

ALGORITMOS DE AJUSTE DE CURVAS PARA MÁQUINAS DE MEDICIÓN POR COORDENADAS

carlos galván
Centro Nacional de Metrología
Apdo. Postal 1-100 CP 7600 Querétaro, México
cgalvan@cenam.mx

Resumen: Durante mucho tiempo el software de las Máquinas de Medición por Coordenadas (MMC), se pensó que no proporcionaba errores en el cálculo de las geometrías: En estudios recientes se demostró que existe una desviación considerable entre los resultados arrojados por algunos de los softwares comerciales. Este conocimiento provocó que se hiciera una estandarización de las ecuaciones para la determinación de las desviaciones entre un punto en el espacio y la geometría ajustada. En el presente trabajo se describe una estrategia empleada para resolver el problema de ajuste empleando el criterio de mínimos cuadrados. Además, de la disminución del error de estimación en comparación con algunos softwares de MMC, se mejora la técnica de muestreo, al permitir que geometrías como el cono o el cilindro, restringidos en algunos softwares, sean mas abiertos para su estimación.

INTRODUCCIÓN

El software matemático, particularmente las rutinas de ajuste para curvas y superficies, es un componente crítico en máquinas de medición por coordenadas (MMC). Una MMC solo es capaz de entregar puntos en tres dimensiones con respecto a un sistema que en ella se establece, el software es el encargado de efectuar el cálculo de los parámetros de ajuste para cada una de las geometrías que el usuario desea parametrizar.

Debido a lo anterior cada diseñador de software establece las rutinas de ajuste, parametrización de geometrías, etc; de ahí el error en algunos de ellos. Existe una norma británica que contiene la parametrización de geometrías para MMC[1], así como las ecuaciones de distancia entre un punto y estas geometrías, pues a partir de estas ecuaciones se debe de encontrar los parámetros de la geometría que mejor ajuste a la forma deseada, a partir de los puntos proporcionados por la MMC.

Este reporte describe algoritmos para encontrar el ajuste de las curvas y superficies que son obtenidas en una MMC, basándose en el criterio de mínimos cuadrados y empleando las fórmulas estandarizadas por [1].

El trabajo esta organizado como sigue. La segunda sección describe los algoritmos empleados para el ajuste de geometrías lineales (planos y líneas). La siguiente sección contienen los algoritmos para las superficies no lineales, para resolverlos se utilizan algoritmos de minimización sin restricciones los cuales emplean las derivadas de las funciones de distancia. En la última sección se describen los resultados del trabajo. Al final se incluye el apéndice

A donde se describen los algoritmos de minimización no lineales empleados en la soluciones planteadas, estos algoritmos se encontraron que funcionan correctamente, aun con datos bastante perturbados, pero no son los únicos en la literatura, estos representan solo una opción que se evaluó como adecuada.

En los algoritmos se hace el supuesto de que los puntos son trasladados usando como pivote el centroide de los puntos originales, logrando con esto que el nuevo centroide sea el origen del sistema de coordenadas

GEOMETRIAS LINEALES

Las geometrías lineales (líneas y planos) se resuelven empleando multiplicadores de Lagrange sobre un problema de optimización con restricciones[2]. Ambos casos se reducen a un problema simple de valores propios (eigen values) o descomposición de valores singulares.

Ajuste de plano

Parámetros que definen un plano:

- x** Un punto sobre el plano
- a** Los cosenos directores de la normal al plano

Ecuación de distancia de un punto a un plano:

$$d_i = d(\mathbf{x}_i) = d(\mathbf{x}_i, \mathbf{x}, \mathbf{a}) = \mathbf{a} \cdot (\mathbf{x}_i - \mathbf{x})$$

Función objetivo: $J(\mathbf{x}, \mathbf{a}) = \sum [\mathbf{a} \cdot (\mathbf{x}_i - \mathbf{x})]^2$

Descripción del método:

El centroide de los datos cae en el plano de mínimos cuadrados. Esto se puede mostrar [2] porque $\nabla J = 0$ (el gradiente de la función), cuando se encuentra la solución de mínimos cuadrados, con lo

Cual se puede decir $\sum [\mathbf{a} \cdot (\mathbf{x}_i - \mathbf{x})] = 0$, si se

multiplica esta ecuación por $1/N$ esto nos da

$$\frac{a}{N} \sum (x_i - x) + \frac{b}{N} \sum (y_i - y) + \frac{c}{N} \sum (z_i - z) = 0$$

si se distribuyen las sumas de esta ecuación se puede reducir la ecuación anterior a $a(\bar{x} - x) + b(\bar{y} - y) + c(\bar{z} - z) = 0$, que no es mas que la distancia del centroide al plano y como es igual a cero, se concluye que el centroide esta en el plano ajustado. Dado que se estableció desde un principio que los datos iban a ser trasladados para que el centroide coincidiera con el origen, se puede establecer entonces que $\mathbf{x} = 0$.

