Anexos

9 Anexo I: Calibración de Parámetros Geométricos mediante el Análisis de Circle-Point

9.1 Notación cinemática

Antes de comenzar el desarrollo de cualquier algoritmo de calibración, se requiere una definición precisa y formal de los parámetros cinemáticos necesarios. Estos parámetros que incluyen los términos fijos y los dependientes de los actuadores deben únicamente especificar la posición y orientación de cada articulación o cuerpo del sistema. La notación y las definiciones de los parámetros cinemáticos precisos se determinan aquí, basados en el trabajo de Duffy y Lipkin. Aunque la notación es ligeramente diferente, y las definiciones matemáticas son más rigurosas, éste en esencia sigue la conocida convención de Denavit-Hartenberg. La Ilustración 25 ilustra todos los parámetros cinemáticos para un brazo manipulador de n-grados de libertad (*DOF = Degrees of Freedom*) completamente arbitrario.

Cada par de ejes adyacentes está conectado por una línea común y mutuamente perpendicular formando una articulación fija. Cada una de las uniones o ejes de movimiento se denotan por el vector $\overrightarrow{S_i}$ para cada eje i. Las perpendiculares comunes o vectores de la articulación se denotan por el vector $\overrightarrow{a_{ij}}$ para cada par de ejes i y j. Estos vectores direccionales, unidos a cada articulación, son rigurosamente definidos debajo:

$\overrightarrow{S_i}$ → Vector dirección del eje de la unión, siempre en la dirección de ahí a $\overrightarrow{a_{ij}}$. Donde los subíndices $hijk$ representan la secuencia de ejes ordenada.

$\overrightarrow{a_{ij}}$ → Vector dirección de la línea perpendicular mutua que une los ejes definidos por $\overrightarrow{S_i}$ y $\overrightarrow{S_j}$, y siempre en la dirección del eje i al eje j.

Es importante hacer notar que esto no es necesariamente una representación física de las articulaciones reales. Las líneas perpendiculares comunes pueden no coincidir con la estructura de miembros real de la articulación. Así, con esta definición geométrica, los parámetros calibrados o sus correcciones pueden no tener relación con ninguna dimensión de diseño. Por tanto, el proceso de calibración no se puede usar como una herramienta de realimentación del proceso y control de producción. Los modelos cinemáticos modificados que se asientan en más de 4 parámetros (el número mínimo requerido para relacionar líneas en el espacio) han sido introducidos así para acomodarse a estos y otros problemas.

Los cuatro parámetros usados para especificar completamente una única articulación son: la longitud de la articulación (a), al ángulo de giro (α), la dimensión del desplazamiento de la articulación (d) y el ángulo de la articulación (θ). Cada uno de estos parámetros del modelo se define a continuación:

1. <u>Longitud de la articulación</u> $\overrightarrow{a_{ij}}$: Es la distancia perpendicular entre el eje adyacente i y j. Esta es la magnitud del vector de la articulación mutuamente perpendicular.

2. Ángulo de giro α_{ij}: Es el ángulo formado por los ejes adyacentes *i,j*, y medido como la rotación dextrógira desde $\vec{S_i}$ hasta $\vec{S_j}$ alrededor del vector $\overrightarrow{a_{ij}}$. La siguiente definición garantiza esta restricción direccional:

$$\cos\alpha_j = c_{ij} = \vec{S_i} \times \vec{S_j} \qquad \text{Ecuación 4}$$

$$\sin\alpha_{ij} = s_{ij} = \left(\vec{S_i} \times \vec{S_j}\right) \cdot \overrightarrow{a_{ij}} \qquad \text{Ecuación 5}$$

3. Dimensión desplazamiento d_j: Es la distancia de desplazamiento entre dos articulaciones dada como la distancia entre $\overrightarrow{a_{ij}}$ y $\overrightarrow{a_{jk}}$ a lo largo del eje $\vec{S_j}$.
4. Ángulo de la articulación θ_j: Este es el parámetro cinemático que define la relación entre articulaciones adyacentes. Es el ángulo entre las articulaciones adyacentes *ij* y *jk*, y se mide como la rotación dextrógira desde $\overrightarrow{a_{ij}}$ a $\overrightarrow{a_{jk}}$ alrededor del vector $\vec{S_j}$. La siguiente definición garantiza esta restricción direccional:

$$\cos\theta_j = c_j = \overrightarrow{a_{ij}} \cdot \overrightarrow{a_{jk}} \qquad \text{Ecuación 6}$$

$$\sin\theta_j = s_j = \left(\overrightarrow{a_{ij}} \times \overrightarrow{a_{jk}}\right) \cdot \vec{S_j} \qquad \text{Ecuación 7}$$

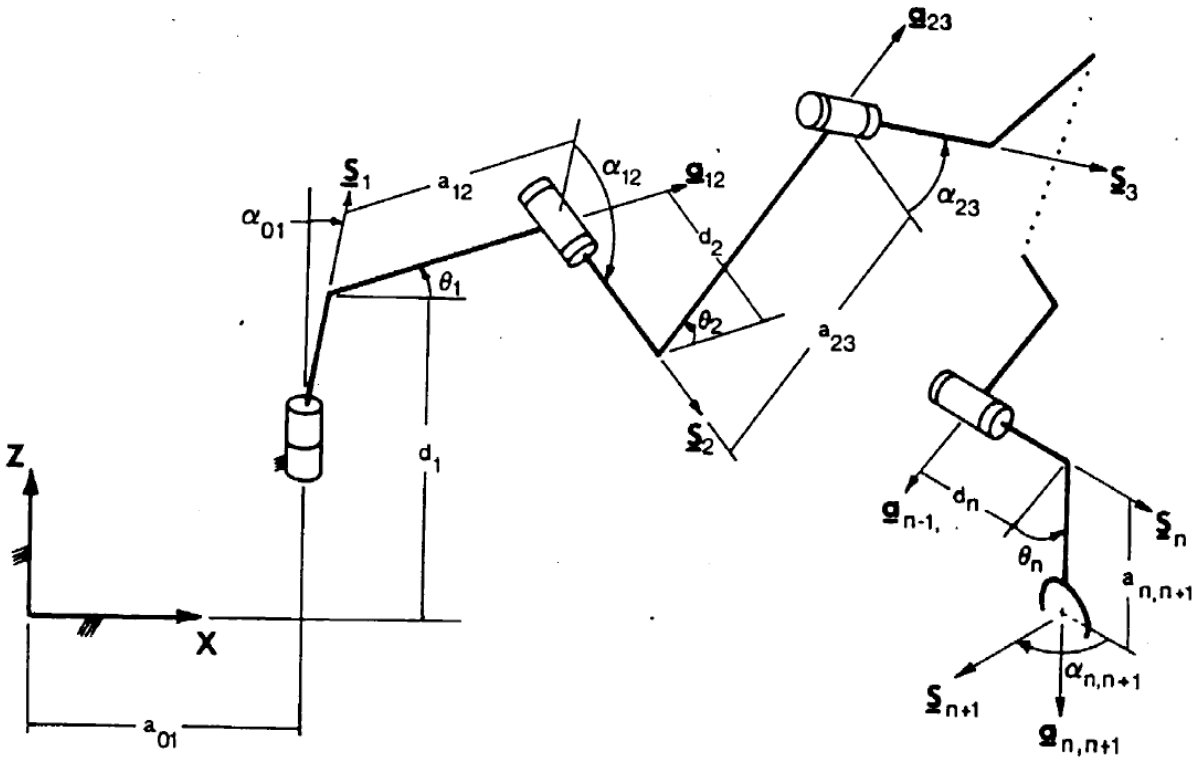


Ilustración 25 – Descripción cinemática de un manipulador general de n-grados de libertad

En esta sección los manipuladores que contienen sólo un único grado de libertad, o actuadores prismáticos, se consideran igualmente. Por tanto, para cualquier articulación dada en el sistema sólo 3

de los 4 parámetros modelados son constantes fijas. O bien la dimensión de desplazamiento (en el caso de las uniones prismáticas) o bien el ángulo de la articulación (en el caso de las uniones en revolución) es variable. Esta dimensión variable constituye la variable actuada. Todos estos parámetros se deben conocer de forma precisa para predecir la posición de la articulación final y controlar el manipulador con precisión. Los procedimientos de calibración deben identificar los parámetros fijos y predecir o modelar errores en los transductores que determinan las variables actuadas.

Además de los parámetros de cada articulación, los parámetros que identifican un sistema de coordenadas mundial o global, y la ubicación de la herramienta del actuador final dentro de la última articulación, también se deben identificar. Como se ve en la figura, la articulación base es la 01. Los parámetros que localizan el sistema de coordenadas de referencia son a_{01} y α_{01}. El primer ángulo de la unión es el ángulo entre la primera dirección de la articulación a_{12}, y el eje global X. Una herramienta del actuador final representada por los vectores ortogonales $\overrightarrow{S_n}$ y $\overrightarrow{a_{n,n+1}}$ se localiza en la última articulación, mediante los parámetros d_n, $a_{n,n+1}$ y $\alpha_{n,n+1}$. Nótese que los vectores ortogonales forma los ejes Z y X de la herramienta. Así, considerando todos los parámetros geométricos fijos requeridos, un total de 3N+2 constantes se deben calibrar. Si un único término de error constante se usa para modelar los errores de los transductores entonces un total de 4N+2 parámetros se deben modelar.

Un cierto número de investigadores ha mostrado que el modelo de 4 parámetros de Denavit – Hartenberg arriba mostrado es inadecuado para sistemas con ejes adyacentes nominalmente paralelos. Esta condición será más adelante ilustrada. Para tratar con este caso más que común (ocurre en virtualmente todos los robots industriales usados hoy en día) se requiere el uso de un parámetro adicional para cada articulación. La técnica del Análisis del Círculo de Puntos es capaz de determinar estos parámetros adicionales. Normalmente el quinto parámetro constituye un término de rotación adicional entre las estructuras adyacentes. Si una estructura local o un sistema de coordenadas se adjunta a cada articulación que contiene un eje Z y X formado por los vectores $\vec{a}$ y $\vec{S}$, entonces el parámetro adicional beta representa una desplazamiento de rotación o error alrededor del eje local y. La naturaleza física de este parámetro, y la razón por la que resuelve el problema de los ejes nominalmente paralelos, se ilustra en el siguiente apartado. Con este modelo de 5-parámetros la transformación total o desplazamiento entre las articulaciones adyacentes ij y jk viene dada por la siguiente transformación compuesta:

$$trans(\overrightarrow{a_{ij}}, a_{ij})\, rot(\overrightarrow{a_{ij}}, \alpha)\, rot(\overrightarrow{b_{ij}}, \beta)\, trans(\overrightarrow{S_j}, d_j)\, rot(\overrightarrow{S_j}, \theta_k) \qquad \text{Ecuación 8}$$

Donde *trans(v,d)* representa la traslación lineal a lo largo del eje v, y *rot(v,th)* representa la rotación alrededor del eje v en una cantidad θ.

9.2 Calibración de Parámetros Geométricos mediante el Análisis de Círculo de Puntos

La precisión de un brazo robot industrial típico puede ser mejorada mediante la calibración precisa o determinando los parámetros geométricos del sistema. La técnica de Círculo de Puntos es un modo de aproximación fácil y sencilla, la cual da una determinación precisa de los parámetros geométricos exactos en un sentido absoluto. Esta aproximación (en la que los términos reales se computan) se encuentra en contraposición con que los errores se tienen con respecto a un conjunto de valores nominales. Así, el conocimiento de los datos nominales del fabricante no se necesita. Esta técnica es un método de ejes independiente. Básicamente, cada uno de las articulaciones de los ejes del sistema se localiza en el espacio para una única configuración. Cada par de líneas adyacentes en el espacio formado por los ejes es entonces analizado para determinar completamente los parámetros para ese enlace. Las primeras ventajas de esta técnica son que se da un indicio geométrico real de cada enlace y quela medición 3D de un punto objetivo único se puede hacer en un sistema de coordenadas arbitrario. Los parámetros fijados relativos a dos líneas en el espacio son independientes del sistema de coordenadas usado para definir las líneas.

El conjunto completo de parámetros geométricos y notación cinemática requerido para calcular de forma precisa la ubicación de un elemento final en un sistema de referencia de coordenadas bien definido, dados los desplazamientos de las articulaciones variables, es bien conocido. Los parámetros cinemáticos que pueden ser determinados por el análisis del Círculo de Puntos incluyen los siguientes:

- Ángulos de giro : $\alpha_{12} \dots \alpha_{N-1,N}$
- Longitudes de los enlaces: $a_{12} \dots a_{N-1,N}$
- Desplazamientos: $d_1 \dots d_{N-1}$
- Ángulos de articulaciones: $\theta_1 \dots \theta_{N-1}$

A notar que los parámetros relativos al sistema de coordenadas de referencia a la primera articulación, a_{01} y $alfa_{01}$, y aquellos relativos a la localización de una herramienta (*tool frame*) en la última articulación, d_N, $a_{N,N+1}$, y $alfa_{N,N+1}$, no pueden ser fácilmente determinados mediante esta técnica. Esto es debido al hecho de que sólo los ejes de los sistemas en los que el movimiento ocurre pueden ser ubicados en el espacio. Así, sólo aquellos parámetros entre la primera y última articulación del eje pueden ser determinados. Aunque no se hace aquí, esta técnica puede ser mejorada de forma inteligente suministrando medios de localizar en el espacio la referencia del eje-Z y del eje-Z de la herramienta local, $\overrightarrow{S_{N+1}}$. Una sugerencia para localizar el eje-Z de la herramienta es montar un objetivo en una fijación unida al conjunto del eje, el cual esté obligado a deslizarse a lo largo o rotar alrededor del eje-Z.

Los parámetros arriba representan los términos cinemáticos puros. Para todos los ejes, dependiendo del tipo de articulación, bien el ángulo θ_i o el valor de desplazamiento d_i, es un parámetro fijo, y los otros representan las variables en movimiento. La técnica del Círculo de Puntos normalmente determina los parámetros geométricos fijos. Sin embargo, es posible determinar también las variables de la articulación para una configuración final o para una serie de movimientos de articulaciones. Esto se explica más adelante en la siguiente sección. Para el caso de una articulación de revolución, la dimensión de desplazamiento d_j es el parámetro fijo y el ángulo cinemático θ_j representa la variable, movimiento de revolución. Lo opuesto es cierto para uniones prismáticas (desplazadores). En este capítulo, las figuras representan todos los sistemas en revolución y el análisis se apoya primeramente

en este hecho. Sin embargo, la técnica se modifica fácilmente para manejar sistemas los cuales contienen ejes prismáticos. Las modificaciones se definirán en breve.

Las técnicas de ejes independientes, basadas en la geometría, no son tan comunes entre los investigadores. Sin embargo, unos pocos investigadores han desarrollado técnicas similares a las del Círculo de Puntos, aquí presentada. El trabajo de Judd y Knasinski en particular es casi una técnica idéntica. La técnica desarrollada aquí se desarrolló completamente independiente de cualquier literatura de calibración, y originalmente fue concebida in 1984.

9.3 Explicación de la Técnica del Círculo de Puntos

El primer objetivo del método de medida mediante el análisis de Círculo de Puntos es ubicar cada uno de los ejes de articulación en el espacio. Hay que notar que la localización de una línea en el espacio requiere tanto la dirección de la línea como las coordenadas de un punto arbitrario en la línea. Un único punto objetivo capaz de ser colocado en cualquier lugar de un brazo y un sistema de medición 3D se necesitan para este experimento. Una vez cada línea es localizada en el espacio para la configuración final del robot, el análisis geométrico de las coordenadas de la línea da los parámetros calibrados geométricos requeridos. Las técnicas analíticas se muestran en las siguientes secciones. El procedimiento experimental básico consiste en los siguientes pasos:

1. El robot se coloca en la configuración inicial, que se graba (como un conjunto de ángulos de las articulaciones). Un único punto objetivo P se adhiere al brazo, después del primer eje (no en la base), a una distancia perpendicular del eje igual a un radio de valor definido R_1.
2. El robot entonces se comanda para moverse alrededor de la primera articulación únicamente, parándose en un número de localizaciones discretas. La localización del objetivo, el cual traza un camino aproximadamente circular, se mide en cada posición en cualquier sistema de coordenadas elegido convenientemente. Esto crea un círculo de puntos de datos, lo que le da el nombre a esta técnica de Círculo de Puntos. Los valores de desplazamiento de la articulación para el primer eje se graban para cada posición. La Ilustración 26 ilustra los puntos en el círculo de la articulación 1, para todo un sistema en completa revolución. Nótese que todas las articulaciones excepto la primera permanecen fijas. Obviamente, cualquier robot calibrado mediante esta técnica deben ser capaces de un control de movimiento por articulación individual.
3. Los datos de puntos del círculo son entonces analizados para definir la ubicación del eje. Primero, los puntos del círculo se encajan en un plano que es el que mejor se aproxima, mediante una aproximación por mínimos cuadrados, que consigue definir la ecuación del plano. La normal del plano resultante define la <u>dirección del eje de la articulación</u>, definida como $\overrightarrow{S_1}$. Los puntos entonces se proyectan sobre el plano, en dirección perpendicular al mismo. Los puntos proyectados se encajan, otra vez usando la solución de mínimos cuadrados, en la ecuación de un círculo en el plano. El centro de este círculo, que se denota como $\overrightarrow{r_1}$, <u>representa un punto en el eje o línea definido por el eje de la articulación</u>.
4. El punto objetivo P se adhiere entonces al brazo en una posición después del segundo eje (es decir, en un punto afectado por el movimiento del eje 2), a una distancia perpendicular al eje aproximadamente igual al radio de valor R_2.

5. El robot se controla entonces para moverse alrededor del eje 2 solamente. Otra vez, el punto P traza un camino aproximadamente circular y se miden y graban los valores individuales de desplazamiento de las articulaciones. Nótese que la articulación 1 no se debe mover, ya que esto causaría el movimiento del eje 2. El objeto de este paso es localizar el eje 2 el cual por consiguiente permanece fijo. Los puntos del círculo son entonces analizados usando la misma técnica analítica que en el paso 3, que permite obtener la dirección $\overrightarrow{S_2}$ del eje y un punto $\overrightarrow{r_2}$ en el eje o línea. Los puntos del círculo para el eje 2 se muestran también en la figura.
6. Un proceso idéntico se debe repetir para cada eje en el sistema. Hay que prestar un cuidado extremo en evitar mover cualquier eje que esté ubicado en la cadena de enlaces cinemática antes del eje que se está moviendo en cada momento.

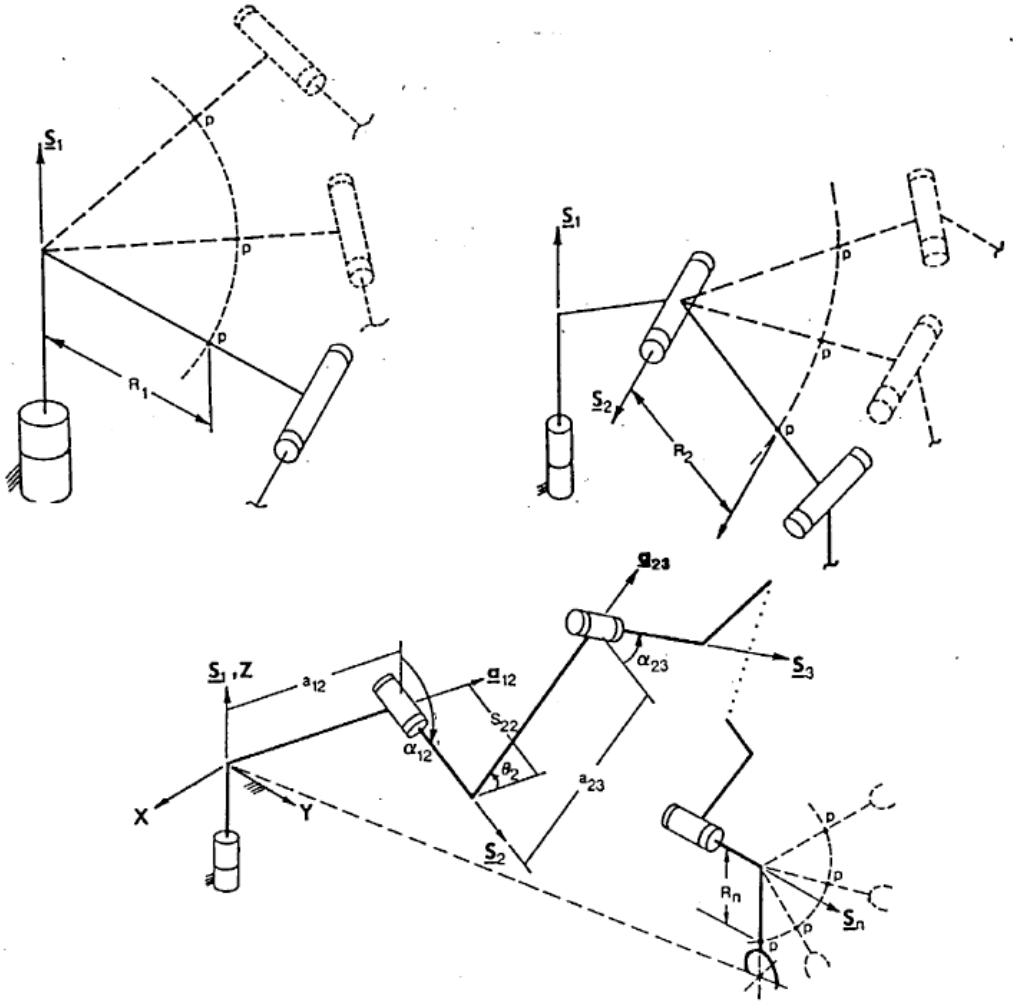


Ilustración 26 – Movimiento del Círculo de Puntos para ejes individuales

El procedimiento antes mostrado no da un indicio de la configuración inicial o final que se debe usar. El rango de movimiento y valores iniciales de cada desplazamiento de la articulación tampoco está definido. Actualmente, el experimentador debe usar su mejor juicio para dar esta información. Las directrices generales son:

1. Cada unión debería moverse a través de un rango tan grande como sea posible para generar su círculo de puntos.
2. La configuración inicial debería resultar en pares de torsión tan bajos como sea posible debido a la gravedad sobre los primeros ejes.
3. La configuración final debería estar cerca de la región que se pretenda que sea para la mayoría de aplicaciones.

Obviamente, para aplicaciones en general, la última directriz no es posible. No está claro actualmente si varias configuraciones causan una diferencia en los parámetros calibrados o no. Se requieren futuras investigaciones para mejorar las configuraciones seleccionadas y los rangos de movimiento para este procedimiento. La razón fundamental por la que el experimento debe ser dependiente de la configuración es que las deformaciones del sistema son causadas por cargas de la gravedad. Un interesante resultado de la técnica del Círculo de Puntos es que las deformaciones debido a la gravedad en realidad deben ser tenidas en cuenta y compensadas usando los parámetros calibrados así determinados. Los parámetros calibrados determinados representan en realidad el sistema en un estado deformado "medio", debido a la carga de la gravedad.

El método de Círculo de Puntos es la técnica más útil para los fabricantes porque provee una indicación directa de los errores de fabricación y ensamblado en la geometría del sistema y un indicio de la integridad mecánica global de cada articulación. Cuando se encaja el Círculo de Puntos en un plano, el valor medio de los errores residuales o distancias de cada punto al plano que "mejor encaja" da un indicio de la dirección del movimiento del eje de la articulación o "cabeceo" ("wobble"). Cuando se encaja los puntos proyectados en el plano en un círculo, el valor medio de los errores residuales o distancia radial entre cada punto y el círculo da una indicación de las traslaciones del eje de la articulación o espacio entre rodamientos, comúnmente llamada como "juego" ("slop").

Los datos resultantes del procedimiento experimental ($\overrightarrow{S_1}, \overrightarrow{r_1}, \overrightarrow{S_2}, \overrightarrow{r_2}, \dots, \overrightarrow{S_n}, \overrightarrow{r_n}$) pueden ser usadas para especificar completamente cada eje como una línea en el espacio para la configuración final alcanzada. Cada par de líneas adyacentes es entonces analizado para determinar los valores de α, a, d y θ para cada enlace en el sistema. Nótese, que para cada configuración final alcanzada, tanto los parámetros d (desplazamientos del enlace) y θ (ángulo cinemático) se pueden determinar para cada articulación, sin importar qué valor se fije. Así, para un eje en revolución, el parámetro variable θ se puede comparar al valor del ángulo del *encoder* del sistema grabado para el último punto del círculo. Este par de datos únicos de ángulos reales versus ángulo del sensor se pueden usar como un parámetro de corrección para los sensores de las articulaciones. Judd and Knasinski incluso dan una mejora que va más allá en su técnica. Para cada eje, la cantidad de movimiento angular desplazado entre cada par de puntos discretos en el círculo se puede calcular. Entonces estos valores se pueden comparar a las diferencias en los valores de los ángulos de sensores grabados, los cuales dan un número de datos par para calcular los parámetros de corrección.

Los ejes prismáticos no dan dificultades inherentes en este análisis. Todas las técnicas y datos mencionados anteriormente se pueden determinar para ejes prismáticos, ya que el análisis está basado únicamente en la localización de líneas o ejes en el espacio. La ubicación de un eje prismático se determina por un punto objetivo, el cual describe una trayectoria aproximadamente a lo largo de una línea recta. Los puntos medidos se pueden encajar de forma simple, a través del método de mínimos cuadrados, en la ecuación de una línea un único paso analítico. Nótese que el método explícito para

encajar puntos en la línea o círculo no se describe en este documento. La mayoría de los sistemas de medición 3D basados en ordenador suministran esta capacidad automáticamente. Los únicos datos presentados al análisis metrológico por ordenador han sido las direcciones de los ejes de las articulaciones ($\vec{S}$) y los puntos en los ejes ($\vec{r}$). Cada par de líneas o ejes 1-2, 2-3, …, (N-1)-N se debe analizar individualmente para dar los parámetros requeridos de la unión o articulación. Los análisis de dos líneas arbitrarias en el espacio se presentan en las siguientes secciones.

9.4 Revisión de la definición de líneas y planos en el espacio

Una definición clara, sea tanto matemática como geométrica, de tanto líneas como de planos en un espacio tridimensional es lo que se necesita para analizar completamente las líneas adyacentes formadas por los ejes de un manipulador. Estas definiciones se revisan aquí brevemente. Una línea general en el espacio es la que se muestra en la Ilustración 27. Los únicos parámetros requeridos para especificar completamente una línea que tiene una dirección y ubicación fijas son:

$\vec{S_i}$ (L, M, N) → Dirección del vector correspondiente a la línea.
$\vec{r_i}$ (x_i, y_i, z_i) → Un punto cualquiera dado en la línea.

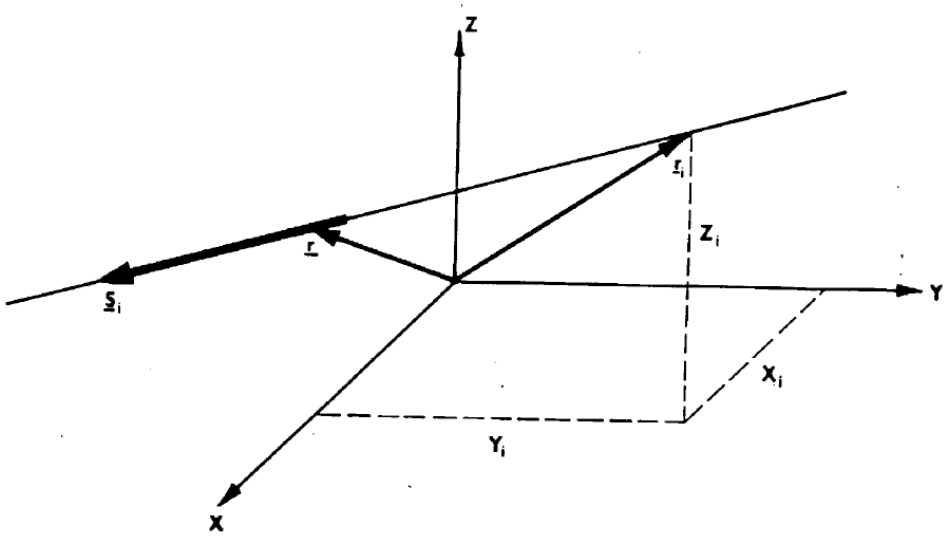


Ilustración 27 - Definición de una Línea en el espacio

Nótese que el vector dirección puede ser considerado como un vector libre que se puede colocar en cualquier lugar a lo largo de la línea. Considerando un punto general en la línea, sea $\vec{r}$(x,y,z), la siguiente ecuación de producto cruzado es obviamente válida:

$$(\vec{r} - \vec{r_i}) \times \vec{S_i} = 0$$ Ecuación 9

Reescribiendo esta ecuación se obtiene la <u>ecuación de una línea</u>:

$$\vec{r} \times \vec{S_i} = \vec{r_i} \times \vec{S_i} = \overrightarrow{S_{0i}}$$ Ecuación 10

Donde $\overrightarrow{S_{0i}}$ representa los momentos de cada componente del vector dirección sobre los ejes. El vector momento $\overrightarrow{S_{0i}}$ (P,Q,R) está definido directamente de los datos dados anteriormente, como sigue:

$$\begin{Bmatrix} P = y_i \cdot N - z_i \cdot M \\ Q = z_i \cdot L - x_i \cdot N \\ R = x_i \cdot M - y_i \cdot L \end{Bmatrix}$$ Ecuación 11

Así, una línea está completamente especificada por las coordenadas

$$\left(\overrightarrow{S_i}\,;\overrightarrow{S_{0i}}\right) = (L, M, N\,; P, Q, R)$$ Ecuación 12

las cuales se refieren a las coordenadas Plücker. Nótese que el vector $\vec{S}$ siempre debe ser especificado como un vector unidad.

Un plano general en el espacio se ilustra en la Ilustración 28. Los parámetros requeridos para especificar completamente un plano ubicado en una ubicación y orientación fijas en el espacio son los siguientes:

$\overrightarrow{S_i}$ (L, M, N) → Dirección del vector normal al plano.
$\vec{r_i}$ (x_i, y_i, z_i) → Un punto cualquiera dado sobre el plano.

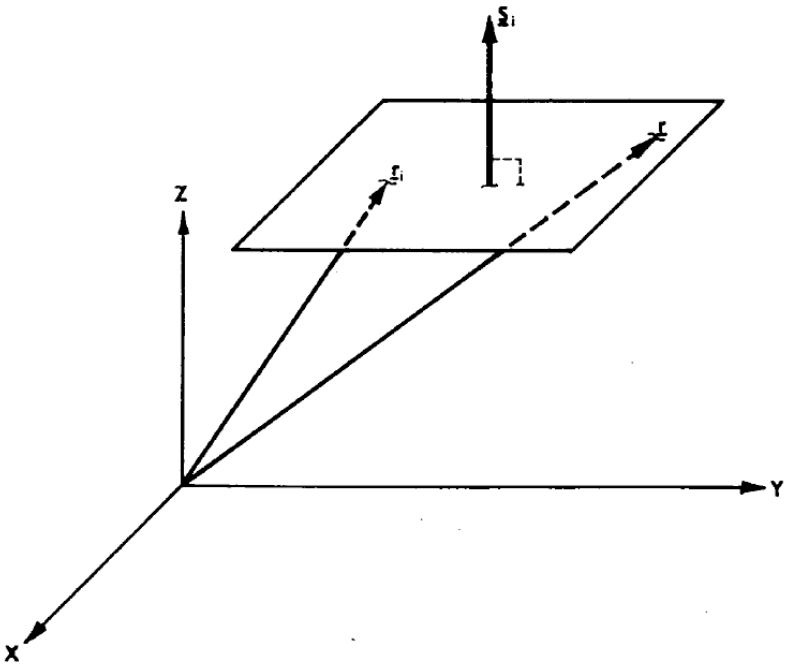


Ilustración 28 - Definición de un Plano en el espacio

Considerando un punto cualquiera sobre el plano, $\vec{r}$(x,y,z), la siguiente ecuación de producto escalar es obviamente válida:

$$(\vec{r} - \vec{r_i}) \cdot \overrightarrow{S_i} = 0$$ Ecuación 13

Reescribiendo esta ecuación se obtiene la ecuación de un plano:

$$\vec{r} \cdot \vec{S_i} = \vec{r_i} \cdot \vec{S_i} = S_{0i}$$ Ecuación 14

Donde S_{0i} es una cantidad escalar definida por los datos dados. Ésta es una ecuación escalar que se puede reescribir en una forma algebraica más familiar, que es la siguiente:

$$L \cdot x + M \cdot y + N \cdot z = S_{0i}$$ Ecuación 15

Los cuatro parámetros que especifican un plano en el espacio son $(\vec{S_i}\,; S_{0i})$, donde $\vec{S_i}$ está siempre escrito como un vector unidad.

9.5 Análisis de Línea adyacente

Las coordenadas Plucker de una línea $(\vec{S_i}\,; \vec{S_{0i}})$ para cada eje se calculan fácilmente de los datos de las líneas $(\vec{r_i}\,; \vec{S_i})$ obtenidos mediante la técnica experimental del Círculo de Puntos. Cada par de líneas adyacentes j y k (por ejemplo, los ejes 1-2, 2-3,…, (n-1)n) se deben analizar para dar los parámetros geométricos requeridos (α_{jk}, a_{jk}, d_j, d_k, θ_j, θ_k), para cada articulación. Esta sección presenta una solución analítica completa para el problema de líneas adyacentes para todos los casos posibles. Como repaso, la definición matemática precisa del ángulo de giro α, requerido para este análisis, se repite debajo:

$$\cos \alpha_{jk} = \vec{S_j} \cdot \vec{S_k}$$ Ecuación 16

$$\operatorname{sen} \alpha_{jk} = \vec{a_{jk}} \cdot \vec{S_j} \times \vec{S_k}$$ Ecuación 17

La Ilustración 29 muestra el caso general de dos líneas o ejes adyacentes en el espacio. Todos los datos requeridos se pueden determinar a partir de las coordenadas de las líneas $L_j(\vec{S_j}\,; \vec{S_{0j}})$ y $L_k(\vec{S_k}\,; \vec{S_{0k}})$ y los puntos de intersección $\vec{P_j}$ y $\vec{P_k}$ de la línea que es mutuamente perpendicular. Nótese que una par de líneas no paralelas sólo tienen una única línea que las intercepta y es perpendicular a ambas. Recapitulando, las ecuaciones de cada línea son:

$$\vec{r_j} \times \vec{S_j} = \vec{P_j} \times \vec{S_j} = \vec{S_{0j}}$$ Ecuación 18

$$\vec{r_k} \times \vec{S_k} = \vec{P_k} \times \vec{S_k} = \vec{S_{0k}}$$ Ecuación 19

Los puntos de intersección obviamente dan de inmediato la longitud de la articulación a_{jk} y el vector dirección de la articulación $\vec{a_{jk}}$. Los valores de desplazamiento se proveen también. Por ejemplo, en el eje *j*, el valor de desplazamiento d_j es la distancia entre el punto de intersección $\vec{P_j^{prev}}$, el cual se determina analizando la articulación *ij*, y $\vec{P_j^{sig}}$, determinado analizando la articulación *j-k*. Así el punto principal de este análisis es el cálculo de los puntos de intersección dado que estos proveen casi todos los parámetros requeridos.

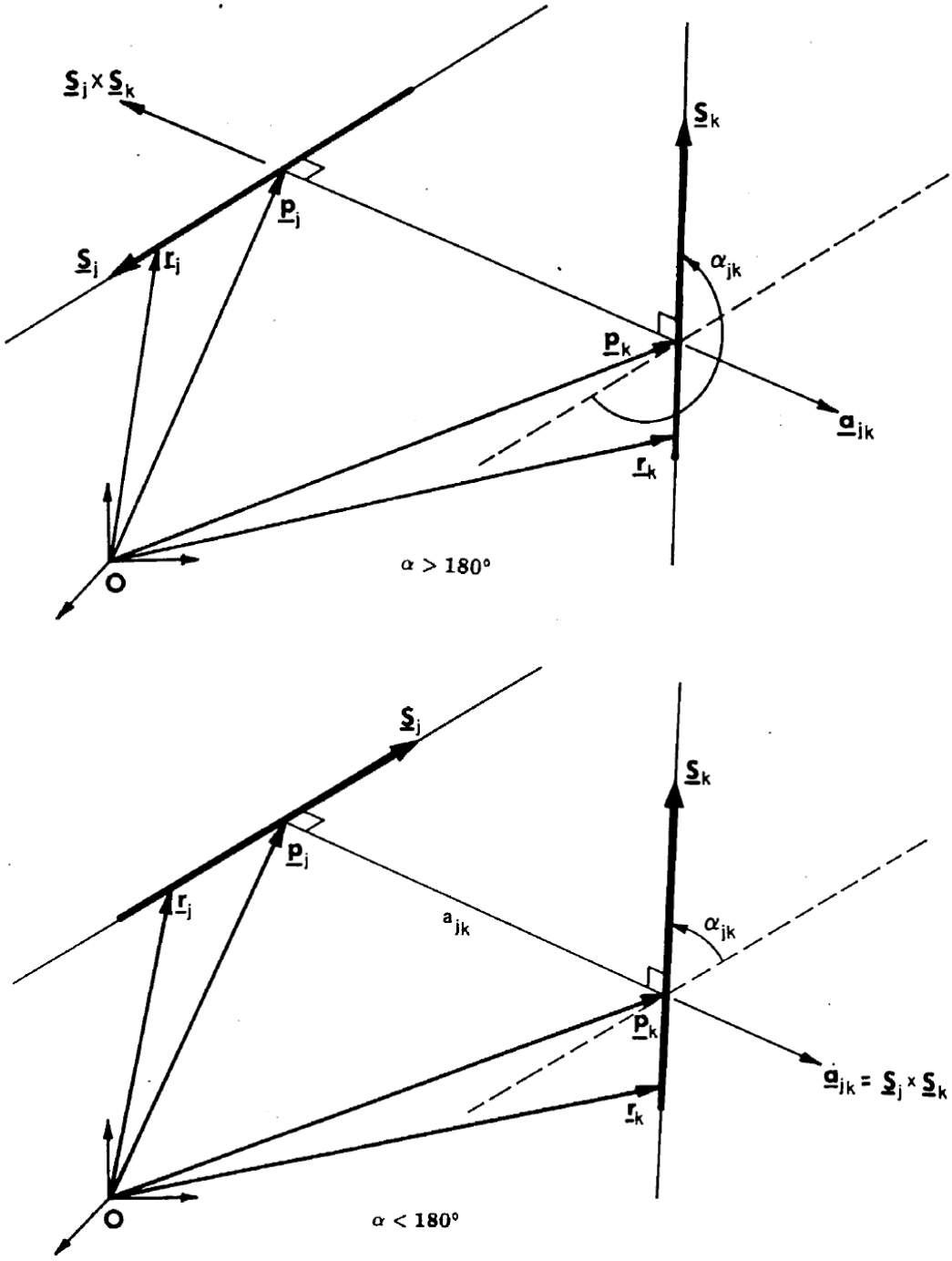


Ilustración 29 - Casos generales de Líneas Adyacentes

Antes de continuar, los siguientes dos vectores identidad (usados extensamente en este análisis) se definen con el propósito de su revisión. Considerando tres vectores generales $\vec{a}, \vec{b}, \vec{c}$, el triple producto escalar es:

$$\vec{a} \cdot (\vec{b} \times \vec{c}) = (\vec{a} \times \vec{b}) \cdot \vec{c} = (\vec{c} \times \vec{a}) \cdot \vec{b}$$ Ecuación 20

Y el triple producto vector identidad es:

$$\vec{a} \times (\vec{b} \times \vec{c}) = (\vec{a} \cdot \vec{c})\vec{b} - (\vec{a} \cdot \vec{b})\vec{c}$$ **Ecuación 21**

La ecuación principal usada para determinar las condiciones iniciales de las dos líneas (paralelas, con intersección, u oblicuas) es llamada como la ecuación del Momento Mutuo (MM). Esta se deriva de expandir las expresiones del momento mutuo de cada línea sobre la otra, las cuales vienen dadas por:

$$\left((\overrightarrow{P_k} - \overrightarrow{P_j}) \times \overrightarrow{S_k}\right) \cdot \overrightarrow{S_j} \qquad (momento\ de\ \overrightarrow{S_k}\ sobre\ \overrightarrow{S_j})$$ **Ecuación 22**

$$\left((\overrightarrow{P_j} - \overrightarrow{P_k}) \times \overrightarrow{S_j}\right) \cdot \overrightarrow{S_k} \qquad (momento\ de\ \overrightarrow{S_j}\ sobre\ \overrightarrow{S_k})$$ **Ecuación 23**

Expandiendo sólo la primera expresión y substituyendo la ecuación de la línea *j* se tiene:

$$\left((\overrightarrow{P_k} - \overrightarrow{P_j}) \times \overrightarrow{S_k}\right) \cdot \overrightarrow{S_j} = \overrightarrow{P_k} \times \overrightarrow{S_k} \cdot \overrightarrow{S_j} - \overrightarrow{P_j} \times \overrightarrow{S_k} \cdot \overrightarrow{S_j} = \overrightarrow{S_{0k}} \cdot \overrightarrow{S_j} + \overrightarrow{S_k} \times \overrightarrow{P_j} \cdot \overrightarrow{S_j}$$ **Ecuación 24**

Substituyendo el triple producto escalar identidad en el segundo término del lado derecho, la ecuación de la línea *k* queda así:

$$\left((\overrightarrow{P_k} - \overrightarrow{P_j}) \times \overrightarrow{S_k}\right) \cdot \overrightarrow{S_j} = \overrightarrow{P_k} \times \overrightarrow{S_k} \cdot \overrightarrow{S_j} - \overrightarrow{P_j} \times \overrightarrow{S_k} \cdot \overrightarrow{S_j} = \overrightarrow{S_{0k}} \cdot \overrightarrow{S_j} + \overrightarrow{S_k} \times \overrightarrow{P_j} \cdot \overrightarrow{S_j}$$ **Ecuación 25**

$$\boxed{MM = \overrightarrow{S_{0k}} \cdot \overrightarrow{S_j} + \overrightarrow{S_k} \cdot \overrightarrow{S_{0j}}}$$ **Ecuación 26**

Esta ecuación final (Ecuación 26) da el momento mutuo en términos de las coordenadas de la línea únicamente. Expandiendo la expresión del segundo momento mutuo arriba quedaría el mismo resultado. Por tanto, el momento de cada línea sobre la otra es idéntico, como el término momento mutuo. El vector formado por los puntos de intersección es en realidad el vector dirección de la articulación:

$$\overrightarrow{P_k} - \overrightarrow{P_j} = a_{jk} \cdot \overrightarrow{a_{jk}} = \overrightarrow{a'_{jk}}$$ **Ecuación 27**

Substituyendo esta expresión en la expresión del primer momento mutuo se tiene lo siguiente:

$$a_{jk} \cdot \overrightarrow{a_{jk}} \times \overrightarrow{S_k} \cdot \overrightarrow{S_j} = \overrightarrow{a'_{jk}} \cdot \overrightarrow{S_k} \times \overrightarrow{S_j} = -\overrightarrow{a'_{jk}} \cdot \overrightarrow{S_j} \times \overrightarrow{S_k}$$ **Ecuación 28**

La cual es también equivalente al momento mutuo. Substituyendo la definición del ángulo de giro, ver Ecuación 17, e igualando con la expresión del momento mutuo queda la ecuación esperada del momento mutuo:

$$\overrightarrow{S_J} \cdot \overrightarrow{S_{0k}} + \overrightarrow{S_k} \cdot \overrightarrow{S_{0J}} = -a_{jk} \cdot sen(\alpha_{jk})$$ **Ecuación 29**

Nótese que el momento mutuo indica si el ángulo de giro es menor o mayor que 180. Dado que a_{jk} es una longitud real que es positiva siempre, la anterior Ecuación 29 nos da las siguientes indicaciones:

$$\alpha_{jk} > 180° \; cuando \; MM > 0$$

$$\alpha_{jk} < 180° \; cuando \; MM < 0$$

La ecuación del momento mutuo (Ecuación 29) indica la condición de las dos líneas. Cuando $MM \neq 0$, entonces se tiene que $a_{jk} \neq 0$ y $\alpha_{jk} \neq 0$, y hay una distancia finita y un ángulo entre las líneas y ejes. Por tanto, las líneas deben ser oblicuas entre sí. Cuando $MM=0$, las líneas se cruzan una con otra si $a_{jk}=0$, o bien son paralelas si $\alpha_{jk}=0$. Los puntos de intersección $\overrightarrow{P_J}$ y $\overrightarrow{P_k}$ se determinan usando una única derivación para cada condición.

Los puntos de intersección para ciertas condiciones se deben escribir en términos del vector unidad $\overrightarrow{a_{Jk}}$. Por tanto, una expresión para este vector únicamente en términos de las coordenadas de la línea se debe derivar primero. Como se ve en la Ilustración 29, los puntos $\overrightarrow{a_{Jk}}$ están en la misma dirección que $\overrightarrow{S_J} \times \overrightarrow{S_k}$, si $\alpha_{jk}<180°$, y en la dirección opuesta cuando $\alpha_{jk}>180°$. Por consiguiente, el vector $\overrightarrow{a_{Jk}}$ se puede escribir como sigue:

$$\overrightarrow{a_{Jk}} = \pm \frac{\overrightarrow{S_J} \times \overrightarrow{S_k}}{|\overrightarrow{S_J} \times \overrightarrow{S_k}|}$$ **Ecuación 30**

De la ecuación del Momento Mutuo, se puede ver que si $\alpha_{jk}<180°$, entonces MM<0, y viceversa. Por tanto, el signo en la anterior ecuación es simplemente el opuesto del signo algebraico de MM. Así, el vector $\overrightarrow{a_{Jk}}$ en la ecuación anterior se puede reescribir así:

$$\overrightarrow{a_{Jk}} = -sgn(\overrightarrow{S_J} \cdot \overrightarrow{S_{0k}} + \overrightarrow{S_k} \cdot \overrightarrow{S_{0J}}) \cdot \frac{\overrightarrow{S_J} \times \overrightarrow{S_k}}{|\overrightarrow{S_J} \times \overrightarrow{S_k}|}$$ **Ecuación 31**

Donde la función sgn representa el signo algebraico. Nótese que a lo largo de este capítulo la siguiente notación que se sigue:

$$|\vec{a}| \rightarrow longitud \; del \; vector$$

$$|a| \rightarrow valor \; absoluto \; de \; una \; cantidad \; escalar$$

La derivación de los puntos de intersección para cada una de los pares de condiciones se da a continuación.

9.5.1 Condición 1: Líneas oblicuas (*MM≠0*)

Los puntos de intersección $\overrightarrow{P_j}$ y $\overrightarrow{P_k}$ para las líneas oblicuas se derivan a través de una técnica directa, por considerar las observaciones geométricas que se ven a continuación. El vector $\overrightarrow{a_{jk}}$ y su línea perpendicular mutuamente asociada están en el plano conteniendo $\overrightarrow{a_{jk}}$ y $\overrightarrow{S_k}$. La línea formada por el eje j se cruza con la línea perpendicular mutua y así como al plano en el punto de intersección $\overrightarrow{P_j}$. Asimismo el punto de intersección $\overrightarrow{P_k}$ es la intersección de la línea k y el plano formado por $\overrightarrow{S_j}$ y por $\overrightarrow{a_{jk}}$. El plano formado por $\overrightarrow{a_{jk}}$ y $\overrightarrow{S_k}$ se usa para determinar $\overrightarrow{P_j}$. La normal a este plano es:

$$\overrightarrow{n_k} = \overrightarrow{a_{jk}} \times \overrightarrow{S_k}$$ **Ecuación 32**

La ecuación del plano viene dada por:

$$\overrightarrow{r_k} \cdot \overrightarrow{n_k} = n_k$$ **Ecuación 33**

Donde $\overrightarrow{r_k}$ es un punto arbitrario en la línea k y obviamente está sobre el plano. Substituyendo en la Ecuación 32 y calculando se obtiene lo siguiente:

$$\overrightarrow{r_k} \cdot (\overrightarrow{a_{jk}} \times \overrightarrow{S_k}) = n_k$$ **Ecuación 34**

$$-\overrightarrow{r_k} \cdot (\overrightarrow{S_k} \times \overrightarrow{a_{jk}}) = n_k$$ **Ecuación 35**

Substituyendo el producto escalar identidad y la ecuación de la línea k queda:

$$\left(-\overrightarrow{r_k} \times \overrightarrow{S_k}\right) \cdot \overrightarrow{a_{jk}} = n_k$$ **Ecuación 36**

$$-\overrightarrow{S_{0k}} \cdot \overrightarrow{a_{jk}} = n_k$$ **Ecuación 37**

En general, el punto de intersección $\overrightarrow{P_j}$ del plano $(\overrightarrow{n_k}; n_k)$ y la línea $(\overrightarrow{S_j}; \overrightarrow{S_{0j}})$ vienen dados como:

$$\overrightarrow{P_k} = \frac{\overrightarrow{n_k} \times \overrightarrow{S_{0j}} + n_k \overrightarrow{S_j}}{\overrightarrow{n_k} \cdot \overrightarrow{S_j}}$$ **Ecuación 38**

Este resultado general se obtiene mediante la manipulación de la ecuación de la línea y substituyendo la ecuación del plano. Este análisis no se mostrará aquí. Substituyendo la expresión para $\overrightarrow{n_k}$ y n_k dada anteriormente en la Ecuación 29 y en la Ecuación 37, dentro de la Ecuación 38, se tiene:

$$\overrightarrow{P_J} = \frac{(\overrightarrow{a_{Jk}} \times \overrightarrow{S_k}) \times \overrightarrow{S_{0J}} - (\overrightarrow{S_{0k}} \cdot \overrightarrow{a_{Jk}})\overrightarrow{S_J}}{(\overrightarrow{a_{Jk}} \times \overrightarrow{S_k}) \cdot \overrightarrow{S_J}}$$ Ecuación 39

La cual es la expresión final para el punto de intersección. El punto de intersección $\overrightarrow{P_k}$ se deriva de manera análoga, calculando la intersección del plano formado por $\overrightarrow{a_{Jk}}$ y $\overrightarrow{S_J}$ y la línea k. La expresión resultante es:

$$\overrightarrow{P_k} = \frac{(\overrightarrow{a_{Jk}} \times \overrightarrow{S_k}) \times \overrightarrow{S_{0k}} - (\overrightarrow{S_{0J}} \cdot \overrightarrow{a_{Jk}})\overrightarrow{S_k}}{(\overrightarrow{a_{Jk}} \times \overrightarrow{S_J}) \cdot \overrightarrow{S_k}}$$ Ecuación 40

9.5.2 Condición 2: Líneas que se cruzan (MM=0, a_{jk}=0)

Las ecuaciones para la derivada de los puntos de la intersección para la condición arriba indicada, que son la Ecuación 39 y la Ecuación 40, no están en una forma simplificada ni conveniente para esta condición. Se necesita una ecuación alternativa, la cual conduzca a una expresión altamente simplificada. La única ecuación de vector disponible es:

$$\overrightarrow{a'_{Jk}} = \overrightarrow{P_k} - \overrightarrow{P_J}$$ Ecuación 41

La cual se puede reescribir como:

$$\overrightarrow{P_k} = \overrightarrow{a'_{Jk}} + \overrightarrow{P_J}$$ Ecuación 42

Esta ecuación se manipula tomando el producto cruzado de cada lado con el vector $\overrightarrow{S_k}$, el cual nos da lo siguiente:

$$\overrightarrow{P_k} \times \overrightarrow{S_k} - \overrightarrow{a'_{Jk}} \times \overrightarrow{S_k} = \overrightarrow{P_J} \times \overrightarrow{S_k}$$ Ecuación 43

Y substituyendo de la ecuación de la línea k nos da:

$$\overrightarrow{S_{0k}} - \overrightarrow{a'_{Jk}} \times \overrightarrow{S_k} = \overrightarrow{P_J} \times \overrightarrow{S_k}$$ Ecuación 44

Tomando el producto cruzado de cada lado de la ecuación con un vector el cual es perpendicular a $\overrightarrow{P_J}$, que lo es a $\overrightarrow{S_{0J}}$, nos conduce a una solución $\overrightarrow{P_J}$ que es como sigue:

$$\overrightarrow{S_{0J}} \times \overrightarrow{S_{0k}} - \overrightarrow{S_{0J}} \times (\overrightarrow{a'_{Jk}} \times \overrightarrow{S_k}) = \overrightarrow{S_{0J}} \times (\overrightarrow{P_J} \times \overrightarrow{S_k})$$ Ecuación 45

Substituyendo el triple producto identidad nos da la siguiente expresión para el lado derecho de la ecuación:

$$\overrightarrow{S_{0J}} \times (\overrightarrow{P_J} \times \overrightarrow{S_k}) = (\overrightarrow{S_{0J}} \cdot \overrightarrow{S_k})\overrightarrow{P_J} - (\overrightarrow{S_{0J}} \cdot \overrightarrow{P_J})\overrightarrow{S_k} = (\overrightarrow{S_{0J}} \cdot \overrightarrow{S_k})\overrightarrow{P_J}$$ Ecuación 46

Dado que $\overrightarrow{S_{0J}} \cdot \overrightarrow{P_J} = 0$. Substituyendo esta expresión en la Ecuación 45, y resolviendo para $\overrightarrow{P_J}$ nos da la expresión final:

$$\vec{P_j} = \frac{\overrightarrow{S_{0j}} \times \overrightarrow{S_{0k}} - \overrightarrow{S_{0j}} \times (\overrightarrow{a'_{jk}} \times \overrightarrow{S_k})}{(\overrightarrow{S_{0j}} \cdot \overrightarrow{S_k})}$$ Ecuación 47

Para esta condición las líneas se cruzan y $\overrightarrow{a'_{jk}} = 0$, obteniéndose una expresión simplificada para el punto de cruce. El proceso completo se puede repetir tomando el producto cruzado de la ecuación original (véase la Ecuación 42), con $\overrightarrow{S_j}$ y después con $\overrightarrow{S_{0k}}$, el cual conducirá a una expresión idéntica para $\overrightarrow{P_k}$, que es lo que se esperaba. El punto de intersección común viene dado por la ecuación:

$$\vec{P} = \vec{P_j} = \vec{P_k} = \frac{\overrightarrow{S_{0j}} \times \overrightarrow{S_{0k}}}{\overrightarrow{S_{0j}} \cdot \overrightarrow{S_k}} = \frac{\overrightarrow{S_{0k}} \times \overrightarrow{S_{0j}}}{\overrightarrow{S_j} \cdot \overrightarrow{S_{0k}}}$$ Ecuación 48

Los denominadores en la ecuación antes mostrada son iguales y opuestos, dado que MM=0, y por tanto las expresiones son equivalentes.

