

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

PRÓ-REITORIA DE GRADUAÇÃO

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**SISTEMA DE CONTROLE PARA RASTREIO DE SATÉLITE
UTILIZANDO PLATAFORMA LABVIEW**

Tiago Rocha Gonçalves

2011048614

Natal – Rio Grande do Norte – Brasil

Novembro de 2015

TIAGO ROCHA GONÇALVES

**SISTEMA DE CONTROLE PARA RASTREIO DE SATÉLITE
UTILIZANDO PLATAFORMA LABVIEW**

Trabalho de Conclusão de Curso
apresentado ao Curso de graduação de
Engenharia Elétrica da Universidade
Federal do Rio Grande do Norte.

Orientador: Professor Dr. José Alfredo Costa

Coorientador: Dr. José Marcelo Lima Duarte

Natal – Rio Grande do Norte – Brasil

Novembro de 2015

Agradecimentos

Aos meus pais por possibilitar uma excelente educação, ao meu irmão e a minha irmã por me incentivarem.

À minha namorada e amiga Carol pelo apoio em todos os momentos.

Ao meu orientador José Alfredo pelas informações compartilhadas.

A José Marcelo por toda a ajuda e orientação durante a bolsa no INPE.

Ao CNPq pelo apoio financeiro de bolsas de IC e pelo Ciência Sem Fronteiras.

A todos os amigos que contribuíram e torcem pelo meu sucesso.

RESUMO

Esse trabalho tem o objetivo de descrever o programa implementado em LabVIEW que proporciona o funcionamento do Subsistema de Controle e Rastreo (SCR) da estação multimissão de Natal (EMMN). Um primeiro sistema de controle foi implementado em plataforma Linux terminal. Entretanto, devido a pouca interface com o usuário e relativa dificuldade de utilização foi decido refazer o sistema de controle em plataforma *Windows* utilizando o software LabVIEW. Tal programa apresenta uma interface mais amigável com o usuário e é capaz de realizar a aquisição de dados em tempo real. Além de utilizar ferramentas poderosas e funções que facilitam a implementação e visualização dos dados.

Lista de Figuras

Figura 1 – Principais tipos de dados no LabView.	9
Figura 2 – Exemplo de programação por fluxo de dados.	10
Figura 3 – Exemplo de um VI.	10
Figura 4 – Estrutura da estação com a conexão entre os módulos	13
Figura 5 –Diagrama de controle do sistema.....	14
Figura 6 – Foto da antena do sistema.	16
Figura 7 – Frente e interior da gaveta de potência.	17
Figura 8 – Frente e interior do módulo de comando.	17
Figura 9 – Lógica Ladder simplificada da alimentação interna da Modulo de Comando.	19
Figura 10 – Lógica Ladder simplificada do acionamento das bobinas KA1 e KA2.	19
Figura 11 – Lógica Ladder simplificada referente ao freio do eixo elevação	20
Figura 12 – Fluxograma simplificado do sistema de controle e de rastreo.....	23
Figura 13 – Fluxograma do subVI Tratar Efemérides.	25
Figura 14 – Diagrama de blocos da subVI-Tratar Efemérides.	26
Figura 15 – Fluxograma simplificado do subVI Mudar Referência.	28
Figura 16 – Diagrama de blocos da subVI Mudar Referência.....	29
Figura 17 – Fluxograma do subVI – Posicionamento.	31
Figura 18 – Diagrama de blocos responsável pelo acionamento da gaveta de potência e do sistema de posicionamento.....	32
Figura 19 – Diagrama de blocos do subVI Apontamento no LabVIEW.....	33
Figura 20 – Conversão entre as escalas.....	34
Figura 21 – Fluxograma do SubVI – Rastreo.....	35
Figura 22 – Diagrama de blocos do LabVIEW do sistema para alterar index do rastreo.....	36
Figura 23 – Diagrama de blocos do LabVIEW do subVI Rastreo.....	36
Figura 24 – Fluxograma simplificado do subVI – Status do Sistema.	37
Figura 25 – Diagrama de blocos LabVIEW do subVI Status do Sistema.....	38

Lista de Abreviaturas e Siglas

CBERS	China-Brazil Earth Resources Satellite
EMMN	Estação Multimissão de Natal
INPE	Instituto Nacional de Pesquisas Espaciais
LabVIEW	Laboratory Virtual Instrument Engineering
PCD	Plataforma de Coleta de Dados
SACI	Satélite Científico
SBCDA	Sistema Brasileiro de Coleta de Dados Ambientais
SCD	Satélite de Coleta de Dados
STK	System Tool Kit

Sumário

Lista de Figuras	5
Lista de Abreviaturas e Siglas	6
1. Contextualização	8
2. Introdução ao LabVIEW	9
3. Introdução ao Sistema	12
3.1. A Teoria do Controle	14
3.2. Módulo Posicionador	15
3.3. Módulo de Potência	16
3.4. Módulo de Comando.....	17
3.5. Módulo de Controle	20
4. Software de Controle e Rastreo em LabVIEW	23
4.1. SubVI – Tratar Efemérides	24
4.2. SubVI – Mudar Referência.....	27
4.3. SubVI – Posicionamento.....	30
4.3.1. Adaptação da leitura do resolver para o fim de curso da antena	33
4.4. SubVI - Rastreo	34
4.5. SubVI – Status do Sistema	37
5. Conclusões e Recomendações	39
6. Referências Bibliográficas	40

1. Contextualização

Foram desenvolvidos no INPE, dois Satélites de Aplicação Científica (SACI) no qual não chegaram a operar em órbita devido a falhas: o SACI-1 e o SACI-2. Segundo [1], o primeiro chegou a ser lançado em órbita, porém houve uma falta de transmissão de qualquer sinal à base devido a problemas de equipamentos de bordo. Já o segundo satélite, nem se quer chegou a sua órbita uma vez que foi necessário abordar o lançamento e acionar a autodestruição do mesmo.

A Estação Multimissão de Natal consiste em um projeto de pesquisa e desenvolvimento tecnológico, que faz o reaproveitamento da estação francesa destinada aos Satélites de Aplicação Científica que não chegaram a operar. A EMM-Natal será uma estação para múltiplas missões, onde está sendo projetada para agregar as funcionalidades de centro de controle e missão.

O Estado Atual do projeto consistiu no desenvolvimento de um software para controle em plataforma Linux sem interface gráfica. Para tal, foi necessário realizar uma engenharia reversa com a finalidade de transferir o sistema de controle embarcado para o PC. O programa desenvolvido era capaz de realizar o rastreamento de satélites com sucesso. Entretanto, devido a relativa dificuldade para operar o sistema devido a ausência de um terminal, foi sugerido desenvolver um novo software para controle de posição da antena com interface gráfica amigável e compatível com o sistema operacional Windows. O escopo desse trabalho consiste na descrição da estação EMM e desenvolvimento de um programa de controle para o rastreamento de satélite com uma interface gráfica amigável.

2. Introdução ao LabVIEW

Para o desenvolvimento desse trabalho foi utilizado o programa LabVIEW, o qual possui uma linguagem de programação gráfica que utiliza ícones ao invés de linhas de texto. Os programas do LabVIEW são chamados de instrumentos virtuais (VI), pois são bem similares em aparência aos instrumentos físicos como osciloscópios e multímetros. Ao criar um novo VI, o usuário se depara com duas janelas principais: o Painel Frontal e o Diagrama de Blocos. O Painel Frontal corresponde à interface do usuário com o programa em execução. No ambiente do Painel Frontal é possível: determinar como os dados serão visualizados, mudar as entradas e ver os dados atualizados em tempo real. No Diagrama de Blocos encontram-se terminais, subVIs, funções, constantes, iterações, estruturas e fios que são utilizados para conectar objetos para permitir a transferência de dados entre eles.

O programa utiliza diferentes cores para indicar o tipo de dado que está sendo conectado. A Figura 1, retirada do manual, resume os principais tipos de dados utilizados nos VI.

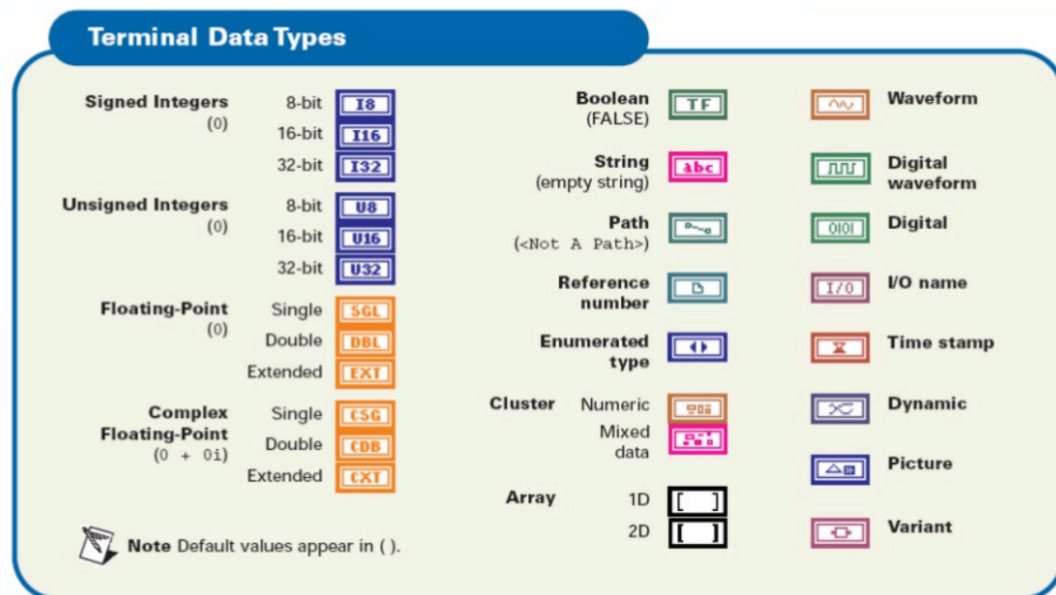


Figura 1 – Principais tipos de dados no LabView.

Diferentemente das outras linguagens de programação, o LabVIEW utiliza programação baseada em fluxo de dados e não uma sequência de códigos no qual a ordem do código determina a execução do sistema. Para esclarecer o funcionamento do programa nesse

tipo de linguagem, será utilizado dois breves exemplos. Diversos exemplos e descrições detalhadas estão presentes no site do fabricante *National Instruments*.

Ao executar um programa o software analisa primeiramente os nós do sistema. Isso significa que determinado nó do sistema é executado apenas quando todas suas entradas tiverem dados disponíveis para posteriormente o resultado ser encaminhado para o nó seguinte. Como exemplo de programação de fluxo de dados, considere o diagrama de blocos da Figura 2, que soma dois números e subtrai de outro terceiro número já pré-fixado como uma constante *double* de valor dez, por exemplo. Nesse caso, existem dois nós correspondendo à função soma e subtração. A função soma é executado primeiro porque a função subtrair precisa do resultado para uma de sua entrada.

