PROGRAMACIÓN, PLANIFICACIÓN Y CONTROL EN ROBÓTICA *

Raúl Suárez

Institut d'Organització i Control de Sistemes Industrials (UPC)
Diagonal 647 planta 11, 08028 Barcelona, España
TE.: +34-934016548, Fax: +34-934016605, e-mail: suarez@ioc.upc.es

Resumen: En esta ponencia se pretende realizar una rápida revisión de tres campos de trabajo en robótica, la programación, la planificación y el control, que si bien han sido tradicionalmente tratados de forma independiente tienen algunos puntos de estrecha relación, a los que se prestará especial atención. La ponencia no pretende llevar a cabo una revisión exhaustiva de cada uno de los temas, que por sí mismos han dado material suficiente para una extensa bibliografía, sino que la pretensión es dar una visión global de ellos, destacando aspectos relevantes e incluyendo ejemplos cuando estos faciliten su ilustración. Este resumen escrito es un guión de la presentación oral en el que se describen sucintamente los principales aspectos tratados.

DE LA PROGRAMACIÓN ...

Ya en las primeras definiciones de robot se usaba el concepto de "...manipulador reprogramable..." para realizar "...movimientos programados..." lo que de por sí muestra que la programación es un aspecto fundamental en robótica. No obstante, en la corta existencia de la robótica como área específica, el enfoque de la programación ha cambiado bastante en términos de investigación y desarrollo pero no mucho en términos de las aplicaciones industriales (posiblemente muy recientemente empieza a haber cambios significativos en este sentido). Son especialmente destacables algunos trabajos de los años ochenta sobre programación de robots que en gran medida quedan hoy en día totalmente superados; buenas revisiones del estado del arte en aquel momento pueden encontrarse en los trabajos de Bonner y Shin (1982), Lozano-Perez (1983) y Gruver *et al.* (1984). Ya en aquellos años se realizó una clasificación de programación de robots que aún es plenamente vigente y que puede resumirse en el esquema mostrado en la Figura 1.

*Este trabajo fue parcialmente financiado por los proyectos CICYT TAP98-0471 y TAP99-0839.

- Programación
  - No-textual
    - Hardware
      - Programa cableado
      - Programa definido mecánicamente
    - Gestual
      - En línea
        - Modo pasivo
        - Modo activo
      - Fuera de línea
  - Textual
    - Explícita
      - Nivel robot
      - Nivel objeto
    - Implícita
      - Nivel objeto
      - Nivel tarea
      - Nivel objetivo

Figura 1. Tipos de programación de robots

Se distinguen dos grandes categorías según exista un programa textual o no, caso este último ya casi inexistente en la práctica pero cuyo principio básico aún es muy utilizado. Dejando de lado la programación por hardware, realmente en desuso, la programación gestual ocupó (y ocupa en combinación con la programación textual explícita) un lugar relevante. Básicamente consiste en mover el brazo manipulador, normalmente mediante un sistema de *joystick* o de botonera, a las posiciones por las que se desea que pase al ejecutar la tarea y memorizarlas para luego repetirlas. Es claro que mientras se realiza este trabajo de toma de posiciones, que además es sumamente tediosos y lento, el robot no está disponible para trabajar en producción, y esa es una de sus principales desventajas. Los avances de la informática, tanto en velocidad de procesamiento como en algoritmos específicos, están permitiendo actualmente realizar este trabajo en simulación fuera de línea, con lo que se ahorra un tiempo considerable ya que una vez trasladado el programa al robot real sólo se deben hacer ajustes finales. Un tema de actualidad como es la "realidad virtual" tiene un claro papel en este campo.

Como complemento a la programación gestual se tiene la programación textual, en la cual el programa puede expresarse mediante un cierto texto, almacenable independientemente del robot. Este sistema permite "escribir" un programa sin necesidad de disponer del robot real. En la práctica así sucede, pero en un alto porcentaje de casos, aunque exista la posibilidad de expresar posiciones mediante una "línea de texto" sucede que las posiciones claves de la tarea siguen obteniéndose mediante guiado del robot, ya que es la forma de superar ciertas incertidumbres en el posicionamiento relativo entre el robot y su entorno. El desarrollo de sistemas sensoriales más complejos y la capacidad de los computadores para tratar más información a mayores velocidades (un claro ejemplo son los sistemas de visión por computador) está permitiendo también reducir la captura de posiciones fijas y aumentar la versatilidad de los robots en entornos no estructurados.

