

Instituto Tecnológico de San Luis Potosí

Departamento de Eléctrica-Electrónica

“ANALISIS DEL ROBOT PUMA 761”

Presenta:

M. I. Elizabeth Rivera Bravo

San Luis Potosí, S.L.P. a 6 de octubre de 2004

Acrónimos y Abreviaciones.

UASLP	Universidad Autónoma de San Luis Potosí.
CONACyT	Consejo Nacional de Ciencia y Tecnología.
CIEP	Centro de Investigación y Estudios de Posgrado.
IICO	Instituto de Investigación en Comunicaciones Ópticas.
FAI	Fondo de Apoyo a Investigadores.
PUMA	Programmable Universal Manipulator for Assembly (por sus siglas en inglés).
CSM	Camera Space Manipulation (por sus siglas en inglés).
DT3155	Data Translation (por sus siglas en inglés) 3155.
VAL	Victor's Assembly Language (por sus siglas en inglés).
UNIMATION	UNIversal autoMATION
NASA	National Aeronautics and Space Administration (por sus siglas en inglés).
(3-D)	Tridimensional
(2-D)	Bidimensional
MFC	Microsoft Foundation Class (por sus siglas en inglés).
GUI	Graphic User Interfase (por sus siglas en inglés)
PC	Personal Computer (por sus siglas en inglés).
CPU	Process Control Unity (por sus siglas en inglés).
RAM	Random Access Memory (por sus siglas en inglés)
led	Light Emitting Diode (por sus siglas en inglés)
Conf. Est. Tool3 F.P.	Configuración Estimada para el Tool 3, en el Face Plate
mm	milímetro
mseg	milisegundos
gdl	grados de libertad
\forall	para todo
\sim	negación
β	beta
γ	gamma
θ	theta
ϕ	phi
Γ	Gamma
Δ	Delta
Θ	Theta
Σ	Sigma
Ψ	Psi

ÍNDICE GENERAL

INTRODUCCIÓN.	II
ÍNDICE DE FIGURAS.	VII
ÍNDICE DE TABLAS	IX
1. MANIPULADORES MECÁNICOS.	1
1.1. Herramientas.	2
1.1.1. VAL II.	2
1.1.2. Programación Orientada a Objetos	2
2. MANIPULADORES MECÁNICOS.	5
2.1. Modelo Cinemático del Manipulador PUMA 761.	10
2.2. Resultados del algoritmo del modelo cinemático del Robot PUMA 761.	17
2.3. Software para la plataforma de 2 grados de libertad.	18
A. Lenguaje VAL II	23
BIBLIOGRAFÍA	27

INTRODUCCIÓN.

En esta época de grandes avances tecnológicos y científicos los robots han incrementado su popularidad, debido a las grandes ventajas que tiene su uso en un proceso automatizado de producción.

Definimos automatización como el “control automático de manufactura de productos por medio de etapas sucesivas; uso de equipo automático para ahorrar trabajo manual y mental.”

De acuerdo a lo anterior y a la época actual observamos que cada vez más al automatizar, muchos industriales introducen robots en sus procesos productivos. Esto debido a que los robots son capaces de realizar tareas repetitivas de forma más rápida, barata y precisa que los seres humanos. Además los robots pueden desarrollar altos volúmenes de producción y trabajar en ambientes peligrosos. A éste respecto se puede decir que existen empresas que, debido a los artículos que elaboran, tienen que trabajar en ambientes con temperaturas muy elevadas, o en lugares donde la acumulación de gases tóxicos en el ambiente puede resultar peligroso para el hombre, pero no para un robot. Ejemplo de este tipo de industria son todas aquéllas dedicadas a pintar y dar el acabado a electrodomésticos y autos.



Figura 1: Aplicación automatizada usando robots.

Existen también una serie de tareas que pueden resultar difíciles y/o desagradables para los humanos; pero debido a que un robot no puede discernir entre estos términos para él una de estas tareas resulta igual que cualquier otra.

En el año de 1995 funcionaban unos 700,000 robots en el mundo industrializado. Más de 500,000 se empleaban en Japón, unos 120,000 en Europa Occidental y unos 60,000 en Estados Unidos. De aquí podemos deducir que sólo unos pocos se encontraban en México y el resto de los países del Mundo.

La relación en porcentaje de estas cifras no ha cambiado mucho en la actualidad, sólo podemos comentar que el número de robots que se encuentran actualmente en todo nuestro país es mucho mayor que hace 5 años. Se afirma entonces que la robótica es una área que está creciendo en nuestro país y existe un buen porcentaje de la industria mexicana, sobre todo de la industria del automóvil, que se auxilia de robots en sus procesos productivos. Entre estas empresas podemos mencionar algunas que se encuentran en nuestro país como son: Ford en la ciudad de Hermosillo, Chrysler de México en el Estado de Coahuila, Volkswagen en el estado de Puebla, Nissan Mexicana en las ciudades de Cuernavaca y Aguascalientes, General Motors en la ciudad de Silao y Honda en el estado de Jalisco. Entre los trabajos que realizan los robots en estas industrias están: soldadura por puntos, pintura, carga de máquinas, transferencia de piezas y montaje. Particularmente las tareas de pintado son especialmente ideales para los robots debido a que son bastante peligrosas, así como lo son también las tareas de montaje, las cuales están incrementando el uso de robots en la industria debido que exigen una gran precisión que es perfectamente proporcionada por los robots modernos, además de que en estas tareas se emplean sistemas de sensores de bajo costo y potentes computadoras que completan el equipo.



Figura 2: Aplicación automatizada de pintado de autos usando robots.

Existen otras actividades que entrañan gran peligro para las personas como la búsqueda de depósitos minerales submarinos o la exploración de volcanes activos. Estas son especialmente apropiadas para emplear robots. Los robots también pueden explorar planetas distantes. Por ejemplo, la sonda espacial no tripulada Galileo, de la NASA, viajó a Júpiter en 1996 y realizó tareas como la detección del contenido químico de la atmósfera de este lejano planeta.

Existe, como en el caso que ocupa este proyecto, el uso de robots en la investigación, cuyo objetivo es experimentar con ellos otras alternativas que resulten de interés para nuevos posibles usos y aplicaciones en la industria. Los investigadores pueden experimentar con robots sus nuevas teorías antes de implantarlas a algún nuevo programa de control. La institución más importante que ha marcado el rumbo en cuanto a tecnologías e investigaciones en robótica es la NASA. Existe el Programa de Telerobótica Espacial de la NASA, el cual está diseñado para desarrollar capacidades de la telerobótica para la movilidad y manipulación a distancia, uniendo la robótica y las teleoperaciones y creando nuevas tecnologías en telerobótica.

Los requerimientos de tecnología de la robótica espacial pueden ser caracterizados por la necesidad del control manual y automático, tareas no repetitivas, tiempo de espera entre el operador y el manipulador, manipuladores flexibles con dinámicas complejas, nueva locomoción, operaciones en el espacio, y la habilidad para recuperarse de eventos imprevistos.

Otra de las razones por las que se ha incrementado la popularidad de los robots es que son sistemas de tipo antropomórficos, es decir, que tienen una gran semejanza con el hombre ya que tienen un cerebro o un dispositivo que piensa, tienen también sentidos, en su caso sensores, brazos y hasta dedos. Existen algunos como los que se encuentran en el laboratorio de robótica de la Facultad de Ingeniería de la UASLP, que cuentan con dispositivos que hacen las veces de ojos, es decir, cuentan con un sistema de visión.

Para los desarrolladores de robots la última idealización de un robot sería uno que pudiera ver, oír, sentir, hablar, manipular objetos y aún pensar.

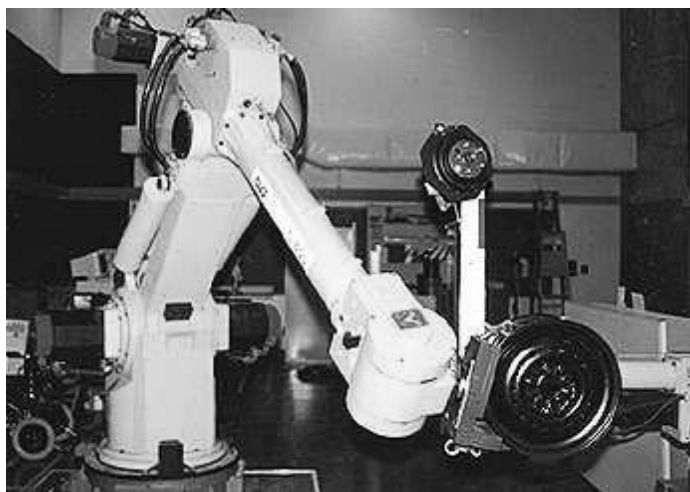


Figura 3: Montaje de ruedas de automóvil usando Manipulación de Espacio de Cámara.

