



Construcció i programació d'un cap robòtic

Autor:

Aniol Garcia i Serrano

Supervisors:

Guillem Alenyà

Sergi Foix

Juliol, 2016



Institut de Robòtica i Informàtica Industrial (IRI)

Consejo Superior de Investigaciones Científicas (CSIC)

Universitat Politècnica de Catalunya (UPC)

Llorens i Artigas 4-6, 08028, Barcelona, Spain

Tel (fax): +34 93 401 5750 (5751)

<http://www.iri.upc.edu>

Corresponding author:

G. Alenyà

tel: +34 93 405 4490

galenya@iri.upc.edu

<http://www.iri.upc.edu/staff/galenya>

1 Introducció

Aquest projecte ha estat desenvolupat durant l'estada a l'empresa de primer de batxillerat. Va ser dut a terme al laboratori de Percepció i Manipulació de l'IRI (Institut de Robòtica i Informàtica Industrial) i proposat i assistit per Guillem Alenyà Ribas i Sergi Foix Salmerón. El projecte consisteix en la creació i programació d'un cap robòtic.

2 Objectius

Es pretén fer que la interacció humà-robot sigui més natural, fent que el robot pugui «comunicar-se» i/o expressar intencions. La manera d'aconseguir aquest objectiu és dotant el robot d'un component més humà. Això es durà a terme mitjançant la construcció d'un cap robòtic dotat d'una càmera i una pantalla on aparegui una cara per tal que l'usuari pugui seguir d'una manera més còmode els moviments del robot en qüestió.

El projecte tindrà els següents requeriments:

- Haurà d'estar muntat en un suport mòbil.
- Tindrà mobilitat en dos eixos mitjançant motors.
- Es podrà anticipar als moviments que el robot faci, per tal que l'usuari els pugui seguir.
- En cas que el robot esperi una acció, s'haurà de mirar a l'usuari.
- S'haurà de poder canviar de posició el suport, adaptant-se al nou sistema de referència.

Aquesta cara podria mostrar les intencions del robot i si, per exemple, el robot pretén moure el braç cap a una posició concreta, el cap miri primer cap a aquell punt. D'aquesta manera, l'usuari podrà predir quins seran els moviments del robot. Per tal de donar més mobilitat, tota l'estructura pot estar muntada en un trípod.

3 Muntatge

L'estructura constarà de tres parts:

- El suport: s'haurà de fer una estructura lleugera que pugui ser muntada sobre un trípod, de manera que faciliti el moviment.
- La part motriu: Serà del tipus pan-tilt (dos motors i per tant dos eixos de moviment), amb motors Dynamixel RX-64.
- El cap: Constarà de la pantalla de 11" on s'imprimirà la cara i de la càmera que s'encarregarà de l'anàlisi d'imatge.

S'ha de tenir en compte que l'estructura final haurà de ser el màxim lleugera possible, de manera que els motors puguin treballar sense forçar. Per aquest mateix motiu, també s'hauran d'evitar els braços de palanca. Una estructura semblant a l'esbós de la figura 1 compliria els requisits: les peces poden fer-se fàcilment amb una impressora 3D, i per tant serà lleugera. Al tenir els motors en aquesta disposició, les forces es repartiran de manera uniforme.