Dentro de la programación textual pueden establecerse dos niveles, según las especificaciones de movimiento sean realizadas de forma explícita o implícita. En el primer caso, las especificaciones pueden referirse al robot mismo o al objeto que este manipula (ejemplos representativos clásicos: lenguaje VAL de 1975 y sucesores (Unimation, 1980) y lenguaje AML de 1977 (Taylor, Summers, and Meyer, 1982)). Los lenguajes de programación de robots comerciales caen en su gran mayoría en el apartado de lenguajes de programación textuales explícitos referidos al robot e incluyen la capacidad de captación de posiciones mediante sistemas no-textuales de guiado activos en línea. En el segundo caso, cuando los movimientos son indicados de forma explícita, existen diferentes niveles de abstracción y en función de ello se suelen distinguir tres niveles: nivel robot, nivel tarea y nivel objetivo (ejemplos representativos clásicos: lenguaje AUTOPASS de 1977 (Lieberman and Wesley, 1977) y RAPT de 1978 (Popplestone, Ambler, and Bellos, 1978)).

En paralelo a este esquema cabe destacar otros dos aspectos. Por un lado está la influencia en la programación textual de la programación orientada a objetos, que ha dado lugar a algunos lenguajes de programación de robot en esta línea pero que no han llegado a versiones comerciales de relevancia. Por otro lado está la influencia de las técnicas de aprendizaje típicas del área de inteligencia artificial, que intentan que el robot "aprenda" o se "autoprograme" a partir de sucesivas demostraciones de la tarea realizadas por un operario, ya sea utilizando el robot real y los sensores necesarios para capturar señales que adecuadamente tratadas permitan deducir la estrategia de acción seguida, o mediante sistemas de simulación, por lo general con interfases gráficas que el operador usa para realimentación visual. Este último enfoque, que es una ambiciosa extensión de la programación gestual, tiene cierto auge dado que una solución robusta permitiría transferir de forma intuitiva al robot la estrategia que sigue el operador humano para realizar una tarea sin necesidad de conocer complicados y poco flexibles lenguajes de programación. Este enfoque ha sido denominado "programación por demostración" (ej.: Takahashi, Ogata and Muto, 1993; Kang and Ikeuchi, 1995; Kaiser and Dillman, 1996; Wan *et al.*, 1996).

En cualquier caso, cuando el lenguaje de programación es de alto nivel de abstracción, y por tanto permite especificar tareas sin entrar en detalles de las mismas, es el "interprete" o "compilador" del programa quien de alguna manera debe pasar de las ordenes de alto nivel a comandos de movimiento de nivel robot. De este modo, instrucciones como por ejemplo "Insertar la pieza A en la pieza B" sin más detalle implican que el propio programa ha de resolver por lo menos los siguientes aspectos: dónde están actualmente las piezas A y B, cómo se ha de sujetar la pieza A en la pinza del robot, qué trayectoria debe seguir el robot hasta llegar a la pieza A para aprehenderla y cuál con la pieza A en la pinza para insertarla en B (ambas sin colisionar con otros elementos del entorno), qué estrategia y tipo de control se ha de utilizar para insertar A en B sin que se produzca un atasco entre las piezas, entre otros aspectos. La resolución automática de estas cuestiones de forma transparente al operador, imprescindible en la programación de alto nivel, nos sitúa ya en un campo específico como es la **planificación de tareas y movimientos en robótica**.

... A LA PLANIFICACIÓN ...

La planificación de tareas y movimientos de un robot también admite, como la programación, una clasificación según el nivel de abstracción al que se realice el trabajo. En cualquier caso los generadores de planes admiten una descripción genérica (independientemente de que sea para robótica o para otro campo) como un módulo al que se le debe dar un objetivo y una situación inicial y que en función de ellos devuelve un plan de acción (Torras, 1989). El módulo debe poseer (o ser capaz de obtener), por ejemplo, una descripción del entorno, las acciones posibles, estrategias predefinidas, restricciones especificas, entre otra información. El enfoque más generalizado de funcionamiento es mediante el fraccionamiento de la tarea en un número finito de estados diferentes y hacer un plan consistente en una secuencia de estados que lleve del inicial al deseado acompañada de los operadores para realizar las transiciones de un estado a otro. La forma más usual de representación del problema consiste en el uso de grafos de estados, en los que los nodos representan los estados de la tarea y los arcos las distintas transiciones posibles, puede que con algún índice de coste de la transición asociado a los mismos. De esta manera, el problema de planificación se convierte en el problema de búsquedas de caminos en grafos, donde se pueden aplicar diversas estrategias.

Dentro del ámbito de la robótica, una vez resuelto el problema de las distintas fases o estado de la tarea se debe pasar a la planificación de las acciones concretas del brazo manipulador. A este nivel pueden distinguirse dos problemas básicos, la planificación de la aprehensión y la planificación de las trayectorias del robot, que a su vez pueden ser en el espacio libre evitando colisiones o en contacto cuando este sea inevitable.

