

## **Aplicação da realidade virtual na representação de erros cinemáticos de máquinas de medir por coordenadas**

**Leonam J Leal de Paula (UNIMEP)** [ljpaula@unimep.br](mailto:ljpaula@unimep.br)

**Alvaro J Abackerli (UNIMEP)** [abakerli@unimep.br](mailto:abakerli@unimep.br)

**Nivaldi Calônego Jr (UNIMEP)** [ncalonego@unimep.br](mailto:ncalonego@unimep.br)

### **Resumo**

*Máquinas de medir por coordenadas (MMCs) são equipamentos versáteis empregados no controle de qualidade de peças mecânicas. São equipamentos de operação simples, mas que envolvem conceitos complexos ligados aos resultados das medições. Um destes conceitos são os erros paramétricos, que são movimentos micrométricos indesejáveis que afetam o resultado da medição e podem ser examinados usando as chamadas máquinas virtuais de medir. A equipe deste projeto tem em desenvolvimento um software que representa graficamente uma MMC real em ambiente virtual 3D. Nesta MMC virtual foram programados inicialmente apenas os movimentos preferenciais X, Y e Z, análogos ao de uma MMC, sem representação dos erros. Entretanto, sabe-se da importância de compreender os efeitos dos erros nos resultados das medições e este ferramental 3D tem potencial para representá-los, auxiliando esta compreensão. Pela importância do problema e pelo potencial aparente de solução, busca-se discutir a implementação de um ferramental para representação e análise dos erros em ambiente 3D, utilizando rotinas implementadas em linguagem JAVA, da modelagem polinomial de erros cinemáticos e da representação de rotações diferenciais usando quaternios. Os resultados do projeto proporcionam aos usuários de MMCs a visualização de erros cinemáticos para análise de diferentes fatores que interferem no desempenho das MMCs.*

*Palavras-chave: Máquina de Medir por Coordenadas; Máquina Virtual de Medir por Coordenadas; Erros Paramétricos, Realidade Virtual.*

### **1. Introdução**

Devido a grande exigência por qualidade imposta pelo mercado, a presença das máquinas de medir por coordenadas (MMCs) nos setores de qualidade das indústrias hoje em dia é cada vez mais importante, principalmente nas indústrias ligadas ao setor metal-mecânico. Estas máquinas são responsáveis pela garantia da qualidade dimensional dos produtos, e, portanto, vitais para a geração de lucros destas empresas. Todas elas possuem um sistema apalpador que é o responsável por medir os pontos coordenados X, Y, ou Z. São equipamentos de alto desempenho que detectam variações microscópicas na medição dimensional. As MMCs típicas podem medir desde peças como cabeçotes e blocos de motores, até carros inteiros e asas de aviões. A figura abaixo ilustra uma MMC tridimensional com os comprimentos dos eixos de 711 X 711 X 600 mm nos eixos X, Y e Z respectivamente, usada como base para o desenvolvimento aqui discutido.



**Figura 1** – MMC tridimensional 771x711x600 mm

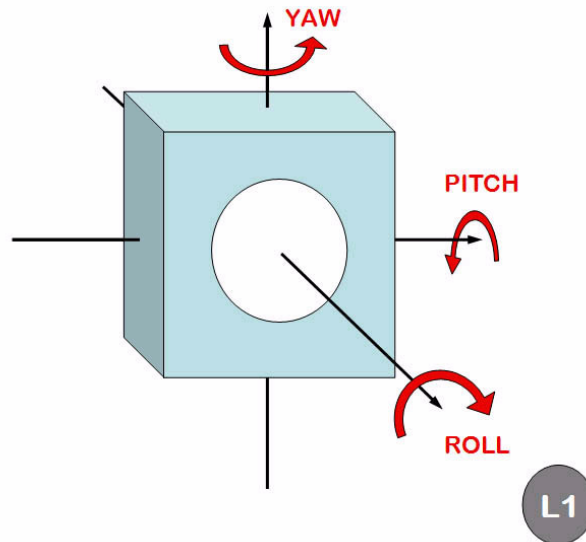
Mesmo sendo versáteis e de fácil manuseio, os usuários de MMCs se deparam cotidianamente com os diversos conceitos complexos que envolvem a medição por coordenadas. Segundo ABACKERLI (2003) essa versatilidade confere às MMCs uma capacidade de aplicação em tarefas complexas, porém dificulta as decisões relativas à medição, particularmente devido aos erros existentes em todas as máquinas. Em particular, merecem atenção especial os erros paramétricos, que são erros que variam em função da coordenada do volume de trabalho ocupada pelo apalpador máquina.

