ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA TESE DEFENDIDA POR Dario Amaya Hurtado E APROVADA PELA COMISSÃO JULGADORA EM 15 / 06 / 2011

ORIENTADOR

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

Dario Amaya Hurtado

Proposta de Modelagem e Projeto de Arquitetura de Controle de Sistemas Híbridos Aplicado em Processos de Manufatura Industrial

Campinas, 2011.

79/2011

Dario Amaya Hurtado

Proposta de Modelagem e Projeto de Arquitetura de Controle de Sistemas Híbridos Aplicado em Processos de Manufatura Industrial

Tese apresentada ao Curso de Doutorado da Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas, como requisito para a obtenção do título de Doutor em Engenharia Mecânica.

Área de Concentração: **Mecânica dos Sólidos e Projeto Mecânico**

Orientador: **Prof. Dr. João Mauricio Rosário**

Campinas

2011

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Am15p

Amaya Hurtado, Dario.
Proposta de modelagem e projeto de arquitetura de controle de sistemas híbridos aplicado em processos de manufatura industrial / Dario Amaya Hurtado. -- Campinas, SP: [s.n.], 2011.

Orientador: João Mauricio Rosário.
Tese de Doutorado - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Sistemas híbridos. 2. Automação industrial. 3. Prototipagem rápida. I. Rosário, João Mauricio. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

Título em Inglês: A modeling proposal and design of a control architecture of hybrid systems applied on manufacturing industrial processes
Palavras-chave em Inglês: Hybrid systems, Industrial automation, Rapid prototyping
Área de concentração: Mecânica dos Sólidos e Projeto Mecânico
Titulação: Doutor em Engenharia Mecânica
Banca examinadora: Antonio Batocchio, Francisco Carlos Parquet Bizarria, Cintia Kimie Aihara, Leonimer Flavio de Melo
Data da defesa: 15-06-2011
Programa de Pós Graduação: Engenharia Mecânica

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO

TESE DE DOUTORADO

Proposta de Modelagem e Projeto de Arquitetura de Controle de Sistemas Híbridos Aplicado em Processos de Manufatura Industrial

Autor: **Dario Amaya Hurtado**
Orientador: **João Mauricio Rosário**

A Banca Examinadora composta pelos membros abaixo aprovou esta Tese:

Prof. Dr. João Mauricio Rosário, Presidente.
Universidade Estadual de Campinas - FEM/DPM

Prof. Dr. Antônio Batocchio
Universidade Estadual de Campinas – FEM/DEF

Prof. Dr. Francisco Carlos Parquet Bizarria
Instituto de Aeronáutica e Espaço-DCTA

Profa. Dra. Cintia Kimie Aihara
Universidade Estadual de Campinas - COTUCA

Prof. Dr. Leonimer Flavio de Melo
Universidade Estadual de Londrina- UEL/DEEL

Campinas, 15 de Junho de 2011

Dedicatória

Dedico este trabalho à minha esposa por cuidar de nossos filhos e de nosso amor, a meus filhos pela compreensão de minha ausência, minha mãe por tudo o que fez por mim ao longo de minha vida, meus irmãos por toda a colaboração e os sogros pela sua ajuda incondicional.

Agradecimentos

Este trabalho não poderia ser terminado sem a ajuda de diversas pessoas às quais presto minha homenagem:

Ao meu orientador, Prof. Dr. João Mauricio Rosário pela confiança que ele deu e a maneira como levou o desenvolvimento do trabalho.

Ao minha família que sempre deu sua voz de alento para continuar

Ao Dr. Oscar Fernando Avilés pela sua grande amizade.

À "Univerdidad Militar Nueva Granada" pelo seu apoio na minha permanência na UNICAMP.

A meus colegas por seu companheirismo e apoio incondicional

Aos professores e funcionários do Departamento de Projeto Mecânico da Faculdade de Engenharia Mecânica da UNICAMP, com os que tive a oportunidade de trabalhar.

“O que sabemos é uma gota, o que ignoramos é um oceano”

Isaac Newton

Resumo

Com o avanço da tecnologia têm sido incrementados a complexidade dos sistemas de produção industrial. Essa complexidade está marcada pela interação das variáveis que respondem a uma dinâmica baseada em eventos discretos e as variáveis que são descritas em relação ao tempo. Isto levou para uma mudança no paradigma dos sistemas automáticos de controle industrial, onde tradicionalmente essas dinâmicas são analisadas de maneira independente, para propor uma estratégia de modelagem e controle na que levem em consideração a interação das duas dinâmicas constituindo um sistema de dinâmica hibrida.

Neste trabalho se desenvolve uma proposta de modelagem e controle para sistemas que respondem a uma dinâmica de arquitetura híbrida. A proposta esta baseada no recente formalismo desenvolvido como método de integração para Equações diferenciais ordinárias, denominado sistema de estado quantificado (QSS), que foi utilizado na discretização das variáveis de estado que descrevem a dinâmica continua de um sistema. Além disso, foi utilizado o formalismo de especificação de eventos discretos (DEVS), com o proposito de representar a dinâmica regida pela ocorrência de eventos e a interação com a dinâmica regida em relação ao tempo em um único modelo, permitindo projetar estratégias de controle. Com o proposito de validar a proposta, foram realizados dois estudos de caso aplicados em um processo de manufatura industrial, levando em consideração sua arquitetura dinâmica híbrida permitindo a implementação em sistemas embarcados utilizando o conceito da prototipagem rápida.

Palavras Chave: Controle híbrido, Controle por Evento, Automação industrial, Prototipagem Rápida.

Abstract

With the advancement of technology has been enhanced complexity of industrial production systems. This complexity is characterized by the interaction of variables that respond to a dynamic and discrete event based on the variables that are described in relation to time. This led to a paradigm shift in industrial automatic control systems, where traditionally these dynamics are analyzed independently, to propose a strategy for the modeling and control that take into account the dynamic interaction of the two forming a hybrid dynamic system.

In this work a proposed modeling and control systems that respond to a dynamic hybrid architecture. The proposal is based on the formalism recently developed as a method of integration for ordinary differential equations, called quantified state system (QSS), which was used in the discretization of state variables that describe the dynamics of a system continues. In addition, we used the formalism of discrete event specification (DEVS) with the purpose of representing the dynamics governed by the occurrence of events and interaction with the dynamics governed over time in a single model, allowing to design control strategies. In order to validate the proposal, were carried out two case studies used in a process of industrial manufacturing, taking into account its dynamic hybrid architecture allowing the implementation in embedded systems using the concept of rapid prototyping.

Key Words: Hybrid Control, Event Control, Industrial Automation, Rapid Prototyping, Process Modeling.

Lista de Ilustrações

Lista de Tabelas

Lista de Abreviaturas e Siglas

Letras Latinas

$\boldsymbol{X}$**-** Conjunto de Eventos de Entrada

$\boldsymbol{Y}$ - Conjunto de Eventos de Saída

$\boldsymbol{S}$ - Conjunto de Estados do Sistema

$\boldsymbol{t_a}$ - Função do Avanço do Tempo

$\boldsymbol{Q}$ - Conjunto de Estados

$\boldsymbol{H}$-Tupla Representação DEVS

$\boldsymbol{D}$ - Conjunto de Nomes DEVS Acoplado

$\boldsymbol{e}$ - Acumulador Avanço do Tempo DEVS

$\boldsymbol{EIC}$ - Conjunto de Enlaces de Entradas

$\boldsymbol{EOC}$ - Conjunto de Enlaces de Saída

$\boldsymbol{IC}$ - Conjunto de Enlaces Internos

$\boldsymbol{p}$ - Porto

$\boldsymbol{X_p}$ - Vetor portas de Entrada

$\boldsymbol{Y_p}$ - Vetor portas de Saída

$\boldsymbol{D}$ - *C*onjunto de Modelos DEVS

$\boldsymbol{EIC}$ - Conjunto Enlaces de Entradas DEVS Acoplado

$\boldsymbol{EOC}$ - Conjunto Enlaces de Saída DEVS Acoplado

$\boldsymbol{IC}$ - Conjunto de Enlaces Internos DEVS Acoplado

$\triangle\ \boldsymbol{Q}$ - Quantum

$\boldsymbol{K_E}$ - Constante de Voltagem

INL - Corrente sem Carga

SNL - Velocidade sem Carga

IPK - Corrente Máxima

TPK - Torque Máxima

Jr - Inércia do Rotor

$\boldsymbol{i_a}$ - Corrente de Armadura

$\boldsymbol{L_a}$ - Enrolamento de Armadura

$\boldsymbol{V_a}$ - Voltagem de Armadura

$\boldsymbol{R_a}$ - Resistência de Armadura

$\boldsymbol{w_a}$ - Velocidade Angular

$\boldsymbol{\dot{x_1}}$ - Variável de Estado de Velocidade

$\boldsymbol{\dot{x_2}}$ - Variável de Estado de Corrente

$\boldsymbol{\dot{x_3}}$ - Variável de Estado de Posição

$\boldsymbol{J}$ - Inércia

$\boldsymbol{B}$ - Atrito

$\boldsymbol{T_L}$ - Torque de Carga

$\boldsymbol{k_t}$ - Constante do Motor

$\boldsymbol{\lambda}$ - Função de Saída DEVS Atômico

RPM - Revoluções por Minuto

Kp - Constante Proporcional

Ki - Constante Integral

Kd - Constante Diferencial

L - Longitude da Esteira

$\boldsymbol{t_{tr}}$ - Tempo de Retraço da Esteira

$\boldsymbol{v_m}$ - Velocidade Meia

$\boldsymbol{v_{f(i)}}$ - Velocidade Final

$\boldsymbol{t_x}$ - Tempo de Desaceleração

$\boldsymbol{t_{tr}}$ - Tempo de Transporte na Esteira

$\boldsymbol{a}$ - Pendente de Aceleração e Desaceleração

$\boldsymbol{v_{fmax}}$ - Velocidade Final Máxima

$\boldsymbol{v_{mmax}}$ - Velocidade Média Máxima

$\boldsymbol{S_i}$ - Sensor de Entrada

$\boldsymbol{T_{da}}$ - Tempo de Atraso Anterior

Tempo - Tempo Faltante para Completar Processo

Longitude - Longitude da Esteira.

S_o - Sensor de Saída

***lamb_dint* -** Cálculo do Valor de Saída.

Dext - Função de Cálculo do Estado Seguinte

TL - Torque de Carga.

Siglas

DPM - Departamento de Projeto Mecânico

FEM - Faculdade de Engenharia Mecânica

LAIR - Laboratório de Automação Integrada e Robótica

UNICAMP - Universidade Estadual de Campinas

IEEE - Instituto de Engenheiros Elétricos e Eletrônico

Letras Gregas

γ - Transições das Variáveis de Dinâmica Contínua,

$\boldsymbol{\delta}$ - Função de Transição

$\boldsymbol{\lambda}$ - Função de Saída

Superescritos

$\oplus$ - Or Exclusiva

Subscritos

i - Índice do Número da Esteira

Abreviações

CAD -Projeto Assistido por Computador

CAE - Engenharia Assistida por Computador

CIM -Manufatura Integrada por Computador

CLP - Controlador Lógico Programável

CNC - Controle Numérico Computadorizado

DEVS - Especificação de Sistemas a Eventos Discretos

FMS - Sistema Flexível de Manufatura

GRAFCET - Grafo Funcional de Comando Estado - Transição

SCADA - Aquisição de dados, Controle e Supervisão.

SED - Sistema de Eventos Discretos

HIL - Hardware-in-the-loop

TIC - Tecnologias da Informação e Comunicação

JIT - No Tempo Justo

MRP - Planejamento de Necessidades de Materiais

ERP - Planejamento dos Recursos de Manufatura

PID - Proporcional Integral Derivativo

C3I - Comunicações, Comando, Controle e Inteligência nas Forças Armadas

QSS - Sistema de Estados Quantificados

VHDL - Linguagem de Descrição de Hardware

QSC - Controle por Estados Quantificado

MIMO - Múltiplas Entradas e Múltiplas Saídas.

ECU - Unidades de Controle Eletrônico

DC/DC - Conversor de Corrente Direita para Corrente Direita

PR - Prototipagem Rápida

ASCII - Código Padrão Americano para Troca de Informações

SRAM - Memória Flash

USB - Bus Serial Universal

ODE's - Equações. Diferenciais. Ordinárias

SUMÁRIO

Capítulo 1

1. Introdução

Na atualidade, com o desenvolvimento da tecnologia e a intensificação da tendência da globalização, o desenvolvimento de sistemas de manufatura industrial necessita cada vez mais de uma abordagem que englobe a estrutura dinâmica de controle, que responda rapidamente às exigências e mudanças nos mercados, os quais são cada vez mais exigentes (HEGNY et al., 2010)

Essas exigências estão relacionadas com a flutuação das demandas dos clientes, a capacidade para diversificar os produtos oferecidos, as mudanças contínuas nos sistemas tecnológicos, crescimento constante dos concorrentes e o curto ciclo de vida dos produtos, são algumas das questões que levam as empresas a implementarem sistemas de produção com grande agilidade e flexibilidade, assegurando assim a sua capacidade competitiva (SHIN et al., 2009).

No transcorrer do tempo, tem sido essa competitividade que levou a indústria a diversificar e melhorar seus modelos de produção. Desde o início da indústria tradicional, com sua origem na produção em massa através do modelo Fordista, que evoluindo para modelos como Toyota, sistemas de manufatura flexíveis e integrados por computadores, até as novas propostas em manufatura inteligente (PAOLUCCI e SACILE 2005).

Essa evolução dos sistemas de manufatura tem sido possível pela evolução da tecnologia, a qual tem permitido o desenvolvimento dos sistemas de controle dos processos industriais.

Sistemas de controle que em seu início foram realizados de maneira empírica até a formalização matemática proposta por Maxwell no meio do século XIX, quem propôs a modelagem dos sistemas através de Equações diferenciais (ROSENBROCK, 1969). Com essa

ideia, começou a definir uma estrutura formal pela teoria do controle, proposta iniciada por Lyapunov ao final do século XIX.

Até os anos 50 do século XX, a maioria dos esforços teóricos no campo do controle foi para a análise e projeto de arquiteturas de controle focadas a sistemas que variavam em relação ao tempo. Foram desenvolvidas técnicas de modelagem, análise de estabilidade e resposta no tempo. Além disso, foram desenvolvidas diferentes arquiteturas de controle, mas todas aplicáveis para sistemas de tempo contínuo.

Nos anos 60 do século XX se iniciou a tecnologia digital e a aplicação dessa no campo industrial, dando origem a um novo conceito na dinâmica dos sistemas que agora são descritos em relação a eventos discretos. Esse novo tipo de sistema, em sua maioria são criados pelo homem, como é o caso da arquitetura dos computadores, as redes de comunicação, os sistemas automatizados de fabricação, sistemas inteligentes de transporte, software distribuído, entre outros (CASSANDRAS e LAFORTUNE, 2008).

Entretanto, a teoria do controle desenvolvida para analisar e controlar os sistemas que respondem no domínio do tempo, os quais em sua maioria estão regidos pela análise das leis da natureza. São geralmente inadequadas e insuficientes para analisar e controlar os sistemas que respondem a eventos discretos. Para isso, foi necessário propor vários formalismos para realizar a análise e controle dos sistemas que respondem em relação a eventos. Entre eles, estão as redes de Petri, autômatos híbridos, Grafcet, Max-min-plus, Discrete Event System Specification (DEVS), entre outros (KOCÍ e JANOUSEK, 2010).

Na dinâmica dos processos industriais atuais existe uma forte interação entre a dinâmica regida em relação ao tempo e a dinâmica de eventos discretos, mas tradicionalmente essas duas dinâmicas são analisadas de maneira independente, impedindo considerar e controlar os efeitos da interação de uma dinâmica respeito à outra (VAN DER SCHAFT e SCHUMACHER, 2000).

Portanto, foram necessárias novas áreas de trabalho e investigação, para estudar o comportamento das duas dinâmicas nos processos industriais, dando origem a uma nova

dinâmica nos sistemas de produção industrial a qual foi denominada sistemas dinâmicos híbridos ou simplesmente sistemas híbridos.

Os sistemas híbridos são definidos como aqueles que estão constituídos por variáveis de estado de natureza continua, que respondem em relação ao tempo e variáveis que respondem em relação a eventos discretos (GARCIA; GOMEZ; MIYAGI, 2008). Na Figura 1.1 se apresenta um sistema de arquitetura híbrida, no qual as variáveis que respondem a uma dinâmica contínua interagem com a dinâmica de eventos discretos. Além disso, as dinâmicas de eventos discretos também influem nas variáveis descritas em relação ao tempo. Essa interação define um sistema em sua dinâmica, como um sistema de arquitetura hibrida. Este tipo de sistemas de dinâmica hibrida estão constituído por sinais que pode ser descritos através do conjunto dos números reais e que geralmente correspondem com sinais de natureza física, misturada com sinais que podem pegar valores que são descritas através de um conjunto discreto de estados que representa seu comportamento. Uma abordagem para conseguir realizar a modelagem de um sistema de dinâmica hibrida, consiste em representar essas dinâmicas que são descritas através do tempo, em função de eventos de tempo discreto. Isto permitirá estabelecer as relações que existe entre as variáveis de dinâmica continua e as variáveis de dinâmica a eventos discretos.

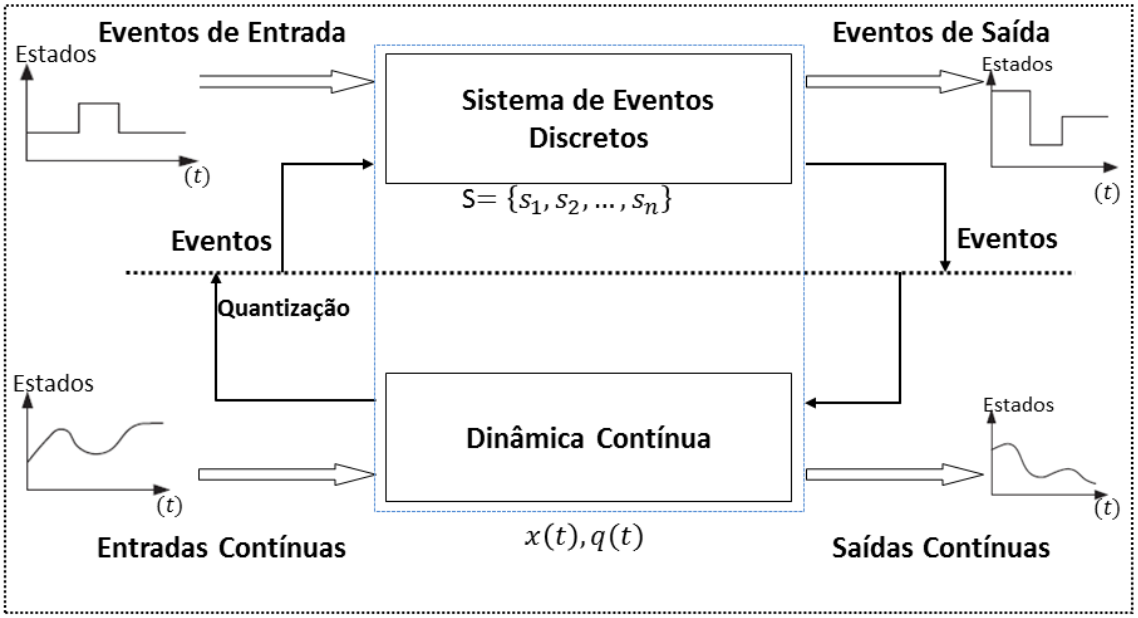

Figura 1.1. Sistema de arquitetura híbrida

Com esse novo conceito na dinâmica dos sistemas de produção industrial, este trabalho de pesquisa apresenta uma proposta para a modelagem e projeto de arquitetura de controle aplicável em sistemas cuja dinâmica está composta da interação de variáveis descritas em relação ao tempo e de eventos discretos, com ênfase na síntese através de sistemas embarcados utilizando o conceito da prototipagem rápida.

Para o desenvolvimento desse trabalho, foram realizados dois estudos de caso aplicados em um sistema de manufatura industrial composto por várias unidades de produção, que respondem a uma dinâmica híbrida de acordo com a sua configuração física.

1.1 Apresentação do problema de pesquisa

Tradicionalmente, os sistemas industriais têm sido analisados de acordo ao comportamento de suas variáveis em função de uma dinâmica em relação ao tempo ou uma dinâmica em relação a eventos de maneira independente. Mas, com a evolução da tecnologia e a complexidade dos sistemas, tem surgido uma nova dinâmica a qual permite a interação das variáveis que são descritas em relação ao tempo e as variáveis descritas através de eventos, surgindo os sistemas dinâmicos de arquitetura híbrida (KHAN e IRAVANI, 2007).

Para atingir esse objetivo torna-se necessário analisar os formalismos existentes de modelagem e controle dos sistemas descritos através de eventos discretos. Além das teorias desenvolvidas para as variáveis regidas em função do tempo, com o propósito de representar e controlar os sistemas de produção onde essas duas dinâmicas interagem.

O propósito de obter modelos dos sistemas de produção industrial é gerar a possibilidade da análise de seu compartimento, permitindo propor estratégias de controle para definir o regime de trabalho dos processos, neste caso em sistemas com arquitetura híbrida (LYGEROS; GODBOLE; SASTRY, 1996).

Consequentemente nesta tese é proposta uma arquitetura de controle baseada em eventos, aplicável nos sistemas que responde a dinâmica de arquitetura híbrida, realizando uma análise comparativa com o método tradicional de controle em tempo discreto.

Com o propósito de realizar a validação dos conceitos desenvolvidos neste trabalho, foram realizados dois estudos de casos de modelagem e controle de sistemas que respondem às dinâmicas de arquitetura híbrida. Para isto, foi utilizada a Plataforma Industrial para Pesquisa, Ensino e Formação em Automação (PIPEFA), disponível na Faculdade de Engenharia Mecânica da UNICAMP.

1.2 Motivação

Considerando-se que a maioria dos processos industriais, são representados por dinâmicas que envolvem a interação de variáveis que respondem em relação ao tempo e em relação a eventos, mais tradicionalmente essas dinâmicas são analisadas de maneira independente. Surgiu a motivação de realizar uma proposta de modelagem e controle aplicáveis em sistemas que estão constituídos por uma dinâmica híbrida, baseado no formalismo DEVS e QSS. Fazendo ênfase na síntese de sistemas de controle utilizando o conceito da prototipagem rápida, baseados na geração automática de código para sistemas embarcados, com o intuito de prover uma ferramenta que permita ser utilizada na automação dos processos industriais.

1.3 Objetivos

Realizar uma proposta de modelagem de um sistema de produção composto de unidades de arquitetura híbrida (dirigidos pelo tempo e pela ocorrência de eventos), e projetar uma estrutura de controle fazendo ênfase na prototipagem rápida.

Realizar uma analise das vantagens do controle de sistemas contínuos discretizados através de eventos, em relação aos sistemas contínuos discretizados no tempo.

Analisar o custo computacional dos sistemas de controle em tempo discreto em relação aos sistemas de controle baseado em eventos.

Realizar a síntese dos sistemas de controle projetados baseados no conceito da prototipagem rápida.

1.4 Organização do trabalho

Este trabalho foi desenvolvido levando em conta a arquitetura do sistema de produção plataforma Industrial para Pesquisa, Ensino e Formação em Automação (PIPEFA), desenvolvida no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da Universidade de Campinas (UNICAMP). Esta tese de doutoramento está estruturada em seis capítulos descritos a seguir.

Capítulo 1: É realizada uma breve introdução do problema que será tratado na tese. São mostrados conceitos iniciais, motivação do trabalho, descrição dos principais objetivos e a descrição da distribuição deste trabalho de pesquisa.

Capítulo 2: Realiza-se uma revisão bibliográfica dos temas de interesse no desenvolvimento da tese, começando com uma descrição dos aspectos mais relevantes dos sistemas de manufatura e trabalhos desenvolvidos nessa área. Ainda neste capítulo, é realizada uma revisão da literatura referente a ferramentas de modelagem de sistemas híbridos, especificamente à modelagem através de DEVS e QSS, e finalmente do conceito de prototipagem rápida.

Capítulo 3: Neste capítulo realiza-se um estudo de ferramentas matemáticas utilizadas para modelagem e controle de sistemas de manufatura industrial cuja arquitetura responde ao conceito de sistemas híbridos, e é realizada uma proposta para a modelagem controle e síntese desse tipo de sistemas.

Capítulo 4: Este capítulo é dedicado a descrever o sistema de produção utilizado nos estudos de caso

Capítulo 5: Este capítulo é dedicado ao estudo de validação das ferramentas propostas a partir de um estudo de caso representando elementos de um sistema de manufatura.

Capítulo 6: Neste capítulo vai ser apresentado um estudo de caso da modelagem e controle de um sistema de manufatura industrial. Este sistema está composto de varias unidades de produção, as quis devem garantir tempos de produção uniformizados na confecção dos produtos.

Capítulo 7: Neste capítulo é realizada a validação da proposta de modelagem aplicada a processos de arquitetura híbrida, desenvolvida no estudo de caso descrito no capítulo 5. Para isto, utiliza-se o conceito da prototipagem rápida a partir da utilização de ferramentas para concepção de sistemas embarcados.

Capítulo 8: Finalmente, neste capítulo da tese, são apresentadas as conclusões deste trabalho de pesquisa e perspectivas futuras. Em anexo são apresentados alguns trabalhos publicados em revistas e eventos nacionais e internacionais, resultado desta pesquisa.

Capítulo 2

2. Revisão da Literatura e Aspectos Teóricos

2.1 Introdução

Como foi descrito no capítulo introdutório desta tese de doutorado, este trabalho concerne à modelagem e controle de sistemas que respondem a uma arquitetura híbrida. Neste capítulo é realizada uma revisão dos conceitos mais importantes de interesse para o desenvolvimento deste trabalho. Inicialmente, foi realizada uma descrição dos aspectos mais relevantes dos sistemas de manufatura industrial, a evolução dos modelos mais destacados e trabalhos desenvolvidos nesta área. Além disso, foi realizada uma revisão da literatura referente a ferramentas de modelagem de sistemas de arquitetura híbrida, especificamente no referente à modelagem através de DEVS e QSS, finalmente é realizada uma revisão bibliográfica do conceito de prototipagem rápida, baseado na técnica de Hardware-In-the-Loop (HIL), que permite o desenvolvimento e teste de sistema de controle industrial utilizando sistemas embarcados trabalhando em tempo real.

2.2 Sistemas de Manufatura, Conceitos e Definições.

Uma das áreas de maior contribuição no desenvolvimento econômico e social de um país é a manufatura. Desde o início do século XVIII, tem sido a principal força de crescimento que tem contribuído significativamente na dinamização e impulso do modelo econômico de muitas das nações. Pode-se afirmar que a manufatura é a coluna vertebral de qualquer nação industrializada, com uma relação diretamente proporcional entre a atividade manufatureira e o nível da vida de um país (LIHUI e WEIMING, 2007).

A manufatura evoluiu como consequência do desenvolvimento do capitalismo, foi inicialmente realizada pelo homem, até chegar às denominadas revoluções industriais. Mair (1984) considera a primeira revolução industrial, como a utilização das máquinas de vapor no

impulso dos sistemas de produção. Além disso, considera que a segunda revolução foi dada pela utilização da energia elétrica e os sistemas mecatrônicos nos sistemas produtivos.

Segundo Thue (2008) a terceira revolução industrial, acontece na atualidade com as tecnologias de informação e as comunicações (TICs). Essas levaram para o conceito da industrialização da produção, através de estratégias de controle, supervisão, coordenação em rede dos processos contínuos e a cibernetização. De acordo com Rifkin (2008), um adicional da terceira revolução industrial, está na utilização das energias renováveis, o que segundo ele, permite garantir o futuro da raça humana. Tem sido as revoluções industriais que permitem grandes mudanças na implementação e configuração dos sistemas produtivos ou sistemas de manufatura.

Kalpakjian e Schmid (2001) define um sistema de manufatura como uma atividade que se encarrega de realizar a transformação de matérias primas em produtos industrializados para sua distribuição no mercado. Essa transformação inclui as fases de projeto do produto, escolha da matéria prima, planejamento dos processos, compras, controle de produção, serviço de apoio, marketing, vendas, distribuição do produto, serviço ao cliente, entre outras.

No transcorrer do tempo, a indústria tem pesquisado diferentes maneiras de organizar seu sistema produtivo, o propósito é melhorar a produtividade de seus processos para serem mais competitivos no mercado mundial. Quando se quer realizar uma análise da evolução dos sistemas produtivos, sempre se referencia os sistemas produtivos das montadoras de automóveis, pois é o setor com maior atividade em seus sistemas produtivos. Alguns deles são apresentados nas seguintes seções.

2.2.1 Modelo de Produção Fordista

Desenvolvido por Henry Ford, no início do século XX, aplica-se o conceito do modelo produtivo proposto por Taylor ao qual ele agregou a mecanização (LIPIETZ, 1994).

Ford que tinha o conhecimento das necessidades do campo, por sua origem (filho de um fazendeiro irlandês), percebeu que tais como os donos de granja como os profissionais independentes precisavam mobilizar-se. Isso permitiu a Ford desenvolver o modelo T, o que representava uma solução prática para a mobilidade das famílias, sem grandes custos ao ser utilizado por somente uma pessoa. A ideia era conquistar mais clientes, com uma versão econômica, sem grandes luxos, com peças padronizadas que fossem facilmente intercambiáveis por qualquer pessoa que não precisasse de uma grande especialização, com várias carrocerias que satisfizessem as necessidades de seus clientes (BOYER e FREYSSENET, 2001).

O modelo de Ford não foi um modelo baseado somente nas atividades do trabalhador, se fundamentou na organização de linha de montagem, com o propósito de gerar produção em massa, a qual foi uma das características relevantes do modelo. Com isso, se conseguiu reduzir os custos de produção, permitindo a Ford baixar o preço de venda de seus veículos e tornando se um modelo de produção que foi implementado em outros sistemas produtivos no inicio do século XX.

2.2.2 Manufatura Enxuta

Como resposta à crise do Fordismo na década dos anos 50, embora alcançasse seu maior auge na década de 70, nasce na companhia Toyota o conceito de produção enxuta, também conhecida como JIT (Just In Time) ou modelo Toyota. É um modelo não baseado na produção em massa, com um conceito mais integral que afeta a organização da empresa em sua totalidade, não só envolve o processo de produção, mas também os aspectos gerenciais da mesma (PIAO; HUANG; WANG, 2007).

Os aspectos importantes no que está baseado o modelo são a eliminação de qualquer fonte de desperdício no interior da empresa, a atitude para a qualidade "Fazer bem a primeira vez" e a participação dos empregados (ESPEJO e MOYANO, 2007). O modelo se fundamenta na lógica de não produzir nada até que seja necessário.

Vários aspectos mudaram com respeito ao modelo Fordista, entre eles, a integração do operário no modelo produtivo, agora o operário deixa de ser um operário especializado em apenas uma tarefa, para se tornar um operário mais qualificado, com uma maior participação nas decisões da empresa, com a possibilidade de desempenhar diferentes atividades e ser capaz de trabalhar com várias máquinas simultaneamente.

Além dos aspectos operacionais, o modelo de produção enxuta incorporou ferramentas e metodologias em seus processos produtivos. Isso facilitou a análise dos problemas que poderiam presentar-se em uma linha de montagem. Além da maneira de soluciona-los através de conceitos de melhora da qualidade (RAJADELL e SÁNCHEZ, 2010). Dentro desses conceitos estão os diagramas de causa-efeito, para fazer análise dos problemas do sistema produtivo. O conceito das 5S, com o propósito de ter sistemas produtivos melhor organizados, mais ordenados e limpos. O conceito de layout celular para ter fábricas, onde sua distribuição seja modular. Tudo isso acompanhado de uma política de motivação para os funcionários, com alto grau de participação nas decisões da empresa (DE SOUZA, 2004).

2.2.3 Manufatura Flexível

Depois do modelo enxuto e devido à evolução da indústria eletrônica e do computador, começou-se a falar na indústria sobre o conceito da automação. Conseguiram-se desenvolver melhores máquinas e ferramentas que levaram a transformar os meios de manufatura às vezes rígidos. (ROCHA e RAMOS, 1994); (REZAIE *et al*., 2009).

Conseguiu-se integrar as redes de comunicação, os robôs e os controladores de lógica programável (CLP), o que permitiu desenvolver melhor as células de trabalho automatizadas que foram a base dos sistemas de manufatura flexível (SMF).

A característica principal do modelo de produção SMF, é organizar as máquinas para trabalhar sequencialmente, conectadas através dos sistemas de transferência de materiais

automatizados, geralmente uma esteira, com um controle central governado por um computador (GAITHER e FRAZIER, 2000).

Utilizando redes de comunicação, cada estação de trabalho e o sistema de transferência recebem as instruções da maneira como devem trabalhar. A flexibilidade consistia em poder mudar com facilidade a configuração do sistema produtivo, mudando simplesmente alguns parâmetros na programação, feita desde os computadores utilizados para o controle das estações (HEIZER e RENDER, 2001).

Uma das grandes vantagens do SMF é permitir configurar sistemas produtivos com capacidade de produzir lotes pequenos com variedade de modelos. Além disso, o sistema produtivo virou menos dependente da mão de obra não qualificada, para um sistema de produção automatizado que facilita a integração (SAVORY; MACKULAK; COCHRAN, 1991).

2.2.4 Sistemas de Manufatura Integrada por Computador

Seguida à manufatura flexível, nasce o conceito da manufatura integrada por computador (CIM). Essa filosofia de produção evoluiu como resultado do avanço dos sistemas computacionais, que se colocaram a serviço dos processos produtivos. A utilização dessas ferramentas permitiu agilizarem-se as etapas de projeto e a fabricação de um produto (AARDAL, 1995).

A ideia na arquitetura CIM é que todas as operações da empresa que estão relacionadas com o sistema produtivo sejam integradas através de sistemas computacionais, os quais permitam ter controle desde o projeto do produto passando pelo recebimento e avaliação das matérias primas, continuando com o desenvolvimento, até chegar ao armazenamento e distribuição (GAITHER e FRAZIER, 2000).

O modelo CIM correspondeu a uma arquitetura piramidal de integração, onde se conseguiu dividir o sistema de produção em vários níveis, que foram integrados através de redes de

comunicação. Nesse modelo se definiu um primeiro nível que corresponde ao chão de fábrica, num segundo nível se estabeleceram os sistemas de controle e definiu se também atividades de projeto e prototipagem do produto, utilizando conceitos como o Projeto Auxiliado por Computador (CAD), manufatura auxiliada por computador, Engenharia Auxiliada por Computador (CAE), prototipagem rápida entre outros (KATEEL; KAMATH; PRATT, 1996); (WANG; ZHANG; LIU, 2007).

Integrado aos dois primeiros níveis, tem se o nível de supervisão industrial, através de sistemas SCADA (Supervisory Control Architecture and Data Acquisition), onde é realizado o monitoramento e a configuração do sistema produtivo. Um nível superior, da pirâmide CIM se tem a parte gerencial do processo, é o nível que permite configurar o sistema produtivo, foram estruturados os sistemas de planejamento de necessidades de materiais (MRP) e planejamento dos recursos de manufatura (ERP) principalmente (RAY, 1988; ZHAO e CHEN, 2009).

2.2.5 Sistemas Inteligentes de Manufatura

Como resultado da rigidez na hierarquia do modelo CIM, nasceu a ideia da manufatura baseada em agentes, que recebeu principal atenção no início dos anos noventa, quando se realizaram os primeiros casos de estudo no programa japonês “sistemas de manufatura inteligente”, e na “U.S. National Center for Manufacturing Science”, que decidiram iniciar vários programas de manufatura baseado em agentes.

Não existindo um consenso na definição do conceito de agente, pode se encontrar uma definição geral, como uma aplicação da ciência da computação na área da inteligência artificial, como uma entidade única que está na capacidade de perceber o entorno através dos sensores, que lhe permite realizar ações de maneira autônoma e flexível através de seus atuadores Castelfranchi (1997).

Os agentes estão caracterizados de acordo com as seguintes propriedades:

Autonomia: A capacidade dos agentes de interagir de maneira independente, sem nenhum tipo de controle de suas ações.

Habilidade social: que lhe permita interagir com outros agentes, inclusive com agentes humanos.

Reativos: capacidade de perceber um ambiente para responder de maneira oportuna às mudanças do mesmo.

Pró-Atividade: é a capacidade que deve ter um agente para interagir sob sua própria iniciativa com outros agentes, responder às mudanças do ambiente no qual se encontra.

Levando em conta as características mais relevantes dos agentes, os quais podem ser percebidos como componente tecnológico que podem realizar uma tarefa, foi desenvolvida a estrutura organizacional de produção denominada de multiagentes, a qual é a organização de vários agentes, que através de regras e padrões de interação estabelecidos, se coordenam entre si na procura de alcançar objetivos gerais do sistema, ou os objetivos particulares de cada agente (ARGENTE, 2008).

Esse tipo de organização dos sistemas de produção industrial tem tido um pilar fundamental nos sistemas de controle, iniciando com a modelagem a análises e o projeto de estratégias de controle que permitem às máquinas trabalharem de acordo com os parâmetros estabelecidos.

Uma descrição dos conceitos básicos da modelagem e controle dos processos de produção industrial é apresentada na seção 2.3.

2.3 Modelagem e Controle da Dinâmica dos Sistemas de Manufatura.

O controle automático tem realizado um grande aporte na evolução dos sistemas de produção industrial, garantindo que os processos cumpram certas especificações de funcionamento e permitindo configurações dinâmicas dos sistemas produtivos (HOSSAIN e SUYUT, 1997).

Um aspecto importante na implementação de estratégias de controle nos processos industriais, é a representação matemática de sua dinâmica. Isso permite realizar a análise do comportamento dos sistemas, no tempo e na frequência, o qual facilita a simulação e emulação dos processos utilizando software de computador ou sistemas embarcados para tal fim.

Em sua evolução os sistemas de produção industrial, estão constituídos por dinâmicas que a principio só funcionavam em relação ao tempo, mas com a evolução tecnológica nasceu uma nova dinâmica que responde em relação a eventos e conformaram sistemas de arquitetura híbrida (KHAN, 2007).

Na seção 2.3.1se realiza um breve histórico da evolução da dinâmica em relação ao tempo dos sistemas de produção industrial.

2.3.1 Sistemas de Dinâmica Contínua

Antes do século XVIII, a produção de bens e consumo era meramente artesanal, até a invenção da máquina propulsionada pelo vapor, criada no ano de 1782 por James Watt (1736-1819). Essa permitiu substituir a mão de obra do homem pelas máquinas, esse foi o início da chamada primeira revolução industrial. Sendo implementadas com sucesso as primeiras linhas de montagem, inicialmente arquitetadas por Henry Ford. Isso criou a necessidade de projetar máquinas cada vez mais precisas, rápidas e de alta qualidade, trazendo consequência a evolução dos sistemas de controle industrial.

Controle industrial que teve o começo com o controle de velocidade proposto por Watt, que tinha dificuldades para trabalhar a altas velocidades, pois ocorriam problemas na estabilidade no sistema de controle. Isso cria a necessidade de representar a dinâmica do sistema através de seu modelo matemático, e foi G. B. Airy (1801-1892) quem apresentou a dinâmica do regulador de Watt modelado através de Equações diferenciais. Estas foram posteriormente utilizadas por Maxwell (1868), com o propósito de estabelecer as condições de estabilidade para os sistemas de controle.

Foi o trabalho apresentado por Maxwell (1868), a origem da formalização matemática, que permitiu representar através de Equações diferenciais lineares a dinâmica de um sistema e o conhecimento da teoria do controle. Após essa importante contribuição, surgiriam novos aportes especificamente na análise da estabilidade dos sistemas de controle, alguns deles desenvolvidos por Routh (1877) e Hurwitz em 1895. No ano de 1897 o matemático russo A. Lyapunov realizou uma descrição formal da teoria de controle em um trabalho que foi inicialmente traduzido para o francês Liapounoff (1907) e que permitiu aos russos liderarem a teoria de modelagem e análise de sistemas não lineares. Após isso, foi interessante a utilização de métodos de análise de resposta em frequência dos sistemas de controle, especificamente o trabalho realizado por Bode (1940).

Nessa mesma época muitos dos sistemas de controle industrial utilizavam o controlador Proporcional Integral Derivativo (PID), os quais eram sintonizados através das regras propostas por Ziegler-Nichols ou projetados através do método proposto por Evans (1948) do lugar geométrico das raízes.

Ao final da década de 50, os métodos no domínio da frequência e do lugar geométrico das raízes constituíram o núcleo da teoria do controle clássico. Nessa época, a teoria do controle tinha alcançado sua madureza e consolidação através das aplicações nos sistemas de produção industrial, os sistemas utilizados na primeira guerra mundial e o início da era aeroespacial.

Uma característica importante até a década de 50 do século XX foi a aplicação da teoria de controle na modelagem, análise da estabilidade e projeto de sistemas de controle no tempo contínuo, aplicável em sistemas de produção cuja dinâmica envolvia o comportamento das variáveis físicas descritas em relação ao tempo, como temperatura, fluxo, nível, pressão, velocidade, entre outras. As mesmas estavam presentes na maioria dos processos de produção industrial dessa época, e a ferramenta matemática utilizada para descrever o comportamento dos processos produtivos e que virou o paradigma da modelagem desse tipo de sistema foram as Equações diferenciais.

Na década dos anos de 1950s, iniciou-se o desenvolvimento dos dispositivos digitais e essa evolução tecnológica permitiram ao homem criar uma nova dinâmica, a dinâmica regida pela ocorrência de eventos. Uma descrição da evolução dos sistemas que responde dinamicamente em relação a eventos é realizada na seção 2.3.2.

2.3.2 Sistemas Dinâmicos Descritos Através de Eventos

A partir dos anos 1960s, com a evolução dos sistemas digitais e o surgimento do computador, o homem cria uma nova dinâmica de sistemas os quais são regidos pela ocorrência de eventos (CASSANDRAS e LAFORTUNE, 2008).

É assim que a tecnologia permitiria desenvolver sistemas cada vez mais complexos, como a arquitetura do computador, as redes de comunicação, sistemas de tráfego em terra, mar e ar, logística de negócios, projeto de software, C3I "Military Systems" e os sistemas de manufatura industrial, são alguns dos sistemas que respondem à ocorrência assíncrona de eventos e que tradicionalmente são denominados sistemas de eventos discretos (SED).

Um SED está caracterizado por possuir uma dinâmica que evolui de acordo com a ocorrência abrupta de eventos físicos. Esses eventos geralmente sucedem de maneira assincrônica no tempo. Os eventos estimulam à dinâmica causando uma transição do estado no sistema (RAMADGE e WONHAM, 1989).

Um evento pode corresponder, por exemplo, com a ocorrência da finalização de uma tarefa em um sistema de produção, à chegada e saídas de clientes de uma loja, à transmissão de um pacote em um sistema de comunicação, à chegada de requisições e seu atendimento por um servidor, a falha de uma máquina em um sistema de fabricação, à chegada e saída de peças em um processo industrial, à ligação ou desligamento de um sinal pelo operador de um processo.

(HRÚZ e ZHOU, 2007) realizam a apresentação de um SED e definem os estados como os valores que podem tomar o sistema no transcorrer do tempo, no caso da Figura 2.1 q1, q2, q3, q4. Além disso, definem os eventos como os impulsos ocorridos no tempo e que estimulam a

evolução do sistema, neste exemplo e_1 ate e_7. É importante destacar como a mudança de estado do SED acontece de maneira assincrônica no tempo.

Uma característica importante na análise dinâmica dos SED é a inoperância das ferramentas matemáticas, centradas especificamente nas Equações diferenciais e nas Equações diferenças, utilizadas durante muitas décadas, desenvolvidas para modelar e projetar sistemas de controle aplicáveis em processos governados pelas leis da natureza. Mas esses métodos não são agora os apropriados para o estudo dos complexos sistemas desenvolvidos no final do século XX e os inicios do século XX1 (WAINER, 2009).

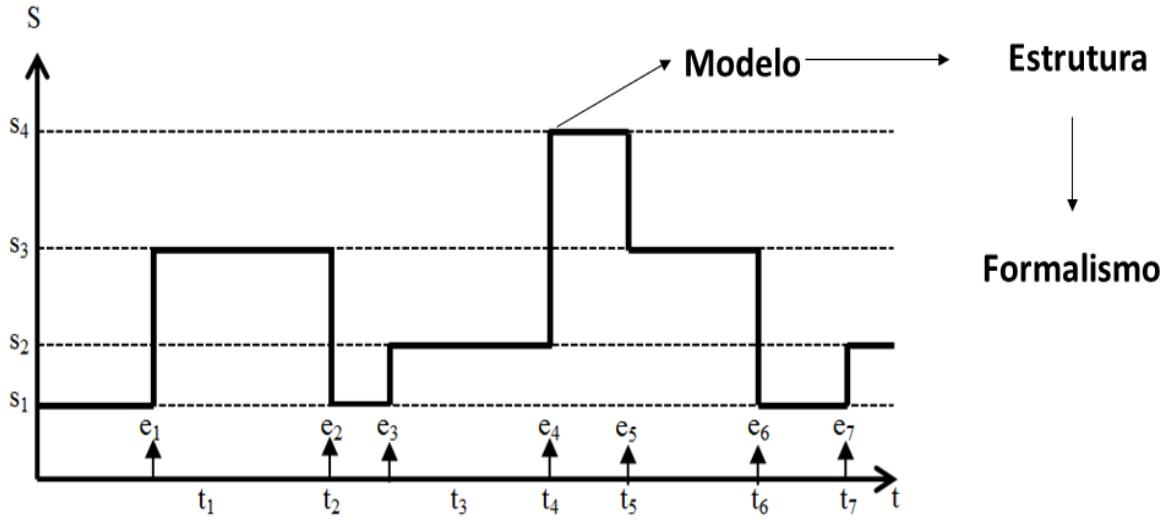

Figura 2.1 Dinâmica de Eventos Discretos

Essa necessidade marcada no desenvolvimento de métodos matemáticos que resolvam os problemas práticos das diferentes áreas que trabalham em função de eventos, foi rapidamente coberta com propostas de formalismos que permitiram descrever o comportamento desse tipo de sistemas, além de projetar estruturas de controle (DAÍ, 2001).

São vários os formalismos desenvolvidos para a modelagem, análise e controle de sistemas descritos através de eventos. Entre os mais destacados estão os autômatos finitos, as redes de Petri e o formalismo DEVS (Discrete Event System Specification). Mas sendo assim nenhuma dessas ferramentas de modelagem converteu-se ainda em um paradigma que permita destacar em

respeito às outras, tendo cada uma sua potencialidade. Mas o formalismo DEVS, desenvolvido pelo matemático Bernard Zeigler nos anos 70, por estar baseado na teoria de sistemas, vem-se tornando um formalismo universal, portanto todos os outros formalismos utilizados na modelagem de sistemas de eventos discretos podem ser absorvidos por DEVS (ZEIGLER e VAHIE, 1993).

Mas os processos industriais modernos estão constituídos, além da dinâmica que responde em relação ao tempo, pela dinâmica em função de eventos, constituindo sistemas de arquitetura híbrida. Uma introdução a esse tipo de sistemas se realiza na seção 2.3.3.

2.3.3 Sistema de Manufatura de Arquitetura Híbrida.

Tradicionalmente, a investigação no controle de processos industriais tem sido focada para a análise de sistemas de dinâmica continua ou de eventos discretos, as quais são analisadas usualmente de maneira independente, mas são vários os processos industriais nos quais essas duas dinâmicas interatuam entre si (PEPYNE e CASSANDRAS, 2000).

Essa interação da dinâmica contínua com dinâmica descrita em relação a eventos em um processo deu origem para um novo tipo de sistemas, como são os sistemas de arquitetura híbrida (SAMAD, 2000).

Para Mosterman (2007) aput Mosterman (2002, p. 5), a representação da dinâmica de um sistema de arquitetura híbrida, considera três diferentes tipos de comportamentos, como se apresentam na Figura 2.2.

Um primeiro comportamento leva em consideração a dinâmica continua, a qual tipicamente é modelada através das Equações diferenciais que tradicionalmente são solucionadas por meio de métodos de integração numérica, para facilitar sua simulação (representado na Figura 2.2 por f). Os eventos gerados da detecção dos umbrais e as transições das variáveis de dinâmica continua (representado na Figura 2.2 por γ). Os eventos para a inicialização das variáveis de estado de

dinâmica continua (representado na Figura 2.2 por g). Esta interação de variáveis regidas em relação ao tempo e em relação a eventos discretos constituem um sistema de arquitetura híbrida.

Mosterman (2002) considera que os sistemas de dinâmica continua, que apresentam não linearidades e pendentes pronunciadas, que dificultam a simulação, baseada em métodos de integração numérica e que são considerados sistemas rígidos, podem ser modelados de maneira simples, se esse tipo de sistema é considerado como um sistema híbrido, levando em conta seu comportamento como contínuo por seções e analisando as descontinuidades como eventos.

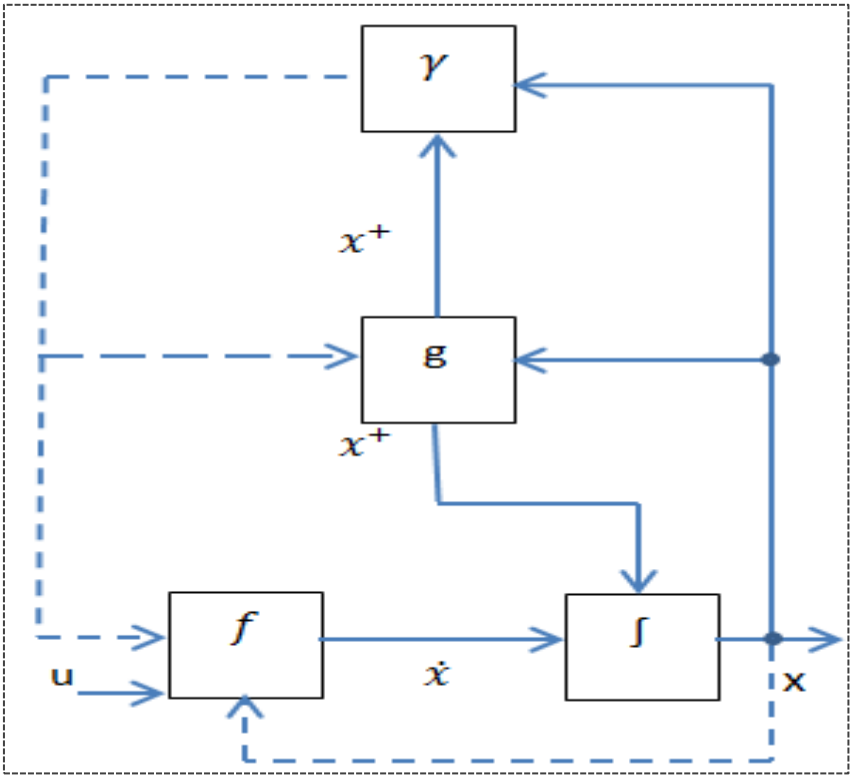

Figura 2.2. Dinâmica de um Sistema de Arquitetura Híbrida

Para Antsaklis (2000), a investigação de sistemas híbridos é uma nova e interessante área que realiza uma interação da engenharia de controle, matemática e a ciência da computação. Permitindo a combinação dos sinais de um sistema que podem tomar valores em um conjunto contínuo, com valores reais que podem ser descritos através de Equações diferenciais ordinárias e valores em um conjunto discreto, em geral finito, dentro dos números naturais, que podem ser descritos através de formalismos como autômatos finitos e outros.

Heemels., et al (2009) apresentam diversas razões para estudar um sistema de arquitetura híbrida. A primeira é a interação de tecnologias que envolvem o projeto de um sistema, como os sistemas mecatrônicos. A segunda, pelos sistemas de controle baseados em rede, como os sistemas de manufatura, controle de tráfego aéreo e terrestre, entre outros, e a terceira, os sistemas que em sua dinâmica possuem não linearidades ou descontinuidades.

Mesmo assim os sistemas de arquitetura híbrida tem despertado um grande interesse na atualidade, há décadas se tem referências de sua abordagem. Uma das primeiras conhecidas foi realizada por Witsenhausen (1966), quem fez uma descrição dos sistemas contínuos que interagem com variáveis booleanas, realizando uma proposta de controle ótimo, para esse tipo de sistema.

Cellier (1979) desenvolveu em seu trabalho uma técnica de simulação de sistemas contínuos e a eventos discretos. Stiver e Antsaklis (1992) realizaram uma proposta formal de modelagem de sistemas híbridos, de igual maneira Mosterman e Biswas (1996) realizaram uma proposta de modelagem baseando sua análise através do formalismo "Bond Graph".

Lygeros; Godbole; Sastry (1998) fizeram uma proposta com o intuito de aumentar a segurança e melhorar o rendimento nas rodovias, através da automação do tráfego. Foi um exemplo de grande escala, de um sistema de dinâmica complexa, considerando-se o conceito de multiagentes. A ideia foi à modelagem do sistema com uma grande quantidade de agentes (veículos), que através de sensores de comunicação e um sistema de controle híbrido poderiam realizar um uso mais eficiente do percurso, no caso de uma rodovia. A dinâmica contínua estava refletida na velocidade dos agentes e a dinâmica de eventos foram os sinais dos sensores que indicavam a colisão dos agentes. A proposta foi baseada na estrutura de autômatos híbridos, eles conseguiram propor um sistema de controle que regulava o tráfego nas rodovias, evitando colisões e aumentando os níveis de segurança.

Mosterman; Vangheluwe (2004) apresentam a dinâmica do sistema de subida ou descida do vidro da janela do veículo como um sistema de arquitetura híbrida. Nesse processo, interagem a

dinâmica do motor de corrente contínua no controle de velocidade de deslocamento e a corrente do mesmo. Além disso, podem ser obtidas informações de comando pelo sistema de comunicação, indicando os eventos de comando dados pelo usuário. Esse trabalho permitiu-lhe propor o conceito de multiparadigma de modelagem para automação computorizada, na qual consiste nas interações entre os paradigmas de descrição da dinâmica baseada em eventos e a teoria de controle que representa a dinâmica das variáveis físicas.

Goebel; Sanfelice; Teel (2009) apresentaram uma proposta pela descrição da modelagem da dinâmica de sistemas híbridos, realizando uma análise de estabilidade e projeto de sistemas de controle robusto. O trabalho está acompanhado de vários exemplos que ilustram os conceitos fundamentais no tema de interesse.

Nesses trabalhos foram utilizadas diferentes ferramentas matemáticas na descrição de sistema de arquitetura híbrida, uma descrição de uma delas se realiza na seção 2.4.

2.4 Modelagem e Controle de Sistemas de Arquitetura Híbrida

De igual maneira que os sistemas que trabalham em relação ao tempo e os sistemas em relação a eventos, foram necessários desenvolver ferramentas matemáticas que permitiram realizar a descrição dos sistemas de arquitetura híbrida.

Como uma evolução do autômato celular proposto por Von Neumann (1966) e uma generalização do autômato temporizado, surge a proposta de modelagem através do autômato híbrido (ALURY *et al*., 1995). Além disso, como são utilizadas as redes de Petri na modelagem e controle de sistemas de eventos discretos, essas foram apresentadas como formalismo para modelar o sistema em tempo contínuo por (DAVID e ALLA 1987), convertendo-se pronto em um formalismo que permite a representação das duas dinâmicas (de tempo contínuo e de eventos discretos) em um único modelo (LEBAIL; ALLA; DAVID 1991).

De igual maneira que evoluíram os autômatos híbridos e a redes de Petri como formalismos para representar a dinâmica híbrida dos sistemas, o formalismo DEVS pode ser utilizado na

representação da descrição dos sistemas em tempo contínuo, aproximando as Equações diferenciais ordinárias por sistemas de eventos discretos, dando origem aos métodos de integração por quantificação (ZEIGLER; PRAEHOFER; KIM, 2000) o que permitira a (KOFMAN, 2003) propor um método numérico de integração de Equações diferenciais ordinárias que denominou sistemas de estados quantificados (QSS).

Considerando-se os formalismos a serem utilizados na proposta de modelagem, controle e sínteses neste trabalho, nas seções 2.4.1 e 2.4.2 se realizarão uma descrição dos formalismos utilizados.

2.4.1 Formalismo DEVS

Tradicionalmente durante muito tempo têm sido utilizadas ferramentas matemáticas como as Equações diferenciais e em diferenças, para representar o comportamento dos sistemas governados pelo tempo, Atualmente, considerando-se desenvolvimentos na área tecnológica, cujo comportamento é governado por eventos de ocorrência assíncrona no tempo, se tem desenvolvido novas teorias de modelagem (HONG e KIMM, 2006). Uma delas foi apresentada em 1972, pelo matemático Bernard Zeigler, quem propõe um formalismo para representar os SED e que denominou DEVS (PALANIAPPAN; SAWHNEY; SARJOUGHIAN, 2006)

NIKOLAIDOU *et al*., (2008) descrevem o formalismo DEVS como uma ferramenta conceitual para especificar o modelo e a simulação de sistemas de eventos discretos. A modelagem pode ser feita através da decomposição do sistema em modelos menores, especificando o acoplamento entre eles.

Os modelos menores estão definidos como o modelo atômico, que representa a unidade "molecular" de processamento e se constitui no elemento fundamental do modelo (BERGERO *et al*, 2008; ZEIGLER *et al*, 1996).

Esses modelos atômicos podem interagir através de um modelo acoplado. Isto permite dividir um problema de modelagem de um sistema complexo em unidades pequenas que possam ser acopladas e tornam mais fácil a sua representação (ZEIGLER, 2003).

São várias as áreas em que tem sido utilizado o formalismo DEVS para representar a dinâmica e a simulação de sistema.

Um dos trabalhos em que foi realizada uma aplicação do DEVS é apresentado por Capocchi *et al.*, (2003). Eles realizaram uma proposta de um método de transformação de modelos desenvolvidos em VHDL para modelos DEVS acoplados. Como validação do método realizaram a modelagem e simulação de um registrador de oito bits, deixando aberta a possibilidade de desenvolver uma modelagem integrada e um ambiente de simulação para essa tarefa de conversão de modelos.

De Gentili; De Cicco; Santucci (2005) descrevem a modelagem de um processo de fabricação de produtos lácteos através do formalismo DEVS. Através do modelo proposto, conseguiram separar o modelo do sistema de controle do modelo de dados, providenciando flexibilidade e portabilidade no desenvolvimento do modelo geral do processo, destacando a facilidade do modelo DEVS acoplado, para permitir representar em unidades um modelo de um sistema complexo.

Na área da Robótica Hu; Ganapathy; Zeigler (2005) apresentam um trabalho focando o estudo da robótica colaborativa baseada no projeto incremental, utilizando a simulação como primeiro passo para a evolução gradual dos robôs reais. Os modelos e o entorno de simulação desenvolvido estão baseados no formalismo DEVS. Com os resultados quantitativos das simulações e os experimentos, eles puderam demonstrar a facilidade de implementação deste modelo do projeto.

Orooji; Sarjoughian; Taghiyareh (2010), realizaram uma modelagem e simulação de um sistema baseado no conceito de multiagentes, de uma plataforma de ajuda acadêmica, que dá aos alunos uma assessoria da oferta de cursos que podem escolher no momento de realizarem suas

matrículas. A modelagem e posterior simulação permitiu lhes validar sua proposta, antes de ser implementada.

A modelagem dos SED tem o propósito de permitir a análise de seu comportamento. Além disso, permite realizar propostas de sistemas de controle que garantam um comportamento desejado. Os conceitos de modelagem de SED têm sido evoluídos como formalismos para representar e controlar a dinâmica dos sistemas que são regidos em função do tempo, dando origem ao controle de processos de dinâmica contínua através de eventos. Uma descrição de um formalismo utilizado para os SED é apresentada na seção 2.4.2.

