying simple scheduling tasks in a reflective and modular architecture, in J. Treur and Th. Wetter eds. *Formal Specification of Complex Reasoning Systems*, Ellis Horwood Pub (in press) (1993)

[Sierra, 98] C.Sierra, P. Noriega. Institucions Electròniques, a 1r Congrés Català d'Intel·ligència Artificial, Tarragona, octubre 1998. Publicat al Butlletí de l'ACIA, Primavera-Estiu 1998, 14-15: 125-129 (1998)

[Singh, 90] M.P.Singh. Towards a theory of situated know-how. *In Proceedings of the Ninth European Conference on Artificial Intelligence (ECAI-90)*, pp. 604-609, Stockholm, Sweden (1990)

[SMI, 94] Col·lectiu S.M.I., IRIT, Univ. Paul Sabatier. Vers l'intelligence artificielle collective. *Journée multi-agent du PRC IA*. París (1994)

[Smith, 80] R.Smith. The contract net protocol: high-level communication and control in a distributed problem solver. *IEEE Trans. Computer*. pp. 1104-1113 (1980)

[Smithers, 92] T.Smithers. Taking eliminative seriously: a methodology for autonomous systems research. *Toward a Practice of Autonomous Systems (Varela, F.J. and Bourgine, P., eds.)*, MIT Press, pp. 31-40 (1992)

[Smithers, 94] T.Smithers. What the dynamics of adaptive behaviour and cognition might look like in agent-environment interaction systems. *In On Role of Dynamics and Representation in Adaptive Behaviour and Cognition DRAB'94* (1994)

[Stankovic, 88] J.A.Stankovic. Misconceptions About Real-Time Computing. *IEEE Computer*. 12(10):10-19 (1988)

[Steeb, 84] R.Steeb, D.McArthur, S.Cammarata, S.Narain, and W.Giarla. Distributed problem solving for air fleet control: framework and implementation. *The Rand Corporation, N-2139-ARPA* (1984)

[Steels, 89] L.Steels. Components of expertise. *Artificial Intelligence Magazine*. 12:38-49 (1989)

[Steels, 90] L.Steels. Cooperation between distributed agent through self organization. In Demazeau, Y. and Müller, J.-P., editors, *Decentralized AI - Proceedings of the Third European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91)*, pages 175-196. Elsevier Science Publishers B. V. Amsterdam, The Netherlands (1990)

[Steiner, 90] D. Steiner, D. Mahling, and H. Haugeneder. Human Computer Cooperative Work. *In Proceedings of the 10th AAAI International Workshop on Distributed Artificial Intelligence* (1990)

- [Stone, 98] P.Stone. Layered Learning in Multi-Agent Systems. *PhD. Thesis CMU*. School of Computer Science Dept. Carnegie Mellon University, Pittsburgh, PA, USA. Accessible com a *Technical Report CMU-CS-98-1487* (1998)
- [Suzuki, 95] T.Suzuki, K.Yokota, et al. Cooperation between the Human Operator and the Multi-Agent Robotic System: Evaluation of Agent Monitoring Methods for the Human interface. *In Proceedings of IROS'95*. pp. 206-211 (1995)
- [Tani, 99] J.Tani and S.Nolfi. Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. *Neural Networks*,12: 1331-1141 (1999)
- [Thomas, 92] S.R.Thomas. A logic for representing action, belief, capability, and intention. Stanford Working Document. Standford, CA (1992)
- [Thomas, 93] S.R.Thomas. PLACA, an Agent Oriented Programming Language. PhD Thesis, Stanford University. Accessible from Computer Science Dept. Technical Report STAN-CS-93-1487 (1993)
- [Travé, 89] L.Travé, J.L. Dormoy. Qualitative calculus and applications. *Artificial Intelligence in Scientific Computation: Towards Second Generation Systems*, pp. 53-61, IMACS'89 (1989)
- [Turner, 90] R.Turner. Truth and Modality for Knowledge Representation. *Pitman Publishing, London* (1990)
- [Ueyama, 93] T.Ueyama, and T.Fukuda. Knowledge Acquisition and Distributed Decision Making". *Proc. of IEEE Int'l Conf. on Robotics and Automation (ICRA'93)*. 3: 167-172 (1993)
- [van Gelder, 94] T.van Gelder. What might cognition be if not computation. *In On Role of Dynamics and Representation in Adaptive Behaviour and Cognition DRAB'94* . També publicat el 1995 al *Journal of Philosophy*, 91:345-381. (1994)
- [Vere, 90] S.Vere, and T.Bickmore. A basic agent. *Computational Intelligence*, 6:41-60 (1990)
- [Vivancos, 98] E.Vivancos, L. Hernández, V.J. Botti. Inteligencia Artificial Distribuida en Entornos de Tiempo Real. *Revista Iberoamericana de Inteligencia Artificial. Especial Monográfico: Inteligencia Artificial Distribuida y Sistemas Multiagentes. Vol. IX/98*. pp. 24-35 (1998)
- [Voorhees, 94] E.M.Voorhees. Software agents for information retrieval. In Etzioni O. Editor. *Software Agents - Papers from the 1994 Spring Symposium*, pp. 126-129, AAAI Press (1994)
- [Wang, 93] J.Wang. Establish a Globally Consistent Order of Discrete Events in Distributed Robotic Systems. *In Proceedings of ICRA'93*. pp. 853-858 (1993)
- [Weiss, 93] G.Weiss. Learning to coordinate actions in Multi-Agent Systems. *Proc. IJCAI-93*, pp. 311-316 (1993)

- [Weiss, 78] S.M.Weiss, C.S.Kulikowski, S.Amarel, A.Safir. A model-based method for computer-aided medical decision-making, *Artificial Intelligence* 11: 145-172 (1978)
- [Weihmayer, 94] R.Weihmayer and H.Velthuijsen. Application of distributed AI and cooperating problem solving to telecommunication. In *AI Approaches to Telecommunications and Network Management* (eds. J.Liebowitz and D.Prereau) IOS Press (1994)
- [Werger, 98] B.B.Werger. Principles of Minimal Control for Comprehensive Team Behavior. In *Proc. of the 1998 IEEE Int'l Conference on Robotics & Automation* 3504-3509 (1992)
- [Werner, 92] G.Werner, M.Dyer. Evolution of herding behavior in artificial animals. In *Proc. Simulation of Adaptive Behavior*. (1992)
- [Westhead, 95] M.D.Westhead. Synchronous Systems for Behaviour Based Robot Control. *Workshop on Mobile Robotics at AISB'95*. (1995)
- [Wielinga, 92] B.J.Wielinga, A.T.Sreiber., J.A.Breuker. KADS: A modelling approach to knowledge engineering. *Knowledge Acquisition* 4: 5-53 (1992)
- [Wittig, 91] T.Wittig. Sistemas Expertos Cooperantes. *Workshop "La inteligencia Artificial y el Control en Tiempo Real"*, Santander, 25-28 Juny (1991)
- [Wooldridge, 92] M.Wooldridge. The Logical Modelling of Computational Multi-Agent Systems. *PhD. thesis Department of Computation, UMIST, Manchester, UK* (1992)
- [Wooldridge, 93] M.Wooldridge and D.Vandekerckhove. MYWORLD: An Agent-Oriented Testbed for Distributed Artificial Intelligence. In *S.M. Deen, editor, Proceedings of the Special Interest Group on Cooperating Knowledge Based Systems 1993*. pp. 263-274. Univ. of Keele (1993)
- [Wooldridge, 94] M.Wooldridge. Coherent social action. In *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI-94)*, pp. 279-283 (1994)
- [Wooldridge, 94a] M.Wooldridge and N.R.Jennings. Agent theories, architectures and languages : A survey. *Workshop ECAI - Jennings&Wooldridge Eds*. (1994)
- [Wooldridge, 95] M.Wooldridge and N.R.Jennings. Intelligent agents: theory and practice. *Knowledge Engineering Review* 10 (2): 115-152 (1995)
- [Zang, 94] Y.Zang and A.K.Mackworth. Will the robot do the right thing? *Proc. Artificial Intelligence'94*. pp. 255-262, Banff, Alberta. May (1994)

Apèndixs

Apèndix I: Aspectes tècnics dels robots mòbils utilitzats

Apèndix II: Limitacions dinàmiques del robot mòbil

Apèndix III: Descripció del simulador PHAGSIM

Apèndix IV: Control del moviment d'un robot mòbil amb tracció diferencial

Apèndix I: Aspectes tècnics dels robots mòbils utilitzats

La plataforma de robots mòbils que s'ha utilitzat en aquest treball pertany al grup de recerca eXiT (Enginyeria de Control i Sistemes Intel·ligents) i està muntada al Departament d'Electrònica, Informàtica i Automàtica de la Universitat de Girona. Aquesta plataforma està en un procés de canvi continu donat que tots els elements que la componen han estat desenvolupats per membres del mateix Departament.

Les característiques de la plataforma responen a les exigències de recerca actuals en el camp del control de sistemes autònoms, i permet experimentar tècniques de control distribuït i d'intel·ligència artificial en un equip de petits robots mòbils.

El funcionament del sistema es basa en tres components diferenciades que s'explicaran a continuació: sistema de visió, sistema de control d'alt nivell, i els robots mòbils (figura A.1.1.).