La planificación de la aprehensión es un problema que se complica cuanto más versátil es la pinza o “mano” utilizada. Básicamente ha de realizar la determinación de los puntos de la superficie del objeto a aprehender sobre los que se deben apoyar los dedos o elementos de agarre de la pinza utilizada. Esto requiere acciones complejas teniendo en cuenta que se debe considerar: la forma del objeto a manipular, la forma (cinemática) de la pinza, las características físicas del contacto (ej. fricción), y la acción a realizar posteriormente (para saber por donde se debe asir el objeto); eventualmente también se deben tener en cuenta la estabilidad del agarre cuando la masa o las fuerzas de contacto sean significativas (Ponce and Faverjon, 1991; Bruyninckx, Demey and Kumar, 1998). En caso de que se utilicen dispositivos “multi-dedo”, del tipo de la mano humana, es posible acudir a estrategias o modos de agarre predefinidos en función de la tarea a realizar (Craig, Kenneth and Andrew, 1998).

La planificación de movimientos libres de colisión es un campo que ha tenido un gran desarrollo, en particular debido a los denominados robóts móviles, que, en la mayoría de los casos, de forma natural reducen el problema a tres grados de libertad, haciéndolo accesible a aplicaciones reales. El caso de un manipulado robótico con 6 o más grados de libertad genera un problema bastante más complejo. La herramienta más popular en este campo es el Espacio de Configuraciones (Lozano-Perez, 1983), es decir el espacio determinado por el conjunto de variables necesarias para determinar unívocamente la posición de los

GEOMÉTRICAS

Tolerancias en la manufactura

- Forma y tamaño de los objetos

Observación de los objetos

- Medición de la posición de un punto de un objeto
- Determinación de la configuración de un objeto

Manipulación del robot

- Posición y orientación del elemento terminal
- Deslizamientos del objeto en la pinza del robot

DE FUERZA

Precisión del sensor de fuerza y par

DE VELOCIDAD

Errores del sistema de control del robot

Figura 2. Fuentes de incertidumbre en una tarea de ensamblado automatizada

objetos móviles involucrados en la tarea; por ejemplo, si como elemento móvil sólo existe un robot de seis grados de libertad bastan seis variables (aunque pueden usarse diferentes conjuntos de seis variables) para describir la posición de todos los elementos en el espacio de trabajo. En el Espacio de Configuraciones pueden representarse de manera muy directa las posiciones accesibles e inaccesibles del robot, y por tanto realizar un plan de movimientos se reduce a encontrar un camino continuo en dicho espacio cuyo puntos sean todos accesibles. Para ello existen diferentes métodos, como por ejemplo, entre los más usados, los basados en: "grafos de visibilidad", diagramas de Voronoi, particiones del espacio libre y búsqueda de caminos entre cada par de subconjuntos vecinos (existen varias maneras de hacerlo), y funciones potenciales (Latombe, 1991; Sheu and Xue, 1993; Bolles, Bunke and Noltemeier, 1997; Gupta and Del Pobil, 1998).

Los movimientos libres de colisión como los mencionados, en los que hay un margen de seguridad, se denominan "movimientos groseros" (en inglés *gross motion*) en tanto que aquellos movimientos que pueden producir (o que con seguridad producen) contacto entre el robot u objeto manipulado y el entorno se denominan "movimientos finos" (en inglés *fine motion*). La planificación de movimientos finos ha sido menos tratada y es un tema en el que aun se buscan soluciones definitivas que sean robustas y a la vez de fácil aplicación práctica (Rosell, Basañez and Suárez, 1997). La principal restricción es que las incertidumbres que afectan la tarea son difíciles de predecir y modelar y que la ocurrencia de contacto real genera varias complicaciones de carácter práctico, entre ellas la necesidad de un sistema de control adecuado. Los movimientos finos son necesarios en tareas como el ensamblado con holguras pequeñas donde difícilmente se puede evitar el contacto entre las partes antes de llegar a la posición deseada final ya que la precisión del robot está en el orden de las incertidumbres existentes (Suárez and Basañez, 1991). Las fuentes de incertidumbre que afectan una tarea se listan en la Figura 2 (Basañez and Suárez, 1991; Rosell, Basañez and Suárez, 1999).

En el campo de la planificación de movimientos finos se pueden distinguir básicamente tres enfoques. El primero de ellos se basa en el concepto de "retroproyección" (en inglés *backprojection*) y tuvo una muy amplia difusión entre la comunidad científica (Lozano-Perez, Mason and Taylor, 1984; Erdmann, 1984; Buckley, 1989). Una retroproyección de un conjunto de puntos del espacio, llamado objetivo, es el subconjunto de puntos de este espacio desde el cual se puede garantizar que con una dirección de movimiento dada se alcanza el objetivo y que, cuando se alcance, el objetivo será detectable. Este concepto se aplica en el Espacio de Configuraciones de forma bastante directa cuando se consideran sólo grados de libertad de traslación e incertidumbre en la dirección de movimiento. La idea es buscar un conjunto de retroproyecciones que lleven desde el objetivo final deseado hasta la configuración inicial de sistema, y luego aplicar los correspondientes movimientos para ir cubriendo los objetivos parciales. El enfoque es formalmente correcto pero de complicada implementación práctica.

