



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

CENTRO DE TECNOLOGIA



PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DA COMPUTAÇÃO

**Haulisson Jody Batista da Costa**

**Modelagem em SystemC-AMS de uma  
plataforma compatível com o Sistema de Coleta  
de Dados Brasileiro**

Natal, RN, Setembro de 2009

**Haulisson Jody Batista da Costa**

**Modelagem em SystemC-AMS de uma plataforma  
compatível com o Sistema de Coleta de Dados  
Brasileiro**

**Dissertação de Mestrado** apresentado ao Programa de Pós-Graduação em Engenharia e da Computação da Universidade Federal do Rio Grande do Norte como um dos pré-requisitos para a obtenção do título de Mestre em Ciências.

Orientador: Prof. Dr. Fernando Rangel de Sousa.

Natal, RN, Setembro de 2009



Costa, Haulisson Jody Batista da  
Modelagem em SystemaC-AMS de uma Plataforma  
Compatível com o Sistema de Coleta de Dados Brasileiro / Haulisson  
Jody Batista da Costa – Natal, RN, 2009.  
86 f.: il.: 39

Orientador: Fernando Rangel de Sousa.  
Dissertação – Universidade Federal do Rio Grande do Norte.  
Centro de Tecnologia. Programa de Pós-Graduação em Engenharia  
Elétrica e da Computação. 2009

1. Coleta de dados via satélite. 2. Modelagem em SystemC-  
AMS. 3. Especificação de Sistemas. I. Título.

**Haulisson Jody Batista da Costa**

**Modelagem em SystemC-AMS de uma plataforma  
compatível com o Sistema de Coleta de Dados Brasileiro**

**Dissertação submetida ao corpo docente da Universidade Federal do Rio Grande do Norte, como parte dos requisitos necessários à obtenção do grau de Mestre em Ciências.**

**Data de aprovação: \_\_\_\_/\_\_\_\_/\_\_\_\_**

**Aprovada por:**

---

**Prof. Dr. Fernando Rangel de Sousa - Orientador**  
**Universidade Federal do Rio Grande do Norte**

---

**Prof. Dr. David Simonetti Barbalho**  
**Universidade Federal do Rio Grande do Norte**

---

**Prof. Dr. Sebastian Yuri Cavalcanti Catunda**  
**Universidade Federal do Maranhão**

**Natal - RN**  
**2009**

Dedico este trabalho aos meus pais a quem agradeço a vida e o amor que a mim dedicaram.

## **AGRADECIMENTOS**

Agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro no desenvolvimento do trabalho científico e aos amigos e familiares que apoiaram neste breve, porém intenso período de trabalho.

“A percepção é forte e a visão é fraca. Em estratégia, é importante ver o que está distante como se estivesse próximo e ter uma visão distanciada do que está próximo”.

Miyamoto Musashi



## RESUMO

Este trabalho apresenta resultados de simulação de uma plataforma de identificação compatível com o Sistema de Coleta de Dados Brasileiro do INPE, modelado com SystemC-AMS. SystemC-AMS, que é uma biblioteca de classes C++ dedicada à simulação de sistemas heterogêneos, oferece um recurso poderoso para descrever modelos nos domínios digital, analógico e de RF, bem como sistemas mecânicos e óticos. O modelo projetado foi dividido em quatro partes. O primeiro bloco leva em consideração a órbita do satélite, necessário para modelar corretamente o canal, inclui o efeito Doppler, a atenuação e o ruído térmico. O bloco identificação que detecta a presença de satélite é composto por um amplificador de baixo ruído, filtro passa-banda, detector de potência e um comparador lógico. O bloco controlador é responsável por habilitar o transmissor RF, quando a presença do satélite é detectada. O controlador foi modelado por uma rede de Petri, devido à natureza assíncrona do sistema. O quarto bloco é o transmissor, que realiza a modulação da informação em BPSK  $\pm 60^\circ$ . Este bloco é composto por oscilador, misturadores, somador e amplificador. Todo o sistema foi simulado simultaneamente. Os resultados serão utilizados para especificar componentes de sistema e para a elaboração de banco de testes para a verificação do projeto.

**Palavras-chave:** Modelagem em SystemC-AMS, Coleta de dados, Especificação de Sistemas.

## ABSTRACT

This work presents simulation results of an identification platform compatible with the INPE Brazilian Data Collection System, modeled with SystemC-AMS. SystemC-AMS that is a library of C++ classes dedicated to the simulation of heterogeneous systems, offering a powerful resource to describe models in digital, analog and RF domains, as well as mechanical and optic. The designed model was divided in four parts. The first block takes into account the satellite's orbit, necessary to correctly model the propagation channel, including Doppler effect, attenuation and thermal noise. The identification block detects the satellite presence. It is composed by low noise amplifier, band pass filter, power detector and logic comparator. The controller block is responsible for enabling the RF transmitter when the presence of the satellite is detected. The controller was modeled as a Petri net, due to the asynchronous nature of the system. The fourth block is the RF transmitter unit, which performs the modulation of the information in BPSK  $\pm 60^\circ$ . This block is composed by oscillator, mixer, adder and amplifier. The whole system was simulated simultaneously. The results are being used to specify system components and to elaborate testbenches for design verification.

**Keywords:** Modeling in SystemC-AMS, Data Collection, System Specification.

# SUMÁRIO

<b>AGRADECIMENTOS.....</b>	<b>iii</b>
<b>RESUMO .....</b>	<b>v</b>
<b>ABSTRACT .....</b>	<b>vi</b>
<b>SUMÁRIO .....</b>	<b>vii</b>
<b>LISTA DE ILUSTRAÇÕES .....</b>	<b>x</b>
<b>LISTA DE TABELAS E QUADROS.....</b>	<b>xii</b>
<b>LISTA DE SIGLAS, ACRÔNIMOS E ABREVIACÕES .....</b>	<b>xiii</b>
<b>1 INTRODUÇÃO.....</b>	<b>15</b>
<b>2 SISTEMA DE COLETA DE DADOS BRASILEIRO (SCDB) .....</b>	<b>17</b>
2.1 INTRODUÇÃO.....	17
2.2 PLATAFORMA DE COLETA DE DADOS.....	19
<b>3 MODELAGEM .....</b>	<b>23</b>
3.1 INTRODUÇÃO.....	23
3.2 DIAGRAMA DE BLOCOS .....	23
3.2.1 <b>Modelo Orbital</b> .....	24
3.2.1.1 Efeito Doppler .....	25
3.2.1.2 Modelo do Canal .....	26
3.2.1.3 Modelo do Ruído .....	27
3.2.2 <b>Módulo de Detecção</b> .....	28
3.2.2.1 Modelo de Amplificador de Baixo Ruído .....	30
3.2.2.2 Filtro Passa-Banda.....	31
3.2.2.3 Detector de Potência.....	31
3.2.2.4 Comparador com Histerese .....	31

3.2.3	<b>Módulo Controlador</b> .....	32
3.2.3.1	Controlador.....	33
3.2.3.2	Registrador de Saída.....	39
3.2.3.3	Contador.....	39
3.2.3.4	Conversor Serial.....	40
3.2.4	<b>Módulo de Transmissão</b> .....	41
3.2.4.1	Oscilador.....	42
3.2.4.2	Defasador.....	43
3.2.4.3	Misturador.....	44
3.2.4.4	Seletor.....	44
3.2.4.5	Amplificador de Potência.....	45
<b>4</b>	<b>DESCRIÇÃO EM SYSTEMC</b> .....	<b>46</b>
4.1	INTRODUÇÃO.....	46
4.2	SYSTEMC.....	47
4.3	SYSTEMC-AMS.....	50
4.4	IMPLEMENTAÇÃO.....	52
4.4.1	<b>Descrição dinâmica do satélite e do canal</b> .....	53
4.4.2	<b>Módulo detector</b> .....	54
4.4.3	<b>Módulo controlador</b> .....	55
4.4.4	<b>Módulo transmissor</b> .....	57
<b>5</b>	<b>RESULTADOS</b> .....	<b>59</b>
5.1	PERÍODO DE DETECÇÃO.....	59
5.2	SENSIBILIDADE DO DETECTOR.....	60
5.3	SISTEMA CONTROLADOR.....	62
5.4	SISTEMA DE TRANSMISSÃO.....	63
5.5	CONCLUSÃO.....	65
<b>6</b>	<b>CONCLUSÕES E PERSPECTIVAS</b> .....	<b>66</b>

<b>REFERÊNCIA BIBLIOGRÁFICA .....</b>	<b>67</b>
<b>APÊNDICE A.....</b>	<b>69</b>
APÊNDICE A.1 - Descrição da Distância Orbital em SystemC-AMS.....	69
APÊNDICE A.2 - Descrição do canal <i>Downlink</i> em SystemC-AMS.....	70
APÊNDICE A.3 - Descrição do amplificador de baixo ruído em SystemC-AMS .....	71
APÊNDICE A.4 - Descrição do filtro do <i>Wake-up</i> em SystemC-AMS.....	72
APÊNDICE A.5 - Descrição do detector de potência em SystemC-AMS.....	73
APÊNDICE A.6 - Descrição do comparador com histerese em SystemC-AMS .....	74
APÊNDICE A.7 - Descrição da Rede de Petri em SystemC.....	75
APÊNDICE A.8 - Descrição do contador em SystemC.....	76
APÊNDICE A.9 - Descrição do Registrador de Saída em SystemC.....	77
APÊNDICE A.10 - Descrição do conversor paralelo para serial em SystemC .....	78
APÊNDICE A.11 - Descrição do conversor Biphase-L em SystemC .....	79
APÊNDICE A.12 - Descrição do Seletor em SystemC-AMS .....	80
APÊNDICE A.13 - Descrição do Oscilador em SystemC-AMS .....	81
APÊNDICE A.14 - Descrição do defasador em SystemC-AMS .....	82
APÊNDICE A.15 - Descrição do misturador em SystemC-AMS .....	83
APÊNDICE A.16 - Descrição do Somador em SystemC-AMS .....	84
APÊNDICE A.17 - Descrição do amplificador de Potência em SystemC-AMS.....	85

## LISTA DE ILUSTRAÇÕES

Figura 1 - Passagens Diárias dos Satélites SCD1, SCD2 e CBERS-2 Sobre a Estação de Rastreio de Cuiabá. ....	18
Figura 2 - Os componentes de enlace da comunicação do Sistema Brasileiro de Coleta de Dados. FONTE: <a href="http://satelite.cptec.inpe.br/PCD/sistema.jsp">http://satelite.cptec.inpe.br/PCD/sistema.jsp</a> .....	19
Figura 3 - Localização das PCD do sistema SCD1 realizada em 2006. ....	20
Figura 4 - Característica temporal do Sinal Codificado em <i>Biphase-L</i> .....	21
Figura 5 - Blocos que representam a organização da plataforma de identificação. ....	23
Figura 6 - Diagrama representativo da trajetória do satélite em relação à plataforma de identificação. ....	24
Figura 7 - Esquema da representação do ruído na entrada do sistema.....	28
Figura 8 - Diagrama de blocos do modelo do processamento do detector. ....	29
Figura 9 - Conjuntos de blocos que constituem o sistema controlador.....	32
Figura 10 - Representação temporal referente ao lugar. ....	34
Figura 11 - Representação da associação de tempo a transição.....	34
Figura 12 - <b>Rede de Petri caracterizando o sistema satélite e o controle da plataforma</b> .....	36
Figura 13 - Simulação comportamental do conversor paralelo para serial .....	40
Figura 14 - Diagrama do codificador <i>Biphase-L</i> descrito comportalmente em SystemC...	40
Figura 15 - Resultado da simulação da codificação dos dados em <i>Biphase-L</i> na frequência de 400Hz. ....	41
Figura 16 - Representação gráfica do módulo de transmissão comportamental implementada em SystemC-AMS.....	41
Figura 17 - Espectro na frequência de um oscilador com a injeção de ruído de fase. ....	43
Figura 18 - Comparação entre as plataformas de simulações integradas por domínios e níveis de abstração. ....	46
Figura 19 - Organização da bibliotecas de Classes do SystemC. ....	47
Figura 20 - Diagrama de abstração da implementação da arquitetura do SystemC.....	47
Figura 21 - SystemC em ambiente de desenvolvimento C++.....	48
Figura 22 - Máquina de estados do núcleo de simulação do SystemC. ....	49
Figura 23 - SystemC e SystemC-AMS em ambiente de desenvolvimento C++. ....	50

Figura 24 - Estruturas de Camadas do SystemC-AMS está representada de core diferente. .....	51
Figura 25 - Encapsulamento de blocos contínuos em um Fluxo de dados de um <i>Cluster</i> ..	52
Figura 26 - Estrutura do modelo orbital.....	53
Figura 27 - Resultado de simulação extraída do modelo orbital.....	54
Figura 28 - Resultado de simulação simultânea de cada bloco que constitui o módulo detector.....	55
Figura 29 - Interligações do módulo controlador.....	56
Figura 30 - Comportamento do módulo controlador considerando uma base de um tempo reduzida.....	56
Figura 31 - Simulação de teste para validar a modulação BPSK $\pm 60^\circ$ .....	57
Figura 32 - Espectro do Oscilador centrado na frequência de 401,62MHz.....	58
Figura 33 - Espectro de potência do sinal transmitido pela plataforma.....	58
Figura 34 - Potências do sinal durante o período de passagem do satélite.....	60
Figura 35 - Avaliação de sensibilidade no detector.....	61
Figura 36 - Relação sinal ruído em toda a cadeia do detector.....	62
Figura 37 - Resultado completo da plataforma empregando um período de execução reduzido em relação ao real que é de 14 minutos; 1 – potência do sinal, 2 – Ângulo de elevação, 3 – Sinal captado, 4 – Sinal transmitido.....	63
Figura 38 - Espectro de saída da plataforma, mostrando a harmônica de terceira ordem. .	64
Figura 39 - Representação da área de simetria da constelação.....	64
Figura 40 - Avaliação do erro de Bit durante a transmissão dos dados ao satélite.....	65

## LISTA DE TABELAS E QUADROS

Tabela 1 - Formato da mensagem transmitida pelas PCDs e MTRs do SBCD. ....	21
Tabela 2 - Características de potência das plataforma de coletas de dado do SBCD .....	21
Tabela 3 - Regulamento da UIT para máxima densidade de fluxo de potência. ....	22
Tabela 4 - Guia de valores utilizado na fase de teste do módulo de transmissão. ....	42
Tabela 5 - Parâmetros de entrada do modelo orbital.....	53
Tabela 6 - Parâmetros utilizados nesta simulação do detector.....	54
Tabela 7 - Parâmetros da descrição do transmissor. ....	57



## LISTA DE SIGLAS, ACRÔNIMOS E ABREVIACÕES

AMS:	<i>Analog and Mixed-Signal</i>
ANA:	Agência Nacional de Águas
ANEEL:	Agência Nacional de Energia Elétrica
ARGOS:	<i>Satellite Data Collection System</i>
ASIC:	<i>Application Specific Integrated Circuit</i>
CAD:	<i>Computer Aid Design</i>
CBERS:	<i>China Brazil Earth Resources Satellite</i>
CMCD:	Centro de Missão Coleta de Dados
CNES:	<i>National d'Etudes Spatiales</i>
CRC:	<i>Cyclic Redundancy Check</i>
CRN:	Centro Regional do Nordeste
DC:	<i>Direct Current</i>
DE:	<i>Discrete event</i>
ETR:	Estação Terrena de Recepção
EIRP:	<i>Effetive Isotropic Radiation Power</i>
GMT:	<i>Greenwich Mean Time</i>
INPE:	Instituto Nacional de Pesquisas Espaciais
LNA:	<i>Low Noise Amplifier</i>
MoC:	Modelo Computacional
MTR:	Mini-transmissores Remotos
NASA:	<i>National Aeronautics and Space Administration</i>
NOAA:	<i>National Oceanic and Atmospheric Administration</i>
OSCI:	<i>Open SystemC Initiative</i>

PCD:	Plataformas de Coleta de Dado
RdP:	Rede de Petri
RF:	Radio Frequência
SCD:	Satélite de Coleta de Dados
SCDB:	Sistema de Coleta de Dados Brasileiro
SDF:	<i>Synchronous Data Flow</i>
SIVAN:	Sistema de Vigilância da Amazônia
SoC:	<i>System on Chip</i>
UFRN:	Universidade Federal do Rio Grande do Norte
UHF:	<i>Ultra High Frequency</i>
UIT:	União Internacional de Telecomunicações
UTC:	<i>Universal Time Coordinated</i>

## 1 INTRODUÇÃO

O Brasil concentra atualmente uma riqueza incalculável de recursos naturais em seus 8,5 milhões de quilômetros quadrados. Este vasto país, de dimensões continentais, apresenta um imenso potencial energético e uma biodiversidade inexplorada que o coloca em vantagem estratégica em relação aos outros países. Tendo em vista a garantia de sua biodiversidade, mostrou-se necessário integrar e guarnecer seus recursos expandindo suas fronteiras para o espaço.

O Instituto Nacional de Pesquisas Espaciais (INPE), procurando concretizar esta conquista, desenvolveu no programa espacial Brasileiro, o Sistema de Coleta de Dados Brasileiro (SCDB), um dos mais bem sucedidos projetos do programa espacial Brasileiro. O SCDB consiste em uma rede de satélites de órbita baixa que captam e retransmitem via satélite os dados ambientais coletados de diversos pontos do território nacional (Kuga, 2007). Estes dados, por sua vez, são provenientes de Plataformas de Coleta de Dados (PCDs) ou de Mini-transmissores Remotos (MTRs) espalhados pelo país. Atualmente esse sistema possui alto custo tecnológico para países em desenvolvimento, provocando assim, uma dependência no fornecimento de equipamentos por outras nações que detêm esta tecnologia. No entanto, o INPE, como responsável pelo desenvolvimento e operação desse sistema, se encontra frente ao novo cenário de desenvolvimento tecnológico em microeletrônica no Brasil. Esta posição, favorável, se mostra propícia para o desenvolvimento de sistemas com tecnologia nacional.

A importância técnico científica que a microeletrônica impõe na vida moderna é indubitável. Computadores, celulares entre outros dispositivos eletrônicos são provas irrefutáveis desta revolução tecnológica. Tal evolução na complexidade dos produtos eletrônicos conduziu, gradativamente, à elevação do nível de abstração nas atividades de projetos. Neste contexto, a evolução da capacidade de integração dos sistemas frente a este cenário de evolução tecnológica exige, cada vez mais, redução de custos na hora de projetar um sistema. Nestas últimas décadas, as linguagens de modelagem de *hardware* têm sido amplamente empregadas no desenvolvimento de sistemas digitais, analógico e RF, bem como em sistemas mecânicos e ópticos. Apesar disso, as linguagens HDLs demonstraram

limites nas simulações em conjunto de diferentes domínios. Mostrando assim, limites quanto à sua utilização.

Seguindo outra vertente na descrição de sistemas, o SystemC-AMS que é dedicada à simulação de sistemas heterogêneos, oferece um recurso poderoso para descrição de modelos mistos. A biblioteca de classes SystemC-AMS é constituída por uma organização de modelos computacionais (MoC) otimizados para cada domínio de simulação, o que possibilita maior velocidade durante as simulações (Michel Vasilevski, 2007).

SystemC-AMS, foi utilizada para modelar uma plataforma de identificação, compatível com o sistema de coleta de dados brasileiro do INPE. O foco deste trabalho consiste em utilizar esta plataforma de simulação para avaliar e extrair um conjunto de especificações que servirão como orientações na validação do projeto. Tendo em vista este objetivo, o presente trabalho encontra-se dividido da seguinte forma: Na seção seguinte serão apresentadas as características e especificações exigidas pelo sistema de coleta de dados brasileiro, a fim de auxiliar a modelagem; Na seção 2 serão expostas as arquiteturas e modelos utilizados para a caracterização de todos os modelos empregados nas simulações. A seção 3 tratará das descrições acerca das plataformas de simulação SystemC e SystemC-AMS, além de apresentar as implementações das descrições em SystemC e SystemC-AMS. Seção 4 será exposto um banco de teste que avaliará o desempenho da plataforma de identificação. E na seção 5 serão apresentadas as conclusões acerca do trabalho, assim como as suas perspectivas e contribuições.

## 2 SISTEMA DE COLETA DE DADOS BRASILEIRO (SCDB)

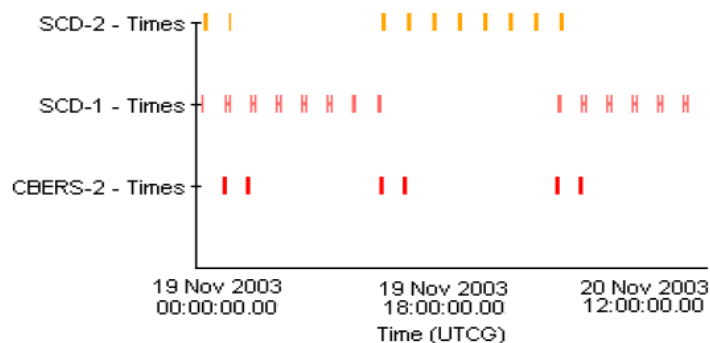
### 2.1 INTRODUÇÃO

O Sistema de Coleta de Dados Brasileiro, como apresentado em (Kuga, 2007) é constituído por uma constelação de satélites SCD1, SCD2 (Satélite de Coleta de Dados) e CBERS2 (Satélite Sino-Brasileiro de Recursos Terrestres – *China Brazil Earth Resources Satellite*) que captam os sinais provenientes das plataformas de coleta de dados e das micro transmissoras (PCDs e MTRs) espalhadas pelo território brasileiro e retransmitem para as Estações Terrenas de Recepção (ETRs) em Cuiabá ou Alcântara.