Estes erros são micrométricos ( $10^{-6}$ m), podendo ser considerados movimentos indesejáveis que impossibilitam a perfeição dos movimentos das estruturas das MMCs; afetam assim os resultados da medição. Ao todo são 21 os erros paramétricos, dentre os quais dezoito são cinemáticos e três são erros de perpendicularismo entre os planos XY, XZ e YZ. Para fins de classificação, os dezoito erros cinemáticos podem ser separados em nove erros de translação e nove erros de rotação, sendo, na verdade, três erros de translação e três de rotação em cada corpo rígido que compõe a máquina nos eixos preferenciais X, Y e Z, totalizando os dezoito erros citados.

Dos erros de translação em cada eixo temos um de posição e dois de retilineidade. O erro de posição é aquele que não permite que a estrutura da máquina se posicione na posição comandada pelo usuário, mas sim em uma outra muito próxima à mesma. Ou seja, trata-se de um erro na direção preferencial de movimento (X, Y ou Z). Existem ainda os dois erros de retilineidade em cada eixo preferencial, que provocam movimentos micrométricos de translação em direções perpendiculares às do movimento. Por exemplo, o eixo Y tem-se dois erros de retilineidade nas direções X e Z; o eixo X tem um na direção Y e outro na Z e assim sucessivamente. Estes erros de retilineidade impedem que a estrutura em movimento execute uma trajetória perfeitamente retilínea. Para entendê-los melhor, imaginemos um avião voando retilinearmente. Enquanto ele se movimenta na direção reta de vôo ele balança levemente para cima e para baixo, ou para a esquerda e para a direita. Da mesma forma, os erros de retilineidade atuam nos movimentos da máquina fazendo com que ao se movimentar ela

também varie transversalmente para cima e para baixo, para a esquerda e para a direita durante o movimento em um eixo horizontal qualquer.

Além dos erros de translação, também os erros de rotação são três, nominalmente: *roll*, *pitch* e *yaw*. Eles também atuam nos três eixos a máquina (X, Y e Z), contabilizando o total de nove erros de rotação. O erro de *roll* causa uma rotação micrométrica na estrutura durante seu movimento, para a qual o eixo de rotação é o mesmo eixo de movimento. Já o *pitch* e o *yaw* geram rotações micrométricas em eixos perpendiculares à direção preferencial de movimento. A figura abaixo tenta ilustrar uma representação para estes importantes erros de rotação, onde L1 indica a direção preferencial de movimento.