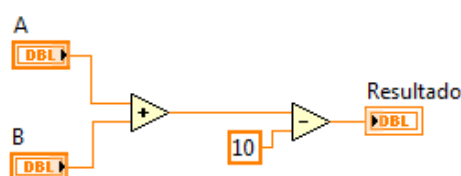


Figura 2 – Exemplo de programação por fluxo de dados.

Considerando um sistema um pouco maior com mais entradas mostrado na Figura 3, pode ser feita a seguinte questão: qual seria o segmento de código executado primeiro? – a função de soma, número aleatório ou divisão. A resposta é que não é possível saber, pois as entradas da função soma e divisão são disponibilizadas ao mesmo tempo e já a função de gerar um número aleatório não possui sequer entrada. Nos casos quando há necessidade de que um determinado segmento de código seja executado antes que outro, e assim seguir uma determinada sequência, é possível utilizar estruturas de sequências para forçar a ordem da execução.

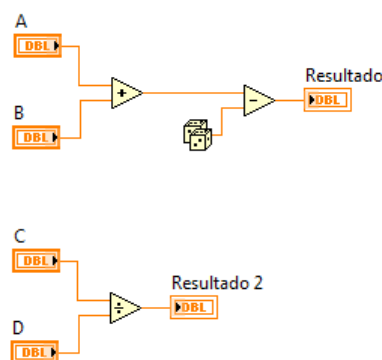


Figura 3 – Exemplo de um VI.

O uso do software LabVIEW provê diversas vantagens como: melhor interação visual com o usuário, facilidade no processo de aquisição e análises dos dados. Com relação às linguagens baseadas em texto, uma grande vantagem consiste na facilidade em criar componentes que executam paralelamente. Devido a essas vantagens, o LabVIEW foi escolhido para implementar o programa de controle de posição da antena.

Por outro lado, a desvantagem de utilizar blocos é a poluição visual no programa, visto que são utilizados diversos blocos às vezes para realizar operações que em linha de código seria bastante breve. Para contornar esse problema, o LabVIEW permite ao programador utilizar subVIs que consistem em seções menores de um VI que realizam uma específica função utilizada em um VI mais externo. Um subVI é similar a uma subrotina em programação baseada em texto. Permite, então, obter um código mais compacto, subdividido e mais organizado. Com esse intuito foram utilizados subVIs na implementação do programa.

3. Introdução ao Sistema

Este trabalho descreve o funcionamento de todo o sistema: hardware da Estação Multimissão de Natal e software responsável pelo controle e rastreamento da EMMN. Para um melhor entendimento, será abordado primeiramente características e informações relacionadas aos componentes básicos do sistema (como os módulos e motores) e posteriormente será introduzido a técnica de controle utilizada para o rastreamento junto com explicações detalhadas de como foi implementado no software LabVIEW.

O objetivo do trabalho consiste em implementar um programa no LabVIEW para permitir que a antena da estação seja capaz de apontar e seguir a trajetória do satélite durante toda sua passagem. Para isso, são utilizadas efemérides, que são um conjunto formado pelos ângulos azimute e elevação e um instante de tempo associado que determinam a posição e o momento do apontamento da antena. Para esse trabalho, as efemérides consistem nas entradas do sistema, ou seja, referências que carregam a que ângulo e instante de tempo a estação deverá apontar para o céu. As efemérides são obtidas a partir de softwares específicos como, por exemplo, o STK. Nele é inserida a posição geográfica da antena e, então, é selecionado qual satélite deseja-se seguir. Por fim, o software gera uma lista com horários e ângulos azimute e elevação correspondente a passagem de um determinado satélite.

Devido às falhas das missões dos satélites SACI-1 e SACI-2 a estação de rastreamento dos satélites SACI ficou sem uso. Nesse sentido, surgiu a ideia por parte de engenheiros do próprio INPE de reutilizar os equipamentos para outro objetivo como realizar o rastreamento de satélites que operam e contribuem para o SBCDA como o satélite CBERS-4, o SCD-1 e o SCD-2. A antena da estação trabalha na banda S, que engloba frequências de 2,0 GHz e permite ter aplicações de transmissão e recepção das mais variáveis possíveis como comandar futuros satélites via telecomando.

Os componentes hardware que foram utilizados corresponderam a quatro módulos. Cada um destes módulos será abordado detalhadamente. Porém, para fins de introdução ao sistema será brevemente comentado sobre cada um destes. Os módulos são:

- Módulo Posicionador
- Módulo de Potência (gaveta de potência – *puissance*)
- Módulo de Comando (gaveta de *pilotage*)
- Módulo de Controle (computador)

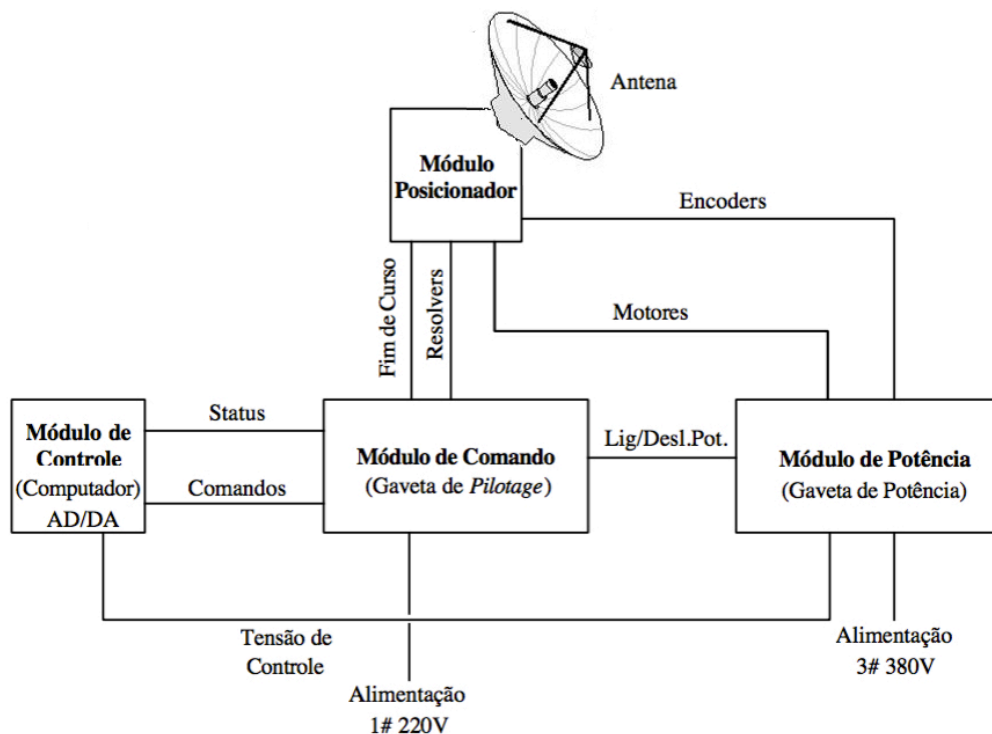


Figura 4 – Estrutura da estação com a conexão entre os módulos .

O Módulo Posicionador é basicamente composto pelos dois motores que movimentam separadamente a antena no sentido azimuth e elevação, por sensores de fim de curso, redutores coaxiais (responsável por diminuir a velocidade e aumentar o torque) e pelos sensores de posição chamados de Resolvers.

O Módulo de Potência é onde estão instalados os dois inversores que acionam os motores que estão localizados no Módulo Posicionador. Estes inversores funcionam comandados pelo o módulo de comando.

O Módulo de Comando corresponde ao centro do sistema, no qual inclui a alimentação do sistema e LEDs que possuem o objetivo de indicar o usuário sobre possíveis eventos que podem ocorrer como: fim de curso acionado, rastreo iniciado ou gaveta de potência ligada etc. Assim, esse módulo se comunica com o próximo módulo mandando constantemente o status do sistema.

O Módulo de Controle é composto pela placa de aquisição de dados PCI 6025 E da *National Instruments* e pelo computador. O programa implementado em LabVIEW atua neste computador verificando constantemente o status do sistema (repasado pelo módulo de comando) e indicando para o usuário final como também sendo responsável pelo início do rastreo, acionando o sistema completo automaticamente e desligando quando necessário.

3.1. A Teoria do Controle

O sistema de controle deve posicionar adequadamente a antena em ambos os eixos, azimute e elevação, a cada intervalo de amostragem. Como a antena mantém sua posição constante quando o sinal de controle aplicado nela é nulo, seu comportamento pode ser modelado por um integrador. Os sinais de referência para o azimute e para a elevação podem ser aproximados por uma rampa se o período de tempo de análise desses sinais for na ordem de grandeza do tempo de estabilização da planta. Entretanto, se todo o período do rastreo for considerado, não é possível representar o sinal de referência por uma reta. Logo é necessário a utilização de um controlador do tipo Proporcional Integrativo (PI) para melhorar o erro em regime. A Figura 5 apresenta o diagrama de blocos de controle do sistema.

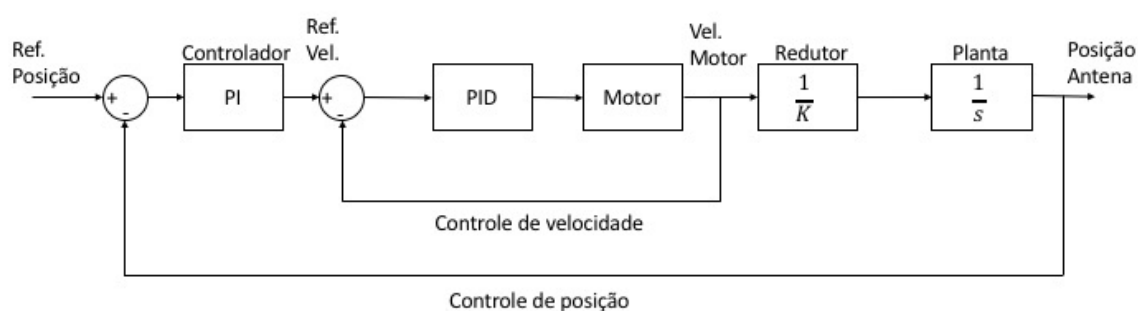


Figura 5 –Diagrama de controle do sistema.

Os inversores, localizados no módulo de potência, controlam a velocidade dos motores com um controlador do tipo Proporcional Integrativo Derivativo (PID) presente nos próprios inversores. Ao analisar a malha interna, responsável pelo controle de velocidade, é possível considerar que quando o inversor recebe a referência da velocidade na qual ele deve impor ao motor da antena, todo esse processo é feito de maneira extremamente mais rápida comparado com a malha externa que corresponde ao controle de posicionamento. Assim, para fins de simplificação, o controle de velocidade pode ser desconsiderado já que é muito mais rápido que o controle de posição.

