

# Navegación de Robots en Formación: un Enfoque Reactivo con Restricciones.

Luis Vicente Calderita, Vidal Moreno y Belén Curto

Universidad de Salamanca.  
{lv.calderita, vmoreno, bcurto}@usal.es

**Resumen** El presente artículo propone una técnica de navegación donde los robots mantienen una determinada posición en la formación y describen al mismo tiempo un movimiento relativo entre ellos, de manera que un observador deduzca que el conjunto se mueve de manera sincronizada y respetando un modelo. Tras estudiar las técnicas más habituales para mantener a un grupo de robots en formación, se ha propuesto una solución reactiva líder-seguidor basada en el seguimiento por parte de los robots seguidores de unas posiciones imaginarias e inalcanzables relativas al líder, que se van actualizando en cada iteración. Para conseguir el giro del conjunto sin perder la forma se ha limitado el radio del movimiento curvilíneo del robot líder, de modo que este describe curvas suaves consiguiendo que el resto de robots las ejecute con precisión. El control de la formación se ha implementado de forma centralizada, ejecutándose en un ordenador que recibe los datos de la formación y envía las órdenes correspondientes a los robots.

**Keywords:** navegación en formación, enfoque reactivo, ley de control, robots móviles.

## 1. Introducción

Si observamos la naturaleza y pensamos en las bandadas de los pájaros, o en los bancos de peces, contemplamos un grupo de animales sincronizados, que de manera natural van hacia un rumbo con aparente sencillez. El movimiento de cada uno confiere la sensación de que el grupo se desplaza guiado desde el exterior llegando a parecer, por la precisión con que cambian de sentido, que el control de todos los movimientos está centralizado por un ente superior que los coordina con maestría, sin embargo, todo indica que el desplazamiento global es resultado de las actuaciones individuales, comprendiendo que cada uno actúa sobre la base de su percepción local.

En nuestro conjunto de robots, al contrario que en la naturaleza, sí podemos pensar en otro tipo de control distinto al descentralizado, ya que podemos dirigir a un grupo desde un ordenador central que procese las posiciones relativas de cada miembro y envíe los movimientos a ejecutar.

En los últimos años, ha habido mucha actividad de investigación sobre sistemas de múltiples robots móviles. La justificación de estos sistemas dentro de la robótica se podría sintetizar en estas dos razones: [8]

- Existen tareas demasiado complejas para ser realizadas por un solo robot o en las que emplear varios robots supondría una gran ventaja.
- Posiblemente sea más sencillo y económico construir varios robots simples que trabajen en equipo que un solo robot suficientemente potente como para realizar la tarea por si mismo.

El estudio de sistemas multi-robot extiende la investigación en sistemas mono-robot, siendo una disciplina por si misma ya que múltiples robots pueden cumplir tareas que no pueden ser realizadas por un solo robot, ya que este por muy capaz que sea está limitado por lo menos espacialmente [8], [22], [16].

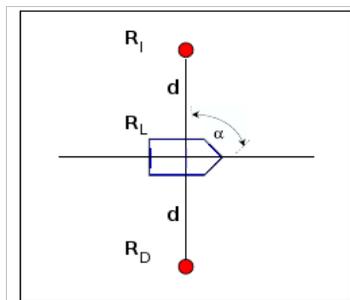
En la literatura se encuentran distintos métodos o enfoques para abordar la navegación de robots en formación. Se aplican las leyes de teoría de control, realizando el seguimiento de un líder mediante control realimentado, manteniendo una distancia y ángulo relativo prefijados [11], [12]. La utilización de las denominadas Estructuras Virtuales, consiguen que un sistema multi-robot se comporte como si fuera un sólido rígido [24]. También se utiliza la técnica de los campos de potencial, la cual tiene su fundamento en [17]. Esta consiste básicamente en dirigir al robot como si fuera una partícula moviéndose en un campo vectorial gradiente, el robot es una partícula con carga positiva bajo la influencia del campo de fuerzas que lo atraen al objetivo cargado negativamente, los obstáculos se representan con carga positiva generando la fuerza repulsiva que lo aleja. Para conseguir que una formación de robots se mueva conjuntamente con este método bastaría con atraer a cada robot hacia su posición en la formación. En [23] podemos ver el diseño completo basado en controladores del campo de potencial, modificados on-line para acomodarlos a las restricciones inducidas por el movimiento de los otros robots. Para cada robot se crean tres regiones: segura, crítica e insegura, estas modifican el potencial aplicado para conseguir la formación deseada. Por su parte, Barfoot y Clark [3] crearon un sistema de trayectorias

autónomas que genera una trayectoria independiente para cada robot a partir de la trayectoria deseada para la formación, empleando un sistema de coordenadas curvilíneo muy apropiado para robots no holonómicos. Este método emplea un sistema de referencia global para estimar la posición de cada robot y no requiere de comunicación entre robots excepto con un generador de trayectorias para la formación. A partir de esta trayectoria y de su posición relativa prefijada cada robot calcula su propia trayectoria.

