

Un CNC Modular, Multieje, Apto para el Comando de Robots y Sistemas Especiales

† ‡ Carlos Candiani, *Member IEEE*, ccandiani@ieee.org, † ‡ Juan Luzuriaga, jluzuriaga@scdt.fr.utn.edu.ar,
 ‡ Daniel Petrone, servicio@contraut.com.ar
 † CUDAR, Facultad Regional Córdoba, UTN, ‡ ContraAUT, Empresa de Ingeniería.

Resumen - Se describe el desarrollo de un comando numérico (CNC) de tipo modular apto para controlar sistemas de hasta seis ejes. (Máquinas herramienta, transfers, robots, etc.) El mismo se basa en módulos idénticos e independientes para cada eje que se desea comandar. Estos módulos, integrados sobre una placa única, contienen todos los elementos requeridos para el cierre del lazo correspondiente, su propia inteligencia abordo, así como un importante número de E/S digitales para realizar las funciones PLC relacionadas con el eje comandado. Un módulo extra –idéntico– es utilizado para la coordinación general, centralización de funciones e interfaz con un periférico inteligente.

Palabras clave: Control numérico, control digital, máquinas herramienta, robots industriales.

I. INTRODUCCIÓN

El comando de máquinas especiales, sistemas robóticos y equipos productivos similares demandan hoy la utilización de controles numéricos con características especiales que muchas veces no pueden ser cubiertas por los equipos comerciales disponibles (o su utilización debe ser complicadamente adaptada) En muchos de estos casos –adicionalmente– los requerimientos hacia el sistema de comando son simples, no requiriendo generación de curvas de orden superior, bastando con la generación de trayectorias rectilíneas, es decir capacidad de realizar interpolaciones lineales, característica a la que, siempre, debe adicionarse la presencia de un poderoso controlador digital tipo **PLC** para la generación de las funciones auxiliares y el mando de los periféricos necesarios. Se ha diseñado un comando numérico especialmente apto para este cometido al que se ha denominado AlfaTROL 49 y se ha dotado de características tales como: *Modularidad*: El número de ejes comandados se determina con el número de módulos idénticos incluidos; *Discriminación*: Es parametrizable; *Comando de elementos auxiliares*: Cada eje posee la posibilidad de controlar un husillo u otro elemento similar vía señal analógica. Cada eje posee la posibilidad de ser comandado en forma manual a través de un volante digital, uno –independiente– por cada eje o uno único conmutable. Como se indicó, un módulo adicional –también idéntico a los controladores de ejes– coordina el conjunto y centraliza la función PLC. De esta forma el número de módulos –placas– totales en una configuración dada resulta:

$$NM = NE + 1, \quad (1)$$

donde **NM** es el número de módulos de control y **NE** es el número de ejes controlados.

II. ELECTRÓNICA: DESCRIPCIÓN

A- Módulos de Control

Cada *Módulo de Control* (denominados **MC_j**, con $j=0, 1...NE$) (Fig. 1) (generalmente **MC₁** comanda el Eje X; **MC₂** el Eje Y... etc.) se estructura alrededor de una única placa, la que posee hasta tres placas auxiliares (*submódulos*) acoplables a la misma. El primero de éstos contiene el procesador (Submódulo SMP) (Fig. 2), motivo por el cual su presencia resulta imprescindible. La segunda contiene los elementos necesarios para el acondicionamiento de las señales de un encoder o regla óptica, así como su correspondiente decodificador de direcciones. Este submódulo de realimentación (denominado SMRD), *no es requerido* en el caso que se trate del módulo central de control del CNC (**MC₀**). El tercer submódulo contiene un acondicionador de señales similar al anterior, más simple, y está destinado a servir de interfaz con un volante digital (SMV). Esto permite que *cada eje controlado posea* –si se lo desea– un volante propio para el accionamiento manual del mismo. Mediante funciones implementadas en el PLC interno, esta función puede ser realizada con sólo uno de dichos volantes, siendo entonces éste utilizado por el eje que el operador designe. Este tercer submódulo se requiere, exclusivamente, cuando se desea un volante en el eje considerado.

Siguiendo los requerimientos del procesador utilizado (Texas 430) se adoptó una lógica de 3,3V.