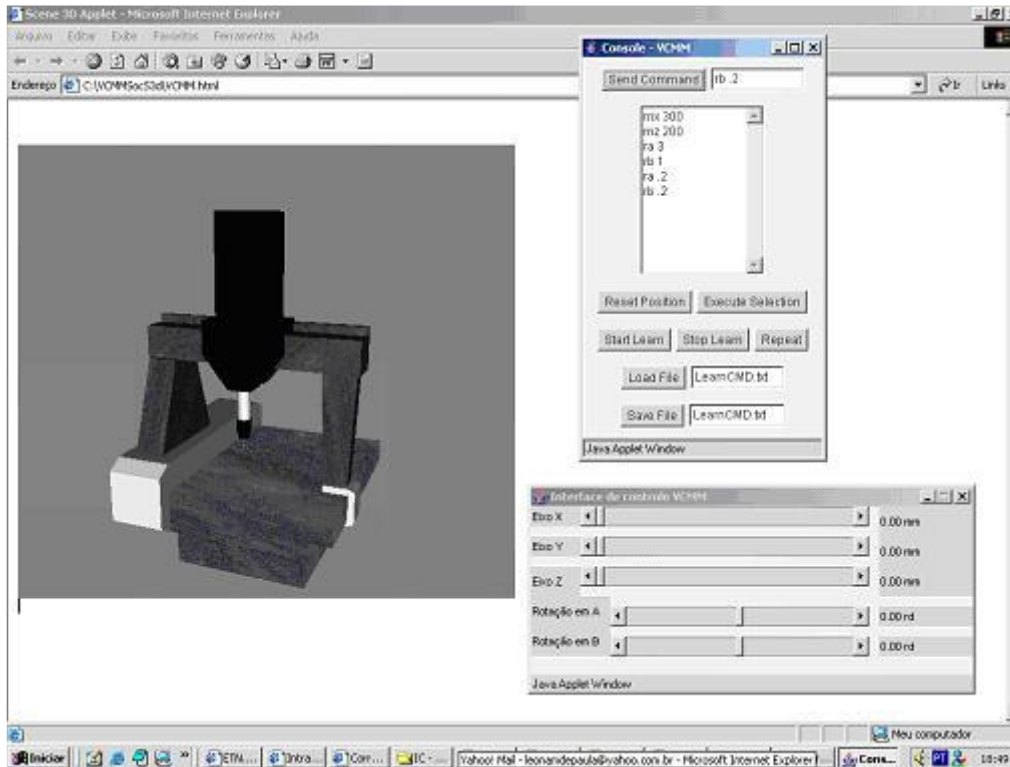


**Figura 2** - Os erros angulares *Roll*, *Pitch* e *Yaw*.  
Fonte: <<http://www.psfc.mit.edu>>

Para que sejam obtidos resultados otimizados na medição, é necessário que o usuário saiba interpretar o efeito destes erros das MMCs. Porém, como revela a discussão acima, esta análise é relativamente complicada se feita por um não especialista porque os erros são invisíveis a olho nu e se comportam de forma complexas devido às suas interações simultâneas.

Uma solução para este problema de interpretação do efeito dos erros paramétricos nas medições são as máquinas virtuais de medir por coordenadas. Existem diversas destas máquinas no mercado, como as produzidas pela SILMA ([www.silma.com](http://www.silma.com)) ou pela MetroSage ([www.metrosage.com](http://www.metrosage.com)), dentre outras. Porém os softwares simuladores disponíveis no mercado não oferecem a representação gráfica dos erros das máquinas de medir para fins de análise, não favorecendo por isso essa visualização direta dos efeitos dos erros paramétricos.

A equipe deste projeto tem em processo de desenvolvimento um software que simula graficamente uma MMC tridimensional. Ele possui uma interface que apresenta uma MMC de cinco graus de liberdade, desenhada em VRML (*Virtual Reality Modelling Language*), que possui movimentos análogos aos de uma MMC real. A figura 3 ilustra o aplicativo.



**Figura 3** – VCMM, interface de comandos e interface de barras.

Toda a programação feita em JAVA® permite a geração de comandos de movimento no modelo VRML® para os eixos X, Y e Z, além da livre movimentação por meio das barras de rolagem. Especificamente, a interface recebe comandos com sintaxe “*mx Coord*”, “*my Coord*” e “*mz Coord*” que representam deslocamentos lineares para as posições “*Coord*” de cada eixo preferencial. Além destes, pode-se comandar as rotações típicas “A” e “B” das cabeças indexadoras com instruções “*ra Ang*”, “*rb Ang*”, onde “*Ang*” representa a posição angular de indexação do apalpador. A composição destas instruções básicas na janela de comando permite simular a movimentação da máquina em tarefas rotineiras de medição.