## 2. Enfoque reactivo en el control de una formación.

La idea básica del enfoque propuesto está inspirada en el concepto de los potenciales sociales de Balch descrito en [2]. Simplificando que significa estar en formación desde el punto de vista de un individuo, no es más que mantenerse en una posición relativa a otro individuo o a varios, en ese momento un observador ajeno advierte una forma entre los individuos concluyendo que sus posiciones no son fruto de la casualidad. Si añadimos el movimiento al conjunto, para ir en formación, mantener la posición implica necesariamente mantener una determinada velocidad y orientación a lo largo del tiempo. Entonces, en un enfoque reactivo, el robot debe ser capaz de realizar la tarea de ir a una posición determinada y una vez alcanzada mantenerse a la misma velocidad y orientación que sus compañeros. En el enfoque reactivo propuesto, el robot simplemente comprueba si se encuentra o no en la posición correcta, si no está, debe reaccionar e ir a esa posición.

Se ha optado por el planteamiento típico, líder-seguidor, ampliamente utilizado en el control de formaciones [15] [4]. El líder va marcando un camino y los demás se colocan y mantienen en una posición relativa respecto a él. Figura 1.



**Figura 1.** Formación de tres robots.

Como método de ubicación se ha recurrido, por su sencillez, a la odometría, y de esta forma sabemos para cada robot, aproximadamente, en que lugar del mundo se encuentra. Las ecuaciones que describen las posiciones relativas respecto a un líder para una formación en paralelo como la mostrada en la figura 1,

podrían ser las descritas en (1) para el robot izquierdo. Donde  $RL$  representa al robot líder, en el centro de la imagen 1, y  $RI$  y  $RD$ , son el robot izquierdo y el robot derecho respectivamente, representados por sendos puntos arriba y abajo en la imagen 1.  $RL$ ,  $RI$  y  $RD$  son estructuras con la información necesaria para representar la posición de un robot en el mundo, sus coordenadas  $x$  e  $y$  y su ángulo  $\alpha$ .

$$\begin{aligned} RI_x &= RL_x + d \cos(RL_\alpha + \alpha) \\ RI_y &= RL_y + d \sin(RL_\alpha + \alpha) \\ RI_\alpha &= RL_\alpha \end{aligned} \tag{1}$$

Lo realmente interesante de este planteamiento a la hora de definir formaciones, es la sencillez. Se podrían describir formaciones muchos más complejas, no solo con posiciones relativas al líder sino con posiciones relativas a cualquiera de los seguidores. El cambio de la estructura en paralelo anterior, a una en triángulo equilátero sería simplemente definir  $\alpha = 120$  grados.

El tipo de control de formación elegido es flexible, en tanto en cuanto, los ejemplares pueden perder su posición en el conjunto, por ejemplo, si aparece un obstáculo y más tarde retomarla, por la aplicación del deseo de volver a la posición en la formación.

### 3. Control de formación con velocidad variable con restricciones de giro.

La idea base es que el robot líder envía las posiciones en las que deben estar sus seguidores, lo interesante del método es que se puede hacer fácilmente a un robot seguidor, líder de otro seguidor. El planteamiento reactivo que bien podría ejemplificarse con la fábula de la zanahoria y el burro, donde al burro le ponemos una zanahoria y él con ganas de comerla nos sigue, y de esta forma lo dirigimos hacia donde queremos. Es lo mismo que ocurre con los robots seguidores siguen unas posiciones imaginarias inalcanzables a velocidades relativas al líder. Si alcanzamos la posición será por que el líder ha parado y en ese momento también pararemos, siguiendo con la analogía, el burro se habrá comido la zanahoria. Se desarrolla una ley de control de velocidad para los seguidores, dejando en segundo plano la tarea de planificar trayectorias, construida una trayectoria y determinadas las velocidades para seguirla teniendo en cuenta que las curvas deben ser suaves, por aquello de la restricción en el giro, el control funcionaría correctamente. Se ha limitado de manera analítica la restricción del giro a un radio de cuatro metros, nuestra única variable para el líder será su velocidad lineal  $v_l$  y determinaremos su velocidad angular  $\omega_l$  con la siguiente relación  $\omega_l = v_l/4$ , donde 4 es el radio en metros. Esa sería la restricción para un planificador de trayectorias, el radio mínimo de giro para ir de un punto a otro no deberá ser inferior, por supuesto no habría ningún problema en que fuera mayor. Un aspecto interesante para el futuro sería determinar matemáticamente el radio mínimo de giro que admite el tipo de formación deseada.

El desarrollo del algoritmo se divide en las siguientes partes:

1. Conocer la posición actual del líder y estimar su posición futura en un tiempo  $t$ .
2. Estimar la posición futura del seguidor relativa a la posición futura del líder y a su posición en la formación.
3. Calcular las velocidades necesarias para ir desde la posición actual del seguidor a su posición futura.

