

<b>1</b>	<b>OBJECTIU .....</b>	<b>1</b>
<b>2</b>	<b>ABAST DEL PROJECTE.....</b>	<b>2</b>
2.1	COMPORTAMENTS.....	2
2.2	RECURSOS.....	2
2.3	LLIBRERIES .....	3
<b>3</b>	<b>ORGANITZACIÓ DEL PROJECTE .....</b>	<b>4</b>
<b>4</b>	<b>RESULTATS.....</b>	<b>5</b>
<b>5</b>	<b>RESUM.....</b>	<b>6</b>
5.1	DESCRIPCIÓ DE L'ARQUITECTURA SAPHIRA.....	6
5.1.1	<i>Arquitectura utilitzada: Saphira .....</i>	6
5.1.2	<i>Comportaments (behaviors).....</i>	6
5.1.3	<i>Activitats: Programació en Colbert .....</i>	7
5.2	DESENVOLUPAMENT DE LLIBRERIES .....	7
5.2.1	<i>Sonar: Accés a les dades del sonar .....</i>	7
5.2.2	<i>sfUtils: Creació d'una llibreria d'utilitats .....</i>	8
5.3	CREACIÓ DE COMPORTAMENTS .....	8
5.3.1	<i>Seguiment de punts en moviment: sfReachPosition.....</i>	8
5.3.2	<i>Seguiment de trajectòries complexes: sfDtTrajectory.....</i>	9
5.3.3	<i>Seguiment de contorns: sfTracking .....</i>	9
5.4	PROVES DE FUNCIONAMENT.....	9
5.4.1	<i>sfReachPosition.....</i>	9
5.4.2	<i>sfDoTrajectory .....</i>	10
5.4.3	<i>sfTracking.....</i>	10
5.5	MANUAL D'USUARI.....	10



# 1 Objectiu

L'objectiu d'aquest projecte és la programació i control de comportaments en un robot mòbil. Aquests comportaments podran ser utilitzats per realitzar tasques de més alt nivell. Exemples d'aquestes tasques serien les següents:

- Aprenentatge d'objectes
- Seguiment de persones

Les funcionalitats dels comportaments han de ser les següents:

- Seguiment de parets (rectilínies).
- Seguiment de contorns no rectilinis.
- Seguiment de punts en moviment, els quals a més representen una orientació que ha d'assolir el robot.
- Seguiment de trajectòries complexes. Aquestes trajectòries han de poder ser definides per Corbes de Bézier o per B-splines

Hi ha uns altres objectius que es deriven dels primers:

- Es demana que les llibreries creades siguin suficientment generals i útils com per poder ser utilitzades en altres projectes similars. Això implica que algunes parts han de poder ser reutilitzades.
- Es vol obtenir una documentació en la qual s'expliqui clarament el procés de creació de comportaments, de manera que pugui ser realitzat amb més facilitat.

També serà necessari fer proves experimentals per determinar la utilitat dels comportaments, així com possibles limitacions o restriccions de funcionament.

Finalment és necessari tenir un manual d'usuari en el que s'expliqui el procés d'instal·lació tant de les llibreries com dels comportaments.



## 2 Abast del projecte

### 2.1 Comportaments

Hi ha una sèrie de restriccions a tenir en compte en els comportaments. Aquestes restriccions depenen del comportament, tot i això hi ha unes restriccions comuns a tots ells:

- Es vol un control de la velocitat. Tot i això és acceptable que en determinades circumstàncies es pugui disminuir, per tal d'adequar-se a les limitacions físiques que té el robot
- De la mateixa forma que el robot físic, tots els comportaments tenen certes limitacions en els paràmetres, especialment en la velocitat d'execució. Es fixarà un rang de velocitats vàlides per cada comportament, de forma que aquest sigui al menys un 75% de la velocitat límit del robot. També es fixarà un rang d'utilització vàlida dels sonars, que és l'únic que s'utilitzarà per tal de distingir contorns.
- Els comportaments seran provats tant en simulació com amb el robot real

Es requereix un seguiment correcte del punt (i orientació), però no es demana un comportament intel·ligent d'evasió d'obstacles o de planificació de trajectòries. Aquesta restricció també serà aplicable al seguiment de trajectòries.

En el cas de seguiment de parets i contorns hi ha dos tipus principals de restriccions: forma i distribució dels objectes, i característiques dels materials. La forma i la distribució d'objectes ha de ser tal que pugui ser distingible utilitzant només el sonar. A més, les propietats de reflexió del material han de ser les adequades per tal de garantir unes bones lectures.

