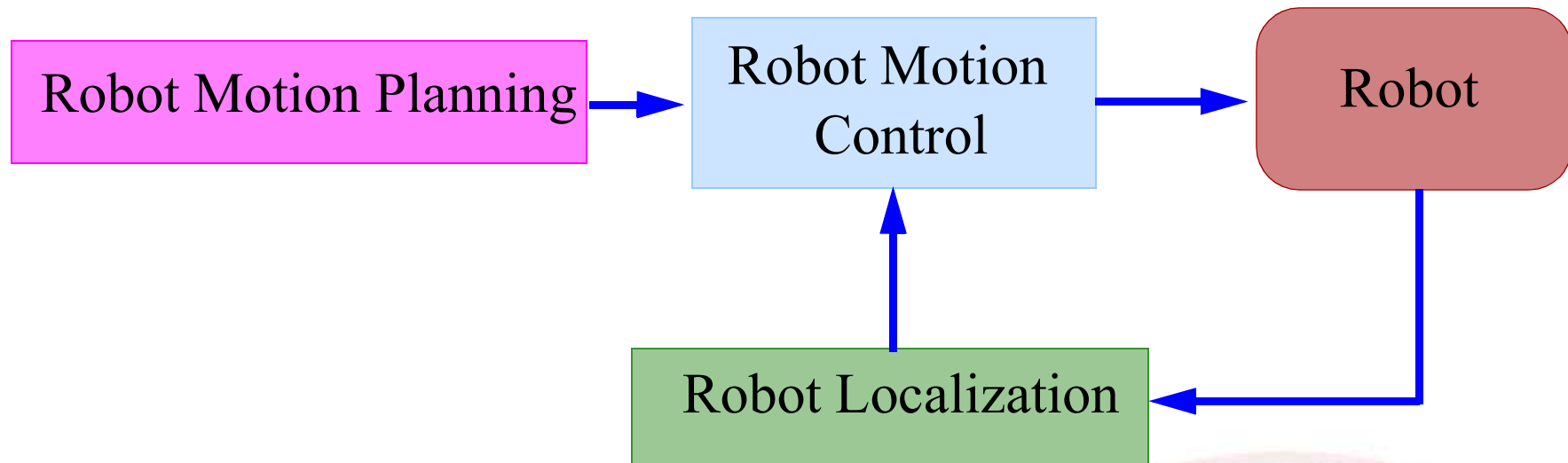




3 ROBOTS MÓVILES

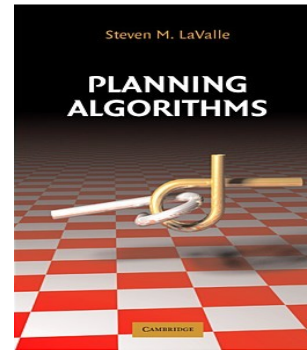
- 3.1 Introducción: Preliminares y Conceptos.
 - 3.2 Características de los Robots Móviles.
 - 3.3 Estrategias de Control.
 - 3.4 Seguimiento de Trayectorias.
 - 3.5 Algoritmos de Planificación.
 - 3.6 Introducción a la Localización.
 - 3.7 Control reactivo
 - 3.8 Slam
 - 3.9 Navegación Topológica
-

Las tres Grandes Áreas de la Navegación Planificada



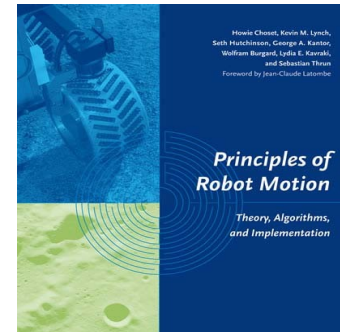
Las tres Grandes Áreas de la Navegación Planificada

Robot Motion Planning



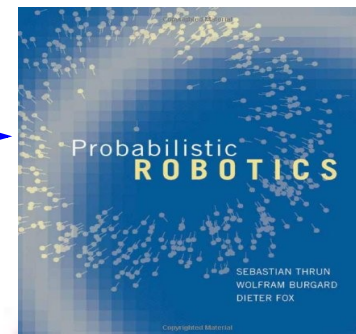
Planning Algorithms (2006)
by Steven M. LaValle

Robot Motion
Control



Principles of Robot Motion (2005)
by Howie Choset, S Thrum et al

Robot Localization



Probabilistic Robotics (2006)
by S Thrum et al



3.4 Seguimiento de Trayectorias y Camino: Algoritmos de path tracking

- Dada una trayectoria o camino, se pretende que el robot la siga de la forma más aproximada posible:
 - ☐ Seguimiento de Trayectoria.
 - ☐ Seguimiento de Caminos.
- Existen numerosos métodos de seguimiento, basados en: Teoría de control no lineal, control predictivo, linealización del modelo cinemático, métodos geométricos.





- **Seguimiento de Trayectorias:**

- ☐ Se especifica la evolución temporal de la posición $[x(t), y(t)]$ y de la velocidad $[\dot{x}(t), \dot{y}(t)]$.

- ☐ Para cada instante t_0 se calcula el error en posición y en velocidad teniendo en cuenta la posición y velocidad del robot y la posición y velocidad planificadas para ese instante: $[x(t_0), y(t_0)]$ $[\dot{x}(t_0), \dot{y}(t_0)]$.