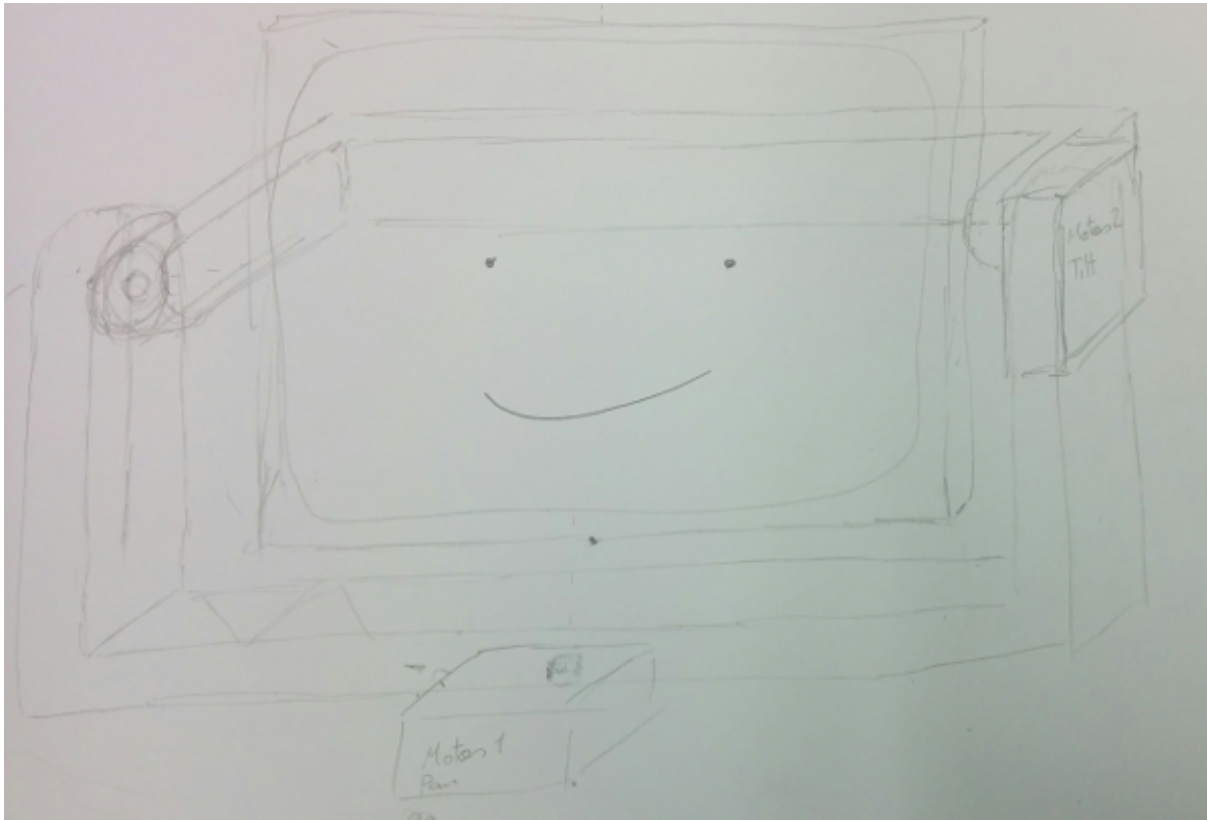


Figura 1: Esbós de l'estructura general del cap robòtic

4 Procediments

El primer que es va fer al iniciar el projecte va ser definir quins serien els objectius finals, el que es pretenia fer (veure l'apartat d'objectius). Un cop es va tenir aquest primer punt, es va pensar com hauria de ser el cap, des del punt de vista mecànic. En principi s'havia d'utilitzar una estructura de tipus "pan-tilt" (figura 2) amb la pantalla i la càmera muntats a sobre.

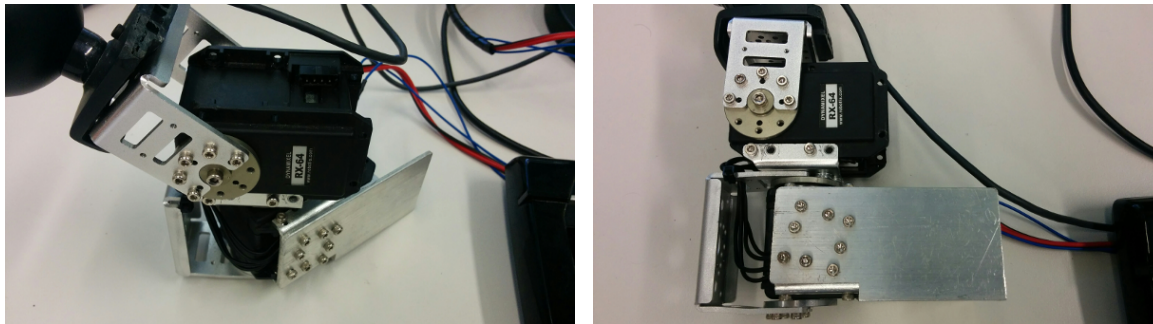


Figura 2: Estructura pan-tilt

El problema d'aquest disseny era que els motors (especialment el motor del tilt) havien de vèncer grans braços de palanca, de manera que es necessitava massa força. D'aquesta manera, i modificant les posicions dels motors respecte el cap es va arribar a una configuració provisional amb les forces molt més repartides (veure figura 1).

