



# A Graphical Development Environment applied to the Design and Debugging of an Application Framework for an Integrated Speed Governor and Voltage Regulator System

Carlos Matias Billo  
Reivax Automação e Controle  
[cmb@reivax.com.br](mailto:cmb@reivax.com.br)

João Marcos Castro Soares  
Reivax Automação e Controle  
[jms@reivax.com.br](mailto:jms@reivax.com.br)

Tiago César Busatta  
Reivax Automação e Controle  
[tcb@reivax.com.br](mailto:tcb@reivax.com.br)

Daniel Maurício Kamers  
Reivax Automação e Controle  
[dmk@reivax.com.br](mailto:dmk@reivax.com.br)

Rafael Bertolini Paiva  
Reivax Automação e Controle  
[rpb@reivax.com.br](mailto:rpb@reivax.com.br)

Gabriel Aurélio de Oliveira  
Reivax Automação e Controle  
[gao@reivax.com.br](mailto:gao@reivax.com.br)

Rodrigo Mussatto  
Reivax Automação e Controle  
[rm@reivax.com.br](mailto:rm@reivax.com.br)

**Abstract** — This paper presents the design, implementation, use and maintenance of an application framework for an integrated speed governor and voltage regulator system, through the use of an own-made graphical development environment, with advanced editing and debugging features. The project requirements are evidenced and the results and benefits of the use of the proposed architecture are presented, based on the experience of REIVAX in the supply of voltage regulators and speed governors.

**Index Terms** — framework, graphical programming, regulator, IEC61131-3.

## I. INTRODUÇÃO

No controle da geração de usinas hidrelétricas, o uso de reguladores de tensão e velocidade integrados em um único equipamento tem mostrado algumas vantagens sobre os sistemas individuais, como facilidade operacional, redução de custos, redução da complexidade do sistema, facilidades para o comissionamento e incremento de qualidade do controle [1].

Na programação do software aplicativo destes reguladores é freqüente a necessidade de adaptação de interfaces de entrada e saída e modificações de algumas lógicas de acionamento e automatismo. Em poucos casos torna-se necessária alguma mudança no núcleo básico do regulador ou em alguma estrutura de controle mais complexa. Além disso, na manutenção dos reguladores são essenciais ferramentas eficientes para a monitoração e edição de sinais em tempo de execução, que permitam diagnósticos rápidos e confiáveis, reduzindo os custos de projeto, instalação e comissionamento das máquinas.

Estas perspectivas evidenciam o benefício da criação de um framework aplicativo para um sistema integrado que possibilite:

- Integração modular dos algoritmos do regulador de tensão e regulador de velocidade;
- Padronização da arquitetura da aplicação e das interfaces para adaptação do programa;
- Programação intuitiva e manutenção simplificada (acréscimo e inibição de funcionalidades);
- Facilidade de depuração da aplicação.

Um framework é uma aplicação “semicompleta” e reutilizável que pode ser adaptada para atender aplicações específicas. Para concretizar a criação e viabilizar o uso de um framework nos projetos de reguladores de tensão e velocidade, percebeu-se a importância de se dispor de uma ferramenta de desenvolvimento robusta e eficiente, e com interface centrada no usuário. Esta percepção advém da compreensão, adquirida com o tempo, do impacto nos custos operativos por deficiências no processo de “Interação Humano-Computador” (IHC) associadas a tempo de aprendizado longo para o usuário, linguagens de programação pouco intuitivas e má ergonomia.

## II. ARQUITETURA

### A. Regulador Integrado de Tensão e Velocidade

O desempenho da geração de uma central elétrica é definido pelos equipamentos primários de controle - sistemas de regulação de velocidade e excitação. Isto se dá pela necessidade de partidas suaves, sem sobreexigências sobre o equipamento, na sincronização rápida e precisa, no controle das grandezas básicas obedecendo as solicitações do Centro de Operações do Sistema (COS) ou Controle Automático de Geração (CAG), na correta divisão de cargas entre unidades e enfim, em um grande número de tarefas que, na inexistência de dispositivos realimentados de controle, se inviabilizariam [1].