O SCD1 e o SCD2 foram colocados em órbitas circulares semelhantes, da ordem de 750km de altitude, e 25° de inclinação em relação ao plano do Equador da Terra. O CBERS-2 se encontra em uma órbita polar, isto é, com inclinação do plano orbital da ordem de 90° em relação ao plano do Equador. Devido à altitude orbital, mas também ao fato de que suas órbitas são quase circulares, estes elementos se deslocam a uma velocidade média de aproximadamente 27.000km/h, completando uma volta em torno do globo terrestre em aproximadamente 100 minutos. Com isso, eles realizam, na sua totalidade, 14 órbitas a cada 24 horas.

Destas 14 órbitas diárias, devido à inclinação do plano orbital em relação ao Equador, em média, apenas oito passagens consecutivas são visíveis pela estação de Cuiabá. Deste modo, a cada 24 horas, ocorre um intervalo de tempo correspondente em média a seis órbitas (aproximadamente 10 horas) em que não há passagens de um dos satélites sobre Cuiabá. Para que se tenha uma visibilidade constante, os intervalos de passagens diárias são intercalados, de modo que um Satélite cubra a passagem dos outros, como ilustrado na Figura 1, que mostra as passagens típicas dos satélites SCD1, SCD2 e CBERS-2 sobre a

estação terrena de Cuiabá ocorridas no período de 48 horas UTC<sup>1</sup>. Conforme mencionado, as passagens do SCD2 cobrem todo o período diário em que não ocorrem passagens do SCD1 e vice-versa.



**Figura 1 - Passagens Diárias dos Satélites SCD1, SCD2 e CBERS-2 Sobre a Estação de Rastreo de Cuiabá.**

Os satélites funcionam como retransmissores de mensagens, como representado na Figura 2. Quando um dos satélites do sistema entra na região de alcance (visibilidade) de uma das estações de recepção (Cuiabá ou Alcântara), a comunicação satélite-solo (*Downlink*) é estabelecida. Os sinais continuamente transmitidos pelas PCDs ou MTRs ao satélite são transmitidos durante sua passagem e, imediatamente, retransmitidos em direção à estação receptora. Na estação, esses dados são armazenados e, após cada passagem dos satélites, são transmitidos ao Centro de Missão Coleta de Dados (CMCD). As estações, por sua vez, demodulam as mensagens e as enviam para o centro de missões de coleta de dados em Cachoeira Paulista. As informações recebidas no centro de missões são armazenadas em forma de dados e, em seguida, são tratadas para serem disponibilizadas via internet aos usuários do sistema (Rae, 2005).

<sup>1</sup> UTC (“Universal Time Coordinated” ou Tempo Universal Coordenado) é sinônimo de GMT (“Greenwich Mean Time” ou Tempo Médio de Greenwich). Trata-se do fuso horário de referência a partir do qual se calculam todas as outras zonas horárias do mundo. É o sucessor do GMT, que por convenção deveria ser utilizado o termo UTC. Contudo, por razões históricas, o termo GMT tende a se perpetuar.



**Figura 2 - Os componentes de enlace da comunicação do Sistema Brasileiro de Coleta de Dados.**  
**FONTE:** <http://satelite.cptec.inpe.br/PCD/sistema.jsp>

Assim, a comunicação entre as redes de PCDs e as estações de recepção é estabelecida através dos satélites. As plataformas são geralmente configuradas para transmitir, a cada 200 segundos, por volta de 32 bytes de dados úteis. Os satélites SCD1, SCD2 e CBERS-2 operam em duas faixas de frequência UHF para recepção das mensagens transmitidas pelas plataformas de Coleta de Dados (em torno de 401,62MHz e de 401,65 MHz). Os sinais das PCDs visíveis são recebidos a bordo dos satélites e retransmitidos para o solo na Banda de frequência S (em torno de 2.267,52MHz). No solo, os sinais são recebidos nas estações de Cuiabá ou Alcântara e enviados para o CMCD para o processamento, armazenamento e disseminação aos usuários. O envio desses dados ao usuário é feito através da Internet.

## 2.2 PLATAFORMA DE COLETA DE DADOS

As PCDs - Plataformas de Coleta de Dados surgiram da necessidade de várias empresas, instituições e órgãos governamentais (Agência Nacional de Águas - ANA, Agência Nacional de Energia Elétrica - ANEEL, etc.) em obter informações colhidas em lugares remotos como demonstrado na Figura 3. Estas PCDs e MTRs adquirem dados climáticos, ambientais, hidrológicos e de movimentação de embarcações ou de animais e as transmitem para os satélites pertencentes a MECB.

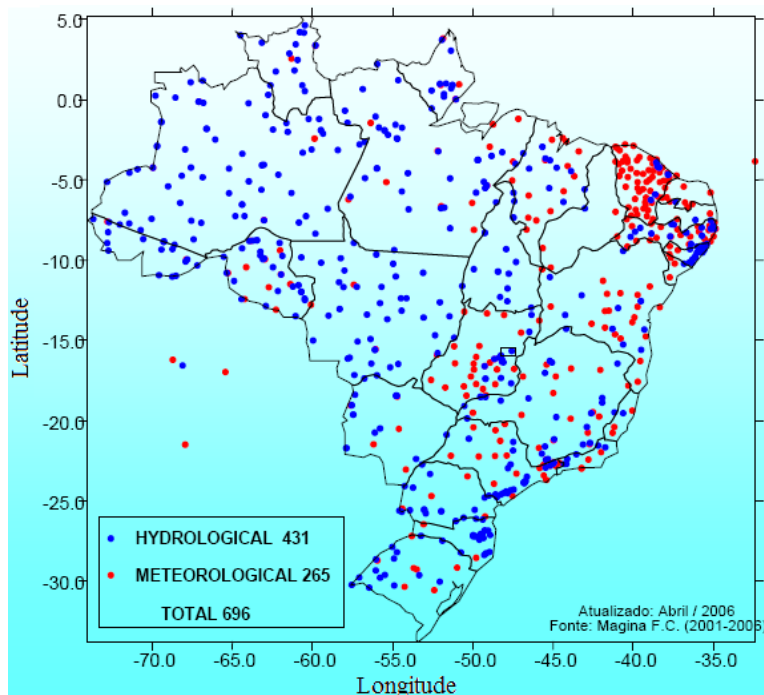


Figura 3 - Localização das PCD do sistema SCD1 realizada em 2006.

FONTE: <http://satellite.cptec.inpe.br/PCD/redes.jsp>

As MTRs e PCDs, após a aquisição dos dados, transmitem os dados em pacotes de mensagens que possuem duração de 360 a 920ms, com intervalos de recorrência entre 40 e 220s. Estas mensagens são codificadas em Biphase-L (conhecido também como codificado *Manchester*) e seguem uma taxa de transmissão de 400 bps, com modulação de fase  $\pm 60^\circ$ , na frequência de 401,62MHz (faixa MECB) ou 401,65MHz (faixa ARGOS)<sup>2</sup> (Rae, 2005). Os dados enviados para os satélites que se encontram em uma altitude média de 750 km, sofrem ajustes de frequência devido ao efeito *Doppler*<sup>3</sup> na faixa de 401,635MHz  $\pm 30$ kHz.

A Codificação *Manchester* é caracterizada por efetuar a transição na metade do intervalo do Bit, como ilustrado na Figura 4.

<sup>2</sup> ARGOS - *Satellite Data Collection System*: O serviço Argos foi desenvolvido no âmbito de um Memorando de Entendimento (MOU) entre o *Centre National d'Etudes Spatiales* (CNES, a agência espacial francesa), a *National Aeronautics and Space Administration* (NASA, E.U.A.) e da *National Oceanic and Atmospheric Administration* (NOAA, Administração Nacional Oceânica e Atmosférica dos E.U.A. ).

<sup>3</sup> Variação do comprimento de onda provocada pela variação da fonte em relação ao observador.



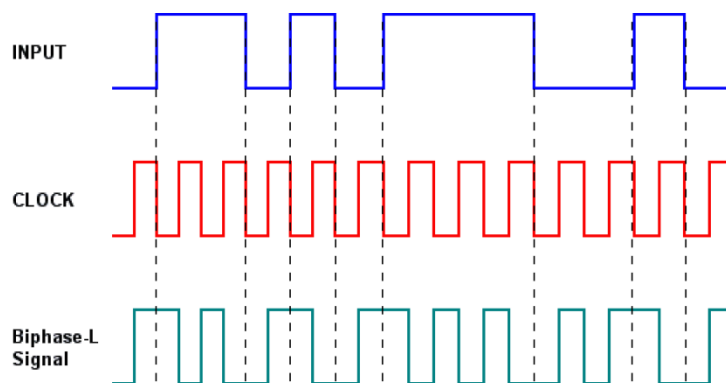


Figura 4 - Característica temporal do Sinal Codificado em *Biphase-L*

Esta codificação possui algumas vantagens que envolvem a auto-sincronização dos dados pela simples recuperação do *Clock* integrado à mensagem e a detecção de erro proveniente da falha de transição durante as transmissões da mensagem. Isto possibilita seu emprego no Sistema de Coleta de Dados.

Os dados recebidos pelo satélite requerem um sinal de reconhecimento em portadora pura, com um intervalo de duração de  $160 \pm 2,5$ ms. Assim, esta mensagem está estruturada em um pacote conforme apresentado na Tabela 1.

Tabela 1 - Formato da mensagem transmitida pelas PCDs e MTRs do SBCD.

Portadora Pura	Portadora modulada em Fase						
160ms	A	B	C	D	E	F	G

A – Sincronização de Bit:	15 bits “111111111111111”
B – Sincronização de palavra:	8 bits “00010111”
C – Inicialização da mensagem	1 bit “1”
D – N <sup>o</sup> de blocos de 32 bits (N <sub>b</sub> )	4 bits (N <sub>b</sub> = 1,2,...,8)
E – Identificação da PCD:	20 bits
F – Dados dos Sensores:	32bits * N <sub>b</sub>
G– CRC (opcional):	8 bits

Caracterizando a transmissão em termos de potência, tem-se que as PCDs e MTRs, tipicamente empregadas no SCDB apresentam variações de potência de transmissão. Estas variações obedecem aos critérios de sistema e atividade, conforme exemplificado na Tabela 2.

Tabela 2 - Características de potência das plataforma de coletas de dado do SBCD

PCD/MTR:	PCDs da ENGESPACO	PCDs do SIVAN	MTRs Baixa Potência
Potência Transmissão	3,0dBW(2,0W)	1,46dBW (1,4W)	-6,0dBW (0,25W)
Ganho da Antena	2,0dB/ elevação 90° -2,0dB/ elevação 15°	0,0dB/ elevação 90° 3,0dB/ elevação 15°	0,0dB/ elevação 90° 0,0dB/ elevação 15°

Seguindo o regulamento internacional de radiocomunicação da União Internacional das Telecomunicações (UIT), o enlace de descida do satélite foi dimensionado para transmitir o máximo de fluxo de potência estipulado pelo órgão internacional. A Tabela 3, que segue a regulamentação da UIT, apresenta um resumo de potências permitidas na superfície da terra compreendendo a faixa S<sup>4</sup>, cuja banda de frequência é de 4 kHz.

**Tabela 3 - Regulamento da UIT para máxima densidade de fluxo de potência.**

<b>Ângulo de elevação da Radiação Incidente</b>	<b>Máxima Densidade de Fluxo Recomendado</b>
$0^\circ \leq \phi < 5^\circ$	-154dB (W/m <sup>2</sup> )
$5^\circ \leq \phi < 25^\circ$	$-154 + \frac{\phi - 5^\circ}{2^\circ} \text{ dB (W/m}^2\text{)}$
$25^\circ \leq \phi \leq 90^\circ$	-144dB (W/m <sup>2</sup> )

---

<sup>4</sup>Faixa de frequência que varia de 2 a 4 GHz. A banda S é utilizada para aplicações de Radar, meteorologia e comunicação.

### 3 MODELAGEM

#### 3.1 INTRODUÇÃO

A crescente complexidade dos sistemas impôs metodologias que são destinadas ao aperfeiçoamento do fluxo de projeto em circuitos integrados. Seguindo o objetivo proposto por este trabalho, destaca-se a definição dos blocos como o início da modelagem. Esta interação que se caracteriza por definir o comportamento da plataforma de identificação em blocos. Partindo dos requisitos impostos pelo sistema de coleta de dados, foi definido que o sistema deveria identificar a presença do satélite a fim de evitar a transmissão contínua da plataforma. Com este requisito definido, inicia-se a caracterização dos modelos em um alto nível de abstração e interação entre os blocos. Nesta etapa, o sistema será caracterizado comportamental e matematicamente.

#### 3.2 DIAGRAMA DE BLOCOS

A plataforma foi estruturada em três blocos que se relacionam com o propósito de cumprir o objetivo de transmitir os dados da identificação da plataforma. A representação do sistema, baseada nos requisitos, deu origem a três blocos responsáveis por: realizar a detecção do satélite, o controle da plataforma e a transmissão dos dados ao satélite, conforme ilustrado na Figura 5.

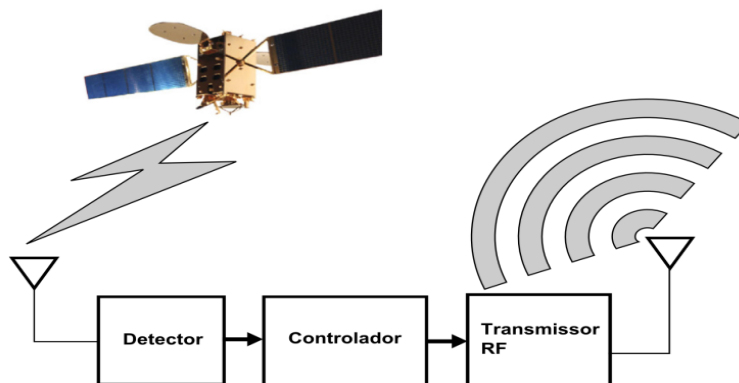
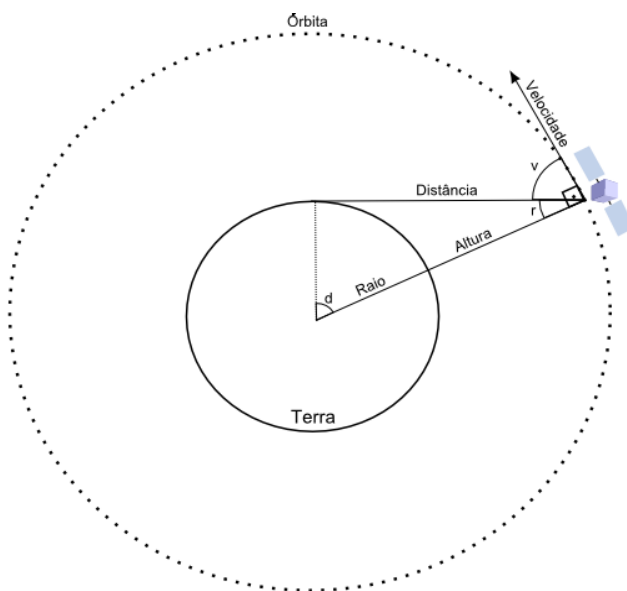


Figura 5 - Blocos que representam a organização da plataforma de identificação.

Como visto na ilustração anterior, a plataforma foi estruturada a partir de três subsistemas: o detector, o controlador e o transmissor. Estes blocos incorporam, respectivamente, funções de gerenciamento de energia, controle da plataforma e modulação dos dados.

### 3.2.1 MODELO ORBITAL

O sistema de comunicação via satélite tem características orbitais que influenciam diretamente o desempenho dos sistemas de comunicação. Para isso, foi feita uma modelagem do sistema orbital que englobou o satélite e a plataforma de identificação. Esta modelagem considerou primeiramente, a definição da distância entre os pontos, como ilustrada no esquema da Figura 6. Esta distância em conjunto com o raio terrestre e a altitude do satélite caracterizam a órbita. O satélite em questão possui uma altitude em torno de 750km conforme exposto na seção 2.2. Segundo (Louis J. Ippolito, 2008), esta característica define que o satélite é de baixa órbita aproximando-se de uma circunferência. Esta definição é caracterizada por apresentar poucas perdas, aplicação bastante empregada em sistemas de rastreamento e comunicação móvel que exigem baixa potência. O fator limitante para estes sistemas que operam em órbita baixa é o curto período de operação, isto é, curtos períodos de visibilidade (aproximadamente 10 minutos).



**Figura 6 - Diagrama representativo da trajetória do satélite em relação à plataforma de identificação.**

O modelo adotado para emular a passagem do satélite considerou a relação da distância com a velocidade angular “ $v_d$ ” (velocidade correspondente ao ângulo “ $\hat{d}$ ”). Considerando a distância, altura e raio da terra, representados respectivamente por  $D$ ,  $A$  e  $R$ , obtêm-se a equação (1) que representa esta relação.

$$D = \sqrt{(R + A)^2 + R^2 - 2 \cdot (R + A) \cdot R \cdot \cos(v_d \cdot t - 26,6^\circ)} \quad (01)$$

O valor inicial do ângulo que corresponde a  $-26,6^\circ$  na equação (1) representa a elevação  $0^\circ$  do satélite em relação a plataforma de identificação.

### 3.2.1.1 Efeito Doppler

Dando continuidade com a caracterização do modelo orbital, incluiu-se a variação da frequência (efeito Doppler). Esta variação, que é originária da velocidade relativa entre o receptor e transmissor, é obtida a partir da equação descrita por:

$$\left\{ \begin{array}{l} F_r = \text{Frequência de recepção} \\ F_t = \text{Frequência de transmissão} \\ V_o = \text{Velocidade do observador} \\ V_t = \text{Velocidade do transmissor} \\ V = \text{Velocidade de propagação} \end{array} \right. \quad F_r = F_t \frac{V \pm V_o}{V \pm V_t} \quad (02)$$

A velocidade de transmissão, que coincide com a distância relativa entre o receptor e o transmissor, é obtida com a reorientação do vetor, sendo desta forma expressa por:

$$V_t = \cos(\hat{v}) \cdot V_{satélite}, \quad \hat{v} = \frac{\pi}{2} - \hat{r} \quad (03)$$

onde o ângulo  $\hat{r}$  é obtido de acordo com a seguinte equação:

$$\hat{r} = \cos^{-1} \left( \frac{R^2 - (R+A)^2 - D^2}{-2 \cdot (R+A) \cdot D} \right) \quad (04)$$

Esta característica é importante durante a detecção do sinal recebido, pois insere o efeito da variação de frequência na simulação.

### 3.2.1.2 Modelo do Canal

O canal de comunicação do satélite apresenta características como perdas do espaço livre, ganho da antena, banda passante, distribuição de potência e ruído. Este canal de comunicação estabelecido entre o satélite e a plataforma é um ponto crítico no projeto e no desempenho do sistema de comunicação, pois define as características do sinal captado que servirá para despertar o sistema e para dimensionar o transmissor. Os parâmetros que caracterizam este canal são o ganho da antena do transmissor  $g_t$ , a potência de transmissão  $p_t$ , o ganho da antena de recepção  $g_r$ , potência recebida  $p_r$  e a distância entre os dispositivos  $r$ . A combinação dos parâmetros  $g_t$  e  $p_t$  conforme demonstrado na equação (5) determinam uma figura de mérito importante para análise do sinal transmitido permitindo caracterizar a potência efetiva da radiação isotrópica (Effective Isotropic Radiation Power - EIRP).

$$eirp = p_t \cdot g_t, \text{ Watts} \quad (05)$$

Considerando a transmissão do satélite como um modelo isotrópico, pode-se definir que a densidade do fluxo de potência no receptor é expressa conforme a equação (6).

$$pfd_r = 10 \log \left( \frac{p_t \cdot g_t}{4\pi r^2} \right), \text{ Watts/m}^2 \quad (06)$$

Este parâmetro é importante na definição do canal, pois está relacionando a potência requerida para a recepção conforme exposto na Tabela 3. O fluxo sofre perdas devido ao espaço livre. A perda do espaço livre existe em todos os sistemas de propagação por rádio frequência, que utiliza meios uniformes como a atmosfera para a propagação do sinal. Esta perda, conforme expressa em (Louis J. Ippolito, 2008), possui a seguinte representação em termos de frequência:

$$I_{fs} = \left( \frac{4\pi \cdot r \cdot f}{c} \right)^2 \quad (07)$$

Para uma distância em quilômetros e uma frequência em GHz, tem-se o resultado expresso em dB dado por (8).

$$L_{fs} (dB) = 20 \cdot \log(f) + 20 \cdot \log(r) + 92,44 \quad (08)$$

Para completar as perdas inerentes ao canal é importante destacar outras perdas que têm origem de fenômenos meteorológicos como chuva, densidade de nuvens entre outros, possam ser considerados. Desta forma, destacamos as outras perdas como  $l_o$ . Com todos os

elementos básicos do canal equacionados, é possível determinar a potência na antena do receptor. Assim, a potência nos terminais da antena do receptor,  $p_r$ , é dado por:

$$p_r = eirp \cdot \left( \frac{1}{l_{fs} \cdot l_o} \right) \cdot g_r \quad (09)$$

### 3.2.1.3 Modelo do Ruído

O ruído tem origem aleatória e descorrelacionada do sinal de interesse. Existem muitas fontes de ruído em um sistema de comunicação. Segundo (Razavi, 1998), a forma mais significativa de contribuição de ruído presente em sistemas de comunicação é o ruído térmico. A potência de ruído  $n_N$  que segue a definição de Nyquist é definida em (10) como a quantificação da potência do ruído térmico.

$$n_N = k \cdot T_e \cdot b_N, \text{ Watts} \quad (10)$$

O  $k$  representa a constante de Boltzmann,  $T_e$  representa a temperatura equivalente da fonte de ruído e  $b_N$  é a largura de banda. Este ruído, que independe da frequência de operação, pode ser definido por um elemento passivo que dissipa potência de ruído por cada unidade de largura de banda. Esta nomenclatura expressa a densidade de potência de ruído que é apresentada em (11).

$$n_o = \frac{n_N}{b_N}, \text{ Watts/Hz} \quad (11)$$

O ruído térmico é considerado o ruído dominante em um sistema, sua representação pode ser composta por duas fontes na entrada, onde uma delas corresponde à fonte de tensão conectada em série e a outra corresponde a fonte de corrente conectada em paralelo com a carga de entrada. A análise do ruído inicia com a fonte  $\overline{V_{Rant}^2}$ , que representa a parcela de ruído proveniente da resistência na antena. Esta parcela, juntamente com as fontes de ruídos  $\overline{V_n^2}$  e  $\overline{I_n^2}$ , que representam a contribuição da resistência de entrada, integram o ruído dominante na entrada do sistema. Esta representação define o modelo do ruído na entrada do sistema analisado, conforme ilustrado na Figura 7.

