

Práctica FINAL

Diseño de Controladores Borrosos y Neuroborrosos para un Robot Móvil



26 de enero de 2020

pastor moreno, javier

DURAND BARTOLO, DANIELA CAMILA

Contenido

[Parte 1: Control Borroso 2](#_Toc62431890)

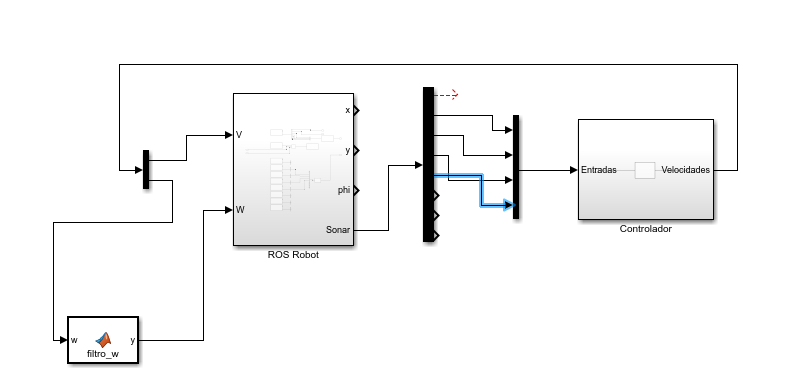
[Controlador 3](#_Toc62431891)

[*Reglas de inferencia* 5](#_Toc62431892)

[Parte 2: Diseño automático de un controlador neuroborroso tipo SUGENO. 7](#_Toc62431893)

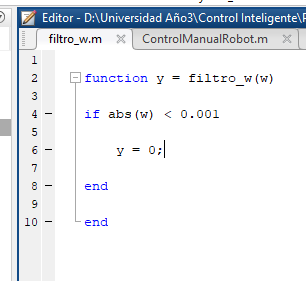
# Parte 1: Control Borroso

En primera instancia, deberemos de llevar a cabo la configuración del modelo Simulink, para lo cual, llevaremos a cabo modificaciones sobre el bloque del robot que nos ha sido proporcionado con el enunciado de la práctica.



De este esquema, resaltaremos varias cuestiones:

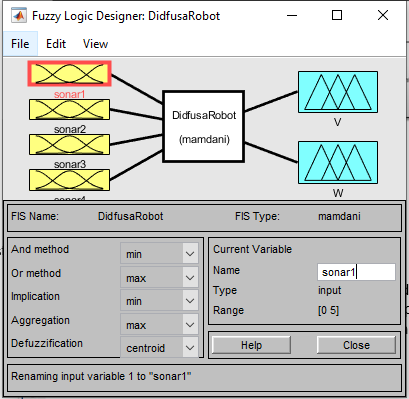
* La primera, la existencia del bloque de filtrado para la velocidad angular W, ya que en el caso de que el robot tenga una velocidad angular muy pequeña este se detendrá por un error, por lo que deberemos llevar a cabo un filtrado como el que puede verse a continuación:



* El segundo sería denotar que sónares se encuentran activos, ya que como puede observarse, solo obtendremos las salidas de los sónares 1,2,3,4.
* Por último, el controlador borroso, que tendrá como parámetro el archivo DidfusaRobot.fis que se detallará en el siguiente apartado.

## Controlador

A continuación se muestra una configuarción inicial del controlador que se ha empleado.



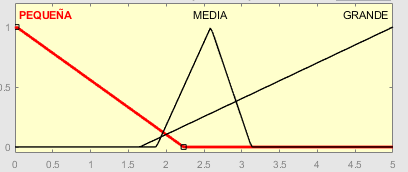
Como puede observarse, en el se encuentran los 4 sónares que se ha comentado previamente que iban a ser utilizados en forma de entradas, y con las salidas “V” y “W”, siendo cada una de ellas la velocidad y la velocidad angular respectivamente.

El empleo de estos 4 sónares no ha sido aleatorio, ya que al encontrarse en la parte delantera del vehículo permiten ir observando de forma constante y con antelación todos aquellos obstáculos o giros inminentes a los que se aproxime el robot.