B- El CNC

Como se indicó, **NM** módulos de control componen el CNC. De estos, $NE = NM - 1$, corresponden a ejes controlados (**MC₁...MC_{NE}**), debiendo adicionarse el Módulo Centralizador (**MC₀**). Todos ellos deben poseer el *submódulo procesador* (SMP). Los que comandan ejes deben poseer, ade-

más, el *submódulo de realimentación* (SMRD) y, como se indicó, algunos (o todos) *pueden* poseer el *submódulo de volante* (SMV). En caso de una máquina muy sencilla (Vg.: Una templadora, un eje comandado) el CNC sólo requiere dos módulos principales (MC_0 y MC_1), sus procesadores SMP (2x) y, en el MC_1 un SMRD para la realimentación, eventualmente incorporado al mismo, un SMV para generar movimientos manuales vía volante. En el caso de un torno, o cualquier sistema 2D, a lo anterior debe sumarse un MC más (MC_2) con su procesador y SMRD. Para sistemas del tipo de 3D, uno más y así siguiendo hasta alcanzar seis ejes controlados (estructura tipo robot) que demandará siete módulos MC (MC_0 hasta MC_6) sus procesadores (7x), seis de ellos con SMR, y –eventualmente- uno o más SMVs.

Dada la velocidad de intercambio de información entre módulos, el volumen de la misma y otras consideraciones de tiempos, es que se ha fijado este número como la cantidad máxima de ejes comandables por el sistema, tal como se halla concebido al día de la fecha.

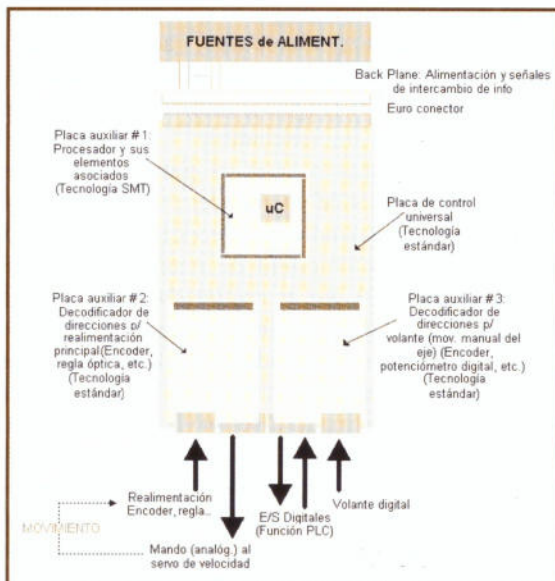


Fig. 1: Un MC_j con su SMP, SMRD y SMV asociados

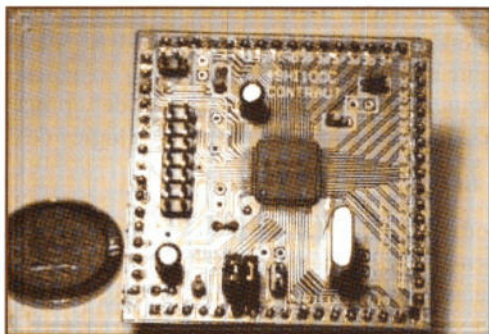


Fig. 2: Placa Auxiliar #1 – Sub. Módulo Procesador (SMP) (Placa con tecnología de montaje superficial)

III. PROGRAMACIÓN INTERNA: PRINCIPIO DE FUNCIONAMIENTO: CNC

A- Movimientos comandados

Dado un punto móvil tomado como referencia (ejemplo: el extremo de la herramienta en un torno o fresadora), el CNC que comanda la máquina, debe ser capaz de desplazarlo siguiendo una curva cualquiera *deseada y programable*, con *velocidad compuesta constante y también programable*.

La curva a generar puede ser **coplanar** (caso 2D) o **espacial** (caso 3D o más ejes) y la velocidad debe ser tal que, *componiendo vectorialmente las velocidades individuales de cada eje, el vector resultante debe ser –en todo momento- de módulo constante*.

La velocidad instantánea *en cada eje* controlado indica la pendiente de la curva generada sobre dicho eje en ese instante. Ésta puede ser *constante* en caso de seguir una forma rectilínea o *variable* –siguiendo leyes aleatorias- dependiendo de las formas que se deseen.

B- Programación

Un *programa de partes* (“Part Program”) consiste en una serie de instrucciones y cotas que –por una parte- *definen la curva a recorrer* y por otra impone las *condiciones* en que deberá ser recorrida (velocidades, acciones extra a ejecutar en puntos o zonas definidas de la misma¹, etc.). De aquí en más, y hasta tanto se traten las acciones del PLC integrado, se prescindirá de las acciones extra y se concentrará la atención en la generación de las curvas a recorrer.

