

DESARROLLO DE UN ROBOT CON ARQUITECTURA DE SUBSUNCION (ROBOT NIMROD)

Realizado por : Ing. Msc. Ricardo J. Alonso C. Profesor
Universidad Tecnológica del Centro
(ralonso@unitec.edu.ve , ricardo_alonso@yahoo.com)

Resumen

El presente trabajo tiene la finalidad de describir un robot móvil con conducción diferencia tipo oruga, el cual tiene una programación basada en una arquitectura de subsunción la cual sustenta 3 comportamientos, prioritzados en el siguiente orden: Escape, Evitar obstáculos, Navegar siguiendo un rumbo señalado y controlado con una brújula analógica.

Dicha arquitectura se encuentra sustentada físicamente por un microcontrolador ATOM28 embebido en una tarjeta de desarrollo IBOARD desarrollada por la compañía ROSO Control de Valencia, Venezuela.

Como sensores, el robot cuenta con dos sensores ultrasónicos tipo Ping de Parallax, ubicados a ambos lados frontales del robot, con una divergencia de aproximadamente 10 grados. Dichos sensores alimentan los comportamientos de Escape y de Evitar obstáculos.

Una brújula marca Dinsmore analógica, proporciona al robot información referente a la orientación magnética del vehículo. Este sensor alimenta al comportamiento de navegar, pero su lectura es complementaria a los comportamientos de Escape y Evitar obstáculos.

Los actuadores del robot son dos servos Parallax modificados para ser de velocidad variable usando la función SERVO del léxico del ATOM28. Esto permite la orientación fina del robot tanto como para esquivar obstáculos como para conseguir el rumbo deseado para el vehículo.

Como aspecto resaltante, el comportamiento de esquivar obstáculos usa como filosofía la medición comparativa del espacio libre disponible a ambos lados del robot. El vehículo girará hacia el lado con más espacio libre. Dicho giro seguirá hasta que ambos lados tengan el mismo espacio libre. En caso de que este espacio sea menor a 10 centímetros, se dispara el comportamiento de Escape. El comportamiento de Esquivar se acciona solo cuando el espacio en cualquier lado es menor a 40 centímetros.

El robot ha demostrado un comportamiento curioso ante la interacción de todos los comportamientos. Su tendencia es a explorar autónomamente una amplia área del espacio de pruebas del mismo, pero siempre manteniendo como objetivo el avanzar en el rumbo magnético definido por el usuario del mismo.

Palabras claves : brújula magnética, subsunción, sonar, Nimrod

Abstract

The main purpose of this work is to present a report of my experiences developing a mobile robot, based on subsumption architecture that I called Nimrod.

The robot was based on the ATOM 28 hardware, developed by MicroBasic. The PIC is embedded on a card called iboard, developed by ROSO Control in Valencia, Venezuela.

As sensor I used a Dinsmore analog magnetic compass and 2 Sonar Ping of Parallax. The actuators of robot are 2 servo motors modified for continuous rotation, such a way I can have a speed control using SERVO instruction of ATOM28.

The subsumption architecture has 3 behaviours, with the following priority order: Escape, Avoid obstacles and Navigation using compass.

The sonar is the sensor that uses the escape and avoiding behaviour for working. But the magnetic compass is used for the navigation behaviour and the escape and avoiding ones.

The avoiding behaviour uses the differential distance measurement of both sonar, installed both sides front of vehicles with a divergence of 10 degrees. The differential measurement gave a signal of what side of path is clearer of obstacles. As a consequence the robot turns to the clearer side.

The robot shows a complex behaviour because it can explore autonomously a big area of the test alley, but keeping in mind the basically navigational aim of the magnetic compass.

Keywords : magnetic compass, subsumption, Nimrod

1. Introducción

El siguiente artículo constituye el Segundo de una serie de artículos técnicos que escribiré acerca del desarrollo de vehículos robóticos totalmente desarrollados con el máximo de elementos de origen casero, es decir, mecanismos desarrollados totalmente por el autor sin la compra de esos robots que se anuncian en la Internet a un costo relativamente elevado.

La excepción se hace con los microprocesadores y sensores, los cuales prefiero comprarlos hechos, aunque selecciono los que tengan la mayor relación beneficio / costo del mercado.