Además de esto, existe una segunda razón para el empleo de estos sónares en concreto, y es que la cantidad de conjuntos borrosos a realizar se reducen a la mitad, debido a la simétrica disposición de los sónares 1,4 y 2,3. Por ello sólo será necesario mostrar a continuación los dos conjuntos borrosos para los pares mencionados previamente.

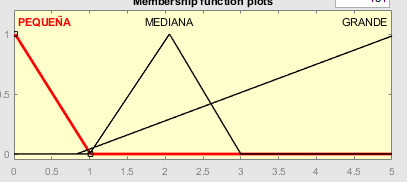
Sónares 1 y 4:

Al ser los que se encuentran más próximos a los laterales, hemos considerado que el momento de la activación de la distancia que denominamos pequeña debiere estar en un punto referente a los 2’2, de tal forma que en caso de que un giro fuese necesario por el lateral este fuese detectado de la mejor forma posible.



Sónares 2 y 3:

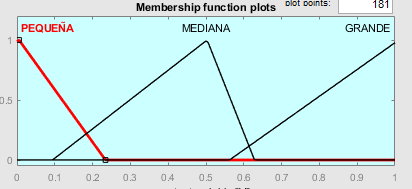
En este caso puede verse como la distancia considerada como cerca se ha reducido drásticamente debido a que es menos importante en estos sónares frontales ( Aunque de vital importancia ante un choque inminente), por lo que se ha elegido una mayor distancia para el parámetro denominado como “GRANDE” con la finalidad de poder ver los obstáculos a una mayor distancia y comenzar la maniobras de manera sutil en base a estos.



V:

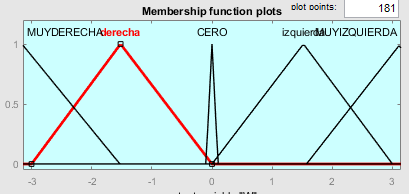
En cuanto a la velocidad, destacar que como se señalaba en el enunciado, está irá variando entre (0 y 1), empleando las velocidades pequeñas en las situaciones de mayor peligro y las distancias mediana y grande en aquellos momentos en los que las distancias que marquen los distintos sónares sea más lejana y permita una mayor velocidad.

Aunque se podrá ver en cuanto se revisen las distintas reglas de inferencia, en la práctica, la velocidad “MEDIANA” ha acabado por no usarse en el caso de que se quiera que el robot circule a la máxima velocidad posible sin chocarse ( A una velocidad media de unos 0.823ms), aunque para una mayor precisión y menor riesgo de colisión sería recomendable emplearlas en las reglas 7 y 8.



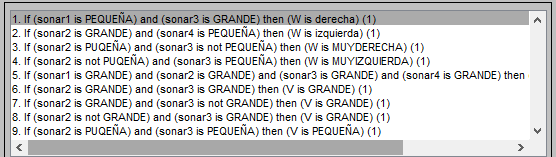
W:

Mediante la velocidad angular se irán controlando los distintos grados que el robot deberá realizar en cada momento, variando desde giros menos agresivos a más agresivos conforma más se acerque al obstáculo o a la curva.



## Reglas de inferencia

Las reglas de inferencia que se han creado son las siguientes:



Podría decirse que estás divididas en dos categorías.

De la regla 1 a la regla 4 se van a tratar de reglas que van a tratar principalmente los distintos giros.

El siguiente grupo, de la regla 5 a la regla 9, van a ir controlando la velocidad del robot.

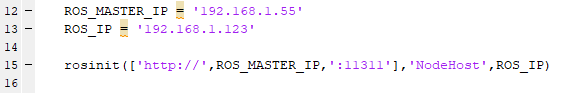
Como se ha comentado, para una mayor seguridad para evitar una colisión del robot, sería necesario modificar las velocidad de las reglas 7 y 8 a V MEDIANA, cuyo resultado sería una velocidad media de unos 0.56 ms, pero ya que uno de los objetivos de la práctica era intentar obtener la máxima velocidad posible, actualmente V se encuentra dispuesta en GRANDE para obtener una velocidad media de 0.85.

Con respecto a esta parte, cabe destacar que se intentó reducir el número de reglas a emplear, pero por desgracia el control simultáneo en una misma regla tanto de la velocidad como de la velocidad angular no dio lugar a buenos resultados, propiciando giros continuos sobre sí mismo, o constantes choques con los objetos.