El equipo actualmente admite al menos- tres formas de ingresar los programas de partes a ejecutar:

- I. Planilla de programación
- II. Teaching
- III. CAMs y Programación Automática

Se dice que se tiene un programa de partes *listo para ejecutar*, ingresado en cualquiera de las tres formas indicadas, cuando *se dispone en el CNC de una serie ordenada de puntos (en dos, tres, o más dimensiones o ejes) definidos por sus coordenadas, los que unidos por una poligonal definen una curva que difiere de la deseada en valores acotados y dentro de tolerancias parametrizable*.

- I. La **Planilla de Programación** sustituye en este equipo a la programación tradicional, realizada en general acorde a la norma DIN 66025. En este caso el equipo presenta una planilla que el

¹ Entre las acciones extra a programar se cuentan: Gama y velocidad del/los husillos, herramienta/s a utilizar, refrigerante, extractor de virutas Si/No, avance de barra, apertura/cierre de mordazas, etc., así como todas las demás funciones “M” acorde con DIN66025.

operador debe llenar con las cotas, velocidades en cada segmento y demás condiciones de marcha. Los puntos programados serán los vértices de la poligonal a describir. El ajuste entre una eventual curva continua y la posición de los puntos programados corre a cargo exclusivo del programador.

- II. En el modo **Teaching**, el operador debe recorrer *una vez*, utilizando los comandos manuales de movimientos, la curva que se desea describir, marcando los puntos que el programa tomará como pivots de la poligonal.
- III. **Modos Automáticos de Programación y CAMs.** Actualmente la mayoría de las máquinas CNC son programadas en forma automática a través de sistemas CAM o similares, los que generan caminos y movimientos acordes a cualquier perfil (en un plano -2D- o en el espacio -3D-.) Estos sistemas admiten, a través de un sistema CAD integrado, la definición de las curvas a generar y entregan como salida -en su forma más común de funcionamiento- una serie de puntos definidos por sus coordenadas, los que unidos por una poligonal generada en el CNC, definen una curva que difiere de la deseada en valores acotados. El equipo desarrollado acepta también este tipo de programación.

Finalmente, se halla en proceso de desarrollo un módulo de programación que admite -además- la carga de funciones estandarizadas para la generación de recorridos circulares (funciones G02 y G03 acorde a **DIN 66025**) u otros no rectos (contornos parabólicos, elípticos o similares). Este módulo halla, define y posiciona las ubicaciones de puntos intermedios que pueden ser utilizados como *pivots* en las mismas condiciones anteriores: La poligonal que los une se halla dentro de una banda (en dos o tres dimensiones) que incluye la curva teórica deseada y cuya amplitud es acotada y parametrizable.

C- Interpolación [1], [2].

El equipo dispone de un *interpolador lineal* que coordina la simultánea generación de pulsos en todos los ejes permitiendo la unión de los puntos pivots programados con segmentos de rectas en las condiciones indicadas antes, es decir generando recorridos rectilíneos con velocidad compuesta constante.

D- Puntos programados – Puntos Insertados

El contorno a recorrer queda determinado por los mencionados **Puntos Pívor**, o **Puntos Programados** los que han sido cargados al programa *listo para ejecutar* por cualquiera de los métodos menciona-

dos antes.

Como se mencionó antes, la principal característica de dichos puntos es que **unidos por segmentos rectilíneos, el contorno generado resulta aceptable** (Fig. 3).

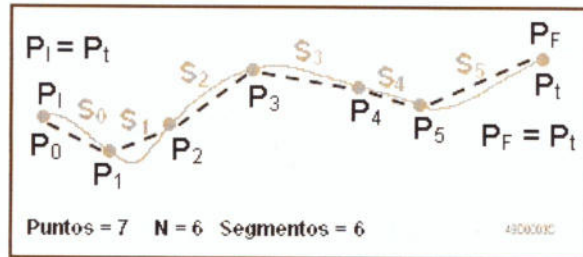


Fig. 3: Puntos Pívor y los Segmentos que los unen. Ejemplo

Así, se denominará **Segmento 0 (S_0)** al tramo recto que une los puntos programados P_0 (*Punto Inicial PI*) con P_1 ; **Segmento 1 ó S_1** , al tramo recto comprendido entre P_1 y P_2 ; S_2 al segmento de unión de P_2 con P_3 y así continuando con S_3, S_4, \dots hasta alcanzar el segmento final S_{N-1} , que une el punto P_{N-1} con el final (*PF*) P_N (En el ejemplo: S_5 , que une P_5 con P_6). El sistema CNC debe hacer que el punto móvil comandado recorra este contorno **con velocidad constante y programable**: en todo momento, **el módulo del vector velocidad debe ser constante e igual al valor programado** aún cuando los valores de las componentes sobre cada eje individual varíen debido a los cambios de dirección de dicho vector.