ÍNDICE DE FIGURAS

1.	Aplicación automatizada usando robots.	III
2.	Aplicación automatizada de pintado de autos usando robots.	IV
3.	Montaje de ruedas de automóvil usando Manipulación de Espacio de Cámara.	VI
1.1.	Robot PUMA 761.	1
2.1.	Sistema del Robot Puma 761	6
2.2.	Juntas del robot Puma 761	7
2.3.	Teach Pendant del Robot PUMA 761.	7
2.4.	Dimensiones externas del robot Puma 761	11
2.5.	Representación gráfica del modelo cinemático del robot Puma 761	12
2.6.	Alineamiento inicial del sistema de coordenadas de la herramienta del robot PUMA 761	15
2.7.	Interfaz gráfica del algoritmo del modelo cinemático del robot PUMA 761.	17
2.8.	Unidad mecánica de 2 grados de libertad	19
2.9.	Diagrama de bloques del sistema de motores de pasos modelo MD-2a	20
2.10.	Interfaz para el manejo de la unidad de 2 gdl y el sistema MD-2.	21

ÍNDICE DE CUADROS

2.1. VALORES PARA EL MODELO CINEMÁTICO DEL ROBOT PUMA 761.	10
2.2. RESULTADOS DE VARIAS CONFIGURACIONES PARA EL ROBOT PUMA.	18

CAPÍTULO 1

MANIPULADORES MECÁNICOS.

Se hará referencia a la pieza o herramienta sujeta por un robot manipulador como objeto manipulable y a la pieza de trabajo fija como objeto no manipulable.

La información tridimensional (3-D) es proporcionada por el modelo cinemático del manipulador.

La figura 1.1 muestra un robot PUMA 761, objeto de estudio de este trabajo.

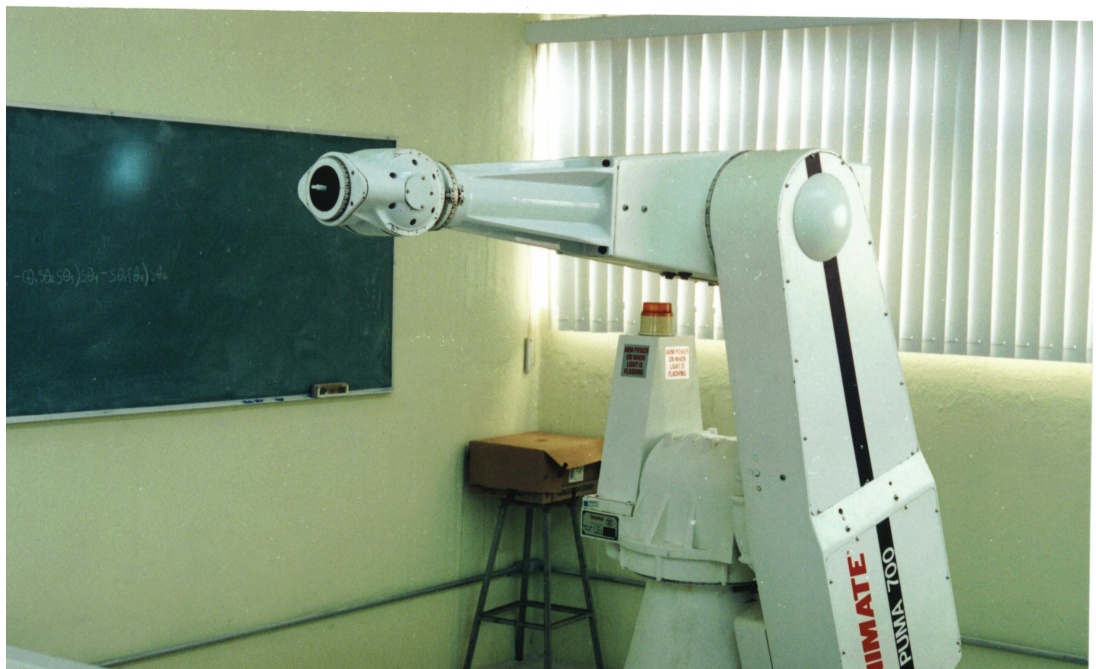


Figura 1.1: Robot PUMA 761.

Al trabajar con este robot, se conocen sus características físicas y se puede desarrollar su modelo cinemático, y por medio de las características de programación de VAL II se pueden corroborar los resultados obtenidos con un algoritmo desarrollado en la computadora personal en el Lenguaje de programación Visual C++.

1.1. Herramientas.

Para realizar el presente trabajo fue necesario recurrir a una serie de herramientas tanto inherentes al robot como a herramientas de software externas.

Para conocer el software del robot PUMA 761 se trabajó durante algún tiempo con el controlador de este robot auxiliados únicamente por los manuales del robot.

1.1.1. VAL II.

Para el robot PUMA 761 el software del sistema se denomina VAL II, el cual fue desarrollado por Unimation, Inc. VAL (Victor's Assembly Lenguaje, después de Victor Scheinman) fue introducido en 1979 por Unimation, Inc. para su serie de robots PUMA, posteriormente fue reemplazado por VAL II y lanzado al mercado en 1984.

El propósito de este apartado es dar una idea general del lenguaje de programación VAL II, de los conceptos e instrucciones básicas del lenguaje y de sus comandos.

Para empezar se mencionará que VAL II es un lenguaje y sistema de control diseñado para los robots industriales de Unimation. Proporciona la capacidad para facilitar la definición de la tarea que va a realizar un robot PUMA, puesto que las tareas se definen mediante los programas escritos por el usuario.

En el caso de este robot, el controlador puede mantener en memoria diversas localizaciones o ubicaciones en el espacio y programas. Estas localizaciones pueden ser usadas indistintamente cuando el usuario lo desee, incluso dentro de programas diferentes. Debido a que el sistema operativo de VAL II y DOS son incompatibles, la edición de programas para este robot sólo puede realizarse directamente en el controlador.

Se mencionarán sólo algunas de las instrucciones y comandos de programación que se utilizan para editar programas utilizando el Lenguaje VAL II, en el Apéndice A. También los modos en los que se pueden realizar los movimientos del robot PUMA.

1.1.2. Programación Orientada a Objetos

Se vive en un mundo de objetos. Estos objetos existen en la naturaleza, en entidades hechas por el hombre, en los negocios, y en los productos que se usan. Ellos pueden ser clasificados, descritos, organizados, combinados, manipulados y creados. Por esto no es sorprendente que se proponga

una visión orientada a objetos para la creación de software de computadora, una abstracción que modela el mundo de forma tal que ayuda a entenderlo y gobernarlo mejor.

Los lenguajes orientados a objetos tienen como ventaja la reutilización (de componentes de software), y llevan a un desarrollo de software más rápido y programas de mejor calidad.

La ingeniería de software orientado a objetos hace hincapié en la reutilización. Por lo tanto, las clases se buscan en una biblioteca (de clases Orientadas a Objetos existentes) antes de construirse. Cuando una clase no puede encontrarse en la biblioteca, el desarrollador de software aplica análisis orientado a objetos y Programación Orientada a Objetos, para crear la clase y los objetos derivados de la clase. La nueva clase se pone en la biblioteca de tal manera que pueda ser reutilizada en el futuro.

Las tecnologías de objetos reflejan una visión natural del mundo. Los objetos están categorizados en clases y las clases en jerarquías. Cada clase contiene un conjunto de atributos que la describen y un conjunto de operaciones que definen su comportamiento. Los objetos modelan casi todo aspecto identificable del dominio del problema: entidades externas, cosas, ocurrencias, roles y estructuras pueden ser representadas como objetos. Es importante destacar que, los objetos y las clases de las que se derivan encapsulan datos y procesos. Las operaciones de proceso son parte del objeto y se inician al pasarle un mensaje al objeto. Una definición de clase, una vez establecida, constituye la base para la reutilización en los niveles de modelado, diseño e implementación.

