1.INTRODUCCION:

Los robots son agentes físicos que realizan tareas mediante la manipulación física del mundo, para lo cual están equipados con efectores como piernas, ruedas, articulaciones, etc.., y con sensores como cámaras, ultrasonidos, giroscopios, etc.., que les permiten percibir el entorno.

2.HARDWARE ROBÓTICO

Los robots hacen uso de una arquitectura de agente se compone de procesadores sensores y efectores. es en estos dos últimos, y concretamente en su correcto diseño y funcionalidad, en los que se basa el éxito real de los robots actuales. A continuación, daremos una breve explicación sobre ambos.

3.SENSORES

Los sensores son para los robots lo que para nosotros nuestros sentidos. Es con lo que perciben el entorno y aunque existen sensores de todo tipo, se pueden agrupar en tres clases principales:

De posición, como GPS o sónares.

De imagen, con cámaras o visión estereoscópica.

Perceptores, para informar al robot sobre su propio estado.

4.EFECTORES

Los efectores son el medio por el cual los robots se mueven y cambian la forma de sus cuerpos. Dependiendo de a cuantas direcciones independientes puedan moverse sus efectores, así contaremos sus grados de libertad. Por sí solos los sensores y efectores no hacen un robot. Es necesaria una fuente de energía para hacerlos funcionar. Usualmente un motor eléctrico, pero también pueden utilizar actuación neumática o hidráulica.

5.PERCEPCIÓN ROBÓTICA

La percepción robótica es el proceso por el cual los sensores del robot elaboran un mapa del entorno utilizando una determinada representación interna.

Esta representación interna debe tener tres propiedades:

Tiene que contener suficiente información para que le robot tome las decisiones apropiadas.

Ha de estar estructurada para que se pueda actualizar eficazmente

Tiene que ser natural, es decir, las variables internas han de corresponder con estados del mundo real.

6.LOCALIZACIÓN

La localización es el problema de determinar dónde están las cosas.

Si la posición del objeto es conocida, la localización es un problema de seguimiento (estos problemas tienen incertidumbre limitada)

Si no se conoce la posición inicial del objeto es un problema de localización global, que se convierte en un problema de seguimiento una vez se conoce la posición de este.

7.GENERACIÓN DE MAPAS

Suele denominarse localización y mapeado simultáneo,

Ya que el robot no solo debe ser capaz de construir un mapa, lo debe hacer sin ningún conocimiento sobre donde está.

El método mas utilizado es el filtrado de Kalmam extendido que suele combinarse con un modelo de detección de marcadores y requiere que todos los marcadores sean distinguibles

8.PLANEAR EL MOVIMIENTO

El movimiento punto a punto consiste en llevar al robot o a sus efectores a una posición determinada, pero lo interesante es el llamado movimiento obediente, que se trata de que el robot se mueva mientras está en contacto físico con un obstáculo como podría ser aflojar una bombilla o levantar una caja. Será necesario definir un espacio de configuración y la planificación de trayectorias.

9.ESPACIO DE CONFIGURACIÓN

El primer paso hacia la solución del problema es encontrar una representación apropiada. En este caso, A y B son posiciones del espacio y la mancha roja un obstáculo, pero, aunque pudiera parecer, este espacio no tendría por qué representar un sistema cartesiano, sino el espacio de trabajo con la posición del robot con respecto a sus articulaciones

10.DESCOMPOSICIÓN EN CELDAS

Para la planificación de rutas podemos descomponer el espacio en un número limitado de regiones más pequeñas, haciendo que llegar hasta el objetivo final pase por un conjunto de pequeñas metas más fáciles de alcanzar individualmente. Divide y vencerás. Pero esta representación tiene el problema de que el número de celdas crece exponencialmente con la distancia haciendo que sea muy poco eficiente en espacios grandes.

11.ESQUELETIZACIÓN

Otra opción es la esqueletización, que consiste en dividir el espacio en puntos finitos donde tres o más curvas unidimensionales intersectan. De esta manera, encontrar el camino más corto se reduce a la búsqueda en un grafo discreto. Este tipo de diagramas se conocen como diagramas de Voronoi, pero esta representación también sufre, aunque en menor medida, de que es difícil de aplicar en espacios de configuración de dimensiones más altas.

