Uma Arquitetura Altamente Disponível Aplicada a Sistemas de Controle Embutidos de Tempo Real[1]

Cesar Ossamu Ida, **Taisy Silva Weber**

Instituto de Informática - Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15064 – 91501-970 – Porto Alegre – RS

{cesarida, taisy}@inf.ufrgs.br

Resumo. *Uma arquitetura baseada em redundância de controladores para aumentar a segurança (safety) e disponibilidade de sistemas embutidos de tempo real é apresentada. Dois controladores (de prateleira) processam os mesmos dados de entrada e os dados computados são comparados, como forma de detecção de erros. Quando um erro é detectado, uma rotina de diagnóstico tenta identificar sua localidade. Isto pode demorar alguns ciclos de controle, o que viola a propriedade de tempo real da aplicação controlada; entretanto, a violação pode ocorrer na classe de aplicações embutidas sendo considerada, desde que não cause o colapso do processo controlado.*

Abstract*. An architecture based on controller redundancy for increased embedded real-time systems safety and availability is presented. Two COTS controllers process the same inputs and the computed data is compared as a means to detect errors. When an error is detected, a diagnosis routine tries to identify its locality. This may take some control cycles, which violates the real-time property of the controlled application; however, this situation is allowed to happen for the embedded systems we consider as long as it does not cause the failure of the controlled process.*

1. Introdução

Um dos maiores obstáculos à ampla utilização de sistemas tolerantes a falhas em computadores embutidos de tempo real é o alto custo inerente à redundância empregada, seja ela temporal ou espacial. Durante alguns anos, sua aplicação foi restrita a sistemas críticos de custos altíssimos, justificados pelos potenciais prejuízos causados por falhas nestes sistemas. Entretanto, fatores como a redução nos preços dos componentes de hardware para computadores, a redução do próprio tamanho desses componentes e produção em larga escala, contribuíram para sua maior utilização em diferentes aplicações comerciais e industriais. Atualmente, técnicas de tolerância a falhas têm sido utilizadas em sistemas de controle como plantas industriais, automóveis, usinas de eletricidade, plataformas de petróleo, controle de tráfego, navegação, controle de trens, sistemas autônomos, aplicações militares, edifícios inteligentes, etc..

Em algumas destas aplicações, um defeito no sistema pode colocar em risco a vida de pessoas, danificar equipamentos ou levar a prejuízos monetários bastante elevados [Laprie 1998], o que justifica altos investimentos em técnicas de tolerância a falhas para manter estes sistemas funcionando, mesmo na presença de falhas. Estas aplicações envolvem os sistemas

[1] Este trabalho é parcialmente financiado pelo projeto CT-Petro, desenvolvido em cooperação entre a Universidade Federal do Rio Grande do Sul, UNISINOS e a empresa Altus Sistemas de Informática.

chamados de "ultra-seguros" (*ultra-dependable*), bem como os sistemas "altamente-seguros" (*highly-dependable*) [Suri 1995]. Por outro lado, algumas destas aplicações possuem requisitos de confiabilidade, disponibilidade e segurança mais baixos que as primeiras e não são consideradas críticas. Estas, são as aplicações "altamente disponíveis" (*highly-available*). Este tipo de aplicação criou um novo nicho para o emprego de sistemas tolerantes a falhas, e se caracteriza pelo custo mais baixo com relação aos sistemas "ultra-seguros", requisitos temporais de resposta da ordem de centenas de milissegundos, execução cíclica (a maioria), computadores embutidos, e normalmente possuem um operador remoto que monitora o sistema. Nestas aplicações, a parada do sistema durante intervalos de tempo curtos são aceitáveis, desde que o mesmo seja colocado em um estado seguro em situações de falha. As aplicações "altamente-disponíveis" controladas por CLPs (Controladores Lógico-Programáveis)[2] são o foco deste trabalho.