La implementación está centralizada en un programa que consigue las posiciones de los robots implicados en la formación y estimando sus posiciones futuras calcula las velocidades necesarias, encargándose de enviar las nuevas órdenes a los robots. En la implementación centralizada, recordemos que los robots están en movimiento, hacemos una lectura de su posición y hasta que no volvemos a hacer otra no sabemos que ha ocurrido, como es nuestro programa el que se encarga de comunicarle donde estará en un tiempo futuro debe ir lo más sincronizado con el tiempo real invertido en las lecturas. Para estar el menor tiempo posible “ciegos” se suele utilizar una frecuencia de muestreo alta en torno a las 10 Hz, en la robótica real ese tiempo tiene que ser suficiente para que los datos sean recibidos, para que los cálculos sean realizados, para que las órdenes sean enviadas, etc. La solución inmediata debida a la naturaleza centralizada de esta implementación era encerrar las lecturas de los sensores entre dos llamadas al tiempo del sistema del ordenador central y calcular la diferencia entre una y otra, de esta forma obtenemos el tiempo exacto de la lectura, fundamental para el correcto desarrollo de la solución, y con ello estimaremos más fielmente las posiciones futuras.

El primero de los pasos del algoritmo era, “Conocer la posición actual del líder y estimar su posición futura en un tiempo  $t$ ”, del líder conocemos su velocidad lineal, en nuestro caso se mueve a velocidad constante pero esto no es ni mucho menos una restricción, ya que es totalmente irrelevante la velocidad que lleve el líder. Obligamos, eso sí, para que el control funcione a que entre cada lectura de velocidad y la siguiente, las velocidades se mantengan constantes, es decir que no haya una aceleración. Esto no implica que el robot no pueda variar de velocidad, como demostrarán las pruebas realizadas. En cualquier caso, para solucionar la primera parte del algoritmo partimos de una posición a la que denominaremos posición actual expresada en coordenadas  $X$ ,  $Y$ , en metros y un determinado  $\alpha$  en radianes, esta posición está referenciada en coordenadas del mundo, a su vez, conocemos la velocidad lineal y angular con la que se está moviendo el robot, como sabemos el tiempo que ha transcurrido en las lecturas reales de los datos del robot, podemos estimar en que posición futura del mundo nos encontraremos tras dicho tiempo  $t$ , de la siguiente forma:

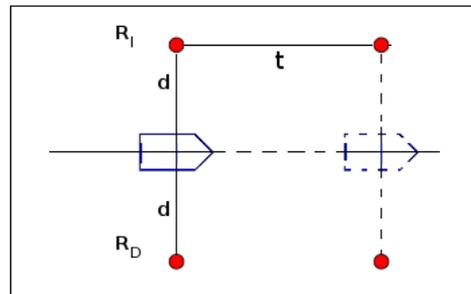
$$\begin{aligned}e &= v t \\ \alpha &= w t \\ F_x &= A_x + e \cos(\alpha) \\ F_y &= A_y + e \sin(\alpha) \\ F_\alpha &= A_\alpha + \alpha\end{aligned}\tag{2}$$

Donde  $v$  es velocidad lineal del robot líder,  $\omega$  es la velocidad angular del robot líder,  $t$  es el tiempo empleado en obtener las posiciones de todos los robots de la formación,  $e$  es el espacio recorrido a esa velocidad en ese tiempo,  $\alpha$  es el ángulo que habría girado el robot,  $F$  y  $A$  serían sendas estructuras de datos que contendrían lo necesario para referenciar la posición de robot en el mundo, sus coordenadas  $x$  e  $y$  y su ángulo  $\alpha$ ,  $F$  haría referencia a la posición futura del líder, obtenida a partir de la posición actual  $A$ , leída de las funciones encargadas de la odometría a bordo del robot.

El segundo de los pasos del algoritmo era, “Estimar la posición futura del seguidor relativa a la posición futura del líder y a su posición en la formación”. Digamos que vamos a proyectar la posición futura de un seguidor en relación a la posición futura de un líder, en  $t + 1$ , el líder si está en movimiento habrá modificado sus coordenadas y su ángulo, la idea original de decirle al seguidor donde tiene que estar, de ponerle la zanahoria para dirigirlo, encaja ahora perfectamente. Si pensamos en la forma elegida para construir una formación, simplemente con la distancia y el ángulo al que queríamos que se encontrase el robot seguidor teníamos suficiente, por lo tanto la posición futura del seguidor se resuelve con los mismos cálculos aplicados esta vez a la posición futura estimada del líder. La ecuación similar a la (1), quedaría descrita por la ecuación (3), en esta primera explicación omitimos el ángulo del robot seguidor. Por tanto pensemos en un movimiento uniforme rectilíneo con velocidad angular nula. La situación sobre la que vamos a calcular la velocidad estaría descrita en la imagen 2.

$$\begin{aligned} FS_x &= FL_x + d \cos(FL_\alpha + \alpha_l) \\ FS_y &= FL_y + d \sin(FL_\alpha + \alpha_l) \end{aligned} \quad (3)$$

En la ecuación (3)  $FS$  representa la posición futura del seguidor siendo una estructura como las anteriores, con las coordenadas  $x$  e  $y$  y el ángulo  $alpha$ ,  $FL$  representa la posición del líder mediante sus coordenadas  $x$  e  $y$  y su ángulo  $alpha$ ,  $d$  representa la distancia a la que queremos que se encuentre el seguidor respecto al líder. Estas variables junto con el ángulo al líder representado por  $\alpha_l$  nos situarán al robot en su posición futura.