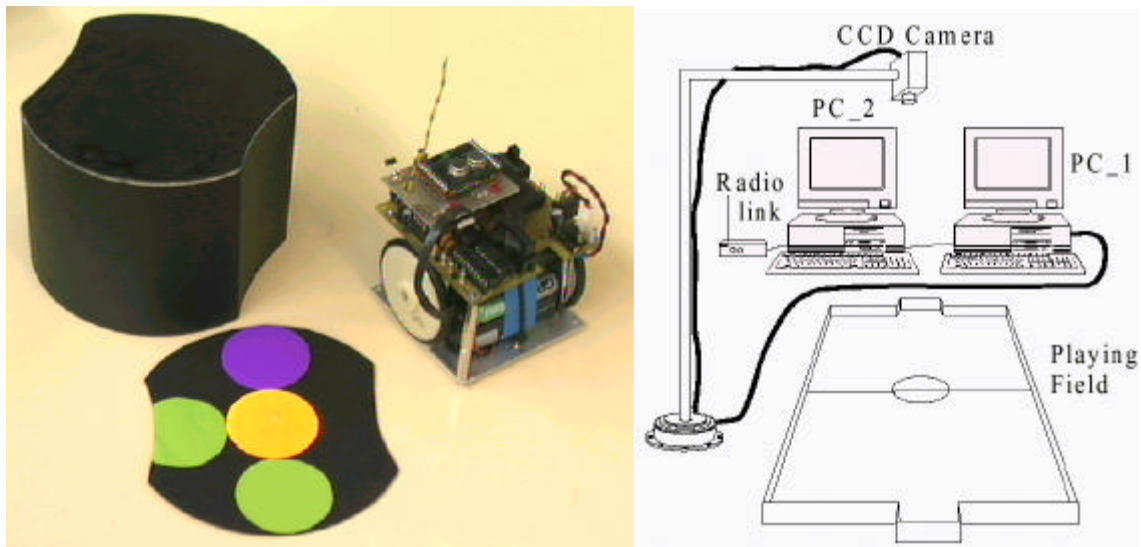


Figura A.1.1.: A l'esquerra un robot amb la carcassa i la disposició dels colors que l'identifiquen. A la dreta, un esquema on es veuen les diferents components del sistema

1. **Sistema de visió:** consisteix en un sistema que processa i analitza les imatges d'una camera en color. Les imatges de la camera consisteixen en una visió azimuthal de l'entorn en què es poden moure els robots (figures A.1.1. i A.1.2.). Aquest entorn té una forma rectangular i unes condicions de llum adequades per a poder distingir diversos colors amb molt fidelitat.

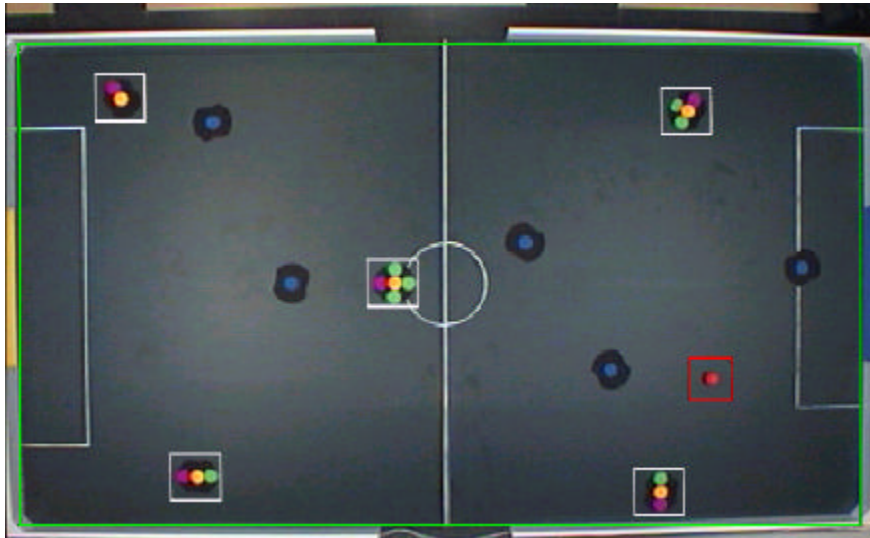


Figura A.1.2.: Exemple d'imatge típica proporcionada per la camera.

L'escena conté onze objectes mòbils: deu robots i una pilota.

El sistema està preparat per a poder fer el seguiment del moviment de fins a deu robots (cinc de color groc i cinc de blaus segons es veu a la figura A.1.2.) i el seguiment d'un altre objecte (normalment una pilota de golf taronja). Aquest sistema proporciona les coordenades (x, y, θ) per als cinc robots a controlar (els grocs) i les coordenades (x, y) per als demés objectes. Tant el processat com l'anàlisi es realitzen en temps de vídeo, és a dir, 25 imatges/s a través d'una tarja dissenyada especialment (figura A.1.3.). Més informació tècnica del funcionament d'aquest sistema pot trobar-se a [García, 98].

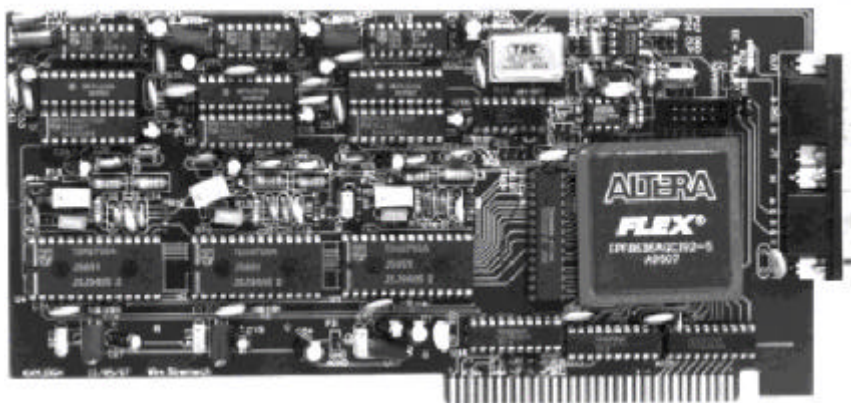


Figura A.1.3.: Imatge de la tarja que processa i analitza les imatges en color [García, 98].

2. **Control d'alt nivell:** consisteix en un sistema que obté les dades del sistema de visió (les coordenades dels onze objectes de l'escena) i s'encarrega de decidir quins són els moviments que han de fer els robots. Seguidament, aquests moviments es codifiquen en forma de trames i s'envien per ràdio als diferents robots (figura A.1.4.). Aquest

sistema està implementat en un segon ordinador (PC_2 de la figura A.1.1.). El procés de decisió dels moviments dels diferents robots s'implementa en forma de sistema multi-agent en què cada robot s'associa a un agent. Els processos de decisió individual, la negociació i la decisió final se simulen en aquest ordinador i canvien en funció del tipus d'experiment que es faci amb els robots.

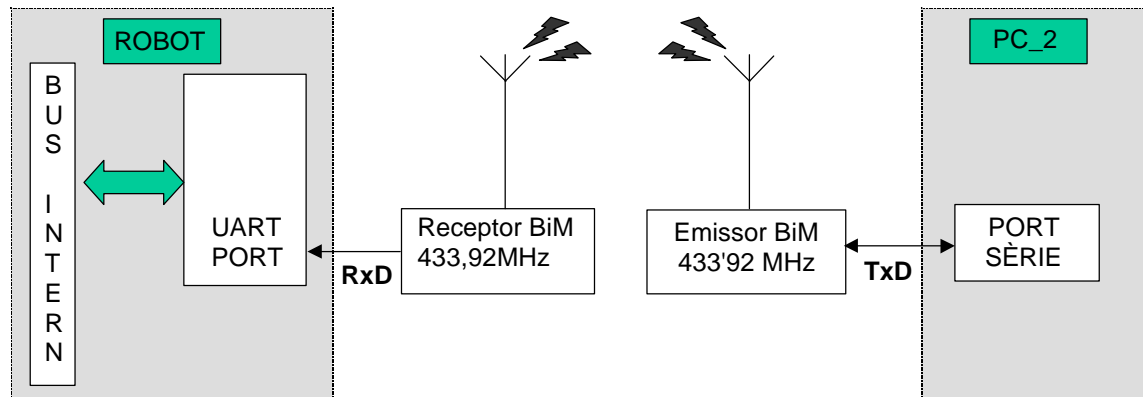


Figura A.1.4.: Esquema funcional que detalla la relació entre el control d'alt nivell i els robots.

Amb aquesta plataforma s'han realitzat experiments molt diversos: algorismes de control predictiu per controlar la posició dels robots, moviment de col·laboració entre robots per al moviment d'objectes grans, formació de convois, i emulació del joc de futbol entre dos equips de com a màxim cinc jugadors cadascun. L'experiment de l'emulació del joc de futbol és el que s'ha realitzat amb més continuïtat des de l'any 1996 fins a l'actualitat, el qual permet fer els experiments més relacionats amb el treball d'aquesta tesi.

3. **Robots mòbils:** el funcionament final d'aquesta plataforma està destinat a l'anàlisi del funcionament de sistemes autònoms, és a dir, robots mòbils. Per les dimensions de la plataforma, els robots utilitzats són de petites dimensions, ràpids de moviment, amb capacitats de càlcul i de control, i amb autonomia energètica.

