

HISTORIA

Mirosot Robot Soccer se juega en un campo de 150cm x 130cm con tres robots tratando de insertar una bola naranja de golf dentro de la portería del equipo contrario. Hay dos jugadores y un guardameta y el juego es dividido en dos periodos que duran 5 minutos cada uno. Micro robots están restringidos en dimensiones de 7.5cm de altura, ancho y largo. La fuente de poder del robot, el motor, el micro controlador, el módulo de comunicación, etc. están integrados dentro del armazón del robot y la comunicación externa solo puede hacerse a través de métodos inalámbricos. Los sistemas de Robot Soccer son básicamente tres, el sistema de visión, equipo de comunicación y una computadora. La cámara se monta a 2m sobre el campo y es usada para distinguir los robots y la bola. Para funciones que el robot no puede realizar por el mismo, el sistema de visión le ayudará en el problema. Después que la cámara graba o detecta los movimientos del robot envía la información y la procesa para así dar órdenes de respuesta a los robots.

Micro Robots podrán observarse muy sencillos pero requieren de una alta tecnología en comunicación para unir en red la computadora con todos los robots, ingeniería de control para el ágil movimiento de los robots, inteligencia artificial para detectar la posición de la bola y de los jugadores así como sus movimientos, programarlos para trabajar en conjunto y llegar a una solución táctica por medio de cálculos de las posiciones de los robots etc. Cada computadora de cada equipo es libre de mandar cualquier tipo de información, como la posición de la bola o de otros robots, velocidad y dirección, tácticas y estrategia, pero el control remoto hecho por humanos esta completamente prohibido.

INTRODUCCION

En este trabajo se pretendrá obtener un modelo valido para poder simular el comportamiento dinámico de un robot. Estos tipos de robots son capaces de seguir una trayectoria determinada

Para poder validar el modelo y efectuar diversas pruebas utilizaremos el paquete de software ECOSIMPRO. Este programa nos permitirá obtener la respuesta de nuestro sistema a partir de la implantación del modelo en diagrama de bloques.

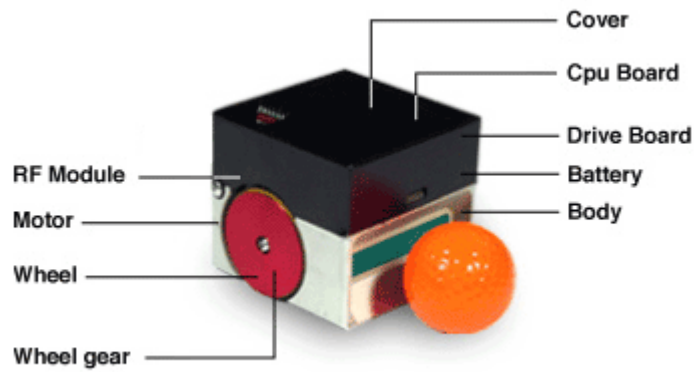
CARACTERISTICAS DEL ROBOT

Los robots presentan unas medidas de construcción de 7.5 x 7.5 x 7.5 (cm.) y un peso de 600g. consta de dos motores motrices. Los ejes de las ruedas están alineados en el centro de masa de los robots.

Las ruedas presentan un radio de 2cm y un ancho de 0.65cm. están impulsadas por 2 motores de corriente continua. La relación de transmisión de reducción es de 8:1 y las características del motor son las que figuran en la página siguiente.

COMPONENTES

YSR-A Type



Medidas 7.5 x 7.5 x 7.5 (cm.)

Peso 600 g

Motor Minimotor Swiss
(Encoder : 512
ppr)Procesador
32bit 80C296SA

Comunicación

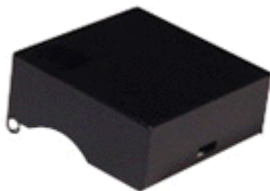
modulo RF (418 & 433 MHz) Programa : Visual C++

Velocidad Máxima

2.4 m/s

Distinciones

La ventaja de tener tu propio código fuente es que puedes modificarlo y así incluir tu propia estrategia en el programa.



CUBIERTA

Fabricada de acero para proteger el tablero de los impactos que existen en el fútbol de robots.



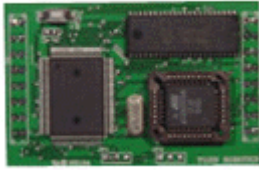
CUERPO

Hecho completamente de 'duralumin' para prevenir corto circuito interno y mantener la estabilidad.



Tablero de MANDO

Usa un tablero tipo 'L298' y así puede controlar dos motores con un chip.



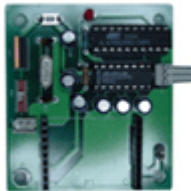
Tablero del CPU

Usa el INTEL 80C296 y el ROM (29C010).



Módulo RF

Usa un módulo 'RADIOMATRIX BIM 418 o el 433'.



Transmisor

Para la comunicación en serie.



MOTOR

MINIMOTOR SWISS DC motor (Tipo 2224 SR)
Encoder – Encoder integral IE2-512 512 líneas por revolución 2 canales, Output Digital.



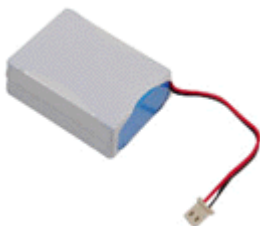
RUEDA

Usa un chip HP para la comunicación infrarroja.



ENGRANAJE

Los engranajes del motor están a un ratio de 8:1.



BATERIA

7.4V, 1200mA batería de lithium ion, protección del circuito incluida.