A implementação destes dois controladores em um mesmo equipamento aliada à definição de um programa aplicativo padrão, através do framework criado, traz vantagens importantes quanto aos aspectos estruturais e funcionais destes controladores:

- Redução de custos: uso de uma única CPU e IHM, redução de painéis, redução de interfaces e cablagens entre os controladores, possibilidade de compartilhamento de transdutores e dispositivos de acionamento e comando;
- Redução de indisponibilidade das máquinas por problemas associados aos controles, pela diminuição de componentes e conseqüente aumento de

confiabilidade;

- ❑ Construção de leis de controle e limitação com recursos para atuar nas duas malhas, trocando informações diretamente via software;
- ❑ Padronização do núcleo do programa aplicativo do regulador e definição clara das interfaces utilizadas para adaptação de aplicações específicas, incluindo a definição de lógicas de automatismo da usina incorporadas ao regulador;
- ❑ Organização da documentação relativa ao software, com ganhos evidentes na compreensão e legibilidade do programa aplicativo, bem como na rapidez de adaptação da aplicação;

Todo o controle e lógicas de acionamento principais do regulador são executadas por uma CPU baseada em processador X86 de 32 bits, com recursos associados de hardware e software bastante poderosos.

## B. Hardware e Software

### 1) Plataforma de Hardware

A plataforma de hardware, denominada CPX, utilizada nas aplicações é baseada em uma placa de aquisição, denominada MX12, associada a uma CPU x86 conectada através de um barramento padrão PC/104.

Na sua configuração padrão a placa de aquisição pode ter 16 entradas e 8 saídas analógicas, 48 entradas/saídas digitais configuráveis, sendo três rápidas que podem ser utilizadas para obter medição precisa de tempo.

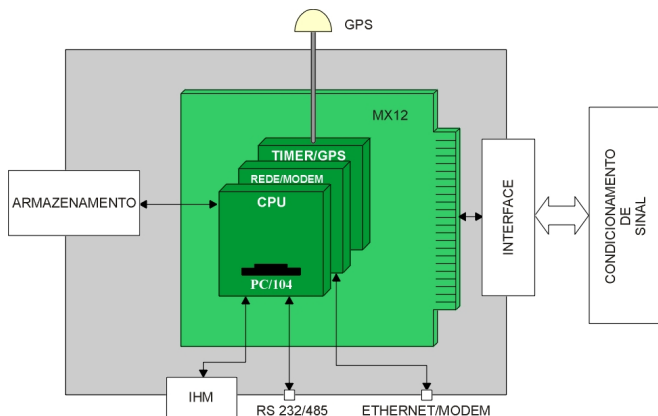


Fig. 1. Hardware simplificado do CPX.

### 2) Software Básico Embarcado

O software embutido no CPX é baseado num RTOS (Sistema Operacional de Tempo Real) preemptivo, denominado Software Básico Embarcado (SBE), e é o responsável pela execução das aplicações programadas pelo usuário no ambiente de desenvolvimento. É o SBE que

executa o framework aplicativo do regulador integrado. Suas principais características são:

- ❑ Execução em modo protegido, 32 bits, em plataforma de hardware x86 ou compatível;
- ❑ Suporte à multitarefa, permitindo o uso de tarefas cíclicas para execução de mais de um programa, tornando possível assim, a otimização de uso da CPU;
- ❑ Desempenho para aplicações de tempo real;
- ❑ Conectividade em rede TCP/IP;
- ❑ Execução de aplicações (algoritmos) baseados na interconexão de blocos;
- ❑ Biblioteca com mais de 150 tipos de blocos funcionais, contendo alguns blocos especialmente desenvolvidos para aplicação em controle de usinas;
- ❑ Suporte a monitoração de sinais, visualização e edição de parâmetros em tempo de execução;
- ❑ Registro de sinais através de lógica de disparo pré programada;
- ❑ Gerenciamento dos arquivos presentes nos discos do módulo;
- ❑ Cadastro de usuários, senhas e permissões de acesso;
- ❑ Suporte para interface com IHMs locais e remotas;
- ❑ Drivers para protocolos de comunicação padrão da indústria;
- ❑ Flexibilidade de adaptação e uso de periféricos de entrada/saída através de drivers de dispositivo;

### 3) Modelo de programação

Para a criação, manutenção e adaptação do framework, o ambiente de desenvolvimento oferece uma interface gráfica de edição com linguagem de programação orientada a blocos de função, FBD, baseada na IEC61131-3.

