

Calculo da Incerteza de Medição em Tarefas Especificas de CMMS através de Simulação

Autores

Octavio Luis Cazzonatto Carvalho

Orientador

Alvaro Jose Abackerli

Apoio Financeiro

Pibic

1. Introdução

Na tentativa de solucionar o problema de cálculo de incerteza de medição em tarefas específicas de Máquinas de Medir por Coordenadas - MMCs, iniciou-se o “desenvolvimento de um núcleo básico de simulação para medições por coordenadas em forma de um “toolbox” Matlab®” (ABACKERLI, 2005) que esteve inicialmente limitado por sua característica mono-usuário e por possuir diversas limitações relativas à entrada de dados e à comunicação com o usuário. Houve várias modificações nessa fase (1ª), pois em função do seu conteúdo técnico complexo a interface ficou difícil de ser configurada, dificultando a sua aplicação mesmo em problemas simples de simulação.

A 2ª fase as dificuldades de representação da configuração da medição fizeram as interfaces evoluir ainda na forma de um aplicativo *stand-alone* Delphi ®” (ABACKERLI, 2005). Nessa fase, foram resolvidos alguns problemas de representação da medição, permitindo a validação da proposta para casos simples de simulação, mas ainda manteve a forma de um aplicativo mono-usuário.

A 3ª fase tratou de melhorar os aspectos práticos necessários para o desenvolvimento desse tipo de software, pesquisando junto a usuários de MMCs que forneceram novos parâmetros para um estudo aprofundado sobre o desenvolvimento e o teste de software aplicado à simulação em metrologia.

Estudos também foram realizados através de um simulador comercial em desenvolvimento pela MetroSage , que permitiu gerar alguns parâmetros para o projeto e validação de simuladores que tem por objetivo realizar a estimativa de incertezas na medição por coordenadas. Os estudos tiveram sua continuidade com a análise de aspectos de usabilidade, configuração e similaridade (CARVALHO, 2005) destacando vários pontos importantes da interface que podem ser aperfeiçoados e assim melhorar o uso da aplicação sob o enfoque do usuário.

Destes estudos, partes do “toolbox” Matlab® foram aprimoradas já na sua 3ª versão. Essa versão trouxe vários aspectos que melhoraram a compreensão sobre a medição por coordenadas e sobre o processo de simulação. Após realizados os aperfeiçoamentos dessas e de outras características na representação da medição, criou-se uma proposta do novo modelo de aplicação, que se encontra em implementação no “projeto mãe”, no qual a proposta aqui discutida é incorporada.

Para que a realização da proposta aqui discutida foi necessário investigar alguns aspectos importantes das MMCs conforme Abackerli (2000) e Bosch (1995), além de estudar a operação da interface do simulador, agora implementada como um aplicativo da WEB em linguagem Java.

Esses estudos auxiliaram o desenvolvimento e a implementação da comunicação entre interfaces (Java) e rotinas de cálculos desenvolvidas em Matlab® (CHAPMAN, 2003; MATSUMOTO, 2004), permitindo realizar uma comparação entre um software disponível no mercado e o simulador aqui discutido.

2. Objetivos

O presente artigo discute a análise das interfaces do simulador e a ligação via WEB entre o aplicativo de configuração e as rotinas de cálculo.

3. Desenvolvimento

A metodologia adotada seguiu as etapas de familiarização com a interface, a implementação da comunicação Matlab® via Web e o teste da simulação.

Com essa metodologia como base para o desenvolvimento do trabalho foi realizada uma pesquisa utilizando o laboratório de metrologia da UNIMEP, Campus Santa Bárbara d'Oeste, São Paulo. No laboratório utilizou-se como fonte de pesquisas, artigos e trabalhos disponíveis no Portal da Capes. Também foi utilizado o laboratório de metrologia para a familiarização com as interfaces, para a implementação das rotinas de comunicação com o Matlab® e para as comparações com o software comercial.

A busca de base teórica foi também realizada por meio de consultas a obras especializadas na área de metrologia, engenharia de software, entre outras áreas do conhecimento da engenharia, que permitiram através dos cálculos, realizar a implementação das rotinas responsáveis pela simulação da MMC.

A implementação do código Matlab® foi feita seguindo procedimentos padrão de implementação de softwares básicos, consistindo da implementação das rotinas de interesse e do seu teste com dados coletados da interface WEB no formato XML.