Un segundo enfoque plantea una solución de dos etapas, en la primera se realiza un plan utilizando modelos nominales ignorando los efectos de las incertidumbres y en la segunda se hacen arreglos locales del plan original donde se prevé que la incertidumbre pueda afectarlo (Xiao and Volz, 1989; Rosell, Basañez and Suárez, 1999). Existen diferentes maneras de llevar a cabo cada una de las dos fases.

El tercer enfoque se basa en un análisis del espacio de contacto, y la tarea se representa como un conjunto de estados definidos por la situación de contacto entre el objeto manipulado por el robot y el entorno de trabajo (Suárez, Basañez and Rosell, 1995). Los mayores problemas en este enfoque aparecen en relación a la identificación del estado actual (es decir de la situación de contacto) y en la determinación de los operadores para cambio de estado, en ambos casos y como sucede en los principales problemas de los enfoques anteriores debido a la incertidumbre.

En cualquiera de los enfoques anteriores, cuando se pretende seguir una trayectoria (libre o de contacto) y se produce un contacto (previsto o no), la trayectoria real que va a llevar a cabo el robot, así como las fuerzas de interacción que aparezcan entre los objetos, dependerá lógicamente de la situación y condiciones de los objetos en contacto, pero además dependerá también, y en gran medida, de como reaccione el **sistema de control del robot**.

... Y EL CONTROL.

El sistema de control de un robot tiene fundamentalmente tres objetivos: el control de movimiento (sea para llegar a una posición o para seguir una trayectoria), el control de la fuerza que ejerce sobre el entorno, o el control conjunto de movimiento y de fuerza. En cada caso existen algunas características particulares.

El robot como sistema a controlar tiene como principales características el ser un sistema no lineal, altamente acoplado y variable en el tiempo (nótese que la dinámica del brazo cambia con el peso de la carga). La forma más común de representar este tipo de sistemas es mediante la ecuación $M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau$, donde q es el vector de variables articulares, M es la matriz de inercias, V es la matriz de las fuerzas centrífugas y de Coriolis, G es el vector de fuerzas gravitatorias y τ el vector de los pares aplicados en las articulaciones. En un sistema real M, V y G no suelen ser conocidas con precisión, pero su error está acotado. Así, el problema de control de movimiento consiste en: dados unos valores de q, y a veces $\dot{q}$, determinar los pares τ necesarios para llevarlos a cabo (Paul, 1981; Craig, 1986; Asada and Slotine, 1986; Spong and Vidyasagar, 1989; Murray, Li and Sastry, 1993).

Existen diferentes maneras de diseñar el sistema de control y de ajustar sus parámetros (como en programación y en planificación en este tema se podría hablar de forma general sin mencionar que se está pensando en un robot). Los enfoques van desde el control independiente de cada articulación mediante un controlador PID convencional ($\tau = -k_p(q - q^*) - k_v(\dot{q} - \dot{q}^*) - k_i \int (q - q^*)\mathrm{dt}$ con q^* el valor de consigna y k_p, k_v y k_i constantes positivas) (Spong and Vidyasagar, 1989) hasta el uso del propio modelo del manipulador para realizar un control por par calculado, es decir, se utiliza un modelo de la dinámica y cinemática inversa del brazo para calcular el par que debe producir un cierto movimiento deseado, luego este par se aplica al robot real con la intención de conseguirlo. Debido a que el modelo del robot no es conocido con precisión, y a que la carga es desconocida en el momento del diseño del controlador este método está sujeto a imprecisiones (que pueden ser acotadas). Para tratar de minimizar estos efectos y no depender de modelos exactos de los manipuladores se han utilizado también técnicas de control robusto (Addallah *et al.*, 1991), aplicando por ejemplo control deslizante (Asada and Slotine, 1986).

Dado que los robots suelen ser utilizados en tareas repetitivas también se ha utilizado el control adaptativo, intentando que el robot se adapte a unas condiciones de trabajo determinadas (por ejemplo a una carga dada) o a un ciclo de trabajo (Hsia, 1986; Craig, Hsu and Sastry, 1987; Slotine and Li, 1987), sin embargo este tipo de enfoques parece haber perdido un poco de peso en los últimos años. También basándose en la idea de que los robots realizan tareas repetitivas, y que por lo tanto el accionar del robot puede mejorarse a medida que se va repitiendo la tarea, se han presentado enfoques de control con aprendizaje, usando por ejemplo redes neuronales para "aprender" la dinámica del robot (Zalzala and Morris, 1996).

Otros campos que merecen un tratamiento particular son el control de manipuladores flexibles, que aunque no son frecuentes en aplicaciones industriales son promisorios en otras aplicaciones, y el control de manipuladores redundantes (ya sea un brazo con numerosas articulaciones o el control conjunto de más de un brazo), cuya particular estructura permite múltiples soluciones.