A padronização da linguagem de programação, e em particular a adequação ao padrão IEC61131-3, favorece [2]:

- ❑ Diminuição do tempo de aprendizado de linguagens específicas;
- ❑ Melhoria da qualidade dos programas;
- ❑ Reutilização;
- ❑ Independência da plataforma de execução;

A IEC61131-3 define 5 linguagens, das quais a FBD (Function Blocks Diagrams) é a mais apropriada para modelagem de sistemas com grandes fluxos de sinal, tal como sistemas de controle.

Usando a linguagem de programação oferecida pelo ambiente o usuário constrói o programa aplicativo que será embarcado e executado pelo SBE. O modelo conceitual de software para desenvolvimento de aplicações é mostrado na Figura 2.

O framework desenvolvido é uma aplicação baseada no modelo apresentado, construída de forma organizada de modo a conter a estrutura básica do regulador integrado de tensão e velocidade, definindo as fronteiras e interfaces para que seja adaptado às características de um projeto específico.

O uso deste framework em um ambiente de desenvolvimento com interface gráfica intuitiva e poderosos recursos de edição, permite um avanço enorme em termos de programabilidade em nível de usuário.

Nota-se uma redução significativa do tempo de aprendizado, habilitando, por exemplo, um usuário com pouco conhecimento em modelagem e conceitos de programação a implementar modificações simples no programa de modo intuitivo e rápido.

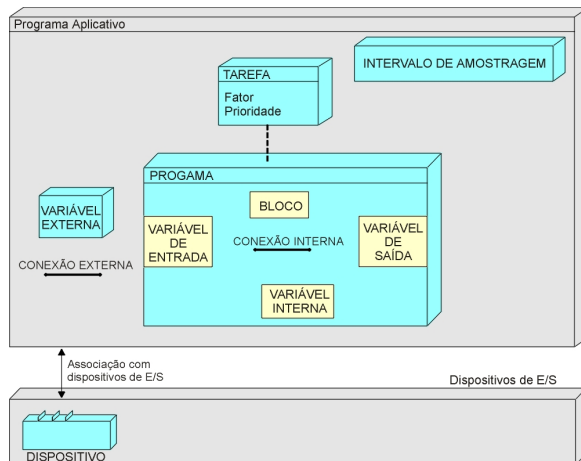


Fig. 2. Modelo conceitual de software.

Em relação ao programa aplicativo de um regulador integrado, a programação gráfica mostra-se eficiente tanto para os desenvolvedores da estrutura principal do modelo, aqui tratado como framework aplicativo, quanto para os engenheiros de aplicação, ou mesmo clientes finais do produto, que irão adaptar ou incluir funções específicas de uma dada aplicação.

### C. Framework aplicativo

O framework ao qual se refere este artigo é um projeto criado no ambiente de desenvolvimento que define a estrutura essencial de software de um regulador integrado de tensão e velocidade padrão REIVAX. Seu objetivo é definir lógicas de operação e o comportamento básico do regulador, bem como fornecer, de modo sistemático e estruturalmente organizado, as interfaces necessárias para interação com estas funcionalidades.

Para criação do framework foi feito um intenso levantamento das características dos programas aplicativos utilizados em reguladores de tensão e de velocidade, os quais, até então, vinham sendo criados de acordo com as necessidades de cada projeto e sem possuir uma estrutura padronizada. Estes dados, juntamente com as solicitações de novas funcionalidades, foram utilizados como requisitos na construção da arquitetura padrão do framework.