4. Resultados

As interfaces desenvolvidas em Java são apresentadas em dispositivos de visualização utilizados em *browsers* para acesso a *Internet* que possibilitam ao usuário realizar simulações da máquina virtual de medir por coordenadas – MVMC por meio de sua configuração, tornando possível uma simulação via *WEB*. Atualmente o software é composto por um conjunto de 12 interfaces de configuração, que são:

- 1 - Definição do nível de usuário e escolha das características de medição;
- 2 - Definição da MMC e suas características;
- 3 - Definição do modelo de apalpador;

- 4 - Escolha do tipo de ponta do apalpador;
- 5 - Definição dos parâmetros de medição dos apalpadores;
- 6 - Definição dos parâmetros de temperatura;
- 7 - Configuração da peça;
- 8 - Posição do volume;
- 9 - Informação da fabricação;
- 10 - Resultado da simulação da *task specific*;
- 11 - Resultado da simulação do *probe system*;
- 12 - Resultado da simulação do *volume mapping*.

Como exemplo aqui é discutida a configuração do apalpador que pode ser executada após a escolha do tipo de usuário e a configuração da MMC; interfaces 1 e 2. Nessa etapa se realiza a escolha do apalpador (Figura 1) iniciando-se pela definição do Modelo da cabeça do apalpador (*Probe head model*), cujas opções são entre a Cabeça Fixa (*Fixed*) ou a Articulada (*Articulated*). Feito isso, define-se a orientação da cabeça no eixo da máquina e após isso o usuário deve então definir o comprimento da extensão do apalpador (*Probe extension length*), a extensão para a ponta de contato (*Stylus extension length*), o Modelo do apalpador (*Probe model*) e o tipo de ponta do apalpador, que o levará a uma outra interface para a interface de escolha do tipo de ponta do apalpador; interface 4.

Após realizar a configuração do tipo de ponta do apalpador, a configuração continua pela definição de todas as características desejadas na simulação até chegar à interface 9 onde são definidos os detalhes de fabricação da peça medida, seguindo-se a visualização dos resultados simulados por meio das interfaces 10, 11 e 12.

Figura 1 - Definição do modelo de apalpador (interface 3).

Depois de configuradas, elas enviam via WEB ao simulador Matlab® os dados definidos pelo usuário em forma de uma estrutura Extensible Markup Language – XML. Um exemplo de estrutura XML pode ser vista na Figura 2. Nesse exemplo são mostrados os dados que definem o tipo de MMC, a quantidade de repetições de simulações, o tipo de gráfico de saída de resultados, as dimensões e os sentidos dos eixos da MMC. Tais informações devem ser interpretadas pelo Matlab® para que seja realizada a simulação por meio de rotinas de cálculo implementadas para este fim.

A rotina que realiza a interpretação do XML executa uma varredura no código buscando os nodos terminais, que são as “partes finais” do XML e que carregam os dados configurados pelo usuário. Um exemplo de nodo terminal é a estrutura *1000* que “carrega” o valor de 1000 que representa o número simulações definido pelo usuário.

Figura 2: Exemplo de estrutura XML.

Depois de realizada a leitura do XML e com os dados já disponíveis no ambiente Matlab®, as rotinas de cálculos executam a simulação seguindo a lógica ilustrada nas etapas descritas a seguir:

- a) – Lê a configuração da MMC no XML;
- b) – Lê o tipo de simulação a executar;
- c) – Lê e calcula a geometria ideal a simular;
- d) – Lê e calcula a temperatura e o modo de compensação térmica;
- e) – Lê as posições do volume a simular a geometria;
- f) – Coloca a geometria desejada do volume da máquina (posição e orientação);
- g) – Verifica se os dados dos passos 3, 4, 5 e 6 acima “cabem” na MMC;
- h) – Lê os erros da MMC e seleciona o modelo da máquina a usar nos cálculos;
- i) – Calcula as incertezas de medição;
- j) – Calcula o “off-set” do apalpador;
- l) – Calcula a geometria simulada com erros e incertezas;
- m) – Compara geometria ideal com geometria contendo erros/incertezas;
- n) – Apresenta os resultados.

Neste projeto essa lógica de simulação foi implementada para realizar a simulação de tarefas de MMC envolvendo um tipo específico de máquina (ponte móvel), um tipo específico de sistema apalpador (*touch-trigger*), uma tarefa de medição (geometrias básicas) em um ambiente de medição (temperatura a 20°C).

Ao serem simuladas, essas rotinas geram os resultados mostrados ilustrativamente nos gráficos da Figura 3, que correspondem às interfaces disponíveis nos passos 10, 11 e 12 acima citados, que foram comparados à forma de representação dos resultados do simulador comercial. Com essa comparação notou-se que os gráficos do simulador “toolbox” Matlab®, além de fornecerem mais opções de análise dos resultados padrão, oferecem resultados alternativos que não estão disponíveis no simulador comercial; como é o caso dos gráficos que mostram os erros do apalpador (Fig. 3-11) e do volume da máquina (Fig. 3-12).