2.4.2 Controle Baseado em Eventos

Geralmente quando se realiza um sistema de controle de variáveis regidas em função do tempo utilizando sistemas digitais, é necessário efetuar uma discretização dessas variáveis. Essa discretização na maioria das aplicações se realiza através de conversores analógicos para digitais que trabalham de maneira sincrônica. Ou seja, a conversão gera mostras do sinal de entrada em intervalos de tempo precisos definidos por um período de amostragem (ARZEN, 1999). O problema com esse tipo de discretização do sinal é a perda de controle que sucede nos intervalos de cada uma das amostras, o que pode levar o sistema controlado ao campo da instabilidade (KOFMAN, 2003). Uma alternativa promissora é o controle através de eventos, controle aperiódico ou assíncrono. Esse se fundamenta em uma discretização através dos eventos produzidos pela mudança do processo, a amostragem será realizada unicamente quando ocorram eventos significativos produto dessa mudança, por exemplo, quando uma variável exceda um limite estabelecido ou um “encoder” mude seu sinal (CERVIN e ASTROM, 2007).

São várias as aplicações de sistemas de controle que tem se implementado, baseado no conceito de *controle* através de eventos, mas ainda são escassas as aplicações que no campo industrial utiliza esse tipo de controle nos níveis inferiores da hierarquia de um sistema automatizado, são mais utilizadas nos níveis superiores da hierarquia, por exemplo, no nível de supervisão (DORMIDO; SANCHEZ; KOFMAN, 2008).

Com o propósito de realizar o controle das variáveis regidas em função do tempo, nos processos industriais utilizando o conceito de eventos discretos, (KOFMAN e JUNCO, 2001) desenvolveram um método de integração que denominaram sistemas de estados quantificados (QSS). Eles partiram da proposta realizada por Zeigler; Praehofer; Kim, (2000) para simular sistemas em tempo contínuo através da quantização de estados, e propuseram uma discretização das variáveis de estado baseado nos eventos produzido pela mudança do sinal, em um intervalo definido que denominaram quantum. Adicionalmente à proposta da quantização de estados, QSS adiciona uma histerese que evita as oscilações apresentadas nos sistemas que são discretizados utilizando conversões sincrônicas. Além de, melhorar a resposta na dinâmica de controle em um processo industrial, QSS permite utilizar uma menor quantidade de capacidade computacional no momento de ser implementado, pois somente quando ocorre uma mudança significativa das variáveis é realizada a quantização dos sinais.

São vários os métodos de quantização desenvolvidos por Kofman, entre eles estão os métodos de quantização de primeira ordem, o qual utiliza uma função linear como sinal de quantificação. Além disso, propõe o método de quantificação de segunda ordem, o método de quantização para sistemas rígidos, entre outros.

Baseado no conceito de quantificação por estados, Kofman (2005) propõe um sistema de controle por estados quantificados QSC (Quantized State Control). O método está baseado na implementação de controladores digitais, os quais são projetados primeiro como controladores contínuos, realizando posteriormente a quantização de suas variáveis de estado através de QSS . Esse tipo de controle digital é descrito como um sistema de eventos discretos baseado num sistema de amostragem assíncrono.

Entre os trabalhos desenvolvidos no controle de sistemas contínuos através de eventos, estão o apresentado por Lygeros *et al*.,(2003). Eles analisam as propriedades dinâmicas do autômato híbrido o qual é uma linguagem para modelagem e análise de sistemas dinâmicos contínuos e de eventos discretos e que precisam ser avaliados em tempo real. Como resultado relevante deste trabalho está a análise da estabilidade para esses tipos de sistemas.

Alminde; Bendtsen; Jakob (2006) realizaram a descrição do projeto e implementação de uma ferramenta para simulação de sistemas híbridos, especificamente trabalhando em linha e baseados neste trabalho, Alminde et al., (2007) apresentaram um método de controle complexo para plantas de multiplex entradas e saídas (MIMO) baseado no método de integração QSS.

Dormido; Sánchez; Kofman, (2008) apresentam uma proposta de controle e comunicação de sistemas baseado em eventos como uma aplicação do algoritmo de controle desenvolvido por (ÅRZÉN, 1999), onde destaca a pouca maturidade ainda dessa teoria e as poucas aplicações significativas no campo industrial.

(BERGUERO *et al*., 2008) realizaram um simulador para sistemas híbridos em tempo real, apresentando em seu trabalho diferentes exemplos de aplicação da plataforma proposta.

(MIGONI e KOFMAN, 2009) eles apresentaram dois métodos de integração para a solução de Equações diferenciais ordinárias de modelos rígidos. Esses métodos comparados com os métodos de integração tradicional permitiram obter um ganho importante no custo computacional. Além disso, puderam demonstrar a propriedade de erro limitado do método QSS.

Com o propósito de permitir implementações mais rápidas e ajustadas com a medidas das necessidades das indústrias, neste trabalho foi proposto um conceito de desenvolvimento de produto baseado na prototipagem rápida. Na seção 2.5, será apresenta uma descrição do conceito da prototipagem rápida e ferramentas para o seu desenvolvimento.

2.5 Conceitos e Ferramentas da Prototipagem Rápida

Um aspecto importante para levar-se em conta no elevado custo de fabricação, está relacionado com o desenvolvimento do produto e especificamente os produtos que envolvem sistemas de controle.

Tradicionalmente a engenharia do controle tem sido desenvolvida baseada no modelo, fundamentado nos conceitos matemáticos que permitem descrever o comportamento de um

processo. Na atualidade são vários os conceitos e ferramentas utilizados para desenvolver produtos baseados no modelo. Algumas destas características foram estruturadas em um modelo em V, que inicialmente fora proposto no desenvolvimento do produto de software e que evoluiu para o desenvolvimento do ciclo de vida do produto na área mecatrônica (ISERMANN, 2008). Na Figura 2.3 se apresenta uma aproximação ao modelo em V para o desenvolvimento de sistemas de controle baseado em modelo, incluindo ferramentas utilizadas nas diferentes fases.

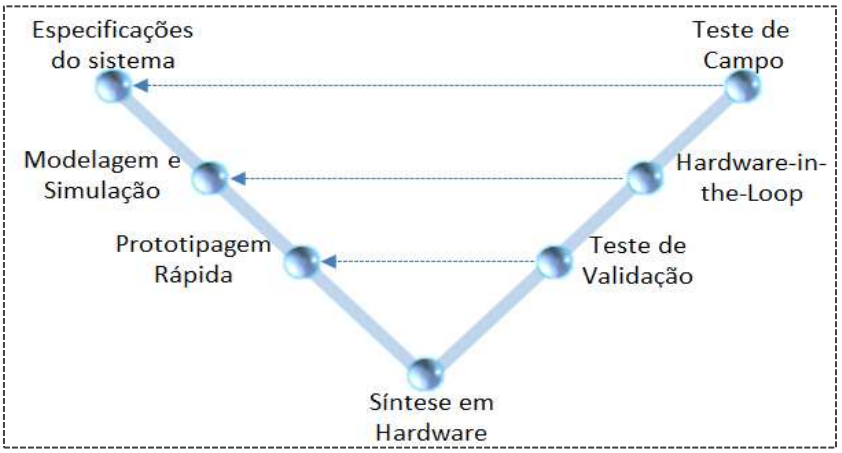

Figura 2.3 Modelo de Desenvolvimento de Produto

Esse modelo está especificado em quatro níveis. O primeiro nível orienta as especificações e requisições do produto e ao teste do produto terminado. Constituindo o princípio e o final do projeto de desenvolvimento. No nível dois se realiza a representação numérica e a simulação do produto. Além disso, é feita a calibração e o teste funcional. No nível três é realizada a emulação do produto baseado no conceito da prototipagem rápida utilizando sistemas embarcados e é realizado o teste do sistema através da interação do sistema embarcado com os sinais reais do processo através de sensores e atuadores. No quarto nível se realiza a implementação do produto.

2.5.1 Prototipagem Rápida de Sistemas de Controle

A prototipagem rápida foi uma ideia desenvolvida nos anos de 1990 na indústria automotiva que foi rapidamente difundida para outras indústrias. Corresponde a um novo conceito de controle personalizado, desenvolvido de maneira direta, rápida e otimizada,

utilizando sistemas embarcados trabalhando no tempo real, com ferramentas padronizadas de blocos predeterminados, que permitem a geração de código de maneira automática (EISEMANN, 2006). O conceito de prototipagem rápida é permitir testes de sistemas de controle implementados em sistemas computacionais, interagindo com o modelo embarcado e seu ambiente de atuação real através de sensores e atuadores. Um esquema do conceito da prototipagem rápida se apresenta na Figura 2.4. A prototipagem está dividida em quatro níveis de desenvolvimento, a especificação e modelo correspondem com as necessidades do cliente. Na simulação, é apresentada uma descrição matemática e uma validação através da simulação do produto. No nível de síntese se realiza uma descrição em VHDL ou em linguagem C, para finalmente embarcar no dispositivo de Hardware.

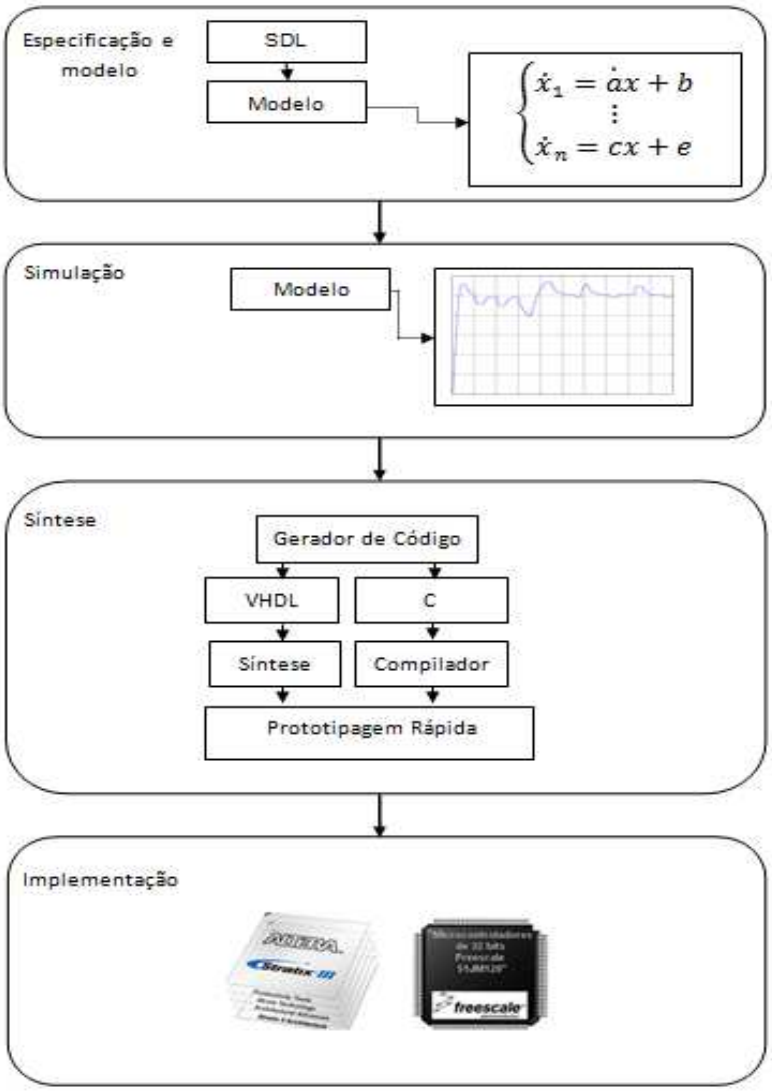

Figura 2.4 Prototipagem Rápida

2.5.2 Hardware-In-the-Loop

A simulação Hardware-In-the-Loop (HIL) é uma das ferramentas mais utilizadas na implementação do conceito de desenvolvimento de produto (MARTIN; SCOTT; EMAMI, 2006). HIL é uma técnica que permite realizar uma representação rápida do modelo virtual de um sistema, em um protótipo físico através de sistemas embarcados, com o intuito de permitir a realização de testes mais pertos com a realidade (DEL SIGNORE; KROVI; MENDEL, 2005).

O HIL permite desenvolver produtos com uma maior rapidez, minimizando custos nos testes de prova, permitindo uma maior rigorosidade na avaliação do produto, permitindo detectar e antecipar possíveis falhas no projeto antes que o produto seja industrializado. Ele difere da prototipagem rápida em que a planta é simulada e a unidade de controle é real. A ideia do conceito HIL é gerar e receber todas as informações provenientes dos sensores existentes na planta real, com o propósito de testar diferentes estratégias de controle.

O HIL teve seu início na indústria aeroespacial e de defesa, principalmente para emular processos de tempo real nos que a vida humana corria perigo. Essa tecnologia foi depois utilizada pela indústria automobilística com o propósito de baixar custos de desenvolvimento dos projetos sendo difundida para outras áreas da indústria (NABI et al., 2004).

Os primeiros trabalhos na área automotiva utilizando HIL foram conhecidos na década de 1980, onde foi utilizado na avaliação de novos componentes para suspensões ativas. Na década dos 90, incrementou-se sua utilização baseado nas Unidades de Controle Eletrônico (ECU), que quando finalizadas as mesmas foram embarcadas para testes de sistemas de comunicação, permitindo na atualidade representar a totalidade da dinâmica de um veículo e suas malhas de controle (PRAEHOFER e WAELTERMANN, 2005).

No contexto da engenharia de produção a simulação HIL, permite que um sistema de controle ligado a uma máquina virtual ou uma planta através de um sistema de comunicação, possa simular o comportamento da máquina, com o propósito de testar estruturas de controle e supervisão (RÖCK, 2011).

Na atualidade são várias as empresas dedicadas a desenvolver ferramentas para implementação do conceito da prototipagem rápida e HIL. Entre algumas destas estão dSPACE®, Altera®, NationalInstruments®, Matlab®, entre outras.

Alguns trabalhos feitos na área da prototipagem rápida foram desenvolvidos por Lindsey, (1992). Ele realizou uma descrição do procedimento automatizado de ensaios aplicados em sistemas eletrônicos. O procedimento executado está constituído de uma metodologia de cinco fases, que incorpora modelos VHDL, permitindo desenvolver aplicações concorrentes em sistemas complexos, associando procedimentos de testes em cada uma das fases. A combinação de ferramentas de software proprietário com o modelado em VHDL permitiu a migração em cada uma das fases. Esse foi um primeiro intento da companhia Motorola para gerar testes automáticos de dados e levou a uma redução significativa dos tempos de testes nos procedimentos manuais. Isto permitiu a detecção de erros na fabricação.

(SCHLAGER; ELMENREICH; WENZEL, 2006) apresentam um trabalho baseado na utilização no conceito de HIL, desenvolveu uma interface que chamaram de transdutor virtual inteligente. Com isto, eles puderam separar a execução do modelo simulado e a interação através de uma interface com seu sistema real.

Uma análise da utilização da ferramenta SimEvents™ de Matlab®, para a simulação de sistemas de controle de arquitetura híbrida, é realizada por (CASSANDRAS; CLUNE; MOSTERMAN, 2006), ele fez uma descrição dos módulos que compõem o SimEvents™ e como pode ser utilizado na simulação de sistemas de eventos discretos, também destaca o potencial da ferramenta de Stateflow®, para a simulação deste mesmo tipo de sistemas.

Li; Cassandras; Clune (2006) utilizou a ferramenta SimEventsTM para a simulação da aplicação da técnica de estimação do gradiente em linha para o controle de potência de unidade de controle eletrônico. Os resultados obtidos permitiram demonstrar como o SimEvents™ pode ser utilizado no projeto de controle para otimizar o uso da energia nos processadores entre os tempos de chegada de trabalhos numa linha de produção.

Ayasun *et al.*, (2007) desenvolveu uma interface de simulação integrada com hardware, para teste e desenvolvimento de equipamento eletrônico. (RUSSO; TERZO; TIMPONE, 2007) apresentaram um sistema de controle da dinâmica de frenagem de um veículo que vira em uma esquina, o que requer o conhecimento da velocidade instantânea das quatro rodas, e propõem um algoritmo de controle que pode ser utilizado em veículos equipados com sistemas de frenagem ABS. O trabalho permitiu apresentar um esquema para a realização de dois casos de teste e validação: em malha aberta e em malha fechada.

(NICA; GANEA; DONCA, 2008) utilizaram a ferramenta de SimEvents™ na simulação de filas em sistemas de manufatura com o propósito de poder definir dimensões dos espaços de armazenagem e permite estabelecer às equipes requeridas para a implementação de uma linha de produção.

Gawthrop *et al*., (2008) desenvolveram uma estratégia de controle baseada na simulação de hardware, aplicada a um atuador. Através dessa emulação conseguiu solucionar os problemas de fidelidade e estabilidade causada pela dinâmica não desejada de sua função de transferência.

Um modelo de controle não linear é desenvolvido por Canale; Fagiano; Razza (2009), com o propósito de melhorar a estabilidade de um veículo, a fim de mostrar de uma maneira mais realista e eficaz da abordagem do sistema de controle, foi implementada a lei de controle em um dispositivo embarcado o qual foi testado através de simulações. Eles demostraram que o sistema tinha um alto amortecimento e fizeram testes a estabilidade variando a exigência do sistema.

Luo e Zhao (2009) desenvolveu uma aplicação de um conversor DC/DC para veículos elétricos, os algoritmos de controle do conversor são implementados sob o conceito da prototipagem rápida e são testados através das ferramentas de MATLAB®, Simulink™ e Stateflow™. Ao final este trabalho mostrou como é fácil o desenvolvimento de controladores utilizando o conceito da prototipagem rápida, podendo converter os algoritmos feitos na ferramenta Stateflow™ ao padrão em código C.

Um elemento essencial na implementação da prototipagem rápida é a CPLD, uma descrição desses dispositivos é realizada na seção 2.5.3.

2.5.3 Dispositivos Reprogramáveis

Uns dos elementos mais utilizados atualmente na implementação dos conceitos de desenvolvimento de produto utilizando HIL e PR são os dispositivos lógicos programáveis do tipo FPGAs, os quais correspondem a circuitos integrados constituídos por blocos lógicos alocados em forma de Matriz, que podem ser configuráveis e interconectáveis através do software especializado (MAXFIELD, 2009).

Um esquema da arquitetura de uma FPGA se pode ver na Figura 2.5. Cada bloco está constituído por flip-flop ou registros e são interconectados de acordo com o programa desenvolvido. Esta programação para as FPGA pode ser realizada através da Linguagem de Descrição de Hardware (VHDL).

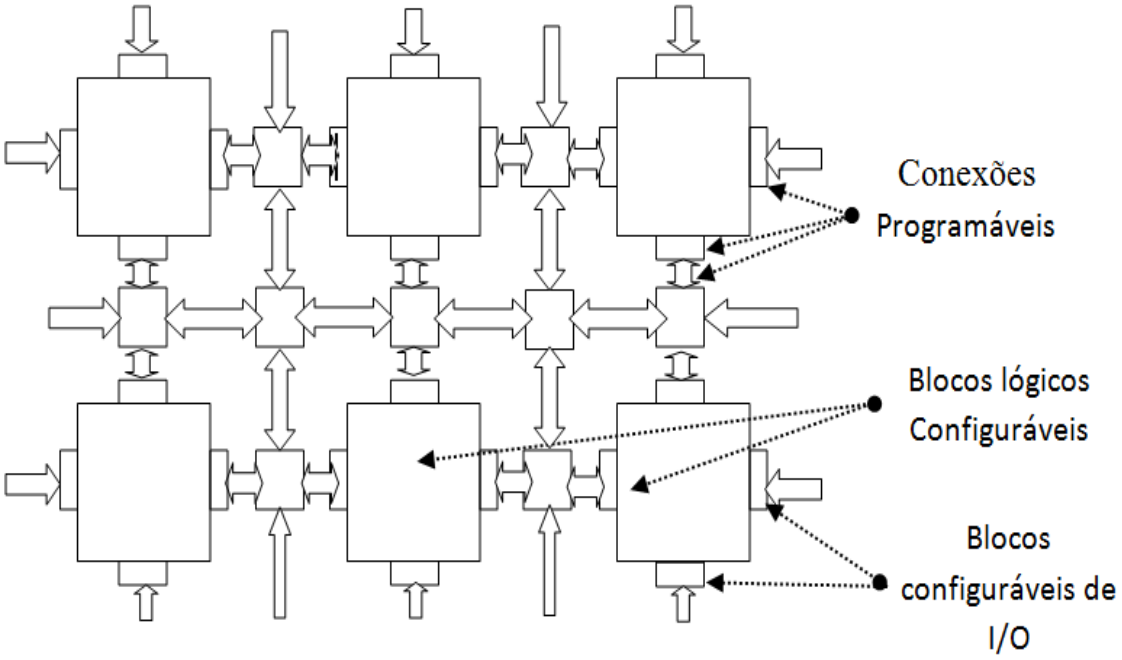

Figura 2.5 Arquitetura de uma FPGA.

VHDL corresponde ao standard IEEE Std 1076-1987, e tem sido revisado nos anos 1993, 2000 e 2002. Além disso, permitem a realização de modelagem e sínteses, mediante HDL, que corresponde a una descrição do programa através de um arquivo ASCII e que pode ser transportado e interpretado em outras ferramentas informáticas (GROUT, 2008).

São duas as empresas que lideram o mercado das FPGA, Xilinx e Altera, desenvolvendo produtos que são aplicados nas diferentes áreas da engenharia e entre eles estão os kits de desenvolvimento, os quais estão equipados de periféricos que facilitam a implementação de sistemas de aquisição e controle de maneira rápida e flexível.

Uma das placas de desenvolvimento de Altera é a Ciclone II, possui dispositivo de controle serial, Controlador USB-Blaster para programação, SRAM, SDRAM, memória flash, Clocks internos de 50 MHz, entre outras características (MUSTAFA *et al.*, 2009).

2.6 Considerações Finais

Neste capítulo foram apresentados conceitos básicos dos sistemas de manufatura industrial, realizando um percurso no tempo na configuração dos sistemas produtivos, destacando a importância da modelagem e projeto de sistemas de controle na evolução desse sistema. Além disso, pode-se concluir a importância de levar em conta as dinâmicas descritas em relação ao tempo e as dinâmicas regidas pelos eventos discretos ao momento de realizar a descrição de um processo.

Ao realizar a descrição do sistema através da modelagem, é importante a utilização de ferramentas que permitam emular em sistemas embarcados, esses modelos propostos, com o intuito de permitir testes rápidos e de menor custo no desenvolvimento de um produto.

É importante destacar o alto custo que leva o desenvolvimento de um produto e a necessidade de utilizar ferramentas que possibilitem uma maior competitividade no seu desenvolvimento.

No próximo capítulo vai ser apresentado as ferramentas para a modelagem dos sistemas que respondem a uma arquitetura hibrida dos processos automatizados.

Capítulo 3

3. Modelagem de Sistemas de Arquitetura Híbrida em Processos Automatizados

3.1 Introdução

Neste capítulo são descritos conceitos básicos utilizados no desenvolvimento deste trabalho. Inicialmente, se descreve o formalismo de modelagem DEVS, em sua representação atômica, realizando uma análise detalhada de sua arquitetura. Posteriormente, se apresentará o conceito do acoplamento dos modelos DEVS acoplados. Ao mesmo tempo, serão descritos conceitos teóricos utilizados no projeto de sistemas de controle baseado em eventos. Nesses conceitos está o QSS, que permite representar sistemas dinâmicos descritos em relação ao tempo, em um sistema equivalente discretizados de maneira assincrônica. A seguir, se descreve a representação de sistemas quantificados em QSS, através do formalismo DEVS. Mesmo assim se realiza uma descrição das ferramentas matemáticas para calcular o erro nos sistemas discretizados através de QSS. Na parte final deste capítulo é apresentada uma síntese de controladores baseado em eventos.

3.2 Modelagem dos Sistemas de Arquitetura Híbrida para automação

Os sistemas híbridos são sistemas dinâmicos que consistem em componentes com comportamento contínuo e de eventos discretos. A modelagem, análise e projeto desses tipos de sistemas requerem certas considerações metodológicas, pois para descrever o sistema eles exigem a combinação da descrição das variáveis contínuas, representadas através de Equações diferenciais e em diferenças, em união com modelos de eventos discretos que inicialmente foram representados por autômatos ou redes de Petri (LUNZE e LAMNABHI-LAGARRIGUE, 2009). Novos formalismos foram propostos com o intuito de representar o comportamento dos sistemas dirigidos pela ocorrência de eventos. Entre eles se destaca o formalismo DEVS descrito a seguir.

3.3 Formalismo DEVS

(NIKOLAIDOU, *et al.*, 2008) descreve o formalismo DEVS como base conceitual para especificação do modelo e sua simulação de sistemas através de eventos discretos. A modelagem pode ser feita através da decomposição do sistema em modelos menores que são denominados modelos atômicos. Através desses modelos atômicos, podem ser representados os diferentes componentes de um sistema e o acoplamento entre os mesmos representam o modelo geral do sistema. Na próxima seção deste trabalho é apresentada uma descrição do formalismo de modelagem DEVS em sua representação atômica.

3.3.1 Formalismo DEVS Atômico

O modelo atômico representa a unidade "molecular" de processamento e se constitui no elemento fundamental do modelo, e o segundo é o modelo acoplado que está constituído pela união dos modelos atômicos (ZEIGLER, e*t al.*, 1996). O modelo atômico pode ser definido como a sêxtupla da Equação (3.1).

$$\mathbf{H} = (\mathbf{X}, \mathbf{Y}, \mathbf{S}, \boldsymbol{\delta}_{\mathbf{int}}, \boldsymbol{\delta}_{\mathbf{ext}}, \boldsymbol{\lambda}, \mathbf{t}_{\mathbf{a}}) \tag{3.1}$$

Onde:

X : É o conjunto de valores de eventos de entrada.

Y : É o conjunto de valores de eventos de saída.

S : É o conjunto de valores de estado.

$\delta_{\mathrm{int}}, \mathrm{t_a}$: É a função de transição interna e a função do avanço do tempo, respectivamente.

δ_{ext}: Q x X $\rightarrow$ S, é uma função de transição externa, onde Qé o conjunto de estados de M.

$M = \{(s, e) | s \in S \mathrm{and} 0 \leq e \leq ta(s)$

$\delta_{\mathrm{int}}: \mathrm{S} \rightarrow \mathrm{S}$ É a função de transição interna

$\lambda : \mathrm{S} \rightarrow \mathrm{y}$ É a função de saída

$\mathrm{t_a} : \mathrm{S} \rightarrow \mathrm{R}^{+}_{0,\infty}$É uma função de avanço do tempo, que determina a máxima duração que o sistema pode permanecer no estado atual. Se transcorre um tempo *ta(s)* sem a ocorrência de eventos externos, o sistema realiza uma transição interna indo a um novo estado s_2, que é

calculado por $s_2 = \delta_{int}(s_2)$, isto ocorre no caso que t_a não seja definida como zero ou infinito.

Se t_a é definido como zero e o sistema permanecerá no estado atual um tempo zero antes de ir para o seguinte estado. Nesse caso o sistema atual é considerado um estado de passo transitório.

Se t_a é definido como infinito o sistema que permanece no estado atual até o acontecimento de um evento externo o que converte ao estado atual em um estado passivo.

A transição do estado s_1 a s_2, produz igualmente um evento de saída, definido por $y = \lambda(s_1)$. A equação de estado do modelo atômico está representada pela Equação (3.2).

$$q' = \delta_{int}(q) \oplus \delta_{ext}(q, X) \tag{3.2}$$

A qual significa que o novo estado do sistema depende da função de transição externa, correspondente a função de transição interna, mas não das duas simultaneamente.

No modelo DEVS se espelham duas variáveis para representar os intervalos de tempo, são elas (e,σ), onde a variável (e) armazena o tempo de avanço do tempo transcorrido desde a última transição do estado e a variável (σ) armazena o tempo que falta para ocorrer a seguinte transição interna.

A Figura 3.1 apresenta a semântica do modelo atômico DEVS (Kim, *et al.*, 1997). Quando os eventos externos de entrada chegam, o modelo responde de acordo a sua função de transição externa, se não existirem eventos em um determinado tempo definido pela função de avanço do tempo, o modelo modifica o estado de acordo com à função de transição interna, gerando um evento de saída para os outros modelos.

No fluxograma apresentado na Figura 3.2 se representa uma interpretação da execução do formalismo DEVS (URQUÍA, 2008). Se em um instante de tempo t_x o sistema está no estado S_x e ocorre um evento de entrada, se realizam as seguintes atividades:

- Desabilita a execução da transição interna.
- O sistema muda de estado de acordo com a execução da função de transição externa, então o seguinte estado estará dado por $s_{x+1}= \delta_{ext}$ (s_x, e, x_x), levando em conta os argumentos do estado atual do sistema (s_x), o tempo transcorrido a partir da ultima transição (e), pode ser interna ou externa e o valor do evento de entrada.
- O evento interno que está pronto para sua execução no seguinte tempo $\mathbf{t_a}$, é eliminado da lista de eventos.
- Realiza-se o planejamento do evento interno que ativara a função de transição interna em ausência de eventos de entrada em um lapso de tempo igual a t_a.

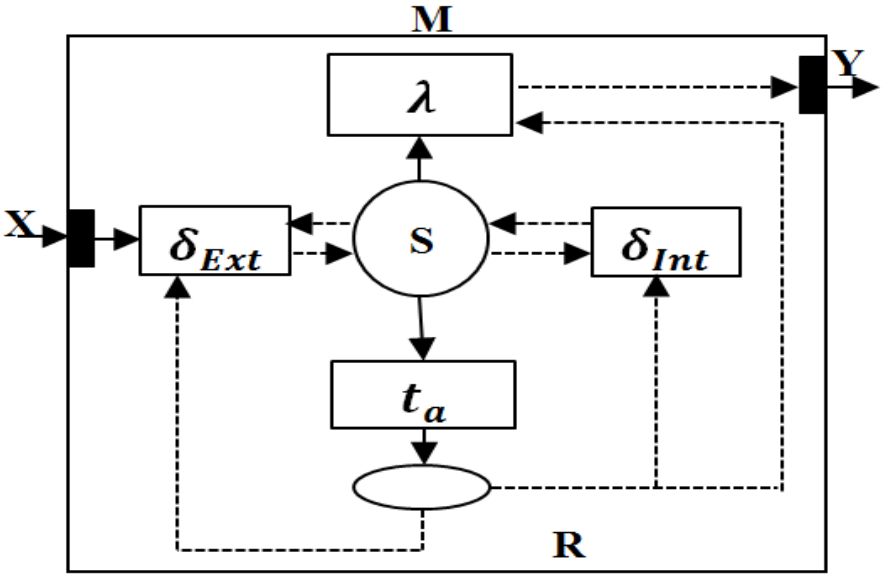

Figura 3.1 Especificação Atômica DEVS.

Se em um instante de tempo t_x o sistema está no estado s_x e não ocorre um evento de entrada, se realizam as seguintes atividades:

- É afetada à variável e o tempo que transcorreu a partir da última transição, ou seja, se faz e = $\mathbf{t_a}$(sx).

- Atualiza-se a saída do sistema de acordo com a função $\boldsymbol{y = \lambda(s_1)}$.
- É habilitada para seu posterior execução a seguinte transição interna

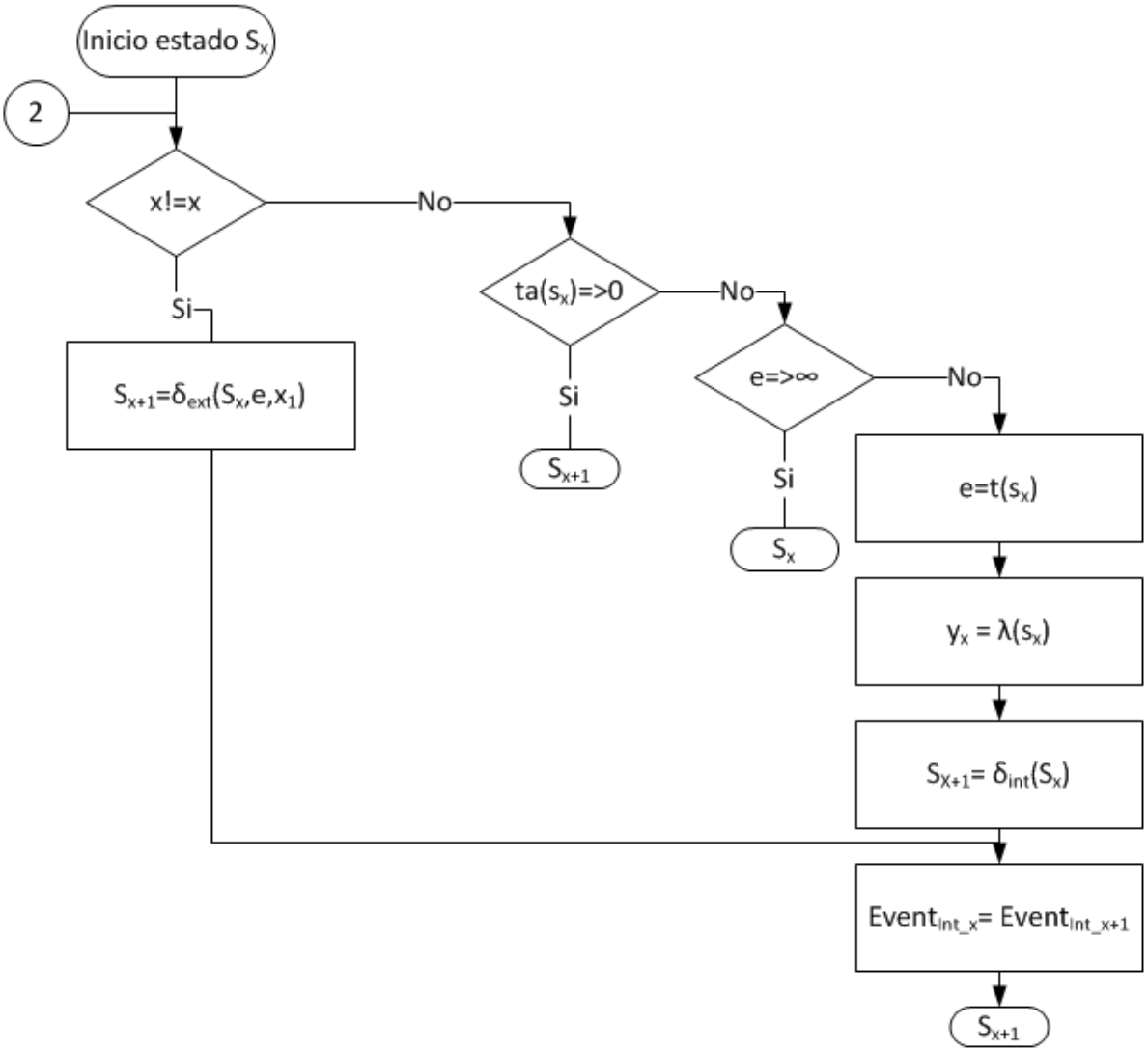

Figura 3.2 Fluxograma do Modelo Atômico DEVS.

A Figura 3.3 apresenta um exemplo da descrição do modelo DEVS num processo que possui três estados, e a maneira como os mesmos, evoluem no tempo. Inicialmente o sistema está no estado s0 e em t1 e t2 o sistema muda de estado pela ocorrência das transições internas, atualizando sua saída, no t3, com a ocorrência de um evento externo o sistema muda de estado de acordo à execução da transição externa.

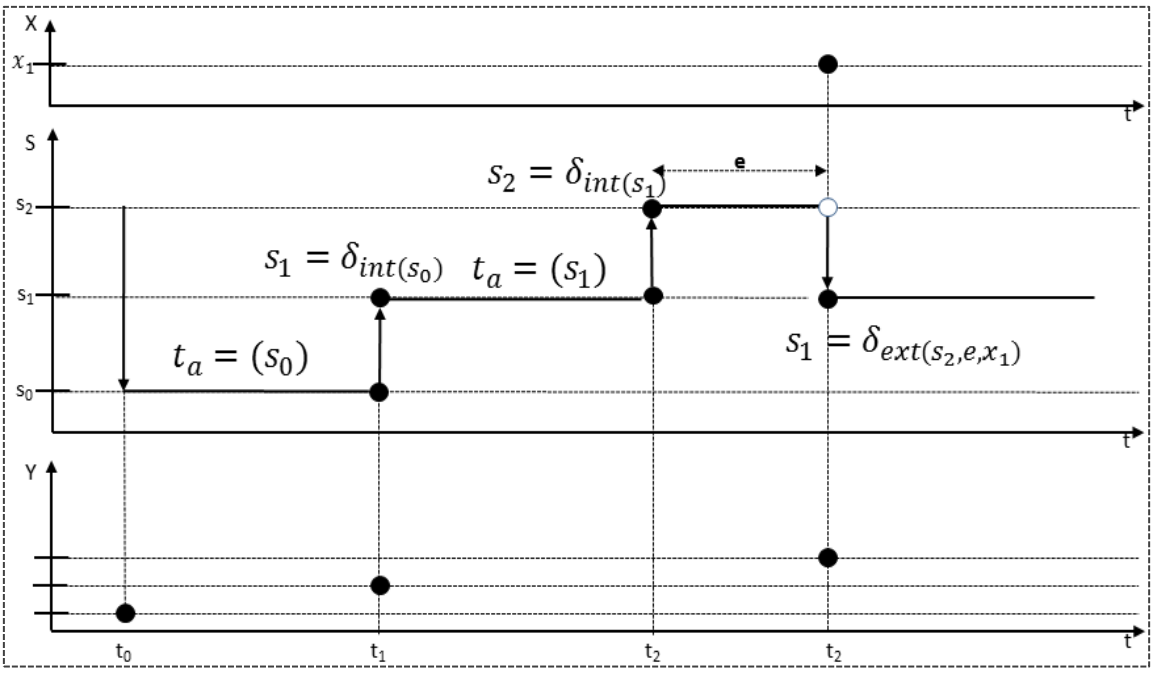

Figura 3.3 Execução do Modelo DEVS.

3.3.2 Formalismo DEVS para Sistemas Acoplados

A modelagem de um sistema pode ser realizada através do modelo atômico, muitas vezes resulta uma descrição complexa do sistema, consequentemente, com objetivo de simplificar o trabalho de modelagem de sistemas complexos, foi definido o modelo acoplado. Um esquema do modelo DEVS acoplado é apresentado na Figura 3.4 (ZEIGLER, 2003). A ideia básica é dividir através de modelos SED atômicos a representação de um sistema de dinâmica complexa, realizando a interligação e realimentação dos sinais que interagem no sistema. Assim, o modelo DEVS acoplado está definido pela tupla da Equação 3.3.

$$CM = \langle X, Y, D, \{M_d | \text{ d} \in D\}, EIC, EOC, IC, select \rangle \tag{3.3}$$

X : É o conjunto de portas de entradas para a recepção de eventos, onde:

$X = \{(p, v) | p \in Inport, v \in X_p$

Y : É o conjunto de portas de saídas para valores de eventos de saída, para a emissão de eventos externos.

$Y = \{(p, v) | p \in OutPorts, v \in Y_p$

D :É o conjunto de componentes (modelos básicos)

M_d: É o modelo DEVS para cada $d \in D$.

$M_d = \langle X_d, S, Y_d, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$, É um modelo DEVS clássico, que em geral tem várias portas de entrada e saída.

$X_d = \{(p, v) | p \in InPorts_d, v \in X_p$

$Y_d = \{(p, v) | p \in OutPorts_d, v \in Y_p$

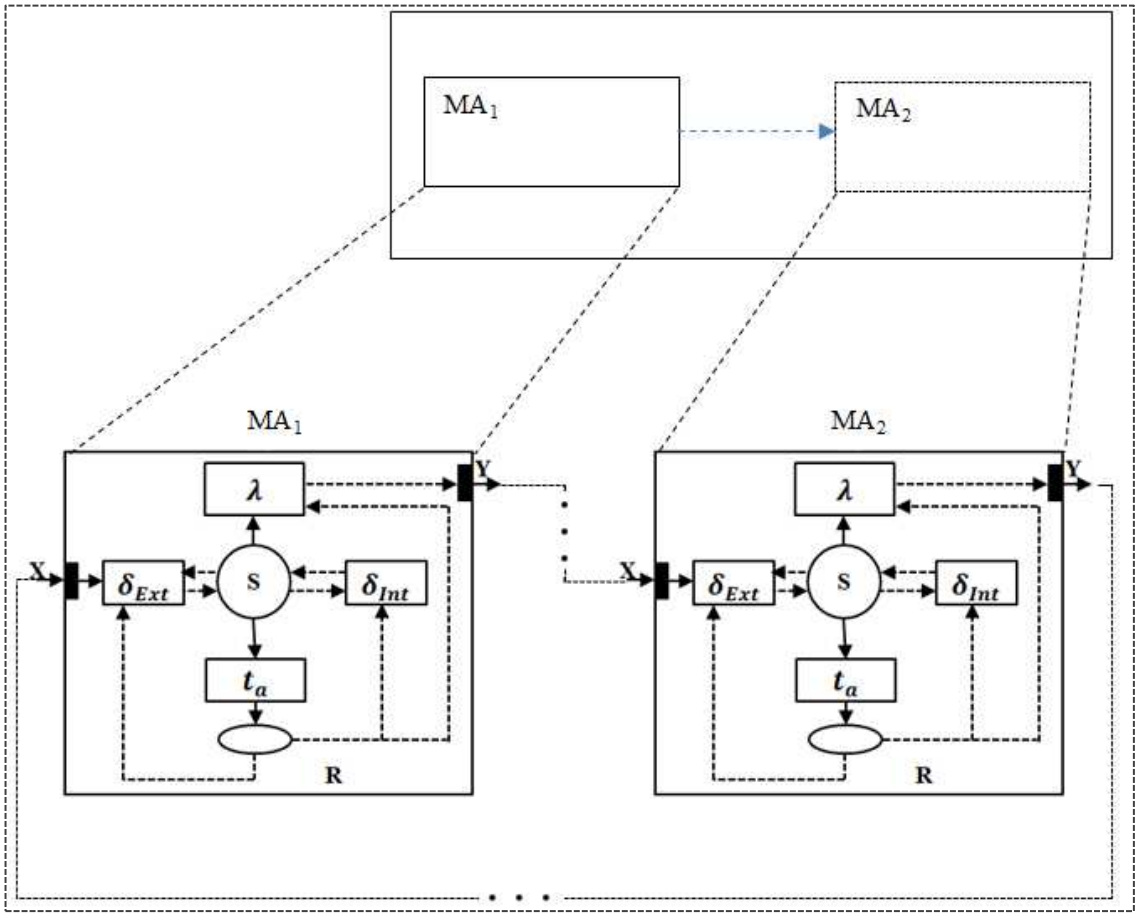

Figura 3.4 Modelo DEVS Acoplado

EIC: É o conjunto de malhas de entradas que se conectam as entradas do modelo acoplado para unir mais entradas de seus componentes, sendo constituído pela Equação 3.4.

$$\text{EIC} \subseteq \{((N, ip_N), (d, ip_d) \mid ip_N \epsilon\ InPorts, d \epsilon D, ip_d \epsilon\ InPorts_d\} \quad (3.4)$$

Onde:

(N,ip_N) Representa o porta de entrada $i\text{p}_\text{N}$do modelo acoplado.

(d,$i\text{p}_\text{d}$) Representa a porta de entrada $i\text{p}_\text{d}$ do componente d

((N,$i\text{p}_\text{N}$),(d,$i\text{p}_\text{d}$)) Representa a conexão entre os dois portas.

EOC : É o conjunto de malhas de saída que conectam as saídas de um ou mais dos componentes para saída do modelo acoplado, sendo representado através da Equação 3.5.

$$EOC \subseteq \{((d, op_d), (N, op_N)) \mid op_N \epsilon\ OutPorts, d \epsilon D, op_d \epsilon\ OutPorts_d\} \quad (3.5)$$

Onde:

(d, op_d)Representa o porta de saída op_d do componente d

(N, op_N) Representa o porta de saída op_N do modelo composto

$((d, op_d),\ (N, op_N))$ Representa a conexão entre ambos os portas.

IC : É o conjunto de malhas internas que conectam os portas de saída dos componentes com os portas de entrada dos modelos acoplados e está representada pela Equação 3.6.

$$\text{IC} \subseteq \left\{\left((\text{a}, \text{op}_\text{a}), (\text{b}, \text{ip}_\text{b})\right| \text{a}, \text{b} \in \text{D con a} \neq \text{b}, \text{op}_\text{a} \in \text{OutPorts}_\text{a}, \text{ip}_\text{b} \in \text{InPorts}_\text{b}\right\} \quad (3.6)$$

Onde:

$(\text{a}, \text{op}_\text{a})$: Representa o porta de entrada ip_Ndo modelo acoplado.

$(\text{b}, \text{ip}_\text{b})$: Representa o porta de entrada ip_d do componente d.

$((\text{a}, \text{op}_\text{a}),\ (\text{b}, \text{ip}_\text{b}))$: Representa a conexão entre os dois portas.

Select: Permite estabelecer a prioridade, se existe um evento interno planejado no mesmo instante.

3.4 Sistemas de Estados Quantificados

Nos sistemas descritos através do tempo, sua dinâmica geralmente é expressa através de ODE. Quando essa dinâmica é simulada através de sistemas computacionais ou exige a realização de ações de controle utilizando sistemas digitais, torna-se necessário a solução das Equações diferenciais que descrevem esse sistema, utilizando tradicionalmente métodos de integração numérica, convertendo as ODE em Equações diferenciais. Isto permite a representação de sistemas dinâmicos contínuos no tempo em sistemas descritos em relação ao tempo discreto.

Uma alternativa de solução das ODE, são os sistemas quantificados QSS (Quantized State Systems). Este é um método de integração numérica de eventos discretos, aplicado a ODEs. Além do método de quantização proposto por (Zeigler, Praehofer *et al.*, 2000), uma modificação foi proposta por (Kofman, 2003), que agregou uma função de histerese, evitando assim, infinitas transições em um período fixo de tempo que ocorre quando a variável de estado está próxima ao seu ponto de transição.

Assim, os métodos de quantificação QSS são baseados na quantificação das variáveis de estado da Equação diferencial ordinária (ODE). O método de quantização propõe para um sistema representado pela Equação 3.7(3.7).

$$\dot{x}(t) = f\big(x(t), u(t)\big) \tag{3.7}$$

Onde $\in \mathbb{R}^{\boldsymbol{n}}$, $\boldsymbol{f}: \mathbb{R}^{\boldsymbol{n}} \rightarrow \mathbb{R}^{\boldsymbol{n}}$ pode se aproximar através do sistema descrito pela Equação 3.8(3.8), que representa um Sistema de Estados Quantificados (QSS) (KOFMAN, 2003).

$$\dot{x}(t) = f\big(q(t), u(t)\big). \tag{3.8}$$

Onde $q(t) ex\ (t)$, tem uma relação componente a componente, inicialmente através de funções de quantificação com histereses em QSS.

3.4.1 Discretização Baseada na Quantização do Estado

Quando se realiza a discretização de um sistema através de métodos de integração numérica, se requer de uma amostragem assincrônica, pois para certo tipo de sistemas é difícil selecionar uma taxa de amostragem que seja o suficientemente rápida para um caso crítico do sistema. Além disso, é necessário garantir a eficiência no sistema computacional, para evitar escravizar o sistema somente na tarefa de amostragem.

QSS propõe um método alternativo de discretização, que consiste em pegar as amostras somente quando a variável mude em certo valor, que está definida pelo Quantum. Um exemplo da diferença do conceito de amostragem utilizando discretização através da amostragem sincrônica e através de quantização é apresentado na Figura 3.5.

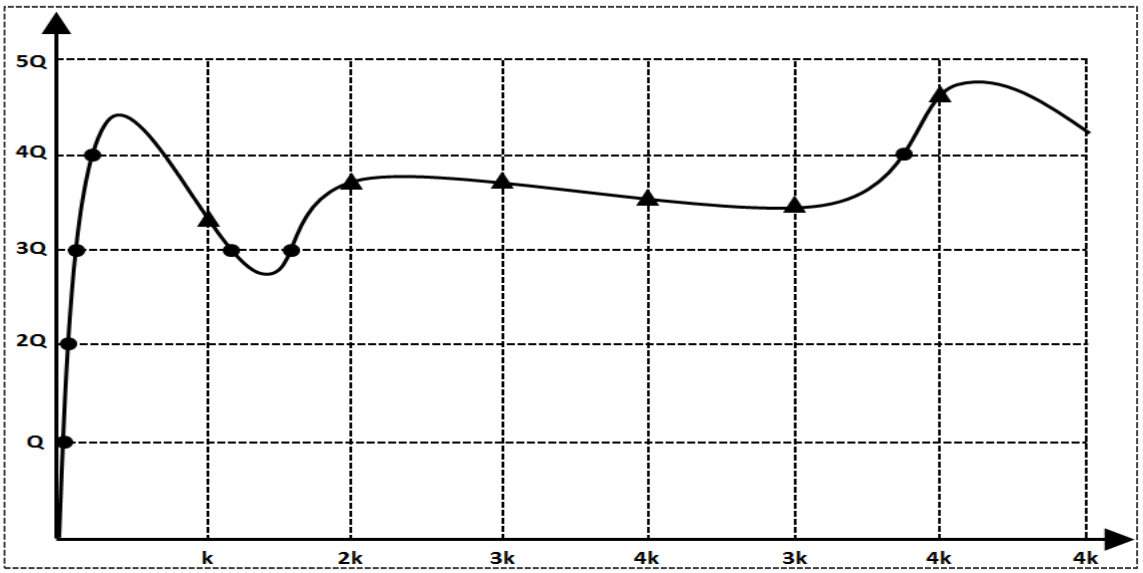

Figura 3.5 Amostragem Sincrônica e por Quantificação

As amostras representadas pelos triângulos são as realizadas de maneira sincrônica, quando se realiza uma discretização baseada no tempo. As representadas pelas bolinhas são adquiridas de maneira assincrônica, quando é realizada uma discretização baseada na mudança da variável de estado do sistema, o que permite garantir que o gasto computacional se ajuste às mudanças do processo.

O método QSS requer a escolha do intervalo em que será realizada a próxima quantização do sistema. O cálculo da mesma é baseado na mudança da distância de quantização ΔQ em relação com a atual derivada $\dot{x}$, de acordo com a Equação 3.9. Uma representação gráfica do cálculo da distância de quantização denominada quantum, é apresentada na Figura 3.6 (ALMINDE, 2009).

$$\Delta t = \frac{\Delta Q}{|\dot{x}(t_k)|} \tag{3.9}$$

Depois de transcorrer o intervalo de tempo, o estado se atualiza de acordo com a Equação 3.10.

$$x(t_k + \Delta t) = x(t_k) + sgn(\dot{x}(t_k))\, \Delta Q \tag{3.10}$$

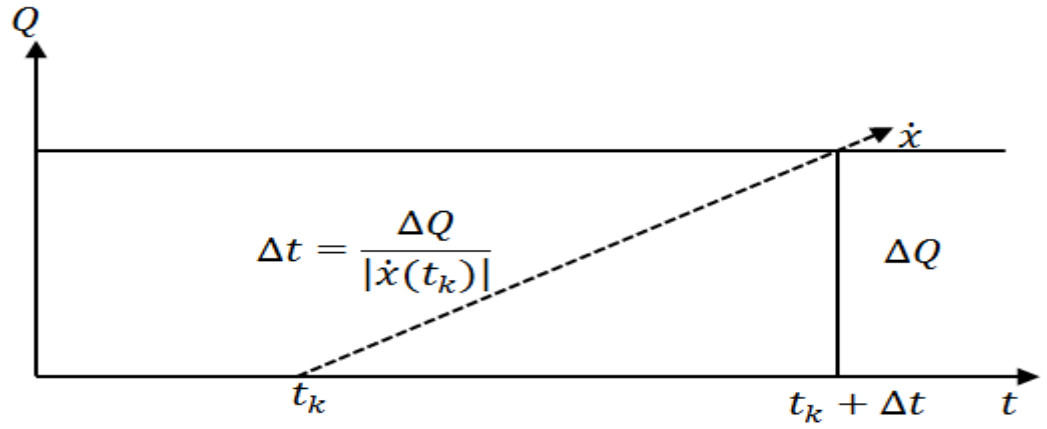

Figura 3.6. Sistema de Quantificação

3.4.2 Representação em DEVS do Integrador de Estado

O método de amostragem apresentado na seção anterior permite realizar a modelagem de sistemas contínuos baseados em eventos através de DEVS, mas existe uma dificuldade na descrição de variáveis definidas em função do tempo através deste formalismo, quando vai ser implementado. Esta dificuldade tem a ver com a não

possibilidade de DEVS para definir a dinâmica de um sistema, quando em um tempo finito ocorre um número infinito de transições discretas, pois estas transições produzem a sua vez oscilações infinitamente rápidas. Isto devido que as trajetórias de saída deixaram de ser segmentadas (ZEIGLER *et al.*, 2000). Esta dificuldade foi solucionada por Kofman (2003), quem adicionou histereses na quantização das variáveis de estado que descreve a dinâmica um sistema. Com esta adição de histerese vão ser requeridas oscilações muito grandes na variável de estado, para que as mesmas sejam refletidas na variável quantificada, e devido à continuidade na trajetória de estado, se requerer de um mínimo espaço de tempo para acontecer, o qual constitui uma condição suficiente para garantir a legitimidade do modelo.

De acordo a (KOFMAN, 2003): Seja $Q = \{Q_0, Q_1, \dots, Q_r\}$ um conjunto de números reais, onde $Q_{k-1} < Q_K$, com $1 \leq K \leq r$. Seja Ω o conjunto das trajetórias reais seccionalmente contínuas E seja $x \in \Omega$ uma trajetória continua, seja b um mapeamento de: $b: \Omega \rightarrow \Omega$, seja $q = b(x)$, onde a trajetória q satisfaça as Equações 3.11 e 3.12. Então se pode dizer que o mapeamento de b(x) é uma função de quantificação com histereses.

$$q(t) \begin{cases} Q_m & f\, t = t_0 \\ Q_{K+1} \text{ if } x(t) = Q_{k+1}\, Tq(t^-) = Q_k\, T \;\; k < r \\ Q_{K-1} \text{ if } x(t) = Q_{k+1}\, Tq(t^-) = Q_k\, T \;\; k < r \\ q_{(t^-)} \quad \text{otherwise} \end{cases} \tag{3.11}$$

$$m = \begin{cases} 0 & \text{if } x(t_0) < Q_0 \\ r & \text{if } x(t_0) \geq Q_r \\ j & \text{if } Q_J \leq x(t_0) < Q_{J+1} \end{cases} \tag{3.12}$$

O mapeamento b é a função de quantização com histereses, os valores discretos Q_i são os níveis de quantização e a distancia $Q_{k+1} - Q_k$ é denominado o quantum, como se apresenta na Figura 3.7. O parâmetro ε corresponde ao intervalo da janela de histerese.

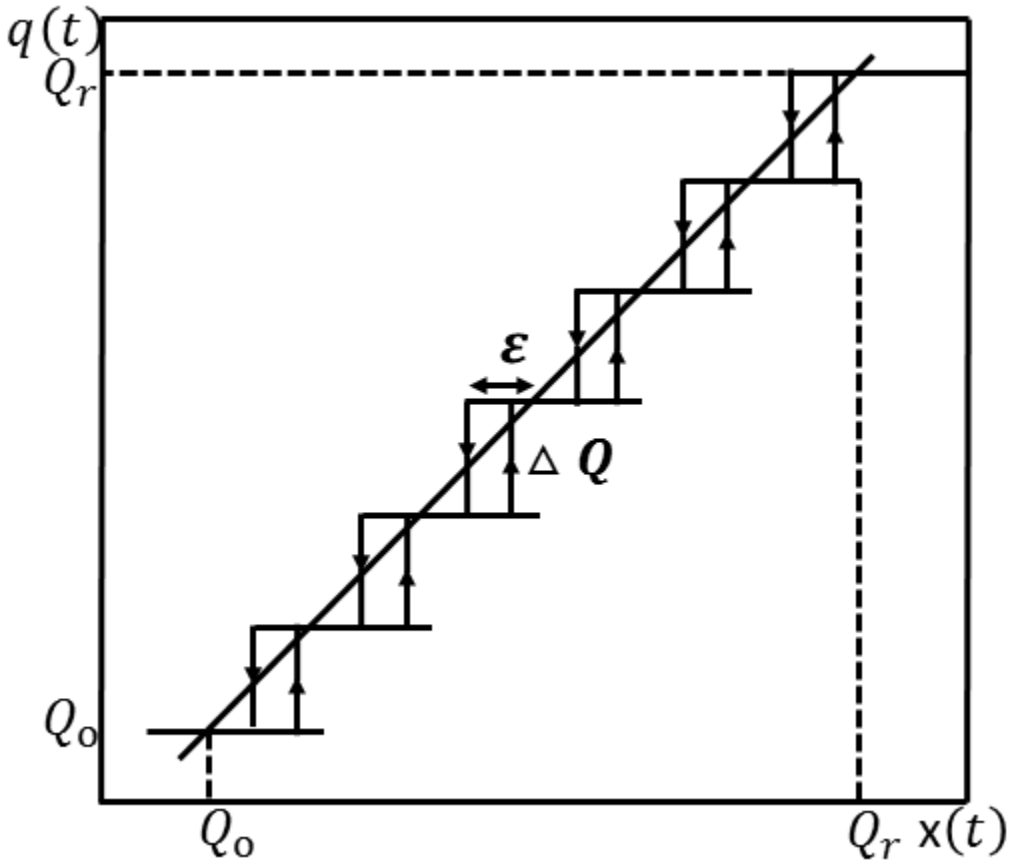

Figura 3.7. Histereses de Quantificação

Consequentemente, o método dos sistemas quantificados QSS é definido pela quantização através de histereses. Um esquema do conceito QSS é apresentado na Figura 3.8.

O bloco integrador é aplicado para cada variável de estado do sistema descrito pela Equação 3.7, mantendo o modelo representado pela Equação 3.13 (ALMINDE, 2009).

$$\bar{x}_i(t) = \bar{x}_i(t) + \bar{\dot{x}}_i(t - t_i) \tag{3.13}$$

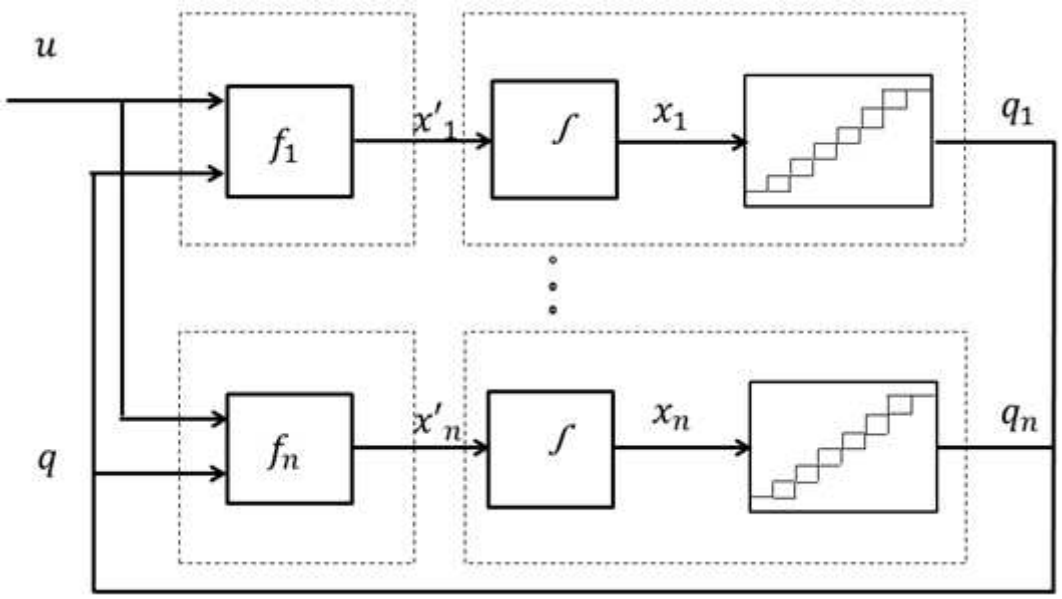

Figura 3.8. Estrutura do Sistema de Estados Quantificados

Onde t_i corresponde ao tempo do último evento interno DEVS do bloco integrador, que corresponde a sua vez à última saída produzida pelo bloco. A estrutura em DEVS do integrador quantificado e dada pela Equação 3.14 (ZEIGLER, *et al.*, 2000).

$$\mathrm{H}_2 = (\mathrm{X}, \mathrm{Y}, \mathrm{S}, \delta_{\mathrm{int}}, \delta_{\mathrm{ext}}, \lambda, \mathrm{t_a}) \tag{3.14}$$

Onde:

$X = Y = \mathbb{R} \times \mathbb{N}$

$S = \mathbb{R}^2 \times \mathbb{Z} \times \mathbb{R}_0^+$

$\delta_{int}(\mathrm{s}) = \delta_{int}(x, d_x, k, \sigma) = (x + \sigma \cdot d_x, d_x, k + sign(d_x), \sigma_1)$

$\delta_{ext}(s, e, x_u) = \delta_{ext}(x, d_x, k, \sigma, e, x_v, p) = (x + e \cdot d_x, x_v, k, \sigma_2)$

$\lambda(s) = \lambda(x, d_x, k, \sigma) = Q_{k+sign(d_x)}, 0)$

$ta(s) = ta(x, d_x, k, \sigma) = \sigma$

Onde o cálculo das variáveis σ_1 e σ_2 pode ser realizado levando em conta as Equações 3.15 e 3.16.

$$\sigma_1 = \begin{cases} \dfrac{Q_{k+2} - (x + \sigma \cdot d_x)}{d_x} & if\ d_x > 0 \\ \dfrac{(x + \sigma \cdot d_x - (Q_{k-1} - \varepsilon)}{|d_x|} & if\ d_x < 0 \\ \infty & if\ d_x = 0 \end{cases} \tag{3.15}$$

$$\sigma_2 = \begin{cases} \dfrac{Q_{k+1} - (x + e \cdot d_x)}{x_v} & if\ x_v > 0 \\ \dfrac{(x + e \cdot d_x - (Q_k - \varepsilon)}{|x_v|} & if\ x_v < 0 \\ \infty & if\ x_v = 0 \end{cases} \tag{3.16}$$

3.4.3 Projeto de Sistema de Controle Baseado em Eventos

A implementação digital de controladores contínuos é basicamente um problema de discretização de uma Equação diferencial, geralmente realizado utilizando o método de Euler ou alguma outra regra de aproximação em tempo discreto, razão pela qual a conversão A/D se realiza utilizando um tempo de amostragem fixa, o que pode acarretar como consequência que o sistema fique sem controle entre cada uma das amostras.