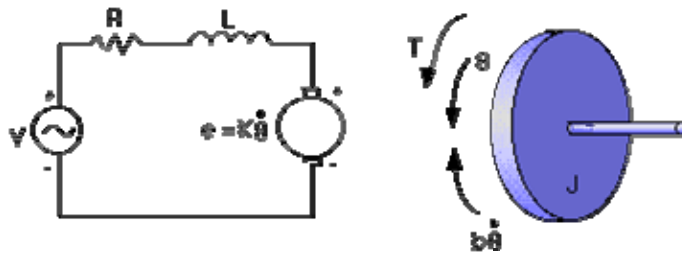
MODELIZACION DEL ROBOT

Después de detallar los componentes y las características de los robots el próximo paso es determinar las ecuaciones que regirán a nuestro sistema. Con el fin de simplificar un poco esta tarea se ha considerado conveniente efectuar la primer modelización del subsistema que formen los motores de corriente continua al que posteriormente se le agregaran los del movimiento del robot.

MODELIZACION DEL MOTOR DE CORRIENTE CONTINUA

El modelo de que se pretende obtener de este motor tiene como entrada un voltaje variable y nos proporcionara como salida en forma de velocidad angular, que nos servirá posteriormente para la entrada del subsistema del movimiento.

A continuación se muestra el esquema del motor de corriente continua



El motor de CC es un dispositivo que actúa sobre la potencia para proporcionar energía a una carga, como se muestra en la figura.

La función de transferencia del motor de CC se establecerá para una aproximación lineal de un motor real y se despreciaran los efectos de segundo orden, como la histéresis y la caída de potencia en las escobillas. El voltaje de entrada se aplica a las terminales de la armadura. El flujo en el espacio de aire del motor es proporcional a la corriente del campo, siempre que el campo no este saturado, de modo que:

$$\phi = K_f \cdot i_f$$

Se supone que el momento de torsión desarrollado por el motor esta relacionado linealmente con ϕ y con la corriente de la armadura como sigue:

$$T_m = K_1 \cdot \phi i_a(t) = K_1 \cdot K_f \cdot i_f(t) \cdot i_a(t)$$

Según esta ecuación es evidente que para tener un elemento lineal debe mantenerse una corriente constante, mientras que la otra se vuelve corriente de entrada. Primero consideraremos controlada la corriente del campo, lo que proporciona una sustancial amplificación de potencia. Entonces tenemos, en la notación de la transformada de Laplace,

$$T_m(s) = (K_1 \cdot K_f \cdot I_a) \cdot I_f(s) = K_m \cdot I_f(s)$$

Donde $i_a = I_a$ es una corriente de armadura constante y K_m se define como la constante del motor. La corriente del campo esta relacionada con el voltaje del campo por

$$V_f(s) = (R_f + L_f s) \cdot I_f(s)$$

El momento de torsión del motor $T_m(s)$ es igual al proporcionado a la carga. Esta relación puede expresarse como

$$T_m(s) = T_L(s) + T_d(s) \Rightarrow T_L(s) = T_m(s) - T_d(s)$$

Donde $T_L(s)$ es el momento de torsión de la carga y $T_d(s)$ es el momento de torsión de perturbación, el cual se debe tomar en cuenta ya que tenemos acción de fricción en nuestro sistema. El momento de torsión de la carga para una inercia de rotación como la mostrada anteriormente se escribe como

$$T_L(s) = Js^2(s) + fs\theta(s)$$

El *motor de cd controlado por la armadura* utiliza una corriente de campo constante y, por tanto, el momento de torsión del motor es

$$T_m(s) = K_m \cdot I_a(s)$$

La corriente de la armadura se relaciona con el voltaje de entrada aplicado a la armadura por

$$V_a(s) = ((R_a + L_a s) \cdot I_a(s)) + V_e(s)$$

Donde $V_e(s)$ es el voltaje de la fuerza electromotriz proporcional a la velocidad del motor. Por tanto, tenemos

$$V_e(s) = K_b \cdot \omega(s)$$

Y la corriente de la armadura es

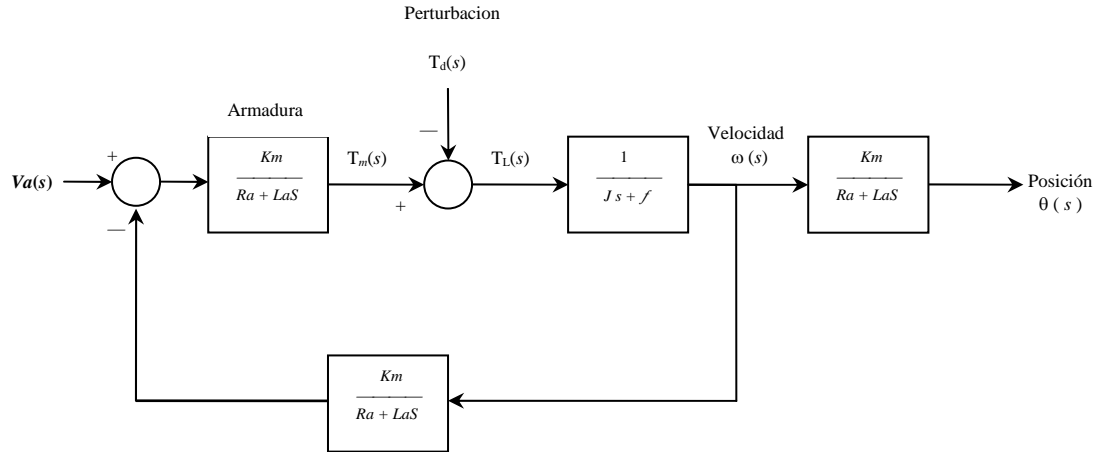
$$I_a(s) = \frac{V_b(s) - K_b \cdot \omega(s)}{R_a + L_a s}$$

Por lo tanto desarrollando nuestras ecuaciones quedan de la siguiente forma

- (1) $V_a(s) = ((R_a + L_a s) \cdot I_a(s)) + V_e(s)$
- (2) $V_e(s) = K_b \cdot \omega(s)$
- (3) $T_m(s) = K_m \cdot I_a(s)$
- (4) $T_L(s) = Js^2(s) + fs\theta(s)$