Existen 3 conceptos muy importantes para Programación Orientada a Objetos: encapsulamiento, herencia y polimorfismo. El encapsulamiento empaqueta datos y las operaciones que manejan estos datos en un objeto simple con denominación. La herencia permite que los atributos y operaciones de una clase sean heredados por todas las subclases y objetos que se instancian de ella. El polimorfismo permite que una cantidad de operaciones diferentes posean el mismo nombre, reduciendo la cantidad de líneas de código necesarias para implementar un sistema y facilita los cambios en caso que se produzcan.

VISUAL C++.

Visual C++ es un Lenguaje Orientado a Objetos, con todas las características arriba mencionadas. Se eligió este lenguaje como base para desarrollar todos los algoritmos de programación que se utilizan para realizar una tarea utilizando la técnica de Manipulación de Espacio de Cámara.

Visual C++ es un lenguaje basado en el sistema operativo Windows desarrollado por la empresa Microsoft. Este lenguaje incluye ya un conjunto bastante amplio de clases reutilizables conocidas como MFC o Microsoft Foundation Class las cuales simplifican la programación en este lenguaje.

Puesto que este entorno trabaja en base a mensajes manejados desde el sistema operativo Windows, la mayoría de los mensajes son estrictamente definidos y aplicados a todos los programas. La ventaja es que el usuario final no se preocupará de la forma en que operan estos mensajes.

El lenguaje Visual C++ nos permite realizar aplicaciones con las características del entorno

de Windows, conocidas como Interfaces de Usuario Gráficas. En éstas aplicaciones el programador inserta las características que el desea que tenga ésa aplicación y su objetivo debe ser ante todo desarrollar una interface que sea comprensible a simple vista para cualquier usuario que no está familiarizado con ésta forma de programación.

El trabajo con este lenguaje ha consistido en desarrollar los algoritmos para: cálculo de la configuración interna del manipulador, manejo de una unidad de dos grados de libertad y otros.

CAPÍTULO 2

MANIPULADORES MECÁNICOS.

Robot PUMA 761.

El conjunto de elementos de que consta el sistema para el robot PUMA 761 son 3 principalmente: el controlador, el brazo de robot y el control manual o Teach Pendant. Estos elementos son mostrados en la figura 2.1.

Conocidas las partes que conforman este sistema, se mencionará la importancia de cada una de ellas.

En el controlador se realizan diversas operaciones, tales como el encendido y apagado del equipo, energizar el brazo del robot, enviar comandos al robot, y varias otras más. En el controlador se encuentran también la unidad de procesamiento central del equipo, memoria de programa, control de los dispositivos de entrada y salida del sistema, además de los servocontroladores de las articulaciones.

Además, el controlador cuenta con una tarjeta de comunicaciones que utiliza el protocolo RS-232 para comunicaciones a través del puerto serial de una computadora personal.

El brazo del robot cuenta con 6 ejes, la figura 2.2 muestra los ejes y describe el movimiento que puede realizar cada uno de ellos, además de su rango de rotación.

Otra parte muy importante del sistema lo constituye el control manual o Teach Pendant¹ el cual puede ser utilizado una vez que es activado el modo TEACH del controlador. Este dispositivo permite mover manualmente al robot a localizaciones deseadas. Entre las funciones que se encuentran en este control está un botón para paro de emergencia en caso necesario, también la activación de los modos en los que se pueden realizar los movimientos del robot PUMA 761 (éstos se explican más ampliamente en el apéndice A) como son el modo JOINT, WORLD y TOOL.

¹conocido también como terminal de enseñanza

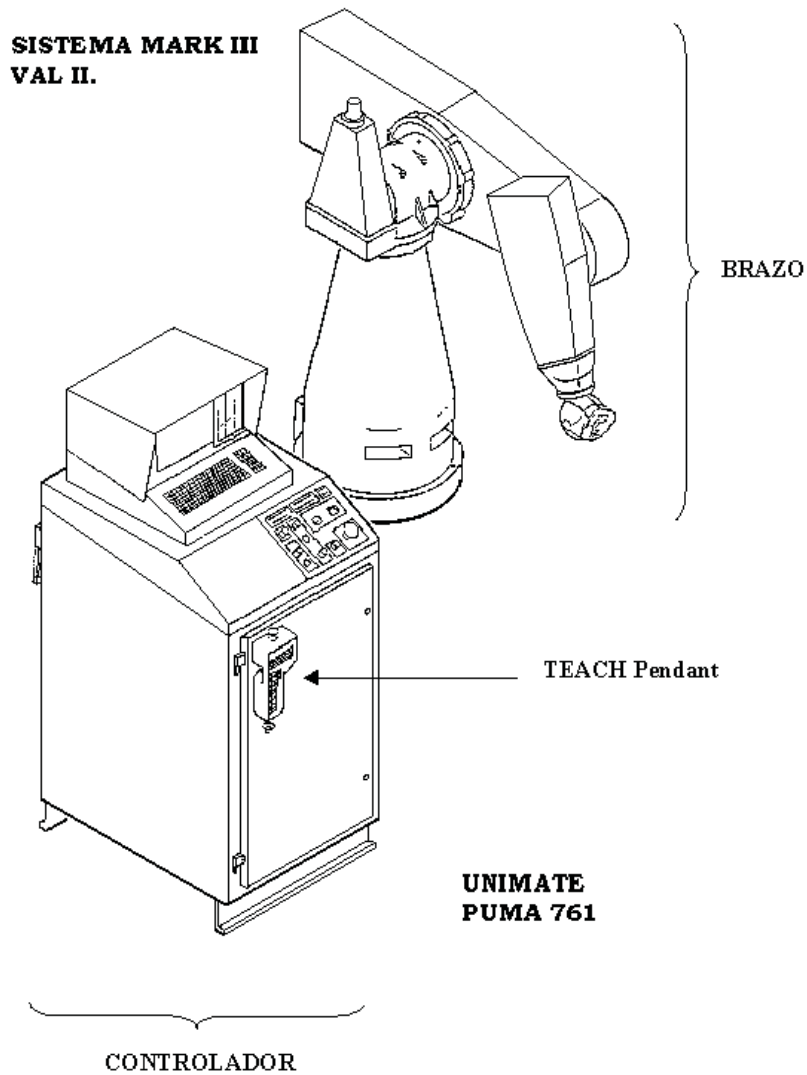


Figura 2.1: Sistema del Robot Puma 761

La figura 2.3 muestra el Teach Pendant del robot PUMA.

Al utilizar este dispositivo en la creación de un programa para desempeñar una tarea, es necesario grabar cada uno de los puntos que componen la trayectoria a lo largo de la cual se debe desplazar el robot. Durante este procedimiento se emplean los modos anteriormente descritos. Al realizar estos pasos se dice que se está enseñando al robot como desempeñar una tarea.

En el control manual también se encuentra un interruptor para indicar la velocidad con la que se moverá el robot, de las que existen 3 opciones en modo Teach que son baja, media y alta. Este interruptor se encuentra en un costado del Teach Pendant.