Alguns aspectos estruturais foram muito importantes para a concretização do projeto de um framework aplicativo para reguladores integrados de tensão e velocidade, satisfazendo

todos os requisitos de projeto pertinentes e com a confiabilidade e o desempenho requeridos por um sistema de controle da geração de energia elétrica:

- ❑ Uso de um Sistema Operacional de Tempo Real com escalonamento preemptivo, permitindo a criação de um software aplicativo multitarefa com prioridades e passos de execução definidos pelo usuário;
- ❑ Desenvolvimento de uma biblioteca de blocos proprietária, contemplando funções especializadas para o uso em um sistema de controle da geração de energia elétrica;
- ❑ Uso de um ambiente de desenvolvimento proprietário criado para suprir os requisitos de projeto, manutenção e depuração do software aplicativo.

O desenvolvimento do framework para o software aplicativo de um regulador integrado trouxe, entre outros, os seguintes benefícios:

- ❑ Padronização do núcleo do programa aplicativo e criação de interfaces básicas para sua adaptação a aplicações específicas, reduzindo o número de erros e diminuindo o esforço nas etapas de conformação do software no projeto dos reguladores;
- ❑ Maior integração entre projeto de software e projeto elétrico do regulador;
- ❑ Arquitetura modular em programas, com programação otimizada no escalonamento de tarefas (prioridades e passos de execução) e definição clara das características funcionais de cada módulo, bem como de suas interfaces de entrada/saída;
- ❑ Distinção entre os módulos que controlam o sistema de excitação e regulador de velocidade, permitindo uma adaptação rápida do framework a uma aplicação como regulador individual;
- ❑ Robustez da aplicação a partir da consolidação e padronização de blocos funcionais com base na experiência da REIVAX em projetos e comissionamentos em campo;
- ❑ Padronização do mapa de registros do regulador, fornecendo suporte estruturado e transparente para parametrização e monitoração de grandezas através de dispositivos de IHM;
- ❑ Documentação detalhada e fiel da arquitetura estrutural e funcional do software e das interfaces utilizadas para configuração dos programas para uma dada aplicação;
- ❑ Definição de uma política de controle e gerenciamento de configuração para o software desenvolvido, com suporte para o recebimento e execução de melhorias solicitadas pelo usuário interno e externo, garantindo rastreabilidade de alterações, gerenciamento de incompatibilidades e melhoria contínua;

O framework desenvolvido foi denominado FAP – RTVX100 Framework Aplicativo, e executa basicamente as seguintes funções:

# A Graphical Development Environment applied to the Design and Debugging of an Application Framework for an Integrated Speed Governor and Voltage Regulator System

- ❑ Algoritmos de transdução, calibração e escala;
- ❑ Algoritmos de controle;
- ❑ Lógicas de operação e comando;
- ❑ Procedimentos de configuração e sistema;
- ❑ Procedimentos de ensaio e operação em modo teste;
- ❑ Procedimentos de alarme e proteção;
- ❑ Procedimentos de monitoração: indicações e sinalização;
- ❑ Registro de eventos e sinais;

Estas funções são implementadas em módulos ou programas distintos no framework. Estes programas são associados a tarefas do núcleo de execução do SBE, cuja prioridade e passo de execução são escolhidos de modo a otimizar o uso da CPU e garantir o correto funcionamento do sistema, em função das características de cada programa. São definidas três tarefas de execução no framework, de acordo com a Tabela 1.

TABELA 1  
TAREFAS DEFINIDAS PELO FRAMEWORK

Tarefa	Prioridade	Intervalo*	Função Principal
Transdução	Máxima	1,6 ms	Conformação e execução de cálculos primários dos sinais adquiridos pela placa de aquisição.
Controle	Normal	10 ms	Execução de malhas e algoritmos de controle além de lógicas de acionamento.
Sistema	Baixa	100 ms	Execução de funções de sistema e interface com dispositivos de IHM

\* Estes valores são utilizados em aplicações típicas. Em aplicações especiais, o intervalo pode ser reduzido através da utilização de uma CPU x86 intercambiável, de maior desempenho.