O controle de posicionamento é feito então por um controlador do tipo PI que é responsável por comparar a posição de referência (no caso das efemérides) com a posição atual da antena. A diferença entre as duas grandezas é o erro de posição passado para o controlador PI para geração do sinal de controle. O referido sinal de controle corresponde a

um sinal analógico que varia entre -10 V e +10 V e corresponde a velocidade de rotação desejada para o motor. O período de amostragem e as constantes do controlador PI digital foram mantidas as mesmas do projeto anterior e correspondem a:

$$T = 1 \text{ s}$$

$$k_{pELE} = k_{pAZ} = 0,6 \quad (1)$$

$$k_{iELE} = k_{iAZ} = 0,05 \quad (2)$$

3.2.Módulo Posicionador

O Módulo Posicionador é composto pelos motores e redutores coaxiais que juntos são responsáveis por mover a antena nos dois eixos: azimute e elevação de forma independente. Acoplado a duas de suas engrenagens, estão dois *resolvers*, um para medição do azimute e outro para medição da elevação da antena. O *resolver* é um sensor que mede a posição angular instantânea de um eixo. Fisicamente, o *resolver* se assemelha a um pequeno motor, tendo uma bobina de excitação em seu rotor e duas bobinas no estator com 90 graus de defasagem entre elas, aqui denominadas de x e y. Um dispositivo denominado Conversor Resolver para Digital gera o sinal senoidal de excitação e processa os sinais senoidais induzidos nas bobinas x e y. A amplitude e polaridade relativa dos sinais x e y permitem a obtenção do ângulo do eixo. O resultado da medição é disponibilizado pelo conversor através de uma saída digital.

Para a proteção do sistema, existem três mecanismos que trava o sistema e não permite que a antena ultrapasse certa posição.

1. A proteção via software (LabVIEW) que, a partir da análise prévia do rastreamento, impede que seja enviado um comando para antena que irá ultrapassar o fim de curso.
2. A proteção elétrica (Módulo de Comando) que é acionada a partir da sensibilidade de sensores de fim de curso que através de contactores corta a alimentação dos motores.
3. A proteção mecânica (Módulo Posicionador) que consiste em borrachas no fim de curso que em último caso atua amortecendo o impacto.

Como visto no diagrama da Figura 5, existe um redutor descrito pelo ganho ($1/K$) no diagrama. Um redutor é uma caixa de engrenagens responsável por reduzir a velocidade de rotação para aumentar o torque. Isso se faz necessário, visto que a antena é relativamente pesada. Além do redutor, contrapesos são utilizados para facilitar a movimentação da antena. A Figura 6 ilustra a antena da estação.



Figura 6 – Foto da antena do sistema.

3.3.Módulo de Potência

O Módulo de Potência possui dois inversores, um para o azimuth e outro para a elevação, responsáveis pelo controle de velocidade e acionamento dos motores. A velocidade de referência para os motores é estabelecida através de um sinal de tensão analógico externo entre -10V e +10V, sendo o valor de 10V correspondendo à velocidade nominal de 3000 rpm, 0V à velocidade nula e -10V à velocidade nominal, porém no sentido contrário. O sistema de controle de velocidade do inversor garante que a velocidade de referência seja atingida com precisão graças ao sensor de velocidade de rotação que fecha a malha de controle.



Figura 7 – Frente e interior da gaveta de potência.

3.4.Módulo de Comando

No módulo de comando (gaveta *pilotage*) estão presentes os LEDs para visualização dos eventos ocorridos, fontes de alimentação do sistema, a placa conversora SOTEREM 2266-1 e circuitos de comando para alimentação dos inversores. Além disso, existe um autotransformador (230V/48V) responsável por alimentar o circuito de aquecimento do Módulo Posicionador. Porém, não se faz necessário para utilização no Brasil visto que não ocorre o congelamento dos componentes ao longo do ano.

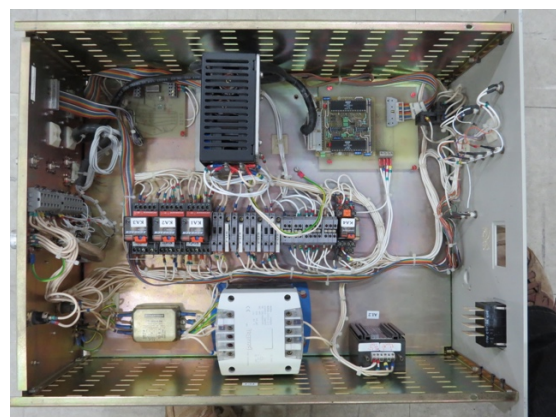


Figura 8 – Frente e interior do módulo de comando.

A placa SOTEREM 2266-1 é um conversor Resolver para Digital que faz interface com os dois *resolvers* do Módulo Posicionador. O resultado da medição dos ângulos é fornecido através de duas saídas digitais de 16-bit em paralelo, uma para a elevação e outra para o azimuth. Como cada eixo exige 16 bits, ao total seriam necessários 32 bits, ou seja, 32 canais na placa de aquisição de dados no Módulo de Controle. Entretanto, esta possui uma limitação de 32 canais digitais no total. Como são necessárias outras entradas e saídas digitais da placa para realizar a comunicação com o sistema, é inviável o uso dos 32 canais apenas para leitura da posição de cada eixo.

Com a finalidade de reduzir o número de canais necessários para leitura da posição atual da antena, foi criada a placa INPE-01 que conecta os 16 bits de cada posição (azimute e elevação) em paralelo. O funcionamento é similar a de um multiplexador que possui uma entrada de seleção responsável por liberar uma ou a outra entrada na saída. Nesse sentido, os pinos 19B e 25A da placa SOTEREM 2266-1 permitem controlar os instantes da leitura da posição da antena do eixo azimuth e elevação, respectivamente. Quando essas entradas se encontram em nível lógico alto o processo de medição do conversor Resolver para Digital continua em ação, assim o resultado da conversão não pode ser lido, e as saídas ficam com impedância alta. Já em nível lógico baixo, o processo de conversão é parado e a leitura pode ser feita. Assim, foram configurados como saídas digitais da placa conversora no módulo de controle os pinos 55 e 57 responsáveis por liberar leitura azimuth e elevação uma vez que estes se conectam com os pinos 19B e 25A da placa SOTEREM, respectivamente.

A Figura 10 apresenta o esquema geral de alimentação do Módulo de Comando em lógica Ladder. A alimentação das fontes AL1 e AL2 é feita após o acionamento da Chave Principal do módulo de comando. Entretanto, para energizar o módulo de potência é preciso acionar as bobinas KA2 e KA3.

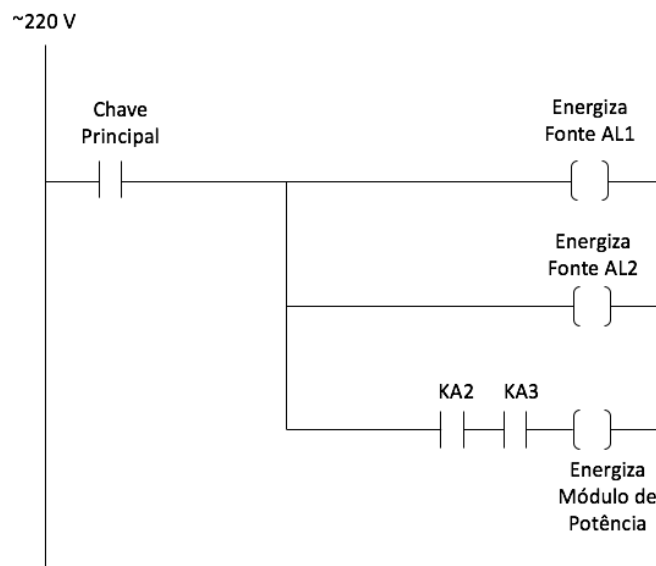


Figura 9 – Lógica Ladder simplificada da alimentação interna da Módulo de Comando.

A Figura 11 representa o funcionamento em lógica Ladder para o acionamento das bobinas KA1 e KA2. O botão de emergência se encontra no Módulo Posicionador, próximo à antena, e o botão *Forçage Butée* está localizado no painel frontal do Módulo de Comando.

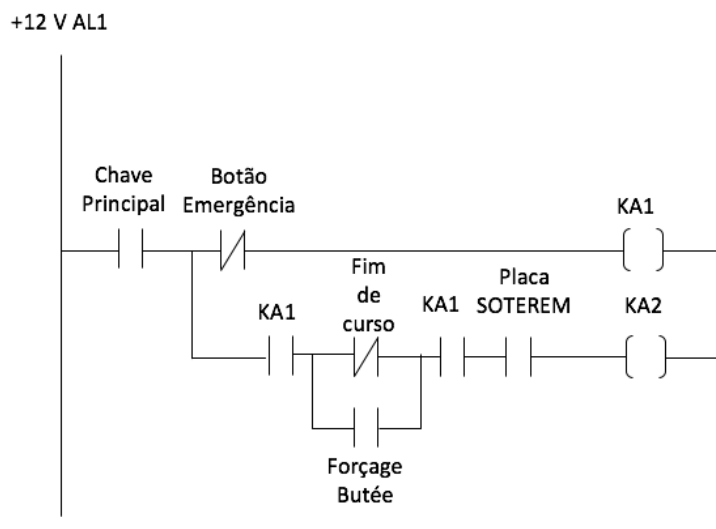


Figura 10 – Lógica Ladder simplificada do acionamento das bobinas KA1 e KA2.

Uma vez energizada, a bobina KA2 é responsável por retirar o freio do eixo elevação. A Figura 12 mostra a lógica Ladder simplificada referente ao freio do eixo elevação e o acionamento da bobina KA3.

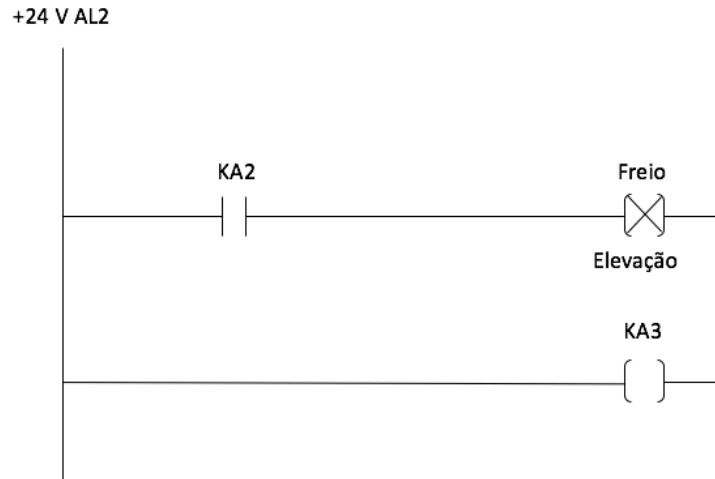


Figura 11 – Lógica Ladder simplificada referente ao freio do eixo elevação

É possível concluir que a bobina KA2 é responsável pelo acionamento do módulo de potência visto que a bobina KA3 é energizada diretamente a partir da alimentação da fonte AL2. Entretanto, a bobina KA2 depende do contato da placa SOTEREM para ser energizada. Os contatos da placa SOTEREM são comutados pelo Módulo de Controle através da placa de aquisição de dados, o que a torna responsável pelo acionamento do módulo de potência.