Hay un problema con la definición del vector dirección $\overrightarrow{a_{jk}}$ de la articulación, para esta condición. La definición expone los puntos $\overrightarrow{a_{jk}}$ en la dirección del eje j al eje k, cuyo enrevesamiento no tiene sentido en el caso de líneas cruzadas. Por tanto la convención estándar que se seguirá para esta condición será que $\overrightarrow{a_{jk}}$ está en la dirección de $\overrightarrow{S_j} \cdot \overrightarrow{S_k}$, y así $\alpha < 180°$, siempre.

Uno podría inmediatamente preguntarse por qué esta derivación no se usó para la condición de líneas oblicuas. La razón es que la presenta una indeterminación, o tiende a infinito cuando $\overrightarrow{S_{0j}} \cdot \overrightarrow{S_k} = 0$. La ecuación para $\overrightarrow{P_k}$ que tenga la misma forma, presenta problemas idénticos cuando $\overrightarrow{S_j} \cdot \overrightarrow{S_{0k}} = 0$. Ninguno de estos casos representa una condición especial o particular. Estos suceden para muchas combinaciones de líneas estándar oblicuas, y así las ecuaciones no se pueden usar para el caso de líneas oblicuas.

9.5.3 Condición 3: Líneas paralelas (MM=0, sin α_{jk}=0)

Para el caso de líneas paralelas únicas, los puntos de cruce mutuamente perpendiculares no existen. Hay un número infinito de líneas perpendiculares mutuas conectando dos líneas paralelas. Sin embargo, el vector $\overrightarrow{a_{jk}}$ se debe determinar para calcular la longitud de la articulación $\overrightarrow{a_{jk}}$. En esta condición, $\cos\alpha_{jk} = \pm 1$, y los vectores dirección son paralelos pero deben estar en direcciones opuestas. Por tanto se tiene:

$$\overrightarrow{S_j} = \cos\alpha_{jk}\,\overrightarrow{S_k}$$ Ecuación 49

O bien

$$\overrightarrow{S_k} = \cos\alpha_{jk}\,\overrightarrow{S_j}$$ Ecuación 50

La ecuación que relaciona cualquier par de puntos que se cruzan se escribe como en la condición anteriormente vista, que sería la Ecuación 42 ($\overrightarrow{P_k} = \overrightarrow{a'_{jk}} + \overrightarrow{P_j}$).

Tomando el producto cruzado con el vector $\overrightarrow{S_j}$ queda:

$$\overrightarrow{P_j} \times \overrightarrow{S_j} + \overrightarrow{a'_{jk}} \times \overrightarrow{S_j} = \overrightarrow{P_k} \times \overrightarrow{S_j}$$ **Ecuación 51**

Substituyendo las ecuaciones de las líneas y en la Ecuación 49 queda la siguiente ecuación:

$$\overrightarrow{S_{0j}} + \overrightarrow{a'_{jk}} \times \overrightarrow{S_j} = \cos\alpha_{jk}\,\overrightarrow{S_{0k}}$$ **Ecuación 52**

Donde los puntos de intersección no-únicos han quedado eliminados. La solución para $\overrightarrow{a'_{jk}}$ se consigue tomando el producto cruzado de la ecuación con un vector perpendicular a $\overrightarrow{a'_{jk}}$, el cual $\overrightarrow{S_j}$, quedando como sigue:

$$\overrightarrow{S_j} \times \overrightarrow{S_{0j}} + \overrightarrow{S_j} \times (\overrightarrow{a'_{jk}} \times \overrightarrow{S_j}) = \overrightarrow{S_j} \times \cos\alpha_{jk}\,\overrightarrow{S_{0k}}$$ **Ecuación 53**

El término del triple producto escalar se expande como sigue:

$$\overrightarrow{S_j} \times (\overrightarrow{a'_{jk}} \times \overrightarrow{S_j}) = (\overrightarrow{S_j} \cdot \overrightarrow{S_j})\overrightarrow{a'_{jk}} - (\overrightarrow{S_j} \cdot \overrightarrow{a'_{jk}})\overrightarrow{S_j} = \overrightarrow{a'_{jk}}$$ **Ecuación 54**

Dado que $\overrightarrow{S_j}$ es un vector unidad perpendicular a $\overrightarrow{a'_{jk}}$. Substituyendo dentro de la Ecuación 53 queda el resultado explícito para $\overrightarrow{a'_{jk}}$.

$$\overrightarrow{a'_{jk}} = \cos\alpha_{jk}\,\overrightarrow{S_j} \times \overrightarrow{S_{0k}} - \overrightarrow{S_j} \times \overrightarrow{S_{0j}}$$ **Ecuación 55**

$$\overrightarrow{a'_{jk}} = \overrightarrow{S_j} \times (\cos\alpha_{jk}\,\overrightarrow{S_{0k}} - \overrightarrow{S_{0j}})$$

La magnitud de este vector nos da la longitud de la articulación j_{jk}.

Calcular los desplazamientos en este caso presenta un problema. Dado que no hay unos puntos de intersección únicos que conecten las líneas j y k, los desplazamientos d_j y d_k no se pueden calcular. Sin embargo, en el eje j, realmente existe un punto de intersección de la línea perpendicular común al eje previo i, que será$\overrightarrow{P_j^{prev}}$. En el eje k, existe un punto de intersección de la línea perpendicular común al siguiente eje l, $\overrightarrow{P_k^{prox}}$ (los puntos $\overrightarrow{P_j^{prox}}$ y $\overrightarrow{P_j^{prev}}$ son los puntos que no existen). La distancia entre estos puntos medida a lo largo de la dirección común $\overrightarrow{S_j}$ o $\overrightarrow{S_k}$ representa la suma de los desplazamientos individuales d_j+d_k. Así, para el caso de líneas paralelas, esta suma se calcula en lugar de los desplazamientos individuales. La posición de la articulación final está completamente determinada usando la suma de los desplazamientos.

Las tres soluciones anteriores representan una base teórica completa para analizar líneas adyacentes en el espacio, para cualquier condición posible. Desafortunadamente, esta es una aproximación puramente teórica, la cual muestra fuertes limitaciones in la práctica real para el caso de líneas oblicuas que son casi paralelas. Este es exactamente el caso los brazos robot que tienen ejes adyacentes nominalmente paralelos, lo que sucede en la gran mayoría de los sistemas disponibles actualmente. La próxima sección muestra tres aproximaciones diferentes para manejar este problema.

9.6 Análisis de ejes nominalmente paralelos

Los ejes adyacentes nominalmente paralelos, situación que ocurre en muchos robots, no serán exactamente paralelos debido a errores de producción y ensamblado. Como se mencionó anteriormente, esto causa problemas numéricos difíciles en ciertos casos, que algunos investigadores han solucionado añadiendo un parámetro cinemático extra. La técnica del Círculo de Puntos, a través del uso del análisis de geometría de líneas, se ha extendido para manejar esta dificultad. Se han desarrollado tres métodos alternativos. La desventaja de añadir un parámetro cinemático extra al modelo de geometría del robot es que requiere grandes modificaciones al programa controlador, haciéndolo difícil de implementar en sistemas actuales. Dos de los métodos presentados aquí evitan esta dificultad, y no requieren un parámetro extra en el modelo.

El problema numérico con ejes casi paralelos se muestra en las figuras siguientes. El vector $\overrightarrow{S_k^{nom}}$ representa exactamente la condición nominal o paralela, y $\overrightarrow{S_k^{real}}$ es la orientación actual de la línea k. Todo el error es modelado como una traslación y orientación de la línea k con respecto a la línea j, y por tanto j sólo tiene una posición nominal y exacta. En la Ilustración 30 el error, d_α, representa un giro solamente sobre la línea nominal perpendicular. Esto no provoca ninguna dificultad. Tanto el $\overrightarrow{a_{jk}}$ como los puntos de intersección ($\overrightarrow{P_j}$ y $\overrightarrow{P_k}$ no mostrados) permanecen sin cambios del caso nominal. El único error en los parámetros de la articulación sucede en el ángulo de giro α. Esto se identifica fácilmente en casi todas las técnicas. Desafortunadamente, no hay garantías de que los ejes estarán descolocados en esta forma ideal.

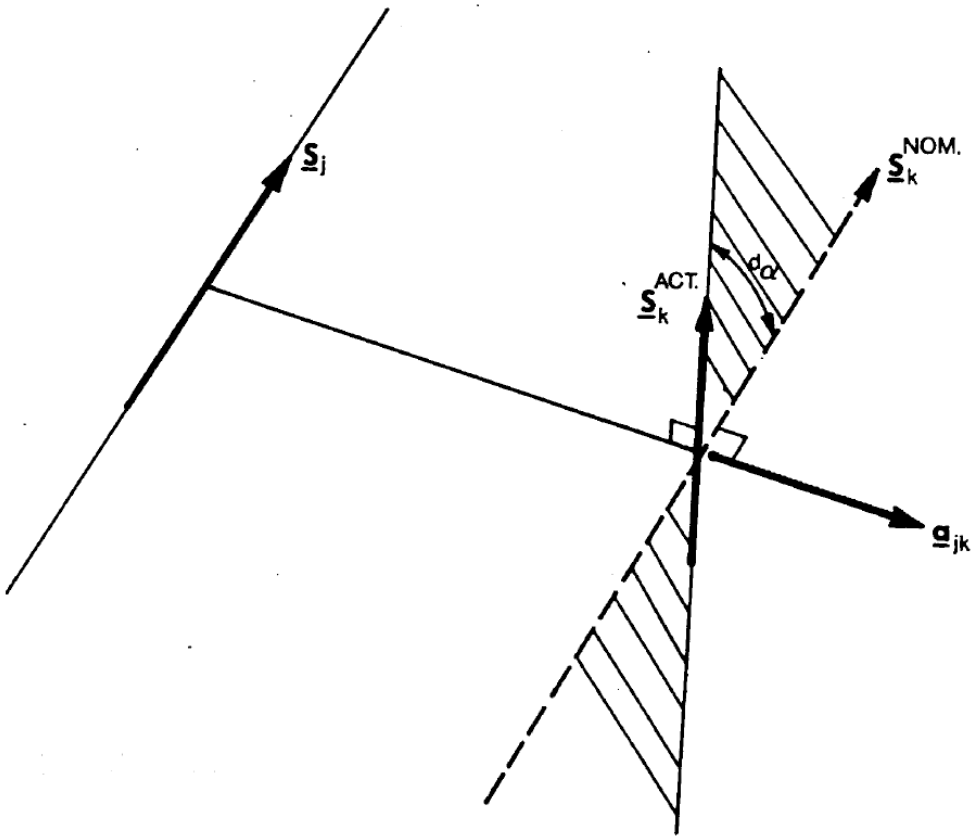


Ilustración 30 - Ejes adyacentes paralelos - Caso Ideal

En lugar del error, d_α, puede ser debido a un giro del eje k alrededor de la dirección normal al plano original que contiene los ejes nominalmente paralelos. Esto se muestra en la Ilustración 31. El

desplazamiento del eje k sucede en el plano original. Esto es que los 3 vectores $\overrightarrow{S_j}$, $\overrightarrow{S_k^{real}}$ y $\overrightarrow{S_k^{nom}}$ están sobre el plano. El efecto de esto es que las líneas reales se cruzan y $\overrightarrow{a_{jk}}$va desde una longitud finita hasta 0. El vector dirección de la articulación real, $\overrightarrow{a_{jk}^{real}}$ (perpendicular a los vectores $\vec{S}$) se convierte en perpendicular a la dirección nominal original $\overrightarrow{a_{jk}^{nom}}$. Un problema añadido es que el error d_α es normalmente pequeño (menor que unos pocos grados) y así las líneas se cruzan casi en el infinito, provocando que los valores de desplazamiento d_j y d_k vayan desde un valor finito hasta el infinito. Así todos los parámetros de las articulaciones cambian drásticamente debido a unos errores muy pequeños en el radio de giro. La extrema sensibilidad numérica hace difícil determinar de forma preciso los parámetros. En la práctica, normalmente se da una combinación del caso ideal y del peor caso. Esto es, que el error es debido a tanto un giro alrededor de $\overrightarrow{a_{jk}^{nom}}$ como a otro giro alrededor de la dirección normal al plano original. Los métodos para manejar esta dificultad se presentan a continuación.

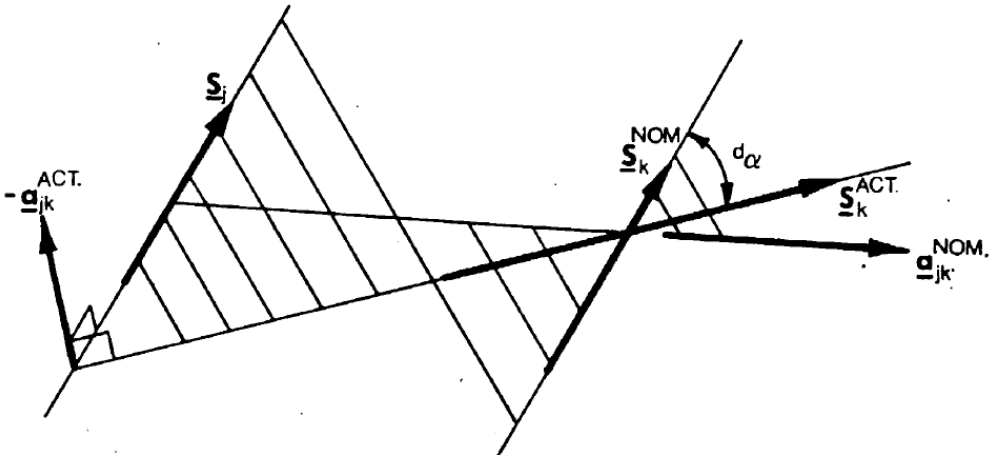


Ilustración 31 - Ejes adyacentes paralelos - Peor Caso

9.6.1 Solución 1: Análisis exacto

El método más directo es tener confianza en la medición y en la precisión del cálculo numérico, y asumir que los parámetros exactos para las líneas adyacentes, sin importar los problemas de sensibilidad numérica, se pueden calcular. Esto significa simplemente tratar las líneas como líneas oblicuas o que se cruzan en caso extremo. Los métodos desarrollados en la sección previa se usan para este análisis. Si esto se puede conseguir de forma precisa, todavía queda el problema con la determinación de los ángulos de la articulación medidos por el transductor. El transductor mide los ángulos de la articulación (ángulos entre los vectores de la articulación $\vec{a}$) con respecto al vector nominal $\overrightarrow{a_{jk}^{nom}}$. Sin embargo, los parámetros de la articulación calibrada determinados representan un modelo en el cual la dirección de la articulación real $\overrightarrow{a^{real}}$ está desplazada de $\overrightarrow{a^{nom}}$ por una cantidad significativa. Por tanto, los valores del transductor medidos se deben corregir.

Los ángulos de corrección que se deben añadir a los valores del transductor, cuando se usa el modelo exacto, se determinan fácilmente. Nótese, que dado que $\overrightarrow{a_{jk}}$ se usa para calcular tanto θ_j como θ_k, ambos ángulos se deben corregir. El ángulo final para el eje j requerido para calcular la posición del robot es el ángulo cinemático real θ_j^{real}, que es el ángulo entre $\overrightarrow{a_{ij}}$ y $\overrightarrow{a^{real}}$. Dado que el ángulo nominal θ_j^{nom}, que es el ángulo entre $\overrightarrow{a_{ij}}$ y $\overrightarrow{a_{jk}^{nom}}$, se conoce, la corrección es el ángulo entre $\overrightarrow{a_{jk}^{nom}}$ y $\overrightarrow{a_{jk}^{real}}$. Estos

ángulos se ilustran en la Ilustración 32. La definición matemática del ángulo de corrección es la siguiente:

$$\sin \phi_j^C = \overrightarrow{a_{jk}^{nom}} \times \overrightarrow{a_{jk}^{real}} \cdot \overrightarrow{S_j}$$ Ecuación 56

El problema aquí es que sólo $\overrightarrow{a_{jk}^{real}}$ se determina de la localización real de los ejes y vectores $\overrightarrow{S_j}$ y $\overrightarrow{S_k^{real}}$. La dirección nominal $\overrightarrow{a_{jk}^{nom}}$, que no tiene existencia física debe ser aproximada. Las soluciones 2 y 3 también hacen uso del vector $\overrightarrow{a_{jk}^{nom}}$. Una vez los ángulos de corrección están calculados, el ángulo cinemático real viene dado por la siguiente fórmula:

$$\theta_j^{real} = \theta_j^{nom} + \phi_j^C$$ Ecuación 57

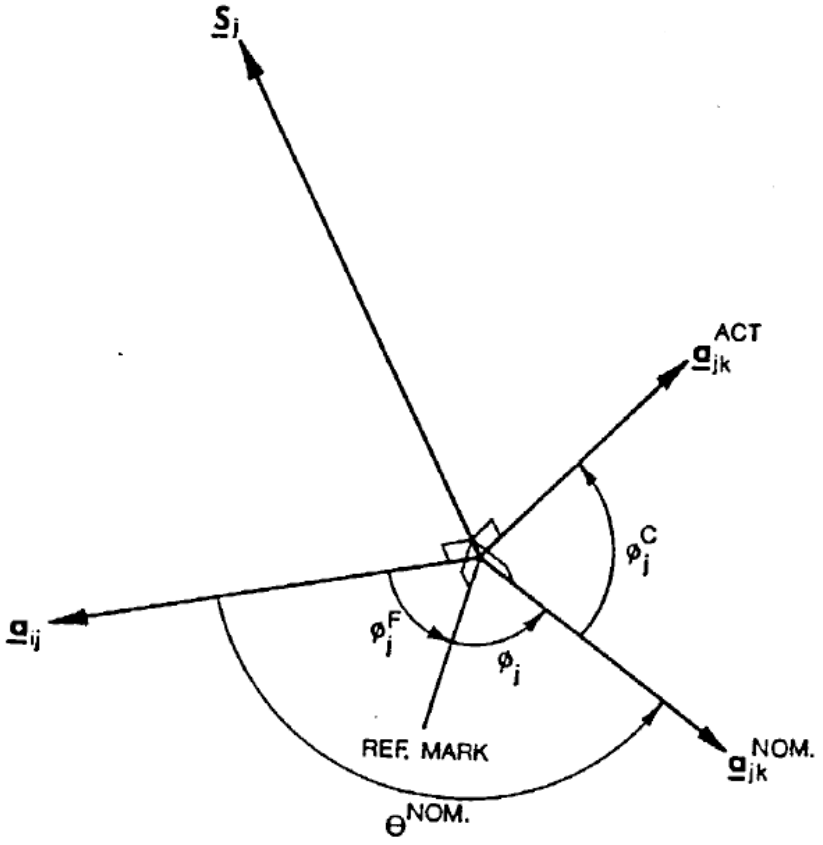


Ilustración 32 - Modelo exacto de corrección del ángulo del transductor para eje j

El ángulo nominal sin calibrar θ_j^{nom} depende del ángulo del transductor medido ϕ_j, y un ángulo dado de desplazamiento de la referencia fija ϕ_j^F, el cual se calcula como:

$$\theta_j^{nom} = \phi_j + \phi_j^F$$ Ecuación 58

El concepto de valores de desplazamiento del transductor, introducido por primera vez aquí, debería ser bien entendido. Los errores en la medida del transductor se pueden modelar como variaciones o correcciones de estos valores. El ángulo cinemático real final es simplemente el siguiente:

$$\theta_j^{real} = \phi_j + \phi_j^F + \phi_j^C$$ Ecuación 59

En la práctica, dado que ambos son en esencia constantes de calibración, ϕ_j^F y ϕ_j^C simplemente se combinan en un desplazamiento general para el modelo exacto.

El ángulo para el eje k se corrige de una forma similar usando la misma constante de corrección. Esta corrección se ilustra en la Ilustración 33. El ángulo cinemático real viene dado por la fórmula:

$$\theta_k^{real} = \theta_k^{nom} - \phi_j^C$$ Ecuación 60

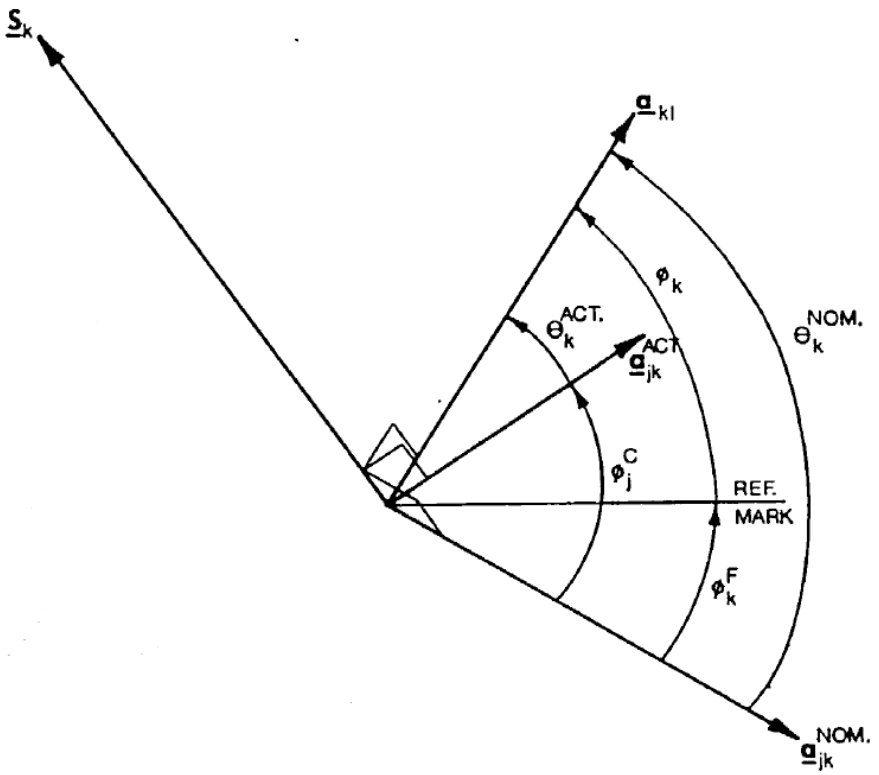


Ilustración 33 - Modelo exacto para corrección del ángulo del transductor para eje k

Otra vez, el ángulo nominal sin calibrar θ_k^{nom} depende del ángulo del transductor y un desplazamiento de la referencia fijo, y viene dado por:

$$\theta_k^{nom} = \phi_k + \phi_k^F$$ Ecuación 61

Y así el ángulo cinemático real final es:

$$\theta_k^{real} = \phi_k + \phi_k^F - \phi_j^C$$ Ecuación 62

Así, un modelo exacto se debe usar mientras los ángulos del transductor son correctos. Obviamente, para el caso ideal de errores en el ángulo de giro sólo, $\overrightarrow{a^{real}} = \overrightarrow{a_{jk}^{nom}}$, y así el ángulo de corrección es 0.

9.6.2 Solución 2: Análisis aproximado

El segundo método de manejar ejes nominalmente paralelos evita el problema de la sensibilidad numérica, mediante la aproximación del ángulo de giro a cero, y calcular un error o valor calibrado sólo para la longitud de la articulación $\overrightarrow{a_{jk}}$. Esto es que α_{jk} se modela o fuerza a ser 0 o π, y que se calcula la longitud real de la articulación $\overrightarrow{a_{jk}}$ para este modelo. Dado que se asume que las líneas son paralelas, los desplazamientos d_j y d_k no se pueden calcular mediante esta técnica. La longitud de la articulación $\overrightarrow{a_{ij}}$, que es el único parámetro calculado, se determina mediante el siguiente análisis. El único vector dirección usado para representar tanto a $\overrightarrow{S_j}$ como a $\overrightarrow{S_k}$ se calcula como una dirección media de los valores exactos de las direcciones de los ejes determinados en el experimento del Círculo de Puntos. Ambas líneas se modelan como apuntando en la dirección media, pero giradas sobre los puntos de intersección originales para el caso nominal, $\overrightarrow{P_j}$ y $\overrightarrow{P_k}$. El punto $\overrightarrow{P_j}$ se calcula del desplazamiento nominal (ya conocido) para los ejes j, d_j, y por tanto viene dado por:

$$\overrightarrow{P_j} = \overrightarrow{P_j^{prev}} + d_j\overrightarrow{S_j} \qquad \text{Ecuación 63}$$

Donde $\overrightarrow{P_j^{prev}}$ es el punto de intersección de la línea perpendicular común entre los ejes i y j. El punto $\overrightarrow{P_k}$, el cual representa el punto de intersección cuando j y k son exactamente paralelos, se calcula como la intersección de la línea real ($\overrightarrow{r_k}$, $\overrightarrow{S_k^{real}}$.) para el eje k, y un plano que es perpendicular a la línea real ($\overrightarrow{P_j}$, $\overrightarrow{S_j^{real}}$.) para el eje j. Los ejes aproximados o modelados se denotan por los vectores $\overrightarrow{S^{med}}$, que se muestran en la Ilustración 34([9]).

[9] En la figura se denota como $\overrightarrow{S^{ave}}$, del inglés *average* (promedio).

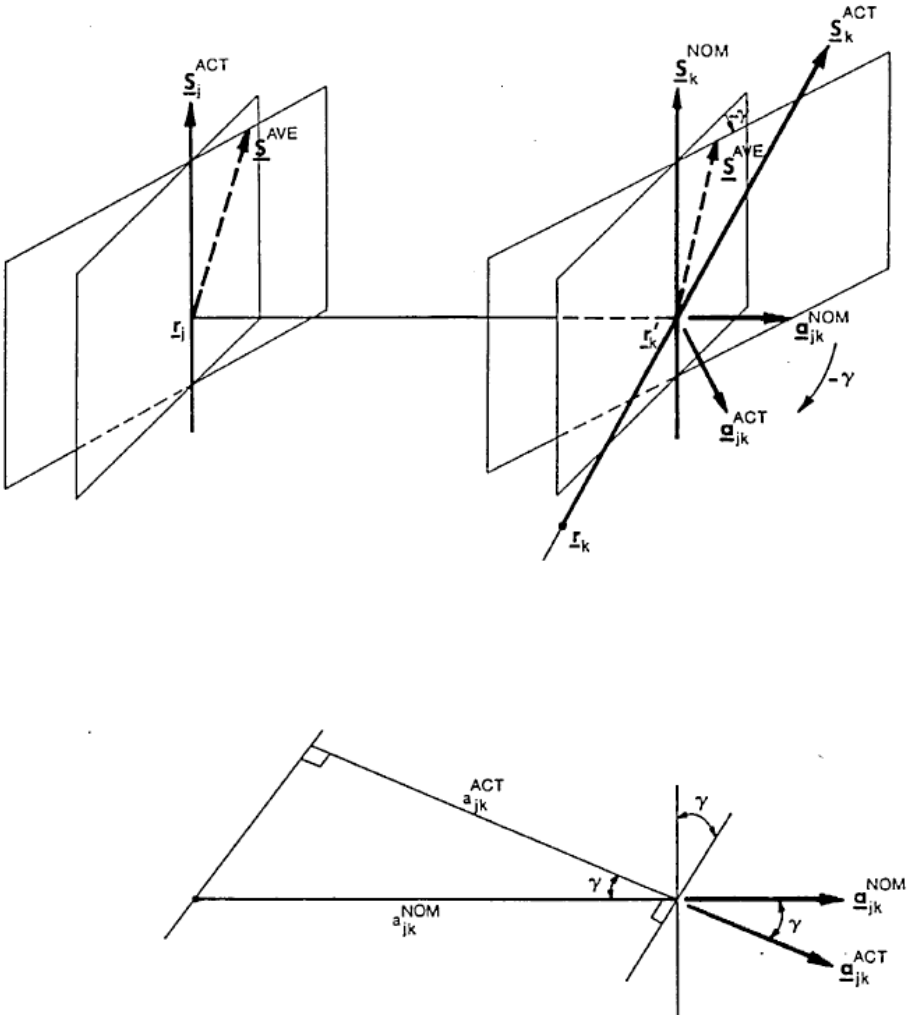


Ilustración 34 - Aproximación de ejes adyacentes mediante vector promedio

Nótese que $\overrightarrow{S_k^{nom}}$ es paralelo a $\overrightarrow{S_J}$ como se ve en la figura. Esto es debido al hecho de que todos los errores están en la ubicación del eje k con respecto a j, como se explica en la solución 1. La dirección media de los dos vectores $\overrightarrow{S_k^{real}}$ y $\overrightarrow{S_J}$ (o $\overrightarrow{S_k^{nom}}$), es simplemente la bisectriz de los vectores en el plano que forman, como viene dado por:

$$\overrightarrow{S^{med}} = \frac{(\overrightarrow{S_J} + \overrightarrow{S_k})}{|\overrightarrow{S_J} + \overrightarrow{S_k}|}$$

Ecuación 64

Este vector hace bisectriz ya que las magnitudes de los vectores unidad, $\overrightarrow{S_J}$ y $\overrightarrow{S_k}$, son equivalentes. Nótese que esta operación de promediar no tiene sentido si el ángulo de giro nominal es 180° y los vectores están en direcciones casi opuestas. En este caso, uno de los vectores debe apuntar en la dirección opuesta antes de promediar. Por convenio, el eje k se cambiará si es necesario para este análisis. Por tanto, la expresión correcta para el vector promedio es:

$$\overrightarrow{S^{med}} = \frac{\overrightarrow{S_J} + (\cos\alpha_{jk})\overrightarrow{S_k}}{|\overrightarrow{S_J} + (\cos\alpha_{jk})\overrightarrow{S_k}|}$$

Ecuación 65

Donde α_{jk} es el valor nominal de 0° o de 180°. Nótese, a través de esta solución, todos los ajustes hechos a los ejes actuales son temporales y se hacen únicamente con el propósito de calcular la longitud de la articulación calibrada $\overrightarrow{a_{jk}}$.

El punto de intersección nominal se determina desde el plano perpendicular a $\overrightarrow{S_j}$, el cual pasa a través del punto $\overrightarrow{P_j}$. El plano viene dado por:

$$\vec{r} \cdot \overrightarrow{S_j} = \overrightarrow{P_j} \cdot \overrightarrow{S_j}$$ Ecuación 66

El punto de intersección es la intersección de este plano y el real (medido) eje k ($\overrightarrow{S_k}$; $\overrightarrow{S_{0k}}$). El resultado se determina mediante la intersección general de una línea y un plano dados en la Ecuación 38, teniéndose que:

$$\overrightarrow{P_k} = \frac{\overrightarrow{S_j} \times \overrightarrow{S_{0k}} + (\overrightarrow{P_j} \cdot \overrightarrow{S_j})\overrightarrow{S_k}}{\overrightarrow{S_j} \cdot \overrightarrow{S_k}}$$ Ecuación 67

Este es el punto de intersección nominal, que representaría la intersección perpendicular común si ambos ejes *j* y k fueran exactamente paralelos sin ningún error. Las dos líneas modificadas para los ejes *j* y k respectivamente usadas para esta solución vienen dadas por:

$$\left(\overrightarrow{S^{med}}; \overrightarrow{P_j} \times \overrightarrow{S^{med}}\right) = (\overrightarrow{S'_j}; \overrightarrow{S'_{0j}})$$ Ecuación 68

$$\left(\overrightarrow{S^{med}}; \overrightarrow{P_k} \times \overrightarrow{S^{med}}\right) = (\overrightarrow{S'_k}; \overrightarrow{S'_{0k}})$$ Ecuación 69

El vector perpendicular común entre estas líneas paralelas modificadas, $\overrightarrow{a_{jk}^{real}}$, se puede determinar de la solución de la condición de líneas paralelas (dadas en la sección 9.5.3 *Condición 3: Líneas paralelas (MM=0, sin αjk=0)*). El resultado después de sustituir las coordenadas de la línea modificada, dadas en Ecuación 68 y en la Ecuación 69, en la Ecuación 55, es el siguiente:

$$\overrightarrow{a'^{real}_{jk}} = a_{jk}^{real} \cdot \overrightarrow{a_{jk}^{real}} = \overrightarrow{S^{med}} \times (\overrightarrow{S'_{0k}} - \overrightarrow{S'_{0j}})$$ Ecuación 70

Donde precisamente la convención establecida, $\cos \alpha_{jk} = \pm 1$, se ha sustituido aquí. Sustituyendo los momentos para las líneas modificadas como se ve en la Ecuación 68 y en la Ecuación 69, conduce a la siguiente ecuación:

$$\begin{aligned}\overrightarrow{a'_{jk}} &= \overrightarrow{S^{med}} \times \left(\left(\overrightarrow{P_k} \times \overrightarrow{S^{med}}\right) - \left(\overrightarrow{P_j} \times \overrightarrow{S^{med}}\right)\right) \\ &= \overrightarrow{S^{med}} \times \left(\left(\overrightarrow{P_k} - \overrightarrow{P_j}\right) \times \overrightarrow{S^{med}}\right)\end{aligned}$$ Ecuación 71

Sustituyendo el triple producto escalar identidad, y con el hecho de que $\overrightarrow{S^{med}}$ es un vector unidad, queda el siguiente resultado final:

$$\overrightarrow{a'^{real}_{jk}} = \left(\overrightarrow{P_k} - \overrightarrow{P_j}\right) - \left(\overrightarrow{S^{med}} \cdot \left(\overrightarrow{P_k} - \overrightarrow{P_j}\right)\right)\overrightarrow{S^{med}}$$ Ecuación 72

La única constante deseada para esta técnica es la longitud de la articulación, la cual es simplemente la magnitud siguiente:

$$a^{real}_{jk} = \left|\overrightarrow{a'^{real}_{jk}}\right|$$ Ecuación 73

Nótese que el ángulo ϒ en la Ilustración 34, que representa el ángulo entre los planos de rotación o error de desplazamiento entre los casos de errores ideal y real, en el ángulo girado. Como se ve en la vista de planta del dibujo, el error en la longitud $\overrightarrow{a_{jk}}$ para esta técnica de modelado es una función directa de este ángulo. Esta función se puede determinar a partir de la figura, y viene dada por la ecuación siguiente:

$$a^{real}_{jk} = a^{nom}_{jk} \cdot \cos\gamma$$ Ecuación 74

Esta técnica de aproximación no está limitada por problemas de sensibilidad extrema. La precisión será razonablemente buena en casos en los que el error del ángulo de giro sea leve y la longitud de la articulación sea grande. Esta será una solución especialmente precisa para errores de ángulo de giro muy pequeños, lo cual sucede en el caso del peor escenario donde los ejes j y k se acercan a una condición en la que se cruzan.

9.6.3 Solución 3: Modelo cinemático de parámetro adicional

Un cierto número de investigadores han considerado usar un modelo cinemático con un 5º parámetro, para cada articulación, con la técnica de calibración de datos de la posición del robot. La técnica del Círculo de Puntos es también capaza de calibrar un sistema usando el modelo modificado. El parámetro adicional representa una rotación adicional entre los sistemas de coordenadas de las articulaciones adyacentes. Esto alivia el problema de la sensibilidad numérica para ejes adyacentes. El parámetro extra dado como ángulo β, representa una rotación de un sistema de coordenadas de la articulación adyacente sobre un eje y local. Así, la transformación total entre los sistemas de coordenadas o ejes adyacentes contiene dos traslaciones (a_{jk}, d_j) y tres rotaciones (θ, β, α). Para el caso de ejes j y k casi paralelos, la rotación del eje y llamada β_{jk} tiene el efecto de colocar los ejes en planos paralelos. Así, el término α_{jk} representa el caso ideal de ejes nominalmente paralelos, descrito al principio de esta sección y que se ve en la Ilustración 30 - Ejes adyacentes paralelos - Caso Ideal.

Usando un modelo de 5 parámetros, la transformación total desde el sistema de coordenadas en la articulación k, $b_k\left[\overrightarrow{a_{k\ell}}\ \overrightarrow{b_{k\ell}}\ \overrightarrow{S_k}\right]$, hasta el sistema en la articulación j, $b_j\left[\overrightarrow{a_{j\ell}}\ \overrightarrow{b_{j\ell}}\ \overrightarrow{S_j}\right]$, vendrá dada por:

$$\left[T^k_j\right] = trans(\vec{a}, a)\ rot(\vec{a}, \alpha')\ rot\left(\vec{b}, \beta\right)\ trans\left(\vec{S}, d\right)\ rot(\vec{S}, \theta_k)$$ Ecuación 75

Donde las traslaciones y rotaciones genéricas son transformaciones homogéneas. Para mayor claridad, los subíndices jk o k se han dejado sin poner. El ángulo de giro α'_{jk} representa el ángulo modificado para este modelo y no es el ángulo total entre las direcciones $\overrightarrow{S_j}$ y $\overrightarrow{S_k}$, como en el modelo normal de 4 parámetros. Expandiendo esta transformación queda la única transformación completa para $[T_j^k]$ usando el modelo modificado:

$$\begin{bmatrix} \cos\beta_{jk}\cdot\cos\theta_k & -\cos\beta_{jk}\sin\theta_k & \sin\beta_{jk} & \sin\beta_{jk}\,d_k + a_{jk} \\ \cos\alpha_{jk}\sin\theta_k + \sin\alpha_{jk}\sin\beta_{jk}\cos\theta_k & \cos\alpha_{jk}\cos\theta_k - \sin\alpha_{jk}\sin\beta_{jk}\sin\theta_k & -\sin\alpha_{jk}\cos\beta_{jk} & -\sin\alpha_{jk}\cos\beta_{jk}\,d_k \\ \sin\alpha_{jk}\sin\theta_k - \cos\alpha_{jk}\sin\beta_{jk}\cos\theta_k & \sin\alpha_{jk}\cos\theta_k + \cos\alpha_{jk}\sin\beta_{jk}\sin\theta_k & \cos\alpha_{jk}\cos\beta_{jk} & \cos\alpha_{jk}\cos\beta_{jk}\,d_k \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 76

Donde se puede usar también la siguiente notación abreviada, para reducir la longitud de la matriz:

$$\begin{matrix} \cos\beta_{jk} = cb_{jk} & \cos\alpha_{jk} = c_{jk} & \cos\theta_k = c_k \\ \sin\beta_{jk} = sb_{jk} & \sin\alpha_{jk} = s_{jk} & \sin\theta_k = s_k \end{matrix}$$

Ecuación 77

Nótese que el parámetro extra sólo se calcula y usa para ejes nominalmente paralelos. Para articulaciones estándar, el parámetro β es igual a cero.

El análisis para esta solución es similar a la técnica aproximada enseñada en la Solución 2. Los parámetros cinemáticos y la localización de los ejes nominales y reales se ilustra en la Ilustración 35. Los parámetros a_{jk}, α'_{jk}, β_{jk} y los puntos de intersección $\overrightarrow{P_j}$ y $\overrightarrow{P_k}$ deben ser determinados. De nuevo, los errores en el eje k se modelan asumiendo que los ejes giran alrededor del punto de intersección nominal original $\overrightarrow{P_k}$. Esto es idéntico a la asunción hecha en la Solución 2 y así el punto de intersección $\overrightarrow{P_k}$ previamente derivado en la Ecuación 67 es válido. Nótese que para este caso $\overrightarrow{a_{jk}}$ tiene una dirección equivalente para tanto el caso nominal como para el caso real. Por tanto la longitud de la articulación es simplemente la siguiente:

$$a_{jk} = \left|\overrightarrow{P_k} - \overrightarrow{P_j}\right|$$

Ecuación 78

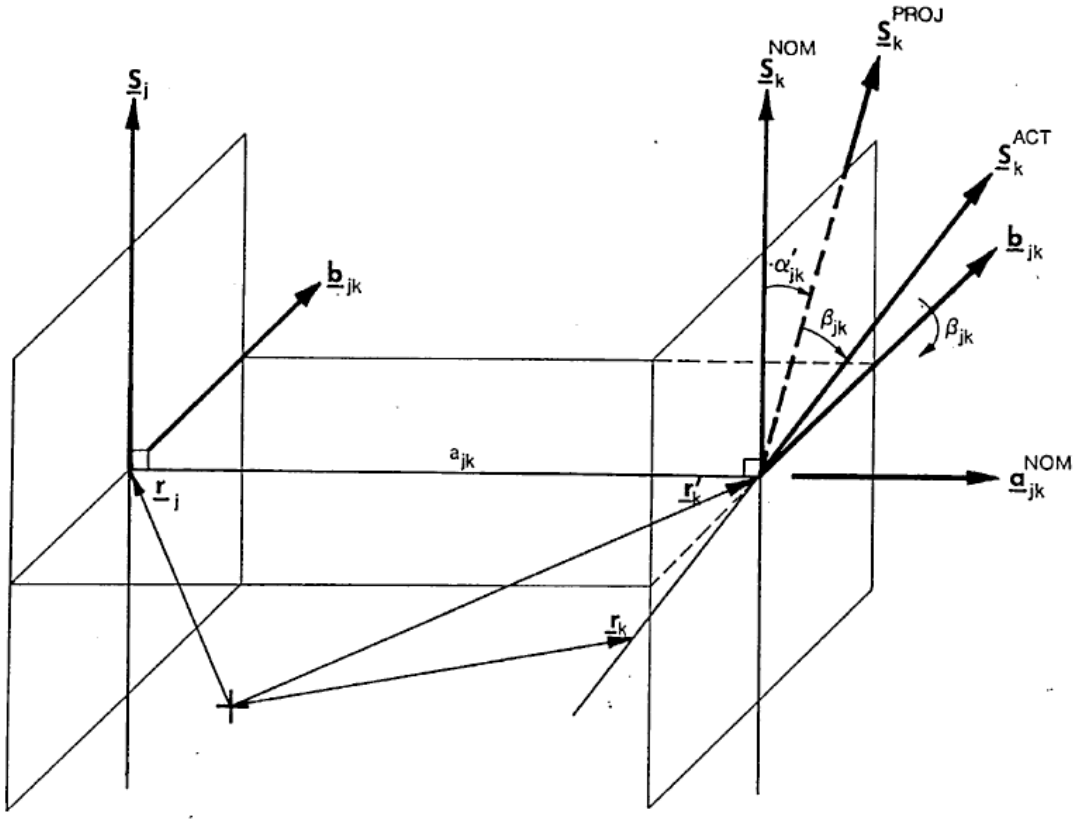


Ilustración 35 - Descripción del modelo cinemático de 5 parámetros

El punto $\overrightarrow{P_j}$ es el punto de intersección nominal el cual también se dio en la Solución 2. El ángulo β_{jk} es el ángulo entre el eje real $\overrightarrow{S_k^{real}}$ y el plano que es perpendicular a $\overrightarrow{a_{jk}}$. Este ángulo se determina por proyección. La proyección de $\overrightarrow{S_k^{real}}$ en el plano, $\overrightarrow{S_k^{prj}}$, viene dada por:

$$\overrightarrow{S_k^P} = \overrightarrow{S_k^{prj}} = \overrightarrow{S_k} - (\overrightarrow{S_k} \cdot \overrightarrow{a_{jk}^{nom}})\overrightarrow{a_{jk}^{nom}}$$

Ecuación 79

Donde la notación abreviada $\overrightarrow{S_k^P}$ ha sido introducida. El ángulo β_{jk} representa un giro dextrógiro alrededor del vector b_{jk} y, por tanto, la definición matemática queda así:

$$\cos\beta_{jk} = \frac{\overrightarrow{S_k^P}}{\left|\overrightarrow{S_k^P}\right|} \cdot \overrightarrow{S_k}$$

Ecuación 80

$$\sin\beta_{jk} = \frac{\overrightarrow{S_k^P}}{\left|\overrightarrow{S_k^P}\right|} \times \overrightarrow{S_k} \cdot \overrightarrow{b_{jk}}$$

Ecuación 81

Substituyendo la Ecuación 79 en la Ecuación 80 queda el siguiente resultado más simplificado:

$$\cos\beta_{jk} = \sqrt{\left[1 - (\overrightarrow{S_k} \cdot \overrightarrow{a_{jk}^{nom}})^2\right]}$$ Ecuación 82

La expresión para seno(β) no se puede simplificar. El nuevo ángulo de giro α'_{jk} es el ángulo entre $\overrightarrow{S_j}$ y $\overrightarrow{S_k^{prj}}$ alrededor del eje $\overrightarrow{a_{jk}}$ y viene dado por las siguientes definiciones:

$$\cos\alpha'_{jk} = \frac{\overrightarrow{S_k^P}}{\left|\overrightarrow{S_k^P}\right|} \cdot \overrightarrow{S_j}$$ Ecuación 83

$$\sin\alpha'_{jk} = \overrightarrow{S_j} \times \frac{\overrightarrow{S_k^P}}{\left|\overrightarrow{S_k^P}\right|} \cdot \overrightarrow{a_{jk}^{nom}}$$ Ecuación 84

Todos los parámetros requeridos por este modelo se pueden calcular mediante las ecuaciones anteriores. El análisis del Círculo de Puntos, como se ha visto en las anteriores 3 soluciones especiales para ejes nominalmente paralelos, es capaz de entregar los mismos parámetros dados por las técnicas de los Datos de Postura (*Pose Data techniques*).

9.7 Resumen de la técnica analítica del Círculo de Puntos

La técnica analítica y experimental del Círculo de Puntos precisamente aquí descrita suministra en un sentido absoluto todos los parámetros geométricos fijos requeridos para calcular la posición y orientación de la articulación final con un grado mayor de precisión que el que fuera posible con los parámetros nominales. Los únicos datos no suministrados por esta técnica son las constantes de errores del transductor de las uniones. Sin embargo, se han sugerido mejoras para incluir las constantes de error, basadas en pares de datos de errores múltiples o únicos. Un resumen completo del procedimiento se da a continuación.

El experimento entrega datos de puntos en un círculo que representan los movimientos individuales de cada eje. Para cada eje, los puntos primeramente se encajan en un plano. La dirección normal al plano es el vector dirección al plano. El usuario debe tener cuidado en este paso para asegurar que los puntos vectores en la dirección asumida por el modelo cinemático nominal, están en contraposición a la dirección opuesta. Dentro del plano obtenido, los puntos son entonces encajados en un círculo cuyo centro representa un punto arbitrario en el eje, el cual está representado por una línea en el espacio. Así, el experimento da las líneas de dirección y puntos arbitrarios $(\overrightarrow{S_i}, \overrightarrow{r_i})$ para cada *i = 1 hasta N*. Las coordenadas de líneas Plucker para cada línea $(\overrightarrow{S_i};\ \overrightarrow{r_i} \times \overrightarrow{S_i})$ se calculan entonces. Cada par de ejes adyacentes *j-k*, para *j = 1 hasta N-1*, se analiza entonces para determinar los parámetros

cinemáticos para cada articulación. El análisis se basa en las derivadas mostradas en las secciones anteriores, dependiendo de si las líneas son paralelas, casi paralelas, se cruzan o son oblicuas. El análisis para un par dado de ejes o líneas adyacentes en el espacio se revisa a continuación.

Las coordenadas de las líneas son $(\overrightarrow{S_j};\ \overrightarrow{S_{0j}})$ y $(\overrightarrow{S_k};\ \overrightarrow{S_{0k}})$:

1. Se calcula $\cos\alpha_{jk}$, y si es igual a ±1, dentro de una tolerancia especificada las líneas son exactamente paralelas y la longitud de la articulación a_{jk} se determina por la ecuación:

$$\overrightarrow{a'_{jk}} = a_{jk} \cdot \overrightarrow{a_{jk}} = \overrightarrow{S_j} \times (\cos\alpha_{jk}\,\overrightarrow{S_{0k}} - \overrightarrow{S_{0j}})$$ Ecuación 85

El ángulo de giro es simplemente:

$$\alpha_{jk} = \cos^{-1}(\overrightarrow{S_j} \cdot \overrightarrow{S_k})$$ Ecuación 86

Los puntos de intersección únicos no existen y dj, dk no se pueden calcular explícitamente.