Otro elemento muy importante es su lenguaje de programación que se denomina VAL II. En el

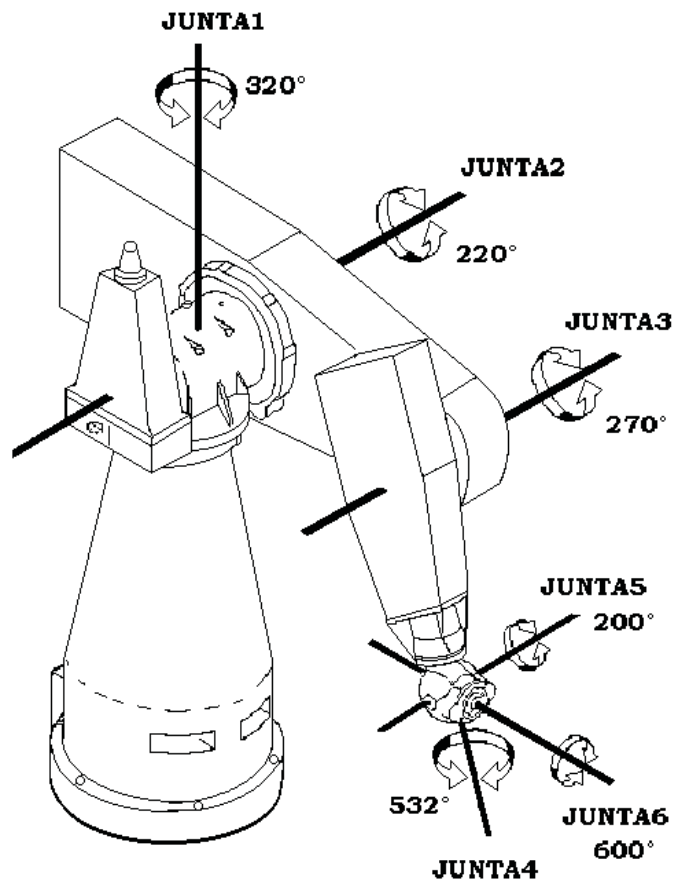


Figura 2.2: Juntas del robot Puma 761

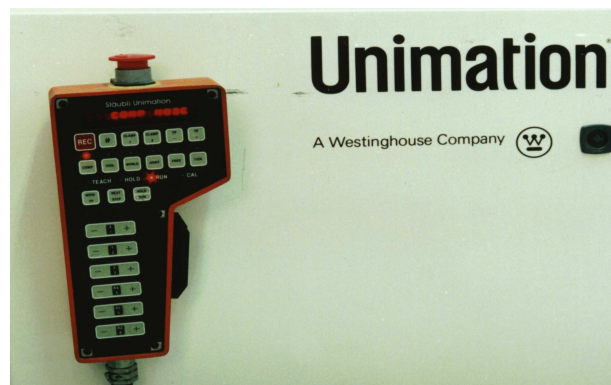


Figura 2.3: Teach Pendant del Robot PUMA 761.

Apéndice A se da una referencia más amplia de él. Este se aplica en el controlador del robot.

Entre las medidas de seguridad que se deben tomar en consideración para este robot se encuentran:

- Asegurarse que el brazo no pueda chocar con otro equipo en cualquier circunstancia.
- Verificar que ninguno de los botones de paro de emergencia se encuentren presionados. Existen 2 botones de paro de emergencia, uno en el Teach Pendant y otro en el controlador.
- Constatar que no existan obstáculos con los que pueda impactarse el robot al realizar sus maniobras.
- Utilizar un botón de paro de emergencia sólo en caso necesario.
- Al dar mantenimiento o realizar algún ajuste al robot es muy importante desenergizar el sistema.
- Realizar correctamente los procedimientos de encendido y apagado del sistema.

Análisis Cinemático Directo.

Un robot puede considerarse como un conjunto de eslabones rígidos conectados en varios puntos. El objetivo del análisis cinemático directo es determinar el efecto acumulativo de todo el conjunto de variables de junta para un manipulador determinado. Dicho de otro modo, el objetivo en este punto es determinar las ecuaciones que describen el comportamiento geométrico del robot, relacionando los desplazamientos lineales o angulares con la posición y orientación del elemento terminal. Los desplazamientos dependen del tipo de juntas; si las juntas son de tipo prismático los desplazamientos son lineales, si las juntas son de tipo de revoluta los desplazamientos son angulares. Este es el caso que ocupa este proyecto, las juntas del robot PUMA 761 son angulares.

En la mayoría de las aplicaciones con robots es imprescindible determinar la posición y orientación del elemento terminal respecto a una referencia fija. Usualmente se considera a ésta fija en la base del robot; pero ésta puede cambiarse a otro punto según se desee o requiera.

Se emplea el concepto de transformación homogénea para describir, respecto a un sistema de referencia fijo, la posición y orientación de cada uno de los eslabones que componen el brazo del robot.

El brazo del robot manipulador está constituido, básicamente, por una secuencia de eslabones rígidos formando una cadena cinemática abierta. La inmensa mayoría de los robots industriales o de investigación responden a una configuración de este tipo.

El movimiento relativo entre dos eslabones adyacentes es causado por el movimiento del par cinemático que conecta a los dos eslabones. Un robot que tiene n grados de libertad que responda a la configuración antes citada, estará constituido por $n+1$ eslabones (incluyendo el eslabón fijo) unidos por n pares cinemáticos. Los eslabones se enumerarán en orden ascendente, comenzando por la base a la que usualmente le corresponderá el número cero, el i -ésimo par conectará a los eslabones $i-1$ e i .

La Notación de Denavit y Hartenberg.

La Notación de Denavit y Hartenberg(1955) es una convención para la selección sistemática de los sistemas coordenados unidos a cada eslabón. En ésta convención cada transformación homogénea \mathbf{A}_i está representada por un producto de 4 transformaciones básicas como sigue:

$$\mathbf{A}_i = \mathbf{Rot}_{z,\theta_i} * \mathbf{Tras}_{z,d_i} * \mathbf{Tras}_{x,a_i} * \mathbf{Rot}_{x,\alpha_i} \quad (2.1)$$

En la cual al substituir cada una de las matrices y realizar el producto se obtiene como resultado:

$$\mathbf{A}_i = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) * \cos(\alpha_i) & \sin(\theta_i) * \sin(\alpha_i) & a_i * \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) * \cos(\alpha_i) & -\cos(\theta_i) * \sin(\alpha_i) & a_i * \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

donde:

- $a_i \longrightarrow$ longitud;
- $\alpha_i \longrightarrow$ giro;
- $d_i \longrightarrow$ desplazamiento de una junta prismática;
- $\theta_i \longrightarrow$ desplazamiento de una junta de revoluta.

La matriz \mathbf{A}_i es función de una sola variable y 3 de las 4 cantidades son constantes para un eslabón determinado.

θ_i es variable para una junta de revoluta y d_i es variable para una junta prismática. En el caso de los robots del laboratorio de la UASLP se consideran variables de tipo θ_i ya que ambos robots tienen sus 6 juntas de tipo de revoluta.

Una vez aplicada ésta convención se tiene la información necesaria para generar cada una de las matrices de transformación homogénea \mathbf{A}_i que dependen sólo de θ_i o sea $\mathbf{A}_i = \mathbf{A}_i(\theta_i)$ donde θ_i es la variable de junta. Se sabe que la matriz homogénea que transforma las coordenadas de un punto desde el marco de referencia n hasta el sistema inercial o es \mathbf{H} , donde esta matriz tiene la forma:

$$\mathbf{H} = \begin{bmatrix} \mathbf{R}_{o/n} & \vec{d}_{o/n} \\ \vec{o} & 1 \end{bmatrix} \quad (2.3)$$

donde:

- $\mathbf{R}_{o/n}$ es una matriz de rotación de $[3 \times 3]$ que contiene la información para encontrar la orientación del sistema en el extremo del manipulador, es decir en el marco de referencia n , con respecto a la base que es el marco de referencia o .
- $\vec{d}_{o/n}$ es un vector de $[3 \times 1]$ que contiene la información para encontrar la posición del sistema en el extremo del manipulador, es decir en el marco de referencia n con respecto a la base, que es el marco de referencia o .

La forma más general de transformación homogénea que se considera puede ser escrita como:

$$\mathbf{T}_{0/n} = \begin{bmatrix} nx & sx & ax & dx \\ ny & sy & ay & dy \\ nz & sz & az & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n & s & a & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

donde:

- $n = [n_x, n_y, n_z]^T$ es un vector que representa la dirección del eje x_n referido al sistema $x_0y_0z_0$
- $s = [s_x, s_y, s_z]^T$ es un vector que representa la dirección del eje y_n referido al sistema $x_0y_0z_0$
- $a = [a_x, a_y, a_z]^T$ es un vector que representa la dirección del eje z_n referido al sistema $x_0y_0z_0$
- $d = [d_x, d_y, d_z]^T$ es un vector que representa al vector dirigido desde el origen del sistema $x_0y_0z_0$ al origen $x_ny_nz_n$ definido en relación al sistema $x_0y_0z_0$

2.1. Modelo Cinemático del Manipulador PUMA 761.

Como se mencionó anteriormente el Manipulador PUMA 761 es un brazo mecánico de configuración antropomórfica. Este manipulador tiene las dimensiones mostradas en la figura 2.4.