Ao mesmo tempo em que se busca maior disponibilidade e segurança, o trabalho desenvolvido objetiva, também, manter os custos do sistema baixos, visto que um dos maiores obstáculos à ampla utilização comercial de sistemas tolerantes a falhas é o seu elevado custo. Por este motivo, a duplicação de CLPs é bastante utilizada comercialmente quando comparada a outros esquemas de redundância, como a triplicação e a quadruplicação. Outra maneira de se manter os custos do sistema baixos - além da utilização de apenas um CLP redundante - é através da utilização de componentes de prateleira (COTS). No sistema desenvolvido, todos os componentes de hardware (UCPs, redes de comunicação, interfaces de comunicação, etc..) e software (sistema operacional, drivers) utilizados são COTS. Este trabalho está sendo desenvolvido dentro do projeto CT-Petro, o qual visa o desenvolvimento de um sistema embutido de tempo real para o controle de plataformas de petróleo *off-shore*, que atenda a requisitos de confiabilidade e disponibilidade das licitações para plataformas da Petrobrás.

Este artigo trata da especificação e desenvolvimento de uma arquitetura de hardware baseada na duplicação de CLPs, que tem por objetivo adicionar características de tolerância a falhas a um sistema de controle embutido de tempo real. É apresentada, também, uma estratégia de tolerância a falhas implementada em software, complementar à arquitetura de hardware redundante. Esta, é baseada na técnica de redundância dinâmica [Pradhan e Banerjee 1996] e detecção de erros por comparação. A seção 2 do artigo descreve a arquitetura geral do sistema e do CLP empregado; a seção 3 o modelo de falhas; a seção 4 a estratégia de tolerância a falhas; a seção 5 a plataforma utilizada na implementação; e na última seção é feita uma análise geral do sistema.

2. Arquitetura de Hardware do Sistema

O sistema desenvolvido pelos autores, denominado HARTS (Highly Available Real Time System) é dividido em 3 subsistemas: supervisão, controle e campo (fig. 1).

O subsistema de supervisão liga o CLP aos computadores dos usuários através de uma rede de supervisão padrão Ethernet, e permite que os operadores monitorem o sistema. O subsistema de campo liga o CLP aos dispositivos de E/S, através de uma rede de campo padrão Profibus e módulos intermediários de E/S. O subsistema de controle é composto pelos próprios CLPs, que são interligados através de um canal de sincronização padrão Ethernet,

[2] O CLP é uma UCP, convencional ou não, somada a interfaces para periféricos e redes específicos para ambiente industrial [Jones 1983]

utilizado para troca de dados referentes ao controle de redundância. Se ocorrer falha nesta rede, os CLPs utilizam a rede de supervisão como um canal alternativo para troca de informações referentes ao diagnóstico da falha. Elimina-se, deste modo, a necessidade de duplicação da rede de sincronização.