2. Si $\cos\alpha_{jk}$ es aproximadamente ±1 dentro de una tolerancia dada, las líneas se tratan como casi paralelas y una de las 3 soluciones especiales (solución exacta con correcciones de ángulo, técnica de media aproximada, o modelo de parámetro adicional) se usa para analizar las líneas (ver apartado 9.6 Análisis de ejes nominalmente paralelos).
3. Se calcula el Momento Mutuo (MM), el cual viene dado por la siguiente fórmula:

$$MM = \overrightarrow{S_j} \cdot \overrightarrow{S_{0k}} + \overrightarrow{S_k} \cdot \overrightarrow{S_{0j}}$$ Ecuación 87

Si MM=0, entonces las líneas se cruzan y $a_{jk} = 0$. El punto de intersección común vendrá dado por la Ecuación 48, que es, como ya se ha visto:

$$\overrightarrow{P_j^{prox}} = \overrightarrow{P_k^{prev}} = \frac{\overrightarrow{S_{0j}} \times \overrightarrow{S_{0k}}}{\overrightarrow{S_{0j}} \cdot \overrightarrow{S_k}}$$

Por convención, $\alpha_{jk} < 180°$, y por tanto se tiene que:

$$\sin\alpha_{jk} = |\overrightarrow{S_j} \times \overrightarrow{S_k}|$$ Ecuación 88

$$\alpha_{jk} = \tan^{-1}\left(\frac{\sin\alpha_{jk}}{\cos\alpha_{jk}}\right)$$ Ecuación 89

4. Si MM≠0, y $\cos\alpha_{jk} \neq \pm 1$, lo cual sucede en la mayoría de los casos, las líneas son oblicuas. El vector dirección de la articulación viene dado por la Ecuación 31, que es como se ha visto anteriormente:

$$\overrightarrow{a_{jk}} = -sgn(MM) \cdot \frac{\overrightarrow{S_j} \times \overrightarrow{S_k}}{|\overrightarrow{S_j} \times \overrightarrow{S_k}|}$$

Los puntos de intersección se calculan a partir de las ecuaciones. La longitud a_{jk} es la distancia entre estos puntos. El ángulo girado se calcula en base a la Ecuación 29, que, como se ha visto anteriormente, es la siguiente ecuación:

$$\overrightarrow{S_j} \cdot \overrightarrow{S_{0k}} + \overrightarrow{S_k} \cdot \overrightarrow{S_{0j}} = -a_{jk} \cdot sen(\alpha_{jk})$$

Y despejando el ángulo se tiene la siguiente ecuación:

$$\alpha_{jk} = \tan^{-1}\left(\frac{\sin \alpha_{jk}}{\cos \alpha_{jk}}\right)$$ **Ecuación 90**

5. Para todos los casos, el valor de desplazamiento d_j, si está disponible, se calcula a partir de los puntos de intersección comunes a lo largo del eje j determinados mediante el análisis de los pares de líneas *i-j* y j-k:

$$d_j = \left|\overrightarrow{P_j^{prox}} - \overrightarrow{P_j^{prev}}\right|$$ **Ecuación 91**

El procedimiento arriba expuesto permite obtener los parámetros disponibles para todas las condiciones físicas posibles. Sin embargo, las ecuaciones presentadas antes se obtuvieron para suministrar una comprensión clara del análisis, y no deben ser el conjunto más eficiente. Se deben buscar mejoras en la eficiencia, aunque el problema de la calibración no es normalmente un procedimiento sensible al tiempo. Esta técnica provee unos parámetros geométricos reales, y es el único que da un indicio exacto de la naturaleza y fuente de los errores en una manipulador manufacturado. Por consiguiente, será el método más útil para calibraciones del fabricante dentro de sus instalaciones.

10 Anexo II: Formulación de los errores de rotación y traslación de un sistema

10.1 Errores de rotación de un eje

Se considera el cuerpo en rotación mostrado en la **¡Error! No se encuentra el origen de la referencia.**. Idealmente, el cuerpo gira sobre su eje de rotación sin ningún error. Sin embargo, en realidad los ejes de rotación giran alrededor de un eje del marco de coordenadas de referencia con errores radiales δ_X y δ_Y, un error axial δ_Z, y errores que se podrían llamar de "cabeceo" de los ejes, llamados ε_X y ε_Y. Todos estos errores pueden también ser una función del ángulo de rotación θ_Z. Para un punto en el marco de coordenadas del eje Xn Yn Zn, uno usaría primero el ángulo de rotación θ_Z para transformar el punto dentro del marco de referencia.

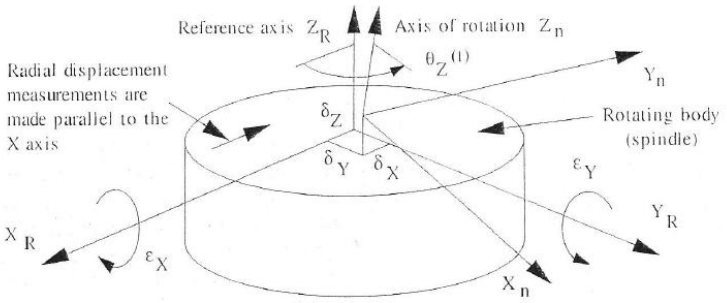


Ilustración 36 - Movimientos y errores sobre un eje de rotación

Entonces, dado que los otros errores de movimiento son pequeños, el orden de multiplicación de las matrices de transformación homogéneas no sería crítico. La multiplicación en secuencia de las matrices de transformación homogénea, que da la matriz resultante de los errores RT_{err}, tendría así la siguiente forma:

$$RT_{err} = MT_x \cdot MT_y \cdot MT_z \cdot MG_{\theta_x} \cdot MG_{\theta_y} \cdot MG_{\theta_z}$$ Ecuación 92

Teniendo en cuenta la siguiente convención en cuanto a la notación:

- MT_n es la matriz de traslación en una cantidad n a lo largo del eje n, para n=x, y ó z.
- $MG_{\theta n}$ es la matriz de giro en un ángulo θ alrededor del eje n, para n=x, y ó z.

El resultado general de desarrollar la matriz en la ecuación anterior sería el siguiente:

$$RT_{err} = \begin{bmatrix} \cos\varepsilon_Y \cos\theta_Z & -\cos\varepsilon_Y \operatorname{sen}\theta_Z & \operatorname{sen}\varepsilon_Y & \delta_X \\ \operatorname{sen}\varepsilon_X \operatorname{sen}\varepsilon_Y \cos\theta_Z + \cos\varepsilon_X \operatorname{sen}\theta_Z & \cos\varepsilon_X \cos\theta_Z - \operatorname{sen}\varepsilon_X \operatorname{sen}\varepsilon_Y \operatorname{sen}\theta_Z & -\operatorname{sen}\varepsilon_X \cos\varepsilon_Y & \delta_Y \\ -\cos\varepsilon_X \operatorname{sen}\varepsilon_Y \cos\theta_Z + \operatorname{sen}\varepsilon_X \operatorname{sen}\theta_Z & \operatorname{sen}\varepsilon_X \cos\theta_Z + \cos\varepsilon_X \operatorname{sen}\varepsilon_Y \operatorname{sen}\theta_Z & \cos\varepsilon_X \cos\varepsilon_Y & \delta_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ Ecuación 93

Nótese que este resultado general también puede usarse para el caso general de un movimiento lineal si ε_Z se substituye por θ_Z. Más a menudo ocurre que los términos de segundo orden, como el caso de $\varepsilon_X \cdot \varepsilon_Y$ se pueden despreciar, y se pueden asumir aproximaciones para los ángulos muy pequeños (por ejemplo, cos ε=1, sen ε = ε), con lo que esto lleva a convertir la ecuación anterior de la forma siguiente:

$$RT_{err} = \begin{bmatrix} \cos\theta_Z & -\operatorname{sen}\theta_Z & \varepsilon_Y & \delta_X \\ \operatorname{sen}\theta_Z & \cos\theta_Z & -\varepsilon_X & \delta_Y \\ \varepsilon_X \operatorname{sen}\theta_Z - \varepsilon_Y \cos\theta_Z & \varepsilon_X \cos\theta_Z + \varepsilon_Y \operatorname{sen}\theta_Z & 1 & \delta_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ Ecuación 94

Para los cálculos que se han empleado en este trabajo, se ha usado la ecuación sin aplicar aproximaciones, esto es, la Ecuación 93.

11 Anexo III: Cálculo de proyección de un punto en un plano

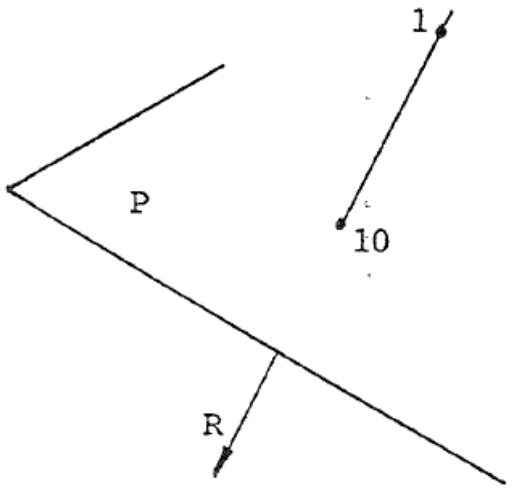
11.1 Proyección de un punto en un plano

Se incluye aquí los cálculos necesarios para realizar la proyección de un punto sobre un plano.

PROYECCION ORTOGONAL DE UN PUNTO SOBRE UN PLANO

DATOS: 1≡ (X1; Y1; Z1) ; P≡ (A; B; C; D')

Dividiendo los cuatro parámetros de P por SQR(A**2+B**2+C**2) éste toma la forma P≡ (A1; A2; A3; D)



Así sabremos que la recta proyectante R es:

R≡ (X1; Y1; Z1; A1; A2; A3)

El cálculo de la intersección de una recta sobre un plano se ve en el siguiente apartado.

11.2 Punto de intersección de recta y plano

DATOS: R≡ (X1; Y1; Z1; A1; A2; A3) ; P≡ (A; B; C; D)

RESULTADO: Punto resultante de resolver el sistema de ecuaciones (P1; P2; P) siendo P1 y P2 la pareja de planos conteniendo a R que corresponden según las condiciones del Punto 1.9. La intersección será según el Punto 1.2 .

El cálculo de la pareja de planos que contiene a R se ve en el siguiente apartado.

11.3 Pareja de planos disponibles conteniendo a una recta

1.9 PAREJA DE PLANOS DISPONIBLES CONTENIENDO A UNA RECTA

DATO: R≡ (X1; Y1; Z1; A1; A2; A3)

RESULTADO: A(X-X1)+B(Y-Y1)+C(Z-Z1)=0 (haz radial, Punto 1.2)

Con la condición de 1.6: A*A1+B*A2+C*A3=0

Se ve que las tres incógnitas A,B,C se pueden reducir a dos mediante la condición de paralelismo. Para determinarlas usaremos de la libertad que nos brinda el haz sin obligarnos a unos planos en particular. Así fijaremos dos coeficientes y obtendremos el tercero; y ello para llegar cada vez a dos planos distintos.

a) Haremos: A=1 para P1 y P2; B=1 para P1; B=2 para P2

$$\left.\begin{array}{l} P1\equiv (X-X1)+(Y-Y1)-((A1+A2)/A3)(Z-Z1)=0 \\ P2\equiv (X-X1)+2(Y-Y1)-((A1+2A2)/A3)(Z-Z1)=0 \end{array}\right\}$$

$$\left\{\begin{array}{l} P1\equiv (1;\ 1;\ -(A1+A2)/A3;\ Z1(A1+A2)/A3-X1-Y1) \\ P2\equiv (1;\ 2;\ -(A1+2A2)/A3;\ Z1(A1+2A2)/A3-X1-2Y1) \end{array}\right\} \text{a condición de que } A3\neq 0$$

b) Haremos: A=1 para P1 y P2; C=1 para P1; C=2 para P2

$$\left.\begin{array}{l} P1\equiv (X-X1)-((A1+A3)/A2)(Y-Y1)+(Z-Z1)=0 \\ P2\equiv (X-X1)-((A1+2A3)/A2)(Y-Y1)+2(Z-Z1)=0 \end{array}\right\}$$

$$\left\{\begin{array}{l} P1\equiv (1;\ -(A1+A3)/A2;\ 1;\ Y1(A1+A3)/A2-Z1-X1) \\ P2\equiv (1;\ -(A1+2A3)/A2;\ 2;\ Y1(A1+2A3)/A2-2Z1-X1) \end{array}\right\} \text{a condición de que } A2\neq 0$$

c) Haremos: B=1 para P1 y P2; C=1 para P1; C=2 para P2

$$\left.\begin{array}{l} P1\equiv -((A2+A3)/A1)(X-X1)+(Y-Y1)+(Z-Z1)=0 \\ P2\equiv -((A2+2A3)/A1)(X-X1)+(Y-Y1)+2(Z-Z1)=0 \end{array}\right\}$$

$$\left\{\begin{array}{l} P1\equiv (-(A2+A3)/A1;\ 1;\ 1;\ X1(A2+A3)/A1-Y1-Z1) \\ P2\equiv (-(A2+2A3)/A1;\ 1;\ 2;\ X1(A2+2A3)/A1-Y1-2Z1) \end{array}\right\} \text{a condición de que } A1\neq 0$$

Con estas ecuaciones de los planos P1, P2, R y P se calcula el punto de intersección.

11.4 Punto intersección de tres planos

1.2 PUNTO INTERSECCION DE 3 PLANOS

El plano lo definiremos por el conjunto de los 4 coeficientes (A; B; C; D) según su ecuación

$$Ax+By+Cz+D=0 \qquad (12)$$

Como se verá en el punto 1.15 el mismo plano puede adoptar la forma que sigue, en función de los cosenos directores de su normal y de la distancia de él al origen de coordenadas:

$$A1x+A2y+A3z+d=0 \qquad (13)$$

Comoquiera que A1, A2, A3 cumplen que:

$$A1**2+A2**2+A3**2=1$$

se ve que las 4 variables definidoras del plano se podrían reducir a tres adoptando entonces éste la forma canónica:

$$x/a+y/b+z/c=1 \qquad (14)$$

Lo normal será,pues, manejar la forma (12); la (13) se utilizará cuando sea preciso, pero no sistemáticamente: como es una derivación de la (12) conviene no hacer manipulaciones innecesarias con sus parámetros para evitar errores por redondeo del ordenador. La (14) no da la elasticidad necesaria cuando uno o más parámetros toman el valor 0.

Aprovecharemos la ocasión para indicar que el haz de planos que pasan por un punto viene dado por:

$$A(x-x1)+B(y-y1)+C(z-z1)=0$$

es decir: $Ax+By+Cz-(Ax1+By1+Cz1)=0$

siendo A,B,C, respectivamente los cosenos directores de la normal al plano (pueden representar cualquier dirección en el espacio).

Volviendo al plano por tres puntos, tenemos los siguientes

DATOS: (A1, B1, C1, D1);(A2, B2, C2, D2);(A3, B3, C3, D3)

Ellos dan lugar a un sistema de tres ecuaciones con tres incógnitas que resuelto matricialmente es:

$$\left.\begin{array}{l} A1x+B1y+C1z=-D1 \\ A2x+B2y+C2z=-D2 \\ A3x+B3y+C3z=-D3 \end{array}\right\}$$

$$\begin{pmatrix} A1 & B1 & C1 \\ A2 & B2 & C2 \\ A3 & B3 & C3 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -D1 \\ -D2 \\ -D3 \end{pmatrix}$$

MAT U(3,3)*MAT V(3,1)=MAT T(3,1)

MAT G=INV(U)

MAT V=G*T

RESULTADO: x=V(1,1) y=V(2,1) z=V(3,1)

a condición de que

$$\begin{vmatrix} A1 & B1 & C1 \\ A2 & B2 & C2 \\ A3 & B3 & C3 \end{vmatrix} \neq 0$$

Si este determinante valiera cero, ello significaría que los planos:

- o coinciden (los 3 ó 2)
- o son paralelos entre sí (los 3 ó 2)
- o los tres son paralelos a una misma recta y se cortan en tres rectas paralelas entre sí.

12 Anexo IV: Cálculo de la Matriz de Transferencia del sistema de referencia del robot al Láser Tracker

Se han tomado varios puntos, medidos tanto en el sistema de coordenadas del robot como del Láser Tracker.

Tabla 8 - Puntos en SR Robot y SR LT

DATOS LT EXPRESADOS EN SR ROBOT (mm)			DATOS DEL LT (en metros)		
X	Y	Z	X	Y	Z
344,9256783	99,88142547	375,1565793	-0,102366	-1,519710	-0,261207
344,5654692	-100,1148223	374,7865266	0,097617	-1,521401	-0,259562
544,6713185	-99,93138254	574,9023273	0,09672719	-1,32259382	-0,05815785
544,8399442	100,2268689	575,0680954	-0,103417	-1,321092	-0,060010
444,6902531	-0,058512255	474,9712401	-0,002786	-1,421260	-0,159741
344,9259599	99,88393839	375,1535316	-0,10236848	-1,51970968	-0,26121007
344,5645347	-100,1178993	374,7935022	0,097620	-1,521402	-0,259555
544,6710543	-99,92809454	574,8922084	0,096724	-1,322594	-0,058168
544,8409511	100,2219163	575,0640386	-0,103412	-1,321091	-0,060014
444,6903253	-0,064562252	474,9761789	-0,002780	-1,421260	-0,159736

Con estos puntos se ha usado la función de MatLab *Matriz_LT_Robot*, que a su vez llama a la función *AbsoluteOrientationQuaternion*, que permite calcular la matriz de transformación de un sistema de referencia a otro. La matriz así obtenida es la siguiente:

Tabla 9 - Matriz de transformación del LT al Robot

MATRIZ DE TRANSFORMACIÓN HOMOGÉNEA

0,0066005	0,999957	0,0065197	1866,94894
-0,9999269	0,0066661	-0,0100871	5,0186305
-0,0101301	-0,0064527	0,9999279	625,501561
0	0	0	1

Pero esta es la matriz de cambio del Sistema de referencia del LT al robot, y la matriz que se busca es la que convierte del sistema de referencia del robot al LT, es decir, se tiene que calcular la inversa de esta matriz. Por tanto, la matriz que se obtiene de cambio del sistema de referencia del robot al Láser Tracker es la siguiente:

Tabla 10 - Matriz de transformación del Robot al LT

Del SR Robot al SR LT

MATRIZ DE TRANSFORMACIÓN HOMOGÉNEA

0,00660053	-0,999926909	-0,010130127	-0,968178687
0,999956924	0,006666069	-0,006452643	-1862,865839
0,006519756	-0,010087073	0,999927839	-637,5778528
0	0	0	1

13 Anexo V: Cálculo de los parámetros cinemáticos del robot Kuka KR 5 sixx empleado

Se ha realizado el modelado cinemático del robot empleado en el trabajo, que es un robot Kuka, y para ello se ha empleado el modelo cinemático de Denavit-Hartenberg, modificado según el modelo de Hayati-Mirmirani.

El robot Kuka es el que se ve en las figuras siguientes.

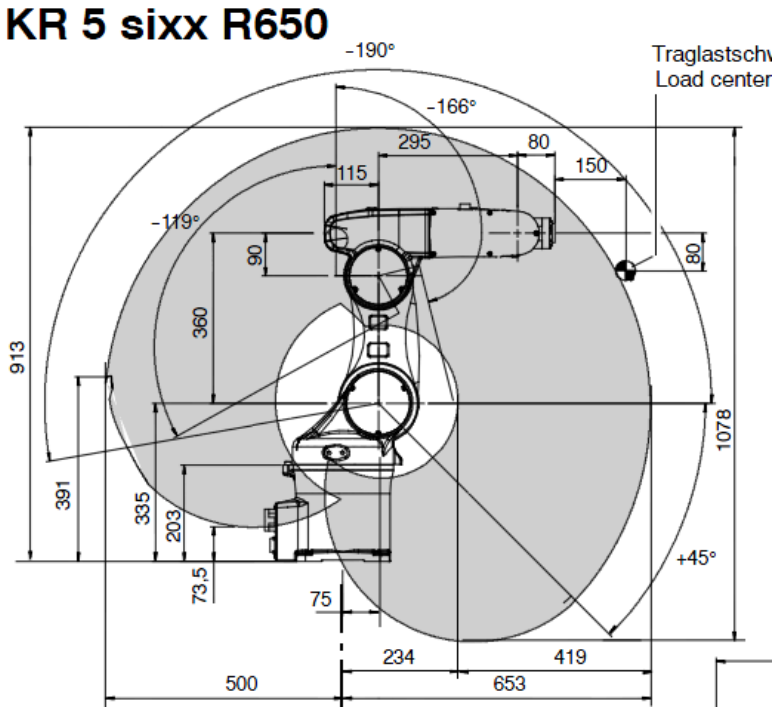


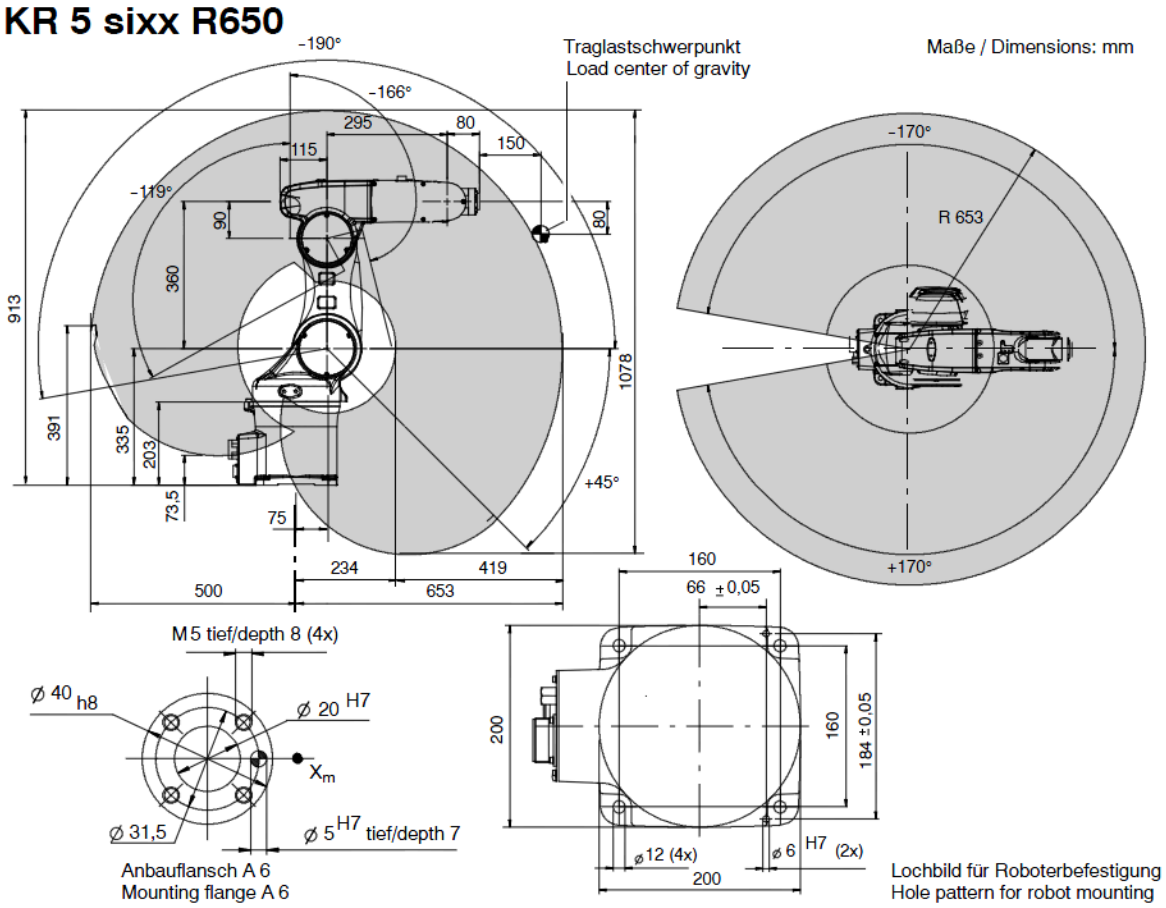
Ilustración 37- Dimensiones robot Kuka KR 5 sixx

Aplicando el modelo anteriormente citado, se obtiene la tabla de parámetros cinemáticos, que permite conocer la relación entre las articulaciones del robot.

Tabla 11 - Parámetros del modelo cinemático del robot Kuka

	d	θ	a	α	β
1	335	0	75	90	0
2	0	0	270	0	0
3	0	90	90	90	0
4	295	0	0	90	0
5	0	180	0	90	0
6	80	0	0	0	0

14 Anexo VI: Robot KUKA KR 5 sixx R650



Traglast / Payload:	5 kg	
Anzahl der Achsen / Number of axes:	6	
Anbauflansch A 6 / Mounting flange A 6:	DIN ISO 9409-1-A31,5	
Einbaulage / Mounting position:	Boden, Decke / Floor, ceiling	
Wiederholgenauigkeit (ISO 9283) / Repeatability (ISO 9283):	± 0,02 mm	
Steuerung / Controller:	KR C2sr	
Gewicht (ohne Steuerung) ca. / Weight (excl. controller) approx.:	28 kg	
Arbeitsraumvolumen / Work envelope volume:	1,000 m^3 1)	
max. Trägheitsmoment Achse 4/5 / Max. moment of inertia, axis 4/5:	0,295 kgm^2	
max. Trägheitsmoment Achse 6 / Max. moment of inertia, axis 6:	0,045 kgm^2	
max. Geschwindigkeit / Max. velocity:	8 200 mm/s	
Achsdaten / Axis data:	Bereich (Software) / Range (software)	Geschwindigkeit / Speed
Achse / Axis 1 (A 1)	± 170°	375 °/s
Achse / Axis 2 (A 2)	+ 45° / -190°	300 °/s
Achse / Axis 3 (A 3)	+ 166° / -119°	375 °/s
Achse / Axis 4 (A 4)	± 190°	410 °/s
Achse / Axis 5 (A 5)	± 120°	410 °/s
Achse / Axis 6 (A 6)	± 350°	660 °/s

1) Bezogen auf Schnittpunkt Achse 4/5. / Referred to intersection of axes 4 and 5.

Ilustración 38 - Datos robot Kuka KR 5 sixx

15 Anexo VII: Datos de entrada para las simulaciones

15.1 Valor de Variables iniciales para las simulaciones

Teniendo en cuenta que la llamada a la función de simulación por Montecarlo es:

>> [Pmed,uPmed,I_inf,I_sup, maximos, minimos, DifP, CP_Param,tiempo] = Montecarlo_Circle_Point (500000, Ruidomed, Initpos, SRLT, Preflector, Angcir, Nptos, Err, Param5, r0, IntCov)

A continuación se ve una tabla en la que se muestran las variables iniciales con las que se han llevado a cabo las simulaciones.

Tabla 12 - Datos variables iniciales para realizar las simulaciones

Ruidomed	15	0	Desviación estándar de la distribución normal con la que se generará aleatoriamente ruido normal debido al instrumento de medida																		
Initpos	15	0	0	0	0	0	Posición inicial del robot - vector con el ángulo inicial de cada articulación														
SRLT	1	0	0	0	Matriz de transformación del Sistema de Referencia del robot al SR del Láser Tracker																
	0	1	0	0																	
	0	0	1	0																	
	0	0	0	1																	
Preflector	100	0	0	Matriz 6x3 con las coordenadas X Y Z de la posición del reflector del LT (en posición inicial) expresadas en el SR de cada articulación																	
	200	0	0																		
	0	50	0																		
	20	0	0																		
	0	80	0																		
	0	60	0																		
Angcir	45	80	100	50	200	300	Vector 1x6 con el ángulo de giro que cubrirá cada articulación														
Nptos	15	20	30	15	25	30	Vector 1x6 con el número de puntos que se generarán en cada círculo														
Err	0	0	0	0	0	6,2832	6,2832	6,2832	6,2832	6,2832	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	6,2832	6,2832	6,2832	6,2832	6,2832	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	6,2832	6,2832	6,2832	6,2832	6,2832	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	6,2832	6,2832	6,2832	6,2832	6,2832	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	6,2832	6,2832	6,2832	6,2832	6,2832	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	6,2832	6,2832	6,2832	6,2832	6,2832	0	0	0	0	0	0	0	0	0	0	
	dx	dy	dz	Ex	Ey	T1	T2	T3	T4	T5	fi1	fi2	fi3	fi4	fi5	R1	R2	R3	R4	R5	
	Matriz 6x20 con los errores de giro a inducir en cada una de las articulaciones																				
Param5	335	0	75	90	0	Vector con los parámetros DH del robot (d, thita, a, alfa, beta)															
	0	0	270	0	0	Es matriz 1x30															
	0	90	90	90	0																
	295	0	0	90	0																
	0	180	0	90	0																
	80	0	0	0	0																
r0	35	35	35	35	35	35	Vector 1x6 con los estimadores de los radios de los círculos descritos por el reflector en cada articulación														
IntCov	95	Porcentaje de datos para el intervalo de cobertura																			

15.2 Cálculo de las diferentes matrices de transformación para la simulación

Para realizar las simulaciones propuestas en el capítulo 5, apartado 5 (*Para el mismo caso de equipo de medición, se ha cambiado la matriz de cambio del robot al Láser Tracker, considerando 3* valores diferentes (ver los desplazamientos tomados y las matrices calculadas en el apartado 15.2 Cálculo de las diferentes matrices de transformación para la simulación, página 82).), se tienen las siguientes matrices de transformación:

1. Matriz de transformación inicial → ver anexo anterior, tabla Tabla 10 - Matriz de transformación del Robot al LT.

2. Matriz segunda → se le aplica un desplazamiento de -400mm. La matriz de desplazamiento es la siguiente:

$$M_2 = \begin{pmatrix} 1 & 0 & 0 & -200 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ Ecuación 95

Por tanto, la matriz de cambio del sistema de referencia del robot al sistema de referencia del equipo de medición será la siguiente:

$$SRLT_2 = SRLT * M_2$$ Ecuación 96

$$SRLT_2 = \begin{pmatrix} 0.00661849 & -0.99992843 & -0.00996556 & -2.29810429 \\ 0.99995540 & 0.00668515 & -0.00667031 & -2062.84782 \\ 0.00673645 & -0.0099209 & 0.99992809 & -638.952407 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ Ecuación 97

3. Matriz tercera → se le aplica un desplazamiento de -2500mm, y se levanta +200mm en Z. La matriz de este desplazamiento es:

$$M_3 = \begin{pmatrix} 1 & 0 & 0 & -2500 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 200 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ Ecuación 98

Por tanto, la matriz de cambio del sistema de referencia del robot al sistema de referencia del equipo de medición será la siguiente:

$$SRLT_3 = SRLT * M_3$$ Ecuación 99

$$SRLT_3 = \begin{pmatrix} 0.00661849 & -0.99992843 & -0.00996556 & -19.51375337 \\ 0.99995540 & 0.00668515 & -0.00667031 & -4364.079327 \\ 0.00673645 & -0.0099209 & 0.99992809 & -454.4606377 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ Ecuación 100

16 Anexo VIII: Resultados en detalle de las simulaciones

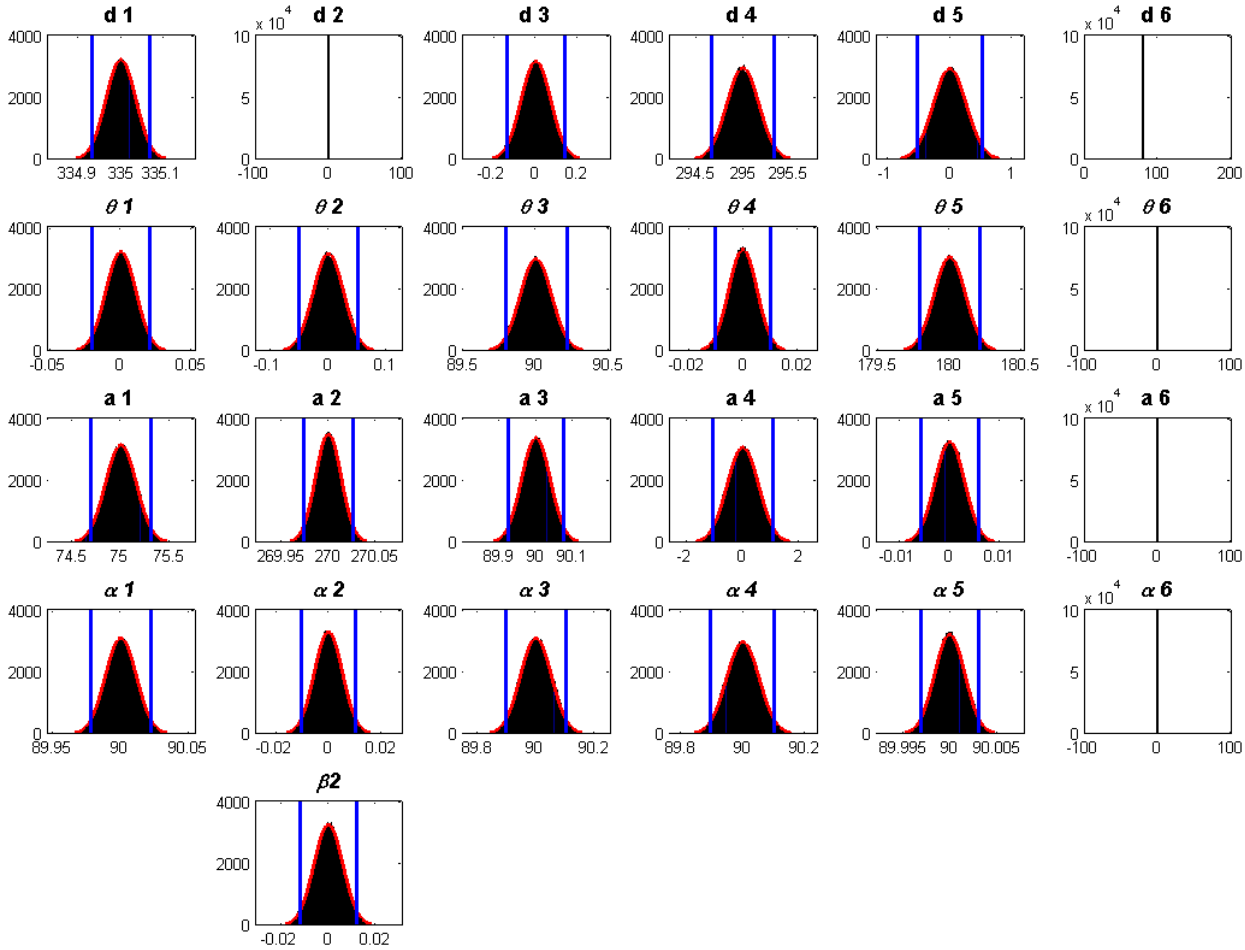
A continuación se pueden ver los resultados de las simulaciones, mostrados en detalle.

16.1 Análisis de la influencia del uso de diferentes equipos de medición

Se considera en este análisis el uso de diferentes equipos de medición (ver los 5 equipos de medición mostrados en la Tabla 13 - Incertidumbre de equipos de medición, en página 153).

16.1.1 Simulación 11 – 100.000 iteraciones – CPS

Ejecutada en ordenador con S.O. Windows XP Professional 64 Edition, v2003, SP2, con una CPU Intel Core 2 6300, 1.86GHz, 3.93Gb Ram.



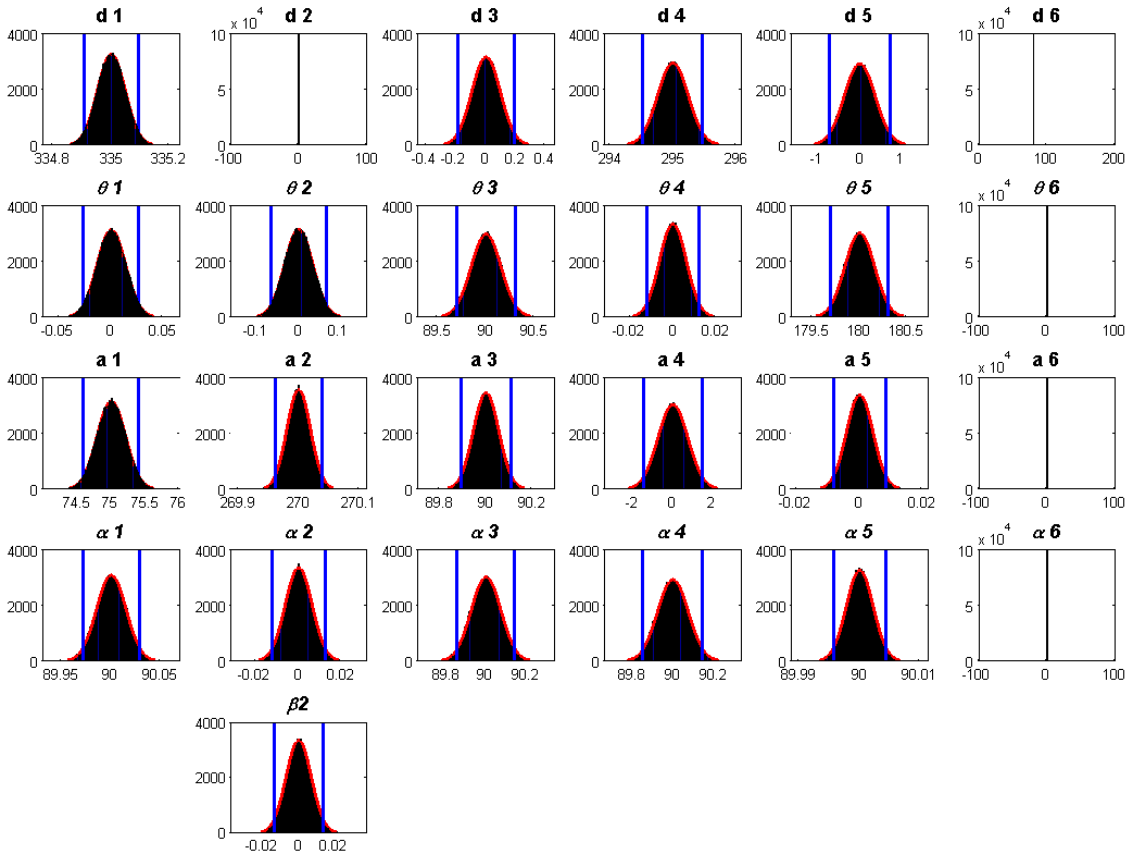
	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9999	0	2.1992e-04	294.9996	-3.3398e-04	80	9.7505e-06	-1.1284e-04	90.0000	-8.9277e-05	180.0000	0
Incertidumbre	0.0351	0	0.0714	0.1698	0.2646	0	0.0106	0.0258	0.1080	0.0052	0.1080	0
Lim.Sup	335.0680	0	0.1387	295.3299	0.5143	80	0.0205	0.0499	90.2113	0.0100	180.2102	0
Lim.Inf	334.9315	0	-0.1398	294.6679	-0.5186	80	-0.0207	-0.0501	89.7895	-0.0102	179.7889	0
Maximo	335.1442	0	0.2766	295.6528	0.9811	80	0.0421	0.1012	90.4235	0.0224	180.4022	0
Minimo	334.8576	0	-0.2977	294.3806	-0.9846	80	-0.0435	-0.1046	89.6100	-0.0203	179.5721	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0010	269.9999	89.9999	4.8620e-05	4.9012e-06	0	90.0000	-1.3111e-05	90.0001	89.9999	90.0000	0
Incertidumbre	0.1556	0.0135	0.0386	0.5519	0.0029	0	0.0112	0.0053	0.0522	0.0519	0.0016	0
Lim.Sup	75.3024	270.0263	90.0746	1.0737	0.0058	0	90.0217	0.0103	90.1014	90.1008	90.0031	0
Lim.Inf	74.6979	269.9735	89.9241	-1.0776	-0.0057	0	89.9782	-0.0104	89.8982	89.8984	89.9969	0
Maximo	75.6238	270.0647	90.1661	2.1280	0.0119	0	90.0448	0.0210	90.2020	90.1936	90.0065	0
Minimo	74.3806	269.9451	89.8357	-2.1666	-0.0121	0	89.9568	-0.0232	89.7923	89.8042	89.9936	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-6.6229e-06	0	0	0	0
Incertidumbre	0	0.0062	0	0	0	0
Lim.Sup	0	0.0121	0	0	0	0
Lim.Inf	0	-0.0121	0	0	0	0
Maximo	0	0.0248	0	0	0	0
Minimo	0	-0.0262	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	3	21	24.3783

16.1.2 Simulación 12 – 100.000 iteraciones - CPS



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9999	0	1.5024e-04	294.9990	-2.6028e-04	80	1.4369e-05	-1.4699e-04	89.9999	-1.8218e-04	180.0002	0
Incertidumbre	0.0469	0	0.0949	0.2413	0.3740	0	0.0137	0.0343	0.1535	0.0063	0.1535	0
Lim.Sup	335.0909	0	0.1846	295.4684	0.7271	80	0.0267	0.0664	90.3000	0.0122	180.2982	0
Lim.Inf	334.9082	0	-0.1850	294.5278	-0.7314	80	-0.0268	-0.0666	89.7016	-0.0125	179.7007	0
Maximo	335.1946	0	0.3829	295.9043	1.3601	80	0.0554	0.1336	90.5904	0.0268	180.5718	0
Minimo	334.8090	0	-0.3747	294.1065	-1.3917	80	-0.0528	-0.1397	89.4366	-0.0263	179.4024	0

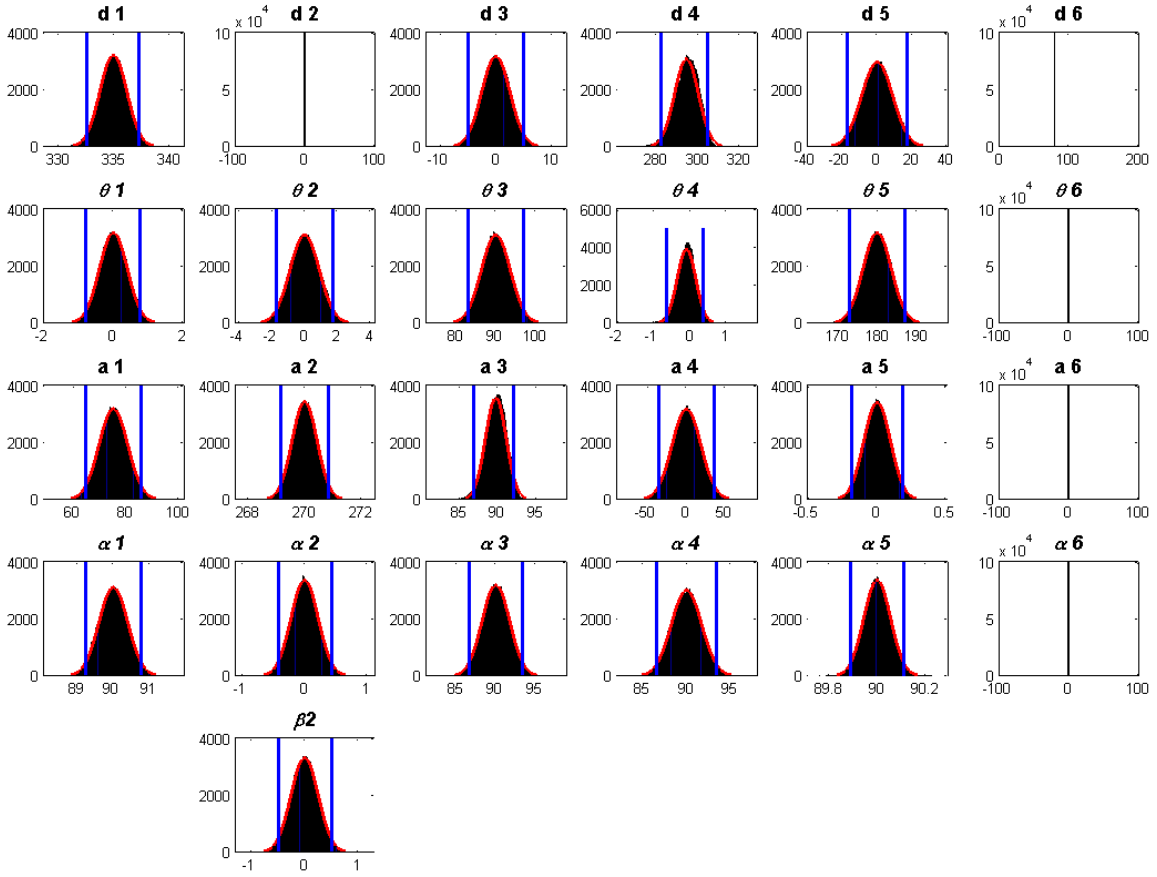
	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0013	269.9998	89.9997	7.4128e-04	2.5104e-06	0	90.0000	-1.6475e-05	90.0001	90.0000	90.0000	0
Incertidumbre	0.2062	0.0193	0.0551	0.7810	0.0042	0	0.0148	0.0063	0.0741	0.0738	0.0022	0
Lim.Sup	75.4008	270.0376	90.1061	1.5215	0.0083	0	90.0287	0.0123	90.1440	90.1433	90.0044	0
Lim.Inf	74.5992	269.9620	89.8916	-1.5227	-0.0083	0	89.9711	-0.0123	89.8555	89.8559	89.9956	0
Maximo	75.8339	270.0937	90.2314	2.9884	0.0178	0	90.0578	0.0263	90.2768	90.2682	90.0088	0
Minimo	74.1907	269.9195	89.7508	-2.9913	-0.0184	0	89.9426	-0.0267	89.7128	89.7232	89.9905	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-1.2331e-05	0	0	0	0
Incertidumbre	0	0.0073	0	0	0	0
Lim.Sup	0	0.0144	0	0	0	0
Lim.Inf	0	-0.0144	0	0	0	0
Maximo	0	0.0292	0	0	0	0
Minimo	0	-0.0329	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	5	33	56.4110

Ejecutada en ordenador con S.O. Windows XP Professional, versión 2002, SP2, con una CPU Pentium 4, 2.40GHz, 1.00Gb Ram.

16.1.3 Simulación 13 – 100.000 iteraciones - CPS



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9669	0	-0.1829	294.5632	0.0030	80	5.8866e-04	-0.0045	90.0062	-0.1033	179.9958	0
Incertidumbre	1.2131	0	2.5238	5.6730	8.8292	0	0.4005	0.8954	3.5946	0.2539	3.5947	0
Lim.Sup	337.2380	0	4.7268	304.3809	17.2859	80	0.7786	1.7291	97.0620	0.3547	186.9587	0
Lim.Inf	332.5226	0	-5.1195	282.3999	-17.1225	80	-0.7852	-1.7402	83.0428	-0.6486	172.9438	0
Maximo	339.5511	0	9.6292	310.7553	33.4103	80	1.5259	3.4705	104.6870	0.7929	193.9579	0
Minimo	329.6844	0	-10.6899	266.5073	-32.8593	80	-1.6910	-3.5613	76.4227	-1.7146	165.0141	0

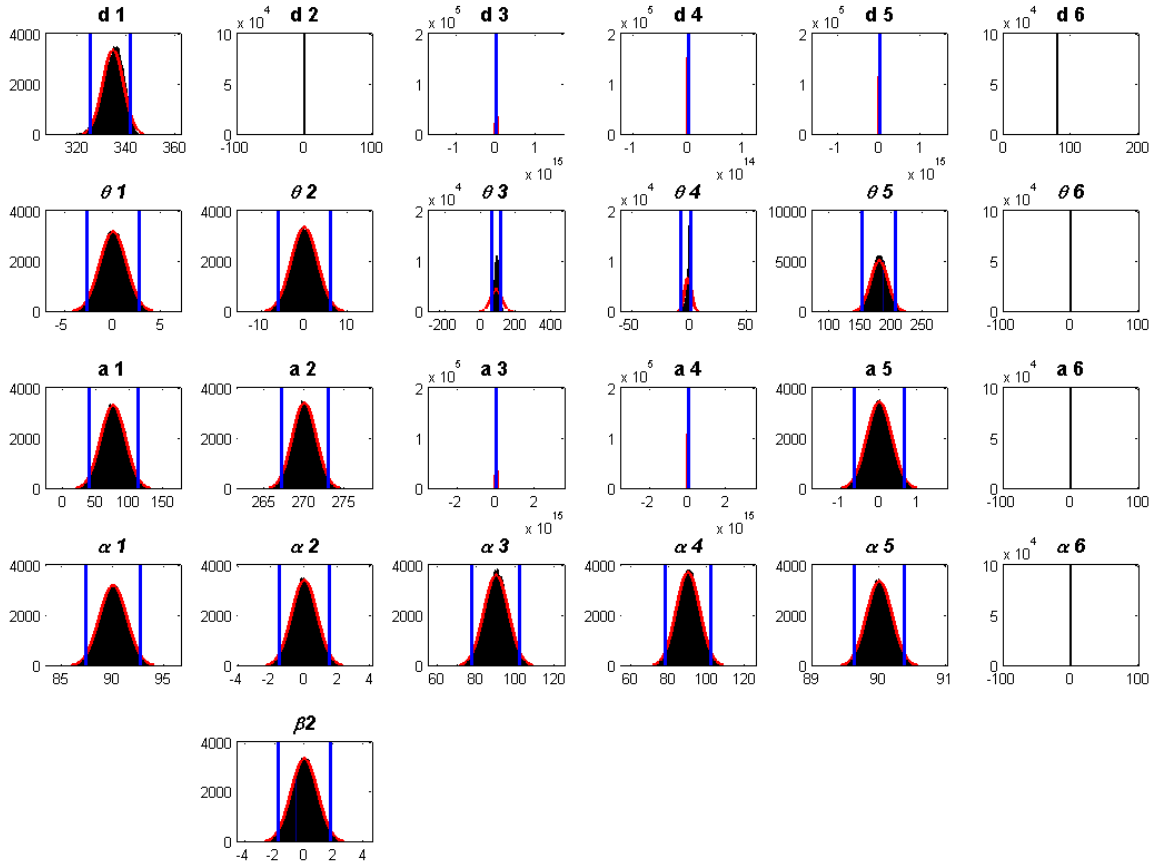
	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0473	269.9988	89.7491	0.0535	1.8981e-04	0	90.0012	-1.5540e-04	90.0009	89.9940	89.9998	0
Incertidumbre	5.4437	0.4382	1.3310	18.4773	0.0967	0	0.3911	0.2189	1.7406	1.7232	0.0552	0
Lim.Sup	85.6851	270.8554	92.0884	36.0984	0.1894	0	90.7578	0.4282	93.3635	93.3424	90.1076	0
Lim.Inf	64.5007	269.1446	86.8910	-35.8983	-0.1896	0	89.2387	-0.4270	86.5814	86.6276	89.8910	0
Maximo	97.3173	272.0490	94.2054	73.1948	0.4083	0	91.5959	0.9360	96.6345	96.4164	90.2273	0
Minimo	53.3875	268.2471	82.1824	-75.5108	-0.4277	0	88.5151	-0.9247	82.5687	83.3038	89.7582	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	1.3536e-04	0	0	0	0
Incertidumbre	0	0.2559	0	0	0	0
Lim.Sup	0	0.5030	0	0	0	0
Lim.Inf	0	-0.5002	0	0	0	0
Maximo	0	1.0568	0	0	0	0
Minimo	0	-1.0953	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	4	26	0.7663

Ejecutada en ordenador con S.O. Windows XP Professional x64 Edition, versión 2003, SP2, con una CPU Intel Core 2 6300, 1.86GHz,3.93Gb Ram.

