

Um Framework para o Teste Automatizado de Sensores

Darlan Felipe Klotz¹, Marco Tulio Guerreiro¹, Marcelo Teixeira¹, Richardson Ribeiro¹

¹Programa de Pós-Graduação em Engenharia Elétrica – Universidade Tecnológica Federal do Paraná (UTFPR) – Pato Branco – PR – Brasil

darlan.klotz@ifc.edu.br, marcotulio.guerreiro@gmail.com,
marceloteixeira@utfpr.edu.br, prof.richard@gmail.com

Abstract. *This paper proposes an automated procedure to test the operation of capacitive soil moisture sensors. The test procedure is modeled using Finite Automata and Language concepts. From the monolithic synthesis, we obtained a controller that was converted to Ladder language and implemented in a Programmable Logic Controller. A SCADA application monitors and communicates with the process via Modbus RTU protocol and RS 485 serial.*

Resumo. *Este trabalho propõe um procedimento automatizado para testar a operação de sensores capacitivos de umidade do solo. O procedimento de teste é modelado utilizando Autômatos Finitos e conceitos de Linguagens. A partir da síntese monolítica, foi obtido um controlador que foi convertido em linguagem Ladder e implementado em um Controlador Lógico Programável. Uma aplicação SCADA monitora e se comunica com o processo através do protocolo Modbus RTU e do padrão serial RS 485.*

1. Introdução

Procedimentos de testes possuem um papel fundamental na indústria [Tahera, Earl e Eckert 2012]. A partir de rotinas específicas de testes, é possível verificar o funcionamento de um produto, em sua forma final ou parcial. Um exemplo que é de interesse particular desse trabalho, envolve a fabricação de sensores capacitivos para a medição de umidade de solo. Nesse caso, um teste manual de validação apresenta potencial para alterar o resultado do teste em função de erros causados pela ação humana [Caetano 2008].

O presente artigo propõe um método automático para a execução de uma rotina de teste para sensores capacitivos de umidade de solo, visto que, atualmente, observa-se que essa prática é realizada manualmente. O procedimento consiste em inserir o sensor em um tanque com água e medir sua resposta em dois níveis diferentes, a fim de verificar, com maior confiabilidade, o correto funcionamento do sensor. Em cada nível, uma resposta pré-definida é esperada. O sensor é aprovado no teste caso as medidas nos dois níveis de água sejam iguais às medidas esperadas e reprovado caso contrário.

Inicialmente, o processo de teste foi observado como um Sistema a Eventos Discretos (SED) [Cassandras e Lafortune 2009]. Essa abordagem permite modelar a dinâmica do sistema por meio de um Autômato Finito (AF) tal que as sequências de evolução do sistema correspondem à evolução desse modelo em seu espaço de estados. A partir do modelo do sistema exposto como um AF, foram construídas especificações

de comportamento que, em conjunção, viabilizam o cálculo de um controlador para o processo. Esse controlador inclui todas e somente as sequências de operação que são permitidas para o sistema sob controle, ou seja, que levam a um comportamento seguro, robusto, minimamente restritivo e não bloqueante. O processo de cálculo desse controlador é denominado de síntese, e a sua concepção matemática é estruturada pela Teoria de Controle Supervisório (TCS) de SEDs [Wonhan 2002].

O controlador foi convertido em código implementável, nesse artigo em um Controlador Lógico Programável (CLP), usando a linguagem Ladder [Bryan e Bryan 1997]. Para fins de supervisão, foi desenvolvida e implementada uma aplicação SCADA que se comunica com o CLP através de uma rede industrial RS 485 regida pelo protocolo Modbus® RTU.

2. Descrição do Processo de testes

O dispositivo, cuja sequência de testes é automatizada nesse trabalho, é um sensor capacitivo que detecta a umidade do solo, como mostra a Figura 1. Suas hastes detectam o contato com a matéria, alterando a resposta do sensor conforme o meio é alterado.



Figura 1. Sensor capacitivo de umidade de solo a ser testado.