En un primer artículo hablé de un robot con un software basado en Redes Neuronales del tipo autoaprendizaje (mecanismo Hebbiano no supervisado), el cual adaptaba la red neuronal para el correcto funcionamiento de un robot que se mueve en un medio lleno de obstáculos y cuyo comportamiento más simple era el de Escapar y el de Evitar obstáculos.

El presente robot hace uso de otra filosofía, la cual es la arquitectura de subsunción. Dicha arquitectura se basa en la presencia de varios tipos de comportamientos simples y elementales, cuya interacción da lugar a

comportamientos emergentes mucho mayor grado de complejidad.

La subsunción es una filosofía de programación de vehículos robóticos mucho más sencilla que aquellas derivadas de la Inteligencia Artificial, por cuanto los comportamientos básicos pueden ser adicionados al vehículo de manera modular y progresiva, a la vez que hace al robot poco vulnerable a la falla de uno o más de los sensores del mismo, en un principio denominado degradación elegante sensorial.

2. La Arquitectura de subsunción y Programación Basada en Comportamientos

Los comportamientos son rutinas sencillas de implementar, las cuales son autónomas en el sentido de que basta y sobra que la subrutina y los periféricos del robot sean implementados para que el mismo actúe como un autómatas; limitado pero suficiente.

Es decir, que no se requiere de otra subrutina para ella opere; es una entidad por sí sola.

Cada comportamiento es implementable, comenzando por los más elementales para la supervivencia del autómatas hasta los más avanzados para el cumplimiento de niveles de más alto nivel en el robot.

Los comportamientos tienen muchas veces objetivos que son contradictorios. El comportamiento de escape solo garantiza que el robot escape de una situación de atropamiento del robot, mientras que el de navegación hace que el robot siga un patrón determinado de viaje, el cual podría colocarlo en una situación de atrapamiento.

Lo que dispara un determinado comportamiento es el estado de los sensores del robot y/o el disparo o no de otros comportamientos en el robot. La estructura de subsunción tiene como objetivo el resolver los conflictos de interés generados por los comportamientos al activarse simultáneamente ante determinados estados sensoriales.

Las estructuras de subsunción pueden ser:

- Redes neuronales
- Sistemas basados en lógica difusa
- Sistemas basados en reglas.

Esta última estructura es la más sencilla y rápida de implementar, aunque las dos primeras son de sumo interés para lograr comportamientos más avanzados, cercanos al comportamiento inteligente del vehículo.

3. Comportamientos incorporados en Nimrod

El robot Nimrod ha incorporado los siguientes comportamientos a una estructura de subsunción basada en reglas. Los comportamientos son:

- Comportamiento de escape: el comportamiento hace que el robot retroceda por un tiempo de 2 segundos y luego gire 1.5

segundos en sentido dextrógiro. El comportamiento se acciona cuando la distancia medida por uno o más de los dos sensores de sonar caiga a 10 o menos centímetros.

- Comportamiento de evitar obstáculos: dicho comportamiento hace que el robot gire en el sentido del sonar que tiene el mayor espacio libre. La tasa de giro es proporcional a la diferencia entre las mediciones de distancia de ambos sensores, por lo que el robot dejara de girar tan pronto ambas distancias sean casi iguales (una diferencia absoluta de 10 centímetros). Para que el comportamiento se active, la distancia en uno o más de los sensores debe ser menor a 40 centímetros y el comportamiento de escape no debe estar activado.
- Comportamiento de navegar: el robot avanza en el plano de navegación usando la brújula del robot y un punto de ajuste del rumbo de navegación del vehículo. Solo se activa cuando ninguno de los otros dos comportamientos está activado.

Es de hacer notar que cuando el robot sale de una maniobra de evitar obstáculo, el mismo continúa con el comportamiento de navegar. Dicha acción hace que el robot retome su rumbo enfrentándolo así de nuevo con el obstáculo que en primer lugar dio origen a la maniobra de evitar obstáculo. De esta manera es posible que el vehículo oscile de manera alternativa entre ambos comportamientos a perpetuidad.

Para evitar esto, el comportamiento de evitar obstáculo reprograma el rumbo deseado de navegación al último rumbo que aleja al vehículo de un rumbo de colisión con el obstáculo. De esta forma el comportamiento de navegación adopta durante 10 ciclos de programación ese rumbo, alejando al robot del obstáculo. Después de eso, se reprograma el rumbo original deseado.