O Controle de Estados Quantificados (QSC) é um método de controle baseado em eventos discretos, que realiza a aproximação de um controlador em tempo contínuo com o método QSS, para isto utiliza uma amostragem de tipo assíncrono. Ou seja, quantifica as variáveis de estado do controlador e o modelo do processo, conectado esses através de conversores A/D e D/A assíncronos, que realizam as operações de conversão quando a diferença entre a entrada e a saída supera um valor definido, conhecido como intervalo de quantificação. A diferença do sistema de discretização baseado no tempo o qual realiza a amostragem de maneira síncrona em intervalos de tempo fixo.

Um controlador QSC pode-se representar de acordo com a Equação (3.17) (KOFMAN, 2003).

$$
\begin{cases}
\begin{cases} \dot{x}_p(t) = f_p(x_p(t), u_p(t)) \\ y_p(t) = g_p(x_p(t)) \end{cases} \\
\begin{cases} \dot{x}_c(t) = f_p(q_c(t), u_c(t), u_r(t)) \\ y_c(t) = g_c(q_c(t), u_c(t), u_r(t)) \end{cases} \\
\{u_p(t) = y_{cq}(t) \\
\{u_c(t) = y_{pq}(t)
\end{cases}
\tag{3.17}
$$

Essas Equações correspondem respectivamente à planta continua, ao controlador e à interconexão ideal entre os dois componentes do sistema QSC. No sistema de Equações do controlador, pode ser observado que as variáveis de estado que descrevem o comportamento do sistema se encontram quantificadas, além disso, a entrada se compõe de um valor de referência ur(t) e da realimentação da malha de controle uc(t), que corresponde à saída da planta. Na terceira Equação a entrada da planta up(t) é conectada com a saída do controlador quantificada pelo conversor D/A (ycq(t)), enquanto que a entrada de controle é conectada com a saída da planta, que se quantifica através do conversor A/D (ypq(t)).

3.4.4 Erro do Controle QSC para Sistemas LTI

As Equações (3.17) podem ser escritas para um sistema Linear e Invariante no Tempo (LTI) como mostra a Equação (3.18) de acordo com Kofman, (2003).

$$
\begin{cases}
\begin{cases} \dot{x_p}(t) = A_p \cdot x_p(t) + B_p \cdot u_p(t) \\ y_p(t) = C_p \cdot x_p(t) \end{cases} \\
\begin{cases} \dot{x_c} = A_c \cdot (x_c + \Delta x_c) + B_c \cdot u_c + B_r \cdot u_r \\ y_c(t) = C_c \cdot (x_c + \Delta x_c) + D_c \cdot u_c + D_r \cdot u_r \end{cases} \\
\{u_c = y_p + \Delta y_p \\
\{u_p = y_c + \Delta y_c
\end{cases}
\tag{3.18}
$$

Δxc representa a diferença entre a variável de estado e sua versão quantificada, Δyp corresponde ao efeito da quantificação da saída da planta através do conversor A/D e Δyc corresponde ao efeito da quantificação da saída do controlador pelo conversor D/A.

A Equação 3.19 representa o sistema em malha fechada.

$$\dot{x} = A(x + \Delta x) + F\Delta y + B(u_r)$$
$$y_p(t) = C_p \cdot x_p(t) \tag{3.19}$$

Onde,

$$x = \begin{bmatrix} x_p \\ x_c \end{bmatrix}, A = \begin{bmatrix} A_p + B_p D_c C_p & B_p C_c \\ B_c C_p & A_c \end{bmatrix}, B = \begin{bmatrix} B_p D_r \\ B_r \end{bmatrix}$$

$$\Delta x = \begin{bmatrix} \Delta x_p \\ \Delta x_c \end{bmatrix}, \Delta y = \begin{bmatrix} \Delta y_p \\ \Delta y_c \end{bmatrix}, F = \begin{bmatrix} B_p D_c & B_p \\ B_c & 0 \end{bmatrix}$$

$$u_c = y_p + \Delta y_p \ ; \ u_p = y_c + \Delta y_c$$

A é a matriz de evolução do sistema em malha fechada e o vetor Δx contém os valores de quantificação das variáveis de estado da planta e do controlador, por isso o primeiro valor é zero, já que a planta não se quantifica.

Da matriz A se obtém os valores e vetores próprios do sistema em malha fechada, que junto com as outras matrizes permitem conhecer o erro máximo que se apresentará entre as variáveis de um Sistema de Controle Contínuo (CCS) e um sistema QSC, descrito através da Equação 3.20 (KOFMAN, 2003).

$$| \tilde{x}(t) - x(t) | \leq | V | (| \mathbb{R}e(\Lambda)^{-1}\Lambda || V^{-1} | \Delta q_x + | \mathbb{R}e(\Lambda)^{-1}V^{-1}F | \Delta q_y) \tag{3.20}$$

Onde o operador | | significa a magnitude da componente da matriz que se aplica, Λ é a matriz diagonal de valores próprios, V é a matriz de vetores próprios, Δqx é o vetor Δx apresentado anteriormente e que contém os quantum da planta e do controlador, e Δqy é o vetor dos valores de quantificação da saída da planta e do controlador (é o mesmo vetor Δy mostrado anteriormente).

Consequentemente, ao serem fornecidos valores de quantificação para as variáveis de estado do controlador e para os conversores A/D e D/A, a diferença entre as variáveis do sistema CCS e o sistema QSC, estará sempre limitada, além disso, quanto menores sejam os intervalos de quantificação menor será a cota de erro.

3.5 Síntese de Controladores para Sistemas de Arquitetura Híbrida

Considerando-se os formalismos DEVS atômico descrito na Equação 3.1, DEVS acoplado na Equação 3.3 apresentado por (ZEIGLER, *et al.*, 2000) e QSS como método de integração, que permite representar sistemas de dinâmica regida em relação ao tempo através do formalismo DEVS, e que é descrito pela Equação 3.14. Além disso, levando em consideração a proposta de (KOFMAN, 2004) na descrição de sistemas de arquitetura hibrida. Realiza-se uma proposta de síntese de sistemas de controle aplicáveis em processos de arquitetura hibrida, em função da implementação em sistemas embarcados sob o conceito da prototipagem rápida. Neste trabalho de pesquisa tem-se como objetivo a modelagem, controle, simulação e síntese utilizando sistemas embarcados descritos através de VHDL.

A proposta inclui sete etapas com diferentes fases de desenvolvimento e implementação, descrevendo as respectivas tarefas realizadas para a modelagem da dinâmica do processo considerando sua estrutura híbrida, a proposta de controle, uma primeira validação através da simulação e a validação através da síntese de sistemas embarcados. Uma descrição geral das diferentes etapas propostas neste trabalho de pesquisa é descrito através do algoritmo apresentado na Figura 3.9.

3.5.1 Descrição das Etapas Propostas

A descrição de cada uma das etapas e as diferentes fases necessárias para o desenvolvimento das mesmas é apresentada na Tabela 3.1.

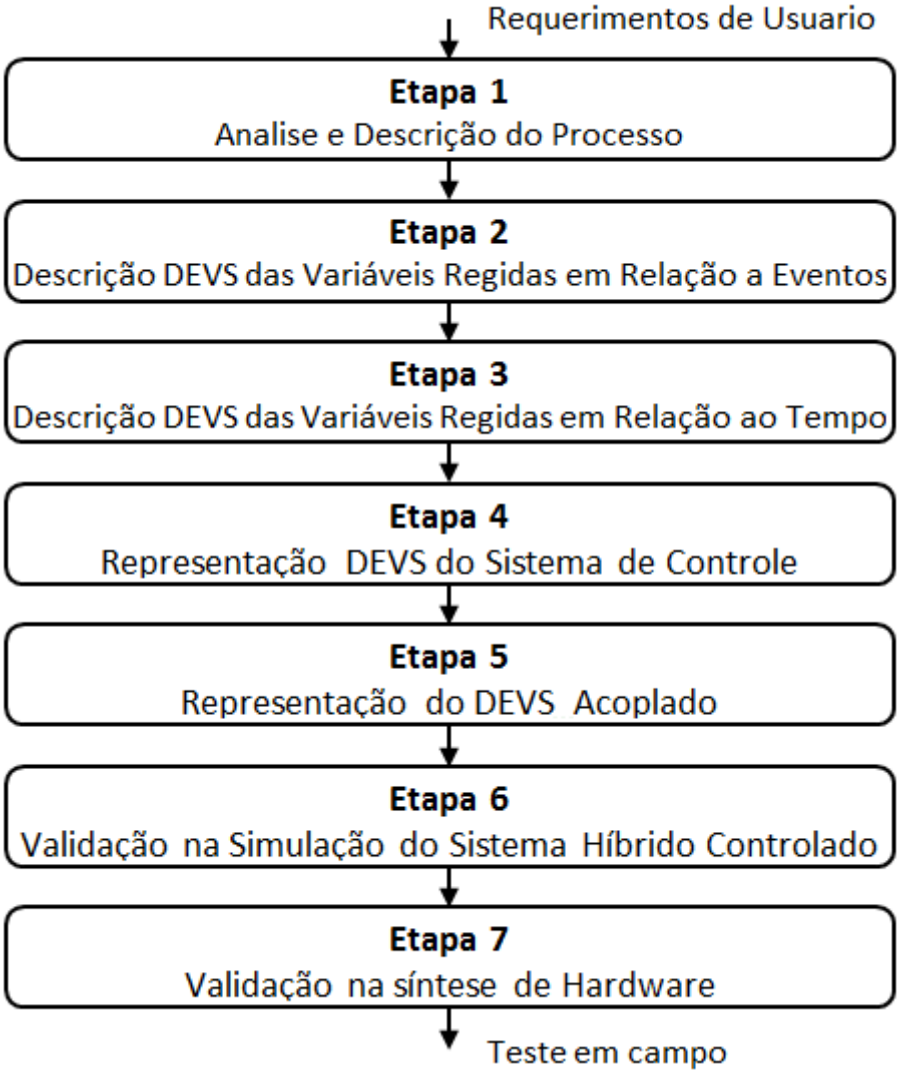

Figura 3.9 Proposta de Síntese de Sistemas de Arquitetura Hibrida.

Tabela 3.1 Etapas da proposta de síntese de sistemas de controle

Etapa 1	Nesta etapa se realiza uma descrição detalhada do processo a controlar, fazendo-se uma seleção das variáveis que compõem o sistema, definindo as variáveis que são regidas pela ocorrência de eventos e as variáveis descritas em relação ao tempo, mesmo assim é definida a interação existente entre elas.
Etapa 2	Nesta etapa descreve-se em função do formalismo DEVS atômico híbrido, as variáveis regidas em relação a eventos, com a interação das variáveis

	quantificadas, para o qual se realiza a definição do range de entrada e saída do modelo e as funções de transição interna, externa, saída e de avanço do tempo Equação 3.21.
Etapa 3	Nesta etapa se realiza a descrição das variáveis regidas em relação ao tempo no formalismo DEVS híbrido acoplado Equação 3.22. Define-se o range de entrada e saída, funções de transição interna, externa, saída e de avanço do tempo.
Etapa 4	Nesta etapa realiza-se o projeto do sistema de controle das variáveis descritas em relação ao tempo em variáveis de estado, para ser representado em DEVS, depois de ser quantificada (Equação 3.14), definido o range de entrada e saída, funções de transição interna, externa, saída e de avanço do tempo. Além da definição do tempo da seguinte transição ($\sigma 1$,$\sigma 2$) (Equações 3.15 e 3.16). Define-se também o sistema de controle das variáveis descritas através de eventos para serem representadas através de modelos DEVS (Equação 3.1).
Etapa 5	Nesta etapa se realiza a representação do modelo DEVS acoplado com a interação das variáveis descritas em relação ao tempo e a eventos discretos, realizando uma representação gráfica da interação das variáveis que compõem o sistema híbrido.
Etapa 6	Nesta etapa se realiza a validação do sistema de controle através da simulação utilizando para este caso as ferramentas Simulink® e Stateflow® de Matlab®.
Etapa 7	Neta etapa se realiza a síntese do sistema híbrido para o sistema embarcado utilizando o conceito da prototipagem rápida através das ferramentas de geração de código VHDL com o intuito de realizar uma validação do sistema de controle através de sua implementação.

Para descrição do sistema híbrido, se apresentam duas tuplas que descrevem à dinâmica das variáveis descritas em relação ao tempo e a dinâmica de eventos discretos, correspondendo a interação existente num processo industrial. Um esquema representativo do modelo proposto é apresentado na Figura 3.10.

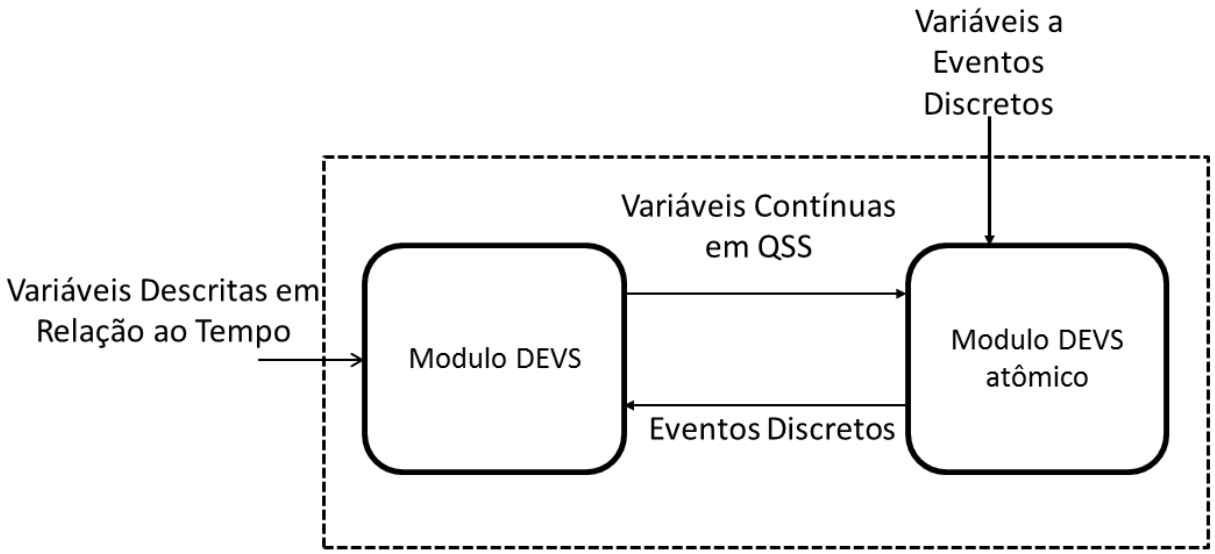

Figura 3.10. Componentes do modelo híbrido

3.5.2 Modelo Híbrido Acoplado

A parte continua do modelo pode ser representado como uma estrutura DEVS acoplada contendo um modelo contínuo quantificado por QSS. A descrição deste modelo se apresenta através da tupla representada na Equação 3.21, que é baseada no modelo DEVS acoplado.

$$MCD = (X_c, Y_c, X_d, Q_S, E_f, M_q, M_e, EIC_h, EOC_h, IC_h, select) \tag{3.21}$$

T: Constitui a base do tempo, não está explicita no modelo.

X_c: Conjunto de entrada das variáveis de estado descritas em relação ao tempo.

Y_c: Conjunto de variáveis de saída quantificadas.

X_d: Conjunto de eventos discretos de entrada.

Q_S: Conjunto de nome dos integradores QSS.

E_f: Conjunto de nome das funções estáticas.

M_q: Conjunto de modelos DEVS dos integradores QSS.

M_e: Conjunto de modelo DEVS das funções estáticas.

EIC_h: Acoplamento externo de entradas.

EOC_h: Acoplamento Externo de saídas.

IC_h: Acoplamento interno.

Cada uma das funções estáticas e cada integrador correspondem a blocos atômicos DEVS. O conjunto de entrada e saída podem conter vários portas, podendo representar tanto modelos DEVS SISO como modelos DEVS MIMO.

3.5.3 Modelo Atômico Híbrido

O modelo atômico híbrido considera as variáveis regidas em função do tempo e os eventos das variáveis quantificadas por QSS é representado na Equação (3. **22**3.22.

$$MD = (X_d, Y_d, X_c, S, \delta_{int}, \delta_{ext_{Ev}}, \delta_{ext_Q}, t_a) \quad (3.\ 22)$$

A diferença principal deste modelo em relação ao modelo DEVS atômico clássico, está no conjunto de variáveis de entrada quantificadas e a função de transição externa de variáveis quantificadas.

T $= \mathbb{R}$ Constitui a base do tempo, não está explicita no modelo.
t_a = Função de avanço do tempo.
X_d= Conjunto de variáveis de eventos discretos de entrada seccionalmente constante.
Y_d = Conjunto de variáveis de eventos discretos de saída.
X_c = Conjunto de variáveis de entrada quantificadas.
S = Conjunto de estados sequenciais.
δ_{int} = Função de transição interna, ativada pela terminação de t_a.
$\delta_{ext_{Ev}}$ = Função de transição externa, ativada por uma mudança em uma das variáveis de entrada de eventos discretos.
δ_{ext_Q} = Função de transição externa, ativada por uma mudança em uma das variáveis de entrada quantificadas.
$Select$ = determina a prioridade da transição interna quando dois componentes tenham planejado uma transição interna simultaneamente.

3.6 Considerações Finais

A partir da proposta orientada à implementação de sistemas de controle aplicáveis em processos de dinâmica hibrida, neste trabalho pretende-se implementar uma ferramenta metodológica de desenvolvimento de aplicação na automação de processos industriais e especificamente em empresas de pequeno e médio porte, permitindo realizar propostas neste campo ajustadas com a medida das necessidades e capacidades deste setor cada vez mais presente e exigido em aplicações industriais.

Uns dos aspectos relevantes do controle híbrido são, os poucos trabalhos de aplicação industrial desenvolvidos na atualidade, têm-se muitos trabalhos desenvolvidos na teoria e simulação, mas poucas propostas ainda na aplicação foram encontradas.

No próximo capítulo desta tese será apresentado o sistema de produção industrial que vai ser modelado e controlado desde a perspectiva hibrida.

Capítulo 4

4. Descrição da Plataforma Industrial para Pesquisa, Ensino e Formação em Automação.

4.1 Introdução

Neste capítulo se apresenta uma descrição de um sistema de produção industrial denominado de Plataforma Industrial para Pesquisa, Ensino e Formação em Automação (PIPEFA) implementada no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP, que representa um chão de fabrica industrial retratando aspectos reais de produção. Além de realizar uma descrição da estrutura física, vai ser apresentado a maneira como está controlado atualmente.

4.2 Descrição da Plataforma PIPEFA

A Plataforma PIPEFA representa um chão de fabrica refletindo as principais características de um sistema de produção industrial. A parte operacional é constituída de unidades de trabalho que funcionam de maneira independente, coordenadas por um sistema de gestão de produção corporativo. O propósito deste sistema de produção é realizar a confecção de um produto genérico, baseado em operações de carregamento/descarregamento, montagem/desmontagem e inspeção de produtos montados em uma placa de base na qual se inserem peças em diferentes posições e até dois níveis de montagem (AIHARA, 2005). As peças utilizadas são cubos e placas de base da LEGO^{TM}, que apresentam boa precisão mecânica e custo relativamente baixo. Este sistema é constituído de cinco módulos descritos a seguir.

O primeiro é o módulo de carregamento, no qual se inicia a confecção do produto através da inserção de uma placa base no sistema de transferência, que transporta até as duas estações encarregadas de montar ou desmontar as peças nas posições laterais e centrais da placa.

O processo continua na estação de inspeção, onde é verificado o produto como bom, rejeitado ou em processo, para finalmente ser levado até a estação de armazenamento. A representação deste sistema de produção é mostrada na Figura 4.1 onde pode ser observar a distribuição física de cada uma das estações dispostas num sistema de transferência.

O sistema de transferência é constituído de quatro esteiras independentes que correspondem a cada uma das estações de trabalho, e permite levar os produtos desde a estação de carregamento até as estações de montagem e desmontagem lateral, e dessas para a estação de inspeção, terminando na estação de armazenamento com os produtos terminados.

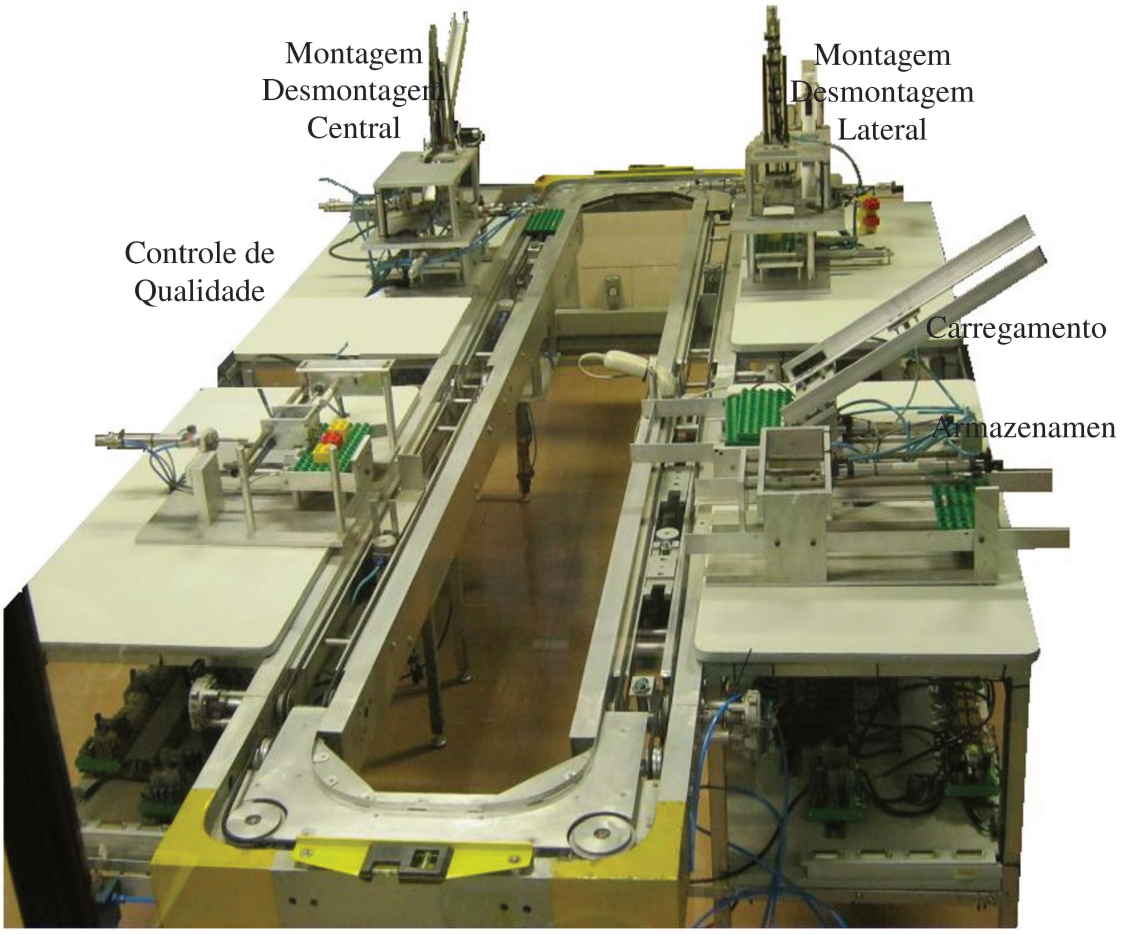

Figura 4.1 Plataforma PIPEFA.

4.2.1 Descrição do Posto de Carregamento

Este é a primeira unidade da linha de montagem denominada de carregamento, e fisicamente é constituído de atuadores e sensores de final de curso, cuja dinâmica respondem em relação a eventos discretos e correspondendo a seção 1 da Figura 4.2. A seção 2 corresponde à esteira transportadora que permite levar o produto até a próxima estação. A movimentação da esteira é realizada por um motor de corrente continua que é controlado através de um sistema liga desliga.

Uma das razões para realizar uma modelagem baseada em arquitetura híbrida desse sistema tem que ver com a possibilidade de flexibilizar a configuração da linha de montagem, considerando que normalmente em aplicações industriais semelhantes, o mesmo é controlado através de eventos discretos, tornando-se assim o sistema muito rígido.

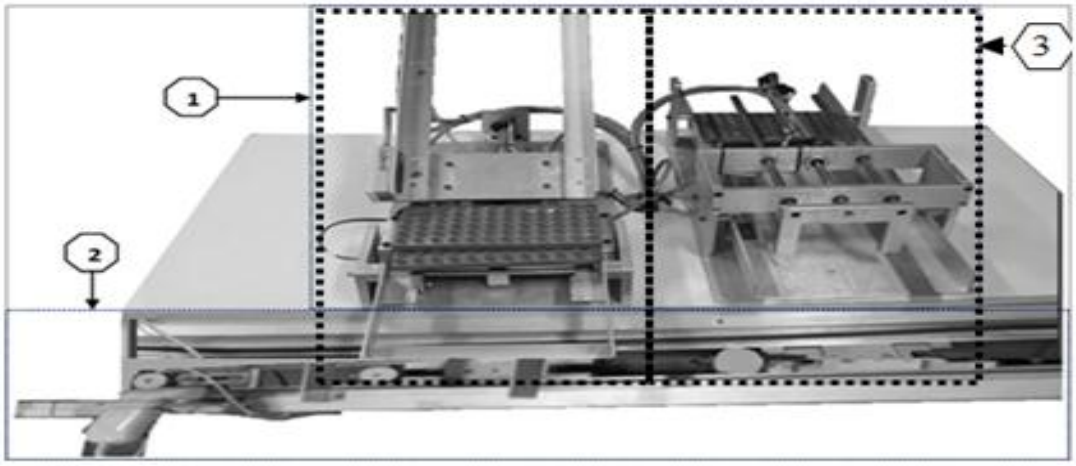

Figura 4.2 Estação de Carregamento

4.2.2 Descrição da Unidade de Montagem e Desmontagem Lateral

Esta unidade de trabalho começa a transformação de matérias primas, neste caso uma placa de base e cubos de montagem LEGO™, em um produto terminado. O trabalho realizado nesta estação é receber a placa base, colocar ela em uma plataforma e depositar cubos na parte lateral esquerda e direita de acordo com os requisitos de montagem. A montagem pode ser

realizada em até dois níveis, o que permite ter diferentes tipos de produtos. Além da montagem de cubos esta estação também possibilita a desmontagem dos mesmos, isto facilita o processo de remontagem do produto em caso que este tiver algum erro de montagem. A Figura 4.3 apresenta esta unidade.

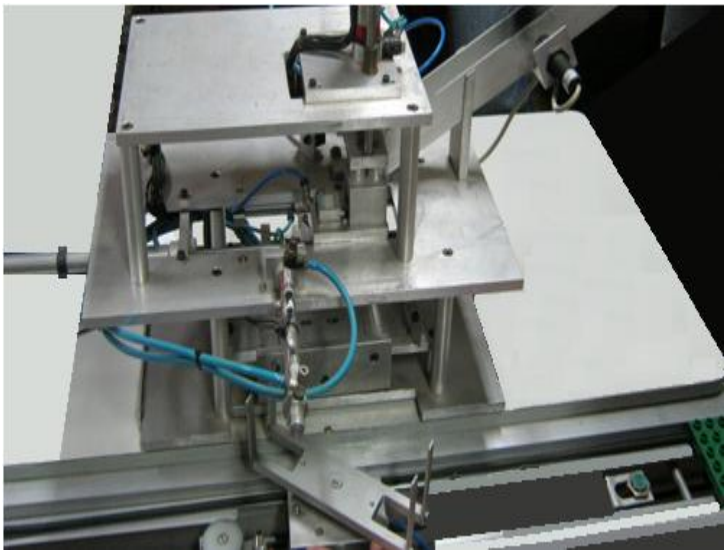

Figura 4.3 Unidade de Montagem e Desmontagem Lateral

4.2.3 Descrição da Unidade de Montagem Central.

A atividade programada nesta estação é a montagem dos cubos na parte central da placa base, de igual maneira que se realiza a montagem e desmontagem lateral, pode-se colocar até dois níveis de cubos de acordo aos requerimentos realizados, esta estação é apresentada na Figura 4.4.

Depois de colocar os cubos na parte central, o produto é devolvido para o sistema de transferência, responsável pelo transporte do mesmo até a próxima estação de trabalho, que é o posto de controle de qualidade.

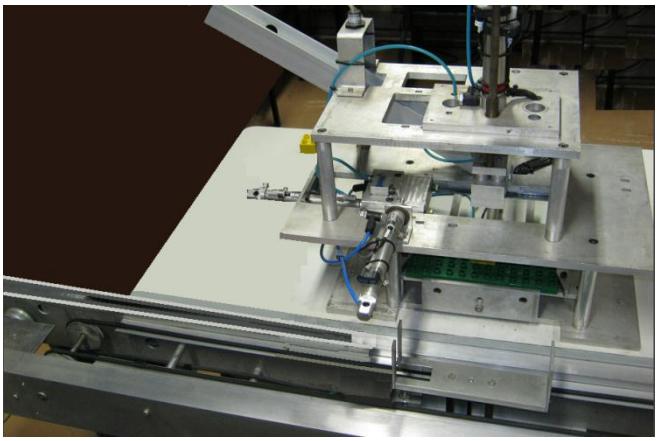

Figura 4.4 Unidade de Montagem Central

4.2.4 Descrição da Unidade de Inspeção

Este posto de trabalho é responsável pela verificação se a manufatura dos produtos esteja realizada de acordo aos requerimentos da produção. Esta tarefa se realiza através de sensores que verificam se a montagem ou desmontagem dos cubos sobre a placa de base esta conforme o requisito de produção. A partir desta informação o sistema decide o armazenamento do mesmo como produto terminado ou se ele será refugado ou colocado novamente no sistema de transferência para remontagem. A Figura 4.5 mostra este posto de trabalho.

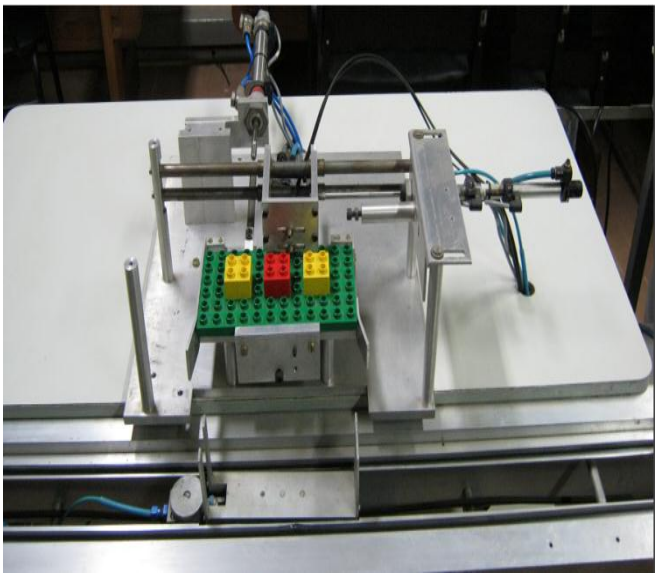

Figura 4.5 Unidade de Inspeção.

4.2.5 Descrição do Sistema de Transferência.

Este sistema está constituído por uma esteira de transporte independente para cada unidade de produção e permite transportar as matérias primas e os produtos em processo para cada uma destas unidades. Cada seção é movimentada através de motores de corrente contínua (CC) e originalmente tem um controle de tipo liga desliga (on-off) o que pode acarretar num desgaste do sistema e um consumo de potência elevado.

4.3 Descrição Funcional através de eventos.

O diagrama funcional do trabalho representado através de seu Grafcet funcional (Figura 4.6) correspondente ao funcionamento da linha de montagem. Nesta aplicação é apresentado a sequência de trabalho necessária para realizar a montagem de um produto composto unicamente de cubos na parte lateral. Com esse objetivo, participam desta atividade as estações de carregamento, montagem e desmontagem lateral, inspeção e armazenamento de produtos. Dentro do contexto proposto neste trabalho de pesquisa as unidades são apresentadas como um módulo encarregado de realizar uma tarefa correspondente, sem necessidade de apresentar eles detalhadamente.

O primeiro que é realizado é a liberação da placa base, uma vez realizada esta operação, o motor movimenta a seção da esteira correspondente a esta estação de trabalho, sendo acionado até que o produto chegue à estação de montagem lateral, onde é realizada esta tarefa. Uma vez terminada a operação de montagem, o motor de acionamento desta estação de trabalho é acionado até que o produto chegue à estação de inspeção. Nesta estação é verificada a conformidade do produto, definindo se o mesmo irá ser armazenado, rejeitado ou se continuará na linha automatizada para outra montagem.

Com a atual dinâmica de eventos sequenciais implementada nesta linha de montagem, é difícil conseguir conhecer os tempos de produção do sistema, pois não está implementado um controle regulatório no sistema de transferência que existe entre cada uma das estações, que será

um dos objetivos deste trabalho de pesquisa que implementará uma estratégia de controle híbrido .

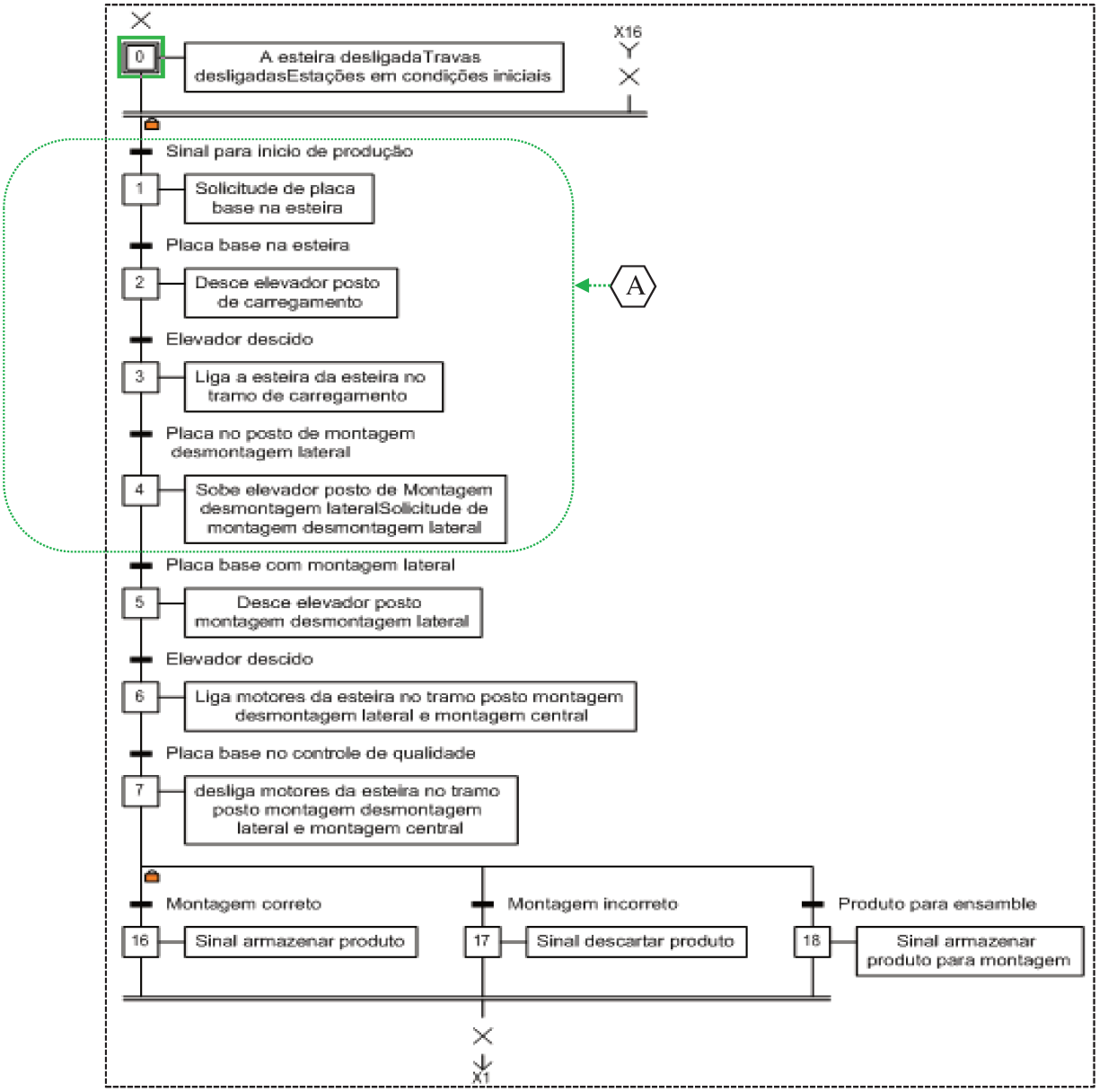

Figura 4.6 Diagrama Funcional do Sistema de Produção

4.4 Modelagem e Controle da Estação de Carregamento

Como validação da proposta de modelagem, controle e síntese de sistemas de arquitetura híbrida, será proposto o estudo de um sistema híbrido correspondente a parte A da Figura 4.6

apresentada anteriormente que consiste da estação de carregamento da PIPEFA e parte do sistema de transferência.

Neste trabalho de pesquisa foi realizado a modelagem desta unidade do sistema de produção e o projeto de uma estratégia de controle considerando a interação dos sinais que são regidos em função de eventos e em relação ao tempo. Como se apresenta nos capítulos seguintes deste trabalho, como atividade final foi realizada uma síntese do sistema de controle, utilizando o conceito da prototipagem rápida. Uma descrição funcional, representada pelo grafo de comando de etapa e transição é mostrada na Figura 4.7.

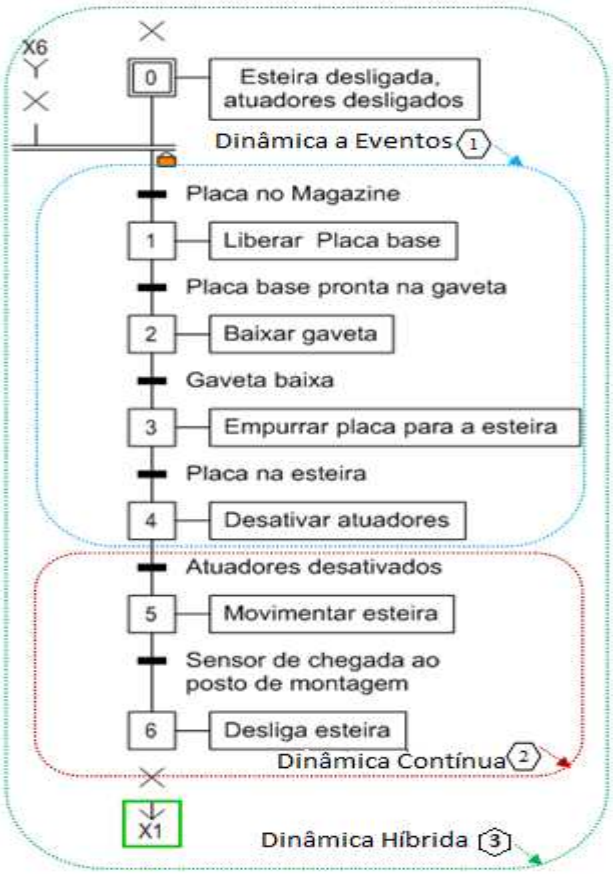

Figura 4.7 Diagrama Funcional da Estação de Carregamento

A seção apresentada como 1 na Figura 4.7 corresponde à dinâmica de eventos discretos, que neste modelo híbrido vai continuar da mesma forma, enquanto a seção 2 vai ser modelada

como um sistema dinâmico que responde em relação ao tempo, a integração das duas dinâmicas podem alcançar a representação do sistema em sua dinâmica híbrida.

O sistema será analisado com um sinal de evento discreto que representa a liberação da placa na esteira pela unidade de carregamento, um sinal de saída da placa da esteira e a dinâmica da esteira levando em conta seu comportamento em relação ao tempo que vai ser modelada em variáveis de estado para logo ser quantificadas utilizando QSS e seu posterior mapeamento do modelo para o formalismo DEVS.

A parte correspondente a eventos discretos vai ser modelada utilizando o formalismo DEVS, e essa terá uma interação direta com a dinâmica da esteira, colocando carga adicional quando entra o produto na esteira e tirando a mesma quando o produto saia.

O modelo da parte dinâmica de eventos discretos e a dinâmica que trabalha em relação ao tempo não formalismo DEVS, se acoplarão com o propósito de ter a dinâmica hibrida do sistema. Finalmente vai ser realizada a simulação do sistema, gerando eventos de entrada e saída o que muda a inercia de carga, evidenciando dessa maneira a vantagem do modelo híbrido na simulação da interação das duas dinâmicas em um processo industrial.

4.4.1 Modelo do Sistema de Transferência

O sistema de transferência é composto de uma esteira que é movimentada por um motor que trabalha em relação ao tempo. Para modelagem dinâmica do sistema de acionamento foi utilizado os parâmetros obtidos a partir das características técnicas de um motor de corrente contínua GM9234S031 de Pittman™,

De acordo com (ISERMANN, 2002) o sistema pode ser modelado a partir da Equação 4.1 que representa o comportamento elétrico e mecânico do sistema em estudo.

$$\frac{d}{dt}i_a = -\frac{R_a}{L_a}i_a - \frac{k_v}{L_a}w_a + \frac{V_a}{L_a} \tag{4.1}$$

Onde,

R_a é a resistência do enrolamento do rotor, L_a é a indutância do mesmo e K_v corresponde à constante do motor, que está relacionada com a força contra eletromotriz produzida pela velocidade angular.

Na parte mecânica do motor considera sua carga inercial que corresponde à produzida pelo rotor, um atrito viscoso que corresponde à fricção das escovas com o rotor, a fricção do eixo e a carcaça do motor. Além disso, considera-se o torque de carga o qual é independente da velocidade. Por tanto a descrição matemática da parte mecânica está dada pela Equação (4.24.2.

$$\frac{d}{dt}w_a = \frac{k_t}{J}i_a - \frac{B}{J}w_a - \frac{T_L}{J} \tag{4.2}$$

Onde J é a inércia correspondente ao rotor e carga acoplada, w_a a velocidade angular que considera que o torque produzido pelo motor é diretamente proporcional á corrente de armadura, e através de um fator de proporcionalidade K_T, obtém-se a relação eletromecânica apresentada na Equação 4.3. Onde $T_e$corresponde com o torque elétrico do motor.

$$T_e(t) = K_T i_a(t) \tag{4.3}$$

É importante destacar que no sistema internacional de medidas (SI) a constante de armadura e a constante do motor são aproximadamente iguais, ou seja, pode-se considerar $K_T = K_E$. Além disso, a força contra-eletromotriz é proporcional a velocidade de rotação do motor. Assim, substituindo a Equação 4.3 obtêm-se a Equação 4.4.

$$K_T i_a(t) = J\alpha(t) + bw(t) + T_F \tag{4. 4}$$

Selecionando como as variáveis de estado: corrente do motor, velocidade e posição, como mostra a Equação 4.5, chega-se à representação do espaço de estado do sistema, expresso pelas Equações 4.6 e 4.7.

$$\begin{cases} \dot{x}_1(t) = \dfrac{di_a(t)}{dt} \\ \dot{x}_2(t) = \alpha(t) \\ \dot{x}_3(t) = x_2(t) \end{cases} \tag{4.5}$$

$$\dot{x}_1(t) = -\frac{\mathrm{R_a}}{\mathrm{L_a}} x_1(\mathrm{t}) - \frac{\mathrm{k_v}}{\mathrm{L_a}} x_2(\mathrm{t}) + \frac{\mathrm{V_a}(t)}{\mathrm{L_a}} \tag{4.6}$$

$$\dot{x}_2(\mathrm{t}) = \frac{\mathrm{k_t}}{\mathrm{J}} \mathrm{x_1}(\mathrm{t}) - \frac{\mathrm{B}}{\mathrm{J}} \mathrm{x_2}(\mathrm{t}) - \frac{\mathrm{T_L}(\mathrm{t})}{\mathrm{J}} \tag{4.7}$$

Tomando como saída à velocidade se tem a Equação 4.8.

$$y(t) = x_2(\mathrm{t}) \tag{4. 8}$$

Onde pode-se representar o sistema de acordo com a Equação (4.94.9.

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{dt}} \begin{bmatrix} \mathrm{i_a} \\ \mathrm{w_a} \end{bmatrix} = \begin{bmatrix} -\dfrac{\mathrm{R_a}}{\mathrm{L_a}} & -\dfrac{\mathrm{K_v}}{\mathrm{L_a}} \\ \dfrac{\mathrm{k_t}}{\mathrm{J}} & -\dfrac{\mathrm{B}}{\mathrm{J}} \end{bmatrix} \begin{bmatrix} \mathrm{i_a} \\ \mathrm{w_a} \end{bmatrix} + \begin{bmatrix} \dfrac{1}{\mathrm{L_a}} & 0 \\ 0 & -\dfrac{1}{\mathrm{J}} \end{bmatrix} \begin{bmatrix} \mathrm{V_a} \\ \mathrm{T_L} \end{bmatrix} \\ \begin{bmatrix} \mathrm{y_1} \\ \mathrm{y_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{i_a} \\ \mathrm{w_a} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{V_a} \\ \mathrm{T_L} \end{bmatrix} \end{cases} \tag{4.9}$$

4.4.2 Simulação do sistema de transferência da estação de carregamento

Utilizando-se os parâmetros fornecidos pelo fabricante do motor do sistema de acionamento apresentados na Tabela 4.1 e Equações 4.6 e 4.7, as equações de estado do sistema serão as descritas pela Equação 4.10.

$$
\begin{cases}
\dot{x}_1(t) = \dfrac{V_a(t) - 1{,}26x_1(t) - 0{,}0232x_2(t)}{0{,}00102} \\
\dot{x}_1(t) = 980{,}39\left[V_{a(t)} - 1{,}26x_1(t) - 0{,}0232x_2(t)\right] \\
\dot{x}_2(t) = \dfrac{0{,}0232x_1(t) - 2{,}6 \times 10^{-6}x_2(t) - T_L(t)}{0{,}00102} \\
\dot{x}_2(t) = 0{,}238 \times 10^6[0{,}0232x_1(t) - 2{,}6 \times 10^{-6}x_2(t) - T_L(t)] \\
\dot{x}_3(t) = x_2(t)
\end{cases}
\tag{4.10}
$$

Substituindo em 4.9, se obtém o modelo em tempo contínuo do motor que movimenta a esteira e que se apresenta em 4.11.

$$
\dot{\bar{x}}(t) = \begin{bmatrix} -1235{,}29 & -22{,}74 & 0 \\ 5523{,}81 & -0{,}62 & 0 \\ 0 & 1 & 0 \end{bmatrix} \bar{x}(t) + \begin{bmatrix} 980{,}39 & 0 \\ 0 & -0{,}238 \times 10^6 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_S(t) \\ T_L(t) \end{bmatrix}
\tag{4.11}
$$

Utilizando-se o modelo obtido foram realizadas simulações utilizando-se blocos em Matlab-Simulink®, implementando-se um bloco correspondente a Equação mecânica e outro para a parte elétrica do motor. A Figura 4.4.8 apresenta o bloco correspondente à Equação elétrica, e a Figura 4.99 o bloco correspondente a parte mecânica, onde a realimentação da velocidade expressa em rad/s, é convertida em RPM para uma melhor visualização.

.

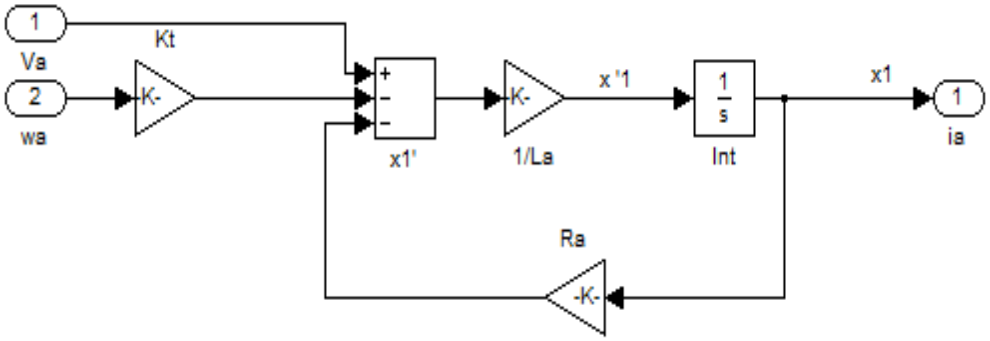

Figura 4.8 Diagrama correspondente a Equação Elétrica

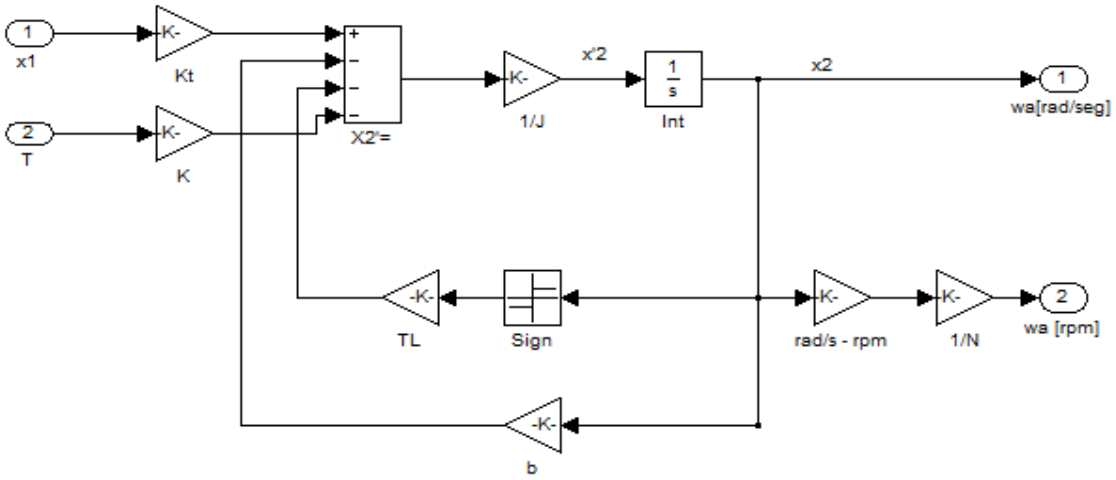

Figura 4.9 Diagrama correspondente ao Modelo Mecânico.

Inicialmente foi realizada a simulação apenas do motor, não considerando a carga aplicada. A resposta temporal é apresentada na Figura 10, onde se pode observar que alcança seu estado de regime estacionário em aproximadamente 22 RPM, que corresponde aos dados fornecidos pelo fabricante do motor.

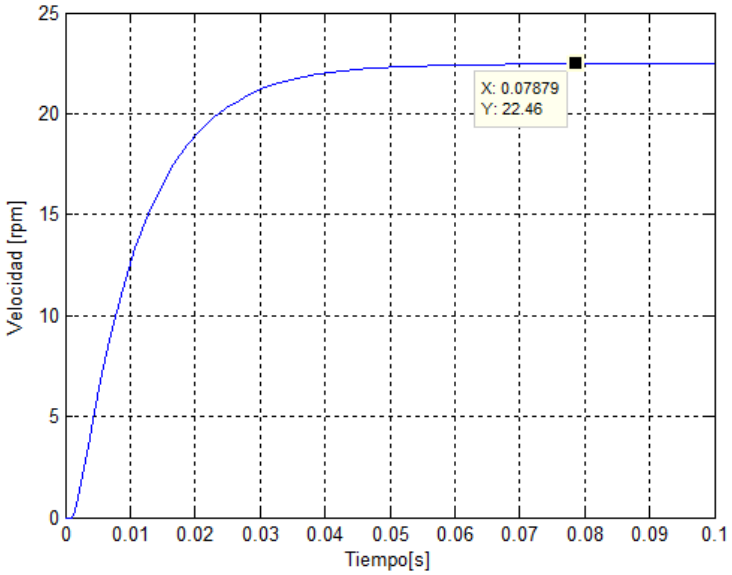

Figura 4.10. Saída de Velocidade do Motor.

4.5 Considerações Finais

Neste capítulo se apresentou uma descrição da Plataforma Industrial para Pesquisa, Ensino e Formação em Automação. Realizando ênfase na unidade de produção de carregamento, que vai ser objeto de estudo no capítulo 5 desta tese.

A unidade de carregamento vai ser analisada levando em conta sua arquitetura híbrida que mistura a dinâmica regida através de eventos da parte que libera a placa de montagem e a parte continua da secção de transferência. Essa mistura das duas dinâmicas é o que constitui um sistema hibrido.

No próximo capítulo vai ser apresentado um estudo de caso de modelagem y projeto de controle da unidade de carregamento, vista como um sistema de arquitetura híbrida, utilizando os formalismos DEVS e QSS.

Capítulo 5

5. Estudo de caso: Modelagem e controle do sistema de arquitetura híbrida

5.1 Introdução

Como parte da validação da proposta de modelagem, controle e síntese de um sistema de produção industrial com arquitetura de controle hibrida baseada nos formalismos DEVS e o método de estados quantificados (QSS), neste capítulo é realizado um estudo de caso aplicado a uma unidade de trabalho de um sistema de produção industrial integrante da Plataforma Industrial para Pesquisa, Ensino e Formação em Automação (PIPEFA) implementada no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP que representa um chão de fabrica industrial retratando aspectos reais de produção.

5.2 Modelo Quantificado

A partir das Equações de estado considerando a Equação 4.11, foi realizada a quantificação das Equações de estado do sistema, com o objetivo de convertê-lo num sistema de estados quantificados. Assim, o sistema de Equações 5.1 representa o sistema em QSS, considerando a Equação 3.8.

$$\begin{cases} \dot{x}_1(t) = 980{,}39 \cdot V_S(t) - 1235{,}29 \cdot (x_1(t) + \Delta x_1) - 22{,}74 \cdot (x_2(t) + \Delta x_2) \\ \dot{x}_2(t) = 5523{,}81 \cdot (x_1(t) + \Delta x_1) - 0{,}62 \cdot (x_2(t) + \Delta x_2) - 0{,}238 \times 10^6 \cdot T_L(t) \\ \dot{x}_1(t) = 980{,}39 \cdot V_S(t) - 1235{,}29 \cdot q_1(t) - 22{,}74 \cdot q_2(t) \\ \dot{x}_2(t) = 5523{,}81 \cdot q_1(t) - 0{,}62 \cdot q_2(t) - 0{,}238 \times 10^6 \cdot T_L(t) \end{cases} \tag{5.1}$$

Cada variável é quantificada através de um valor Δxn, denominada Quantum, convertendo ela em uma variável qn.

Um esquema representativo do sistema em estados quantificados é apresentado na Figura 5.1 FE1 e FE2 representam as funções estáticas, e são responsáveis de realizar as operações matemáticas básicas das Equações de estado. VS e TL correspondem ao vetor de entradas do sistema e as variáveis q1 e q2 correspondem ao vetor das variáveis de estado quantificadas. Pode-se também observar que os blocos de função estática geram como resultado as derivadas das variáveis de estado, que logo são integradas e quantificadas através dos blocos integradores QSS.

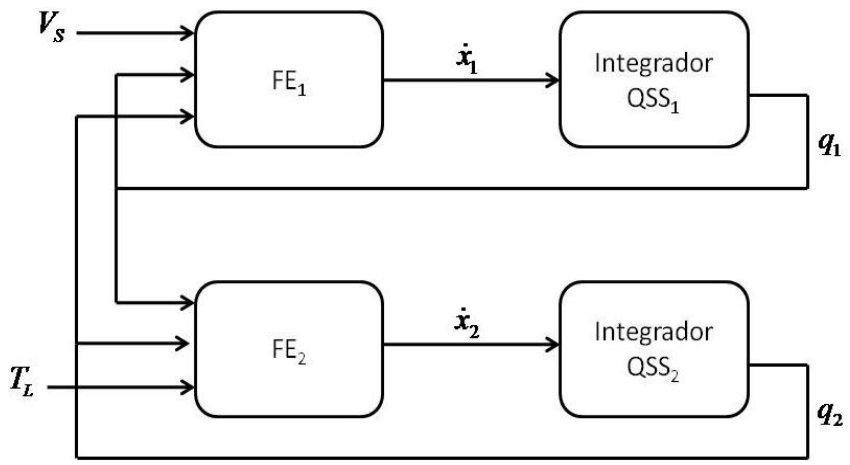

Figura 5.1 Diagrama de Blocos do Sistema em QSS.

Tomando como base a estrutura geral DEVS que representa um integrador quantificado com histereses, representada através da Equação 3.14, a Equação 5.2 descreve a estrutura DEVS para a Equação estática de x'1:

$$\begin{gathered}
M_{FE1} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{onde} \\
X = Y = \mathbb{R} \times \mathbb{N}; S = \mathbb{R}^3 \times \mathbb{R}_0^+ \\
\delta_{int}(s) = \delta_{int}(u, q_1, q_2, \sigma) = (u, q_1, q_2, \infty) \\
\delta_{ext}(s, e, (x_v, port)) = \delta_{ext}(u, q_1, q_2, \sigma, e, (x_v, port)) = \begin{Bmatrix} (x_v, q_1, q_2, 0) & \text{si port} = 0 \\ (u, x_v, q_2, 0) & \text{si port} = 1 \\ (u, q_1, x_v, 0) & \text{si port} = 2 \end{Bmatrix} \\
\lambda(s) = \lambda(u, q_1, q_2, \sigma) = (y, port) = (980.39 * u - 1235.29 * q_1 - 22.74 * q_2, 0) \\
ta(s) = ta(u, q_1, q_2, \sigma) = \sigma
\end{gathered} \tag{5.2}$$

Nessa estrutura pode-se observar que o bloco DEVS conta com três portas de entrada, pelas quais ingressam eventos com os valores de tensão de entrada (variável u) e das variáveis de

estado quantificadas. Além disso, conta com somente um ponto de saída, cujo valor é calculado pela derivada de x1 a partir da Equação de estado, e dependendo somente a eventos externos ao sistema, pelo qual se atualiza a variável σ com um valor de infinito ao executar-se uma transição interna.