Na interface de barras, cada eixo possui uma barra e o intervalo de movimento realizado pelo eixo é proporcional ao intervalo da respectiva barra. Assim cada vez que se muda de lugar o marcador de uma das barras a estrutura correspondente na VCMM se moverá de acordo com o deslocamento gerado pelo marcador. O ambiente de realidade virtual permite a visualização da máquina por diferentes ângulos e com diferentes ampliações, favorecendo a observação de efeitos micrométricos em qualquer condição desejada pelo usuário.

Como pode ser notado, a interface inicialmente disponível modelava a representação física da máquina, porém não seus erros cinemáticos que traz especial interesse na presente discussão. Por isso, discute-se aqui a implementação gráfica dos erros cinemáticos para fins de representação e apoio ao ensino/aprendizado dos efeitos destes erros na medição por coordenadas. Esta implementação exige a elaboração de interfaces que permitam aos usuários configurar características dos erros, implementação esta discutida nos itens a seguir.

## 2. Desenvolvimento do aplicativo

Dentre as linguagens existem para a geração de animações em desenhos VRML, tem-se a linguagem JAVA. Ela possibilita que este software seja executado pela Internet através de uma *Applet* e também disponibiliza uma grande variedade de opções na construção de

interfaces. Por estes motivos, foi escolhida para estruturar o código gerador de movimentos da máquina virtual em questão.

Uma das formas de se modelar os erros cinemáticos é utilizando equações polinomiais, tornando o valor do erro expresso em função da posição em que se encontra o corpo rígido da MMC em que o erro atua; mais especificamente os eixos X, Y e Z. Todos os dezoito erros possuem uma equação polinomial que diferem apenas pelo valor dos coeficientes dos polinômios usados na representação matemática do erro. Por serem equações polinomiais de grau seis, são previstos aqui sete coeficientes para cada equação erro, totalizando 126 coeficientes que ditam a intensidade dos erros sobre a MMC. Utilizando desta modelagem introduziu-se no software uma rotina de programação que provoca o re-cálculo dos valores dos erros a cada deslocamento dos eixos. Calculados os erros para cada posição tridimensional, segue-se a lógica para a representação dos erros de translação e de rotação acima discutidos.

Existem no ambiente JAVA duas funções importantes para a geração de movimentos de translação e rotação na MMC virtual quando ela se movimenta em qualquer dos eixos X, Y ou Z, a função *set\_translation()* e a *set\_rotation()*. Estas funções geram movimentos nos objetos virtuais alterando a cena do desenho. Utilizando pequenas alterações introduzidas sequencialmente são gerados os movimentos na máquina virtual.

Quando a função de translação é usada, a estrutura especificada é reposicionada em uma nova coordenada no sistema de referências do ambiente virtual, que deve ser informada à função. Para a representação dos erros de translação, primeiramente os valores dos erros de translação são calculados, sendo depois acrescidos às coordenadas genéricas ( $x_0, y_0, z_0$ ) em quem serão reposicionadas as estruturas. Ou seja, tomando erros de translação genéricos na direção Z e as posições genéricas ( $x_0, y_0, z_0$ ) para a qual a máquina deve ser comandada, calculam-se inicialmente os erros de posição ( $dz_z$ ) e retilidade ( $dx_z$  e  $dy_z$ ) pelas equações abaixo.

$$dx_z = b_0 + b_1 z_0 + b_2 z_0^2 + b_3 z_0^3 + b_4 z_0^4 + b_5 z_0^5 + b_6 z_0^6 \quad (1)$$

$$dy_z = c_0 + c_1 z_0 + c_2 z_0^2 + c_3 z_0^3 + c_4 z_0^4 + c_5 z_0^5 + c_6 z_0^6 \quad (2)$$

$$dz_z = a_0 + a_1 z_0 + a_2 z_0^2 + a_3 z_0^3 + a_4 z_0^4 + a_5 z_0^5 + a_6 z_0^6 \quad (3)$$

Calculados os erros, a máquina comandada nas posições  $(x_0', y_0', z_0') = (x_0 + dx_z, y_0 + dy_z, z_0 + dz_z)$  pela simples aplicação da função de translação na forma *set\_translation*( $x_0 + dx_z, y_0 + dy_z, z_0 + dz_z$ ). Desta forma, quando o comando já contém os erros na nova coordenada sua representação gráfica pode ser feita e visualizada pela aplicação da amplificação apropriada na imagem a renderizar.