16.1.4 Simulación 14 – 100.000 iteraciones – Simulación en CPS



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.6386	0	1.2037e+09	7.4380e+08	-1.2527e+10	80	0.0047	-0.0191	86.9672	-1.4907	179.9404	0
Incertidumbre	4.1893	0	8.6024e+12	4.4384e+11	8.0953e+12	0	1.4023	3.0757	31.5226	3.0965	13.9134	0
Lim.Sup	341.7891	0	16.9799	314.0889	57.3743	80	2.7273	5.9500	114.5626	1.0973	206.1798	0
Lim.Inf	325.5508	0	-21.7024	228.1328	-56.6152	80	-2.7427	-5.9940	60.9602	-7.7654	153.3083	0
Maximo	347.7282	0	1.4209e+15	1.0366e+14	1.0446e+15	80	5.4287	12.9287	124.9995	2.7550	269.5229	0
Minimo	311.7774	0	-9.7021e+14	-7.1751e+13	-1.3724e+15	80	-5.8556	-13.1171	-234.9967	-50.7355	90.0433	0

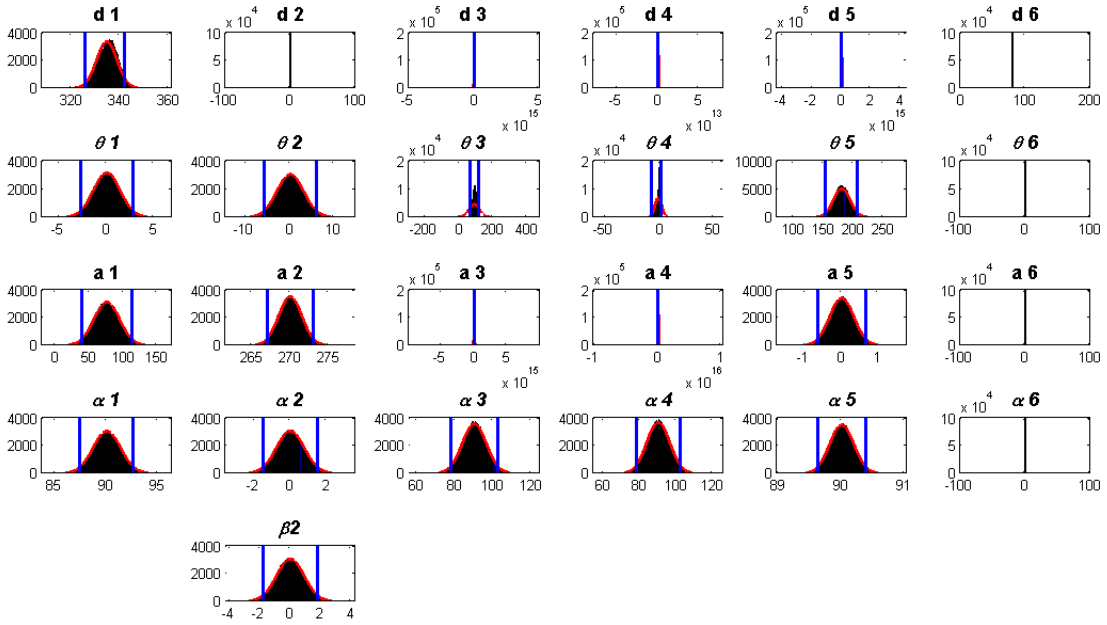
	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.2963	270.0199	1.7529e+10	-1.2752e+10	3.3118e-04	0	90.0042	-0.0014	89.9804	89.9722	89.9993	0
Incertidumbre	18.7240	1.4918	1.8039e+13	1.8269e+13	0.3302	0	1.3434	0.7750	6.3060	6.2332	0.1905	0
Lim.Sup	112.2668	272.9553	95.9407	131.6081	0.6451	0	92.5967	1.5109	102.0618	101.9743	90.3711	0
Lim.Inf	39.4701	267.1421	68.1676	-134.7724	-0.6477	0	87.3782	-1.5126	77.6313	77.8960	89.6233	0
Maximo	159.6456	277.0365	2.9096e+15	2.1686e+15	1.4064	0	95.4697	3.3995	119.2768	118.8844	90.7850	0
Minimo	2.5634	264.1279	-2.2097e+15	-2.9334e+15	-1.4787	0	84.6140	-3.2878	61.6285	60.2604	89.1614	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-6.5474e-05	0	0	0	0
Incertidumbre	0	0.9063	0	0	0	0
Lim.Sup	0	1.7809	0	0	0	0
Lim.Inf	0	-1.7709	0	0	0	0
Maximo	0	3.8361	0	0	0	0
Minimo	0	-3.8256	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	6	38	2.1402

Ejecutada en ordenador con S.O. Windows XP Professional x64 Edition, versión 2003, SP2, con una CPU Intel Core 2 6300, 1.86GHz,3.93Gb Ram.

16.1.5 Simulación 14 – 100.000 iteraciones – Simulación con Laptop



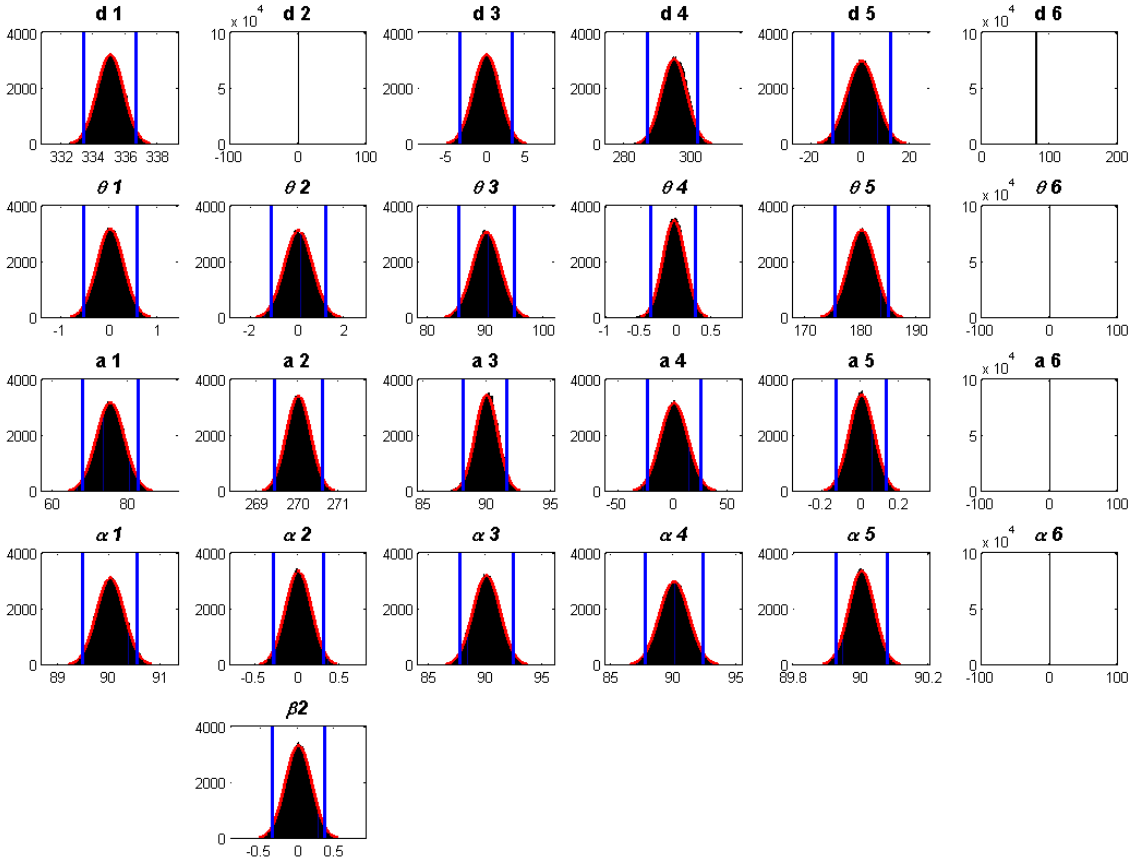
	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.6314	0	-6.6401e+10	-1.3045e+09	6.6675e+10	80	6.5818e-04	-0.0167	87.0375	-1.4899	179.9423	0
Incertidumbre	4.1906	0	1.8015e+13	4.2754e+11	1.7467e+13	0	1.4091	3.0791	31.3250	3.0562	13.9183	0
Lim.Sup	341.8162	0	17.0878	314.1657	57.6344	80	2.7549	5.9881	114.6499	1.0910	206.4677	0
Lim.Inf	325.5526	0	-21.7372	227.3022	-56.6897	80	-2.7405	-6.0069	60.6592	-7.8530	153.3324	0
Maximo	348.0134	0	1.3492e+15	6.6590e+13	3.6906e+15	80	5.7001	11.2633	124.9950	2.9717	269.6925	0
Minimo	312.3902	0	-4.1884e+15	-6.3465e+13	-1.2892e+15	80	-5.5467	-12.4200	-234.9946	-51.1438	90.2559	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.2687	270.0349	-1.5182e+11	1.5156e+11	-1.3650e-04	0	90.0030	-0.0031	89.9767	89.9707	90.0001	0
Incertidumbre	18.7752	1.4948	3.9044e+13	3.9283e+13	0.3289	0	1.3453	0.7759	6.3053	6.2587	0.1905	0
Lim.Sup	112.4564	272.9891	95.9326	133.1574	0.6420	0	92.5945	1.5205	102.1686	102.1064	90.3727	0
Lim.Inf	39.2472	267.1359	68.1261	-134.6592	-0.6477	0	87.3733	-1.5152	77.6427	77.9322	89.6265	0
Maximo	154.9443	276.3227	3.0001e+15	8.5195e+15	1.3674	0	95.0892	2.9964	118.6160	119.5288	90.8304	0
Minimo	6.5039	263.0597	-8.2871e+15	-3.0290e+15	-1.4668	0	84.7250	-3.0127	61.5273	60.9476	89.1453	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-0.0048	0	0	0	0
Incertidumbre	0	0.9088	0	0	0	0
Lim.Sup	0	1.7835	0	0	0	0
Lim.Inf	0	-1.7775	0	0	0	0
Maximo	0	3.5607	0	0	0	0
Minimo	0	-3.5135	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	4	51	37.8869

16.1.6 Simulación 15 – 100.000 iteraciones - CPS



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9838	0	-0.0820	294.7979	0.0040	80	3.1098e-04	-0.0030	90.0027	-0.0478	179.9974	0
Incertidumbre	0.8308	0	1.7290	3.8529	6.0458	0	0.2802	0.6137	2.4462	0.1619	2.4461	0
Lim.Sup	336.5580	0	3.2763	301.7374	11.8241	80	0.5436	1.1856	94.7818	0.2590	184.7419	0
Lim.Inf	333.3269	0	-3.4606	286.7699	-11.7601	80	-0.5501	-1.1931	85.2509	-0.3782	175.2184	0
Maximo	338.1888	0	6.5707	307.2700	22.9856	80	1.0713	2.3964	99.8780	0.5580	189.3335	0
Minimo	331.4509	0	-7.4472	277.5207	-22.5369	80	-1.1836	-2.4502	80.9028	-0.8627	169.8510	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0295	269.9986	89.8839	0.0209	1.5472e-04	0	90.0008	-3.1092e-04	89.9994	89.9977	89.9999	0
Incertidumbre	3.7393	0.2973	0.8806	12.6076	0.0661	0	0.2686	0.1551	1.1870	1.1745	0.0381	0
Lim.Sup	82.3105	270.5792	91.4866	24.5903	0.1293	0	90.5203	0.3026	92.3008	92.2865	90.0743	0
Lim.Inf	67.7839	269.4189	88.0535	-24.4887	-0.1293	0	89.4772	-0.3027	87.6784	87.7062	89.9246	0
Maximo	90.2032	271.3748	93.0977	49.2358	0.2823	0	91.1012	0.6719	94.6464	94.3582	90.1571	0
Minimo	60.0577	268.8012	85.4102	-51.7325	-0.2956	0	88.9794	-0.6461	85.0319	85.4674	89.8319	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-5.7402e-05	0	0	0	0
Incertidumbre	0	0.1814	0	0	0	0
Lim.Sup	0	0.3566	0	0	0	0
Lim.Inf	0	-0.3547	0	0	0	0
Maximo	0	0.7536	0	0	0	0
Minimo	0	-0.7803	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	4	16	18.9375

Ejecutada en ordenador con S.O. Windows XP Professional, versión 2002, SP3, con una CPU Intel Celeron E1400, 2.00GHz, 0.99Gb RAM.

16.3 Análisis de la influencia de la variación del ángulo cubierto en la incertidumbre

Tomando un equipo de medición, que será el Láser Tracker, se elige la articulación 4, y se lanzan simulaciones, tomándose los siguientes ángulos: 15°, 40°, 90° y 180°.

16.3.1 Simulación 21 – 100.000 iteraciones – PC i7

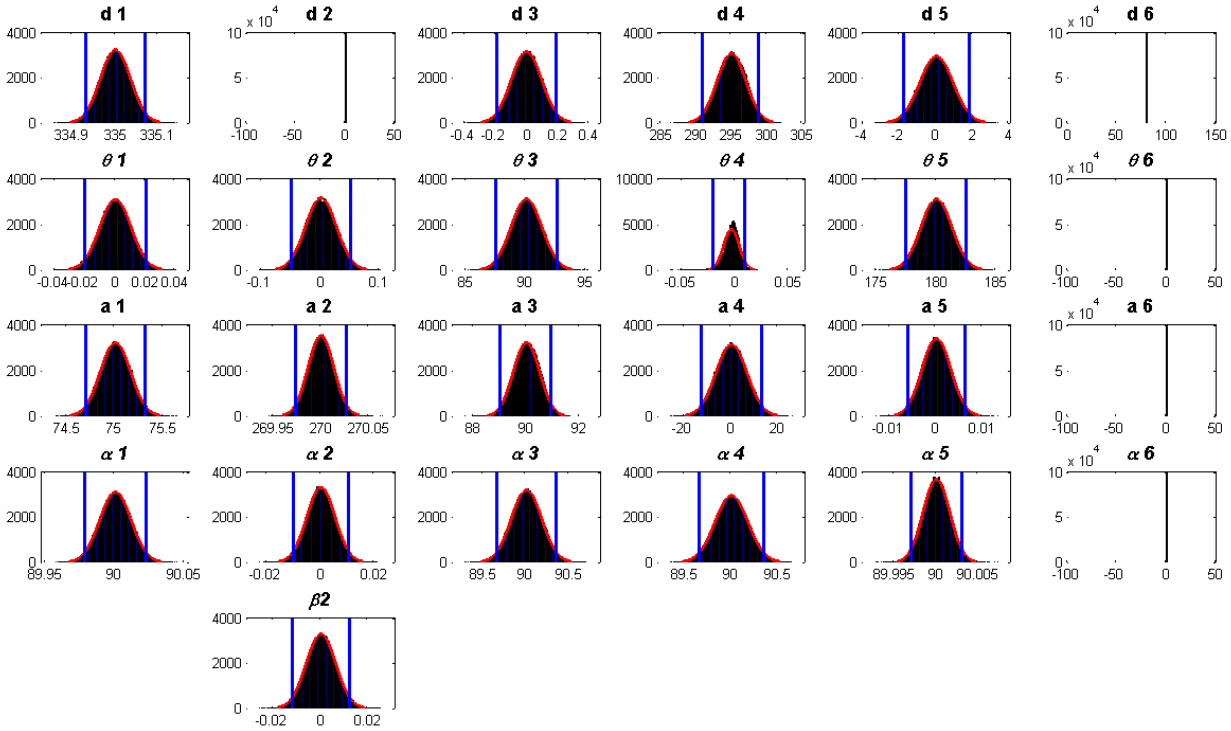


Tabla de datos de Parámetros

File Edit View Insert Tools Desktop Window Help

	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9999	0	-0.0070	294.9266	2.3815e-04	80	-1.2325e-05	-7.1586e-05	90.0002	-0.0038	179.9998	0
Incertidumbre	0.0352	0	0.0968	2.0320	0.9061	0	0.0105	0.0259	1.2928	0.0073	1.2928	0
Lim.Sup	335.0683	0	0.1794	298.6814	1.7718	80	0.0205	0.0502	92.5227	0.0082	182.5163	0
Lim.Inf	334.9314	0	-0.1989	290.7678	-1.7696	80	-0.0207	-0.0503	87.4833	-0.0208	177.4754	0
Maximo	335.1442	0	0.3675	301.9432	3.3835	80	0.0404	0.1012	95.1845	0.0205	184.9421	0
Minimo	334.8576	0	-0.4030	286.1837	-3.3566	80	-0.0416	-0.1046	85.0710	-0.0616	174.8014	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0007	269.9999	89.9676	0.0087	2.5346e-06	0	90.0000	-9.7765e-06	90.0001	89.9998	90.0000	0
Incertidumbre	0.1561	0.0135	0.4901	6.6489	0.0031	0	0.0112	0.0053	0.1765	0.1764	0.0016	0
Lim.Sup	75.3041	270.0263	90.8846	12.9653	0.0061	0	90.0218	0.0104	90.3430	90.3430	90.0031	0
Lim.Inf	74.6965	269.9736	88.9760	-12.9535	-0.0061	0	89.9782	-0.0104	89.6550	89.6551	89.9969	0
Maximo	75.6435	270.0647	91.6252	26.3900	0.0134	0	90.0448	0.0210	90.6983	90.6515	90.0075	0
Minimo	74.3795	269.9451	87.6293	-26.2290	-0.0132	0	89.9568	-0.0232	89.2764	89.3379	89.9927	0

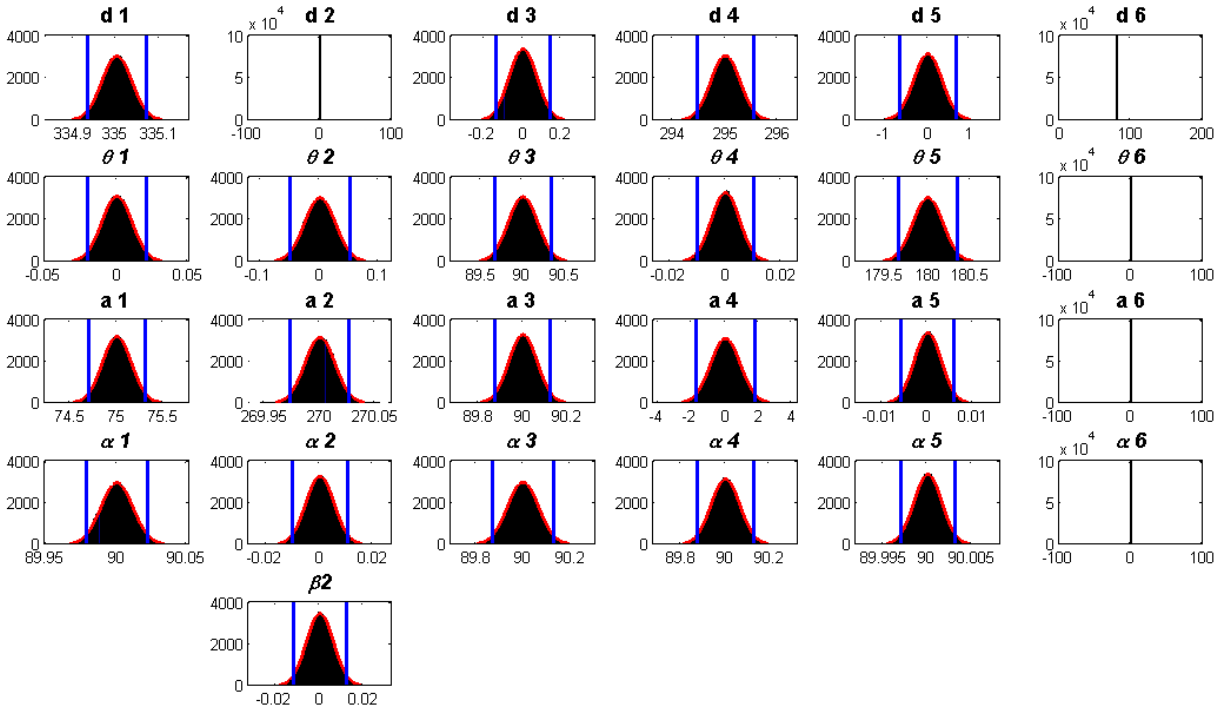
	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-1.2308e-07	0	0	0	0
Incertidumbre	0	0.0062	0	0	0	0
Lim.Sup	0	0.0121	0	0	0	0
Lim.Inf	0	-0.0121	0	0	0	0
Maximo	0	0.0250	0	0	0	0
Minimo	0	-0.0262	0	0	0	0

Duración de la Simulación

File Edit View Insert Tools Desktop Window Help

	Horas	Minutos	Segundos
Tiempo empleado	4	42	13.7049

16.3.2 Simulación 22 – 100.000 iteraciones – Laptop



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	335.0002	0	-6.1040e-04	294.9997	-0.0010	80	-2.8224e-06	1.7057e-04	90.0006	-1.8631e-04	179.9994	0
Incertidumbre	0.0352	0	0.0731	0.2737	0.3346	0	0.0106	0.0259	0.1742	0.0051	0.1741	0
Lim.Sup	335.0687	0	0.1424	295.5286	0.6503	80	0.0207	0.0505	90.3399	0.0098	180.3392	0
Lim.Inf	334.9318	0	-0.1429	294.4633	-0.6528	80	-0.0207	-0.0502	89.6618	-0.0102	179.6599	0
Maximo	335.1294	0	0.3158	295.9857	1.4145	80	0.0422	0.1006	90.6398	0.0215	180.6965	0
Minimo	334.8600	0	-0.3080	293.8682	-1.2372	80	-0.0406	-0.0971	89.2892	-0.0207	179.3630	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	74.9992	269.9999	89.9994	-0.0027	3.0058e-06	0	89.9999	-1.1191e-05	89.9998	89.9998	90.0000	0
Incertidumbre	0.1562	0.0134	0.0635	0.8915	0.0030	0	0.0112	0.0053	0.0657	0.0655	0.0016	0
Lim.Sup	75.3040	270.0263	90.1218	1.7388	0.0058	0	90.0218	0.0103	90.1279	90.1272	90.0031	0
Lim.Inf	74.6964	269.9737	89.8751	-1.7431	-0.0058	0	89.9782	-0.0103	89.8716	89.8725	89.9969	0
Maximo	75.6110	270.0534	90.2554	3.6194	0.0133	0	90.0424	0.0211	90.2540	90.2742	90.0068	0
Minimo	74.3517	269.9464	89.7294	-3.3651	-0.0121	0	89.9586	-0.0224	89.7580	89.7547	89.9935	0

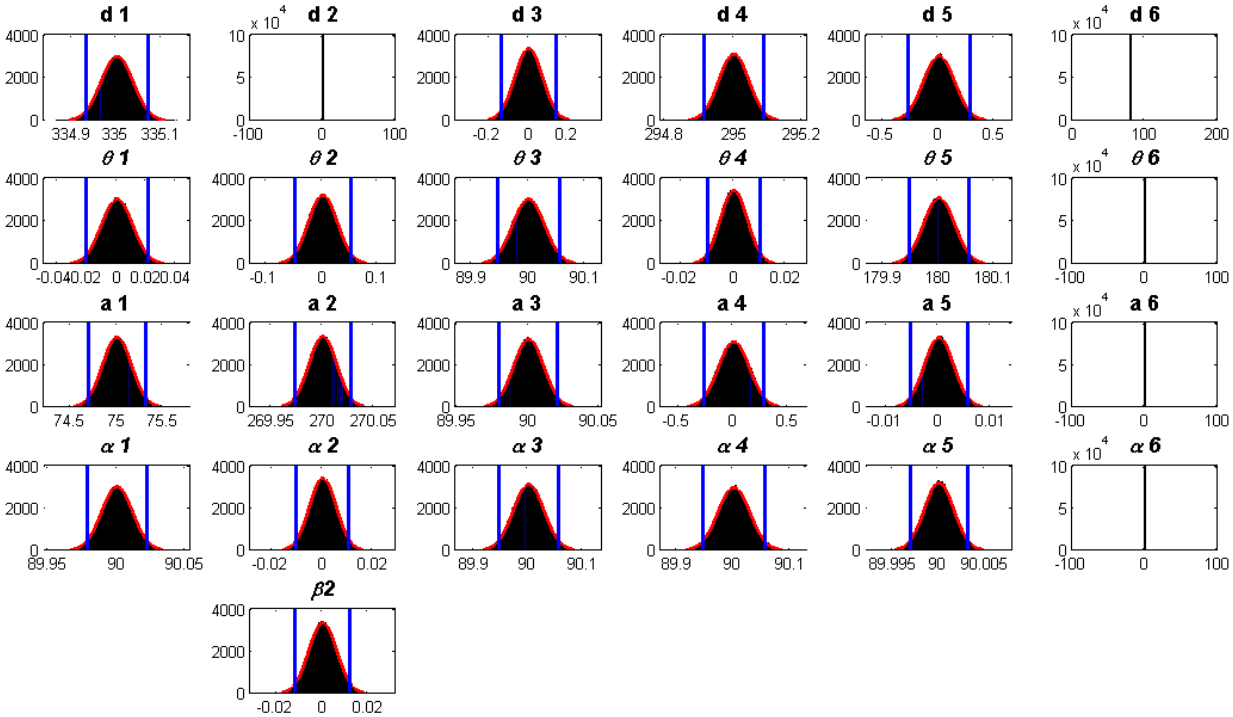
	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-1.8749e-05	0	0	0	0
Incertidumbre	0	0.0062	0	0	0	0
Lim.Sup	0	0.0120	0	0	0	0
Lim.Inf	0	-0.0120	0	0	0	0
Maximo	0	0.0267	0	0	0	0
Minimo	0	-0.0271	0	0	0	0

Duración de la Simulación

File Edit View Insert Tools Desktop Window Help

	Horas	Minutos	Segundos
Tiempo empleado	2	32	26.5053

16.3.3 Simulación 23 – 100.000 iteraciones – Laptop



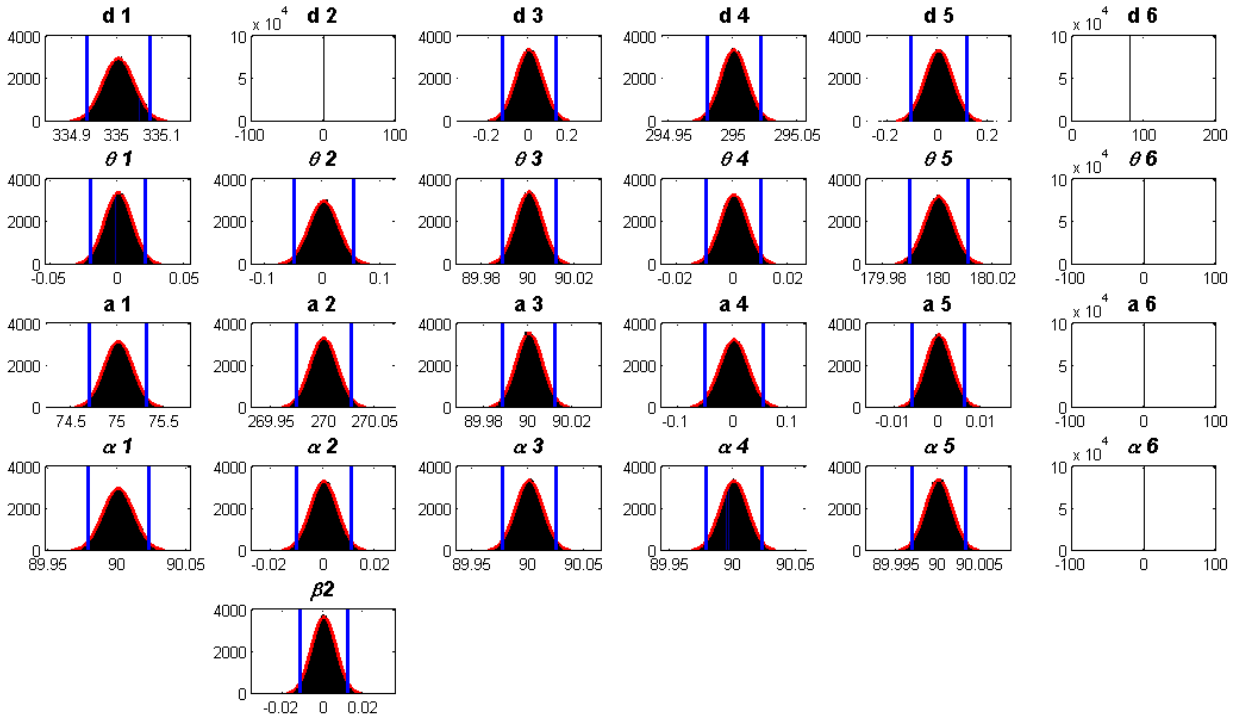
	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	335.0001	0	-1.0505e-04	294.9999	4.0901e-04	80	-7.1077e-06	5.0262e-05	89.9999	-1.4586e-05	180.0000	0
Incertidumbre	0.0354	0	0.0700	0.0440	0.1402	0	0.0105	0.0260	0.0278	0.0052	0.0277	0
Lim.Sup	335.0688	0	0.1365	295.0853	0.2759	80	0.0206	0.0507	90.0539	0.0102	180.0538	0
Lim.Inf	334.9310	0	-0.1369	294.9139	-0.2723	80	-0.0206	-0.0507	89.9456	-0.0102	179.9465	0
Maximo	335.1340	0	0.2872	295.1667	0.5262	80	0.0393	0.1025	90.1058	0.0235	180.1091	0
Minimo	334.8655	0	-0.3088	294.8218	-0.5461	80	-0.0413	-0.1084	89.8929	-0.0217	179.8935	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	74.9998	270.0000	90.0000	1.7909e-04	3.7278e-06	0	90.0000	1.6893e-05	90.0000	90.0001	90.0000	0
Incertidumbre	0.1567	0.0135	0.0104	0.1409	0.0028	0	0.0113	0.0053	0.0282	0.0277	0.0016	0
Lim.Sup	75.3061	270.0264	90.0202	0.2741	0.0054	0	90.0220	0.0103	90.0549	90.0542	90.0031	0
Lim.Inf	74.6951	269.9737	89.9797	-0.2727	-0.0054	0	89.9781	-0.0103	89.9453	89.9462	89.9968	0
Maximo	75.6620	270.0560	90.0422	0.5561	0.0116	0	90.0417	0.0219	90.1105	90.1031	90.0064	0
Minimo	74.3481	269.9418	89.9576	-0.5491	-0.0116	0	89.9554	-0.0238	89.8879	89.8940	89.9933	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	2.4131e-05	0	0	0	0
Incertidumbre	0	0.0062	0	0	0	0
Lim.Sup	0	0.0121	0	0	0	0
Lim.Inf	0	-0.0121	0	0	0	0
Maximo	0	0.0267	0	0	0	0
Minimo	0	-0.0258	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	2	27	27.6193

16.3.4 Simulación 24 – 100.000 iteraciones – Laptop



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	335.0001	0	-7.1370e-05	295.0000	2.4302e-04	80	3.1727e-07	6.6946e-05	90.0000	1.6351e-05	180.0000	0
Incertidumbre	0.0352	0	0.0690	0.0107	0.0566	0	0.0106	0.0259	0.0057	0.0050	0.0053	0
Lim.Sup	335.0690	0	0.1340	295.0209	0.1109	80	0.0204	0.0508	90.0112	0.0098	180.0104	0
Lim.Inf	334.9316	0	-0.1344	294.9789	-0.1103	80	-0.0207	-0.0504	89.9887	-0.0099	179.9895	0
Maximo	335.1315	0	0.3059	295.0439	0.2354	80	0.0451	0.0925	90.0258	0.0217	180.0209	0
Minimo	334.8677	0	-0.2825	294.9530	-0.2414	80	-0.0450	-0.1026	89.9763	-0.0201	179.9781	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	74.9997	270.0000	90.0000	-7.4686e-05	2.0487e-05	0	90.0000	-1.6384e-05	90.0000	90.0000	90.0000	0
Incertidumbre	0.1561	0.0135	0.0060	0.0266	0.0031	0	0.0113	0.0053	0.0124	0.0111	0.0016	0
Lim.Sup	75.3021	270.0264	90.0117	0.0519	0.0060	0	90.0219	0.0103	90.0243	90.0219	90.0032	0
Lim.Inf	74.6950	269.9736	89.9883	-0.0523	-0.0060	0	89.9779	-0.0104	89.9757	89.9784	89.9968	0
Maximo	75.6397	270.0578	90.0271	0.1075	0.0139	0	90.0434	0.0228	90.0541	90.0459	90.0073	0
Minimo	74.3966	269.9445	89.9734	-0.1094	-0.0131	0	89.9592	-0.0213	89.9490	89.9524	89.9934	0

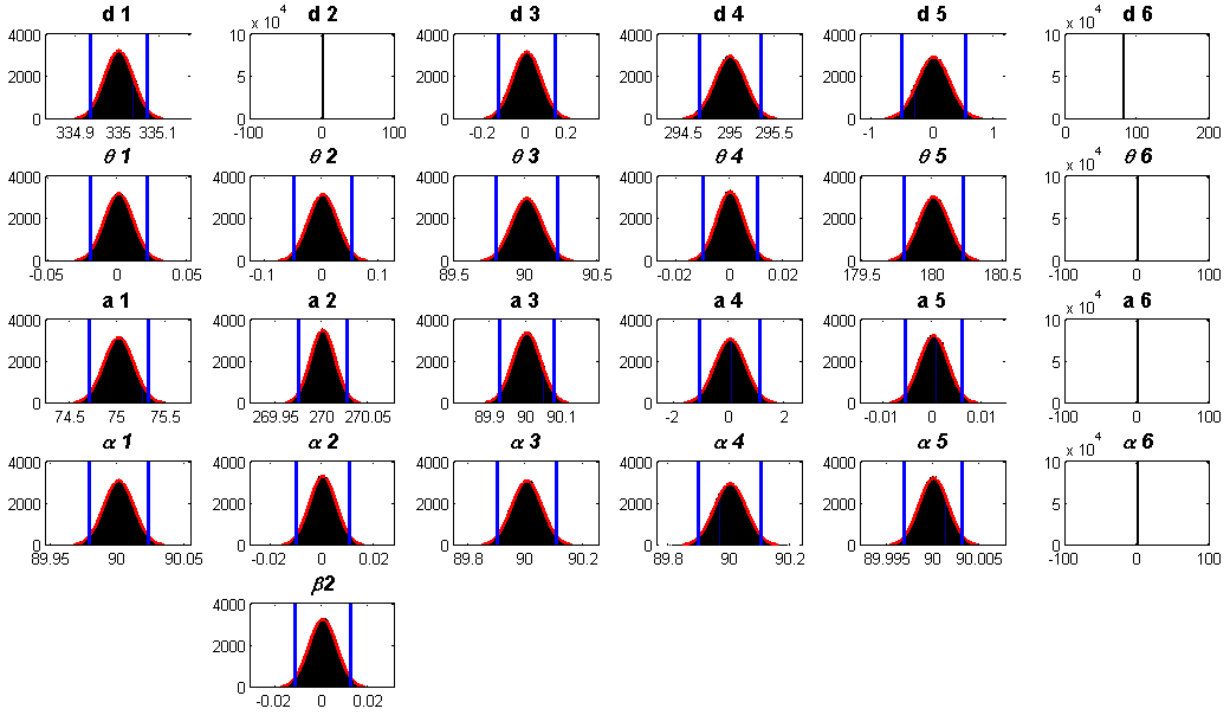
	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-1.3072e-05	0	0	0	0
Incertidumbre	0	0.0062	0	0	0	0
Lim.Sup	0	0.0121	0	0	0	0
Lim.Inf	0	-0.0121	0	0	0	0
Maximo	0	0.0272	0	0	0	0
Minimo	0	-0.0301	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	2	26	25.1218

16.4 Análisis de la influencia de la variación del ángulo cubierto en la incertidumbre

Con el mismo equipo de medición tomado anteriormente, se modifica la posición del reflector (variable *Preflector*), en una articulación (articulación 3), fijando 3 valores diferentes, y comprobando su influencia en la incertidumbre obtenida. Se fijan los valores (0,50,0), (0,70,20) y (20,100,50), en coordenadas (X,Y,Z).

16.4.1 Simulación 31 – 100.000 iteraciones – Laptop



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9999	0	2.1992e-04	294.9996	-3.3398e-04	80	9.7505e-06	-1.1284e-04	90.0000	-8.9277e-05	180.0000	0
Incertidumbre	0.0351	0	0.0714	0.1698	0.2646	0	0.0106	0.0258	0.1080	0.0052	0.1080	0
Lim.Sup	335.0680	0	0.1387	295.3299	0.5143	80	0.0205	0.0499	90.2113	0.0100	180.2102	0
Lim.Inf	334.9315	0	-0.1398	294.6679	-0.5186	80	-0.0207	-0.0501	89.7895	-0.0102	179.7889	0
Maximo	335.1442	0	0.2766	295.6528	0.9811	80	0.0421	0.1012	90.4235	0.0224	180.4022	0
Minimo	334.8576	0	-0.2977	294.3806	-0.9846	80	-0.0435	-0.1046	89.6100	-0.0203	179.5721	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0010	269.9999	89.9999	4.8620e-05	4.9012e-06	0	90.0000	-1.3111e-05	90.0001	89.9999	90.0000	0
Incertidumbre	0.1556	0.0135	0.0386	0.5519	0.0029	0	0.0112	0.0053	0.0522	0.0519	0.0016	0
Lim.Sup	75.3024	270.0263	90.0746	1.0737	0.0058	0	90.0217	0.0103	90.1014	90.1008	90.0031	0
Lim.Inf	74.6979	269.9735	89.9241	-1.0776	-0.0057	0	89.9782	-0.0104	89.8982	89.8984	89.9969	0
Maximo	75.6238	270.0647	90.1661	2.1280	0.0119	0	90.0448	0.0210	90.2020	90.1936	90.0065	0
Minimo	74.3806	269.9451	89.8357	-2.1666	-0.0121	0	89.9568	-0.0232	89.7923	89.8042	89.9936	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-6.6229e-06	0	0	0	0
Incertidumbre	0	0.0062	0	0	0	0
Lim.Sup	0	0.0121	0	0	0	0
Lim.Inf	0	-0.0121	0	0	0	0
Maximo	0	0.0248	0	0	0	0
Minimo	0	-0.0262	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	2	24	59.7504

16.4.2 Simulación 32 – 100.000 iteraciones – PC i7

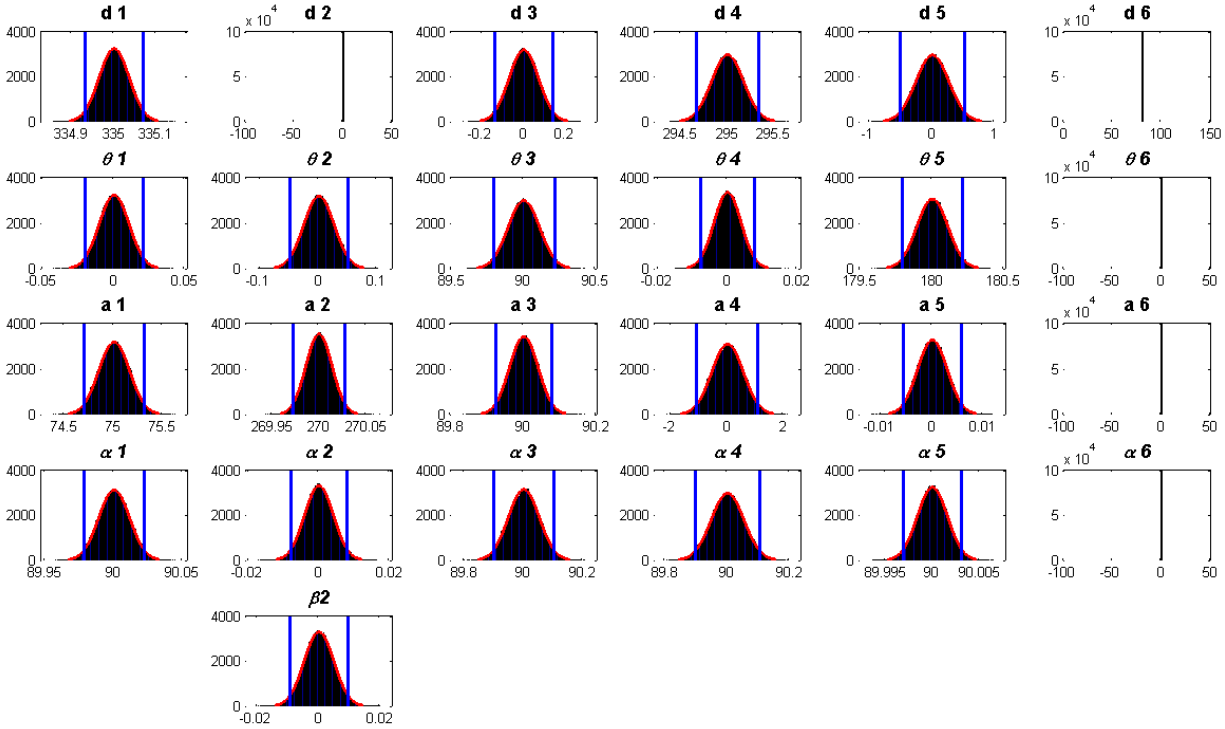


Tabla de datos de Parámetros

File Edit View Insert Tools Desktop Window Help

	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9999	0	2.2048e-04	294.9996	-3.3398e-04	80	9.7505e-06	-1.1275e-04	90.0000	-8.8938e-05	180.0000	0
Incertidumbre	0.0351	0	0.0712	0.1698	0.2646	0	0.0106	0.0258	0.1080	0.0039	0.1080	0
Lim.Sup	335.0680	0	0.1381	295.3299	0.5143	80	0.0205	0.0499	90.2113	0.0076	180.2102	0
Lim.Inf	334.9315	0	-0.1392	294.6674	-0.5186	80	-0.0207	-0.0501	89.7895	-0.0078	179.7889	0
Maximo	335.1442	0	0.2743	295.6529	0.9811	80	0.0421	0.1012	90.4238	0.0176	180.4022	0
Minimo	334.8576	0	-0.2967	294.3826	-0.9846	80	-0.0435	-0.1050	89.6104	-0.0153	179.5721	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0010	269.9999	89.9999	4.8620e-05	4.9012e-06	0	90.0000	-1.1196e-05	90.0001	89.9999	90.0000	0
Incertidumbre	0.1556	0.0138	0.0387	0.5519	0.0029	0	0.0112	0.0041	0.0521	0.0519	0.0016	0
Lim.Sup	75.3024	270.0268	90.0746	1.0737	0.0058	0	90.0217	0.0080	90.1012	90.1008	90.0031	0
Lim.Inf	74.6979	269.9729	89.9239	-1.0776	-0.0057	0	89.9782	-0.0080	89.8984	89.8984	89.9969	0
Maximo	75.6238	270.0651	90.1670	2.1280	0.0119	0	90.0448	0.0165	90.2046	90.1936	90.0065	0
Minimo	74.3806	269.9426	89.8335	-2.1666	-0.0121	0	89.9568	-0.0174	89.7925	89.8042	89.9936	0

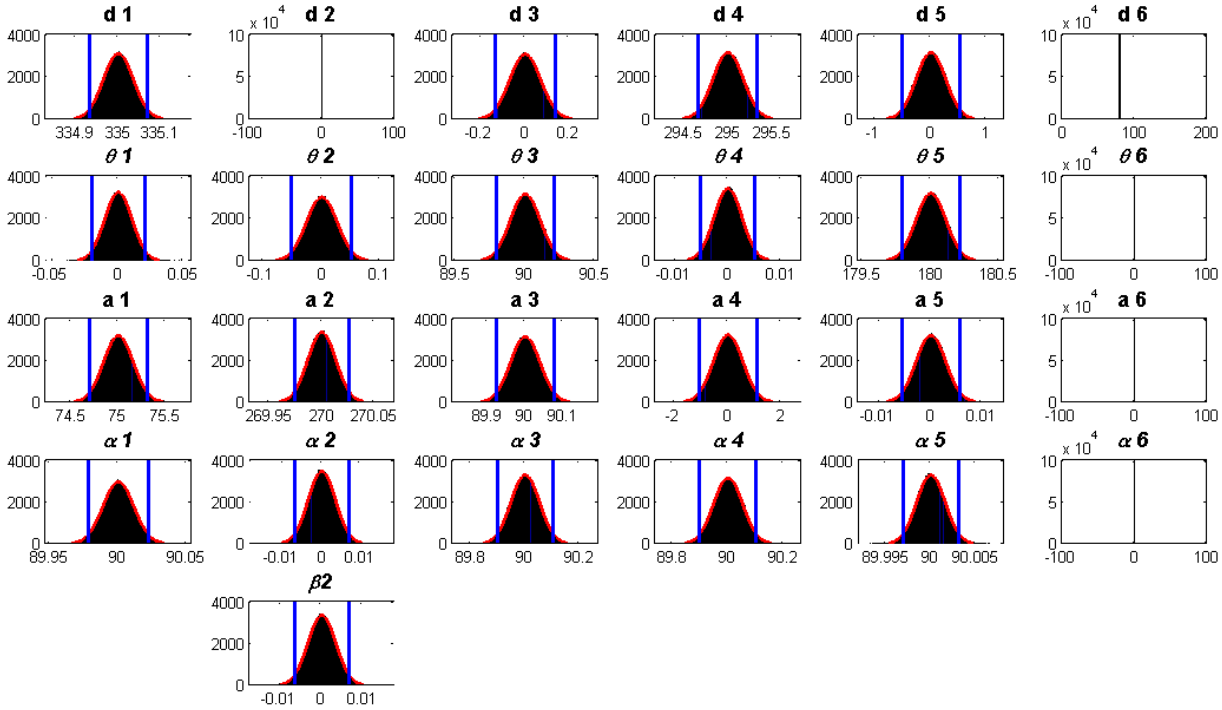
	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-6.6426e-06	0	0	0	0
Incertidumbre	0	0.0048	0	0	0	0
Lim.Sup	0	0.0094	0	0	0	0
Lim.Inf	0	-0.0094	0	0	0	0
Maximo	0	0.0199	0	0	0	0
Minimo	0	-0.0197	0	0	0	0

Duración de la Simulación

File Edit View Insert Tools Desktop Window Help

	Horas	Minutos	Segundos
Tiempo empleado	4	1	35.6926

16.4.3 Simulación 33 – 100.000 iteraciones – Laptop



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9999	0	1.2320e-04	294.9988	0.0012	80	-2.9504e-05	-3.7775e-05	89.9995	-1.1033e-04	180.0005	0
Incertidumbre	0.0352	0	0.0713	0.1699	0.2649	0	0.0106	0.0260	0.1081	0.0026	0.1080	0
Lim.Sup	335.0689	0	0.1395	295.3274	0.5170	80	0.0206	0.0507	90.2093	0.0050	180.2113	0
Lim.Inf	334.9308	0	-0.1396	294.6662	-0.5128	80	-0.0208	-0.0507	89.7889	-0.0052	179.7911	0
Maximo	335.1345	0	0.2750	295.6725	1.0960	80	0.0467	0.1020	90.4364	0.0111	180.4335	0
Minimo	334.8566	0	-0.2785	294.3151	-1.0177	80	-0.0399	-0.0958	89.5757	-0.0116	179.5619	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0004	270.0000	89.9999	0.0028	1.3587e-05	0	90.0000	-1.5837e-06	90.0003	90.0002	90.0000	0
Incertidumbre	0.1565	0.0135	0.0386	0.5522	0.0029	0	0.0113	0.0036	0.0520	0.0520	0.0016	0
Lim.Sup	75.3067	270.0264	90.0751	1.0807	0.0058	0	90.0220	0.0070	90.1017	90.1014	90.0031	0
Lim.Inf	74.6949	269.9737	89.9243	-1.0677	-0.0057	0	89.9782	-0.0070	89.8995	89.8995	89.9969	0
Maximo	75.6435	270.0576	90.1486	2.2769	0.0120	0	90.0402	0.0156	90.2217	90.2175	90.0065	0
Minimo	74.3795	269.9435	89.8411	-2.2271	-0.0118	0	89.9558	-0.0158	89.7875	89.8017	89.9932	0

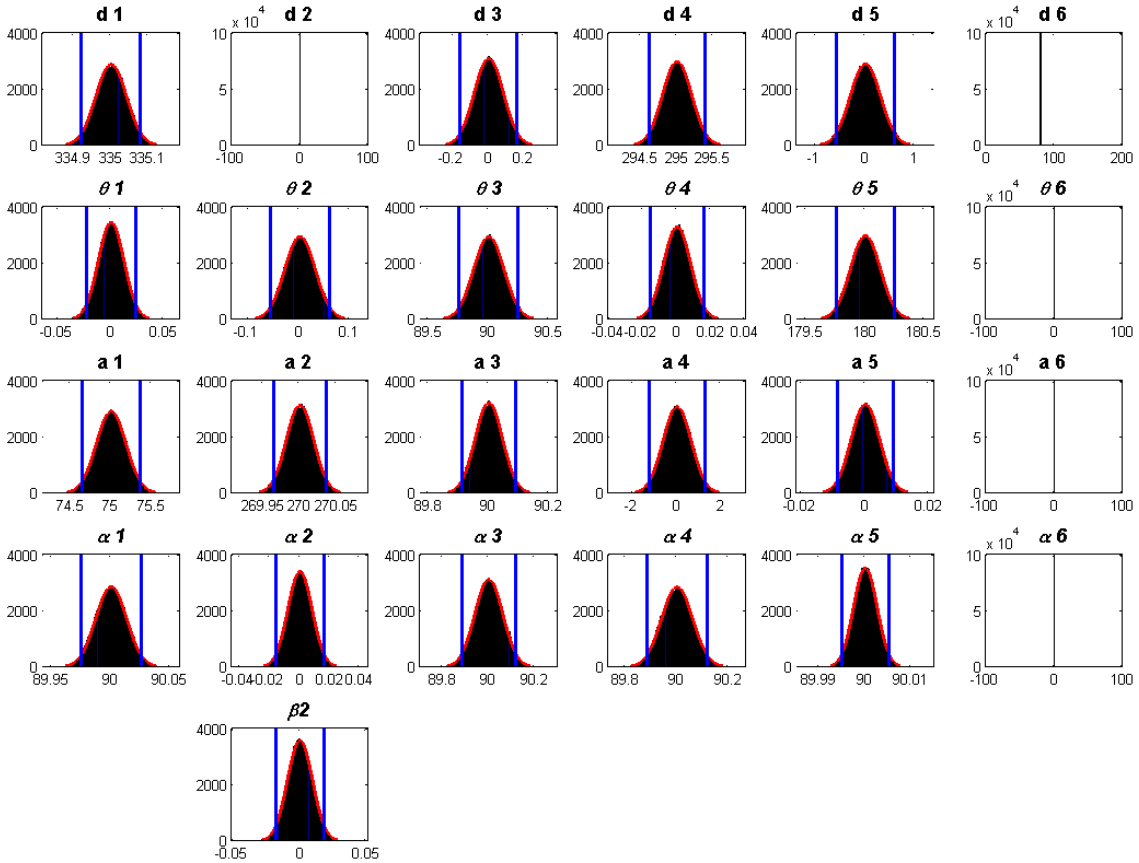
	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	2.3883e-06	0	0	0	0
Incertidumbre	0	0.0035	0	0	0	0
Lim.Sup	0	0.0068	0	0	0	0
Lim.Inf	0	-0.0068	0	0	0	0
Maximo	0	0.0149	0	0	0	0
Minimo	0	-0.0144	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	2	26	55.6594

16.5 Análisis de la influencia de la variación del número de puntos considerado que se miden para cada articulación

Para el mismo caso considerado anteriormente, se varía el número de puntos considerados (10, 40, 100).