De maneira similar, a estrutura DEVS para a Equação estática de x'2 está dada pela equação 5.3.

$$\begin{gathered}
M_{FE2} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{onde} \\
X = Y = R \times N; S = R^3 \times R_0^+ \\
\delta_{int}(s) = \delta_{int}(q_1, q_2, u, \sigma) = (q_1, q_2, u, \infty) \\
\delta_{ext}(s, e, (x_v, port)) = \delta_{ext}(q_1, q_2, u, \sigma, e, (x_v, port)) = \begin{Bmatrix} (x_v, q_2, u, 0) & \text{si port} = 0 \\ (q_1, x_v, u, 0) & \text{si port} = 1 \\ (q_1, q_2, x_v, 0) & \text{si port} = 2 \end{Bmatrix} \\
\lambda(s) = \lambda(q_1, q_2, u, \sigma) = (y, port) = (5523.81 * q_1 - 0.62 * q_2 - 0.238 \times 10^6 * u, 0) \\
ta(s) = ta(u, q_1, q_2, \sigma) = \sigma
\end{gathered} \tag{5.3}$$

De modo similar ao modelo anterior, o mesmo contém três portas de entrada e uma de saída, através do qual se envia o valor calculado da derivada, segundo com a Equação de estado correspondente. Através das portas de entrada se tem respectivamente as variáveis q1, q2 e o torque de carga, representado novamente como uma variável de entrada u.

Uma vez estabelecidas às estruturas DEVS que definem as Equações de estado quantificadas, pode-se gerar um modelo de simulação, pelo qual se utiliza a ferramenta Stateflow® de Matlab®. A Figura 5.2 mostra o diagrama de estados criado de acordo à estrutura e funcionamento de um modelo DEVS atômico, que permite simular qualquer sistema representado neste formalismo, escrevendo as operações correspondentes a cada uma de suas funções.

Em geral os modelos da função estática e do integrador quantificado têm a mesma estrutura de código apresentada anteriormente, sendo diferenciadas apenas no conteúdo das funções de transição e a função de saída.

O diagrama de estados mostrado na Figura 5.2 apresenta dois estados principais chamados Estados_DEVS e Avanço_Tempo contidos em um quadro pontilhado, mostrando que esses estados são executados em paralelo.

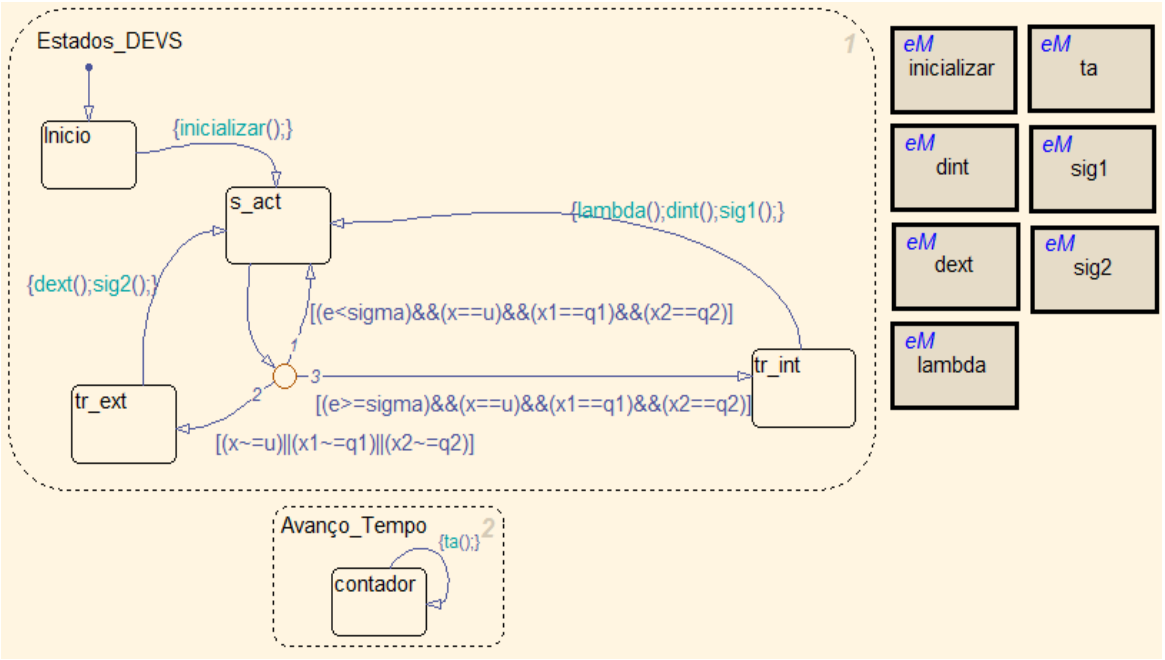

Figura 5.2 Diagrama de Estados em Stateflow para o Modelo DEVS Atómico.

No interior do primeiro estado concentram-se quatro estados: *Inicio, s_act, tr_ext e tr_int,* que fazem referencia respectivamente aos estados do sistema: inicial, atual, de transição externa e de transição interna. Dentro do estado *Avanço_Tempo*, tem-se um estado que realiza á função de avanço de tempo ta() em cada ciclo de simulação, onde essa função incrementa a variável de tempo transcorrido e.

Na transição do estado inicio é executada a função inicializar, com a qual se estabelecem as condições iniciais das variáveis do estado do sistema, e consequentemente passa-se ao estado *s_act,* permanecendo o sistema até que haja um evento externo ou se cumpra o tempo previsto para ocorrência de uma transição interna.

Quando é detectado um evento de entrada o sistema passa ao estado *tr_ext*, e são executadas as funções *dext* e *sig2*. Essas funções realizam a atualização das variáveis do sistema

e muda o valor da variável de tempo *sigma*, gerando uma transição interna. Quando o valor da variável *e,* alcança o valor da variável de tempo *sigma*, o sistema passa ao estado *tr_int*, e são executadas as funções *lambda*, que gera um evento de saída, *dint* que atualiza as variáveis de estado do sistema e *sig1*que define o novo valor para a variável de tempo. A Figura 5.3 mostra o diagrama de blocos de simulação da esteira modelado através de DEVS, e utilizando integradores QSS.

Neste diagrama, cada bloco corresponde a um modelo DEVS atômico e os blocos de atraso de um ciclo são para evitar possíveis erros na simulação. Além disso, pode-se ver que se utiliza um bloco de ganho para observar a saída em RPM e compará-la com o modelo contínuo do sistema.

Na Figura 5.3 se observa a presença de um terceiro integrador que permite a obtenção da posição da esteira. Na mesma figura, pode-se observar que se tem disponível também a variável de estado quantificada q1 correspondente à corrente de armadura.

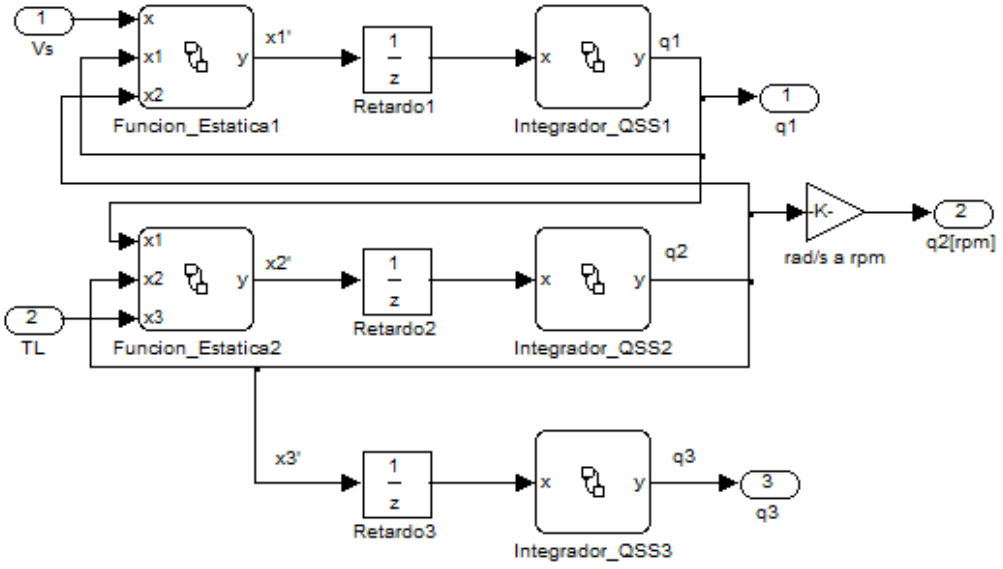

Figura 5.3 Modelo da Dinâmica Contínua.

Com o intuito de validar e comparar através da simulação o modelo em QSS obtido da parte continua do sistema, é implementado em Simulink® o modelo do sistema continuo e em

QSS, como é mostrado na Figura 5.4. Além disso, é incluído um bloco correspondente a diferença entre a saída quantificada com a saída continua, permitindo assim a obtenção da informação relativa ao erro do sistema.

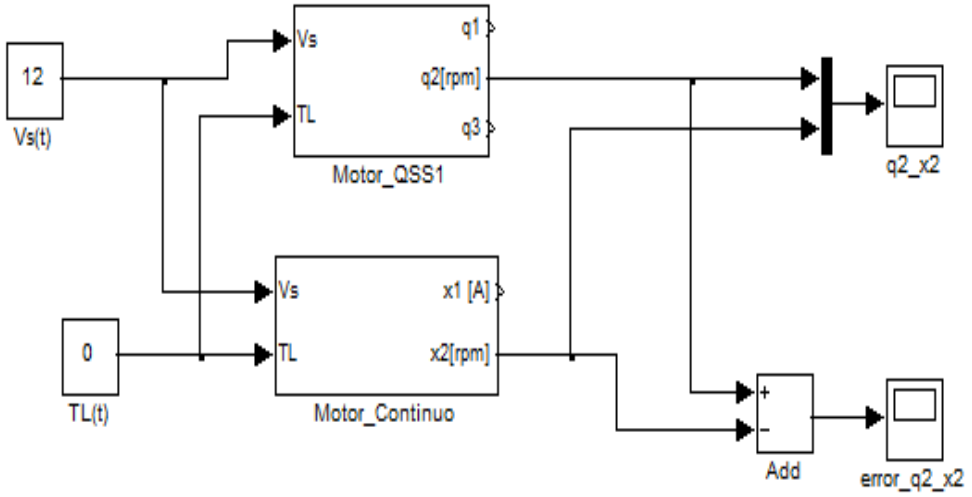

Figura 5.4 Diagrama Geral de Simulação do Modelo da Esteira.

5.3 Cálculo do Erro do Sistema Quantificado

Os valores de quantificação (ΔQ) para os três integradores da Figura 5.3 são:

$$\Delta Q_1 = 0{,}5A$$
$$\Delta Q_2 = 11 rad/s \approx 0{,}5 rpm$$
$$\Delta Q_3 = 20 rad$$

A partir desses valores, e das matrizes de valores próprios e os vetores próprios do sistema apresentados anteriormente, pode-se calcular o valor do erro correspondente a cada variável de estado quantificada por QSS, em relação à variável de estado contínuo de acordo com a Equação 3.18. Para este cálculo utiliza-se somente as Equações do sistema para x'1 e x'2, já que as mesmas são as que definem o comportamento dinâmico da esteira (equações elétrica e mecânica do motor). Consequentemente a matriz de estados utilizada e o vetor de quantum se apresentam na equação 5.4.

$$A = \begin{bmatrix} -1235{,}29 & -22{,}74 \\ 5523{,}81 & -0{,}62 \end{bmatrix} \quad ; \quad \Delta Q = \begin{bmatrix} \Delta Q_1 \\ \Delta Q_2 \end{bmatrix} = \begin{bmatrix} 0{,}5 \\ 11 \end{bmatrix} \tag{5.4}$$

Assim, os valores e vetores próprios do sistema obtidos através do Matlab® são respectivamente, representados na Equação 5.5.

$$\Lambda = \begin{bmatrix} 1123{,}4 & 0 \\ 0 & -112{,}5 \end{bmatrix} \quad ; \quad V = \begin{bmatrix} -0{,}1992 & 0{,}0202 \\ 0{,}98 & -0{,}998 \end{bmatrix} \tag{5.5}$$

O cálculo do valor do erro e(t) entre a variável de estado quantificada $\tilde{x}(t)$ e a variável de estado continua $x(t)$, pode ser obtido através da Equação 3.20. Considerando ΔQ = ε, onde ε é a largura da histerese, pode-se assegurar uma baixa frequência de oscilação final juntamente com um erro pequeno. Assim, a Equação de erro pode ser representada através da Equação 5.6.

$$|e(t)| = |V| \cdot \left|\mathrm{Re}\left(\Lambda^{-1}\right) \cdot \Lambda\right| \cdot \left|V^{-1}\right| \cdot (2\Delta Q) \tag{5.6}$$

Avaliando a expressão com as matrizes correspondentes, o valor máximo do erro para cada variável de estado é definido através da Equação 5.7(5.7).

$$|e(t)| = \begin{bmatrix} 2{,}21 \\ 37{,}79 \end{bmatrix} \tag{5.7}$$

Conforme mostra este vetor, o erro máximo obtido na variável de estado quantificada q1 é de 2,21A, enquanto o erro máximo na variável q2 é de 37,79 rad/s, ou seja, 1,65 RPM. A Figura 5.5 mostra a saída do sistema contínuo (linha verde) e a saída do sistema QSS (linha azul).

Através desta simulação pode-se observar que o modelo da esteira obtido mediante o método de QSS, está muito próximo do comportamento do sistema contínuo, permitindo assim validar através desta simulação o modelo em QSS.

Uma característica importante a ser considerada é com a quantidade de passos realizados para esta simulação. Neste caso foi realizado um total de 49 passos durante os 0,1s, o que permite concluir que o baixo custo computacional no modelo QSS em relação à simulação do sistema contínuo.

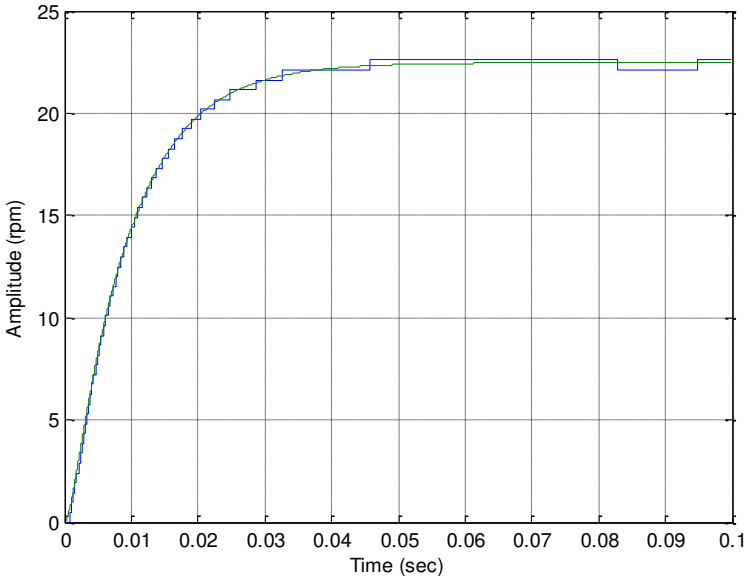

Figura 5.5 Saída do Sistema QSS Comparada com a Saída do Sistema Contínuo.

O erro correspondente a diferença entre a saída quantificada (q2) e a saída continua (x2) é apresentada na Figura 5.6, onde se pode observar que o valor (absoluto) máximo é de 0,55 RPM, e o mesmo se encontra dentro da faixa de erro calculado que é de 1,65 RPM.

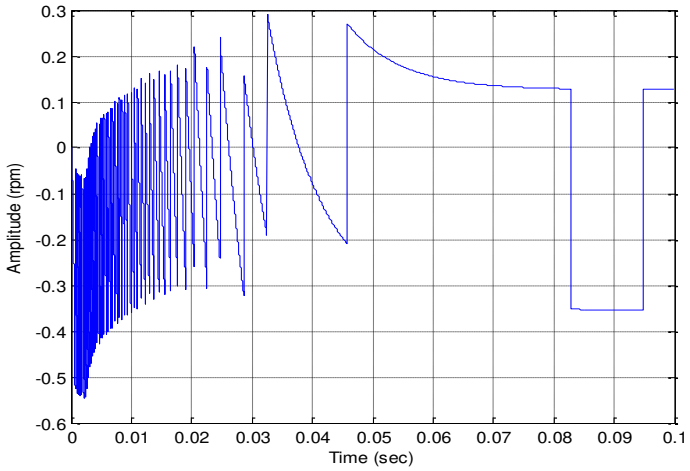

Figura 5.6 Erro da Diferença da Dinâmica em QSS e em Variáveis de Estado Contínuas.

5.4 Comparação do Modelo Quantificado com o Modelo de Tempo Discreto

Com o objetivo de se verificar o erro obtido entre o modelo quantificado e modelo em tempo discreto, torna-se importante realizar um trabalho de comparação da simulação do modelo da esteira utilizando QSS com o método tradicional de discretização temporal. Isto permitirá avaliar, a obtenção de um erro menor com uma quantidade de passos de interação menor utilizando QSS, ou em caso contrario, se é melhor utilizar o método tradicional de modelagem e simulação baseado em tempo. Com este objetivo será utilizado o método de aproximação numérica de Euler.

A Figura 5.7 mostra o diagrama de blocos utilizado para esta simulação. Neste diagrama tem-se o modelo contínuo da planta em variáveis de estado, que serve como referência para comparar o erro em relação à saída do motor, utilizando discretização temporal e QSS. Além disso, utilizam-se dois blocos de multiplicação e dois integradores, que permitem determinar o parâmetro de desempenho ISE (Integral Quadrática do Erro), de tal modo que se tenha um critério de validação e para comparação do erro obtido através dos dois modelos.

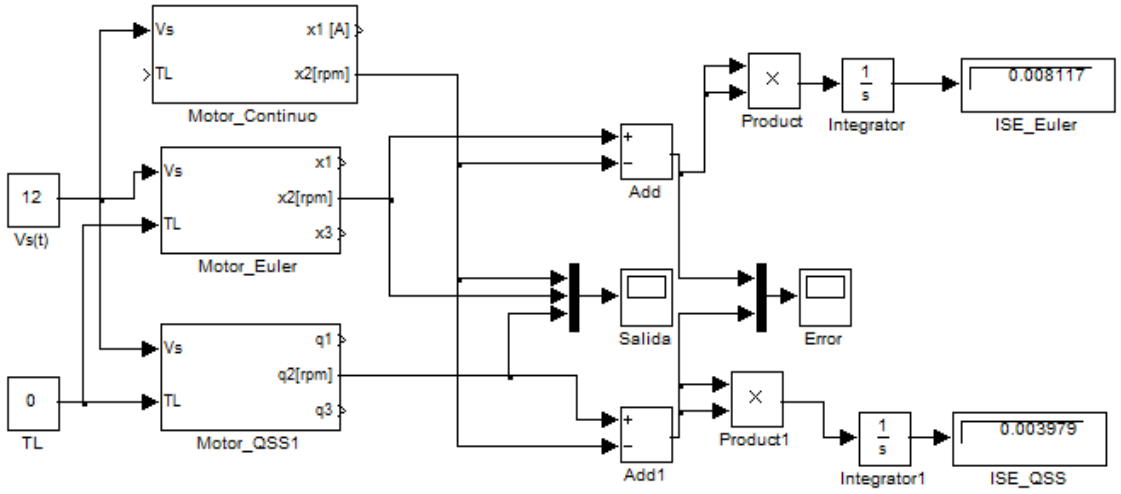

Figura 5.7 Modelo da Esteira em Discretização Temporal e QSS.

A simulação é realizada para um tempo de 100ms, utilizando um tempo de amostragem de 1ms para o modelo em tempo discreto (Motor_Euler), um valor de quantificação ΔQ1 de 0,5A e um ΔQ2 de 11rad/s. Através da Figura 5.7 pode-se observar que aparecem nos blocos "display" os valores da integral do erro para estas condições de simulação, onde o valor ISE para o modelo de tempo discreto clássico (bloco ISE_Euler) é de 0,0081 e o valor ISE para o modelo QSS (bloco ISE_QSS) é de 0,0039.

Levando em conta os resultados obtidos, pode-se observar que o modelo em QSS apresenta um menor erro de saída que o modelo em tempo discreto, comparando esses dois modelos com o modelo contínuo.

A Figura 5.8 mostra a simulação da saída do sistema, onde a linha 3 corresponde à saída do modelo contínuo, a linha 1 é a saída do modelo de tempo discreto através da aproximação de Euler e a linha 2 corresponde à saída do motor modelado mediante QSS. Nesta figura podemos observar que os dos modelos, discreto e QSS, seguem a trajetória do modelo contínuo. É importante observas que a simulação do modelo correspondente ao comportamento da esteira em tempo discreto necessitou de 100 passos de iteração, enquanto o modelo QSS necessitou apensas de 49 passos.

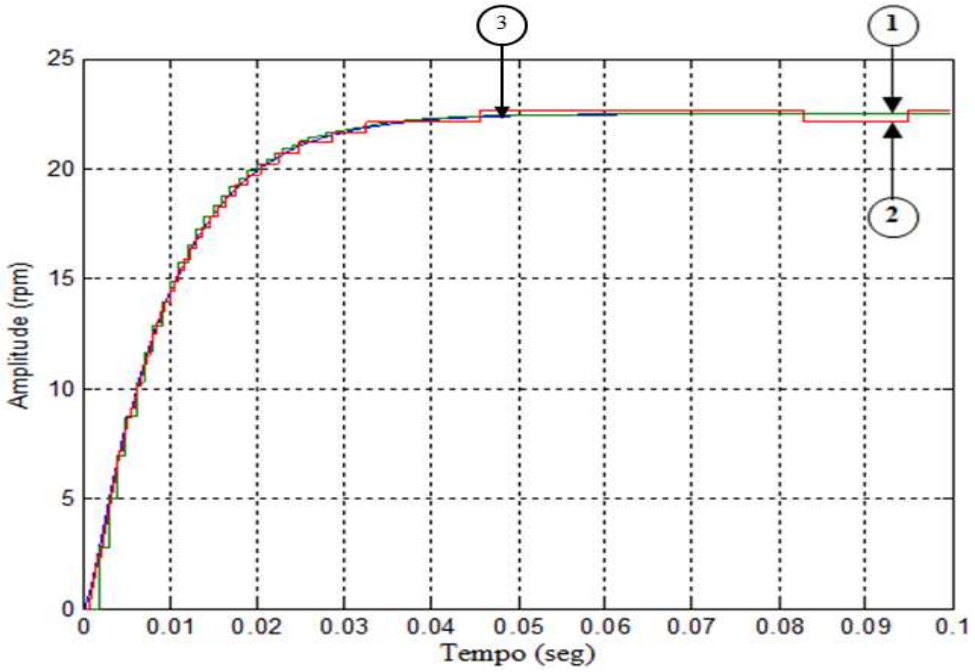

Figura 5.8 Saídas do Modelo Contínuo de Tempo Discreto e QSS

A Figura 5.9 mostra o erro correspondente a diferença entre a saída do modelo discreto (linha 2) e a saída do modelo QSS (linha 1) em relação ao modelo contínuo, confirmando o resultado apresentado na Figura 5.7 com respeito ao parâmetro ISE, já que se observa uma grande diferença entre o erro do modelo QSS e o do modelo discreto, que apresenta uma amplitude maior no sinal de erro relativo ao modelo discreto, enquanto que no estado de regime estacionário acontece o contrario.

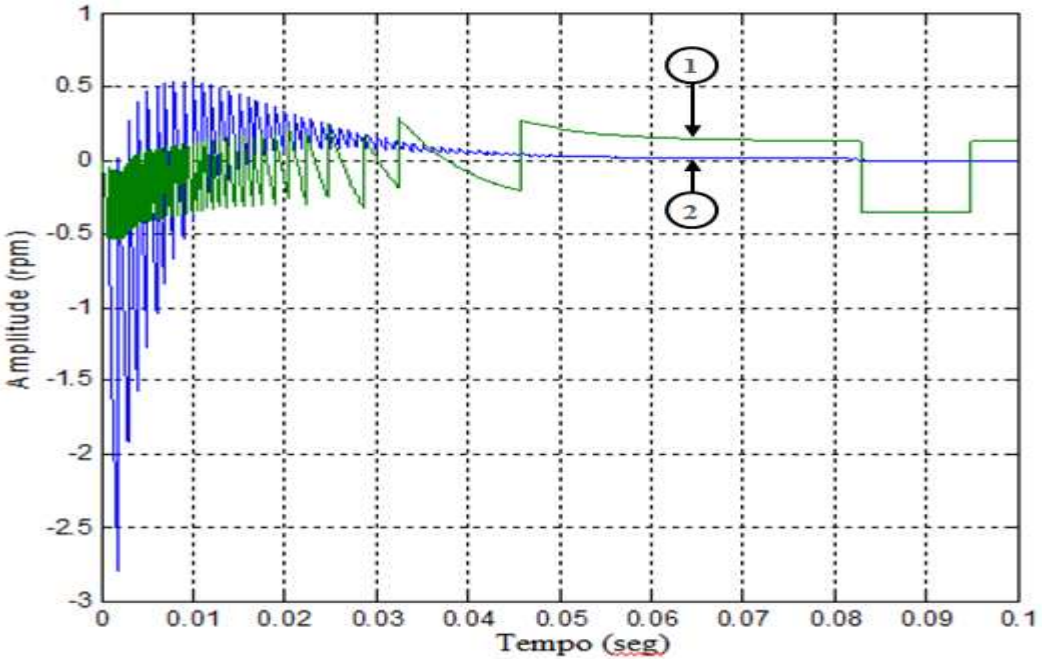

Figura 5.9 Erro correspondente a Diferença na Saída do Sistema Modelo Discreto e QSS.

Os resultados obtidos através das simulações apresentadas anteriormente encontram-se resumidas na Tabela 5.1, onde se pode concluir que o modelo QSS não só requer uma menor quantidade de passos que o modelo de tempo discreto para simular o comportamento do motor, mas também que o erro relativo à saída do modelo contínuo é também menor em relação ao modelo QSS.

Tabela 5.1 Resultados da simulação através da aproximação por Euler e QSS.

Parâmetro	Euler	QSS

Número de passos	100	49
Parâmetro ISE	0,0081	0,0039

5.5 Parâmetros do Controlador

Para a implementação do sistema de controle foi proposto um controlador do tipo PID. Para determinar as constantes Kp, Ki e Kd, o sistema será aproximado como um sistema de primeira ordem com tempo morto.

A Figura 5.10 mostra o comportamento da saída do sistema em estudo submetido a uma entrada em degrau. Nesta figura pode-se observar que o comportamento da saída do sistema que apresenta uma redução dos 72% no tempo de estabilização (utilizando o critério de 2%), em relação ao sistema em malha aberta e que o máximo sobressinal é do 2,7%.

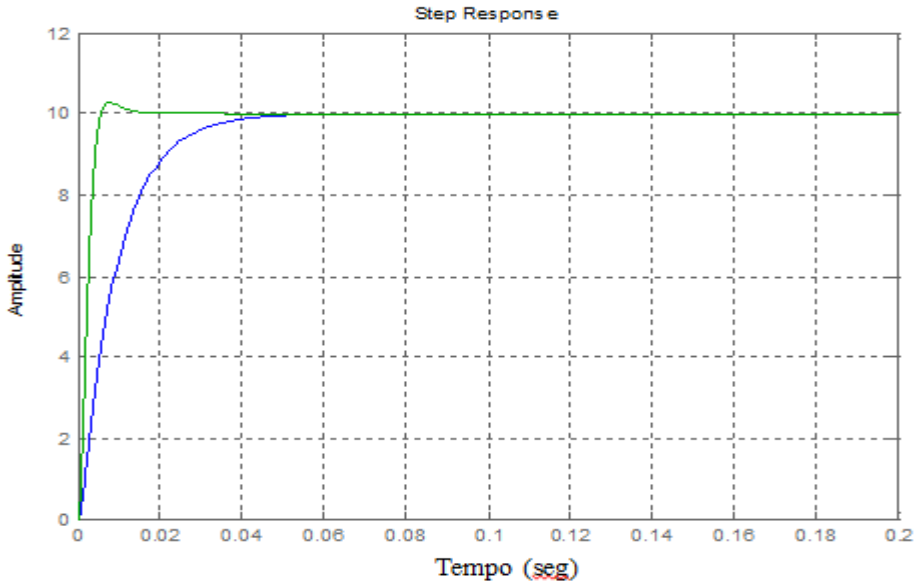

Figura 5.10 Saída do Sistema Controlado.

Os valores obtidos para sintonização do controlador PID contínuo foram calculados a partir de Ziegler Nichols, correspondendo aos parâmetros:

$$k_p = 25 \quad k_i = 5 \quad k_d = 0{,}2$$

A função de transferência de um controlador PID contínuo no tempo é descrito pela Equação 5.8.

$$\frac{Yc(s)}{E(s)} = \frac{k_d \cdot s^2 + k_p \cdot s + k_i}{s} \tag{5.8}$$

Que pode ser representada através da Equação 5.9.

$$\frac{Yc(s)}{E(s)} = \frac{b_0 \cdot s^2 + b_1 \cdot s + b_2}{s^2 + a_1 \cdot s + a_2} \tag{5.912}$$

Onde,

$$b_0 = k_d, \quad b_1 = k_p, \quad b_2 = k_i, \quad a_1 = 1, \quad a_2 = 0$$

5.5.1 Descrição do Controlador em Variáveis de estado

As variáveis de estado que descrevem o controlador pode ser escritas de acordo com o sistema de Equações 5.10.

$$\begin{aligned} x_1 &= y - \beta_0 \cdot u \\ x_2 &= \dot{x}_1 - \beta_1 \cdot u \\ \dot{x}_2 &= -a_2 \cdot x_1 - a_1 \cdot x_2 + \beta_2 \cdot u \end{aligned} \tag{5.10}$$

Com,

$$\beta_0 = b_0 = k_d$$
$$\beta_1 = b_1 - a_1 \cdot \beta_0 = k_p - k_d$$
$$\beta_2 = b_2 - a_1 \cdot \beta_1 - a_2 \cdot \beta_0 = k_i - k_p + k_d$$

As Equações de estado do PID devem ser definidas pelo sistema da Equação 5.11.

$$\left\{\begin{array}{c} \dot{x}_1 = x_2 + (k_p - k_d) \cdot e \\ \dot{x}_2 = -x_2 + (k_i - k_p + k_d) \cdot e \\ y_c = x_1 + k_d \cdot e \end{array}\right\} \tag{5.11}$$

Através desta descrição em variáveis de estado a função de transferência será representada através da Equação 5.12.

$$\frac{Yc(s)}{E(s)} = \frac{k_d \cdot s^2 + k_p \cdot s + k_i}{s(s+1)} \tag{5.12}$$

5.5.2 Representação do Sistema de Controle em DEVS

Para realizar a representação das variáveis de estado do controlador através do modelo DEVS utiliza-se o sistema da Equação 3.14. O diagrama de blocos que representa o controlador em estados quantificados com uma entrada e duas saídas é mostrado na Figura 5.11.

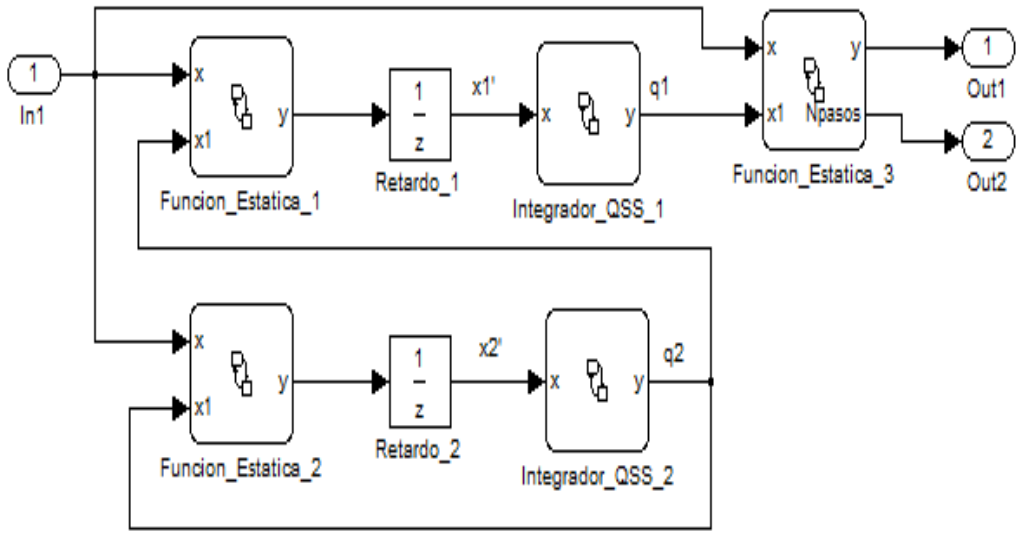

Figura 5.11 Diagrama de Blocos de Simulink™ do Controlador Quantificado.

A saída Out2 foi implementada unicamente para a simulação, pois permite visualizar o número de passos realizados pelo controlador (número de ações de controle). Pode-se observar também a função bloco *Funcion_Estatica_1, que* executa as operações aritméticas concernentes a primeira Equação de estado e o bloco Integrador_QSS_1 integra a variável x1' para gerar q1. Na

parte inferior encontram-se os blocos que realizam as operações da segunda variável de estado e a integração da mesma. Finalmente, o bloco *Funcion_Estatica_3* realiza as operações da Equação de saída do controlador.

Os blocos de funções estáticas e os blocos integradores contêm diagramas de estados idênticos aos apresentados na Figura 5.13, já que cada um é um modelo DEVS atômico. A estrutura do modelo DEVS do integrador QSS é mostrada na Equação 3.14, e as funções estática1 e 2, de acordo ao modelo em variáveis de estado para o controlador mostradas na Equação 5.11(5.11), onde as funções estão descritas pelas Equações 5.13(5.13), 5.14 e 5.15, apresentadas a seguir:

Função Estática 1:

$$\begin{gathered}
M_{FE1} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{onde} \\
X = Y = \mathbb{R} \times \mathbb{N}; S = \mathbb{R}^3 \times \mathbb{R}_0^+ \\
\delta_{int}(s) = \delta_{int}(u_r, u_c, q_1, \sigma) = (u_r, u_c, q_1, \infty) \\
\delta_{ext}(s, e, (x_v, port)) = \delta_{ext}(u_r, u_c, q_1, \sigma, e, (x_v, port)) = \begin{Bmatrix} (x_v, u_c, q_1, 0) & \text{si port} = 0 \\ (u_r, x_v, q_1, 0) & \text{si port} = 1 \\ (u_r, u_c, x_v, 0) & \text{si port} = 2 \end{Bmatrix} \\
\lambda(s) = \lambda(u_r, u_c, q_1, \sigma) = (y, port) = \left(q_1 + (u_r - u_c) * \left(k_p - k_d\right), 0\right) \\
ta(s) = ta(u_r, u_c, q_1, \sigma) = \sigma
\end{gathered} \tag{5.13}$$

Função Estática 2:

$$\begin{gathered}
M_{FE2} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{onde} \\
X = Y = \mathbb{R} \times \mathbb{N}; S = \mathbb{R}^3 \times \mathbb{R}_0^+ \\
\delta_{int}(s) = \delta_{int}(u_r, u_c, q_1, \sigma) = (u_r, u_c, q_1, \infty) \\
\delta_{ext}(s, e, (x_v, port)) = \delta_{ext}(u_r, u_c, q_1, \sigma, e, (x_v, port)) = \begin{Bmatrix} (x_v, u_c, q_1, 0) & \text{si port} = 0 \\ (u_r, x_v, q_1, 0) & \text{si port} = 1 \\ (u_r, u_c, x_v, 0) & \text{si port} = 2 \end{Bmatrix} \\
\lambda(s) = \lambda(u_r, u_c, q_1, \sigma) = (y, port) = \left(q_1 + (u_r - u_c) * \left(k_i - k_p + k_d\right), 0\right) \\
ta(s) = ta(u_r, u_c, q_1, \sigma) = \sigma
\end{gathered} \tag{5.14}$$

Função Estática 3:

$$
\begin{gathered}
M_{FE3} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{donde} \\
X = Y = \mathbb{R} \times \mathbb{N}; S = \mathbb{R}^3 \times \mathbb{R}_0^+ \\
\delta_{int}(s) = \delta_{int}(u_r, u_c, q_1, \sigma) = (u_r, u_c, q_1, \infty) \\
\delta_{ext}(s, e, (x_v, port)) = \delta_{ext}(u_r, u_c, q_1, \sigma, e, (x_v, port)) = \begin{Bmatrix} (x_v, u_c, q_1, 0) & \text{si port} = 0 \\ (u_r, x_v, q_1, 0) & \text{si port} = 1 \\ (u_r, u_c, x_v, 0) & \text{si port} = 2 \end{Bmatrix} \\
\lambda(s) = \lambda(u_r, u_c, q_1, \sigma) = (y, port) = \begin{Bmatrix} (q_1 + (u_r - u_c) * k_d, 0) \\ (Npasos + 1{,}1) \end{Bmatrix} \\
ta(s) = ta(u_r, u_c, q_1, \sigma) = \sigma
\end{gathered}
\tag{5.15}
$$

Essas três funções representam as operações matemáticas das Equações de estado, utilizando os valores kp, ki e kd obtidos para o controlador PID (25, 5 e 0.2 respectivamente), com uma saída adicional referente a Função Estática 3, utilizada para contar o número de passos realizados pelo controlador. A Figura 5.12 apresenta o diagrama de blocos referente ao controlador em tempo contínuo e em espaço de estados, utilizado para a comparação dos resultados da simulação.

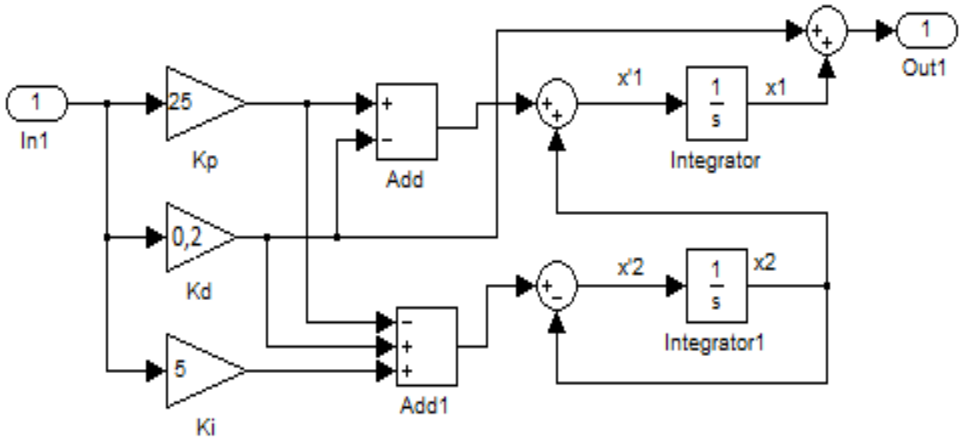

Figura 5.12 Diagrama de Blocos do Controlador Contínuo

5.6 Comparação do Controlador CDTC com o Controlador QSC

A simulação do sistema em estudo utilizando os dois tipos de controle é apresentada na Figura 5.13, onde foram colocados blocos de visualização, com o propósito de contar o número de passos (ações de controle) dos dois controladores. O valor do quantum para as variáveis de

estado do controlador QSC foi de 0,1 e o tempo de amostragem para o controlador de tempo discreto é de 0.001s. A simulação se realiza em um período de 0,1s.

O diagrama de blocos referente à Figura 5.13 pode ser dividido em duas partes principais: a parte superior que contém um controlador contínuo, utilizado como referencia para comparação com os resultados obtidos com o controlador QSC que se encontra na parte inferior da figura. Foram utilizados blocos de saturação que limitam o sinal de controle entre 12 e -12, simulando as limitações de energia que se tem num controlador real.

Neste diagrama o bloco PID_Discreto, igual que o bloco PID_QSS, que contém um elemento gerador de eventos para o controlador, isto porque ao modelar o controlador digital baseado na discretização temporal como um sistema de eventos discretos, deve-se garantir que a entrada seja seccionalmente constante. Por isso, um bloco conversor A/D síncrono, gera eventos de entrada correspondente à discretização temporal da saída da planta, com o mesmo período de amostragem do controlador.

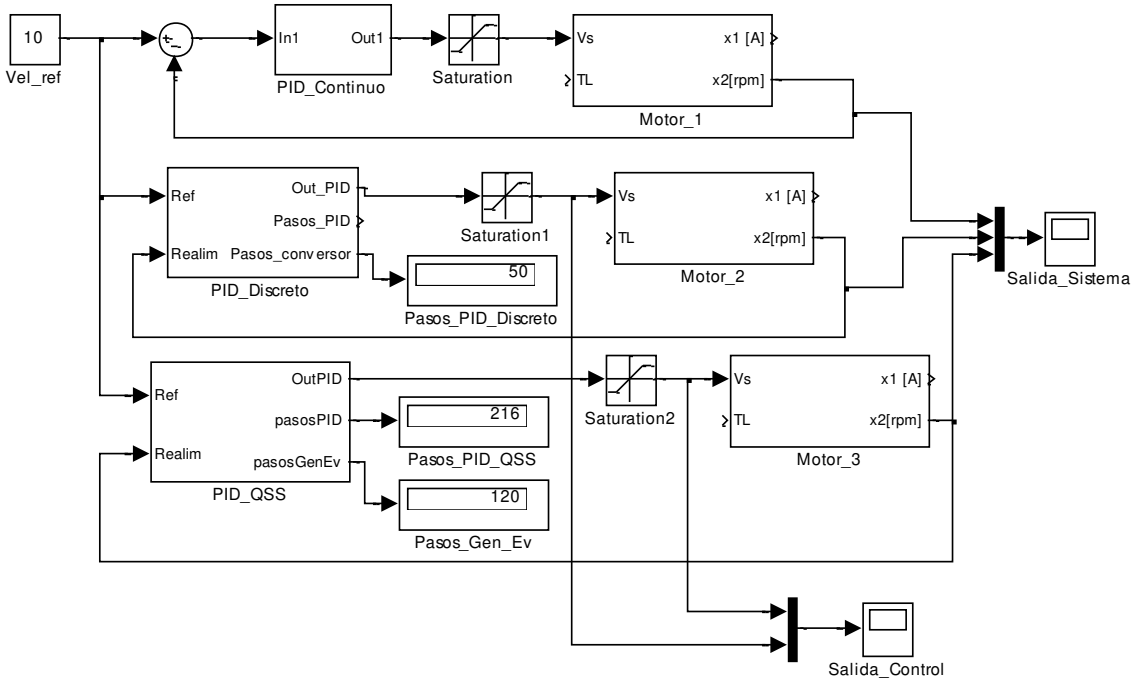

Figura 5.13 Simulação do Controlador Quantificado e Contínuo.

A Figura 5.14 mostra a saída do sistema, onde a linha 2 é a saída com o controlador QSC e a linha 1 é a saída utilizando o controlador CDTC.

Pode-se observar claramente que o controlador discreto no tempo faz com que a saída do sistema apresente um sobressinal de 18%, enquanto que o controlador de estados quantificados mantém sobressinal em 10,2%, que está mais próximo dos resultados obtidos para um controlador contínuo no tempo.

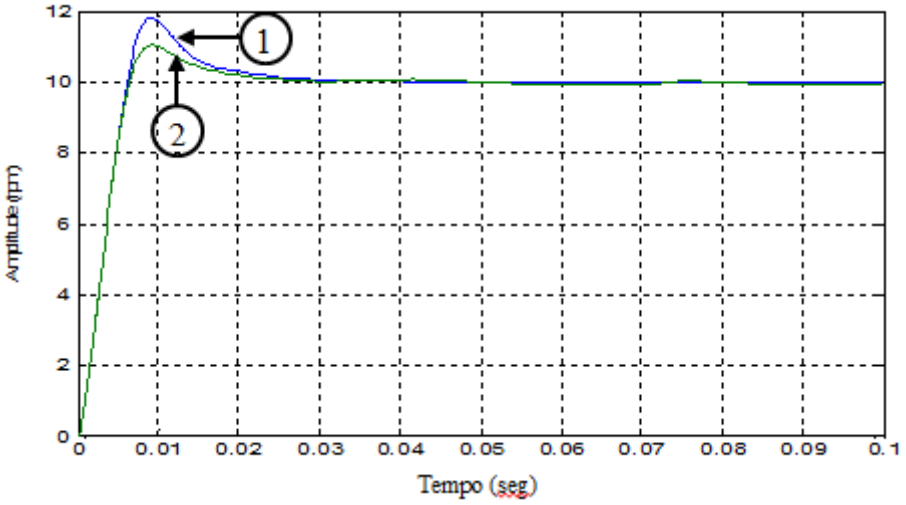

Figura 5.14 Saída da Planta com o Controlador CDTC e QSC

O controlador QSC tem duas saídas adicionais, uma chamada *pasosGenEv* que serve para visualizar o número de passos realizados pelo conversor A/D assíncrono (representado pelo bloco *Gen_Eventos* na Figura 5.13) e outra chamado *pasosPID*, mostrado na Figura 5.15, onde é contado o número de ações de controle.

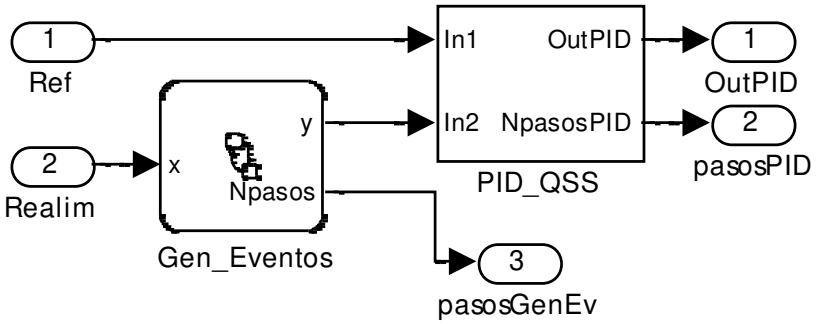

Figura 5.15 Divisão interna do bloco principal PID_QSS.

A estrutura interna do bloco gerador de eventos é apresentada na Figura 5.16, onde se tem dois estados, um para inicializar variáveis utilizadas, e outro para realizar a comparação da entrada e a saída atual do bloco, em cada transição.

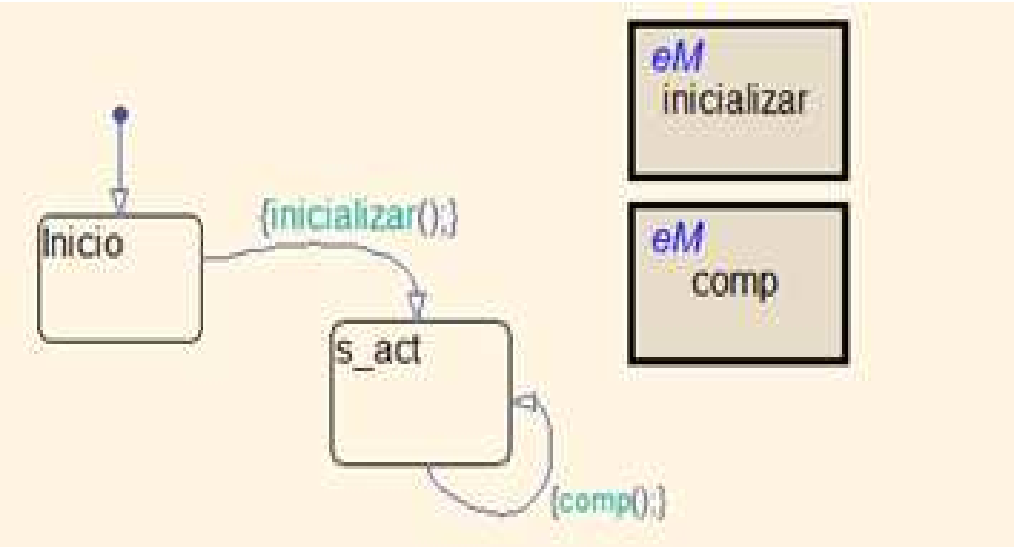

Figura 5.16 Diagrama de Estados do Bloco Gen_Eventos

Quando o valor de sua diferença supera o intervalo de quantificação, fixado em 0,05 RPM, a saída é atualizada como "y + 0,1", caso a diferença seja positiva, e "y – 0,1" caso seja negativa. Ou seja, de acordo com as Equações apresentadas anteriormente no sistema de Equações 3.19, o valor da variável Δyp é 0,1, já que para efeitos da análise do controlador, este valor representa a quantificação da saída da planta através do conversor A/D assíncrono.

5.7 Quantificação do Controlador

Considerando-se que a entrada para o controlador seja definida pelo erro do sistema em malha fechada (e = ur – uc), que representa a diferença entre o valor de referência e da variável, as Equações de estado do controlador podem ser expressas de acordo com a Equação (5.16).

$$\begin{Bmatrix} \dot{x}_1 = x_2 + (k_p - k_d) \cdot e \\ \dot{x}_2 = -x_2 + (k_i - k_p + k_d) \cdot e \\ y_c = x_1 + k_d \cdot e \end{Bmatrix} \rightarrow \begin{Bmatrix} \dot{x}_1 = x_2 + 248 \cdot e \\ \dot{x}_2 = -x_2 - 198 \cdot e \\ y_c = x_1 + 2 \cdot e \end{Bmatrix} \tag{5.16}$$

$$\rightarrow \begin{Bmatrix} \dot{x}_1 = x_2 + 248 \cdot u_r - 248 \cdot u_c \\ \dot{x}_2 = -x_2 - 198 \cdot u_r + 198 \cdot u_c \\ y_c = x_1 + 2 \cdot u_r - 2 \cdot u_c \end{Bmatrix}$$

As Equações de estado do sistema QSC devem considerar os valores de quantificação das variáveis de estado, onde para este caso foi Δxc1 = Δxc2 = 0,1. Consequentemente, o controlador quantificado será descrito através do sistema de Equações (5.17).

$$\begin{Bmatrix} \dot{x}_c = A_c \cdot (x_c + \Delta x_c) + B_c \cdot u_c + B_r \cdot u_r \\ y_c(t) = C_c \cdot (x_c + \Delta x_c) + D_c \cdot u_c + D_r \cdot u_r \end{Bmatrix}$$

$$\begin{Bmatrix} \dot{x}_1 = (x_{c2} + 0{,}1) + 248 \cdot u_r - 248 \cdot u_c \\ \dot{x}_2 = -(x_{c2} + 0{,}1) - 198 \cdot u_r + 198 \cdot u_c \\ y_c = (x_{c1} + 0{,}1) + 2 \cdot u_r - 2 \cdot u_c \end{Bmatrix} \tag{5.17}$$

O valor de quantificação para as duas variáveis de estado do controlador (Δxc1 e Δxc2) é de 0,1 e como a saída do controlador se conecta diretamente com a entrada da planta, o valor de Δyc também é 0,1. Utilizando esses valores junto com as matrizes de estado da planta e do controlador, pode-se analisar o erro do sistema de controle quantificado em relação ao sistema contínuo.

Como foi apresentada anteriormente, a Equação 3.18 permite determinar o erro das diferentes variáveis de estado do sistema quantificado, em relação às variáveis do sistema contínuo, pelo qual se utilizam as diferentes matrizes que compõem a planta, o controlador e o sistema completo em malha fechada.

Sem levar em conta a variável x3, pois a mesma não define o comportamento do sistema e sem considerar a entrada de torque de carga TL, já que não se encontra conectada com o controlador, o modelo contínuo da planta é descrito pela Equação 5.18.

$$\dot{\bar{x}}(t) = \begin{bmatrix} -1235{,}29 & -22{,}74 \\ 5523{,}81 & -0{,}62 \end{bmatrix} \bar{x}(t) + \begin{bmatrix} 980{,}39 \\ 0 \end{bmatrix} V_S(t) \tag{5.18}$$

$$y(t) = \begin{bmatrix} 0 & 0{,}0437 \end{bmatrix} \bar{x}(t)$$

onde a saída do sistema está multiplicada pelo fator de conversão de rad/s a RPM, sendo definidas através da Equação 5.19(5.19).

$$\begin{aligned} \boldsymbol{A_p} &= \begin{bmatrix} -1235{,}29 & -22{,}74 \\ 5523{,}81 & -0{,}62 \end{bmatrix} \\ \boldsymbol{B_p} &= \begin{bmatrix} 980{,}39 \\ 0 \end{bmatrix} \\ \boldsymbol{C_p} &= \begin{bmatrix} 0 & 0{,}0437 \end{bmatrix} \end{aligned} \tag{5.19}$$

As matrizes do controlador são descritas através da Equação 5.20.

$$\begin{gathered} A_c = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \; ; \; B_r = \begin{bmatrix} 248 \\ -198 \end{bmatrix} \; ; \; B_c = \begin{bmatrix} -248 \\ 198 \end{bmatrix} \\ C_c = \begin{bmatrix} 1 & 0 \end{bmatrix} ; \; D_r = 2 \; ; \; D_c = -2 \end{gathered} \tag{5.20}$$

A Equação do sistema em malha fechada é definida pela Equação 5.21.

$$\dot{x} = A(x + \Delta x) + F\Delta y + B(u_r) \tag{5.21}$$

Onde,

$$x = \begin{bmatrix} x_p \\ x_c \end{bmatrix}, A = \begin{bmatrix} A_p + B_p D_c C_p & B_p C_c \\ B_c C_p & A_c \end{bmatrix}, B = \begin{bmatrix} B_p D_r \\ B_r \end{bmatrix}$$

$$\Delta x = \begin{bmatrix} \Delta x_p \\ \Delta x_c \end{bmatrix}, \Delta y = \begin{bmatrix} \Delta y_p \\ \Delta y_c \end{bmatrix}, F = \begin{bmatrix} B_p D_c & B_p \\ B_c & 0 \end{bmatrix}$$

Como a planta não se quantifica já que se utiliza seu modelo contínuo, o valor de Δxp é zero e seu resultado se expressa no sistemas de Equação 5.22.

$$\Delta x = \begin{bmatrix} 0 \\ \Delta x_c \end{bmatrix}, \Delta y = \begin{bmatrix} \Delta y_p \\ \Delta y_c \end{bmatrix}, F = \begin{bmatrix} B_p D_c & B_p \\ B_c & 0 \end{bmatrix} \tag{5.22}$$

Considerando-se que o valor de Δyp é 0,1 e Δyc é 0,1, obtém-se o sistema representado pela Equação (5.23).

$$x = \begin{bmatrix} x_{p1} \\ x_{p2} \\ x_{c1} \\ x_{c2} \end{bmatrix}; A = \begin{bmatrix} -1235,3 & -108,4 & 980,4 & 0 \\ 5523,8 & -0,6 & 0 & 0 \\ 0 & -10,8 & 0 & 1 \\ 0 & 8,7 & 0 & -1 \end{bmatrix}; B = \begin{bmatrix} 1960,8 \\ 0 \\ 248 \\ -198 \end{bmatrix}$$

$$\Delta x = \begin{bmatrix} 0 \\ 0 \\ 0,1 \\ 0,1 \end{bmatrix}; \Delta y = \begin{bmatrix} 0,1 \\ 0,1 \end{bmatrix}; F = \begin{bmatrix} -1960,8 & 980,4 \\ 0 & 0 \\ -248 & 0 \\ 198 & 0 \end{bmatrix} \tag{5.23}$$

5.8 Análise de Erro do Sistema

A partir da matriz A e utilizando Matlab®, obtém-se a matriz de vetores e a matriz diagonal de valores próprios, de acordo com [V, Λ]=eig(A), onde o resultado obtido é apresentado através da Equação 5.24.

$$V = \begin{bmatrix} 0,0994 - 0,0698i & -0,0994 + 0,0698i & -0,0232 & 0 \\ -0,9924 & -0,9924 & -0,994 & 0,0921 \\ -0,0130 - 0,0092i & -0,0130 + 0,0092i & -0,0832 & 0,0102 \\ 0,0104 + 0,0073i & 0,0104 - 0,0073i & 0,0670 & 0,9957 \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} -553,68 + 388,43i & 0 & 0 & 0 \\ 0 & -553,68 - 388,43i & 0 & 0 \\ 0 & 0 & -129,35 & 0 \\ 0 & 0 & 0 & -0,2 \end{bmatrix} \tag{5.24}$$

A partir dos dados obtidos é avaliado o erro a partir da Equação 5.25(5.25).

$$| \tilde{x}(t) - x(t) | \leq | V | (| \mathbb{R}e(\Lambda)^{-1}\Lambda || V^{-1} | \Delta q_x + | \mathbb{R}e(\Lambda)^{-1}V^{-1}F | \Delta q_y) \quad (5.25)$$

Obtendo os resultados apresentados na Equação 5.26.

$$| \tilde{x}(t) - x(t) | \leq \begin{bmatrix} 1{,}4228 \\ 14{,}3932 \\ 0{,}4643 \\ 0{,}6688 \end{bmatrix} \quad (5.26)$$

A partir da análise destes resultados pode-se constatar que a diferença máxima apresentada entre a saída da planta com controle contínuo e com controle quantificado (que é a variável de estado da planta xc2) é de ±14,3932 rad/s., ou seja, o erro entre as duas trajetórias de saída está limitada por 14,3932 rad/s, que equivale a 0,63 RPM. O primeiro valor do vetor resultante indica o erro máximo em relação à variável de estado da planta xp1 e os dois últimos valores indicam o erro máximo para as variáveis de estado do controlador (xc1 e xc2), usando controle contínuo e controle de estados quantificados.

5.9 Simulação do Sistema para o Cálculo do Erro

Uma vez realizado o cálculo teórico do erro, é realizado a simulação utilizando o diagrama de blocos da Figura 5.13. Considerando-se que em QSC o erro é um valor fixo, que não depende da entrada, o erro é expresso em unidades e não em porcentagem.

A Figura 5.17 mostra a resposta do sistema, onde saída do sistema de controle contínuo é representada pela linha 1 e a saída utilizando o controlador de estados quantificados é representado pela linha 2. Pode-se observar que a resposta com o controle quantificado está muito próxima à resposta com controle contínuo, já que a linha 2 segue a linha 1 durante todo o estado transitório.

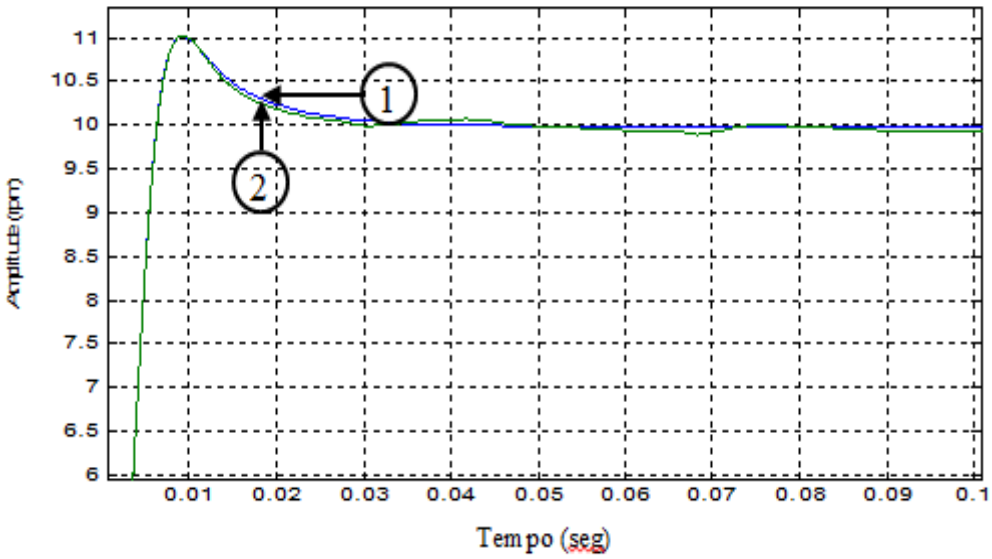

Figura 5.17 Saída do sistema sem controle, com CCS e QSC.

Na Figura 5.18 é mostrada o sinal resultante da diferença entre a saída da planta com controlador QSC e a saída com controlador CCS, para uma simulação de 0,2s. Pode-se constatar que a saída com controlador QSC oscila ao redor do valor estável com um erro máximo de 0,0874 RPM, inferior com o valor de erro calculado de 0,63 RPM.