### 2.2 Recursos

Un factor important és que s'utilitzaran els recursos estàndard que té Saphira (versió 6.2) respecte a la programació de comportaments. Aquest fet inclou tots els programes informàtics que conté Saphira (simulador i traductor de comportaments).

La programació tant de llibreries externes com de comportaments serà realitzada utilitzant Microsoft Visual C++, però només utilitzant C, sense utilitzar disseny orientat a objectes (C++). Està fora de l'abast d'aquest projecte utilitzar altres llenguatges de programació, així com la integració amb altres sistemes diferents del bàsic de Saphira.

El sistema operatiu utilitzat serà Microsoft Windows (versió 98, NT o 2000).



El robot físic que s'utilitzarà serà el model Pioneer 2 DX.

Finalment comentar que en cap cas s'intentarà modificar el robot físic per tal d'augmentar les seves prestacions.

## **2.3 Lliberies**

Les lliberies han de ser en forma de DLL's de Windows, de manera que puguin ser integrades fàcilment en qualsevol altre projecte, a més de l'actual.



### 3 Organització del projecte

El projecte està organitzat de la següent forma:

*Memòria:*

- Descripció arquitectura Saphira: **Capítols 1-4**
- Llibreries de programació:
  - o Sonar: **Capítols 5-6**
  - o sfUtils: **Capítols 7-11**
- Realització de comportaments
  - o sfReachPosition **Capítols 12-15**
  - o sfDoTrajectory **Capítols 16-18**
  - o sfTracking **Capítols 20-23**
- Resultats experimentals **Capítols 24-27**
- Referències **Capítol 28**

*Annexes:*

- 1) Manual d'usuari
- 2) Codi font
- 3) Pressupost



## 4 Resultats

El resultat del projecte ha estat la creació d'una sèrie de comportaments que poden ser utilitzats per tota una família de robots mòbils (tots els que actuen amb Saphira). També s'han creat dues llibreries de programació que poden ser utilitzades de forma independent, i que faciliten la posterior programació de comportaments. Un altre aspecte important a destacar és la descripció que s'ha realitzat de tota l'arquitectura, de manera que s'ha obtingut una documentació compacta que resumeix i descriu tot el procés de disseny d'un comportament (i d'una activitat). Aquest procés s'ha estudiat no solament des del punt de vista teòric, sinó també des del punt de vista del programador, amb tota la problemàtica derivada.

S'han creat tres comportaments:

- sfReachPosition: Seguiment de punts amb orientació
- sfDoTrajectory: Seguiment de trajectòries complexes (Corbes de Bézier i B-Splines)
- sfTracking: Seguiment de parets (rectilínies) i contorns (no rectilinis)

Els comportaments sfReachPosition i sfDoTrajectory són plenament funcionals, però estan dissenyats per poder integrar-se amb altres llibreries. També s'ha vist la forma d'integració, de manera que no és una possibilitat teòrica, sinó un fet ja provat. Queda per posteriors ampliacions la creació d'aquestes llibreries de planificació, i la total integració amb altres sistemes (càmares, sistemes avançats de visió, etc.).

El comportament sfTracking és completament autònom, però necessita ajustar una sèrie de paràmetres per tal de ser operatiu en qualsevol entorn. Aquest fet es deriva de l'elevada complexitat que té un comportament d'aquestes característiques. Una futura ampliació d'aquest projecte seria un estudi estadístic dels paràmetres de funcionament, de manera que s'obtingués un model aplicable de forma automàtica. Aquest model podria ser incorporat "de sèrie" en l'actual comportament.

Finalment s'han creat dues llibreries:

- Sonar: Accés a les lectures del sonar
- sfUtils: Tipus de dades dinàmics i actualitzats amb l'estat intern del robot.

Les llibreries de programació han jugat un paper fonamental en la creació dels comportaments, i tenen un caràcter general, de manera que poden utilitzar-se en qualsevol projecte que utilitzi l'arquitectura Saphira. És per això que poden ser fàcilment reutilitzades en futures implementacions.



## 5 Resum

### 5.1 Descripció de l'arquitectura Saphira

Aquest document està format per varies parts:

- Visió global de l'arquitectura del Saphira
- Presentació del llenguatge Colbert
- Estudi dels comportaments (*behaviors*)
- Introducció a la lògica difusa

En primer lloc s'ha de presentar quin és l'entorn sobre el qual es treballarà, i aquest entorn és el Saphira.