3.5.Módulo de Controle

Esse último módulo é composto pelo computador e pela placa de aquisição de dados responsável pela ponte entre o computador e os sinais de entrada e saída da gaveta de comando. A placa de aquisição de dados é da *National Instruments*, modelo PCI 6025 E, que possui 32 canais digitais para entrada/saída e duas saídas analógicas. Os canais digitais são usados para ler a posição atual da antena, liberar ou bloquear a leitura do azimuth/elevação, ligar a gaveta de potência, acionar LEDs presentes na gaveta de comando que indicam o status do sistema etc. Já as duas saídas analógicas são utilizadas para gerar os dois sinais analógicos de -10V a +10V que vão para os inversores da Gaveta de Potência. A Tabela 1 apresenta todos os canais digitais e analógicos da placa de aquisição de dados e as suas respectivas configurações:

Tabela 1 – Canais digitais e analógicos da placa AD/DA e suas configurações

PLACA PCI 6025 E			
ID	PINO	Descrição	Configuração
P0.0	25	LED Butées	Saída Digital
P0.1	27	Ligar Gaveta de Potência	Saída Digital
P0.2	29	LED Poursuite	Saída Digital
P0.3	31	LED Arrêt d’urgence	Saída Digital
P0.4	26	LED Puissance	Saída Digital
P0.5	28	LED Défaut	Saída Digital
P0.6	30	Defeito AZ	Entrada Digital
P0.7	32	Defeito ELE	Entrada Digital
P1.0	97	Bit00 (LSB)	Entrada Digital
P1.1	95	Bit01	Entrada Digital
P1.2	93	Bit02	Entrada Digital
P1.3	91	Bit03	Entrada Digital
P1.4	89	Bit04	Entrada Digital
P1.5	87	Bit05	Entrada Digital
P1.6	85	Bit06	Entrada Digital
P1.7	83	Bit07	Entrada Digital
P2.0	81	Bit08	Entrada Digital
P2.1	79	Bit09	Entrada Digital
P2.2	77	Bit10	Entrada Digital
P2.3	75	Bit11	Entrada Digital
P2.4	73	Bit12	Entrada Digital
P2.5	71	Bit13	Entrada Digital
P2.6	69	Bit14	Entrada Digital
P2.7	67	Bit15(MSB)	Entrada Digital
P3.0	65	Status Bobina KA2	Entrada Digital
P3.1	63	Status Bobina KA1	Entrada Digital
P3.2	61	Chave Manuel	Entrada Digital
P3.3	59	Status Bobina KA4	Entrada Digital
P3.4	57	Libera Leitura Azimute (19B)	Saída Digital

P3.5	55	Libera Leitura Elevação (25A)	Saída Digital
P3.6	53	LED Manuel	Saída Digital
P3.7	51	LED Survie	Saída Digital
AO-0	20	Sinal pro Inversor Azimute	Saída Analógica
AO-1	21	Sinal pro Inversor Elevação	Saída Analógica

4. Software de Controle e Rastreo em LabVIEW

Nessa seção do relatório é abordado os procedimentos adotados na implementação do software responsável pelo controle e rastreo. Foram criados fluxogramas com o intuito de ajudar no entendimento de cada parte do programa implementado. A Figura 12 descreve o funcionamento geral do sistema de controle e de rastreo de forma simplificada.

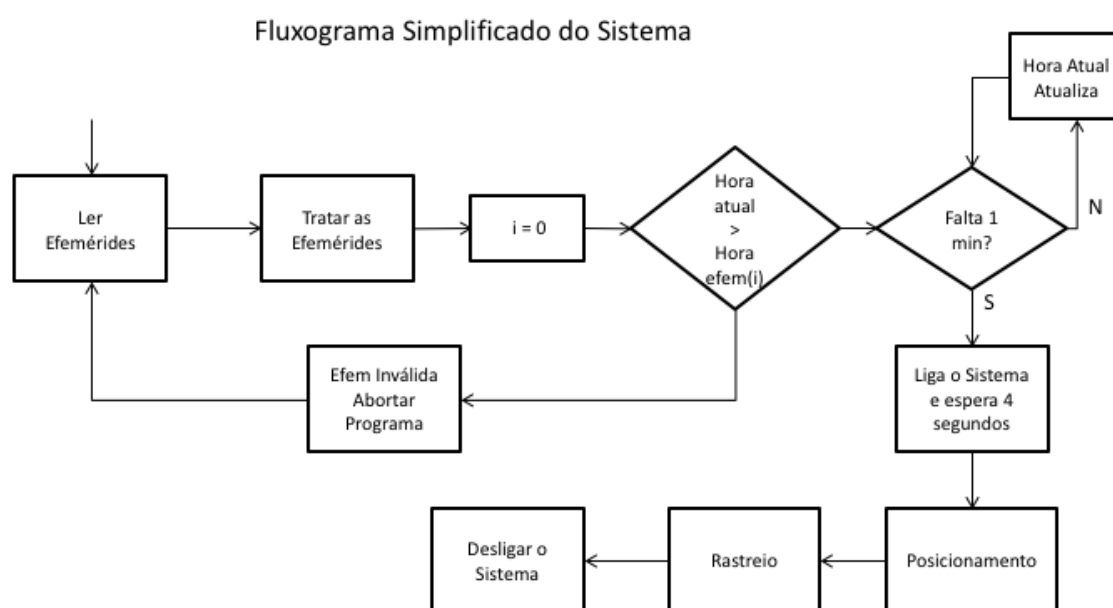


Figura 12 – Fluxograma simplificado do sistema de controle e de rastreo.

Para o funcionamento do programa implementado no LabVIEW, é necessário inserir as efemérides, como mencionado na Seção 3. Independente do software a ser utilizado para se obter o conjunto de efemérides, é primordial que o intervalo entre elas seja de um segundo para o funcionamento adequado do sistema. Isso ocorre, visto que para o momento do rastreo foi implementado no LabVIEW uma comparação da hora atual com a hora de cada efeméride para determinar os instantes de amostragem e como os parâmetros do controlador k_p e k_i foram obtidos para o tempo de amostragem igual a um segundo, é imprescindível o período de amostragem coincidir com o valor adotado no projeto do controlador para que o comportamento do sistema seja conforme o esperado.

Em seguida, o software realiza uma varredura no arquivo de texto inserido a fim de criar uma matriz de dimensão $n \times m$ no formato *string*. Ou seja, n é o número de linhas que equivale ao número total de efemérides inseridas e m é o número de colunas que delimitam as

informações de cada efemérides. Posteriormente é feito o tratamento das efemérides que basicamente consiste em separar a hora/data e o ângulo azimute/elevação de cada efeméride para uso do sistema de controle.

No LabVIEW existe um bloco que retorna a hora atual com precisão de milissegundos, chamado de *Get Date/Time in Seconds*. Esse bloco é utilizado para fazer uma primeira comparação para identificar se o instante da primeira efeméride já passou. Se for o caso, o rastreo é abortado visto que a passagem já ocorreu, ou já se encontra em andamento. Caso contrário se espera até que falte um minuto para a primeira efeméride para realizar o acionamento de todo o sistema. Foi implementado um cronômetro regressivo que disponibiliza para o usuário quanto tempo falta para a primeira efeméride. O sistema de controle de posicionamento é responsável por posicionar a antena para a primeira efeméride antes mesmo de ela ocorrer com a finalidade de realizar o rastreo desde o início da passagem. Após o fim da contagem regressiva de um minuto, é dado início ao rastreo. Nessa etapa do processo, as efemérides são atualizadas a cada um segundo, alterando então as referências que devem ser apontadas a cada instante. É feita constantemente uma análise para saber se a efeméride é a última (se é igual ao número de linhas n da matriz). Assim, quando a análise é verdadeira indica que o rastreo finalizou, e o programa deve ser desligado.

4.1.SubVI – Tratar Efemérides

Foi criado um subVI, intitulado de Tratar Efemérides, com a função de converter o formato dos dados das efemérides de matriz de *strings* para um formato mais adequado para processamento. Como saída, o subVI gera dois vetores de formato *double* de n efemérides com as referências de ângulos azimute e elevação e cria um vetor no formato *TimeStamp* com as horas e datas que devem ser apontadas. A Figura 13 representa o fluxograma do subVI – Tratar as Efemérides.

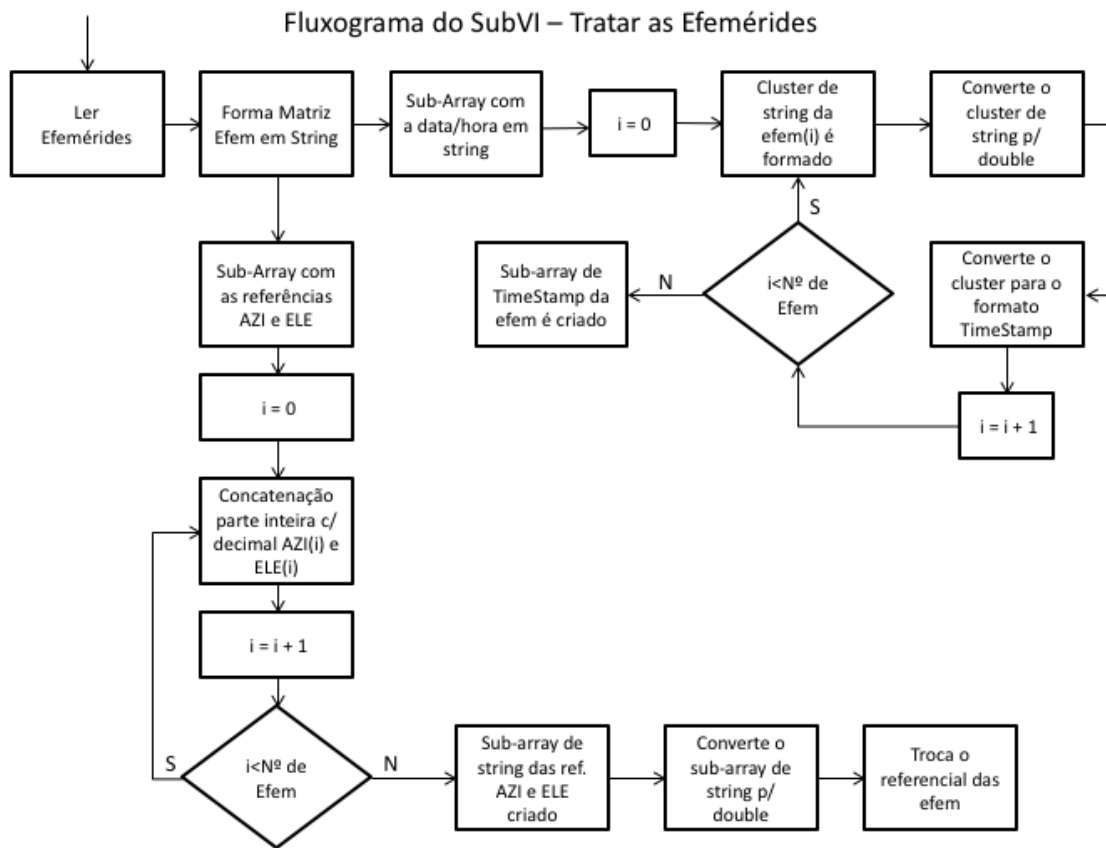


Figura 13 – Fluxograma do subVI Tratar Efemérides.