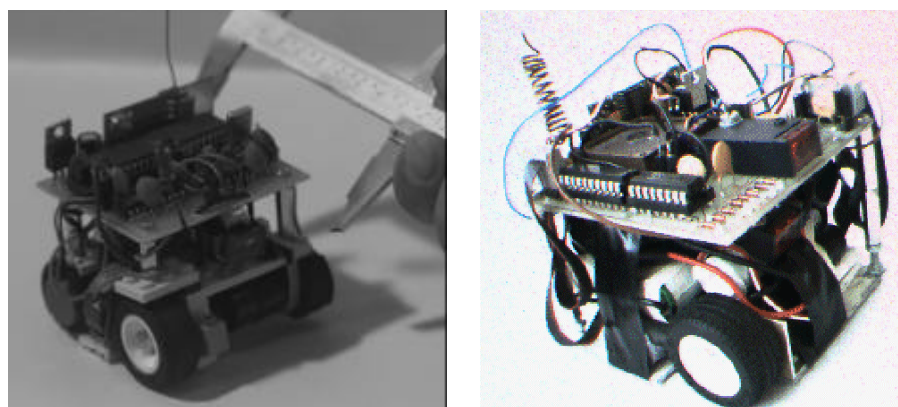


Figura A.1.5. Aspecte dels robots en les primeres generacions dels anys 1996 i 1997

A més, disposen de la capacitat de comunicar-se per ràdio, és a dir, poden rebre comandes de control que han d'executar. El disseny d'aquests robots ha anat variant amb els anys i s'ha passat d'uns primers prototipus relativament simples a uns models actuals amb alta capacitat de càlcul i precisió de moviments. A tall d'exemple, a la figura A.1.5. es mostren els primers prototipus que varen utilitzar-se per a les competicions internacionals de futbol robòtic de MiroSot (Corea del Sud i Àustria) i RoboCup (Japò) els anys 1996 i 1997.

El robot que s'ha utilitzat en aquesta tesi tant en els experiments reals com en els simulats és el que hi ha a la figura A.1.6., el qual té les següents característiques tècniques:

- CPU: 87C552 de Philips, micro-controlador de 8 bits.
- Driver L293: Dos ponts i drivers per al control de velocitat i del sentit de rotació dels motors (16 pins). Corrent de sortida 1 A.
- Sensors: Dos detectors fotoelèctrics reflex E3S de OMROM. Distància de detecció ajustable.
- Encoders: Dos codificadors incrementals magnètics amb 265 polsos per volta, solidaris als eixos dels motors.
- Receptor: Radio-transmissor de FM, Radiometrix BiM a 433,92 MHz.
- Bateries : 10 bateries de 1.2V 1300 mA/h (12 V)
- Consum total robot: 600mA/h
- Driver de Potència L298B: Alimentació de potència: 12 V (màx: 36 V). Alimentació digital: 5 V (màx: 36 V). Corrent màxim constant de sortida: 1 A.
- Motors: Dos motors MAXON DC. 9V de 1.5W, velocitat nominal de 13.200 rpm, amb reductor (16 :1).

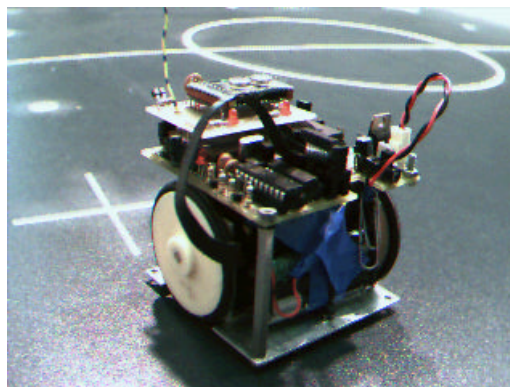


Figura A.1.6. Aspecte del robot utilitzat en els experiments reals d'aquesta tesi

El tipus de tracció dels robots és de tipus diferencial, és a dir, consta de dues rodes independents amb capacitat motriu però no directriu. Per tant, si les rodes giren a la mateixa velocitat el robot es mourà en línia recta, i si giren a velocitats diferents es mourà seguint un arc de circumferència.. Per raons d'estabilitat del centre de masses del robot, s'han posat dos pivots que ajuden a mantenir-lo en posició horitzontal exercint un mínim de fregament amb el terra: en termes mecànics, els pivots es corresponen a dues boles boges, és a dir, no afecten ni a la cinemàtica ni a la dinàmica del robot.

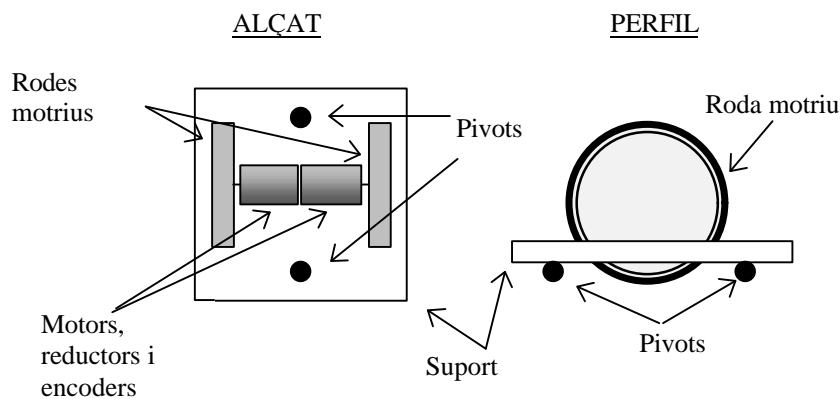


Figura A.1.7.: Esquema de la base del robot. S'observa un tipus de tracció diferencial

Per a les proves simulades en aquesta tesi, s'han tingut en compte diversos aspectes del robot real:

Model realista del motor: a partir dels paràmetres del fabricant del motor, es pot obtenir un model bastant aproximat de quina és la velocitat de gir del motor a partir d'un voltatge d'excitació. A part, el model es pot afinar més incloent no linealitats com són els valors màxims de velocitat, i la zona morta. També s'ha tingut en compte la relació d'engranatges del reductor.

Dimensions del robot: els radis de les rodes, la separació entre elles, i les dimensions del robot també s'han tingut en compte.

Dinàmica de la pilota: el fregament de la pilota amb el terra dóna lloc a un comportament de primer ordre per a la velocitat de la pilota. També s'ha tingut en compte que el rebot de la pilota amb les parets no és elàstic. Finalment, també cal comentar que el xoc de la pilota amb els robots és completament inelàstic fet que dóna lloc a una dinàmica molt especial: en el procés del xut de la pilota la direcció d'aquesta després del xut és la mateixa del robot, és a dir, el xut es produeix en forma d'impuls.

Apèndix II: Limitacions dinàmiques del robot mòbil

Sigui A un robot mòbil amb tracció diferencial que es mou a través d'un camí S lliure de col·lisió, és a dir, els obstacles de l'entorn són estacionaris (figura A.2.1).

Se suposa que el moviment es realitza en el pla \mathcal{R}^2 .

Les dues rodes del mòbil són motrius i la diferència de velocitats entre elles marca la tendència de gir del mòbil seguint un arc de circumferència de radi ρ .

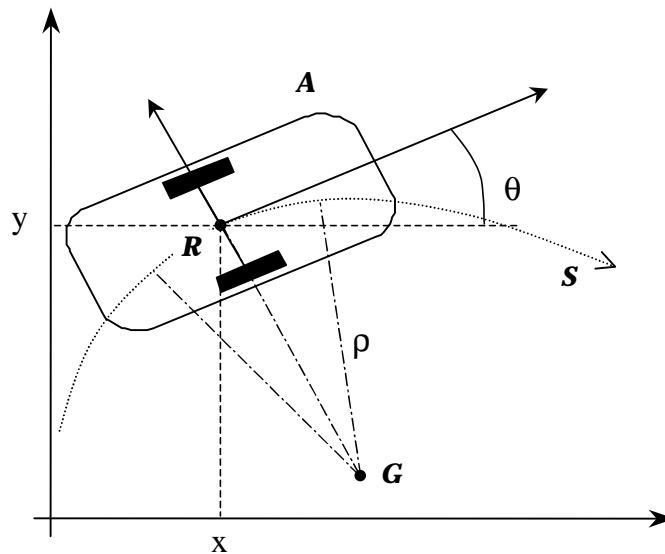


Figura A.2.1.: Diagrama de les variables que defineixen un robot mòbil amb tracció diferencial

La configuració del mòbil es defineix de forma unívoca per la terna $(x, y, \theta) \in \mathcal{R}^2 \times [0, 2\pi[$ on (x, y) són les coordenades de \mathbf{R} (el punt mig d'ambdues rodes) i θ és l'orientació del mòbil.

Un mòbil movent-se en el pla té un centre de rotació \mathbf{G} . Suposant que el moviment s'executa només per tracció de les rodes, cada roda es mou en una direcció normal al seu eix de rotació. Per tant, quan un mòbil es mou, els eixos de les seves rodes intersecten al centre de rotació \mathbf{G} , mentre que \mathbf{R} es mou en una direcció normal a l'eix de les rodes. És a dir, es compleix la següent relació:

$$\tan\theta = \frac{\dot{y}}{\dot{x}} \quad (1)$$

Com que hi ha una separació entre les rodes, resulta evident que el radi de curvatura de S al punt \mathbf{R} està limitat per la següent relació:

$$\rho \geq \rho_{\min} \tag{2}$$

Les relacions (1) i (2) són no-holonòmiques [Barraquand, 89].

El camí que segueix un mòbil consisteix en una seqüència de configuracions, és a dir, una corba al $(xy\theta)$ -espai que ha de verificar les relacions (1) i (2). De fet, les característiques de la corba són més simples donat que (1) estableix una relació entre la terna (x,y) i θ . Així, el camí que segueix un mòbil consisteix en una corba g que segueix el punt R en un (xy) -espai.

Si es considera el cas d'un mòbil amb velocitats altes, la corba g és de classe C^2 (diferenciable 2 vegades i amb la segona derivada contínua) i el màxim radi de curvatura és de $1/\rho_{\min}$. La propietat C^2 assegura que el camí és realitzable sempre i quan se suposi que el mòbil no s'aturi ni canviï de sentit de moviment [Scheuer, 97]

La dinàmica del model del robot

A es pot modelar com un cos rígid aguantat per dues rodes. A efectes d'estabilitat es pot suposar que hi ha una tercera "roda boja". Sense pèrdua de generalitat, es pot assumir que l'eix \vec{t} solidari a A coincideix amb el vector unitari i tangent al camí S en el punt R . L'eix \vec{b} està dirigit cap a la direcció positiva i normal al pla. L'eix \vec{n} s'escull de manera que la terna de vectors $(\vec{t}, \vec{n}, \vec{b})$ compleix la regla de la mà dreta. Es pot observar que la línia que marca el radi de curvatura al punt R coincideix amb \vec{n} .