- **Seguimiento de Caminos:**

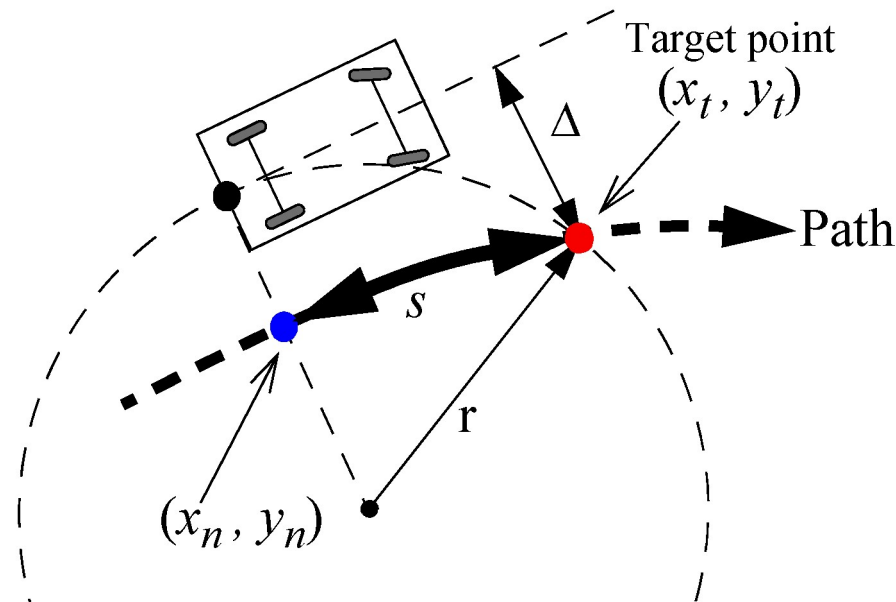
- ☐ Se especifica la serie de configuraciones que compone el camino $[x, y]$

- ☐ Para cada instante t_0 se calcula el error en posición teniendo en cuenta la posición del robot y un punto seleccionado del camino.



Seguimiento de Caminos

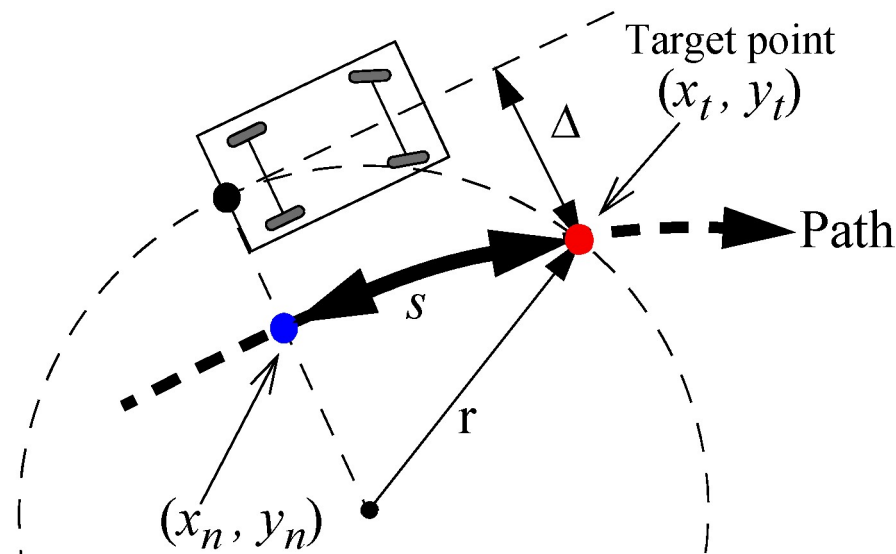
- Dentro de los métodos geométricos, el más conocido y sencillo es el de **Persecución Pura (Pure Pursuit)**.
- Permite seguir cualquier tipo de camino o ruta
- Puede demostrarse que para trayectorias suficientemente suaves permite asegurar bajo error en posición y orientación.





Pure Pursuit

- En cada instante se obtiene el punto del camino más cercano a la posición actual del vehículo (x_n, y_n) .
- El punto objetivo (x_t, y_t) se calcula escogiendo el punto del camino que dista un valor s de (x_n, y_n) .
- Se aplica el algoritmo de control para converger al punto objetivo.
- Se realizan los cálculos para cada vez que se ejecuta el bucle de control.





Algoritmo Pure Pursuit

Pure_pursuit () {

definicion_camino

valores_iniciales

$v=v_0$ // se mantiene constante la velocidad

while(!fin)

{

posicion_actual=estimacion_posicion() // en nuestro caso odometria

punto=punto_mas_cercano()

punto_objetivo=punto+distancia

rho=calcula_curvatura(posicion_actual, punto_objetivo)

potencia=calculo_potencia(v, rho)

actuacion(potencia) // se manda al robot los valores de control calculados

}

}



Algoritmo Pure Pursuit

¡¡¡¡Atención!!!

El punto se mueve según lo hace el vehículo, es éste el que ‘empuja el punto hacia adelante’

