

SISTEMA DE COMANDO Y NAVEGACION PARA ROBOT MOVIL CON ARQUITECTURA DISTRIBUIDA

Fernando G. Pereiro*, Claudio Verrastro**

* Grupo de Inteligencia Artificial - Universidad Tecnológica Nacional F.R.B.A.

** Grupo de Inteligencia Artificial - Universidad Tecnológica Nacional F.R.B.A
Instrumentacion y Control – Comisiion Nacional de Eenergía Atomica.

e-mail: fercasper@yahoo.com.ar, cverra@cae.cnea.gov.ar

Resumen:

En este trabajo se describe sistema para controlar la navegación autónoma para un robot móvil, orientado a la inspección de una planta conocida pero capaz de sortear obstáculos desconocidos. El sistema se comunica con el robot por medio de un radiomodem, enviando comandos a los motores y recibiendo información de los sensores del móvil. Con el operador se comunica por medio de una pantalla en donde se ubica al robot en el plano de la planta y se le dan ordenes complejas en lenguaje coloquial.

Palabras Clave: Robótica Móvil, Navegación Autónoma, Fusión de sensores.

I. INTRODUCCION

El sistema de navegación reside totalmente en la PC, dividido en tres capas: Nivel de comunicaciones, Nivel básico o Kernel y Nivel de Interprete de comandos. Se ha desarrollado un lenguaje de programación de las tareas del robot , que consta de un set de funciones básicas a partir de las cuales se puede construir funciones complejas en lenguaje natural. Se comunica con los controladores del móvil por medio de una interfaz serie y un módem inalámbrico.

II. DESCRIPCIÓN GENERAL

a) Nivel de comunicaciones

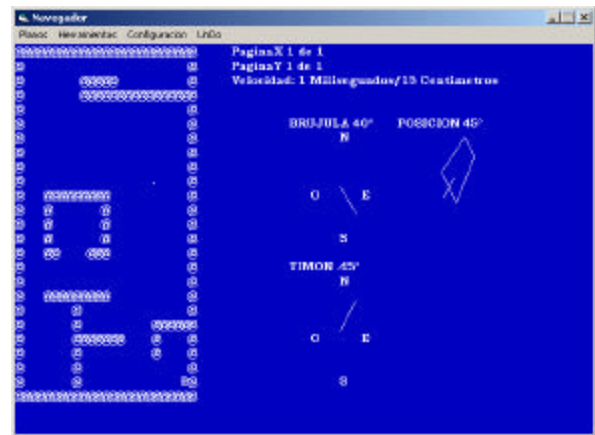
Este es el nivel inferior del sistema y consiste en una serie de funciones las cuales se encargan de enviar y recibir mensajes a las distintas placas controladoras vía puerto serie, resuelve el protocolo de comunicaciones y el direccionamiento de las placas de E/S.

Este nivel solo se comunica con los niveles superiores por medio funciones de entrada y de variables globales .

b) Nivel de funciones

Este nivel consta con varias pantallas para el usuario, y funciones básicas de funcionamiento.

c) Pantallas



Esta es la pantalla principal del Navegador. En la izquierda se puede visualizar el plano conocido del terreno con sus obstáculos. El plano puede ser de dimensiones variables por lo que de no entrar en una sola pantalla se puede ir navegando con los cursores por las distintas paginas del mismo, en la parte superior derecha se puede ver la referencia de la pagina visualizada.

En la parte derecha se pueden ver los datos de posicionamiento :

Brújula: Da una referencia del plano con el Norte, este dato es fijo y es cargado por el operador a la hora de confeccionar el plano.

Posición: Da la posición del móvil con respecto al plano, este dato es actualizado automáticamente por el sistema cuando el móvil cambia de posición.

Timón: Este muestra la posición de las ruedas delanteras con respecto al plano, este dato es actualizado automáticamente por el sistema cada vez que giran las ruedas.

En el margen superior se pueden ver los accesos a las distintas funciones:

Planos:

Configurar:
 X, Y
 Grados

Leer
 Guardar
 Guardar Como
 Nuevo

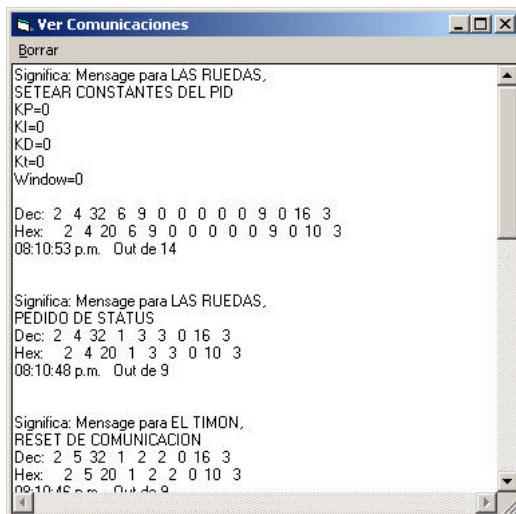
Herramientas:
 Interprete
 Comandos:
 Enviar
 Ver