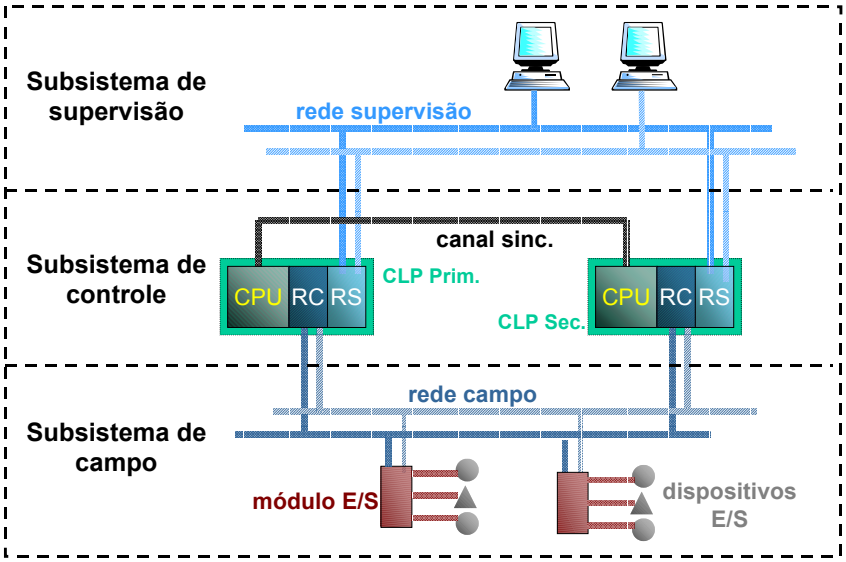

Figura 1. Arquitetura de hardware do sistema HARTS

Os CLPs são nomeados primário e secundário apenas como forma de referência. Ambos executam todas as tarefas em paralelo e possuem programas idênticos. Realizam a leitura dos dados de entrada (adquiridos através da rede de campo), os processam e geram novos valores que são enviados para os equipamentos de campo. Os CLPs comparam seus resultados antes de enviá-los para os equipamentos de campo, como forma de detecção de erros. Uma vez que um erro é detectado, rotinas de diagnóstico, recuperação e reintegração são executadas e, se necessário, uma reconfiguração automática de recursos é realizada para manter o sistema em funcionamento, sem afetar sua disponibilidade.

2.1 Arquitetura do CLP

Cada CLP do sistema HARTS é composto por 3 módulos: Módulo UCP, Módulo de Campo (MC) e Módulo de Supervisão (MS). Estes executam tarefas específicas e possuem baixa capacidade de processamento, sendo compostos por um microcontrolador, fonte de energia elétrica e ligações com suas respectivas redes de comunicação. Todos os módulos (de cada CLP) são interligados através de um barramento, através do qual comunicam-se entre si - o protocolo de comunicação entre módulos utilizado no projeto CT-Petro é o GBL, um protocolo mestre/escravo desenvolvido por, e de propriedade da Altus Sistemas de Informática. Por questões de simplificação, este barramento é assumido como livre de falhas. No HARTS, a falha permanente de qualquer um destes módulos causa a desativação de todo o CLP; neste caso, o sistema pode ser desligado ou continuar funcionando com apenas um CLP.

O Módulo UCP é o principal, controla as atividades do CLP, executa o programa de controle e gerencia as atividades relacionadas à tolerância a falhas. Possui ligação a um canal de sincronização padrão Ethernet (que a interliga a outro CLP), além de ligação ao barramento (GBL) que a interliga aos outros módulos.

O Módulo de Supervisão realiza comunicação com os usuários do sistema através de computadores e rede de supervisão, recebendo as requisições, numerando-as e repassando-as para a UCP. Possui ligação ao barramento GBL e à rede de supervisão (padrão Ethernet), que pode ser redundante (duplicada). As requisições de usuário, geradas através dos computadores de supervisão, fogem à execução cíclica do sistema, pois não se pode prever o momento em que um usuário gera uma requisição. Por este motivo, estas requisições são consideradas dinâmicas e podem causar inconsistências entre CLPs se atendidas em qualquer momento, ou em apenas um dos CLPs. No HARTS, estas requisições são devidamente tratadas pelos módulos de supervisão de maneira consistente e determinística, como é apresentado na seção 3 deste artigo.

O Módulo de Campo perfaz o papel de mestre de rede de campo e realiza comunicação e varredura dos módulos de E/S. Possui ligação à rede de campo (padrão Profibus), além de ligação ao barramento GBL. O HARTS permite que a ligação à rede de campo seja duplicada.

3. Modelo de Falhas

O HARTS suporta falhas de hardware, partindo da premissa que as mesmas sejam independentes e simples. Independente significa que a falha não afeta os dois CLPs[3] ao mesmo tempo (falhas correlacionadas); simples significa que nenhuma outra falha ocorre até que a primeira seja recuperada.

Os tipos de falhas toleradas, quanto à sua duração, são as falhas temporárias e permanentes ocorridas na UCP (Unidade Central de Processamento). As falhas temporárias são aquelas que ocorrem por curtos instantes de tempo, como um eventual ruído. Se não houver como tratar o erro sem parar o sistema, a simples reinicialização do mesmo pode ser suficiente para resolvê-lo. Algumas falhas temporárias são periodicamente recorrentes e, neste trabalho, são consideradas permanentes. Falhas permanentes são falhas de hardware. Não há como recuperar o componente que sofreu falha permanente sem a substituição do mesmo.