### D. Ambiente de desenvolvimento

Para implementação do ambiente de desenvolvimento foi feito um levantamento de requisitos, com base nas necessidades de projeto e utilização do framework para reguladores de tensão e velocidade. Estes requisitos compreendiam facilidade na implementação, leitura e entendimento dos programas, reutilização de algoritmos, capacidade de depuração, monitoração online de sinais e facilidade em obter a documentação da aplicação implementada.

Na implementação do ambiente de desenvolvimento foram utilizadas ferramentas RAD e técnicas de projeto orientado a objeto. Utilizou-se como plataforma de desenvolvimento o Delphi 4 Professional®, da Borland.

O ambiente de desenvolvimento criado, denominado SEC – Sistema de Edição de Configurações, roda no sistema operacional Windows®.

Este ambiente foi projetado de acordo com as necessidades específicas ao desenvolvimento do software aplicativo dos reguladores de tensão e velocidade desenvolvidos pela REIVAX. Esta particularidade permite somar aos

tradicionais recursos de um ambiente de edição padrão, um conjunto de funcionalidades especiais destinados a aplicação específica. Funcionalidades estas que suprem as necessidade do usuário nas etapas de projeto e utilização do framework aplicativo tal como ele foi desenhado, com alto grau de qualidade e eficiência.

Um resumo geral das funcionalidades providas pelo ambiente de desenvolvimento é descrito nos itens a seguir:

- ❑ Programação da lógica da aplicação utilizando linguagem de alto nível baseada na linguagem FBD da IEC 61131-3, com interface amigável para a inserção de blocos e conexões;

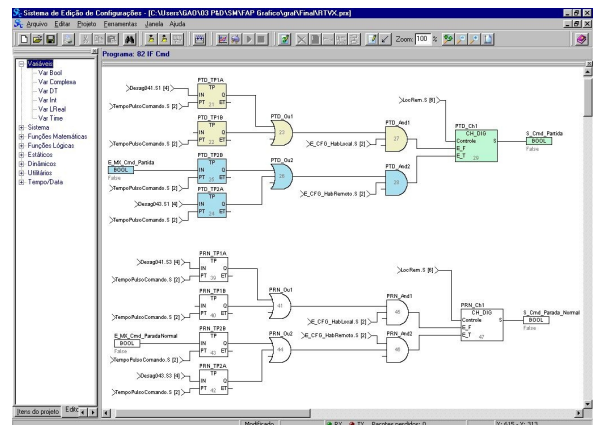


Fig. 3 – Programação da aplicação usando blocos de função

- ❑ Programação modular com definição de tarefas e programas;
- ❑ Recurso de compilação condicional, permitindo habilitar ou desabilitar a compilação de um programa. Esta funcionalidade permite, por exemplo, rapidamente adaptar o framework para uma aplicação individual como regulador de tensão ou velocidade;
- ❑ Depuração e alteração de parâmetros da aplicação em tempo de execução. Permite, entre outras coisas, realizar o ajuste online de alguns parâmetros que não encontram-se disponíveis nas IHMs do sistema;
- ❑ Visualização, em tempo de execução, de qualquer sinal disponível na lógica da aplicação
- ❑ Possibilidade de forçar valores em dispositivos de E/S, independente da aplicação em execução. Esta característica é especialmente útil para testes de interface da aplicação;



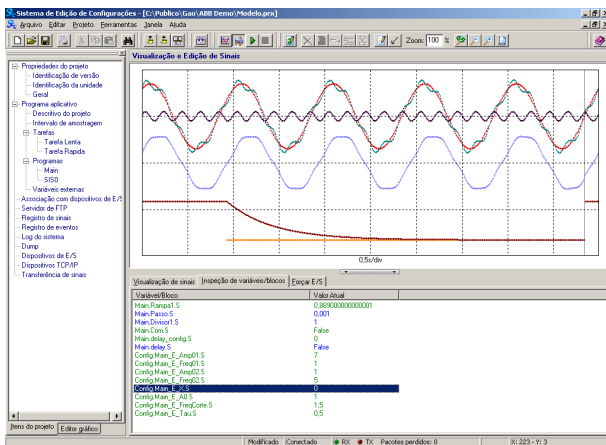


Fig. 4 – Monitoração de sinais e depuração da aplicação

- ❑ Configuração de registradores de sinais e programação da lógica de gatilhos. Esta funcionalidade facilita a análise e o diagnóstico de problemas intermitentes durante a operação dos equipamentos;
- ❑ Configuração de registrador de eventos;
- ❑ Obtenção, a qualquer momento, da documentação das lógicas implementadas.