En función de los radios de curvatura del contorno a recorrer, *las distancias* entre los puntos programados pueden alcanzar valores importantes y muy disímiles entre sí. Esta consideración y la necesidad de actualizar los valores de referencia de los lazos en los instantes de muestreo, hace que estos segmentos deban ser fraccionados en **Tramos**, considerando como *Tramo* a la **porción de segmento que debe ser recorrida durante un período de muestreo** (Fig. 4).

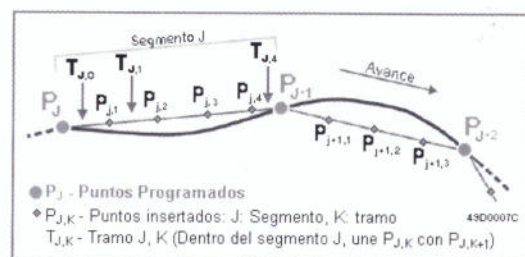


Fig. 4: Puntos Programados e Insertados, Segmentos y Tramos

De esta forma, cada uno de los N **Segmentos** son fragmentados en M **Tramos**² los que quedan com-

² Los valores de M **no son** necesariamente iguales para todos los segmentos. Es decir: $M=M(J)$

prendidos entre los puntos programados de *Inicio* y *Final* del segmento y un conjunto de *M-1 Puntos Insertados*, completando un total del *M+1* puntos totales en el segmento, los que flanquean los *M Tramos* totales (Fig.4).

E- Cálculos: ¿Qué?

Todos los cálculos que se indican tienen un único objetivo final:

Determinar en cada instante de muestreo el mando de velocidad que el CNC debe enviar como referencia a todos los ejes controlados ($VX_{JK}, VY_{JK}, VZ_{JK}, etc.$).

Valores estos obtenidos, *eje a eje*, como el producto de una constante (*Ganancia Proporcional: KX, KY, KZ*) y su correspondiente y actual *Error de Seguimiento: ($ESX_{JK}, ESY_{JK}, ESZ_{JK}, etc.$)* diferencia entre las *posiciones solicitadas ($XS_{JK}, YS_{JK}, ZS_{JK}, etc.$)* y las *posiciones reales ($XR_{JK}, YR_{JK}, ZR_{JK}, etc.$)*.³ Siendo las posiciones solicitadas la suma algebraica de todos los incrementos ocurridos desde el inicio del programa, en los instantes *JK* (uno por eje, $\forall J, \forall K$).

Es decir (Eje X):

$$ESX_{JK} = XS_{JK} - XR_{JK} \quad (2)$$

$$DXU_{JK} = KX * ESX_{JK} \quad (3)$$

Siendo VX_{JK} el valor analógico (0 a $\pm 10,000$ Vdc) resultante de convertir el número DXU_{JK} obtenido en (3) el que, a su vez ha sufrido una serie de comprobaciones y desplazamientos (*offsets*) destinados a asegurar que el mismo se mantiene acotado y centrado en cero. Todo ello acorde con el diagrama de bloques mostrado en la Fig. 5.

Resulta evidente entonces que deben hallarse, en forma individual, en cada instante de muestreo (los que a su vez se corresponden con un juego *JK*), los respectivos valores ($dXS_{JK}, dYS_{JK}, ...$) en que deben ser incrementados (o decrementados) los contadores internos que mantienen las posiciones teóricas (o *solicitadas*) para cada eje controlado. Hallar y disponer de estos valores –en los instantes adecuados– resulta una de las tareas más importantes y relevantes que debe realizar el equipo.

Formalizando:

$$XS_{JK} = dXS_{JK} + XS_{J,K-1} \quad (4)$$

F- Cálculos: ¿Cuándo?