MODELO TEORIA DE CONTROL

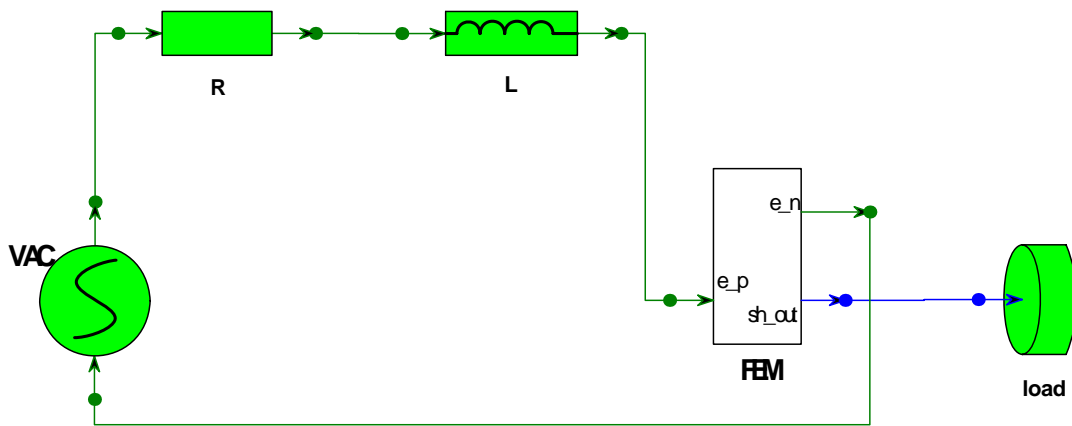
Si se muestra un diagrama de bloques de la modelización desarrollada para los motores de los robots se obtiene algo como lo que se muestra a continuación:



DESARROLLO DEL PROGRAMA EN ECOSIMPRO

Teniendo el modelo teórico y las ecuaciones de generación de la modelización, se puede realizar el programa en Ecosimpro. A continuación se muestran los programas realizados del circuito de los motores tanto en forma eléctrica como de forma de teoría de control. Esto se logro gracias al manejo de Smarts sketch.

ELECTRICO



PROGRAMA EN ECOSIMPRO

Con el modelo esquemático en Smarts sketch se realiza la conversión y compilación del modelo al entorno Ecosimpro. El programa del circuito de los motores queda de la forma siguiente:

```
-- Generated automatically with EcosimPro 3.2, Date: 15/04/2003  
-- Original SmartSketch symbol: motor.igr sheet: Topology
```

```
LIBRARY APROYECTO
```

```
USE ELECTRIC
```

```
COMPONENT motorelec
```

```
PORTS
```

```
TOPOLOGY
```

```
    ELECTRIC.R  R (  
      R = 36.30,  
      TR1 = 0,  
      TR2 = 0,  
      TR = 300)
```

```
    ELECTRIC.VAC  VAC (  
      VAC = 0.0,  
      freq = 0.0,  
      phase = 0,  
      VDC = 24)
```

```
    ELECTRIC.L  L (  
      L = 0.000008,  
      TL1 = 0,  
      TL2 = 0,  
      TL = 300)
```

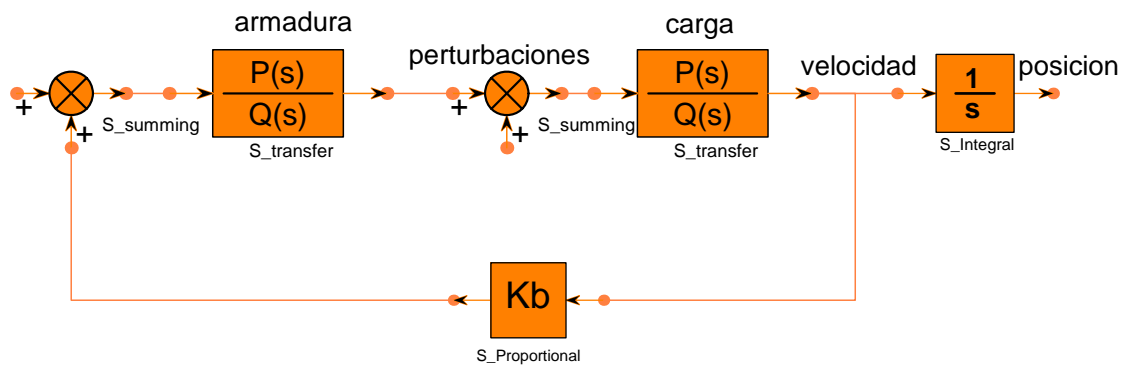
```
    ELECTRIC.load  load (  
      I = 0,  
      kf = 29.10)
```

```
    ELECTRIC.FEM  EMF (  
      k = 3.040)
```

```
CONNECT VAC.e_n TO R.e_p  
CONNECT R.e_n TO L.e_p  
CONNECT L.e_n TO EMF.e_p  
CONNECT EMF.e_n TO VAC.e_p  
CONNECT EMF.sh_out TO load.sh_in
```

```
END COMPONENT
```

TEORIA DE CONTROL



PROGRAMA EN ECOSIMPRO

Con el modelo esquemático en el formato de teoría de control en SmartsSketch se realiza la conversión y compilación del modelo al entorno Ecosimpro. El programa del circuito de los motores queda de la forma siguiente:

```
-- Generated automatically with EcosimPro 3.2, Date: 17/04/2003
17:40:51
-- Original SmartSketch symbol: cochinercontrol.igr sheet: Sheet1
-- Don't modify
```

LIBRARY APROYECTO

USE CONTROL

COMPONENT motorcontrol

PORTS

TOPOLOGY

```
CONTROL.S_transfer (
  nnum = 0,
  ndem = 1) armadura (
  p = {29.10},
  q = {36.30, 0.0008},
  output_o = 0)

CONTROL.S_summing summer_2

CONTROL.S_transfer (
  nnum = 0,
  ndem = 1) carga (
  p = {1},
  q = {2.6, 0.2},
  output_o = 0)

CONTROL.S_Integral Integral
CONTROL.S_summing summer1

CONTROL.S_Proportional kb (
  k = -3.04)
```

```
CONNECT armadura.s_out TO summer_2.s_in_1
CONNECT summer_2.s_out TO carga.s_in
CONNECT summer1.s_out TO armadura.s_in
CONNECT kb.s_out TO summer1.s_in_2
CONNECT carga.s_out TO Integral.s_in, kb.s_in

END COMPONENT
```

VALIDACION DEL MODELO

Para comprobar si el programa que tenemos arriba es correcto se harán una serie de comprobaciones después de obtener la aceptación de Ecosimpro (compilación).