4. Modulo de accionamiento de los motores.

Los comportamientos producen una serie de parámetros, los cuales accionan los motores servos del robot. El comportamiento de escape define un valor para los parámetros de la instrucción SERVO que comandan los dispositivos izquierdo y derecho. En cambio, los comportamientos de evitar obstáculos y escape proporcionan un parámetro llamado M, el cual, es la diferencia que hay entre las lecturas de ambos sensores de sonar (en el comportamiento de evitar obstáculos) y la diferencia entre el rumbo deseado del robot y el la lectura de la brújula magnética del mismo (en el comportamiento de navegar).

Dicho parámetro define la velocidad y sentido de giro de cada servo, estandarizando dicho valor al rango - 1200 / + 1200.

5. Programa del Robot Nimrod

A continuación se presenta el código del programa en Basic escrito para el ATOM28.

```
'== SECCION DE DECLARACION DE LAS VARIABLES Y CONSTANTES ==
lecturaA      var    word
lecturaB      var    word
VA            var    long
VB            var    long
heading       var    long
SP_RUMBO      var    long
flag          var    byte
gradiente     var    long
FLAG_ESCAPE  VAR    BYTE
FLAG_EVITAR   VAR    BYTE
FLAG_NAVEGAR  VAR    BYTE
T             VAR    WORD
M             VAR    LONG
VAVG          VAR    LONG 'AVERAGE DE VELOCIDADES DE AMBOS SONARES
AAVG          VAR    LONG 'AVERAGE DE ACELERACIONES DE AMBOS SONARES
Medicion1     VAR    WORD 'ALMACENA LA MEDICION DEL SENSOR IZQUIERDO
Medicion2     VAR    WORD 'ALMACENA LA MEDICION DEL SENSOR DERECHO EN
Distance1     VAR    LONG 'DISTANCIA DEL OBJETO AL SENSOR PING Iz. cm
Distance2     VAR    LONG 'DISTANCIA DEL OBJETO AL SENSOR PING De. cm
D1OLD        VAR    LONG
D2OLD        VAR    LONG
V1NEW        VAR    LONG
V2NEW        VAR    LONG
V1OLD        VAR    LONG
V2OLD        VAR    LONG
A1           VAR    LONG
A2           VAR    LONG
RUMBO        VAR    LONG
CONTADOR     VAR    BYTE
SIG1         CON    P0 'TERMINAL DE E/S DEL SENSOR PING IZQUIERDO
SIG2         CON    P1 'TERMINAL DE E/S DEL SENSOR PING DERECHO
mot_der      CON    P8
mot_izq      CON    P7
Conversion   FCON  29.034 'CONSTANTE PARA LA MEDICION

rot_izq      var    sword
rot_der      var    sword
temporal     VAR    SWORD

CUENTA       VAR    BYTE

GOSUB SENSORES

SP_RUMBO = Heading

D1OLD = 300.00
D2OLD = 300.00
V1OLD = 0.00
V2OLD = 0.00
RUMBO = SP_RUMBO
CONTADOR = 0
```

'=====

CICLO:

```
GOSUB SENSORES

IF ( Distance1 < 10.0 ) OR ( Distance2 < 10.0 ) THEN
    FLAG_ESCAPE = 1
ELSE
    FLAG_ESCAPE = 0
ENDIF

IF (((Distance1 >= 10.0) AND (Distance1 <= 40.0)) OR ((Distance2 >= 10.0) AND
    (Distance2 <= 40.0))) AND (FLAG_ESCAPE = 0) THEN
    FLAG_EVITAR = 1
ELSE
    FLAG_EVITAR = 0
ENDIF

IF (FLAG_ESCAPE = 0) AND (FLAG_EVITAR = 0) THEN
    FLAG_NAVEGAR = 1
ELSE
    FLAG_NAVEGAR = 0
ENDIF

IF FLAG_ESCAPE = 1 THEN GOSUB ESCAPE
IF FLAG_EVITAR = 1 THEN GOSUB EVITAR
IF FLAG_NAVEGAR = 1 THEN GOSUB NAVEGAR

IF CONTADOR >= 15 THEN
    RUMBO = SP_RUMBO
    CONTADOR = 0
ENDIF

CONTADOR = CONTADOR + 1
```