### III. RESULTADOS

#### A. Projeto e implementação do framework

Com a implementação do ambiente gráfico de desenvolvimento, o tempo de síntese das funções do framework e outros algoritmos diminui sensivelmente. Nota-se que, entendido o problema, basta sintetizá-lo utilizando a biblioteca de blocos disponível, o que é feito diretamente no ambiente de desenvolvimento. Nota-se também que, com um pequeno esforço de aprendizado, se obtém bons resultados na implementação de aplicações.

A vantagem de se utilizar um ambiente de desenvolvimento proprietário aparece com bastante intensidade em alguns aspectos básicos, próprios ao uso deste tipo de solução. Um deles é a perfeita integração com os recursos oferecidos pela plataforma de hardware e software, permitindo o aproveitamento ótimo das funcionalidades existentes e contribuindo para o desenvolvimento de aplicações de modo mais intuitivo e rápido. Outro aspecto importante é a total detenção da tecnologia aplicada, permitindo agilidade e flexibilidade na implementação de melhorias a partir das solicitações dos usuários.

#### B. Aplicação do framework

O desenvolvimento do framework aplicativo trouxe grandes benefícios, principalmente para os engenheiros de aplicação responsáveis pelas adaptações finais na modelagem de software das aplicações envolvendo reguladores de tensão

e velocidade. Vários aspectos podem ser enumerados neste sentido, entre eles a padronização do software, o fornecimento de uma documentação detalhada do programa aplicativo, a definição clara das interfaces para adaptação da aplicação e o suporte para solicitação e acompanhamento de melhorias no framework.

A implementação do framework não só tornou o programa aplicativo mais legível e controlado, como também fez diminuir a taxa de erros e o tempo de engenharia despendido na etapa de conformação do software nos projetos de reguladores. Isto se traduz, entre outras coisas, na introdução de “boas práticas” de programação favorecendo os aspectos de compreensão e reusabilidade do código.

#### C. Documentação

O ambiente de desenvolvimento trouxe a possibilidade de obter a qualquer momento uma documentação confiável, pronta para ser entregue ao usuário final e que representa fielmente a aplicação implementada com a ferramenta. A criação de uma estrutura padrão para o programa aplicativo dos reguladores contribuiu para gerar uma documentação clara e organizada, facilitando a compreensão, análise e manutenção do software do regulador.

#### D. Depuração e testes

Um dos problemas encontrados pelo engenheiro de aplicação, durante a fase de teste e comissionamento dos reguladores, é o fato de as ferramentas não darem suporte à depuração e investigação de problemas no software da aplicação. Com o novo ambiente de desenvolvimento esta lacuna foi preenchida com a disponibilização de recursos de inspeção e visualização de sinais em tempo de execução, bem como o forçamento de E/S. Esses novos recursos reduziram o tempo gasto na resolução de erros e facilitaram sobremaneira os testes com os equipamentos.

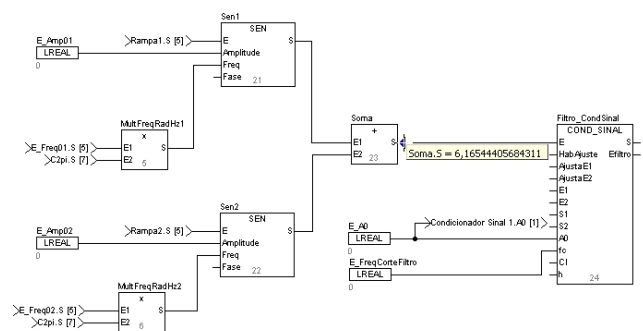


Fig. 7 – Inspeção de variáveis no editor gráfico

#### E. Registro de sinais

Este recurso pode ser utilizado em aplicações específicas

que exijam o registro de sinais a partir de eventos relevantes para o sistema (sistemas de monitoração), em ensaios de campo onde existe a necessidade de registro de algumas operações efetuadas pelo equipamento ou mesmo, para depurar o funcionamento do programa aplicativo em fase de desenvolvimento.