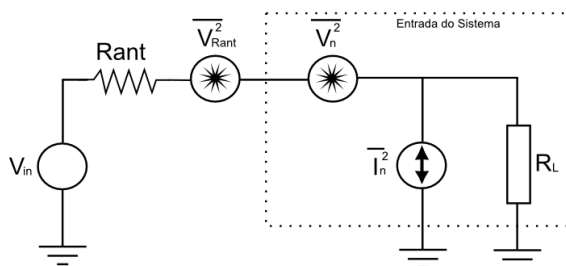


Figura 7 - Esquema da representação do ruído na entrada do sistema.

Definido o modelo de representação do ruído em (Razavi, 1998), observa-se a existência de uma fonte de ruído externa ao sistema. A relação de ruído na entrada do sistema com o ruído na saída exprime a contribuição do sistema no acréscimo de ruído no sinal de interesse. Gerando, desta forma, um parâmetro que serve de métrica para análise do desempenho do sistema, isto é, a figura de ruído.

A análise feita para o ruído na entrada da plataforma segue a proposta apresentada na Figura 7. Este diagrama produz a seguinte representação matemática, que caracteriza o ruído na entrada do sistema, dada por:

$$v_n^2 = \left[ 2\sqrt{\frac{4.k.T}{R_l}} \cdot (R_{ant} \parallel R_l) + \sqrt{4.k.T \cdot R_l \cdot \frac{R_l}{R_{ant} + R_l}} \right]^2 + v_{ant}^2 \cdot \frac{R_{ant}^2}{(R_l + R_{ant})^2} \quad (12)$$

### 3.2.2 MÓDULO DE DETECÇÃO

O circuito de sensoriamento remoto via satélite possui características que demandam consumo de energia elevado, durante o período de atuação do dispositivo. Os amplos períodos de recorrência tornam-se um empecilho, pois os dispositivos de pequeno porte, que utilizam fontes limitadas de energia para o processamento e transmissão têm pequena autonomia. Além disso, o difícil acesso ao local em que o dispositivo se encontra, torna custosa a substituição ou recarga destas fontes. Apesar do período de atuação do equipamento ser relativamente amplo, a transmissão ocorre em intervalos curtos de duração devido à janela de visibilidade do satélite. Esta característica possibilita a otimização da plataforma em termos de energia, tornando-o portátil.

A administração de energia é um ponto importante para prolongar a vida útil destes dispositivos, pois permite o consumo de energia somente quando este for necessário. Uma técnica amplamente utilizada em dispositivos de comunicação para minimizar o consumo de



energia é a detecção de potência (P. Kolinko, 2007). Esta técnica, comumente chamada de *wake-up*, consiste em utilizar a detecção de potência para indicar o momento em que o sistema entrara em operação.

A concepção do módulo detector vem da necessidade de identificação do satélite a partir do sinal recepcionado pela estação terrena, compreendido em uma banda de frequência específica. Quando o sinal detectado for suficiente para acionar os circuitos, o controlador inicia o processamento da informação, caso contrário, o sistema permanecerá em hibernação até que a detecção ocorra. Esta seqüência de operações, que foi baseado em (Cecilia Carbonelli, 2005) e (Alberto Rabbachin, 2004), constitui o módulo detector como ilustrado na Figura 8.

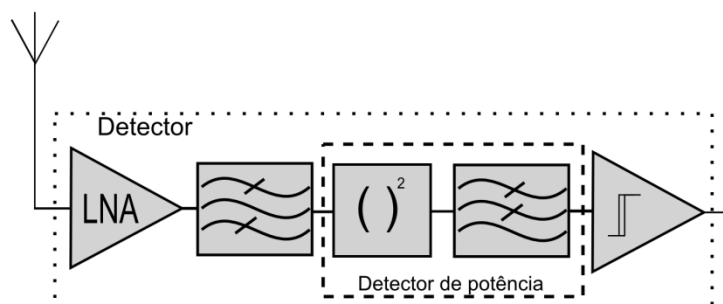


Figura 8 - Diagrama de blocos do modelo do processamento do detector.

Considere um sinal periódico  $x(t)$ , cuja frequência angular  $w_s$  corresponde à frequência da portadora,  $\theta$  a fase e  $A_s(t)$  à amplitude do sinal conforme em (13).

$$x(t) = A_s(t)\cos(w_s t + \theta) \quad (13)$$

Este sinal captado pela antena é condicionado à função quadrática que modifica o sinal recebido. A função quadrática dobra a frequência do sinal e adiciona uma componente DC proporcional a potência do sinal.

$$x^2(t) = A_s^2(t) \left( \frac{1}{2} + \frac{1}{2} \cos(2w_s t + 2\theta) \right) \quad (14)$$

Esta componente continua é utilizada como indicador de nível do sinal recebido. A seguir, cada componente do sistema é modelado.

### 3.2.2.1 Modelo de Amplificador de Baixo Ruído

Um Amplificador de baixo ruído (Low Noise Amplifier - LNA) é um importante componente em aplicações que envolvem Rádio Freqüência, pois amplifica pequenos sinais em uma ampla faixa de freqüência, contribuindo pouco com o ruído. Este dispositivo apresenta, também, comportamento não-linear, que adicionam harmônicos ao sinal.

Um LNA é comumente especificado a partir do seu ganho de potência, fator de ruído e do ponto de compressão de terceira ordem (IP3). O modelo do amplificador especifica o ganho de potência como:

$$A_{Ganho} = \sqrt{10^{\frac{ganho em dB}{10}} \cdot \frac{R_{out}}{R_{in}}} \quad (15)$$

Esta expressão permite obter o ganho não linear em termos de componentes harmônicas de primeira e terceira ordem, como demonstrado a seguir:

$$v_{out} = a \cdot v_{in} - b \cdot v_{in}^3 \quad (16)$$

onde

$$a = A_{ganho} \quad (17)$$

$$b = \frac{4}{3} \frac{a}{ip3^2} \quad (18)$$

$$ip3 = \sqrt{10^{\frac{ip3(dBm) - 30}{10}} \cdot 2 \cdot R_{in}} \quad (19)$$

O ruído térmico que o amplificador acrescenta ao sinal é descrito em termos de figura de ruído (NF).

$$N_o = 4KT(NF - 1) \quad (20)$$

onde  $K$  é a constante de *Boltzmann*,  $T$  representa a temperatura.

### 3.2.2.2 Filtro Passa-Banda

O modelo de filtro passa-banda de primeira ordem tem sua caracterização definida pela frequência central e pelo fator de qualidade, conforme representados na função de transferência seguinte:

$$H(s) = \frac{s \cdot \frac{w_o}{Q}}{s^2 + s \cdot \frac{w_o}{Q} + w_o^2} \quad (21)$$

### 3.2.2.3 Detector de Potência

O modelo detector emprega a detecção quadrática do sinal  $x$  a partir da modelagem expressa na equação (22), considerando somente o componente de segunda ordem:

$$y = K \cdot \left( \frac{\beta_d^2}{2!} \cdot x^2 \right) \quad (22)$$

As variáveis  $K$  e  $\beta_d$  são consideradas constantes que definem a responsividade do modelo do detector. A equação (22) em conjunto com a equação (14) permite definir o componente contínuo. Esta definição permite com a filtragem deste sinal obter somente a componente contínua que determina o nível do sinal transmitido pelo satélite.

### 3.2.2.4 Comparador com Histerese

O comparador é empregado na verificação do nível do sinal entregue pelo detector. A implementação da histerese tem um papel fundamental, pois evita as oscilações na saída lógica do comparador, provocada por pequenas flutuações. Assim, sua caracterização se resume à realização do teste lógico precedido de um intervalo de guarda. A especificação do detector parte da definição exposta em (Phillip A. Allen, 2002), que caracterizada na equação (23), estabelece a entrada mínima exigida no comparador.

$$V_{in(mim)} = \frac{V_{out(max)} - V_{out(mim)}}{A_0} \quad (23)$$

Sendo cumprido esse requisito, tem-se o ponto de partida para a definição da responsividade do módulo detector.

### 3.2.3 MÓDULO CONTROLADOR

O conceito de sistema pode ser entendido, como uma combinação de elementos que atuam em conjunto para realizar uma determinada tarefa, que não seria possível de ser concretizada pelas partes constituintes individualmente, ou seja, um sistema corresponde a uma interação de componentes ou uma associação de funções. Realizando uma análise qualitativa e quantitativa do sistema, a fim de obter um modelo que possa ser entendido como uma simples duplicata comportamental do próprio sistema. Neste contexto, foi idealizado um bloco destinado a comandar este sistema, nomeado de Módulo Controlador: Figura 9.

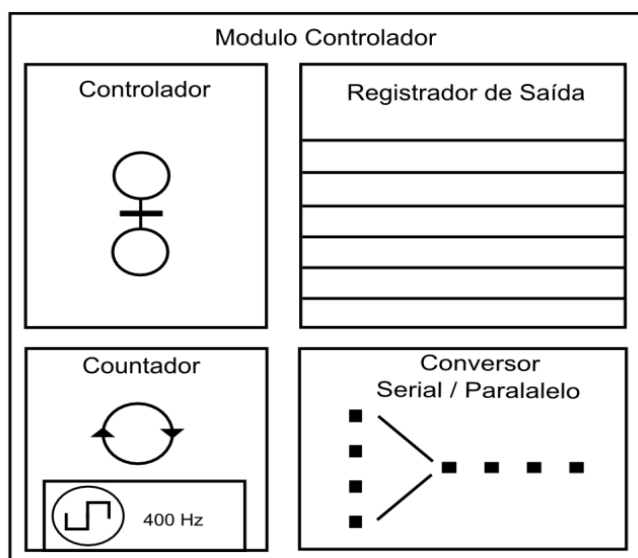


Figura 9 - Conjuntos de blocos que constituem o sistema controlador

O módulo controlador foi definido a partir de quatro componentes fundamentais que o compõem. As características destes componentes seguiram as definições do sistema de coleta de dados. O sub-bloco controlador, que se caracteriza por efetuar o controle das operações nos componentes registrador, contador e no conversor serial. Esse controle que é feito, assincronamente, foi modelado e implementado a partir de uma descrição em SystemC de uma Rede de Petri (RdP) (Uzam, et al., 2008). O componente contador efetua a contagem do tempo, além de controlar a taxa de transmissão do conversor serial.

Já o componente “registrador de saída” possui a formatação da memória de dados, conforme a Tabela 1. Por fim, o bloco conversor que é responsável por transmitir os dados serialmente do registrador no formato *Biphase-L* para o módulo transmissor.

### 3.2.3.1 Controlador

O controlador, que foi implementado através de uma RdP, pode ser entendida como uma representação gráfica que possui uma relação explícita com as funções de transições do sistema a evento discreto (Christos G. Cassandras, 2008). A rede manipula os eventos de acordo com as regras que permitem a evolução da RdP. Isto admite uma ampla representação estrutural do sistema discreto, dependendo somente da complexidade com que a rede é concebida. Esta característica, somada à ampla gama de técnicas que já foram desenvolvidas para analisar essas redes, motivou o emprego desta ferramenta de análise e síntese para automatização da plataforma. A linguagem que é utilizada para descrever a RdP na visão de autômatos é representada por uma quintupla  $(P, T, A, w, x)$  onde:

$P$  é a definição do vetor finito e não-vazio de lugares;

$T$  é a definição do vetor finito e não-vazio de transições;

$A \subseteq (P \times T) \cup (T \times P)$  define as matrizes de pré-condições e de pós-condições (MURATA, 1989), isto é, define as ligações e orientações dos arcos que vão do lugar até a transição e da transição até o lugar;

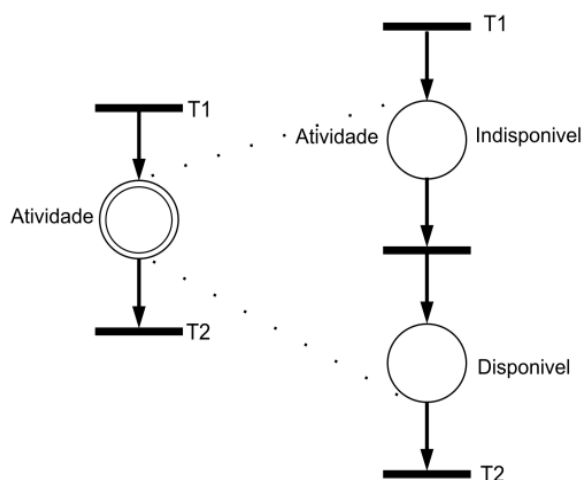
$w: A \rightarrow \{1,2,3, \dots\}$  representa o conjunto de valores dos pesos de cada arco;

$x$ : representa o vetor coluna da quantidade de fichas existentes em cada lugar, ou seja,  $x = [x(p_1), x(p_2), \dots, x(p_n)] \in \mathbb{N}^n$ .

As redes regulares, segundo (Valette, 2002), descrevem um relação de casualidade com os eventos. Num evento “a” que é a causa de “b”, pressupõe-se uma ordenação destes eventos no tempo. Associando a esta ordenação no tempo, com as componentes da rede, obtêm-se a associação de tempo para as transições ou para os lugares.

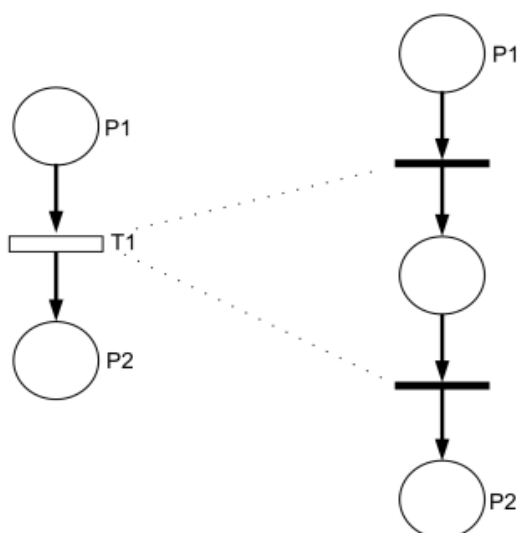
A representação do lugar com associação ao tempo é uma representação que significa a duração da atividade. Assim, as atividades que ocorrerem após o disparo de uma transição corresponde ao horário em que a atividade teve início, ou seja, a transição seguinte

só estará disponível para disparar após um determinado tempo que está associado ao lugar. Esta representação é ilustrada na Figura 10.



**Figura 10 - Representação temporal referente ao lugar.**

A associação do tempo às transições só faz sentido, quando uma transição é interpretada como uma atividade e não como um evento instantâneo, podendo ser considerado, desta forma, que esta transição é constituída por uma seqüência: transição, lugar e transição. Assim a primeira transição corresponde ao início da atividade e a última transição corresponde ao final da atividade. E isso corresponde ao atraso da transição T1 como exemplificado na ilustração Figura 11.



**Figura 11 - Representação da associação de tempo a transição.**

Com o objetivo de tornar à RdP mais compacta possível, a mesma foi acrescida de noção de tempo às transições, a partir da inclusão de um vetor de disparos no tempo (Ramchandani, 1974). Essa modificação, representada por uma sêxtupla  $(P, T, A, w, x, V)$ , é

bastante importante para o controle da plataforma. Esta estrutura, agora, associada a um conjunto de transições temporizadas  $T_D \subseteq T$  onde  $V = \{v_j : t_j \in T_D\}$ , define uma seqüência de tempo dado por:

$$v_j = \{v_{j,1}, v_{j,2}, \dots\}, t_j \in T_D, v_{j,k} \in \mathbb{R}^+, k = 1, 2, \dots \quad (24)$$

O modelo é baseado no disparo dinâmico das transições temporizadas da forma  $\{\tau_{j,1}, \tau_{j,2}, \dots\}$ ,  $j = 1, \dots, m$ , onde  $\tau_{j,k}$  é o  $k$ ésimo disparo da transição  $t_j$  para  $k = 1, 2, \dots$ . Deste modo, a RdP temporizada se decompõem em vários sub-modelos de lugares temporizados independentes, isto é, onde cada lugar é formado por uma seqüência de equações de estados que encadeiam, por sua vez, sucessões de disparos. Esta decomposição de disparos dinâmicos defini  $\pi_{i,k}$  como o instante em que o lugar  $p_i$  recebe uma ficha no  $k$ ésimo disparo, onde  $k = 1, 2, \dots$ . Esta seqüência no tempo, encadeia os disparos tornando o comportamento da RdP como o de um sistema de fila.

Partindo para a modelagem do sistema de controle e empregando RdP temporizada, como ilustrado na Figura 12, tem-se a interpretação das ações definidas pelo conjunto dos lugares  $P = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8\}$  como:

- P<sub>1</sub> - Satélite fora do alcance;
- P<sub>2</sub> - Osciladores desligados;
- P<sub>3</sub> - Sistema em espera;
- P<sub>4</sub> - Osciladores em operação;
- P<sub>5</sub> - Transmissão da portadora pura;
- P<sub>6</sub> - Transmissão da mensagem contida no registrador;
- P<sub>7</sub> - Sistema em espera de recorrência;
- P<sub>8</sub> - Satélite na visibilidade.

A Marcação inicial em que se encontra a rede é dada por  $M_0 = \{1, 1, 1, 0, 0, 0, 0, 0\}$ , com a temporização das transições dadas por  $T_D = \{a, d, e, f, g\}$ .

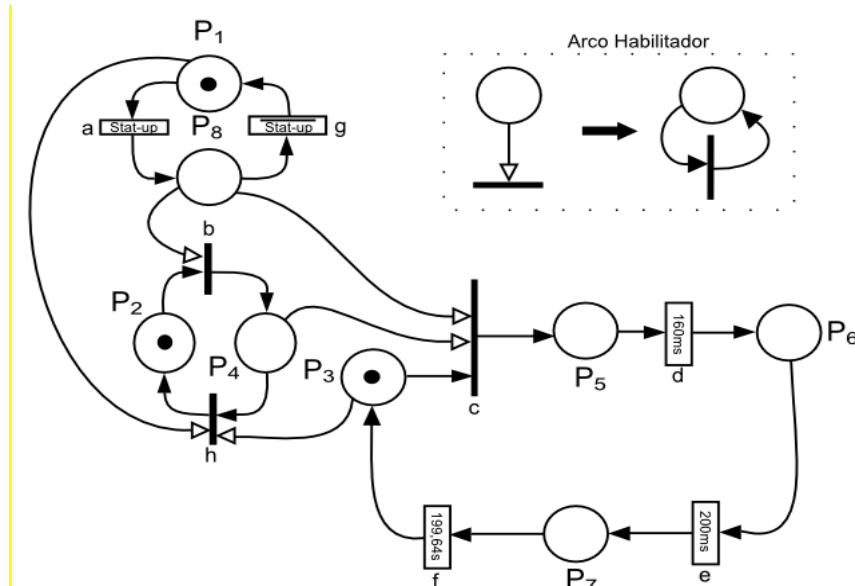


Figura 12 - Rede de Petri caracterizando o sistema satélite e o controle da plataforma.

A representação dos disparos é descrita por um conjunto de equações de disparos dinâmicos que convertem a RdP em um simples sistema de filas. Sabendo que as representações de chegada das fichas no  $k$ ésimo instante são dadas por:  $\pi_{P_1,k}, \pi_{P_2,k}, \pi_{P_3,k}, \pi_{P_4,k}, \pi_{P_5,k}, \pi_{P_6,k}, \pi_{P_7,k}, \pi_{P_8,k}$  e com suas adjacentes transições dadas por:

$a_k$  –  $K_{th}$  início da detecção do satélite (*start-up*);

$b_k$  –  $K_{th}$  início do funcionamento dos osciladores;

$c_k$  –  $K_{th}$  condição estável de transmissão;

$d_k$  –  $K_{th}$  transmissão da portadora pura;

$e_k$  –  $K_{th}$  modulação da informação;

$f_k$  –  $K_{th}$  intervalo de guarda para reinício da transmissão;

$g_k$  –  $K_{th}$  sinal do satélite fora do alcance;

$h_k$  –  $K_{th}$  fim do funcionamento dos osciladores.