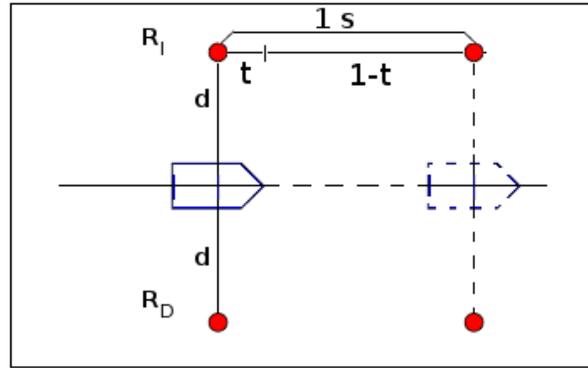


**Figura 2.** Formación de tres robots y sus posiciones futuras.

Desde el punto de vista del seguidor el control le está diciendo donde tiene que ir, y conocemos el tiempo empleado por el líder para llegar al futuro, con lo cual tenemos resuelta la velocidad lineal con la conocida relación entre el espacio y el tiempo,  $v = \frac{e}{t}$ . En este punto estaríamos en el paso tres del algoritmo propuesto, estaríamos estimando la velocidad del seguidor, es decir, resolviendo la ley de control, al menos para la ausencia de giros. La mayoría de las personas darían por bueno lo anterior, si tengo que estar en una posición y dispongo de un tiempo para llegar a ella, tengo que ir a la velocidad que me haga estar en ese punto cuando haya pasado ese tiempo, y parece que si pensamos en el futuro, el líder seguirá avanzando y volverá a pedirme que esté en una determinada posición en un nuevo intervalo de tiempo, aumentaré o disminuiré mi velocidad en función de la distancia y todo funcionará correctamente. El problema surge al plantearse ¿que pasa cuando alcance mi posición?, volviendo a la fábula, ¿si el burro muerde la zanahoria, que le pasa al burro? que se para, la idea de que la zanahoria dirige al burro se centra en que nunca la alcance. En nuestro caso cuando estemos en posición debemos mantenernos a la velocidad del líder. En el caso de ir recto, si no tenemos en cuenta esto los robots seguidores “van a saltos”, de pronto alcanzan su posición y su distancia a la siguiente es mínima, en la siguiente iteración es posible que el tiempo empleado fuese mayor y la velocidad necesaria muy alta y entre que se produce la nueva lectura incluso adelante al líder que sigue ajeno a los movimientos de sus compañeros de formación.

Para resolver este problema la idea ha sido extender la posición futura del líder a la hora de calcular la proyección futura del seguidor. Conocemos la posición futura del líder y su velocidad si hacemos que el cálculo siempre sea donde estará el líder tras un tiempo constante, por ejemplo, un segundo, en movimiento a esa velocidad, simplemente estamos alejando la posición futura del seguidor para calcular la velocidad que debo mantener para alcanzar dicha posición en un segundo. No es lo mismo que esperar a que el líder se mueva durante un segundo. La posición futura de la que partimos, aunque virtual, era la estimada en un tiempo  $t$  conocido, pero era real porque sería la que alcanzaría el líder en ese tiempo, ahora se crea otra virtual más lejos en el futuro, simplemente para conseguir una ley de control que resuelva de manera elegante, la aproximación a la posición deseada en la formación y dote al seguidor de la capacidad de mantenerse a la velocidad adecuada cuando esté correctamente posicionado dentro de la formación. Todo queda más claro observando la figura 3. Donde  $R_I$  y  $R_D$  representan con sendos puntos, al robot izquierdo y derecho de la formación respectivamente.

De la figura 3 y su explicación obtenemos la nueva ecuación (4) que estima la proyección futura del seguidor respecto a la posición futura del líder. Donde  $FS$  y  $FL$  son estructuras como las anteriores, con las coordenadas  $x$  e  $y$  y el ángulo  $alpha$  que representan la posición futura del seguidor y la posición futura del líder respectivamente,  $d$  representa la distancia a la que queremos que se encuentre el seguidor respecto al líder, esta variable junto con el ángulo respecto al líder representado por  $\alpha_l$  nos situarían al robot en su posición futura.  $vl$  es la velocidad lineal del líder en ese instante, al multiplicarla por el tiempo que nos



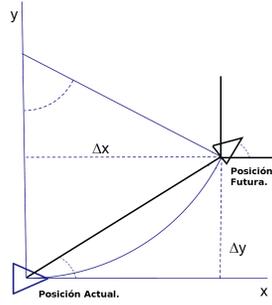
**Figura 3.** Formación de tres robots y la hipotética tras un segundo.