EXPERIMENTO

```
EXPERIMENT expl ON motorcontrol.default

DECLS

INIT    -- set initial values for variables
        -- Dynamic variables
        Integral.s_out.signal = 0

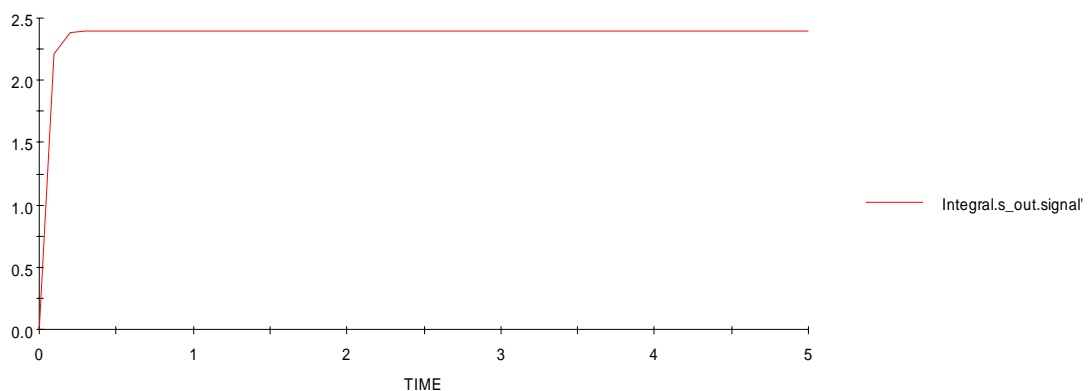
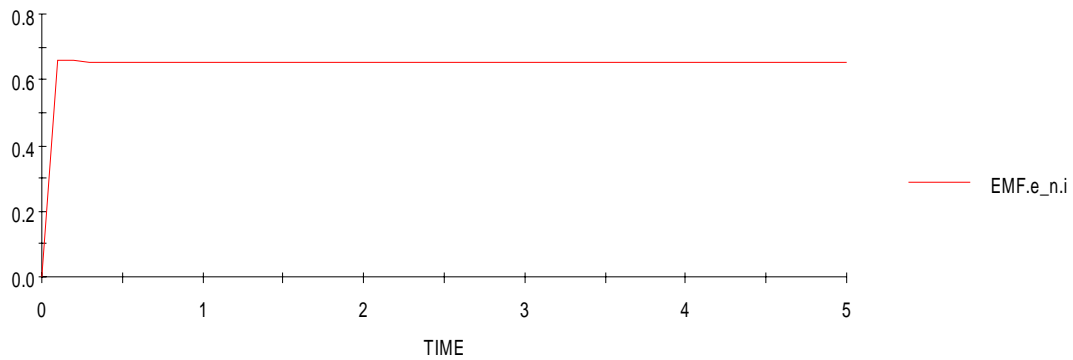
BOUNDS  -- set expressions for boundary variables: v = f(t,...)
        summer1.s_in_1.signal = 24
        summer_2.s_in_2.signal = -7.15

BODY
        REPORT_TABLE("reportAll", " * ")
        TIME = 0
        TSTOP = 5
        CINT = 0.1
        INTEG()

END EXPERIMENT
```

COMPROBACION

Se comprobará el trabajo realizando el experimento y obteniendo 2 graficas. Una de la corriente verificando que no sobrepase la intensidad máxima y la otra de la velocidad.



Como se aprecia en las graficas los datos obtenidos son razonables de acuerdo a las especificaciones del fabricante de los robots Robosot.

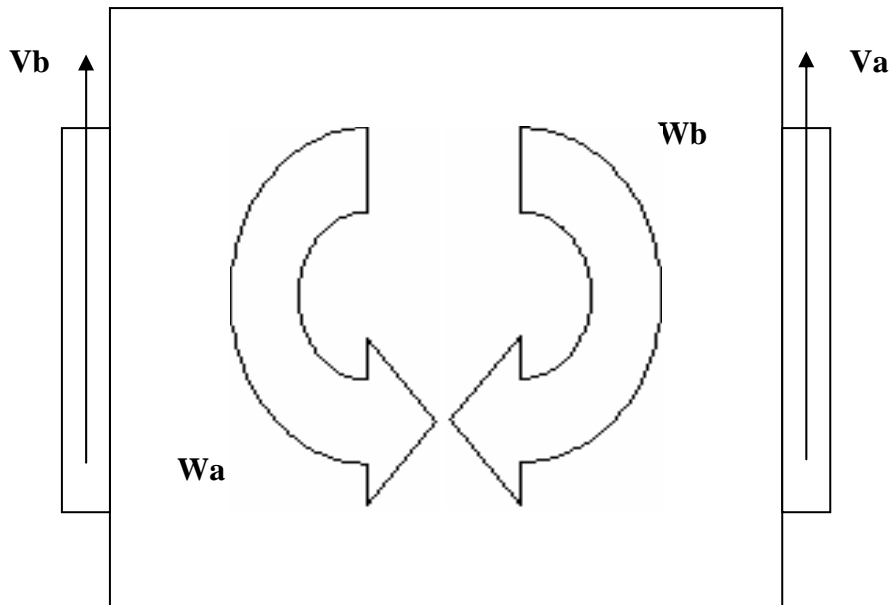
MODELO CINEMATICO DEL MOVIMIENTO DEL ROBOT

Lo importante de este proyecto es lograr que el modelo de los motores sea la entrada del modelo cinemático del movimiento del robot.