Pode-se obter, a partir da marcação inicial, o primeiro instante em que cada lugar recebe suas fichas. Com a marcação  $M_0$  definida pela Figura 12, obtém-se o tempo inicial dos lugares como:  $\pi_{P_1,1} = 0$ ,  $\pi_{P_2,1} = 0$  e  $\pi_{P_3,1} = 0$ . Esta condição, inicia a evolução da RdP de forma seqüencial. Assim tem-se que:

$$a_k = \pi_{P_1,k} + v_{a,k} \quad k = 1,2, \dots, \quad (25)$$

$$b_k = \max\{\pi_{P_2,k}, \pi_{P_8,k}\} \quad k = 1,2, \dots, \quad (26)$$



$$c_k = \max\{\pi_{P_4,k}, \pi_{P_3,k}, \pi_{P_8,k}\} \quad k = 1, 2, \dots, \quad (27)$$

$$d_k = \pi_{P_5,k} + v_{d,k} \quad k = 1, 2, \dots, \quad (28)$$

$$e_k = \pi_{P_6,k} + v_{e,k} \quad k = 1, 2, \dots, \quad (29)$$

$$f_k = \pi_{P_7,k} + v_{f,k} \quad k = 1, 2, \dots, \quad (30)$$

$$g_k = \pi_{P_8,k} + v_{g,k} \quad k = 1, 2, \dots, \quad (31)$$

$$h_k = \max\{\pi_{P_1,k}, \pi_{P_3,k}, \pi_{P_4,k}\} \quad k = 1, 2, \dots, \quad (32)$$

e

$$\pi_{P_1,k} = g_{k-1} \quad k = 2, 3, \dots, \quad \pi_{P_1,1} = 0 \quad (33)$$

$$\pi_{P_2,k} = h_{k-1} \quad k = 1, 2, \dots, \quad \pi_{P_2,1} = 0 \quad (34)$$

$$\pi_{P_3,k} = f_{k-1} \quad k = 2, 3, \dots, \quad \pi_{P_3,1} = 0 \quad (35)$$

$$\pi_{P_4,k} = b_k \quad k = 1, 2, \dots, \quad (36)$$

$$\pi_{P_5,k} = c_k \quad k = 1, 2, \dots, \quad (37)$$

$$\pi_{P_6,k} = d_k \quad k = 1, 2, \dots, \quad (38)$$

$$\pi_{P_7,k} = e_k \quad k = 1, 2, \dots, \quad (39)$$

$$\pi_{P_8,k} = a_k \quad k = 1, 2, \dots, \quad (40)$$

Combinando (25) à (40), pode-se substituir os lugares  $\pi_{P_1,k}, \pi_{P_2,k}, \pi_{P_3,k}, \pi_{P_4,k}, \pi_{P_5,k}, \pi_{P_6,k}, \pi_{P_7,k}, \pi_{P_8,k}$  e obter para  $k = 1, 2, \dots$ ,

$$a_k = g_{k-1} + v_{a,k} \quad k = 1, 2, \dots, \quad g_0 = 0 \quad (41)$$

$$b_k = \max\{a_k, h_{k-1}\} \quad k = 1, 2, \dots, \quad h_0 = 0 \quad (42)$$

$$c_k = \max\{b_k, f_{k-1}, a_k\} \quad k = 1, 2, \dots, \quad f_0 = 0 \quad (43)$$

$$d_k = c_k + v_{d,k} \quad k = 1, 2, \dots, \quad (44)$$

$$e_k = d_k + v_{e,k} \quad k = 1, 2, \dots, \quad (45)$$

$$f_k = e_k + v_{f,k} \quad k = 1, 2, \dots, \quad (46)$$

$$g_k = a_k + v_{g,k} \quad k = 1, 2, \dots, \quad (47)$$

$$h_k = \max\{g_{k-1}, f_{k-1}, b_k\} \quad k = 1, 2, \dots, \quad (48)$$

Isolando  $a$  e  $f$  que corresponde, respectivamente, ao início da detecção e ao início da recorrência, obtém-se uma relação importante de entrada e de saída para este sistema de fila, dado por:

$$f_k = \max\{a_k, f_{k-1}\} + (v_{d,k} + v_{e,k} + v_{f,k}) \quad k = 1, 2, \dots, \quad f_0 = 0 \quad (49)$$

$$a_{k+1} = a_k + (v_{a,k} + v_{g,k}) \quad k = 1, 2, \dots, \quad a_0 = 0 \quad (50)$$

As relações (49) e (50) caracterizam o comportamento dinâmico do sistema de fila em termos de tempo de chegada e de partida das fichas no sistema. Estas equações demonstram, que a partida da ficha ocorre no instante  $(v_{d,k} + v_{e,k} + v_{f,k})$ , depois da  $(k - 1)$  éxima partida. Para simplificar a análise, assumiu-se que  $(C_a = v_{a,k} + v_{g,k})$  e que  $(C_d = v_{d,k} + v_{e,k} + v_{f,k})$  para  $k = 1, 2, \dots$ . Nota-se que  $C_a$  representa a constante de tempo de recorrência da visibilidade do satélite e que  $C_d$  representa a constante de processamento da plataforma. Assumindo que  $C_a > C_d$ , que é aceitável, pois se fosse considerado de outra forma, seria obtido um tempo de visibilidade do sinal proveniente do satélite insuficiente para se realizar todo o processo de identificação e envio da mensagem, causando assim, instabilidade no sistema.

Utilizando a álgebra de Dioides para reescrever as equações (51) e (52), a fim de, analisar a estabilidade do sistema temos:

$$f_k = (a_k \otimes C_a) \oplus (f_{k-1} \otimes -L) \quad (51)$$

$$a_{k+1} = (a_k \otimes C_d) \oplus (f_{k-1} \otimes C_d) \quad (52)$$

A variável  $-L$  representa um valor suficientemente pequeno e negativo para tornar a sentença  $\max\{a_k + C_a, f_{k-1} - L\} = a_k + C_a$  verdadeira. Com base em (51) e (52) mudamos a notação para a matricial, tendo desta forma a expressão obtida em (53).

$$\begin{bmatrix} a_{k+1} \\ f_k \end{bmatrix} = \begin{bmatrix} C_a & -L \\ C_d & C_d \end{bmatrix} \begin{bmatrix} a_k \\ f_{k-1} \end{bmatrix} \quad (53)$$

Definindo a matriz como:

$$x_k = \begin{bmatrix} a_{k+1} \\ f_k \end{bmatrix}, \quad A = \begin{bmatrix} C_a & -L \\ C_d & C_d \end{bmatrix} \quad (54)$$

onde:

$$x_{k+1} = Ax_k, \quad x_0 = \begin{bmatrix} C_a \\ 0 \end{bmatrix} \quad (55)$$

A equação (53) apresenta a similaridade com os modelos lineares de sistemas dinâmicos de variáveis contínuas, cuja matriz  $A$  apresenta informações relevante ao sistema dinâmico estudado. Entre estas informações, pode-se destacar as condições de estabilidade dada por  $C_a > C_d$ , de oscilação dada por  $C_a = C_d$ , além da informação de instabilidade do sistema dada pela condição  $C_a < C_d$ .

### 3.2.3.2 Registrador de Saída

O registrador de saída é um elemento de memória com capacidade de armazenar 80 bits de dados. Este elemento é responsável por efetuar o armazenamento dos dados necessário para formar o bloco de mensagem, descrito na Tabela 1. Sua finalidade é de armazenar antes de ser enviado ao conversor paralelo-serial.

### 3.2.3.3 Contador

O sub-bloco contador fornece a base de tempo necessária para que os circuitos digitais sincronizem suas operações. Este circuito que é composto por um *clock* (relógio interno), que envia sinais regulares aos contadores e a outros circuitos, a fim de fornecer a referência de tempo para as atividades requisitadas. O *clock* é um pulso alternado de sinais de tensão. Cada um destes intervalos de tempo é delimitado pelo início da descida do sinal que corresponde a um ciclo de execução ou pulso. O tempo de ciclo equivale ao período de oscilação que, seguindo as especificações, necessita ter uma frequência de 400Hz com tolerância de 5Hz. Assim, o objetivo primordial deste sub-bloco é fornecer uma base de tempo a partir do relógio (*clock*) e informar através dos contadores o tempo de execuções das operações. Segundo as especificações exigidas pelo controlador, que neste caso é a rede de Petri, têm-se que os dois contadores deverão respeitar os tempos exigidos nas transições  $d$  e  $f$  com base no *clock* de 400Hz que correspondem, respectivamente, a 64 e 79.856 vezes.



A saída codificada se destina a controlar o módulo transmissor. O resultado comportamental da descrição deste circuito está ilustrado na Figura 15, onde se constata que as informações seriais e a informação de sincronização do *clock* estão encapsuladas em uma única via de dados.

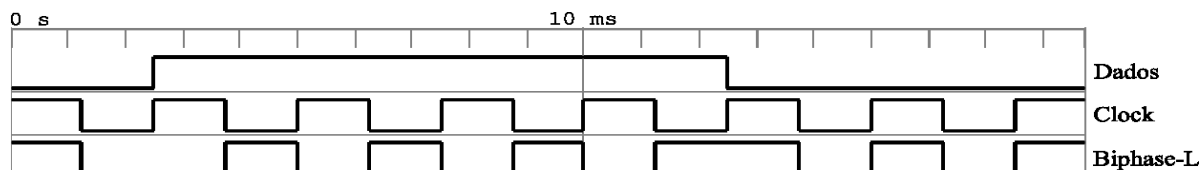


Figura 15 - Resultado da simulação da codificação dos dados em *Biphase-L* na frequência de 400Hz.

### 3.2.4 MÓDULO DE TRANSMISSÃO

O módulo de transmissão abrange um sistema de comunicação digital no domínio da Rádio Frequência (RF). Este módulo tem como finalidade modular a informação digital disponibilizada serialmente codificada em *Biphase-L*. A modulação desta informação segue uma variação de fase de  $\pm 60^\circ$  para o satélite SCD1 da MECB, a uma frequência de 401,62 MHz, como especificado na seção 2.2. Existem vários tipos de modulações que oferecem soluções em termos de custos e qualidade para o receptor. Contudo a modulação em fase apresenta uma boa imunidade ao ruído (SERIOJA O. TATU, 2005). Esta imunidade traz vantagens para o sistema de transmissão de dados, pois o enlace de subida (*uplink*) apresenta perdas inerentes à atmosfera que provocam diminuição da relação sinal-ruído do sinal transmitido. Seguindo esta linha, a modulação em fase do sistema foi modelada usando a combinação dos sinais em fase e quadratura como ilustrado na Figura 16.

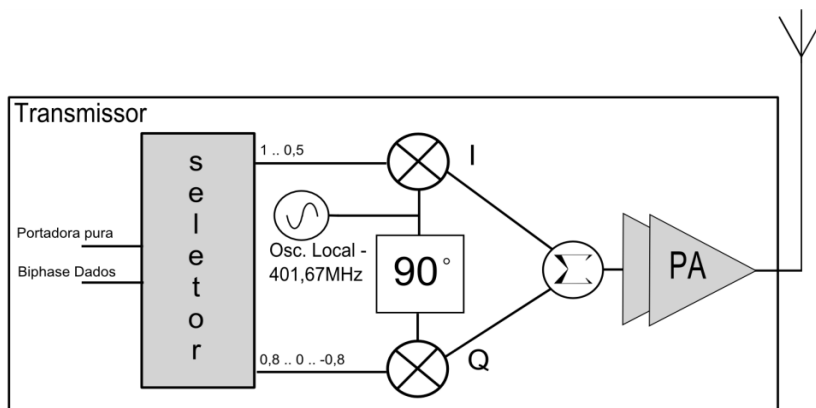


Figura 16 – Representação gráfica do módulo de transmissão comportamental implementada em SystemC-AMS.

A modulação ocorre com a separação do sinal em duas vias. Uma via  $I(t)$  que está em fase com o sinal e a outra  $Q(t)$  que está em quadratura. Ambos os sinais  $I(t)$  e  $Q(t)$  são multiplicados, respectivamente, pelo sinal portador  $\cos(\omega_p t)$  e pelo sinal defasado exatamente de  $90^\circ$ , para obter o sinal modulado resultante  $S(t)$ . Para o sinal recém formado, a mudança de fase ocorre instantaneamente. Desta forma, o sinal  $S(t)$  pode ser expresso por:

$$S(t) = I(t) \cdot \cos(\omega_p t) + Q(t) \cdot \cos\left(\omega_p t + \frac{\pi}{2}\right) \quad (56)$$

Tendo em vista o embasamento teórico da modulação efetuada pela plataforma, observa-se que a escolha das amplitudes das componentes  $I(t)$  e  $Q(t)$  do sinal é um fator a ser controlado. Estas amplitudes são definidas pelo componente “Seletor” que, de acordo com os comandos realizados nas entradas: **Biphase Dados** e **Portadora pura** variam as amplitudes de  $I(t)$  e  $Q(t)$ , para que se obtenha o sinal desejado conforme mostrado na Tabela 4.

**Tabela 4 - Guia de valores utilizado na fase de teste do módulo de transmissão.**

Entradas digitais		Saídas analógicas	
Biphase Dados	Portadora pura	I(t)	Q(t)
0	1	1*X	0
1	1	1*X	0
0	0	0,5*X	-0,866*X
1	0	0,5*X	0,866*X

“X” corresponde a o valor da amplitude do sinal desejado.

### 3.2.4.1 Oscilador

O oscilador consiste em um circuito autônomo que segue os critérios de *Barckhausen* e gera em sua saída um sinal periódico. Este circuito possui um conjunto de características que definem seu comportamento, entre elas podemos destacar a saída que acompanha a seguinte representação para um tom:

$$V_{osc}(t) = V_{DC} + V_{Lin} \cdot \cos(2\pi \cdot f \cdot t + \theta_{fase}) \quad (57)$$

onde a tensão  $V_{lin}$  corresponde à amplitude da tensão na saída que é definida por:

$$V_{Lin} = \sqrt{2 \cdot r_{out} \cdot 10^{\frac{V_{dBm} - 30}{10}}} \quad (58)$$

Outra característica que define o oscilador está relacionada ao ruído de fase. A representação deste fenômeno, conforme descrito em (Leung, 2002), é representado por:

$$V_{osc}(t) = V_{DC} + V_{Lin} \cdot \cos(2\pi \cdot f \cdot t + \Delta\theta_{fase}(t)) \quad (59)$$

Sendo  $\Delta\theta_{fase}(t) = A_{\Delta\theta}(f_n) \cdot \cos(2\pi \cdot f_n \cdot t)$  uma variação randômica da fase devido ao ruído existente no oscilador que, por sua vez, modula o sinal na fase. Pode-se assumir que esta variável é uma função randômica da frequência  $\Delta\theta_{fase}(f_n)$ . Esta função gera uma seqüência aleatória de pares de pulsos espaçados de  $\pm f_n$  da frequência central como visto na Figura 17.

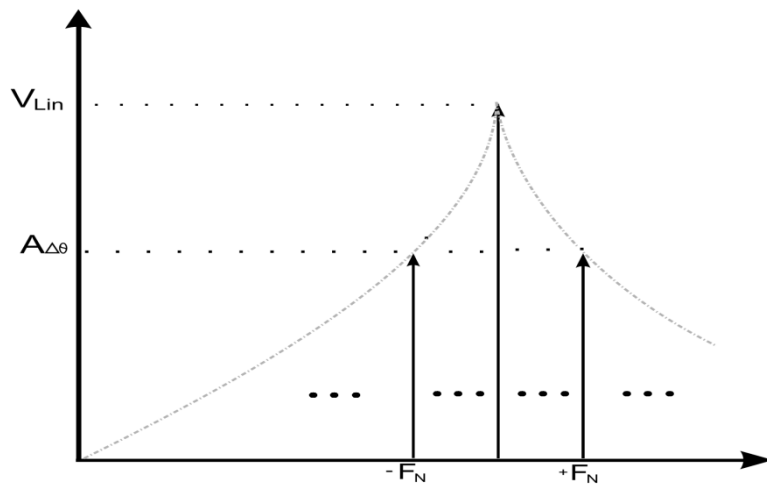


Figura 17 - Espectro na frequência de um oscilador com a injeção de ruído de fase.

#### 3.2.4.2 Defasador

O defasador pode ser modelado por um filtro passa - tudo, (Rogers, et al., 2003), que tem como função, defasar o sinal em  $\theta^\circ$ . A caracterização do modelo se procede na definição dos pólos e zeros em função da frequência e da defasagem. A função que descreve este comportamento está expressa na função de transferência apresentada em (60).

$$H(s) = \frac{\frac{\tan(\theta)}{\omega_c} s - 1}{\frac{\tan(\theta)}{\omega_c} s + 1} \quad (60)$$

### 3.2.4.3 Misturador

O misturador ou conversor de frequência converte uma frequência em outra frequência com um ganho de conversão (Leung, 2002). A caracterização do misturador se inicia com a definição matemática simplificada da célula de Gilbert dada por:

$$V_{rf}(t) = V_{hf}(t) \cdot V_{lo}(t) \quad (61)$$

Considerando a não-linearidade do misturador, inclui-se a distorção de terceira ordem ao sinal. A não linearidade do misturador pode ser incluída no modelo, da seguinte maneira:

$$V_{rf}(t) = \alpha \cdot V_{rf}(t) - \gamma \cdot V_{rf}^3 \quad (62)$$

onde:

$$\alpha = \sqrt{10^{\frac{G_{dB}}{10} \cdot \frac{R_{out}}{R_d}}} \quad (63)$$

$$\gamma = \frac{4}{3} \frac{\alpha}{ip3^2} \quad (64)$$

$$ip3 = \sqrt{10^{\frac{ip3(dBm) - 30}{10}} \cdot 2 \cdot R_d} \quad (65)$$

Os valores dos sinais do Seletor são definidos pelos ganhos de conversão do misturador. O sinal proveniente dos misturadores são responsáveis por implementar a modulação da informação em banda básica.

### 3.2.4.4 Seletor

O componente seletor tem como designação traduzir os sinais digitais disponibilizados pelo controlador em nível de tensão. Estes níveis, por sua vez, são responsáveis por modular a informação em banda básica e têm a definição da saída correlacionada com a não-linearidade do misturador. Desta forma, a Tabela 4 que representa a transformação do sinal em banda básica terá seus valores alterados conforme a caracterização do ganho de conversão imposta pelo componente misturador.



#### 3.2.4.5 Amplificador de Potência

Amplificadores de potência, diferentemente dos amplificadores de pequenos sinais, operam com grandes sinais em condições de operação de forte não-linearidade. Apesar desta característica, o modelo adotado pouco se diferencia do amplificador convencional, isto é, a função básica que caracteriza esse amplificador é a mesma. Deste modo, a única diferença na modelagem deste elemento consiste nos valores dos parâmetros que o caracterizam e no ponto de operação. Assim, a modelagem do ganho e da não-linearidade corresponde ao modelo do amplificador de baixo ruído apresentados na seção 3.2.2.1.

## 4 DESCRIÇÃO EM SYSTEMC

### 4.1 INTRODUÇÃO

O projeto de sistemas em Chip (SoC) como definido em (H. Chang, 1999), integrou componentes digitais, analógicos, de Rádio Frequência (RF) e softwares, assim como partes não pertencentes ao domínio elétrico como sensores, dispositivos micro eletromecânicos e ópticos. Esta heterogeneidade mostrou a necessidade de modelos computacionais (MoC) robustos, capazes de simular a descrição do sistema em diferentes níveis. A descrição feita a partir de modelos os quais, espera-se um comportamento próximo ao físico, requisita uma plataforma capaz de simular conjuntamente estes aspectos heterogêneos com eficiência e consistência. Procurando atender as necessidades de descrever estes comportamentos, a associação sem fins lucrativos *Open SystemC Initiative* (OSCI) desenvolveu o *framework*<sup>5</sup> de código aberto, SystemC<sup>6</sup> para projeto, modelagem e verificação como ilustrado na Figura 18.

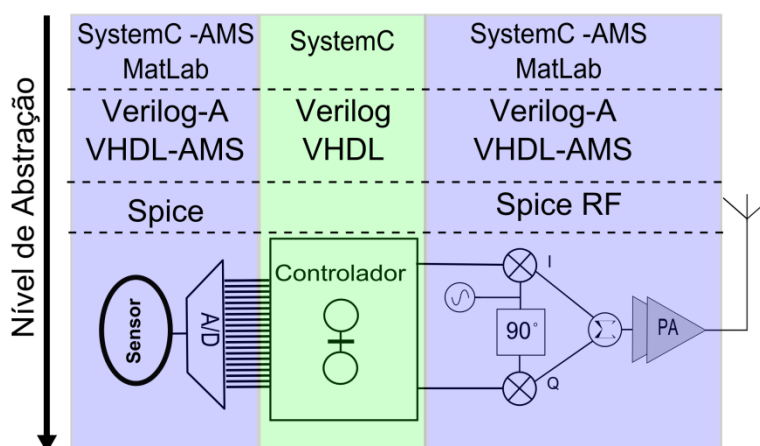


Figura 18 - Comparação entre as plataformas de simulações integradas por domínios e níveis de abstração.

<sup>5</sup> Abstração que integra códigos entre vários projetos de *software* fornecendo uma funcionalidade genérica ou específica dependendo da configuração. Isto é, um conjunto de classe que contribuem para uma determinada aplicação.

<sup>6</sup> Padrão IEEE 1666-2005, "Linguagem de Descrição SystemC".

## 4.2 SYSTEMC

SystemC é definido por (Bhasker, 2002) como um framework baseado na linguagem de programação orientada a objetos, por exemplo, C++, que incorpora características da linguagem como concorrência e temporização de eventos. O SystemC é entendida como uma classe da linguagem que não altera a sintaxe do C++, organizado como na Figura 19.

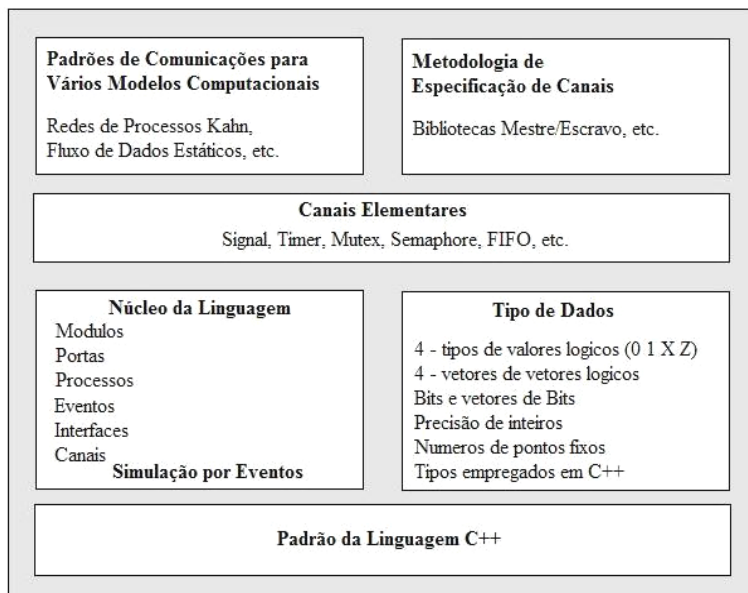


Figura 19 - Organização da bibliotecas de Classes do SystemC.