A rotina de teste dos sensores é assim descrita. Após a montagem do circuito, os sensores são postos em uma caixa ou recipiente, denominada de “testes”. Em seguida, um operador retira um sensor por vez da caixa de testes, conectando-o a um cabo para energizá-lo. Então, o sensor é inserido em um tanque com água, sendo que, inicialmente, 90 % do comprimento das hastes do sensor é submerso.

A resposta do sensor é lida por um circuito eletrônico e convertida para valores de tensão, sendo que o circuito final deve apresentar uma tensão de cerca de 3V. Caso o valor de leitura seja diferente de $3V \pm 0,1 V$, o sensor é reprovado no teste e descartado em uma caixa denominada de “nok”. Do contrário, i.e., caso o sensor seja aprovado no teste, o próximo passo consiste em elevar o sensor para o segundo nível de medição, de forma que as hastes se encontrem 50% submersas. Nesse caso, a tensão esperada é de $2.5V \pm 0,1 V$. Caso a tensão não esteja dentro da faixa aceitável, o sensor é reprovado e, igualmente, depositado em “nok”. Do contrário, o sensor é aprovado no teste e colocado em uma caixa de sensores aprovados, denominada de “ok”.

Para o caso em que uma máquina executa a rotina de testes, um braço robótico pode ser usado para buscar um novo sensor na caixa de testes, executar na ordem correta e segura as rotinas para verificação das medidas de tensão e, após o resultado, depositá-lo no recipiente de sensores aprovados ou reprovados [Rosário 2010].

3. Modelagem do processo de testes

Para a modelagem do sistema, foi considerada a inserção de um braço robótico, para a manipulação dos componentes a serem testados, e um conjunto de sensores que

identificam os componentes e auxiliam o braço. O manipulador robótico foi modelado com base em todos os movimentos possíveis para realizar o procedimento de teste e os sensores são modelados por auto laços, como mostra a Figura 2.

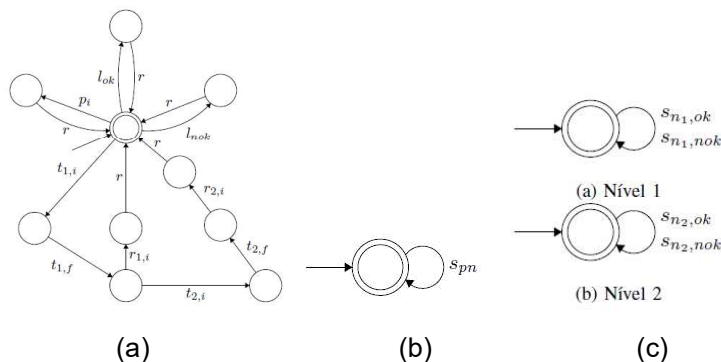


Figura 2. Modelos que compõem a planta do sistema.

Para iniciar o procedimento de testes, o manipulador realiza o movimento de submergir o sensor até nível 1. O comando $t_{1,i}$ inicia o movimento do braço robótico para levar o sensor até o nível 1, emitindo um sinal $t_{1,f}$ para indicar que o manipulador está posicionado com o sensor no nível 1. Então, o manipulador pode retornar, a partir do comando $r_{1,i}$ que inicia o retorno à posição de descanso caso o sensor seja reprovado, ou avançar ao próximo nível a partir do comando $t_{2,i}$ caso seja aprovado. Ao atingir o nível 2, o manipulador emite um sinal $t_{2,f}$ indicando estar em posição de teste. A partir do nível 2, a única possibilidade para o manipulador é voltar para a posição de descanso, que é realizado a partir do comando $r_{2,i}$. Quando o manipulador alcançar a posição de descanso, o sinal r é disparado acusando tal evento. Esse modelo é apresentado na Figura 2a.

