

## Segona Pràctica: Arquitectura d'un Robot Mòbil

L'objectiu d'aquesta pràctica és implementar una arquitectura de control molt simple que permeti a un robot (Pioneer-3DX) moure's per un entorn amb obstacles i realitzar una determinada tasca. El robot haurà de evitar obstacles molt simples i de petit tamany. Per això, es farà servir una arquitectura que constarà de tres nivells:

- Estratègic (seqüenciador): on s'executarà la missió a alt nivell. Indicarà al nivell tàctic quins són els sub-objectius (punts) per on ha de passar el robot per a executar la tasca.
- Tàctic: on s'executaran els comportaments simples que pot dur a terme el robot (anar cap a un objectiu, evitar obstacles, vagar, etc). Entre d'altres funcions, s'encarregarà de portar al robot fins al punt indicat pel nivell estratègic de manera segura (sense col·lisions). Aquest nivell serà cridat periòdicament pel nivell estratègic.
- Executiu: encarregat de interactuar amb la part física del robot (ultrasons i motors). Bona part del procés executiu ja ve implementat per les llibreries de l'ARIA i pel  $\mu$ -controlador del propi robot.

De tots els nivells el que cal explicar de manera més detallada és el nivell tàctic. Per a implementar aquest nivell es faran servir comportaments reactius. Un comportament és un mòdul que a partir d'una sèrie d'estímuls exteriors dóna una determinada resposta. Aquesta resposta és de tipus reactiu, és a dir, no es realitza cap tipus de planificació i només depèn de les dades que es tenen en aquell moment; vendria a ser per tant, com un acte reflex. Durant aquesta pràctica, la resposta de cada comportament serà un vector que indicarà la direcció que, segons ell, hauria de seguir el robot. Per exemple, hi haurà un comportament per a evitar obstacles, i la seva sortida indicarà la direcció que haurà de seguir el robot per a evitar-los. N'hi haurà un altre per a dirigir-se cap a un determinat objectiu, i la seva sortida indicarà la direcció que ha de seguir el robot per a anar cap a l'objectiu, independentment de si hi ha o no obstacles. Existeixen múltiples maneres per a implementar els comportaments, en el nostre cas s'utilitzarà una tècnica anomenada 'camps de potencial'. Els camps de potencial és un mètode que consisteix en modelitzar al robot com si fos una partícula sotmesa a una sèrie de forces. Per exemple, els obstacles generaran forces de repulsió sobre el robot, mentre que el punt objectiu en generarà una d'atracció. Cada comportament generarà una força i la direcció final del robot s'obindrà a partir de la suma ponderada del resultat de cada un dels comportaments. Així, el valor final de la direcció del robot serà igual a:

$$\vec{V} = \sum_{i=1}^n \omega_i * \vec{V}_i$$

on  $\vec{V}_i$  és la resposta del comportament  $i$  (força generada pel comportament),  $n$  és el nombre de comportaments i  $\omega_i$  és el pes que es donarà al comportament  $i$ . Els valors dels pesos,  $w_i$ , dependran de la tasca concreta que s'ha d'executar i el nivell executiu els podrà anar modificant. Com a norma general, el pes del comportament destinat a evitar obstacles haurà de ser superior que no pas al de anar cap a l'objectiu. La figura 1 mostra gràficament la manera com es combinen els diferents comportaments.

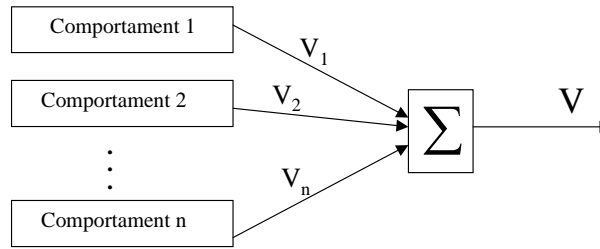


Figura 1. Exemple de combinació de varis comportaments.