Assume-se que falhas no canal de comunicação (sincronização) entre os CLPs (interface ou *link* de comunicação) sejam falhas do tipo colapso (*crash*). Falhas de projeto, implementação e de software não são tratadas, pois necessitam de mecanismos de detecção e tratamento diferentes dos utilizados neste trabalho. Falhas nos dispositivos de E/S não são consideradas, assume-se que os mesmos sempre funcionam corretamente. Os códigos executáveis dos programas são armazenados em memória ROM (Read Only Memory) e, portanto, imunes a interferências e corrupção de dados.

4. Estratégia de Tolerância a Falhas

O CLP, sob uma visão bastante simplificada, lê um conjunto de dados de entrada do ambiente (através dos módulos de E/S e sensores), processa esses dados e gera um conjunto de resultados que são utilizados neste ambiente (através dos módulos de E/S e atuadores). Estas atividades devem ser executadas em um tempo finito, predefinido, porém, variável de acordo com cada aplicação.

[3] Os CLPs são fisicamente e eletricamente isolados um do outro e, portanto, apresentam falhas independentes, dentro dos limites cabíveis

No HARTS, as UCPs dos dois CLPs executam os mesmos programas em paralelo: o sistema operacional e o programa de controle devem ser idênticos e determinísticos. Os resultados dos programas de aplicação (controle) são comparados bit-a-bit a cada ciclo, como forma de detecção de erros. O método de comparação permite a detecção rápida de erros (pequeno atraso), fator importante para contenção de seus efeitos nocivos na UCP, de modo que não se espalhem pelo sistema.

Para evitar que seja um ponto único de falha, o comparador é distribuído em ambas UCPs. Estas, comunicam-se através de troca de mensagens. Após a comparação bem sucedida de resultados, ambas UCPs os enviam para os mestres de rede de campo, que por sua vez os repassam para os módulos de E/S. Os equipamentos de rede de campo também poderiam comparar os resultados antes de utilizá-los, para garantir que nenhum erro tenha ocorrido entre o momento da comparação nas UCPs, transmissão através da rede de campo e sua recepção nos equipamentos. A adoção da solução de comparar resultados pelos equipamentos de campo depende do método de sincronização empregado nos mesmos, mas não é essencial para o funcionamento da estratégia de tolerância a falhas. No HARTS a comparação é realizada na UCP antes do envio dos dados, mas não é realizada pelos equipamentos de campo no momento de sua recepção.

Após a inicialização do sistema, as UCPs fazem a leitura dos dados de entrada, contidos nos respectivos mestres de rede de campo - assume-se que o sistema inicia seu funcionamento livre de falhas. Em seguida, o programa de aplicação é executado, o qual gera um conjunto de dados de saída, que não são imediatamente enviados para os equipamentos de rede de campo. Ao contrário, o procedimento de comparação de resultados é acionado. Se os valores gerados pelos programas de aplicação das duas UCPs forem iguais, o sistema continua seu fluxo de execução normalmente, atendendo às requisições de usuários e iniciando o ciclo seguinte. Se os valores não forem idênticos, significa que ocorreu alguma falha e os procedimentos relacionados à tolerância a falhas são executados. Estes procedimentos são descritos na seção seguinte.

4.1 Tarefas de Tolerância a Falhas

Após a execução do programa de aplicação, são executadas as tarefas relacionadas à tolerância a falhas, quais sejam a comparação, diagnóstico, recuperação e reintegração.

A primeira delas a ser executada é a comparação, para detecção de erros. Se nenhum erro for detectado, os valores de saída são enviados aos dispositivos de saída (atuadores), as requisições de usuário (se existirem) são processadas e o ciclo inicia-se novamente. Por outro lado, se algum erro for detectado no momento da comparação, os dados de saída do ciclo anterior são mantidos nos módulos de Rede de Campo e de E/S, e o fluxo de execução da UCP é desviado para a rotina de diagnóstico. O diagnóstico deve apontar em qual das UCPs a falha ocorreu; dependendo de seu resultado, a rotina de recuperação é invocada na UCP em falha, enquanto a UCP livre de falhas inicia o ciclo seguinte e mantém o sistema operacional. Se a UCP em falha for recuperada, é reintegrada ao sistema e seu estado é sincronizado novamente com o estado da UCP não falha – a reintegração também é realizada quando a UCP é substituída. Se não for possível recuperar o componente em falha, ele é desligado do sistema, que continua seu funcionamento com apenas um CLP controlando o processo; neste caso, não é mais possível realizar a detecção de erros através da comparação, mas mecanismos auxiliares de detecção podem ser empregados.