5. Considerações Finais

O estudo inicial das interfaces do simulador Java foi de grande ajuda para o processo de aprendizagem da simulação, auxiliando na organização e na implementação das rotinas de cálculos desenvolvidas em Matlab®. O estudo dessas interfaces possibilitou o correto entendimento da lógica de uma simulação, que permitiu o seqüenciamento correto dos passos de simulação ilustrados nos itens (a) a (n) acima.

Com a implementação das rotinas de cálculos pode-se implementar a comunicação entre as rotinas e a interface de configuração da simulação. A partir da comunicação via XML pode-se realizar a simulação de modo a verificar a funcionalidade desse processo por meio de testes onde eram verificados os dados de entrada-saída, podendo-se assim validar a comunicação WEB-Matlab® via XML que era o objetivo central deste estudo.

Os autores deste trabalho agradecem ao CNPq pelo apoio concedido na realização deste projeto e ao Professor Luis Augusto Consularo com a colaboração para o desenvolvimento do projeto.

Referências Bibliográficas

ABACKERLI, A. J. *et al.* **Princípios da medição por coordenadas**. 3. ed. Salto: Engenharia & Tecnologia, 2000. Apostila.

ABACKERLI, A. J. **Cálculo da incerteza de medição em tarefas específicas de CMMs através de simulação**. Proposta de Pesquisa: nível graduação. Santa Bárbara d'Oeste: Universidade Metodista de Piracicaba, 2005.

BOSCH, John A. **Coordinate Measuring Machines and Systems**. New York: Marcel Dekker, Inc. 1995.

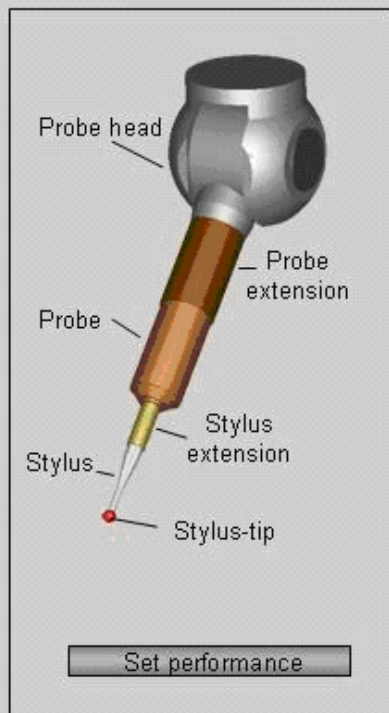
CARVALHO, O.L.C. **Teste de um simulador para a medição por coordenadas**. Projeto PIBIC/UNIMEP 227-04. Santa Bárbara d'Oeste – SP. jul. 2005.

CHAPMAN, Stephen J. **Programação em Matlab® para engenheiros**. Tradução técnica Flavio Soares Correa da Silva. São Paulo: Pioneira Thomson Learning, 2003.

MATSUMOTO, Élia Yathie. **Matlab® 7: Fundamentos**. 1. ed. São Paulo: Érica, 2004.

Anexos

Probe system



Components configuration

Probe head model **PH10**

Reference probe head orientation

Probe orientation for A=90 and B=0 **- Y**

Fixed orientation for the probe **- Z**

☒ Fixed orientation ☐ Articulated

Probe extension length (L) **100** mm

Length (L)

Stylus extension length (L) **50** mm

Length (L)

Stylus setup

Probe model **TP20**

Cancel

Help

Accept

```
<SimulationTask>
  <Type>Task</Type>
  <Runs>1000</Runs>
  <PlotType>Plot1 </PlotType>
</SimulationTask>
<VCMMSetup>
  <MachineConfiguration>
    <CMMXAxis>
      <CMMAxisOrientation>+A</CMMAxisOrientation>
      <CMMAxisLength>400</CMMAxisLength>
    </CMMXAxis>
    <CMMYAxis>
      <CMMAxisOrientation>+B</CMMAxisOrientation>
      <CMMAxisLength>500</CMMAxisLength>
    </CMMYAxis>
    <CMMZAxis>
      <CMMAxisOrientation>-C</CMMAxisOrientation>
      <CMMAxisLength>600</CMMAxisLength>
    </CMMZAxis>
    <CMMTypeSelect>MovingBridge</CMMTypeSelect>
  </MachineConfiguration>
```