Tot i que possiblement n'hàgiu de fer servir més, els dos comportaments bàsics que haureu de implementar obligatòriament són els que s'expliquen a continuació:

- **Anar cap a l'objectiu:** aquest comportament generarà una força que atraurà al robot cap al punt objectiu. Podeu suposar que el mòdul d'aquesta força és constant i igual a 1.
- **Evitar obstacles:** per a cada ultrasò del robot ( $i$ ) que detecti un obstacle, es generarà una força de repulsió amb un mòdul igual a:

$$|\vec{V}_{obs\_i}| = \begin{cases} \frac{MAX\_DIST - d_i}{MAX\_DIST} & \text{si } d_i < MAX\_DIST \\ 0 & \text{en cas contrari} \end{cases}$$

on  $d_i$  és la distància a la que es troba l'obstacle detectat pel sensor  $i$ .  $MAX\_DIST$  és la distància màxima a partir de la qual els objectes detectats ja no es consideren obstacles. El valor final del comportament d'evitar obstacles,  $V_{obs}$  serà el resultat de realitzar la següent suma ponderada:

$$\vec{V}_{obs} = \sum_{i=1}^m \alpha_i * \vec{V}_{obs\_i}$$

on  $\alpha_i$  és el pes donat a la lectura de del ultrasò  $i$ , i  $m$  és el nombre de ultrasons que té el robot. Els valors de  $\alpha_i$  dependran de la posició que ocupa cada ultrasò. Per exemple, els ultrasons que estan al cap del robot tindran més pes que als que es troben al seu darrera.

Quan es detecti algun obstacle molt a prop del robot, a una distància menor a un determinat valor  $D_{min}$ , es suposarà que el perill de col·lisió és imminent. En aquest cas, es deshabilitaran tots els altres comportaments i tan sols estarà actiu el de evitar obstacles.

En algunes arquitectures es suposa que el robot no es pot traslladar-se i rotar al mateix temps. Si el robot no està orientat correctament, segons la direcció indicada pels comportaments, aquest s'aturarà (*robot.setVel(0)*) i començarà a girar fins que s'orienti. Una vegada fet això, ja es podrà desplaçar de nou a la velocitat que es consideri adequada. En el cas d'aquesta pràctica, per a evitar que el robot estigui massa temps aturat, es considerarà que el robot pot també desplaçar-se i rotar al mateix temps sempre que la diferència entre la seva orientació actual i la que hauria de tenir sigui inferior a  $D_r$  graus. Per això, podeu fer servir la funció *isHeadingDone()* de la classe *ArRobot*.

La velocitat de desplaçament del robot haurà d'estar en funció de la quantitat d'obstacles que detecti. Quan el mòdul del vector del comportament d'evitar obstacles

sigui molt gran, la velocitat s'haurà de reduir, en canvi quan el robot es mogui per un entorn menys 'perillos' la velocitat podrà ser major. A més a més, la velocitat haurà de dependre de lo correctament orientat que estigui el robot. La velocitat del robot disminuirà com major sigui la diferència entre la seva orientació actual i la que hauria de tenir. Tal i com ja s'ha explicat durant el paràgraf anterior, si aquesta diferència és superior a  $D_r$  graus el robot s'haurà de aturar. En tot cas, la velocitat mai podrà ser superior a un paràmetre que s'anomenarà  $V_{max}$ . Durant les primeres proves s'ha d'utilitzar una velocitat màxima molt baixa, al voltant de 50 ó 100 mm/s. També s'ha de tenir en compte que és molt difícil que el robot arribi exactament al punt objectiu indicat pel nivell estratègic, per això s'ha de suposar que el robot ja hi ha arribat quan es troba a una distància  $D_o$  de l'objectiu.

El nivell estratègic anirà cridant periòdicament al tàctic per a que aquest decideixi la direcció que ha de prendre el robot. El robot no pot variar ni la seva direcció ni velocitat durant el període de temps que entre dues cridades consecutives del nivell tàctic. A aquest període de temps entre cridades l'anomenarem  $T$  i dependrà de la velocitat màxima a la qual pot anar el robot. Com a norma general, valor típics de  $T$  solen estar entre 200 ms i 500 ms. El funcionament concret del nivell tàctic no s'explica en aquest enunciat ja que depèn de la tasca a alt nivell que es vulgui realitzar.