La dirección del plano ajustado \mathbf{a} , puede ser encontrada planteando el problema como minimización con restricciones, esto es, minimizando J sujeto a la restricción que $|\mathbf{a}| = 1$. Se puede entonces definir una función $G = |\mathbf{a}|^2 - 1$, así el problema es minimizar J sujeto a la restricción de que $G = 0$. Utilizando el método de multiplicadores de Lagrange este nos dice que un mínimo ocurre en punto donde $\nabla J = \mathbf{I} \nabla G$, para algún número real \mathbf{I} (con esto a, b y c son tratados como variables independientes, debido a que la restricción esta acotada por G). Así $\nabla G = 2\mathbf{a}$ y $\nabla J = 2(\mathbf{M}^T \mathbf{M})\mathbf{a}$, considerando que la matriz \mathbf{M} es una matriz de $N \times 3$ que contiene los puntos en cuestión, así pues substituyendo esto en la consideración de Lagrange nos conduce a un problema de valores propios $(\mathbf{M}^T \mathbf{M})\mathbf{a} = \mathbf{I}\mathbf{a}$.

Este es un problema de valores propios de 3×3 el cual puede ser resuelto eficientemente. Sin embargo, este problema se puede ver también como una descomposición de valores singulares (SVD) pues los valores propios de $\mathbf{M}^T \mathbf{M}$ son también los valores singulares de \mathbf{M} . Esto permite tener mas estabilidad numérica aplicando SVD a \mathbf{M} sin calcular $\mathbf{M}^T \mathbf{M}$.

Lo anterior produce 3 valores singulares de los cuales hay que escoger el que indique cual es el vector que soluciona el problema dado. Las ecuaciones siguientes describen el problema original:

$$\begin{aligned} \sum x_i (\mathbf{a} \cdot \mathbf{x}_i) &= \mathbf{I}a & \sum y_i (\mathbf{a} \cdot \mathbf{x}_i) &= \mathbf{I}b \\ \sum z_i (\mathbf{a} \cdot \mathbf{x}_i) &= \mathbf{I}c \end{aligned}$$

Si se multiplican estas ecuaciones por a, b, c respectivamente y sumándolas se obtiene que $\sum (\mathbf{a} \cdot \mathbf{x})^2 = \mathbf{I}|\mathbf{a}|^2 = \mathbf{I}$. Como se puede ver la suma del lado izquierdo de esta ecuación no es mas que la función objetivo inicial por tanto, \mathbf{I} es igual a la función objetivo, de ahí que el vector que contiene la solución de mínimos cuadrados es el que posee el menor valor singular o valor propio (según sea el algoritmo empleado).

Ajuste de línea

Parámetros que definen una línea:

- \mathbf{x} Un punto sobre la línea
- \mathbf{a} Los cosenos directores de la línea

Ecuación de la distancia:

$$d_i = d(\mathbf{x}_i) = d(\mathbf{x}_i, \mathbf{x}, \mathbf{a}) = |\mathbf{a} \times (\mathbf{x}_i - \mathbf{x})|$$

donde \times representa el producto cruz de los vectores. También puede ser expresado como

$$\sqrt{|\mathbf{x}_i - \mathbf{x}| \cdot |\mathbf{a} \cdot (\mathbf{x}_i - \mathbf{x})|^2}$$

$$J(\mathbf{x}, \mathbf{a}) = \sum |\mathbf{a} \times (\mathbf{x}_i - \mathbf{x})|^2$$

Descripción del método:

Al igual que en el plano, y en todas las geometrías, los puntos se trasladan tal que su centroide sea el origen, con lo anterior se puede mostrar que el centroide es un punto sobre la línea, así como se hizo con el plano [2].