El primer punto nos interesa obtener el valor del ángulo de giro que efectuara nuestro robot cuando se introducen ciertas entradas. También habremos de hallar el valor de la posición 'X' y 'Y' en que se encontrará a cada instante de tiempo.

ANGULO DE GIRO

Si se considera toda la masa del robot como un sólido rígido, podemos observar que las velocidades suministradas por cada motor proporcionarían una velocidad angular determinada al sólido:



Si aplicamos y desarrollamos la ecuación cinemática de un movimiento de roto traslación de un sólido rígido se obtiene:

$$\omega = (V_b - V_a) / D$$

Integrando el valor de ω se obtiene directamente el valor del ángulo de giro.

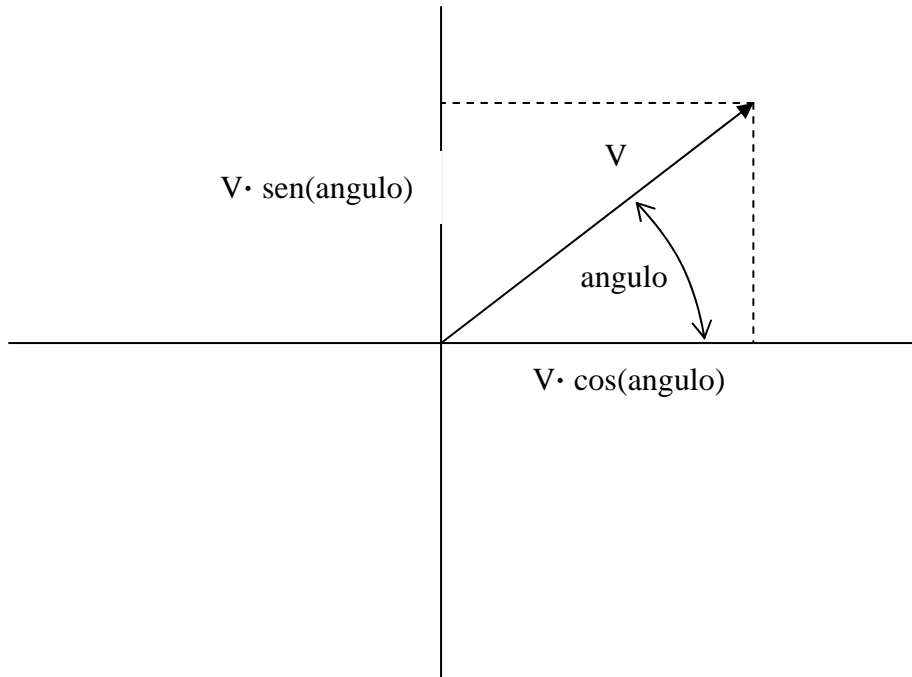
Ahora se buscará el valor de la velocidad lineal adquirida por el robot. Aquí se tiene que aplicar la ecuación anterior, pero en este caso se aplicará entre uno de los motores y el centro de gravedad del robot.

$$V_b = V_a + (\omega * D/2)$$

Ahora ya tenemos el ángulo de giro y la velocidad en que se moverá el robot.

El último paso a realizar será determinar la posición en 'X' y en 'Y' respecto a un eje de coordenadas fijas.

Para poder determinar las coordenadas habremos de fijarnos en que forma se representara la velocidad obtenida. Si se observa el siguiente diagrama:



Con este diagrama podemos afirmar que las velocidades en 'X' y en 'Y' podrán ser determinadas por las ecuaciones siguientes:

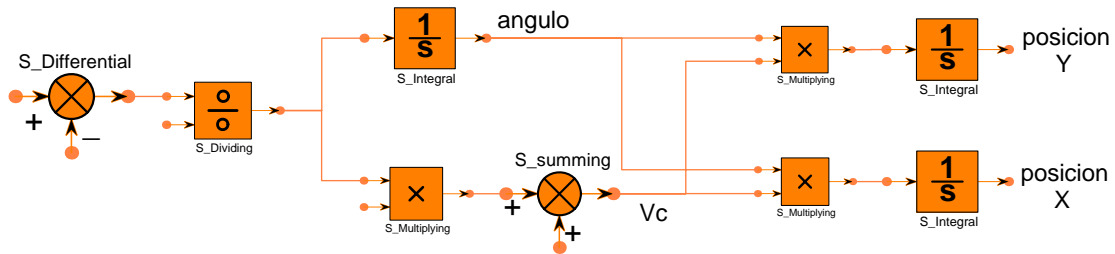
Donde φ representa el ángulo de giro.

$V_x = V_c \cdot \cos\varphi$ $V_y = V_c \cdot \text{sen}\varphi$

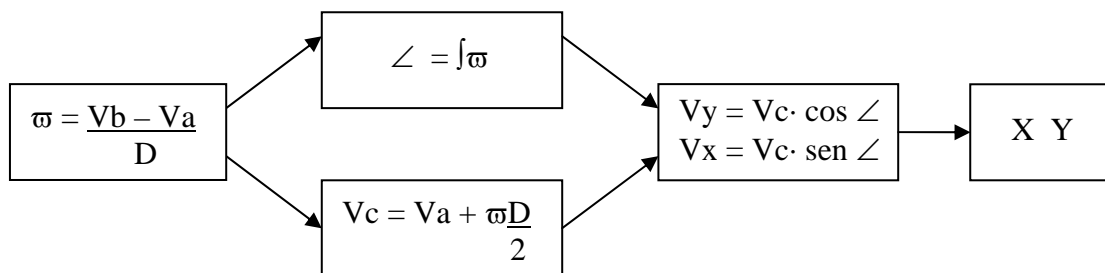
Ahora solo nos faltara integrar las velocidades para obtener la posición en cada instante.