As figuras a seguir mostram registros obtidos durante os testes de um regulador em campo.

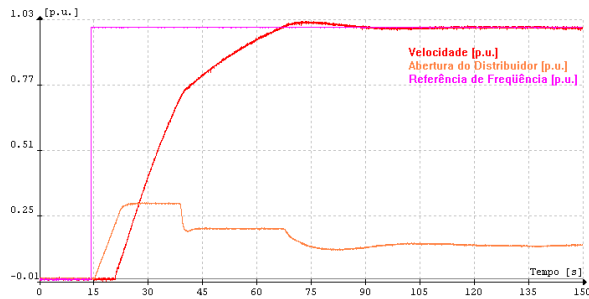


Figura 8 – Partida da máquina.

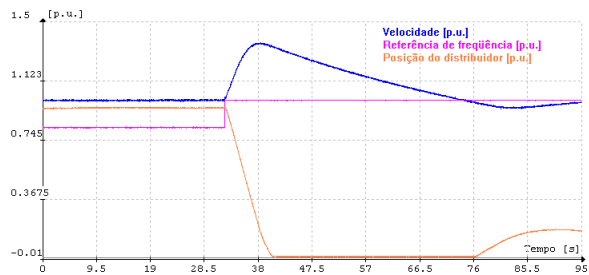


Figura 9 – Rejeição de carga

## IV. TRABALHOS FUTUROS

Para algumas aplicações o uso da linguagem de programação utilizando blocos (FBD) pode não ser tão adequada. Assim, pretende-se que o ambiente de desenvolvimento dê suporte à implementação de aplicações usando outras linguagens definidas pela IEC61131-3 como Ladder e SFC (Structured Function Chart). O uso de ladder, por exemplo, facilita a compreensão e a implementação de lógica a relés, enquanto que a SFC facilita o projeto de programas estruturados, através de cadeias de ações paralelas e seqüenciais.

O uso de estruturas que se repetem dentro do framework aplicativo tornou evidente a importância da criação de blocos definidos pelo usuário dentro do ambiente de desenvolvimento. Isto permitiria ao usuário implementar estruturas de blocos, e a partir destes gerar um novo bloco que poderia ser utilizado em outros programas.

## V. CONCLUSÃO

A utilização de um ambiente gráfico de desenvolvimento com recursos de monitoração e edição de sinais, associada ao uso de um framework aplicativo padrão demonstrou ser uma

estratégia extremamente vantajosa para o projeto, validação e manutenção do software aplicativo de reguladores de tensão e velocidade. Os benefícios diretos da utilização destes recursos incluem:

- Ganho significativo de tempo nas etapas de implementação, teste e depuração dos equipamentos;
- Menor índice de retrabalho em aplicações;
- Confiabilidade da documentação do software;
- Menor tempo de aprendizado da ferramenta de desenvolvimento;
- Menor tempo de colocação em funcionamento do equipamento.

Por fim, o ambiente de desenvolvimento e o framework aplicativo criados, fazendo uso de uma linguagem de programação padronizada de alto nível associada a uma documentação estruturada da aplicação, conferem um suporte poderoso a todas as etapas de engenharia envolvidas com o desenvolvimento do programa aplicativo dos reguladores de tensão e velocidade.

## REFERÊNCIAS

- [1] ZENI Jr., N., et ali. Sistema Integrado de Controle de Velocidade, Tensão e Automatismo. XVI SNPTEE – São Paulo, 2001. Brasil.
- [2] BONFATTI, F. et ali, “IEC61131-3 Programming Methodology”, C.J. International, França, 1997.