La dirección \mathbf{a} puede ser encontrada siguiendo la misma estrategia que en el plano, para el ajuste de la línea las ecuaciones son similares a $(\mathbf{M}^T \mathbf{M})\mathbf{a} = \mathbf{I}\mathbf{a}$, una vez más el valor singular correcto debe ser escogido tal que minimice la suma de cuadrados de los residuales. Así como en el plano se mostró que

$$\sum (\mathbf{a} \cdot \mathbf{x}_i)^2 = \mathbf{I}, \text{ así pues } J = -\mathbf{I} + \sum |x_i|^2 \text{ es para}$$

este caso la función, lo cual conduce a que para minimizar J , se debe maximizar \mathbf{I} , así pues se escoge el valor propio más grande.

GEOMETRIAS NO LINEALES

Las geometrías que se han denominado no lineales en este reporte corresponde al círculo, esfera, cono y cilindro.

Para el cálculo de las distancias de las geometrías no lineales son importantes las ecuaciones de las distancias de línea y plano a un punto. El algoritmo empleado es Gauss Newton, el cual funciona en minimización de funciones sin restricciones, para eliminar las restricciones de los cosenos directores de las diferentes geometrías se puede trabajar en términos de los números directores expresados como $\mathbf{A} = (A, B, C)$, donde $\mathbf{a} = \mathbf{A} / |\mathbf{A}|$. El algoritmo aplicado debe normalizar \mathbf{A} en cada iteración, de tal forma que siempre se trabaje con el vector unitario (cosenos directores) pero sin restringirlos a esos valores.

La ecuación $p(\mathbf{x}_i, \mathbf{x}, \mathbf{A})$ describe la distancia del punto \mathbf{x}_i a la línea con parámetros \mathbf{x} y \mathbf{A} . El valor de la función p está dado por

$$p_i = p(\mathbf{x}_i, \mathbf{x}, \mathbf{A}) = \mathbf{a} \cdot (\mathbf{x}_i - \mathbf{x}) + b(y_i - y) + c(z_i - z).$$

La distancia entre una línea y un punto está dada por $l(\mathbf{x}_i, \mathbf{x}, \mathbf{A}) = |\mathbf{a} \times (\mathbf{x}_i - \mathbf{x})|$, que puede ser expresado como

$$l_i = \sqrt{u^2 + v^2 + w^2} \quad \text{donde} \quad u = c(y_i - y) - b(z_i - z),$$

$$v = a(z_i - z) - c(x_i - x) \quad \text{y} \quad w = b(x_i - x) - a(y_i - y) \text{ según [1].}$$

Las expresiones de las derivadas, que se emplearán en las diferentes geometrías no lineales se pueden expresar fácilmente.

$$\frac{\partial p}{\partial x} = -a; \quad \frac{\partial p}{\partial y} = -b; \quad \frac{\partial p}{\partial z} = -c$$

$$\frac{\partial p}{\partial A} = (x_i - x) - ap_i; \quad \frac{\partial p}{\partial B} = (y_i - y) - bp_i; \quad \frac{\partial p}{\partial C} = (z_i - z) - cp_i$$

$$\frac{\partial l}{\partial x} = [ap_i - (x_i - x)]/l_i; \quad \frac{\partial l}{\partial y} = [bp_i - (y_i - y)]/l_i$$

$$\frac{\partial l}{\partial z} = [cp_i - (z_i - z)]/l_i; \quad \frac{\partial l}{\partial A} = p_i [ap_i - (x_i - x)]/l_i$$

$$\frac{\partial l}{\partial B} = p_i [bp_i - (y_i - y)]/l_i; \quad \frac{\partial l}{\partial C} = p_i [cp_i - (z_i - z)]/l_i$$

Para el caso del cilindro y cono, la línea asociada con las fórmulas de distancia es el eje de la geometría. Para los conos, el plano asociado con las fórmulas es el plano que pasa por el punto \mathbf{x} y es perpendicular a su eje.