Abordaremos ahora el problema de control de la fuerza F que aplica el robot sobre el entorno (normalmente se realiza conjuntamente con un control de la posición P del robot) que tiene una estrecha relación con la planificación de movimientos finos (Suárez, 1988). En relación al control de fuerza cabe distinguir los siguientes casos:

- *Control de P solamente.* Es útil cuando se trabaja en el espacio libre, pero puede presentar problemas cuando hay interacción entre el robot y su entorno de trabajo. Una imprecisión en la posición puede producir atascos o fuerzas desmedidas que dañen las piezas o el robot mismo.

- *Control de F solamente.* Se trata de cumplir una consigna de fuerza, independientemente de la posición en la que se encuentre. Sólo tiene sentido cuando se garantiza contacto entre el robot y el entorno de trabajo, de lo contrario el robot se "perdería" al no tener posición de referencia.

- *Control de P con F proporcional al error de P (Control de elasticidad).* Se establece una relación lineal entre el error de posición ΔP y F, con lo que el efecto logrado es equivalente al de un resorte que ejerce una fuerza de reacción proporcional a su deformación.

- *Control de P con F proporcional al error de velocidad (Control de amortiguamiento).* Se establece una relación lineal entre el error de velocidad ΔV y F, con lo que el efecto logrado es el de un amortiguador viscoso que produce una fuerza de reacción proporcional a la velocidad con que se lo pretenda desplazar. Debe destacarse que cuando el camino geométrico inicial es forzado a cambiar debido a un obtáculo luego no es recuperable, y que la fuerza ejercida contra el obstáculo es proporcional a la variación de la velocidad prevista.

- *Control de P con umbral de fuerza.* El comportamiento es el de un control de posición con la particularidad de que, cuando ante un obstáculo aparecen fuerzas superiores a un umbral predefinido, tiene prioridad el intentar mantener el valor de fuerza dentro del rango permisible.

- *Control de impedancia.* Se intenta producir una fuerza de reacción proporcional a la variación de energía cinética y potencial del móvil respecto a una referencia. El efecto logrado es una combinación lineal de los mencionados para el control de elasticidad y de amortiguamiento.

Estas situaciones asumen un control continuo de la fuerza de contacto, pero también puede usarse la información de fuerza para tomar decisiones lógicas, es decir, usar la fuerza simplemente para detectar una situación límite del movimiento y esa situación realizar un cambio de consigna.

Los esquemas básicos de control de posición/fuerza que han dado lugar a numerosas realizaciones son los siguientes:

- *Control Híbrido de Posición/Fuerza.* El objetivo del control híbrido es controlar la fuerza según algunos ejes de un sistema de coordenadas cartesianas, y la posición en todos los restantes, haciendo que todas las articulaciones del robot contribuyan a seguir ambos conjuntos de consignas mediante dos lazos de control paralelos, uno para la fuerza y otro para la posición (Raibert, 1981).

En este esquema, la señal de control para cada actuador es del tipo,

$$\tau_i = \sum_{i=1}^{n} [\Phi_{ij}(s_j \cdot \delta f_j) + \Psi_{ij}((1 - s_j) \cdot \delta p_j)]$$

donde:

τ_i **:** par a aplicar por el actuador i
δf_j **:** error de fuerza en la componente j del sistema de coordenadas cartesianas
δp_j **:** error de posición en la componente j del sistema de coordenadas cartesianas
Φ_{ij} **:** función de transferencia entre el error de fuerza en la componente cartesiana j y la articulación i
Ψ_{ij} **:** función de transferencia entre el error de posición en la componente cartesiana j y la articulación i
s_j **:** componente del vector de selección $\overline{s}$ compuesto por unos y ceros según corresponda control de fuerza o posición. Por lo tanto, si por ejemplo se desea controlar fuerza y par según el eje z del sistema cartesiano y posición y giro según los ejes x e y, el vector de selección será $\overline{s} = (001001)^T$, representando los 3 primeros elementos traslación o fuerza según cada eje coordenado y los 3 últimos rotación o par sobre los mismos ejes.

Cada uno de los términos de la sumatoria es la aportación de los lazos de control de fuerza y posición respectivamente.

- *Control Activo de Elasticidad en Coordenadas Cartesianas.* El concepto básico de un control de elasticidad, como ya se dijo, reside en establecer una relación lineal entre la fuerza realizada y el error de posición (Salisbury, 1980). Esto equivale a la ley de Hooke, que al extenderla a 6 dimensiones (3 translaciones y 3 rotaciones) puede ponerse en forma matricial como:

$$F_a = K \cdot \Delta P$$

donde (todos expresados en coordenadas cartesianas):
F_a : es el vector de fuerzas a controlar
K : es la matriz de elasticidad
ΔP : es el vector de errores de posición respecto a la posición deseada

Si el error es evaluado en coordenadas del robot, es necesario encontrar la relación equivalente en estas coordenadas. Para ello se parte de la relación entre el error de posición expresado en coordenadas cartesianas y en coordenadas del robot $\Delta P = J \cdot \Delta Q$, siendo J el Jacobiano del manipulador, y de la relación entre los pares en las articulaciones y las fuerzas en coordenadas cartesianas, $T_a = J^T \cdot F_a$. Usando estas expresiones junto a $F_a = K \cdot \Delta P$ se obtiene $T_a = J^T \cdot F_a = J^T \cdot K \cdot J \cdot \Delta Q = K_q \cdot \Delta Q$ donde $K_q = J^T \cdot K \cdot J$ se denomina matriz de elasticidad en las articulaciones.