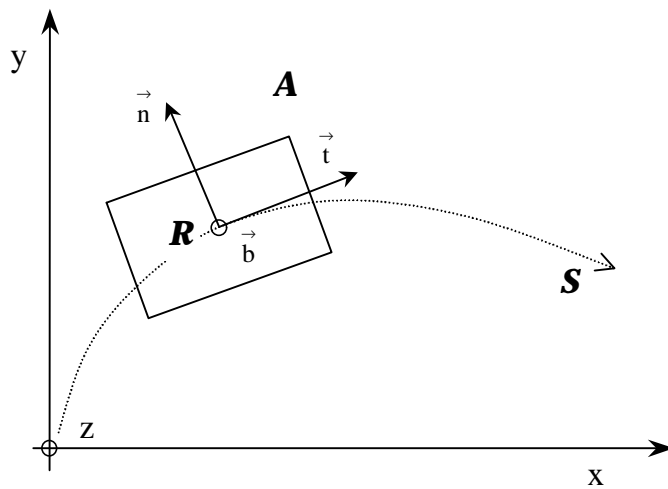


Figura A.2.2.: Sistema de vectors que es defineixen per establir les equacions dinàmiques

El moviment de **A** al llarg de **S** obeeix la dinàmica Newtoniana. Les forces externes que actuen sobre **A** són la força de la gravetat \vec{G} i la força de reacció del terra \vec{R} que es pot descomposar en les seves components ortogonals:

$$\begin{aligned}\vec{G} &= -mg \vec{b} \\ \vec{R} &= R_t \vec{t} + R_n \vec{n} + R_b \vec{b}\end{aligned}\tag{3),(4}$$

on m és la massa de **A** i g és l'acceleració de la gravetat. L'equació del moviment de **A** es pot expressar en funció de la velocitat tangencial \dot{s} i de l'acceleració tangencial \ddot{s} de la següent manera:

$$\vec{G} + \vec{R} = m \ddot{s} \vec{t} + m \kappa_s \dot{s}^2 \vec{n}$$

on κ_s és la curvatura amb signe del camí a la posició s (κ_s és positiu si la direcció radial coincideix amb \vec{n} , és a dir, $-1/\rho_{\min} \leq \kappa_s \leq 1/\rho_{\min}$). Usant (3) i (4), l'equació de moviment es pot reescriure així:

$$\begin{aligned}R_t &= m \ddot{s} \\ R_n &= m \kappa_s \dot{s}^2 \\ R_b &= mg\end{aligned}\tag{5)(6)(7}$$

Aquestes equacions representen les forces que es requereixen per a mantenir la velocitat \dot{s} i l'acceleració \ddot{s} de **A** en una posició s del camí. Aquest model [Shiller, 90] és relativament simple però és suficient per mostrar quines són les limitacions i el caràcter dinàmic d'un robot mòbil.

Les restriccions dinàmiques del robot

En aquesta secció només es tindran en compte la limitació de la força motriu, la restricció que evita que les rodes llisquin, i la limitació de la velocitat tangencial.

Força motriu: Quan el robot es mou, el parell aplicat pels motors a les rodes es converteix a una força planar \vec{F} paral·lela a \vec{t} i de mòdul $m \ddot{s}$. Aquesta força està limitada de la següent manera:

$$F_{\min} \leq F \leq F_{\max}\tag{8}$$

Restricció de lliscament: La component de \vec{R} en el pla $\vec{t} \times \vec{n}$ representa el fregament que actúa del terra cap a les rodes. Aquest fregament ha de complir la següent relació

$$\sqrt{R_t^2 + R_n^2} \leq \mu R_b \quad (9)$$

on μ és el coeficient de fregament entre les rodes i el terra. Si aquesta restricció no es compleix aleshores el robot se separaria del camí.

Velocitat tangencial: Una de les restriccions més importants per al control de la velocitat del mòbil és la limitació de la velocitat \dot{s} :

$$0 \leq \dot{s} \leq \dot{s}_{\text{màx}} \quad (10)$$

on $\dot{s}_{\text{màx}}$ és la velocitat màxima permesa.

Si l'equació (8) es reescriu en forma d'acceleracions s'obté el següent:

$$\frac{F_{\text{mín}}}{m} \leq \ddot{s} \leq \frac{F_{\text{màx}}}{m} \quad (11)$$

Si les equacions (5), (6) i (7) es substitueixen a la (9) i s'aïlla l'acceleració, la relació que expressa el marge de valors de l'acceleració que compleixen la restricció de lliscament és:

$$-\sqrt{\mu^2 g^2 - \kappa_s^2 \dot{s}^4} \leq \ddot{s} \leq \sqrt{\mu^2 g^2 - \kappa_s^2 \dot{s}^4} \quad (12)$$

Per tant, el marge de valors correctes de l'acceleració vindrà donat per les condicions (11) i (12), és a dir:

$$\max\left(\frac{F_{\text{mín}}}{m}, -\sqrt{\mu^2 g^2 - \kappa_s^2 \dot{s}^4}\right) \leq \ddot{s} \leq \min\left(\frac{F_{\text{màx}}}{m}, \sqrt{\mu^2 g^2 - \kappa_s^2 \dot{s}^4}\right) \quad (13)$$

Els arguments de les arrels quadrades a l'expressió (12) han de ser positius i quan $\kappa_s \neq 0$, la velocitat haurà de complir el següent:

$$-\sqrt{\frac{\mu g}{|\kappa_s|}} \leq \dot{s} \leq \sqrt{\frac{\mu g}{|\kappa_s|}} \quad (14)$$

Per tant, el marge de valors correctes de la velocitat vindrà donat per les condicions (10) i (14), és a dir:

$$0 \leq \dot{s} \leq \min\left(\dot{s}_{\text{màx}}, \sqrt{\frac{\mu g}{|\kappa_s|}}\right) \quad (15)$$

Apèndix III: Descripció del simulador PHAGSIM

Per a la obtenció de dades de l'experiment de la passada de la pilota s'ha utilitzat un programa de simulació desenvolupat pel mateix autor d'aquesta tesi¹. Aquest simulador ha estat realitzat íntegrament en l'entorn MATLAB/SIMULINK (ver.4.2.c/ver.1.3) sota Windows 98. El simulador permet realitzar experiments amb robots mòbils i proporciona una interfície gràfica que permet a l'usuari accedir a totes les utilitats que incorpora el simulador i que són les següents:

→ Permet simular el comportament d'un o més robots mòbils de dimensions reduïdes (alguns decímetres de diàmetre) en un entorn simulat que consisteix en un rectangle pla de 130x90 cm delimitat per parets (figura A.3.1).

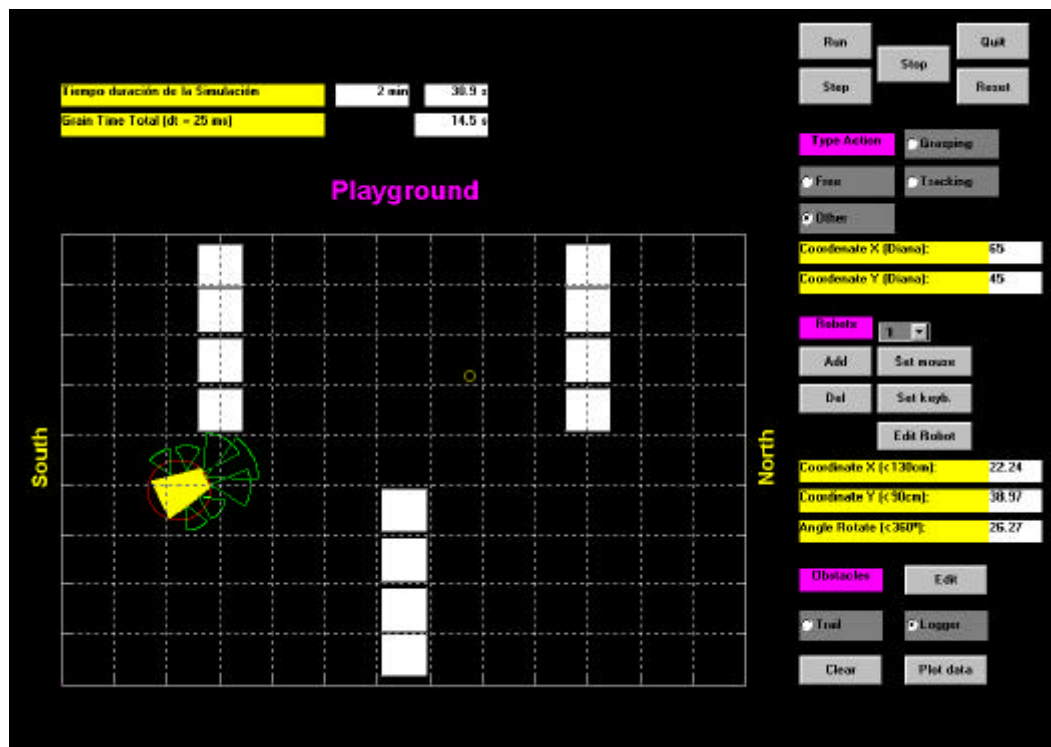


Figura A.3.1: Pantalla principal del simulador de robots mòbils PHAGSIM

→ Els robots són completament configurables gràcies a la pantalla d'edició gràfica de les característiques dels robots (figura A.3.2): mides (la forma no), nombre, col·locació i orientació de sensors damunt del robot. El simulador ja disposa d'una col·lecció de diversos tipus de robots equipats amb diversos sensors. Cal comentar que la

¹ La versió actual del simulador és el resultat d'un treball fi de carrera, i és accessible a Internet a la següent adreça: <http://www.etse.urv.es/~aoller/phagsim/PHAGSIM.htm>

simulació dels xocs del robot sempre té en compte que les dimensions del robot corresponen a les del cercle que inscriu al robot.