En el caso de este robot es perfectamente aplicable la Convención de Denavit y Hartenberg, y la tabla de valores obtenida después de aplicar ésta es la siguiente:

Cuadro 2.1: VALORES PARA EL MODELO CINEMÁTICO DEL ROBOT PUMA 761.

Junta	a_i	d_i	α	θ_i
1	0	0	-90	θ_1^*
2	a_2	0	0	θ_2^*
3	0	d_3	90	θ_3^*
4	0	d_4	-90	θ_4^*
5	0	0	90	θ_5^*
6	0	d_6	0	θ_6^*

donde θ_i^* significa que θ_i es variable.

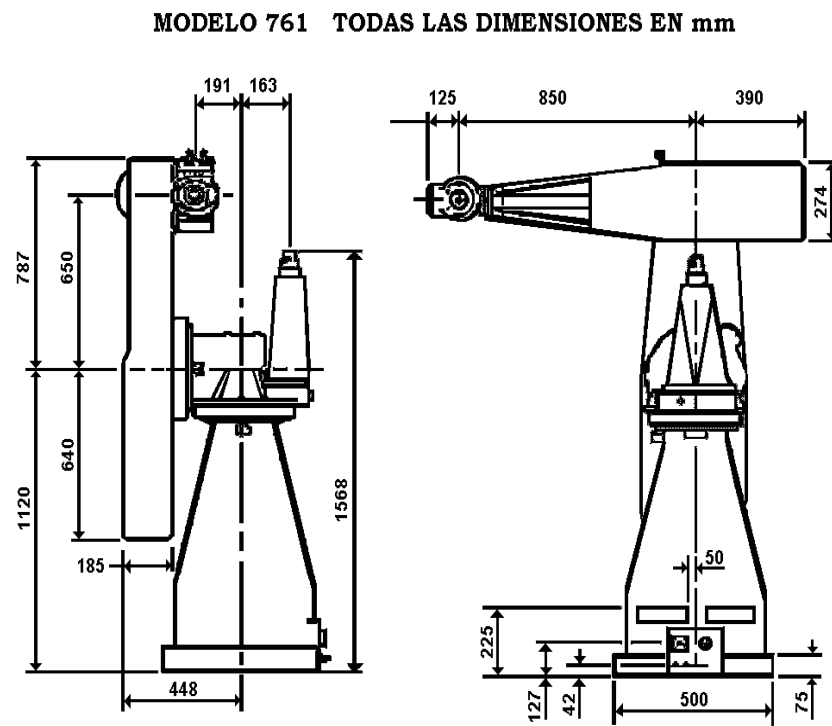


Figura 2.4: Dimensiones externas del robot Puma 761

Y substituyendo los valores de la tabla 2.1 en la matriz A_i se obtiene lo siguiente:

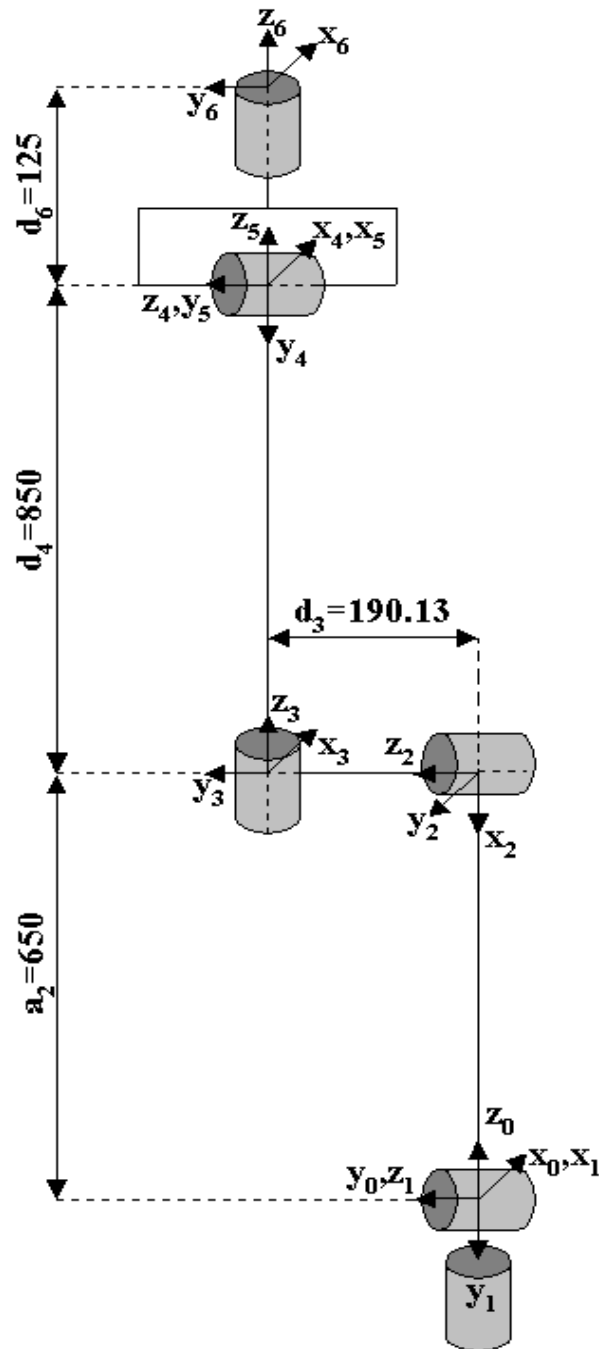


Figura 2.5: Representación gráfica del modelo cinemático del robot Puma 761

$$A_1 = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & \cos(\theta_1) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2 * \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2 * \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} \cos(\theta_3) & 0 & \sin(\theta_3) & 0 \\ \sin(\theta_3) & 0 & -\cos(\theta_3) & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_4 = \begin{bmatrix} \cos(\theta_4) & 0 & -\sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & \cos(\theta_4) & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_5 = \begin{bmatrix} \cos(\theta_5) & 0 & \sin(\theta_5) & 0 \\ \sin(\theta_5) & 0 & -\cos(\theta_5) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_6 = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

donde :

$$a_2 = 650mm; \quad d_3 = 190,13mm; \quad d_4 = 850mm; \quad d_6 = 125mm;$$

Luego se tiene que para este robot:

$$\mathbf{H} = \mathbf{T}_{0/6} = \mathbf{A}_1(\theta_1) * \mathbf{A}_2(\theta_2) * \mathbf{A}_3(\theta_3) * \mathbf{A}_4(\theta_4) * \mathbf{A}_5(\theta_5) * \mathbf{A}_6(\theta_6) \quad (2.5)$$

la matriz \mathbf{H} o $\mathbf{T}_{0/6}$ describe la posición y orientación del elemento terminal con respecto a un sistema de referencia ligado a la base del manipulador y el producto

$$\mathbf{A}_1 * \mathbf{A}_2 * \mathbf{A}_3 * \mathbf{A}_4 * \mathbf{A}_5 * \mathbf{A}_6$$

se conoce como ecuación cinemática del robot PUMA 761. La matriz $\mathbf{T}_{0/6}$ de la ecuación 2.5 se puede representar en forma compacta como:

$$\mathbf{T}_{0/6} = \begin{bmatrix} nx & sx & ax & dx \\ ny & sy & ay & dy \\ nz & sz & az & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Entonces para conocer la posición y orientación de un sistema coordenado en el extremo de la herramienta, o en este caso en la placa colocada al extremo, con respecto al sistema coordenado en

la base del robot², se tiene que :

$$\begin{aligned}
 I_0 &= \mathbf{A}_1 * I_1 \\
 I_1 &= \mathbf{A}_2 * I_2 \\
 I_2 &= \mathbf{A}_3 * I_3 \\
 I_3 &= \mathbf{A}_4 * I_4 \\
 I_4 &= \mathbf{A}_5 * I_5 \\
 I_5 &= \mathbf{A}_6 * I_6
 \end{aligned}$$