Para evitar que o manipulador busque um novo sensor com a caixa “teste” vazia, há um sensor que detecta a presença de peças na caixa por um evento s_{pn} , visto na Figura 2b. Outros dois sensores, modelados conforme a Figura 2c, indicam se o teste retornou verdadeiro ou falso, ou seja, se as medidas de tensão estão dentro dos valores esperados ou não. No caso de sucesso, os sensores emitem os eventos $s_{n1,ok}$ e $s_{n2,ok}$ para o nível 1 e nível dois, respectivamente. Em caso de falha, os sensores emitem os sinais $s_{n1,nok}$ e $s_{n2,nok}$.

3.1. Especificações

Para o controle do sistema, quatro especificações foram modeladas. A primeira especificação, apresentada na Figura 3a, define que o manipulador busque uma peça na caixa “teste” se existe de fato um sensor esperando para ser testado.

A segunda especificação, apresentada na Figura 3b, é um complemento da anterior. Nesse caso, a especificação determina uma condição de manipulador ocupado, permitindo a busca de uma nova peça após o teste ser iniciado (evento $t_{1,i}$) e finalizado (evento l_{ok} ou l_{nok}). Dessa forma, essa especificação indica que o braço só pode testar novos sensores caso tenha terminado um procedimento.

Na terceira especificação, apresentada na Figura 3c, determina o procedimento para o primeiro teste. Após posicionado no nível 1 (ocorrência do evento $t_{1,f}$), o manipulador deve esperar o resultado do teste 1 (ocorrência do evento $s_{n1,ok}$ ou $s_{n1,nok}$). Se o sensor passar no teste, a especificação bloqueia o comando para o manipulador retornar à posição de descanso e permite apenas o movimento para o próximo nível $t_{2,i}$. Se o sensor

reprovar no teste, somente o comando para retornar ao descanso é habilitado $r_{1,i}$, não permitindo que o teste prossiga. Além disso, após retornar ao descanso, o manipulador deve deixar o sensor na caixa “nok” (l_{nok}).

A quarta e última especificação, apresentada na Figura 3d, define o procedimento para o teste no nível 2. Após posicionado neste nível (evento $t_{2,f}$), o manipulador deve aguardar o resultado do teste. Dessa forma, o comando de retorno ao descanso é desabilitado até que a rotina de teste é completada (evento $s_{n2,ok}$ ou $s_{n2,nok}$). Após o teste, o manipulador deve retornar ao descanso, uma vez que não há outros níveis a serem testados e, com isso, o comando $r_{2,i}$ para iniciar o retorno é habilitado independente do resultado do teste. No entanto, se o teste falhar ($s_{n2,nok}$), apenas o comando l_{nok} para deixar o sensor na caixa “nok” é habilitado. Por outro lado, caso o teste passar $s_{n2,ok}$, apenas o comando l_{ok} para deixar o sensor na caixa “ok” é habilitado.

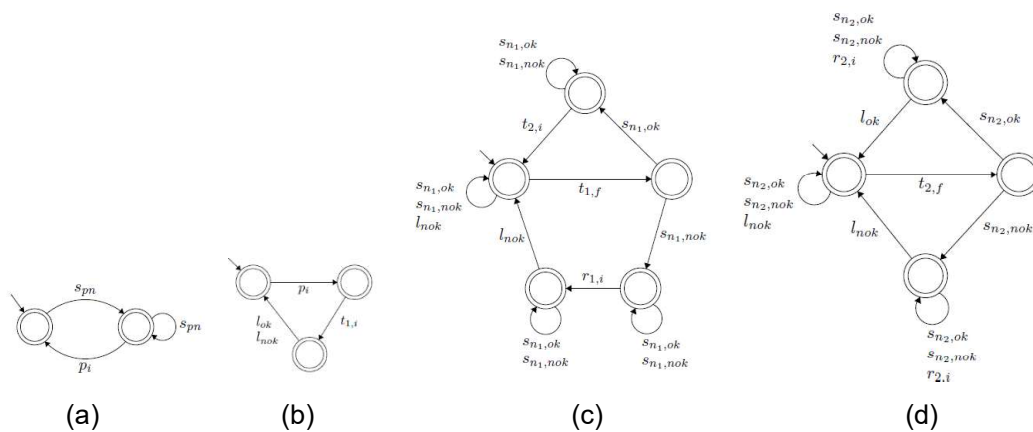


Figura 3. Especificações para o sistema.