A matriz de string, entrada deste subVI, é gerada a partir do processamento de um texto em ASCII com a lista de efemérides que definem um rastreo. Neste processamento, uma coluna de dado é criada para cada delimitador encontrado no texto, e uma linha é criada para cada quebra de linha, sendo os delimitadores o ponto, a vírgula e o espaço. Como os ângulos de azimuth e elevação possuem precisão decimal, e o ponto é um delimitador, a parte inteira e a parte decimal ficam em colunas distintas na matriz gerada. Portanto, é preciso concatenar essas duas *string* e somente depois converter o valor da *string* resultante para *double*. O procedimento para geração dos dois vetores de *double* com a sequência de posições azimuth e elevação é apresentado na fluxograma da Figura 13. A implementação deste processamento no ambiente LabVIEW pode ser visualizado na parte superior da Figura 14 nas cores laranjas, que representam o formato *double*.

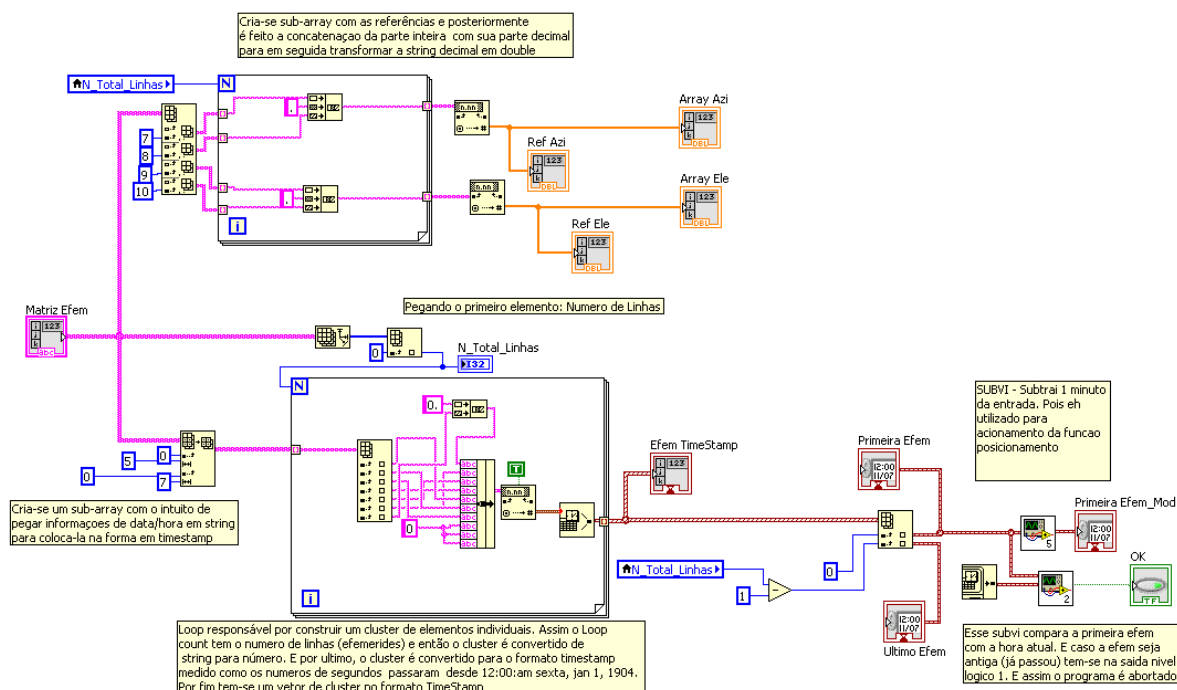


Figura 14 – Diagrama de blocos da subVI-Tratar Efemérides.

O formato de dados *TimeStamp* é comumente utilizado para aplicações que utilizam o tempo com alta precisão. De acordo com [2], esse tipo de dado é armazenado em um cluster e corresponde ao número de segundos que passaram desde as 12:00 a.m., sexta-feira dia 1º Janeiro de 1904, Tempo Universal. Foi criado um *sub-array* com apenas os valores referentes a hora e data, ainda no formato *string*. Em seguida, foi executado um *while loop* com o intuito de criar um cluster de *string* e dentro da interação cada elemento cluster foi convertido de *string* para *TimeStamp* como desejado. A cada interação do laço é feita uma análise para saber se o número de efemérides chegou ao fim. No final, tem-se então um vetor de cluster no formato *TimeStamp* permitindo, então, que seja utilizado pelo sistema de controle e rastreamento. Esse último utiliza o primeiro e o último elemento do vetor para fins de comparação com a hora atual e, assim, permitir o início ou fim do rastreamento. Com esse intuito, foram disponibilizados na saída do subVI: o primeiro elemento, o último elemento e o primeiro elemento modificado, além das referências dos ângulos no formato de vetor em *double*. A modificação do primeiro elemento consiste na subtração de um minuto para permitir que o sistema de posicionamento seja realizado antes mesmo do início da passagem.

4.2.SubVI – Mudar Referência

O objetivo desse subVI é mudar as efemérides de entrada, com posição zero do azimute no norte geográfico, para efemérides com posição zero do azimute no fim de curso da antena. Isto simplifica o algoritmo de controle que precisa evitar o fim de curso. Além disso, esse subVI precisa analisar em que quadrante do eixo elevação a antena deverá trabalhar para evitar que o rastreo passe pelo fim de curso do Azimute e, em seguida, deve mudar as referências segundo o quadrante escolhido. Em [3] é descrito o procedimento para medir a diferença entre o fim de curso no eixo azimute e o norte geográfico. O resultado desta medição, segundo [3], é de -18 graus. Assim, a relação que expressa as alterações feitas nas efemérides para a troca do referencial geográfico para o referencial do fim de curso da própria antena é:

$$azi_{ant} = azi_{geo} - 18^{\circ} \quad (3)$$

O LabVIEW possui um bloco chamado *Formula Node* que permite utilizar a linguagem C para programar uma função. As entradas da função ficam do lado esquerdo do bloco e as saídas do lado direito. Entretanto, de acordo com usuários do fórum da NI, é possível que tenha uma redução de performance ao utilizar programação baseada em texto. Devido a facilidade da programação em C, foi optado por implementar a função dessa maneira visto que o objetivo principal é o seu funcionamento e não necessariamente alta performance.

Como mencionado na Seção 4.1, entre as saídas do subVI Tratar Efemérides estão um vetor com as referências dos eixos azimute e um vetor com as referências para elevação. Essas saídas servem de entrada para esse subVI – Mudar Referência e é por isso que foram criadas duas entradas no bloco *Formula Node*. Para facilitar o entendimento, a Figura 15 apresenta o fluxograma que resume as ações dessa subVI.

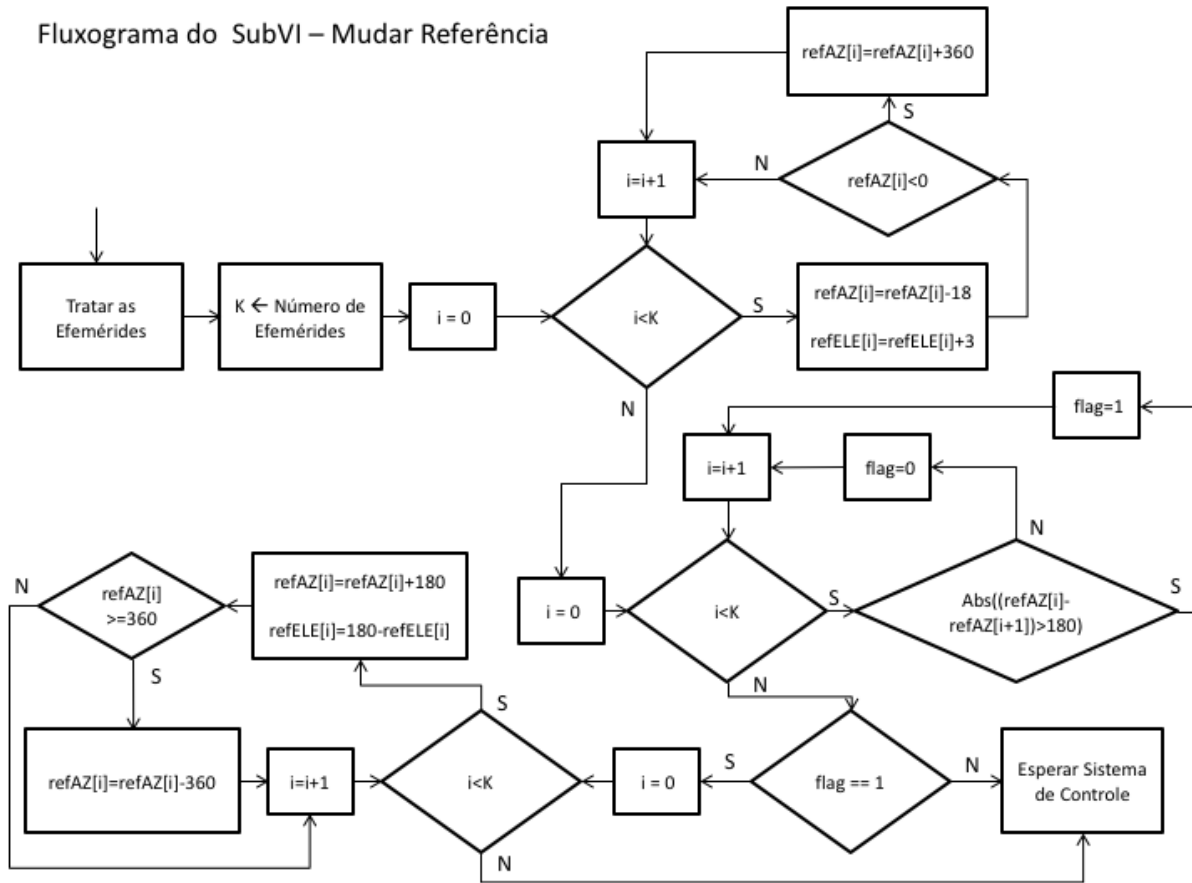


Figura 15 – Fluxograma simplificado do subVI Mudar Referência.

No bloco *Formula Node* são declaradas as variáveis no início e, logo em seguida, é calculada a dimensão do vetor *double* com as efemérides (K). Então, é executado um *for loop* com a intenção de alterar o referencial geográfico para o fim de curso da antena de todos os valores do vetor azimuth de acordo com a equação (3). Durante a interação é feita ainda outra análise com a finalidade de testar se o ângulo azimuth passou a ser negativo após a conversão do referencial. Isso pode acontecer, por exemplo, quando o rastreo iniciar com os ângulos azimuth entre 0° e 17° no referencial geográfico. Então, após a conversão para o referencial do fim de curso da antena temos os novos valores de ângulos azimutes entre -18° e -1°, respectivamente. Para garantir que o novo valor permanecesse no intervalo de 0° a 360° foi criado o seguinte algoritmo:

```

if(refAZ[i] < 0){
    refAZ[i] = refAZ[i] + 360;
}

```

A Figura 16 representa o subVI Mudar Referência implementado no LabVIEW onde é possível identificar o código em C no interior do bloco *Formula Node* e as saídas ao lado direito.