Os elementos do SystemC que formam a base do simulador são constituídos por portas, módulos, interfaces e canais. Estas portas definem as entradas e saídas dos módulos que representam as estruturas lógicas em alto nível nas formas de Métodos e *Threads*<sup>7</sup>.

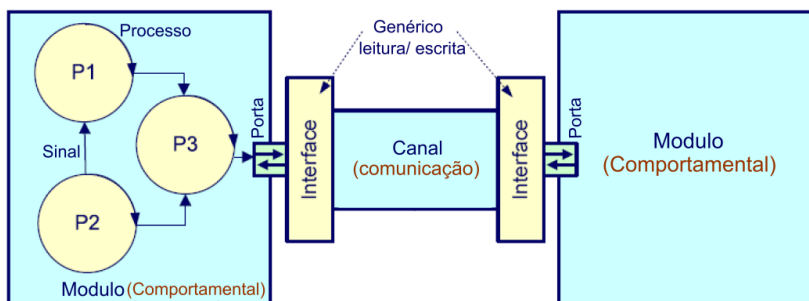
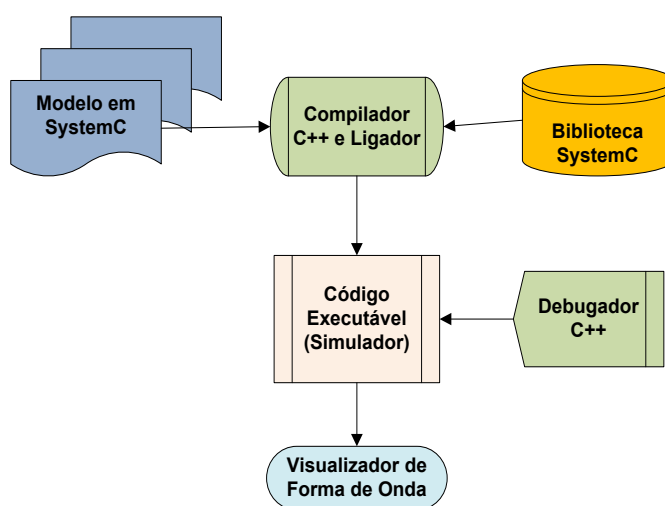


Figura 20 – Diagrama de abstração da implementação da arquitetura do SystemC.

<sup>7</sup> Conhecida também como linha de execução.

Estes processos se comunicam por canais primitivos como requisição, atualização e hierarquias (i.e., Mestre e escravo) assim como filas, sinais, temporizadores e semáforos. Os canais possuem variáveis internas que funcionam como registradores. Estas definições, como ilustradas na Figura 20, permitem a criação de especificações executáveis do sistema modelado como diagramado na Figura 21, proporcionando uma melhor compreensão do sistema. A compreensão do sistema envolve uma verificação no início da implementação seguido de uma validação do desempenho do sistema que impede interpretações ambíguas. Além de evitar inconsistências, erros e contribui para provar a integridade do sistema.



**Figura 21 - SystemC em ambiente de desenvolvimento C++.**

**Fonte: Slide SystemC-AMS 0.15RC4, Fraunhofer IIS/EAS Dresden, Fevereiro de 2004**

SystemC possui uma base de simulações coordenada por eventos que realizam o controle de tarefas a serem executadas (Donovan, 2004). O *kernel*<sup>8</sup> ilustrado na Figura 22 demonstra como ocorre a renovação de tarefas e a noção de tempo do sistema a partir da notificação dos eventos. Como o *kernel* de simulações do SystemC está totalmente voltado à notificações de eventos, o simulador não diferencia as tarefas por sua finalidade, mas tratando-as de forma genérica. A notificação dos eventos possui uma unidade indivisível de tempo que serve para ordenar os eventos no simulador.

---

<sup>8</sup> Parte central responsável pelo gerenciamento dos recursos computacionais do sistema ou núcleo.

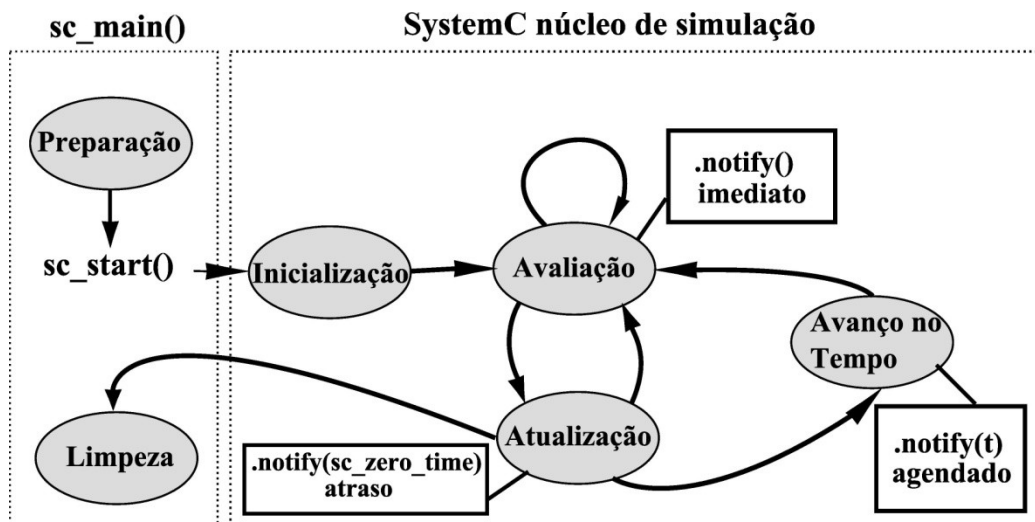


Figura 22 – Máquina de estados do núcleo de simulação do SystemC.  
 Fonte: SYSTEMC FROM THE GROUND UP, 2004, pg. 111.

Embora um evento seja avaliado em um ciclo de simulação, o tempo não avança. Essa avaliação é realizada nos estados de avaliação (*evaluate*) e atualização (*Update*), permitindo, assim, a sincronia do tempo de propagação dos sinais. A seqüência de operação do *kernel* do SystemC está exemplificada a seguir conforme em (Thorsten Grotker, 2002):

- a) Ao Inicializar cada processo, é executado uma vez para os Métodos ou até um ponto de sincronização (i.e., *wait*) para os *Threads*;
- b) O processo de avaliação seleciona os processos que estão prontos para execução ou prontos para continuar. Podendo executar as notificações imediatas, com custo de ciclo de execução, mas com atraso-zero (*zero delay*) na mesma fase de execução. Se as notificações imediatas forem mal utilizadas, a simulação pode não sair deste ponto, caindo, assim, em um estado de bloqueio, pois a avaliação se processa até que não existam mais processos prontos para continuar. Neste ponto, são feitos pedidos de atualização;
- c) A fase de atualização realiza pedidos de atualização que foram realizados no processo de Avaliação (*evaluate*). Estes são executados até que não tenha mais pedidos. A atualização do canal pode gerar novas notificações de eventos, algumas ou todas as notificações podem ser de atraso-zero;
- d) Se há notificações agendadas para o tempo atual de simulação (*delta delay*), se determina que os processos estejam aguardando e voltem à etapa de Avaliação;

- e) Se não há mais notificações agendadas (i.e., sem atraso-zero) para qualquer instante, o simulador termina;
- f) Se o evento de outras notificações para o futuro existir, avança-se o tempo no simulador para o próximo instante, que é a notificação agendada para mais cedo;
- g) Determinar os processos que estão prontos para execução por eventos que tenham notificações pendentes no tempo atual e repetir a partir do estágio de Avaliação.

### 4.3 SYSTEMC-AMS

SystemC havia sido definido para as simulações que envolveram sistemas de Eventos Discretos (DE). Contudo, procurando explorar suas características heterogêneas apresentada em (Alain Vachoux, 2003) propõe a extensão SystemC-AMS, conforme ilustrado na Figura 23, na qual propõe um meio mais eficiente de simular sistemas heterogêneos.

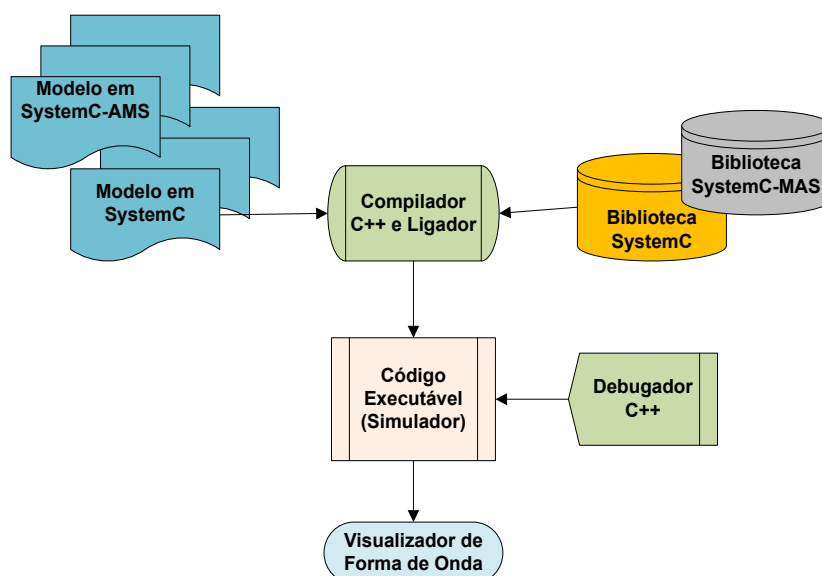
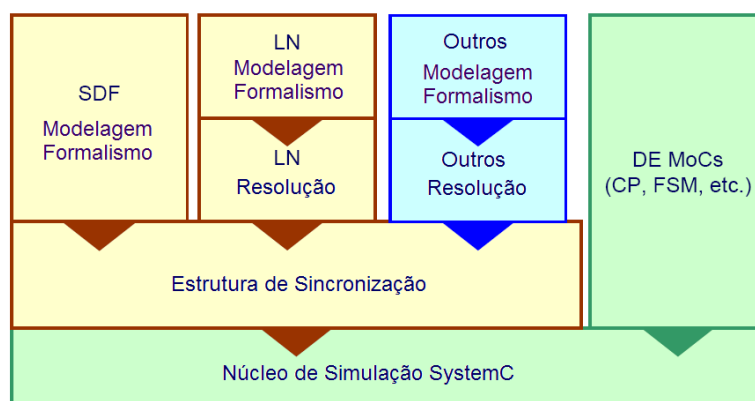


Figura 23 – SystemC e SystemC-AMS em ambiente de desenvolvimento C++.  
 Fonte: Slide SystemC-AMS 0.15RC4, Fraunhofer IIS/EAS Dresden, Fevereiro de 2004

A base de simulação para os sistemas de domínio contínuo foi sendo integrada à estrutura do SystemC e suas camadas de aplicações foram construídas como ilustrado na Figura 24 (Einwich, 2003). Os domínios visíveis ao usuário, representadas por *Synchronous Dataflow* (SDF), *Linear electrical Network* (LN) e outros, oferecem diferentes métodos descritivos para gerar modelos executáveis no domínio contínuo, isto é, modelos comportamentais, equações, funções de transferência e espaço de estados. As camadas de resolução oferecem diferentes métodos de resolução para simular as descrições AMS.



**Figura 24 – Estruturas de Camadas do SystemC-AMS está representada de core diferente.**  
 Fonte : Slide SystemC-AMS 0.15RC4, Fraunhofer IIS/EAS Dresden, Fevereiro de 2007

A última camada de aplicação antes da integração com o *kernel* do SystemC, implementa um mecanismo de sincronização das simulações do SystemC-AMS em suas diferentes camadas de aplicações com a parte dos eventos discretos do SystemC.

O sistema de simulação do domínio contínuo é um fluxo de *clusters*<sup>9</sup> (i.e, uma seqüência de *clusters*) que podem ser decompostos em vários blocos executando ordenadamente em agendamentos estáticos, isto é, o tempo só avança quando o último sinal do bloco é processado (Alain Vachoux, 2004).

---

<sup>9</sup> Aglomerado de blocos de menor complexidade lógica.

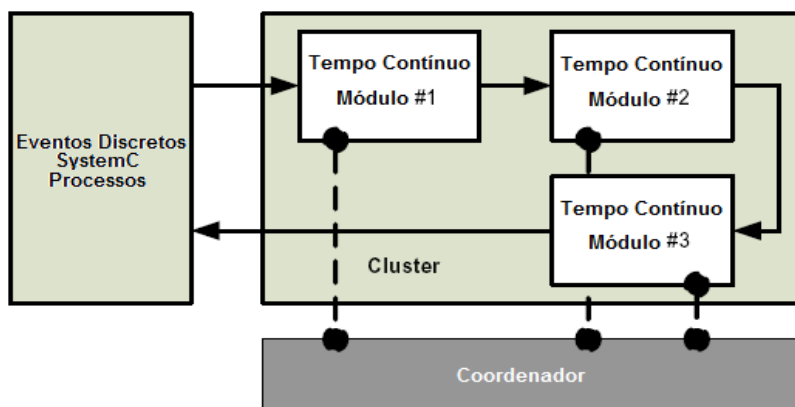


Figura 25 – Encapsulamento de blocos contínuos em um Fluxo de dados de um *Cluster*.

Cada fluxo de *cluster* é considerado um evento discreto denominado de processo do *cluster* que, por sua vez, é coordenado por um processo coordenador Figura 25. O coordenador é responsável por sincronizar as soluções do tempo contínuo e as ocorrências de eventos do *kernel* do SystemC.

#### 4.4 IMPLEMENTAÇÃO

Metodologias de projetos consistem num sistema de gestão com regras e procedimentos que visam garantir o melhor desempenho no tempo e na qualidade do projeto. Essa gestão tem como objetivo geral proporcionar a minimização das falhas inerentes ao processo. Neste contexto, o presente trabalho estabeleceu uma metodologia de projeto *top-down*, que foi definida em (SANDERSON, 1988) e expõe um fluxo de projeto que se inicia por meio de uma visão geral, conforme exposto na seção 2.2, seguindo para uma visão particular do problema definido pelo sistema como exposto na seção 3.2. Assim, visando obter previamente as especificações da plataforma sem a necessidade de implementação física, utilizou-se uma linguagem descritiva de alto nível que não requisitasse muito processamento como nas linguagens HDLs. Desta forma, a implementação de todos os blocos que constituem a plataforma de identificação empregou o SystemC como base de simulação, além de sua extensão SystemC-AMS.



#### 4.4.1 Descrição dinâmica do satélite e do canal

A modelagem da órbita do satélite SCD1, juntamente com o efeito Doppler, as características do canal e o ruído foram baseados no modelo apresentado na seção 3.2.1. Este modelo, ilustrado na Figura 26, foi descrito para gerar o sinal recepcionado pela plataforma.

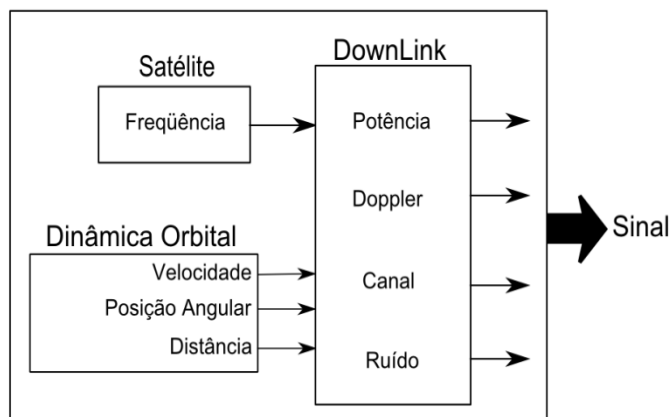


Figura 26 – Estrutura do modelo orbital

As descrições destes modelos, conforme apresentados nos apêndices A.1 e A.2, foram testados, segundo os parâmetros de entrada dos modelos apresentados na Tabela 5.

Tabela 5 - Parâmetros de entrada do modelo orbital.

Entidade	Componentes	Parâmetros						
		Raio	Altitude	Âng. inicial	Tempo	Âng. Final	Velocidade	
Órbita	Dinâmica orbital	6378,14 Km	700 Km	26,61°	2 s	-26,61°	27.000 Km/h	
		freq.	Pot. Trans.	Ganho trans.	Lo	Rin	Rout	Ant. ganho
	Downlink	2,26 GHz	-16,6dBW	1 dBW	-0,5dBW	50	50	0

Empregando um tempo de dois segundos para verificar seu comportamento, o modelo foi simulado e os resultados foram expostos na Figura 27. A dinâmica da órbita considerou três variáveis para modelar o seu comportamento. A elevação do satélite que foi obtida através das relações entre os ângulos da trajetória expostas na Figura 6, a distância entre a plataforma e o satélite que foi modelada a partir da equação (01) e a velocidade relativa descrita segundo a equação (03). As respectivas simulações estão expostas nos gráficos 2,3 e 4 na Figura 27. Por fim, o sinal transmitido considerando o modelo do canal é recebido conforme exposto no gráfico 1 na Figura 27.

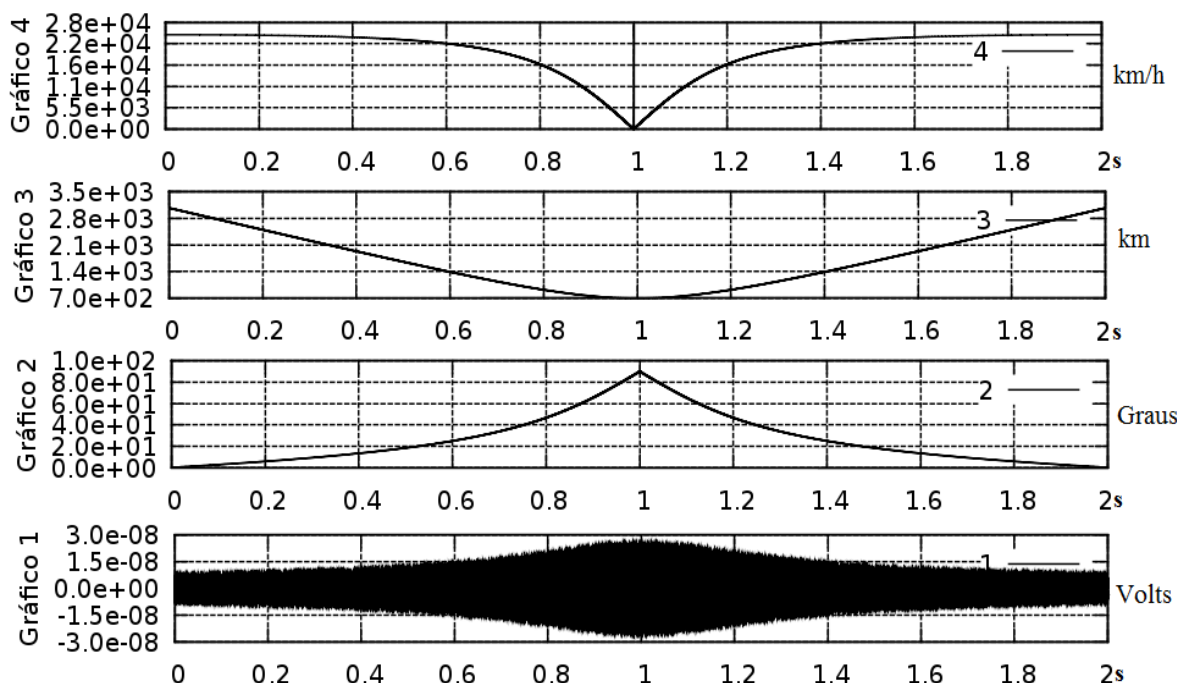


Figura 27 - Resultado de simulação extraída do modelo orbital.

#### 4.4.2 Módulo detector

O modelo do detector foi descrito a partir de equações apresentadas na seção 3.2.2, que levou em consideração as características do amplificador de baixo ruído, do filtro passa-banda, assim como do detector de potência e do comparador, conforme ilustrado na Figura 8. Os testes destes componentes utilizaram as descrições em SystemC-AMS apresentadas nos apêndices A.3,4,5 e 6, além dos parâmetros apresentados na Tabela 6.

Tabela 6 - Parâmetros utilizados nesta simulação do detector.

Entidade	Componentes	Parâmetros					
Detector	LNA	Ganho	IP3	NF	Rin	Rout	Temperatura
		12 dB	-30 dBm	10 dB	50	50	27 °C
	Filtro Passa-Banda	Freq.	Q				
		2,26 GHz	15				
	Detector	Freq.	Q	Rout	Responsividade		
		22.6Hz	10	50	1,60E+05		
	Comparador	Mag. Inf.	Mag sup				
		9 uV	10 uV				

Os resultados apresentados nos Gráficos 1, 2, 3, 4 e 5, da Figura 28, ilustram respectivamente, o sinal recebido, o espectro na saída do LNA, a resposta do filtro segundo a equação (52), a resposta do detecto ao sinal de entrada e a comparação lógica do sinal do detector.