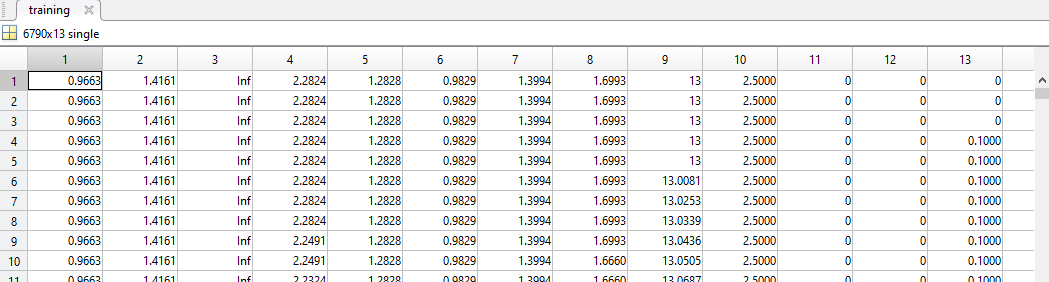
# Parte 2: Diseño automático de un controlador neuroborroso tipo SUGENO.

2. Diseño automático de un controlador neuroborroso tipo SUGENO

En primer lugar, para poder obtener nuestro controlador tipo SUGENO es necesario obtener datos con los que realizar el entrenamiento. Para ello utilizamos el script proporcionado *ControlManualRobot.m*, en el que configuramos las direcciones IP de nuestra máquina y de la máquina virtual.



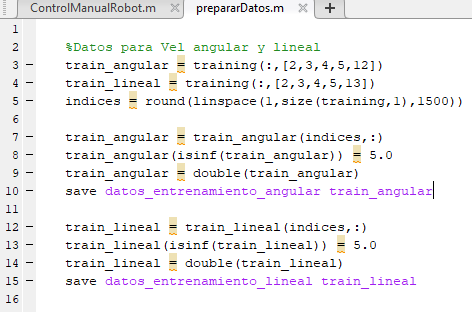
Tras ello arrancamos el simulador en la máquina virtual y ejecutamos el script, con el que cogeremos los datos obtenidos de los sensores del robot teleoperándolo, además de su posición y sus velocidades lineal y angular continuamente. Estos datos se guardan en la variable training, que a su vez se guarda en un .mat (*datos\_entrenamiento.mat*).



Como no podemos utilizar todos los datos obtenidos, nos quedamos con las columnas correspondientes a los sensores cuyas medidas queremos utilizar en el entrenamiento y el valor a predecir (velocidades angular y lineal). Además, cogeremos 1500 del total de filas de la variable training repartidas uniformemente para que los datos sean lo más representativos posible. También es necesario poner a 5 los valores que figuren como infinito, ya que ese es el alcance máximo de los sensores.

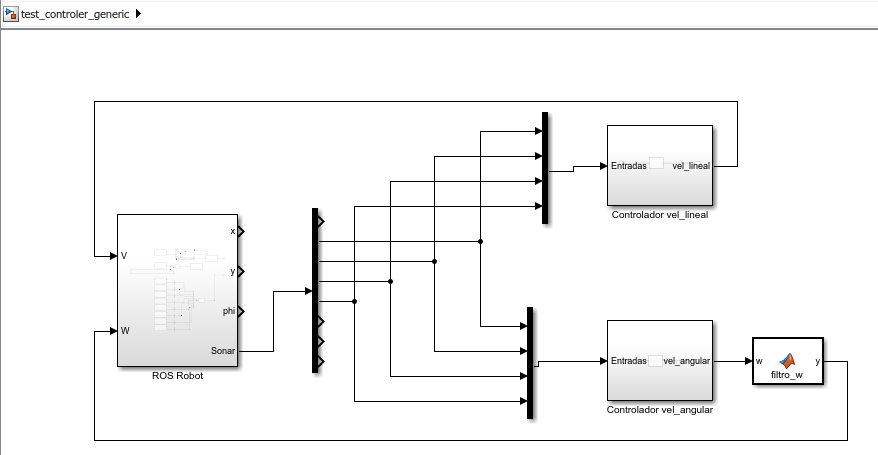
Para ello usamos el código proporcionado en el enunciado de la práctica y lo adaptamos para obtener los datos de entrenamiento para los controladores de la velocidad angular y la lineal por separado: script *prepararDatos.m*.