Se o diagnóstico das UCPs não for satisfatório, ou seja, se não for possível determinar em qual das UCPs ela ocorreu, então não é possível manter o sistema em funcionamento: ambas UCPs são suspeitas e devem ser desligadas para que não executem algum comportamento errôneo. O sistema entra em um estado seguro, livre de falhas. Este processo pode ser melhor visualizado na fig. 2:

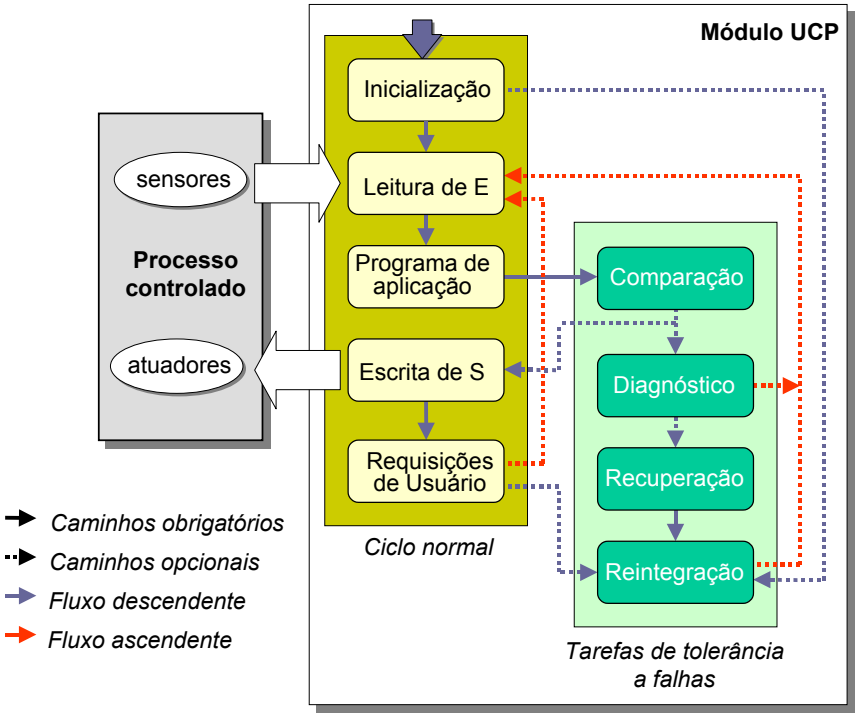

Figura 2. Seqüência de tarefas da UCP

4.2 Requisições de Usuário

As requisições de usuário são fonte potencial de inconsistência entre UCPs se processadas em momentos diferentes. No HARTS, o esquema utilizado para sincronização de módulos de supervisão é similar ao líder seguidor [Brasileiro 1996]. Uma requisição de usuário é enviada pelo computador de supervisão para os dois módulos de supervisão. Assim que o Módulo de Supervisão Líder (MSL) recebe a mensagem, este a ordena e a repassa, juntamente com sua ordem, para o Módulo de Supervisão Seguidor (MSS). Este, responde com uma mensagem de confirmação (*ack*). Desta forma, os dois módulos possuem filas idênticas de requisições ordenadas, prontas para serem enviadas para a UCP. Este processo é ilustrado na fig. 3.

As UCPs, após executarem o programa de controle e a comparação de resultados, verificam se existem requisições de usuários nos módulos de supervisão. Estes, enviam suas respectivas filas de requisições ordenadas para suas respectivas UCPs, as quais verificam a igualdade das mesmas (comunicam-se entre si) e processam as requisições. As respostas das UCPs são enviadas para seus respectivos Módulos de Supervisão (módulo contido no mesmo CLP da UCP). O MSL e MSS comparam as respostas recebidas e, se forem iguais, apenas o MSL as repassa para o computador de supervisão.