Na função de rotação é usado um versor no sistema de coordenadas do ambiente virtual, além de um ângulo de giro em torno deste versor. Para a representação dos erros de rotação necessitou-se da implantação do conceito de quatérnios (BIASI e GATTASS, 2002) de rotação e de matriz de transformação homogênea (ROMANO, 2002). Estes quatérnios transformam as rotações individuais que representam os erros de *pitch*, *yaw* e *roll*, ou seja, os vários erros de rotação, em apenas uma rotação micrométrica em torno de um versor de resultante.

De modo análogo, primeiramente são calculados os erros usando os polinômios que caracterizam os erros cinemáticos rotacionais, seguindo-se o cálculo do quatérnio e do ângulo de rotação resultante dos erros de *pitch*, *yaw* e *roll* para cada posição genérica ( $x_0, y_0, z_0$ ) ocupada pela máquina. As expressões abaixo ilustram os polinômios para cálculo dos erros de *pitch* ( $ex_z$ ), *yaw* ( $ey_z$ ) e *roll* ( $ez_z$ ), além do versor  $\vec{q}$  (BIASI e GATTASS, 2002) que caracteriza o quatérnio e do ângulo  $\alpha$  de rotação, determinados pelos parâmetros da matriz de transformação homogênea  $T$  (ROMANO, 2002).

$$exz = d_0 + d_1 z_0 + d_2 z_0^2 + d_3 z_0^3 + d_4 z_0^4 + d_5 z_0^5 + d_6 z_0^6 \quad (4)$$

$$eyz = e_0 + e_1 z_0 + e_2 z_0^2 + e_3 z_0^3 + e_4 z_0^4 + e_5 z_0^5 + e_6 z_0^6 \quad (5)$$

$$ezz = f_0 + f_1 z_0 + f_2 z_0^2 + f_3 z_0^3 + f_4 z_0^4 + f_5 z_0^5 + f_6 z_0^6 \quad (6)$$

$$T = \begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix} \quad (7)$$

$$q_1 = \frac{1}{2} \sqrt{n_x + s_y + a_z + 1} \quad (8)$$

$$q_2 = \frac{1}{2} \sqrt{n_x - s_y - a_z + 1} \quad (9)$$

$$q_3 = \frac{1}{2} \sqrt{s_y - n_x - a_z + 1} \quad (10)$$

$$q_4 = \frac{1}{2} \sqrt{A_z - N_x - S_y + 1} \quad (11)$$

$$\alpha = 2 \arccos(q_1) \quad (12)$$

$$\vec{q} = \left\{ \frac{q_2}{\sin(\alpha)}, \frac{q_3}{\sin(\alpha)}, \frac{q_4}{\sin(\alpha)} \right\} \quad (13)$$

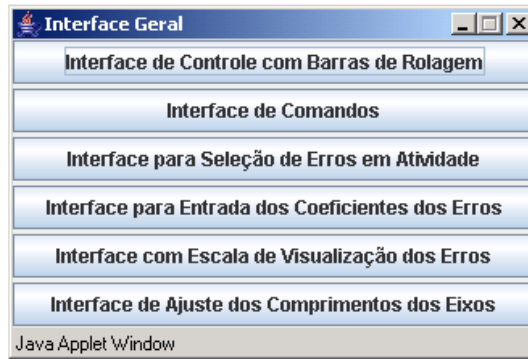
Utilizando estes resultados, executa-se a função de rotação na forma *set \_ rotation*( $\vec{q}$ ,  $\alpha$ ) e para a posição ( $x_0$ ,  $y_0$ ,  $z_0$ ) na qual a máquina foi comandada, gerando-se assim a representação dos erros cinemáticos rotacionais de interesse. Do mesmo modo, essa representação pode ser visualizada de diferentes ângulos de observação e com diferentes ampliações que satisfaçam as necessidades do usuário.