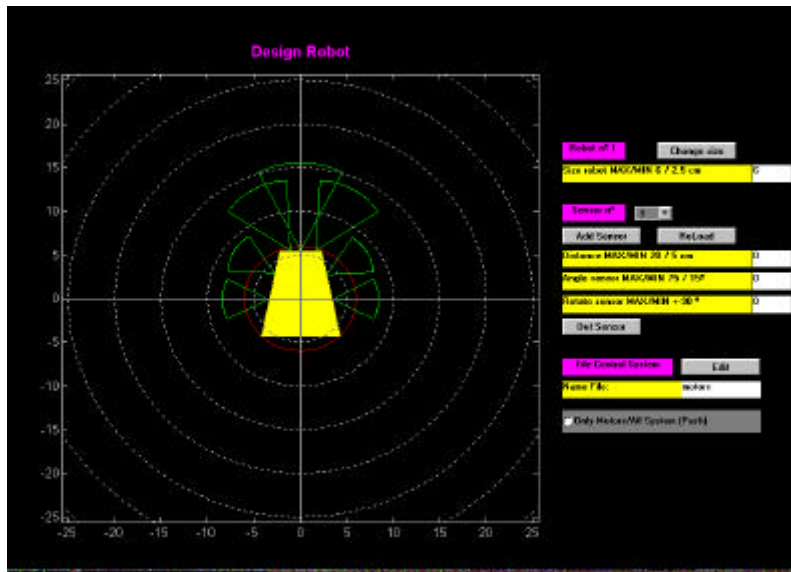


Figura A.3.2: Pantalla d'edició del robot mòbil

→ Els sensors també són configurables. En concret es poden especificar l'angle i la distància de detecció de cadascun d'ells. Cal comentar que el tipus de sensor és únic i correspon a un sensor detector de presència, és a dir, no proporciona cap mesura relativa a la distància.

→ L'entorn també és configurable, és a dir, es poden incorporar un nombre elevat d'obstacles de diverses formes (figura A.3.3). El simulador ja disposa de diversos entorns preparats (passadissos, columnes, barreres, etc.). Cal comentar que la simulació dels xocs té en compte que les dimensions de l'obstacle corresponen a les exactes de l'obstacle.

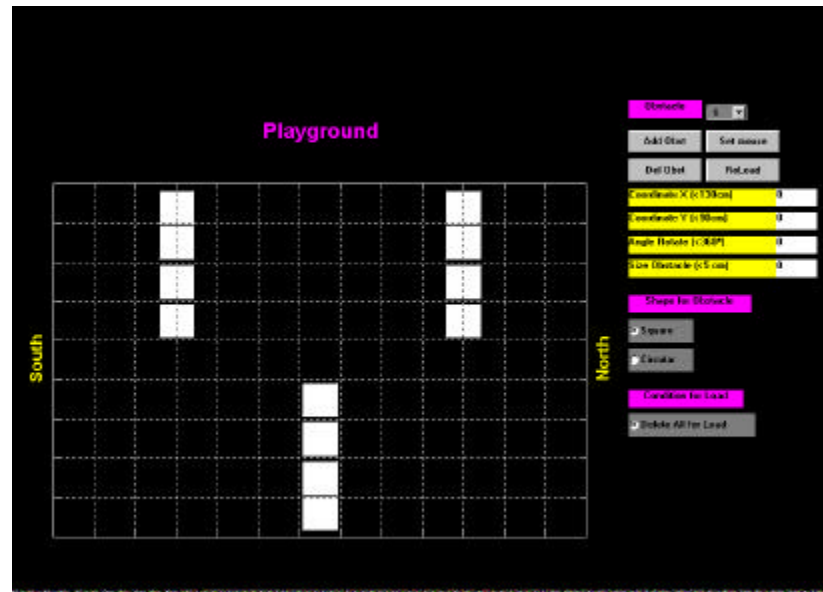


Figura A.3.3: Pantalla d'edició d'obstacles

→ La tracció dels robots és de tipus diferencial (dos rodes motrius independents i fixes), i permet obtenir gràfics sobre dades dels motors (corrents, rpm, voltatges, etc), i coordenades i velocitats dels robots

La concepció modular i oberta del simulador permet realitzar canvis en diversos aspectes de configuració que estan donats per defecte. Així, permet canviar tant el sistema de control de velocitat com el de posició dels robots, i és una tasca relativament ràpida i senzilla donat que el mateix entorn de MATLAB està equipat de *toolboxes* que incorporen eines molt útils (*fuzzy*, xarxes neuronals, control, etc.). Gràcies a la modularitat, el simulador s'ha utilitzat per a simular sistemes multi-robot, programar comportaments amb màquines d'estats, analitzar diferents algorismes de planificació de trajectòries, i monitoritzar i controlar un robot real a través de comunicació sèrie.

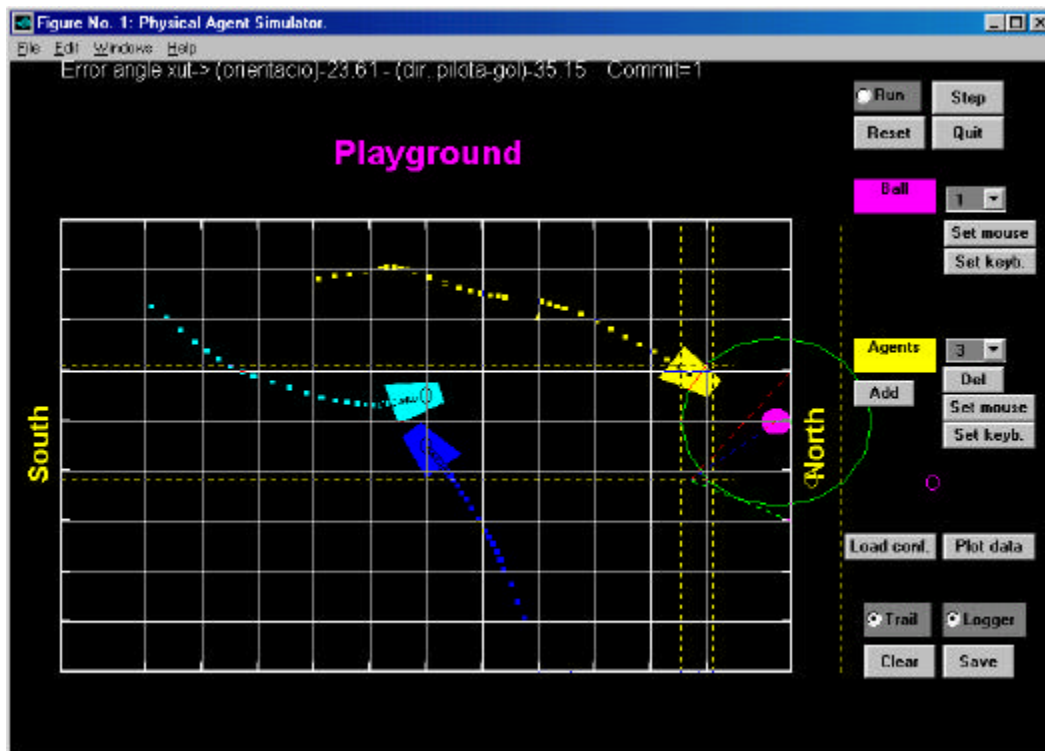


Figura A.3.4: Pantalla principal de PHAGSIM per a l'obtenció de dades. Exemple amb tres agents negociant els moviments a fer en una passada de pilota.

Per a la realització de l'experiment de la passada de la pilota, s'han desactivat diverses utilitats i s'han introduït diverses modificacions per a poder reproduir un entorn adequat. En concret, s'han eliminat tots els obstacles, s'ha afegit una pilota amb una dinàmica que reproduïx el moviment d'una pilota real, i s'han afegit una parell de porteries (figura A.3.4). Quant als robots, s'ha introduït un model realista pels motors, s'ha implementat un control de velocitat semblant al que hi ha als robots reals, s'han configurat els robots amb unes dimensions semblants a les dels robots reals i no s'hi ha posat cap sensor.

L'ús del simulador ha permès realitzar els experiments relacionats amb el problema de la passada de la pilota. Concretament, ha permès fer l'estudi de diversos controladors de posició del robot utilitzant tant l'assignació de consignes a alt nivell mitjançant màquines d'estats, com tancant el llaç de control a baix nivell amb diferents controladors. Tantmateix, proporciona les característiques essencials de l'agent que xuta la pilota, és a dir, trajectòria anterior al xut i els valors de les velocitats lineal i angular del robot (figura A.3.5).