Con el objeto de conocer los elementos en la última columna de la matriz $\mathbf{T}_{0/6}$ se considera el vector I_6 como

$$I_6 = [0 \ 0 \ 0 \ 1]^T \quad (2.7)$$

y se procede en la forma siguiente:

$$\begin{aligned}
 I_4 &= \mathbf{A}_5 * \mathbf{A}_6 * I_6 \\
 I_3 &= \mathbf{A}_4 * \mathbf{A}_5 * \mathbf{A}_6 * I_6 \\
 I_2 &= \mathbf{A}_3 * \mathbf{A}_4 * \mathbf{A}_5 * \mathbf{A}_6 * I_6 \\
 I_1 &= \mathbf{A}_2 * \mathbf{A}_3 * \mathbf{A}_4 * \mathbf{A}_5 * \mathbf{A}_6 * I_6 \\
 I_0 &= \mathbf{A}_1 * \mathbf{A}_2 * \mathbf{A}_3 * \mathbf{A}_4 * \mathbf{A}_5 * \mathbf{A}_6 * I_6
 \end{aligned}$$

es decir:

$$I_0 = \mathbf{T}_{0/6} * I_6 \quad (2.8)$$

se sabe que el vector 2.8 es de la forma

$$I_0 = [dx \ dy \ dz \ 1]^T \quad (2.9)$$

donde dx, dy y dz de este vector I_0 corresponden a los valores de las coordenadas x, y, z del sistema, relacionado al posicionamiento del sistema. Estos valores se pueden comprobar en el controlador del robot.

Para resolver el problema de los ángulos de orientación de un robot PUMA, es decir, encontrar la solución O, A, T donde:

- O (Orientación) es el ángulo formado desde el eje y_0 hasta la proyección del eje Z de la herramienta sobre el plano XY , respecto del eje z_0 .
- A (Altitud) es el ángulo formado desde el plano XY hasta el eje Z de la herramienta.

²que en este caso se toma como el sistema coordinado en el origen del sistema

- T (Tool=Herramienta) es el ángulo formado entre Tool Y y un plano paralelo al plano XY .

se consideró que inicialmente el sistema de coordenadas de la herramienta está alineado con el sistema de coordenadas de la base del robot como lo muestra la figura 2.6. Es decir, cuando $O = A = T = 0^\circ$, el eje y_0 negativo coincide con el eje a de la herramienta, en tanto que el eje x_0 positivo coincide con el eje s de la herramienta.

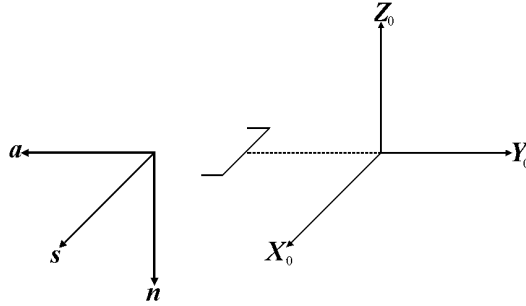


Figura 2.6: Alineamiento inicial del sistema de coordenadas de la herramienta del robot PUMA 761

$$\mathbf{T}_{0/6} = \begin{bmatrix} nx & \mathbf{sx} & ax & dx \\ ny & sy & \mathbf{ay} & dy \\ \mathbf{nz} & sz & az & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

La transformación necesaria que describe la orientación del sistema de coordenadas de la herramienta (n, s, a) , en la ecuación 2.6 respecto del sistema de coordenadas de la base (x_0, y_0, z_0) está dado por

$$\mathbf{Transf} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

De las definiciones de los ángulos arriba mencionadas y la matriz de alineación 2.11, la relación entre la transformación de la herramienta y los ángulos O, A, T está dada por:

$$\mathbf{T}_{0/6} = \mathbf{R}_{z,O} * \mathbf{Transf} * \mathbf{R}_{s,A} * \mathbf{R}_{a,T} \quad (2.12)$$

cada una de las matrices del producto anterior se describe a continuación:

$$\mathbf{R}_{z,O} = \begin{bmatrix} CO & -SO & 0 & 0 \\ SO & CO & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_{s,A} = \begin{bmatrix} CA & 0 & SA & 0 \\ 0 & 1 & 0 & 0 \\ -SA & 0 & CA & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_{a,T} = \begin{bmatrix} CT & -ST & 0 & 0 \\ ST & CT & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

en las ecuaciones anteriores:

$$\begin{aligned} CO &= \cos(O); & SO &= \sin(O) \\ CA &= \cos(A); & SA &= \sin(A) \\ CT &= \cos(T); & ST &= \sin(T); \end{aligned}$$

Después de realizar el producto matricial arriba indicado y de hacer algunas operaciones algebraicas, se llega a la siguiente ecuación para obtener T :

$$nz(ST) + sz(CT) = 0$$

aquí se puede despejar para llegar a

$$T = \arctan\left(\frac{sz}{-nz}\right) \quad (2.13)$$

Luego para obtener la representación de A :

$$SA = -az; \quad CA = -nz(CT) + sz(ST) \quad (2.14)$$

en las ecuaciones anteriores se despeja y se obtiene A como sigue:

$$A = \arctan\left(\frac{-az}{-nz(CT) + sz(ST)}\right) \quad (2.15)$$

Para obtener el ángulo O se cuenta con las siguientes ecuaciones:

$$SO = ny(ST) + sy(CT); \quad CO = nx(ST) + sx(CT) \quad (2.16)$$

se despeja la variable O para llegar finalmente a:

$$O = \arctan\left(\frac{ny(ST) + sy(CT)}{nx(ST) + sx(CT)}\right) \quad (2.17)$$

2.2. Resultados del algoritmo del modelo cinemático del Robot PUMA 761.

Después de obtenido el modelo cinemático del Robot PUMA 761 se creó un programa en Visual C++ para probar éstas ecuaciones. Una vez desarrollado se dieron como parámetros de entrada los valores de las juntas 1 a 6, del programa se tiene como salida los valores de X , Y , Z , O , A , T , éstos valores se compararon con los obtenidos con el comando WHERE (vea el Apéndice A para referencia). La figura 2.7 muestra la interfaz gráfica desarrollada en visual C++ para probar el modelo cinemático anteriormente explicado.

CINEMATICA DIRECTA	
JOINT	
J1	0
J2	0
J3	0
J4	0
J5	0
J6	0
WORLD	
X	650
Y	190.13
Z	975
O	90
A	-90
T	0

Figura 2.7: Interfaz gráfica del algoritmo del modelo cinemático del robot PUMA 761.

A continuación se muestran algunas localizaciones con las que se probó el programa. Se observa de la diferencia entre los valores calculados y medidos que el algoritmo funciona satisfactoriamente. Se observa también que en algunos casos la diferencia no es exactamente cero, esto es debido a que el controlador y el algoritmo no tienen especificado dar la respuesta con el mismo número de cifras decimales. Otra causa puede ser que como los valores para el algoritmo se especifican en radianes, puede haber una pequeña pérdida de decimales en el proceso de conversión de grados a radianes y viceversa.

Cuadro 2.2: RESULTADOS DE VARIAS CONFIGURACIONES PARA EL ROBOT PUMA.