Ajuste de una esfera

Parámetros que definen una esfera:

- \mathbf{x} El centro de la esfera
- r El radio de la esfera

Ecuación de la distancia:

$$d_i = d(\mathbf{x}_i) = d(\mathbf{x}_i, \mathbf{x}, r) = |\mathbf{x}_i - \mathbf{x}| - r$$

Función objetivo: $J(\mathbf{x}, \mathbf{a}) = \sum (|\mathbf{x}_i - \mathbf{x}| - r)^2$

Descripción:

La ecuación de la distancia de un punto a la esfera es una función no lineal, por lo cual es necesario emplear algún algoritmo de minimización no lineal para la solución del problema de ajuste de puntos a la geometría, en este reporte se usa el algoritmo de Gauss-Newton. Este algoritmo, como se describe en el apéndice A, necesita una aproximación a los parámetros reales que se encuentre cercana la verdadera solución

En la iteración del algoritmo de Gauss-Newton se necesita la matriz jacobiana (**G**) y el vector de residuos los cuales se pueden expresar como :

$$G = \begin{bmatrix} \frac{\partial l_i}{\partial x} & \frac{\partial l_i}{\partial y} & \frac{\partial l_i}{\partial z} \\ \dots\dots\dots \end{bmatrix} \quad r_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2}$$

$$\frac{\partial l_i}{\partial x} = \left(\frac{x_i - x}{r_i} \right) \quad \frac{\partial l_i}{\partial y} = \left(\frac{y_i - y}{r_i} \right) \quad \frac{\partial l_i}{\partial z} = \left(\frac{z_i - z}{r_i} \right)$$

$$\frac{\partial l_i}{\partial r} = -1 \quad d_i = r_i - r$$

Ajuste de un cilindro

Parámetros que definen un cilindro:

- x** Un punto sobre el eje del cilindro
- r** El radio del cilindro
- a** Los cosenos directores apuntando en la dirección del eje

Ecuación de la distancia: $d_i = d(\mathbf{x}_i, \mathbf{x}, \mathbf{a}, r) = l_i - r$

Función objetivo: $J = \sum (l_i - r)^2$

Descripción del método:

La función de la distancia de un punto a un cilindro no es mas que la distancia del punto al eje del cilindro menos el radio del cilindro.

Para resolver el problema del cilindro, se hace uso una vez mas del método de Gauss-Newton, lo cual requiere la matriz Jacobiana, así como el vector de residuos los cuales se pueden derivar, quedando de la siguiente manera:

$$G = \begin{bmatrix} \frac{\partial d_1}{\partial x} & \frac{\partial d_1}{\partial y} & \frac{\partial d_1}{\partial z} & \frac{\partial d_1}{\partial A} & \frac{\partial d_1}{\partial B} & \frac{\partial d_1}{\partial C} & \frac{\partial d_1}{\partial r} \\ \dots\dots\dots \end{bmatrix}$$

Las derivadas de la distancia no son mas que las derivadas de la fórmula de la distancia de un punto a una línea (el eje del cilindro) y solo falta especificar la derivada con respecto al radio la cual es muy simple

$$\frac{\partial d}{\partial r} = -1$$

Al igual que en el caso de la esfera, es necesario dar una aproximación para el método de Gauss-Newton, esto se puede lograr mediante una aproximación por mínimos cuadrados lineales.

Ajuste de un círculo en el espacio

Parámetros que definen un círculo:

- x** El centro del círculo
- r** El radio del círculo
- a** Los cosenos directores de la normal al círculo

Ecuación de la distancia: $d(\mathbf{x}, \mathbf{a}, \mathbf{r}) = \sqrt{p_i^2 + (l_i - r)^2}$

Función objetivo: $J(\mathbf{x}, \mathbf{a}, \mathbf{r}) = \sum (p_i^2 + (l_i - r)^2)$

Descripción del método:

Esta fórmula esta basada también en distancias de punto a plano, y de punto a recta.

Para encontrar el círculo se hace uso del método Gauss-Newton, las derivadas que conforman la Jacobiana para este modelo son:

$$\frac{\partial d}{\partial x} = [p_i \partial p_i / \partial x + (l_i - 1) \partial l_i / \partial x] / d_i \quad \frac{\partial d}{\partial y} = [p_i \partial p_i / \partial y + (l_i - 1) \partial l_i / \partial y] / d_i$$

$$\frac{\partial d}{\partial z} = [p_i \partial p_i / \partial z + (l_i - 1) \partial l_i / \partial z] / d_i \quad \frac{\partial d}{\partial A} = [p_i \partial p_i / \partial A + (l_i - 1) \partial l_i / \partial A] / d_i$$

$$\frac{\partial d}{\partial B} = [p_i \partial p_i / \partial B + (l_i - 1) \partial l_i / \partial B] / d_i \quad \frac{\partial d}{\partial C} = [p_i \partial p_i / \partial C + (l_i - 1) \partial l_i / \partial C] / d_i$$

$$\frac{\partial d}{\partial z} = -(f_i - r)/d_i$$

el problema del círculo es algo complicado por ello se sugiere la siguiente estrategia para lograr una rápida convergencia del método:

1. Calcular el plano de mínimos cuadrados que determinan los puntos bajo análisis
2. Rotar los datos de tal forma que el plano de mínimos cuadrados sea paralelo al plano xy. Esto se puede lograr formando una matriz de rotación, con el vector normal al plano calculado anteriormente de tal forma que la normal sea paralela al eje z. Los vectores restantes pueden estimarse como el producto cruz del vector normal con otro vector, posiblemente uno a lo largo del eje x (1,0,0), en tanto; el tercer vector se puede determinar con el producto cruz de los dos anteriores. El usuario debe verificar que alguno de los vectores resultantes no coincida con el eje z.
3. Proyectar los datos rotados sobre el plano xy.
4. Calcular un ajuste de círculo en 2D de acuerdo a la sección anterior
5. Regresar a la orientación original, se puede aplicar las transformaciones inversas a los parámetros encontrados con el fin de tener la estimación en el sistema original
6. Realizar la minimización completa en 3D a partir de esta primera estimación.

Ajuste de cono

Parámetros que definen un cono:

- x** Un punto sobre el eje del cono
- a** Los cosenos directores del eje del cono
- j** La mitad del ángulo del cono
- s** Distancia del punto sobre el eje a la superficie del cono

Ecuación de distancia: $d(\mathbf{x}, \mathbf{a}, \mathbf{r}) = l_i \cos(\mathbf{j}) + p_i \sin(\mathbf{j}) - s$

Función objetivo: $J(\mathbf{x}, \mathbf{a}, \mathbf{r}) = \sum (l_i \cos(\mathbf{j}) + p_i \sin(\mathbf{j}) - s)^2$

Descripción del método:

El algoritmo empleado para encontrar el mejor ajuste a un cono, varía un poco en relación a los métodos

presentados para otras geometrías, debido a su complejidad se intentaron varios métodos antes de encontrar el mas conveniente. Se encontró que la mejor aproximación se hallaba usando el método de Levenberg-Marquardt descrito en el apéndice A. Además, el método para encontrar la primera aproximación es mas complejo que los anteriores para otras geometrías y de eso depende encontrar rápido la mejor aproximación.

El algoritmo de Levenberg-Marquardt al igual que el de Gauss-Newton requieren la matriz Jacobiana y por tanto las derivadas de la expresión de distancia

$$\frac{\partial d}{\partial x} = \partial l / \partial x \cos y + \partial p / \partial x \sin y \quad \frac{\partial d}{\partial y} = \partial l / \partial y \cos y + \partial p / \partial y \sin y$$

$$\frac{\partial d}{\partial z} = \partial l / \partial z \cos y + \partial p / \partial z \sin y \quad \frac{\partial d}{\partial A} = \partial l / \partial A \cos y + \partial p / \partial A \sin y$$

$$\frac{\partial d}{\partial B} = \partial l / \partial B \cos y + \partial p / \partial B \sin y \quad \frac{\partial d}{\partial C} = \partial l / \partial C \cos y + \partial p / \partial C \sin y$$

$$\frac{\partial d}{\partial s} = -1$$

$$\frac{\partial d}{\partial y} = -l \sin y + p \cos y$$

RESULTADOS

Este proyecto de investigación concluyó en un software que fue elaborado en el lenguaje de programación C++, que lee archivos de datos en formato de texto para hacer el ajuste que el usuario solicite. Para mayores detalles del programa el lector puede hacer referencia al manual de usuario del programa [7].

El software en cuestión, fue elaborado en el compilador de C++, Builder de Borland, version 3.0. Con este software es posible leer archivos en formato texto, donde se detallan las coordenadas de los puntos que forman la geometría a parametrizar. Contiene un menú donde el usuario puede elegir la geometría que se desea ajustar. Además, el usuario puede visualizar los puntos leídos desde el archivo en una pantalla gráfica, la cual el usuario puede rotar, trasladar o cambiar de escala por medio de acciones con el mouse de la computadora. Después de hacer el ajuste solicitado, el programa muestra los resultados para la geometría seleccionada, además de que se realiza la graficación de la geometría en la pantalla, junto con los puntos para que el usuario distinga la orientación de la geometría calculada, por último, se puede ver la calidad del ajuste viendo el error cuadrático medio con el ajuste encontrado.