Realizado e testado todo o processo de implementação, documentou-se o código explicando a funcionalidade de cada método e das variáveis nele contidas. A ferramenta utilizada para esta documentação foi o JAVADOC, que gera um documento HTML e que contém os métodos e variáveis, juntamente com seus respectivos comentários que os explicam no código, cujos resultados são discutidos a seguir.

### 3. Resultados e Discussão

Para o usuário poder simular os erros na máquina virtual foi necessária a utilização das interfaces elaboradas para sua configuração. Foram elaboradas cinco interfaces das quais três são para a análise dos erros cinemáticos, uma para a configuração dos comprimentos dos eixos da máquina virtual e a última um console geral de comandos. Seguem abaixo as interfaces criadas e a discussão das suas funcionalidades, onde se discute também as instruções necessárias para a sua operação e visualização dos erros.

Assim que o usuário inicia o software aparece, juntamente com a representação VRML da máquina (figura 3), uma interface nomeada *Interface\_Geral* que possibilita, através de seus botões, a abertura de todas as outras seis interfaces integrantes do aplicativo. Esta interface só pode ser fechada quando o software é finalizado, funcionando portando como um console geral de comandos. Sua função é otimizar o espaço de vídeo, permitindo que as outras interfaces possam ser fechadas e abertas a qualquer momento sem congestionamento do monitor para a melhor visualização dos erros, figura 4.



**Figura 4** – Interface\_Geral

A outra interface, figura 5, permite inserir os coeficientes dos polinômios que modelam os erros, conforme acima discutidos. Sua operação é conseguida inicialmente pela inserção dos valores dos coeficientes que ditam a intensidade dos erros. Para maior funcionalidade, os coeficientes podem ser introduzidos em diferentes formatos numéricos como o decimal, por exemplo, ou mesmo por meio de mantissa e expoente para garantir uma precisão adequada na representação do erro de interesse. Devido ao grande número de dados com que se trabalha nesta interface, ela conta ainda com as opções de Salvar/Abrir, que favorecem o armazenamento e a recuperação de conjuntos de coeficientes que representem os diferentes erros cinemáticos de máquinas que se queira simular.

Interface para Entrada dos Valores dos Coeficientes dos Erros								
	a0	a1	a2	a3	a4	a5	a6	
dx(x)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
dy(x)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
dz(x)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
dx(y)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
dy(y)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
dz(y)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
dx(z)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
dy(z)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
dz(z)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
ex(x)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
ey(x)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
ez(x)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
ex(y)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
ey(y)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
ez(y)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
ex(z)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
ey(z)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
ez(z)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
				Ok	Aplicar	Abrir	Salvar	Cancelar

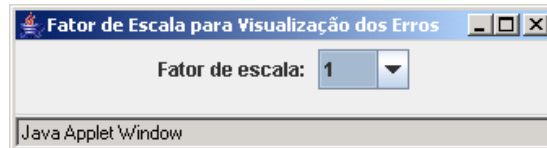
Java Applet Window

**Figura 5** – Interface para a entrada dos coeficientes dos erros

É interessante observar ainda na interface na figura 5 a aparente ausência de dados que modelem os erros discutidos de perpendicularismo. Ao contrário, tais erros podem ser facilmente modelados por meios dos polinômios que representam erros de retilinearidade,

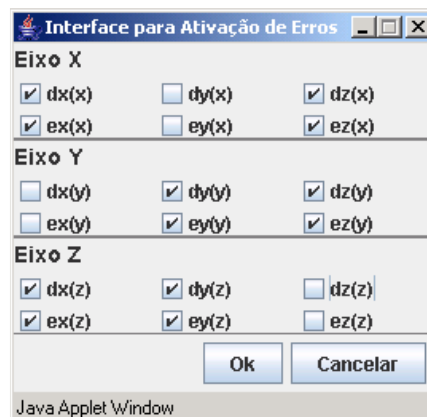
bastando para isso escolher os polinômios apropriados e incluir o perpendicularismo no processo de ajuste numérico das funções polinomiais.