Con el fin de optimizar la utilización del tiempo y los recursos disponibles, los cálculos se efectúan en tres etapas, *debiendo minimizarse* –tanto en cantidad como en complejidad– aquellos que deban ser realizados en *tiempo de ejecución*, fundamental-

³ *JK*: Datos del *Segmento J, Tramo K*. Los valores *XR* y similares son obtenidos del sistema de realimentación de posición. Corresponden al instante de muestreo correspondiente al inicio del tramo *JK*.

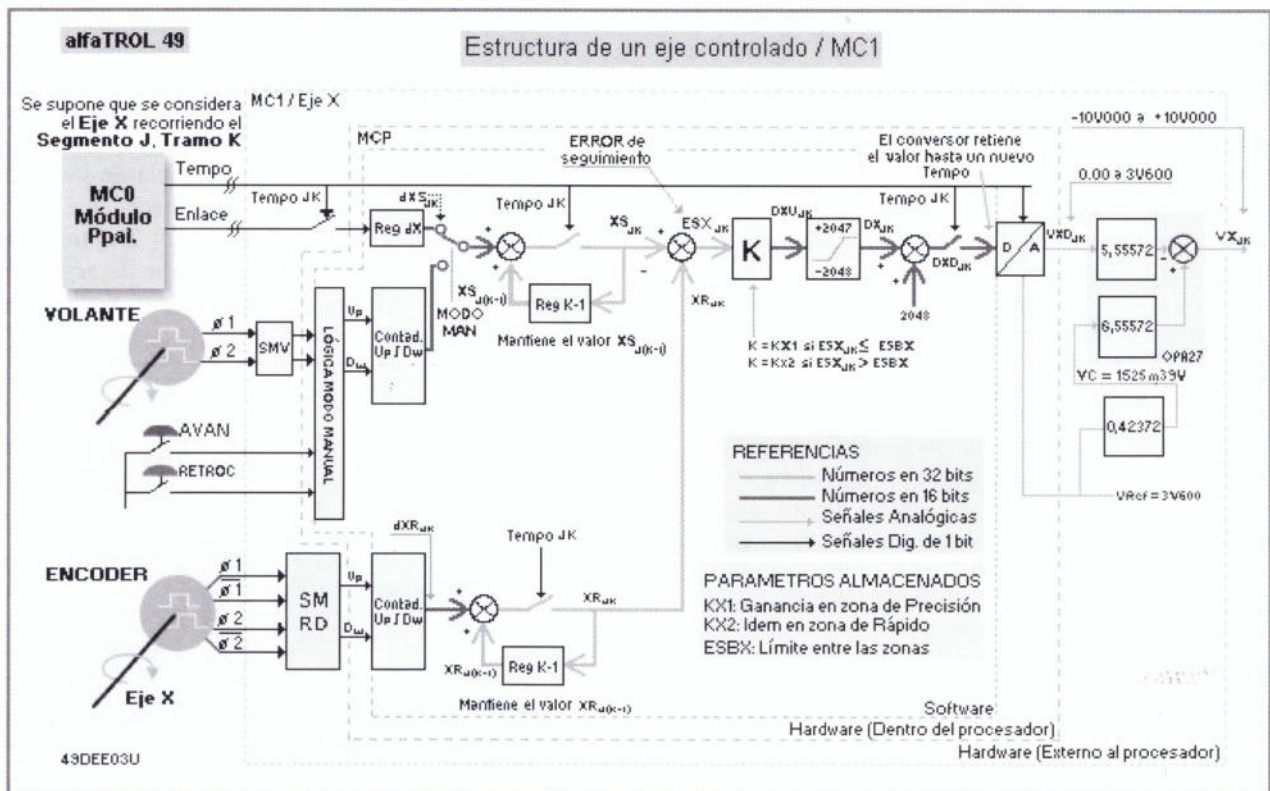


Fig. 5: Diagrama de bloques simplificado de la estructura de Hard- y software de un eje controlado (X)

mente, si el sistema se halla en movimiento. El equipo ha sido desarrollado para realizar cálculos:

1. Durante el *proceso de programación* y
2. Durante la *ejecución del programa*:
 - a. Al inicio del mismo
 - b. Al inicio de cada *segmento*.
 - c. Al inicio de cada *tramo*

Durante 1. y 2a. el uso del tiempo no resulta crítico ya que el sistema está inmóvil. En las dos situaciones últimas (2b. y 2c.), el sistema está en movimiento, lo que obliga a minimizar los tiempos de cálculo ya que, por una parte se requiere respetar rigurosamente los instantes de muestreo y, por otra, el sistema debe realizar –obligatoriamente– otras tareas que también demandan tiempo tales como la función *PLC*, refresco de pantalla, etc. Además, a medida que se avanza en la lista anterior, el índice de repetición se incrementa en forma notable, debiendo repetirse los cálculos un número cada vez mayor de veces: Los cálculos de 2a. se efectúan una única vez, los de 2b. $N - 1$ veces (una vez por cada punto programado sin contar el punto inicial (PI) ni el final (PF)), mientras que los de 2c. se ejecutan un número de veces equivalente a la sumatoria de **todos los tramos** de **todos los segmentos** del programa, es decir un número que puede ser elevadísimo.