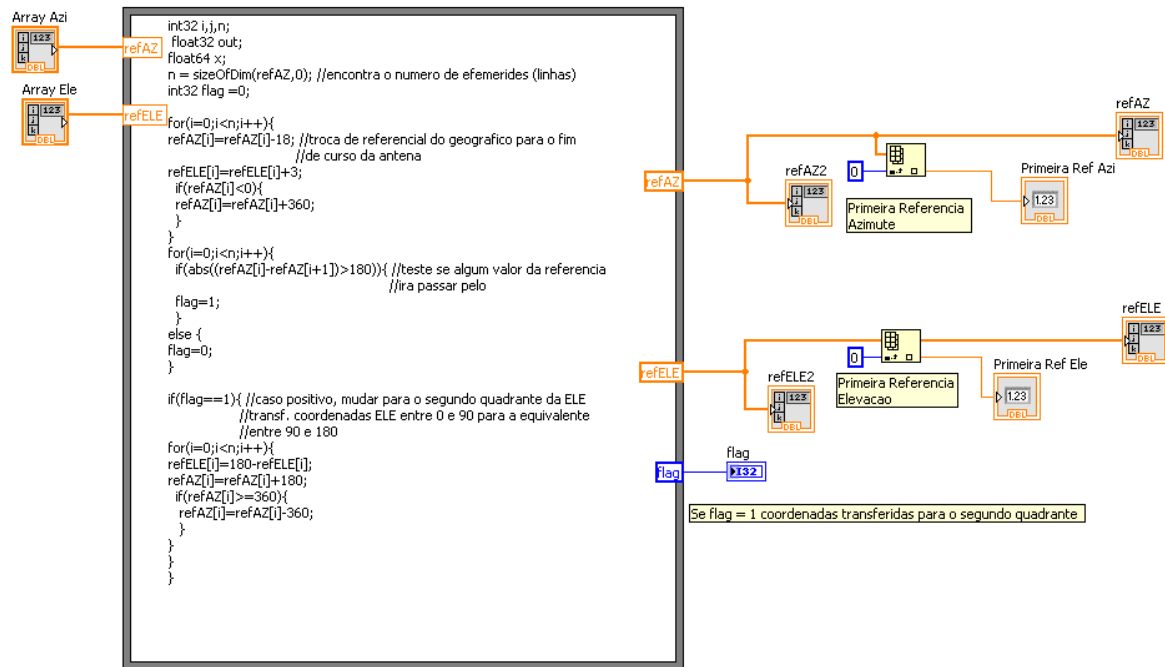


Figura 16 – Diagrama de blocos da subVI Mudar Referência.

Algumas passagens passam pelo fim de curso azimuth da antena. Dessa forma, com o intuito de evitar a interrupção do rastreamento devido ao fim de curso é realizado uma conversão nas coordenadas dessas passagens que cruzam o fim de curso para que seja utilizado o segundo quadrante do eixo elevação (entre 90° e 180°).

Nesse sentido, é verificado primeiramente se ocorrerá a passagem pelo 0°. Sabendo que a região a ser varrida no eixo azimuth, durante a passagem de um satélite é menor ou igual a 180°. Logo, é executado um *for loop* para analisar constantemente se o módulo da diferença da coordenada atual azimuth com a coordenada seguinte é maior que 180°, o que indica uma passagem pelo 0°. E o resultado desse teste condicional é armazenado em uma *flag* para sinalizar a necessidade da alteração. Assim o seguinte algoritmo foi implementado:

```

if(abs(refAZ[i] - refAZ[i + 1]) > 180){
    flag = 1;
}
else flag = 0
    
```

Por fim, é analisado o valor presente na *flag* para, então, alterar toda a passagem utilizando o segundo quadrante da elevação, caso necessário. Uma vez que é possível utilizar o segundo quadrante da elevação (90° a 180°), existem sempre duas opções em que a antena pode apontar para o mesmo ponto. Por exemplo, considerando uma efeméride com 50° az e 50° ele, a sua efeméride equivalente pode ser escrita como 230° az e 130° ele. Existe então a possibilidade que o eixo azimuth ultrapasse o valor de 360° após a alteração das coordenadas. A fim de limitar o ângulo azimuth entre 0° e 360° é subtraído 360° do valor e o ângulo excedente é o correspondente azimuth. Assim, o seguinte algoritmo foi implementado com a função de analisar o valor presente na variável *flag* e então realizar as mudanças para utilizar o segundo quadrante com objetivo de evitar a passagem pelo fim de curso azimuth.

```

if(flag == 1){
    for(i = 0; i < K; i++){
        refELE[i] = 180 - refELE[i];
        refAZ[i] = refAZ[i] + 180;
        if(refAZ[i] > 360){
            refAZ[i] = refAZ[i] - 360;
        }
    }
}

```

A saída desse subVI consiste nos valores das referências ajustadas, considerando a troca de referencial e a análise da passagem pelo fim de curso. É obtido então na saída o vetor no formato *double* de cada eixo azimuth e elevação que serão utilizados para o rastreamento do sistema. Além disso, são criados dois vetores cada um com os correspondentes primeiros elementos de cada eixo que serão utilizados para o posicionamento.

4.3.SubVI – Posicionamento

Nessa seção o foco é o sistema de controle da estação. Na implementação no LabVIEW, o sistema de controle foi separado em dois momentos distintos: o posicionamento e o rastreamento.

O subVI, intitulado de Posicionamento, foi criado com a função de posicionar a antena antes do surgimento do satélite. Como entradas esse subVI possui a posição azimuth e a posição elevação da primeira efeméride. Para o cálculo do erro de posição, é necessário obter a posição atual da antena e por isso foi utilizado um bloco chamado *DAQ Assistant* que é responsável por realizar a comunicação com a placa de aquisição de dados e ler os 16 bits referentes à posição da antena. Esse bloco resume todo o procedimento necessário para a aquisição de dados como: criar, ler, escrever, fechar e limpar o canal, tornando-o bastante prático. Depois de feita a leitura da posição da antena é imprescindível realizar adaptação da leitura do *resolver* para o fim de curso da antena, detalhada na Seção 4.3.1 a seguir. Como foi reduzido o número de canais para a leitura da posição da antena, é preciso realizar a adaptação e o cálculo do sinal de controle em um eixo de cada vez. A Figura 17 resume o fluxograma do subVI Posicionamento.

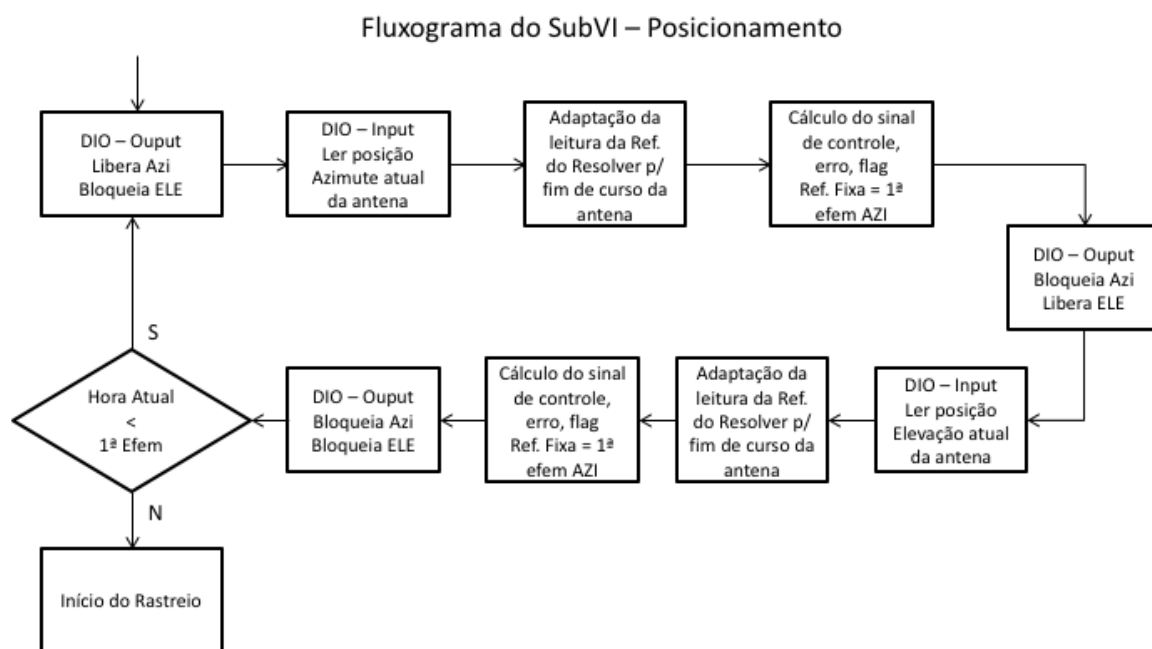


Figura 17 – Fluxograma do subVI – Posicionamento.

No momento que antecede o posicionamento, é constantemente realizada uma comparação entre a hora atual e a hora da primeira efeméride para permitir o início do posicionamento. Uma vez que falta exatamente um minuto para o início da primeira efeméride, o subVI Posicionamento é iniciado. Porém, são necessários alguns segundos para o acionamento por completo do sistema e por isso é garantido quatro segundos para esperar o sistema ligar.

A Figura 18 mostra o diagrama de blocos implementado no LabVIEW responsável pelo acionamento do subVI Posicionamento e por ligar a gaveta de potência. No momento que antecede o início do Posicionamento, foi usado o *DAQ Assistant* para configurar duas portas da placa de aquisição de dados como saídas digitais com a finalidade de ligar a gaveta de potência e ligar o LED *Puissance* localizado na gaveta de comando.

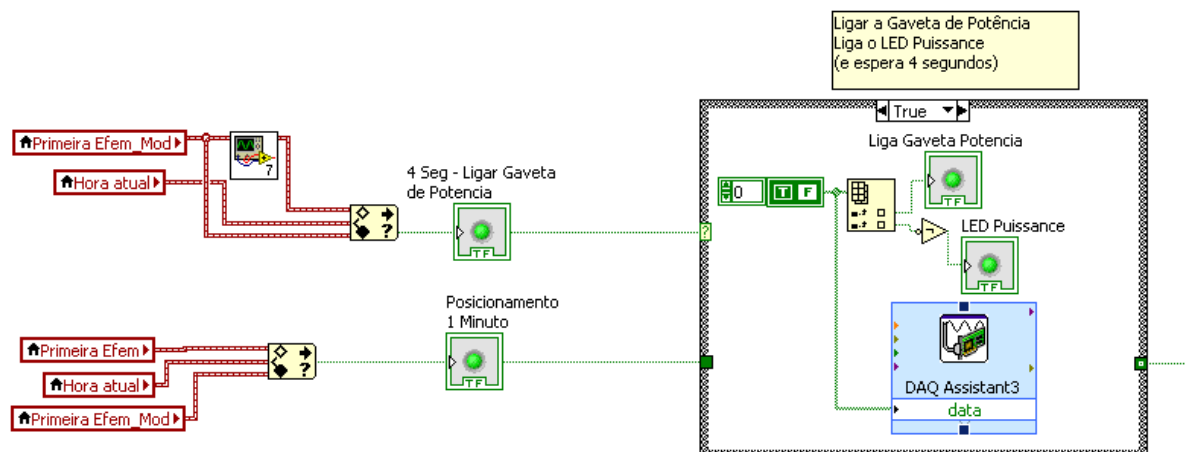


Figura 18 – Diagrama de blocos responsável pelo acionamento da gaveta de potência e do sistema de posicionamento.