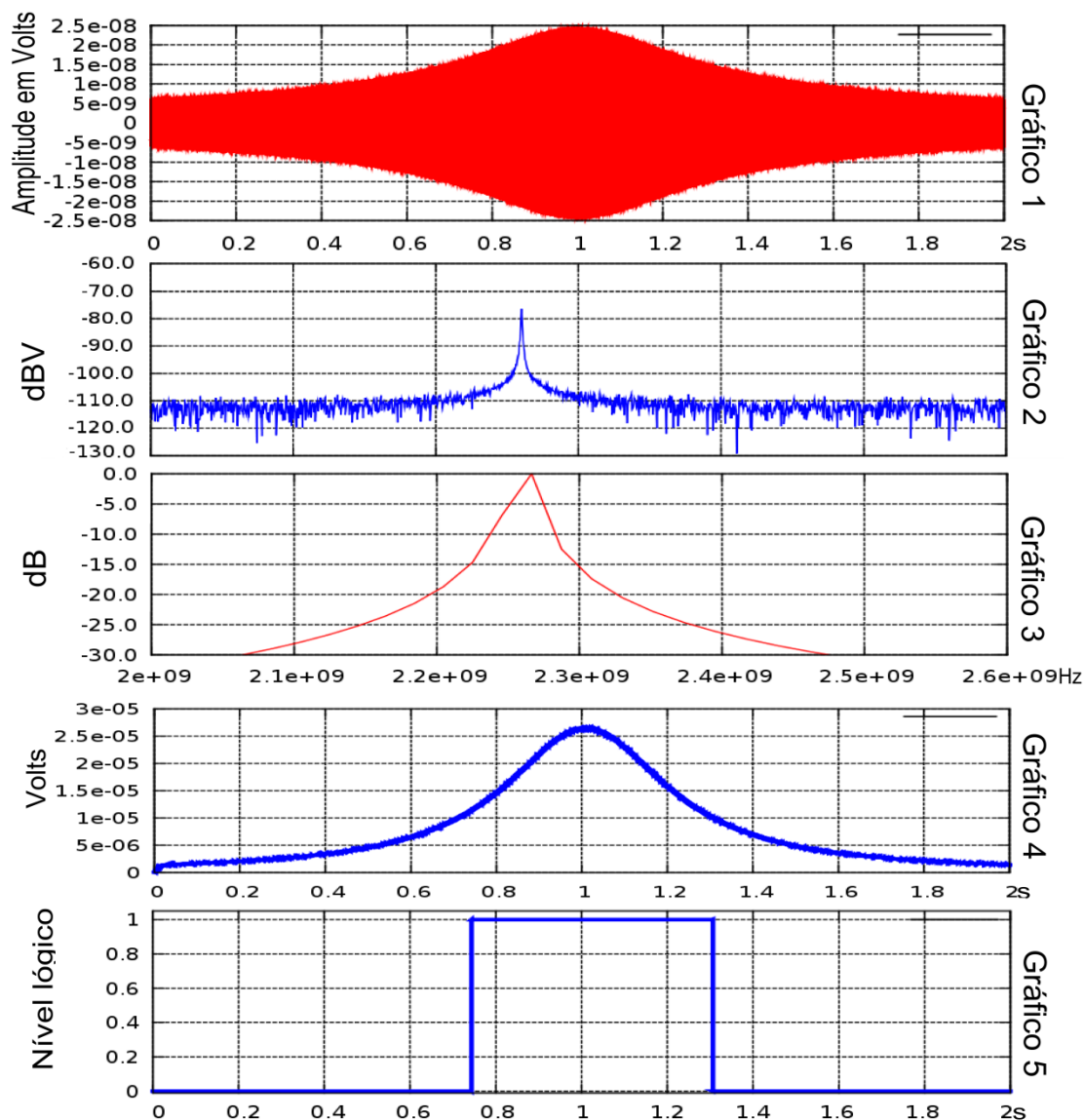


Figura 28 – Resultado de simulação simultânea de cada bloco que constitui o módulo detector.

#### 4.4.3 Módulo controlador

A implementação deste módulo se baseou na modelagem proposta na seção 3.2.3 que levou em consideração as características dos componentes: registrador, contador, conversores, assim como, da rede de Petri que os controla. A interligação destes blocos, que segue a lógica da modelagem da Figura 12, se encontra exposta na Figura 29. As descrições feitas em SystemC destes componentes seguem descritas nos apêndices A.7, 8, 9, 10 e 11.

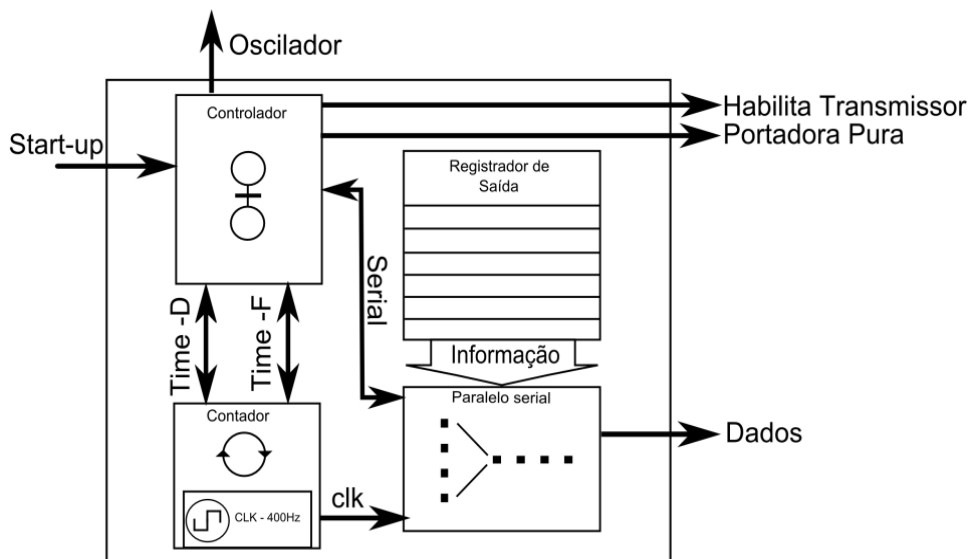


Figura 29 – Interligações do módulo controlador.

A simulação destes modelos que constituem o controlador da plataforma de identificação está demonstrada na Figura 30. O controlador recebe o sinal de identificação (**Start-up**). Este sinal é reconhecido imediatamente pela rede de Petri que liga os osciladores (**Oscilador**). Após um atraso devido à estabilização dos osciladores a transmissão da portadora pura (**Time-D, Habilita Transmissor e Portadora Pura**) é habilitada. Ao término do tempo da portadora pura inicia-se a modulação (**Serial e Dados**). O fim da transmissão é precedido do início do período de recorrência (**Time-F**).

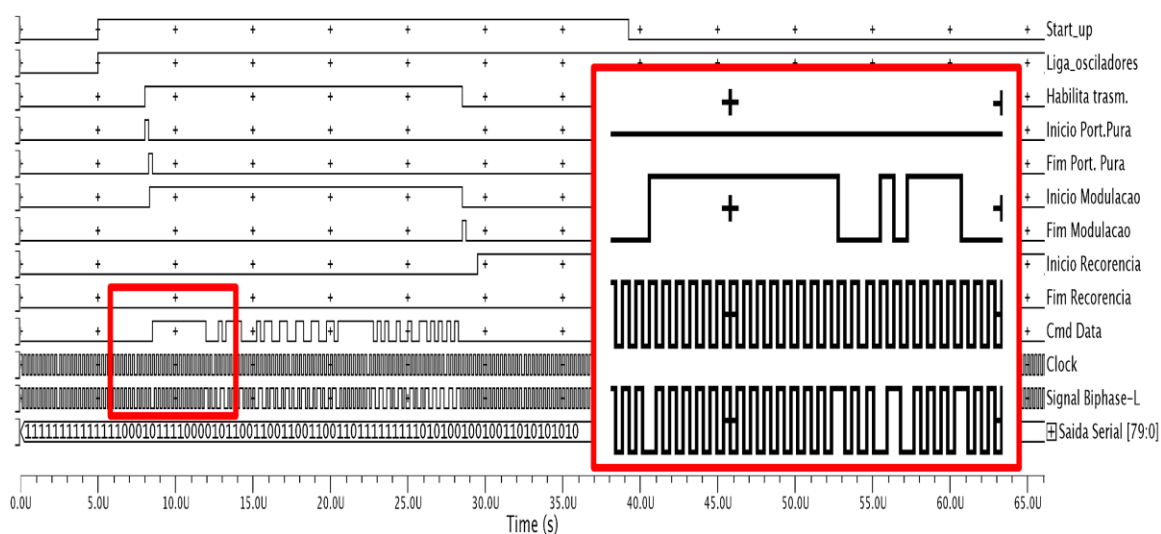


Figura 30 – Comportamento do módulo controlador considerando uma base de um tempo reduzida.

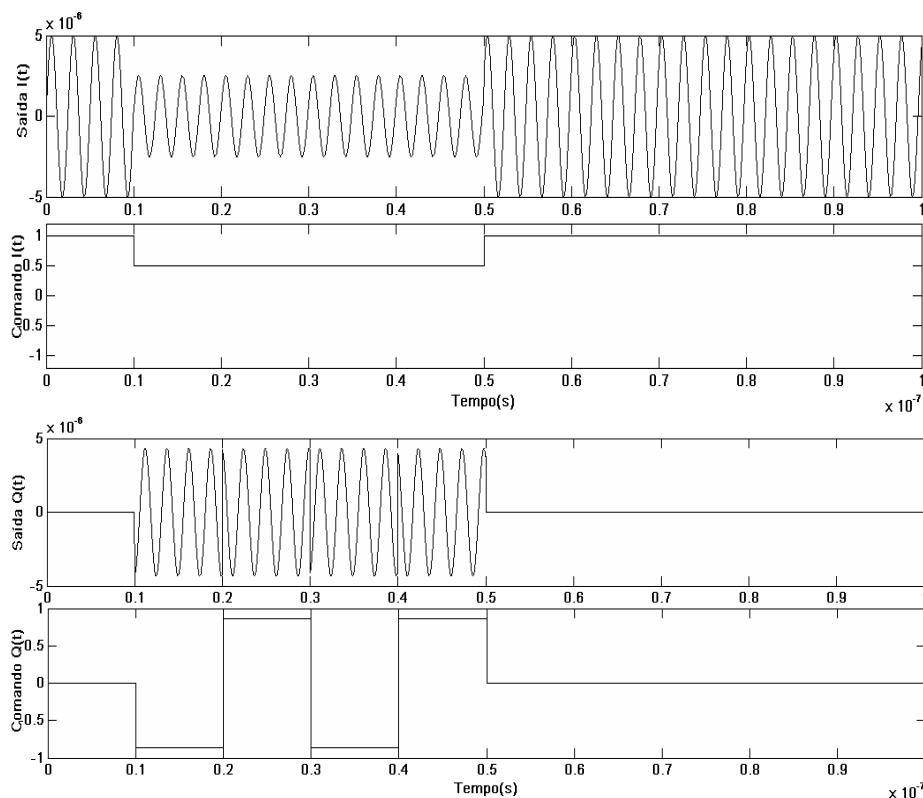
#### 4.4.4 Módulo transmissor

A parte responsável pela transmissão da plataforma está compreendida no módulo de transmissão, que integra os componentes: seletor, *mixer*, oscilador, defasador, somador e por último o amplificador de potência (PA). Os parâmetros utilizados durante as simulações dos componentes anteriormente citados seguem na Tabela 7.

**Tabela 7 - Parâmetros da descrição do transmissor empregados na simulação completa.**

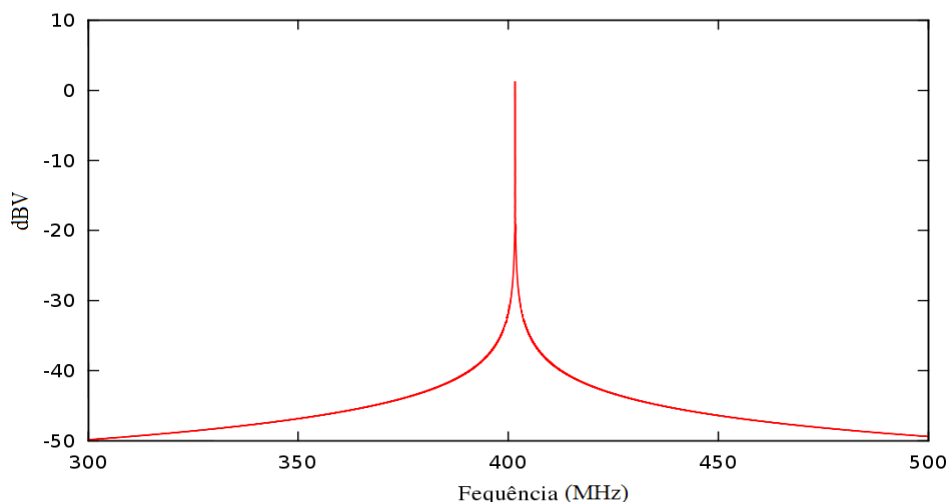
Entidade	Componentes	Parâmetros				
		DC	Amplitude	Freq.	Rout	Ruído de Fase
Transmissor	Oscilador	0	10 dBm	401,62MHz	50	1°
		Freq.	Ângulo			
	Defasador	401,62MHz	45°			
		Ganho de Conversão	IP3	Rin	Rout	
	Mixer	0 dB	20 dBm	50	50	
		Ganho	IP3	Rin	Rout	
	Somador	12 dB	20 dBm	50	50	
		Ganho	IP3	Rin	Rout	
	PA	30dB	10 dBm	50	50	

Os misturadores, que estão descritos no apêndice A.15, são responsáveis pela modulação do sinal. A verificação desta funcionalidade foi realizada a partir de um banco de teste que comprova a modulação da informação pela variação das saídas do seletor. As respostas, conforme ilustrada na Figura 31, demonstram que a mudança de fase está relacionada com as saídas do seletor e corresponde o exposto na Tabela 4.



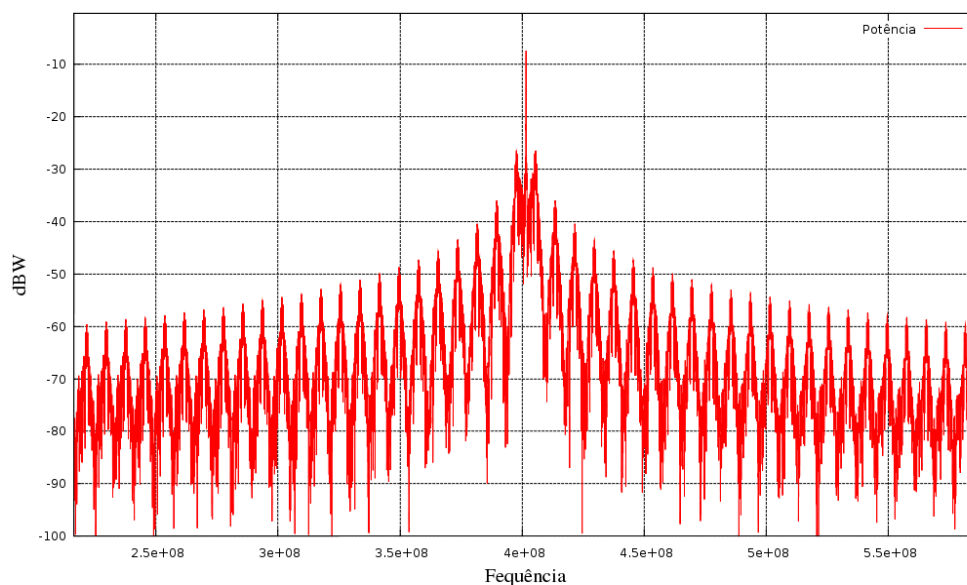
**Figura 31 – Simulação de teste para validar a modulação BPSK  $\pm 60^\circ$ .**

O oscilador, descrito no foi simulado e em seguida feito sua representação na frequência a fim de observar sua resposta. A Figura 32 mostrado o resultado obtido após simulações do modelo do oscilador descrito no apêndice A.13.



**Figura 32 - Espectro do Oscilador centrado na frequência de 401,62MHz.**

O amplificador de potência é responsável por adicionar potência ao sinal transmitido pela plataforma. A descrição deste modelo segue no apêndice A.17. A resposta, ilustrada na Figura 33, mostra a potência do sinal transmitido pela plataforma de identificação que é de -6dBW.



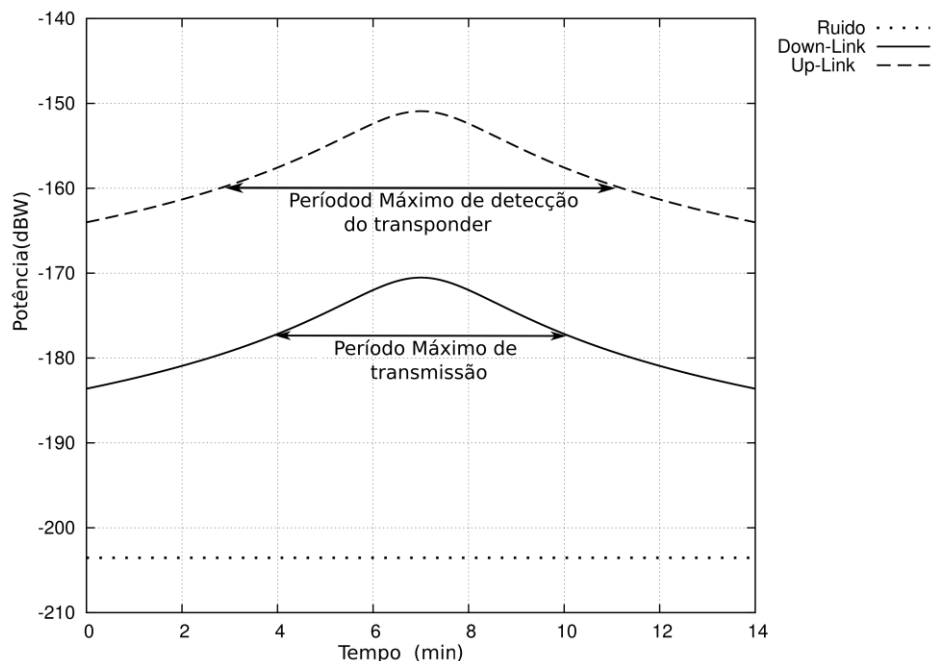
**Figura 33 - Espectro de potência do sinal transmitido pela plataforma.**

## 5 RESULTADOS

Com o objetivo de obter um conjunto de especificações que oriente na implementação da micro-plataforma de identificação, utilizou-se a biblioteca de classes SystemC, acrescida da extensão para domínio dos sinais mistos, SystemC-AMS, além disso, um conjunto de ferramentas *open source* como Octave e Gnuplot foram empregadas para análise e tratamento matemático dos resultados. Para a modelagem e verificação dos componentes referentes ao controlador utilizou-se a biblioteca SystemC e para os componentes referentes a orbita, ao Detector e ao Transmissor utilizou-se SystemC-AMS. Nesta seção, serão analisados os modelos que permitiram, por meio de simulações em conjunto, obter bancos de testes a fim de caracterizar os blocos e adquirir, desta forma, os requisitos necessários para especificação do sistema.

### 5.1 PERÍODO DE DETECÇÃO

O modelo do canal permite observar a variação da potência do sinal ao longo de todo o período de visibilidade que transcorre da elevação  $0^\circ$  a  $180^\circ$ . O modelo proposto na seção 3.2.1 foi simulado a fim de obter a variação da potência do sinal recepcionado pela plataforma, conforme demonstrado na Figura 34. Com base no requisito de sensibilidade do transponder que define  $-160\text{dBW}$  como a potência mínima captada pelo transponder, cuja relação sinal ruído de  $43,59\text{dBHz}$  (Rae, 2005), pode-se determinar o período propício para a realização do envio dos dados. Este período é definido como período máximo de detecção do transponder que se inicia na elevação de  $13^\circ$ . Este período ajuda a definir o instante que a plataforma deverá ficar ativa, pois, abaixo da elevação de  $13^\circ$ , o satélite não detectaria o sinal transmitido pela plataforma.



**Figura 34 - Potências do sinal durante o período de passagem do satélite**

A determinação do período de transmissão permite delimitar o período de detecção. Sabendo que o ciclo de transmissão da mensagem de uma plataforma é de 200s ( $\cong 3,3$ min) optou-se por realizar duas transmissões, pois pelo menos uma única vez a informação seria transmitida no instante de menor perda no canal, isto é, a transmissão será realizada no instante em que o satélite estiver mais próximo da plataforma.

## 5.2 SENSIBILIDADE DO DETECTOR

A implementação deste modelo, que está proposto na seção 3.2.2, exemplifica a extração de requisitos por meio da utilização do SystemC-AMS. Assim, com o objetivo de indicar o momento propício para o envio dos dados, analisou-se o detector a fim de analisar sua responsividade. Esta análise considerou a influência dos ganhos e o nível de comparação exigidos. Esta análise, que oferece uma visão global do detector, permitiu obter a responsividade do detector, a partir da tensão mínima exigida pelo comparador lógico e pelo sinal captado na antena. Este teste gerou, por meio de simulações, uma margem de valores de ganhos para dois níveis de elevação  $46^\circ$  e  $90^\circ$ , conforme apresentado na Figura 35.



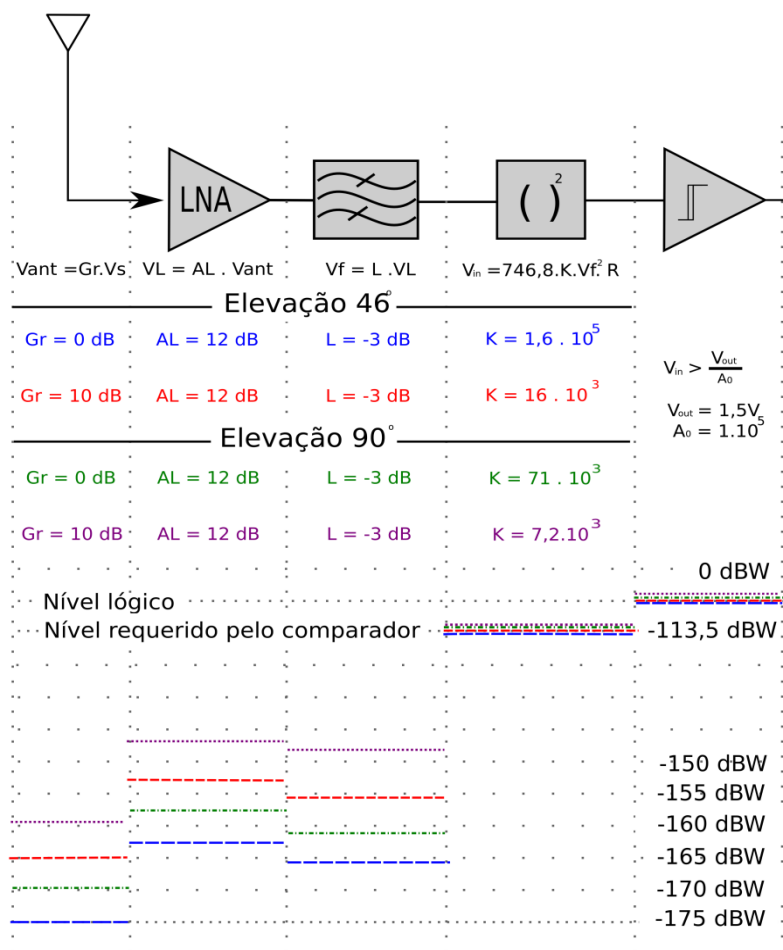
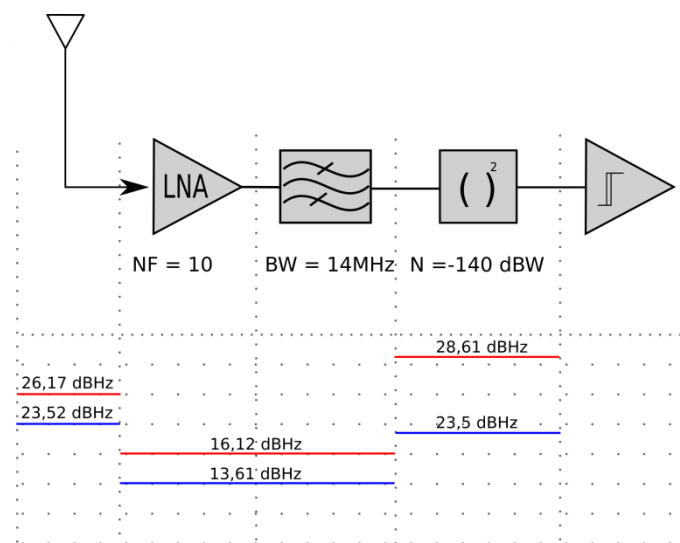


Figura 35 – Avaliação de sensibilidade no detector.