Podría pensarse que este proceso se simplificaría variando la estrategia indicada adoptando –por ejemplo– un procedimiento tal que realizara todos los cálculos necesarios para recorrer todo el contorno durante la programación o en otro tiempo *antes de la ejecución*. Se obtendrían así todos los resultados requeridos, incluso los valores de referencia para todos los ejes para todos los tramos, los que podrían ser almacenados, entregando luego, en tiempo de ejecución, cada uno de ellos como referencia de cada eje controlado al inicio del período de muestreo correspondiente. Esta idea, evidentemente muy simple y atractiva, resulta inviable dado que existen datos variables que intervienen en los cálculos que deben ser *necesariamente adquiridos en tiempo real*, durante la ejecución, hecho que obliga a operar acorde a la estructura adoptada u otra similar.

G- Sistemas de coordenadas

El equipo ha sido preparado para operar controlando sistemas de muy diferente naturaleza y arquitectura. Es así posible comandar mecanismos basados en arquitecturas y sistemas de coordenadas referidos a ejes cartesianos ortogonales, polares, combinación de ellos, etc. A modo de ejemplo se describe su funcionamiento cuando la arquitectura

del sistema controlado es *cartesiana ortogonal* (caso mayoritario). Para sistemas diferentes, el procedimiento de cálculo es también diferente aunque similar. Se debe destacar que en algunos casos (por ejemplo comandando una arquitectura doble polar), puede resultar compleja –o muy compleja– la determinación de las distancias entre puntos *pívots* consecutivos, datos que, como se verá, son imprescindibles para la realización de los cálculos.

H-Interpolación. Cálculos.

-Cálculos: En tiempo de programación:(*Tiempo 1*) Al ingresarse los puntos *pívot*, se calculan las longitudes de los **segmentos** que se van definiendo como unión entre dichos puntos. (Fig. 6 -Caso de dos ejes).

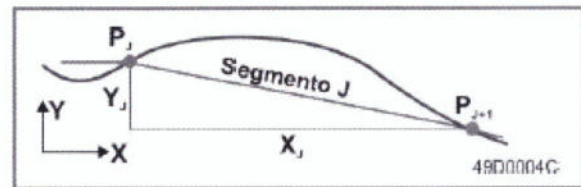


Fig. 6 - Componentes X, Y del SJ (X_J & Y_J)

En general:

$$D_J = \sqrt{X_J^2 + Y_J^2 + Z_J^2} \quad (5)$$

(en el ejemplo que se muestra, sólo existen las componentes X_J , Y_J). Estos valores son siempre números enteros ya que –internamente– todas las distancias son medidas en **UR** (*Unidades de Resolución*). El valor de esta distancia queda internamente almacenada como parte del programa a ejecutar.

- Cálculos: Al inicio de cada segmento (*Tiempo 2.b*)

- *Determinación de M*

M, es decir el número de **tramos** requeridos para recorrer cada **segmento**, debe calcularse –necesariamente– en tiempo de ejecución ya que la *velocidad de contorno*, uno de los parámetros incluidos en el programa y necesario para el cálculo, puede ser modificada en cualquier momento –incluso– durante la ejecución del programa (la altera el operador a través del comando *feed rate - FR -*) debiendo su valor real, actual, ser adquirido al inicio de cada segmento. Se ha determinado que las posibles variaciones del **FR** serán aceptadas e incorporadas al movimiento al *inicio de cada segmento*, manteniendo su valor –al menos– durante el recorrido del mismo. Variaciones ocurridas mientras se esté transitando uno de ellos serán incorporadas al movimiento al concluir el mismo y tendrán recién influ-

encia en el recorrido del próximo.