16.5.1 Simulación 41 – 100.000 iteraciones - CPS



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	335.0001	0	-6.4037e-04	294.9996	3.1224e-06	80	-5.1673e-05	1.3159e-04	90.0001	-1.0380e-04	179.9999	0
Incertidumbre	0.0410	0	0.0834	0.1973	0.3076	0	0.0121	0.0299	0.1254	0.0080	0.1254	0
Lim.Sup	335.0792	0	0.1610	295.3826	0.5996	80	0.0236	0.0578	90.2444	0.0155	180.2440	0
Lim.Inf	334.9204	0	-0.1634	294.6137	-0.5973	80	-0.0238	-0.0581	89.7553	-0.0157	179.7566	0
Maximo	335.1469	0	0.3279	295.7118	1.1674	80	0.0542	0.1112	90.4606	0.0327	180.4832	0
Minimo	334.8465	0	-0.3265	294.2189	-1.0973	80	-0.0518	-0.1121	89.5244	-0.0339	179.5385	0

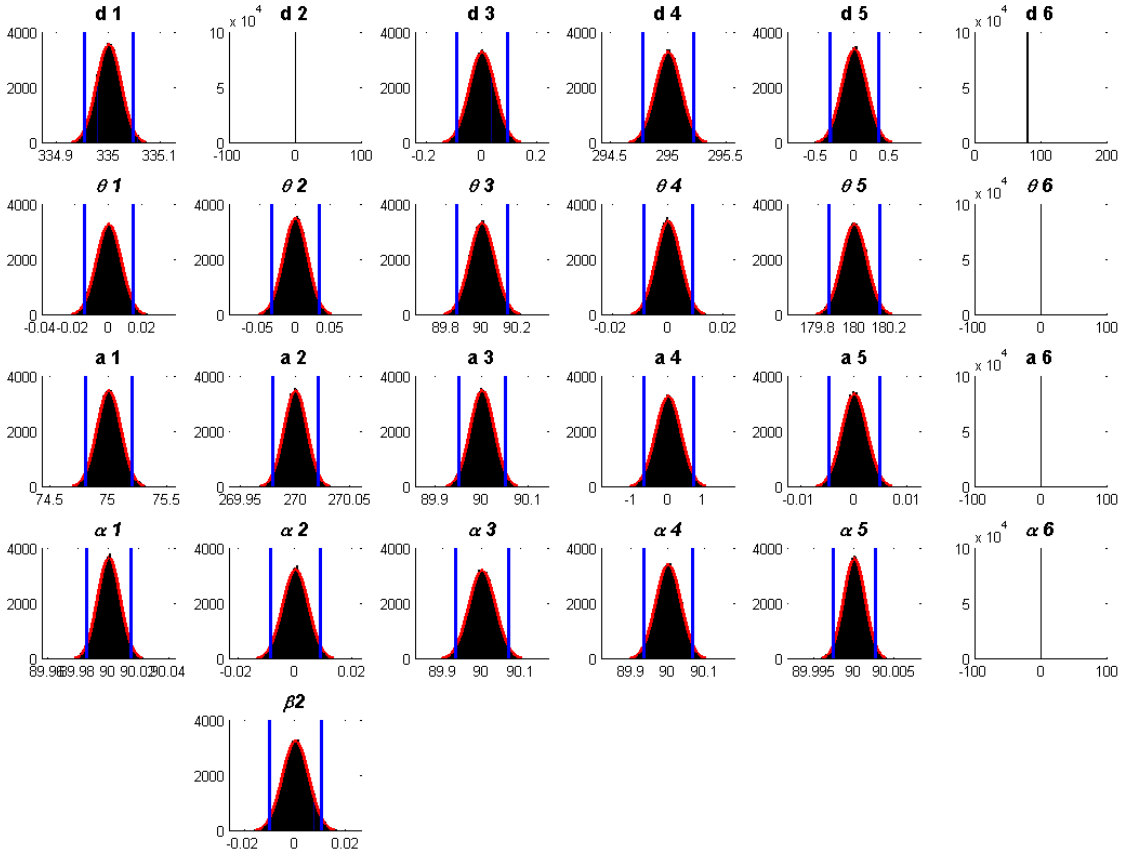
	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	74.9994	270.0000	89.9997	-3.1857e-04	-1.1096e-05	0	89.9999	-1.8527e-05	89.9999	90.0000	90.0000	0
Incertidumbre	0.1803	0.0184	0.0453	0.6411	0.0045	0	0.0130	0.0080	0.0610	0.0604	0.0026	0
Lim.Sup	75.3511	270.0360	90.0876	1.2472	0.0086	0	90.0252	0.0156	90.1182	90.1175	90.0050	0
Lim.Inf	74.6483	269.9641	89.9116	-1.2452	-0.0087	0	89.9747	-0.0156	89.8819	89.8827	89.9950	0
Maximo	75.6972	270.0769	90.1892	2.6115	0.0179	0	90.0485	0.0376	90.2497	90.2235	90.0104	0
Minimo	74.3655	269.9301	89.8205	-2.3864	-0.0178	0	89.9538	-0.0315	89.7673	89.7877	89.9875	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-2.1526e-05	0	0	0	0
Incertidumbre	0	0.0094	0	0	0	0
Lim.Sup	0	0.0183	0	0	0	0
Lim.Inf	0	-0.0183	0	0	0	0
Maximo	0	0.0427	0	0	0	0
Minimo	0	-0.0424	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	4	18	28.3243

Ejecutada en ordenador con S.O. Windows XP Professional, versión 2002, SP3, con una CPU Intel Pentium 4, 2.40GHz, 1.00Gb RAM.

16.5.2 Simulación 42 – 100.000 iteraciones - CPS



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9999	0	1.0344e-04	294.9990	0.0013	80	-3.7379e-05	-5.8949e-05	89.9995	-4.3637e-05	180.0005	0
Incertidumbre	0.0234	0	0.0476	0.1123	0.1748	0	0.0074	0.0171	0.0714	0.0045	0.0714	0
Lim.Sup	335.0455	0	0.0933	295.2175	0.3425	80	0.0144	0.0334	90.1390	0.0088	180.1399	0
Lim.Inf	334.9541	0	-0.0933	294.7788	-0.3405	80	-0.0145	-0.0335	89.8600	-0.0088	179.8611	0
Maximo	335.1059	0	0.1941	295.4785	0.7689	80	0.0281	0.0733	90.3094	0.0183	180.2813	0
Minimo	334.8951	0	-0.2027	294.5440	-0.7312	80	-0.0334	-0.0779	89.7097	-0.0201	179.6889	0

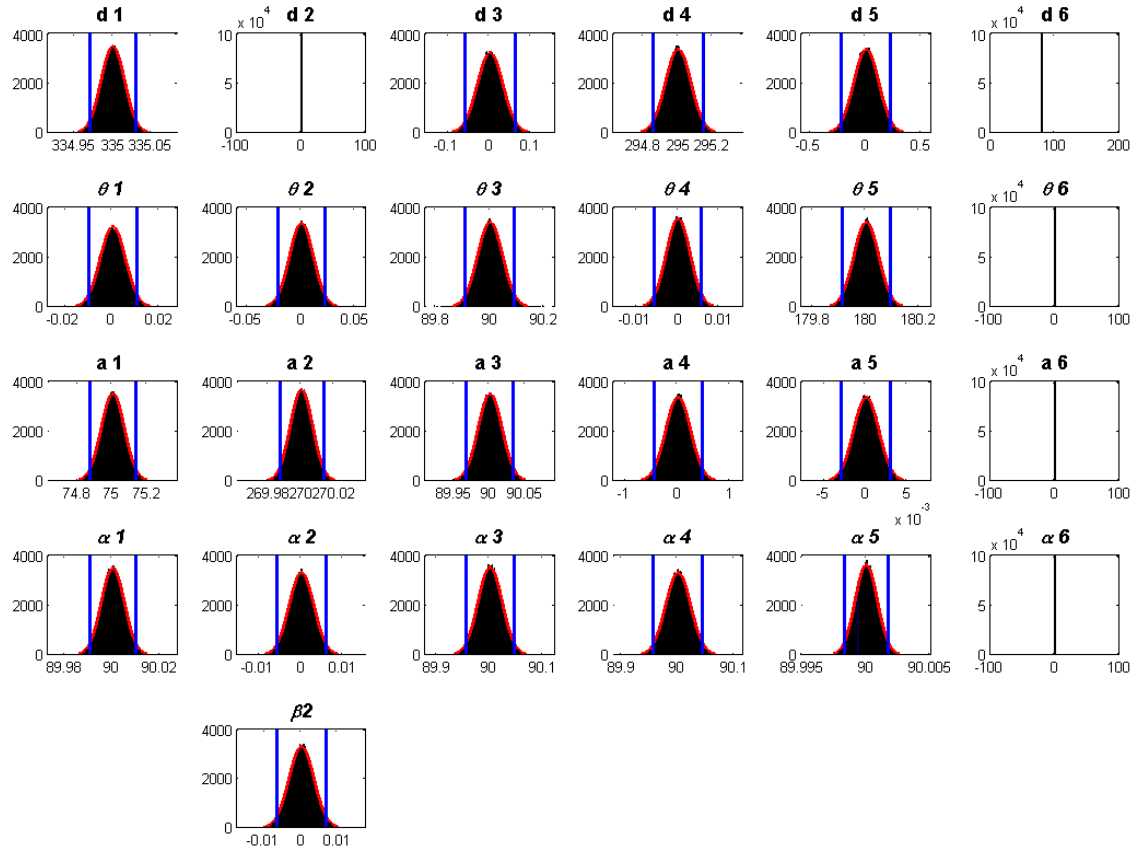
	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0005	270.0000	89.9999	0.0026	1.0473e-06	0	90.0000	2.0650e-05	90.0002	90.0003	90.0000	0
Incertidumbre	0.1032	0.0104	0.0258	0.3648	0.0024	0	0.0074	0.0045	0.0346	0.0343	0.0013	0
Lim.Sup	75.2025	270.0204	90.0503	0.7159	0.0047	0	90.0146	0.0088	90.0676	90.0672	90.0026	0
Lim.Inf	74.7987	269.9795	89.9494	-0.7083	-0.0047	0	89.9855	-0.0089	89.9328	89.9333	89.9974	0
Maximo	75.4773	270.0501	90.1082	1.4217	0.0099	0	90.0367	0.0175	90.1364	90.1520	90.0053	0
Minimo	74.5703	269.9583	89.8801	-1.5956	-0.0105	0	89.9677	-0.0195	89.8563	89.8562	89.9931	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	1.2352e-05	0	0	0	0
Incertidumbre	0	0.0053	0	0	0	0
Lim.Sup	0	0.0104	0	0	0	0
Lim.Inf	0	-0.0104	0	0	0	0
Maximo	0	0.0214	0	0	0	0
Minimo	0	-0.0220	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	7	11	13.5057

Ejecutada en ordenador con S.O. Windows XP Professional, versión 2002, SP3, con una CPU Intel Pentium 4, 2.40GHz, 1.00Gb RAM.

16.5.3 Simulación 43 – 100.000 iteraciones - CPS



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	335.0000	0	2.2041e-05	294.9996	4.9671e-04	80	2.4504e-05	-2.4700e-05	89.9998	-2.6126e-05	180.0002	0
Incertidumbre	0.0153	0	0.0309	0.0728	0.1131	0	0.0054	0.0112	0.0463	0.0029	0.0463	0
Lim.Sup	335.0298	0	0.0608	295.1423	0.2223	80	0.0104	0.0218	90.0906	0.0057	180.0911	0
Lim.Inf	334.9699	0	-0.0604	294.8561	-0.2217	80	-0.0104	-0.0218	89.9088	-0.0057	179.9092	0
Maximo	335.0702	0	0.1215	295.3115	0.4756	80	0.0234	0.0500	90.1999	0.0132	180.2034	0
Minimo	334.9357	0	-0.1310	294.6871	-0.4893	80	-0.0209	-0.0450	89.8002	-0.0130	179.8014	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0002	270.0000	90.0000	0.0012	3.1776e-06	0	90.0000	1.1994e-05	90.0001	90.0001	90.0000	0
Incertidumbre	0.0671	0.0068	0.0168	0.2369	0.0015	0	0.0048	0.0029	0.0225	0.0222	8.5752e-04	0
Lim.Sup	75.1320	270.0134	90.0329	0.4655	0.0030	0	90.0095	0.0058	90.0441	90.0438	90.0017	0
Lim.Inf	74.8683	269.9867	89.9670	-0.4638	-0.0030	0	89.9905	-0.0057	89.9560	89.9565	89.9983	0
Maximo	75.2837	270.0328	90.0735	1.0307	0.0066	0	90.0202	0.0128	90.1008	90.0901	90.0037	0
Minimo	74.6861	269.9693	89.9238	-0.9998	-0.0065	0	89.9774	-0.0118	89.8998	89.9039	89.9958	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	1.6300e-05	0	0	0	0
Incertidumbre	0	0.0034	0	0	0	0
Lim.Sup	0	0.0068	0	0	0	0
Lim.Inf	0	-0.0067	0	0	0	0
Maximo	0	0.0148	0	0	0	0
Minimo	0	-0.0140	0	0	0	0

Duración de la Simulación

File Edit View Insert Tools Desktop Window Help

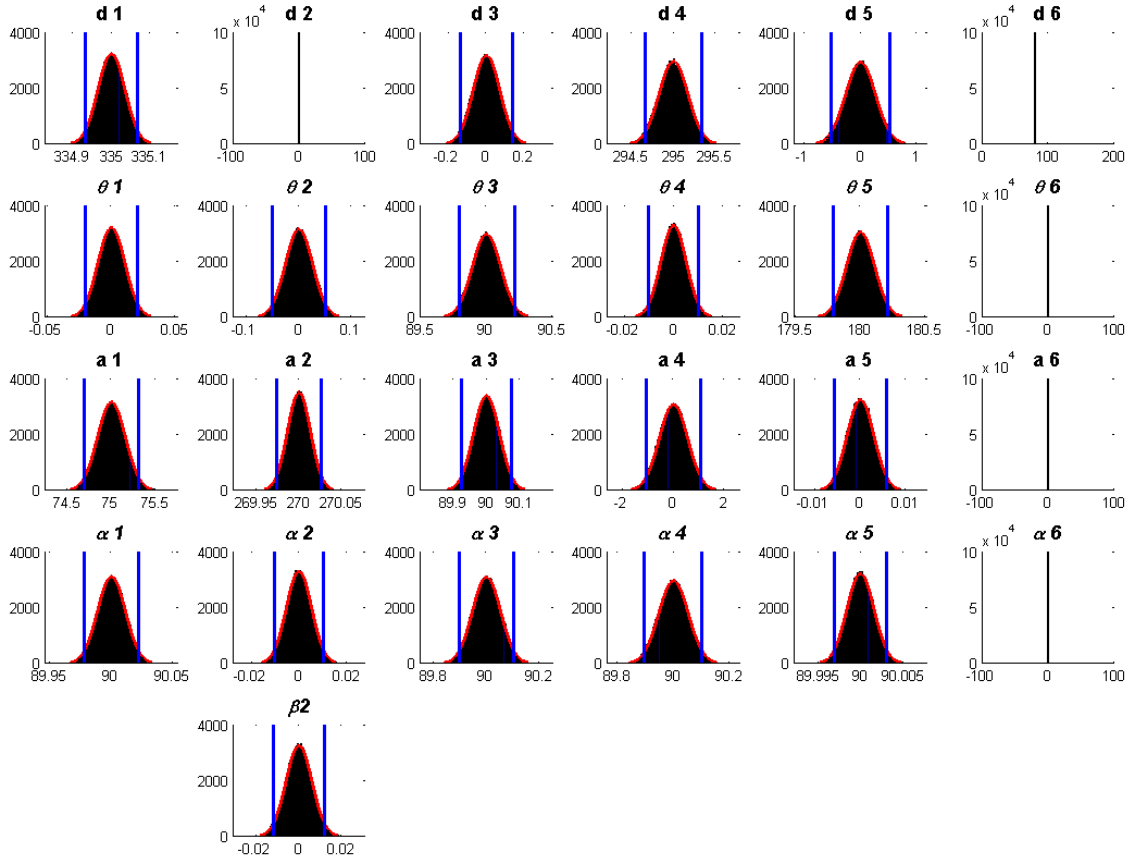
	Horas	Minutos	Segundos
Tiempo empleado	13	2	36.3972

Ejecutada en ordenador con S.O. Windows XP Professional, versión 2002, SP3, con una CPU Intel Pentium 4, 2.40GHz, 0.99Gb RAM.

16.6 Análisis de la influencia de la variación de la matriz de transformación del SR del robot al equipo de medición

Para el mismo caso de equipo de medición, se cambia la matriz de cambio del robot al Láser Tracker, considerando 3 valores diferentes (ver los desplazamientos tomados y las matrices calculadas en el apartado 15.2 Cálculo de las diferentes matrices de transformación para la simulación, página 82).

16.6.1 Simulación 51 – 100.000 iteraciones – CPS



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9999	0	2.1992e-04	294.9996	-3.3398e-04	80	9.7505e-06	-1.1284e-04	90.0000	-8.9277e-05	180.0000	0
Incertidumbre	0.0351	0	0.0714	0.1698	0.2646	0	0.0106	0.0258	0.1080	0.0052	0.1080	0
Lim.Sup	335.0680	0	0.1387	295.3299	0.5143	80	0.0205	0.0499	90.2113	0.0100	180.2102	0
Lim.Inf	334.9315	0	-0.1398	294.6679	-0.5186	80	-0.0207	-0.0501	89.7895	-0.0102	179.7889	0
Maximo	335.1442	0	0.2766	295.6528	0.9811	80	0.0421	0.1012	90.4235	0.0224	180.4022	0
Minimo	334.8576	0	-0.2977	294.3806	-0.9846	80	-0.0435	-0.1046	89.6100	-0.0203	179.5721	0

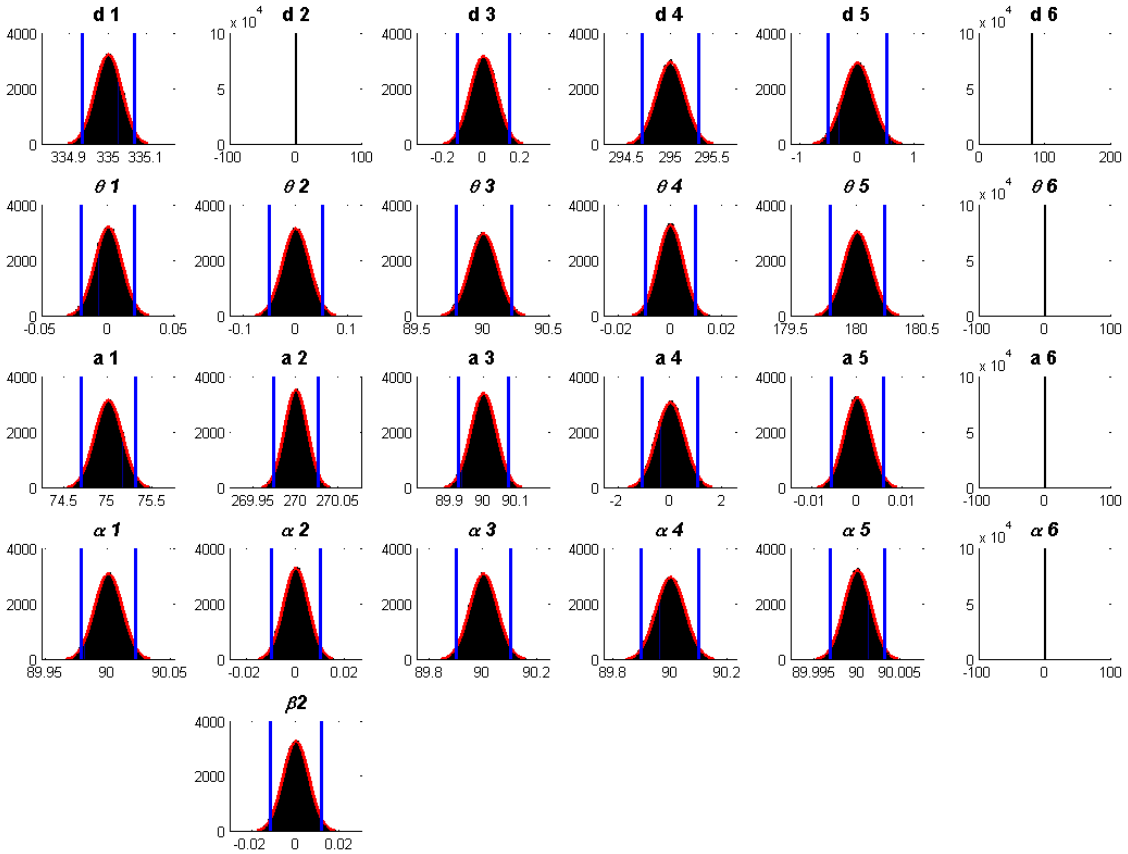
	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0010	269.9999	89.9999	4.8620e-05	4.9012e-06	0	90.0000	-1.3111e-05	90.0001	89.9999	90.0000	0
Incertidumbre	0.1556	0.0135	0.0386	0.5519	0.0029	0	0.0112	0.0053	0.0522	0.0519	0.0016	0
Lim.Sup	75.3024	270.0263	90.0746	1.0737	0.0058	0	90.0217	0.0103	90.1014	90.1008	90.0031	0
Lim.Inf	74.6979	269.9735	89.9241	-1.0776	-0.0057	0	89.9782	-0.0104	89.8982	89.8984	89.9969	0
Maximo	75.6238	270.0647	90.1661	2.1280	0.0119	0	90.0448	0.0210	90.2020	90.1936	90.0065	0
Minimo	74.3806	269.9451	89.8357	-2.1666	-0.0121	0	89.9568	-0.0232	89.7923	89.8042	89.9936	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-6.6229e-06	0	0	0	0
Incertidumbre	0	0.0062	0	0	0	0
Lim.Sup	0	0.0121	0	0	0	0
Lim.Inf	0	-0.0121	0	0	0	0
Maximo	0	0.0248	0	0	0	0
Minimo	0	-0.0262	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	5	28	1.7629

Ejecutada en ordenador con S.O. Windows XP Professional, versión 2002, SP3, con una CPU Intel Pentium 4, 2.40GHz, 1.00Gb RAM.

16.6.2 Simulación 52 – 100.000 iteraciones – CPS



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9999	0	2.0086e-04	294.9996	-3.0016e-04	80	8.1951e-06	-1.1299e-04	90.0000	-8.9020e-05	180.0000	0
Incertidumbre	0.0351	0	0.0711	0.1683	0.2619	0	0.0102	0.0258	0.1070	0.0050	0.1070	0
Lim.Sup	335.0679	0	0.1381	295.3271	0.5096	80	0.0199	0.0499	90.2092	0.0096	180.2083	0
Lim.Inf	334.9315	0	-0.1391	294.6713	-0.5130	80	-0.0201	-0.0501	89.7919	-0.0098	179.7911	0
Maximo	335.1440	0	0.2779	295.6455	0.9687	80	0.0412	0.1010	90.4189	0.0214	180.3984	0
Minimo	334.8576	0	-0.2937	294.3869	-0.9764	80	-0.0418	-0.1046	89.6135	-0.0195	179.5767	0

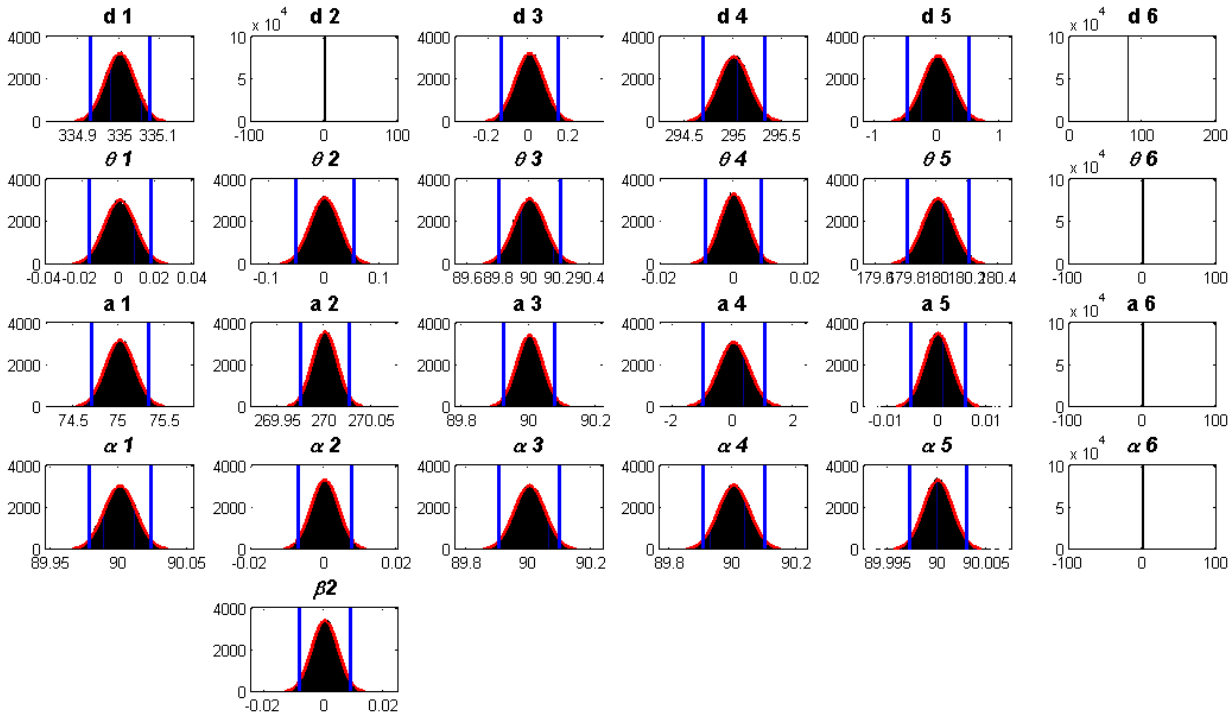
	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0009	269.9999	89.9999	1.1036e-04	3.4434e-06	0	90.0000	-1.2903e-05	90.0001	90.0000	90.0000	0
Incertidumbre	0.1552	0.0134	0.0383	0.5465	0.0029	0	0.0112	0.0051	0.0517	0.0515	0.0016	0
Lim.Sup	75.3018	270.0262	90.0741	1.0632	0.0057	0	90.0216	0.0099	90.1005	90.0999	90.0031	0
Lim.Inf	74.6987	269.9736	89.9246	-1.0664	-0.0057	0	89.9782	-0.0099	89.8990	89.8993	89.9969	0
Maximo	75.6226	270.0645	90.1646	2.1076	0.0117	0	90.0444	0.0198	90.1986	90.1913	90.0064	0
Minimo	74.3841	269.9451	89.8349	-2.1397	-0.0121	0	89.9569	-0.0222	89.7949	89.8058	89.9937	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-7.0881e-06	0	0	0	0
Incertidumbre	0	0.0059	0	0	0	0
Lim.Sup	0	0.0116	0	0	0	0
Lim.Inf	0	-0.0116	0	0	0	0
Maximo	0	0.0238	0	0	0	0
Minimo	0	-0.0250	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	5	33	17.4736

Ejecutada en ordenador con S.O. Windows XP Professional, versión 2002, SP3, con una CPU Intel Pentium 4, 2.40GHz, 1.00Gb RAM.

16.6.3 Simulación 53 – 100.000 iteraciones – Laptop



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9999	0	1.6630e-04	294.9993	1.6505e-04	80	8.3048e-06	-1.1874e-04	89.9998	-7.7398e-05	180.0002	0
Incertidumbre	0.0368	0	0.0730	0.1607	0.2485	0	0.0086	0.0272	0.1022	0.0040	0.1022	0
Lim.Sup	335.0713	0	0.1421	295.3120	0.4832	80	0.0167	0.0526	90.1999	0.0077	180.1992	0
Lim.Inf	334.9282	0	-0.1422	294.6851	-0.4841	80	-0.0168	-0.0528	89.8006	-0.0078	179.8008	0
Maximo	335.1494	0	0.3087	295.6159	0.9801	80	0.0326	0.1052	90.3969	0.0170	180.3951	0
Minimo	334.8496	0	-0.2870	294.3695	-0.9761	80	-0.0336	-0.1104	89.5997	-0.0160	179.5965	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0009	269.9999	89.9999	0.0011	-2.3816e-06	0	90.0000	-1.1351e-05	90.0001	90.0000	90.0000	0
Incertidumbre	0.1616	0.0133	0.0387	0.5195	0.0028	0	0.0117	0.0038	0.0493	0.0491	0.0015	0
Lim.Sup	75.3137	270.0260	90.0748	1.0153	0.0055	0	90.0226	0.0075	90.0959	90.0956	90.0029	0
Lim.Inf	74.6868	269.9739	89.9241	-1.0135	-0.0055	0	89.9773	-0.0075	89.9039	89.9043	89.9971	0
Maximo	75.6729	270.0647	90.1526	1.9996	0.0125	0	90.0451	0.0169	90.1842	90.1918	90.0062	0
Minimo	74.3742	269.9438	89.8179	-2.0514	-0.0123	0	89.9552	-0.0153	89.8040	89.8102	89.9937	0

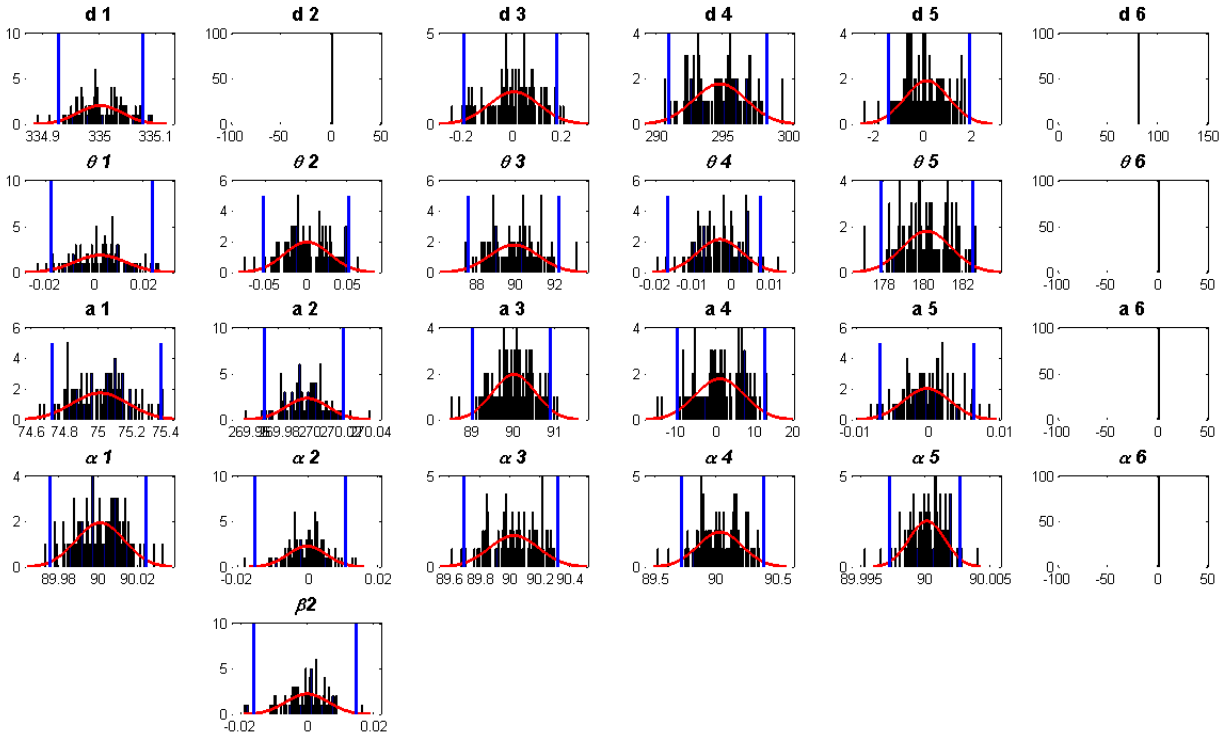
	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-1.3123e-05	0	0	0	0
Incertidumbre	0	0.0045	0	0	0	0
Lim.Sup	0	0.0088	0	0	0	0
Lim.Inf	0	-0.0088	0	0	0	0
Maximo	0	0.0180	0	0	0	0
Minimo	0	-0.0207	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	2	25	50.6777

16.7 Análisis de la influencia de la variación del número de iteraciones ejecutado en los cálculos

Se consideran dos equipos, un PC i7 64 bits, y un ordenador portátil.

16.7.1 Simulación 21 – 100 iteraciones – PC i7



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9998	0	0.0019	294.6998	0.0968	80	0.0018	-7.6922e-04	89.8530	-0.0034	180.1465	0
Incertidumbre	0.0391	0	0.1026	1.9767	0.9103	0	0.0104	0.0280	1.2542	0.0059	1.2523	0
Lim.Sup	335.0745	0	0.1774	298.1770	1.8498	80	0.0232	0.0500	92.1542	0.0073	182.4748	0
Lim.Inf	334.9264	0	-0.2044	290.8446	-1.5077	80	-0.0182	-0.0547	87.5206	-0.0172	177.8313	0
Maximo	335.0911	0	0.2054	299.3899	1.8757	80	0.0259	0.0627	93.0533	0.0121	182.6329	0
Minimo	334.8897	0	-0.2537	290.5530	-2.5050	80	-0.0237	-0.0779	87.3605	-0.0199	176.9696	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0046	269.9976	89.9897	0.7431	-3.7280e-04	0	90.0003	-6.6982e-04	90.0191	90.0190	90.0000	0
Incertidumbre	0.1664	0.0137	0.5133	6.3328	0.0032	0	0.0121	0.0054	0.1626	0.1798	0.0014	0
Lim.Sup	75.3640	270.0214	90.8551	12.2508	0.0061	0	90.0235	0.0102	90.3093	90.3692	90.0025	0
Lim.Inf	74.7193	269.9702	88.9908	-10.4037	-0.0069	0	89.9752	-0.0155	89.6869	89.7071	89.9972	0
Maximo	75.3745	270.0383	91.0393	12.9153	0.0083	0	90.0318	0.0132	90.3137	90.3818	90.0040	0
Minimo	74.6431	269.9576	88.4945	-15.4741	-0.0082	0	89.9726	-0.0184	89.6053	89.5116	89.9952	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-6.1307e-04	0	0	0	0
Incertidumbre	0	0.0063	0	0	0	0
Lim.Sup	0	0.0142	0	0	0	0
Lim.Inf	0	-0.0166	0	0	0	0
Maximo	0	0.0162	0	0	0	0
Minimo	0	-0.0194	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	0	0	18.2865

16.7.2 Simulación 21 – 1.000 iteraciones – PC i7

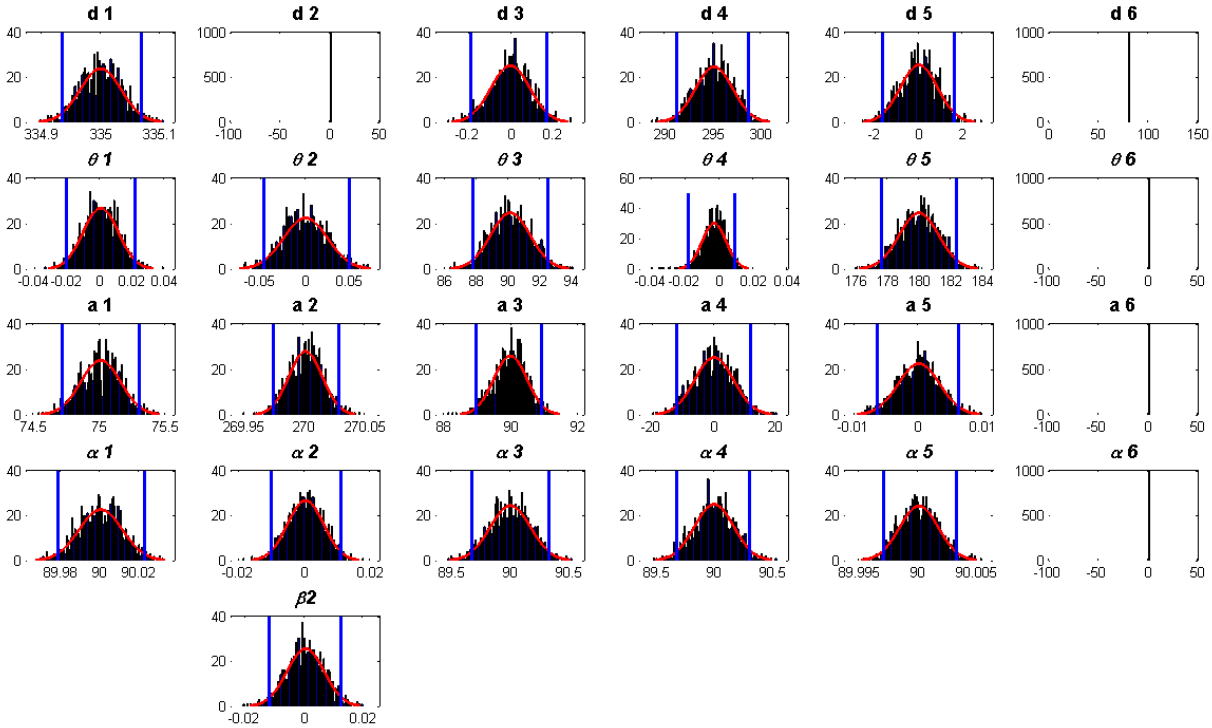


Tabla de datos de Parámetros

File Edit View Insert Tools Desktop Window Help

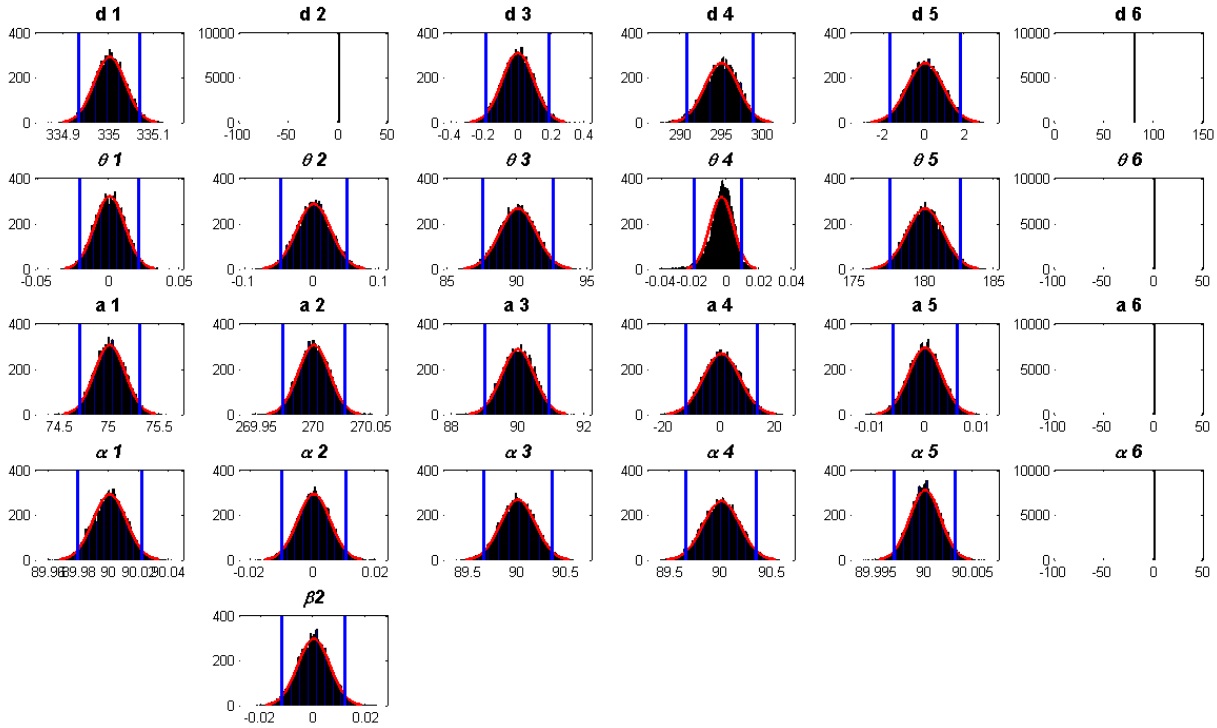
	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9983	0	-0.0097	295.0374	-0.0507	80	3.9727e-06	-7.2310e-04	90.0671	-0.0037	179.9335	0
Incertidumbre	0.0344	0	0.0927	1.9297	0.8572	0	0.0109	0.0251	1.2353	0.0071	1.2358	0
Lim.Sup	335.0672	0	0.1643	298.5343	1.5570	80	0.0209	0.0491	92.4148	0.0084	182.2682	0
Lim.Inf	334.9335	0	-0.2009	291.2056	-1.7421	80	-0.0214	-0.0488	87.7353	-0.0188	177.5681	0
Maximo	335.1030	0	0.2801	300.5304	2.8258	80	0.0387	0.0706	93.9668	0.0130	183.6899	0
Minimo	334.8961	0	-0.3077	288.5907	-2.6755	80	-0.0347	-0.0717	86.3022	-0.0406	176.0283	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0026	270.0006	89.9525	-0.3162	-5.1808e-05	0	90.0001	4.5020e-06	89.9928	89.9897	90.0000	0
Incertidumbre	0.1500	0.0137	0.4958	6.3328	0.0032	0	0.0111	0.0054	0.1704	0.1675	0.0016	0
Lim.Sup	75.2971	270.0274	90.8794	11.3528	0.0061	0	90.0223	0.0106	90.3140	90.2953	90.0032	0
Lim.Inf	74.7081	269.9739	88.9471	-12.3601	-0.0066	0	89.9783	-0.0103	89.6636	89.6679	89.9970	0
Maximo	75.4366	270.0452	91.3188	19.9032	0.0097	0	90.0320	0.0176	90.5112	90.5223	90.0053	0
Minimo	74.5302	269.9495	88.0939	-20.1986	-0.0085	0	89.9691	-0.0189	89.4654	89.4723	89.9953	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	1.3442e-04	0	0	0	0
Incertidumbre	0	0.0063	0	0	0	0
Lim.Sup	0	0.0122	0	0	0	0
Lim.Inf	0	-0.0124	0	0	0	0
Maximo	0	0.0195	0	0	0	0
Minimo	0	-0.0211	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	0	2	57.0512

16.7.3 Simulación 21 – 10.000 iteraciones – PC i7



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	335.0002	0	-0.0080	294.9063	0.0053	80	3.8910e-05	1.2710e-04	89.9868	-0.0038	180.0134	0
Incertidumbre	0.0348	0	0.0959	2.0239	0.9018	0	0.0106	0.0255	1.2869	0.0072	1.2869	0
Lim.Sup	335.0676	0	0.1768	298.6697	1.7501	80	0.0203	0.0497	92.5169	0.0082	182.5332	0
Lim.Inf	334.9317	0	-0.2000	290.7329	-1.7585	80	-0.0210	-0.0496	87.4602	-0.0207	177.4814	0
Maximo	335.1188	0	0.3658	301.0570	2.9315	80	0.0421	0.0924	94.4000	0.0161	184.2807	0
Minimo	334.8625	0	-0.3822	287.4762	-3.0943	80	-0.0435	-0.0924	85.7226	-0.0416	175.6068	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	74.9995	269.9998	89.9642	0.0820	-1.7552e-05	0	89.9999	-4.8520e-05	90.0027	90.0007	90.0000	0
Incertidumbre	0.1537	0.0136	0.4912	6.6130	0.0031	0	0.0111	0.0052	0.1757	0.1757	0.0016	0
Lim.Sup	75.2951	270.0263	90.8923	13.0277	0.0061	0	90.0211	0.0102	90.3439	90.3411	90.0031	0
Lim.Inf	74.7002	269.9731	88.9799	-12.8529	-0.0060	0	89.9783	-0.0103	89.6592	89.6572	89.9969	0
Maximo	75.5634	270.0506	91.6473	22.4573	0.0117	0	90.0412	0.0199	90.5587	90.5651	90.0061	0
Minimo	74.3806	269.9460	88.0960	-22.2176	-0.0112	0	89.9598	-0.0189	89.3731	89.4035	89.9936	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-7.9149e-05	0	0	0	0
Incertidumbre	0	0.0061	0	0	0	0
Lim.Sup	0	0.0119	0	0	0	0
Lim.Inf	0	-0.0121	0	0	0	0
Maximo	0	0.0236	0	0	0	0
Minimo	0	-0.0222	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	0	30	40.0043

16.7.4 Simulación 21 – 100.000 iteraciones – PC i7

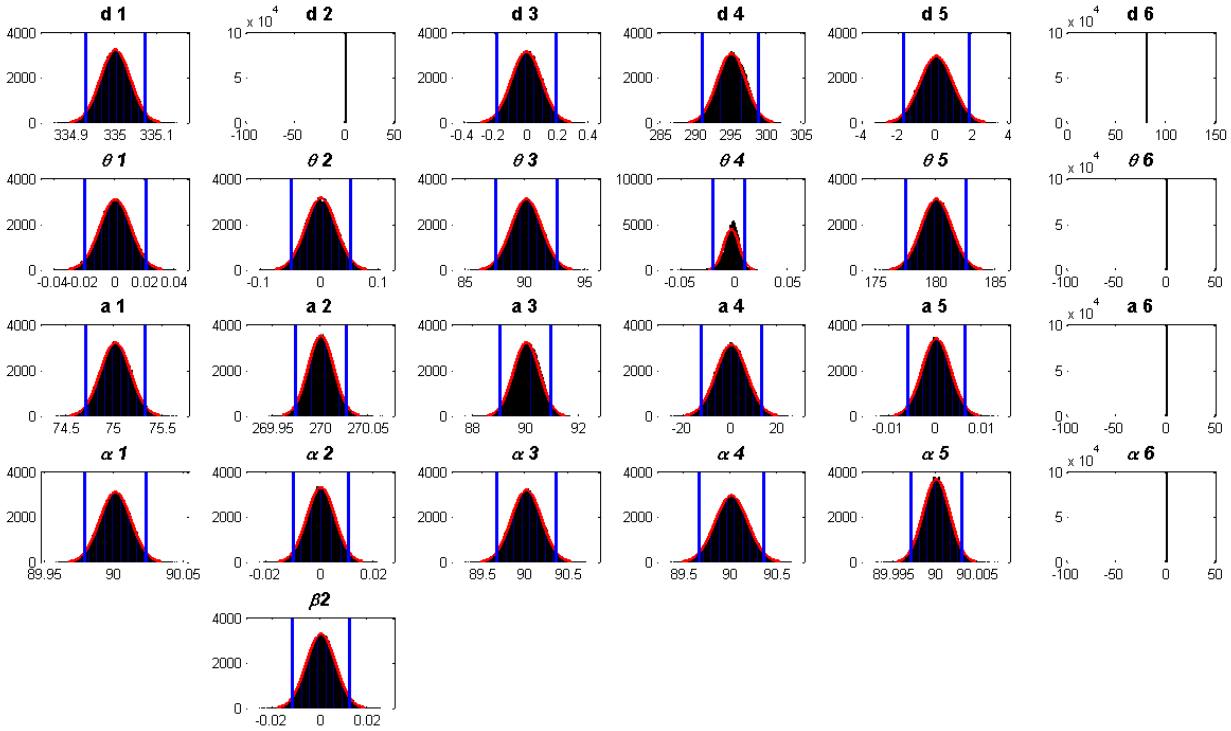


Tabla de datos de Parámetros

File Edit View Insert Tools Desktop Window Help

	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9999	0	-0.0070	294.9266	2.3815e-04	80	-1.2325e-05	-7.1586e-05	90.0002	-0.0038	179.9998	0
Incertidumbre	0.0352	0	0.0968	2.0320	0.9061	0	0.0105	0.0259	1.2928	0.0073	1.2928	0
Lim.Sup	335.0683	0	0.1794	298.6814	1.7718	80	0.0205	0.0502	92.5227	0.0082	182.5163	0
Lim.Inf	334.9314	0	-0.1989	290.7678	-1.7696	80	-0.0207	-0.0503	87.4833	-0.0208	177.4754	0
Maximo	335.1442	0	0.3675	301.9432	3.3835	80	0.0404	0.1012	95.1845	0.0205	184.9421	0
Minimo	334.8576	0	-0.4030	286.1837	-3.3566	80	-0.0416	-0.1046	85.0710	-0.0616	174.8014	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0007	269.9999	89.9676	0.0087	2.5346e-06	0	90.0000	-9.7765e-06	90.0001	89.9998	90.0000	0
Incertidumbre	0.1561	0.0135	0.4901	6.6489	0.0031	0	0.0112	0.0053	0.1765	0.1764	0.0016	0
Lim.Sup	75.3041	270.0263	90.8846	12.9653	0.0061	0	90.0218	0.0104	90.3430	90.3430	90.0031	0
Lim.Inf	74.6965	269.9736	88.9760	-12.9535	-0.0061	0	89.9782	-0.0104	89.6550	89.6551	89.9969	0
Maximo	75.6435	270.0647	91.6252	26.3900	0.0134	0	90.0448	0.0210	90.6983	90.6515	90.0075	0
Minimo	74.3795	269.9451	87.6293	-26.2290	-0.0132	0	89.9568	-0.0232	89.2764	89.3379	89.9927	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-1.2308e-07	0	0	0	0
Incertidumbre	0	0.0062	0	0	0	0
Lim.Sup	0	0.0121	0	0	0	0
Lim.Inf	0	-0.0121	0	0	0	0
Maximo	0	0.0250	0	0	0	0
Minimo	0	-0.0262	0	0	0	0