No Posicionamento são realizados os cálculos do sinal de controle com base no erro de posição. A fim de coincidir o valor adotado no projeto do controlador é necessário garantir que o processo de liberar ou bloquear a leitura azimuth, ler a posição atual, fazer a adaptação do ângulo, realizar os cálculos do controlador e o mesmo para o eixo elevação sejam feitos a cada um segundo. Para isso, foi utilizado o bloco chamado *Timed Loop* que permite definir o tempo no qual cada interação é realizada.

Como a leitura dos eixos da posição atual da antena não podem ser feitas ao mesmo tempo, é necessário realizar a leitura sequencial dos eixos. Entretanto, o LabVIEW não possui uma sequência de execução de código bem definida, e por isso foi necessário utilizar o bloco *Stacked Sequence Structure* para garantir a sequencialidade da execução. Com esse bloco é possível criar abas no qual o LabVIEW garante a execução do que está inserido em cada aba de forma sequencial. O bloco é usado para garantir a leitura da posição atual do eixo azimuth e posteriormente o da elevação. A Figura 19 apresenta o subVI Posicionamento implementado no LabVIEW.

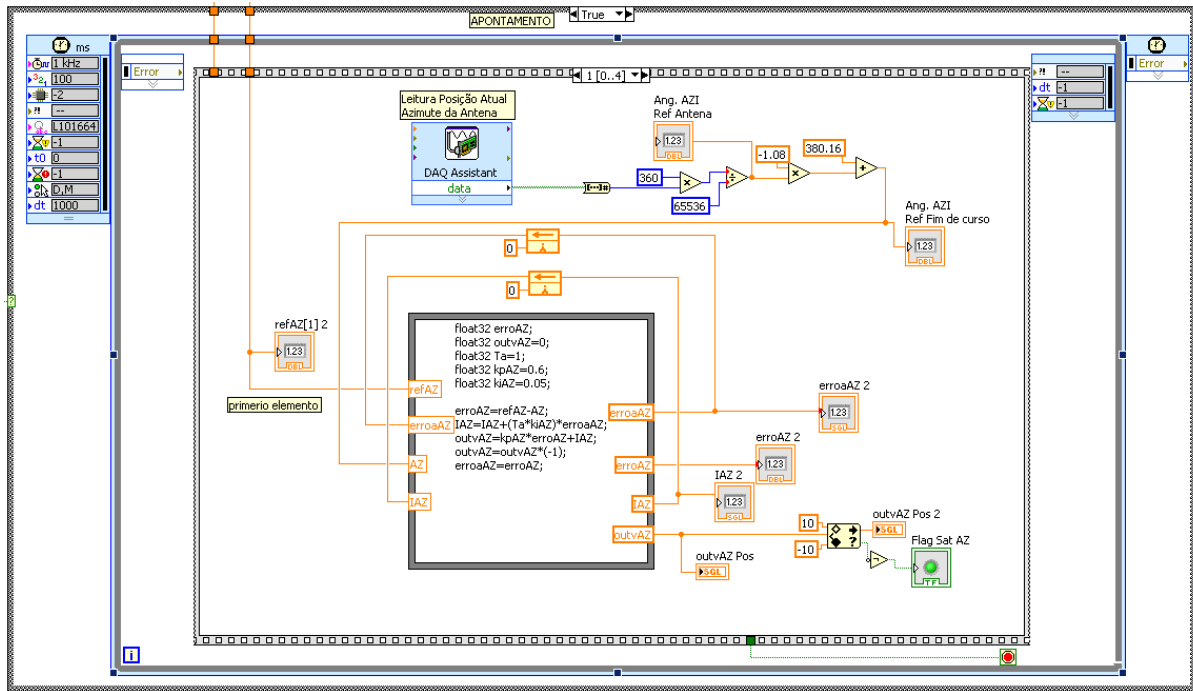


Figura 19 – Diagrama de blocos do subVI Apontamento no LabVIEW.

4.3.1. Adaptação da leitura do resolver para o fim de curso da antena

A adaptação da leitura do *resolver* para o fim de curso da antena é necessária, pois quando os *resolvers* foram instalados no eixo da antena, eles não foram colocados no zero do sistema, ou seja, no fim de curso da antena. Portanto, é necessário que o LabVIEW realize essa conversão de maneira a obter o valor em relação ao fim de curso da antena. Com base nos valores extremos obtidos pelos *resolvers* para cada eixo foi possível encontrar a equivalência a partir de uma regra de três composta.

Para converter os valores fornecidos pela placa SOTEREM 2266-1 (res_{AZ} e res_{ELE}) em valores angulares ($angRes_{AZ}$ e $angRes_{ELE}$) em graus, deve seguir as seguintes equações:

$$angRes_{AZ} = res_{AZ} \times \frac{360}{65536} \quad (4)$$

$$angRes_{ELE} \begin{cases} res_{ELE} \times \frac{360}{65536} & se \quad res_{ELE} \times \frac{360}{65536} < 230^\circ \\ res_{ELE} \times \frac{360}{65536} - 360^\circ, & senão \end{cases} \quad (5)$$

A conversão da equação 5 considera que os valores extremos do eixo elevação se encontram no terceiro quadrante e por isso foi necessário deslocar a descontinuidade para o terceiro quadrante. Isso ocorre, pois no possível trajeto da elevação ocorre uma descontinuidade no zero grau. A Figura 20 demonstra a equivalência entre as escalas.

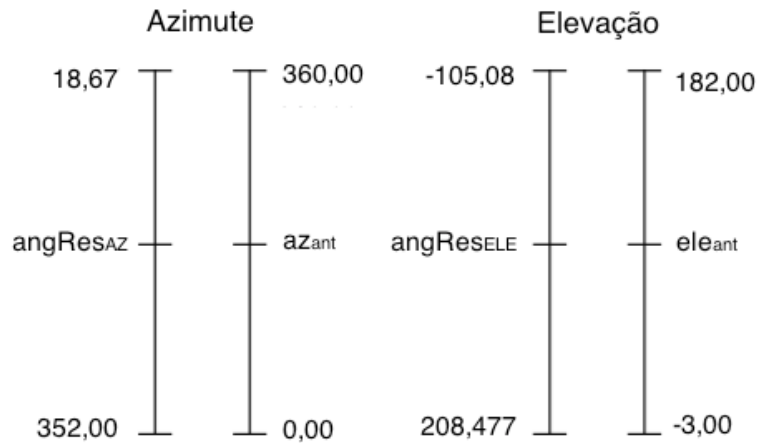


Figura 20 – Conversão entre as escalas.

$$az_{ant} = -1,08 \times angRes_{AZ} + 380,16 \quad (6)$$

$$ele_{ant} = -0,59 \times angRes_{ELE} + 120 \quad (7)$$

A partir dos valores extremos fornecidos pelos *resolvers* é possível observar que um giro completo da antena em ambos os eixos corresponde a menos de um giro completo do *resolver*. Isso é importante, pois mesmo que perdendo um pouco de resolução evita-se uma ambiguidade na leitura da posição da antena.

4.4.SubVI - Rastreo

A principal diferença entre o momento do Rastreo e do Posicionamento está relacionado ao fato que no rastreo as entradas dos ângulos azimute e elevação não são fixas como no Posicionamento. Para o momento do Rastreo é necessário realizar a análise se a hora atual é maior que a hora da efeméride, caso afirmativo indica que a passagem já ocorreu e a referência deve ser atualizada.

Fluxograma do SubVI – Rastreo

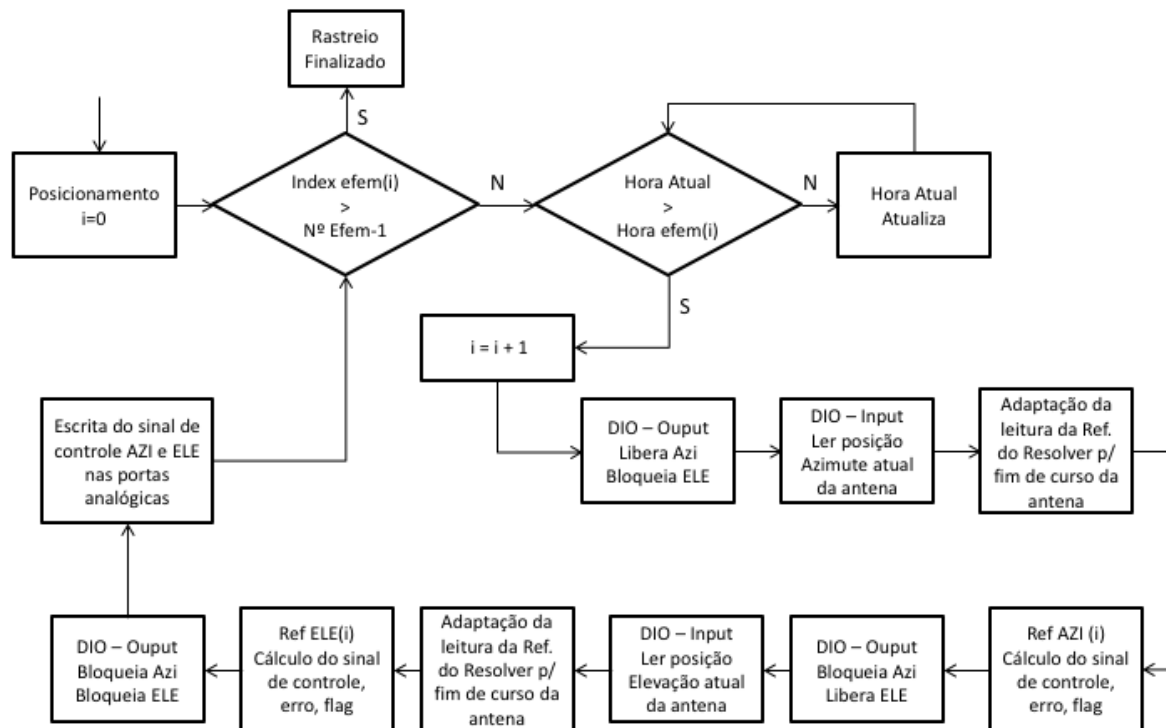


Figura 21 – Fluxograma do SubVI – Rastreo.