La matriz K expresada en el marco de acomodación es por lo general diagonal, siendo sus elementos no nulos la elasticidad deseada según cada eje. La matriz K_q resulta claramente no-diagonal, lo que significa que un error de posición en una articulación puede hacer que se generen pares en otras articulaciones.

- *Control de Amortiguamiento.*

  El control de amortiguamiento implica una relación lineal entre las fuerzas ejercidas y el error de velocidad (Whitney, 1977)

  $$F_a = K_f \cdot \Delta V$$

  Existe un sistema de coordenadas, normalmente el marco de acomodación, en el que la matriz K_f suele ser diagonal. Sus valores no nulos relacionan las fuerzas medidas con los cambios de velocidad. Es decir K_f es del tipo,

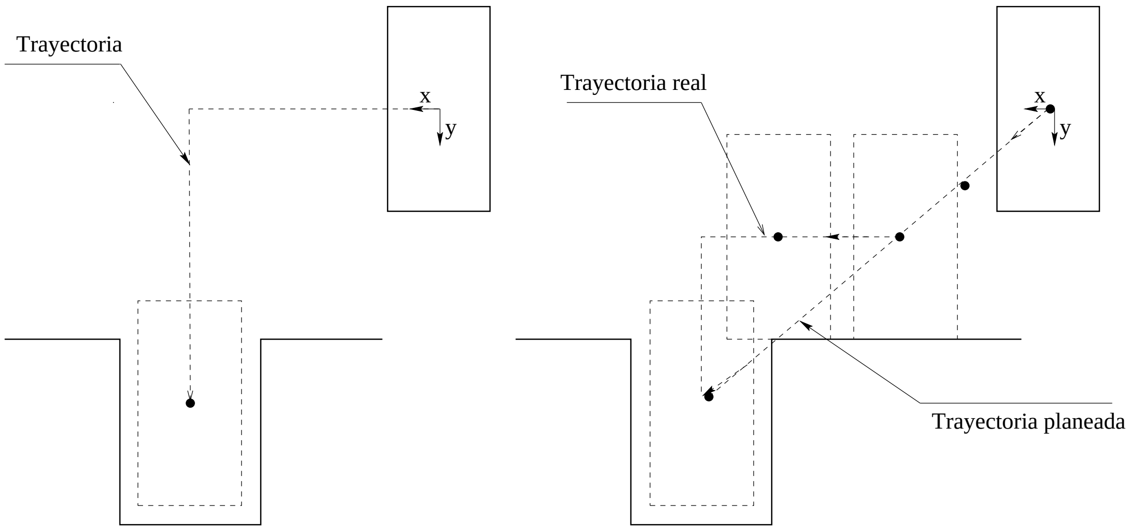
  $$K_f = \begin{pmatrix} k_{f_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{f_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{f_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{f_4} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{f_5} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{f_6} \end{pmatrix}$$

  Otra característica de este sistema es que si la velocidad deseada es cero, es decir el manipulador se halla en una posición deseada, y se le aplica una fuerza externa al extremo del brazo, este saldrá de la posición actual con una velocidad proporcional a la fuerza que se aplique. Por ejemplo, se podría en este caso "mover" el manipulador conduciéndolo con la mano.

- *Control de Impedancia.* El efecto total es la suma de los efectos de amortiguamiento y elasticidad (Hogan, 1980).

RELACIÓN ENTRE LOS TRES CAMPOS

Es claro que en algunas circunstancias la trayectoria planeada y programada puede ser perfectamente ejecutada por el robot, como en el caso mostrado en la Figura 3a. En otros casos el uso de un adecuado sistema de control permite planear trayectorias más sencillas o directas, pero cuya ejecución será diferente, como en el caso de la Figura 3b. El programador o el sistema de planificación han de saber de que tipo de control disponen para poder sacar ventaja de esta situación. El uso de sistemas de control de posición/fuerza permite también, por ejemplo, que mediante la información de un sensor de fuerza el sistema pueda realizar movimientos reactivos a las fuerzas que le aplica un operario, quien podría posicionar el brazo manipulador a su antojo de forma instintiva y muy directa, simplemente empujándolo. De este modo se puede disponer de un método muy práctico para programar el robot por guiado. Así, resulta evidente la relación que existe entre los tres temas tratados, la programación, la planificación y el control en robótica, tal como se ilustra esquemáticamente en la Figura 4.



a) b)

Figura 3. Diferencia entre trayectoria planeada y trayectoria real.