falta para movernos durante un segundo a esa velocidad  $(1 - t)$  obtenemos la proyección futura en un segundo.

$$\begin{aligned} FS_x &= FL_x + vl(1 - t) \cos(FL_\alpha) + d \cos(FL_\alpha + \alpha_l) \\ FS_y &= FL_y + vl(1 - t) \sin(FL_\alpha) + d \sin(FL_\alpha + \alpha_l) \end{aligned} \quad (4)$$

Cuando el movimiento es curvilíneo, nuestra velocidad angular es distinta de cero en esta solución aproximamos la velocidad angular a la diferencia de ángulo entre la orientación inicial del robot y la orientación futura y el tiempo invertido. Se vuelve a aprovechar la naturaleza de la solución para resolver de forma sencilla y con pocos cálculos, la velocidad angular a la que debe ir el seguidor. La estimación es aproximar la curva continua que une los dos puntos, su arco, con la línea recta que también los une. Una vez más en la robótica real la velocidad del movimiento lineal es pequeña, normalmente no supera el metro por segundo, y el tiempo de muestreo es alto, implican que el avance entre lectura y lectura, sea pequeño, por tanto la longitud del arco entre dos puntos  $A$  y  $B$  y la longitud de la línea recta que une ambos, es similar. La figura 4 muestra de forma ampliada, lo anterior.

En las ecuaciones (3), se añadiría un término más relativo al ángulo final futuro estimado que posee el robot  $FS_\alpha$ , este ángulo se obtendría con la función *atan2* de la librería *math.h* de C++ [10], su cometido es calcular la arcotangente de  $\frac{\Delta y}{\Delta x}$  en radianes teniendo en cuenta su signo. Por tanto el nuevo término de la ecuación sería el descrito en (5). Donde  $\Delta x$  y  $\Delta y$  expresan los incrementos en  $x$  e  $y$ , y  $\Delta\alpha$  el incremento del ángulo entre el instante actual y el futuro,  $FS$  es la posición futura del seguidor y  $AS$  es la posición actual del seguidor, son estructuras de datos para representar la posición del robot, necesitando por tanto, sus coordenadas  $x$  e  $y$  y su ángulo *alpha*. Desde el punto de vista del robot este ángulo debe ser sumado al ángulo que ya lleva, para que la odometría estimada



**Figura 4.** Aproximación del arco de la circunferencia.

del futuro se aproxime lo máximo posible a la real cuando haya transcurrido el tiempo.

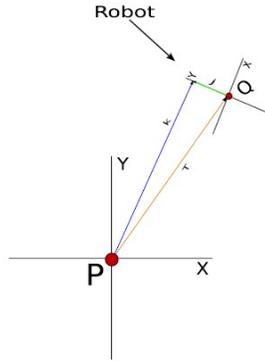
$$\begin{aligned}
 \Delta x &= FS_x - AS_x \\
 \Delta y &= FS_y - AS_y \\
 \Delta \alpha &= \text{atan2}(\Delta x, \Delta y) \\
 FS_\alpha &= AS_\alpha + \Delta \alpha
 \end{aligned} \tag{5}$$

El cálculo de la velocidad angular  $\omega$  para el seguidor consiste con este planteamiento en una simple resta entre el ángulo futuro estimado del robot y el ángulo que posee en ese instante. La velocidad angular se entiende entonces como: estoy con una orientación determinada y tras un tiempo  $t$  quiero acabar con otra orientación ¿a qué velocidad angular necesito girar para conseguirlo?, pues sencillamente, a lo expresado en (6). Si además aplicamos la misma idea de extender la  $\omega$  a intervalos de un segundo tenemos resuelta la ley de control para la velocidad lineal y angular, además el movimiento descrito por el robot seguidor mientras traza una curva será muy similar al que representa el arco, ya que no solo esta girándose sobre sí mismo y luego yendo hacia la posición futura, sino que a medida que va avanzado va girando.

$$\omega = \frac{FS_\alpha - AS_\alpha}{t} \tag{6}$$

Necesitamos una operación crucial en la solución propuesta: hacer un cambio al sistema de referencia del robot de la posición futura, a fin de decidir si el punto está detrás de él o delante. Para cambiar el sistema de referencia del mundo, entendido como el plano 2D en el que se esta moviendo el robot respecto al sistema de referencia situado en el propio robot donde la coordenada  $X$  para valores positivos se extiende en el sentido del avance del robot y la coordenada  $Y$  para valores positivos es perpendicular a la anterior.

Para transformar un vector del sistema de referencia del mundo al sistema de referencia del robot, debemos aplicar la ecuación (7). Obtenida de la imagen 5.