Como as referências não são fixas é necessário ter conhecimento constantemente em qual efeméride deverá ser apontada em cada instante. Para resolução desse problema no LabVIEW, foram utilizados *Shift Registers* que são capazes de passar valores entre as interações do sistema. Nesse contexto, foi feita uma implementação no LabVIEW de um sistema capaz de identificar qual efeméride deve ser considerada como referência para o cálculo do sinal de controle no rastreo. Foi criado um vetor no formato *TimeStamp* das efemérides e utilizado um bloco chamado *Index Array* responsável por retornar apenas o elemento do vetor referente ao index selecionado. O *Shift Register* é utilizado para repassar para as próximas interações, o valor do *index* que aponta para o vetor de efemérides no formato *TimeStamp*. Dessa forma, o bloco *Shift Register* garante que enquanto a hora atual é menor que a hora de determinada efeméride, o index é mantido o mesmo. Caso contrário, o index é incrementado em um, visto que a efeméride já passou e exige outros valores de referência para que a antena siga o satélite. O diagrama de blocos do LabVIEW que altera o index do rastreo é mostrado na Figura 22. O sinal lógico booleano (cor verde), ao lado direito da Figura 22 é responsável pelo acionamento do sistema de rastreo. Dessa maneira, apenas no momento em que a hora atual ultrapassa o horário da efeméride é acionado o sistema de rastreo.

Assistant para realizar a leitura da posição da antena e depois fazer à alteração do referencial do *resolver* para o fim de curso da antena. O bloco *Formula Node* foi escolhido para implementar o controlador digital PI responsável pelo sinal de controle enviado ao inversor.

4.5.SubVI – Status do Sistema

Foi implementado um subVI responsável por fornecer o status do sistema. Este é obtido a partir das condições lógicas de certas entradas digitais e de algumas bobinas localizadas no interior do Módulo de Comando que são transmitidas para o Módulo de Controle. De acordo com o status das bobinas é possível saber se o fim de curso ou botão de emergência foi acionado. Já a partir das entradas digitais saber se a Chave Manuel foi acionada e se o inversor azimuth ou elevação estão com defeito. O diagrama da Figura 24 resume as condições analisadas e as decisões a serem tomadas.

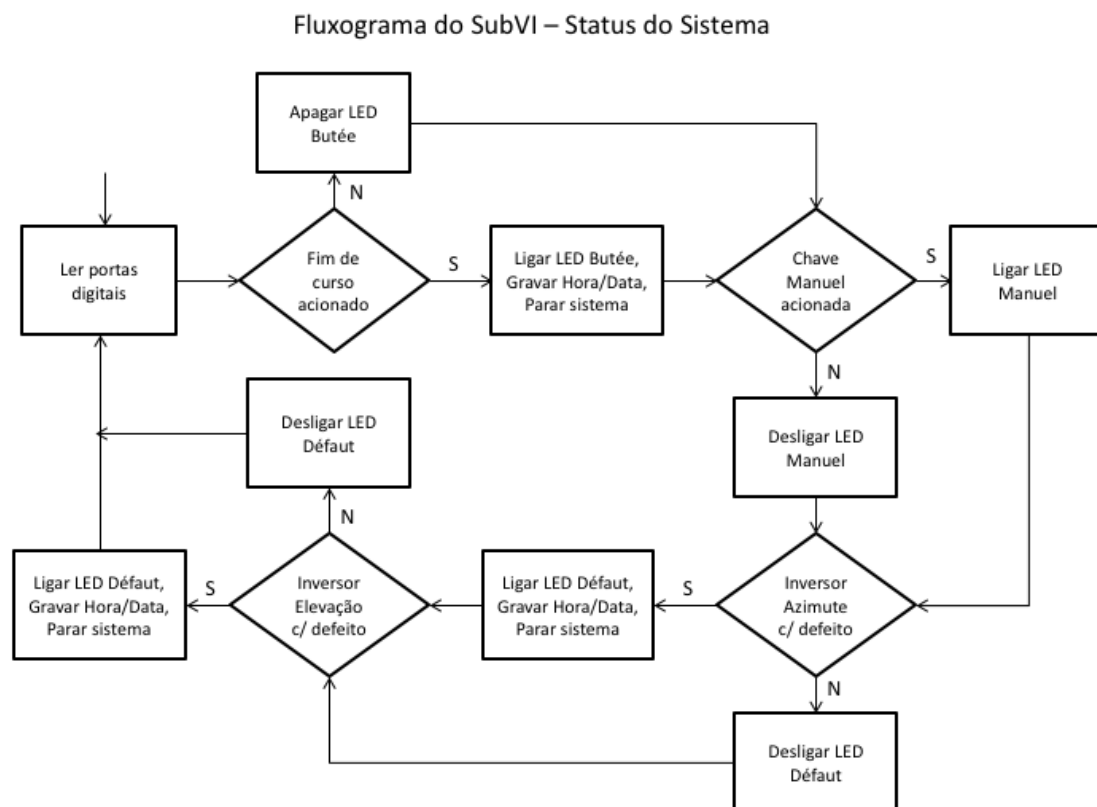


Figura 24 – Fluxograma simplificado do subVI – Status do Sistema.

Após a leitura das portas digitais feita pelo bloco *DAQ Assistant*, é usado o bloco *Index Array* com a finalidade de analisar separadamente o valor de cada porta digital. Para todas as portas analisadas o nível lógico zero indica o funcionamento normal e nível lógico alto indica que o referido evento, relacionado à porta em análise, ocorreu. Para o acionamento

de qualquer LED, presente no painel frontal da gaveta de comando, é imprescindível configurar a saída da porta digital em nível lógico baixo.

O primeiro teste corresponde verificar o status da bobina KA2. Quando ela se encontra desenergizada, é devido a uma das seguintes condições: fim de curso acionado, contato da placa SOTEREM em aberto ou bobina KA1 desenergizada (ver Figura 10). Dessa maneira para concluir que o fim de curso foi acionado, basta analisar o status da bobina KA1. Esta quando energizada indica que o botão de emergência não foi acionado. Assim, na condição da bobina KA2 desenergizada e KA1 energizada, é concluído que o fim de curso foi acionado. Para fins de notificação ao usuário, deve-se configurar a saída digital para ligar o LED *Butée*.

Outra análise foi verificar o status da Chave Manuel. A chave Manuel indica se a gaveta de comando está operando em modo manual ou automático. O subVI implementado, apenas repassa o status referente a Chave Manuel para a saída digital referente ao LED Chave Manuel. Entretanto, a função manual foi removida e, portanto o sistema é operado apenas pelo LabVIEW.

Por fim, é feita a análise se há algum defeito nos inversores azimuth e elevação. É verificado se: as entradas digitais referentes ao defeito azimuth ou ao defeito elevação estão em nível lógico alto e se as bobinas KA1 e KA2 estão energizadas. Nesse caso, pode-se concluir que há defeito no inversor e o usuário é notificado através do LED *Défaut*.

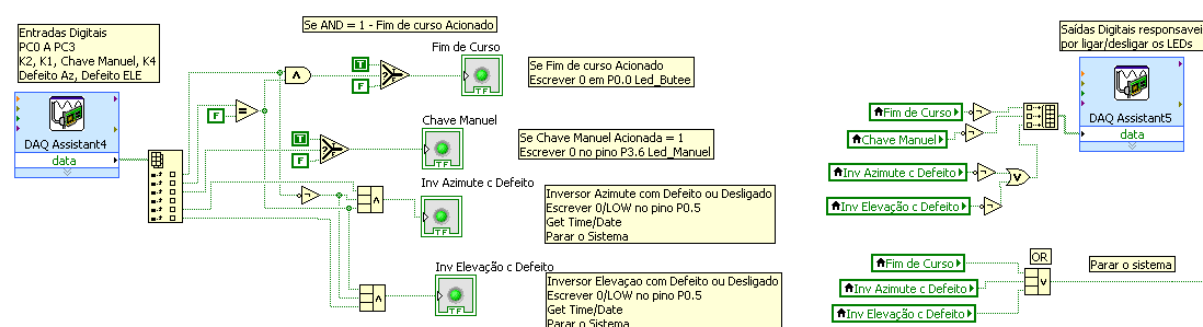


Figura 25 – Diagrama de blocos LabVIEW do subVI Status do Sistema.

No canto superior direito da Figura 25, é feito o acionamento dos LEDs com o uso do *DAQ Assistant* para configurar as portas como saídas digitais. Como os LEDs são acionados em nível lógico baixo, foi necessário utilizar inversores lógicos depois de cada variável do status do sistema. No canto inferior direito da mesma Figura, é utilizado o operador lógico OR com três entradas com a finalidade de parar o sistema para qualquer um dos três eventos que cheguem a ocorrer: fim de curso, inversor azimuth ou elevação com defeito.

5. Conclusões e Recomendações

O trabalho desenvolvido até o momento apresentou diversos avanços com o uso do software LabVIEW na implementação. Os subVIs implementados apresentaram uma boa funcionalidade. Como consiste numa versão em desenvolvimento, o desempenho do sistema não foi o objetivo principal, mas sim o seu funcionamento. Os subVIs que tinham como parâmetro de entrada variáveis como: posição atual da antena, status das bobinas, liberar azimuth/elevação etc não foram testados na prática. Foi sugerido agir com cautela nos momentos em que é necessário a comunicação entre a placa de aquisição de dados e a gaveta de comando da estação. A principal causa dessa precaução é de evitar danos na estação.

A interface entre o programa e a gaveta de comando foi implementada utilizando o *DAQ Assistant* que é em um bloco bastante prático, pois resume todos os procedimentos para aquisição de dados em um só bloco. Entretanto, quando se deseja desempenho é preferível o uso de blocos separados para criar, ler, escrever e limpar o canal etc. Outro bloco que merece atenção é o *Formula Node* que de acordo com usuários experientes do fórum da *National Instrument* deve ser usada em último caso, pois há perda de desempenho no programa. Para os próximos passos do projeto, é sugerido uma implementação alternativa para os blocos *DAQ Assistant* e do *Formula Node*.

Outra sugestão para melhoria seria modificar o subVI Ler Efemérides para que este seja capaz de ler várias passagens, de diferentes satélites, ao longo de um período e organizá-las cronologicamente. No momento o sistema é limitado por apenas uma passagem que tem sua validade testada com base na hora atual que o programa é executado.

6. Referências Bibliográficas

- [1] Orlando, V.; Kuga, H. K.; Capítulo 6 – Rastreo e Controle de Satélites do INPE.
- [2] Manual do LabVIEW – National Instruments.
- [3] QUEIROZ, K. I. P. M. Descrição Funcional do Subsistema de Controle e Rastreo da Estação Multimissão de Natal (EMMN), Instituto Nacional de Pesquisas Espaciais, Natal. 2006.
- [4] YAMAGUIT, W.; ORLANDO, V.; PEREIRA, S.P.; Sistema Brasileiro de Coleta de Dados Ambientais: Status e planos futuros, Instituto Nacional de Pesquisas Espaciais, São José dos Campos. 2006.
- [5] O que é uma PCD? - Instituto Nacional de Pesquisas Espaciais. Disponível em: <http://sinda.crn2.inpe.br/PCD/SITE/novo/site/index.php>. Acesso em 20 de outubro de 2015.
- [6] OGATA, K.: Engenharia de Controle Moderno – 4ª Edição, 2003, Prentice-Hall.