Inseridos os coeficientes o usuário pode executar os movimentos na máquina e os erros já estarão sendo representados, porém, eventualmente, não poderão ser visualizados devido seu tamanho tipicamente micrométrico. Para favorecer o uso de coeficientes que representem erros paramétricos reais de máquinas de medir, desenvolveu-se para fins de visualização o processo de ampliação/dedução dos erros. Modificando o valor da escala na lista de seleção da interface, figura 6, o usuário altera a representação gráfica dos erros de acordo com suas necessidades, permanecendo ainda as alternativas já discutidas de alterar o ângulo de visualização provido pelo ambiente de desenvolvimento.



**Figura 6** – Interface para escolha do fator de escala para a visualização erros

Com o auxílio das funcionalidades das interfaces discutidas, o usuário visualiza a representação gráfica de todos os erros cujos polinômios tenham sido inseridos através dos seus coeficientes. Entretanto, para fins de análise do comportamento da máquina e aprendizado dos efeitos dos erros no seu desempenho, torna-se interessante facilitar o processo de inclusão e exclusão de erros sem a necessidade de excluir os coeficientes. Isso pode ser feito pela interface de ativação dos erros, figura 7, através da qual pode-se escolher quais erros se deseja analisar.

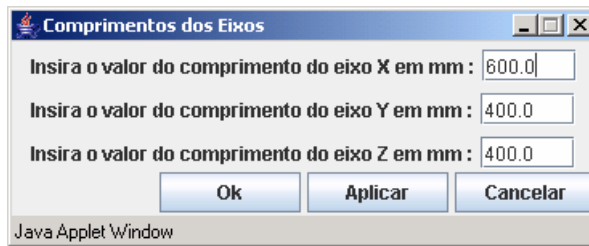


**Figura 7** – Interface para escolha do fator de escala para a visualização erros

A interface disponibiliza caixas de seleção que permitem a escolha de quais erros devem estar ativos durante o movimento da máquina. Desta forma, o usuário pode analisar diferentes combinações de erros e verificar como a combinação dos mesmos interfere no desempenho da sua máquina.

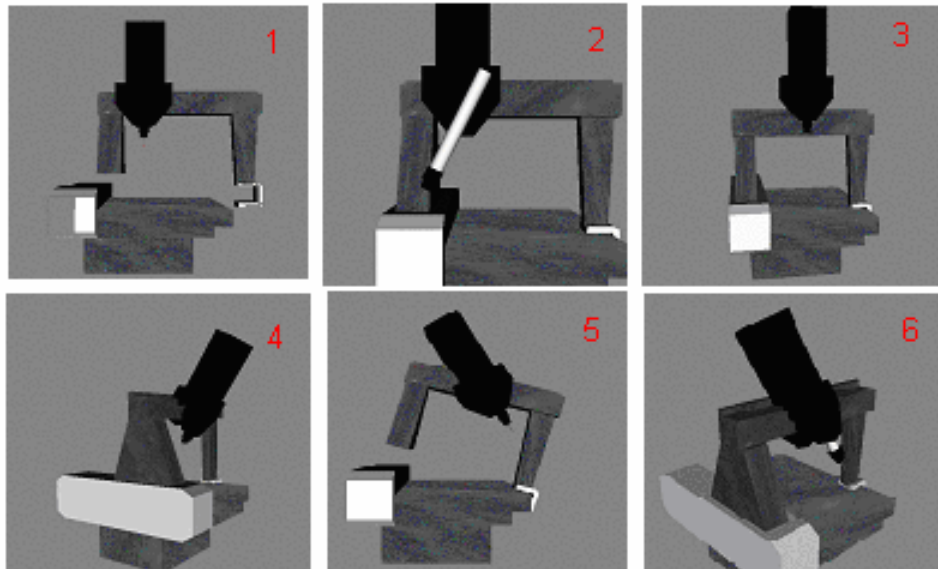
Considerando que existem diferentes MMCs, com dimensões variadas, criou-se uma interface para a configuração dos comprimentos dos eixos da máquina virtual, tal que os polinômios de erros possam ser calculados e representados no domínio completo da máquina simulada. A figura 8 a seguir ilustra a interface com a qual se altera o comprimento dos eixos da máquina.