GOTO CICLO

SENSORES:

```
'     ESTA PARTE DE LA SUBROUTINA SE DEDICA A LA LECTURA DE LOS DOS SONARES DEL ROBOT
'
'     PARA EL SENSOR IZQUIERDO SE COMIENZA LA MEDICION

LOW     SIG1                'GARANTIZAR ESTADO INICIAL DEL PIN
        PULSOUT SIG1,5      'EMITIR PULSO DE ACTIVACION
PULSIN  SIG1,1, Medicion1   'LEER SALIDA DEL SENSOR (ECO)

'EL TIEMPO ALMACENADO EN Medicion1 FUE EL REQUERIDO POR LA ONDA ULTRA-
'SONICA PARA LLEGAR HASTA EL OBJETO Y REGRESARSE
'SE HACE LA MEDICION PARA EL SENSOR DERECHO

LOW     SIG2                'GARANTIZAR ESTADO INICIAL DEL PIN
        PULSOUT SIG2,5      'EMITIR PULSO DE ACTIVACION
PULSIN  SIG2,1, Medicion2   'LEER SALIDA DEL SENSOR (ECO)

'EL TIEMPO ALMACENADO EN Medicion2 FUE EL REQUERIDO POR LA ONDA ULTRA-
'SONICA PARA LLEGAR HASTA EL OBJETO Y REGRESARSE

Medicion1 = Medicion1 /2    'TIEMPO DE VIAJE DE LA SEÑAL
Medicion2 = Medicion2 /2
Distance1 = FLOAT(Medicion1) FDIV Conversion
Distance2 = FLOAT(Medicion2) FDIV Conversion

'DISTANCIA GUARDA LA DISTANCIA ENTRE EL PING Y EL OBJETO EN CM

V1NEW = (D1OLD FSUB Distance1) FDIV 0.3
V2NEW = (D2OLD FSUB Distance2) FDIV 0.3

A1 = (V1NEW FSUB V1OLD) FDIV 0.3
A2 = (V2NEW FSUB V2OLD) FDIV 0.3
```

```

D1OLD = Distance1
D2OLD = Distance2
V1OLD = V1NEW
V2OLD = V2NEW

VAVG = 0.5 FMUL (V1NEW FADD V2NEW)
AAVG = 0.5 FMUL (A1 FADD A2)

'ESTA PARTE DE LA SUBROUTINA SE DEDICA A LA LECTURA DEL COMPAS MAGNETICO 1655 DINSMORE

    pause 1500          ' ESPERE 1.5 SEGUNDOS PARA QUE SE ESTABILIZE LA BRUJULA

    Adin AX0,2,AD_RON,lecturaA
    Adin AX1,2,AD_RON,lecturaB

    VA = ((5.00 fdiv 1024.0) fmul (float lecturaA))
    VB = ((5.00 fdiv 1024.0) fmul (float lecturaB))
    gradiente = (VA fsub VB) fadd 10.00

    IF (VA >= 2.08) AND (VA <= 2.92) AND (gradiente <= 10.00) then
        heading = 45.0 fadd 107.14 fmul ( 2.92 fsub VA )
    endif

    IF (VA >= 2.08) AND (VA <= 2.92) AND (gradiente >= 10.0) then
        heading = 225.0 fadd 107.14 fmul ( VA fsub 2.08)
    endif

    IF (VB >= 2.08) AND (VB <= 2.92) AND (gradiente <= 10.0) then
        heading = 135.0 fadd 107.14 fmul ( 2.92 fsub VB )
    endif

    IF (VB >= 2.5) AND (VB <= 2.92) AND (gradiente >= 10.0) then
        heading = 107.14 fmul ( VB fsub 2.50)
    endif

    IF (VB >= 2.08) AND (VB < 2.50) AND (gradiente >= 10.0) then
        heading = 315.00 fadd 107.14 fmul ( VB fsub 2.08)
    endif

    '      serout s_out,i9600, [ REAL heading\2, " grados ",REAL Distance1, " ", REAL
Distance2,13]

RETURN

'=====

EVITAR:

    M = ((Distance2 FSUB Distance1) fadd 360.00)

    '      SEROUT S_OUT, I9600 , [real Distance1\2," ",real Distance2\2, " ", real M," ",
DEC T, 13]

    GOSUB AJUSTE

    GOSUB SENSORES
    RUMBO = heading
    CONTADOR = 0

    IF (M < 364.00) AND (M > 356.00) THEN GOSUB AVANCE

RETURN

'=====

NAVEGAR:

    M = ((RUMBO FSUB heading) fadd 360.00)

    GOSUB AJUSTE

    IF (M < 370.00) AND (M > 350.00) THEN GOSUB AVANCE

RETURN