IMPLEMENTACION EN ECOSIMPRO

Si pasamos las ecuaciones dinámicas obtenidas anteriormente a un diagrama de teoría de control a Ecosimpro obtendremos el modelo siguiente:

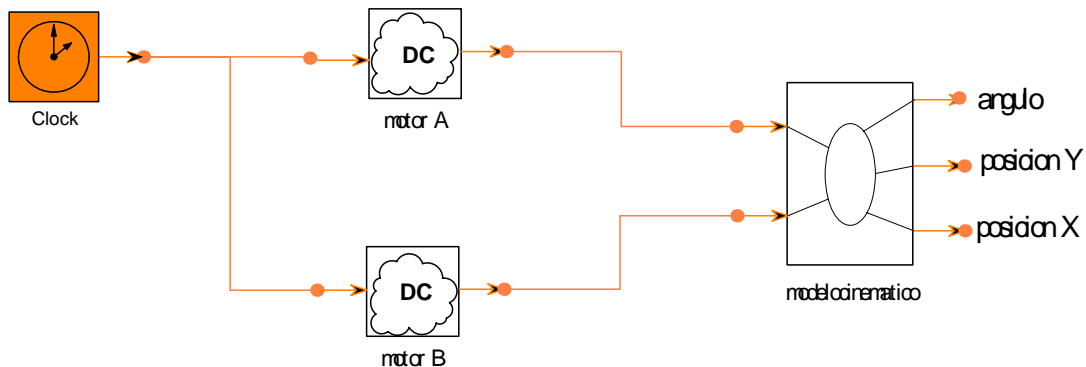


Que significa lo mismo que lo siguiente sustituyendo la forma en la que se obtienen las variables con las ecuaciones dinámicas para no perdernos en el diagrama de teoría de control



AGRUPAMIENTO DE LOS SUBSISTEMAS EN EL DEL MODELO FINAL

Ya se tienen los dos submodelos que intervienen en nuestro sistema. Ahora necesitamos agruparlos en nuestro modelo final. Para facilitar la comprensión y facilitación del sistema se ha agrupado cada submodelo y hecho el modelo final de la siguiente forma:



PROGRAMA FINAL

```
-- Generated automatically with EcosimPro 3.2, Date: 20/04/2003
-- Original SmartSketch symbol: proyecto final.igr sheet: Hoja1
-- Don't modify
```

```
LIBRARY APROYECTO
```

```
USE CONTROL
```

```
COMPONENT realizado2
```

```
PORTS
```

```
TOPOLOGY
```

```
CONTROL.S_summing S_summing2

CONTROL.S_Differential S_Differential1

CONTROL.S_Dividing S_Dividing

CONTROL.S_Integral S_Intangulo

CONTROL.S_Multiplying S_Multiplying

CONTROL.S_Multiplying seno S_Multiplying y

CONTROL.S_Multiplying coseno S_Multiplying x

CONTROL.S_Integral S_Integral x

CONTROL.S_Integral S_Integral y

CONTROL.S_transfer (
  nnum = 0,
  ndem = 1) armadura (
  p = {29.10},
  q = {36.30, 0.0008},
  output_o = 0)

CONTROL.S_summing summer_2

CONTROL.S_transfer (
  nnum = 0,
  ndem = 1) carga (
  p = {1},
  q = {2.6, 0.2},
  output_o = 0)

CONTROL.S_Integral Integral

CONTROL.S_summing summer1

CONTROL.S_Proportional kb (
  k = -3.04)

CONTROL.S_transfer (
  nnum = 0,
  ndem = 1) armadurab (
  p = {29.10},
  q = {36.30, 0.0008},
  output_o = 0)

CONTROL.S_summing summer_2b
```

```

CONTROL.S_transfer (
  nnum = 0,
  ndem = 1) cargab (
  p = {1},
  q = {2.6, 0.2},
  output_o = 0)

CONTROL.S_Integral  Integralb

CONTROL.S_summing  summer1b

CONTROL.S_Proportional  kbb (
  k = -3.04)

CONTROL.S_summing  posicioncentro

CONNECT S_Dividing.s_out TO S_Multiplying.s_in_1, S_Intangulo.s_in
CONNECT S_Multiplying.s_out TO S_summing2.s_in_1
CONNECT S_Differential1.s_out TO S_Dividing.s_in_1
CONNECT S_Multiplyingy.s_out TO S_Integraly.s_in
CONNECT S_Multiplyingx.s_out TO S_Integralx.s_in
CONNECT S_Intangulo.s_out TO S_Multiplyingy.s_in_1,
S_Multiplyingx.s_in_1
CONNECT S_summing2.s_out TO S_Multiplyingx.s_in_2,
S_Multiplyingy.s_in_2
CONNECT armadura.s_out TO summer_2.s_in_1
CONNECT summer_2.s_out TO carga.s_in
CONNECT kb.s_out TO summer1.s_in_2
CONNECT carga.s_out TO Integral.s_in, kb.s_in,
S_Differential1.s_in_2, S_summing2.s_in_2, posicioncentro.s_in_1
CONNECT summer1.s_out TO armadura.s_in
CONNECT armadurab.s_out TO summer_2b.s_in_1
CONNECT summer_2b.s_out TO cargab.s_in
CONNECT kbb.s_out TO summer1b.s_in_2
CONNECT cargab.s_out TO Integralb.s_in, kbb.s_in,
S_Differential1.s_in_1, posicioncentro.s_in_2
CONNECT summer1b.s_out TO armadurab.s_in

END COMPONENT

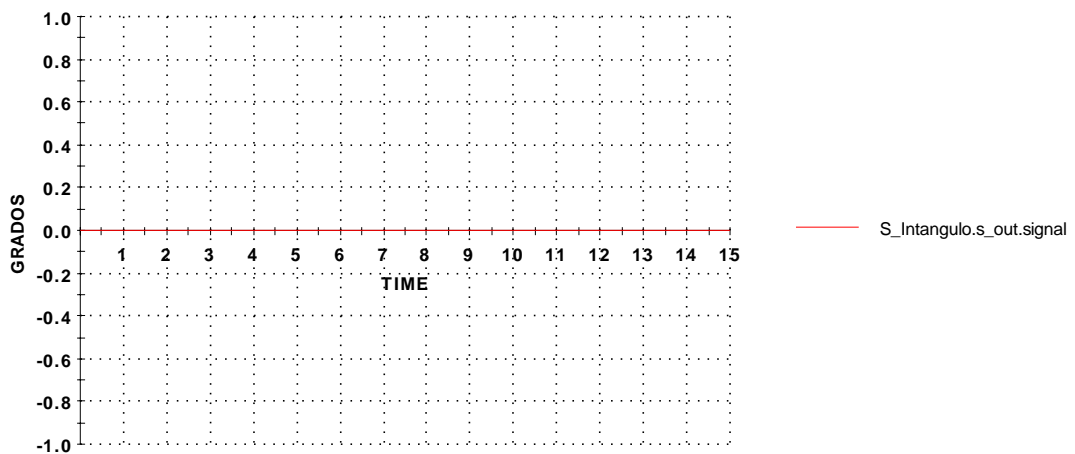
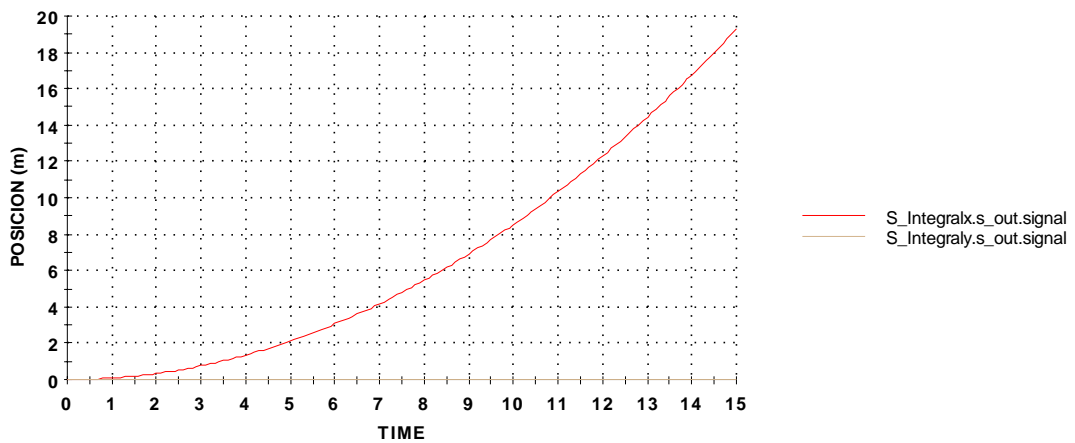
```