Junta	Desp.°	V.	Calculado	V.	Medido	Diferencia
J1 =	0.0000	X =	650.0000	Xm =	650.0000	Dif = 0.0000mm
J2 =	0.0000	Y =	190.1300	Ym =	190.1300	Dif = 0.0000mm
J3 =	0.0000	Z =	975.0000	Zm =	975.0000	Dif = 0.0000mm
J4 =	0.0000	O =	90.0000	Om =	90.0000	Dif = 0.0000°
J5 =	0.0000	A =	-90.0000	Am =	-90.0000	Dif = 0.0000°
J6 =	0.0000	T =	0.0000	Tm =	0.0000	Dif = 0.0000°
J1 =	90.0000	X =	-190.1300	Xm =	-190.1300	Dif = 0.0000mm
J2 =	-45.0000	Y =	459.6194	Ym =	459.6300	Dif = -0.0106mm
J3 =	45.0000	Z =	1434.6194	Zm =	1434.6300	Dif = -0.0106mm
J4 =	0.0000	O =	-180.0000	Om =	-180.0000	Dif = 0.0000°
J5 =	0.0000	A =	-90.0000	Am =	-90.0000	Dif = 0.0000°
J6 =	0.0000	T =	0.0000	Tm =	0.0000	Dif = 0.0000°
J1 =	0.0000	X =	650.1725	Xm =	650.1900	Dif = -0.0175mm
J2 =	-105.1780	Y =	65.1300	Ym =	65.1300	Dif = 0.0000mm
J3 =	179.9890	Z =	850.0198	Zm =	850.0000	Dif = 0.0198mm
J4 =	-89.9900	O =	0.0219	Om =	0.0220	Dif = 0.0000°
J5 =	89.9800	A =	0.0044	Am =	0.0000	Dif = 0.0044°
J6 =	74.8100	T =	-0.0010	Tm =	-0.0050	Dif = -0.0040°
J1 =	7.7620	X =	896.4738	Xm =	896.5000	Dif = -0.0262mm
J2 =	-93.6090	Y =	319.6448	Ym =	319.6300	Dif = 0.0148mm
J3 =	179.1980	Z =	715.9780	Zm =	715.9400	Dif = 0.0380mm
J4 =	35.5300	O =	100.2886	Om =	100.2890	Dif = 0.0000°
J5 =	4.3500	A =	-0.8678	Am =	-0.8620	Dif = -0.0058°
J6 =	-3.8700	T =	31.5436	Tm =	31.5470	Dif = -0.0034°
J1 =	8.2670	X =	320.6730	Xm =	320.6900	Dif = -0.0170mm
J2 =	-51.2950	Y =	214.3050	Ym =	214.3100	Dif = -0.0050mm
J3 =	44.3130	Z =	1465.0200	Zm =	1465.0310	Dif = -0.0110mm
J4 =	-22.4200	O =	70.0498	Om =	70.0430	Dif = 0.0068°
J5 =	30.4500	A =	-65.8708	Am =	-65.8850	Dif = 0.0142°
J6 =	68.5200	T =	75.0319	Tm =	75.0530	Dif = -0.0211°

2.3. Software para la plataforma de 2 grados de libertad.

Características del sistema.

Esta unidad mecánica tiene como objetivo principal sujetar 2 unidades láser. Una de ellas que tiene un sólo punto láser que será la que ayude a obtener los puntos sobre una superficie para obtener sus coordenadas en las 2 cámaras con que se cuenta. Estos puntos son los utilizados para realizar el mapeo.

Las características que tiene esta unidad le permiten abarcar un gran espacio de trabajo gracias a que es posible localizar al apuntador muy cerca del punto donde se intersecan los ejes de giro. A grandes rasgos, la plataforma está construida por una base en forma cónica que sujeta a un motor de pasos, que se encuentra en la parte inferior del sistema. Dicho motor suministra el torque necesario a un reductor de múltiples engranes rectos. Este reductor se encuentra en la parte superior con respecto al plano horizontal. A la salida de este reductor se encuentra unido firmemente, mediante una estructura, otro arreglo motor-reductor dispuesto en forma horizontal, es decir, que la base del cilindro es paralela al plano vertical mostrado, tal como se muestra en la figura 2.8. Finalmente, en la salida de este segundo reductor se sujeta el apuntador láser.



Figura 2.8: Unidad mecánica de 2 grados de libertad

La mayoría de las piezas está hecha en aluminio entre ellas: la carcasa, las tapas de ambos motores, las tapas de ambos reductores, las tapas que dan ajuste a los baleros cónicos, las tapas que se montan en los retenes y la unión U.

Otras piezas están hechas en acero inoxidable como los ejes de los motores, las camisas de los ejes, los coples del tipo unión universal entre los ejes de los motores y los ejes de los reductores, así como entre los ejes de los reductores y los de los motores de pasos.

Toda la tornillería es de acero negro al carbón de cabeza Allen, las juntas son de neopreno y la aceitera es de bronce. Los reductores de engranes poseen una relación 213 a 1.

La plataforma trabaja con motores de pasos modelo MD-2a que fueron comprados a la compañía Arrick Robotics. En la figura 2.8 se observan, a un costado de la unidad de 2 grados de libertad, el dispositivo que alimenta y maneja los motores, y en la parte superior se puede apreciar uno de los motores. El otro motor no se aprecia debido a que se encuentra dentro de la unidad en la parte inferior del mecanismo.

Los motores se activan a través del puerto paralelo de la PC. Puesto que el manejo de los puertos es un poco diferente para Windows 95 que para Windows NT se utiliza una librería especial que permite realizar esta operación. La figura 2.9 muestra la forma en que están conectados al sistema.

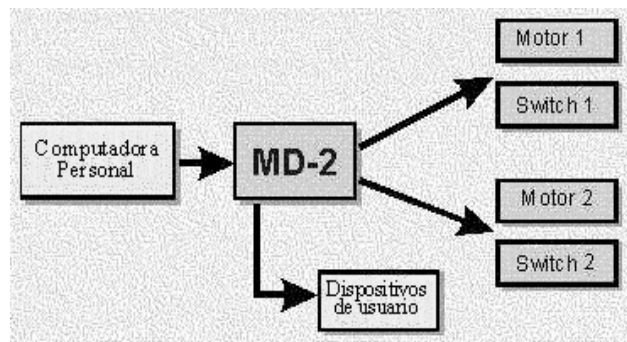


Figura 2.9: Diagrama de bloques del sistema de motores de pasos modelo MD-2a

Se logró manejar exitosamente ambos motores a través del puerto paralelo de la PC, y siguiendo las instrucciones del fabricante de este equipo se logró un completo control de las funciones de los motores a través del puerto paralelo.

Se realizó un programa en ambiente de Windows que permite fácilmente a un usuario trabajar con esta plataforma.

Se asignó el motor 2 para mover el láser a izquierda o derecha sobre la superficie de trabajo y el otro motor para mover el láser hacia arriba o hacia abajo.

Pasos a seguir cuando se usa la plataforma de 2 gdl :

1. Inicializar el puerto paralelo.
2. Inicializar las fases de los motores.

Los pasos anteriores se realizan automáticamente al abrir la aplicación.

3. Al abrir la aplicación oprimir el botón marcado como **Abrir MD-2**. Para activar el sistema de motores MD-2A.
4. Establecer un retardo en milisegundos para cada paso de los motores. El mínimo retardo es de 0.1 mseg ya que cada motor puede avanzar hasta $10000 \frac{\text{pasos}}{\text{seg}}$

5. Establecer un valor en grados de avance, este valor se tomará para ambos motores y se visualizará el número de pasos necesarios para alcanzar ese desplazamiento.
6. Realizar el desplazamiento en el sentido deseado.

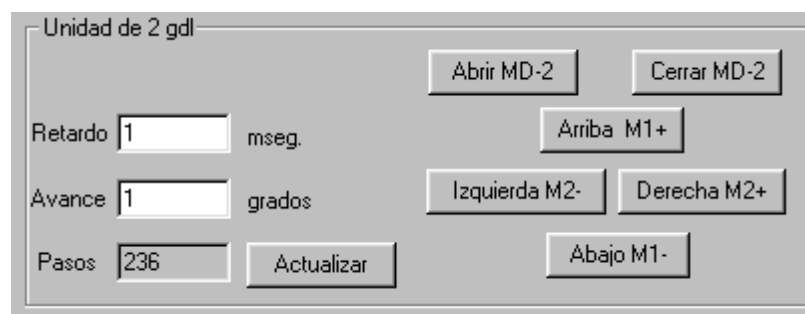


Figura 2.10: Interfaz para el manejo de la unidad de 2 gdl y el sistema MD-2.

APÉNDICE A

Lenguaje VAL II

El propósito de este apéndice no es proporcionar una descripción completa del lenguaje de programación de VAL II, sino más bien, informar acerca de algunos de los conceptos, instrucciones básicas y comandos de este Lenguaje. Para mayor información consultar el Manual de Programación de VAL II.

INSTRUCCIONES Y COMANDOS DEL ROBOT PUMA 761

COMANDOS

- HERE Localización

Este comando asigna el valor de la variable de localización igual a la posición actual del robot.

- WHERE o w

Despliega el valor de la posición actual en coordenadas cartesianas y determina el valor de los ángulos de cada articulación.

- TEACH Localización

Este comando se utiliza para registrar series de localizaciones o ubicaciones utilizando el botón REC del TEACH Pendant. El Teach pendant de este robot es el mostrado en la figura [2.3](#).