Al inicio de cada segmento el sistema adquiere la corrección de velocidad actual FR ($0 \leq FR \leq 1,3$) y calcula el valor exacto de la velocidad para el próximo. Así, en el punto P_J calcula el valor de velocidad V_J para el segmento S_J :

$$V_J = FR * V_P \tag{6}$$

donde V_P es la velocidad programada.

Para la expresión de las velocidades se adopta el uso de las unidades naturales del sistema:

$$[V_P] = \frac{UR}{T} \tag{7}$$

o sea: Unidades de Resolución [UR] por período de muestreo [T]. En el caso de V_P , el sistema habitualmente se programa en mm/min., valores que son convertidos a unidades naturales, en forma automática, al momento de su almacenamiento. De igual manera se procede con los valores de las cotas y otras cantidades programables: Son traducidas automáticamente a unidades naturales en tiempo de programación.

Disponiendo de D_J y de V_J , puede calcularse fácilmente el número de períodos de muestreo necesarios para recorrer el segmento S_J :

$$M_J = \frac{D_J}{V_J} \tag{8}$$

el que queda medido en períodos de muestreo T .

Este número M_J es -en principio- el número de tramos que deberá contener el segmento J, pero, por la forma de cálculo (8), el valor de M_J es un número racional, es decir *puede resultar un número NO entero*. Dado que el número de segmentos **debe** ser un entero, el valor obtenido en (8) se modifica acorde con:

- Si la fracción decimal calculada en (8) es menor que 0,5, simplemente se redondea

$$M_J = INT\left(\frac{D_J}{V_J}\right) \tag{9}$$

- En el caso contrario, en que la parte decimal de (8) resulta mayor a 0,5, se adiciona un segmento extra al recorrido:

$$M_J = INT\left(\frac{D_J}{V_J}\right) + 1 \tag{10}$$

Esto genera un inconveniente menor que habitualmente pasa desapercibido: El último tramo resulta de una longitud levemente mayor en el caso (9) y levemente menor en el caso (10) (la diferencia nunca excede medio tramo, en forma NO repetitiva ni acumulativa.) Como todos los tramos son recorridos durante un tiempo exactamente igual

a T , la velocidad durante este último tramo se ve levemente alterada. Evidentemente esto ocurre durante un período de muestreo, es decir durante un tiempo de uno o algunos milisegundos, motivo por el cual las masas en movimiento -que ejercen un efecto de filtrado PB- tornan indetectable el fenómeno.

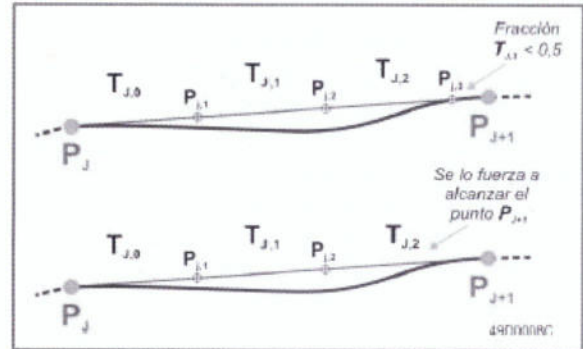


Fig. 7: Ejemplo: M_J en (8) resulta de un valor (aprox.) de 3,4 T

- *Determinación de los valores básicos de dXS_{JK} , dYS_{JK} (y sus homólogos en otros ejes).*

Como puede verse en la Fig. 8, los valores básicos de los incrementos (o decrementos) para cada eje en cada instante de muestreo posee un valor básico (ejemplo en dos ejes -Fig. 8).

$$X_{JK} = dXS_{JK} = \frac{X_J}{M_J} \tag{11}$$

$$Y_{JK} = dYS_{JK} = \frac{Y_J}{M_J} \tag{12}$$

Y así siguiendo para todos los ejes.

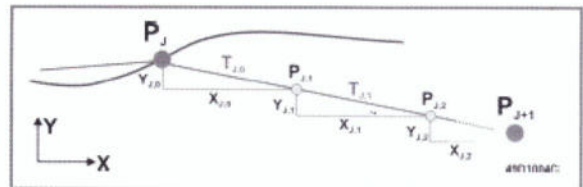


Fig. 8: Ubicación de los puntos insertados

- *Cálculos: Al inicio de cada tramo (Tiempo 2.c)*

Como ocurrió antes, los valores de X_{JK} , Y_{JK} y homólogos, son racionales, es decir, que en general no son enteros: Cada uno de ellos posee una parte decimal, la que -en principio- se ignora, tomando y transfiriendo, al inicio de cada tramo, como valor de incremento, sólo la parte entera de (11), (12), etc. Esto genera un error acumulativo de una fracción de UR por cada tramo que se inicia. La unidad de cálculo registra esta situación y acumula también estas fracciones, las que igualan -o sobrepasan- la unidad en algún momento. Al alcanzarse esta situación, se transfiere como incremento el valor base (11), (12), etc. más uno.