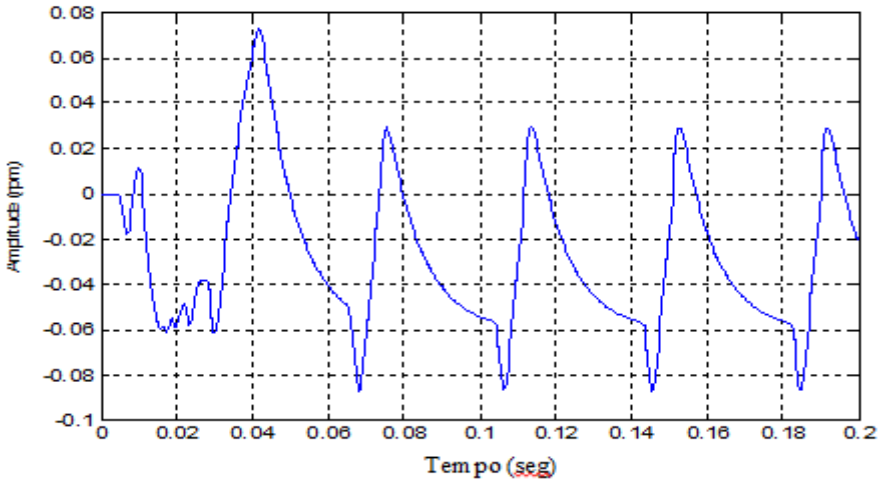

Figura 5.18 Erro de diferença entre a saída da planta com QSC e CCS.

A Figura 5.19 apresenta a saída do controlador QSC para todo o intervalo de tempo da simulação. Pode-se observar que o sinal se compõe de passos com um quantum (Δq) definido e

que ao final, a saída do controlador quantificado oscila entre valores de diferentes magnitudes, o qual explica a oscilação ao redor do estado estável na saída da planta (Figura 5.18).

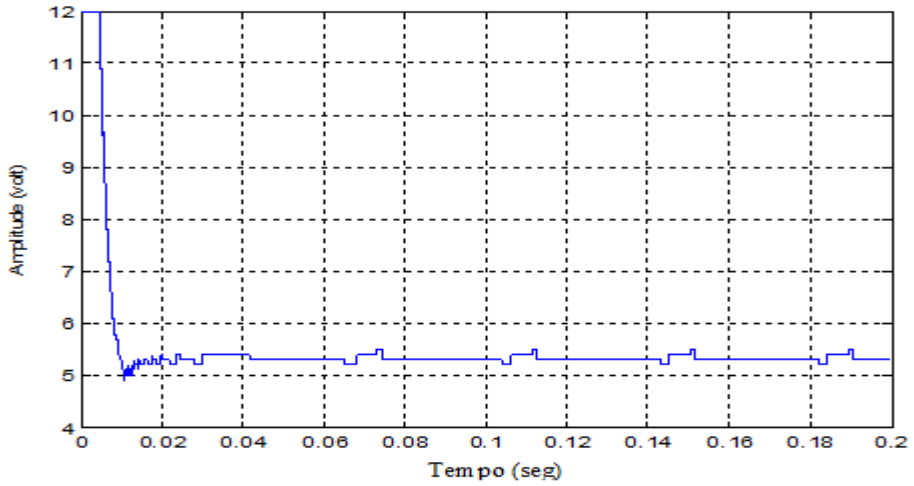

Figura 5.19 Saída do controlador QSC.

A Figura 5.20 mostra uma aproximação nos primeiros 0,02 segundos da simulação, onde pode-se constatar que o controlador quantificado realiza uma grande quantidade de passos no arranque, sem embargo, aproximadamente a partir de t=0,03s (Figura 5.19) os saltos se reduzem consideravelmente. Para os 0,2 segundos de simulação o controlador realiza 232 passos e o conversor A/D, que está representado por o bloco *Gen_Eventos* na Figura 5.15, realiza 128 passos.

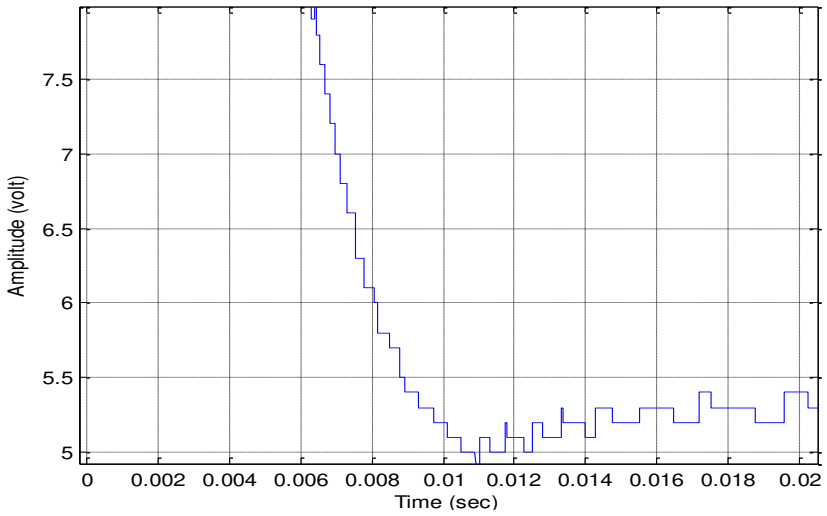

Figura 5.20 Saída do QSC.

5.10 Modelagem do Controlador de Tempo Discreto através de DEVS

Como ferramenta de análise comparativa dos resultados obtidos com o controlador QSC, é implementado um controlador digital convencional discreto no tempo (CDTCs). Assim, foi realizado o modelo do controlador em tempo discreto através do formalismo DEVS, implementando-se a seguir um novo diagrama de simulação utilizando Simulink™, para comparar o desempenho destes dois sistemas de controle.

Para modelar o controlador CDTC utilizando o formalismo DEVS, foi desenvolvido o modelo em variáveis de estado, e a partir do mesmo é gerado o diagrama de blocos apresentado anteriormente na (5.3) para o controlador quantificado. A maneira de converter esta estrutura o controlador digital baseado no tempo, é a partir da troca do integrador QSS por um integrador em tempo discreto, que calcula o valor da saída da variável de estado com intervalos de tempo periódicos. De modo similar, os modelos atômicos das funções estáticas também devem gerar uma saída periódica. A estrutura DEVS do integrador digital é mostrado na Equação 5.27.

$$\begin{gathered}
M_{DT} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{donde} \\
X = Y = \mathbb{R}; S = \mathbb{R}^2 \times \mathbb{R}_0^+ \\
\delta_{int}(s) = \delta_{int}(x_1, dx_1, \sigma) = (x_1 + \sigma \cdot dx_1, dx_1, T) \\
\delta_{ext}(s, e, x_v) = \delta_{ext}(x_1, dx_1, \sigma, e, x_v) = (x_1 + e \cdot dx_1, x_v, \sigma - e) \\
\lambda(s) = \lambda(x_1, dx_1, \sigma) = (y) = (x_1 + \sigma \cdot dx_1) \\
ta(s) = ta(x_1, dx_1, \sigma) = \sigma
\end{gathered} \tag{5.27}$$

Nesta estrutura pode-se observar que as transições internas, que são as que geram eventos de saída, se realizam com um intervalo de tempo periódico T, trocando o valor da saída mediante a função λ(s), segundo o valor da derivada da variável de estado. Além disso, todas as vezes que se gera uma transição externa, os valores da variável de estado (x1) e a derivada (dx1) são atualizados, calculando-se o tempo da próxima transição interna, através da diferença entre o período de amostragem (σ=T) e o tempo transcorrido (e).

5.11 Funções do Controlador CDTC em QSS

De maneira análoga ao modelo quantificado por QSS, as três funções estáticas representam as operações matemáticas das Equações de estado, utilizando os valores do controlador contínuo para kp, ki e kd (25, 5 e 0.2 respectivamente), com uma saída adicional na Função Estática 3 para contar o número de passos realizados pelo controlador. A estrutura para as funções estáticas de tempo discreto, modeladas a partir de DEVS, são apresentadas pelas Equações 5.28, 5.29 e 5.30.

Função Estática 1:

$$\begin{gathered}
M_{FE1} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{onde} \\
X = Y = \mathbb{R} \times \mathbb{N}; S = \mathbb{R}^3 \times \mathbb{R}_0^+ \\
\delta_{int}(s) = \delta_{int}(u_r, u_c, x_1, \sigma) = (u_r, u_c, x_1, T) \\
\delta_{ext}(s, e, (x_v, port)) = \delta_{ext}(u_r, u_c, x_1, \sigma, e, (x_v, port)) = \begin{cases} (x_v, u_c, x_1, \sigma - e) & \text{si port} = \\ (u_r, x_v, x_1, \sigma - e) & \text{si port} = \\ (u_r, u_c, x_v, \sigma - e) & \text{si port} = \end{cases} \\
\lambda(s) = \lambda(u_r, u_c, x_1, \sigma) = (y, port) = \left(x_1 + (u_r - u_c) * \left(k_p - k_d\right), 0\right) \\
ta(s) = ta(u_r, u_c, x_1, \sigma) = \sigma
\end{gathered} \tag{5.28}$$

Função Estática 2:

$$\begin{gathered}
M_{FE2} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{onde} \\
X = Y = \mathbb{R} \times \mathbb{N}; S = \mathbb{R}^3 \times \mathbb{R}_0^+ \\
\delta_{int}(s) = \delta_{int}(u_r, u_c, x_2, \sigma) = (u_r, u_c, x_2, T) \\
\delta_{ext}(s, e, (x_v, port)) = \delta_{ext}(u_r, u_c, x_2, \sigma, e, (x_v, port)) = \begin{cases} (x_v, u_c, x_2, \sigma - e) & \text{si port} = \\ (u_r, x_v, x_2, \sigma - e) & \text{si port} = \\ (u_r, u_c, x_v, \sigma - e) & \text{si port} = \end{cases} \\
\lambda(s) = \lambda(u_r, u_c, x_2, \sigma) = (y, port) = \left(x_2 + (u_r - u_c) * \left(k_i - k_p + k_d\right), 0\right) \\
ta(s) = ta(u_r, u_c, x_2, \sigma) = \sigma
\end{gathered} \tag{5.29}$$

Função Estática 3:

$$
\begin{gathered}
M_{FE3} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{onde} \\
X = Y = \mathbb{R} \times \mathbb{N}; S = \mathbb{R}^3 \times \mathbb{R}_0^+ \\
\delta_{int}(s) = \delta_{int}(u_r, u_c, x_3, \sigma) = (u_r, u_c, x_3, T) \\
\delta_{ext}(s, e, (x_v, port)) = \delta_{ext}(u_r, u_c, x_3, \sigma, e, (x_v, port)) = \begin{cases} (x_v, u_c, x_3, \sigma - e) & \text{si port} = \\ (u_r, x_v, x_3, \sigma - e) & \text{si port} = \\ (u_r, u_c, x_v, \sigma - e) & \text{si port} = \end{cases} \\
\lambda(s) = \lambda(u_r, u_c, x_3, \sigma) = (y, port) = \begin{Bmatrix} (x_3 + (u_r - u_c) * k_d, 0) \\ (Npasos + 1,1) \end{Bmatrix} \\
ta(s) = ta(u_r, u_c, x_3, \sigma) = \sigma
\end{gathered}
\tag{5.30}
$$

5.12 Simulação do Sistema Controlador-Esteira Modelado em QSS

Antes de realizar o modelo e a simulação de toda a planta híbrida, se apresenta a simulação do sistema controlador-esteira quantificado utilizando QSS, como uma aproximação ao modelo e simulação de sistemas dinâmicos contínuos modelados por eventos discretos e conformados por subsistemas de eventos discretos acoplados. Neste caso, os subsistemas são o modelo do controlador QSC e o modelo da planta quantificada por QSS.

A Figura 5.21 apresenta o diagrama de blocos correspondente à simulação. Na parte superior tem-se o sistema contínuo Controlador-Planta e na parte inferior encontra-se o sistema QSS.

Ao mesmo tempo se tem elementos de visualização para contar os passos do conversor A/D assíncrona e do controlador QSC. Observa-se ainda que o conversor A/D assíncrono realiza 126 passos e o controlador executa 168 ações de controle. Além disso, o bloco *Motor_QSS* precisa 60 passos para simular a saída do sistema controlado, evidenciando a vantagem da simulação baseada em eventos discretos, já que para obter um resultado satisfatório necessita menos recursos computacionais que a simulação baseada em tempo discreto, com uma cota de erro limitado em relação à representação continua. A Figura 5.22 apresenta a saída do sistema com controlador contínuo e quantificado

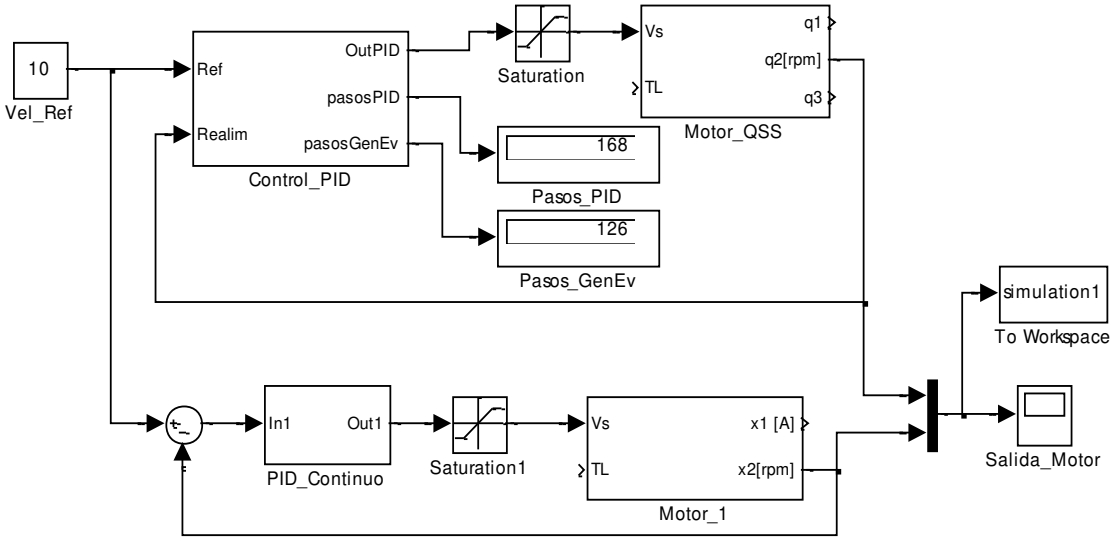

Figura 5.21 Simulação do Sistema Controlado em Representação QSS.

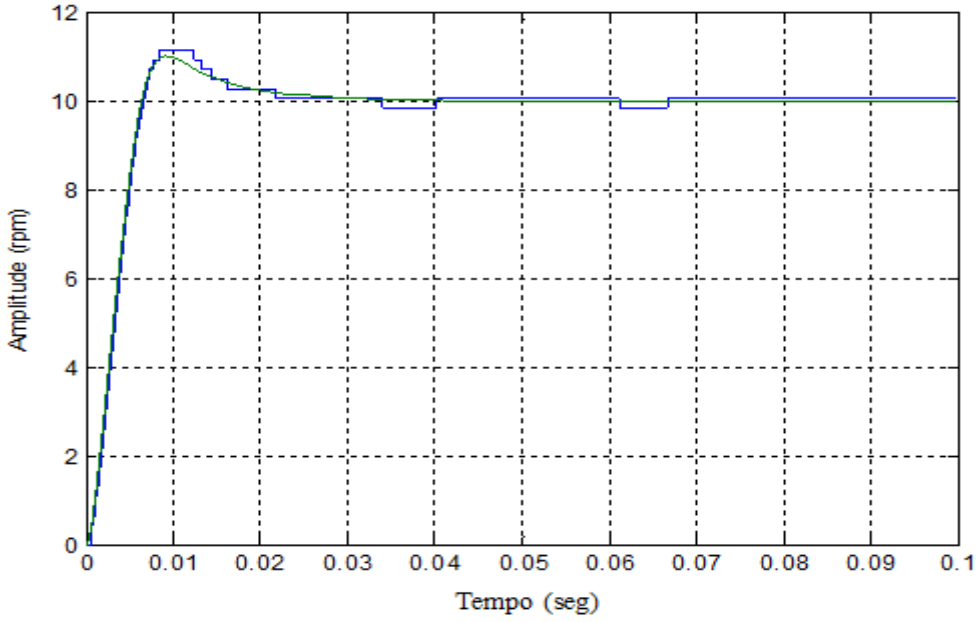

Figura 5.22 Saída do Sistema com Controlador Contínuo e Quantificado.

Na Figura 5.22 a linha verde representa a saída do sistema com controlador contínuo enquanto a linha azul representa a saída do sistema com controlador quantificado por QSS e representado em DEVS. Pode-se observar que o sistema QSS segue corretamente ao sistema contínuo, com um quantum de 0,2 RPM aproximadamente (5 rad/s). O gerador de eventos e o controlador trabalham com valores de quantificação de 0,1rpm.

Utilizando a Equação (3.20) para o cálculo de erro têm-se os resultados apresentados na Equação 5.31(5.31).

$$\Delta x = \begin{bmatrix} \Delta x_p \\ \Delta x_c \end{bmatrix}, \Delta y = \begin{bmatrix} \Delta y_p \\ \Delta y_c \end{bmatrix}$$
$$\Delta x = \begin{bmatrix} 0{,}1 \\ 5 \\ 0{,}1 \\ 0{,}1 \end{bmatrix}; \Delta y = \begin{bmatrix} 0{,}1 \\ 0{,}1 \end{bmatrix} \tag{5.31}$$

O valor de quantificação da variável xp1 é 0,1A e o quantum da variável xp2 é de 5rad/s. O valor do erro obtido para cada variável de estado quantificada no sistema QSS é apresentado através da Equação 5.32.

$$| \tilde{x}(t) - x(t) | \leq \begin{bmatrix} 3{,}5002 \\ 33{,}2853 \\ 0{,}9286 \\ 1{,}0400 \end{bmatrix} \tag{5.32}$$

Pode-se constatar a partir deste resultado, que a diferença máxima entre a variável de saída do sistema quantificado e do sistema contínuo, está limitada a 33,2853rad/s, ou seja, 1,454 rpm. É importante notar que este valor é maior que o obtido a partir da Equação 5.25, já que neste caso se está considerando que a planta se encontre quantificada, a diferença da planta que foi utilizada para calcular o erro na Equação 5.25, a qual se realizou com um controlador quantificado e uma planta continua.

A Figura 5.23 mostra o gráfico do erro, onde pode-se observar que a diferença máxima na variável de saída é de 0,393 RPM, 27% do valor de erro calculado.

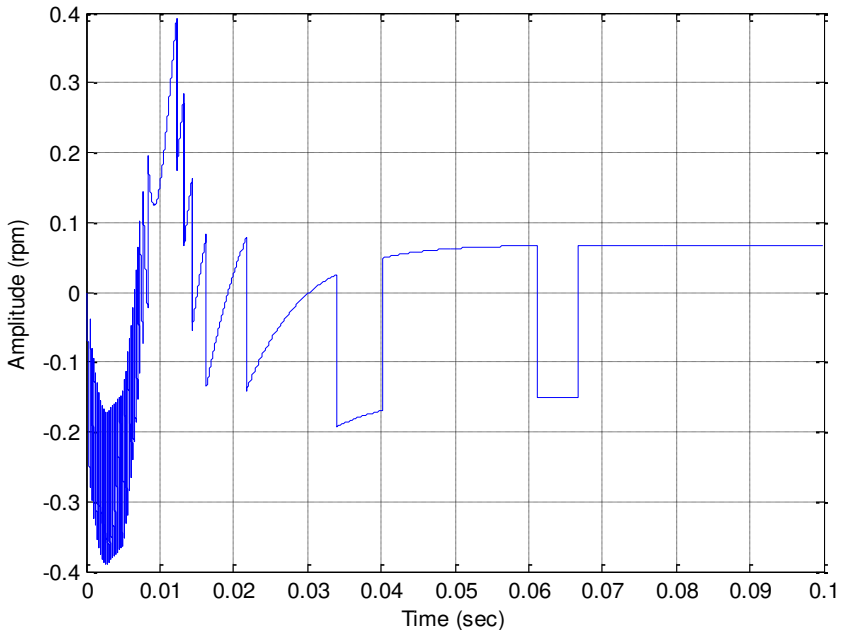

Figura 5.23 Diferença do sinal de Saída do Sistema Quantificado e o Sistema Contínuo

5.13 Modelagem em DEVS do Processo híbrido

O processo a ser modelado é uma esteira transportadora com um sensor de entrada e um sensor de saída. O sensor de entrada indica a chegada de novos produtos à esteira, que representa um torque devido à carga sobre a esteira. O sensor de saída indica os objetos que saem do processo e representa uma diminuição no torque de carga. A máxima quantidade de objetos permitida sobre a esteira é de três, e se simula uma esteira cujo seu comprimento equivale a 12 radianos.

Para modelagem do processo (esteira transportadora), são definidas três estruturas DEVS, a primeira para gerar os eventos de entrada quando um objeto chega até a esteira. A segunda estrutura simula o sensor de saída, quando a esteira deixa a placa de base na próxima unidade, que é montagem e desmontagem lateral, através do cálculo do tempo de chegada ao posto de acordo com a velocidade da esteira, a qual se vê afetada com cada objeto de entrada pelo torque de carga, e finalmente, a terceira estrutura corresponde à geração do torque de carga dependendo da quantidade de objetos que exista sobre a esteira.

A Figura 5.24 apresenta o modelo DEVS acoplado do sistema, composto por 5 blocos DEVS atômicos, um bloco que representa os eventos de entrada ao sistema gerados pelo sensor 1, um bloco que representa os eventos de saída do sistema gerados pelo sensor 2, o bloco gerador de torque que coloca um torque de carga a planta, dependendo da quantidade de objetos que se encontrem em processo, o bloco da esteira que por sua vez é composto por uma série de modelos atômicos DEVS para sua representação em QSS e o bloco do controlador, que igualmente se compõe de vários modelos DEVS que o representam em QSS como um controlador QSC.

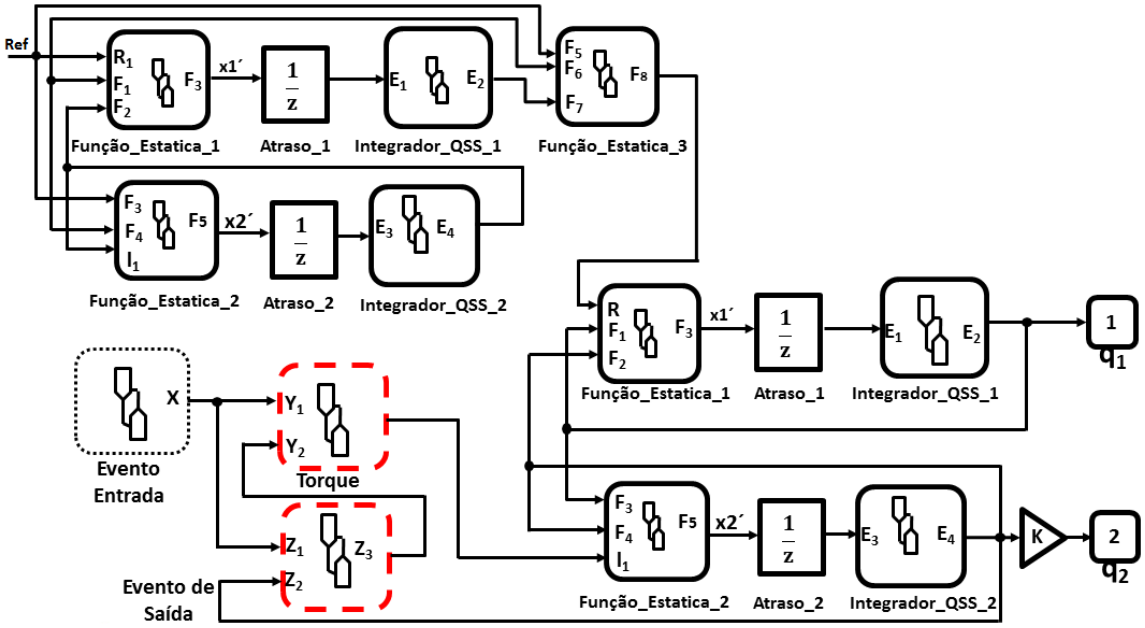

Figura 5.24 Modelo DEVS Acoplado do Sistema Híbrido

O bloco evento de entrada é um gerador de eventos, que simula os objetos que entram no processo e possuem uma única saída para isso. Essa saída se interconecta com o bloco evento de saída e com o bloco Torque. O primeiro calcula o tempo que cada objeto permanecerá na esteira de acordo à velocidade da mesma, apresentando por este motivo uma entrada que transporta eventos de velocidade desde a variável quantizada de velocidade q2. O bloco Torque calcula o torque de carga segundo a quantidade de objetos presentes na esteira, pelo que recebe eventos tanto de evento de entrada como de evento de saída. O bloco que representa o modelo da estira recebe eventos com ações de controle por parte do bloco Controlador através da função estática

3, e envia o esse último evento de velocidade. Em geral, o modelo DEVS acoplado tem uma única saída, que se interconecta com a saída do bloco evento de saída para representar a saída de objetos do processo.

A Figura 5.25 mostra o diagrama de blocos da simulação. Neste diagrama pode-se observar a conexão das três estruturas DEVS mencionadas, onde o bloco *DEVS_Sensor2* gera eventos que dependem dos eventos gerados por o bloco *DEVS_Sensor1* e da velocidade da esteira, o bloco *DEVS_Sensor1* é um gerador de eventos, sem entradas e o bloco *DEVS_TL* gera um torque de carga segundo os eventos de entrada e saída do processo. Além disso, foram implementados blocos de visualização para a saída do sistema, para os sinais de cada sensor, o torque de carga e os contadores de passos do controlador.

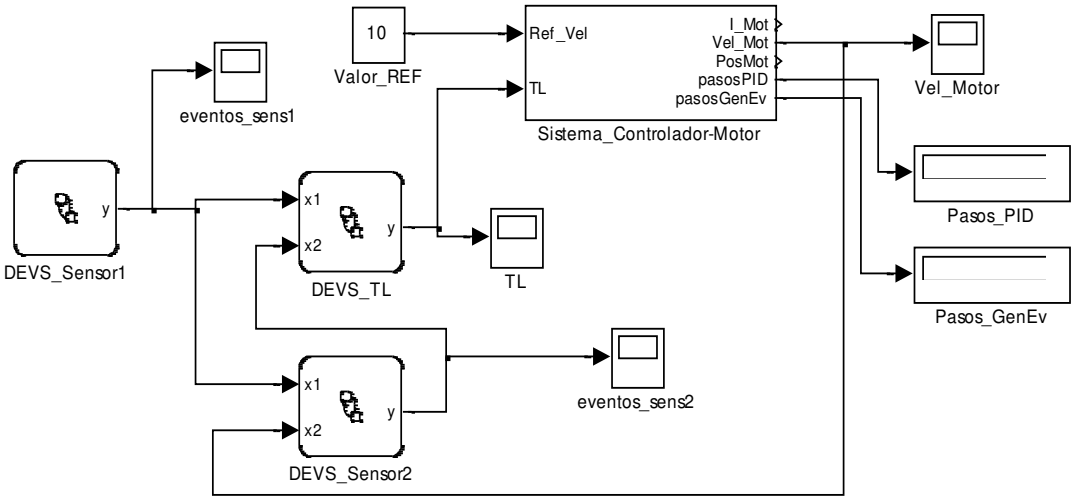

Figura 5.25 Diagrama de Blocos da Processo Híbrido.

A estrutura DEVS do sensor de entrada é um gerador de eventos que permite um número máximo de três objetos, como se descreve na Equação 5.33.

$$
\begin{gathered}
M_{S1} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{onde} \\
X = Y = \mathbb{R}; S = \mathbb{R}^1 \times \mathbb{R}_0^+ \\
\delta_{int}(s) = \delta_{int}(s_1, \sigma) = (s_1 + 1, T) \\
\lambda(s) = \lambda(s_1, \sigma) = (y) = \{(s_1 + 1) \quad \text{si} \quad y < 3\} \\
ta(s) = ta(x_1, dx_1, \sigma) = \sigma
\end{gathered}
\tag{5.33}
$$

Esta estrutura não requer para fins de simulação a função de transição externa, onde o período de amostragem T utilizado é de 0,02s. A estrutura correspondente ao gerador de torque de carga é mostrada na Equação 5.34.

$$\begin{gathered}
M_{TL} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{onde} \\
X = Y = R; S = R^3 \times R_0^+ \\
\delta_{int}(s) = \delta_{int}(in, out, T_L, \sigma) = (in, out, T_L, \infty) \\
\delta_{ext}(s, e, x_v) = \delta_{ext}(in, out, T_L, \sigma, e, x_v) = \begin{Bmatrix} (in + 1, out, T_L, 0) & \text{si port} = 0 \\ (in, out + 1, T_L, 0) & \text{si port} = 1 \end{Bmatrix} \\
\lambda(s) = \lambda(in, out, T_L, \sigma) = (y) = \left(T_L \cdot (in - out)\right) \\
ta(s) = ta(in, out, T_L, \sigma) = \sigma
\end{gathered} \tag{5.34}$$

Como pode ser observada, a saída do sistema se atualiza com a diferença entre a quantidade de objetos de entradas e saídas, multiplicadas pelo valor do torque de carga. A terceira estrutura corresponde ao sensor de saída, como mostra a Equação 5.35.

Nesta estrutura tem-se uma variável chamada est, que representa o estado atual do sistema e toma valores entre 0 e 3, dependendo do número de objetos sobre a esteira. Quando ingressa um novo objeto ativa-se a função de transição externa, que troca o estado atual e calcula novos valores para as variáveis *sig1, sig2, sig3* e *vel*, devido ao cálculo do tempo de cada objeto novo, considerando a velocidade da esteira, que ingressa pela porta1 da estrutura (admite-se no máximo três objetos).

Quando finaliza o tempo de saída do primeiro objeto na fila de objetos presentes sobre a esteira, é gerada uma transição interna que muda o valor do estado atual e gera um evento de saída através da função $\lambda(s)$, incrementando em um a quantidade de elementos que saem do sistema.

A Figura 5.26 mostra a saída do sistema híbrido e a Figura 5.27 apresenta a saída do gerador de torque de carga, onde se utilizou um quantum pela variável de saída de aproximadamente 0,2RPM (5rad/s). O quantum das variáveis de estado do controlador QSC é de 0,1 e geram-se eventos de entrada cada 0,02s.

Pode-se observar que para cada novo objeto que ingressa (Figura 5.27) o torque de carga se incrementa gerando uma perturbação no sistema (Figura 5.26), que o controlador tenta corrigir para voltar à velocidade de 10rpm.

$$
\begin{gathered}
M_{S2} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{onde} \\
X = Y = R; S = R^6 \times R_0^+ \\
\delta_{int}(s) = \delta_{int}(est, sig1, sig2, sig3, vel, dist, \sigma) = \\
\left\{ \begin{array}{c} (0, sig1, sig2, sig3, vel, dist, \infty) \quad \text{se} \quad est = 1 \\ (1, sig2 - e, sig2, sig3, vel, dist, sig2 - e) \quad \text{se} \quad est = 2 \\ (2, sig2 - e, sig3 - e, sig3, vel, dist, sig2 - e) \quad \text{se} \quad est = 3 \end{array} \right\} \\
\delta_{ext}(s, e, x_v) = \delta_{ext}(est, sig1, sig2, sig3, \sigma, e, x_v) = \\
\left\{ \begin{array}{c} \left(1, \frac{dist}{vel}, sig2, sig3, vel, dist, \infty\right) \quad \text{se port} = 0 \ \text{e} \ est = 0 \\ \left(2, sig1 - e, \frac{dist}{vel}, sig3, vel, dist, sig1 - e\right) \quad \text{se port} = 0 \ \text{e} \ est = 1 \\ \left(3, sig1 - e, sig2 - e, \frac{dist}{vel}, vel, dist, sig1 - e\right) \quad \text{se port} = 0 \ \text{e} \ est = 2 \\ (est, sig1 - e, sig2 - e, sig3 - e, vel, dist, sig1 - e) \quad \text{se port} = 0 \ \text{e} \ est = 3 \\ (est, sig1 - e, sig2, sig3, x_v, dist, \infty) \quad \text{se port} = 1 \ \text{e} \ est = 0 \\ (est, sig1 - e, sig2, sig3, x_v, dist, sig1 - e) \quad \text{se port} = 1 \ \text{e} \ est \neq 0 \end{array} \right\} \\
\lambda(s) = \lambda(est, sig1, sig2, sig3, vel, dist, \sigma) = (y) = (y + 1) \\
ta(s) = ta(est, sig1, sig2, sig3, vel, dist, \sigma) = \sigma
\end{gathered}
\tag{5.35}
$$

Cada objeto que sai da esteira reduze o torque, gerando novamente uma perturbação que o controlador deve corrigir.

Isto representa a arquitetura híbrida do sistema, já que os eventos de entrada que simulam a ativação dos sensores na simulação de perturbações pela mudança na carga repercutem através do bloco gerador de torque de carga.

No período de simulação de 0,2s o modelo QSS do motor necessitou de 126 passos para representar a saída sob os eventos que foram simulados, o controlador realizou 430 ações de controle, e o conversor A/D assíncrono realizou 264 passos.

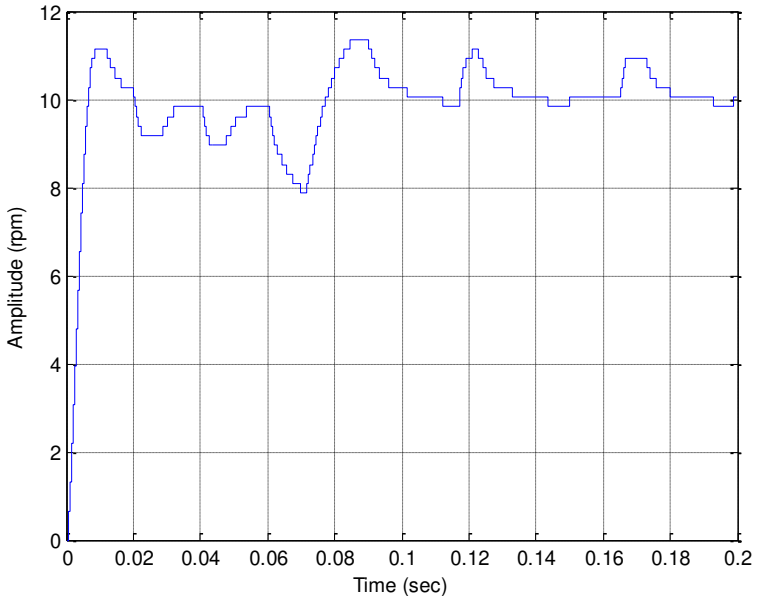

Figura 5.26 Saída do Sistema Híbrido

Pode-se ressaltar que se o número de objetos que entram na esteira transportadora a cada 2s fosse aumentado na simulação, a quantidade de passos necessários seria bastante similar, já que uma vez a planta alcança à estabilidade, o controlador para manter a saída limitada não realiza mais ações de controle. Isto representa uma importante vantagem de utilizar a modelagem e simulação baseada em eventos para sistemas de arquitetura híbrida, como foi apresentado nas seções anteriores deste trabalho.

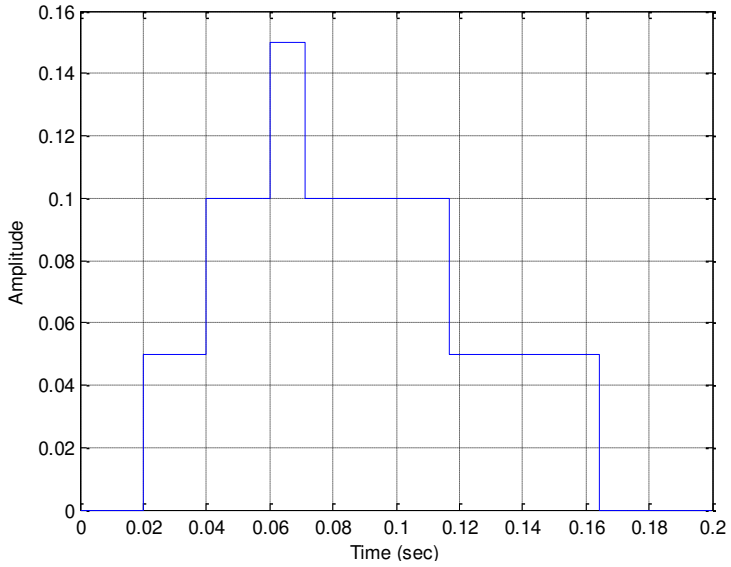

Figura 5.27 Saída do Bloco DEVS do Gerador de Torque de Carga.

É importante ressaltar que para obter um melhor resultado, referente ao erro em relação ao modelo contínuo, através do modelo de tempo discreto, deve-se reduzir o tempo de amostragem, o que acarreta em uma carga computacional maior e reduz a possibilidade de realizar uma implementação em hardware para emulação do comportamento de um sistema como o estudado neste trabalho.

5.14 Considerações Finais

Neste capítulo apresentou um estudo de caso de modelagem e controle de sistemas híbridos, considerando um processo de produção industrial, composto de uma esteira de transporte que leva produtos entre estações de trabalho.

O sistema foi modelado através do formalismo DEVS e o sistema de estados quantificados QSS, utilizando métodos de integração numérica para resolver as Equações diferenciais que representam o processo na parte regida pela ocorrência de eventos. O modelo foi comparado com a discretização através da amostragem temporal, conseguindo resultados muito próximos na resposta transiente e de estado estável.

Foi calculado o erro do sistema quantificado, sendo comparado com o erro do sistema cuja discretização foi realizada através da amostragem temporal, mostrando que o sistema que trabalha em estados quantificados, apresenta um erro menor.

No próximo capítulo desta tese de doutorado será apresentado um estudo de caso, que integra quatro modelos analisados nesta seção, com o propósito de garantir tempos de produção estáveis.

Capítulo 6

6. Controle de um Sistema de Produção de Arquitetura Híbrida

6.1 Introdução

Neste capítulo é apresentado um estudo de caso de modelagem e controle de um processo industrial de arquitetura híbrida, baseado nos formalismos DEVS e QSS. O objetivo deste caso de estudo é a modelagem e controle de um sistema de produção industrial apresentado na seção 4.1 deste trabalho. Considerando a estrutura da linha de montagem descrita foi realizada uma proposta para o controle de tempo de produção do sistema, com o intuito de garantir que esse tempo seja uniforme no desenvolvimento dos produtos.

Para este caso a linha de montagem é composta das unidades de carregamento (UC), montagem central (UMC), inspeção ou controle de qualidade (UCC) e armazenamento (UA). Essas unidades de produção foram modeladas como um sistema de dinâmica híbrido , no qual existe um sinal de eventos discretos que indica a terminação de seu trabalho e a entrega do produto para a próxima unidade de produção através de uma esteira de transporte independente que foi modelada como um sistema que trabalha em relação ao tempo em conjunto com a parte de eventos discretos, configurando assim, um sistema de produção com arquitetura de controle híbrido, como foi apresentado no capítulo 4 deste trabalho. Uma proposta de aplicação é mostrada na Figura 6.1.

A parte continua deste processo, realizada através do controle no tempo da esteira, é modelado através de variáveis de estado, para posteriormente realizar quantização do sistema utilizando QSS. Com o modelo em QSS, é realizado um mapeamento do sistema para representar o sistema continuo através de eventos utilizando modelos atômicos DEVS. A parte de eventos discretos, correspondente à ativação dos sensores a partir da entrada e saída de objetos, se modelando também com modelos atômicos DEVS. Posteriormente é realizada a simulação do sistema, utilizando os eventos de entrada e saída de objetos para gerar eventos no modelo da

esteira, que aumentem ou diminuem com a inércia de carga, evidenciando dessa maneira uma das principais vantagens na utilização da modelagem híbrida para simular a interação de eventos discretos num processo industrial, com os elementos de dinâmica continua que o compõem.

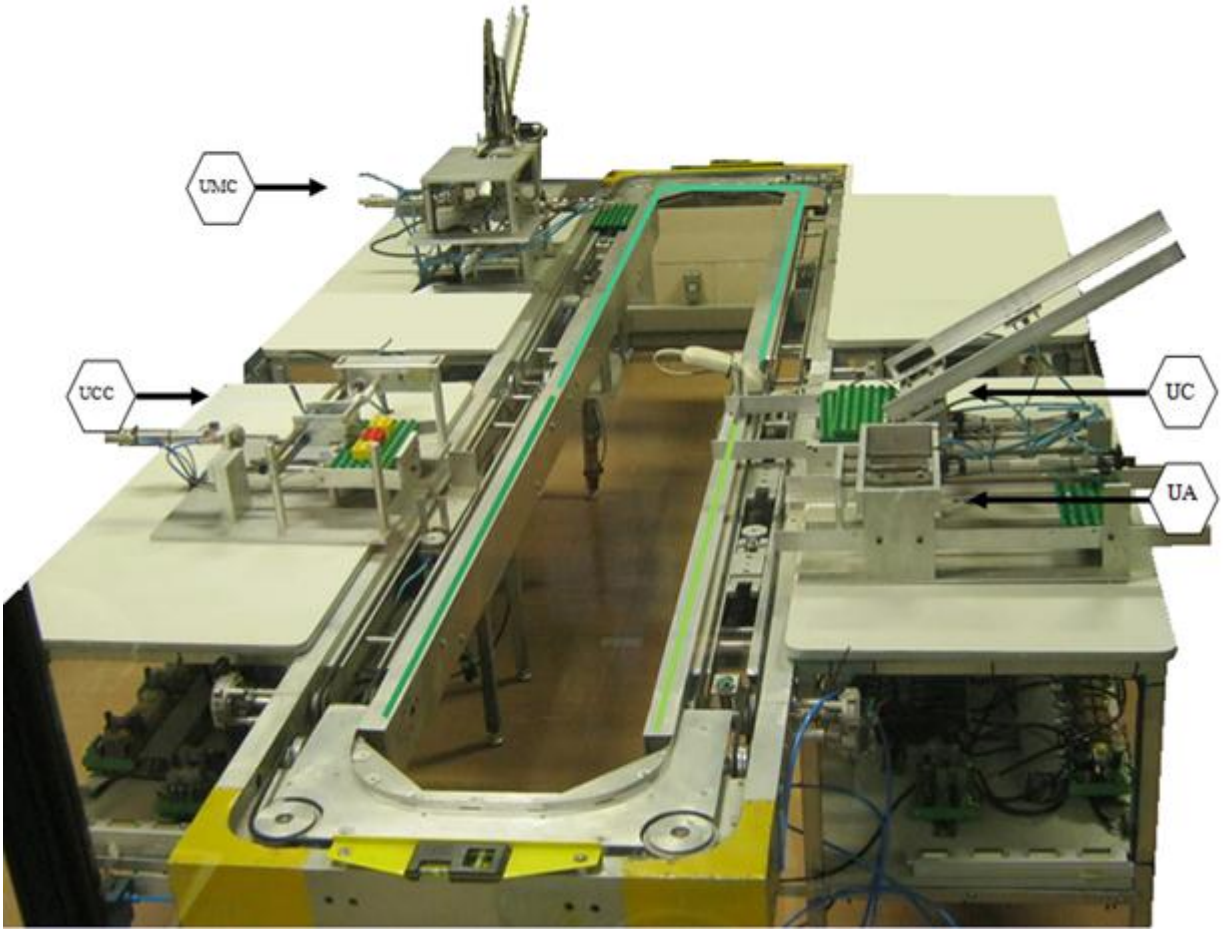

Figura 6.1 Modelo proposto para o caso em estudo.

6.2 Controle do Tempo de Produção de uma Linha de Montagem

O sistema possui quatro esteiras de transporte que possuem as mesmas características mecânicas e possuem igual comprimento. O tempo utilizado em cada unidade para realizar a atividade é aleatório.

No modelo proposto considera-se os eventos produzidos pelos produtos que chegam à esteira e os eventos que produzem os produtos ao sair da esteira. Esses eventos vão influir no comportamento dinâmico da esteira mudando sua velocidade, isto tem que ser corrigido com o intuito de garantir tempos de produção constantes na linha de montagem.

O objetivo do sistema de esteira transportadora é garantir um tempo total constante apesar das variações causadas por atrasos nos tempos de trabalho das unidades ou dos tempos de transporte de um produto. Com o propósito de garantir esse tempo, é realizada a escolha da variável controlada como referência de um controle de velocidade aplicado a cada uma das esteiras transportadoras, ou seja, velocidade da esteira pode ser variada para compensar os atrasos causados pelas unidades de produção anteriores.

6.3 Descrição do Funcionamento do Sistema

O processo começa quando se ativa uma sinal de inicio a partir do sistema de supervisão definida para essa tarefa. Nesse momento a unidade de produção 1, inicia sua atividade e ao terminá-la indica através da ativação do sinal S_i que existe um novo produto sobre a primeira esteira, para ser transportado até a unidade de produção seguinte. Nesse momento o controle da primeira esteira é ativado, e considerando o tempo de atraso anterior, o controlador define uma aceleração determinada. A partir do tamanho da esteira e o tempo estimado do processo, calcula-se um perfil de velocidade na forma trapezoidal. Esse perfil de velocidade é referência do controlador da esteira de transporte, onde cada esteira é movimentada por um motor DC, cuja análise foi realizada na sessão 4.3.1 do capítulo anterior deste trabalho.

Com o propósito de determinar o perfil de velocidade, foi realizado o cálculo da velocidade máxima do motor. Esse é o único parâmetro que pode ser ajustado de acordo com o comportamento das condições do processo.

Inicialmente é realizado o cálculo da velocidade média da esteira utilizando os parâmetros de tempo da esteira ($\mathbf{t_{tr(i)}}$), o tempo de atraso anterior ($\mathbf{t_{d(i-1)}}$) e o comprimento da mesma (L(i)).

Para a compensação dos atrasos que vai acontecendo no processo, utiliza-se o tempo adicional ($\mathbf{t_{b(i)}}$) dos processos anteriores, de acordo com a Equação 6.1.

$$t_{b(i)} = t_{tr(i)} + t_{d(i-1)} \tag{6.1}$$

Para garantir que o tempo da esteira seja constante, realiza-se um ajuste na velocidade considerando-se a magnitude da velocidade a partir da Equação 6.2.

$$t_{tr(i)} = t_{b(i)} - t_{d(i-1)} \tag{6.2}$$

Com isso pode-se realizar o cálculo da velocidade média da esteira de acordo com a equação 6.3.

$$V_{m(i)} = \frac{L_{(i)}}{t_{tr(i)}} = \frac{L_{(i)}}{t_{b(i)} - t_{d(i-i)}} \tag{6.3}$$

Para calcular o perfil de velocidade se leva em conta a velocidade media da esteira. O perfil de velocidade tem a forma da Figura 6.2.

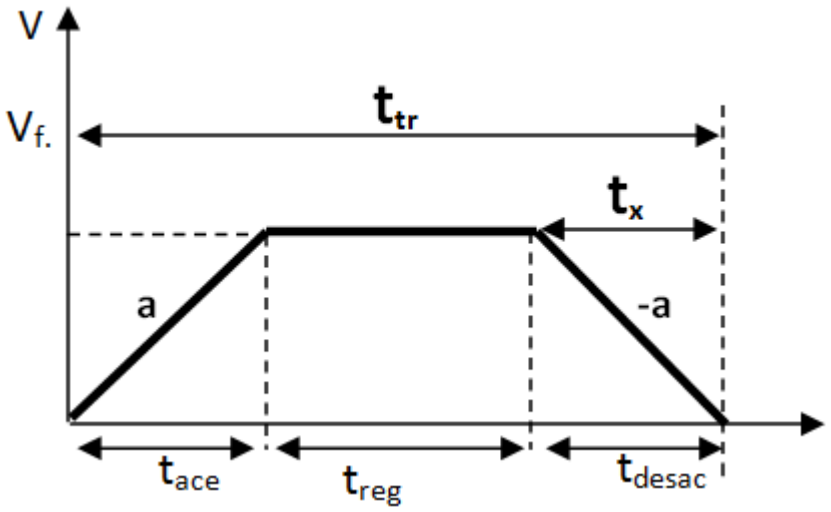

Figura 6.2 Perfil de Velocidade do Sistema de Controle.

A velocidade média deste perfil foi calculada a partir da Equação 6.4.

$$v_{m(i)} = \frac{1}{t_{tr(i)}} \int_0^{t_{tr(i)}} v(t)dt = \frac{1}{t_{tr(i)}} v_{f(i)}\left(t_{tr(i)} - t_x\right) \tag{6.4}$$

Considerando-se as igualdades apresentadas na Equação 6.4, e substituindo a mesma na Equação 6.3, obtém-se à Equação 6.6, levando em conta 6.5.

$$\mathrm{v_{f(i)}} = \mathrm{at_x} \rightarrow \mathrm{t_x} = \frac{\mathrm{v_{f(i)}}}{\mathrm{a}} \tag{6.513}$$

onde t_x é o tempo de desaceleração e a é a aceleração.

$$v_{m(i)} = \frac{1}{t_{tr(i)}} v_{f(i)} \left(t_{tr(i)} - \frac{v_{f(i)}}{a}\right) = v_{f(i)} - \frac{v_{f(i)}^2}{a * t_{tr(i)}} \tag{6.6}$$

Substituindo e igualando, obtém-se a Equação 6.7.

$$\frac{v_{f(i)}^2}{a * t_{tr(i)}} - v_f + v_{m(i)} = 0 \tag{6.7}$$

Solucionando a Equação (6.7), foi selecionada a Equação 6.8, para conseguir que a velocidade obtida seja maior à calculada com um retardo inicial menor.

$$v_f = \frac{a * t_{tr(i)}}{2} \left(1 - \sqrt{1 - 4\left({v_{m(i)}}/{a * t_{tr(i)}}\right)}\right) \tag{6.8}$$

Isto supõe uma limitante no parâmetro do tempo de transporte, já que a quantidade ao interior do radical da Equação 6.8 deve ser maior o igual a zero, pelo tanto se deve cumprir a desigualdade mostrada na Equação (6.9).

$$1 - 4\left({v_{m(i)}}/{a * t_{tr(i)}}\right) \geq 0 \quad ; \quad \frac{4 * v_{m(i)}}{a * t_{tr(i)}} \leq 1 \tag{6.9}$$

Isto significa que quando o tempo de transporte é muito pequeno para uma velocidade media e uma determinada aceleração, não se pode construir um perfil de velocidade que possa satisfazer a condição de aceleração e desaceleração.

A velocidade final máxima possível para condições de aceleração dadas se obtém quando o perfil de velocidade deixa de ser trapezoidal e se converte em um perfil triangular, como o apresentado na Figura 5.3, onde o valor máximo pode ser calculado de acordo com a Equação 6.10(6.10).

$$v_{fmax} = \frac{a * t_{tr}}{2} \tag{6.10}$$

Com este perfil de velocidade, pode-se obter a velocidade média, a mesma pode ser calculada obedecendo à Equação 6.11(6.11).

$$\mathrm{v_{mmax}} = \frac{1}{\mathrm{t_{tr}}}\left(\frac{\mathrm{t_{tr}} * \mathrm{v_{fmax}}}{2}\right) = \frac{1}{\mathrm{t_{tr}}}\left(\frac{\mathrm{t_{tr}} * \mathrm{a} * \mathrm{t_{tr}}}{2 * 2}\right) \tag{6.11}$$

6.4 Descrição do Modelo de Simulação

Cada bloco funcional simula uma unidade de produção composto do sistema que realiza as atividades próprias da unidade, carregamento, montagem central ou controle de qualidade e o transporte dos produtos entre as unidades. Nessas atividades pode existir certa incerteza nos tempos de trabalho das unidades, portanto deve-se realizar o ajuste da velocidade da esteira para compensar atrasos introduzidos nos processos anteriores. Os parâmetros utilizados são apresentados na Tabela 6.1.

Tabela 6.1 Parâmetros da simulação

Parâmetro	Nome	Descrição
S_i	Sensor de entrada	Indica que um novo produto esta na esteira.
T_{da}	Tempo de Atraso	Indica o tempo necessário para compensar o atraso no

		processo anterior, necessário no cálculo do perfil de velocidade.
Tempo	Tempo	Tempo que deve empregar o bloco para completar o processo
Comprimento	Comprimento	Comprimento da esteira.
Aceleração	Aceleração	Mudança da velocidade da esteira transportadora nos momentos de ligação e parada.
S_o	Sensor de Saída	Indicação de término do processo.
T_d	Tempo de Atraso	Indica a diferença entre o tempo empregado pelos processos e o tempo de referência (Variável de entrada).

6.4.1 Modelo do Sistema

A Figura 6.3 apresenta o diagrama de simulação do sistema físico, onde o bloco (Operações e Sistema Físico) realiza os cálculos da simulação da esteira, utilizando os parâmetros de referência, comprimento da esteira, tempo de atraso da unidade anterior e a aceleração do perfil de velocidade. Este bloco determina a posição, velocidades médias e finais da esteira que são calculadas utilizando a Equações 6.8 e 6.2.

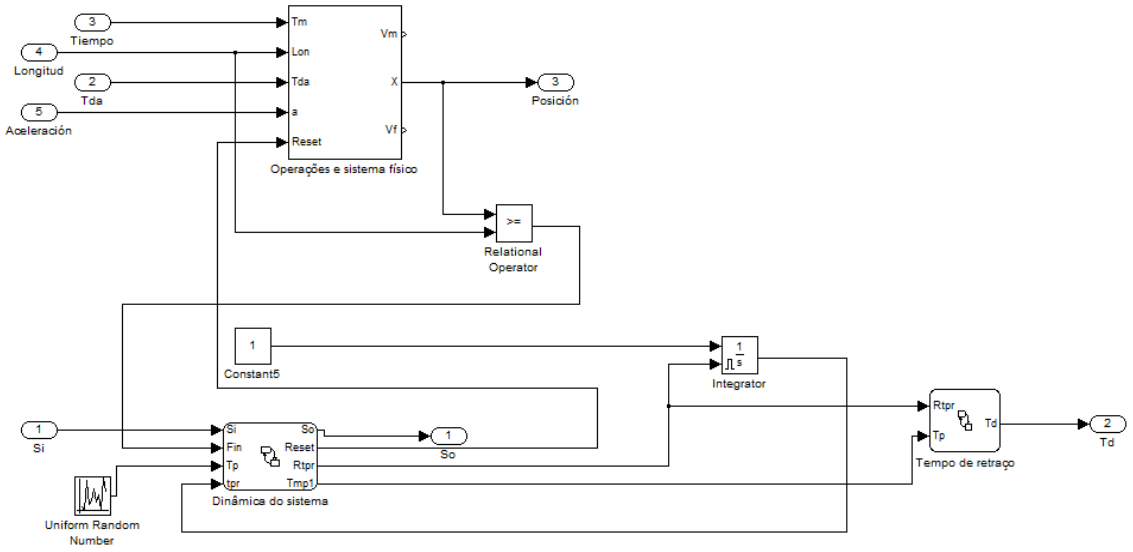

Figura 6.3 Componentes do Processo Controlado.

O bloco (Dinâmica do sistema) corresponde a uma máquina de estados, encarregada de selecionar a dinâmica do bloco funcional dependendo das variáveis de entrada, que são o sensor de entrada (S_i) e a posição do produto. Ao mesmo tempo, é encarregado de ativar o sensor de saída (S_o) marcando o final do processo, de simular a incerteza do processo baseado em um número aleatório com distribuição uniforme entre um limite superior e inferior e finalmente reiniciar as variáveis do sistema quando o produto sai da esteira de transporte. O bloco (Tempo de atraso) é encarregado de realizar o cálculo do tempo de atraso do processo atual e envia este para o seguinte bloco funcional.

6.4.2 Operações e Modelo Físico

No bloco de operações e sistema físico é implementado a partir das Equações 5.47 e 5.48, juntamente com o bloco encarregado de construir o perfil e o controle de velocidade. O diagrama de blocos correspondente a este sistema é mostrado na Figura 6.4.

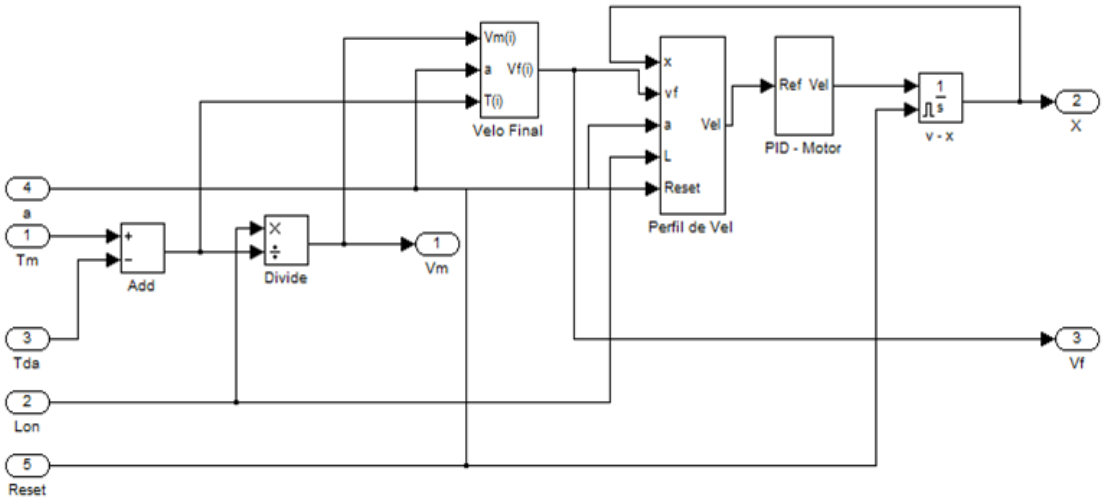

Figura 6.4 Bloco de Operações e Sistema Físico.

6.4.3 Cálculo da Velocidade Final

Para cálculo da velocidade final baseado nas Equações 5.3 e 5.9, foi implementado o sistema apresentado na Figura 6.5. Além disso, se o valor do radical é negativo, a simulação é interrompida.

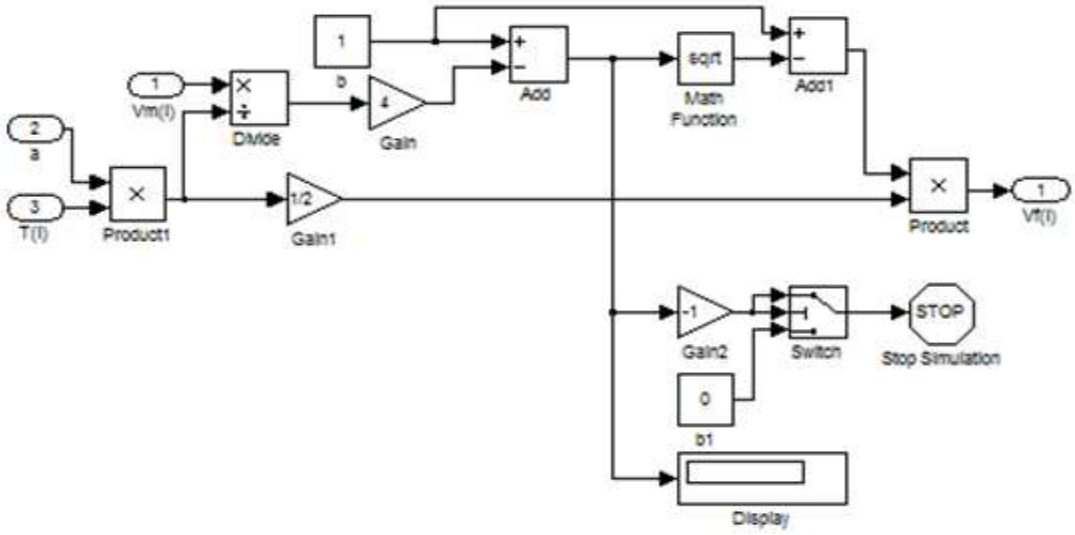

Figura 6.5 Estrutura Interna do Bloco para Cálculo da Velocidade Final.