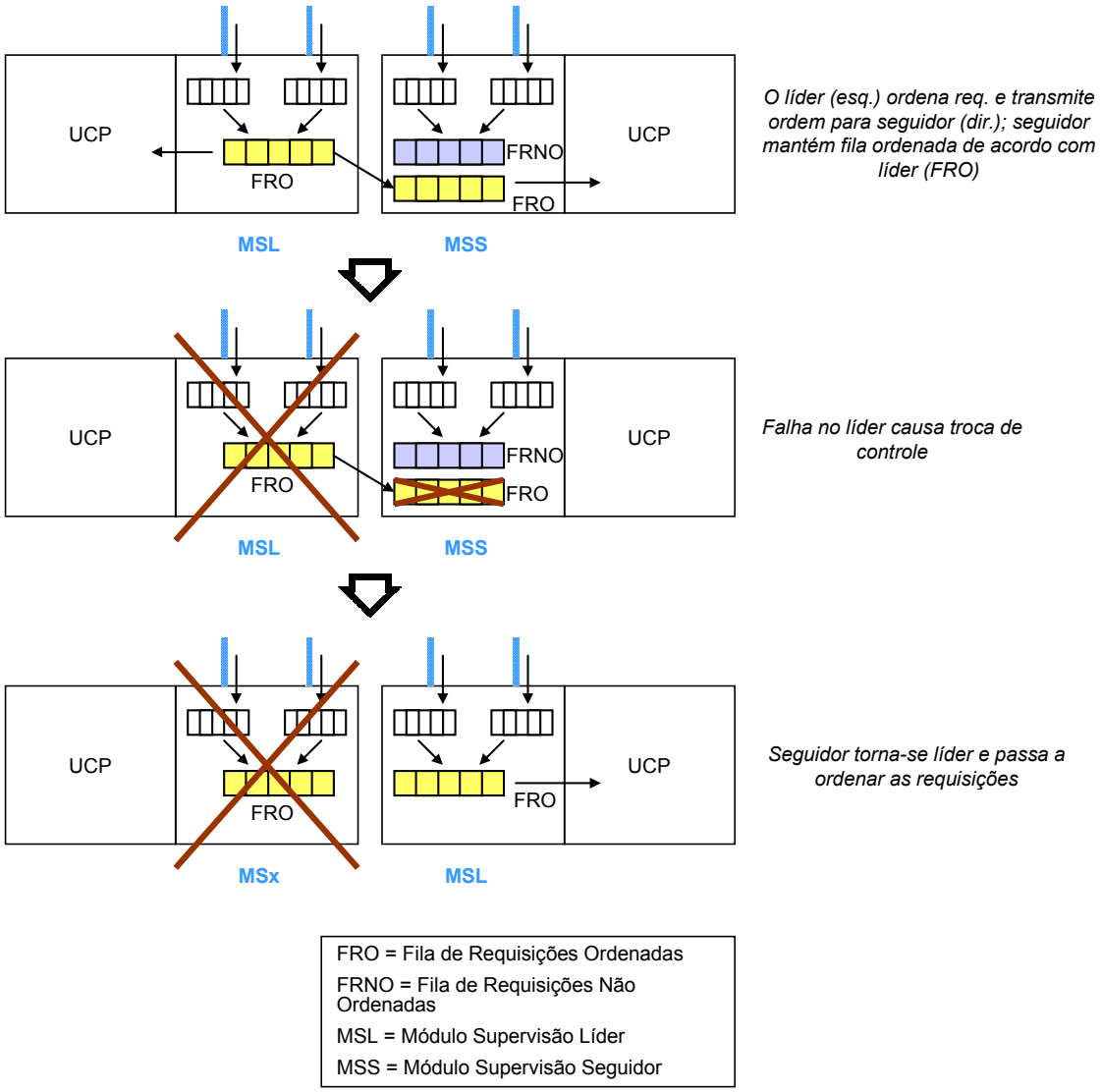

Figura 3. Troca de controle entre MS Líder e MS Seguidor

5. Plataforma

A plataforma do HARTS é composta por equipamentos da série Ponto da empresa Altus Sistemas de Informática, tais como UCPs, Módulos de Campo e Supervisão, mestres e escravos Profibus, FPGAs mestres e escravos de controle do barramento GBL, fontes de alimentação e módulos de E/S. Para implementação da estratégia de tolerância a falhas por software, foi utilizado o sistema operacional de tempo real QNX.