Després s'ha fet una presentació del llenguatge de programació utilitzat, que en aquest cas és el Colbert. Amb el llenguatge Colbert s'han programat les activitats, que són el que controlen a alt nivell les accions que farà el robot.

Finalment s'ha realitzat una descripció exhaustiva dels comportaments (*behaviors*) en Saphira. Els *behaviors* són els encarregats de controlar el robot a un nivell més baix que el Colbert. Els comportaments estan basat en lògica difusa, i és per això que hi ha un apartat dedicat a la seva presentació.

#### 5.1.1 *Arquitectura utilitzada: Saphira*

La programació d'un robot mòbil és una feina complexa, en la que hi ha molts factors a tenir en compte. En aquest projecte s'utilitzarà el Saphira, una arquitectura que permet el control d'un robot autònom. Des de Saphira es poden realitzar totes les tasques de control:

- Interfície per realitzar el control físic del Robot
- Programació de comportaments (control físic del robot mitjançant lògica difusa)
- Programació d'activitats (execució de comportaments, monitorització de tasques, control a alt nivell de les accions a realitzar)
- Simulador del robot

#### 5.1.2 *Comportaments (behaviors)*

Els comportaments són els encarregats de fer el control físic del robot. Tot i que es pot controlar el robot de forma directa, amb els comportaments s'aconsegueixen realitzar de forma més eficient



polítiques d'acció molt més complexes i elaborades que si es controlés directament la velocitat i el gir del robot.

Els comportaments reben informació tant de l'estat del robot com de l'entorn, i tenen una sèrie de regles orientades a assolir uns objectius particulars.

A més, els comportaments tenen una sèrie de característiques que fan que sigui fàcil de combinar varis comportaments (*behaviors*) en la realització de tasques complexes.

La creació de comportaments està basada en la lògica difusa, i és per això que s'ha inclòs un apartat amb la seva descripció. Hi ha un factor a tenir en compte: la descripció que es fa serà funcional. És a dir, no s'intenta fer cap demostració dels axiomes ni de les afirmacions que es fan. L'únic que es vol és tenir una base operativa per tal de poder aplicar el coneixement d'aquesta a la construcció de comportaments. És per això que si el lector té coneixements previs de lògica difusa pot ometre aquest apartat.

### **5.1.3 Activitats: Programació en Colbert**

Per controlar el robot a baix nivell existeixen els comportaments, els quals permeten actuar directament sobre el robot físic. A un nivell superior es troben les *activitats*. Amb les activitats es pot decidir quins són els comportaments que han d'estar actius, es poden parar o fer executar varis al mateix temps, de forma concurrent. És a dir, permet controlar i combinar els comportaments de forma eficient per tal d'assolir un objectiu global de més alt nivell.

A més, hi ha una sèrie de característiques que fa que siguin unes eines de treball molt potents:

- Execució concurrent
- Sincronització d'activitats

La execució concurrent permet tenir més d'una activitat funcionant al mateix temps, mentre que la sincronització permet a les activitats enviar-se una sèrie de senyals entre elles per tal de sincronitzar les accions a realitzar.

## **5.2 Desenvolupament de llibreries**

### **5.2.1 Sonar: Accés a les dades del sonar**

Saphira no té funcions específiques per treballar amb els sonars, i és per això que s'ha creat una llibreria que permeti accedir fàcilment a les lectures realitzades per aquests. Tot i no ser necessària estrictament, aquesta llibreria té unes característiques que la fan indispensable. És per això que s'ha cregut necessària la seva incorporació.





### **5.2.2 *sfUtils*: Creació d'una llibreria d'utilitats**

Aquesta llibreria té com a objectiu disposar d'una sèrie de funcions per manipular de forma eficient estructures de dades dinàmiques. Així s'han fet dos tipus nous de dades:

- Vector
- LlistaPunts

El tipus vector és el més senzill, i només és una construcció que servirà d'ajuda en casos particulars, però tindrà com a principal aplicació donar suport al tipus LlistaPunts.

El tipus LlistaPunts és el més important, i pot mantenir una llista de punts i realitzar operacions sobre ella. En particular es poden consultar valors, afegir i treure elements, i fer càlculs geomètrics. La característica més important és que els punts es mantenen actualitzats en el mapa intern del robot, de forma que els seus valors corresponen a punts reals de l'entorn del robot.