Un cop l'estructura mecànica va ser definida, i abans de començar a programar, es van

estudiar les llibreries i drivers en els que es basaria el codi i que era imprescindible utilitzar. Aquestes llibreries van ser:

- OpenCV: Open Computer Vision, utilitzada per fer l'anàlisi de l'entorn, seguiment de diversos objectes, i per crear i mostrar per pantalla la cara del robot.
- Dynamixel Pan Tilt: Driver de baix nivell pels motors dynamixel que s'han utilitzat. D'aquest driver depenen tots els moviments que hagi de fer el cap del robot. La classe *mechanical*, amb els mètodes que en última instància s'utilitzaran per moure el cap, es basa en aquesta i utilitza les seves estructures de dades. *Dynamixel_Pan_Tilt* necessita la llibreria *DynamixelServer_ftdi*.
- Dynamixel Server: Llibreria per crear el servidor per tal de comunicar els motors amb l'ordinador. És una dependència de Dynamixel Pan Tilt.

Un cop es va conèixer el seu funcionament, es va decidir que tot el programa havia d'estar separat per diverses classes (per facilitar el possible aprofitament de cadascuna de les parts per separat i la detecció i sol·lució d'errors) i es va fer un primer esborrany de l'estructura d'aquestes (veure figura 3).

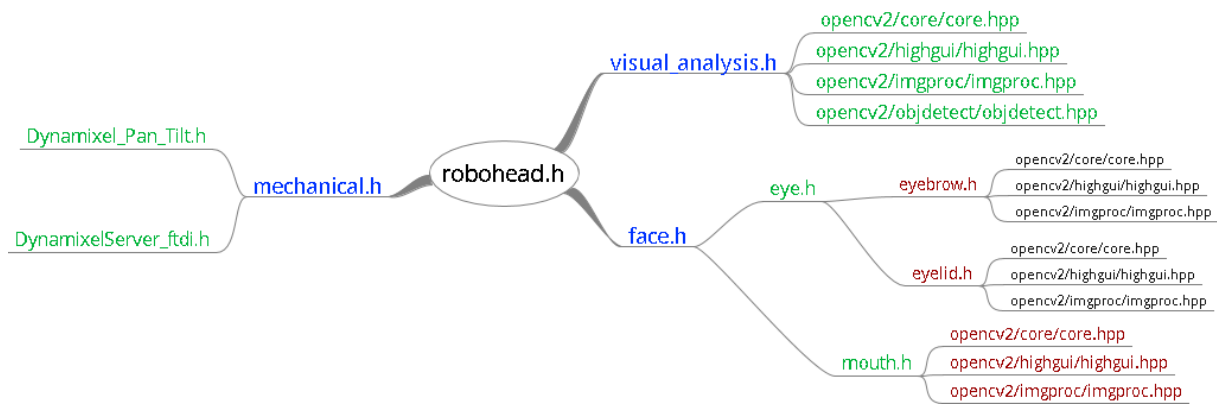


Figura 3: Estructura de classes a l'inici

Tot i això, finalment no es van fer les classes pertanyents a cadascun dels elements de la cara. En comptes d'això es va fer una sola classe per a la cara sencera, que ja tenia totes les parts necessàries incloses. Finalment l'estructura de classes ha quedat de la següent manera (figura 5):

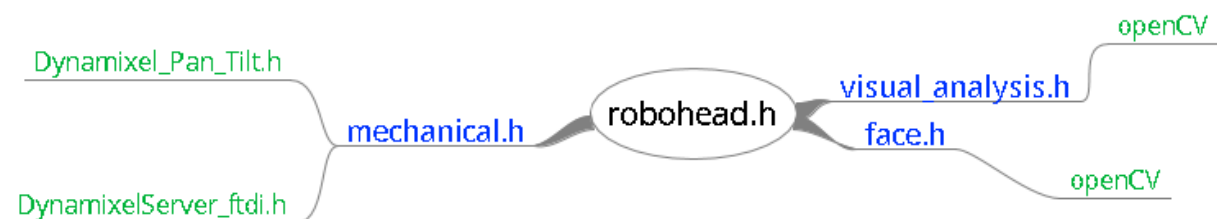


Figura 4: Estructura de classes final

Quan les classes van estar ben definides, es va començar a programar. El primer a fer va ser la part de motors. Aquesta llibreria (la *mechanical*) permet controlar els motors d'una manera més còmode i a més alt nivell que no pas amb la Dynamixel Pan Tilt. El següent pas va ser fer el sistema d'anàlisi visual, que utilitza la llibreria OpenCV. En un principi, el cap només podia detectar i seguir colors, però posteriorment es van afegir mètodes per detectar cares i algunes

formes. Amb aquestes dues classes, el robot ja era capaç de detectar diversos objectes i seguir-los amb el cap.