VERIFICACION Y VALIDACION DEL MODELO FINAL

Una vez obtenido el modelo se habrá de efectuar una serie de pruebas básicas con tal de determinar si el modelo se ajusta a nuestros propósitos, y si responde adecuadamente a las señales que se programen y asegurarnos que se esta obteniendo una respuesta fiable de nuestro sistema simulado.

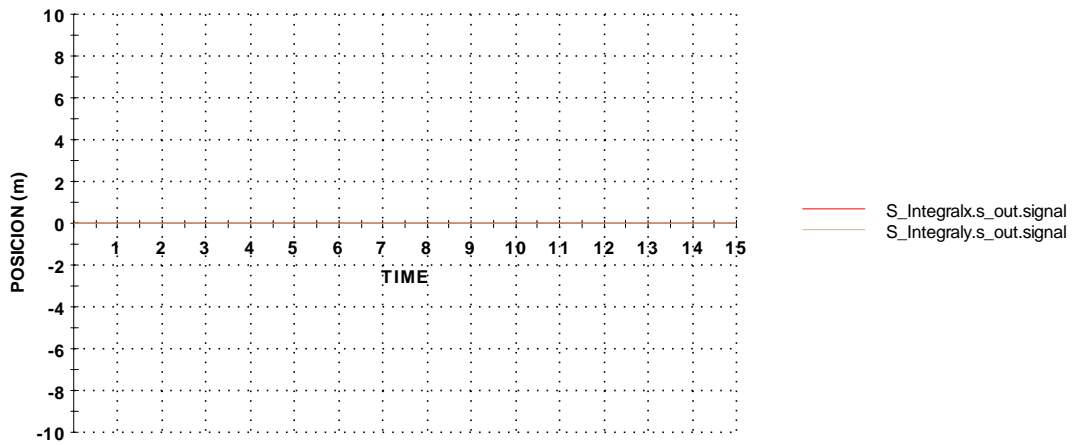
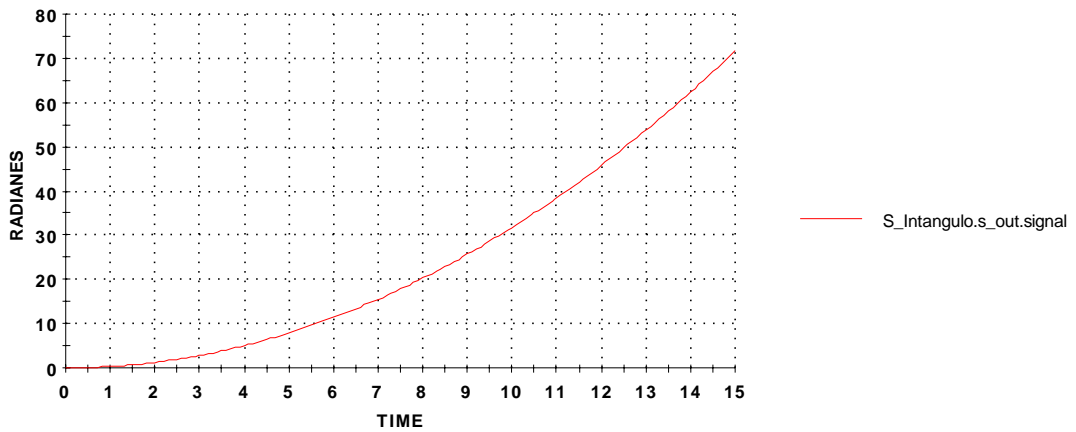
PRUEBAS BASICAS