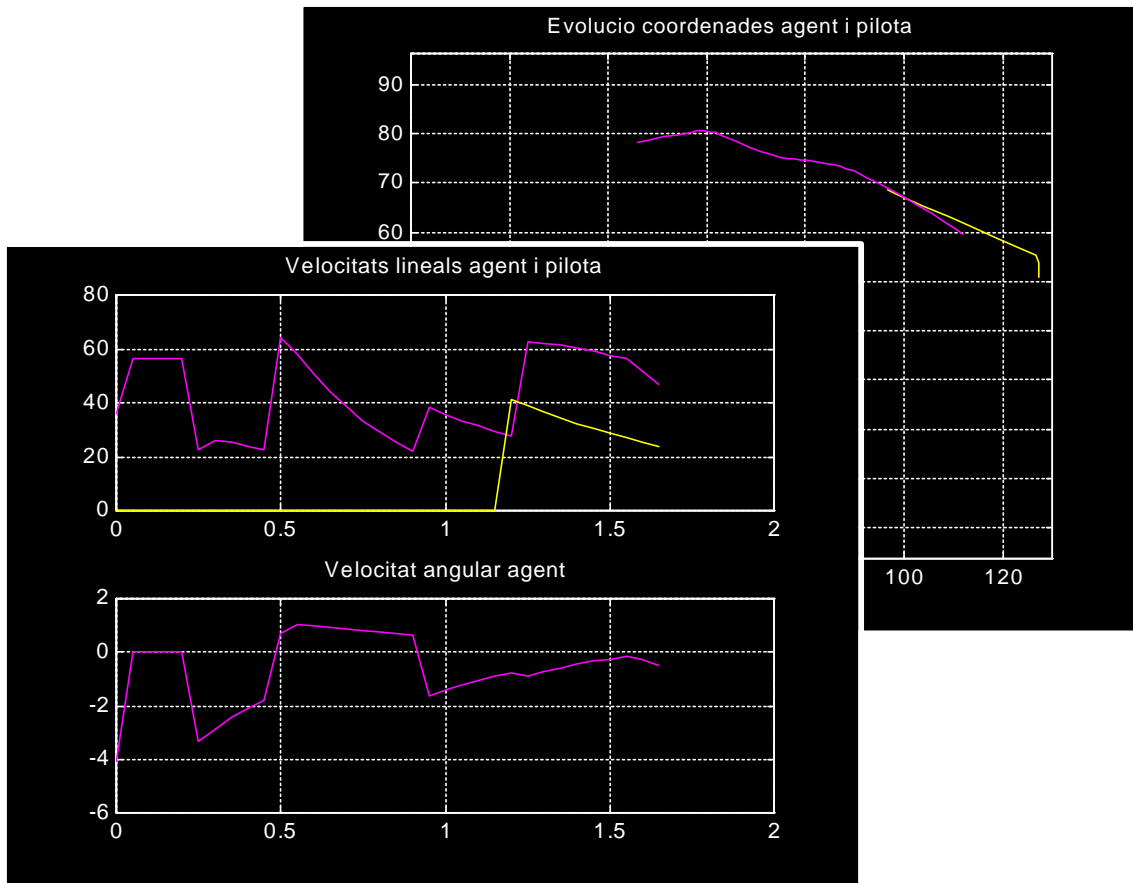


Figura A.3.5: Pantalles que mostren les característiques de l'agent que xuta la pilota

Apèndix IV: Control de moviment d'un robot mòbil amb tracció diferencial

Les equacions que descriuen la cinemàtica d'un robot mòbil que es desplaça a una velocitat lineal v i una velocitat angular ω són les següents:

$$\begin{pmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{pmatrix} = \begin{pmatrix} v \cdot \cos \theta_r \\ v \cdot \sin \theta_r \\ \omega \end{pmatrix} \quad (1)$$

on x_r i y_r representen la posició del robot respecte d'un sistema de coordenades fix, i θ_r és l'angle determinat per la direcció del robot respecte l'eix horitzontal (veure figura A.4.1). Suposem que es vol controlar el moviment del robot de forma que vagi seguint l'eix horitzontal $(x_r, 0)$ amb una velocitat lineal constant v .

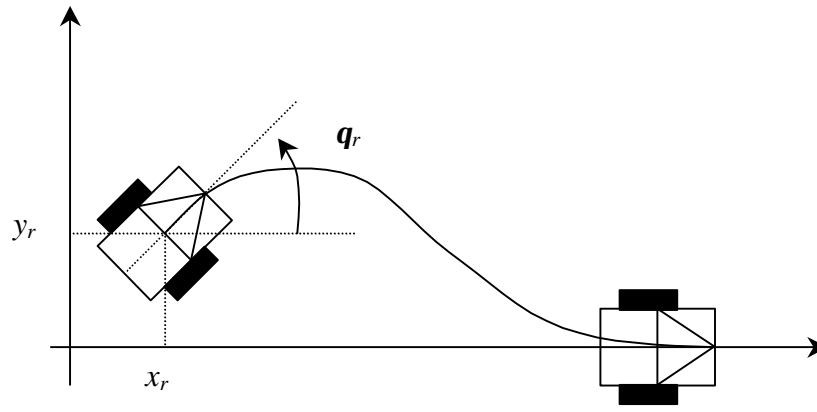


Figura A.4.1: Definició de variables d'un robot mòbil

Per fer aquest control trobarem quina ha de ser la velocitat angular en funció de la posició del robot per tal que el moviment sigui el desitjat, és a dir, el robot tendirà a moure's seguint l'eix horitzontal $(x_r, 0)$. Per tant, obtindrem la llei de control o funció $\mathbf{w}(y_r, \theta_r)$ a cada instant de temps que fa que la terna de valors (y_r, θ_r) tendeixi a $(0, 0)$.

Podem rescriure les equacions cinemàtiques de la següent manera:

$$\dot{x} = F(x, \omega), \quad \text{on } x = (y_r, \theta_r)^T, \quad F(x, \omega) = (v \sin \theta_r, \omega)^T \quad (2)$$

Els punts d'equilibri del sistema $(X_{eq}, \mathbf{w}_{eq})$ són aquells punts x i \mathbf{w} en què la dinàmica s'anul·la, és a dir:

$$X_{eq} = \left(\begin{matrix} Y_{eq} \\ \Theta_{eq} \end{matrix} \right) = \left(\begin{matrix} Y_{eq} \\ \{0, \pm\pi\} \end{matrix} \right) \quad \omega_{eq} = 0$$

Desenvolupant la funció F en sèrie de Taylor a l'entorn del punt d'equilibri $(X_{eq}, \mathbf{w}_{eq})$ s'obté la següent expressió per a $F(x, \mathbf{w})$:

$$F(x, \omega) = F(X_{eq}, \omega_{eq}) + \frac{\partial F}{\partial x} \Big|_{(X_{eq}, \omega_{eq})} (x - X_{eq}) + o(x - X_{eq}) + \frac{\partial F}{\partial \omega} \Big|_{(X_{eq}, \omega_{eq})} (\omega - \omega_{eq}) + o(\omega - \omega_{eq})$$

on $\left(\frac{\partial F}{\partial x}\right)$ és la matriu Jacobiana de F , $\left(\frac{\partial F}{\partial \omega}\right)$ és un vector columna, i els vectors columna $o(x - X_{eq})$ i $o(\omega - \omega_{eq})$ són termes menyspreables respecte de $(x - X_{eq})$ i $(\omega - \omega_{eq})$ respectivament. Com que $F(X_{eq}, \omega_{eq}) = 0$, el comportament del sistema es pot escriure de la següent manera:

$$F(x, \omega) = F(X_{eq}, \omega_{eq}) + \frac{\partial F}{\partial x} \Big|_{(X_{eq}, \omega_{eq})} (x - X_{eq}) + \frac{\partial F}{\partial \omega} \Big|_{(X_{eq}, \omega_{eq})} (\omega - \omega_{eq})$$

Per tant, com que en el nostre problema de control tenim que el punt d'equilibri és l'origen, la dinàmica del sistema linealitzant la tangent ve donada per les següents equacions:

$$\begin{pmatrix} \dot{y}_r \\ \dot{\theta}_r \end{pmatrix} \approx \begin{pmatrix} 0 & v \\ 0 & 0 \end{pmatrix} \begin{pmatrix} y_r \\ \theta_r \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \omega \quad (3)$$

Des del punt de vista de la teoria clàssica de control de sistemes lineals, aquestes equacions són una representació en l'espai d'estat del comportament del moviment del robot mòbil. Concretament, es pot demostrar que el sistema és controlable i que es pot trobar una llei de control que garanteix una evolució del sistema cap al punt d'equilibri. A continuació anem a veure que el comportament de la variable y_r respon a una dinàmica de segon ordre i, per tant, es pot caracteritzar el seu comportament a partir de la col·locació de dos pols.

Anem a suposar que la llei de control és la següent:

$$\omega = -\left(\frac{\alpha_1 \alpha_2}{v}\right) \cdot y_r - (\alpha_1 + \alpha_2) \cdot \theta_r$$

A partir de les equacions (3) obtenim que:

$$\ddot{y}_r + (\alpha_1 \alpha_2) \dot{y}_r = -(\alpha_1 + \alpha_2) v \theta_r$$

Per tant, podem escriure finalment l'equació de segon ordre corresponent al comportament de l'angle θ_r en funció dels paràmetres α_1 i α_2 :

$$\ddot{\theta}_r + (\alpha_1 + \alpha_2) \dot{\theta}_r + (\alpha_1 \alpha_2) \theta_r = 0$$

Aquesta equació diferencial lineal de segon ordre descriu com evoluciona la variable θ_r en funció del temps i dels dos paràmetres. És fàcil veure que els dos paràmetres són les arrels de l'equació característica i, per tant, marquen clarament quin

és el comportament del sistema. L'equació diferencial es pot resoldre utilitzant algunes propietats de la transformada de Laplace i s'obté la següent expressió:

$$\theta_r(s) = \frac{\theta_r(0) \cdot (s + \alpha_1 \alpha_2) + \dot{\theta}_r(0)}{s^2 + (\alpha_1 + \alpha_2)s + \alpha_1 \alpha_2} \quad (4)$$

Per tant, l'equació característica d'aquest sistema és la següent,

$$s^2 + (\alpha_1 + \alpha_2)s + \alpha_1 \alpha_2 = 0,$$

i es pot escriure de la següent manera:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = (s + s_1) \cdot (s + s_2) = 0$$

$$, \text{ on } s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$$

És a dir, l'equació correspon a la d'un sistema de segon ordre. La solució del sistema es pot descriure a partir de la discussió dels valors dels paràmetres ζ i ω_n , o a partir de les relacions entre les arrels s_1 i s_2 (pols).