Duración de la Simulación

File Edit View Insert Tools Desktop Window Help

	Horas	Minutos	Segundos
Tiempo empleado	4	42	13.7049

16.7.5 Simulación 21 – 500.000 iteraciones – PC i7

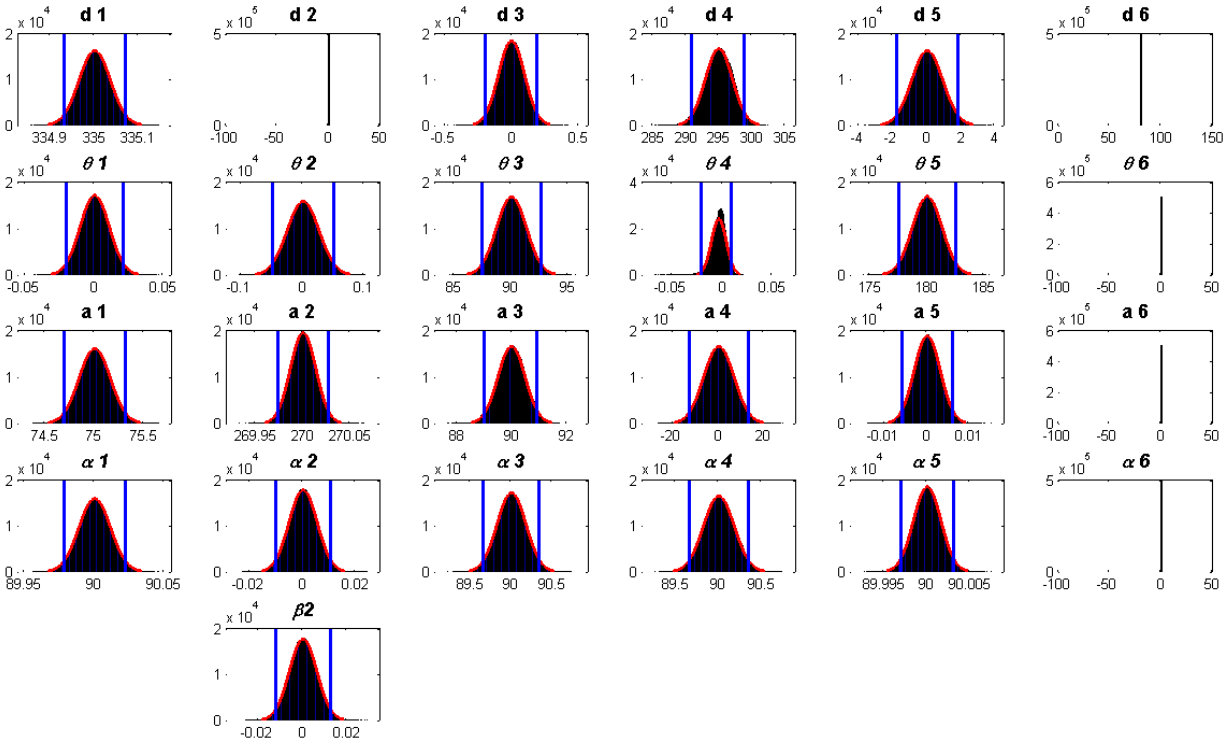


Tabla de datos de Parámetros

File Edit View Insert Tools Desktop Window Help

	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9999	0	-0.0073	294.9258	3.6488e-04	80	-8.4790e-06	-5.8389e-05	89.9997	-0.0038	180.0004	0
Incertidumbre	0.0351	0	0.0966	2.0320	0.9062	0	0.0106	0.0259	1.2925	0.0073	1.2925	0
Lim.Sup	335.0683	0	0.1793	298.6762	1.7658	80	0.0205	0.0503	92.5195	0.0081	182.5142	0
Lim.Inf	334.9313	0	-0.1989	290.7717	-1.7594	80	-0.0207	-0.0504	87.4852	-0.0209	177.4811	0
Maximo	335.1442	0	0.4120	302.3658	3.7563	80	0.0467	0.1014	95.5866	0.0208	185.3960	0
Minimo	334.8566	0	-0.4858	285.2476	-3.6695	80	-0.0446	-0.1046	84.6069	-0.0674	174.4014	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0004	270.0000	89.9661	0.0130	2.2012e-06	0	90.0000	-2.5260e-06	90.0003	89.9998	90.0000	0
Incertidumbre	0.1560	0.0135	0.4910	6.6484	0.0031	0	0.0112	0.0053	0.1764	0.1764	0.0016	0
Lim.Sup	75.3045	270.0264	90.8868	12.9847	0.0061	0	90.0218	0.0104	90.3430	90.3423	90.0031	0
Lim.Inf	74.6968	269.9735	88.9761	-12.9112	-0.0061	0	89.9782	-0.0103	89.6562	89.6568	89.9969	0
Maximo	75.6435	270.0647	91.7341	28.4103	0.0153	0	90.0457	0.0243	90.7401	90.7252	90.0075	0
Minimo	74.3764	269.9316	87.6293	-27.1606	-0.0142	0	89.9556	-0.0232	89.2091	89.2613	89.9926	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-2.9844e-06	0	0	0	0
Incertidumbre	0	0.0062	0	0	0	0
Lim.Sup	0	0.0121	0	0	0	0
Lim.Inf	0	-0.0121	0	0	0	0
Maximo	0	0.0290	0	0	0	0
Minimo	0	-0.0262	0	0	0	0

Duración de la Simulación

File Edit View Insert Tools Desktop Window Help

	Horas	Minutos	Segundos
Tiempo empleado	18	8	52.0519

16.7.6 Simulación 22 – 100 iteraciones – Laptop

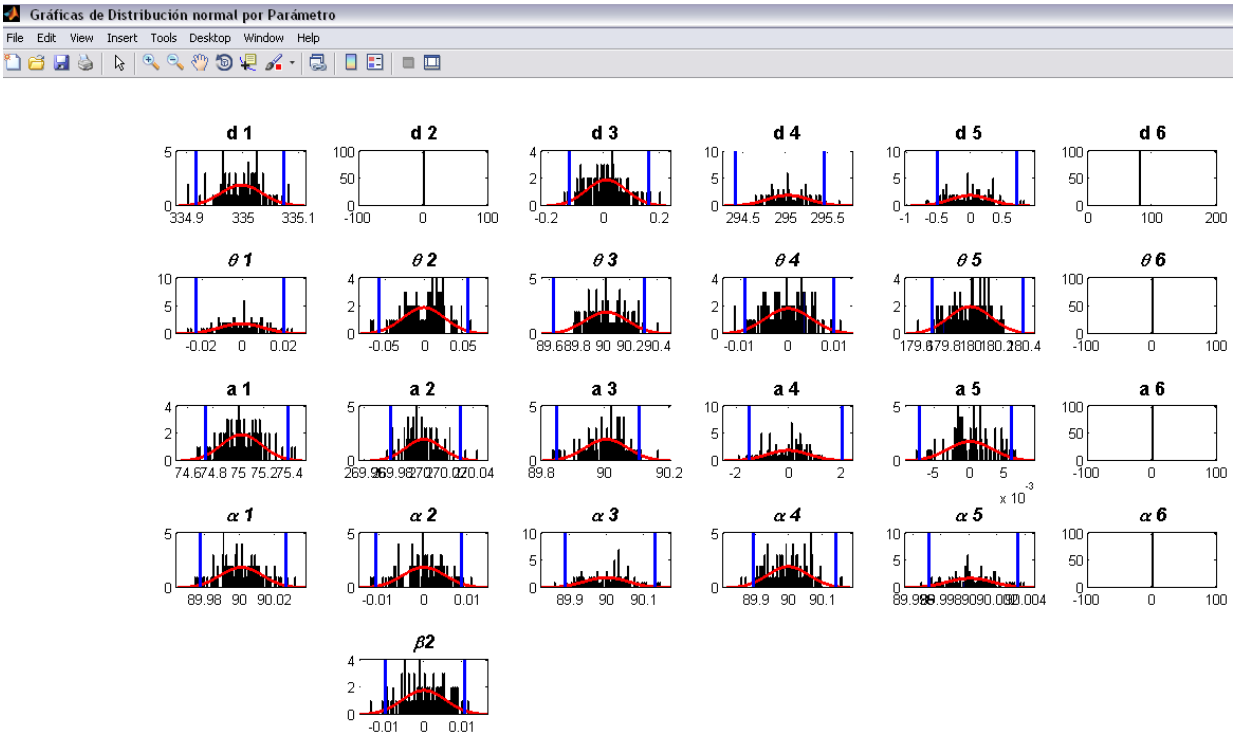


Tabla de datos de Parámetros

File Edit View Insert Tools Desktop Window Help

	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	334.9948	0	0.0046	295.0181	-0.0274	80	-0.0015	-0.0027	90.0125	-7.4597e-05	179.9885	0
Incertidumbre	0.0372	0	0.0718	0.2507	0.3127	0	0.0109	0.0270	0.1595	0.0049	0.1584	0
Lim.Sup	335.0703	0	0.1584	295.4476	0.7029	80	0.0192	0.0533	90.2929	0.0097	180.3812	0
Lim.Inf	334.9150	0	-0.1267	294.3894	-0.5361	80	-0.0235	-0.0609	89.6184	-0.0093	179.7049	0
Maximo	335.0793	0	0.1993	295.6270	0.8204	80	0.0234	0.0580	90.4008	0.0111	180.3885	0
Minimo	334.8988	0	-0.1474	294.3790	-0.7190	80	-0.0280	-0.0717	89.6109	-0.0116	179.6010	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0157	270.0025	90.0005	-0.0589	-2.7600e-05	0	90.0004	-1.7673e-04	89.9965	89.9947	90.0000	0
Incertidumbre	0.1589	0.0132	0.0658	0.8112	0.0032	0	0.0120	0.0048	0.0588	0.0615	0.0016	0
Lim.Sup	75.3833	270.0297	90.1032	1.9922	0.0060	0	90.0255	0.0083	90.1261	90.1366	90.0034	0
Lim.Inf	74.7387	269.9779	89.8475	-1.5648	-0.0073	0	89.9773	-0.0110	89.8858	89.8921	89.9971	0
Maximo	75.4391	270.0419	90.1620	2.0164	0.0066	0	90.0310	0.0106	90.1409	90.1561	90.0038	0
Minimo	74.6661	269.9753	89.8332	-1.9778	-0.0078	0	89.9741	-0.0123	89.8583	89.8543	89.9965	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-1.8444e-04	0	0	0	0
Incertidumbre	0	0.0057	0	0	0	0
Lim.Sup	0	0.0105	0	0	0	0
Lim.Inf	0	-0.0103	0	0	0	0
Maximo	0	0.0115	0	0	0	0
Minimo	0	-0.0142	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	0	0	9.2879

16.7.7 Simulación 22 – 1.000 iteraciones – Laptop

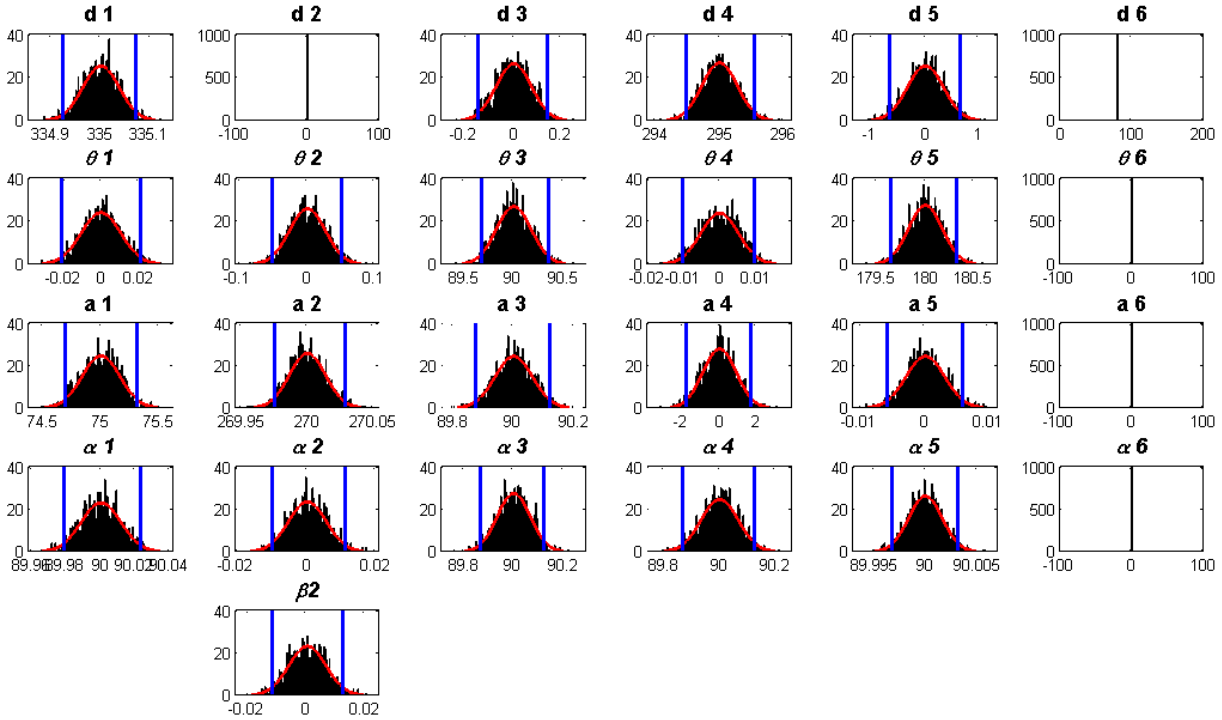


Tabla de datos de Parámetros

File Edit View Insert Tools Desktop Window Help

	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	335.0002	0	-9.2463e-04	294.9994	-0.0023	80	-4.3880e-04	2.9311e-04	90.0005	-3.2464e-04	179.9997	0
Incertidumbre	0.0364	0	0.0742	0.2745	0.3372	0	0.0107	0.0267	0.1747	0.0052	0.1747	0
Lim.Sup	335.0696	0	0.1380	295.5432	0.6386	80	0.0208	0.0512	90.3522	0.0094	180.3237	0
Lim.Inf	334.9253	0	-0.1494	294.4847	-0.6763	80	-0.0216	-0.0523	89.6745	-0.0106	179.6455	0
Maximo	335.1200	0	0.2520	295.9328	1.1164	80	0.0321	0.0888	90.6032	0.0145	180.5901	0
Minimo	334.8861	0	-0.2439	294.0627	-1.0583	80	-0.0332	-0.0852	89.4153	-0.0170	179.3778	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	75.0000	270.0002	90.0004	-0.0024	6.4519e-05	0	89.9999	8.9975e-05	90.0003	89.9996	89.9999	0
Incertidumbre	0.1605	0.0136	0.0616	0.8971	0.0030	0	0.0115	0.0055	0.0661	0.0658	0.0016	0
Lim.Sup	75.3104	270.0287	90.1196	1.6595	0.0059	0	90.0228	0.0111	90.1227	90.1226	90.0030	0
Lim.Inf	74.6928	269.9758	89.8739	-1.7927	-0.0059	0	89.9787	-0.0100	89.8678	89.8668	89.9968	0
Maximo	75.4898	270.0439	90.1813	3.0702	0.0095	0	90.0321	0.0173	90.2191	90.2132	90.0057	0
Minimo	74.4878	269.9553	89.8024	-3.2409	-0.0092	0	89.9649	-0.0152	89.7582	89.8016	89.9950	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	2.6786e-04	0	0	0	0
Incertidumbre	0	0.0064	0	0	0	0
Lim.Sup	0	0.0124	0	0	0	0
Lim.Inf	0	-0.0119	0	0	0	0
Maximo	0	0.0208	0	0	0	0
Minimo	0	-0.0167	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	0	1	31.3868

16.7.8 Simulación 22 – 10.000 Iteraciones – Laptop

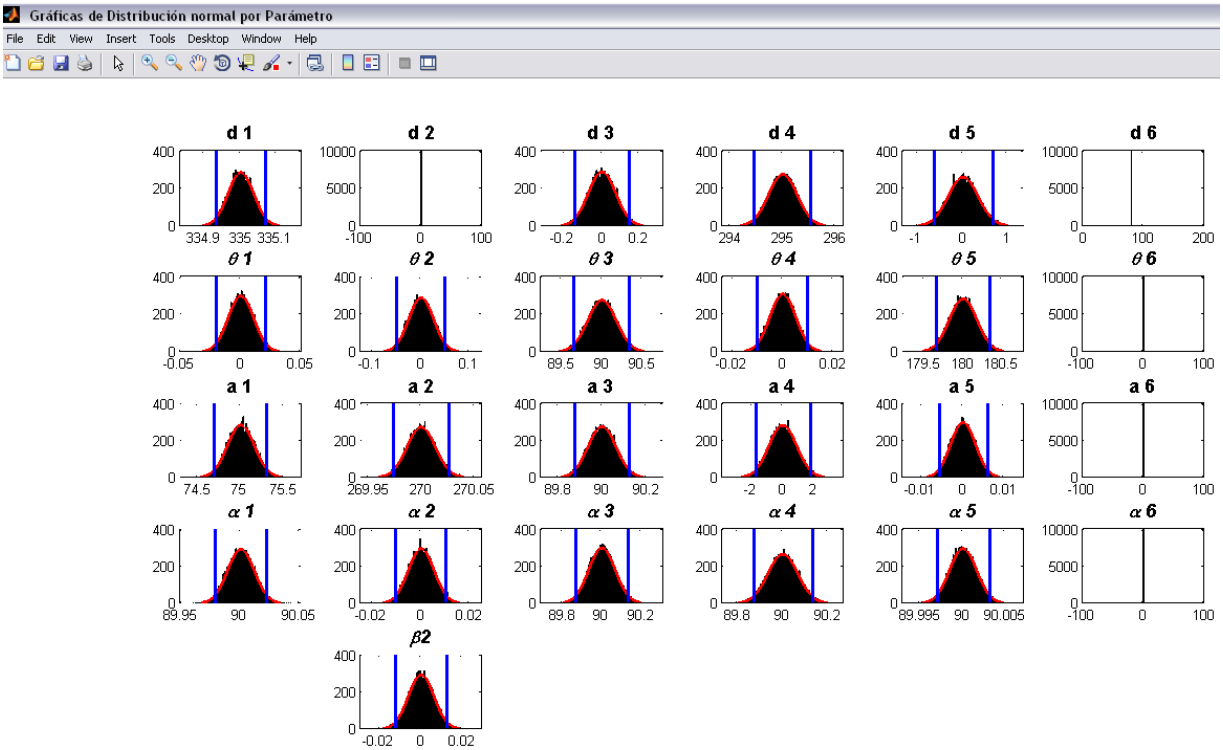


Tabla de datos de Parámetros

File Edit View Insert Tools Desktop Window Help

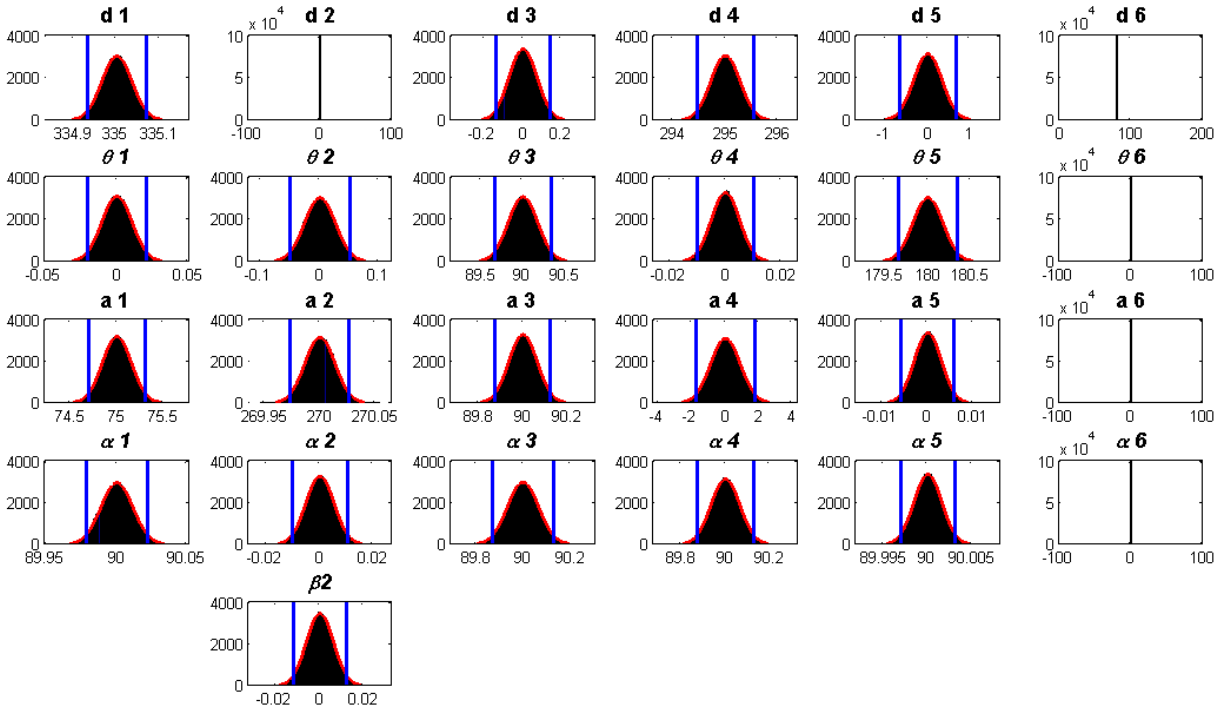
	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	335.0002	0	-8.0984e-04	294.9984	0.0012	80	8.5351e-05	1.0865e-04	89.9997	-1.5514e-04	180.0002	0
Incertidumbre	0.0353	0	0.0728	0.2745	0.3351	0	0.0106	0.0259	0.1747	0.0052	0.1749	0
Lim.Sup	335.0681	0	0.1436	295.5278	0.6657	80	0.0209	0.0500	90.3387	0.0097	180.3438	0
Lim.Inf	334.9317	0	-0.1439	294.4528	-0.6499	80	-0.0203	-0.0501	89.6544	-0.0102	179.6613	0
Maximo	335.1158	0	0.2615	295.9644	1.0818	80	0.0377	0.0831	90.6236	0.0198	180.6081	0
Minimo	334.8610	0	-0.2709	294.0581	-1.1257	80	-0.0418	-0.1051	89.4119	-0.0204	179.3567	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	74.9995	269.9999	89.9989	0.0020	-5.8148e-05	0	90.0000	1.3050e-05	90.0000	90.0002	90.0000	0
Incertidumbre	0.1553	0.0134	0.0626	0.8965	0.0030	0	0.0113	0.0053	0.0662	0.0655	0.0016	0
Lim.Sup	75.3007	270.0262	90.1211	1.7794	0.0058	0	90.0216	0.0104	90.1312	90.1299	90.0032	0
Lim.Inf	74.6973	269.9740	89.8765	-1.7485	-0.0059	0	89.9783	-0.0104	89.8704	89.8741	89.9970	0
Maximo	75.5804	270.0444	90.2284	3.1386	0.0122	0	90.0429	0.0186	90.2450	90.2169	90.0059	0
Minimo	74.4674	269.9523	89.7919	-3.2619	-0.0104	0	89.9596	-0.0209	89.7447	89.7770	89.9941	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-2.0672e-05	0	0	0	0
Incertidumbre	0	0.0062	0	0	0	0
Lim.Sup	0	0.0123	0	0	0	0
Lim.Inf	0	-0.0123	0	0	0	0
Maximo	0	0.0214	0	0	0	0
Minimo	0	-0.0247	0	0	0	0

	Horas	Minutos	Segundos
Tiempo empleado	0	15	15.8115

16.7.9 Simulación 22 – 100.000 iteraciones – Laptop



	d1	d2	d3	d4	d5	d6	thita 1	thita 2	thita 3	thita 4	thita 5	thita 6
Media	335.0002	0	-6.1040e-04	294.9997	-0.0010	80	-2.8224e-06	1.7057e-04	90.0006	-1.8631e-04	179.9994	0
Incertidumbre	0.0352	0	0.0731	0.2737	0.3346	0	0.0106	0.0259	0.1742	0.0051	0.1741	0
Lim.Sup	335.0687	0	0.1424	295.5286	0.6503	80	0.0207	0.0505	90.3399	0.0098	180.3392	0
Lim.Inf	334.9318	0	-0.1429	294.4633	-0.6528	80	-0.0207	-0.0502	89.6618	-0.0102	179.6599	0
Maximo	335.1294	0	0.3158	295.9857	1.4145	80	0.0422	0.1006	90.6398	0.0215	180.6965	0
Minimo	334.8600	0	-0.3080	293.8682	-1.2372	80	-0.0406	-0.0971	89.2892	-0.0207	179.3630	0

	a1	a2	a3	a4	a5	a6	alfa 1	alfa 2	alfa3	alfa 4	alfa 5	alfa 6
Media	74.9992	269.9999	89.9994	-0.0027	3.0058e-06	0	89.9999	-1.1191e-05	89.9998	89.9998	90.0000	0
Incertidumbre	0.1562	0.0134	0.0635	0.8915	0.0030	0	0.0112	0.0053	0.0657	0.0655	0.0016	0
Lim.Sup	75.3040	270.0263	90.1218	1.7388	0.0058	0	90.0218	0.0103	90.1279	90.1272	90.0031	0
Lim.Inf	74.6964	269.9737	89.8751	-1.7431	-0.0058	0	89.9782	-0.0103	89.8716	89.8725	89.9969	0
Maximo	75.6110	270.0534	90.2554	3.6194	0.0133	0	90.0424	0.0211	90.2540	90.2742	90.0068	0
Minimo	74.3517	269.9464	89.7294	-3.3651	-0.0121	0	89.9586	-0.0224	89.7580	89.7547	89.9935	0

	beta 1	beta 2	beta 3	beta 4	beta 5	beta 6
Media	0	-1.8749e-05	0	0	0	0
Incertidumbre	0	0.0062	0	0	0	0
Lim.Sup	0	0.0120	0	0	0	0
Lim.Inf	0	-0.0120	0	0	0	0
Maximo	0	0.0267	0	0	0	0
Minimo	0	-0.0271	0	0	0	0

Duración de la Simulación

File Edit View Insert Tools Desktop Window Help

	Horas	Minutos	Segundos
Tiempo empleado	2	32	26.5053

17 Anexo IX – Funciones MatLab desarrolladas

Se incluyen aquí el listado de las funciones que se han desarrollado para la realización de este proyecto.

17.1 Genera Puntos Circle-Point

```
function [Ptos_Circle_Point,
Ptos_Original,randomnoise]=Genera_puntos_circle_point(Initpos, SRLT,
Preflector, Angcir, Nptos, Ruidomed, Err, Param5, ShowFigure)

%PARÁMETROS
%-Initpos: Posición inicial del robot a partir de la cual se generarán los
%círculos. Vector 1x6 con los ángulos de las articulaciones.

%- SRLT: Matriz de transformación que pasa puntos del SR global del robot
%al SR de medición del LT. Los puntos generados estarán expresados en el SR
%del LT.

%-Preflector: Matriz 6x3 con las coordenadas X Y Z de la posición del
%reflector del LT (en posición  inicial) expresadas en el SR de la
articulación que gira (fijo) para
%todas las articulaciones:
% [X1 Y1 Z1; X2 Y2 Z2;...;X6 Y6 Z6]

%-Angcir: Vector 1x6 con el ángulo de giro que cubrirá cada articulación en
%el ensayo.

%-Nptos: Vector 1x6 con el número de puntos que se generarán en cada círculo

%-Ruidomed: Desviación estándar de la distribución normal con la que se
%generará aleatoriamente ruido normal DEBIDO AL INSTRUMENTO DE MEDIDA
centrado en el valor nominal en cada
%coordenada. Consideraremos con carácter general un valor de desviación tal
%que 2sigma sea aprox el 95% del error máximo del instrumento de medida
%(LT, Estación total, indoor gps,...). Cero si no se desea ruido. EXPRESADA
%EN MICRÓMETROS. Vector 1x2, que significa: Ruido= A + B*L
%siendo A el primer valor, en microm, y B en microm/m.

%-Err: Matriz 6x20 con los errores de giro a inducir en cada una de las
%articulaciones. Cada fila tiene estas componentes:
%[dx dy dz Ex Ey T1 T2 T3 T4 T5 fi1 fi2 fi3 fi4 fi5 R1 R2 R3 R4 R5]
%Son los términos habituales correspondientes al modelado de errores en ejes
%giratorios (matriz Slocum).dx, dy, dz son los deltas en cada uno de los
%ejes para la traslación respecto al sistema de referencia fijo y Ex, Ey
%son giros entorno a X e Y del sistema fijo. Ez no se considera porque la
%rotación es entorno a Z. Todos los errores se van a generar dependientes del
valor del
%ángulo de giro de la articulación en cada momento, por lo que lo que se
%introducirá en esta matriz serán las amplitudes máximas de estos errores.
%Su valor final se modelará como:
%   A sen((2pi/T)tita + fi)
%por tanto dependiente del valor de giro en cada momento. Siendo:
%-A: el valor introducido en los términos de error (amplitud). En la matriz
milímetros
%para desplazamietos y grados para giros. Si no se desean errores los 5
%primeros términos igual a 0.
%-T: Periodo de la oscilación EN RADIANES (normalmente consideraremos 2pi).
%En la matriz se especifica uno para cada error.
%- tita: Será el valor de giro de la articulación en cada momento
%- fi: fase EN RADIANES, una para cada error en la matriz. (normalmente
consideraremos 0)
%Modelando este error en función de tita, en cada posición de giro de una
```

```
%articulación, su SR estará en una posición distinta para cada tita. Se
%obtendrá la matriz de este sistema al SR nominal de la articulación para
%cada ángulo. Los parámetros R1 a R5 son desviaciones estándar de la
distribución normal con la que se
%generará aleatoriamente ruido normal al valor de cada error.0 si no se
%desea generar ruido.

%- Param5: Vector con los parámetros DH del robot
%- ShowFigure: Muestra o no la figura. Variable booleana (true o false)

%SALIDA
%Ptos_Circle_Point, Ptos_Original: EL primero contiene los puntos generados
%contemplando defectos y ruido, mientras que el segundo contiene la
%trayectoria nominal sin considerar defectos ni ruido. Vector 9 elementos
%por fila, de los que son 3 XYZ y los otros 6 los ángulos de las
%articulaciones en ese punto

%-randomnoise: vector 6x3xn con valores aleatorios sumados a cada coordenada
%dentro del intervalo especificado en Ruidomed

%% ************ INICIALIZA VARIABLES ***********
max_nptos=max(Nptos);
Ptos=zeros(max_nptos,9);
Ptos_Orig=zeros(max_nptos,9);
randomnoise=zeros(6,3,max_nptos);
Incang=zeros(1,6);
Ptos_Circle_Point=zeros(6,max_nptos,9);
Ptos_Original=zeros(6,max_nptos,9);
Posact=Initpos;
%cont=0;

%% ******* PARA CADA ARTICULACION de 1 a 6 ************
for i=1:6
    %Incremento de giro para la generación en la articulación actual
    %i es un valor 1..6 que corresponde a la articulación que se está girando
    Incang(i)=Angcir(i)/(Nptos(i)-1);
    k=0;
    %% **** Para cada punto j espaciado un ángulo Incang ******
    for j=Initpos(i):Incang(i):(Initpos(i)+Angcir(i))% COMPROBAR NUMERO DE
PUNTOS
      %Obtener coordenadas del reflector del LT en SR global robot
      %j cotiene el valor del ángulo, para la articulación actual, y
      % va girando en cada punto
      Angi=Posact;
      Angi(i)=j;
      jrad=deg2rad(j);  %jrad contiene el valor del ángulo en radianes
      %Preflector contiene las coordenadas del reflector para la posición
      %inicial de la articulación expresadas en sistema referencia fijo de la
      %articulación que gira. Se aplica un giro entorno al eje Z de ese
sistema
      %de referencia para ir obteniendo la posición del reflector en ese
      %sistema fijo, y posteriorente en el robot

      %% Calcula la matriz GZs de giro para ese punto SIN errores
      %La matriz de giro sobre Z, sin errores, para hacer esto sería:
      %GZs=[cos(jrad) -sin(jrad) 0 0; sin(jrad) cos(jrad) 0 0; 0 0 1 0;0 0 0
1];
      %No se usa esta matriz - solo como información

      %% Calcula los errores, y calcula la matriz de giro GZ con errores
      %Construimos la matriz de giro con errores
      %[dx dy dz Ex Ey T1 T2 T3 T4 T5 fi1 fi2 fi3 fi4 fi5 R1 R2 R3 R4 R5]
```

```
      %A*sen((2pi/T)tita + fi)
      dx=Err(i,1)*sin((2*pi/Err(i,6))*jrad+Err(i,11));
      dy=Err(i,2)*sin((2*pi/Err(i,7))*jrad+Err(i,12));
      dz=Err(i,3)*sin((2*pi/Err(i,8))*jrad+Err(i,13));
      Ex=deg2rad(Err(i,4))*sin((2*pi/Err(i,9))*jrad+Err(i,14));
      Ey=deg2rad(Err(i,5))*sin((2*pi/Err(i,10))*jrad+Err(i,15));
      %Ruido aleatorio a cada error.
      dx=dx+Err(i,16)*randn(1,1);
      dy=dy+Err(i,17)*randn(1,1);
      dz=dz+Err(i,18)*randn(1,1);
      Ex=Ex+Err(i,19)*randn(1,1);
      Ey=Ey+Err(i,20)*randn(1,1);

      %Matriz de giro sobre Z, considerando los errores
      GZ=[cos(Ey)*cos(jrad) -(cos(Ey)*sin(jrad)) sin(Ey) dx;...
          (sin(Ex)*sin(Ey)*cos(jrad))+(cos(Ex)*sin(jrad))
(cos(Ex)*cos(jrad))-(sin(Ex)*sin(Ey)*sin(jrad)) -(sin(Ex)*cos(Ey)) dy;...
          -(cos(Ex)*sin(Ey)*cos(jrad))+(sin(Ex)*sin(jrad))
(sin(Ex)*cos(jrad))+(cos(Ex)*sin(Ey)*sin(jrad)) cos(Ex)*cos(Ey) dz;...
          0 0 0 1];

      %% ** Calcula el punto del círculo con errores y sin errores ****
      Ptocirloc=GZ*[Preflector(i,:) 1]';%Punto con errores
      %Ptocirlocs=GZs*[Preflector(i,:) 1]';%Punto sin errores - no se usa
      %Solo toma los 3 primeros valores, que son XYZ
      Ptocirloc=Ptocirloc(1:3);     %Punto con errores
      %Ptocirlocs=Ptocirlocs(1:3);   %Punto sin errores- no se usa ya


      %Ptocir en SR global del robot
      %ATENCION:
      %MOTIVADO POR COHERENCIA EN LA DEFINICIÓN DE SRs y
      %porque el método del circle point se basa en esta coherencia.
      %Respecto al modelo del robot original, se cambia el SR4 original por
      %un SR4 con el ejeZ al revés. Este cambio afecta a los parámetros
      %alfa34 y alfa45. ahora alfa34=90 y alfa45=-90. Esto tiene como
      %consecuencia que se deben cambiar de signo todas las lecturas de
      %ángulo del robot EXCEPTO LA DEL EJE 5. Dentro del modelo
      %PBETAfRobotMKR5sixxDHHpar no se cambia de signo la lectura del 5
      %para que esto funcione. como alfa45=-90, también se cambia el SR5
      %para que alfa45 sea 90, por lo que tita45 pasa a ser 0 y tita56 pasa
      %a 180.

      %El modelado de la excentricidad se va a basar en considerar que el
      %SR ligado a una articulación determinada permanece fijo y hay otro
sistema
      %de referencia en esa articulación, que inicialmente es igual que ese,
pero
      %que gira con el giro de la articulación y tiene los efectos de la
excentricidad
      %(de la matriz de errores). La matriz de cambio entre ese SR y el fijo
tiene
      %en cuenta la excentricidad Y YA EL GIRO, por lo que la relación entre
el sistema
      %con excentricidad y el siguiente "fijo" será la relación inicial de
los
      %parámetros nominales. POR ESO SE OBTIENEN LAS MATRICES NOMINALES CON
      %ÁNGULO DE GIRO 0, porque el ángulo lo considera GZ.

      %% Calcula la matriz de cambio del origen a la articulación actual
      %Calcula las matrices de cambio de una articulación a la anterior
      %según D/H-H
      if i>1       %sólo realiza el cálculo si es articulacion 2..6
```

```
          [M0A6 M0A1 M1A2 M2A3 M3A4 M4A5 M5A6] =
PBETAfRobotMKR5sixxDHHpar([0 0 0 0 0 0], Param5);
      end
      %Selecciona el valor adecuado de matriz M según en qué articulación
      %se encuentra
      switch i
          case 1
              M=[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];
          case 2
              M=GZ1*M0A1;
          case 3
              M=GZ1*M0A1*GZ2*M1A2;
          case 4
              M=GZ1*M0A1*GZ2*M1A2*GZ3*M2A3;
          case 5
              M=GZ1*M0A1*GZ2*M1A2*GZ3*M2A3*GZ4*M3A4;
          case 6
              M=GZ1*M0A1*GZ2*M1A2*GZ3*M2A3*GZ4*M3A4*GZ5*M4A5;
      end

      %% Cambia la referencia del punto de la articulación actual, al
      % origen del robot y después al Sistema de Referencia del Láser
      % Tracker
      PtocirRobot=M*[Ptocirloc; 1];
      %Expresar el punto en SR LT de medición
      PtocirLT=SRLT*PtocirRobot;
      PtocirLT=PtocirLT(1:3);
      PtocirLTr=PtocirLT;


      %% Añade ruido aleatorio debido al instrumento de medida
      %Añadir defectos del robot antes del ruido de medida
      %El ruido vendra dado por: Ruido= A + B*L
      %Una vez generado el punto nominal añadir ruido a las 3 coordenadas.
      %Será un ruido aleatorio correspondiente a una distribución normal de
      %media el valor de la coordenada y desviación estándar la indicada en
      %el parámetro.

      %r=(ub-lb)*rand(n,1)+lb;
      k=k+1;
      intervalo=Ruidomed(1)+Ruidomed(2)*(PtocirLTr/1000);%int en microm
      randomnoise(i,:,k)=2*intervalo(1:3).*rand(3,1)-intervalo(1:3);%um
      randomnoise(i,:,k)=randomnoise(i,:,k)/1000;%Pasar de microm a mm

      PtocirLTr(1)=PtocirLTr(1)+randomnoise(i,1,k);
      PtocirLTr(2)=PtocirLTr(2)+randomnoise(i,2,k);
      PtocirLTr(3)=PtocirLTr(3)+randomnoise(i,3,k);

      %% Guardar el punto con/sin ruido debido al instrumento de medida
      Ptos_Orig(k,:)=[PtocirLT(1:3)' -Angi];%Negativo porque el robot Kuka da
los ángulos negativos
      Ptos(k,:)=[PtocirLTr(1:3)' -Angi];
    end
    %% *** Guarda puntos del circulo de la articulación actual en las
    % variables de salida y pasa a la siguiente articulación
    Ptos_Circle_Point(i,:,:)=Ptos(:,:);
    Ptos_Original(i,:,:)=Ptos_Orig(:,:);
    Ptos=zeros(max_nptos,9);
    Posact=Angi;

    %% Guardar la matriz de error de la articulación i en la última posición
    %de giro de esta articulación ya que es necesaria para generar puntos
```

```
    %en el resto de articulaciones. todas las que vengan detrás estarán
    %afectadas por este error.
    eval(['GZ' num2str(i) '=GZ;']);

end

%% Muestra los puntos generados
if ShowFigure
    figure;
    col=['r' 'g' 'b' 'c' 'm' 'k'];
    for i=1:6
      Ptos(:,:)=Ptos_Circle_Point(i,:,:);
      plot3(Ptos(1:Nptos(i),1),Ptos(1:Nptos(i),2),Ptos(1:Nptos(i),3),['.'
col(i)]);
      axis equal;
      hold on;
    end
    grid on;
    xlabel('X'), ylabel('Y'), zlabel('Z')
end
```