El programa mencionado fue verificado mediante conjuntos de datos, que fueron generados aleatoriamente introduciéndoles un tipo de error, cosenoidal, sistemático y/o aleatorio, que perturban los datos originales que conforman una geometría teórica ideal, los resultados obtenidos son bastante satisfactorios mejorando en todos los casos los resultados producidos por el software de una MMC comercial.

Además de la mejora en error de ajuste por estos algoritmos, estos permiten mejorar la técnica de muestreo de la pieza a medir. El software de la MMC empleada, por ejemplo, no permite la medición de un círculo en tres dimensiones, lo cual con estos algoritmos es algo factible. Por otro lado el muestreo de cilindro y cono es mas fácil con estos algoritmos pues no es necesario hacer el muestreo de círculos concéntricos como lo exige el software de CENAM para estos casos.

APÉNDICE A: MÉTODOS DE SOLUCIÓN

El algoritmo de Gauss-Newton se basa en el algoritmo de Newton para la determinación de las raíces de un función[3], en este algoritmo se supone que se tiene un estimado del parámetro u_0 donde la función tiene una raíz, entonces $u_1 = u_0 + p$ donde $p = -f(u_0)/f'(u_0)$, se usa este valor como la nueva estimación donde f cruza por eje. El Gauss-Newton sigue la misma idea para encontrar el mínimo de una suma de cuadrados. Supone que se tiene algún estimado u de la solución u^* . Se resuelve el sistema de mínimos cuadrados lineales de la forma $Gp = -d$, donde G es la matriz Jacobiana de $m \times n$ cuyo i -ésimo renglón es el gradiente de d_i con respecto a los parámetros de u , esto es $G_{ij} = \partial d_i / \partial u_j$, evaluada en u , en tanto el i -ésimo componente de d es el residuo de la distancia entre el punto i y la geometría con los parámetros u , $d_i(u)$. Finalmente se actualiza la estimación de los parámetros de solución haciendo $u = u + p$. Estos pasos son repetidos hasta que se estime que se ha convergido a la solución u^* .

El criterio de convergencia de los métodos iterativos en minimización de funciones pueden ser, un pequeño cambio en las nuevas soluciones, o una magnitud pequeña en el vector de actualización.

El algoritmo de Gauss-Newton presenta problemas de convergencia cuando la aproximación inicial no esta en la vecindad de la real. Además, como en el caso del cono, la matriz G puede llevar a una matriz no definida positiva en la solución del problema de mínimos cuadrados lineales, el cual no es posible solucionarlo directamente. Caso que prevee el método de Levenberg-Marquardt[4] mediante la suma de una constante multiplicada por la matriz identidad a la matriz cuadrada (ya en la solución de mínimos cuadrados), provocando que la matriz se vuelva diagonal dominante y buscando así que se vuelva definida positiva. Esta consideración pretende detener la introducción de la matriz hessiana (segundas derivadas) y de esta manera evitar la introducción de los datos originales, lo cual según [3] y [4] provoca desviaciones en la estimación de la solución verdadera.

REFERENCIAS

- [1] "Assesment of position, size and departure from nominal form goemetric features", British Standard Guide, BS7172:1989.
- [2] Shakarji, C. "Least-Squares Fitting Algorithms of the NIST Algorithm Testing System", Journal of Research of NIST, Vol 103, No. 6, 1998
- [3] Golub, H.G. "Matrix Computations", The Johns Hopkins University Press, Baltimore, MD, 1983
- [4] Nash, J.C "Compact Numeriacal Methods for Computers: Linear Algebra and Function Minimization", Adam Hilger, Ltd., Bristol, England, 1979
- [5] Diaz C, Hopp T., "Testing Coordinate Measuring Systems Software". Proceedings of ASQC Measurement Quality Conference, October 26-27, National Institute of Standards and Technology.
- [6] Diaz C., Hopp T, "Special Test Service: The Algorithm Testing and Evaluation Program for Coordinate Measuring Systems", NISTSP 250-41, National Institute of Standards and Technology, Gaithersburg MD.
- [7] Galván, C."Manual de usuario de Pajusta", Reporte de Proyecto Tenológico, maestría en Computación, CIMAT, 1999.