Ejemplo:

TEACH Pos[]

Por cada vez que se presione REC, se le asignará a esa posición el nombre Pos[0], Pos[1],...,Pos[n].

- DO Instrucción del programa

Ejecuta la instrucción especificada como si se tratara de la ejecución de un programa línea por línea. Si no se especifica ninguna instrucción, la última instrucción ejecutada por DO es ejecutada de nuevo.

Ejemplo: DO MOVE Ub1

El brazo se moverá a la posición indicada como Ub1.

- FREE

Despliega el porcentaje de memoria disponible actualmente en el sistema.

- DIRECTORY

Despliega los nombres de todos los programas del usuario residentes en la memoria del sistema actualmente.

- LISTL

Despliega los nombres de todas las variable de localizaciones guardadas en la memoria del sistema con sus respectivos valores. Si el usuario especifica el nombre de alguna localización, el controlador desplegará el valor de cada una de las juntas en la localización especificada.

- LISTP Programa

Despliega un listado con todas las instrucciones del programa especificado.

- EXECUTE [Programa],[Ciclo],[Paso]

Ejecuta el programa especificado por el usuario. El programa se ejecutará una sola vez si el argumento Ciclo es omitido o se le asigna un valor de 0 o 1, en caso de que tome un valor de (-1), se ejecutará continuamente hasta que el usuario lo detenga. Si se desea ejecutar el programa desde un renglón en especial, se asigna el valor de éste en el argumento Paso.

Ejemplo:

EXECUTE DEMO,-1

Ejecutará el programa DEMO continuamente hasta que el usuario lo detenga con el comando HALT o con la instrucción abort.

- SPEED Valor

Determina la velocidad de los siguientes movimientos que realice el robot. El argumento valor tendrá un rango de valores entre 0.39(muy lento) hasta 12800(muy rápido), donde 100 es considerado como velocidad normal. Es importante mencionar que al inicializar el equipo, la velocidad que debe especificar el usuario debe ser 30, o sea que el robot trabajará al 30 % de su velocidad normal.

Existen algunos comandos que se utilizan para la edición de programas como son:

- EDIT Programa

Como su nombre lo indica nos sirve para editar un programa.

- E(End)

Termina con el modo de edición y regresa al modo Terminal del controlador.

- I(Insert)

Permite insertar una nueva línea a partir de la línea actual.

Existen muchos otros comandos, por el momento sólo mencionaremos los anteriores por considerarlos importantes y muy utilizados durante el presente trabajo de tesis.

INSTRUCCIONES DE PROGRAMACIÓN

- RIGHTY o LEFTY

Cambia la configuración del robot de manera que las primeras 3 articulaciones del robot emulan un brazo humano derecho o izquierdo según se especifique. La figura 1.1 muestra al robot PUMA 761 en configuración de hombro izquierdo.

- ABOVE o BELOW

Cambia la configuración del robot de manera que el codo apunte hacia arriba o hacia abajo según lo especifique el usuario. La figura 1.1 muestra al robot PUMA 761 en configuración de codo arriba.

- MOVE Localización

Mueve el brazo de la posición actual a la especificada por el argumento, tal desplazamiento es considerado como un movimiento de interpolación de juntas.

- ALIGN

La herramienta es rotada hasta lograr una alineación del eje Z de la herramienta con el eje más cercano del sistema de coordenadas WORLD del robot.

- **DEPART** Distancia en mm

Desplaza la herramienta desde la posición actual colinealmente sobre su eje z, por interpolación de ejes, la distancia especificada en el argumento. Esta distancia puede ser positiva o negativa, si es negativa el desplazamiento será hacia adelante, y si es positiva el desplazamiento será hacia atrás. Esta instrucción debe usarse con cautela debido a que existen localizaciones que no pueden ser alcanzadas por el robot debido a sus características mecánicas, además éstas localizaciones pueden llevar al robot a posiciones singulares que son indeseadas, éstas hacen que el robot entre en una condición de fallo del sistema.

- **DELAY** tiempo sec

Pone al programa en estado de espera durante el tiempo especificado, el argumento tiempo puede estar en el rango de valores entre 0.01 y 900 , como se indica arriba estos valores se especifican en segundos.

- **READY**

Lleva al robot a la posición de HOME. Podemos mencionar que en esta posición los valores de cada una de las juntas son [0,-90,90,0,0,0].

- **CALL** Programa

Esta instrucción permite llamar otro u otros programas que se consideran como subrutinas del programa que lo(s) está llamando.

Al igual que con los comandos, existen muchas más instrucciones de programación en el Lenguaje VAL II, por el momento solo mencionaremos las anteriores.

MODOS DE MOVIMIENTO PARA EL ROBOT PUMA.

A continuación se presentan los modos en los que se pueden realizar los movimientos del robot **PUMA**, cabe aclarar que estos modos se ejecutan desde el Teach Pendant o Control Manual de este robot.

- **JOINT:** En este modo se realizan los movimientos de cada junta del robot por separado.
- **WORLD:** En este modo el brazo se mueve para que la herramienta se dirija hacia un punto desplazándose en los diversos planos (X,Y,Z) tomando como base el sistema cartesiano del robot cuyo punto (0,0,0) se localiza en la base del robot.
- **TOOL:** En este modo el robot se mueve hacia algún punto deseado, cambiando su sistema de coordenadas al centro de la herramienta y ubicando al eje Z como el centro de ésta.

BIBLIOGRAFÍA

- [1] W. Z. Chen, U. A. Korde, and Steven B. Skaar. Position control experiments using vision. *The International Journal of Robotics Research*, 13(3):199–208, 1994.
- [2] Robert B. Davies. NEWMAT09: A Matrix Library in C++. Disponible gratis en internet: <http://webnz.com/robert>, 1997.
- [3] K. S. Fu, R. C. González, and C. S. Lee. *Robótica – Control, detección, visión e inteligencia*. McGraw-Hill, 1988.
- [4] Emilio J. González Galván. Introducción al Curso de Robótica y Visión por Computadora. Notas del curso impartido en el CIEP-FI, 1999.
- [5] Emilio J. González Galván, Steven B. Skaar, and Michael J. Seelinger. Efficient camera-space target disposition in a matrix of moments structure using camera-space manipulation. *The International Journal of Robotics Research*, 18(8):809–818, 1999.
- [6] John L. Junkins. *An Introduction to Optimal Estimation of Dynamical Systems*. Alphen Aan Den Rijn, The Netherlands: Sijthoff & Noordhoff, 1978. The University of Virginia.
- [7] David J. Kruglinski, George Shepherd, and Scot Wingo. *Programación Avanzada con Microsoft Visual C++*. Serie de Programación Microsoft. Microsoft Press – McGraw Hill, 1999.
- [8] Leslie Lamport. *L^AT_EX₂ ϵ – A document preparation system*. Addison Wesley, 1994.
- [9] Roger N. Nagel Mikel P. Groover, Mitchell Weiss and Nicholas G. Odrey. *Industrial Robotics Technology, programming and Applications*. Ed. McGraw Hill–International Edition.
- [10] Pierre Alain Muller. *Modelado de Objetos con UML*. Ed. Gestión 2000, 1 edition, 1997.

- [11] Scott Muller. *Manual de Actualización y reparación de PC's*. Ed. Prentice Hall, Ediciones Que, 1997.
- [12] William H. Press, William T. Vetterling, Saul A. Teukolsky, and Brian P. Flannery. *Numerical Recipes in C – The Art of Scientific Computing*. Ed. Cambridge University Press, 2 edition.
- [13] Steven B. Skaar and Carl F. Ruoff. *Teleoperation and Robotics. Chap. 9: Versatile and Precise Vision-Based Manipulation*. American Institute of Aeronautics and Astronautics, Inc.
- [14] Chuck Sphar. *Aprenda Microsoft Visual C++ Ya*. Microsoft Press – McGraw Hill, 1999.
- [15] UNIMATION. *Manuales del Robot PUMA 761*.
- [16] FANUC ROBOTICS y LINCOLN. *Manuales del Robot FANUC ArcMATE 100 i y soldadora LINCOLN*.
- [17] Beck Zaratian. *Microsoft Visual C++, Manual del Programador*. Microsoft Press – McGraw Hill, 1999.



-

Este trabajo fue realizado por M.I. Elizabeth Rivera Bravo.