Quan la part d'anàlisi d'imatge va estar acabada, es va fer la part de la cara, utilitzant les funcions de dibuix d'OpenCV, que consisteixen en anar insertant formes dins una matriu. En aquest cas, tan sols es va haver d'utilitzar circumferències, cercles i fragments d'el·lipses. A través de combinacions i canvis de posicions d'aquests elements es van crear un total de 5 cares amb expressions diferents (figura), que poden ser utilitzades per mostrar estats del robot. La classe *robohead* ja incorpora un mètode per mostrar diverses expressions.



Figura 5: Conjunt de possibles expressions de la classe *robohead*

Amb aquests mètodes ja quedaven coberts els requeriments bàsics, però s'han anat implementant altres funcions addicionals, com per exemple el fet de que el robot no només pugui fer el seguiment d'objectes amb el cap, sinó que també el pugui fer amb els ulls o el fet que pugui moure els ulls cap a una coordenada concreta de l'espai (respecte el seu propi sistema de referència).

5 Codi

El codi es basa en un conjunt de quatre classes: *robohead*, *visual_analysis*, *face* i *mechanical*.

La classe *robohead* és la principal, i a través d'aquesta s'accedeix a les altres. Els mètodes que conté són les principals funcions del cap, i les que l'usuari pot necessitar. *Robohead* conté objectes del tipus *mechanical*, *visual_analysis* i *face*. Cadascun d'aquests objectes corresponen a un component del cap, i són necessaris pel funcionament dels mètodes, que es basen en aquests.

- *Visual_analysis* s'encarrega del tractament de la imatge proporcionada per la càmera. Principalment s'utilitza per fer el seguiment de cares i passar les coordenades de pantalla a l'objecte *mechanical*, que s'encarregarà de moure els motors. Tot i això, també pot detectar i fer el seguiment d'objectes a partir del seu color (l'usuari ha d'especificar a través del sistema VHS a quin rang buscar) o forma (tot i que de moment només està implementada la funció per buscar cercles).
- *Face* conté mètodes per expressar diferents emocions o estats d'ànim i per moure diferents parts de la cara. Es va decidir posar tots els components de la cara en la mateixa classe, ja que si es feia com inicialment estava pensat, s'hauria necessitat un objecte per cada part de la cara. Tampoc tenia massa sentit tractar els components per separat i tenir mètodes per moure'ls independentment dels altres. Com que l'únic motiu per moure les diverses parts era canviar d'expressió, s'ha fet una funció per canviar-la on ja hi ha definides les posicions de cada element. *Face* té mètodes per enviar els ulls a unes coordenades del sistema de la pantalla (no de l'espai), i si es combina amb mètodes d'altres classes que retornen posicions a la pantalla, i d'aquesta forma fer el seguiment d'objectes. Es treballa amb les coordenades de la pantalla perquè en aquesta aplicació era més convenient localitzar punts en una matriu que no pas anar-los a buscar a l'espai. Tot i això, finalment també s'ha fet un mètode que agafa coordenades de l'espai (en metres) i mira cap a aquella direcció.

- *Mechanical* és l'objecte que controla els motors, i conté informació i mètodes per fer funcionar un pan-tilt. Es basa en el driver estàndard dels motors Dynamixel, i per tant es necessita un fitxer XML amb la configuració dels motors (més informació a la documentació). Aquest objecte no està pensat per ser utilitzat únicament en aquest projecte, sinó que conté una gran quantitat de mètodes que en aquest cas no s'utilitzen i que poden ser útils per a controlar un parell de motors Dynamixel. S'ha considerat que és més intuïtiva i amigable a l'usuari que no pas utilitzar directament la llibreria Dynamixel Pan Tilt. La llibreria conté funcions per moure's a un angle absolut, per moure's un nombre de graus determinat, parar els motors, tornar a la posició inicial...

Aquí hi ha un quadre amb les variables i els mètodes amb els seus corresponents arguments d'aquestes quatre classes. Per a obtenir informació més detallada del que fa cada funció, és recomanable consultar la documentació i els comentaris del codi.