17.2 Coordenadas Plücker Ejes

```
function [Plucker, P_Circulos, SR0, Distcirculo, Distplano, Distcilindro,
sigmaPlano, sigmaCirculo, sigmaCilindro, Angfin]=...
    Coordenadas_Plucker_Ejes(Ptos_Circle_Point, Nptos, r0, Param5, ShPoints,
ShBestfitCircles, ShNormals, ShrefS, ShEx, SRLT)
%Calcula las coordenadas Plucker correspondientes a los 6 ejes del robot
%con los datos de entrada obtenidos, bien sea de una GENERACION DE PUNTOS
%completa del robot según el método del Circle-Point, o de las lecturas
%tomadas con un instrumento de medición (p.ej., Láser Tracker) de la posición
%de cada articulación del robot. En la simulación se utilizará esta
%función, que permite detectar automáticamente si las normales obtenidas
%están o no en la misma dirección que los ejes Z del robot. Si no lo están
%se cambia su sentido automáticamente. Con datos reales, se necesitará
%disponer y suministrar a esta función con la matriz que pasa de SRLT al
%SR robot.

%PARAMETROS:
%- Ptos_Circle_Point. Matriz 6x1000x9 con los puntos de los círculos
%medidos en SRLT.

%-Nptos: Vector 1x6 con el número de puntos de cada
%círculo.

%-r0: Vector 1x6 con los estimadores de los radios de los círculos
%descritos por el reflector en cada articulación

%-Param5: Parámetros del robot

%-ShPoints, ShBestfitCircles, ShNormals, ShrefS, ShEx: Indican si se
%representarán en el resultado gráfico los puntos, círculos aproximados,
%normales a los planos, sistemas de referencia del robot y gráficas de
%excentricidad. Variables booleanas: Pueden ser TRUE o FALSE.

%- SRLT: Matriz de transformación que pasa puntos del SR global del robot
%al SR de medición del LT utilizada en la generación de puntos. Los puntos
generados estarán expresados en el SR
%del LT.

%SALIDA
%-Plucker: Matriz 6x6 que contiene por filas las coordenadas Pücler de los
%ejes del robot obtenidas por circle-point

%-P_Circulos: Matrix 6x7 que contiene por filas Centro, radio y normal del
%círculo ajustado en cada eje.

%-SR0: Devuelve la matriz de cambio del Sistema de Referencia 0, es decir,
%del origen de coordenadas del robot. Contiene los vectores de cambio del
%eje X,Y,Z, por columnas. (eje X=columna 1, etc). Matriz 4x4.

%-Distcirculo: Vector 6x1000 que contiene por filas las distancias de los
puntos al
%círculo de ajuste en cada eje

%-Distplano: Vector 6x1000 que contiene por filas las distancias de los
puntos al
%plano de ajuste en cada eje

%-Distcilindro: Vector 6x1000 que contiene por filas las distancias de los
puntos al
```

```
%cilindro que contiene al círculo de ajuste en cada eje

%-sigmaPlano: Vector 1x6 con las desviaciones estándar de la distribucion de
distancias al
%plano para los 6 ejes.

%-sigmaCirculo: Vector 1x6 con las desviaciones estándar de la distribucion
de distancias al
%círculo para los 6 ejes.

%-sigmaCilindro: Vector 1x6 con las desviaciones estándar de la distribucion
de distancias al
%cilindro para los 6 ejes.

%% ****** INICIALIZA VARIABLES ********
%Se alojan las variables para mayor velocidad de la función
max_nptos=max(Nptos);
Distplano=zeros(6,max_nptos);
Distcirculo=zeros(6,max_nptos);
Distcilindro=zeros(6,max_nptos);
dplane=zeros(6,max_nptos);
dcircle=zeros(6,max_nptos);
Si=zeros(3,6);
S0i=zeros(3,6);
x0i=zeros(3,6);
ri=zeros(3,6);
rni=[0 0 0 0 0 0];
Plucker=zeros(6,6);
P_Circulos=zeros(6,7);
sigmaPlano=rni;%Vector 1x6 con todo ceros
sigmaCirculo=rni;
sigmaCilindro=rni;
Proy=zeros(6,3);%Guarda un valor por articulacion para colocar el n°
Pproy=zeros(max_nptos,3);
cp=zeros(max_nptos,3);

%Representar círculos de puntos, círculos ajustados y normales con
%dirección
col=['r' 'g' 'b' 'c' 'm' 'k'];
longflecha=[150 150 150 150 150 150];
longSR=[50 50 50 50 50 50];
incnum=10;%Separación del texto N° articulación al primer punto

%Asigna valores de PuntoInicial y AnguloFinal
PtoIni(:,:)=Ptos_Circle_Point(1,1,:);
PtoIni=PtoIni';
Angfin(:,:)=Ptos_Circle_Point(6,Nptos(6),4:9);
Angfin=Angfin';%Se transpone para que de una matriz de 1 fila

%% ***** PARA CADA ARTICULACION ******
for i=1:6
  Ptos(:,:)=Ptos_Circle_Point(i,:,:);
  PtosEje(:,:)=Ptos(1:Nptos(i),:);

  %% Obtiene el plano y circulo que mejor se ajusta a los puntos
  %Obtener el plano que mejor ajusta a los puntos del círculo
  [x0, Eje, dist, normd] = lsplane(PtosEje(:,1:3));
  Si(:,i)=Eje;%Dirección normal - cosenos directores
  x0i(:,i)=x0;%Centroide de los datos
  %S cosenos directores de la normal al plano
  %d distancias de los puntos al plano ajustado
```

```
  %Debe asegurarse que el sentido de la normal coincida con el eje Z de la
  %articulación segun el modelo. Esto se hará visualmente dado que los
  %puntos están expresados en SRLT y a priori no es posible asegurar su
  %orientación. Función Cambia_sentido_ejes

  %Proyección ortogonal de los puntos del círculo en el plano
  %1. Pareja de planos que contienen a la normal al plano
  %calcular la pareja de planos en función de los valores de los cosenos
  %directores de la normal

  %Ajustar círculo con los puntos. Esta función obtiene el plano de ajuste,
  %el círculo y los vectores de distancias al plano, al círculo y al
  %cilíndro
  %El centro círculo estimado ya está calculado por lsplane en x0
  % y la normal al plano es la variable Eje obtenida antes por lsplane
  %[x0n1, an1, rn1, dc1, e1, f1, sigmah, conv, Vx0n, Van, urn, GNlog, a, R0,
R] = ls3dcircle(PtosEje(:,1:3), mean(PtosEje(:,1:3))', Si(:,i), r0(i), 1e-6,
1e-6);
  [x0n, an, rn, dc, e, f, sigmah, conv, Vx0n, Van, urn, GNlog, a, R0, R] =
ls3dcircle(PtosEje(:,1:3), x0, Eje, r0(i), 1e-6, 1e-6);

  %% Guarda los valores de Si, ri, P_Circulos y otros
  Si(:,i)=an; %normal
  ri(:,i)=x0n; %punto del eje (centro del círculo)
  rni(i)=rn; %radio del círculo.

  P_Circulos(i,:)=[x0n' rn an']; %Centro, radio y normal
  Distcirculo(i,1:Nptos(i))=dc';
  Distplano(i,1:Nptos(i))=e';
  Distcilindro(i,1:Nptos(i))=f';
  %Desviaciones estándar de la distribucion de distancias al plano
  sigmaPlano(i)=std(dc);
  sigmaCirculo(i)=std(e);
  sigmaCilindro(i)=std(f);

  PtosEje=[];
end

%% Comprueba si la normal eje Z calculado del robot coincide con normal SRLT
% y si es diferente, cambia la dirección de Si(:,i)
%Reconocer si las normales obtenidas tienen la misma dirección o dirección
%contraria a los ejes Z del modelo del robot.
[M0A6 M0A1 M1A2 M2A3 M3A4 M4A5 M5A6] = PBETAfRobotMKR5sixxDHHpar(Angfin,
Param5);

%Para cada articulación se realiza el producto escalar (cos fi) entre el
%eje Z del SRLT y el eje Z de la articulación, definido por Si(:,i)
%Si es < 0 ==> Si(:,i) tiene que cambiarse de signo para que vaya en la
%misma dirección que el SRLT
%cos_ang no es realmente el cos_ang. Sería cos_ang si:
%cos_ang=dot(norm(x1),norm(x2));

%SR0_1
cos_ang=dot(SRLT(1:3,3),Si(:,1));
if (cos_ang<0)
Si(:,1)=-Si(:,1);
end
%Eje Z nominal articulación 1 expresado en SRLT
EZn(1:3,1)=SRLT(1:3,3);

%SR2
M=SRLT*M0A1;
```

```
cos_ang=dot(M(1:3,3),Si(:,2));
if (cos_ang<0)
Si(:,2)=-Si(:,2);
end
%Eje Z nominal articulación 2 expresado en SRLT
EZn(1:3,2)=M(1:3,3);

%SR3
M=M*M1A2;
cos_ang=dot(M(1:3,3),Si(:,3));
if (cos_ang<0)
Si(:,3)=-Si(:,3);
end
%Eje Z nominal articulación 3 expresado en SRLT
EZn(1:3,3)=M(1:3,3);

%SR4
M=M*M2A3;
cos_ang=dot(M(1:3,3),Si(:,4));
if (cos_ang<0)
Si(:,4)=-Si(:,4);
end
%Eje Z nominal articulación 4 expresado en SRLT
EZn(1:3,4)=M(1:3,3);

%SR5
M=M*M3A4;
cos_ang=dot(M(1:3,3),Si(:,5));
if (cos_ang<0)
Si(:,5)=-Si(:,5);
end
%Eje Z nominal articulación 5 expresado en SRLT
EZn(1:3,5)=M(1:3,3);

%SR5
M=M*M4A5;
cos_ang=dot(M(1:3,3),Si(:,6));
if (cos_ang<0)
Si(:,6)=-Si(:,6);
end
%Eje Z nominal articulación 6 expresado en SRLT
EZn(1:3,6)=M(1:3,3);

%% ****** Guarda coordenadas Plücker ********
for i=1:6
 S0i(:,i)=cross(ri(:,i), Si(:,i));
 Plucker(i,:)=[Si(:,i)' S0i(:,i)'];
end

%% Si tiene que mostrar alguna gráfica ==> PARA CADA ARTICULACION i=1..6
%ShowSth=show something = tiene que mostrar algo
ShowSth=(ShPoints || ShEx || (ShBestfitCircles) || (ShrefS) || ShNormals);

if ShowSth
 for i=1:6
  %Puntos
  Ptos(:,:)=Ptos_Circle_Point(i,:,:);
  %% Si se permite, muestra los puntos del circulo, y el n°articulación
  if (ShPoints)
    if (i==1)
      h3=figure('Color','w','Name','Circle Point Method: Joints circles and
normal direction','NumberTitle','off');
```

```
    end
    figure(h3);
    plot3(Ptos(1:Nptos(i),1),Ptos(1:Nptos(i),2),Ptos(1:Nptos(i),3),['.'
col(i)]);
    axis equal;
    hold on;
    %Representar el número de articulación junto al primer punto capturado en
    %cada una

text(Ptos(1,1)+incnum,Ptos(1,2)+incnum,Ptos(1,3)+incnum,num2str(i),'Color',co
l(i),'FontSize',16,'FontWeight','bold');
  end

  %% Proyectar un punto del círculo en el plano XY de la articulación
  %Punto p=Ptos(i,1:3) ---- Plano XY = [0 0 1]
  %Se define la recta R que pasa por el punto P y es ortogonal al plano XY
  % es decir, que pasa por Si(1:3,i)
  %Se calculan los planos P1 y P2 que contienen a R
  %Se calcula el punto intersección de los 3 planos, P1, P2 y XY
  p=Ptos(1,1:3);
  if (abs(Si(3,i))> 1e-6)
    P1=[1 1 -(Si(1,i)+Si(2,i))/Si(3,i) ((p(3)*(Si(1,i)+Si(2,i)))/Si(3,i))-
p(1)-p(2)];
    P2=[1 2 -(Si(1,i)+2*Si(2,i))/Si(3,i)
((p(3)*(Si(1,i)+2*Si(2,i)))/Si(3,i))-p(1)-2*p(2)];
  elseif (abs(Si(2,i))> 1e-6)
    P1=[1 -(Si(1,i)+Si(3,i))/Si(2,i) 1 ((p(2)*(Si(1,i)+Si(3,i)))/Si(2,i))-
p(3)-p(1)];
    P2=[1 -(Si(1,i)+2*Si(3,i))/Si(2,i) 2
((p(2)*(Si(1,i)+2*Si(3,i)))/Si(2,i))-2*p(3)-p(1)];
  elseif (abs(Si(1,i))> 1e-6)
    P1=[-(Si(2,i)+Si(3,i))/Si(1,i) 1 1 ((p(1)*(Si(2,i)+Si(3,i)))/Si(1,i))-
p(2)-p(3)];
    P2=[-(Si(2,i)+2*Si(3,i))/Si(1,i) 1 2
((p(1)*(Si(2,i)+2*Si(3,i)))/Si(1,i))-p(2)-2*p(3)];
  else
      %ERROR - No puede ser: sólo llega aquí si Si=[0 0 0]
  end
  U=[Si(:,i)'; P1(1:3);P2(1:3)];
  T=[-(-(Si(1,i)*x0i(1,i)+Si(2,i)*x0i(2,i)+Si(3,i)*x0i(3,i))) -(P1(4)) -
(P2(4))]';
  Proy(i,:)=(U\T)';%Punto referenciado al centroide del círculo (usa x0i)
  %Se usa para calcular después la matriz de giro MG del SRplano al SRLT

  %Proyectar todos los puntos en el plano y obtener las distancias con
  %signo. Para ver la excentricidad que tiene la articulación hay que
  %hacerlo respecto al plano nominal. si se hace respecto a los datos con
  %error, no se verá excentricidad. Los datos con error, por ejemplo,
  %pueden estar en un plano perfecto girado respecto al nominal (con una
  %excentricidad senoidal de periodo 2pi en z). En ese caso las distancias
  %al plano calculado serían cero y sin embargo habría excentricidad
  %respecto al nominal.
  %Realmente deberán obtenerse las distancias respecto a un plano paralelo
  %al nominal por el centro del círculo identificado en el circle point.

  %% Dentro de la articulación - Para cada punto del círculo
  for m=1:Nptos(i)

      p=Ptos(m,1:3);
      %% Obtiene PProy=proyección del punto sobre el plano normal
      % al vector Si(,i)
      if (abs(Si(3,i))> 1e-6)
```

```
        P1=[1 1 -(Si(1,i)+Si(2,i))/Si(3,i)
((p(3)*(Si(1,i)+Si(2,i)))/Si(3,i))-p(1)-p(2)];
        P2=[1 2 -(Si(1,i)+2*Si(2,i))/Si(3,i)
((p(3)*(Si(1,i)+2*Si(2,i)))/Si(3,i))-p(1)-2*p(2)];
      elseif (abs(Si(2,i))> 1e-6)
        P1=[1 -(Si(1,i)+Si(3,i))/Si(2,i) 1
((p(2)*(Si(1,i)+Si(3,i)))/Si(2,i))-p(3)-p(1)];
        P2=[1 -(Si(1,i)+2*Si(3,i))/Si(2,i) 2
((p(2)*(Si(1,i)+2*Si(3,i)))/Si(2,i))-2*p(3)-p(1)];
      elseif (abs(Si(1,i))> 1e-6)
        P1=[-(Si(2,i)+Si(3,i))/Si(1,i) 1 1
((p(1)*(Si(2,i)+Si(3,i)))/Si(1,i))-p(2)-p(3)];
        P2=[-(Si(2,i)+2*Si(3,i))/Si(1,i) 1 2
((p(1)*(Si(2,i)+2*Si(3,i)))/Si(1,i))-p(2)-2*p(3)];
      else
          %ERROR - No puede ser: sólo llega aquí si Si=[0 0 0]
      end
      U=[Si(:,i)'; P1(1:3);P2(1:3)];
  %Proyectar los puntos en un plano paralelo al XY del SR nominal que pasa
  %por el centro del círculo. Esta es la forma de obtener la excentricidad
  %respecto al SRnominal sin desfase de excentricidad, dado que se obtendrá
  %la misma situación que produce el movimiento que hace el SR con
excentricidad alrededor del fijo de
  %cada articulación. Dado que este es el generador, se debe tener en
  %cuenta que todas las articulaciones que van detrás de la actual sufrirán
  %esta excentricidad, de modo que, excepto en la primera, la excentricidad
  %que se obtiene es acumulada. Para obtener la excentricidad real de cada
  %articulación a partir de medidas del circle-point habrá que hacer un
  %offset al plano nominal de acuerdo a la posición que ocupe el SR con
  %excentricidad anterior(No creo que haga falta porque no hay giro del
anterior y el centro
  %del círculo por el que pasará el plano ya estará en la pos
correspondoiente a la excentricidad). En principio, los datos que se van a
obtener de
  %la excentricidad son puramente axiales o radiales, de modo que en
  %identificación de los términos de la matriz de excentricidad no será
  %posible a partir de estos datos determianr giros. Es como si únicamente
  %el SR se trasladase,porque solo hay información de la trayectoria que
  %sigue un punto (CON 2 MEDIDAS A DIFERENTES DISTANCIAS PODEMOS SABER ALGO
DEL GIRO!!!)
  %De ese modo, en la posición de la articuaión anterior, el SR con
  %excentricidad tendrá la misma orientación que el nominal pero con el
  %offset que le corresponda. OJO, porque en caso de que existan
  %errores de giro, el efecto de un giro en las excentricidades dependerá
  %de a qué diámetro esté colocado el reflector, cuanto más lejos mayores
  %distancias al plano o círculo tendremos dado un giro. Si esto lo
  %traducimos directamente solo en desplazamientos a partir de datos de
  %excentricidad no iría bien. Por el momento solo representar las
  %excentricidades, después ya veremos las correcciones. OJO también con
  %las correcciones. Una cosa es intentar conocer la excentricidad real (uqe
mediríamos con un comparador)
  %En ese caso efectivamente hay que sacarlo respecto al plano teórico y no
  %al de mínimos cuadrados. Otra cosa es que mteméticamente el modelo va a
  %ir perfectamente si en lugar de los parámetros nominales se consideran
  %los del circle point, que no verían excentricidad por ejemplo axial en
  %el caso de que fuera, por ejemplo una con un periodo de 2pi en dirección
  %axial. El modelo iría bien, aunque no sería la situación real de
  %excentricidad.

  %Habría que proyectar en el plano de circle-point en ese caso, de modo
  %que, aunque realmente tenemos una excentricidad el modelo de
  %circle-point únicamente debe considerar la excentricidad "ficticia" que
  %ve desde su propio modelo que en el ejemplo del plano inclinado sería
  %nula. Así que todo esto depende de cómo se quiera afrontar una posterior
```

```
  %corrección de la excentricidad. Parece que tiene sentido llevarse los
  %sistemas de referencia nominales a los orígenes del circle point y ahí
  %caracterizar la excentricidad REAL. Esto afectaría a los parámetros
  %nominales y daría otro conjunto de parámetros "nominales" con cambios en a
y d, pero
  %no en los ángulos. En ese caso, parece lógico si solo se consideran
  %desplazamientos en la excentricidad proyectar los puntos en el plano
  %nominal (eliminar la excentricidad axial) y ahí caracterizar la
  %excentricidad radial respecto al círculo teórico, no respecto al de
  %mínimos cuadrados. El problema es que el círculo teórico depende de las
  %coordenadas del reflector (p. ej. en la primera posición) respecto a la
  %posición del modelo, dato que no tenemos a partir de las medidas del
  %circle-point. CONCLUSIÓN: Hay que caracterizar la excentricidad en el
  %plano de circle point y respecto al círculo de mínimos cuadrados. Con
  %respecto a la excentricidad radial, de momento se va a representar el
  %valor de excentricidad radial respecto al círculo de mínimos cuadrados en
el plano de mínimos cuadrados.
  %Para una posterior corrección de esta excentricidad habrá que ver la
  %relación de cada lectura con el SR final, de modo que se pueda
  %distribuir adecuadamente la excentricidad entre x e y. La distancia de
  %el centro obtenido al primer punto del reflector debería ser constante.
  %Las variaciones de esa distancia se deben al movimiento del centro, y pr
  %tanto a excentricidad. Además, esto hy ue relacionarlo con el giro
  %medido. aplicando el giro al vector que une el centro y la primera
  %posición del reflector tendríamos dónde debería estar en la segunda y
  %por la medida dónde está realmente. ese incremento en X e Y es el que se
  %debe corregir en cada posición de giro.
      T=[-(-(Si(1,i)*ri(1,i)+Si(2,i)*ri(2,i)+Si(3,i)*ri(3,i))) -(P1(4)) -
(P2(4))]';
      %Punto junto a la circunferencia del círculo (usa ri(1:3,i) )
      Pproy(m,1:3)=(U\T)';%Pproy=inv(U)*T;

      %% Calcula la distancia AXIAL del punto al punto proyectado
      cp(m,:)=p-Pproy(m,:);%Diferencia con 1° punto p, en valor XYZ
      dplane(i,m)=norm(cp(m,:));%Distancia (valor escalar) al plano
      if(abs(dplane(i,m))<1e-6)
          dplane(i,m)=0;%Si se hace 0, cp debe hacerse 0.Si no,se hace NaN o
Inf
          %cp(m,:)=0;
      else
          cp(m,:)=cp(m,:)/dplane(i,m);%cp queda como vector direccion de la
distancia
      end

      %signo de la distancia (positivo en dirección de la normal, negativo al
otro lado del plano)
      %cos_ang=dot(cp(m,:),Si(:,3));
      cos_ang=dot(cp(m,:),Si(:,i));
      if (cos_ang<0)
        dplane(i,m)=-dplane(i,m);
      end

      %% Calcula la distancia RADIAL del punto proyectado a la
      % circunferencia
      %Distancia de los puntos proyectados al círculo. Positiva punto por
      %fuera del círculo, negativa punto por dentro.
      Rm=sqrt((Pproy(m,1)-ri(1,i))^2+(Pproy(m,2)-ri(2,i))^2+(Pproy(m,3)-
ri(3,i))^2);
      dcircle(i,m)=Rm-rni(i);
      if(abs(dcircle(i,m))<1e-6)
          dcircle(i,m)=0;
      end
  end
```

```
  %% Si ShEx-> Muestra el gráfico y las distancias de excentricidad axial y
  % radial calculadas para la articulación
  if (ShEx)
      %plano
      if (i==1)
          h1=figure('Color','w','Name','AXIAL Circle-Point
Eccentricity','NumberTitle','off');
          h2=figure('Color','w','Name','RADIAL Circle-Point
Eccentricity','NumberTitle','off');
      end
      %Ángulos cambiados de signo!!
      hs1=subplot(2,3,i,'Parent',h1);plot(hs1,-
Ptos(1:Nptos(i),i+3),dplane(i,1:Nptos(i)),'-r','LineWidth',0.7);
      title(hs1,['Joint ' num2str(i)],'fontSize',16,'fontweight','b');
      xlabel(hs1,['\it\theta' num2str(i) '
(\circ)'],'fontSize',12,'fontweight','b');
      ylabel(hs1,'Eccentricity d (mm)','fontSize',12,'fontweight','b');
      set(hs1,'YGrid','on');
      if (i==6)
          figure(h1);
          th = subplot_title('AXIAL CIRCLE-POINT ECCENTRICITY');
      end
      %círculo
      hs2=subplot(2,3,i,'Parent',h2);plot(hs2,-
Ptos(1:Nptos(i),i+3),dcircle(i,1:Nptos(i)),'-b','LineWidth',0.7);
      title(hs2,['Joint ' num2str(i)],'fontSize',16,'fontweight','b');
      xlabel(hs2,['\it\theta' num2str(i) '
(\circ)'],'fontSize',12,'fontweight','b');
      ylabel(hs2,'Eccentricity d (mm)','fontSize',12,'fontweight','b');
      set(hs2,'YGrid','on');
      if (i==6)
          figure(h2);
          th = subplot_title('RADIAL CIRCLE-POINT ECCENTRICITY');
      end
  end

  %% Si ShBestfitCircles OR ShrefS -> calcula los puntos del circulo
  % ajustado XYZ con el radio obtenido, y calcula la matriz de transformación
  % del SR del plano al SRLT.
  % Gira los puntos XYZ que forman el circulo con esta matriz
  if ((ShBestfitCircles) || (ShrefS))
      % Cálculo del círculo ajustado XYZ con el radio obtenido
      NOP=100;
      %Genera un vector de NOP puntos, entre 0 y 2*pi, linealm. espaciados
      THETA=linspace(0,2*pi,NOP);%Son los ángulos
      RHO=ones(1,NOP)*P_Circulos(i,4);%P_Circulos(i,4) es el radio
      %Ahora convierte a coordenadas cartesianas, las coor.polares de
      %valor longitud RHO y ángulo THETA
      [X,Y] = pol2cart(THETA,RHO);
      Z=0*ones(1,length(X));

      %V2 es el Vector del centro estimado al punto proyectado.
      % Será perpendicular al vector normal del plano
      V2=[Proy(i,1)-ri(1,i) Proy(i,2)-ri(2,i) Proy(i,3)-ri(3,i)];
      norma= sqrt(V2(1)^2+V2(2)^2+V2(3)^2);
      V2=V2/norma;
      %tercer eje del SR del plano, que debe estar a 90° del eje V2
      V3=cross(Si(:,i)',V2);
      %MG=Matriz que pasa del SR plano al SRLT, en el que se ha generado el
      %círculo ajustado para representación
      MG=[V2; V3; Si(:,i)']';
      MG=cat(2,MG, [ P_Circulos(i,1) P_Circulos(i,2) P_Circulos(i,3)]');
```

```
      MG=cat(1,MG, [0 0 0 1]);
      %Girar/trasladar los puntos con la matriz obtenida para representación
      for k=1:NOP
          pcir=[X(k) Y(k) Z(k) 1]';
          res=MG*pcir;
          X(k)=res(1);
          Y(k)=res(2);
          Z(k)=res(3);
      end
  end

  %% Si ShBestfitCircles -> representa el círculo
  if (ShBestfitCircles)
    figure(h3);
    hold on;
    plot3(X,Y,Z,[col(i) '-']);
    hold on;
  end
  %% Si ShNormals -> Representa las normales con dirección
  %Representar las normales con dirección
  if(ShNormals)
      arrow3D(ri(:,i),longflecha(i)*Plucker(i,1:3)',col(i));
      hold on;
  end

  %% Si ShrefS y es la articulacion 1 --> Pinta los sistema de referencia
  % y los números de cada articulación
  %Representar SR's del robot. Para esta representación se asume que la
normal
  %al plano de giro de la primera articulación es la correcta. Esto servirá
  %para determinar si el sentido de las normales que se han obtenido es el
  %correcto o es necesario cambiar alguna de sentido para hacerla coincidir
  %con el sentido de los ejes de giro en el modelo del robot
  if(ShrefS && i==1)
     %La matriz MG se ha generado con el eje Z del SR de cada plano siendo
     %su normal y el eje X apuntando del centro aproximado al primer punto.
     %Localizar el SR global del robot en el plano en la posición
     %correspondiente a un giro de 0 grados, es decir, en posición del
     %SRGlobal. Hay que tener en cuenta que los puntos corresponden a la
     %posición del reflector, por lo que el centro del círculo no
     %coincidirá con la posición del SR global y por tanto el resto de
sistemas tampoco coincidirán en posición con los círculos.
     %Lo que se pretende con esta representación es solo comprobar el
     %sentido de las normales obtenidas, cuya dirección sí coincidirá con
     %los ejes Z de las articulaciones.

     %Esto pinta los SRs NOMINALES. Como el SR base no está localizado, el
     %SR0 representado se pone de modo que el eje Z0 coincide con la normal
     %al plano y con origen en el centro del primer círculo.
     %PtoIni(1,4) es el ángulo inicial de la articulación 1
     angcero=-PtoIni(1,4);%El ángulo viene cambiado de signo por el sentido
de las lecturas del robot
     GZ=[cos(deg2rad(angcero)) -sin(deg2rad(angcero)) 0 0;
sin(deg2rad(angcero)) cos(deg2rad(angcero)) 0 0; 0 0 1 0; 0 0 0 1];
     SR0=MG/GZ;
     %Pintar SR0
     M=SR0;
     Pinta_SR (1,ri,longSR,incnum,M,col)
     % Obtener las matrices en la última posición del robot
     %Para ello el ángulo del eje 5 debe ir en sentido contrario
     Angfin(5)=-Angfin(5);
     [M0A6 M0A1 M1A2 M2A3 M3A4 M4A5 M5A6] = PBETAfRobotMKR5sixxDHHpar(Angfin,
Param5);
```

```
     Angfin(5)=-Angfin(5);

     M=M*M0A1;
     Pinta_SR (2,ri,longSR,incnum,M,col);
     M=M*M1A2;
     Pinta_SR (3,ri,longSR,incnum,M,col);
     M=M*M2A3;
     Pinta_SR (4,ri,longSR,incnum,M,col);
     M=M*M3A4;
     Pinta_SR (5,ri,longSR,incnum,M,col);
     M=M*M4A5;
     Pinta_SR (6,ri,longSR,incnum,M,col);
  end
 end

else
    %% Si no muestra nada => Calcula SR0
    %El cálculo de SR0 es para la articulación 1
    i=1;

    % Proyectar el primer punto del círculo en el plano XY de la articulación
    %Punto p=Ptos(i,1:3) ---- Plano XY = [0 0 1]
    %Se define la recta R que pasa por el punto P y es ortogonal al plano XY
    % es decir, que pasa por Si(1:3,i)
    %Se calculan los planos P1 y P2 que contienen a R
    %Se calcula el punto intersección de los 3 planos, P1, P2 y XY
    Ptos(1,:)=Ptos_Circle_Point(i,1,:);
    p=Ptos(1,1:3);
    if (abs(Si(3,i))> 1e-6)
      P1=[1 1 -(Si(1,i)+Si(2,i))/Si(3,i) ((p(3)*(Si(1,i)+Si(2,i)))/Si(3,i))-
p(1)-p(2)];
      P2=[1 2 -(Si(1,i)+2*Si(2,i))/Si(3,i)
((p(3)*(Si(1,i)+2*Si(2,i)))/Si(3,i))-p(1)-2*p(2)];
    elseif (abs(Si(2,i))> 1e-6)
      P1=[1 -(Si(1,i)+Si(3,i))/Si(2,i) 1 ((p(2)*(Si(1,i)+Si(3,i)))/Si(2,i))-
p(3)-p(1)];
      P2=[1 -(Si(1,i)+2*Si(3,i))/Si(2,i) 2
((p(2)*(Si(1,i)+2*Si(3,i)))/Si(2,i))-2*p(3)-p(1)];
    elseif (abs(Si(1,i))> 1e-6)
      P1=[-(Si(2,i)+Si(3,i))/Si(1,i) 1 1 ((p(1)*(Si(2,i)+Si(3,i)))/Si(1,i))-
p(2)-p(3)];
      P2=[-(Si(2,i)+2*Si(3,i))/Si(1,i) 1 2
((p(1)*(Si(2,i)+2*Si(3,i)))/Si(1,i))-p(2)-2*p(3)];
    else
        %ERROR - No puede ser: sólo llega aquí si Si=[0 0 0]
    end
    U=[Si(:,i)'; P1(1:3);P2(1:3)];
    T=[-(-(Si(1,i)*x0i(1,i)+Si(2,i)*x0i(2,i)+Si(3,i)*x0i(3,i))) -(P1(4)) -
(P2(4))]';
    Proy(i,:)=(U\T)';%Punto referenciado al centroide del círculo (usa x0i)
    %Se usa para calcular después la matriz de giro MG del SRplano al SRLT

    %Calcula MG, que es la matriz de cambio
    %V2 es el Vector del centro estimado al punto proyectado.
    % Será perpendicular al vector normal del plano
    V2=[Proy(i,1)-ri(1,i) Proy(i,2)-ri(2,i) Proy(i,3)-ri(3,i)];
    norma= sqrt(V2(1)^2+V2(2)^2+V2(3)^2);
    V2=V2/norma;
    %tercer eje del SR del plano, que debe estar a 90° del eje V2
    V3=cross(Si(:,i)',V2);
    %MG=Matriz que pasa del SR plano al SRLT, en el que se ha generado el
    %círculo ajustado para representación
    MG=[V2; V3; Si(:,i)']';
```

```
    MG=cat(2,MG, [ P_Circulos(i,1) P_Circulos(i,2) P_Circulos(i,3)]');
    MG=cat(1,MG, [0 0 0 1]);

    angcero=-PtoIni(1,4);%El ángulo viene cambiado de signo por el sentido de
las lecturas del robot
    GZ=[cos(deg2rad(angcero)) -sin(deg2rad(angcero)) 0 0;
sin(deg2rad(angcero)) cos(deg2rad(angcero)) 0 0; 0 0 1 0; 0 0 0 1];
    SR0=MG/GZ;

end

%% Si ShRefs --> Muestra la imagen de los ejes del robot
if (ShrefS)
    grid on;
    xlabel('X'), ylabel('Y'), zlabel('Z')
    figure;
    I=imread('EjesRobot.bmp');
    image(I);
    axis off;
    axis image;
end
```

17.3 Calcula Parámetros KM

```
function P=Calcula_Parametros_KM(Plucker, P_Circulos, Param5, Angfin, CheckX,
SRLT, SR0, ShowFigure)
%Obtiene los parámetros cinemáticos del modelo del robot a partir de las
%coordenadas Plücker de los ejes obtenidas de los ensayos del método Circle
%Point

%PARÁMETROS
%-Plucker: Matriz 6x6 que contiene por filas las coordenadas Pücler de los
%ejes del robot obtenidas por circle-point

%-P_Circulos: Resultado del ajuste de los círculos tal y como lo da la
%función Coordenadas_Plucker_Ejes.

%- Params: Parámetros nominales del modelo del robot

%-Angfin: Vector 1x6 con el ángulo de giro que indica el robot en la
%última posición de captura

%- CheckX: Comprueba que la dirección de los vectores ajk coincide con el
%eje X nominal de la siguiente articulación. Variable booleana (true o
%false)

%- SRLT: Matriz de transformación que pasa puntos del SR global del robot
%al SR de medición del LT utilizada en la generación de puntos. Los puntos
%generados estarán expresados en el SR del LT.

%-SR0: Matriz de cambio del Sistema de Referencia 0, es decir,
%del origen de coordenadas del robot. Contiene los vectores de cambio del
%eje X,Y,Z, por columnas. (eje X=columna 1, etc). Matriz 4x4.

%- ShowFigure: Variable booleana - indica si se deben mostrar los gráficos.

%Parámetros iniciales
%d = [ 335 0 0 295 0 80];
%tita0 = [ 0 0 90 0 180 0 ];CON EL CAMBIO EN EL SR4 y SR5
%a = [ 75 270 90 0 0 0 ];
%alfa = [ 90 0 90 90 90 0 ];CON EL CAMBIO EN EL SR4 y SR5
%beta=[0 0 0 0 0 0];

%% ****** INICIALIZA VARIABLES ********
%Se preubican las variables para darle mayor velocidad
beta_jk=zeros(1,6);
ajk_v=zeros(6,3);
alfajk=zeros(1,6);
Pn=zeros(6,3);
Pp=zeros(6,3);
ajk=zeros(1,6);
dj=zeros(1,6);
titaj=zeros(1,6);
M=SRLT;
%Tolerancia para considerar que dos ejes son paralelos
TOLPAR_EXACTO=0;
TOLPAR_APROX=0.02;
TOL_CERO=1e-10; %Tolerancia para hacer cero un valor. Valores negativos muy
pequeños cambian el sentido de los ángulos en los cálculos

%% **** PARA CADA ARTICULACION j = 1 a 5
for j=1:5
```

```
    %% Toma valores Sj, S0j, Sk, S0k, cos (alfajk), calcula matriz M
    Sj=Plucker(j,1:3)';
    S0j=Plucker(j,4:6)';
    Sk=Plucker(j+1,1:3)';
    S0k=Plucker(j+1,4:6)';
    %Calcular cos(alfajk)
    cos_alfa_jk=dot(Sj,Sk);
    beta_jk(j)=0;
    %Calcula la matriz de cambio a la articulación actual j
    [M0A6 M0A1 M1A2 M2A3 M3A4 M4A5 M5A6] = PBETAfRobotMKR5sixxDHHpar(Angfin,
Param5);
    Mant=M;%Matriz de cambio de la articulación anterior j-1
    switch j
        case 1
            M=SRLT*M0A1;
        case 2
            M=M*M1A2;
        case 3
            M=M*M2A3;
        case 4
            M=M*M3A4;
        case 5
            M=M*M4A5;
    end

    %por el momento por sensibilidad numérica
    if ((cos_alfa_jk<(0+TOL_CERO)) && (cos_alfa_jk>(0-TOL_CERO)))
       cos_alfa_jk=0;
    end
    %% Si cos(alfajk)=-1 ó +1 --> EJES PARALELOS
    if ((cos_alfa_jk<(-1+TOLPAR_EXACTO)) || (cos_alfa_jk>(1-TOLPAR_EXACTO)))
        %**************DEBERÍA SER <= Y EN EL OTRO LADO >=
        %Los ejes son exáctamente paralelos. Determinar ajk y alfajk
        ajk_v(j,:)=cross(Sj,(cos_alfa_jk*S0k-S0j));
        alfajk(j)=rad2deg(acos(dot(Sj,Sk)));
        %En este caso, dj y dk no pueden calcularse porque los puntos de
        %intersección no son únicos
    end

    %% Si cos(alfajk) es casi -1 ó +1 --> EJES CASI PARALELOS
    if ((cos_alfa_jk<(-1+TOLPAR_APROX)) || (cos_alfa_jk>(1-TOLPAR_APROX)))
        %Los ejes son casi paralelos. 3 posibles soluciones para determinar
        %los parámetros. Sección 5.4 Tesis Sklar. Por el momento se utiliza
        %la Solución 3 (5 parámetros).
        %Obtener los puntos de intersección. Se obtienen los nominales, al
        %igual que en la solución 2.
        Pn(j,:)=(Pp(j-1,:)'+Param5(j)*Sj)';
        %Esto es PARAM5(j)=dj
        Pp(j,:)=(cross(Sj,S0k)+(dot(Pn(j,:)',Sj)*Sk))/dot(Sj,Sk);
        %d será cero y a es d redifinir el modelo en ejes paralelos porque
        %no es realmente un modelo de 5 parámetros
        ajk_v(j,:)=Pp(j,:)-Pn(j,:);
        ajk(j)=norm(ajk_v(j,:));
        ajk_v(j,:)=ajk_v(j,:)/ajk(j);
        %Skp es la proyección de Sk_real sobre el plano normal a ajk
        %Sk_dif es el vector diferencia entre Sk-real y Skp(proyectada).
        %cos_skajk es el coseno del ángulo que forman Sk y ajk_v
        %se forma un triángulo, en el que hipotenusa es Sk_real, coseno es
        %Skproy y el seno sería el Sk_dif, de forma que Skr=Skp+Skdif
        cos_skajk=dot(Sk,ajk_v(j,:)');
        Sk_dif=(cos_skajk)*ajk_v(j,:)';
        Skp=Sk-(Sk_dif);
        %Forma 1 de calcular cos_beta->
```

```
        cos_beta_jk=dot((Skp/norm(Skp)),Sk);
        beta_jk(j)=rad2deg(atan2(cos_skajk,cos_beta_jk));%****
        %Forma 2 de calcularlo: Esta forma es sen=sqrt(h^2 - cos^2), con
        %h=1
        %cos_beta_jk=sqrt(1-((dot(Sk,ajk_v(j,:)))^2));
        %Se usa la forma 1 ya que con la raiz cuadrada siempre da positivo

        %ATENCIÓN: EL CÁLCULO DE ESTE SE HACE SOLO CON
        %EL ACOS EN VEZ DE CON ATAN2 PORQUE EL SENO DE BETA DEPENDE DE UN
        %VECTOR B CUYO CÁLCULO NO SE ESPECIFICA.
        %Metodo anterior
        %beta_jk(j)=rad2deg(acos(cos_beta_jk));

        cos_alfa_p_jk=dot(Skp/norm(Skp),Sj);
        sin_alfa_p_jk=dot(cross(Sj,Skp/norm(Skp)),ajk_v(j,:));
        %Para calcular alfajk, con atan2 -> da un valor entre -pi y +pi
        %si atan2>3pi/4 o atan2<-3pi/4 (cerca de +-180º)
        %  entonces se cambia atan2 entre 0 y 2pi
        alfajk(j)=atan2(sin_alfa_p_jk,cos_alfa_p_jk);%da valor entre -pi y
+pi
        if (alfajk(j)>(3*pi/4)) || (alfajk(j)<(-3*pi/4))
            alfajk(j)=rad2deg(mod(alfajk(j),2*pi));%Cambia entre 0 y 2pi
        else
            alfajk(j)=rad2deg(alfajk(j));
        end
    else
        %% No son ejes casi parelelos - Calcula MM
        %Calcular el momento mútuo de los dos ejes
        MM=dot(Sj,S0k)+dot(Sk,S0j);
        if ((MM<(0+TOL_CERO)) && (MM>(0-TOL_CERO)))
           MM=0;
        end

        if (MM==0)
          %% Si MM=0 --> las líneas se cortan
          %Las líneas se cortan
          ajk(j)=0;%Distancia ajk es 0, ya que se cortan, y Pn=Pp
          ajk_v(j,:)=(cross(Sj,Sk)/norm(cross(Sj,Sk)));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
          %Por el momento, por sensibilidad numérica del cero alfa4 es 90
          %y alfa5 -90. ajk_v identifica el eje x correspondiente en sentido
          %contrario. Hasta ver si con las pruebas reales o con valores que
          %no sean perfectos se obtiene correctamente.
          %Ocurre lo mismo con el MM de las dos últimas, por sensibilidad
          %numérica obtiene el vector al revés

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%
          Pn(j,:)=cross(S0j,S0k)/dot(Sk,S0j);
          Pp(j,:)=Pn(j,:);

          %alfajk por convenio es menor de 180º
          sin_alfa_jk=norm(cross(Sj,Sk));
          %Para calcular alfajk, con atan2 -> da un valor entre -pi y +pi
          %si atan2>3pi/4 o atan2<-3pi/4 (cerca de +-180º)
          %  entonces se cambia atan2 entre 0 y 2pi
          alfajk(j)=atan2(sin_alfa_jk,cos_alfa_jk);%da valor entre -pi y +pi
          if (alfajk(j)>(3*pi/4)) || (alfajk(j)<(-3*pi/4))
              alfajk(j)=rad2deg(mod(alfajk(j),2*pi));%Cambia entre 0 y 2pi
          else
              alfajk(j)=rad2deg(alfajk(j));
```

```
          end
          %alfajk(j)=rad2deg(mod(atan2(sin_alfa_jk,cos_alfa_jk),2*pi));

        else
          %% Si MM<>0 --> las líneas son oblicuas
          %implícito que cosalfajk sea distinto de 1 ó -1
          %ya que ya se ha comprobado antes
          ajk_v(j,:)=-sign(MM)*(cross(Sj,Sk)/norm(cross(Sj,Sk)));

          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
          %Comprobar que la dirección de ajk_v coincide con la dirección
          %del siguiente eje x nominal del modelo.
          ajk_sign=+1;
          if (CheckX)
              cos_ang=dot(M(1:3,1),ajk_v(j,:));
              if (cos_ang<0)
                  ajk_sign=-1;
                  ajk_v(j,:)=-ajk_v(j,:);
                  MM=-MM;
              end
          end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
          %Por el momento, por sensibilidad numérica del cero alfa4 es 90
          %y alfa5 -90. ajk_v identifica el eje x correspondiente en sentido
          %contrario. Hasta ver si con las pruebas reales o con valores que
          %no sean perfectos se obtiene correctamente.
          %Ocurre lo mismo con el MM de las dos últimas, por sensibilidad
          %numérica obtiene el vector al revés

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%

          Pn(j,:)=(cross(cross(ajk_v(j,:)',Sk),S0j)-...
              (dot(S0k,ajk_v(j,:)')*Sj))/dot(cross(ajk_v(j,:)',Sk),Sj);
          Pp(j,:)=(cross(cross(ajk_v(j,:)',Sj),S0k)-...
              (dot(S0j,ajk_v(j,:)')*Sk))/dot(cross(ajk_v(j,:)',Sj),Sk);
          ajk(j)=sqrt((Pp(j,1)-Pn(j,1))^2+(Pp(j,2)-Pn(j,2))^2+...
              (Pp(j,3)-Pn(j,3))^2);
          %Se le aplica signo a ajk para obtener valores entre`+0.1 y -0.1
          ajk(j)=ajk(j)*ajk_sign;%Se le aplica signo, si ha cambiado ajk_v

          %La fórmula que se aplica antes es: sin_alfa_jk=-(MM)/ajk(j);
          %No se puede usar, ya que se ha cambiado el signo de ajk
          %Se aplica otra fórmula, ver ecuación 14
          sin_alfa_jk=dot(ajk_v(j,:),cross(Sj,Sk)');

          %Para calcular alfajk, con atan2 -> da un valor entre -pi y +pi
          %si atan2>3pi/4 o atan2<-3pi/4 (cerca de +-180°)
          %  entonces se cambia atan2 entre 0 y 2pi
          alfajk(j)=atan2(sin_alfa_jk,cos_alfa_jk);%da valor entre -pi y +pi
          if (alfajk(j)>(3*pi/4)) || (alfajk(j)<(-3*pi/4))
              alfajk(j)=rad2deg(mod(alfajk(j),2*pi));%Cambia entre 0 y 2pi
          else
              alfajk(j)=rad2deg(alfajk(j));
          end
          %alfajk(j)=rad2deg(mod(atan2(sin_alfa_jk,cos_alfa_jk),2*pi));

        end
```

```
    end

    if (j==1)
        %% Calcula dj y thitaj para la articulación 1
        %No tengo punto de intersección anterior, por lo que no es posible
        %calcular d según el método del circle point. Se podría calcular
        %una d respecto al centro del círculo ajustado, que no coincidirá
        %con el origen del robot.
        %Como se tiene la matriz de cambio del SR del robot al SR LT,
        %se puede calcular de forma precisa el punto origen del robot
        %Punto origen del robot, en SR del robot
        OrigRob=[0;0;0;1];
        %Se calcula el punto origen del robot, en SRLT
        OrigRob=SRLT*OrigRob;
        OrigRob=OrigRob(1:3);

        %d1 es Pn - punto origen del robot, en SRLT
        dj(j)=norm(Pn(j,:)-OrigRob');

        %Ocurre lo mismo con tita. No tenemos a01, por lo que no es posible
        %calcular tita 1 según Circle-Point. Se podría hacer respecto al
        %SR global del LT, convirtiéndolo así en SR global del robot.
        %Lo que se hace es tomar el SR0, que es el sistema de referencia
        % del origen 0, tal y como se ha calculado en la anterior función
        % "Coordenadas_Plucker", y que ya incluye el desfase dado en
        % Angfin, y que se ha usado para la representación de los ejes.
        % Los vectores "a" van en dirección del eje X siguiente en los ejes
        % de los que podemos tener datos y por eso tita se obtiene como
        %producto escalar de dos vectores "a" consecutivos. Para ser
        %coherentes con los parámetros D-H, se supone que el a01 va en la
        %misma dirección que el a12 (definimos el SR0 igual que el 1 de
        %circle point). De este modo estamos poniendo el eje X del SR
        %en el eje X correspondiente al vector S1
        %Calculamos el ángulo con atan2 y mod para obtenerlo entre 0 y 2pi

        ajk_ejex=SR0(1:3)';%Toma el eje X para la articulación 0 del SR0

        sen_th=cross(ajk_ejex,ajk_v(j,:)');%resultado en fila
        sen_th=dot(sen_th,Sj');%cos del ángulo entre el vector ant. y Sj
        if sen_th<0
            %No está alineado con Sj => lo cambia
            ajk_ejex=-ajk_ejex;
        end

        %% Si ShowFigure --> muestra la figura con los circulos, ajk_v, Sj
        if (ShowFigure)
            longflecha=1;
            longflecha=100;
            scnsize = get(0,'ScreenSize');%Tamaño de la ventana
            pos=[scnsize(3)/10 scnsize(4)/10 8*scnsize(3)/10
8*scnsize(4)/10];
            h1=figure('Color','w','Name','Vectores ajk - Sj - thita',...
                'NumberTitle','off','Position',pos,'Resize','off');
            %hsp=subplot(2,3,1,'Parent',h1);

            plot3(Pn(j,1),Pn(j,2),Pn(j,3),'ko','MarkerSize',20);
            hold on;
            plot3(Pp(j,1),Pp(j,2),Pp(j,3),'ko','MarkerSize',20);
            r=[Pn(j,1),Pn(j,2),Pn(j,3)]';
            hold on;

            ptoini=[0 0 0];
```

```
            ptoini=r;
            arrow3D(ptoini,longflecha*ajk_ejex,'r');
            hold on;
            arrow3D(ptoini,longflecha*ajk_v(j,:),'g');
            hold on;
            arrow3D(ptoini,longflecha*Sj','b');
            hold on;
            arrow3D(ptoini,longflecha*(cross(ajk_ejex,ajk_v(j,:))),'m');
            hold on;
            axis equal;
        end
        %% Calcula tita_jk
        %Para calcular titajk, con atan2 -> da un valor entre -pi y +pi
        %si atan2>3pi/4 o atan2<-3pi/4 (cerca de +-180°)
        %  entonces se cambia atan2 entre 0 y 2pi
        titaj(j) = atan2(dot(cross(ajk_ejex,ajk_v(j,:)),Sj),...
            dot(ajk_ejex,ajk_v(j,:)));%da valor entre -pi y +pi
        if (titaj(j)>(3*pi/4)) || (titaj(j)<(-3*pi/4))
            titaj(j)=rad2deg(mod(titaj(j),2*pi));%Cambia entre 0 y 2pi
        else
            titaj(j)=rad2deg(titaj(j));
        end
        %titaj(j) = rad2deg(mod(atan2(dot(cross(ajk_ejeX,ajk_v(j,:)),Sj),...
        %    dot(ajk_ejeX,ajk_v(j,:))),2*pi));
    else
        %% Calcula dj y thitaj para el resto de articulaciones
        dj_v=(Pn(j,:)-Pp(j-1,:));
        dj(j)=norm(dj_v);
        %Según Denavit-Hartenberg, dj es la distancia desde el origen de la
        %articulación j-1, a lo largo del eje Z(j-1), hasta el eje Xj
        %(siguiente articulación). Por tanto, para obtener el signo de dj,
        %a partir del vector anterior dj_v, se calculo el ángulo que forma
        %éste con el eje Z de la articulación j-1, que es ahora Mant.
        %Si cos_ang es negativo, hay que cambiar el signo.
        %dj_v=(Pn(j,:)-Pp(j-1,:));
        cos_ang=dot(Mant(1:3,3),dj_v);
        if (cos_ang<0)
            dj(j)=-dj(j);
        end

        %% Si ShowFigure --> muestra la figura con los circulos, ajk_v, Sj
        if (ShowFigure)
            %lflec=1;
            lflec=100;
            if j>3  %Reserva el espacio 4 para la tabla
                k=j+1;
            else
                k=j;
            end

            plot3(Pn(j,1),Pn(j,2),Pn(j,3),'ko','MarkerSize',20);
            hold on;
            plot3(Pp(j,1),Pp(j,2),Pp(j,3),'ko','MarkerSize',20);
            r=[Pn(j,1),Pn(j,2),Pn(j,3)]';
            hold on;

            %hsp=subplot(2,3,k,'Parent',h1);
            ptoini=[0 0 0];
            ptoini=r;
            arrow3D(ptoini,lflec*ajk_v(j-1,:),'r');
            hold on;
            arrow3D(ptoini,lflec*ajk_v(j,:),'g');
            hold on;
```

```
            arrow3D(ptoini,lflec*Sj','b');
            hold on;
            arrow3D(ptoini,lflec*(cross(ajk_v(j-1,:),ajk_v(j,:))),'m');
            hold on;

            if (j==5)
                axis equal;
                %En la ultima articulación coloca la "leyenda"
                %Define colores: red, green, blue, magenta
                colrow=[1 0 0;0 1 0;0 0 1;1 0 1];
                nomfil={'    ajk(previo)    ',...
                        '    ajk(actual)    ',...
                        '        Sj         ',...
                        'ajk(pr) X ajk(act)'};
                 dd={'    '};

                figt1=uitable('Position',[60 200 270 20]);
                set(figt1,'data', dd,'RowName',nomfil(1),'ColumnName',[]);
                set(figt1,'ColumnWidth',{60},'BackgroundColor',colrow(1,:));
                figt2=uitable('Position',[60 180 270 20]);
                set(figt2,'data', dd,'RowName',nomfil(2),'ColumnName',[]);
                set(figt2,'ColumnWidth',{60},'BackgroundColor',colrow(2,:));
                figt3=uitable('Position',[60 160 270 20]);
                set(figt3,'data', dd,'RowName',nomfil(3),'ColumnName',[]);
                set(figt3,'ColumnWidth',{60},'BackgroundColor',colrow(3,:));
                figt4=uitable('Position',[60 140 270 20]);
                set(figt4,'data', dd,'RowName',nomfil(4),'ColumnName',[]);
                set(figt4,'ColumnWidth',{60},'BackgroundColor',colrow(4,:));

            end
        end
        %% Calcula tita_jk
        %Para calcular titajk, con atan2 -> da un valor entre -pi y +pi
        %si atan2>3pi/4 o atan2<-3pi/4 (cerca de +-180°)
        %  entonces se cambia atan2 entre 0 y 2pi
        titaj(j) = atan2(dot(cross(ajk_v(j-1,:),ajk_v(j,:)),Sj),...
            dot(ajk_v(j-1,:),ajk_v(j,:)));%da valor entre -pi y +pi
        if (titaj(j)>(3*pi/4)) || (titaj(j)<(-3*pi/4))
            titaj(j)=rad2deg(mod(titaj(j),2*pi));%Cambia entre 0 y 2pi
        else
            titaj(j)=rad2deg(titaj(j));
        end
        %titaj(j) = rad2deg(mod(atan2(dot(cross(ajk_v(j-
1,:),ajk_v(j,:)),Sj),...
        %    dot(ajk_v(j-1,:),ajk_v(j,:))),2*pi));

    end

    %% Calculo del sentido del ajk
    %Calculo de ajk: Segun el metodo de Denavit-Hartenberg, ai Es la
    % distancia que hay desde la intersección del eje zi-1 con el eje xi
    % hasta el origen del sistema i-ésimo, a lo largo del eje xi
    % Esto es la matriz de transformación M, columna 1, para el eje Xi
    % Si cos_ang es negativo, hay que cambiar el signo.
    cos_ang=dot(M(1:3,1),ajk_v(j,:));
    if (cos_ang<0)
        ajk(j)=-ajk(j);
    end


end
```

```
%% Ajusta el ángulo Thita según los ángulos finales obtenidos
%Tita es el ángulo entre los vectores a en la posición final del robot, que
%es en la que se obtienen los vectores S. Este ángulo será pues
%directamente el giro de la articulación respecto a su eje X
%correspondiente. De este modo, es posible obtener el offset que debe
%incluirse en el tita0 nominal para hacer coincidir la lectura del robot
%con el resultado de giro de circle point. De este modo corregimos la
%posición del eje X y por tanto el parámetro tita0 (offset de cero). El
%resultado de este método dará el ángulo TOTAL entre 2 ejes X consecutivos,
%por lo tanto teniendo en cuenta la lectura del encoder (ángulo que ha
%girado la articulación), así como el tita0, es decir, el giro entre ejes X
%antes de hacer el giro.

%OJO AL SIGNO DE ANGFIN
Angfin=-Angfin;
%Cambio de signo a la lectura del 5 por cambio en SRs
%Angfin(5)=-Angfin(5);
for j=1:5
   if (j==5)
       titaj(j)=Angfin(j)-titaj(j);%Esto no debe estar bien*******
       %deberia ser: titaj(j)=titaj(j)+Angfin(j);
   else
       titaj(j)=titaj(j)-Angfin(j);
   end
   %Este tita obtenido será el parámetro nominal más un cierto offset. Este
   %offset será la corrección directa a aplicar al parámetro. Como con
   %atan2 se obtiene el ángulo en el cuadrante correcto, no es necesario
   %hacer correcciones según singnos de senos y cosenos. Con las fórmulas
   %del método original basadas en la tangente los ángulos son una
   %pesadilla. Esto explicarlo bien.
   if (titaj(j)>360)
       titaj(j)=titaj(j)-360;%no se contempla más de 2 vueltas.
   end
end

%% Asigna parámetros al último sistema de referencia
%Faltan los parámetros del último sistema de ref, que en notación de circle
%point sería el 7. Se supondrá que este sistema es igual que el 6 con los
%parámetros del modelo nominal, es decir d=80, a=0, tita= 0(cambio en SR5),
alfa=0.
%También se podría hacer como con el primero, poniendo el origen en el
%centro del círculo
ajk(6)=0;
alfajk(6)=0;
titaj(6)=0;
dj(6)=80;
beta_jk(6)=0;

%% Devuelve dato de salida P
%Nota: si angulo>180 o <-180, se podría ajustar
%tita5 no da bien - da aprox -220
P=[dj titaj ajk alfajk beta_jk];
%*****LO SIGUIENTE ES LO QUE PROVOCA ALFA2=POSITIVO SIEMPRE
%if (P(20)>270)
%    %Puede dar 360 por cero
%    P(20)=360-P(20);
%end
```

17.4 Montecarlo Circle Point

```
function [Pmed,uPmed,I_inf,I_sup,maxmed, minmed, DifP, CP_Param,
timefunc]=Montecarlo_Circle_Point(n, Measunc, Initpos, SRLT, Preflector,
Angcir, Nptos, Err, Param5, r0, IntCov)

%Realiza n simulaciones numéricas de cálculo
%de parámetros mediante el método del CIRCLE_POINT. En salida obtiene la
%media y la desviación estándar de los valores de cada parámetro,
%obteniendo por tanto el valor medido y la incertidumbre en la obtención de
%cada parámetro cinemático del modelo.

%PARÁMETROS
%-n: Número de iteraciones

%-MeasUnc: vector 1x2 con la Incertidumbre de medida nominal del equipo con
%el que se han capturado los puntos (en mm). Por ejemplo:
%Láser tracker 5+0.3L micrometros por metro con L en m --> Measunc=[5 0.0003]

%-IntCov: Porcentaje de datos para el intervalo de cobertura (ej. 95)

%Salida
%-Pmed: valor medio del vector de parámetros 1x30

%uPmed: Desviación estandar (Incertidumbre típica) de cada parámetro 1x30

%-I_inf, I_sup: Vectores con los intervalos de confianza de cada parámetro
%1x30

%maxmed, minmed: Máximos y minimos de los parámetros obtenidos

%-DifP: Error en la determinación de cada parámetro (medio-nominal) 1x30

%-CP_Param: Matriz de parámetros obtenida nx30

%-timefunc: Tiempo que ha necesitado la función en ejecutarse

%% Inicializa variables
%Inicio del contador del tiempo
tic;
CP_Param=[];
tabladat=[];
CP_Param=zeros(n,30);
tabladat=zeros(6,30);
showfig=false;
CheckX=true;%Para chequear que el vector ajk coincide con la direccion X

%% Realiza iteraciones por Montecarlo
for i=1:n
  [Ptos_Circle_Point, Ptos_Original,
randomnoise]=Genera_puntos_circle_point(Initpos, SRLT, Preflector, Angcir,
Nptos, Measunc, Err, Param5, showfig);
  [Plucker, P_Circulos, SR0, Distcirculo, Distplano, Distcilindro,
sigmaPlano, sigmaCirculo, sigmaCilindro, Angfin]=...
    Coordenadas_Plucker_Ejes(Ptos_Circle_Point, Nptos, r0, Param5, showfig,
showfig, showfig, showfig, showfig, SRLT);
  P=Calcula_Parametros_KM(Plucker, P_Circulos, Param5, Angfin, CheckX, SRLT,
SR0, showfig);
  CP_Param(i,:)=P;
```

```
  if ((mod(i,1000))==0);fprintf('Iteracion Numero = %d\n',i);end
end

%% Obtener valor medio, incertidumbre,  intervalos de confianza para cada
%   parámetro
%Media
Pmed=mean(CP_Param);
tabladat(1,:)=Pmed;
DifP=Pmed-Param5;

%desviación estándar
uPmed=std(CP_Param);
tabladat(2,:)=uPmed;

%intervalo de cobertura
CP_Param=sort(CP_Param,1);
I_inf=CP_Param(round((((100-IntCov)/2)/100)*n),:);
I_sup=CP_Param(round((1-(((100-IntCov)/2)/100))*n),:);
tabladat(3,:)=I_sup;
tabladat(4,:)=I_inf;

%Maximos y minimos
maxmed=max(CP_Param);
minmed=min(CP_Param);
tabladat(5,:)=maxmed;
tabladat(6,:)=minmed;

%Fin contaje del tiempo
timefunc=toc;

%% Muestra gráficas distribución normal de cada parámetro
h1=figure('Color','w','Name','Gráficas de Distribución normal por
Parámetro','NumberTitle','off');
for i=1:26
    if i==25; i=i+1; end;%Si es el 25(beta1), pasa a beta2

    hsp=subplot(5,6,i,'Parent',h1);
    histfit(CP_Param(:,i),100);
    hold on;

    %Ajusta el ancho del eje X
    %Se guarda el intervalo, que es un cierto valor a añadir=1/3 media-minimo
    dif=max(abs(Pmed(i)-minmed(i)),abs(maxmed(i)-Pmed(i)));
    xsup=Pmed(i)+(6*dif/5); xinf=Pmed(i)-(6*dif/5);
    xx=[xinf, xsup];
    %Si hay diferencia
    if dif>0
        set(hsp,'Xlim',xx);%Ajusta eje X de la gráfica

        %Coloca las líneas del intervalo de confianza: I_sup - I_inf
        ylim=get(hsp,'Ylim');
        xsup=[];xinf=[];
        yi=[0 ylim(2)];
        xsup=I_sup(i)*ones(1,size(yi,2));
        xinf=I_inf(i)*ones(1,size(yi,2));
        plot(xsup,yi,'-b','LineWidth',2);
        hold on;
        plot(xinf,yi,'-b','LineWidth',2);
        hold on;
    end

    %Ajusta al título del subplot
```

```
    if ((i>=1) && (i<=6))
        title(['d ' num2str(i)],'fontSize',14,'fontweight','b');
    elseif ((i>=7) && (i<=12))
        title(['\it\theta ' num2str(i-6)],'fontSize',14,'fontweight','b');
    elseif ((i>=13) && (i<=18))
        title(['a ' num2str(i-12)],'fontSize',14,'fontweight','b');
    elseif ((i>=19) && (i<=24))
        title(['\it\alpha ' num2str(i-18)],'fontSize',14,'fontweight','b');
    else      %(i==26)
        title(['\it\beta' '2'],'fontSize',14,'fontweight','b');
    end

end


%% Muestra tabla de datos resumen
h2=figure('Position',[50 280 1054 440],'Name','Tabla de datos de
Parámetros','NumberTitle','off');

figt1=uitable('Position',[5 300 1045 130]);
nomfil={'Media','Incertidumbre','Lim.Sup','Lim.Inf','Maximo','Minimo'};
nomcol={'d1','d2','d3','d4','d5','d6','thita 1','thita 2','thita 3','thita
4','thita 5','thita 6'};
set(figt1,'data', tabladat(:,1:12),'ColumnName',nomcol,'RowName',nomfil);

figt2=uitable('Position',[5 160 1045 130]);
nomcol={'a1','a2','a3','a4','a5','a6','alfa 1','alfa 2','alfa3','alfa
4','alfa 5','alfa 6'};
set(figt2,'data', tabladat(:,13:24),'ColumnName',nomcol,'RowName',nomfil);

figt3=uitable('Position',[5 20 595 130]);
nomcol={'beta 1','beta 2','beta 3','beta 4','beta 5','beta 6'};
set(figt3,'data', tabladat(:,25:30),'ColumnName',nomcol,'RowName',nomfil);

axis off;
hold on;

%% Muestra el tiempo empleado
 %Muestra el tiempo
h3=figure('Color','w','Position',[40 500 440 80],'Name','Duración de la
Simulación','NumberTitle','off');
figt31=uitable('Position',[20 20 400 40]);
nomfil={'Tiempo empleado'};
nomcol={'Horas','Minutos','Segundos'};
tiempohms(1)=fix(timefunc/3600);%Horas
trest=timefunc-tiempohms(1)*3600;
tiempohms(2)=fix(trest/60);%minutos
tiempohms(3)=trest-tiempohms(2)*60;%segundos
set(figt31,'data', tiempohms,'ColumnName',nomcol,'RowName',nomfil);
```