12.PLANIFICAR MOVIMIENTOS INCIERTOS:

En robótica llamamos movimientos inciertos a la observabilidad parcial del entorno y los efectos de las acciones del robot. Los errores pueden provenir de algoritmos de aproximación que no prevean que el robot modele el entorno perfectamente.

13.ALGORITMOS

La mayoría de robots de hoy en día usan algoritmos determinísticos para la toma de decisiones, una práctica común es extraer el estado más probable de la distribución de estados obtenida. En este caso obviar la incertidumbre funciona correctamente porque esta es mínima, pero nos puede llevar a problemas cuando esta sea demasiado grande. Para intentar evitar esto se usan las políticas que normalmente se llaman funciones de navegación, que le dice al robot que hacer en cada posible estado y saber que hacer en caso de error.

14.MOVIMIENTO

Hasta ahora hemos hablado de planificar movimientos no de cómo moverse, y asumíamos que el robot puede seguir cualquier ruta que el algoritmo produce, esto no es cierto en el mundo real, los robots tienen inercia y no pueden seguir rutas arbitrarias excepto a bajas velocidades. En la mayoría de los casos más que especificar posiciones los robots trataran de aplicar fuerzas.

15.DINÁMICA Y CONTROL

Los controladores son técnicas para generar controles del robot en tiempo real utilizando retroalimentación del entorno.

Un controlador se dice que es estable si pequeñas perturbaciones llevan a un error limitado entre el robot y la señal de referencia, y es estrictamente estable si capaz de retornar a su ruta de referencia a pesar de tales perturbaciones.

16.CONTROL DEL CAMPO POTENCIAL

Se basa en definir una fuerza atractiva que empuje al robot hacia su configuración objetivo y un campo potencial repulsivo que empuje al robot lejos de los obstáculos.

Esta es una forma muy eficiente de encontrar la ruta al objetivo, pero tiene desventajas importantes como que las fuerzas que generan dependen solo de la posición del robot y de los obstáculos, no de la velocidad del robot. Y pude fallar si el robot se mueve rápidamente .

17.CONTROL REACTIVO

Es el diseño de agentes guiados por reflejos, estos se usan cuando no es posible generar mapas con la suficiente precisión para ser efectivos, o la generación de estos es computacionalmente intratable.

Estos se basan en controladores simples que definen unas reglas básicas, para sortear los obstáculos, y guiados por la retroalimentación que reciben del entorno.

El comportamiento que surge de por la interacción de entre un controlador simple y un entorno complejo es llamado comportamiento emergente.

18.ARQUITECTURAS DE HADWARE ROBOTICO:

Una metodología para estructurar algoritmos es la llamada arquitectura software. Una arquitectura incluye normalmente lenguajes y herramientas para escribir programas, así como una filosofía que nos dice como los programas deben unirse. La mayoría de las arquitecturas usan técnicas reactivas en los niveles más bajos de control y técnicas deliberativas en los niveles más altos por tanto son llamadas arquitecturas hibridas.

19.ARQUITECTURA EN TRES CAPAS

La arquitectura hibrida más popular es la arquitectura de tres capas que son:

1. capa reactiva provee de control de bajo nivel al robot su ciclo de decisión se encuentra en el orden de los milisegundos.
2. capa ejecutiva sirve como unión entre la primera y la tercera capa acepta directivas de la capa deliberativa y las encadena a la capa reactiva integra la información de los sensores en una representación interna del estado.
3. capa deliberativa genera soluciones globales para tareas complejas utilizando la planificación

Además de estas, algunos software robóticos incluyen capas adicionales.

20.LENGUAJES DE PROGRAMACIÓN ROBÓTICA

Algunas partes de la arquitectura son implementadas en el lenguaje de comportamiento basado en reglas y en el tiempo real que compila los controladores, en total lenguajes de programación similares en esencia al lenguaje de comportamientos, pero más poderosos y rápidos en ejecución como puede ser el lenguaje robótico genérico funcional para programar grandes sistemas de control modulares.