6.4.4 Construção do Perfil de Velocidade

Este bloco é composto de uma máquina de estados que realiza a escolha da aceleração que deve ter o sistema de acordo à velocidade atual e à posição do produto na esteira transportadora. Foi utilizado um integrador com o propósito de se obter a velocidade do sistema considerando sua aceleração, conforme mostra a Figura 6.6.

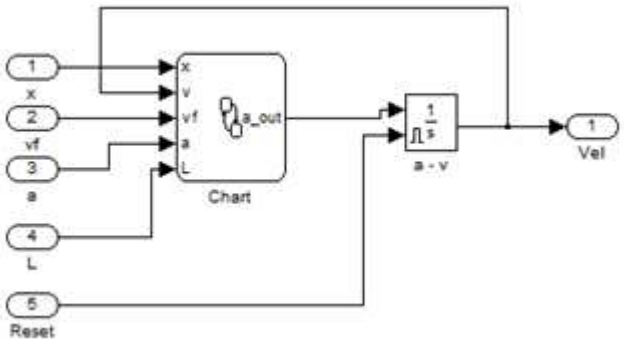

Figura 6.6 Bloco Construtor do Perfil de Velocidade.

A máquina de estados apresentada na

Figura 6.7, inicia no estado "a_1", onde é selecionada a aceleração positiva do perfil de velocidade. Quando a velocidade do sistema alcança sua velocidade final (equação 5.9), a máquina de estados muda para aceleração zero, com o intuito de manter a referência do sistema a uma velocidade constante. Após isso, o sistema chega ao ponto de desaceleração, e a máquina de estados mudará para aceleração negativa. Na parte final, quando a velocidade é igual a zero, o sistema volta para seu estado inicial "a_1" com uma aceleração positiva.

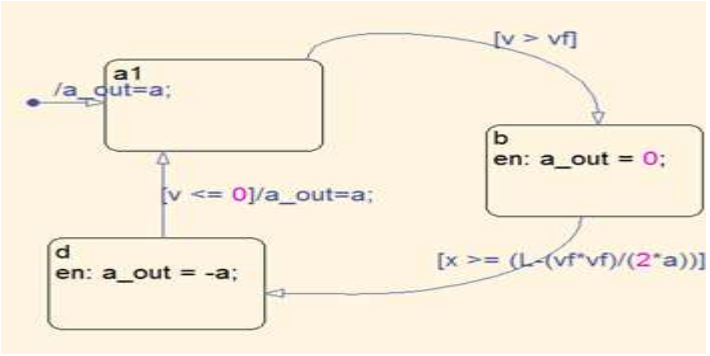

Figura 6.7 Bloco Gerador do Perfil de Velocidade

O integrador que converte a aceleração em velocidade (Figura 6.6) tem um sinal de reinicio externa, esse sinal permanece em um quando não existem produtos no processo e em zero quando um produto termina seu ciclo dentro do processo controlado, logo, a velocidade do sistema será zero quando não existir produtos.

6.5 Projeto do Sistema de Controle

Neste bloco é representada a dinâmica da esteira de transporte de uma unidade de produção, que se movimenta através de um motor de corrente continua através de seu sistema de controle. A dinâmica da esteira e o sistema de controle foram estabelecidos no capítulo 4 deste trabalho. O formalismo matemático descrito a seguir permite representar os sistemas através de eventos discretos (Formalismo DEVS), e usando um método para aproximar sistemas contínuos, expressados como sistemas de Equações diferenciais, usando sistemas de eventos discretos através de QSS (Quantized State System). Em conjunto, os modelos do controlador e a dinâmica

da planta constituem um sistema de controle de estados quantificados (Quantized State Controle (QSC)). A Figura 6.8 apresenta o sistema em QSC.

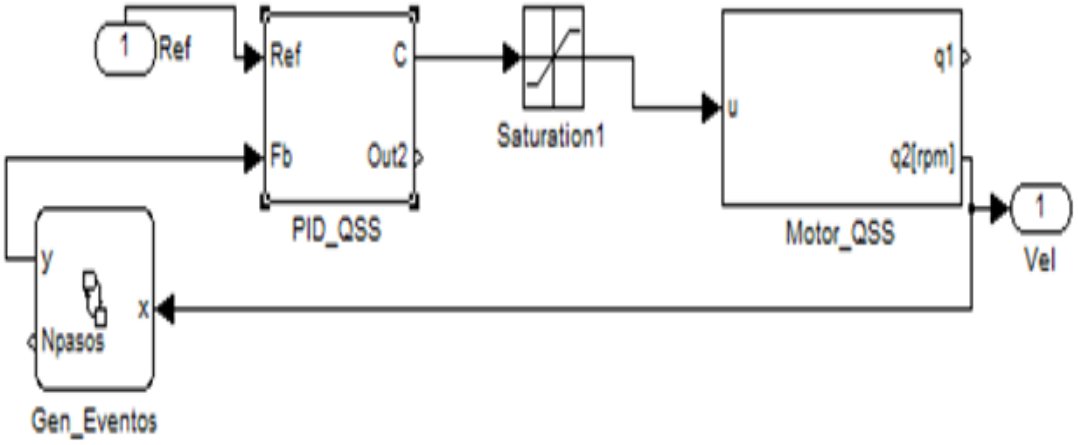

Figura 6.8. Controle de Estados Quantificados - QSC.

Para a seleção da dinâmica do processo implementou-se um bloco para realizar a escolha da parte onde está o produto: na esteira ou na parte final do processo controlado, utilizando-se um tempo aleatório antes de sair. Esse bloco é simulado usando uma máquina de estados em Stateflow® como mostra na Figura 6.9. O processo controlado está inicialmente em estado de "espera" até a chegada de um novo produto, isto acontece quando se ativa o sensor S_0. Depois disso, passa ao estado "Transportando" que corresponde ao percorrido que realiza o produto pela esteira de transporte. Quando se realiza a transição para o estado "transportando", o bloco de operações do sistema físico é ativado (conforme mostra a Figura 6.3). Quando o produto está no final da esteira de transporte, se ativa o sinal "Fin" gerado pelo comparador conectado à entrada do processo controlado chamada de "Longitud" e à saída do subsistema operações. Além disso, se reiniciam as variáveis do sistema físico e a máquina de estados vai para "processo", onde o produto se atrasa um tempo aleatório antes de sair. Quando termina o tempo aleatório, a máquina de estados gera um pulso no sensor de saída e quando passa pelo estado "x" terminando o ciclo e voltando ao estado "espera".

O bloco "*Tiempo de retardo*" se encarrega de calcular o tempo adicional empregado pelo produto no processo controlado, enviando posteriormente para o processo seguinte o tempo de atraso.

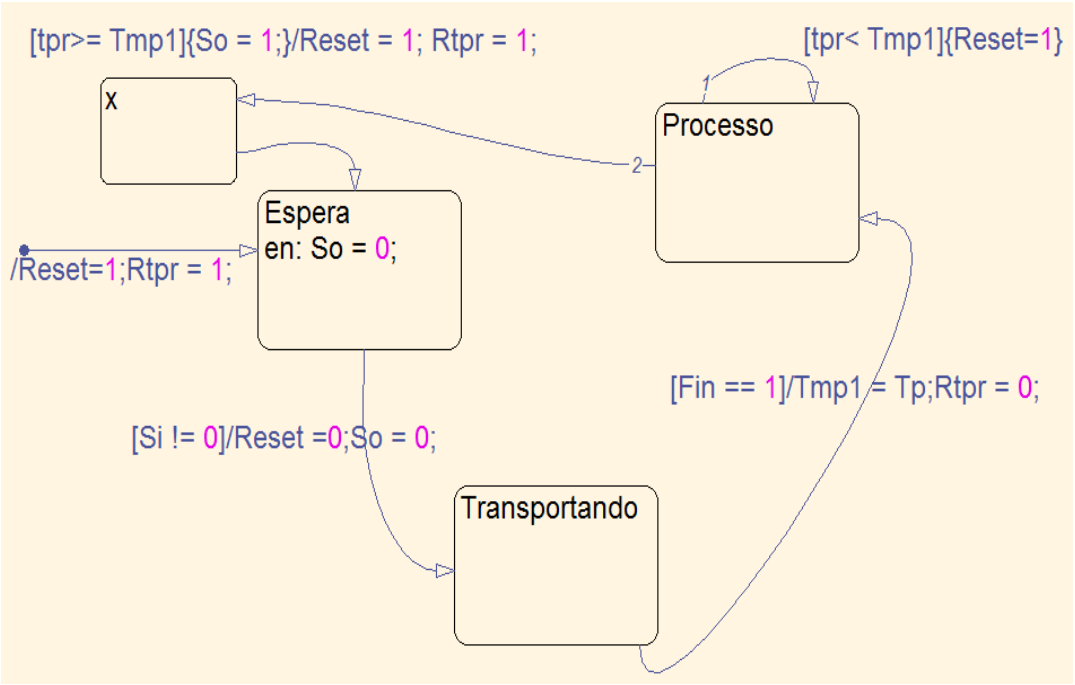

Figura 6.9 Máquina de estados seletora da dinâmica do processo.

6.6 Simulações e Resultados

Com o propósito de testar os processos controlados, é realizada uma simulação de uma linha de montagem composta de quatro unidades de produção. O sistema de simulação é apresentado na Figura 6.10.

Todos os processos foram estimados com uma esteira de transporte com comprimento de cinco metros, utilizando-se uma aceleração de 4m/seg2 para gerar os perfis de velocidade de referência para cada esteira de transporte, e tempo de referência de 5s (tempo que deveria empregar o produto desde a entrada ate a próxima unidade de produção), para um tempo de duração total de 20 segundos em toda a linha de montagem.

O produto passa de uma unidade de produção para outra quando se ativa o sensor S_0 da Figura 6.3 da primeira unidade de produção, ativando o sensor de entrada da unidade seguinte e assim por diante, até percorrer as quatro unidades que compõe o sistema total de transferência.

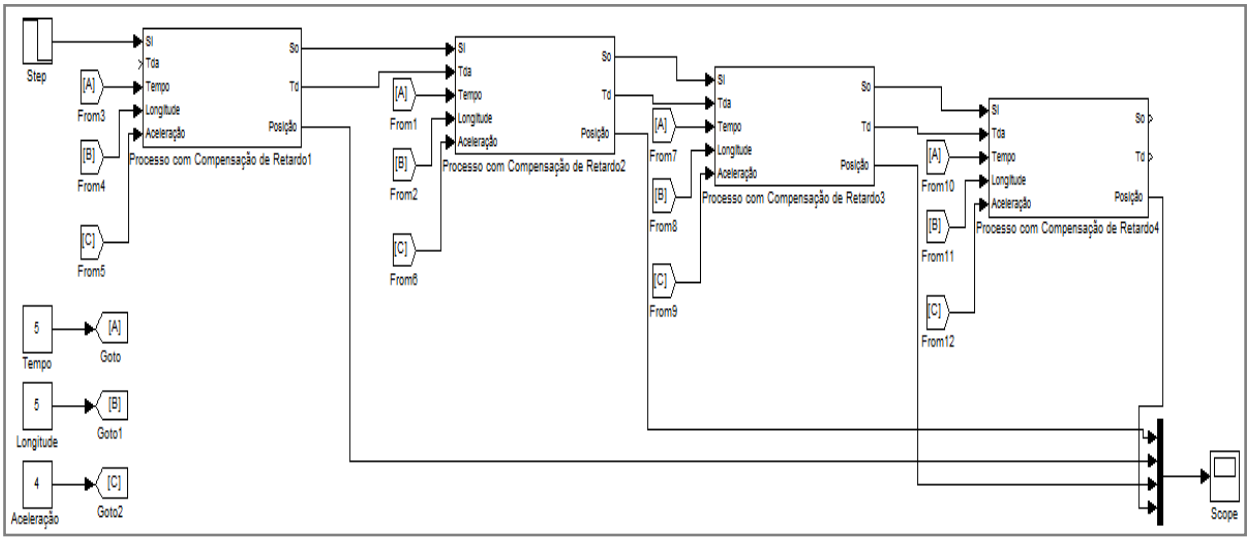

Figura 6.10 Diagrama de Blocos da Linha de Montagem.

Na Figura 6.11 observa-se como a posição do produto aumenta de maneira linear, representando a distância percorrida pelo produto pela esteira de transporte. Quando o produto chega até o final da esteira, a posição volta para zero, indicando que o produto saiu da esteira, passando-se a parte final do processo controlado, onde é utilizado um tempo aleatório limitado entre dois valores, antes de ir para a unidade seguinte de produção. Como pode-se constatar o tempo na parte final do processo controlado onde a posição do objeto permanece em zero, é diferente para cada processo controlado, em especial entre as unidades de produção três e quatro.

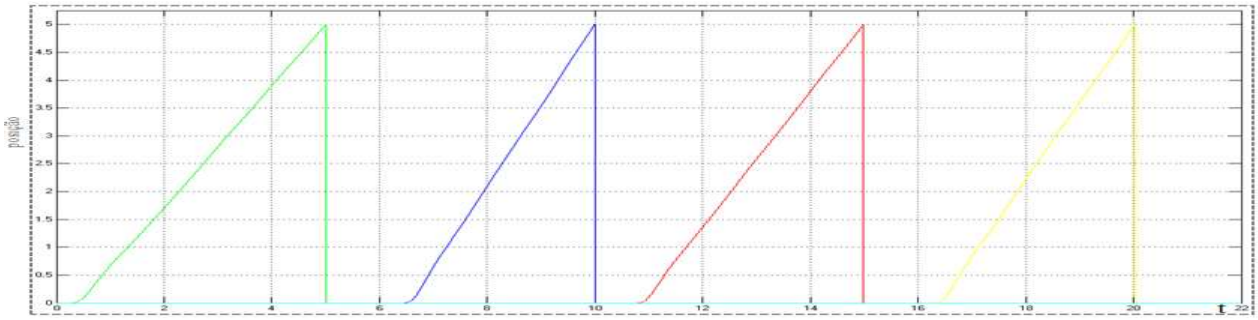

Figura 6.11 Posição do Produto nas Diferentes Unidades de Produção.

Os resultados apresentados na Figura 6.11, mostram a dinâmica da evolução das unidades de produção, referente ao tempo utilizado. Existe um pequeno atraso no sistema, ocasionado pelo sistema de controle, que introduz este atraso no início e no final dos perfis de velocidade, como pode ser observado na Figura 6.12.

A resposta do sistema em estado estável apresenta pequenas oscilações que são próprias dos sistemas de variáveis quantificadas, devido ao fato de que a variável não está definida em certos intervalos, então quando a referência toma um dos valores não definido, o controle começa a acumular erro ate que a saída do sistema alcança um valor acima do valor de referência.

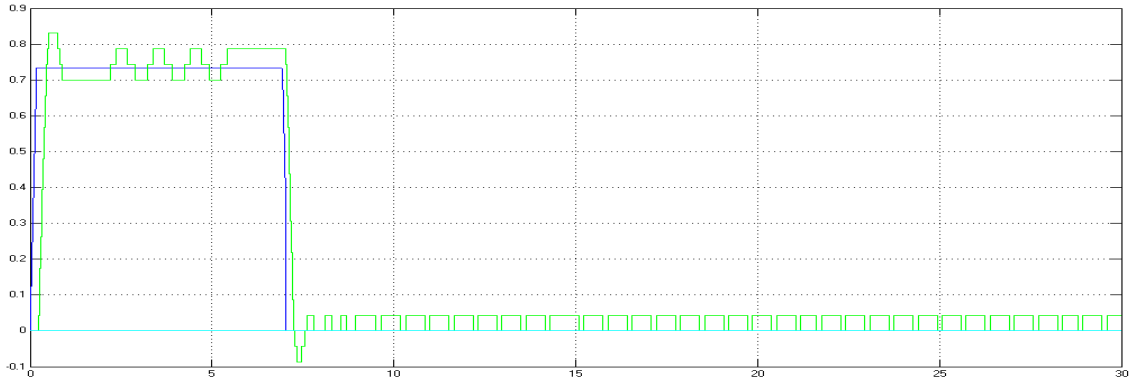

Figura 6.12 Perfil de Velocidade usado como Referência

É importante ressaltar que esta oscilação também influirá no atraso apresentado na parte final de toda a linha de montagem.

6.7 Considerações Finais

Neste capítulo realizou-se a modelagem e projeto de arquitetura de controle para um sistema de produção industrial composto de várias estações, cuja dinâmica corresponde com uma dinâmica hibrida. O propósito é garantir tempos de produção uniformes no processo produtivo.

Com a arquitetura de controle proposta conseguiu-se que os tempos da linha de produção fossem uniformes, mesmo com as perturbações existente no sistema.

Capítulo 7

7. Prototipagem Rápida de um Sistema de Arquitetura Híbrida

7.1 Introdução

Como parte do processo de validação da proposta de modelagem dos sistemas de produção industrial de arquitetura híbrida, baseada no formalismo DEVS e o método dos Sistemas de Estados Quantificados (QSS), realizou-se a implementação em um sistema embarcado do modelo e controle desenvolvido no capítulo 4 utilizando prototipagem rápida.

O processo utilizado corresponde ao descrito na sessão 4.3, cujo modelo foi descrito na seção 5.2 e que corresponde a um sistema de transportes de objetos através de uma esteira, com um sensor de entrada e um sensor de saída do sistema.

Nesta aplicação são considerados os modelos DEVS apresentados na seção 5.13, onde as variáveis foram aproximadas, com o propósito de realizar o cálculo no modelo em ponto fixo, evitando-se assim possíveis dificuldades para descrição em hardware de operações que utilizem cálculos com números decimais.

7.2 Modelo do Processo

Nas sessão 5.13 desta tese foram realizadas a modelagem e simulação do processo de arquitetura híbrido , onde foi utilizada a saída de velocidade da esteira para determinar o instante de saída dos objetos da esteira de transporte, ou seja, utilizou-se uma única saída do bloco *Motor*. No caso deste estudo neste capítulo, foram utilizadas duas saídas que são apresentadas como q_1 *e* q_2 no bloco esteira da Figura 5.24, e que correspondem à velocidade e posição da esteira. Isto porque para a implementação em hardware se utilizou a posição dos objetos na esteira para

controlar os eventos de saída do *sensor2* e a saída de velocidade será utilizada como o sinal de realimentação do sistema de controle.

A diferença com o modelo apresentado na sessão 5.13, é a utilização da saída de velocidade na geração dos eventos para os blocos (F_8 e Z_3). Para facilitar a implementação eliminou-se no bloco *Evento de Saída* a divisão entre o tempo transcorrido e a velocidade, que permitia calcular o instante de saída dos produtos da esteira, onde isto será realizado considerando-se a posição. Isto pode ser observado na Equação 7.1 que representa o novo modelo DEVS do Bloco *Evento de Saída.*

$$
\begin{gathered}
M_{S2} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad onde \\
X = Y = R; S = R^4 \times R_0^+ \\
\delta_{int}(s) = \delta_{int}(est, pos1, pos2, pos3, \sigma) = \\
\begin{Bmatrix} (0, pos1, pos2, pos3, \infty) \quad se \quad est = 1 \\ (1, pos2, pos2, pos3, \infty) \quad se \quad est = 2 \\ (2, pos2, pos3, pos3, \infty) \quad se \quad est = 3 \end{Bmatrix} \\
\delta_{ext}(s, e, x_v) = \delta_{ext}(est, pos1, pos2, pos3, \sigma, e, x_v) = \\
\begin{Bmatrix} (1{,}20, pos2, pos3, \infty) \quad se \ port = 0 \ y \ est = 0 \\ (2, pos1{,}20, pos3, \infty) \quad se \ port = 0 \ y \ est = 1 \\ (3, pos1, pos2{,}20, \infty) \quad se \ port = 0 \ y \ est = 2 \\ (est, pos1, pos2, pos3, \infty) \quad se \ port = 0 \ y \ est = 3 \\ (est, pos1, pos2, pos3, \infty) \quad se \ port = 1 \ y \ est = 0 \\ (est, pos1 - 1, pos2, pos3, \infty) \quad se \ port = 1 \ y \ est = 1 \ y \ (pos1 - 1) > 0 \\ (est, pos1 - 1, pos2, pos3{,}0) \quad se \ port = 1 \ y \ est = 1 \ y \ (pos1 - 1) = 0 \\ (est, pos1 - 1, pos2 - 1, pos3, \infty) \quad se \ port = 1 \ y \ est = 2 \ y \ (pos1 - 1) > 0 \\ (est, pos1 - 1, pos2 - 1, pos3{,}0) \quad se \ port = 1 \ y \ est = 2 \ y \ (pos1 - 1) = 0 \\ (est, pos1 - 1, pos2 - 1, pos3 - 1, \infty) \quad se \ port = 1 \ y \ est = 3 \ y \ (pos1 - 1) > 0 \\ (est, pos1 - 1, pos2 - 1, pos3 - 1{,}0) \quad se \ port = 1 \ y \ est = 3 \ y \ (pos1 - 1) = 0 \end{Bmatrix} \\
\lambda(s) = \lambda(est, pos1, pos2, pos3, \sigma) = (y) = (y + 1) \\
ta(s) = ta(est, pos1, pos2, pos3, \sigma) = \sigma
\end{gathered}
\tag{7.1}
$$

Na Equação 7.1 o cálculo do tempo que permanece cada objeto (*sig1, sig2, sig3*) na esteira, se realiza através da divisão entre a distância estabelecida para o comprimento da esteira e a velocidade da mesma. Um dos inconvenientes deste procedimento é que na simulação através de Matlab® as divisões não apresentam maiores dificuldades, enquanto que na implementação em FPGA, essas operações são mais difíceis de serem realizadas, considerando que a descrição em

hardware de um bloco divisor exige-se a utilização de muitos componentes (multiplexores, comparadores, máquinas de estados finitos de controle, e outros) (Paschalakis, 2003). Além disso, o bloco combinacional da divisão que está disponibilizado no software Quartus® do fabricante de produtos de FPGA Altera que permite realizar divisões utilizando um único ciclo de máquina, ocupa uma grande quantidade de elementos lógicos (dependendo do tamanho das variáveis pode ocupar ate 1.000 elementos lógicos). Por esse motivo foi necessário utilizar diretamente a posição de cada objeto para realizar o cálculo do tempo de saída no momento de realizar a implementação em hardware do sistema híbrido .

A representação do sistema híbrido para ser embarcado em FPGA, é apresentada na Figura 7.1, onde a saída de posição do bloco Controlador-Motor se conecta ao bloco Sensor2 para determinar o instante em que os objetos que tenham ingressado ao processo, saem da esteira de transporte.

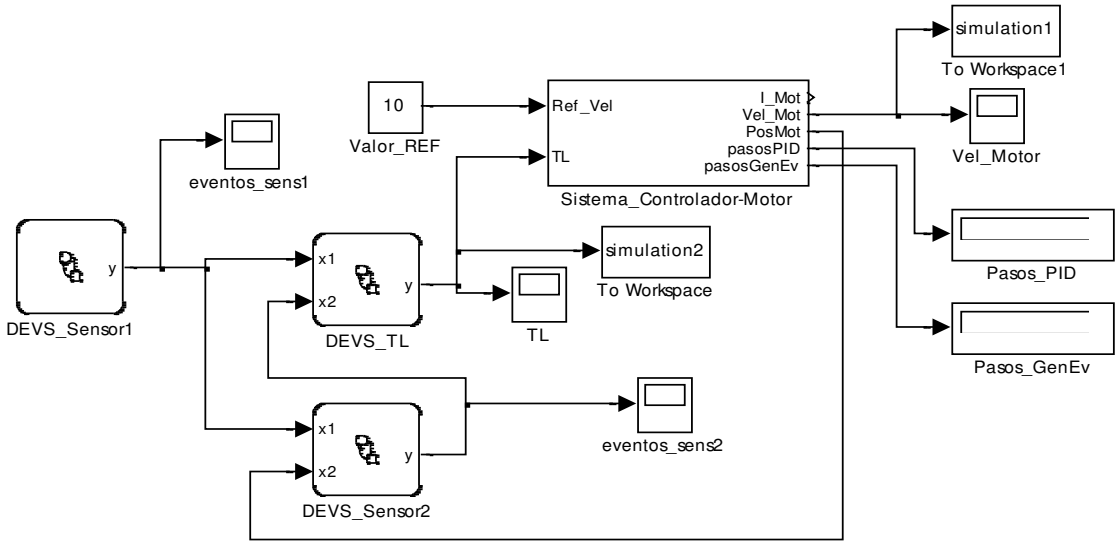

Figura 7.1. Diagrama de Blocos da Simulação do Processo Híbrido

Nessa estrutura tem-se a variável *est*, que representa o estado atual do sistema, e considera valores entre 0 e 3, dependendo do número de objetos que estejam sobre a esteira nesse instante. No momento de ingressar um objeto na esteira de transporte ocorrerão os eventos seguintes:

- É gerado um evento de entrada no porta 0 do modelo.
- Ativa-se a função de transição externa.

- Muda-se o estado atual.
- Atualização dos valores para as variáveis pos1, pos2 e pos3 colocando o Número 20 na variável de posição correspondente, com o intuito de contar os produtos na esteira.
- Atualização do tempo de estado σ, fazendo ele infinito para desativar as transições internas do modelo.

O valor de 20 corresponde com a quantidade de passos da variável posição da esteira, o que determina uma distância fixa que corresponde ao longo da mesma. Esse valor pode ser mudado no momento em que se quer realizar uma variação no comprimento da esteira.

Quando ocorre um evento no porta 1, indica que a posição tenha mudado em um certo valor que depende do quantum da variável de posição no modelo. Isto ativa novamente a função de transição externa atualizando os valores das variáveis de posição dos objetos na esteira. Além disso, se o objeto alcançou o final da esteira, ou seja, se o contador *pos1* chegou a 0, coloca em 0 o tempo σ. Isto ocasiona os seguintes fatos:

- Agenda-se uma transição interna imediatamente.
- Ativa-se a função de transição interna, encarregada de gerar um evento de saída através da função $\lambda(s)$
- Incrementa em um a quantidade de elementos que tem deixado à esteira.
- Atualiza-se os valores das variáveis *pos1*, *pos2* e *pos3 através d*a função de transição interna, copiando na variável *pos1* o conteúdo da variável *pos2* e na variável *pos2* o conteúdo da variável pos3. Com isto sempre estará na variável pos1 o seguinte objeto por sair da esteira.

A simulação é apresentada na Figura 7.2, onde o gráfico da direita mostra o comportamento do torque de carga, considera o ingresso e a saída de objetos no processo. No gráfico da esquerda é apresentada a saída do sistema controlado. Pode-se observar que o controlador mantém a velocidade do sistema no estado estável, mas conforme se apresentam a entrada e saída dos objetos do processo, existem certas perturbações ao ser mudadas o torque de carga da esteira de transporte.

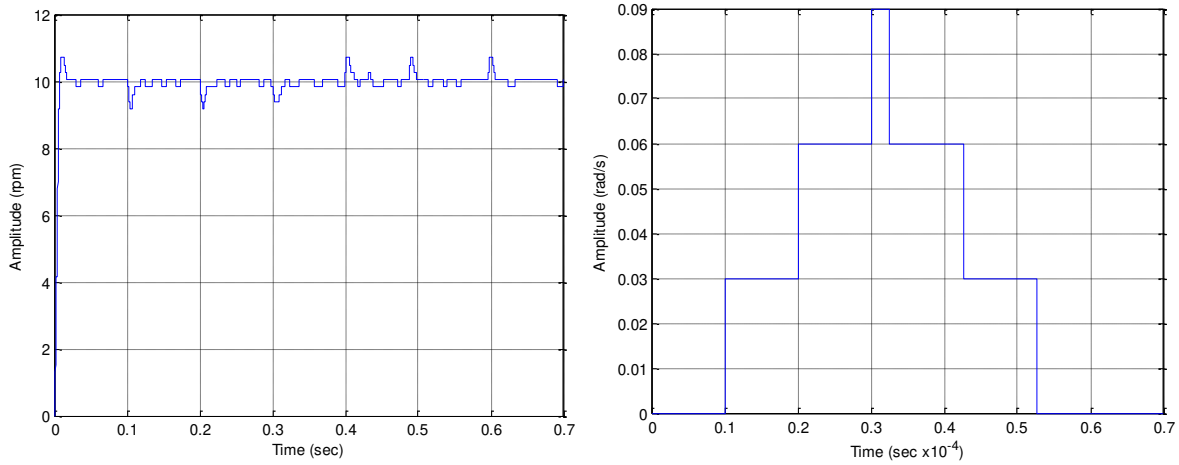

Figura 7.2. Simulação do Modelo Híbrido

Na Figura 7.3 são apresentados os gráficos de eventos de entrada de objetos na esquerda e de saída dos objetos da direita.

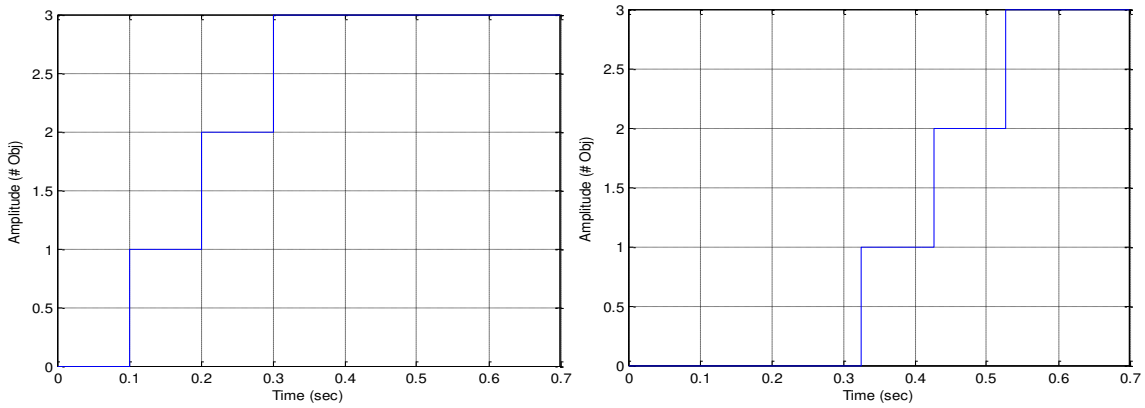

Figura 7.3. Eventos de Entrada e Saída do Sistema.

7.3 Ajuste do Modelo para a Geração da Descrição em VHDL

Antes de gerar o código mediante a ferramenta de prototipagem rápido HDL Coder de Matlab® devem-se realizar alguns ajustes ao modelo, que são descritas a continuação:

- Realiza-se um escalamento para que todas as variáveis trabalhem em números inteiros.
- Ajusta-se o modelo DEVS dos integradores QSS para poder realizar as divisões que calculam o tempo de integração que depende do quantum. Para isto foi incorporado um estado adicional que realiza a divisão através de subtrações sucessivas.

Na Figura 7.4 se apresenta a implementação do diagrama de estados do modelo DEVS para um integrador QSS, utilizando a ferramenta de Stateflow.

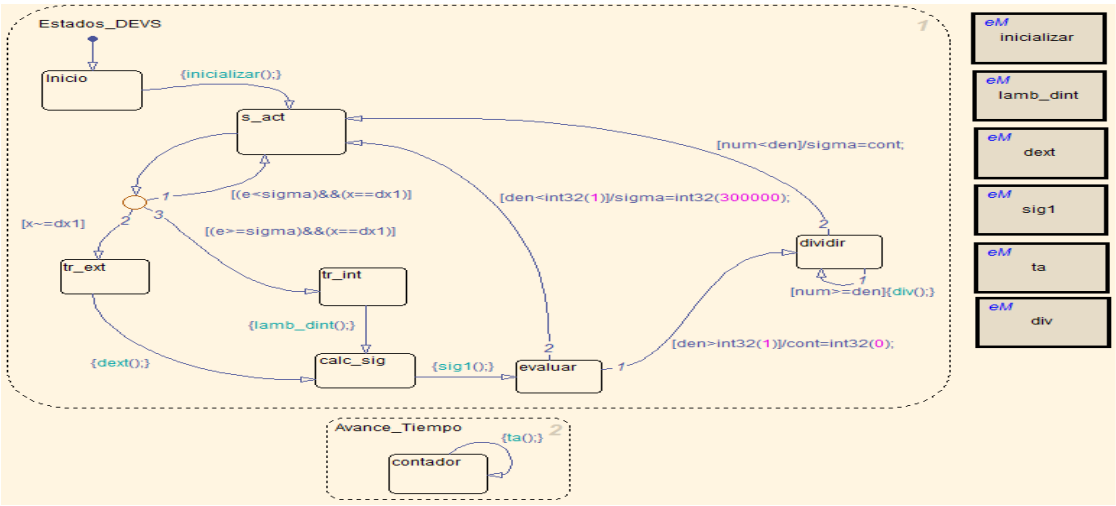

Figura 7.4 Diagrama de Stateflow do Modelo DEVS para um Integrador QSS.

Na Tabela 7.1 é realizada uma descrição das funções desenvolvidas no diagrama de estado do modelo DEVS do integrador QSS.

Tabela 7.1 Funções do diagrama DEVS para o integrador QSS

FUNÇÃO	DESCRIÇÃO
inicializar	Inicializa os valores das variáveis do modelo.
lamb_dint	Quando ocorre uma transição interna, realiza o cálculo do valor da saída e o valor do estado seguinte.
dext	Quando ocorre um evento de entrada externo, se executa esta função de transição externa, calculando o valor do estado seguinte.
sig1	Avalia o valor da derivada para dar passo ao cálculo do tempo σ no que permanece o sistema antes de executar uma transição interna.
ta	Incrementa os contadores de tempo transcorrido.
Div	Calcula a divisão entre a variável de estado e sua derivada para obter o tempo σ. Este é o tempo calculado para o "seguinte passo" de acordo ao valor de quantificação (quantum).

Na Figura 7.5 se apresenta os códigos que compõe a função estática do modelo DEVS da esteira que calcula a variável x_2'. Essa função corresponde à inicialização de variáveis do sistema. Na linha 10 mostra-se a mudança na operação matemática, onde a função original é representada pela Equação 7.2.

$$\dot{x}_2(t) = 5523{,}81 \cdot q_1(t) - 0{,}62 q_2(t) - 0{,}238 \times 10^6 \cdot T_L(t) \tag{7.2}$$

Entretanto a descrição VHDL vai ser trabalhada com números inteiros, de tal maneira que na linha 10 se muda 5523.81 por uma variável inteira de valor 5524 e -0.62 é trocada por -1. Além disso, 238000 se trocam por 238.

Com isto é realizada a compensação das variáveis no modelo DEVS de geração do torque de carga, já que seu valor se encontra no intervalo [0,1] e por tanto se escala por 1000, como se comenta na linha 11.

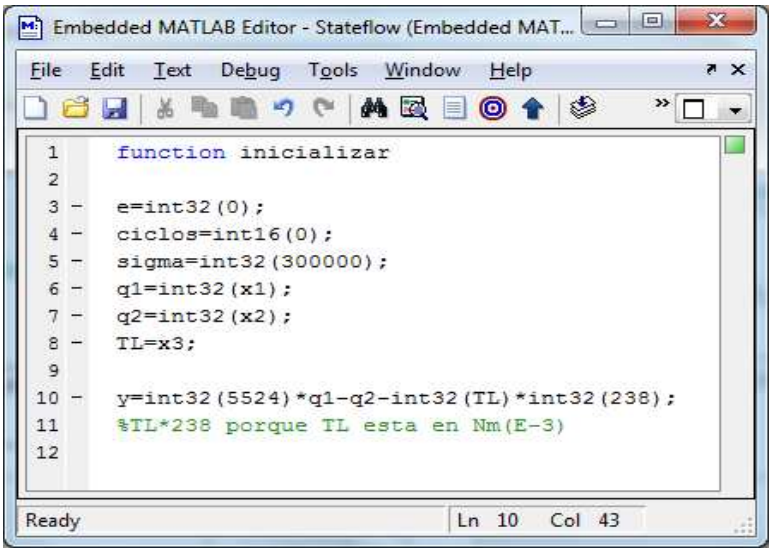

Figura 7.5 Código Ajustado para a Geração da Descrição em VHDL

Essa aproximação tem repercussão na dinâmica da esteira, já que se modificaram os valores da matriz de estado, apresentando-se uma pequena diferença, no comportamento do sistema

simulado utilizando variáveis em ponto flutuantes, respeito à utilização das variáveis em ponto fixo.

Na Figura 7.6 se compara a saída do modelo QSS da esteira usando os valores em ponto flutuante (linha azul) e o resultado com a aproximação através de valores inteiros (linha verde). Pode-se observar como a resposta transitória é a mesma e a diferença se encontra no valor de estado estável. Mas o modelo representado através de variáveis inteiras pode ser considerado como uma boa aproximação para descrever o sistema híbrido.

Um aspecto importante a levar em conta na implementação é a especificação do formato das constantes. Sendo a variável *TL* de 16 bits o resultado de toda a operação deve ser especificado em uma variável de 32 bits, entretanto foi necessário mudar o tamanho de *TL* para 32 bits. Além disso, se deve especificar o formato da multiplicação de *TL* e 238 antes de ser realizada. Isto com o propósito de evitar que a ferramenta HDL Coder gere muitas variáveis temporais no código VHDL. Isto acontece porque a ferramenta, se não conhece o formato da operação matemática, realiza varias comprovações de transbordamento de dados e muda o formato VHDL para encontrar o indicado.

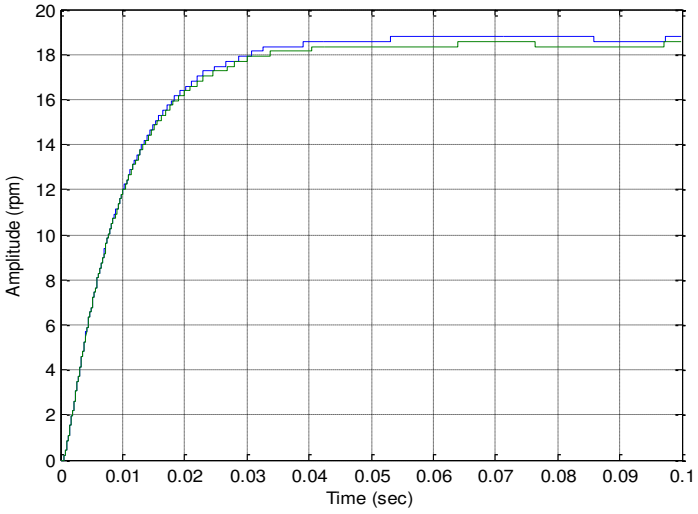

Figura 7.6 Saída do Modelo QSS da Esteira

A função *sig1* avalia o valor da derivada para determinar o numerador e denominador com os que se calcula o tempo σ através da função *div* descrito na Tabela 7.2. Nesta função se escala o

tempo por 10000 para converter-lho em centenas de microssegundos. Isto pela rápida resposta da esteira, que tem repercussões na quantidade de elementos lógicos necessários na implementação da descrição em VHDL. Entretanto, quanto maiores forem estas variáveis, maiores devem ser os registros e blocos que realizam as diferentes operações aritméticas. A função *sig1* é apresentada na Figura 7.7

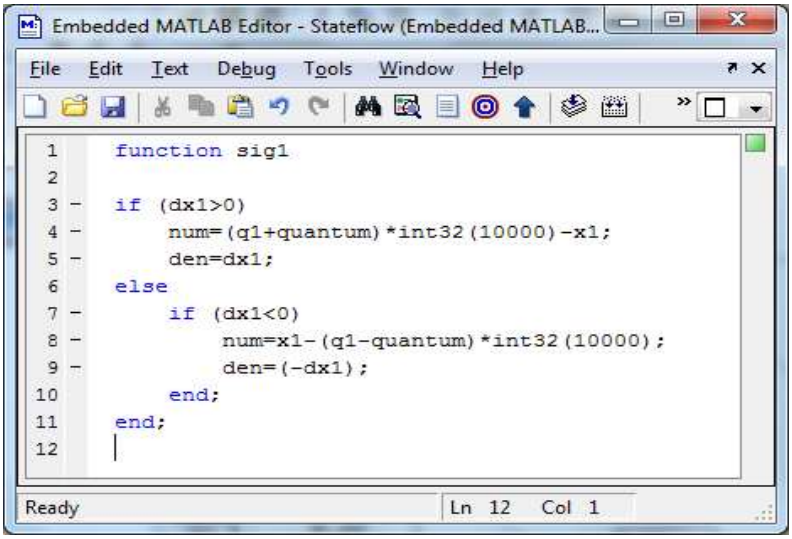

Figura 7.7 Cálculo do Tempo e Escalado para sua Descrição em VHDL.

Uma vez se tenha ajustado o tipo das variáveis a utilizar em todos os blocos do modelo, muda-se o tipo da variável dos conetores de entrada e de saída dos subsistemas. No diagrama de blocos da Figura 7.8, por exemplo, os conectores de entrada e de saída são respectivamente *Vs*, *TL*, *q1*, *q2[rad/s]* e *q3*, a descrição destas variáveis se mostra na Tabela 7.2. Dentro das propriedades destas variáveis deve ser selecionado o mesmo tipo de dado das conexões de entrada e saída.

Tabela 7.2 Descrição das variáveis de entrada e saída do diagrama DEVS da esteira

VARIÁVEL	DESCRIÇÃO
Vs	Tensão aplicada no motor da esteira
TL	Torque de carga.
q1	Variável de estado quantificada para a corrente de armadura da esteira.
q2[rad/s]	Variável de estado quantificada para a velocidade da esteira em rad/s.
q3	Variável de estado quantificada para a posição da esteira que resulta de

	integrar a velocidade.

A simulação do sistema híbrido se apresenta na Figura 7.9, onde se pode observar a mudança da velocidade da esteira devido a os objetos que entram e saem. Como é apresentado na Figura 7.10. O valor de referencia para a velocidade da esteira é de 300 rad/s e o tempo está escalado, de tal forma que 1s na simulação deste modelo equivale a 100µs na simulação do modelo inicial.

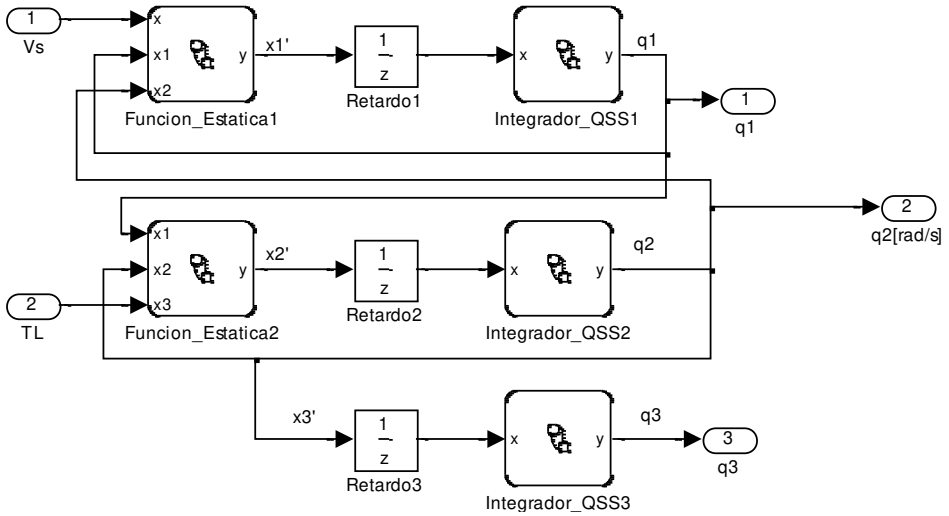

Figura 7.8 Diagrama de Blocos do Modelo Híbrido representado Através de DEVS

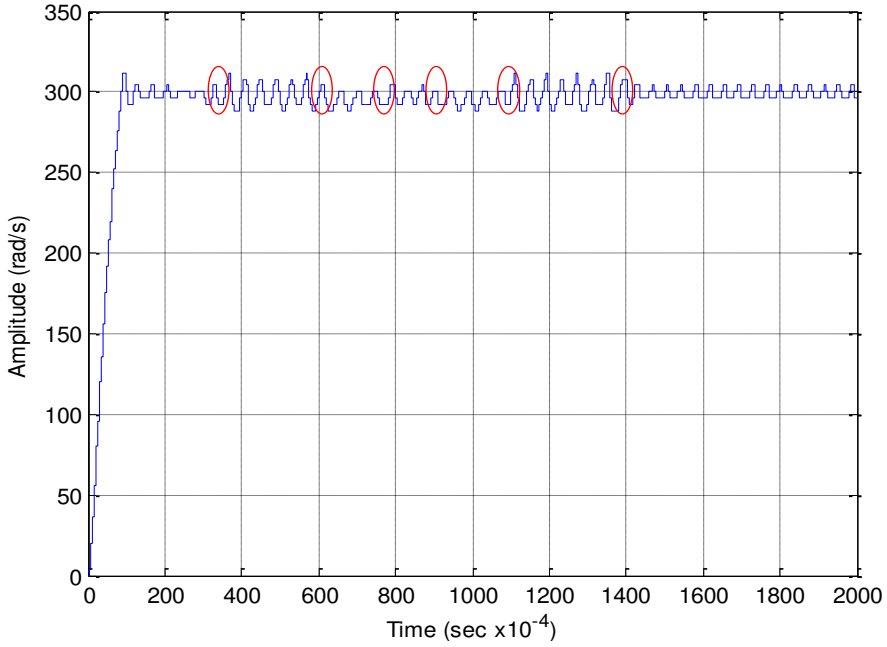

Figura 7.9 Resposta da Velocidade da Esteira

Na Figura 7.11 se apresenta os eventos de entrada (esquerda) e de saída (direita) do sistema que geram perturbações ao mudar o torque de carga.

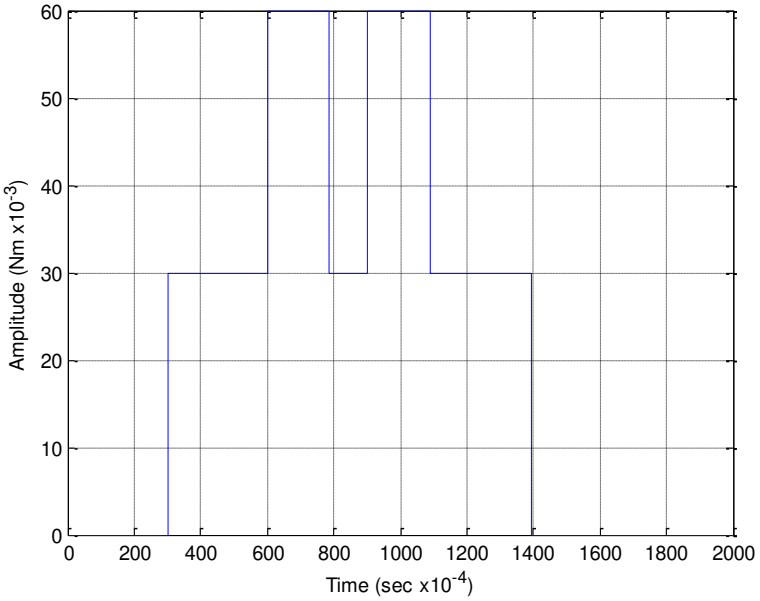

Figura 7.10 Sinal do Bloco Gerador de Torque

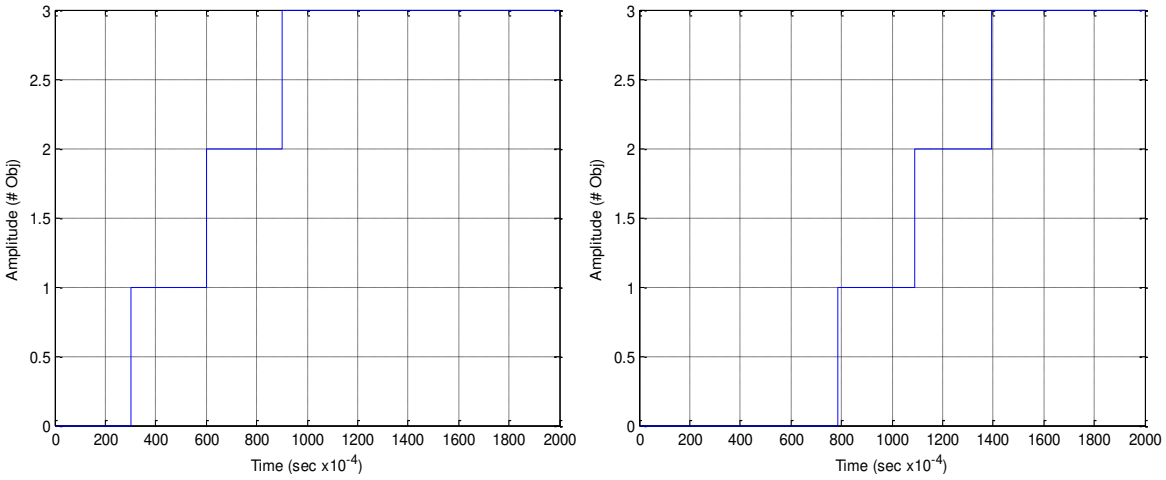

Figura 7.11. Eventos de Entrada e Saída de Objetos do Processo

7.4 Geração da descrição em VHDL

Para a geração da descrição em VHDL do sistema se utilizou a ferramenta HDL Coder. Para isso foram eliminados do modelo os blocos de visualização e se colocaram dos elementos de

saída, denominados *Velmotor* e *Obj_salida*, como se mostra na Figura 7.12. Esses correspondem os portas de saída do modelo de hardware que se está descrevendo. Estabelece-se um valor de referência em 300 rad/s.

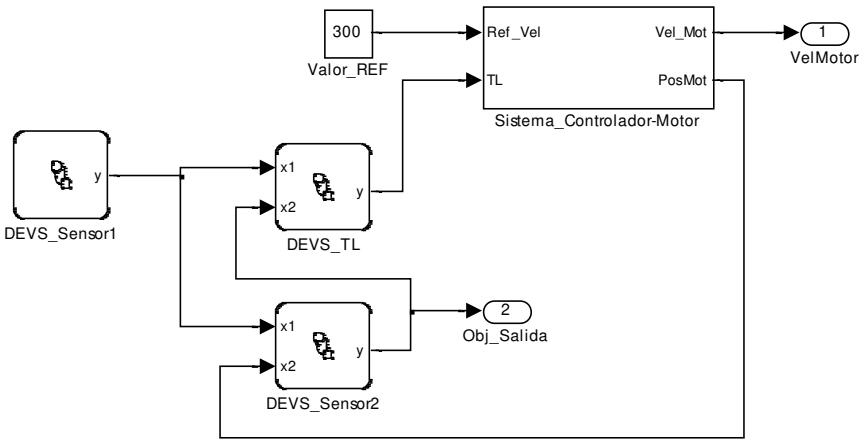

Figura 7.12 Diagrama de Blocos para a Geração da Descrição em VHDL

Utilizando a ferramenta HDL coder se selecciona o Hardware da implementação, na lista de navegação, realizando a seleção do dispositivo a ser utilizado, neste caso *ASIC/FPGA*. Com isso se configura como dispositivo programável uma FPGA. Não é necessário selecionar um fabricante em especial, já que a linguagem VHDL gera as descrições de hardware estandardizados que podem ser implementadas em diversos dispositivos FPGA o CPLD.

7.5 Síntese do Sistema Híbrido em FPGA

Utilizando os arquivos gerados com HDL Coder, realiza-se um projeto na ferramenta Quartus II do fabricante Altera, a interface utilizada foi uma Cyclone II EP2C8Q208C7.

A Ferramenta HDL Coder gera algumas variáveis que não vão ser utilizadas no projeto de Quartus II. Algumas delas são o habilitador de relógio e um sinal de saída que se conecta ao relógio do sistema. Essas variáveis e portas não utilizadas devem ser eliminados com o intuito de poupar espaço na FPGA.

Na Figura 7.13 é apresentado o código gerado como descrição do submodelo *sensor 1,* constituída de dois portas de entrada, relógio e reset, e como pode-se observar na Figura 7.12 não dispõe de nenhuma variável de entrada e a saída pela qual se geram os eventos de ingresso de objetos à esteira de transporte corresponde ao porta *y*.

```
15
16      LIBRARY ieee;
17      USE ieee.std_logic_1164.all;
18      USE ieee.numeric_std.all;
19
20    ENTITY DEVS_Sensor1 IS
21        PORT (
22            clk : IN std_logic;
23            --clk_enable : IN std_logic;
24            reset : IN std_logic;
25            y : OUT std_logic_vector(7 DOWNTO 0));
26      END DEVS_Sensor1;
27
28
29    ARCHITECTURE fsm_SFHDL OF DEVS_Sensor1 IS
30
31        TYPE T_state_type_is_Estados_DEVS is (IN_NO_ACTIVE_
32        TYPE T_state_type_is_Avance_Tiempo is (IN_NO_ACTIVI
33        SIGNAL is_Estados_DEVS : T_state_type_is_Estados_DI
34        SIGNAL is_Avance_Tiempo : T_state_type_is_Avance_T:
35        SIGNAL sigma : signed(15 DOWNTO 0);
36        SIGNAL e : signed(15 DOWNTO 0);
37        SIGNAL T : signed(15 DOWNTO 0);
38        SIGNAL ciclos : signed(7 DOWNTO 0);
39        SIGNAL y_reg : signed(7 DOWNTO 0);
```

Figura 7.13. Código Gerado do Submodelo *Sensor 1.*

Além do habilitador e a saída do relógio, HDL Coder gera alguns códigos para comprovação de transbordamento de dados, estes podem ser eliminados, pois já na simulação de Matlab, isto é comprovado. Esse código se encontra comentado na Figura 7.14 e se repete em varias partes das descrições dos diferentes blocos, pelo qual quando este código redundante é eliminado se reduz de maneira considerável a quantidade de elementos lógicos ocupados pelo modelo.

```
149         is_Estados_DEVS_next <= IN_s_act;
150     WHEN IN_calc_sig =>
151
152         IF dx1 > 0 THEN
153             mul_temp_1 := (resize(q1, 33) + resize(quantum, 33)) * to_signed(100
154
155             --IF (mul_temp_1(47) = '0') AND (mul_temp_1(46 DOWNTO 31) /= "000000
156             --    sub_cast := "01111111111111111111111111111111";
157             --ELSIF (mul_temp_1(47) = '1') AND (mul_temp_1(46 DOWNTO 31) /= "111
158             --    sub_cast := "10000000000000000000000000000000";
159             --ELSE
160                 sub_cast := mul_temp_1(31 DOWNTO 0);
161             --END IF;
162
163             sub_cast_0 := resize(sub_cast, 33);
164             sub_temp := sub_cast_0 - resize(x1, 33);
165
166             --IF (sub_temp(32) = '0') AND (sub_temp(31) /= '0') THEN
167             --    num_next <= "01111111111111111111111111111111";
168             --ELSIF (sub_temp(32) = '1') AND (sub_temp(31) /= '1') THEN
169             --    num_next <= "10000000000000000000000000000000";
170             --ELSE
171                 num_next <= sub_temp(31 DOWNTO 0);
172             --END IF;
```

Figura 7.14. Código de Comprovação do transbordamento de dados.

7.5.1 Descrição do código para o bloco Sensor1

Para explicar a maneira como a ferramenta HDL Coder traduz os modelos de Stateflow para máquinas de estados finitos descritas em VHDL, apresentando o código completo gerado pelo modelo DEVS do bloco *Sensor1*. Na Figura 7.13 mostrou-se a entidade criada pela ferramenta, e esta tem o nome do bloco correspondente no diagrama da Figura 7.12.

Na Figura 7.15 mostra o nome dado ao HDL Coder à arquitetura do bloco e os sinais criados. Estes sinais correspondem aos nomes das variáveis utilizadas no modelo de Matlab®, junto com sinais terminadas em *_next*, que serão utilizadas pelos valores nos estados futuros.

```
29  ARCHITECTURE fsm_SFHDL OF DEVS_Sensor1 IS
30
31      TYPE T_state_type_is_Estados_DEVS is (IN_NO_ACTIVE_CHILD, IN_Inicio, IN_s_act, IN_tr_int);
32      TYPE T_state_type_is_Avance_Tiempo is (IN_NO_ACTIVE_CHILD, IN_contador);
33      SIGNAL is_Estados_DEVS : T_state_type_is_Estados_DEVS;
34      SIGNAL is_Avance_Tiempo : T_state_type_is_Avance_Tiempo;
35      SIGNAL sigma : signed(15 DOWNTO 0);
36      SIGNAL e : signed(15 DOWNTO 0);
37      SIGNAL T : signed(15 DOWNTO 0);
38      SIGNAL ciclos : signed(7 DOWNTO 0);
39      SIGNAL y_reg : signed(7 DOWNTO 0);
40      SIGNAL is_Estados_DEVS_next : T_state_type_is_Estados_DEVS;
41      SIGNAL sigma_next : signed(15 DOWNTO 0);
42      SIGNAL e_next : signed(15 DOWNTO 0);
43      SIGNAL T_next : signed(15 DOWNTO 0);
44      SIGNAL ciclos_next : signed(7 DOWNTO 0);
45      SIGNAL y_reg_next : signed(7 DOWNTO 0);
```

Figura 7.15. Arquitetura de Sinais do Bloco *Sensor1*.

Na Figura 7.16 são apresentadas as linhas geradas pelo processo de relógio e reset. Quando se recebe um sinal de reset, se reiniciam todos os valores das variáveis e se regressa ao estado inicial, se isto não ocorre, todas as vezes que ocorra um ciclo de relógio se passa ao estado seguinte. Neste caso se eliminou a condição do habilitador de relógio (*clk_enable*).

```
47  BEGIN
48      initialize_DEVS_Sensor1 : PROCESS (clk, reset)
49      BEGIN
50          IF reset = '1' THEN
51              sigma <= to_signed(0, 16);
52              e <= to_signed(0, 16);
53              T <= to_signed(0, 16);
54              ciclos <= to_signed(0, 8);
55              y_reg <= to_signed(0, 8);
56              is_Estados_DEVS <= IN_Inicio;
57          ELSIF clk'EVENT AND clk = '1' THEN
58              --IF clk_enable = '1' THEN
59                  is_Estados_DEVS <= is_Estados_DEVS_next;
60                  sigma <= sigma_next;
61                  e <= e_next;
62                  T <= T_next;
63                  ciclos <= ciclos_next;
64                  y_reg <= y_reg_next;
65              --END IF;
66          END IF;
67      END PROCESS initialize_DEVS_Sensor1;
```

Figura 7.16. Processo de relógio

Na Figura 7.17 apresenta-se o inicio do processo correspondente às transações entre os estados, na qual se criam as variáveis temporais para as diversas operações aritméticas e se atualizam os valores dos sinais do sistema.

```
DEVS_Sensor1 : PROCESS (is_Estados_DEVS, sigma, e, T, ciclos, y_reg)
    VARIABLE e_temp : signed(15 DOWNTO 0);
    VARIABLE ciclos_temp : signed(7 DOWNTO 0);
    VARIABLE b_ciclos_temp : signed(7 DOWNTO 0);
    VARIABLE add_temp : signed(8 DOWNTO 0);
    VARIABLE add_temp_0 : signed(8 DOWNTO 0);
    VARIABLE add_temp_1 : signed(16 DOWNTO 0);
BEGIN
    e_temp := e;
    T_next <= T;
    b_ciclos_temp := ciclos;
    is_Estados_DEVS_next <= is_Estados_DEVS;
    sigma_next <= sigma;
    y_reg_next <= y_reg;
```

Figura 7.17. Processo de Transições de Estados

Após o inicio do processo é executado o ***case*** que seleciona o estado seguinte de acordo ao estado atual e as condições das variáveis do sistema.