Com o resultado desta análise, pode-se ter uma idéia da margem de valores de ganhos por faixa de elevação para que ocorra a detecção. A importância deste resultado contribui para a definição da arquitetura do detector de potência. Esta definição será responsável por definir se a plataforma realizará duas transmissões ou somente uma transmissão. Como a sensibilidade na elevação de 46°, que corresponde a uma potência de -175dBW, exige uma sensibilidade 225,35% maior que na elevação 90°, pressupõe assim que é mais viável a realização de uma transmissão.

Analisando agora a influencia da degradação do sinal ao longo do detector, obteve-se o resultado exposto na Figura 36. A análise foi realizada considerando a responsividade de  $1,6 \cdot 10^5$  nas elevações de 46° e 90° respectivamente representadas por azul e vermelho. A análise mostra a regeneração da relação sinal ruído para as duas faixas de elevação. Este fato ocorre, pois o sinal ao entrar no detector muda de domínio, isto é, passa da variação periódica de tensão para uma variação gradual de potência. A regeneração na primeira elevação ocorre quase que completa, já na elevação máxima ocorre um acréscimo de

aproximadamente 2,5dBHz em relação a relação sinal ruído inicial. Este resultado permite dimensionar a imunidade do comparador ao ruído, isto é, permite definir a característica da histerese que anulará a variação lógica do *start-up*.

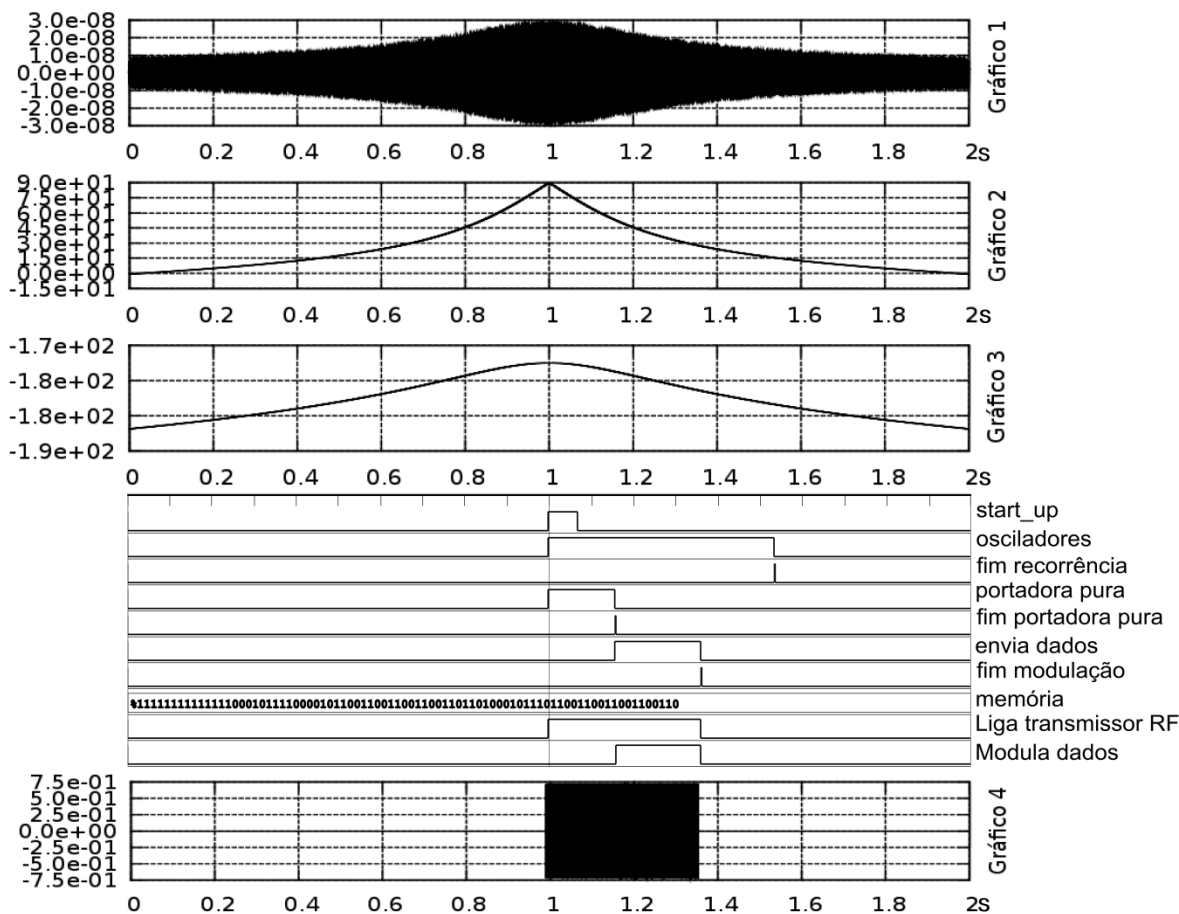


**Figura 36 - Relação sinal ruído em toda a cadeia do detector.**

### 5.3 SISTEMA CONTROLADOR

A modelagem do elemento controlador da plataforma foi descrito utilizando SystemC. Este bloco é constituído por três subsistemas que são comandados pela implementação da RdP em SystemC. Fazendo a correspondência da execução com a modelagem em RdP temporizada ilustrada na seção 3.2.3, pode-se, através do banco de teste, avaliar este modelo. A execução do teste foi realizada com a plataforma completa que obteve os resultados ilustrados na Figura 37.

A Figura 37 mostra nos gráficos 1, 2 e 3 respectivamente, o comportamento no tempo do sinal recepcionado, a elevação do satélite no mesmo momento e a potência na antena da plataforma de identificação. Dimensionada para identificar o satélite na melhor proximidade, o detector responde ao estímulo de entrada com um pulso de aproximadamente 90ms, representado pelo *start-up*. A informação de início é reconhecida pelo sistema que liga os osciladores e em seguida realizando a transmissão da portadora pura e da informação modulada, como observada no gráfico 4.



**Figura 37 - Resultado completo da plataforma empregando um período de execução reduzido em relação ao real que é de 14 minutos; 1 – potência do sinal, 2 – Ângulo de elevação, 3 – Sinal captado, 4 – Sinal transmitido.**

Pode-se observar na Figura 37, o sinal promove a execução da rede que desenvolve sequencialmente as tarefas de ligar os osciladores, habilitar a transmissão e comandar a modulação dos dados existentes na memória interna do dispositivo que evolui desta forma, até o estágio de verificação do estado lógico do *start-up*. Esta verificação, que ocorre logo após o período de recorrência, verifica as condições de retransmissão dos dados, caso não ocorra, o sistema entra em hibernação.

#### 5.4 SISTEMA DE TRANSMISSÃO

Tem-se como requisito de transmissão do sistema de coleta de dados uma potência mínima de -6dBW e uma diferença das raias parasitas inferior a -30dBW (Jean Paul Dubut, 1983). O requisito de distorção está demonstrado na Figura 38.

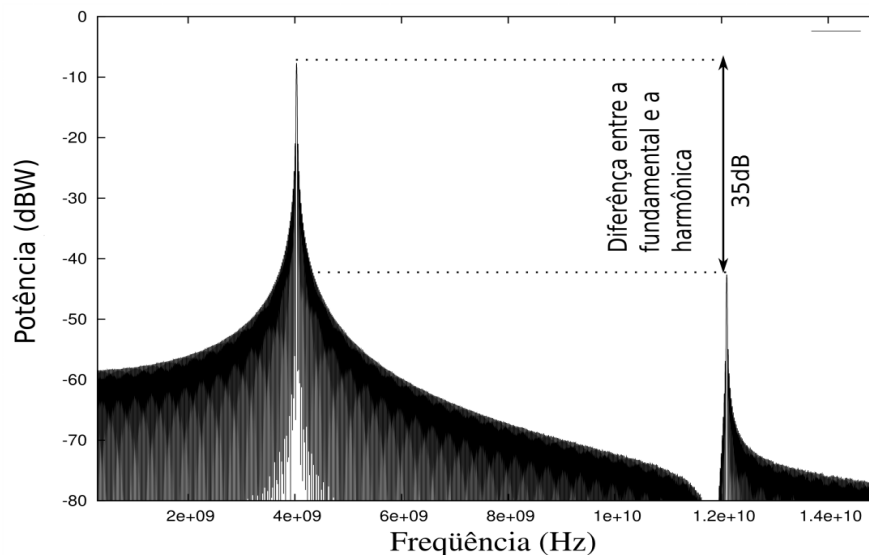


Figura 38 - Espectro de saída da plataforma, mostrando a harmônica de terceira ordem.

Outra avaliação sobre o sistema que pode ser obtida a partir das simulações é a análise de simetria Figura 39. Esta análise considerou a contribuição da variação do ganho de conversão dos *mixers* e da distorção de terceira ordem. A análise consistiu em variar estes parâmetros até o limite de distorção que é de 1% (Jean Paul Dubut, 1983).

Assim, os seguintes resultados foram obtidos para o *mixer* em fase (I(t)):

- O menor valor para o IP3 que respeita a simetria é de 15dBm;
- A variação de  $\pm 1.9$ dB no ganho.

Para o *mixer* em quadratura Q(t) obteve-se os seguintes resultados:

- O menor valor para o IP3 que respeita a simetria é de 15dBm;
- Variação de  $\pm 1.2$ dB no ganho de conversão.

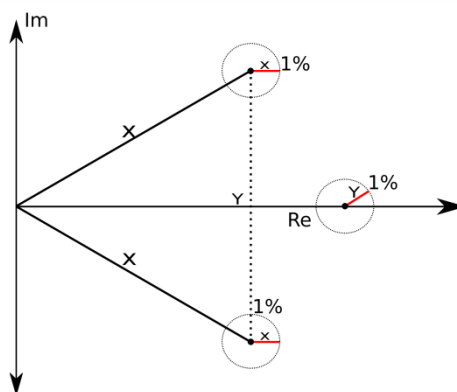


Figura 39 – Representação da área de simetria da constelação.

O resultado do desempenho deste sistema avaliou, também, a taxa de erro de bit no canal de subida “*up-link*”. Esta análise comparou a curva teórica com os erros obtidos, por meio de simulações, considerando a distorção do PA em 20dBm e 10dBm. Observando os resultados expressos na Figura 40, percebe-se a dependência da taxa de erro de bit com a componente IP3.

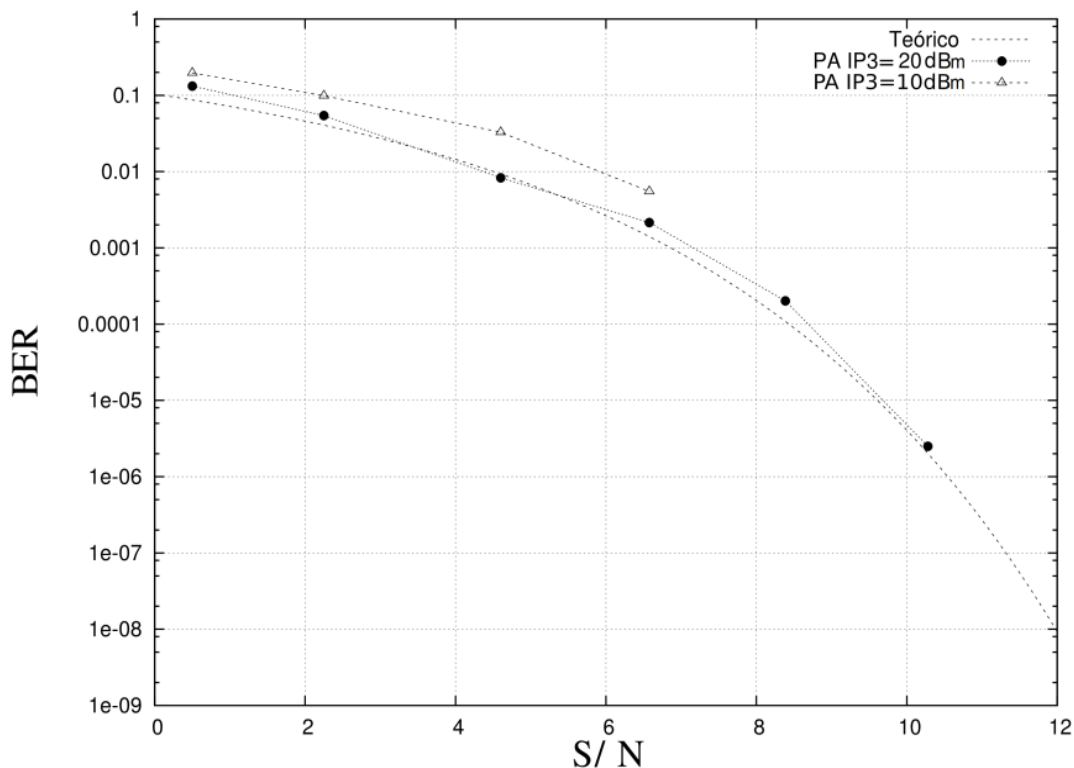


Figura 40 - Avaliação do erro de Bit durante a transmissão dos dados ao satélite.

## 5.5 CONCLUSÃO

Os resultados destas análises mostram, através da simulação de vários domínios em SystemC-AMS, a verificação dos modelos que constituem a plataforma de identificação. Nesta verificação foi possível a determinação da margem de valores que tornam a detecção do satélite viável, assim como avaliar a implementação de um sistema controlador por rede de Petri para comandar a plataforma. O resultado que fecha o ciclo de funcionamento desta implementação, reside na verificação dos dados enviados ao satélite que avalia o canal e a modulação dos dados. Assim, pode-se avaliar a utilização de uma plataforma de simulação capaz de simular, conjuntamente, vários domínios para determinar previamente o comportamento do sistema.

## 6 CONCLUSÕES E PERSPECTIVAS

Uma ferramenta de análise, que englobe diversos domínios, é uma importante estratégia de projeto. A maioria dos efeitos que influenciam o sistema podem ser modelados antes de sua implementação, permitindo avaliar e analisar o desempenho do circuito projetado de forma mais precisa. Este tipo de modelagem, que utiliza SystemC-AMS, apesar de ser uma iniciativa de código aberto, é capaz de atender essas expectativas, pois partindo de um requisito funcional do sistema como: identificação do satélite, controle e transmissão do dados, foi possível verificar a proposta de implementação da plataforma que atende-se ao requisitos do sistema de coleta de dados do INPE. As análises permitiram apontar um conjunto de especificações que, obtidas a partir de simulações, viabilizam a plataforma e orienta na implementação em chip do sistema. A contribuição, deste trabalho, reside na obtenção das especificações que orientaram nas escolhas das arquiteturas de circuitos, conforme as exigências das especificações e na demonstração da viabilidade do sistema. Além disso, as simulações em conjunto de diferentes domínios em uma única plataforma de simulação contribui para o enriquecimento dos estudos de casos do SystemC-AMS.

O trabalho expõe várias perspectivas de trabalhos futuros, entre elas, a implementação física da plataforma, a comparação dos resultados simulados com os mensurados, além do aperfeiçoamento e inclusão de novos modelos ao projeto.

## REFERÊNCIA BIBLIOGRÁFICA

Alain Vachoux, Christoph Grimm and Karsten Einwich. 2003. *Analog and Mixed Signal Modelling with SystemC-AMS*. Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on. pp. III-914-III-917, 2003, Vol. 3, 25-28 May 2003.

—. 2004. Towards Analog and Mixed-Signal SOC Design with SystemC-AMS. *Proceedings of the Second IEEE International Workshop on Electronic Design, Test and Applications (DELTA'04)*. 28-30 Jan, 2004.

Alberto Rabbachin, Ian Oppermann. 2004. Synchronization Analysis for UWB Systems with a Low-Complexity Energy Collection Receiver. *Ultra Wideband Systems Joint with Conference on Ultrawideband Systems and Technologies and Joint UWBST & IWUWBS*. 2004.

Bhasker, J. 2002. *A SystemC Prime*. s.l. : star Galaxy Publishing, 2002.

Cecilia Carbonelli, Umberto Mengali. 2005. Low Complexity Synchronization for UWB Noncoherent Receivers. *Vehicular Technology Conference, 2005*. VTC 2005-Spring, 2005, Vol. 2005 IEEE, 61st.

Christos G. Cassandras, Stéphane Lafortune. 2008. *Introduction to Discrete Event Systems*. New York : Springer Science, 2008. 978-0-387-33332-8.

Donovan, David C. Black and Jack. 2004. *SYSTEMC: FROM THE GROUND UP*. s.l. : Kluwer Academic Publishers, 2004.

Eduardo Antonio Parada Tude, Carlos Alberto Iennaço Miranda, Leopoldo Edgardo Messenger Parada, Max Herique Machado Costa, Sérgio de Paula Pereira, Válder Matos de Medeiros. 1986. *Análise do Sistema de Coleta de Dados da MECB/SS*. 1986.

Einwich, K. 2003. SystemC-AMS: Steps towards an implementation. *Proc. FDL 2003*. 2003.

Frevert, Ronny, et al. 2005. *Modeling and Simulation for RF System Design*. s.l. : Springer, 2005. 10 0-387-27584-3.

H. Chang, L. Cooke, M. Hunt, G. Martin, A. McNelly, L. 1999. *Surviving the SOC Revolution: A Guide to Platform-Based*. San Jose, California : Kluwer Academic Publishers, 1999.

Jean Paul Dubut, Carlos Dantas Aquino. 1983. *Plataforma de Coleta de Dados para o Sistema Argos - Manual Técnico*. 1983.

Kuga, Valcir Orlando e Hélio Koiti. 2007. Os satélites SCD1 e SCD2 da Missão Espacial Completa Brasileira - MECB. [A. do livro] Antonio Fernando Bertachini de Almeida Prado Othon Cabo Winter. *A Conquista do Espaço do Sputnik à Missão Centenário*. São Paulo : Livraria da Física , 2007.

Leung, Bosco H. 2002. *VLSI for wireless Communication*. s.l. : Prentice Hall, 2002.

Louis J. Ippolito, Jr. 2008. *Communications Systems Engineering: atmospheric effects, satellite link design, and system*. s.l. : JohnWiley & Sons Ltd, 2008. 978-0-470-72527-6.

- Michel Vasilevski, Francois Pecheux, Hassan Aboushady, Laurent de Lamarre. 2007. *Modeling Heterogeneous Systems Using SystemC-AMS Case Study: A Wireless Sensor Network Node*. Behavioral Modeling and Simulation Workshop . BMAS 2007. IEEE International, 2007.
- Murata, T. 1989. *Petri Nets: Propriets, Analysis and Aplications*. Proceeding of the IEEE. 1989.
- Open SystemC Initiative. [Online] <http://www.systemc.org>..
- P. Kolinko, L.E. Larson. 2007. *Passive RF Receiver Designa for Wireless Sensor Networks*. Microwave Symposium, 2007. IEEE/MTT-S International. 2007.
- Phillip A. Allen, Douglas R. Holberg. 2002. *CMOS Analog Circuit Design*. New York : Oxford university press, 2002.
- Rae, João Carlos Pécala. 2005. *Detector de Sinais para os Satélites do Sistema brasileiro de Coleta de Dados usando Análise Espectral Digital*. São José dos Campos : Tese de mestrado - Instituto Tecnológico de Aeronáutica, 2005.
- Ramchandani, Chander. 1974. *Analysis Of Asynchronous Concurrent Systems By Timed Petri Nets*. February Massachusetts : Massachusetts Institute of Technology, 1974.
- Razavi, Behzad. 1998. *RF Microeletronics* . s.l. : Prentice Hall PTR, 1998. 0-13-887571-5.
- Rogers, John e Plett, Calvin. 2003. *Radio frequency integrated circuit design*. s.l. : ARTECH HOUSE, INC, 2003. 1-58053-502-x.
- Sanderson, M. 1988. *Eletronic Devices: a top-down systems approach*. New Jersey : Prentice-Hall, 1988. 0132508796.
- Serioja o. Tatu, renato g. Bosisio,. 2005. *Quadrature Phase Shift Keying (QPSK)*. [A. do livro] Kai Chang. *ENCYCLOPEDIA OF RF AND MICROWAVE ENGINEERING*. New Jersey : A John Wiley & Sons, Inc., Publication, 2005.
- Sousa, Fernando Rangel de. 2004. *Application du corrélateur « Five-Port » aux PLLs, à la récupération de porteuse et à un MODEM de télécommunications dans la bande 1,8 - 5,5 GHz*. Paris : s.n., 2004.
- Thorsten Grotker, Stan Liao, Grant Martin, and Suart Swan. 2002. *System Design with SystemC*. New York, Boston, Dordrecht, London, Moscow : Kluwer Academic Publishers, 2002.
- Uzam, Murat, et al. 2008. *Asynchronous implementation of discrete event controllers based on safe automation Petri nets*. 2008.
- Valette, Robert. 2002. Les Réseaux de Petri. *LAAS-CNRS*. [Online] Setembro de 2002. <http://www.laas.fr/~robert>.