Leer Gráfico
 Bordes
 Obstáculos
 Limpiar
 Buscar Recorrido:
 En un paso
 Paso a paso
 Pidiendo confirmación
 Gráficamente

Configuración:
 Determinar obstáculo máximo
 Caracteres
 Escala
 Puntos
 Optimizar:
 Por Punto/Por Línea

Este es el modulo central del sistema desde el cual se pueden leer guardar y editar planos así como también ir siguiendo las trayectorias del móvil ya sea que se lo este operando desde el nivel inferior (enviando comandos básicos), desde el intermedio (se le da una coordenada y el navegador traza una trayectoria y la ejecuta) o desde el interprete de comandos.

En la siguiente pantalla se pueden ver todos los comandos que pasan por la red interna de comunicaciones (RS485) ya sean o no para o del Navegador con una descripción de cada uno de ellos y su trama en Decimal y en Hexadecimal.



Esta pantalla del modo teleoperacion, sirve para enviar comandos o pedidos de respuesta a las placas de control de motores, estos comandos son básicos o de nivel de comunicaciones.



Normalmente no se usa esta pantalla mas que para testear el funcionamiento de las placas. O para un modo de operación manual.

Funciones destacadas de este nivel:

Una de las funciones necesarias para la ejecución de cualquier trayectoria es la planificación de una ruta a través del plano conocido, para ello este modulo cuenta con dos sistemas para este propósito.

El primer paso para encontrar una trayectoria es saber si existe un camino posible, para esto se utiliza una función recursiva sencilla, la cual dado un punto de partida, devuelve Verdadero si alguno de los puntos que la rodean (que no son obstáculos) es el punto de llegada, pero de no serlo antes de responder se llama a si misma

preguntando por los puntos que no son el punto de llegada. También lleva un registro de los puntos ya consultados para evitar un "Abrazo Mortal".

De esta manera con una función de pocas líneas y una variable se determina si existe alguna forma de llegar al punto deseado, por mas intrincada que esta sea.

En el caso de ser satisfactorio el punto anterior se pasa a buscar un recorrido posible, sin necesidad de que sea el mas corto ya que luego se optimiza, esta parte de la función esta configurado por el usuario para utilizar una de las dos funciones disponibles:

Método A:

El primero intenta una trayectoria directa desde donde esta el móvil y hacia el punto de llegada y en el caso de encontrar un obstáculo en su camino intenta encontrar un camino bordeándolo , este algoritmo es una función recursiva típica utilizada para resolver laberintos que va guardando un registro de los últimos movimientos realizados, una vez que encuentra una salida vuelve a intentar una trayectoria directa hacia su destino y así sucesivamente hasta llegar a su punto de llegada.

Esta función es muy complicada, utiliza menos recursos del sistema pero es sumamente lenta y produce caminos no óptimos.

Método B:

La segunda función no es mas que una versión mas completa de la función de búsqueda de un camino posible la cual además de irse llamando a si misma hasta encontrar el punto de llegada, una vez que lo encuentra registra todos los puntos por los que paso para llegar a dar ese camino.

Con esto se obtiene un camino completamente ridículo en la mayoría de los casos pero luego de un proceso de mejora resultan ser buenos y en muchos casos mejor que el encontrado por la otra función.

En ambos casos se obtiene un recorrido el cual dista de ser optimo ya que suelen tener muchas idas y vueltas por lo que se lo optimiza de la siguiente forma:

Se van tomando uno a uno los puntos del recorrido desde el primero hacia el ultimo y en cada uno de ellos se traza una recta hasta cada uno de los puntos restante del recorrido teniendo en cuenta que no haya obstáculos en medio, en cada caso en el cual la respuesta es satisfactoria se

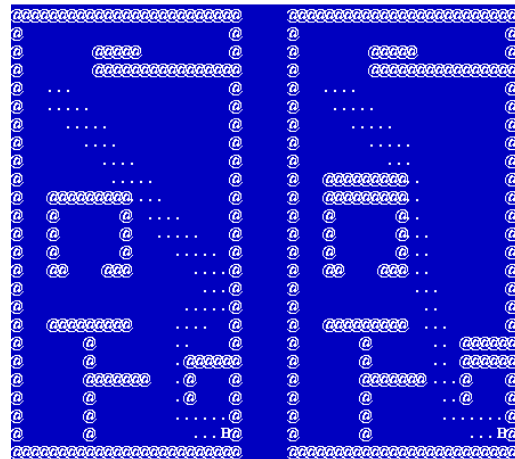
eliminan todos los puntos intermedios de la trayectoria.