ROBOHEAD
<i>mechanical</i> neck
<i>face</i> virtual_face
<i>visual_analysis</i> camera
CRobohead()
~CRobohead
void follow_face()
void follow_ball()
void follow_color(int min_H_, int max_H_, int min_S_, int max_S_, int min_V_, int max_V_)
void orient_to(int x, int y, int z)
void look_at(double x, double y, double z)
void expression(int face)

VISUAL ANALYSIS
<i>VideoCapture</i> camera
int min_H, max_H
int min_S, max_S
int min_V, max_V
bool success
visual_analysis()
~visual_analysis()
bool get_img(Mat& original_img)
int get_color(double& x, double& y, Mat& original_img, int min_H_, int max_H_, int min_S_, int max_S_, int min_V_, int max_V_)
int get_ball(double& x, double& y, Mat& original_img)
int get_face(double& x, double& y, Mat& original_img)
void get_circle(double& x, double& y, Mat& original_img)

FACE

```

int eye_radius, eyelid_radius, step
int pixel_size, focal_distance
int width, height
face(int width, int height, double pixel_size)
face()
~face()
void follow(int x, int y)
void look_at(double x, double y, double z)
void expression(int face)

```

MECHANICAL

```

double deg_x, deg_y, deg, default_speed
string serial
Dynamixel_pan_tilt_info info
Dynamixel_pan_tilt_data pos, default_pos, desired_pos, desired_speed, desired_effort
Dynamixel_pan_tilt_config config
Dynamixel_pan_tilt_compliance compliance
Dynamixel_pan_tilt_pid pid
CDynamixelServerFTDI *dyn_server
CDynamixel_Pan_Tilt *pan_tilt
double max_angle_error, time_interval, max_time_sec, t, uperiod, step, timeout, torque_timeout
mechanical(std::string& name, int pan_id, int tilt_id, std::string& config_file)
mechanical()
~mechanical()
void move_x(double deg)
void move_y(double deg)
void add_x(double deg)
void add_y(double deg)
void follow(double x, double y)
void move(double& deg_x, double& deg_y)
void add(double& deg_x, double& deg_y)
void start_position()
void stop()
void set_default_speed(int speed)
double get_deg_x()
double get_deg_y()
double max_angle_x()
double min_angle_x()
double max_angle_y()
double min_angle_y()

```

6 Conclusions i treball futur

Es pretenia fer un cap robòtic per fer que la interacció del robot amb l'usuari sigués més natural i, s'han posat les eines suficients per fer-ho d'una manera més senzilla. S'ha escrit codi que permet al robot detectar i seguir cares, i a més s'han afegit funcions per identificar també formes i colors. Aquestes funcions en un principi eren per fer proves amb elements més simples que les cares, però es va considerar que també podien ser útils. Malgrat els problemes que han sorgit amb els motors, s'ha escrit una llibreria que permet moure l'estructura pan-tilt de manera més senzilla que amb la del driver en el qual es basa, i s'hi ha afegit altres funcionalitats complementàries. Finalment també s'ha fet un codi per controlar la cara, aconseguint que pugui mostrar expressions, cosa que no estava planejada a l'inici. A través de la combinació de totes aquestes opcions, es poden aconseguir la majoria dels objectius plantejats inicialment: el robot pot indicar les accions que farà i fer-ne un seguiment mitjançant el moviment del cap i dels ulls, pot mirar l'usuari quan esperi una acció per part d'aquest i pot expressar diverses emocions.

Per completar els objectius inicials ha faltat: poder muntar el cap sencer i poder fer els canvis de sistema de referència respecte el robot. Respecte aquest últim punt, és important dir que el robot haurà de ser l'encarregat de passar les dades tal com les necessita el cap, ja que aquest no té (de moment) funcions per poder-les calcular, essent necessari un procés de calibració. D'aquesta manera, el cap pot estar situat fora del robot i continuar funcionant de manera normal.

Evidentment, aquest projecte no s'ha d'acabar aquí. Es poden afegir moltes altres funcions, i de moment ha quedat pendent l'opció de donar unes coordenades de l'espai al cap per tal que s'orienti cap a aquella posició (tot i que sí hi ha una petita funció per fer-ho amb els ulls), i el fet de poder tractar amb els canvis de sistemes de referència.

A nivell personal, la valoració del projecte és molt positiva, ja que m'he vist forçat a aprendre com funcionaven coses amb les que mai havia tractat abans i, encara que ha quedat feina a fer, estic satisfet amb el resultat final. Com ja he dit abans, aquest projecte no ha d'acabar aquí, i penso que pot ser molt interessant el fet de tenir un cap que pugui ser utilitzat en els robots per millorar la interacció entre humans i màquines.

IRI reports

This report is in the series of IRI technical reports.

All IRI technical reports are available for download at the IRI website

<http://www.iri.upc.edu>.