## **5.3 Creació de comportaments**

S'han creat una sèrie de comportaments, els quals poden ser utilitzats de forma independent o de forma integrada amb altres.

### **5.3.1 *Seguiment de punts en moviment: sfReachPosition***

Aquest comportament té com a objectiu fer el seguiment d'un punt en moviment, però la forma amb la que es realitzarà té unes característiques especials:

- Està basat en la construcció d'una API de funcions que fan d'interfície amb altres activitats (o altres API's diferents).

Aquest fet és particularment útil, ja que permet integrar les diferents parts d'un control complex. Així es té per un costat el control del moviment, i de forma independent es poden tenir els algorismes que decideixin cap a on s'ha de dirigir el robot. D'aquesta forma es pot utilitzar el comportament de forma integrada amb dades rebudes del sonar, de càmares d'estereovisió, o de qualsevol altre sistema de sensors.

Les característiques d'aquest comportament fan que es pugui modificar de manera dinàmica la posició i la orientació del punt a seguir, de manera que aquest punt pot estar calculat en temps real, i pot pertànyer a un objecte en moviment.



### **5.3.2 Seguiment de trajectòries complexes: *sfDtTrajectory***

Aquest comportament té com a objectiu el seguiment de trajectòries complexes. La trajectòria està determinada per una sèrie de punts de control, els quals són utilitzats per determinar la forma que ha de tenir. Així hi ha dues construccions possibles:

- Corbes de Bézier
- B-Splines

S'ha decidit modelar les corbes amb aquestes funcions per la senzillesa de control que tenen, unit a la potència i generalitat de les seves representacions.

Les trajectòries poden ser modificades en temps real, i és possible afegir nous punts amb el robot en moviment, sense haver de reiniciar el comportament.

Igual que el comportament anterior estarà basat en un API de funcions, de manera que es pot separar el procés de creació de trajectòries del seguiment d'aquestes.

### **5.3.3 Seguiment de contorns: *sfTracking***

L'objectiu d'aquest comportament és el de seguir un contorn arbitrari, amb les següents característiques de funcionament:

- Velocitat constant (pot ser elegida al executar el comportament)
- Distància constant (pot ser elegida al executar el comportament)
- Es fa ús únicament de la informació rebuda pels sonars

L'objectiu d'aquest comportament es pot subdividir en tres parts diferenciades:

- Seguiment d'una paret
- Seguiment d'un contorn no lineal
- Integració i generalització (seguiment de paret + seguiment de contorn, simultàniament)

## **5.4 Proves de funcionament**

Tots els comportaments han estat provats en simulació i amb el robot real. S'ha de dir que el funcionament òptim amb el robot real en tots els comportaments està fora l'abast d'aquest projecte, tot i això es compararan els resultats obtinguts en els dos entorns (físic i simulat).

### **5.4.1 *sfReachPosition***

El comportament obtingut en simulació ha estat correcte, de forma que es podia actualitzar la posició a seguir en temps d'execució, sense que es produïssin alteracions en el moviment del



robot. Els resultats que s'han obtingut en el robot real han estat molt similars als anteriors, de manera que es pot afirmar que aquest comportament pot ser utilitzat en tots els casos.

#### **5.4.2 *sfDoTrajectory***

Igual que el comportament anterior, es pot afirmar que els resultats obtinguts han estat satisfactoris, tant amb l'entorn de simulació com amb el robot real. Igual que abans es pot concloure que aquest comportament pot ser utilitzat en qualsevol cas.

#### **5.4.3 *sfTracking***

Els resultats que s'han obtingut són diferents als anteriors comportaments.

Es pot afirmar que en simulació els resultats han estat correctes, i el robot ha funcionat perfectament tant en el seguiment de parets com en el de contorns, així com en la integració de tots dos.

Els resultats en el robot real també han estat satisfactoris en el seguiment de parets i de contorns, però en el cas de la integració ha estat necessari un procés de prova i error per tal de determinar els paràmetres òptims de funcionament. Un cop trobats aquests paràmetres també es pot afirmar que els resultats han estat positius.

### **5.5 Manual d'usuari**

Ha estat necessària la creació d'un manual d'usuari dels comportaments realitzats. En aquest manual s'han indicat instruccions per carregar els comportaments des de Colbert (Saphira), així com els passos a seguir per incloure les llibreries (Sonar i sfUtils) des de C.

També s'ha detallat el procediment de creació de nous comportaments, per tal de tenir una guia útil de consulta.