**Figura 8** – Interface para a configuração dos comprimentos dos eixos

Como esperado, usando as interfaces discutidas podem ser representados os diversos tipos de erros, conforme ilustra a figura 9 através de imagens aqui exageradas apenas para fins de ilustração. Em (1) tem-se representação dos erros de reticidade  $dxy$  e  $dzy$  atuando conjuntamente. Em (2) observa-se o erro de *yaw* ( $dyz$ ) em Z. Em (3) mostra-se o erro de reticidade em Z de X, ou seja,  $dzx$ . Em (4) representa-se o erro de *roll* ( $exx$ ) em X. Em (5) estão ativos os erros de *pitch* de X e de *roll* de Y, ou seja,  $eyx$  e  $eyy$ . Finalmente, em (6) estão sendo simultaneamente representados os erros de *roll* ( $exx$ ), *pitch* ( $eyx$ ) e *yaw* ( $ezx$ ) no eixo X.



**Figura 9** – Erros cinemáticos em representação gráfica ampliada

Estas são apenas algumas das possíveis combinações de representação de erros disponíveis no software. Cabe ao usuário especificar ao software as configurações que atendam às necessidades na simulação desejada.

## 5. Considerações finais

Este projeto completa mais uma fase do desenvolvimento da VCMM em estudo, que demonstra ser uma ferramenta metrológica realmente de grande potencial, tanto para fins de orientação aos usuários de MMCs como para fins didáticos no ensino de Engenharia.

A implementação tornou possível a visualização dos erros que são dos invisíveis a olho nu numa máquina real. Com o aplicativo aqui discutido, eles podem ser visualizados e entendidos sem maiores dificuldades, por meio de operações simples feitas no software através de acesso remoto e com boa funcionalidade.

Estes resultados têm particular importância devido ao fato de que não foi encontrado produto comercial que satisfaçam as necessidades de representação de erros aqui discutida, caracterizando, portanto, uma contribuição interessante no contexto da metrologia e da engenharia.

Finalmente, é relevante destacar que, devido ao modo pelo qual o aplicativo foi desenvolvido, ele permite representar os erros de diferentes modelos de máquinas com cinco graus de liberdade (*moving table, cantiliver, gantry, horizontal arm, etc.*), pela simples substituição do desenho VRML que pode ser obtido automaticamente de diversos aplicativos CAD comerciais sem maiores dificuldades.

## **6. Referências Bibliográficas**

1. ABACKERLI, A. J.; ORREGO, R. M. M. **Intercomparação de Medições em Máquinas de Medir por Coordenadas - II. Parte 2/2.** Relatório Final de Pesquisa. Projeto FAP 205-99. Universidade Metodista de Piracicaba – UNIMEP. Janeiro 2003.
2. ABACKERLI, A. J.; ORREGO, R. M. M.; CAUCHICK MIGUEL, P. A. **Princípios da Medição por Coordenadas.** Apostila de treinamento técnico especializado. 3ª edição revisada e ampliada. Setembro 2000. Publicação EGP – Engenharia & Tecnologia. Rua 9 de Julho 709, Salto – SP.F
3. BIASI, S. C.; GATTASS, M. **Uso de quatérnios para a representação de rotações em 3D.** Tecgraf. Disponível em: <http://www.tecgraf.puc-rio.br/~mgattass/fcg/material/Quaternios.pdf>. Acesso em: 01/ago/2005.
4. FURGERI, S. **Java 2: Ensino Didático: Desenvolvendo e Implementando Aplicações.** SP: Érica, 2000.
5. ROMANO, V. F. **Robótica Industrial.** SP: Edgar Blücher LTDA, 2000.

## **Agradecimentos**

Os autores deste trabalho agradecem o CNPq pelo apoio concedido na realização deste projeto.