3.2. Controlabilidade de eventos

Para fins de controle, é importante saber se um evento é de fato controlável, ou seja, se ele é passível de intervenção por via controlador, ou se, do contrário, ele não pode ser diretamente proibido. Esse conceito é um dos fundamentos da TCS de SEDs [Wonhan, 2002]. Para a aplicação em questão, apenas $\{p_i, l_{ok}, l_{nok}, t_{1,i}, t_{2,i}, r_{1,i}, r_{2,i}\}$ são controláveis.

4. Síntese, Implementação e supervisão do sistema de controle

Essa seção apresenta detalhes sobre a síntese do controlador para a aplicação proposta, bem como detalhes sobre a implementação física do sistema.

4.1. Composição e Síntese

O modelo do sistema e suas especificações foram construídos na ferramenta *Supremica*® [Akesson, Fabian, Flordal e Malik 2006]. Essa ferramenta incorpora as funcionalidades de modelar, compor e sintetizar o controlador. Para a aplicação proposta, a planta, denotada por G , é derivada da composição síncrona dos modelos apresentados na Figura 2. Essa composição possui 10 estados. Ainda, o modelo da especificação, denotado por E , resulta da composição das especificações da Figura 3. O autômato que representa E possui 120 estados.

A partir de G e E , é possível se obter o modelo $K = G \parallel E$, representando o comportamento desejado para o sistema G quando sob influência de E . O modelo K pode então ser usado como entrada para a síntese de um controlador, denominado S , que restringe G reconhecendo a impossibilidade de desabilitar diretamente eventos não controláveis. O comportamento modelado por S é, por construção da TCS, considerado minimamente restritivo em relação à planta e aos requisitos, não bloqueante, e alinhado com as especificações do problema.

4.2. Implementação e supervisão

A implementação física do sistema de controle foi realizada a partir de um CLP TPW03-30HRA®. A escolha do CLP foi feita visando a comunicação com um sistema supervisão, utilizando o protocolo Modbus RTU [Goldenberg e Wool 2013]. A linguagem *Ladder* foi utilizada para programar o CLP. O controlador S foi convertido para a linguagem *Ladder*, criando-se blocos que são ativados/desativados conforme o modelo formal obtido na síntese monolítica.

Para supervisionar a rotina de testes automatizada, uma aplicação SCADA® (Supervisory Control and Data Acquisition) foi desenvolvida utilizando o software Elipse E3® para desenvolvimento da interface [Dnaeels e Slater 1999], [Becker, Pardi e Pereira 1999]. A comunicação entre o software supervisão e o CLP é feita a partir do protocolo Modbus® RTU. Na camada física, o CLP utiliza o protocolo serial RS 485. Dessa forma, há um conversor RS485/USB para a conexão serial entre o CLP e o computador utilizado na supervisão. A Figura 4 ilustra o procedimento de supervisão do sistema.

Essa interface permite, além da monitoração on-line, a contagem de sensores aprovados, descartados e totais. Também é possível visualizar o *layout* da planta fabril, com o manipulador robótico, tanque aquático de teste, esteiras de entrada e saída de peças (sensores a serem testados). Por fim, também é possível intervir via SCADA sobre o processo físico, uma vez que ele implementa um formato de comunicação dual.