Na Figura 7.18 se mostra a transição para os estados *inicio* e *s_act.* Durante o estado *inicio* se inicializa os valores das variáveis do bloco, passando ao estado *s_act*. Por outra parte, durante

o estado *s_act* se avalia o valor da variável de avance de tempo *e*, quando iguala ao tempo de estado *sigma*, se passa ao estado de transição interna *tr_int* e si é menor permanece em *s_act*.

```
CASE is_Estados_DEVS IS
     WHEN IN_Inicio =>
          e_temp := to_signed(0, 16);
          b_ciclos_temp := to_signed(0, 8);
          T_next <= to_signed(300, 16);
          sigma_next <= to_signed(300, 16);
          y_reg_next <= to_signed(0, 8);
          --Inicialmente no hay objetos en la banda
          is_Estados_DEVS_next <= IN_s_act;
     WHEN IN_s_act =>

          IF e < sigma THEN
               is_Estados_DEVS_next <= IN_s_act;
          ELSIF e >= sigma THEN
               is_Estados_DEVS_next <= IN_tr_int;
          END IF;
```

Figura 7.18. Estados de *Inicio* e *S_Act*

Na Figura 7.19 se apresenta a função de transição interna do modelo DEVS deste bloco, na qual é incrementada a saída sempre e quando seja menor que 3, ou seja, se descreve una esteira que suporta um máximo de 3 objetos simultaneamente. Além disso, é reiniciada a variável *e*, carregando-se na variável de tempo de estado (*sigma*) o período de tempo *T* com o qual se simula a chegada de objetos e se passa ao estado *s_act*. Finalmente se encerra o processo ***case***.

```
     WHEN IN_tr_int =>

          IF y_reg < 3 THEN
               add_temp := resize(y_reg, 9) + 1;

               --IF (add_temp(8) = '0') AND (add_temp(7) /= '0') THEN
               --    y_reg_next <= "01111111";
               --ELSIF (add_temp(8) = '1') AND (add_temp(7) /= '1') THEN
               --    y_reg_next <= "10000000";
               --ELSE
                    y_reg_next <= add_temp(7 DOWNTO 0);
               --END IF;

          END IF;

          e_temp := to_signed(0, 16);
          sigma_next <= T;
          is_Estados_DEVS_next <= IN_s_act;
     WHEN OTHERS =>
          is_Estados_DEVS_next <= IN_Inicio;
END CASE;
```

Figura 7.19. Função de Transição Interna (*Tr_Int*)

7.5.2 Simulação do modelo em hardware

Depois de comentado e eliminado todo o código redundante e as variáveis necessárias, passa-se a sintetizar o modelo, com o que se obtém os resultados que se apresentam na Figura 7.20.

Com estes resultados se observa que a quantidade de elementos lógicos excede os 8.000 disponíveis na FPGA EP2C8Q208C7, então foi necessário trocar a FPGA pela EP2C20F484C7 a qual tem 20.000 elementos lógicos.

Entity	Logic Cells
Cyclone II: EP2C20F484C7	
CAMP_Modelo_Hibrido_...	8912 (1)
DEVS_Sensor1:u_DEVS_Se...	62 (62)
DEVS_Sensor2:u_DEVS_Se...	225 (225)
DEVS_TL:u_DEVS_TL	144 (98)
Sistema_Controlador_Motor:u...	8485 (10)

Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Met timing requirements	Yes
Total logic elements	8,912 / 18,752 (48 %)
Total combinational functions	8,778 / 18,752 (47 %)
Dedicated logic registers	2,117 / 18,752 (11 %)
Total registers	2117
Total pins	26 / 315 (8 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	52 / 52 (100 %)
Total PLLs	0 / 4 (0 %)

Figura 7.20. Resultados da Síntese do Modelo

Do processo de síntese do modelo obtém-se o diagrama RTL que é apresentado na Figura 7.21, onde se podem observar os quatro blocos principais do modelo híbrido e sua interconexão, que corresponde com o diagrama de Simulink® da Figura 7.12.

Na Figura 7.22 se apresenta a simulação do modelo em Quartus, onde podem ser observados os valores da velocidade de saída durante os primeiros instantes do transitório. O

tempo tem-se dividido por 10 para conseguir uma simulação mais rápida, pois o processo de simulação é muito lento.

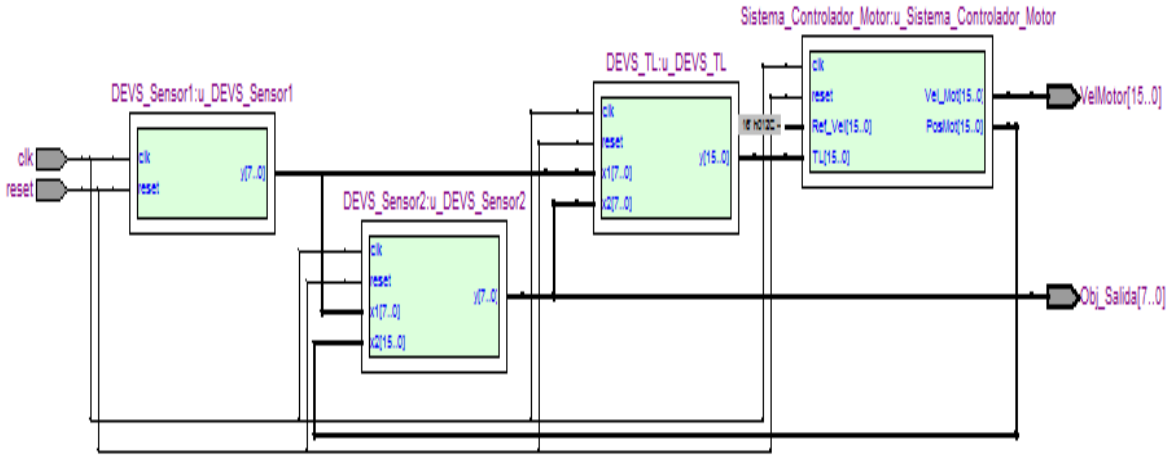

Figura 7.21. Diagrama RTL da Discrição de Hardware do Modelo Híbrido

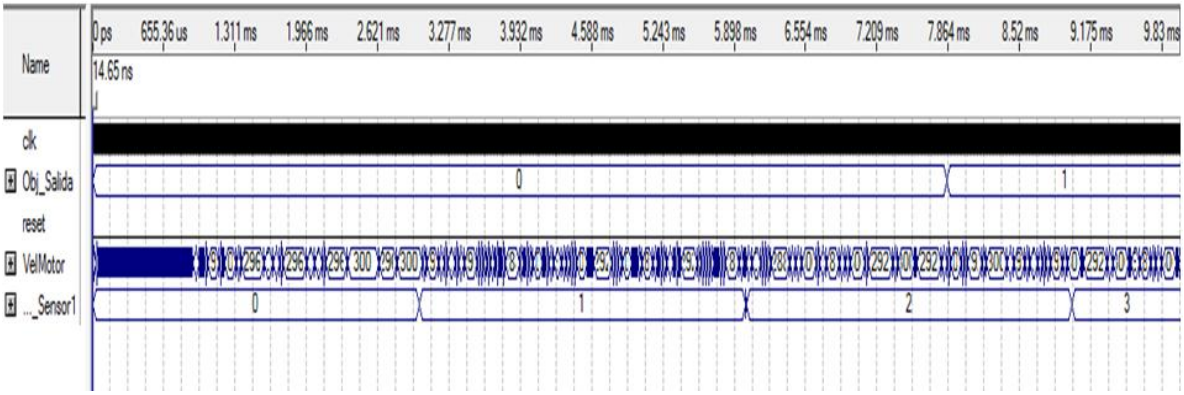

Figura 7.22. Simulação do Modelo Híbrido em Quartus.

Na Figura 7.23 se apresentam dois gráficos, em que se corrobora o resultado do modelo de hardware para o sobre impulso e parte estável do sistema, mediante a simulação em Quartus durante 2ms, equivalentes a 20ms no modelo real. Deve-se levar em conta que no gráfico de Excel os pontos se unem através de linha e no gráfico da simulação do sistema QSS os passos são escalonados. Não obstante, os pontos coincidem tanto em magnitude como em tempo, verificando a correta implementação em hardware do modelo realizado.

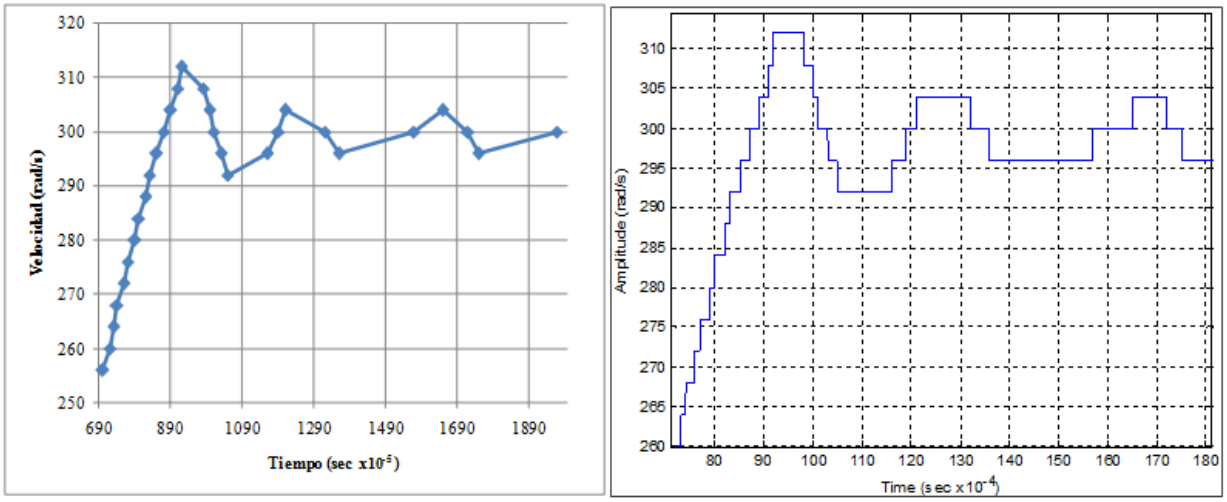

Figura 7.23 Comparação do Modelo de Simulação e o Modelo de Hardware

7.6 Considerações do Capítulo

A ferramenta HDL Coder do Matlab mostrou para esta aplicação uma alta confiabilidade para a geração de condigo em VHDL. Entretanto na conversão produz uma grande quantidade de código redundante que deve ser analisado e eliminado para conseguir um modelo equivalente mais leve, que ocupe uma menor quantidade na implementação de Hardware.

A simulação do modelo híbrido realizada através de Simulink® respondeu de maneira similar com a simulação realizada em descrição de hardware, tanto em magnitudes como em tempos, o qual monstra a capacidade do formalismo DEVS e o método QSS para modelar sistemas híbrido.

Devido à aproximação a números inteiros que foi realizada no modelo, o mesmo apresenta oscilações no estado estável, não obstante, conseguindo obter a resposta esperada, com pequenas mudanças na dinâmica do sistema.

A ferramenta HDL Coder permite uma implementação em hardware pronta para ser embarcada, já que gera as entidades, arquiteturas, funções, operações internas e as interconexões entre os diferentes subsistemas.

Deve-se levar em conta a maneira em que são descritas as operações nas funções de Matlab, vá gerar uma quantidade de variáveis temporais, que vão ocupar um espaço no momento da descrição do sistema.

Para a implementação de um sistema híbrido que leve em conta as operações com números de ponto flutuante, é necessário descrever os módulos de hardware que realizam estas operações, razão pela qual seria necessário modificar a maneira de descrever o modelo DEVS em Matlab, ou modelar todo o sistema diretamente em VHDL sim a ajuda de Matlab.

Como recomendações principais se propõe implementar este tipo de modelos sobre arquiteturas que permitam facilitar as operações com números de ponto flutuante, como um Microcontrolador ou um computador que disponha de placas de aquisição para realizar a emulação.

Devem ser realizadas provas para eliminar algumas variáveis temporais dos blocos gerados pelo HDL Coder, verificando o correto funcionamento do hardware, com o intuito de reduzir ainda mais a quantidade necessária de elementos lógicos. Não obstante isto requerera investir um tempo adicional para realizar esta modificação de maneira manual.

Finalmente se propõe a investigação dos métodos BQSS e LIQSS, com o intuito de reduzir as oscilações finais no sistema.

Capítulo 8

8. Conclusões e Sugestões para Trabalhos Futuros

8.1 Conclusões

Com o resultado do desenvolvimento deste trabalho foi realizada uma proposta para a modelagem dos sistemas que respondem a uma arquitetura híbrida, na qual contêm interações de variáveis físicas descritas em relação ao tempo e variáveis que respondem a eventos discretos. Baseado nesta proposta de modelagem foi proposto adicionalmente uma estrutura de controle aplicável nesse tipo de sistemas e a síntese dos mesmos utilizando lógica embarcada.

A proposta da modelagem e controle foi baseada nos formalismos DEVS e QSS a partir dos quais se conseguiu estruturar uma tupla que permite realizar a representação semântica de um processo que responda a um comportamento híbrido.

Ao mesmo tempo, a modelagem realizada facilita a representação de sistemas que por sua dinâmica resulta complexa, a interagir dinâmicas regidas em relação ao tempo e dinâmicas de eventos discretos, permitindo propor estratégias de controle que levem em consideração essa interação.

A proposta de modelagem inclui a utilização do método de integração QSS, com o propósito de resolver as Equações diferenciais que representam a dinâmica contínua do sistema, realizando-se uma análise comparativa com a utilização do método de integração de Euler, onde QSS permite uma melhor resposta do sistema de controle por estar baseado na digitalização em função de eventos.

Este trabalho permitiu realizar uma análise comparativa na implementação de controladores para variáveis contínuas baseado em eventos e controladores baseados em tempo discreto, onde

se conseguiu um melhor desempenho em função do custo computacional ao implementar o controlador baseado em eventos, pois requer uma quantidade de passos menor.

Com a modelagem realizada no sistema híbrido, foi realizada uma análise do modelo contínuo do sistema no que foi utilizado o método de integração QSS, encontrando um menor erro de estado estável do sistema, em relação ao modelo realizado utilizando o método de integração numérica de Euler.

A proposta de modelagem facilitou a síntese do modelo híbrido utilizando o conceito de hardware-in-the-loop e prototipagem rápida o que permite realizar testes mais rápidos e de mais baixo custo no desenvolvimento de um produto ou na automação de um processo industrial.

A proposta de modelagem realizada permite a simulação dos modelos utilizando ferramentas computacionais com Simulink® da Matlab®, permitindo realizar a validação numérica dos modelos e dos sistemas de controle projetados, além de permitir uma análise comparativa com outros formalismos de modelagem tradicional.

Esta proposta de modelagem e arquitetura de controle aplicado em sistemas híbridos tem um grande aporte na automatização de processos industriais da pequena e meia indústria, facilitando a representação da dinâmica de seus processos que são regidos em relação ao tempo e por eventos discretos, permitindo realizar testes rápidos e sua implementação final.

8.2 Sugestões para Trabalhos Futuros

Como resultado desse trabalho, novos trabalhos de pesquisa e desenvolvimento podem ser propostos. Entre as principais sugestões para pesquisas futuras derivadas do desenvolvimento deste trabalho estão:

- Proposta de novas arquiteturas de controle que leve em consideração a dinâmica híbrida dos sistemas, levando em conta esta proposta realizada.

- Uma análise da profundidade da estabilidade dos sistemas que estão conformados por dinâmica que respondem em relação tempo e a eventos discretos.
- Validação da proposta de modelagem em áreas diferentes à aplicada neste trabalho, relacionadas com a manufatura industrial.
- Validar o formalismo de modelagem proposto com outros formalismos como a redes de Petri os autômatos finitos e max- min-plus.
- Utilizar na síntese do modelo híbrido em sistemas embarcados com aritmética de ponto flutuante em vez de aritmética de ponto fixo, isto com o propósito de conseguir uma melhor definição na implementação.

Ao longo do desenvolvimento deste trabalho e como resultado dele foi publicado um trabalho científico em revista internacional indexada e dois trabalhos em anais de congresso internacional. Esses trabalhos se encontram no anexo I desta tese de doutoramento.

Referências

AARDAL, M. A model of CIM. In: ADVANCED SEMICONDUCTOR MANUFACTURING CONFERENCE AND WORKSHOP, Austin Texas. **ASMC 95 Proceedings IEEE/SEMI** 1995 p. 148-158

AIHARA, C. K. **Uma Abordagem Interativa para o Problema de Capacitação e Pesquisa em Automação**. 2005. 141p Teses (Doutorado). Faculdade de Engenharia Mecanica, Universidade Estadual de Campinas, Campinas.

ALMINDE, L.; BENDTSEN J.; JAKOB, S. A Quantized State Approach to On-line Simulation for Spacecraft Autonomy. In: MODELING AND SIMULATION TECHNOLOGIES CONFERENCE PROCEEDINGS, Keystone, Colorado, American Institute of Aeronautics and Astronautics, 2006.

ALMINDE, L.; BENDTSEN J.; JAKOB, S.; KRISTIN, P. Objective Directed Control using Local Minimisation for an Autonomous Underwater Vehicle. In: PROCEEDINGS OF IAV, Toulouse, France, 2007.

ALMINDE, L. A Quantised State Systems Approach Towards Declarative Autonomous Control. 2009. 223p. Teses (Doutorado). Department of Electronic Engineering. Universidade do Aalborg, Aalborg East.

ALURY, R., COURCOUBETISZ C. HALBWACHS, N; HENZIGER T; HOX, P; NICOLINZ, X; OLIVEROZ, A; SIFAKISZ, J; YOVINEZ, S. The algorithmic analysis of hybrid systems. Theoretical Computer Science, n.138, p. 3–34, 1995

ANTSAKLIS, P. J. Special issue on hybrid systems: theory and applications a brief introduction to the theory and applications of hybrid systems. **Proceedings of the IEEE** v.88, n.7, p. 879-887, 2000.

ARGENTE, E. **Guías Para El Desarrollo De Sistemas Multiagente Abiertos Basados En Organizaciones**. 2008. 428p. Tese. Departamento de Sistemas Informáticos y Computación Universidad Politécnica De Valencia, Valencia, España.

ÅRZÉN, K.-E. Simple event-based PID controller. In: PREPRINTS 14TH WORLD CONGRESS OF IFAC, Beijing, P.R. China, 1999

AYASUN, S.; FISCHL, R.; VILLIEU, S; BRAUN, J; CADIRLI, D. Modeling and stability analysis of a simulation–stimulation interface for hardware-in-the-loop applications. Simulation Modelling Practice and Theory. v.15, n.6, p.734-746, July 2007

BERGERO, F.; KOFMAN, E; BASABIBASO, C; ZUCCOLO, J. Desarrollo de un simulador de sistemas híbridos en tiempo real. In: XXI CONGRESO ARGENTINO DE CONTROL AUTOMÁTICO, Buenos Aires, Argentina. 2008.

BODE, H. W. Relations between attenuation and phase in feedback amplifier design. **Bell System Technical Journal,** n.19, p. 412–454, 1940.

BOYER, R.; FREYSSENET, M. **Los Modelos Productivos**. Madrid: Fundamentos, 2001. 160p.

CANALE, M.; FAGIANO, L.; RAZZA, V. Vehicle lateral stability control via approximated NMPC: real–time implementation and software–in–the–loop test. In: DECISION AND CONTROL, 2009 HELD JOINTLY WITH THE 2009 28TH CHINESE CONTROL CONFERENCE ON, Shangai, 2009. p.4596-4601.

CAPOCCHI, L.; BERNARDI F., et al. Transformation of VHDL descriptions into DEVS models for fault modeling. In: SYSTEMS, MAN AND CYBERNETICS, 2003. IEEE INTERNATIONAL CONFERENCE ON Corcéga, 2003, Universidade do Corsica, 2003. p.120-1210.

CAPOCCHI, L.; BERNARDI F., FEDERICI, D.; BISGAMBIGLIA, P. BFS-DEVS: A general DEVS-based formalism for behavioral fault simulation. **Simulation Modelling Practice and Theory**.v.14, n.7, p.945-970, 2006.

CASTELFRANCHI, C. Modeling Social Action For AI Agents. In: PROCEEDINGS OF THE FIFTEENTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICAL, 1997. P.1567-1576.

CASSANDRAS, C.; CLUNE, M.I.; MOSTERMAN, P.J. Hybrid system simulation with simevents. In: Discrete Event Systems, 2006 8th International Workshop on, Miami, 2006. p.286-287.

CASSANDRAS, C. G., LAFORTUNE S. **Introduction to Discrete Event Systems**, Springer, 2008.776p.

CELLIER, F. E. **Combined Continuous/Discrete System Simulation by Use of Digital Computers: Techniques and Tools ETH**. 1979 . Teses (Doutorado) Universidade do Zurich

CERVIN, A.; ASTROM,K. On limit cycles in event-based control systems. In: DECISION AND CONTROL, 46TH IEEE CONFERENCE ON. New Orleans, 2007.

DAI, L. Introduction to discrete event systems Book Review. **Automatic Control, IEEE Transactions on** v.46, n.2, p. 353-354, 2001.

DAVID, R.; ALLA, H. Continuous Petri nets. In: 8TH EUROPEAN WORKSHOP ON APPLICATION AND THEORY OF PETRI, NETS, Zaragoza, España, 1987. p.275-294

DE GENTILI, E.; DE CICCO A.; SANTUCCI, J. Devs and Fuzzy logic to model and simulate a manufacturing process. In: COMPUTATIONAL INTELLIGENCE FOR MODELLING, CONTROL AND AUTOMATION, 2005 AND INTERNATIONAL CONFERENCE ON INTELLIGENT AGENTS, WEB TECHNOLOGIES AND INTERNET COMMERCE, INTERNATIONAL CONFERENCE ON. Washington, DC, USA. 2005. p.557-564

DEL SIGNORE, M. J.; KROVI, V.; MENDEL, F. Virtual prototyping and hardware-in-the-loop testing for musculoskeletal system analysis. In: MECHATRONICS AND AUTOMATION, 2005 IEEE INTERNATIONAL CONFERENCE. Ontario, Canada. 2005. p.394-399

DE SOUZA, M. Abstração de Princípios da Competitividade a Partir da Releitura do Caso Toyota. XXIV ENCONTRO NACIONAL DE ENGENHERIA DE PRODUÇÃO. Florianópolis, SC, Brasil, 2004.

DORMIDO, S.; SÁNCHEZ, J.; KOFMAN, E. Muestreo, Control e Comunicaciones Basadas en Eventos. **Revista Iberoamericana de Automática e Informática Industrial**. v.5, n.1, p 5-26, 2008

EISEMANN, U. Guidelines for a model-based development process with automatic code generation. In: EMBEDDED WORLD CONFERENCE. Nuremberg, Germany. 2006.

ESPEJO, M.; MOYANO, J. Lean production: Estado Actual y Desafios Futuros de la Investigación. **Investigaciones Europeas de Direccion y Economia de la Empresa**. v.13, n.2, p.179-202, 2007.

EVANS, W. R. Graphical Analysis of Control Systems. **American Institute of Electrical Engineers, Transactions of the.** v.67, n.1, p. 547-551, 1948

GAITHER, N.; FRAZIER, G. **Administración de Producción y Operaciones**. México: Thomson, 2000.

GARCIA, J.; GOMEZ, R.; MIYAGI, P. Supervisory system for hybrid productive systems based on Bayesian networks and OO-DPT Nets. Emerging Technologies and Factory Automation, 2008. In: ETFA 2008. IEEE INTERNATIONAL CONFERENCE ON, 2008. p.1108-1111.

GAWTHROP, P; VIRDEN, D; NEILD, S; WAGG, D. Emulator-based control for actuator-based hardware-in-the-loop testing. **Control Engineering Practice.** v.16, n.8, p. 897-908, 2008.

GOEBEL, R.; SANFELICE, R., TEEL, A. Hybrid dynamical systems. **Control Systems, IEEE,** v.29, n.2, p.28-93, 2009.

GROUT, I. Digital System Desing whit FPGAs and CPLDs, Oxford: Newnes, 2008. 784p

HEEMELS, W.; LEHMANN, D; LUNZE,J; DE SCHUTTER, B. Handbook of Hybrid Systems Control. In: LUNZEN, Jan; LAMNABHI-LAGARRIGUE, Françoise (Ed.). **Introduction to hybrid systems**, Cambridge: University Press, 2009. chap. 2, p.27-66.

HEIZER J.; RENDER, B., **Operations Management**. Prentice Hall, 2001. 784p.

HEGNY, I.; WENGER, M.; ZOITL, A. IEC 61499 based simulation framework for model-driven production systems development. In: EMERGING TECHNOLOGIES AND FACTORY AUTOMATION (ETFA), 2010 IEEE CONFERENCE ON, 2010. 13-16 Sept. 2010. p.1-8.

HONG, K.; KIM, T. DEVSpecL: DEVS specification language for modeling, simulation and analysis of discrete event systems. **Information and Software Technology**, v.48, n.4, p. 221-234, 2006.

HOSSAIN, A.; SUYUT, S. Monitoring and controlling of a real-time industrial process using dynamic model control technology. In: DYNAMIC MODELING CONTROL APPLICATIONS FOR INDUSTRY WORKSHOP, 1997 IEEE INDUSTRY APPLICATIONS SOCIETY. 1997. p20-25.

HRÚZ, B.; ZHOU, M. C. **Modeling and Control of Discrete-event Dynamic Systems**. London: Springer. 2007.341p.

HU, X.; GANAPATHY, N.; ZEIGLER, B. Robots in the loop: supporting an incremental simulation-based design process. In: SYSTEMS, MAN AND CYBERNETICS, 2005 IEEE INTERNATIONAL CONFERENCE ON. 2005. p.2013-2018

ISERMANN, R. Optimal State Space Control of DC Motor. **Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul**, v.42, n.5, p.1070-1077, 2002.

ISERMANN, R.. Mechatronic systems--Innovative products with embedded control. **Control Engineering Practice** v.16 n.1 p.14-29, 2008.

KALPAKJIAN, S.; SCHMID, S. **Manufacturing and technolog**y, Prentice-Hall,2001. 1320p.

KATEEL G.; KAMATH M.; PRATT, D. An Overview Of Cim Enterprise Modeling Methodologies. In: WSC '96 PROCEEDINGS OF THE 28TH CONFERENCE ON WINTER SIMULATION, WASHINTONG, 1996.

KHAN, M. S.; IRAVANI, M. R. Supervisory Hybrid Control of a Micro Grid System. ELECTRICAL POWER CONFERENCE, 2007. EPC 2007. IEEE, Canada. 2007. p.20-24

KIM, K.; SEONG Y.; GON, T.; HO, K. Ordering of simultaneous events in distributed DEVS simulation. **Simulation Practice and Theory**. v.5, n.3, p.253-268, 1997.

KOCÍ, R.; JANOUSEK, V. OOPN and DEVS Formalisms for System Specification and Analysis. In: SOFTWARE ENGINEERING ADVANCES (ICSEA), 2010 FIFTH INTERNATIONAL CONFERENCE ON, 2010. p.305-310.

KOFMAN, E.; JUNCO S., Quantized State System. A Devs Aproach For Continuos System Simulation. Transactions of the Society for Computer Simulation International v.18, n.3, 2001.

KOFMAN, E. **Discrete event based simulation and control of continuous systems.** 2003. Teses (Doutorado) Faculdade de Ciências Exatas, Universidade Nacional de Rosário, Rosário Argentina.

KOFMAN, E. Discrete Event Simulation Of Hybrid Systems. **Society for Industrial and Applied Mathematics**. v. 25, n.5, p. 1771–1797, 2004

KOFMAN, E. Discrete Event Controlo of Time-Varying Plant. **Latin American Applied Research**, v.35, n.2,2005

LEBAIL, J.; ALLA, H.; DAVID, R. Hybrid Petri Nets. **Proc. 1st ECC, Grenoble** p.187–191, 1991

LIAPOUNOFF, A. **Problème général de la stabilité du mouvement.** Toulouse: Princenton Univ., 1907. 474p

LIHUI, W.; WEIMING, S. **Process Planning and Scheduling for Distributed Manufacturing**. London: Springer, 2007. 429p.

LINDSEY, M. Automatic test procedure generation from system specifications. In: RAPID SYSTEM PROTOTYPING, SHORTENING THE PATH FROM SPECIFICATION TO PROTOTYPE INTERNATIONAL WORKSHOP ON, Triangle Park, NC, 1992. p.301-310.

LIPIETZ A., El posfordismo y sus espacios las relaciones capital - In: TRABAJO EN EL MUNDO, SEMINARIOS INTENSIVOS DE INVESTIGACIÓN, Centro de Estudios de Investigaciones Laborales, Buenos Aires, 1994.

LUNZE, J.; LAMNABHI-LAGARRIGUE, F. Handbook of Hybrid Systems Control: Theory, Tools, Applications. Cambridge University Press, 2009. 582p.

LUO, B; ZHAO R. Rapid Control Prototype Design of the DC/DC Converter for Fuel Cell Electric Vehicles. POWER AND ENERGY ENGINEERING CONFERENCE, 2009. APPEEC 2009. Asia-Pacific, 2009. 27-31 March 2009. p.1-4.

LYGEROS, J.; GODBOLE, D.; SASTRY, S. A game-theoretic approach to hybrid system design. In: ALUR, R.;HENZINGER, T., et al (Ed.). Hybrid Systems III: Springer Berlin / Heidelberg, v.1066, 1996. p.1-12.

LYGEROS, J.; GODBOLE, D. N.; SASTRY, S. Verified hybrid controllers for automated vehicles. **Automatic Control, IEEE Transactions** on. v.43, n.4, p.522-539, 1998.

LYGEROS, J; JOHANSSON, K; SIMIC, S; JUN, Z.; SASTRY, S. Dynamical properties of hybrid automata." **Automatic Control, IEEE Transactions on**. v.48, n.1, p.2-17, 2003.

MAIR, A. C. The Second Industrial Revolution is World Wide. **Power Engineering Review IEEE.** v.4, n.12, p.15-17,1984

MARTIN, A.; SCOTT, E.; EMAMI, M. R. Design and Development of Robotic Hardware-in-the-Loop Simulation. In: CONTROL, AUTOMATION, ROBOTICS AND VISION, 2006. ICARCV '06. 9TH INTERNATIONAL CONFERENCE ON. Singapore. p.1-6

MAXFIELD, C. **FPGAs: World Class Designs**. :ELSEIVER, 2009. 488p.

MAXWELL, J. C. On governors. **Proceedings of Royal Society**. v.16, 1868

MIGONI, G.; KOFMAN E.. Linearly implicit discrete event methods for stiff ODEs **Latin Amer Appl Res**. v.39, n.3, p.245–254, 2009

MOSTERMAN, P. J.; BISWAS, G. A formal hybrid modeling scheme for handling discontinuities in physical system models. In: PROCEEDINGS OF THE THIRTEENTH NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1996, Portland, Oregon, p.985-990.

MOSTERMAN, P. J. HyBrSim - A Modeling and Simulation Environment for Hybrid Bond Graphs. **Journal of Systems and Control Engineering** v.special issue, p.35–46, 2002.

MOSTERMAN, P. J.; VANGHELUWE, H. Computer Automated Multi-Paradigm Modeling: An Introduction. **Simulation**,v.80, n.9, p.433-450, 2004.

MOSTERMAN, P. J. Hybrid Dynamic Systems: Modeling and Execution In: Paul A. Fishwick, (ed). **Handbook of Dynamic System Modeling.** 2007.

MUSTAFA, R.; MOHD M; UMAT, C.; AL-ASADY, D. Design and implementation of least mean square adaptive filter on Altera Cyclone II Field Programmable Gate Array for active noise control. In: INDUSTRIAL ELECTRONICS & APPLICATIONS, 2009. ISIEA 2009. IEEE SYMPOSIUM ON, Kuala Lumpur, 2009. p.479-484.

NABI, S.; BALIKE, M.; ALLEN, J.; RZEMIEN, K. An Overview of Hardware-In-the-Loop Testing Systems at Visteon. SAE WORLD CONGRESS, Detroit, Michigan, 2004

NICA, M.; GANEA, L.; DONCA, G. Simulation Of Queues In Manufacturing Systems, Annals Of The Oradea University. **Fascicle of Management and Technological Engineering,** v.7, 2008.

NIKOLAIDOU, M.; DALAKAS, V.; MITSI, L.; KAPOS, G. A SysML Profile for Classical DEVS Simulators. In: SOFTWARE ENGINEERING ADVANCES, 2008. ICSEA '08. THE THIRD INTERNATIONAL CONFERENCE ON. 2008, Washington, p.445-450.

OROOJI, F.; SARJOUGHIAN, H.; TAGHIYAREH, F. Modeling & simulation of educational multi-agent systems in DEVS-suite. In: TELECOMMUNICATIONS (IST), 2010 5TH INTERNATIONAL SYMPOSIUM ON. Theran. 2010. p956-961.

PALANIAPPAN, S.; SAWHNEY, A. SARJOUGHIAN, H. Application of the DEVS Framework in Construction Simulation. In: PROCEEDINGS OF THE 2006 WINTER SIMULATION CONFERENCE ON, 2006. p. 2077-2086.

PAOLUCCI, M,; SACILE R. Agent-Based Manufacturing and Control Systems. CRP PRESS,2005. 288p

PASCHALAKIS, S.; LEE, P.; Double Precision Floating-Point Arithmeticon FPGAs. In: PROCEEDINGS 2003 IEEE INTERNATIONAL CONFERENCE ON FIELD PROGRAMMABLE TECHNOLOGY (FPT '03), Tokyo, Japan, 2003, p. 352-358.

PEPYNE, D.L.; CASSANDRAS, C.G. Optimal Control of Hybrid Systems in Manufacturing. **Proceedings of the IEEE**,v.88, n.3, p.398-415,2000.

PIAO, H.; HUANG, J.; WANG, C. The Process Modeling and Integration for JIT Automotive Supply Logistics Based on IDEF9000. INTERNATIONAL CONFERENCE ON AUTOMATION AND LOGISTICS. IEEE, P. O. T. Jinan, China, 2007. p.215-219

PRAEHOFER, H.; WAELTERMANN, P. Hardware-in-the-Loop Testing of Vehicle Dynamics Controllers – A Technical Survey. **SAE**, v.1, 2005.

RAJADELL, M.; SÁNCHEZ, J. **Lean Manufacturing: La evidencia de una necesidad**. Cataluña: Ediciones Díaz de Santos, 2010. 260p.

RAMADGE, P.; WONHAM, W. The Control of Discrete Event Systems. **Proceedings of the IEEE**, v.77, n.1, p.81- 98, 1989

RAY, A. . Networking for computer-integrated manufacturing. **Network IEEE**, v.2, n.3, p. 40-47, 1988

REZAIE, K.; SHIRKOUHI, S.; ALEM, S. Evaluating and Selecting Flexible Manufacturing Systems by Integrating Data Envelopment Analysis and Analytical Hierarchy Process Model. In: MODELLING & SIMULATION, 2009. AMS '09. THIRD ASIA INTERNATIONAL CONFERENCE ON. 2009. p.460-464.

RIFKIN, J. "The third industrial revolution. **Engineering & Technology**, v.3, n.7, p.26-27, 2008.

RÖCK, S. Hardware in the loop simulation of production systems dynamics. **Production Engineering**, p.1-9,2011.

ROCHA, J.; RAMOS, C.. Task planning for flexible and agile manufacturing systems. In: Intelligent Robots and Systems '94. Advanced Robotic Systems and the Real World, IROS '94. PROCEEDINGS OF THE IEEE/RSJ/GI INTERNATIONAL CONFERENCE ON. Munich, 1994. p.105-112

ROSENBROCK, H. H. Control: past, present and future. **Radio and Electronic Engineer,** v. 37, n. 1, p. 30-32, 1969.

ROUTH, E. J. **A treatise on the stability of a given state of motion.** London: MacMillan, 1877. 108p.

RUSSO, R. TERZO, M. TIMPONE, F. Software-in-the-loop development and validation of a Cornering Brake Control logic. **Vehicle System Dynamics**, v. 45, n.2, p.149-163, 2007.

SAMAD, T. Perspectives in Control Engineering Technologies, Applications, and New Directions. **IEEE Robotics & Control Systems**, p.165-188, 2000.

SAVORY, P.; MACKULAK, G.; COCHRAN, J. Material handling in a flexible manufacturing system processing part families. In: SIMULATION CONFERENCE, 1991. PROCEEDINGS., WINTER, 1991. p.375-381.

SCHLAGER, M.; ELMENREICH, W.; WENZEL, I. Interface design for hardware-in-theloop: simulation. In: PROCEEDINGS OF THE IEEE INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS (ISIE'06) 2006. p. 1554–1559.

SHIN, M.; MUN, J.; JUNG, M. Self-evolution framework of manufacturing systems based on fractal organization. Computers & Industrial Engineering, v. 56, n. 3, p. 1029-1039, 2009.

STIVER, J. A.; ANTSAKLIS, P. J. Modeling And Analysis Of Hybrid Control Systems. DECISION AND CONTROL, 1992. In: PROCEEDINGS OF THE 31ST IEEE CONFERENCE ON. 1992, p. 3748-3751.

THUE, L. From Large Technical Systems To Technological Complexes. In: HISTORY OF TELECOMMUNICATIONS CONFERENCE, 2008. HISTELCON 2008. IEEE. 2008. p.52-55.

VAN DER SCHAFT, A.; SCHUMACHER, H. **An Introduction to Hybrid Dynamical Systems**. London: Springer, 2000.

VON NEUMAN J. Theory of Self-Reproducing Automata. **University of Illinois Press, Illinois. Edited and completed by A.W. Burks.** 1966.

WAINER, G. A. **Discrete-Event Modeling And Simulation: A Practitioner's Approach.** Boca Raton, FL.: CRC Press, 2009. 520p.

WANG, S.; ZHANG, J.; LIU, W. A Novel Flexible System for Denture Design and Manufacture. In: COMPLEX MEDICAL ENGINEERING, 2007. CME . IEEE/ICME INTERNATIONAL CONFERENCE ON, 2007. p. 1965-1969.

WITSENHAUSEN, H. A class of hybrid-state continuous-time dynamic systems. Automatic Control, IEEE Transactions on, v.11, n.2, p.161-167, 1966.

ZEIGLER, B.; VAHIE, S. Devs formalism and methodology: unity of conception/diversity of application. In; PROCEEDINGS OF THE 25TH WINTER SIMULATION CONFERENCE. Washington, 1993.

ZEIGLER, B.; MOON, Y.; LOPES, V.; KIM, Y. DEVS approximation of infiltration using genetic algorithm optimization of a fuzzy system. Mathematical and Computer Modelling, v.23, n.11, p.215-228, 1996.

ZEIGLER, B.; PRAEHOFER, H.; KIM, T. Theory of Modeling and Simulation. New York: Academic Press, 2000. 510p.

ZEIGLER, B. DEVS today: recent advances in discrete event-based information technology. In: MODELING, ANALYSIS AND SIMULATION OF COMPUTER TELECOMMUNICATIONS SYSTEMS, 2003. MASCOTS 2003. 11TH IEEE/ACM INTERNATIONAL SYMPOSIUM ON. 2003. p.148-161.

ZHAO, Q.; CHEN D. The Research to Power SCADA Based on J2EE Framework. In: INFORMATION ENGINEERING, 2009. ICIE '09. WASE INTERNATIONAL CONFERENCE ON, 2009. p.435-438.

Anexo A- Trabalhos Publicados

EVENT-BASED CONTROL FOR A SECOND ORDER CONTINUOUS SYSTEM

Dario Amaya H.[1], João M. Rosário[2], Deivy J. Mayorquin[3]

1. Nueva Granada Military University, Dario.amaya@unimilitar.edu.co
2. State University of Campinas, rosario@fem.unicamp.br
3. Centre for Research and Technological Development of the Electric Electronic and Informatics Industry – CIDEI, dmayorquin@cidei.net

ABSTRACT

Traditionally modern industrial control systems tend to be implemented using digital platforms, therefore it is necessary the discretization of systems, which is done mostly through a temporal discretization, reaching the so-called discrete-time systems, which evaluates the change of system variables synchronously in specific instants of time. However, in this type of discretization, there is the difficulty of evaluating changes in the signal from sample to sample. This is where the event-based control, which performs an asynchronous discretization, becomes a promising tool in applications requiring a low computational cost. In this work we make a comparative analysis of discrete-time control, based on Euler's method to perform discretizationand event-based control. We present a case study based on the implementation of a discrete-time control applied to a second-order system, comparing the results using anevent-based control system.

KEY WORDS

event-based control systems, PI control, discrete-time systems.

1. INTRODUCTION

Being the most of the industrial processes analog variables, currently the majorityof automatic control systems use digital technologyarchitecture in the implementation of their structures, thanks to an approximation of analog systems into numerical systems through analog to digital converters, traditionally samplingsignals synchronously, using fixed times toobtain a discrete time system [3]. However, the difficulties to implement control systems using synchronous sampling, lies in the loss of process control at intervals of two successive samples[1]. In this topicis where we have been working with a new philosophy,based on an event-based sampling, which defines the time at which the analog signal sample is made at the instant of a certain level crossing[4].

There are several advantages of performing the control of an event-based process, among these are that event-based controller is closer to reality, as well as that when implementing a controller on aCPU, the fact ofload the CPU when nothing significant occurs, generates unnecessary computational costs [7].

Between some of the applications that use the concept of event-based control, is the one made by[5], who presented a proposal for an event-based PID controller, which was modified by[4], taking into account only the quantification error to determine the sampling instants in which are executed the control actions. This algorithm is the fundamental basis for the control strategy implemented in this work.

[6]Performed an event-based control in order to maintain a stable temperature in a greenhouse, for which used a wireless sensor network and presented it as a case study, managing to obtain a significant result in reducing the number of switching operations performed by the controllerat a value close to 80% compared to the traditional discrete-time controller.

[7]Developed an event-based PID controller in order to guarantee low computational cost in the implementation. This application that was based on the algorithm proposed by[5]and modified by[4], made a contribution to minimize the sampling interval in order to reduce the

sample number in the transientsthat the system presented, whichallowed them to reduce the margin of error in asteady-state interval.

[8]Presented two complementary approaches to event-based control in order to ensure a final limitationin a control loop. These methods are based on consideration of the feedback of states taking into account that the control loop structure is based on events. [9]Presented the application of an event-based controller applied to a system of two degrees of freedom. The architecture proposed is event-based control, which works uncoupled for the purpose of tracking the operating point and rejecting load disturbances. There were three aspects that characterized thisapplication, the absence of asynchronizationclockin the control strategy, the reduction in the number of calculations of control actions and the independence of the rules in the controller tuning.

Although there are still few works in event-based control compared to traditional control based in the discretization of systems using synchronous samplingand specifically based on the method of Euler, for the discretization of systems. It is of growing interest in areas such as control of combustion engines, speed controlwherethe measuring element is an encoder, wireless measurement systems, among others.

Outline of the paper

The first part is an introduction to event-based control systems and a review of work done in this area, in the second part there is a theoretical review of event-based control systems and specifically the PID controller proposed by [5] and modified by [4]. In the third part thereis performed a case study of design and simulation of an event-based PID applied to a second-order system. In the final part, conclusions are made and theused references are presented.

2. EVENT-BASED CONTROL SYSTEMS

Event-based control systems are characterized by having a digitalization of analog systems using an analog to digital conversion that unlike synchronous converters are based on the occurrence of an event that lets you decide the time that should make the capture of the controlled variable [2].This allows you to set the time at which the variable changes its value, reducing the response times of the controller and sometimes avoiding unnecessary evaluations of the variable as occurs when using synchronous sampling. An example of the difference in synchronous and asynchronous conversion is shown in Figure 1.

In the synchronous conversion, the samplingis performed every time you fulfill a time k, which has fixed steps; in the asynchronous conversion, a sample of the signal is taken every time it changes its magnitude in a value exceeding Q. As it can be seen in some sampling periods using the discrete time, it is not possible to distinguish some of the variable changes, as shown from the time period 0 to 1k and in some sampling periods it is unnecessary to evaluate the signal, because the variable does not has significant changes in the case of the sampling periods from 2 to 5.

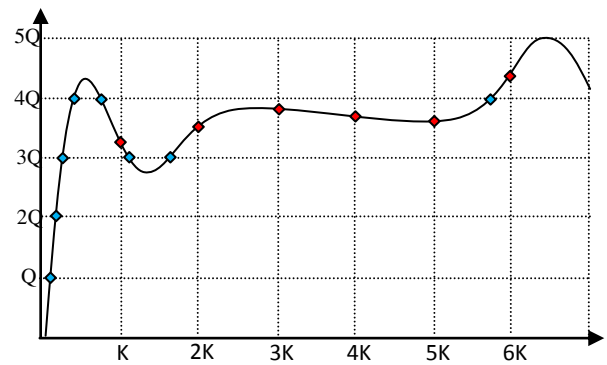

Figure 1.Synchronous and asynchronous discretization.

The controller implemented as control structure is a PI, represented by equation 1 in continuous time.

$$U(s) = k_p(1 + \frac{1}{sT_i(s)})E_s \qquad 1$$

Where $E_s$represents the system error signal corresponding to the difference between setpoint and actual value of the variable. The difference between the current error signal and the error signal after the last control action defines the event that lets to decide a new control action, provided that this difference varies in an absolute value> ΔQ, as shown in Equation 2.

$$|e(t) - e(t_k)| > \Delta Q \quad 2$$

Wheree(t) is the current error signal and $e(t_k)$ is the error signal of the last step, i.e. the error signal obtained after the last control action.

Based on Equation 1, we proceeded to perform thediscretization of the PI controller and make adjustments that alter the conversion of analog to digital signal based on the equation 2. The code implemented for PI controller has the following structure.

```
//Reading the system output and Set Point
y = ADIn(ch1);
ysp = ADIn(ch2);
```

```
//Error Calculation

e = ysp – y;
IF (abs(e – es)>Q) THEN
es = e;
//Asses of the control action

up = Kp*e
u = up + ui
//Send control action to the output

DAOut(u,ch3)
//Update status by calculating the integral action

ui = ui + (ti - ti-1)*Ki*e
ENDIF;
```

3. CASE STUDY

As a case study, we designed an event-based PI controller applied to a second-order system. We also carried out a comparative analysis between the responses of a continuous-time controller, adiscrete-timecontroller and an event-based controller, in order to determine the efficiency compared to impulse, the steady-state error and the number of steps required for implementation of discrete-time controller from the event based controller.

The second-order system that was used in this work is an electric motor which was obtained by the analysis of the power represented as a series RL circuit, followed by a voltage source dependent of the angular velocity of motor, this is shown in Figure 2.

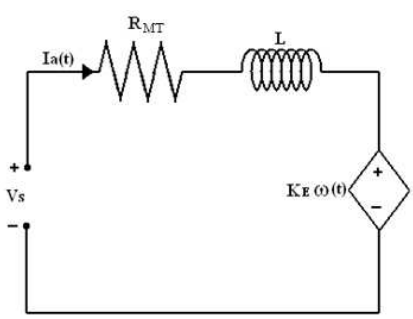

Figure 2. Electrical model of a DC motor

The electrical equation of the system using the Kirchhoff voltage law is:

$$V_S(t) = i_a(t)R_{MT} + L\frac{di_a(t)}{dt} + K_E\omega(t) \qquad 3$$

Where R_{MT} is the rotor winding resistance, L is the inductance and $K_E$corresponds to the motor voltage constant, which is related to the EMF produced due to its angular velocity.

The motor mechanical part is represented as in Figure 3, which shows an inertial load that corresponds to that produced by the rotor, a viscous friction corresponding to the friction of the brushes (thin) with the rotor and friction shaft with the motor housing and a torque load that is independent of the angular velocity of motor, so it should be considered as a second input to the system.

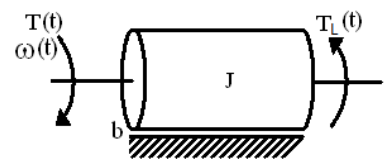

Figure 3.Mechanical representation of a DC motor.

The mechanical equation of the system is defined by equation 4.

$$T(t) = J\alpha(t) + b\omega(t) + T_L(t) \qquad 4$$

Where J corresponds to the inertia of the rotor and α(t) is the angular acceleration. The torque produced by the motor is proportional to the current of the winding, thus replacing the mechanical equation of the system and neglecting the friction torque is obtained theequation 5.

$$K_T i_a(t) = J\alpha(t) + b\omega(t) + T_L(t) \qquad 5$$

Where KT is the torque constant of the motor.Choosing as state variables rotor current $x_1(t)=ia(t)$and the angular velocity of motor $x_2(t)=\omega(t)$, we get the system of equations represented by 6.

$$\dot{x}_1(t) = \frac{di_a(t)}{dt}$$
$$\dot{x}_2(t) = \alpha(t) \qquad 6$$

Substituting into the mechanical and electrical equations yields the following system equations are represented in 7.

$$V_S(t) = x_1(t)R_{MT} + L\dot{x}_1(t) + K_E x_2(t)$$
$$K_T x_1(t) = J\dot{x}_2(t) + bx_2(t) + T_L(t) \qquad 7$$

Solving higher order variables, it is possible to find out that the system state equations can be represented by 8.

$$\dot{x}_1(t) = \frac{1}{L}V_S(t) - \frac{R_{MT}}{L}x_1(t) - \frac{K_E}{L}x_2(t)$$

8

$$\dot{x}_2(t) = \frac{K_T}{J_R}x_1(t) - \frac{b}{J_R}x_2(t) - \frac{T_L(t)}{J_R}$$

The output y (t) of the system is the angular velocity:

$$y(t) = x_2(t)$$

Expressing the equations in matrix form:

$$\dot{\bar{x}}(t) = \begin{bmatrix} -\frac{R_{MT}}{L} & -\frac{K_E}{L} \\ \frac{K_T}{J} & -\frac{b}{J} \end{bmatrix}\bar{x}(t) + \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix}\begin{bmatrix} V_S(t) \\ T_L(t) \end{bmatrix}$$

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix}\bar{x}(t)$$

Replacing the variables by parameters given by the engine manufacturer:

$$L = 1{,}02mH \qquad R_{MT} = 1{,}26\Omega$$

$$K_E = 0{,}0232\frac{V}{rad/s} \qquad K_T = 0{,}0232\frac{Nm}{A}$$

$$J = J_R = 4{,}2\times10^{-6}\,Kg\cdot m^2 \qquad b = D = 2{,}6\times10^{-6}\frac{Nm}{rad/s}$$

The continuous model of the plant is defined by:

$$\dot{\bar{x}}(t) = \begin{bmatrix} -1235{,}29 & -22{,}74 \\ 5523{,}81 & -0{,}62 \end{bmatrix}\bar{x}(t) + \begin{bmatrix} 980{,}39 & 0 \\ 0 & -0{,}238\times10^6 \end{bmatrix}\begin{bmatrix} V_S(t) \\ T_L(t) \end{bmatrix}$$

Figure 4 shows the plant response to a step of 12V without load. It is desirableto see the output in rpm instead of rad/s, therefore it is scaled by a value of 0.0437, according to the conversion formula and thereduction ratio from the manufacturer's specification sheet (30/(π*218.4) = 0.0437).

According to Figure 4, for purposes of the synthesis of the PI controller, the plant can be approximated as a second-order system. From the graph, we calculated the constants of the PI, in this case the used technique was Ziegler-Nichols tuning, obtaining the constants:

$$k_p = 2{,}52 \qquad k_i = 600$$

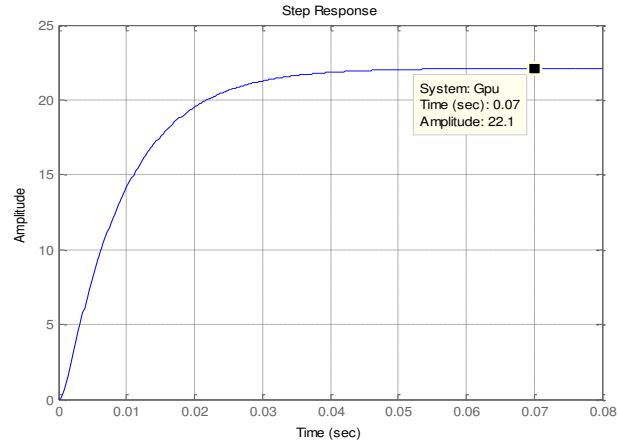

Figure. 4. Motor Speed Output

The output of the controller-plant system, with these parameters is shown in Figure 5.It can be seen that the output has a 70% reduction in the stabilization time (2%), but the overshoot is 20%. Therefore, the parameters kp and ki are adjusted, reducing them to 2 and 300, respectively. With these new parameters, the output of the system (Figure 6) has an overshoot less than 10% (6.5%) and a stabilization time (2%) of 14ms, 62% lower than the stabilization time of the plant in open loop.

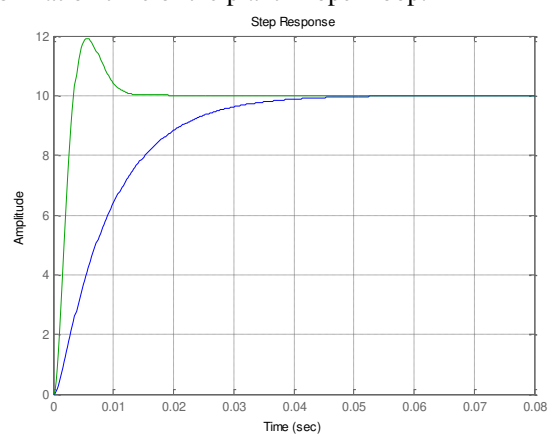

Figure 5.Output of the plant with the PI controller.

With the obtained controller parameters , the simulation of the system is performed using two digital controllers, the first onebased on time discretization and the second onebased on the quantification of the error, ie, based on discrete events, for which the algorithms of event-based PI control of the part 2were used. This was made in order to conduct a comparative analysis between the two control strategies and clearly identify the advantages and disadvantages of implementing a control system based on events on a plant of rapid response, such as a DC motor.

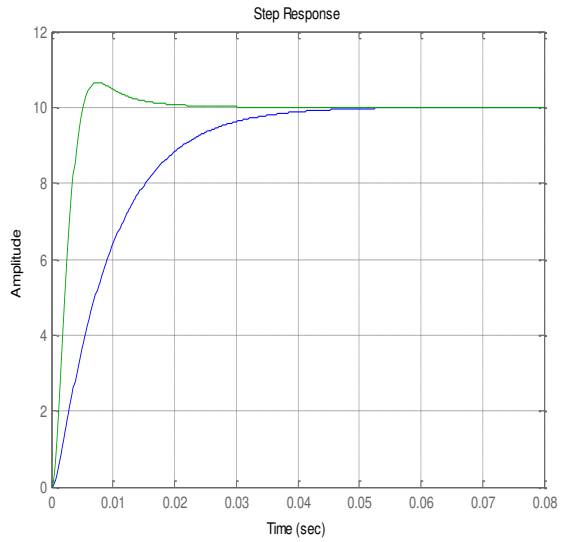

Figure 6.Output of the plant with the adjusted control

The parameter governing the behavior of time-based controller is the sampling period, while in the event-based controller is the quantification of error interval. The time-based digital controller usesa sampling period of 200µs and the event-based digital controller is working with a quantization for the error signal of 0.02 rad/s. The simulation time is 40ms. This quantization value is selected in order to obtain a steady state error less than 0.1 rad/s, i.e. less than 1% regard the set point.

As an instrument of comparison, we analyze the system step response with each controller, using as reference the response to the continuous controller Figure 7.

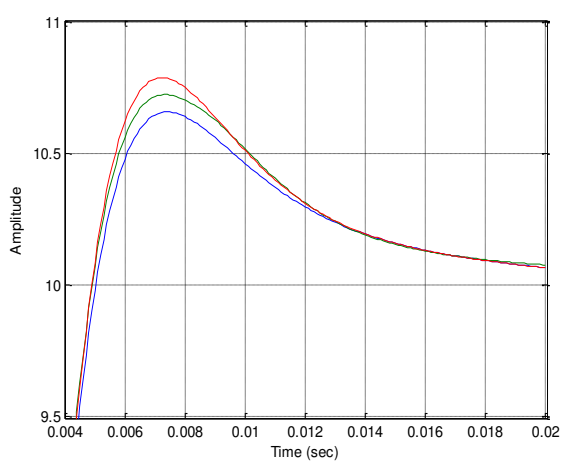

Figure7. Output of the plant with the three controllers.

In the figure 7, the blue lines is the output of the plant with the continuous controller, the green line is the output using the event based controller and the red line is the output using the discrete time based control. In this figure it is possible to note that, for the established conditions of sampling time and quantification interval, the plant with event based digital control have a faster response with less overshoot than the discrete time based digital control. In addition the asses of the required steps for the implementation of the event based controller and the discrete time based controller was made; finding a greater computational cost in this last one.

Table 1 summarizes the results of this simulation, which also includes the number of steps performed by the two digital controllers. It is important to observe, than besides presenting a smaller overshoot, event-based controller requires less than half of steps that the discrete-time controller to make the output of the plant to the reference value with a steady-state error of 0,03%. This error can be accepted, bearing in mind that such a significant reduction in the control calculations, as presented in this case the event-based digital controller, means a significant saving of energy and computational costs.

Table 1.Simulation results of the figure 6.

	Controller		
	Continuous	**Event-based**	**Discrete time**
Mp	6,6%	7,2%	8,0%
ts (2%)	13,9ms	13,8ms	13,9ms
tr (100%)	5,02ms	4,91ms	4,89ms
Steps	---	99	200
SSE	0%	0,03%	0%

If we extend the simulation time, the time based controller would keep calculating control actions, which would add more steps, while the event-based controller would remain pending of the disturbances in the system, therefore, not makingadditionalcalculations. This is shown in Figure 8, which shows the controllers simulation output.

Figure 8 shows the last 3 steps of the event-based controller (green line) and it is possible to see that while it made significant adjustments in irregular time intervals, the time-based digital controller (red line) takes a lot of small actions to control a fixed period. This represents a major disadvantage for the classic controller compared to discrete-time event-based controller, due to the high computational cost which involves the periodic calculation of control actions, because it is done even when the system output value has reached the stable.

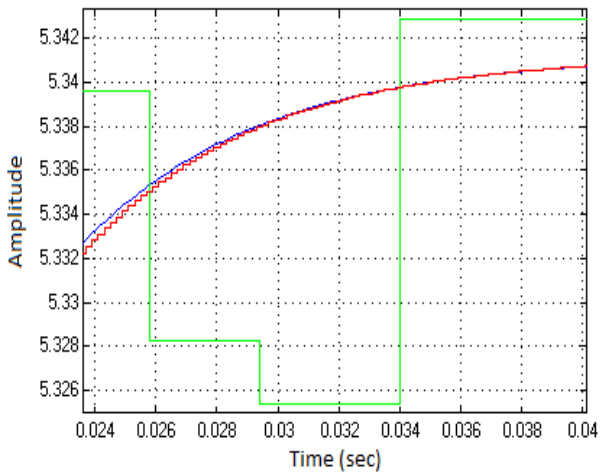

Figure. 8. Controllers Output

In order to reduce the number of steps performed by the discrete-time controller, it increases to 800µs sampling period. With this adjustment, the time-based digital controller (red line) takes 50 steps during the first 40ms of the simulation, however the output of the plant responds more slowly at the beginning and has an overshoot of 13%, as shown in Figure 9.

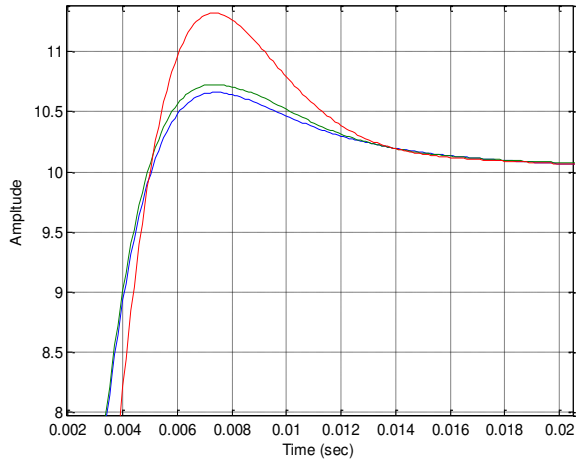

Figure 9.Output of the plant with the three controllers.

4. CONCLUSION

This work allowed a comparative analysis of the implementation of event-based controllers and discrete-timecontrollers, where we could get a better cost-performance computing by implementing an event-based controller, because it requires fewer steps.

In addition to the advantage of event-based controller in the computational cost, the simulations allowed to observe, that it is possible to obtain a steady state error pretty close to the error obtained when using a discrete time controller.

The simulation of the controlled system allows us to observe that the event-based controller has a faster transient response than the discrete-time controller, furthermore, the second one presents a higher overshoot, which could be reduced by using a faster sampling time, which would increase even more the computational cost.