De esta manera se obtienen unos pocos puntos que al unirlos con rectas darán la trayectoria optima para llegar al destino.

Normalmente estas trayectorias no solo tienden a ser rectas sino que también tienden a formar curvas que rodean los obstáculos eficazmente.

Metodo A

Metodo B



Quando los sistemas de vision o de ultrasonido detectan un obstaculo, este es marcado en el plano como objeto nuevo y si interfiere en la trayectoria del movil, la ruta es recalculada tomando como punto de partida la coordenada actual.

III. INTERPRETE DE COMANDOS

El Interprete de comandos es el nivel superior del sistema, para su funcionamiento utiliza como plataforma los niveles inferiores a los cuales les deriva las funciones a realizar.

La pantalla no es mas que un cuadro de texto el cual sirve como interfaz del motor del interprete que hace el trabajo. Cuya flexibilidad permite que casi cualquier sentencia sea válida ya que el sistema va aprendiendo de las explicaciones que les suministra el usuario.

La idea es muy simple, se parte de una serie de comandos básicos, 29 en total, divididos en tres grupos:

Primer Grupo

- #TIME
- #IDBARRAS

#BRUJULA
 #SONAR
 #GRADOS
 #DISTANCIA
 #VELOCIDAD
 #X
 #Y
Segundo grupo
 #RESPONDE
 #RESETS
 #CUANDO
Tercer grupo
 #IF
 #ELSE
 #=
 #<>
 #>
 #<
 #*
 #/
 #+
 #-
 #AND
 #OR
 #XOR
 #NOT
 #(
 #)

Todas las palabras que comiencen con ## se las considera como una variable

Un comando extra se utiliza para cambiar de base de datos de palabras asociadas, se puede usar para personalizaciones de distintos usuarios.

El primer paso es dar una oración cualquiera, el sistema busca todas las palabras que son comandos básicos (primer grupo), Luego busca en el diccionario del usuario las palabras que el usuario ya ingreso significado, se reemplazan estas por su significado, por último pregunta por el significado de las palabras nuevas, que es guardado en la base de datos (se puede responder que no existe significado). Recursivamente, se realizan todas las la sustituciones hasta que se obtiene una oración (usualmente mucho mas extensa que la original) con palabras básicas, números o variables. Recordemos que las palabras a las que no se le asigno significado se guardan en el diccionario y sirven solo de conectores.

Las palabras básicas del primer grupo son variables que representan o dan valores a periféricos del sistema por lo que el siguiente paso

es reemplazar todas aquellas que son datos de entrada que se obtienen de los niveles inferiores del sistema (sensores, controladores, etc.), dejando aquellas que se usan para enviar datos, que pueden ser suministrados por el usuario o calculados por el navegador.

Ejemplo:

#TIME es solo de lectura por lo que siempre se lo reemplaza por la hora del sistema, en cambio #X significa la coordenada X del móvil en el plano. Si esta precedido por una variable y un signo = se le asigna a la variable el valor de X

El siguiente paso es resolver las operaciones matemáticas, Disyunciones y pruebas lógicas (Tercer grupo) lo cual se resuelve con una serie de maquinas de estado típicas de un compilador.

Finalmente quedan Variables de escritura (#X, #Y, #TIMON, #VELOCIDAD, #DISTANCIA) , Números o palabras de funciones (#CUANDO, #RESPONDE, #RESETS), que evaluadas de izquierda a derecha, , de encontrarse funciones de comandos se ejecutan, a las variables de escritura seguidas de números se les asigna ese valor. (con las inconsistencias se emite un mensaje de advertencia)

Las palabras del Segundo grupo son comandos que realizan funciones particulares como:

#RESPONDE:

Envía un mensaje al operador con el resultado de lo que esté a la derecha de la palabra.

Ejemplo:

#RESPONDE #TIME

Envía un mensaje al operador que dice la hora.

#CUANDO:

Debe estar seguido de un conjunto de números y variables de escritura o par de variables de lectura/escritura o de una serie de pares unidos por operadores lógicos que ya fueron resueltos

Todo lo que sigue se considera como tarea a realizar por lo que se copia en una base de datos y un subsistema se encarga de reenviar el comando al interprete de comandos cuando la condición se cumpla.

Ejemplo:

#CUANDO #X #= 5 #X = 10

Cuando el móvil pase por el punto X= 5, , se emitirá un comando para desplazarse a la coordenada X = 10.

Ejemplo2:

#CUANDO #X #= 14 #RESPONDE #TIME

Cuando pase por la coordenada X = 14 enviara un mensaje al operador con la hora de ese momento.

#RESETS

Este comando no tiene parámetros y sirve para resetear el sistema.