6. Conclusões

Este artigo apresentou o sistema HARTS, composto de uma arquitetura de hardware baseada na duplicação de CLPs e uma estratégia de tolerância a falhas implementada por software sobre o hardware redundante. O HARTS visa controlar processos industriais e plataformas de petróleo com restrições temporais de execução e requisitos de alta disponibilidade e segurança e que empregam computadores embutidos.

A tolerância a falhas do HARTS é alcançada através da detecção de erros nos CLPs por comparação de resultados, diagnóstico e recuperação de falhas, reintegração de CLP e troca de controle. Para o programador da aplicação, os recursos relacionados à tolerância a falhas são transparentes, inclusive a arquitetura com CLPs redundantes. Desta forma, O HARTS oferece altos níveis de disponibilidade e segurança no controle de processos, sem adicionar complexidade à sua programação e utilização, ao mesmo tempo em que permite configuração de acordo com os requisitos de funcionamento da aplicação.

A execução do diagnóstico ocupa alguns ciclos de controle, durante os quais as UCPs tornam-se indisponíveis para o resto do sistema. Nesta situação, os módulos de campo repetem os valores de saída dos ciclos anteriores nos dispositivos de E/S, até que uma ou as duas UCPs tornem-se disponíveis novamente ou que ocorra um *time-out*. Isto pode ocasionar violação da propriedade de tempo real da aplicação, mas não deve causar seu colapso. De acordo com experiência prática [Cunha 2001], eventuais erros nos valores de saída não são dramáticos, desde que os valores subseqüentes sejam corrigidos, observação válida inclusive para alguns sistemas de tempo real rígido (*hard real-time*).

Através deste trabalho, foi possível constatar a viabilidade de utilização de componentes de hardware e software do tipo COTS na implementação de sistemas de controle embutidos de tempo real com requisitos de alta disponibilidade e segurança, desde que devidamente adaptados. O HARTS é voltado para plataformas de petróleo, mas pode ser empregado em aplicações cíclicas que utilizem CLPs e que possam ter algumas saídas repetidas durante o seu funcionamento.

Referências Bibliográficas

[Brasileiro 1996] Braileiro, F.V. et al. Implementing fail-silent nodes for distributed systems. **IEEE Transactions on Computers**, Los Alamitos, v.45, n.11, p.1226-1238, Nov. 1996.

[Cunha 2001] Cunha, J.C. et al. A study of failure models in feedback control systems. In: International Conference on Dependable Systems and Networks, Goteburg, 2001. **Proceedings.** Los Alamitos: IEEE Computer Society, Jul. 2001.

[Jones 1983] Jones, C.T.; Bryan, L.A. **Programmable controllers – concepts and applications**. USA: International Programmable Controls, 1983. 329p.

[Laprie 1998] Laprie, J.C. Dependability of computer systems: from concepts to limits. In: IFIP International Workshop on Dependable Computing and its Applications. Johannesburg, South Africa, 1998. **Proceedings.** [S.l.: s.n.], Jan. 1998.

[Pradhan e Banerjee 1996] Pradhan, D.K.; Banerjee, P. Fault-tolerant multiprocessor and distributed systems: principles. In: PRADHAN, D. K., **Fault-tolerant computer system design**. Upper Saddle River: Prentice Hall, 1996. 550p. cap.3, p.135-235.

[QNX 2003] QNX Developer's Network. Disponível em <http://www.qnx.com/developer/docs> Acesso em: abril de 2003.

[Suri 1995] Suri, N.; Walter, C.J e Hugue, M.M. **Advances in Ultra-Dependable distributed systems**. cap.1. Los Alamitos: IEEE Computer Society, 1995. 467p.