### **Aplicació a entregar:**

Emprant l'arquitectura explicada, el robot haurà de desplaçar-se fins a un punt objectiu evitant algun obstacle molt simple, com per exemple papereres, cadires, etc. El robot haurà de llegir d'un fitxer els paràmetres de l'arquitectura (els pesos dels comportaments,  $MAX\_DIST$ ,  $V_{max}$ ,  $D_{min}$ ,  $T$ , punt objectiu, etc.). El format d'aquest fitxer el podeu determinar vosaltres mateixos, sempre que s'especifiqui clarament a la documentació. Una vegada el robot ja es pugui moure evitant obstacles, s'haurà de implementar alguna tasca a més alt nivell. Aquesta tasca també la podeu triar vosaltres, sempre que compti amb el vist i plau previ del professor. Alguns exemples de tasques que podeu implementar són les següents:

- Passar per un conjunt de punts: el robot en lloc de tan sols anar a un sol punt haurà de passar conjunt d'ells. Podeu intentar optimitzar la ruta efectuada pel robot, de manera que la distància recorreguda sigui la mínima. Fixeu-vos que en aquest darrer cas es tractaria de solucionar el problema del viatjant de comerç.
- Cerca d'objectes: el robot ha de trobar una sèrie d'objectes que estan situats a posicions desconegudes. Dins del fitxer de configuració s'indicaria la posició d'aquests objectes. Podeu suposar que es tracta d'objectes virtuals, no existeixen realment. El robot tan sols els pot detectar si la distància entre ell i l'objecte és inferior a un determinat valor. Mentre no detecti cap objecte, el robot haurà d'anar cercant-los per l'entorn movent-se de manera aleatòria. Per implementar el moviment de cerca podeu definir un altre comportament, anomenat *renou*, que generarà un vector aleatori. També podeu intentar introduir nous objectius des de el teclat del ordinador.
- Cobrir una zona de l'entorn (tasca de neteja): el robot haurà de intentar passar per tots els punts de l'entorn emulant la tasca de neteja que realitzaria un robot aspirador. Per això, haureu de especificar les dimensions de l'espai a cobrir que, per simplicitat, podeu suposar que es un rectangle. Pot succeir que, degut a un

obstacle, el robot no pugui cobrir alguns punts. Com que aquests obstacles poden ser fixes (una cadira, una paperera, etc), o mòbils (una persona que passava, un altre robot, etc. ), una vegada el robot ha arribat al final del seu recorregut, haurà de intentar tornar a passar pels punts on anteriorment no havia ho havia pogut fer. A més, durant la tasca rutinària de neteja, pot ocorre alguna emergència, com per exemple que s'hagi trencat algun tassó. Si això succeeix, el robot hauria de abandonar el que estava fent i anar immediatament a atendre l'emergència. Les emergències les podeu simular amb el teclat, indicant les coordenades del punt on s'ha produït. Per a implementar aquesta missió, molt probablement necessitareu fer servir un mapa de l'entorn que indiqui els punts per on ha passat el robot i els obstacles que s'ha trobat.

El robot podrà executar tant la tasca simple d'anar fins a un punt de l'espai, com la aplicació d'alt nivell que heu triat. Dins del fitxer de configuració s'indicarà quina de les dues tasques s'ha de dur a terme.

**Possibles ampliacions:**

- Implementació d'altres tipus de mecanismes d'evitació d'obstacles: *Vector field Histogram* (VFH).
- Implementació d'algun algorisme de “seguiments de parets”.

**Documentació a entregar:**

- Descripció de l'arquitectura de control que heu implementat: comportaments utilitzats, formules emprades per el càlcul de les forces, càlculs de les velocitats, etc.
- Descripció detallada de la tasca a alt nivell que heu triat.
- Altres aspectes que considereu importants de la vostra pràctica i que no han estat comentats a l'enunciat d'aquesta.
- Codi font comentat.

**Criteris d'avaluació:**

- Complexitat de la tasca d'alt nivell que s'ha triada.
- Qualitat dels moviments del robot: realització per part del robot de moviments segurs (lliures de col·lisions), però ajustats a la trajectòria que ha de seguir per a arribar a l'objectiu.
- Estructura del programa.
- Qualitat de la documentació.

L'ordre en que apareixen aquests criteris no indica, necessàriament, la seva importància.

**Data límit d'entrega:** 31 de Maig de 2011.

S'haurà de mostrar el correcte funcionament de la pràctica davant el professor, tant amb el simulador com amb el robot real (pioneer-3DX).