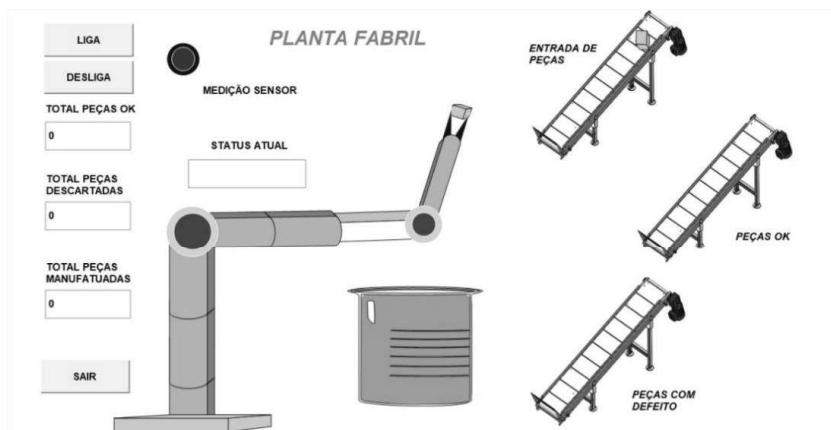


Figura 4. Interface de Controle Supervisório do sistema.

4.3. Práticas de testes com o framework de controle e supervisão

Foram realizados ensaios para ilustrar o comportamento do sistema sob controle do CLP. Usando botões para simular os sensores de entrada e observando as saídas do CLP, foi possível observar as ações permitidas e bloqueadas do manipulador através de possíveis sequências de eventos. A primeira sequência testada foi $\{p_i, r\}$ resultando no evento p_i

bloqueado pois necessita da antecedência do evento s_{pn} indicando presença de nova peça na caixa “teste”. A segunda sequência testada foi $\{s_{pn}, p_i, r, s_{pn}, p_i\}$ em que o evento p_i foi novamente desabilitado pelo controlador após a primeira transição desta cadeia, restringindo a busca por uma nova peça antes da atual ser processada. Outras cadeias de eventos foram aplicadas (como o teste em cada nível do tanque) e nenhum comportamento excedeu às condições esperadas sob controle.

5. Conclusão

A aplicação desse framework traz benefícios imediatos, como redução significativa no tempo do teste, redução de erros na validação, monitoramento on-line do sistema, maior confiabilidade nos resultados, dinamismo na confecção de relatórios com número de sensores aprovados e rejeitados, entre outros.

Trabalhos futuros pretendem implementar a comunicação do sistema via Modbus® TCP, na qual possibilitará o armazenamento de dados em rede e contribuirá com a tendência da indústria 4.0.

Referências

- K. Tahera, C. Earl e C. Eckert, “The role of testing in the engineering product development process”, Proceedings of TMCE 2012, Maio, 2012.
- K. Karhu, T. Repo, O. Taipale e K. Smolander, “Empirical Observations on Software Testing Automation,”2009 International Conference on Software Testing Verification and Validation, Denver, CO, 2009, pp. 201- 209.
- C.G. Cassandras e S. Lafortune. “Introduction to Discrete Event Systems”, 2 ed, 2009.
- W. Wonhan. “Notes on Discrete Event Systems”, Universidade de Toronto, 2002.
- L.A. Bryan e E.A. Bryan. “Programmable Controllers: Theory and Implementation”. Industrial Text Company, 2 ed, 1997.
- K. Akesson, M. Fabian, H. Flordal, e R. Malik, “Supremica—na integrated environment for verification, synthesis and simulation of discrete event systems”, 8th Int. Workshop on Discrete Event Systems, 384–385, 2006.
- N. Goldenberg e A. Wool, “Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems”. Journal of Critical Infrastructure Protection, pp. 63-75, 2013.
- A. Dnaeels e W. Slater, “What is SCADA?”. International Conference on Accelerator and Large Experimental Physics Control Systems, It´alia, 1999.
- L. Becker, W. Pardi, C. Pereira, “Integrated Object- Oriented Environment for the Design of Supervisory Software for Real-Time Industrial Automation Systems”. Workshop on Object-Oriented Real-Time Dependable Systems, 1999.
- C. Caetano, "Melhores Práticas na Automação de Testes." Revista Engenharia de Software Magazine, 5ª edição, p42-47, 2008.
- J.M. Rosário, “Robótica Industrial I Modelagem, Utilização e Programação”, Vol. 1, Editora Baraúna, 2010.