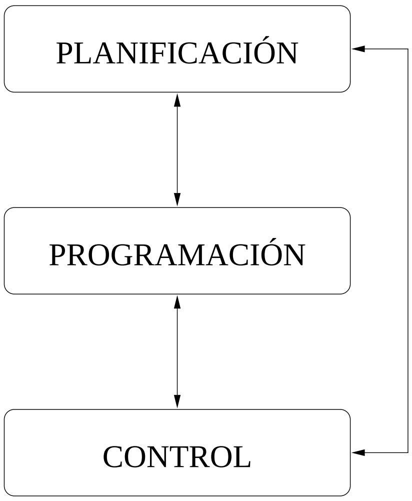


Figura 4. Relación entre la programación, la planificación y el control de robots

AGRADECIMIENTOS

Agradezco a los organizadores del Primer Encuentro Nacional de Robótica y Manufactura Flexible de la Fundación Universidad del Norte el haberme invitado a impartir esta conferencia, así como su paciencia a la espera de este resumen enviado casi al agotar los plazos. A Sonia Vadala por su colaboración en la edición de este resumen.

REFERENCIAS

Addallah, Dawson, Dorato and Jamshidi (1991). IEEE Control System Magazine, Vol.11, Num. 2, pp 24-30.

Asada and Slotine (1996). *Robot analysis and control.* John Wiley & SIns Inc.

Basañez L. and Suárez R. (1991). "Uncertainty modelling in configuration space for robotic motion planning". Proceedings of the IFAC/IFIP/IMACS Symposium on Robot Control, SYROCO'91, Viena, Austria, septiembre de 1991, Pergamon Press, ISBN 0-08-041276-9, Oxford 1992, pag. 247-252.

Bolles, R. C. Bunke, H. Noltemeier, H. (Ed). (1997) "Intelligent Robots. Sensing, Modeling and Planning". World Scientific.

Bonner S., and K G. Shin. " A Comparative Study of Robot Languages". AML: IBM robot System/1, Vols. 1-3. IBM Corporation, Sept. 1.981.

Bruyninckx H., Demey S and Kumar V., (1998). Generalized Stability of Compliant Grasps, Proc. of the IEEE int. Conf. On Robotics and Automation, Leuven, pp. 2396-2402.

Buckley S.J. (1989). Planning Compliant Motion Strategies. *International Journal of Robotics Research*, Vol. 8 (5), pp. 28-44.

Craig J. (1986) *Introduction to robotics: mechanics & control.* Addison Wesley Publishing Comp.

Craig R., Kenneth H. and.Andrew E., (1998). On Optimizing the Kinematic Geometry of a Dextrous Robot Finger, International Journal of Robotic Research, Vol. 17, No. 10, October 1998.

Craig, Hsu and Sastry (1987). Adaptive control of Mechanical Manipulators. Int. Journal of Robotics Research, vol 6, num 2.

Erdmann M. (1984). On Motion Planning with Uncertainty. *MIT Artificial Intelligence Laboratory report AI-TR-810 (Master Thesis).*

Gruver W. A., Soroka B., J. J. Craig, and T. Turner " Industrial Robot Programming languages: A Comparative Evaluation". IEEE Transactions on Systems, man, and Cybernetics, Vol. SMC-14, no. 4, July/August 1.984.

Gupta, Kamal. Del Pobil, Angel P. (Ed). "Practical Motion Planning in Robotics". John Wiley & Sons. 1.998.

Hogan N. (1980).] Mechanical Impedance Control in Assistive Devices and Manipulators. *Proceedings of the 1980 Joint Automatic Control Conference*, San Francisco, USA, pp. TA10-B.

Hsia T. (1986). Adaptive control for Robot Manipulators - A review. Int. Conf. on Robotics and Automation, San Francisco,USA.

Kang S. and Ikeuchi K. Toward Automatic Robot Instruction from Perception-Temporal Segmentation of Tasks from Human Hand Motion. IEEE Transactions on Robotics and Automation. Volume 11, No 5, October 1995. Pags 670-681

Kaiser M. and Dillmann R. (1996). Building Elementary Robot Skills from Human Demostration. Proc. IEEE International Conference on Robotics and Automation. Minneapolis, Minnesota, Pags 2700-2705.

Latombe, Jean-Claude (1991). "Robot Motion Planning". Kluwer Academic Plublishers.

Lieberman, L.I. and M.A. Wesley, (1977) "AUTOPASS: An automatic programming system for computer controlled mechanical assembly", IBM J. Res. Devel., vol. 21, no. 4, pp. 321-333, 1977.

Lozano Perez T. (1983). Spatial Planning: A Configuration Space Approach. *IEEE Trans. on Computers*, Vol. C-32 (2), pp. 108-120.

Lozano-Perez, Tomás. "Robot Programming". Proceedings of the IEEE. Vol. 71. No. 7, July 1983.

Lozano-Perez T., M.T. Mason and R.H. Taylor, (1984). "Automatic Synthesis of Fine-Motion Strategies" *The International Journal of Robotics Research*, Vol.3, No.1, pp.3-24.