**Figura 5.** Cambio de sistema de referencia en 2D.

$$\vec{j} = R^T (\vec{k} - \vec{T}) \quad (7)$$

Donde  $\vec{j}$  es el vector en el sistema del robot,  $R^T$  es la traspuesta de la matriz de rotación definida en (8) y  $(\vec{k} - \vec{T})$  es el vector diferencia entre  $\vec{k}$  y  $\vec{T}$ .

$$R = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \quad (8)$$

En el caso en el que punto esté delante del robot su coordenada  $X$  en su sistema de referencia será positiva, y será negativa, por tanto, en caso contrario. El control comprobará el signo y decidirá que hacer, operando de la siguiente forma, en caso de que el punto esté efectivamente delante del robot estimará la velocidad lineal para llegar a él como hemos visto anteriormente si por el contrario se encuentra detrás, quiere decir que estamos mal orientados, debemos girar para alcanzar nuestro objetivo, por lo tanto su velocidad angular tendrá algún valor. En general siempre se hallará la velocidad angular ya que esta será cero por las propias ecuaciones, y para la velocidad lineal se tendrá en cuenta la situación del punto, igualándola a cero si se encuentra detrás. Gracias a este método podríamos incluso utilizar una velocidad lineal negativa para que el robot fuese hacia su posición en la formación girando y avanzando marcha atrás pero por motivos meramente visuales se ha decidido, parar al seguidor y girarlo. Para el giro de los seguidores no se han tenido en cuenta restricciones con lo cual su comportamiento no es distinto al de cualquier otro robot no holomónico, pudiendo girar sobre sí mismo. La esencia del algoritmo permite que los robots pudiesen estar dotados con sistemas locales para evitar obstáculos pudiendo desviarse de su posición en la formación sin miedo, ya que siempre que no le ocurriese nada al líder, seguirían obteniendo, posiciones a las que dirigirse, esta idea esencial es, sin duda, la clave de todo el desarrollo y la que le confiere robustez y sencillez. Por el contrario la limitación del giro, puede desembocar en problemas en el

movimiento de la formación. El más claro es que si la formación se dirige hacia un callejón sin salida, simplemente no puede salir. Pensemos en un automóvil, en ellos existe también una restricción de giro y no pueden girar sobre sí mismos, es decir describen una curva de un determinado radio y cuando ocurre una situación como la descrita, la única opción es dar marcha atrás para salir del atolladero. Al enfocar nuestra solución a la restricción del giro, por otro lado inevitable si queríamos describir curvas sin perder la formación, nos encontramos con estas dificultades, si queremos que la formación navegue libre de obstáculos sin perder en ningún momento la forma, debemos incluir un planificador de trayectorias a más alto nivel que los guíe a su destino.

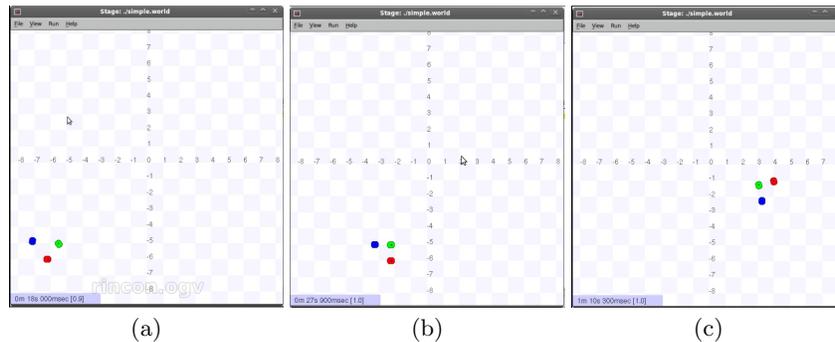
## 4. Experimentos

Para los experimentos se ha utilizado el software de simulación Player/Stage [21]. Player es un servidor que permite controlar los dispositivos de un robot y obtener información de sus sensores. Es una capa software que abstrae los detalles hardware del robot. Los algoritmos de control de los robots funcionan como clientes del servidor Player [5]. La arquitectura de este software sigue un modelo cliente-servidor sobre una red TCP/IP. El servidor Player se ejecuta en un robot, permitiendo a un programa cliente (o varios concurrentemente) leer datos de los sensores y enviar comandos a los actuadores del robot. Por su parte, Stage es un software de simulación de sistemas robóticos y sistemas percepto-efectores que evolucionan también en un entorno simulado, normalmente de interior, de dos dimensiones. Player puede ejecutarse utilizando Stage como sistema robótico simulado, en lugar de un sistema real.

El modelo de robot elegido es un robot Pioneer [7] este es un robot diferencial, con tres ruedas: dos motrices, con sendos motores, y una rueda loca.