```

```

=====
AJUSTE:

    temporal = int(M)

    rot_izq = 3*(temporal - 360)
    rot_der = 3*(temporal - 360)

'    serout s_out , i9600, [sdec rot_der, " ", sdec rot_izq,13]

    servo mot_der,rot_der,1
    servo mot_izq,rot_izq,1

RETURN

=====

AVANCE:

    FOR CUENTA = 1 TO 5

        rot_der = 400
        rot_izq = -400

        servo mot_der,rot_der,1
        servo mot_izq,rot_izq,1

    NEXT

RETURN

=====

ESCAPE:

FOR CUENTA = 1 TO 50

    servo mot_izq, 200,1
    servo mot_der,-200,1

NEXT

FOR CUENTA = 1 TO 20

    servo mot_izq, 200,1
    servo mot_der, 200,1

NEXT

GOSUB  SENSORES
RUMBO  = heading
CONTADOR = 0

RETURN

```

6. Descripción del Hardware

Como ya ha sido dicho, el robot Nimrod tiene las siguientes características de hardware:

- Cerebro : Procesador ATOM28 de BasicMicro, embebido en una tarjeta de desarrollo iBOARD.
- Actuadores : dos motores servo marca Parallax.
- Plataforma de robot tipo oruga
- Sensores
 - Sonares modelo PING de Parallax. Son dos, instalados con un ángulo de divergencia lateral de 10°.
 - brújula Dinsmore analógica modelo R.1655.

Las fotografías 6.1, 6.2, 6.3 y 6.4 muestran varias vistas del robot vehicular (frente, planta y laterales derecho e izquierdo).



Figura 6.1



Figura 6.2



Figura 6.3

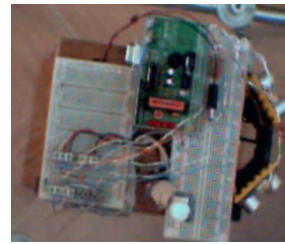


Figura 6.4

El diagrama 6.1 muestra un diagrama de conexionado de los diversos elementos de Nimrod.

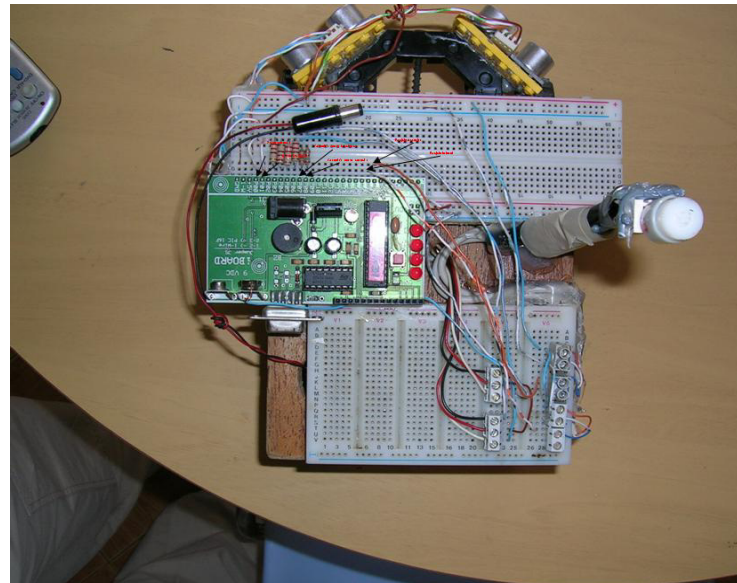


Diagrama 6.1 (ampliar foto para ver conexiones)

7. Ensayos efectuados

El Diagrama de la figura 7.1 muestra el recorrido de prueba del robot. El rumbo designado para el robot es establecido desde el mismo momento en que se energiza o resetea el vehículo. Lo que pasa es que la primera instrucción a ejecutarse en el arranque es la consulta del rumbo magnético dado por la brújula Dinsmore. Esta lectura será su rumbo designado durante toda la prueba.