17.5 Simulacion_MCP

```
function tabladat=Simulacion_MCP(simul, n)

%Selecciona la simulación elegida y realiza la llamada a la misma

%PARÁMETROS
%-simul: Selecciona la simulación a realizar

%-n: Número de iteraciones


%% Inicializa variables
tabladat=[];
tabladat=zeros(6,30);
%Faro Laser Tracker ION
Ruidomed=[10 2.5];%Ruido del equipo medición
%Ruidomed=[5 0.3];%Ruido del equipo medición
Initpos=[15 0 0 0 0 0];%Posición inicial
SRLT1=[1 0 0 0;...
    0 1 0 0;...
    0 0 1 0;...
    0 0 0 1];%Matriz de cambio del SR robot al SR Laser Tracker- identidad
SRLT=[0.00660053 -0.999926909 -0.010130127 -0.968178687;...
0.999956924 0.006666069 -0.006452643 -1862.865839;...
0.006519756 -0.010087073 0.999927839 -637.5778528;...
0 0 0 1];%Matriz de cambio del SR robot al SR Laser Tracker
SRLT1=[0.0066005 0.999957 0.0065197 1866.94894;...
    -0.9999269 0.0066661 -0.0100871 5.0186305;...
    -0.0101301 -0.0064527 0.9999279 625.501561;...
    0 0 0 1];%Matriz de cambio 3 - del LT al robot

Preflector=[100 0 0; 200 0 0; 0 50 0;...
    20 0 0; 0 80 0; 0 60 0];%Posiciones iniciales del reflector
Angcir=[45 80 100 50 200 300];%Angulos a describir por el reflector
Nptos=[15 20 30 15 25 30];%Número de puntos por articulación
Err=[0 0 0 0 0 2*pi 2*pi 2*pi 2*pi 2*pi 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 2*pi 2*pi 2*pi 2*pi 2*pi 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 2*pi 2*pi 2*pi 2*pi 2*pi 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 2*pi 2*pi 2*pi 2*pi 2*pi 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 2*pi 2*pi 2*pi 2*pi 2*pi 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 2*pi 2*pi 2*pi 2*pi 2*pi 0 0 0 0 0 0 0 0 0 0];%Errores
%Parámetros iniciales según D-H
Param5=[335 0 0 295 0 80 0 0 90 0 180 0 75 270 90 0 0 0 90 0 90 90 90 0 0 0 0
0 0 0];
r0=[35 35 35 35 35 35];%Radio que describe el reflector
IntCov=95;

%% Selecciona la simulación a realizar, modificando el parámetro
switch simul
    %Simulaciones 10's: diferentes equipos de medición
    case 11
        %Faro Laser Tracker ION
        Ruidomed=[10 2.5];%Ruido del equipo medición
    case 12
        %Leica Laser Tracker Absoluto AT-901MR
        Ruidomed=[15 6];%Ruido del equipo medición
    case 13
        %Leica Estación Láser TDRA6000
        Ruidomed=[300 13];
    case 14
```

```
        %Leica Estación Total TCA
        Ruidomed=[1000 1];
    case 15
        %Nikon Indoor GPS
        Ruidomed=[200 0];

    %Simulaciones 20's: Variar el ángulo cubierto para la
    % articulación 4 (15,40,90,180)
    case 21
        Angcir(4)=15;
    case 22
        Angcir(4)=40;
    case 23
        Angcir(4)=90;
    case 24
        Angcir(4)=180;

    %Simulaciones 30's: modificando la posición del reflector
    case 31
        %Preflector posición nominal
        Preflector(3,:)=[0 50 0];
    case 32
        %Reflector en posición 2
        Preflector(3,:)=[0 70 20];
    case 33
        %Reflector en posición 3
        Preflector(3,:)=[20 100 50];

    %Simulaciones 40's: modificando el número de puntos
    case 41
        %Número de puntos = 10
        Nptos=[10 10 10 10 10 10];%Número de puntos por articulación
    case 42
        %Número de puntos = 40
        Nptos=[40 40 40 40 40 40];%Número de puntos por articulación
    case 43
        %Número de puntos = 100
        Nptos=[100 100 100 100 100 100];%Número de puntos por articulación

    %Simulaciones 50's: modificando la matriz de transformación
    case 51
        %Matriz original
        SRLT=SRLT;
    case 52
        %Matriz segunda, con un primer desplazamiento para alejarla 400mm
        SRLT=[0.006618494 -0.999928439 -0.009965562 -2.298104298;...
            0.999955407 0.006685151 -0.006670312 -2062.847829;...
            0.006736456 -0.00992097 0.999928095 -638.9524073;...
            0 0 0 1];
    case 53
        %Matriz tercera, con un desplazamiento de 2500mm, y +200mm en Z
        SRLT=[0.006618494 -0.999928439 -0.009965562 -19.51375337;...
            0.999955407 0.006685151 -0.006670312 -4364.079327;...
            0.006736456 -0.00992097 0.999928095 -454.4606377;...
            0 0 0 1];
    otherwise
        fprintf('No es una simulacion definida');

end

%% Llama al simulador
[Pmed,uPmed,I_inf,I_sup, maximos, minimos, DifP, CP_Param,tiempo]=...
            Montecarlo_Circle_Point(n, Ruidomed, Initpos, SRLT,...
```

```
            Preflector, Angcir, Nptos, Err, Param5, r0, IntCov);

%% Salva datos
tabladat(1,:)=Pmed;
tabladat(2,:)=uPmed;
tabladat(3,:)=I_sup;
tabladat(4,:)=I_inf;
tabladat(5,:)=maximos;
tabladat(6,:)=minimos;
```

17.6 Matriz LT Robot

```
function [LTRB,RB, M]=Matriz_LT_Robot(LT, AngRobot, Param5)

%Obtiene la matriz de transformación homogénea que transforma coordenadas
%en SR LT a coordenadas Robot.

%PARAMETROS
%-LT: matriz nx3 con las coordenadas de los puntos del LT
%-AngRobot: matriz nx6 con los ángulos de las articulaciones del robot
%correspondientes a cada punto
%-Param5: Vector de parámetros del robot

%SALIDA
%-M: matriz de transformación
%-LTRB: Puntos del LT expresados en sist. de ref Robot

n=size(AngRobot,1);

for i=1:n
  [M0A6 M0A1 M1A2 M2A3 M3A4 M4A5 M5A6] =
PBETAfRobotMKR5sixxDHHpar(AngRobot(i,:), Param5);
  RB(:,i)=M0A6(1:3,4);
end


B=RB;
A=LT';
[s R T err] = absoluteOrientationQuaternion(A,B,1);
%R=inv(R);
M=[R(1,:) T(1,1); R(2,:) T(2,1); R(3,:) T(3,1); 0 0 0 1];


for i = 1:n
  p4=[LT(i,:)'; 1];
  p41=M*p4;
  LTRB(i, :)=p41(1:3)';
end
RB=RB';
```

18 Anexo X – Funciones MatLab usadas de apoyo

A continuación se incluyen aquellas funciones que se han utilizado, como apoyo al desarrollo del proyecto, y que se han tomado de diversas fuentes.

18.1 Arrow 3D

```
function arrowHandle = arrow3D(pos, deltaValues, colorCode, stemRatio)

% arrowHandle = arrow3D(pos, deltaValues, colorCode, stemRatio)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%     Used to plot a single 3D arrow with a cylindrical stem and cone
arrowhead
%     pos = [X,Y,Z] - spatial location of the starting point of the arrow
(end of stem)
%     deltaValues = [QX,QY,QZ] - delta parameters denoting the magnitude of
the arrow along the x,y,z-axes (relative to 'pos')
%     colorCode - Color parameters as per the 'surf' command.  For example,
'r', 'red', [1 0 0] are all examples of a red-colored arrow
%     stemRatio - The ratio of the length of the stem in proportion to the
arrowhead.  For example, a call of:
%                 arrow3D([0,0,0], [100,0,0] , 'r', 0.82) will produce a red
arrow of magnitude 100, with the arrowstem spanning a distance
%                 of 82 (note 0.82 ratio of length 100) while the arrowhead
(cone) spans 18.
%
%     Example:
%       arrow3D([0,0,0], [4,3,7]);  %---- arrow with default parameters
%       axis equal;
%
%    Author: Shawn Arseneau
%    Created: September 14, 2006
%    Updated: September 18, 2006
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if nargin<2 || nargin>4
        error('Incorrect number of inputs to arrow3D');
    end
    if numel(pos)~=3 || numel(deltaValues)~=3
        error('pos and/or deltaValues is incorrect dimensions (should be
three)');
    end
    if nargin<3
        colorCode = 'interp';
    end
    if nargin<4
        stemRatio = 0.75;
    end

    X = pos(1); %---- with this notation, there is no need to transpose if
the user has chosen a row vs col vector
    Y = pos(2);
    Z = pos(3);

    [sphi, stheta, srho] = cart2sph(deltaValues(1), deltaValues(2),
deltaValues(3));
```

```
    %****************************************** CYLINDER == STEM
*******************************************
    %cylinderRadius = 0.05*srho;
    cylinderRadius = 0.03*srho;
    cylinderLength = srho*stemRatio;
    [CX,CY,CZ] = cylinder(cylinderRadius);
    CZ = CZ.*cylinderLength;    %---- lengthen

    %----- ROTATE CYLINDER
    [row, col] = size(CX);      %---- initial rotation to coincide with X-
axis

    newEll = rotatePoints([0 0 -1], [CX(:), CY(:), CZ(:)]);
    CX = reshape(newEll(:,1), row, col);
    CY = reshape(newEll(:,2), row, col);
    CZ = reshape(newEll(:,3), row, col);

    [row, col] = size(CX);
    newEll = rotatePoints(deltaValues, [CX(:), CY(:), CZ(:)]);
    stemX = reshape(newEll(:,1), row, col);
    stemY = reshape(newEll(:,2), row, col);
    stemZ = reshape(newEll(:,3), row, col);

    %----- TRANSLATE CYLINDER
    stemX = stemX + X;
    stemY = stemY + Y;
    stemZ = stemZ + Z;

    %******************************************** CONE == ARROWHEAD
*********************************************
    coneLength = srho*(1-stemRatio);
    coneRadius = cylinderRadius*1.5;
    incr = 4;  %---- Steps of cone increments
    coneincr = coneRadius/incr;
    [coneX, coneY, coneZ] = cylinder(cylinderRadius*2:-coneincr:0);  %-------
--- CONE
    coneZ = coneZ.*coneLength;

    %----- ROTATE CONE
    [row, col] = size(coneX);
    newEll = rotatePoints([0 0 -1], [coneX(:), coneY(:), coneZ(:)]);
    coneX = reshape(newEll(:,1), row, col);
    coneY = reshape(newEll(:,2), row, col);
    coneZ = reshape(newEll(:,3), row, col);

    newEll = rotatePoints(deltaValues, [coneX(:), coneY(:), coneZ(:)]);
    headX = reshape(newEll(:,1), row, col);
    headY = reshape(newEll(:,2), row, col);
    headZ = reshape(newEll(:,3), row, col);

    %---- TRANSLATE CONE
    V = [0, 0, srho*stemRatio];    %---- centerline for cylinder: the
multiplier is to set the cone 'on the rim' of the cylinder
    Vp = rotatePoints([0 0 -1], V);
    Vp = rotatePoints(deltaValues, Vp);
    headX = headX + Vp(1) + X;
    headY = headY + Vp(2) + Y;
    headZ = headZ + Vp(3) + Z;

%*************************************************************************
******************************
```

```
    hStem = surf(stemX, stemY, stemZ, 'FaceColor', colorCode, 'EdgeColor',
'none');
    hold on;
    hHead = surf(headX, headY, headZ, 'FaceColor', colorCode, 'EdgeColor',
'none');

    if nargout==1
        arrowHandle = [hStem, hHead];
    end
```

18.2 Lsplane

```
  function [x0, a, d, normd] = lsplane(X)
% ---------------------------------------------------------------------
% LSPLANE.M   Least-squares plane (orthogonal distance
%             regression).
%
% Version 1.0
% Last amended   I M Smith 27 May 2002.
% Created        I M Smith 08 Mar 2002
% ---------------------------------------------------------------------
% Input
% X        Array [x y z] where x = vector of x-coordinates,
%          y = vector of y-coordinates and z = vector of
%          z-coordinates.
%          Dimension: m x 3.
%
% Output
% x0       Centroid of the data = point on the best-fit plane.
%          Dimension: 3 x 1.
%
% a        Direction cosines of the normal to the best-fit
%          plane.
%          Dimension: 3 x 1.
%
% <Optional...
% d        Residuals.
%          Dimension: m x 1.
%
% normd    Norm of residual errors.
%          Dimension: 1 x 1.
% ...>
%
% [x0, a <, d, normd >] = lsplane(X)
% ---------------------------------------------------------------------

% check number of data points
  m = size(X, 1);
  if m < 3
    error('At least 3 data points required: ' )
  end
%
% calculate centroid
  x0 = mean(X)';
%
% form matrix A of translated points
  A = [(X(:, 1) - x0(1)) (X(:, 2) - x0(2)) (X(:, 3) - x0(3))];
%
```

```
% calculate the SVD of A
  [U, S, V] = svd(A, 0);
%
% find the smallest singular value in S and extract from V the
% corresponding right singular vector
  [s, i] = min(diag(S));
  a = V(:, i);
%
% calculate residual distances, if required
  if nargout > 2
    d = U(:, i)*s;
    normd = norm(d);
  end
% ---------------------------------------------------------------------
% End of LSPLANE.M.
```

18.3 Ls3dcircle

```
  function [x0n, an, rn, d, e, f, sigmah, conv, Vx0n, Van, urn, ...
            GNlog, a, R0, R] = ls3dcircle(X, x0, a0, r0, tolp, tolg, w)
% ---------------------------------------------------------------------
% LS3DCIRCLE.M   Least-squares circle in three dimensions using
%                Gauss-Newton.
%
% Version 1.0
% Last amended   I M Smith 27 May 2002.
% Created        I M Smith 08 Mar 2002
% ---------------------------------------------------------------------
% Input
% X        Array [x y z] where x = vector of x-coordinates,
%          y = vector of y-coordinates and z = vector of z-coordinates.
%          Dimension: m x 3.
%
% x0       Estimate of the circle centre.
%          Dimension: 3 x 1.
%
% a0       Estimate of the normal to the plane containing the circle.
%          Dimension: 3 x 1.
%
% r0       Estimate of the circle radius.
%          Dimension: 1 x 1.
%
% tolp     Tolerance for test on step length.
%          Dimension: 1 x 1.
%
% tolg     Tolerance for test on gradient.
%          Dimension: 1 x 1.
%
% <Optional...
% w        Weights.
%          Dimension: m x 1.
% ...>
%
% Output
% x0n      Estimate of the circle centre.
%          Dimension: 3 x 1.
%
% an       Estimate of the normal direction
%          Dimension: 3 x 1.
```

```
%
% rn        Estimate of the circle radius
%           Dimension: 1 x 1.
%
% d         Vector of distances from the points to the circle
%           Dimension: m x 1.
%
% e         Vector of distances from the points to the plane
%           containing the circle
%           Dimension: m x 1.
%
% f         Vector of distances from the points to the cylinder
%           containing the circle
%           Dimension: m x 1.
%
% sigmah    Estimate of the standard deviation of the weighted
%           residual errors.
%           Dimension: 1 x 1.
%
% conv      If conv = 1 the algorithm has converged,
%           if conv = 0 the algorithm has not converged
%           and x0n, rn, d, e, f and sigmah are current estimates.
%           Dimension: 1 x 1.
%
% Vx0n      Covariance matrix of circle centre.
%           Dimension: 3 x 3.
%
% Van       Covariance matrix of normal direction.
%           Dimension: 3 x 3.
%
% urn       Uncertainty in circle radius.
%           Dimension: 1 x 1.
%
% GNlog     Log of the Gauss-Newton iterations.
%           Rows 1 to niter contain
%           [iter, norm(f_iter), |step_iter|, |gradient_iter|].
%           Row (niter + 1) contains
%           [conv, norm(d), 0, 0].
%           Dimension: (niter + 1) x 4.
%
% a         Optimisation parameters at the solution.
%           Dimension: 6 x 1.
%
% R0        Fixed rotation matrix.
%           Dimension: 3 x 3.
%
% R         Upper-triangular factor of the Jacobian matrix
%           at the solution.
%           Dimension: 6 x 6.
%
% Modular structure: NLSS11.M, GNCC2.M, FG3DCIRCLE.M, ROT3Z.M, GR.M,
%                    FGRROT3.M, FRROT3.M, DRROT3.M.
%
% [x0n, an, rn, d, e, f, sigmah, conv, Vx0n, Van, urn, GNlog, a, ...
%   R0, R] = ls3dcircle(X, x0, a0, r0, tolp, tolg <, w >)
% ----------------------------------------------------------------------

% check number of data points
  m = size(X, 1);
  if m < 6
    error('At least 6 data points required: ' )
  end
%
% if no weights are specified, use unit weights
```

```
  if nargin == 6
    w = ones(m, 1);
  end % if nargin
%
% find the centroid of the data
  xb = mean(X)';
%
% transform the data to close to standard position via a rotation
% followed by a translation
  R0 = rot3z(a0); % R0 * a0 = [0 0 1]'
  xb1 = R0 * xb;
  x1 = R0 * x0;
  X1 = (X * R0');
% find xp, the point on axis nearest the centroid of the rotated data
  xp = x1 + (xb1(3) - x1(3)) * [0 0 1]';
% translate data, mapping xp to the origin
  X2 = X1 - ones(m, 1) * xp';
  x2 = x1 - xp;
%
  ai = [0 0 0 0 0 r0]';
  tol = [tolp; tolg]';
%
% Gauss-Newton algorithm to find estimate of roto-translation
% parameters that transform the data so that the best-fit circle
% is one in standard position
  [a, ef, R, GNlog] = nlss11(ai, tol, 'fg3dcircle', X2, w);
  e = ef(1:m);
  f = ef(m+1:2*m);
  d = sqrt(e .* e + f .* f);
%
% inverse transformation to find circle centre and normal
% corresponding to original data
  rn = a(6); % radius
  [R3, DR1, DR2, DR3] = fgrrot3([a(4:5); 0]);
  an = R0' * R3' * [0 0 1]'; % axis
  x0n = R0' * (xp + [a(1) a(2) a(3)]'); % centre
%
  nGN = size(GNlog, 1);
  conv = GNlog(nGN, 1);
  if conv == 0
    beep;
    warning('*** Gauss-Newton algorithm has not converged ***');
  end % if conv
%
% calculate statistics
  dof = 2 * m - 6;
  sigmah = norm(d)/sqrt(dof);
  G = zeros(7, 6);
  G(1:3, 1) = R0' * [1 0 0]';
  G(1:3, 2) = R0' * [0 1 0]';
  G(1:3, 3) = R0' * [0 0 1]';
  G(4:6, 4) = R0' * DR1' * [0 0 1]';
  G(4:6, 5) = R0' * DR2' * [0 0 1]';
  G(7, 6)   = 1;
  Gt = R'\(sigmah * G'); % R' * Gt = sigmah * G'
  Va = Gt' * Gt;
  Vx0n = Va(1:3, 1:3); % covariance matrix for x0n
  Van = Va(4:6, 4:6); % covariance matrix for an
  urn = sqrt(Va(7, 7)); % uncertainty in rn
% ----------------------------------------------------------------------
% End of LS3DCIRCLE.M
```

18.4 Subplot_title

```
function title_handle = subplot_title(title_string)
% subplot_title - displays a title for a subplot, across all subplots
%
% Usage: title_handle = subplot_title(title_string)
%
% Displays a title for a set of subplots at the top of the current figure.
% Said title goes across all of the subplots.  Returns the handle to the
% subplot.
%
% subplot_title is a modified version of code by Keith Rogers and posted
% on comp.soft-sys.matlab on 1995/05/14 in the thread entitled
% "Global title for subplot".
%
% This function is part of froi, available from http://froi.sourceforge.net,
% and is governed under the terms of the Artistic License.
 %
% $Id$

ax = gca;
fig = gcf;

title_handle = axes('position',[.1 .9 .8 .05],'Box','off','Visible','off');

title(title_string);
set(get(gca,'Title'),'Visible','On');
set(get(gca,'Title'),'FontSize',24);
set(get(gca,'Title'),'FontWeight','bold');
axes(ax);
```

18.5 AbsoluteOrientationQuaternion

```
% [s R T error] = absoluteOrientationQuaternion( A, B, doScale)
%
% Computes the orientation and position (and optionally the uniform scale
% factor) for the transformation between two corresponding 3D point sets Ai
% and Bi such as they are related by:
%
%     Bi = sR*Ai+T
%
% Implementation is based on the paper by Berthold K.P. Horn:
% "Closed-from solution of absolute orientation using unit quaternions"
% The paper can be downloaded here:
% http://people.csail.mit.edu/bkph/papers/Absolute_Orientation.pdf
%
% Authors:      Dr. Christian Wengert, Dr. Gerald Bianchi
% Copyright:    ETH Zurich, Computer Vision Laboratory, Switzerland
%
% Parameters:   A           3xN matrix representing the N 3D points
%               B           3xN matrix representing the N 3D points
%               doScale     Flag indicating whether to estimate the
%                           uniform scale factor as well [default=0]
%
% Return:       s           The scale factor
%               R           The 3x3 rotation matrix
%               T           The 3x1 translation vector
%               err         Residual error
%
% Notes: Minimum 3D point number is N > 4
function [s R T err] = absoluteOrientationQuaternion( A, B, doScale)


%Argument check
if(nargin<3)
    doScale=1;
end
%Return argument check
if(nargout<1)
    usage()
    error('Specify at least 1 return arguments.');
end
%Test size of point sets
[c1 r1] = size(A);
[c2 r2] = size(B);
if(r1~=r2)
    usage()
    error('Point sets need to have same size.');
end
if(c1~=3 | c2~=3)
    usage()
    error('Need points of dimension 3');
end
if(r1<4)
    usage()
    error('Need at least 4 point pairs');
end

%Number of points
Na = r1;

%Compute the centroid of each point set
```

```
Ca = mean(A,2);
Cb = mean(B,2);

%Remove the centroid
An = A - repmat(Ca,1,Na);
Bn = B - repmat(Cb,1,Na);

%Compute the quaternions
M = zeros(4,4);
for i=1:Na
    %Shortcuts
    a = [0;An(:,i)];
    b = [0;Bn(:,i)];
    %Crossproducts
    Ma = [  a(1) -a(2) -a(3) -a(4) ;
            a(2)  a(1)  a(4) -a(3) ;
            a(3) -a(4)  a(1)  a(2) ;
            a(4)  a(3) -a(2)  a(1)  ];
    Mb = [  b(1) -b(2) -b(3) -b(4) ;
            b(2)  b(1) -b(4)  b(3) ;
            b(3)  b(4)  b(1) -b(2) ;
            b(4) -b(3)  b(2)  b(1)  ];
    %Add up
    M = M + Ma'*Mb;
end

%Compute eigenvalues
[E D] = eig(M);

%Compute the rotation matrix
e = E(:,4);
M1 = [  e(1) -e(2) -e(3) -e(4) ;
        e(2)  e(1)  e(4) -e(3) ;
        e(3) -e(4)  e(1)  e(2) ;
        e(4)  e(3) -e(2)  e(1)  ];
M2 = [  e(1) -e(2) -e(3) -e(4) ;
        e(2)  e(1) -e(4)  e(3) ;
        e(3)  e(4)  e(1) -e(2) ;
        e(4) -e(3)  e(2)  e(1)  ];


R = M1'*M2;

%Retrieve the 3x3 rotation matrix
R = R(2:4,2:4);
%Compute the scale factor if necessary
if(doScale)
    a =0; b=0;
    for i=1:Na
        a = a + Bn(:,i)'*R*An(:,i);
        b = b + Bn(:,i)'*Bn(:,i);
    end
    s = b/a;
else
    s = 1;
end

%Compute the final translation
T = Cb - s*R*Ca;

%Compute the residual error
if(nargout>3)
```

```
    err =0;
    for i=1:Na
        d = (B(:,i) - (s*R*A(:,i) + T));
        err = err + d'*d;
    end
    err = sqrt(err)/Na;
end


%Displayed if an error occurs
function usage()
disp('Usage:')
disp('[s R T error] = absoluteOrientationQuaternion( A, B, doScale)')
disp(' ')
disp('Return values:')
disp('s         The scale factor')
disp('R         The 3x3 rotation matrix')
disp('T         The 3x1 translation vector')
disp('err       Residual error (optional)')
disp(' ')
disp('Input arguments:')
disp('A         3xN matrix representing the N 3D points')
disp('B         3xN matrix representing the N 3D points')
disp('doScale   Optional flag indicating whether to estimate the uniform
scale factor as well [default=0]')
disp(' ')
```

19 Anexo XI: Equipos de Medición considerados

19.1 Resumen de equipos de medición

Se incluye aquí a continuación una tabla resumen con todos los equipos de medición que se han considerado en el cálculo.

Tabla 13 - Incertidumbre de equipos de medición

Equipo de medición	Rango de trabajo	Resolución (mínima)	Incertidumbre máxima en medición longitudinal	Incertidumbre máxima en medición angular
FARO Láser Tracker ION	110m (diámetro)	0.5 µm ±2"	±2µm + 0.4µm/m	±10µm + 2.5µm/m
LEICA Láser Tracker absoluto AT901-MR	50m	0.32 µm 0.14"	±0.5µm	±15µm + 6µm/m
LEICA Estación Láser TDRA6000	100m[10]	0.5mm	0.3mm + 13µm/m	*No aplicable*
LEICA Estación Total TCA2003	Hasta 2500m[11]	0.5 arc sec 1mm	1mm + 1ppm	*No aplicable*
NIKON Indoor GPS	Sin límite	0.2mm	0.2mm	*No aplicable*

Como otros equipos de medición se pueden considerar los siguientes:

Tabla 14 - Otros equipos de medición

Equipo de medición	Rango de trabajo	Resolución (mínima)	Precisión mínima (Incertidumbre máx)
LEICA T-Probe AT901-MR	18m	---	100µm
LEICA T-Mac	18m	---	±15µm + 6µm/m

[10] Distancia mínima de trabajo: 2 metros.
[11] Distancia mínima de trabajo: 5 metros.

19.2 FARO Láser Tracker ION

www.faro.com

FARO Laser Tracker ION™

Agile ADM

Acquire targets even if they are moving; no need to switch between ADM and IFM systems

SelfComp

Automatically tunes Laser Tracker parameters to ensure high accuracy

Versatile Mounting Options

Mounts vertically, horizontally or upside down, providing versatility in tight or congested areas*

**Inverted mounting requires the use of the integrated threaded ring.*

Instant-On Laser

No warm-up of the laser tube is required

Smart Warm-Up

Accelerates the stabilization time in order to minimize the initial temperature changes' impact on measurements

Integrated Weather Station

Monitors and compensates for changes in temperature, air pressure and humidity

Integrated Precision Level

Establishes level to gravity within the measurement job

World's Most Accurate Large Volume Laser Tracker

The FARO Laser Tracker ION is an extremely accurate, portable coordinate measuring machine that enables you to build products, optimize processes, and deliver solutions by measuring more quickly, simply and precisely than previously possible. The ION is the most accurate laser tracker available based on the most common types of measurement applications. It also features a longer measurement range, lighter weight, and contains the fastest, most sophisticated distance measuring system: Agile Absolute Distance Meter (aADM).

Common Applications

Alignment: Real-time feedback of object positioning
Installation: Lay out / level machine foundation
Part Inspection: Digital record of actual vs nominal data
Tool Building: Set up and inspect tools with only one person
Reverse Engineering: Acquire high accuracy digital scan data

Features

- 0.049mm volumetric accuracy at 10m
- 110m* diameter range
- Agile ADM instant beam acquisition
- As light as 17.7kg
- High performance, real-time dynamic measurements

**with selected targets*

FARO Laser Tracker ION

System Specifications

Dimensions
Head size: 311(W) x 556(H) mm
Head weight: 17.7kg (19.5kg w/IFM option)
Controller size: 282(L) x 158(D) x 214(H) mm
Controller weight: 5.2kg

Range
Horizontal envelope: +/- 270°
Vertical envelope: +75° to -50°
Minimum working range: 0 meters
Maximum working range: 55m with select targets
40m with standard 1.5" & 7/8" SMRs
30m with standard 1/2" SMR

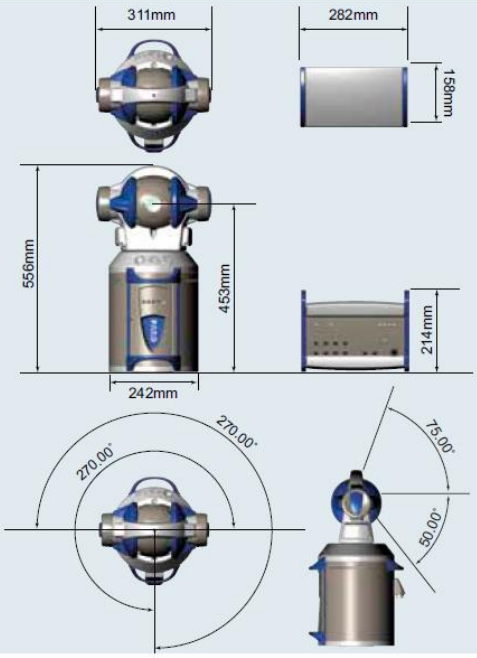
Environmental
Altitude: -700 to 2,450 meters
Humidity: 0 to 95% non-condensing
Operating Temperature: -15°C to 50°C

Laser Emission**
633-635 nm Laser, 1 milliwatt max/cw.
Class II Laser Product

Distance Measurement Performance***

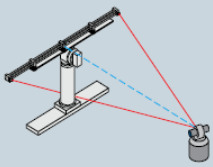
Agile ADM	Optional Interferometer
Resolution: 0.5µm	Resolution: 0.158µm
Sample rate: 10,000/sec	Accuracy: 2µm + 0.4µm/m
Accuracy: 8µm + 0.4µm/m	Maxim. radial velocity: 4m/sec
R0 Parameter: 8µm	R0 Parameter: 8µm

Angle Measurement Performance***
Angular accuracy: 10µm + 2.5µm/m
Maximum angular velocity: 180°/sec
Precision Level Accuracy: +/- 2 arcseconds

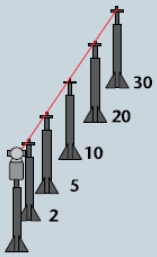


Point-to-Point Typical Accuracy***

Horizontal Scale Bar Measurement (2.3 m)		
Range (m)	ADM (mm)	IFM (mm)
2	0.022	0.021
5	0.032	0.032
10	0.049	0.049
20	0.085	0.085
30	0.120	0.120
40	0.156	0.156
50*	0.191	0.191
55*	0.209	0.209



In-Line Distance Measurement			
Length (m)	Distance (m)	ADM (mm)	IFM (mm)
2 - 5	3	0.009	0.003
2 - 10	8	0.011	0.005
2 - 20	18	0.015	0.009
2 - 30	28	0.019	0.013
2 - 40	38	0.023	0.017
2 - 50*	48	0.027	0.021
2 - 55*	53	0.029	0.023



** With selected targets.*
*** Product complies with radiation performance standards under the food, drug, and cosmetics act and international standard IEC 60825-1 2001-08.*
**** Typical Accuracy shown is half the Maximum Permissible Error (MPE) and variation in air temperature is not included. MPE and all accuracy specifications are calculated per ASME B89.4.19 - 2006 Standard.*

Specifications, descriptions, and technical data may be subject to change. **Protected by U.S. patents:** 7327446, 7352446, 7466401

19.3 Láser Tracker LEICA

Leica Absolute Tracker	AT 901-B	AT 901-MR	AT 901-LR
	Accuracy shown is the maximum permissible error (MPE). Typical results are half the MPE.		
Compatibility			
Compatible with Leica T-Cam	No	Yes	Yes
Compatible with Leica T-Probe	No	Yes	Yes
Compatible with Leica T-Scan	No	Yes	Yes
Compatible with Leica T-Mac	No	Yes	Yes
Measurement Volume			
Typical Volume (Ø)	160 m (525 ft)	50m (164 ft)	160 m (525 ft)
Horizontal	360°	360°	360°
Vertical	± 45°/360° *	± 45°/360° *	± 45°/360° *
			* in horizontal setup
Measuring and tracking performance			
Measuring rate	3,000 points per second		
Measuring rate output	1,000 points per second		
Tracking speed lateral	> 4 m/s (13 ft/s)		
Tracking speed radial	> 6 m/s (19 ft/s)		
Acceleration lateral	> 2 g		
Acceleration radial	unlimited		
Absolute Interferometer (AIFM)	**Yes**	**Yes**	**Yes**
Principle technology	Single Beam Heterodyne Interferometer with Polarization Modulation Absolute Reference		
Wavelength	633 nm / 795 nm (visible / IR)		
Warm-up time, cold start	8 min		
Warm-up time, warm start	5 min		
Largest Beam diameter	4 mm		
Interferometer Distance Resolution	0.32 µm (0.000013")		
Interferometer Distance Accuracy	± 0.5 µm/m (± 0.000006"/ft)		
Dynamic Lock-On Accuracy	± 10 µm (± 0.00039")		
Typical Lock-On working range	1.0 – 80.0 m (3.3 – 262 ft)	1.0 – 9.0 m (3.3 – 29 ft)	1.0 – 80.0 m (3.3 – 262 ft)
Laser Safety			
The Laser Tracking System is a Class 2 Laser Product	IEC 60825-1; Second Edition (2007-03)		
Accuracy information			
Angular resolution	0.14 arc sec		
Angular repeatability, full range	± 7.5 µm + 3 µm/m		
and in 2.5 x 5 x 10 m volume	(±0.0003" + 0.00004"/ft)		
Angle accuracy, full range	± 15 µm + 6 µm/m (±0.0006" + 0.00007"/ft)		
Angle accuracy in 2.5 x 5 x 10 m volume	± 10 µm + 5 µm/m (±0.0004" + 0.00006"/ft)		

Size and weight	
Sensor size	620 / 290 / 240 mm (24 / 11 / 9")
Sensor weight	22 kg (48.5 lbs)
Controller size	510 / 485 / 200 mm (20 / 19 / 7.9")
Controller weight	17 kg (37.5 lbs)

19.4 LEICA T-Probe

Leica T-Probe system accuracy

The measurement uncertainty of 3D points "U_{3d}" is defined as the distance between a measured point and the nominal position of that point. This measurement uncertainty is specified as a function of the distance between the laser tracker and the measured point.

The measurement uncertainty of spatial length "U_L" is defined as the deviation between a measured length and its nominal value. This measurement uncertainty is specified as a function of the shortest distance between the laser tracker and the measured length. The length can be up to 6 m and is positioned perpendicularly to the laser beam (Leica T-Probe held in constant orientation).

Measurement uncertainty of sphere radius "U_R" is defined as the deviation between a measured sphere radius and its nominal value. This specification assumes a reference sphere with a radius between 10 mm and 50 mm. This measurement uncertainty is specified as a function of the distance between the Laser Tracker and the measured sphere (Leica T-Probe held in constant orientation).

The uncertainty specified below is achieved with Leica T-Probe (110 mm stylus in mount 1/2) and a measurement mode of 1s per point under stable environmental conditions. Accuracy shown is the maximum permissible error (MPE). Typical results are half the MPE.

Measurement uncertainty of 3D Point (MPE)
U_{3D} = 100 µm if under 7 m (0.004" if under 23 ft)

U_{3D} = 30 µm + 10 µm/m if greater than 7 m
(0.0012" + 0.00012"/ft if greater than 23 ft)

Comfort through true intelligence

Feature	Benefit
Armless operations	Full "Walk-Around" freedom
Wireless operations	Speed and safety increase
No direct line of sight between probe tip and laser tracker required	Reaches hidden, recessed or hard-to-reach parts without needing to reposition laser tracker
Probe self-identification	Eliminates handling errors
Stylus self-identification	Eliminates handling errors
Acoustic system feedback in your hand	Accelerates measurement process
Visual system feedback in your hand	Accelerates measurement process
Precision quick release for stylus	Flexibility with speed
Wide variation of styli type and length	Highest probing flexibility
6DOF Dynamic Lock-On	No more waiting for the tracker to "lock on" – AIFM can instantly re-establish a broken laser beam.

Measurement uncertainty of spatial length (MPE)
U_L = ± 60 µm if under 8.5 m (± 0.0024" if under 27.9 ft)

U_L = ± 7 µm/m if greater than 8.5 m
(± 0.00008"/ft if greater than 27.9 ft)

Measurement uncertainty of sphere radius (MPE)
U_R = ± 20 µm + 2 µm/m (± 0.0008" + 0.00002"/ft)

Leica T-Probe	AT 901-MR	AT 901-LR
	Accuracy shown is the maximum permissible error (MPE). Typical results are half the MPE.	
Measurement volume		
Maximal volume (Ø)	18 m (59 ft)	30 m (98 ft)
Horizontal	360°	
Vertical	± 45°	
Acceptance angle		
(Freedom to rotate)		
Pitch angle	± 45°	
Jaw angle	± 45°	
Roll angle	360°, unlimited	
Measuring and tracking performance		
Measuring rate output	1,000 points per second	
Tracking speed all directions	> 1 m/s (≈ 3.3 ft/s)	
Acceleration, all directions	1 g	
Weight		
Leica T-Probe with standard tip and battery	670 g (≈ 1.48 lb)	
Leica T-Probe with standard tip and without battery	570 g (≈ 1.26 lb)	
Leica T-Cam MR and LR	4.7 kg (≈ 10.36 lb)	

19.5 LEICA T-Mac system

Leica T-Mac system accuracy

Accuracy of rotation angles
0.01° = 18 µm/100mm (0.002"/ft)

Accuray of time stamp
‹5 µs

Positioning accuracy for typical robotic drilling applications
50 µm (0.002")

Positional accuracy
± 15 µm + 6 µm/m (±0.0006" + 0.00007"/ft)

Accuracy shown is the maximum permissible error (MPE). Typical results are half the MPE.

Comfort through true intelligence

Feature	Benefit
Sealed housing	Work in toughest industrial environments
Robust design, with no internal moving parts or mechanisms	Designed for use in real production environments, easy to service and with no "wear and tear"
Simple mechanical interface	Direct mounting on a robot or a machine
Adaptation of tool exchange interface	Automatic connection to robot with high repeatability of under 3 µm (0.00012")
Multiple reflector nests on Leica T-Mac housing	Calibrated reflector locations as known home points for easy establishing of object orientation
6DOF Dynamic Lock-On	No more waiting for the tracker to "lock on" – AIFM can instantly re-establish a broken laser beam.

Leica T-Mac	AT 901-MR	AT 901-LR
Accuracy shown is the maximum permissible error (MPE). Typical results are half the MPE.		
Measurement volume		
Maximal volume	18 m (59 ft)	30 m (98 ft)
Horizontal	360°	
Vertical	± 45°	
Acceptance angle (Freedom to rotate)		
Pitch angle	± 45°	
Jaw angle	± 45°	
Roll angle	360°, unlimited	
Measuring and tracking performance		
Measuring rate output	1,000 points per second	
Tracking speed all directions	> 1 m/s (≈ 3.3 ft/s)	
Acceleration all directions	1 g	
Accuracy		
Accuracy of rotation angles	0.01° = 18 µm/100mm (0.002"/ft)	
Accuray of time stamp	‹5 µs	
Positional accuracy	± 15 µm + 6 µm/m (±0.0006" + 0.00007"/ft)	
Weight		
Leica T-Mac	1,480 g (≈ 3.26 lb)	

Leica Geosystems

Dispositivo de seguimiento 6GdL para aplicaciones automatizadas

El Leica T-Mac (sensor de control de máquinas tracker), la última generación del dispositivo de seguimiento 6GdL para aplicaciones automatizadas, da respuesta a las necesidades de un número de clientes de Leica Geosystems cada vez mayor que o bien han modificado el Leica T-Probe existente para aplicaciones de medición automatizada o bien han expresado interés en hacerlo.

Leica T-Mac (sensor de control de máquina Tracker)

Descripción | **Ventajas y Características** | Aplicaciones | Descargas

Especificaciones de funcionamiento e información de precisión, incluyendo el rango máximo de 30 m (98 ft) que corresponde al del Leica T-Probe.

- PowerLock: Por primera vez en la historia del Laser Tracker, el rayo láser se mueve hacia el usuario y no al contrario.
- Carcasa especial para sencilla instalación en máquinas o robots
- Nidos de reflector para calibración y orientación sencilla
- Interfaz mecánica para herramienta de calibración
- Interfaz para unidad precisa de cambiador de herramientas
- El mayor volumen de medición de hasta 30 m (98 ft)
- Ángulo de aceptación más ancho que cualquier sistema comparable: ángulo de inclinación ±45°, guiñada ±45°, balanceo 360°
- Feedback de estado encendido (ON), a la vista y a distancia, así como para información de proceso
- Feedback acústico de información de proceso
- Insensible a la luz medioambiental
- Cadencia de medición de 1.000 puntos por segundo
- Velocidad de seguimiento > 1 m/s
- Precisión de medición de longitud de 2 sigma de 60 µ en un volumen de medición de 15 m (49 ft)

19.6 LEICA Estación Láser TDRA6000

Precisiones de la estación láser Leica TDRA6000

Todas las mediciones se realizan bajo condiciones de entorno estables en interior y con el objetivo debidamente alineado. Las mediciones de distancia se verifican con un interferómetro láser en 120 m. Si no se indica lo contrario, todas las precisiones son de error máximo admisible MPE (≈ ±3 σ), con precisiones típicas que son la ½ del MPE (o aproximadamente ±1.5 σ).

Precisión de punto 3D

La incertidumbre de medición de una coordenada "U_{XYZ}" se define como la desviación entre una coordenada medida y la coordenada nominal de ese punto. Esta incertidumbre de medición se especifica como una función de la distancia entre la estación láser y el punto medido. La precisión especificada abajo se logra con los reflectores de anillo rojo de 1,5" de Leica Geosystems 1.5" y el modo de medición "Preciso". La especificaciones se indican en error máximo admisible (MPE). Los resultados típicos son la ½ del MPE.

$U_{XYZ} \leq$ 30 m de volumen	U_{XYZ} > 30 m de volumen
± 0,5 mm	± 0,3 mm + 13 µm/m

Rendimiento de PowerSearch

Rendimiento de PowerSearch con BRR y RRR de 1,5"

Volumen máx.	100 m
Distancia mín.	2,0 m

Reproducibilidad de punto 3D

La reproducibilidad de punto 3D "PRE" se define como la desviación estándar entre mediciones múltiples de un punto 3D y se especifica como una función de la distancia entre la estación láser y el punto medido. La precisión especificada abajo se logra con reflectores TBR y/o RFI de 0,5" de Leica Geosystems utilizando el modo de medición "Preciso".

$P_{RE} \leq$ 30 m de volumen	± 0,2 mm

Rendimiento EDM

Rendimiento EDM	*1.5" RRR & BRR*[2]	*0.5" TBR & RFI*[2]	*Cinta reflectante*[3]	*Sin contacto (RL)*[4]
Volumen máx	600 m	600 m	600 m	600 m
Distancia mín	2,0 m	2,0 m	5,0 m	2,0 m
Precisión (MPE)[1]	0,5 mm	0,5 mm	1,0 mm	2,0 mm

[1] *La precisión indicada es el error máximo admisible (MPE). Los resultados típicos son la 1/2 del MPE.*
[2] *El RRR de 1,5" se mide en el rail láser como estándar.*
[3] *20 mm x 20 mm objetivo de 5 m a 60 m.*
[4] *Las mediciones RL se realizan con el riel láser de 2 a 60 m.*

Rendimiento ATR

Rendimiento ATR	*1.5" RRR & BRR*	*0.5" TBR & RFI*[2]	*Dianas*
Volumen máx	600 m	400 m	40 m
Distance mín.	2,0 m	2,0 m	5,0 m
Precisión (MPE)[1]	± 0,3 mm + 13 µm/m	N/A	± 2,0 mm

19.7 LEICA TCA2003 Total Station

Angle measurement

		TCA1800	TCA2003	TC2003
Accuracy (standard deviation, ISO 17123-3)	Hz, V	1" (0.3 mgon)	0.5" (0.15 mgon)	0.5" (0,15 mgon)
	Display resolution	1" (0.1 mgon)	0.1" (0.01 mgon)	0.1" (0.01 mgon)
	Method:		absolute, continuous, diametrical	
Compensator	Working range:	4' (0.07 gon)	4' (0.07 gon)	4' (0.07 gon)
	Setting accuracy:	0.3" (0.1 mgon)	0.3" (0.1 mgon)	0.3" (0.1 mgon)
	Method:		electronic dual axis compensator	

Distance measurement (IR)

Range (average atmospheric conditions)	Round prism (GPR1):		2500 m	
	360° reflector (GRZ4):		1300 m	
	Mini prism (GMP101):		900 m	
	Reflective tape (60 mm x 60 mm)		200 m	
Accuracy / measurement time (standard deviation, ISO 17123-4)	Standard mode:	1 mm + 2 ppm / 3.0 s	1 mm + 1 ppm / 3.0 s	1 mm + 1 ppm / 3.0s
	Fast mode:		3 mm + 2 ppm / 1.5 s	
	Tracking mode:		5 mm + 2 ppm / 0.3 s	
	Display resolution:	0.1 mm	0.01 mm	0.01 mm
	Method:		Phase measurement (coaxial, invisible infrared laser)	

Guide light (EGL)

Range (average atmospheric conditions)	Working range:	5 m - 150 m
Accuracy	Positioning accuracy:	5 cm at 100 m

Motorized

Maximum speed	Rotating speed:	45° / s

Automatic target recognition (ATR)

Range ATR mode / LOCK mode (average atmospheric conditions)	Round prism (GPR1):	1000 m / 500 m
	360° reflector (GRZ4):	500 m / 350 m
	Shortest measurable distance:	5 m
Accuracy / measurement time	Up to 200 m: 1 mm, > 200 m: as per angle measuring accuracy +1 mm / 3-4 s	
Max. speed (LOCK-Modus)	Tangential (standard mode):	5 m / s at 100 m, 1 m / s at 20 m
	Tangential (with EDM tracking mode):	1 m / s at 100 m, 0.2 m / s at 20 m
	Method:	Digital image processing (laser beam)

19.8 NIKON Indoor GPS

iGPS - Large scale metrology

Large volume tracking and assembly

The iGPS is a modular, large volume tracking system enabling factory-wide localization of multiple objects with metrology accuracy, applicable in manufacturing and assembly. iGPS is mainly used by aerospace manufacturers, but is also adopted by automotive and industrial manufacturers both for positioning and tracking applications.

Key features

- Integrated with 3rd Party software such as Spatial Analyzer (SA)
- Transmitters expandable from 2 m to 300m volume size
- Indoor/Outdoor GPS application
- Fixed and portable receivers (vector bar)
- Unlimited number of client setups

Configurations and specifications

iSpace setups, configurations, tools and options

	4 series	6 series	8 series
Transmitter count	4	6	8
Available configurations	x, s	x, s, i	x, s, i
Typical measurement volume footprint	from 10x10m up to 40x40m		
Typical use case	Smaller volume / Unobstructed line of sight / Lower accuracy	General purpose / Standard system size / Moderate line of sight obstructions	Larger volumes / Obstructed line of sight / Maximum accuracy

	x	s	i
Hardware configuration	Free network of transmitters	Transmitters plus base stations	Transmitters plus monuments
Calibration method	Traceable scale-bar	Traceable scale-bar	Traceable reference network
Accuracy[1,2]	<250 µm over 3x scale-bar length[3]	<250 µm over 3x scale-bar length	<200 µm uniformly over monument volume
Typical use case	Portable/Deployable setup Short term measurement and tracking Engineering or R&D focused Less health monitoring	Semi-portable or semi-fixed installations Longer term measurement and tracking Production focused High level of health monitoring	Fixed installation Always-on production system Maximum system robustness accuracy and reliability

[1] For optimum performance, it is recommended to keep the distance between sensors and transmitters between 3 and 35m and guarantee individual sensors to be within line of sight of at least 4 transmitters.
[2] System performance testing according to B89 and ISO10360 in conjunction with an independent accredited measurement standards lab. iSpace is a network based metrology system that does not follow traditional uncertainty models. Coordinate uncertainties are interdependent, do not scale with range and are repeatable within the volume.
[3] Requires stable transmitter positions and environmental conditions

	DGA requirements	Metris iGPS characteristics (6 transmitter and 4 receiver setup)
Measurement range	70 meter	Unlimited, as more transmitters can be added
Motion tracking and measurement	Yes	Yes
Indoor and outdoor use	Yes	Yes
Real-time process	Yes	Yes
Synchronization	< 50 milliseconds	Approx. 25 milliseconds
Static positional accuracy	< 1 millimeter	0.3 millimeter
Dynamic positional accuracy	< 5 millimeter	1 millimeter (speed-dependent)
Sensor information	5DOF (2 x 3DOF synchronized)	6DOF (4 x 3DOF synchronized) : x, y, z, r_x, r_y, r_z
Data sampling rate	> 20 Hertz	Approx. 30 Hertz
Horizontality	0.05 degrees	0.00001 degrees
Line of sight	Not sensitive	Not sensitive

Table : Metris iGPS system characteristics versus DGA system requirements