Ejemplo del Interprete: *(Comandos en Italica)*

Supongamos que el sistema se esta utilizando por primera vez por lo que no conoce ningún significado y se tipea el siguiente comando:

Cuando sea de día anda a la esquina .

El interprete preguntara los significados de las palabras que no conoce

Finalmente termina ejecutando el siguiente comando:

#CUANDO #TIME #> 7 #AND #TIME #< 19,5 #X #= 32 #Y #= 54 .

Si se emite este comando de día, el móvil se moverá a la dirección dada en el plano, internamente envió al nivel inferior la orden de moverse a la coordenada dada, pero antes busca una ruta y la optimiza en el nivel intermedio para generar una serie de comandos al nivel de comunicaciones quien por ultimo emite los comandos en la secuencia adecuada a los distintos controladores.

Si le escribiese el mismo comando nuevamente solamente respondería:

ENTENDIDO:#cuando #TIME #> 7 #AND #TIME #< 19,5 #X #= 32 #Y #= 54 .

Pero no haría nada ya que si bien ejecutaría el comando #CUANDO este no movería el móvil ya que el generador de trayectoria descubriría que el punto de partida es igual al de llegada.

Este es un ejemplo muy simple pero se pueden realizar verdaderos programas tan extensos como se quiera los cuales se ejecuten con solo escribir una palabra como por ejemplo el comando:

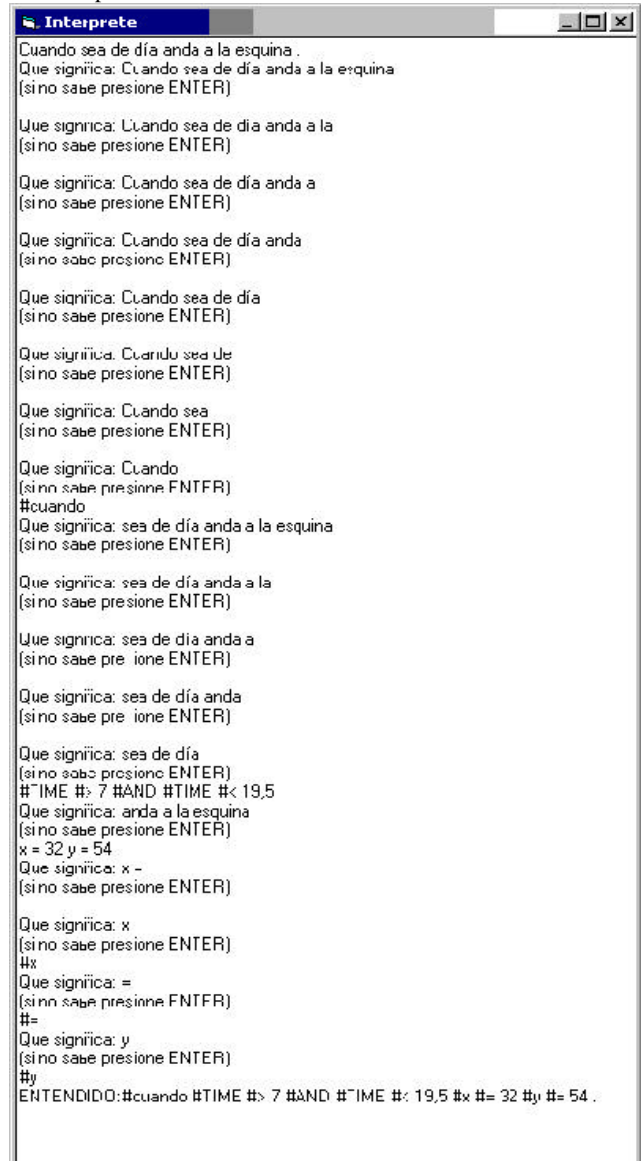
Che Veni .

Que, según esté definido, retornará a la posición de origen

IV. CONCLUSIONES

Se ha desarrollado un entorno de trabajo para la programación de tareas de un robot móvil, que permite el control del mismo en lenguaje natural. Actualmente se esta trabajando en la integración de dos módulos, ya desarrollados, un sonar y una

cámara de visión que funden la información con la odometría por medio de un control difuso.



REFERENCIAS

- Villegas J. Et al. Navegación autónoma, localización y desplazamiento de piezas en entornos estáticos conocidos
- Siemiłtkowska B., Weigl M.: "New World Modelling Methods for Autonomous Mobile Robots With Various Type Sensors", Proc. of International Workshop on Intelligent Robotic
- Weigl M., Siemiłtkowska B., Sikorski K., Borkowski A.: "Grid-based mapping for autonomous mobile robot", Robotic and Autonomous Systems, Amsterdam, Holland, 1993