Finalmente, para el último tramo, el incremento que se transfiere, es el valor base más (o menos) el número de URs necesario para alcanzar exactamente el próximo punto programado (en este caso P_{j+1}).

Todos estos cálculos los efectúa la unidad central, transfiriendo, vía el sistema de comunicaciones entre módulos, los valores a cada uno de los módulos de control de eje.

IV. PROGRAMACIÓN INTERNA: PRINCIPIO DE FUNCIONAMIENTO: PLC

El equipo ha sido dotado de un poderoso controlador digital programable. Éste, que se halla distribuido entre sus módulos de control de ejes (MC_j) y el módulo de control central (MC_0), posee un número de entradas, salidas y otros recursos que se va incrementando a medida que se incrementa el número de ejes comandados (y –por consiguiente– el número de módulos controladores). Cada uno de ellos posee un cierto número de Entradas Digitales (ED) (en general $8x$), así como salidas digitales (SD) (igual número).

Al inicio de cada período de muestreo –y durante el tiempo de intercambio de información– las diferentes unidades envían al módulo central de comando los estados de las entradas asociadas a su eje controlado, a lo que el módulo central replica con los estados que deben adoptar las salidas asociadas al mismo (con un período de muestreo de retardo). Todo el procesamiento se realiza en el módulo central, sirviendo los módulos de control de ejes como sistema de adquisición de datos y de salida de señales de comando. En otros trabajos se describirán detalladamente las funciones disponibles en el PLC, así como otras funciones auxiliares incorporadas.

V. CONCLUSIÓN

El equipo, que se halla operativo con siete unidades en producción, ha demostrado una gran solvencia y una fuerte adaptabilidad a las muy disímiles funciones a las que han sido destinados los ejemplares fabricados. Los primeros se hallan en producción, todos fuertemente demandados (alta producción, medio ambiente hostil, etc.), algunos desde hace más de dos años, habiendo demostrado

una confiabilidad extrema: Prácticamente no ha demandado tareas de mantenimiento desde su puesta en marcha.

EQUIPOS OPERANDO:

1. Máquina p/preseting de herramientas. (2D) **Metalúrgica ROMA S.A.** (Córdoba)
2. Máquina especial (Centro de Torneado – Centro de Mecanizado) para la producción de lentes intraoculares. (2D) (Discriminación $0,1 \mu\text{m}$, -Calidad óptica) **Productores Ópticos Asoc.** (Córdoba)
3. Máquina aplicadora de pegamento bicompuerto en el espacio (3D) - **PILOT Automotive S.A.** (Ford) (Parque Industrial TIGRE - B.Aires)
4. Máquina *transfer* para la fabricación de extremos de dirección. (3 cabezales CNC, 3 husillos, y comando PLC de 3 estaciones no CNC.) - **DANA Argentina S.A.** (San Luis).
5. Máquina aplicadora de producto formador de junta. (2D – *Doble Polar*) - **FIAT Auto Argentina S.A.** (Córdoba)
6. Laminador Parabólico en caliente (2D) - **Allevard Rejna Argentina S.A.** (Córdoba)
7. Máquina para la fabricación flexible de resortes. (3D) **Resortes Argentina S.A.** (Córdoba) (En período de puesta en marcha)

AGRADECIMIENTOS

Los autores agradecen a las empresas que han adoptado el sistema desarrollado confiando en sus posibilidades productivas a pesar de la escasa tradición existente al respecto, así como al **Superior Gobierno de la Provincia de Córdoba (República Argentina)**, el que ha conferido a este desarrollo el Premio “*A la Innovación en Alta Tecnología*” (Ministerio de Producción y Trabajo – Secretaría de Industria y Comercio - Año 2007.)

REFERENCIAS

- [1] G. Stute, et. al., “*Regelung an Werkzeugmaschinen*”, Carl Hansen Verlag, München, Wien, 1981.
- [2] D. Binder, “*Interpolation in numerischen Bahnsteuerungen*”, Springer-Verlag, Berlin – Heidelberg – N. York, 1979. 