The event-based controller has some disadvantages to consider. This controller has required less control actions than the discrete-time controller; however, the number of steps in the dynamic portion of the output signal is higher due to the rapid change of the error regard the quantization interval. There is another disadvantage and is the steady state error, however it is very small (less than 0.5% in this case) and it could be accepted if one considers that the event-based controller does not executes control actions in steady state.

As future work we purpose the hardware implementation of the event-based PI controller to verify its performance under a real work environment, considering that the simulation models are based on physical parameters of such a real system as the Pittman D.C. motor.

Some work must be also done in order to compare the event-based controller with other discrete time algorithms, different than the algorithm based on Euler approximation we used for comparison purposes in this paper. Furthermore, it is important to model different control structures like state feedback and other kind of compensators, using asynchronous sampling to evaluate the performance of event-based sampling applied to different controller architectures.

5. REFERENCES

[1] E. Kofman, Discrete event based simulation and control of continuous systems (Ph.D. Dissertation thesis. Faculty of Exact Sciences, National University of Rosario, Argentina, 2003).

[2] S. Dormido, J. Sánchez, E. Kofman, Sampling, event-based control and communication (in Spanish). RevistaIberoamericana de Automática e Informática Industrial, 2008.

[3] k. J. Astrom, W. Bjorn, Computer-controlled systems, theory and design, Prentice-Hall information and system sciences series,1997.

[4] S. Durand, N. Marchand, Further Results on Event-Based PID Controller, European Control Conference, 2010.

[5] K. E. Arzén. A simple event-based PID controller.In Preprints of the 14th World Congress of IFAC, Beijing, P.R. China, 1999.

[6] A. Pawlowski, J.L. Guzman, et al, Event-based control and wireless sensor network for greenhouse diurnal temperature control: A simulated case study, IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008), 2008.

[7] S. Durand, N. Marchand, An EventBased PID Controller With Low Computational Cost, 8th International Conference on Sampling Theory and Applications (SampTA'09), 2009.

[8] L. Grüne, S. Jerg, O. Junge, D. Lehmann, J. Lunze, F. Müller, M. Post, "Two Complementary Approaches To Event-based Control", at-Automatisierungstechnik, 58(4): 173-182, 2010.

[9] J. Sáncheza, A. Visioli, S. Dormido, A two-degree-of-freedom PI controller based on events, Journal of Process Control, 2010.

Modeling and Simulation of Hybrid Architecture Manufacturing Systems

Dario Amaya H.[1], Ricardo A. Castillo[2], João M. Rosário[3.]

Abstract – This paper presents a methodology proposal for hybrid architecture processes modeling, it takes into account concepts of both DEVS (Discrete Event System Specification) formalism and QSS (Quantized State System) discretization technique to obtain an event based representation of a complete mechatronic system. Initially this work briefly describes the basics of the two formalisms used in the proposed methodology. In this methodology the system is dynamically evaluated in order to identify its continuous and discrete parts, which are afterward mathematically represented and engaged as modules in a single model able to be subsequently simulated. This proposal is validated through the modeling of a work station responsible for supply and transportation operations in a production system, within it is possible to identify variables that change continuously over time besides others whose behavior is defined by the occurrence of events, in such way constituting a hybrid architecture system. Finally, the obtained events model is simulated (Stateflow™, Matlab-Simulink™) showing the advantages of an events based methodology for systems modeling and simulation respect to other ones that take only continuous time into account.

Keywords: Hybrid Systems Modeling, DEVS, QSS.

NOMENCLATURE

$\boldsymbol{H}$: DEVS atomic.	$\boldsymbol{R_a}$: Armature resistance
$\boldsymbol{X}$: Set of input events.	$\boldsymbol{L_a}$: Armature inductance
$\boldsymbol{Y}$: Set of output events.	$\boldsymbol{k_v}$: Constant Voltage
$\boldsymbol{S}$: Set of sequential states.	$\boldsymbol{V_a}$: Armature Voltage
$\boldsymbol{\delta_{int}}$: An internal transition function.	$\boldsymbol{k_t}$: Constant de torque
$\boldsymbol{\delta_{ext}}$: External transition function.	$\boldsymbol{J}$: Inertia
$\boldsymbol{Q}$: Total state set.	$\boldsymbol{B}$: Friction
$\boldsymbol{\lambda}$: output function	$\boldsymbol{T_L}$: Load torque
$\boldsymbol{t_a}$: a time advance function	$\boldsymbol{T}$: Applied Torque
$\boldsymbol{i_a}$: Armature current	$\boldsymbol{w_a}$:Rotor speed

I. INTRODUCTION

So far technology is changing rapidly. This evolution is reflected mainly in computation, communication and manufacturing systems, consequently some new methodologies for modeling and analyzes this kind of events based systems have emerged. Such systems are characterized by the capacity to performing real time tasks with high efficiency and accuracy. Thus, any representation of these characteristics must by associated to modeling methods based in systematic computational tools supporting the development and simulation of these process models and coping with this type of behavior [1].

The concept of a system model has been traditionally associated with the derivation of a set of differential equations, which describes the identified variables and the relation among them through physical laws governing the system dynamic. This approach is typically employed in order to describe continuous time varying systems. However to obtain the full description of many mechatronic processes it should be identified and considered variables that change depending on events occurrence in addition to continuous variables that change over the time.

Although these two types of processes in a system are conventionally modeled using an independent and disjunctive form, in a variety of systems there is a strong interaction and even a dynamic dependence between variables belonging to continuous and discrete parts. Such systems are known as hybrid architecture systems [6]. Hence, a new research area focused on the coordinated behavior the two types of processes within a system arise, this area is called: hybrid dynamic systems [4]. Hybrid systems are defined as those where is possible to identify

state variables as with continuous behavior as described by occurrence of discrete events.

Figure 1 shows the structure of a hybrid architecture dynamic system in which there is a continuous dynamic interacting with a discrete dynamics through entry and exit events

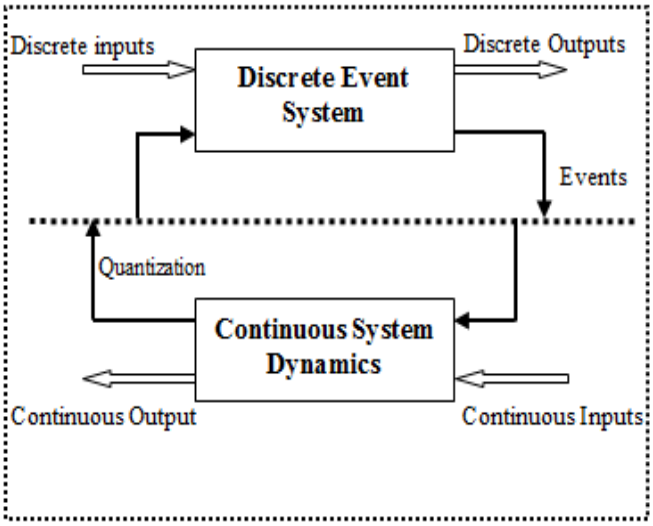

Figure 1 Hybrid System

The research on hybrid systems have two origins that eventually converged at the same point: studies on continuous-time systems that initiated the incorporation of discrete events, and studies on discrete event dynamic systems that in the same way started incorporating continuous variables [7], [5]. This is clearly observed in industrial production systems, which can be defined as a set of interconnected cells that collaborate to achieve a certain goals, e. g. receipt of raw materials to transform them into pieces or assembled products.

Such systems (pneumatic or hydraulic manipulators, end of course sensing systems, etc) according to its application are composed by cells which are controlled through events based variables. Similarly some of the cells forming a production system involve the control of variables that change over time (temperature, flow, pressure, pH, etc), hence the modern production systems require modeling and simulation methodologies that enable the development of control systems taking into account the concept of hybrid control. Several formalisms have been developed to analyze and simulate hybrid systems. One of the most recently used is DEVS (Discrete EVent Specification) formalism [2], [3].

The rest of paper is organized as follows: Section II presents an introduction to DEVS and QSS formalisms. Section 3 describes the modeling and simulation methodology for hybrid system developed in this work. Section IV discusses the validation of the methodology trough a case study. Finally, Section V rounds up the paper with conclusions and directions for future work.

II. DESCRIPTION OF USED FORMALISMS

Traditionally tools as differential equations and equations in differences have been used to represent the behavior of systems described in relation to time. Currently taking into account the technological development have appeared many systems with a behavior based on the occurrence of asynchronous events, as computers, transportation, manufacturing and communications systems [10]. This type of systems is typically called a Discrete Event Systems (DES), and its analysis can be said complex due to a variety of requirements which includes proprietary functions, desired performance, time response limitations, etc [3].

II.1. Discrete Event Specification (DEVS)

In 1972 the mathematician Bernard Zeigler proposed a formalism called DEVS (Discrete EVent Specification) in order to represent DES [11]. In [12] this formalism is described as an environment to specify the model and simulation of discrete event systems in a hierarchical modular manner, with the formalism the model of complex systems can be obtained easily by the coupling of smaller sub-systems models, according to a specifying form.
The smaller models are defined as atomic models, representing "molecular" processing units, which constitute the fundamental elements to construct the complete model in complex systems [13], [14], [8].

An atomic model can be defined as the 7-tuple:

$$\mathbf{H} = (\mathbf{X}, \mathbf{Y}, \mathbf{S}, \boldsymbol{\delta}_{\mathbf{int}}, \boldsymbol{\delta}_{\mathbf{ext}}, \boldsymbol{\lambda}, \mathbf{t_a}) \quad \textbf{(1)}$$

Where:

X : Set of input events;
Y : Set of output events;
S : Set of sequential states;
$\boldsymbol{\delta}_{\mathbf{ext}}$**:** $\mathbf{Q}$ x X $\rightarrow$ S, an external transition function, where **Q is the total state set of** $\boldsymbol{M} = \{(\boldsymbol{s}, \boldsymbol{e}) | \boldsymbol{s} \in \boldsymbol{S}$ and $\mathbf{0} \leq \boldsymbol{e} \leq \boldsymbol{ta}(\boldsymbol{s})$
$\boldsymbol{\delta}_{\mathbf{int}}$: $\mathbf{S} \rightarrow \mathbf{S}$, an internal transition function;
$\boldsymbol{\lambda}$: $\mathbf{S} \rightarrow \mathbf{y}$, an output function;
$\mathbf{t_a}$: $\mathbf{S} \rightarrow \mathbf{R}^{+}_{0,\infty}$ a time advance function which determines the maximum duration that the system can remains in its current state. If after a time ***ta(s)*** external events do not occur, the system performs an internal transition going to a new state s_2 that is defined by: $\mathbf{s_2} = \boldsymbol{\delta}_{\mathbf{int}}(\mathbf{s_2})$

The transition from the state s_1 to s_2 also produces an output event defined by $\boldsymbol{y} = \boldsymbol{\lambda}(\mathbf{s_1})$**.**

The state equation describing this atomic model is presented by the following expression:

$$\mathbf{q}' = \boldsymbol{\delta}_{\mathbf{int}}(\mathbf{q}) \oplus \boldsymbol{\delta}_{\mathbf{ext}}(\mathbf{q}, \mathbf{X}) \quad \textbf{(2)}$$

This means that the new state of the system depends on only one of the two transition functions. Figure 2 illustrates a DEVS atomic model [15].

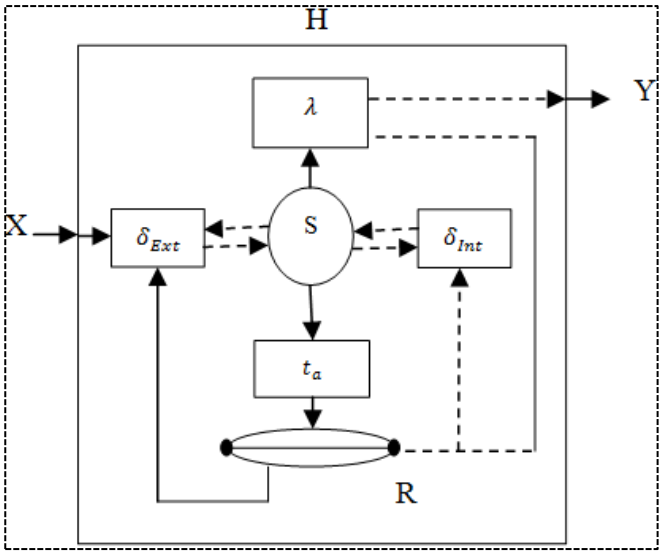

Figure 2 DEVS Atomic Specification

II.2. Quantized State System (QSS)

Traditionally in continuous systems simulation have been used numerical methods for integrating differential equations, all of these have a common characteristic: time discretization. That is, the variable sampled at discrete instants of time, whose step is always the same. The problem in relation with such systems is the uncertainly with the variable value between consecutive samples, it can produce an increasing error, that in some cases can grow to unacceptable values causing instability.

In [14] are proposed new methods for continuous systems discretization based on the concept of quantify the state variables of the ordinary differential equation (ODE). These methods are called quantified states systems.

A system with the following form:

$$\dot{x}(t) = f\big(x(t), u(t)\big) \tag{3}$$

Where $\in \mathbb{R}^n$, $f: \mathbb{R}^n \to \mathbb{R}^n$, can be approximated by the system:

$$\dot{x}(t) = f\big(q(t), u(t)\big) \tag{4}$$

Where $q(t)$ e $x(t)$ are related component to component, initially through quantification functions with hysteresis. This concept is illustrated in Figure 3. Where Q is the quantified state vector and **U** is the system inputs vector.

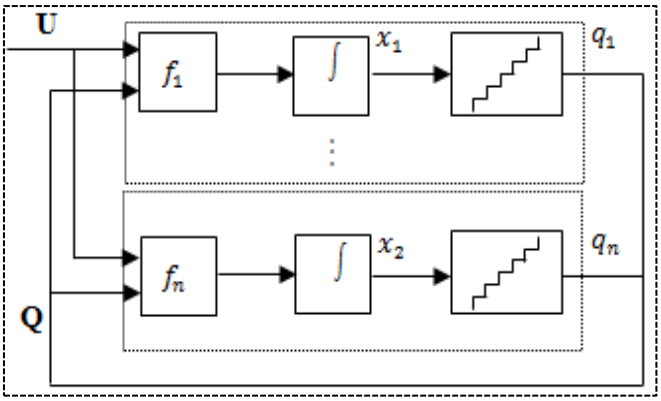

Figure 3. QSS Block Diagram (Kofman, 2003)

In the QSS block diagram is shown that the model can be divided into DEVS modules grouping both the integrators and the static functions quantified with hysteresis. The static functions perform the basic mathematical operations related to multiplications by system gains, in addition they are the inputs of quantified variables, generating an event when any variable change its value. When this happens an input event is created at the integrators, which perform a recalculation of the time at which the variable will change in a value equals to ΔQ, and then is scheduling an internal transition. The DEVS structure that represents the quantified integrator with hysteresis is given in [14].

$$\mathbf{H_2 = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, t_a).} \tag{5}$$

Where:

$X = Y = \mathbb{R} \times \mathbb{N}$
$S = \mathbb{R}^2 \times \mathbb{Z} \times \mathbb{R}_0^+$
$\boldsymbol{\delta_{int}(s)} = \delta_{int}(x, d_x, k, \sigma) = (x + \sigma \cdot d_x, d_x, k + sign(d_x), \sigma_1)$
$\boldsymbol{\delta_{ext}(s, e, x_u)} = \boldsymbol{\delta_{ext}}(x, d_x, k, \sigma, e, x_v, p) = (x + e \cdot d_x, x_v, k, \sigma_2)$
$\boldsymbol{\lambda(s)} = \lambda(x, d_x, k, \sigma) = Q_{k+sign(d_x)}, 0)$
$\boldsymbol{ta(s)} = ta(x, d_x, k, \sigma) = \sigma$

With:

$$\boldsymbol{\sigma_1} = \begin{cases} \dfrac{\boldsymbol{Q_{k+2} - (x + \sigma \cdot d_x)}}{\boldsymbol{d_x}} & \boldsymbol{if\ d_x} > 0 \\ \dfrac{\boldsymbol{(x + \sigma \cdot d_x - (Q_{k-1} - \varepsilon)}}{\boldsymbol{|d_x|}} & \boldsymbol{if\ d_x} < 0 \\ \infty & \boldsymbol{if\ d_x = 0} \end{cases}$$

And,

$$\boldsymbol{\sigma_2} = \begin{cases} \dfrac{\boldsymbol{Q_{k+1} - (x + e \cdot d_x)}}{\boldsymbol{x_v}} & \boldsymbol{if\ x_v} > 0 \\ \dfrac{\boldsymbol{(x + e \cdot d_x - (Q_k - \varepsilon)}}{\boldsymbol{|x_v|}} & \boldsymbol{if\ x_v} < 0 \\ \infty & \boldsymbol{if\ x_v = 0} \end{cases}$$

III. MODELING AND SIMULATION

The diagram in Figure 4 shows the activities in the proposed.

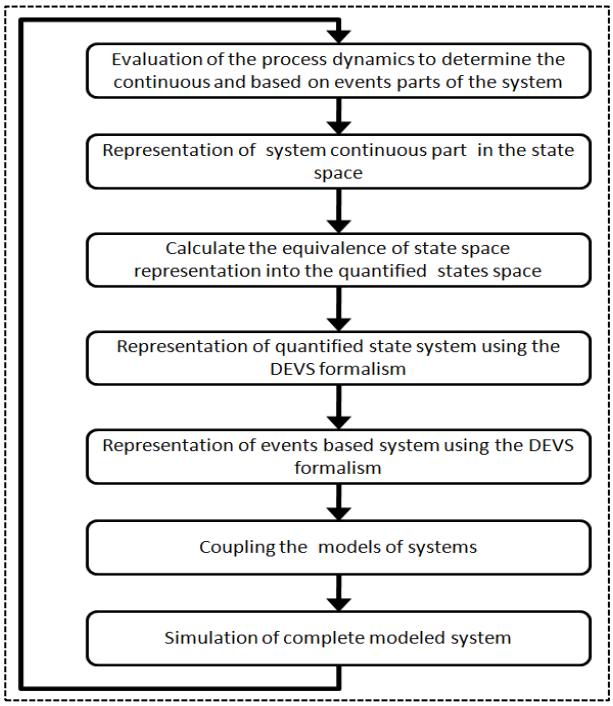

Figure 4. Modeling and Simulation Methodology

In this methodology, initially is realized a review of the system to be modeled in order to clearly establish its continuous and described by events parts respectively. Once the continuous part is described it is obtained the state space representation of the variables of interest. After that, it is realized a representation of each considered variable in the system using the states quantization technique according to equation (4), in order to facilitate the specification the continuous system through DEVS formalism. Subsequently it is performed a coupling of it with the discrete part model previously specified in DEVS according to equation (1), obtaining in such way a hybrid architecture model of the system[9].

Having obtained the general model of the system, it is possible to simulate it (Matlab-Simulink, Stateflow) with the aim of represent the static functions describing the modeled process behavior. The proposed methodology finishes with the analysis of simulation results; it may go back to the beginning to take new features into account within the model.

IV. HYBRID MANUFACTURING SYSTEM

The work is based on the Platform for Industrial Research, Education and Training on Automation (PIPEFA), this is composed of work stations, each one realizing a specific task within a product assembly process. In each station is possible to identify continuous and described by events signals, so it can be stated that these stations are hybrid architecture systems. This production system will be presented as a case study in order to evaluate the proposed methodology since this system can be considered as a hybrid architecture production system according to its component variables.

IV.1 PIPEFA Description

This production system is an industrial platform for research, teaching and training on automation, and is composed by five work stations that perform the assembly of LEGO bricks over a base plate. This platform was implemented in the Laboratory of Integrated Automation and Robotics in the Faculty of Mechanical Engineering at University of Campinas (UNICAMP), it intends to represent a factory floor coping with real aspects of production such as technology integration, flexibility and production management techniques. This platform performs the assembly of a generic product consisting of a base plate and smaller LEGO bricks [16].
Representation of this system is presented in Figure 5, where it can be shown the physic distribution of work stations and the transfer system allowing the transport of the products between stations in order to perform determined process activities. The location of the base plate on the conveyor belt is made by the assembly station, the assembly of the buckets on the base side is performed by the lateral assembly and disassembly station, location of bricks on central position is performed by the central assembly and disassembly station, quality control is achieved in inspection station and finished parts storage is performed by the storage station.

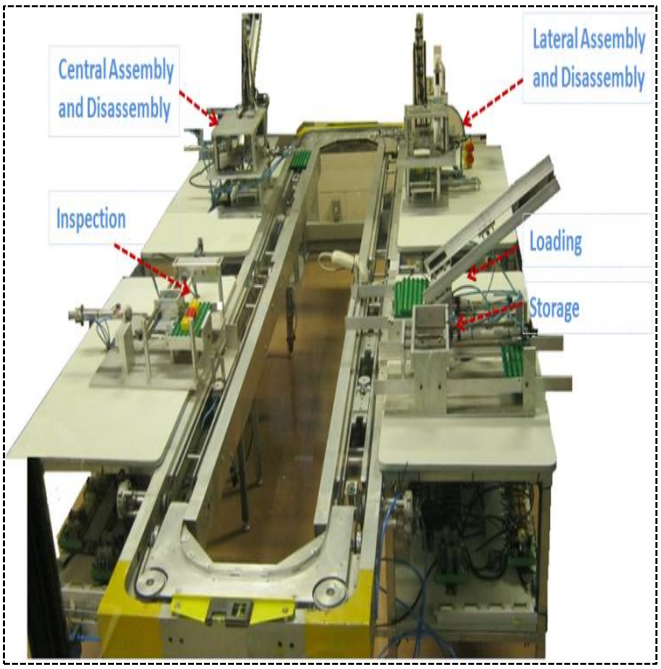

Figure 5. PIPEFA Platform.

IV.1.1 Assembly Station Description

This is the first station of the assembly line which is physically constituted of pneumatic actuators and limit switch sensors in order to determine the state of the actuators. The operational part of the process responsible for releasing and transport the base plate to the assembly line is composed by elements that respond to discrete events. The transport conveyor belt is driven by a DC motor which is currently governed by events (on and off). Figure 6 shows this work station, which is the subject of this work.

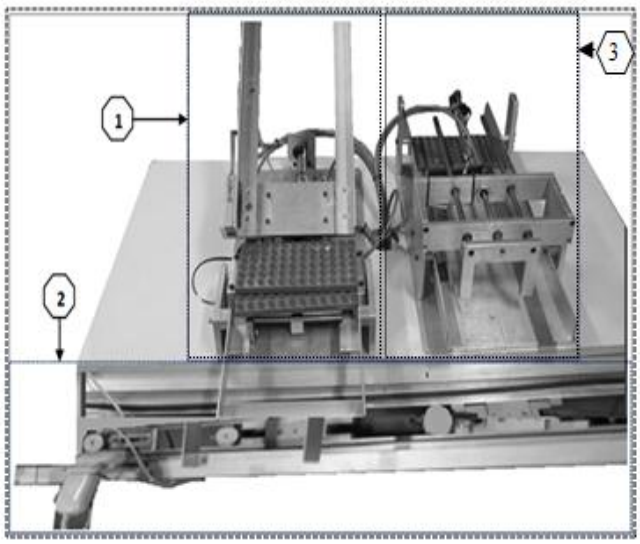

Figure 6. Station Considered in the case Study

The part marked with number 1, consists of actuators that work in terms of discrete events. The part marked with number 2, is originally functioning with events and is this portion that will be modeled as a continuous time system, in order to construct a hybrid model taking into sections 1 and 2 of within this work station.

One reason to develop a hybrid architecture model of this system takes in consideration the possibility of having a flexible configuration of the assembly line, due to a control solely based in discrete events, makes this system too rigid.

A functional specification in terms of GRAFCET graph is presented in Figure 7.

The section marked with number 1 in Figure 6, is responsible for releasing the base plate into the conveyor belt, the section marked with number 2 is responsible for transporting the board to the next work station. Initially this process works under discrete events concept, in this proposal the model of Section 2 is modified in order to be represented in continuous time to be then integrate with the discrete events model of a section number 1 thus forming a hybrid dynamics system in which the two dynamics interact in a single model.

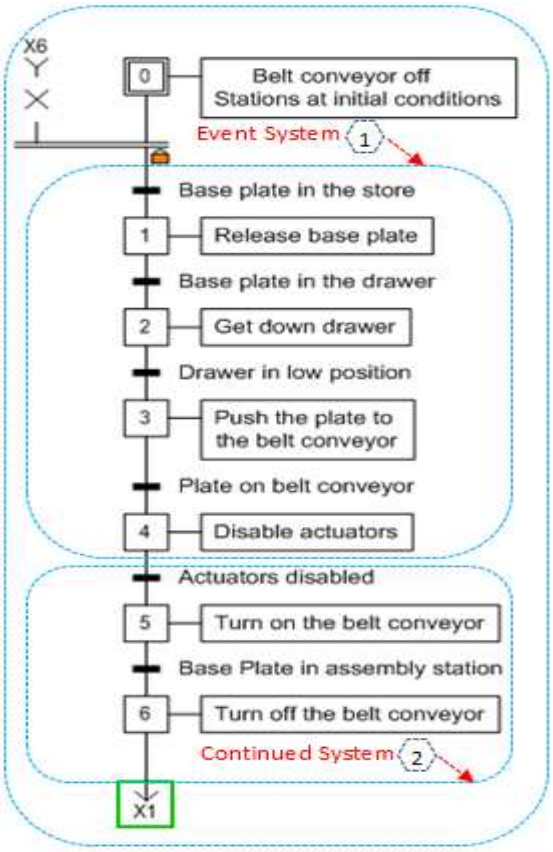

Figure 7. Functional GRAFCET Diagram - Charging Station

IV.1.2 Description of Lateral Assembly Station

This job began the transformation of raw materials (in this case a base plate and LEGO-type blocks) in a finished product. The task performed at this station is to receive the base plate, place it in a platform and then locate the bricks in its left and right sides according to installation requirements.

The assembly can be made even in two different heights, allowing two different types of products. Figure 8 shows the lateral assembly and disassembly station.

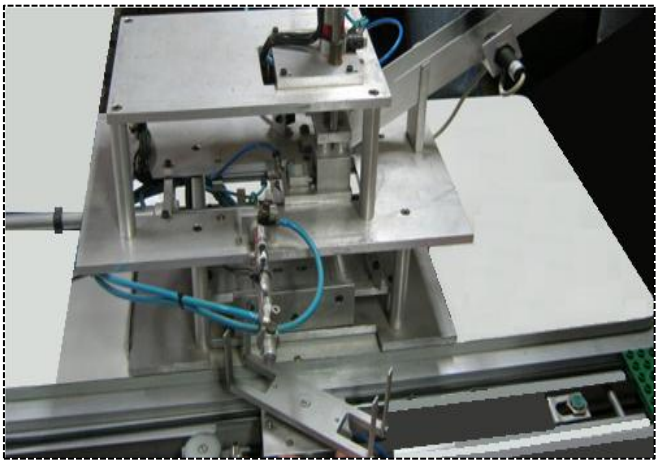

Figure 8. Lateral Assembly Station

IV.1.3 Description of Central Assembly Station

The activity that is performed in this station is to locate the LEGO bricks on the central part of the base plate. Just as in the previous station, it can be put even two levels of blocks according to the requirements. In order this station to operate it uses actuation and measuring devices both responding to discrete events. This station is shown in Figure 9.

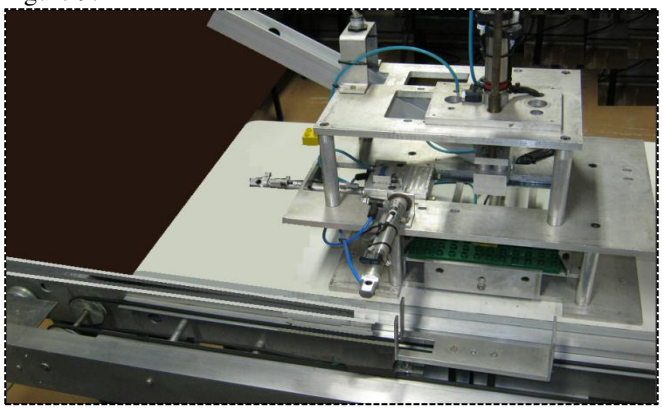

Figure 9. Central Assembly Station

IV.1.4 Description of Transport System

This system consists of independent conveyor belts for each job. Each conveyor belt is driven by a DC motor. The transport system allows carrying raw materials and products between the stations that make up the complete production system activation and deactivation control for these motors is done using discrete event commands.

IV.1.5 Description of Inspection Station

The inspection station verifies that the manufacturing of the products is made according to production requirements; the inspection task is realized done by means of sensors that determine if the location of the bricks is correct and if the required amount of each product is appropriate. With this information the system determines if the product is finished and should be storage or if the product is defective and consequently it should be rejected or sent for re-assembly in the production system. The inspection station is shown in Figure 10.

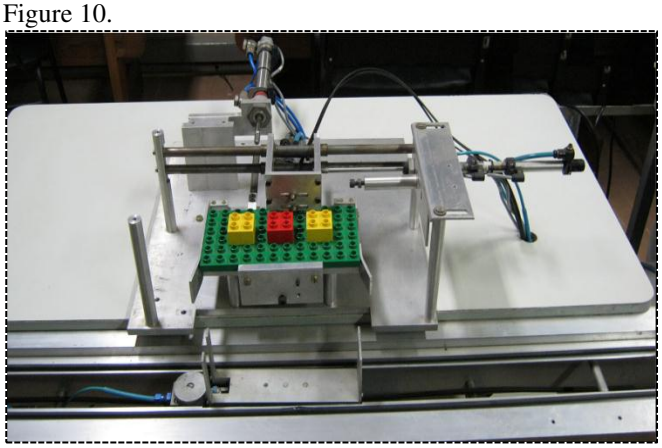

Figure 10. Inspection Station

IV.1.6 Functional Description of the Assembly Line

An example of the assembly line configuration is presented in Figure 11, using functional GRAFCET. In this application is presented the tasks sequence to complete the assembly of a product only with bricks on its lateral side. In this activity take part: charging, lateral assembly and disassembly, inspection and storage stations. These stations are presented as modules responsible to perform specific tasks.

All actuators are ON - OFF type, including the DC motors that perform the movement of the sections in the assembly line. The first task to be performed is to release the base plate, once finished this task, the motor moves the conveyor belt corresponding to this station to bring the product until lateral assembly station where is made this task. Then the conveyor motor corresponding to assembly station is turned on, moving the product until inspection station.
When the product arrives to this station is defined if it will be stored, discarded or continues in the system for reassembly.

Taking into consideration only an events based dynamics it is difficult to know the production times due to it cannot be developed a control of the transfer system existing between the stations.

IV.2 Case Study

The methodology proposed in this article is applied and evaluated through a case study considering the modeling and simulation, of one of the five work station comprising the PIPEFA, this station supplies LEGO boards to a assembly line which transport them until a second station. This station is a hybrid architecture system.
The session identified with number 1 in Figure 6 represents the part controlled by the occurrence of events, it is comprised of pneumatic type cylinders as actuators which give the dynamics of this sub-process, together with final course sensors identifying the state of the actuators.

According to the tasks sequence of the process the events based dynamic presented in Figure 6 can be modeled taking into account (1), and taking Figure 7 as reference: where *Sx* represents the states, Rx represents the transitions and *EXTx* represents the final course sensors of the actuators.

$$\mathbf{X} = \{R_x | x \in \{0,1\}\} : \qquad (6)$$
$$\mathbf{Y} = \{EXT_A, EXT_B, EXT_C, EXT\_D\} X \mathbb{R}_0^+$$
$$\mathbf{S} = \{S1, S2, S3, S4, S5\} X \mathbb{R}_0^+$$
$$\boldsymbol{\delta}_{\mathbf{int}}(s, \sigma) = (s, \infty)$$
$$\boldsymbol{\delta}_{\mathbf{ext}}(s, e, x) = (x, 0)$$

$$\boldsymbol{\lambda}(\boldsymbol{s}) = \begin{Bmatrix} \mathrm{S1} = \mathrm{R1}\ and\ D1 \\ \mathrm{S2} = \mathrm{R2}\ and\ S1 \\ \mathrm{S2} = \mathrm{R2}\ and\ S1 \\ \mathrm{S4} = \mathrm{R4}\ and\ S3 \\ \mathrm{S5} = \mathrm{R5}\ and\ S4 \end{Bmatrix}$$

$$\mathbf{t_a}(\mathrm{S}, \sigma) = \sigma$$

The continuous part of the process is still represented by number 2 in Figure 6, is a conveyor responsible for transporting the bases until another workstation. The model of the transfer system can be present as a composition of an electrical part coupled with a mechanical system. The electrical part corresponds to a permanent magnet DC motor, with the following characteristics.

A schematic of the transfer system is presented in Figure 12, where is the representation of the motor equivalent circuit engaged with the load to move. First is obtained the motor model, relating the applied armature voltage with the velocity of it.

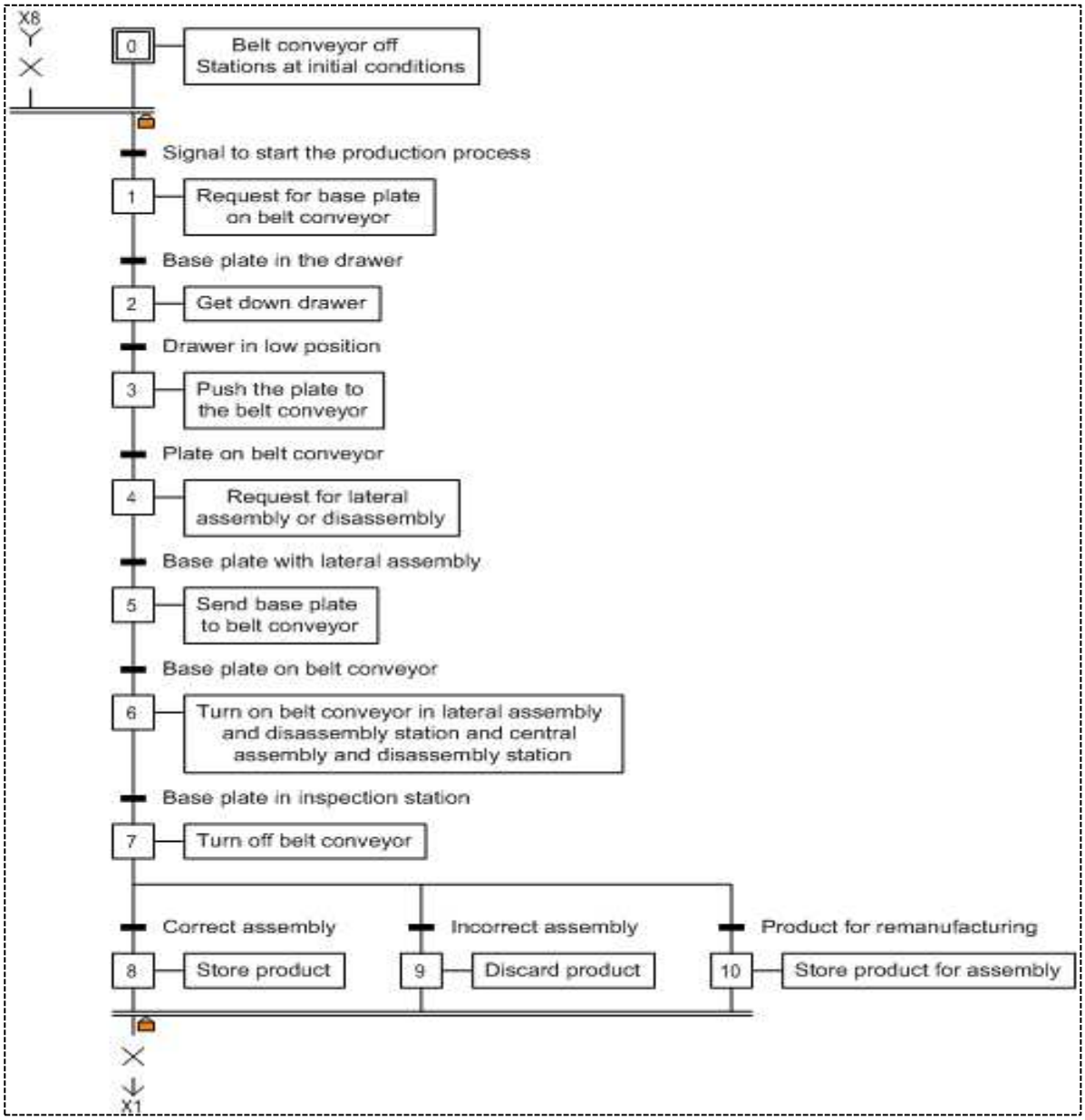

Figure 11. Functional diagram

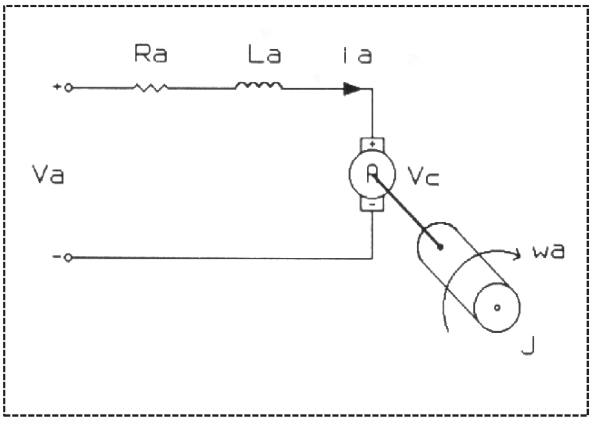

Figure 12. Conveyor equivalent Circuit

According to [12] the system can be modeled taking into account equations 5 and 6.

$$\frac{d}{dt}i_a = -\frac{R_a}{L_a}i_a - \frac{k_v}{L_a}w_a + \frac{V_a}{L_a} \tag{7}$$

$$\frac{d}{dt}w_a = \frac{k_t}{J}i_a - \frac{B}{J}w_a - \frac{T_L}{J} \tag{8}$$

Where J is the rotor inertia plus the load inertia w_a is the angular. Taking in consideration that the motor torque is proportional to the armature current and k is the motor constant, it is obtained the electromechanical relation presented in equation 7:

$$T(t) = K_T i_a(t) \tag{9}$$

Substituting equation 9 is obtained equation 10 as follows:

$$K_T i_a(t) = J\alpha(t) + bw(t) + T_F \tag{10}$$

Selecting the current, velocity and position of the motor as state variables, is derived the state space representation (equation 11)

$$\dot{x}_1(t) = \frac{di_a(t)}{dt}$$
$$\dot{x}_2(t) = \alpha(t) \tag{11}$$
$$\dot{x}_3(t) = x_2(t)$$

Table 1. Motor Parameters

Parameter	Symbol	Value	Units
Supply Voltage	V_a	12	V
Constant	K_v	0,0213	
Torque constant	K_t	0,0232	Nm/A
Resistance	R_a	1,26	Ω
Inductance	L_a	1,02	mH
Rotor inertia	J_r	4,2 E - 06	Kg m^2
Speed without load	V_{sc}	21	RPM

Substituting equations 7 and 8 (mechanical and electrical) is obtained equation 12.

$$V_a(t) = x_1(t)R_a + L_a\dot{x}_1(t) + K_v x_2(t)$$
$$K_V x_1(t) = J\dot{x}_2(t) + Bx_2(t) + T_L \tag{12}$$

As well, the state space model is presented in equation 13.

$$\frac{d}{dt}\begin{bmatrix} i_a \\ w_a \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_v}{L_a} \\ \frac{k_V}{J} & -\frac{B}{J} \end{bmatrix}\begin{bmatrix} i_a \\ w_a \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix}\begin{bmatrix} V_a \\ T_L \end{bmatrix} \tag{13}$$
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} i_a \\ w_a \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} V_a \\ T_L \end{bmatrix}$$

Substituting the motor parameters (given by the manufacturer) in the state space model parameters and taking table 14 in consideration, is obtained:

$$\dot{\bar{x}}(t) = \begin{bmatrix} -1235{,}29 & -22{,}74 \\ 5523{,}81 & -0{,}62 \end{bmatrix}\bar{x}(t) + \begin{bmatrix} 980{,}39 & 0 \\ 0 & -0{,}238\times 10^6 \end{bmatrix}\begin{bmatrix} V_a(t) \\ T_L(t) \end{bmatrix} \tag{14}$$

Taking into account that the engine load for this application will be depreciated due to the inertia of the load is much smaller than the inertia of the engine and taking the parameters of the engine according to the table 1. The simulation of the state variable speed in continuous time is presented in Figure 13. This corresponds to the transfer system belt speed. The system reaches a steady speed that corresponds to that given by the manufacturer.

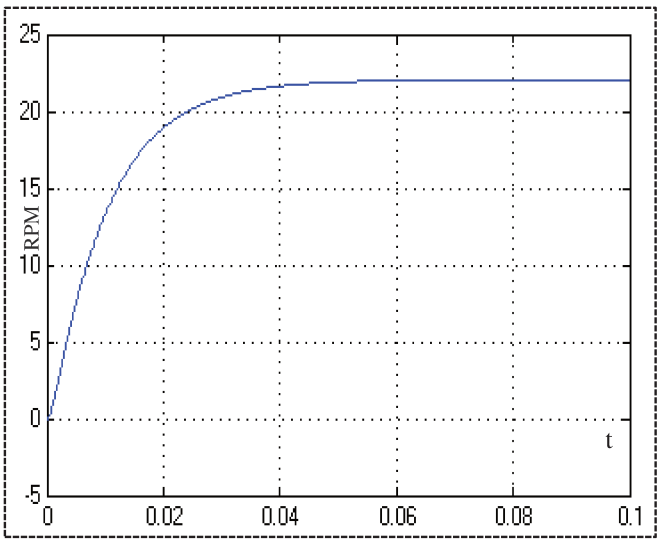

Figure 13. Conveyor velocity

The other state variable is the current consumed by the engine. According to the data and the model obtained, the behavior of this variable is presented in Figure 14.

This represents a current consumption of around 8 A. to overcome the inertia of the motor and load that makes up the conveyor, but when the motor is in steady state the current consumption is less than 1 A. This transition of power consumption occurs rapidly, less than 3 ms this agrees with the time for the engine speed reaches its steady state.

Having the state space model of the system is realized a quantization according to the tuples represented in (1), according to Figure 3, which gave a result presented in equation 15. Static functions and integrator are shown in Figure 15.

Then, it is conducted a simulation of the system in states quantified according to Figure 3, for this was used Matlab-Stateflow™ tool, to create internal and external transition functions and forward-time function representing in such way the system dynamics using DEVS.

$$\begin{aligned}
\dot{x}_1(t) &= 980{,}39 \cdot V_a(t) - 1235{,}29 \cdot \left(x_1(t) + \Delta x_1\right) \\
&\quad - 22{,}74 \cdot \left(x_2(t) + \Delta x_2\right) \\
\dot{x}_2(t) &= 5523{,}81 \cdot \left(x_1(t) + \Delta x_1\right) - 0{,}62\left(x_2(t) + \Delta x_2\right) \\
&\quad - 0{,}238 \times 10^6 \cdot T_L(t) \\
\dot{x}_1(t) &= 980{,}39 \cdot V_a(t) - 1235{,}29 \cdot q_1(t) \\
&\quad - 22{,}74 \cdot q_2(t) \\
\dot{x}_2(t) &= 5523{,}81 \cdot q_1(t) - 0{,}62 q_2(t) \\
&\quad - 0{,}238 \times 10^6 \cdot T_L(t)
\end{aligned} \tag{15}$$

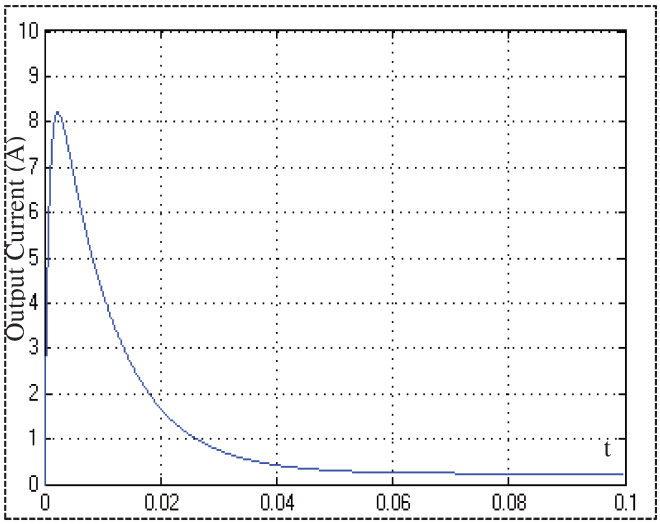

Figure 14. Motor current consumption

With the model obtained was carried out simulation presented in Figure 16. This simulation allowed us to compare the equivalence of the model in continuous time which is represented by green color and graphic simulation system that is quantified in states represented by the graph in blue. There is a very good approximation of the system in its two models.

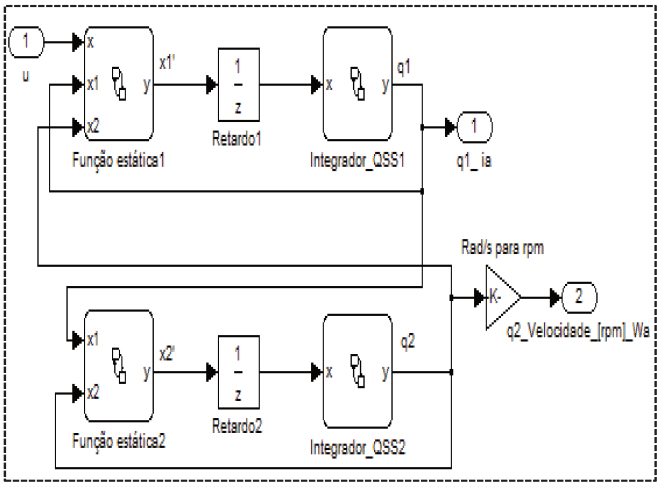

Figure 15. Quantified representation of the states.

In Quantified graph of the system can be seen how the system is carrying out this quantization with a variable step discretization, which depends on the change of state variable. It may be noted as when the system starts, the change of variable *y* is faster when the stationary state on the quantization step is larger, unlike the system that is simulated by discretization of the variable, where the discretization step is fixed.

An important aspect of the results is the amount of steps required in the simulation of quantified in states system, (around 48 steps) and simulation of continuous-time system, (requiring around 80 steps).

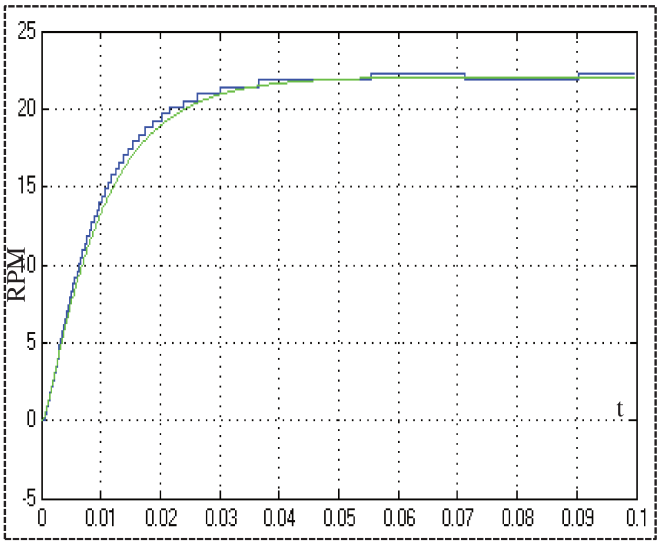

Figure 16. Conveyor velocity

This is very important to implement a system of digital architecture, because it requires a lower processing speed for quantified in states system.

The DEVS structure for $\dot{x}_2$ static equation is:

$$M_{FE2} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{where} \tag{16}$$
$$X = Y = R \times N; S = R^3 \times R_0^+$$
$$\delta_{int}(s) = \delta_{int}(q_1, q_2, u, \sigma) = (q_1, q_2, u, \infty)$$
$$\delta_{ext}(s, e, (x_v, port)) = \delta_{ext}(q_1, q_2, u, \sigma, e, (x_v, port))$$
$$= \begin{Bmatrix} (x_v, q_2, u, 0) & si\ port = 0 \\ (q_1, x_v, u, 0) & si\ port = 1 \\ (q_1, q_2, x_v, 0) & si\ port = 2 \end{Bmatrix}$$
$$\lambda(s) = \lambda(q_1, q_2, u, \sigma) = (y, port)$$
$$= (5523.81 * q_1 - 0.62 * q_2 - 0.238 \times 10^6 * u, 0)$$
$$ta(s) = ta(u, q_1, q_2, \sigma) = \sigma$$

Using the two DEVS models, which represents the dynamics of discrete events and the DEVS model of dynamics still represented in states quantified, it is realized the coupling to obtain a global model through the concept of coupled DEVS, according to (2).
A schematic of the coupled model is shown in Figure 17, it represents the event in which a base plate arrives to the conveyor belt, this event is generated by means of an events generator. These events are generated through the output Out_1, the DEVS model of this generator is presented in equation 17, where the maximum number of plates in the conveyor belt is 3 for this application.

$$M_{S1} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{where} \tag{17}$$
$$X = Y = \mathbb{R}; S = \mathbb{R}^1 \times \mathbb{R}_0^+$$
$$\delta_{int}(s) = \delta_{int}(s_1, \sigma) = (s_1 + 1, T)$$
$$\lambda(s) = \lambda(s_1, \sigma) = (y) = \{(s_1 + 1) \quad si \quad y < 3\}$$
$$ta(s) = ta(x_1, dx_1, \sigma) = \sigma$$

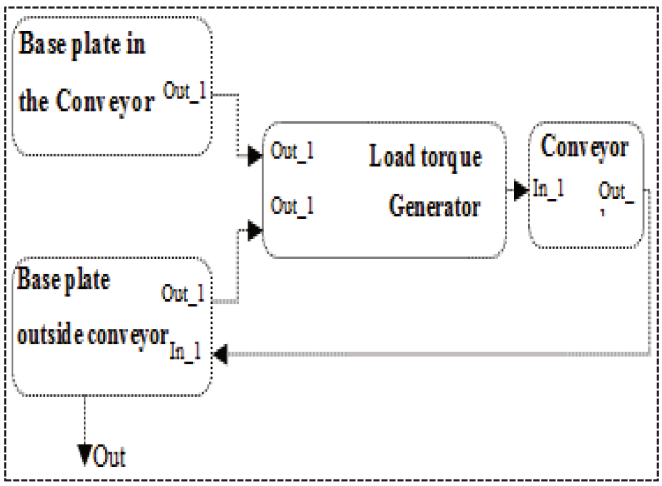

Figure 17. Coupled Hybrid model

The load generator block performs the calculation of load torque according to the corresponding number of base plates on the conveyor belt at this time; this computation is done as a sum in accordance with the way the plates enter and leave the conveyor belt, as presented in equation 18.

$$M_{TL} = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta) \quad \text{where} \tag{18}$$
$$X = Y = R; S = R^3 \times R_0^+$$
$$\delta_{int}(s) = \delta_{int}(in, out, T_L, \sigma) = (in, out, T_L, \infty)$$
$$\delta_{ext}(s, e, x_v) = \delta_{ext}(in, out, T_L, \sigma, e, x_v) =$$
$$\begin{Bmatrix} (in + 1, out, T_L, 0) & si\ port = 0 \\ (in, out + 1, T_L, 0) & si\ port = 1 \end{Bmatrix}$$
$$\lambda(s) = \lambda(in, out, T_L, \sigma) = (y) = (T_L \cdot (in - out))$$
$$ta(s) = ta(in, out, T_L, \sigma) = \sigma$$

The block output representing the conveyor belt circuit performs the calculation of the time each object is over the conveyor belt according to the speed of it; therefore it presents a velocity input from the conveyor belt. The signals and connectivity ports between the two DEVS models are defined generating scheme presented in Figure 18.

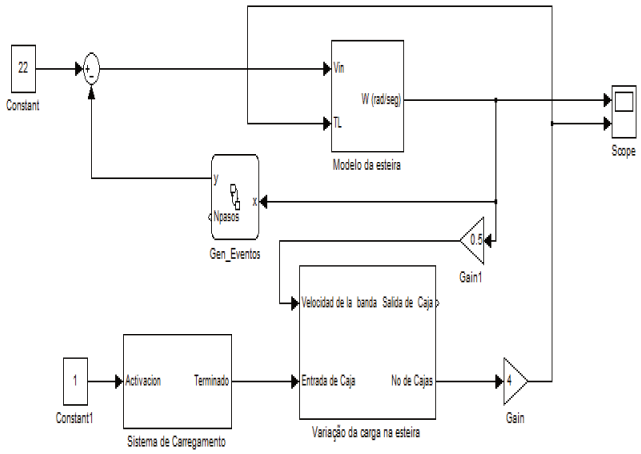

Figure 18. Hybrid model

The conveyor belt block has a dynamic that corresponds to a DC motor working on steady-state regime, when its load characteristic varies due to torque variation. The operating point is moved but the motor remains stable, as shown in Figure 19 .

As a final result is obtained the simulation of the coupled system that represents the continuous dynamic behavior coupled with discrete events dynamics, which is a hybrid system architecture.

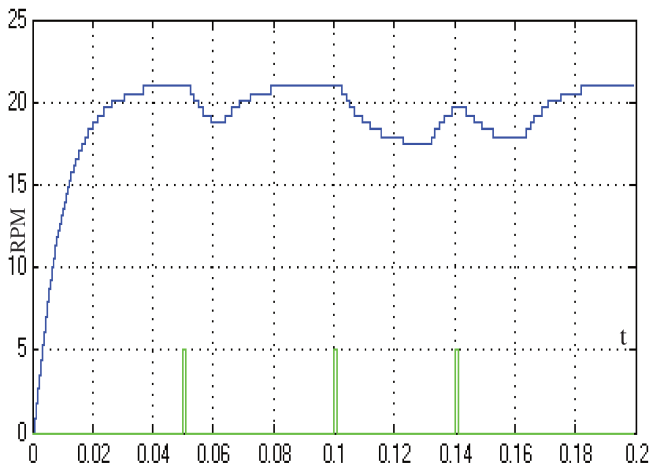

Figure 19. Response of hybrid system

V. CONCLUSIONS AND FUTURE WORK

This paper proposes a methodology to model systems that integrates continuous and discrete events variables into its dynamic, a case study taking in consideration one station of the assembly line is presented, this station was initially modeled under the discrete events concept where all its sensors and actuators are ON – OFF type so having mechanical wear and high energy consumption that is specially truth for the motors that move each section of the conveyor belt. Modeling the dynamics of the system as an hybrid system architecture was possible to simulate the process, perform the conveyor belt operational analysis and to study the form in which the products on the conveyor affect its performance in terms of speed

O system simulates the products drop over the conveyor belt so modifying the system inertia in a value determined by the mass of each product.

With this concept in mind, from the modeling of hybrid systems architecture can be developed more flexible control strategies allowing production systems to adapt quickly to market changes, allowing determining more accurate of production times.

With the results obtained in this work can be proposed works oriented to new supervision and control strategies that may be applicable to industrial production systems in order to ensure more accurate production times, ease of implementation of control and supervision strategies, actuators lower energy consumption (specifically motors moving sections of the conveyor belt), perform stability analysis using processes hybrid models, etc. Furthermore it can determined how the based in events variables affect the continuous time variables, even it can be analyzed the interaction and dependence between continuous variables and based in events variables.

VI. REFERENCES

[1] Capocchi, L., F. Bernardi, et al. A general DEVS-based formalism for behavioral fault simulation. Simulation Modelling Practice and Theory, 2006.

[2] Filippi, J.-B. and P. Bisgambiglia. An implementation of a DEVS based formal framework for environmental modelling." Environmental Modelling & Software, 2004.

[3] Hong, K. and T. Kim, DEVS specification language for modeling, simulation and analysis of discrete event systems." Information and Software Technology, 2006.

[4] Van der Schaft A., Schumacher H., An Introduction to Hybrid Dynamical Systems, 2000

[5] Branicky M., Studies in Hybrid Systems: Modeling, Analysis, and Control. Tesis Massachusetts Institute of Technology, 1995

[6] Alur, R., T. A. Henzinger, et al. "Discrete abstractions of hybrid systems." Proceedings of the IEEE, 2000

[7] Palaniappan, S., A. Sawhney, et al. Application of the DEVS Framework in Construction Simulation. Simulation Conference, 2006. WSC 06. Proceedings of the Winter, 2006.

[8] Nikolaidou, M., Dalakas, V., Mitsi, L., Kapos G. A SysML Profile for Classical DEVS Simulators. Software Engineering Advances, The Third International Conference on, IEEE, 2008.

[9] Bergero, F., Kofman, E., Basabilbaso, C., Zúccolo, J. desarrollo de un simulador de sistemas hibridos en tiempo real. XXI Congreso Argentino de Control Automático, 2000.

[10] Zeigler, B. P., Y. Moon, et al. DEVS approximation of infiltration using genetic algorithm optimization of a fuzzy system, Mathematical and Computer Modelling,1996.

[11] Kim, K. H., Y. R. Seong, et al, Ordering of simultaneous events in distributed DEVS simulation, Simulation Practice and Theory, 1997

[12] Isermann R. 2002(Aput M. Ruderman, et al), Optimal State Space Control of DC Motor, Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, 2008.

[13] Mital A., et al, Product Development, Elsevier Inc, 2008.

[14] Kofman E., Simulación y Control de Sistemas Continuos por Eventos Discretos, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, 2003.

[15] Chen Liu, et al, Extend Srml Schema Based On Devs: An Executable Devs Language, Proceedings of the Winter Simulation Conference, 2005.

[16] Cintia k. A., Uma abordagem interativa para o prblema de capacitacao e pesquisa em automacao,2005

Authors' information

[1]Nueva Granada Military University, Bogotá-Colombia, dario.amaya@unimilitar.edu.co
[2]Nueva Granada Military University, Bogotá-Colombia, ricardo.castillo@unimilitar.edu.co
[3]Campinas State University, Campinas-Brasil, rosario@fem.unicamp.br

Dario Amaya Hurtado was educated at UAN, Bogotá, Colombia receiving the B Sc. degree in Electronics Engineering in 1995 and the M.Sc. degree in Teleinformatic in 2007 by the Faculty of Engineering at the Francisco José de Caldas District University, UFJC in Bogotá, Colombia. Currently he is completing the Ph.D. degree in Mechanical Engineering at Campinas State University, São Paulo, Brazil, working on hybrid control – He has worked

as a professor and researcher at the Military University, Colombia since 2007 been involved in Robotics, Mechatronics and Automation areas.

Ricardo A. Castillo was born in Colombia in 1980, and received his B.Sc. degree in Mechatronics Engineering by the Military University, Bogota, Colombia in 2004 and the M.Sc degree in Mechanical Engineering by the Campinas State University. Currently he is completing a Ph.D. degree in Mechanical Engineering at Campinas State University, São Paulo, Brazil working on Collaborative Automation Modeling and Coordination agent based strategies.
He has worked as a professor and researcher at the Military University, Colombia since 2005 been involved in Robotics, Mechatronics and Automation areas. In addition Prof. Castillo has developed projects related to Artificial Intelligence and mobile autonomous robotics.

João M. Rosário, was educated at Campinas State University, São Paulo, Brazil receiving the B Sc. degree in mechanical engineering in 1981 and the M.Sc. degree in systems and control in 1983 and Specialization Degree in Production and Automation Systems in 1986 at Nancy University, France. He was awarded the Ph.D. degree in 1990 by Ecole Centrale – Paris, France, for research into Automation and Robotics.
He worked briefly as a control engineer and robotics in the Hispano Suiza, France and underwater robotics at GKSS, Germany. Actually, he's invited professor at Automation and Control department, at SUPELEC, France. Currently he is an associated professor at Faculty of Mechanical Engineering at the University of Campinas, UNICAMP, responsible of the Automation and Robotics Laboratory, and coordinator of Robotics and Automation in the Brazilian Manufacturing Network. From 1998-2002 was the head of the graduate course of Automation and Control Engineering (Mechatronics) Department. Actually develops various industrial research projects at national and international level in different areas such as: Industrial Automation, Control Design of Mechatronics Systems and Biomechanics.