A partir de l'equació (4) s'obté la següent solució analítica:

$$\theta_r(t) = e^{-\frac{\zeta\omega_n t}{\sqrt{1-\zeta^2}}} \sin(\omega_n t \sqrt{1-\zeta^2} + \arccos \zeta)$$

El paràmetre ζ s'anomena **coeficient d'esmoreïment** i el paràmetre ω_n s'anomena **frequència natural**. Tot i que l'equació diferencial té la solució analítica anterior, en teoria clàssica de control se sol classificar el comportament de $\theta_r(t)$ en tres tipus segons el valor que pren el paràmetre ζ :

0 < z < 1 → Comportament sub-esmoreït. El sistema presenta unes oscil·lacions que s'esmoreeixen de forma exponencial. La freqüència de les oscil·lacions és ω_n . Els pols són complexa conjugats.

z = 1 → Comportament críticament esmoreït. El sistema ja no presenta oscil·lacions, i s'esmoreeix de forma exponencial. Els pols són iguals, reals i negatius.

z > 1 → Comportament sobre-esmoreït. El sistema presenta un comportament d'atenuació exponencial. Els pols són diferents, reals i negatius.

A partir dels gràfics de la figura A.4.2 pot observar-se que hi ha alguns punts característics a la resposta. En concret ens fixarem en dos punts d'aquests donat que són importants a l'hora d'especificar quin ha de ser el comportament desitjat de la resposta. A partir del punt que defineix el primer màxim a la corba, es pot definir el **sobre-nivell màxim** (S_p):

$$S_p = q_{r\text{màx}} \cdot 100\%$$

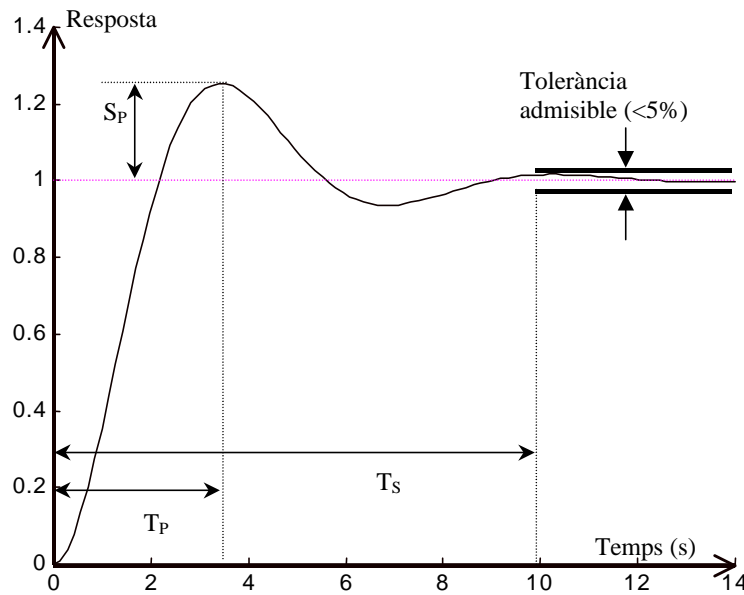


Figura A.4.2: Resposta típica d'un sistema de segon ordre per a valors de $z < 1$

El valor de S_P s'utilitza com indicador de l'estabilitat relativa d'un sistema: normalment no s'aconsellen valors alts de S_P . El valor de S_P també pot obtenir-se a partir del paràmetre ζ amb la següent expressió:

$$S_P = \mathbf{q}_r(0) \cdot \exp\left(-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right) \quad (5)$$

L'altre punt fa referència a l'instant de temps en què la resposta es considera que ha deixat d'oscil·lar, és a dir, l'instant de temps en què el sistema passa del règim estacionari al règim permanent. Aquest valor de temps s'anomena **temps d'establiment** (T_S) i es calcula imposant a la resposta que l'amplitud d'oscil·lació sigui inferior al 5%. Es pot demostrar que una bona aproximació pel càlcul de T_S és la següent:

$$T_S \cong \frac{4}{\zeta\omega_n} \quad (6)$$

Vistos aquests resultats, es pot concloure que és possible realitzar el control del moviment d'un robot mòbil amb tracció diferencial mantenint la seva velocitat lineal v constant i variant la velocitat angular ω seguint la següent **lleï de control**:

$$\omega = -\left(\frac{\alpha_1\alpha_2}{v}\right) \cdot y_r - (\alpha_1 + \alpha_2) \cdot \theta_r \quad (7)$$

Els paràmetres de la lleï de control α_1 i α_2 es poden escollir de forma apropiada per tal que el comportament de segon ordre de la variable θ_r sigui el desitjat. Aquest comportament està caracteritzat pels paràmetres ζ i ω_n , que es poden rescriure en funció dels paràmetres α_1 i α_2 de la següent manera:

$$\begin{cases} \zeta = \frac{1}{2} \left(\frac{1}{\alpha_1} + \frac{1}{\alpha_2} \right) \sqrt{\alpha_1 \alpha_2} \\ \omega_n^2 = \alpha_1 \alpha_2 \end{cases}$$

Els valors de ζ i ω_n també poden obtenir-se a partir de les anomenades especificacions de disseny (equacions 5 i 6), és a dir, les limitacions s'imposen a la dinàmica del sistema. Normalment aquestes limitacions s'expliciten numèricament posant valors màxims als valors de T_S i S_P .

Finalment, el diagrama de blocs funcionals del sistema de control de posició del robot és el següent:

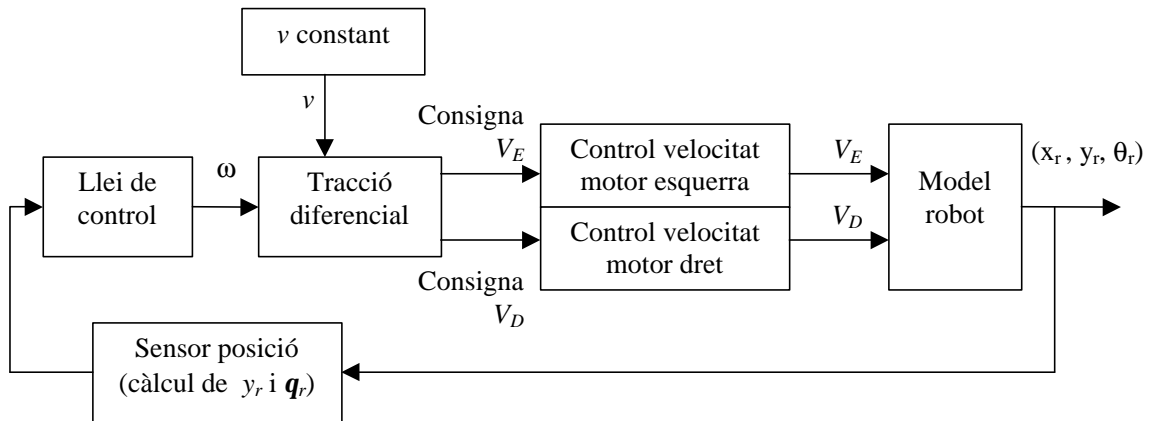


Figura A.4.3: Diagrama de blocs del sistema de control de posició del robot mòbil

A la dreta del diagrama s'observa com el robot es mou a partir de les velocitats de les dues rodes. Aquestes velocitats són les que realment tenen les rodes esquerra i dreta tenint en compte que cada roda per separat disposa d'un sistema independent de control de velocitat. Aquests dos controladors necessiten com a entrada les consignes de velocitat que es generen a partir de la llei de control i del valor constant de la velocitat lineal. L'element sensor de posició és l'encarregat d'obtenir els valors de y_r i θ_r de forma indirecta a partir de mesures basades en qualsevol tipus d'instrument d'auto-localització absoluta o relativa, o bé basant-se en les mesures realitzades en els dos llaços de control de velocitat.

A continuació es mostren diversos resultats en relació a aquesta llei de control de posició del robot mòbil. En concret la figura A.4.4 mostra les corbes de la resposta temporal de l'orientació relativa θ_r per a diferents valors de ζ i ω_n , és a dir, dels pols α_1 i α_2 . Aquestes corbes s'han obtingut a partir de l'equació (4) i considerant que $\dot{\theta}_r(0) = 0$ i $\theta_r(0) = 90^\circ$. Pot observar-se com l'augment del coeficient d'esmoreïment fa disminuir l'amplitud de l'oscil·lació màxima (S_P) i fa disminuir el temps d'establiment (T_S), tal

com era d'esperar. També es pot observar com l'augment del coeficient d'esmoreïment per sobre de la unitat fa augmentar el temps d'establiment.