Murray, Li and Sastry (1993). *A mathematical introduction to robot manipulation*, CRC Press.

Paul R. (1981) *Robot manipulators : mathematics, programming & control* MIT Press, 1981

Ponce J., Faverjon B., (1991). On Computing Three-Finger Force-Closure Grasp of Poligonal Objects, Proc. of the IEEE Trans. On Robotics and Automation, pp 1018-1023.

Popplestone, R. J. , A. P. Ambler, and I. Bellos, (1978) "RAPT, A language for describing assemblies", Industr. robot, vol. 5, no. 3, pp. 131-137, 1978.

Raibert M.H. y Craig J.J. (1981). Hybrid Position/Force Control of Manipulators. *Trans. of ASME, Journal of Dynamic Systems, Measurement and Control*, Vol. 102, June, pp. 126-133.

Rosell J., Basañez L. and Suárez R., (1997). "Determining Compliant Motions for Planar Assembly Tasks in the Presence of Friction" *Proceedings of the 10th IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'97*, (ISBN 0-7803-0123-4) Grenoble, France, September 1997, pp. 946-951.

Rosell J., L. Basañez and R. Suárez, (1999). "Improving the Performance of Compliant Motions by On-line Geometric Uncertainty Reduction" Proceedings of the 1999 IEEE International Symposium on Assembly and Task Planning, ISATP'99, (ISBN 0-7803-5704-3), Porto, Portugal, July 21-24, 1999, pp. 169-174.

Rosell J., Basañez L. and Suárez R., (1999). "Compliant-motion Planningand Execution for Robotic Assembly" *Proceedings of the 1999 IEEE International Conference on Robotics and Automation, ICRA'99,* Detroit, Michigan, USA, May 10-15, 1999, pp. 2774-2779.

Salisbury J.K. (1980). Active Stiffness Control of a Manipulator in Cartesian Coordinates. *Proceedings of the 19th IEEE Conference on Decision and Control*, pp. 95-100.

Sheu, Phillip C-Y. Xue, Q. (1993). "Intelligent Robotic Planning Systems". World Scientific.

Slotine and Li, (1987). On the adaptive control of robot manipulators. Int. Journal of Robotics Research, vol.6 num 3.

Spong and Vidyasagar (1989) *Robot dynamics and control* John Wiley & SIns Inc.

Suárez R. (1988) "Control de Fuerza en Robótica" *Automática e Instrumentación*, Num.185, pag.221-235, nov. de 1988.

Suárez R. and Basañez L. (1989). "Automatic fine-motion planning based on position/force states" *Proceedings of the IFAC/IFORS/IMACS/IFORS 6to. Symp. on Information Control Problems in Manufacturing Technology,* INCOM'89, Madrid, España, septiembre de 1989, Pergamon Press, ISBN 0-08-037023-3, London 1990, pag. 301-307.

Suárez R. and Basañez L. (1991). "Assembly with robots in presence of uncertainty" *Proceedings 22nd International Symposium on Industrial Robots, ISIR'91*, pag. 19/1–19/15, Detroit, EEUU, octubre de 1991.

Suárez R., Basañez L. and Rosell J., (1995). Using Configuration and Force Sensing in Assembly Task Planning and Execution *Proc. of the 1995 IEEE Int. Symposium on Assembly Task Planning*, pp.273-279.

Taylor, R. H. , P.D. Summers, and J.M. Meyer, (1982) "AML: A manufacturing language", Robotics res., vol. 1, no. 3, Fall 1.982.

Takahashi T., Ogata H. and Muto S. A (1993). Method for Analyzing Human Assembly Operations for Use in Automatically Generating Robot Commands. Proc. IEEE International Conference on Robotics and Automation., Pags 695-700.

Torras C. (1989). "Planning for Problem Solving: A Survey". In *Artificial Intelligence In Scientific Computation*, Baltzer AG. Scientific Publishing Co., IMACS Transaction Series, pp.3-14.

Unimation (1980). Unimation inc. "User's guide to VAL: A robot programming and control system", Unimation Inc., Danbury, CT, version 12, June 1.980.

Wang Q., De Schutter J., Witvrouw Wim and Graves S. (1996). Derivation of Compliant Motion Programs Based on Human Demostration. Proc. IEEE International Conference on Robotics and Automation. Minneapolis, Minnesota, 1996. Pags 2616-2621

Whitney D. (1977). Force Feedback Control of Manipulator Fine Motions. *Trans. of ASME, Journal of Dynamic, Systems, Measurement and Control*, June, pp. 91-97.

Xiao J. and R. Volz, (1989). "On Replanning for Assembly Tasks Using Robots in the Presence of Uncertainties", *Proc. of the 1989 IEEE ICRA*, pp. 638-645.

Zalzala, A.M.S. Morris, A.S. (Ed). (1996) "Neural Networks for Robotic Control. Theory and Applications". Ellis Horwood.