Como se ha comentado, si queremos que los robots no pierdan la forma en un mundo real deben ir por un camino libre de obstáculos y lo suficientemente espacioso para la dimensión de la formación y el objeto que transporten, si por el contrario se admite una formación flexible deben estar provistos de sistemas de navegación locales a bordo de cada uno para evitar colisiones. En el siguiente vídeo se demuestra la validez de la solución, por su puesto se han realizados otras pruebas, con las que se ha pretendido abarcar las situaciones posibles a las que se puede enfrentar una formación en movimiento, como puede ser tener que frenar, tener que girar a ambos lados, variar la velocidad, incluso variando el aspecto de la formación en triángulo, en convoy o en rombo. El vídeo muestran un tipo de formación poco habitual en una especie de  $\neg$  donde el robot líder es el posicionado en el extremo inferior derecho. En la figura 6 se muestra una secuencia de imágenes del experimento. Al principio los robots parten de una posición distinta a la deseada y a lo largo del tiempo se observa como, gracias al comportamiento reactivo, se colocan correctamente mientras van navegando. La distancia de separación entre todos es de un metro. Recordamos que nuestra manera de definir la estructura de la formación es en relación al líder, con lo cual

el robot azul se encuentra, según el conocido teorema de Pitágoras a  $\sqrt{2}$  metros del líder, el robot rojo.



**Figura 6.** Experimento de navegación de la formación. Capturas del vídeo demostrativo *rincon.org*. (a) Sin formación. (b) Formación avanzando. (c) Formación girando.

Los resultados del experimento son satisfactorios, el movimiento de los robots es preciso y no pierden la forma cuando trazan curvas. En este vídeo la formación de la secuencia de capturas expuestas en 6, describe un movimiento pseudo-aleatorio variando para el líder su velocidad lineal incrementando y/o descendiendo según el cuadrante en el que se encuentre, primero describe un movimiento recto, luego giran a la izquierda al empezar el cuadrante negativo y más tarde lo hacen a la derecha, con objeto de demostrar el funcionamiento correcto del código implementado. El movimiento se realiza en un escenario completamente limpio de obstáculos, se ha dejado por tanto para futuros trabajos la integración con un planificador de trayectorias junto con la resolución de manera matemática del radio mínimo de giro según el tipo de formación.

Reflexionando sobre la solución propuesta, la principal característica de la ley de control, es la estimación de las posiciones imaginarias dependiendo tras un tiempo  $t$  para obtener ese futuro hipotético, pero es al mismo tiempo su principal defecto, aún funcionando correctamente como se demuestra en los vídeos. La razón fundamental de esta afirmación, es que está demasiado orientada a un control centralizado, el bloque principal para calcular  $t$  es una secuencia de llamadas o lecturas a los sensores para cada robot de la formación, tras el bloque se calcula el tiempo invertido en esas llamadas y así podemos estimar el resto de operaciones.

## 5. Conclusiones

Un problema típico y común de los algoritmos de desarrollo basados en la odometría es su dependencia en que la estimación de la posición sea lo más

correcta posible. Los cálculos en entornos simulados entre la posición estimada y la posición real suelen ser muy exactos pero en el mundo real, debido a los errores sistemáticos o deterministas y los no sistemáticos o no deterministas [9], hacen que dicha estimación no sea tan exacta.

Además estos desarrollos tienen otro inconveniente, el problema del conocimiento del entorno, es decir, nuestro mundo tiene unas dimensiones, está representado por unas coordenadas  $X$  e  $Y$ , es un mapa rudimentario de los límites de la habitación, y está posicionado en una determinada coordenada respecto a ese sistema de referencia. Este problema es común en muchos algoritmos y siempre quedan relegados o bien a poseer un mapa estático del entorno previamente cargado en memoria, o bien a conseguir en tiempo real mediante el uso de distintos sensores, cámaras, láser, sonares..., un mapa del entorno y moverse en relación a un punto determinado. El uso de los mapas y los problemas de la localización son una constante en el mundo de la robótica [25] [13] [14].

El método propuesto en el que el líder es el que de alguna manera sabe realmente hacia donde se dirige, la tolerancia a fallos del resto es manifiesta, la formación no sabe de la existencia del resto y si fuera, Dios no lo quiera, un escuadrón de soldados dispuestos a atacar un objetivo, si alguno de los robots seguidores fuera abatido el resto seguiría adelante hasta perder a su líder o capitán.

Otras tareas mucho más interesantes para las que se puede utilizar la navegación de robots en formación es para realizar tareas cooperativas entre varios de ellos. El ejemplo típico consiste en una serie de robots limpiadores que van navegando juntos por un edificio y al llegar a una determinada habitación uno de ellos se sale de su posición y procede a limpiar la habitación. Habría que implementar este comportamiento en los seguidores, y por ejemplo que el líder esperase al final del pasillo enviando las posiciones donde deberían estar los seguidores. Cuando el robot termina de limpiar la habitación y vuelve al pasillo vuelve a su vez al estado de ir en formación, y se dirigiría a la posición que le está mandando el líder. Con el resto ocurriría lo mismo, una vez estuviesen en sus posiciones seguirían con la ruta.

El ejemplo mostrado en los vídeos es ideal para la tarea de arrastrar un objeto en el interior de la formación o para transportar un objeto encima, ya que en ningún momento se pierde la forma de la estructura imaginaria que los une, si colocásemos una carga pesada o en un ambiente hostil, como una plancha de acero, o que estuviesen los robots protegidos y el acero estuviese a altas temperaturas, podrían transportarla a otro punto siguiendo una ruta que cumpliera las restricciones del giro, como si de una cinta transportadora se tratase pero mucho más flexible a la hora de crear nuevos caminos.