Además, los datos correspondientes a cada velocidad también son guardadas en archivos .mat.



Posteriormente pasamos a crear el controlador. Para ello usamos la herramienta *anfisedit*. Cargamos los datos y configuramoes el tipo de controlador, el tipo y número de funciones de pertenencia y el número de épocas, para luego realizar el entrenamiento (*Train Now*) y el tras él el test (*Test Now*). Por último, exportamos el controlador en un *.fis*. Este proceso se realiza para cada velocidad.

Una vez obtenidos los controladores, creamos los bloques necesarios en simulink y especificamos en ellos los nombres de cada controlador. Ejecutamos el esquema y verificamos los resultados del movimiento del robot en el simulador de la máquina virtual.



Tras varias pruebas se ha optado por los siguientes controladores:

* Velocidad Angular (*anfisW.fis*): Recibe como entradas las medidas de los 4 sensores delanteros: columnas 2,3,4 y 5 (como está especificado en *prepararDatos.m*), cada una con 3 funciones de pertenencia de tipo gbellmf. La salida tiene una función de pertenencia constante. Para la etapa de entrenamiento se usó un método de optimización híbrido y 150 épocas.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

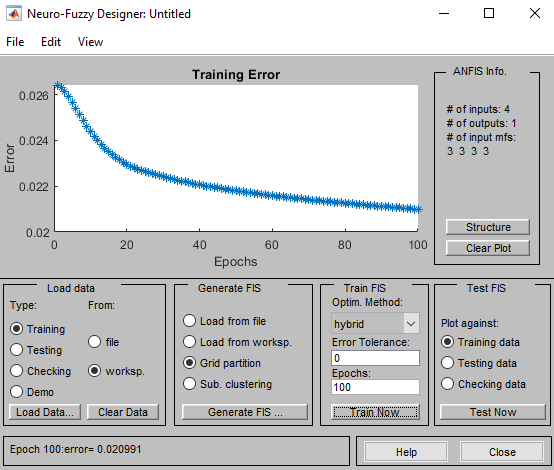
Como se puede observar, el error se va reduciendo hasta obtener uno de 0.042873.

Tras realizar el test se obtiene el siguiente resultado:

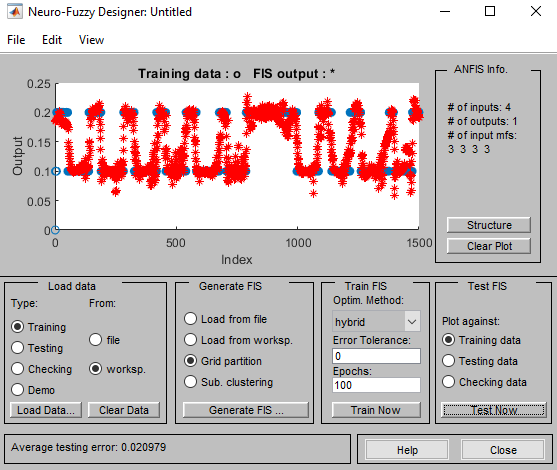
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* Velocidad lineal (*anfisV.fis*): También se tienen 4 entradas (sensores delanteros): filas 2,3,4,5 cada una con 3 funciones de pertenencia de tipo gbellmf. La salida también tiene una función de pertenencia constante. Para el entrenamiento se ha usado le método de optimización híbrido y 100 épocas.



Se consigue un error de 0.020991. El resultado del test es el siguiente:



Con estos controladores se ha conseguido que el robot no choque y pueda dar la vuelta completa, aunque a una velocidad bastante pequeña, ya que en el control manual se ha llegado a una velocidad máxima de 0.2 m/s en zonas sin obstáculos y de 0.1m/s en zonas con obstáculos. Ésto es porque al ir a velocidades más altas, los giros debían ser más bruscos, por lo que los valores capturados de velocidad angular no permitían un buen entrenamiento.