## APÊNDICE A

### APÊNDICE A.1 - Descrição da Distância Orbital em SystemC-AMS

```

SCA_SDF_MODULE (distancia)
{
// distancia(km) & angulo de elevação(rad) & velocidade do atelite(km/s)
sca_sdf_out<double> out,ang,V;

    double Re, Rs, Ws, Wi, Dist_mtr_sat, w, Ts, V_ang, V_sat, V_f;

void init(double raio_terra, double altura_sat, double angulo_sat, double tempo_pass, double Ang_atual,
double V_sat){

    this -> V_sat = V_sat;           // velocidade do satélite em km/h
    this -> Re = raio_terra;         // raio da terra em km
    this -> Wi = Ang_atual*M_PI/180; // angulo inicial em rad
    this -> Rs = altura_sat;         // altura do satélite km
    this -> Ts = tempo_pass;         // Tempo de passagem no percurso de visibilidade 0 a 90 em s
    this -> Ws = angulo_sat*M_PI/180; // angulo final em rad
    }
void sig_proc(){

// variação da angulação do satélite em rad/s
    this -> w = Wi+(Ws/Ts*sc_time_stamp().to_seconds());
// distancia entre o satélite e o micro transmissor
    this -> Dist_mtr_sat = sqrt(pow(Re+Rs,2)+pow(Re,2)-2*(Re+Rs)*Re*cos(w));
    double V_ang = acos((pow(Re,2)-pow(Re+Rs,2)-pow(Dist_mtr_sat,2))/(-2*(Re+Rs)*Dist_mtr_sat));
if (w <= 0){ // se o valor do angulo for negativo
// velocidade de transmissão em relação ao transmissor
    this -> V_f = cos((M_PI/2)-V_ang)*V_sat;
}
else
{
    this -> V_f = cos((M_PI/2)+V_ang)*V_sat;
}

// Transcrição das saídas
out.write(Dist_mtr_sat);
ang.write(w);
V.write(V_f);
}
SC_CTOR (distancia) {}
};

```

## APÊNDICE A.2 - Descrição do canal *Downlink* em SystemC-AMS

```

SCA_SDF_MODULE (downlink)
{
    sca_sdf_in<double> in,Vf; //distancia da fonte transmissora, velocidade relativa
    sca_sdf_out<double> out,pot,lsf,eirp,nois; //sinal que chega no receptor
    double f ,fr ,PT ,PFD ,GT ,LSF ,EIRP ,LO ,pr ,SIGMA ,I_ant ,PR ,R_ant ,c ,N0 ,Rin ,Gr ,k , T , R_eq;

    // banda do ruído, frequência do sinal, potencia de transmissão, ganho da antena,
void init(double fn, double freq, double Pot_tras, double Gain_tras, double othr_loss, double r, double rin,
double g_ant){

    this -> c = 1079252848.8; // velocidade da luz em Km/h
    this -> k = 1.38e-23; // constante de Boltzmann J/K
    this -> T = 273 + 27; // temperatura em kelvins
    this -> Gr = g_ant; // ganho da antena receptora
    this -> R_ant = r; // resistência da antena
    this -> Rin = rin; // resistência de entrada do bloco seguinte
    this -> LO = othr_loss; // outras perdas em dB
    this -> f = freq;
    this -> PT = Pot_tras; // potencia de transmissão
    this -> GT = Gain_tras; // ganho da antena
    srand(time(NULL));
    this -> R_eq = ((Rin*R_ant)/(Rin+R_ant));
// ruído estimado na entrada do sistema dado em v/sqrt (Hz) médios
    this->SIGMA = sqrt(pow((sqrt(4*k*T/Rin)*R_eq) + (sqrt(4*k*T*Rin)*(R_ant/(Rin+R_ant))),2) ...
+4*k*T*R_ant*pow(R_ant/(Rin+R_ant),2))*sqrt(fn);

}

void sig_proc(){
///// processo de potencia do sinal recepcionado //////////
    double dist = in.read();
    this -> LSF = 20*log10(f) + 20*log10(dist) + 92.44; // perda do espaço livre
    this -> EIRP = PT + GT; // effective isotropic radiated power dBW
    this -> PR = EIRP + Gr - LSF - LO; // inclusão das perdas
    double pr = pow(10,PR/10); // potencia em watts
    double noise = (rand()*(2*SIGMA/RAND_MAX))-SIGMA; // ruído branco dado V/sqrt(Hz)
    double V = sqrt(Rin*2*pr); // amplitude do sinal de entrada

///// inicio do processo do efeito Doppler//////////
    this -> fr = f*c/(c+Vf.read());

///// forma do sinal transmitido //////////
// Saída do modulo
    out.write(V*sin(2*M_PI*fr*sc_time_stamp().to_seconds()) + noise );
// sinais utilizados para verificação
    pot.write(PR);
    eirp.write(EIRP);
    lsf.write(LSF);
    nois.write(noise);

}
SC_CTOR (downlink) {}
};

```

## APÊNDICE A.3 - Descrição do amplificador de baixo ruído em SystemC-AMS

```

SCA_SDF_MODULE (Ina)
{
    sca_sdf_in<double> in;
    sca_sdf_out<double> out;
    double gain_power, a1, a3, AIP3, sigma;
    double rin, rload;

    void init(double gain_power_db, double iip3, double nf, double rin, double rload, double T){

        double f = pow(10,nf/10), N0 = 4*(f-1)*pow(10,-228.6/10)*(T+273)*50; //potencia de ruído
        this -> sigma = sqrt(N0); // valor médio de ruído v/sqrt(Hz)
        srand(time(NULL)); // inicio da contagem da função rand
        this -> rin = rin; // resistência de entrada
        this -> rload = rload; // resistência da carga
        this -> AIP3 = sqrt(pow(10,iip3/10)*2*rin*0.001); // ip3 dBm
        this -> gain_power=pow(10,gain_power_db/10); // ganho do amplificador
    }

    void sig_proc(){
        double noise = (rand()*sigma/RAND_MAX); // geração de ruído a partir da potencia de ruído
        this -> a1=sqrt(gain_power*rload/rin); // componente de distorção de primeira ordem
        this -> a3=a1/(3*pow(AIP3,2)/4); // componente de distorção de terceira ordem
        double input = in.read() + noise; // sinal de saída com ruído

        out.write(a1*input-a3*pow(input,3));
    }

    SC_CTOR (Ina) {}
};

```

## APÊNDICE A.4 - Descrição do filtro do *Wake-up* em SystemC-AMS

```

SCA_SDF_MODULE (band_pass)
{
    sca_sdf_in<double> inp;
    sca_sdf_out<double> outp;
    sca_ltf_nd ltf1;
    sca_vector<double> num, den;
    // descrição inicial da equação no domínio da frequência
    void init(double fc, double Q){
        num(0) = 0;
        num(1) = (2*M_PI*fc)/Q;
        den(0) = pow(2*M_PI*fc,2);
        den(1) = num(1);
        den(2) = 1;
    }
    // análise ac do filtro
    void ac_sig_proc(){
        sca_ac(outp) = sca_ac_ltf_nd(num,den,sca_ac(inp));
    }
    void sig_proc(){
        outp = ltf1(num,den,inp.read());
    }
}

SC_CTOR (band_pass) {}
};

```

## APÊNDICE A.5 - Descrição do detector de potência em SystemC-AMS

```

SCA_SDF_MODULE (detc)
{
    sca_sdf_in<double> in;
    sca_sdf_out<double> out;
    sca_ltf_nd ltf1;
    double Rout, S, B;
    sca_vector<double> num, den;

    // descrição inicial da equação no domínio da frequência
    void init(double fc, double Q, double Rout, double sen){
        double q = 1.6e-19;           // valor da carga em C
        double N = 1;                 // fator de rampa
        double K = 1.38e-23;          // constante de Boltzmann J/K
        double T = 273+27;            // Temperatura em K
        this -> B = q/(N*K*T);
        this -> Rout = Rout;
        this -> S = sen;               // Sensibilidade
    }

    // definição do filtro
    num(0) = 1;
    den(2) = 1;
    den(1) = (2*M_PI*fc)/Q;
    den(0) = pow(2*M_PI*fc,2);
}

void sig_proc(){
    double Id = S*pow(in.read(),2); // modelo do detector em series de Taylor
    out.write(ltf1(num,den,Id));
}

SC_CTOR (detc) {}
};

```

## APÊNDICE A.6 - Descrição do comparador com histerese em SysemC-AMS

```
SCA_SDF_MODULE(comp){
    sca_sdf_in<double> in;
    sca_scsdf_out<bool > out;

    void init(double threshold_sup, double threshold_inf){
        thrs = threshold_sup;
        thri = threshold_inf;
    }
    void sig_proc(){
        if (in.read() >= thrs) out.write(1);
        if (in.read() <= thri) out.write(0);
    }

    SCA_CTOR(comp){}
    private:
    double thrs, thri;
};
```

## APÊNDICE A.7 - Descrição da Rede de Petri em SystemC

```

SC_MODULE(Petri){
    sc_in<bool> stup, ack_c, ack_p2s, ack_e;
    sc_out<bool> source_osc, source_sel, req_c, req_p2s, req_e;
    sc_event p1, p2, p5, p6, p7, ps;

    // notificação do estabelecimento das condições para ocorrer a transição
    void lugar(){
        if(stup == 1)    {p1.notify();}
        if(ack_c == 1)  {p5.notify();}
        if(ack_p2s == 1) {p6.notify();}
        if(ack_e == 1)  {ps.notify();}
    }
    void thread1(){
        wait(p1);
        source_osc = 1;
        wait(2,SC_US);
        for(;;){
            wait(1,SC_US);
            req_c = 1;
            source_sel = 1;
            wait(p5);
            req_c = 0;
            wait(0.1,SC_US);
            req_p2s = 1;
            wait(p6);
            req_p2s = 0;
            source_sel = 0;
            wait(1,SC_US);
            req_e = 1;
            wait(ps);
            req_e = 0;
            if(!stup) {break;}
        }
        source_osc = 0;
    }
}

SC_CTOR(Petri){
    SC_METHOD(lugar);sensitive << stup << ack_c << ack_p2s << ack_e ;
    dont_initialize();
    SC_THREAD(thread1);
}
};

```

## APÊNDICE A.8 - Descrição do contador em SystemC

```

SC_MODULE(contador){
    sc_in<bool> clk;
    sc_in<bool> req_cont_1,req_cont_2;
    sc_out<bool> ack_cont_1,ack_cont_2;
    sc_event e1,e2;
    double cont_1,cont_2;
    //////////////////////////////////////
    void incremento_160(){
        if(req_cont_1 == 1){
            cont_1++;
            if (cont_1 == 64){
                ack_cont_1 = 1;
            }
        }else {ack_cont_1 = 0; cont_1 = 0;}
    }
    //////////////////////////////////////
    void incremento_200(){
        if(req_cont_2){
            cont_2++;
            if (cont_2 == 79856){
                ack_cont_2 = 1;
            }
        }else {ack_cont_2 = 0; cont_2 = 0;}
    }
    //////////////////////////////////////
    SC_CTOR(contador){
        SC_METHOD(incremento_160);
        sensitive << clk.pos();
        SC_METHOD(incremento_200);
        sensitive << clk.pos();
    }
};

```



## APÊNDICE A.9 - Descrição do Registrador de Saída em SystemC

```
SC_MODULE(registrador){
    sc_out<sc_biguint<80>> out_p2s;

    void prc_registrador(){
        sc_biguint<80> regist;
        sc_uint<15> sic_bit = "0b111111111111111";
        sc_uint<8> sic_pal = "0b00010111";
        bool inic_msg = 1;
        sc_uint<4> n_blocos = "0b0001";
        sc_uint<20> id_pcd = "0b01100110011001100110";
        sc_uint<32> in_sensor = "0b11010001011101100110011001100110";
        regist.range(79,66) = sic_bit;
        regist.range(65,58) = sic_pal;
        regist[57] = inic_msg;
        regist.range(56,52) = n_blocos;
        regist.range(51,32) = id_pcd;
        regist.range(31,0) = in_sensor;
        out_p2s = regist;
    }

    SC_CTOR(registrador){
        SC_METHOD(prc_registrador);
    }
};
```

## APÊNDICE A.10 - Descrição do conversor paralelo para serial em SystemC

```

SC_MODULE(seripar){
    sc_in<sc_biguint<80>> in_mem;
    sc_in<bool> req_s2p, clk;
    sc_out<bool> ack_s2p,cmd_tra, cmd_data;
    sc_biguint<80> aux1;
    int cont_bit, aux2;

    void thread(){
        cont_bit = 80;
    }

    void trans(){
        if(req_s2p){
            if(cont_bit > 0){
                cmd_tra = 1;
                cmd_data = aux1[cont_bit-1];
                cont_bit--;
                if(cont_bit == 0 and aux2 == 0){cont_bit = 1; aux2 = 1;}
                if(cont_bit == 0 and aux2 == 1){cmd_tra = 0;ack_s2p = 1;}
            }else{cmd_tra = 0;}
        }else{ack_s2p = 0; cont_bit = 80;aux2 = 0;}
    }

    void trans1(){
        aux1 = in_mem;
    }

    SC_CTOR(seripar){
        SC_THREAD(thread);
        SC_METHOD(trans);
        sensitive << clk.pos();
        dont_initialize();
        SC_METHOD(trans1);
        sensitive << clk.pos();
    }
};

```

## APÊNDICE A.11 - Descrição do conversor Biphase-L em SystemC

```
SC_MODULE(nrz){
    sc_in<bool> a, b;
    sc_out<bool> c;
    void prc_port_or(){
        c = (!(a&b) & !((!a)&!b));
    }
    SC_CTOR(nrz){
        SC_METHOD(prc_port_or);
        sensitive << a << b;
    }
};
```

## APÊNDICE A.12 - Descrição do Seletor em SystemC-AMS

```

SCA_SDF_MODULE(Seletor)
{
    sca_sc sdf_in<bool> in_1, in_2, in_3;
    sca_sdf_out<double> out_1, out_2;

    void attributes(){out_1.set_T(sc_time(1,SC_US)); out_2.set_T(sc_time(1,SC_US));}

    void sig_proc(){
        if(in_3 == 0) {
            out_1.write(0);
            out_2.write(0);
        }
        else{ if(in_1 == 0 ) { out_1.write(0.76); out_2.write(0); }
        else if( in_2 == 1) { out_1.write(0.375); out_2.write(0.635);} //out_1.write(0.5); out_2.write(+sqrt(3)/2);}
        else { out_1.write(0.375); out_2.write(-0.635);}} //out_1.write(0.5); out_2.write(-(sqrt(3)/2));}}
    }

    SC_CTOR(Seletor){}
};

```

## APÊNDICE A.13 - Descrição do Oscilador em SystemC-AMS

```

SCA_SDF_MODULE (osc)
{
    sca_sdf_out<double> out;
    double freq_0, k_freq, DC, ampl, teta_p, sigma;

    void init(double DC, double ampl_dbm, double freq, double k_freq, double rin, double rout, double PN){
        double N0 = 4*1.38e-23*(27+273)*rout; // potencia de ruído de fase - dBc/Hz
        this -> teta_p = pow(10,(PN - 30)/10); // variação pico a pico da fase
        this -> sigma = sqrt(N0); // valor medio de ruído v/sqrt(Hz)
        double ampl_lin = pow(10,(ampl_dbm -30)/10); // sinal de saída medio - dbm
        this -> ampl = sqrt(ampl_lin*2.0*rout); // saída pico a pico
        this -> freq_0 = freq; // frequencia da fundamental - Hz
        this -> k_freq = k_freq; // frequencia de offset - Hz
        this -> DC = DC; // tensão dc do offset do oscilador - V
        srand(time(NULL));
    }

    void sig_proc(){
        // regaração de ruído aparti da potencia de ruído
        double noise = (rand()*sigma/RAND_MAX);
        double phi_P = teta_p
        out.write(DC+ampl*sin(2*M_PI*(freq_0)*sc_time_stamp().to_seconds() + phi_P*(rand() % 100)/100 + noise));
        // saída do vco
    }

    SCA_CTOR(osc){
};

```

## APÊNDICE A.14 - Descrição do defasador em SystemC-AMS

```
SCA_SDF_MODULE(defasador)
{
    sca_sdf_in<double> in;
    sca_sdf_out<double> out;
    sca_ltf_nd ltf;
    sca_vector<double> num, den;

    void init(double fc , double ang ){
        // ang = angulo de defasagem/2
        num(0)= -1.0;
        num(1)= tan(ang*M_PI/180)/(2*M_PI*fc);
        den(0)= 1.0;
        den(1)= tan(ang*M_PI/180)/(2*M_PI*fc);
    }

    void sig_proc(){
        out.write(ltf(num,den,in.read()));
    }

    SC_CTOR(defasador){}
};
```

## APÊNDICE A.15 - Descrição do misturador em SystemC-AMS

```

SCA_SDF_MODULE(mixer)
{
    sca_sdf_in<double> rf_in, lo_in;           // portas de entrada
    sca_sdf_out<double> out;                 // portas de saída
    double GP_LIN, IP3_LIN, FNOISE_DB, RIN, ROUT, fn, sigma, A, B;

    void init(double GP_DB, double IP3_DBM, double FNOISE_DB, double RIN, double ROUT){
        this -> GP_LIN = pow(10,GP_DB/10);           // ganho de malha aberta - dB
        this -> A = sqrt(GP_LIN*ROUT/RIN);          // ganho linear da tensão
        this -> IP3_LIN = pow(10,(IP3_DBM - 30)/10); // entrada de referencia ao ip3 - dBm
        this -> B = A/pow(sqrt(IP3_LIN*2*RIN),2)*(4/3); // coeficiente de terceira ordem
        this -> fn = pow(10,FNOISE_DB/10);          // figura de ruído - dB
        this -> RIN = RIN;                          // resistência de entrada do sinal - Ohm
        this -> ROUT = ROUT;                         // resistência de saída
        double N0 = 4*(fn - 1)*pow(10,-228.6/10)*(27+273)*50; // potencia de ruído
        this -> sigma = sqrt(N0);                   // valor médio de ruído
        srand(time(NULL));                          // inicio da contagem da função rand
    }

    void sig_proc()
    {
        double rf = rf_in.read();                 // entrada do sinal
        double lo = lo_in.read();                 // entrada do oscilador
        double rf_dist = ((A*rf)-(B*pow(rf,3))); // implementação da distorção
        double mix_dist = rf_dist * lo;          // definição do mixer
        double noise = (rand()*sigma/RAND_MAX); // geração do ruído
        out.write(mix_dist + noise);
    }

    SCA_CTOR(mixer) {}
};

```

## APÊNDICE A.16 - Descrição do Somador em SystemC-AMS

```

SCA_SDF_MODULE(somador)
{
sca_sdf_in<double> in_1, in_2;
sca_sdf_out<double> out;
double gain_power, a1, a3, AIP3, sigma;
double rin, rload;

void init(double gain, double iip3, double nf, double rin, double rload){
    double f = pow(10,nf/10), N0 = 4*(f-1)*pow(10,-228.6/10)*(27+273)*50; //potencia de ruído
    this -> sigma = sqrt(N0); // valor medio de ruído v/sqrt(Hz)
    srand(time(NULL)); // inicio da contagem do rand
    this -> rin = rin; // resistencia de entrada
    this -> rload = rload; // resistencia da carga
    this -> AIP3 = sqrt(pow(10,iip3/10)*2*rin*0.001); // ip3 dBm
    this -> gain_power=pow(10,gain/10); // ganho do amplificador
}
void sig_proc(){

double noise = (rand()*sigma/RAND_MAX); // regaração de ruído aparti da potencia de ruído
this -> a1=sqrt(gain_power*rload/rin); // componente de distorção de primeira ordem
this -> a3=a1/(3*pow(AIP3,2)/4); // componente de distorção de terceira ordem
double input = in_1.read()+in_2.read() + noise; // sinal de saída com ruído
out.write(a1*input-a3*pow(input,3));
}

SC_CTOR(somador){}

};

```



## APÊNDICE A.17 - Descrição do amplificador de Potência em SystemC-AMS

```

SCA_SDF_MODULE (PA)
{
sca_sdf_in<double> in;
sca_sdf_out<double> out;
double gain_power, a1, a3, AIP3, sigma;
double rin, rload;

void init(double gain_power_db, double iip3, double nf, double rin, double rload){
double f = pow(10,nf/10), N0 = 4*(f-1)*pow(10,-228.6/10)*(27+273)*50; //potencia de ruído

this -> sigma = sqrt(N0); // valor medio de ruído v/sqrt(Hz)
srand(time(NULL)); // inicio da contagem da função rand
this -> rin = rin; // resistencia de entrada
this -> rload = rload; // resistencia da carga
this -> AIP3 = sqrt(pow(10,iip3/10)*2*rin*0.001); // ip3
this -> gain_power=pow(10,gain_power_db/10); // ganho do amplificador
}

void sig_proc(){

double noise = (rand()*sigma/RAND_MAX); // regaração de ruído a partir da potencia de ruído
this -> a1=sqrt(gain_power*rload/rin); // componente de distorção de primeira ordem
this -> a3=a1/(3*pow(AIP3,2/4)); // componente de distorção de terceira ordem
double input = in.read() + noise; // sinal de saída com ruído
out.write(a1*input-a3*pow(input,3));
}

SC_CTOR (PA) {}
};

```