De la misma forma con un sistema de navegación local a bordo de cada robot en caso de encontrar un obstáculo sería evitado y una vez fuera de peligro el concepto flexible de nuestra estructura le permite cambiar de forma. En el campo relativo a las ideas, de cómo resolver el comportamiento "formar", hay un abanico grande de propuestas sobre el concepto, sobre qué estructura crear, por ejemplo, mediante un control descentralizado y negociando entre los

robots que forma adoptar [20], o bien buscando la posición más apropiada de cada sujeto dentro de la misma [18], algoritmos que optimizan la creación de estas estructuras, e incluso plantean algoritmos para cambiar de forma según las circunstancias [12], estas ideas pueden ser aplicadas a nuestro concepto o idea de construcción de la formación. Además este tipos de formaciones que varían su aspecto tienen otro tipo de aplicaciones, quizás más cooperativas, como por ejemplo la exploración [6], los robots abandonan su posición y regresan según las circunstancias.

## Referencias

1. T. Balch and R. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transaction on Robotics and Automation*, 14(6), (1998).
2. T. Balch and M. Hybinette. Social potentials for scalable multi-robot formations. ICRA. U.S.A. (2000).
3. T. D. Barfoot and C. M. Clark. Motion planning for formation of mobile robots. *Robotic and Autonomous Systems*. (2003).
4. N. Basilico, N. Gatti and F. Amigoni Leader-Follower Strategies for Robotic Patrolling in Environments with Arbitrary Topologies. AAMAS. Hungary. (2009)
5. A. Botía, D. Gallardo, M. I. Alfonso. Comportamientos coordinados en formaciones de robots usando percepción visual y comunicación punto a punto. VII Workshop de agentes físicos. (2006).
6. W. Burgard, M. Moors, D. Fox, R. Simmons and S. Thrun Collaborative Multi-Robot Exploration. I.C.R.A. (2000).
7. J.M.Cañás. Programación de robots. Technical report TR-GSYC-2004-6. Universidad Rey Juan Carlos. (2004)
8. Y. U. Cao, A. S. Fukunaga, and A. B. Cooperative Mobile Robotics: Antecedents and Directions.(1997).
9. B. Curto, V. Moreno, F. J. Blanco Comportamientos odométricos. Máster en Sistemas Inteligentes.(2010)
10. C ++ Reference, <http://www.cppreference.com>
11. J. P. Desai, J. P. Ostrowski, and V. Kumar. Controlling formations of multiple mobile robots. In Proc. 1998 IEEE Int. Conf. Robotics and Automation, vol. 4, Leuven, Belgium, pp. 2864–2869, (1998).
12. J. P. Desai, J. P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Trans. on Robotics and Automation*, vol. 17, no. 6, December (2002).
13. H. Durrant-Whyte and T. Bailey Simultaneous Localization and Mapping: Part I. *IEEE Robotics and Automation Magazine* (2006).
14. H. Durrant-Whyte and T. Bailey Simultaneous Localization and Mapping: Part II. *IEEE Robotics and Automation Magazine* (2006).
15. A. Fujimori, T. Fujimoto and G. Bohács. Distributed Leader-Follower Navigation of Mobile robots. ICCA. Hungary. (2005).
16. L. Iocchi, D. Nardi, and M. Salerno. Reactivity and deliberation: a survey on multi-robot systems. *Lecture Notes in Computer Science*, volume 2103, (2001).
17. O. Khatib. The Potential Field Approach and Operational Space Formulation in Robot Control. In K. S. Narendra. *Adaptive and Learning Systems: Theory and Applications*. Plenum Press, New York, NY, USA, , p. 367-377, (1986).
18. M. Lemay, F. Michaud, D. Létourneau and J. Valin Autonomous Initialization of Robot Formations. I.C.R.A., (2004).
19. Robin R. Murphy. *An Introduction to AI Robotics (Intelligent Robotics and Autonomous Agents)*. MIT Press, (2000).
20. D. J. Naffin and G. S. Sukhatme. Negotiated Formations. In *Proceedings of the Eighth Conference on Intelligent Autonomous Systems*. (2004)
21. J. Owen. *How to Use Player/Stage*. (2009).
22. L. E. Parker. Current state of the art in distributed autonomous mobile robotics. In *Distributed Autonomous Robotic Systems*.(2000).
23. G. A. S. Pereira, A. K. Das, V. Kumar, and M. F. M. Campos. Formation control with configuration space constraints. (2003).

24. K. Tan and M. A. Lewis. Virtual structures for high-precision cooperative mobile robotic control. *Autonomous robots*. (1997).
25. S. Thrun. Robotic Mapping: A survey. Chapter 1 in *Exploring Artificial Intelligence in the New Millenium*. M. Kauffmann, (2002).