La primera prueba que se ha realizado consiste en comprobar si cuando se le administra la misma tensión a los dos motores efectivamente realiza una trayectoria rectilínea. El resultado, aplicando una tensión de 10 volts a cada motor es el siguiente:



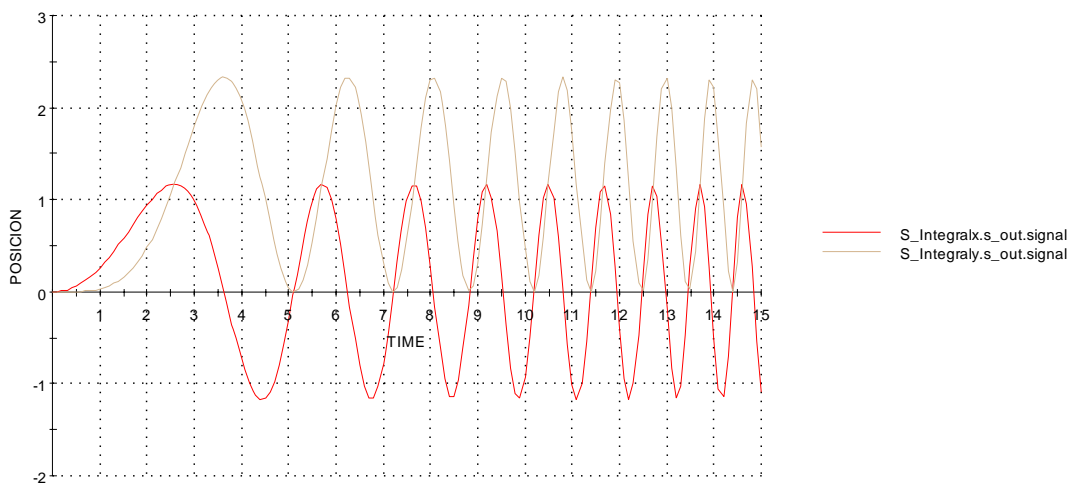
Vemos que de momento el sistema respondió de la manera prevista.

Una segunda prueba será suministrar al sistema tensiones de la misma magnitud pero de polaridad contraria; al motor izquierdo 10 volts y al derecho -10 volts. En este caso el robot debe de mantenerse quieto en la posición inicial de movimiento de rotación puro. Los resultados son los siguientes:

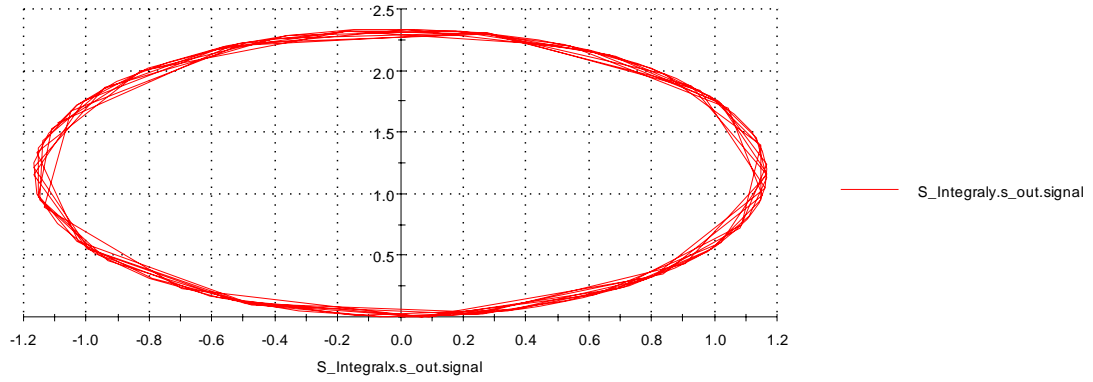


Vemos como efectivamente sucede esto; No existen movimientos en X ni en Y y el ángulo de giro se dispara indefinidamente.

Otra comprobación que podemos realizar es si se suministra una tensión mayor a uno de los motores y ver si se produce una trayectoria curvilínea



Ahora veremos como se observa en un plano X Y comun



CONCLUSION

La modelización realizada nos va a servir posteriormente para comprobar las especificaciones del fabricante de los robots Robosot: Yuijn Electronics.

Se podrán efectuar varias pruebas reales de movimiento y sincronización de los robots y después poderlas comprobar por medio del programa generado en Ecosimpro. Este proyecto sirve también como un apoyo para la modelización en este software y aprender el manejo de entorno que se maneja con el software Smartsketch.

Este modelo nos sirve para cualquier robot rastreador de dos motores de CD como entrada y de salida dos ruedas en posiciones paralelas. El programa está compuesto solo con el esqueleto del programa para analizar al robot rastreador en cuestión y el investigador es el encargado de darle los valores correspondientes para obtener la modelización correcta y así compararla con el experimento real.

Lo que se debe tomar en cuenta como más importante es que para cada modelo se tiene que hacer una serie de comprobaciones como las que se generaron en este trabajo de investigación. Las pruebas obtenidas son los datos expuestos en las gráficas de control de las páginas anteriores.