La siguiente película muestra el primer recorrido de Nimrod colocándolo en un rumbo de 300°.

Película recorrido Nimrod

8. Resultados obtenidos

Tal y como se ve en la película de Nimrod, el robot sigue los siguientes pasos:

- Robot avanza siguiendo el rumbo que tenía al momento de ser inicializado su microcontrolador.
- Robot corrige desviaciones del rumbo asignado. Dichas desviaciones son causadas por obstáculos en el suelo no detectables por el sonar o por las diferencias de conducción de los servomotores.
- El robot detecta un obstáculo. Su sensor derecho indica una lectura de distancia inferior a la de su sensor izquierdo. Sin embargo, mientras esta lectura no sea inferior a 40 centímetros, el comportamiento de navegación predominará.
- El robot está a menos de 40 centímetros del obstáculo. En este momento gobierna el comportamiento de evitar obstáculos. El sensor gira en el sentido levógiro al ser la lectura izquierda mayor que la derecha.
- El robot gira hasta que ambas lecturas son iguales y mayores a 40 centímetros. En este caso el comportamiento de evitar deja de gobernar y pasa el gobierno al comportamiento de navegar.
- En este caso, el robot retoma el rumbo original, el cual lo pone de nuevo enfrente al obstáculo. Esto hace que el control se pase de manera alternativa entre ambos

comportamiento, produciendo que el robot quede inmovilizado.

- Para evitar esto, altera momentáneamente el rumbo deseado a fin de que se mantenga en todo momento un rumbo paralelo al obstáculo por unos 10 ciclos de programa.
- Luego de esto, el comportamiento de navegar recupera su rumbo original. Si aun hay obstáculos en ese rumbo, se repite el ciclo de evitar – navegar arriba descrito.
- Si no detecta obstáculo, el robot sigue navegando el rumbo deseado.
- De esta forma, la programación del robot permite mantener el rumbo deseado a la vez que evita obstáculos, sin que hayan conflictos entre ambos comportamientos.

9. Conclusiones y recomendaciones

Como se ve en esta monografía, la programación de comportamientos usando la teoría de la subsunción permite:

1. Coordinar la actividad de varios comportamientos a fin de evitar conflictos.
2. Los comportamientos pueden ser modificados de forma independiente.
3. Se pueden incorporar nuevos comportamientos y coordinarlos con la arquitectura con un mínimo de modificaciones al programa.
4. El comportamiento global del robot es complejo y adaptativo a los diversos ambientes del robot.
5. La incorporación de nuevo hardware adicional es fácil dado el sistema de breadboard utilizado.

Hemos visto que se hace necesario incorporar nuevos sensores que permitan: a) enriquecer el universo que percibe el robot, b) respaldar los sensores existentes a fin de utilizar los beneficios del modelo de degradación graciosa de sensores a través de la mezcla sensorial.

Para ello planeamos instalar dos nuevos sensores en el robot u otro similar:

1. Un detector GPS que permita al robot apuntar no sólo a un rumbo, si no que permita navegar al robot de un punto

- dado a otro definido por su latitud y longitud.
2. El sensor GPS tiene incorporado un reloj GMT y un detector de velocidad y rumbo geográfico, además de altura. Estas lecturas permiten reemplazar el dinmore usado en Nimrod, además de inferior data para un control PID de velocidad, sensores de gradiente y sensores de éxtasis.
 3. Estos dos últimos sensores permitirán:
 - a. Detectar pendientes que pudieran volcar al robot, además de seguir una línea de contorno en un terreno.
 - b. Hacer que un robot acciones el comportamiento de escape en caso de que no cambie su posición en un tiempo dado o su velocidad sea cero.

El objetivo es desarrollar hardware que nos permita experimentar varios tipos de esquemas con un mínimo de inversión y pérdida de tiempo.

Se planea que el nuevo robot sea llamado Galileo y será realizado en el transcurso del año 2007.

10. Referencias Bibliografiitas

- Kachroo,P & Mellodge,P. "Mobile robotic Car Design", TAB Robotic-McGrawHill, 2002.
- Jones, J. "Robot Programing; A practical guide to Behavior Based robotics", TAB Robotic-McGrawHill,2002.
- Braitenberg, V. "Vehicles; experiments in syntethic psychology", Bradford Books,2002.
- Murphy, R. "Introduction to AI robotics". Bradfor Book, 2002.