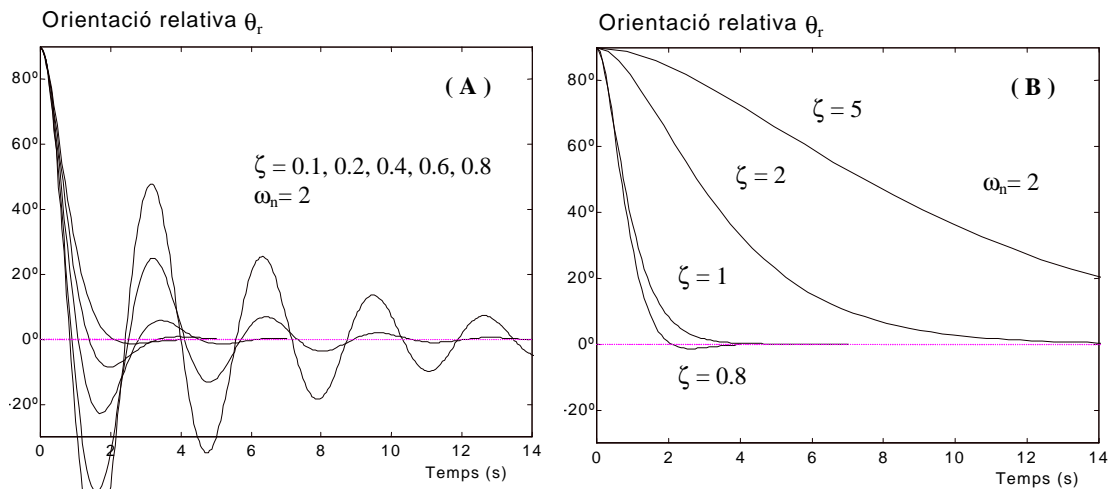


Figura A.4.4: Resposta temporal de l'orientació relativa: (A) $q_r(0)=90$, (B) $q_r(0)=90$.

Respecte a l'estabilitat del controlador, es pot demostrar que està completament garantida a partir de l'aplicació del criteri d'estabilitat de Lyapunov i que la convergència de y_r cap a 0 està garantida quan $\theta_r \rightarrow 0$. A partir de les equacions (3) i (4) s'obté la següent funció de transferència corresponent a y_r :

$$y_r(s) = \frac{y_r(0) \cdot (s + \alpha_1 + \alpha_2)}{s^2 + (\alpha_1 + \alpha_2)s + \alpha_1\alpha_2}$$

Es pot comprovar a partir d'aquesta funció de transferència que la dinàmica de $y_r(t)$ és del mateix tipus que la de $\theta_r(t)$, i així ho mostren els següents gràfics:

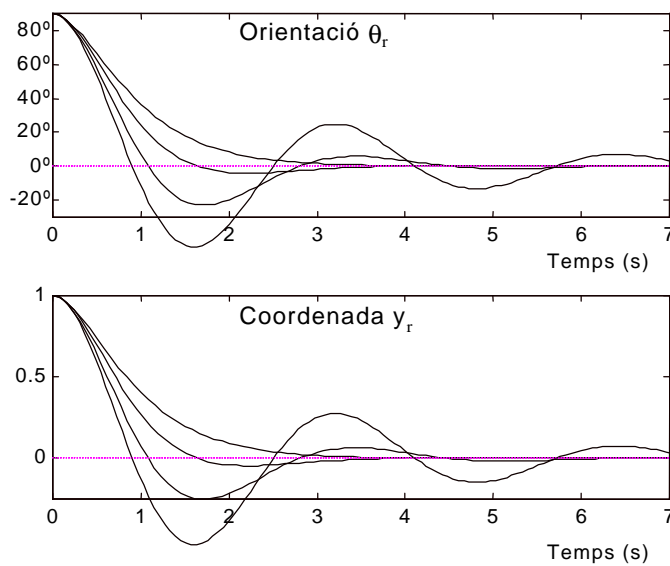


Figura A.4.5: Respostes temporals de $q_r(t)$ i $y_r(t)$ per a diferents valors de z ($\omega_n=2$).

Comentaris sobre comportaments dinàmics d'ordre superior

En el moviment controlat del robot mòbil hem vist que la dinàmica de θ_r és de segon ordre donat que s'ha suposat en tot moment que no hi ha presència de cap altre efecte. A partir del diagrama de la figura A.4.3 és fàcil deduir que els llaços de control de velocitat també presenten una dinàmica pròpia, la qual afectarà al moviment del robot. Aquests efectes són difícils de valorar donat que depenen del model del motor i de les característiques d'aquests llaços de control. En molts casos és acceptable suposar que el model d'un motor respon a un sistema de primer ordre, i que els llaços de control es realitzen utilitzant un controlador PI (un pol). Per tant, la dinàmica de θ_r és de quart ordre: quatre pols.

En teoria de sistemes de control s'intenta treballar amb sistemes de baix ordre i s'aconsegueix en molts casos utilitzant el concepte de la dominància de pols: uns pols dominen sobre uns altres quan tenen una dinàmica molt més lenta. És a dir, cada pol contribueix a la resposta transitòria d'un sistema, i quan un pol és dominant significa que la seva component transitòria és molt lenta. A efectes numèrics significa moltes vegades que els pols no dominants poden eliminar-se. En el nostre cas els llaços de control de velocitat han de presentar una dinàmica molt ràpida respecte de la dinàmica del llaç de control de posició de forma que és correcte prescindir dels pols associats al motor i al controlador PI. Per tant, podem concloure que és vàlid fer l'estudi de θ_r basant-se en una dinàmica de segon ordre, tal i com s'ha fet en aquest apèndix.

Termes específics i acrònims

AOP: De l'anglès, Agent-Oriented Programming. Metodologia per a programar agents proposada per Y. Shoham [Shoham, 90].

BDI: De l'anglès, Belief, Desire, Intention.

Benchmark: Problema o tasca de referència representativa de problemes que podrien aparèixer en dominis reals.

Blackboard: Pissarra en anglès. Fa referència a la tècnica relacionada amb els sistemes distribuïts on tota la informació (missatges) es centralitza en una llista.

Bottom-up: De baix cap a dalt. Fa referència a tècniques de desenvolupament a o processos que s'executen.

CACSD: De l'anglès, Computer Aided Control Systems Design.

COSY: De l'anglès, Complex System.

Croniça: Equivalent al mot crònica.

CRS: De l'anglès, Cellular Robotic System.

Dead-line: Punt en el temps que marca el fi d'alguna cosa. Ultimatum temporal.

Debugging: Procés de comprovació d'un programa en temps d'execució.

Forward chaining: Mètode de raonament on s'analitzen les causalitats futures d'una decisió. En contraposició hi ha el "backward chaining", on l'anàlisi es basa en els antecedents.

Fuzzy: De l'anglès, difús o borros. La tècnica de control fuzzy o la lògica fuzzy és aquella que utilitza informació imprecisa per obtenir resultats.

GA: De l'anglès, Genetic Algorithms.

IA: Intel·ligència Artificial. En anglès AI.

IAD: Intel·ligència Artificial Distribuïda. En anglès DAI.

Mainframe: Marc de referència conceptual.

NN: De l'anglès, Neural Networks.

Off-line: Fora de línia, o fet abans. Fa referència a una tipologia de sistemes reals de control.

On-line: En línia, en contacte directe. Fa referència a una tipologia de sistemes reals de control.

OOP: De l'anglès, Object-Oriented Programming.

RTAIS: De l'anglès, Real-Time Artificial Intelligence Systems.

Scheduling: Mètode per assignar recursos amb restriccions temporals i espacials.

Sinergistics: Efectes sinergistics. Els seus efectes són diferents si es comparen els que s'obtenen d'una execució seqüencial dels que s'obtenen d'una execució simultània.

SMA: Sistema Multi-Agent

Subsumpció: Nom assignat per R.Brooks [Brooks, 86] a una forma d'estructurar i combinar els comportaments. També s'anomena arquitectura horitzontal.

Testbeds: Entorns en què s'implementen els "benchmarks".

Top-down: De dalt cap a baix. Fa referència a tècniques de desenvolupament a o processos que s'executen.

Adreces web interessants

• Informació i organismes sobre IAD i agents

AgentLink: <http://www.agentlink.org>

AIPS'98 (*A.I. and Planning Systems*): <http://www.cs.cmu.edu/~aips98>

Cognitive & Agent Architectures: <http://ai.eecs.umich.edu/cogarch0/index.html>

Coordination in MAS: <http://www.elec.qmw.ac.uk/dai.html>

EUROBOT-EU (*European Research on new Testbeds*): <http://eurobot.udg.es>

FIPA (*Foundation of Intelligent Physical Agents*): <http://www.fipa.org>

Knowledge Systems Lab. (Y. Shoham): <http://www-ksl.stanford.edu>

KQML (*Knowledge Query and Manipulation Lang.*): <http://www.csee.umbc.edu/kqml/>

MAS Lab. (V.Lesser): <http://dis.cs.umass.edu>

• Futbol robòtic

MiroSot (*Micro-robot Soccer Tournament*): <http://www.fira.net>

PHAGSIM: <http://www.etse.urv.es/~aoller/phagsim/PHAGSIM.htm>

RoboCup (*Robot-Soccer World Cup*): <http://www.robocup.org>

RogiTeam: <http://rogiteam.udg.es>

• Plataformes de disseny d'agents

AgentBuilder®: <http://www.agentbuilder.com/>

April Agent Platform (AAP): <http://www.nar.fujitsulabs.com/aap/index.html>

Agent X: <http://www.iks.com/agentx.htm/>

Gypsy: <http://www.infosys.tuwien.ac.at/Staff/lux/Gypsy/>

IBM-ABE: <http://networking.ibm.com/iag/iagsoft.htm/>

FIPAOS: <http://www.fipa.org>

Jack: <http://www.agent-software.com.au/jach.html>

JAFMAS: <http://www.eecs.edu/~abaker/JAFMAS/>

JATLite: http://java.stanford.edu/java_agent/html/

Kafka: <http://fujitsu.co.jp/hypertext/free/kafka/>

LALO: <http://www.crim.ca/sbc/english/lalo/>

MadKid : <http://www.madkit.org>

Via: <http://www.kinetoscope.com/via/default.htm>

Voyager: <http://www.objectspace.com/>

Zeus : <http://www.lab.bt.com/projects/agents/zeus>