

Automatização do descarregamento de contêineres em navios cargueiros de grande porte e segmentação de posicionamento baseada em cores

Mateus Matzkin Gurza, Thiago Daquino Velasquez, Thiago Rosa Carvalho

Resumo – Este trabalho propõe, no contexto de Sistemas a Eventos Discretos (SED), um método de automação e controle para o descarregamento de navios cargueiros de grande porte, incluindo a segmentação espacial de contêineres de diferentes marcas, aproximada por cores distintas. O processo atual depende fortemente de operadores humanos, o que representa vulnerabilidade devido a fatores como fadiga e desatenção, além de reduzir a eficiência operacional. A automação visa mitigar esses riscos e otimizar o tempo de operação, diminuindo a ociosidade dos navios e o tempo de espera para atracação. O controle foi modelado com 39 sinais, sendo 30 de entrada, seguindo a metodologia da disciplina PMR3305: definição da arquitetura física e modelo estrutural, modelagem comportamental via PFS e Redes de Petri, validação no PIPE 4.3 e conversão para Ladder Diagram (LD) por isomorfismo. Conclui-se que a implementação de sistemas automatizados de descarregamento é viável e pode aumentar a segurança, a organização e a eficiência das operações portuárias em larga escala.

Palavras-chave – Automação; Controle; Navios; Descarregamento; Container

1 Introdução

O presente projeto se propõe a discutir e idealizar um método de automação do descarregamento de navios cargueiros em razão da importância desse procedimento do ponto de vista das redes de transporte de mercadorias. Em um contexto onde a crescente complexidade e escala dos processos logísticos globais criam a necessidade de se tornar cada vez mais eficiente, o desafio de realizar a operação de descarregamento de navios cargueiros em portos comerciais de maneira mais rápida, organizada e segura ganha destaque. Para enfrentar esse desafio, a automação e controle desse processo é uma solução promissora, com o potencial de reduzir vulnerabilidades associadas ao trabalho humano, melhorar a eficiência de tempo e otimizar o uso dos recursos portuários. Atualmente, o processo de descarregamento ainda depende intensivamente da intervenção humana, com operadores controlando o maquinário responsável pela movimentação dos containers. Essa dependência constitui um risco significativo, uma vez que fatores como fadiga, desatenção e erros humanos podem comprometer tanto a segurança quanto a integridade da carga, além de acarretar perdas financeiras consideráveis.

Para a questão do descarregamento de um navio cargueiro não importa apenas a retirada dos contêineres do barco, mas também a forma como esses são dispostos em seguida. Um navio de grande porte costuma transportar containers de diversas marcas, e o descarregamento eficiente desses containers, com sua distribuição em zonas específicas de acordo com as marcas, é essencial para facilitar o transporte e a redistribuição subsequente. No entanto, o processo manual e fragmentado atual torna essa segmentação complexa e demorada, o que por muitas vezes a leva a ser ignorada. A automação do processo completo de descarregamento envolve, então, pensar na organização de como são depositados os containers no porto. Portanto, automatizar integralmente a operação de descarga não só promete acelerar o processo, diminuindo o tempo de ociosidade dos navios no porto e o tempo de espera para a atracação de outros, mas também favorece uma distribuição inteligente dos containers, o que beneficia diretamente as redes de transporte e as empresas proprietárias das mercadorias.

Pensando na modelagem do sistema proposto, simplificações serão utilizadas para tornar o problema mais tangível. Por exemplo, a diferenciação recém-comentada entre marcas de contêineres será abordada por meio de um esquema de cores. Assumiu-se que cada marca de container é identificada por uma cor distinta (ex.: vermelho, verde, azul e amarelo), o que facilita a implementação de sensores para a identificação das marcas. Além disso, o descarregamento de um navio normalmente é realizado por vários guindastes, cada um responsável por uma seção do navio. Contudo, podemos assumir que o método de controle aplicado para cada seção é o mesmo, e assim, com alguns pequenos ajustes explicados ao longo deste documento, modulariza-se o projeto de automação, permitindo detalhar melhor o desenvolvimento do sistema para a descarga automatizada de uma seção, abordando de forma prática e eficiente as complexidades desse processo.

2 Revisão bibliográfica e metodologia

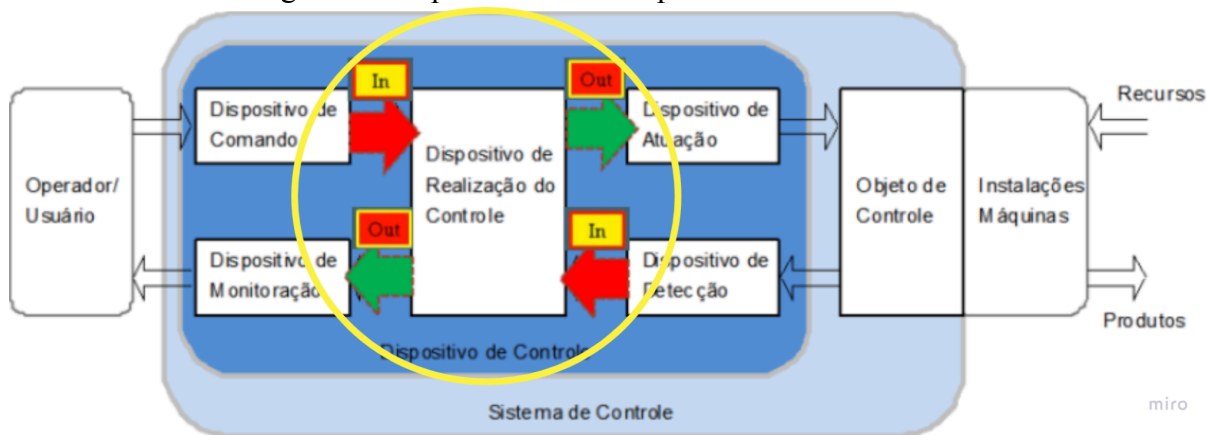
A modelagem do problema abordado neste trabalho exigiu o conhecimento de técnicas associadas à discretização de problemas de engenharia do mundo real usando a teoria de Sistemas a Eventos Discretos (SED). Esse campo trata da utilização de um espaço de estados discretos lógicos para lidar com a modelagem e entendimento de processos cuja dinâmica pode ser associada à ocorrência de eventos pontuais. Assim, a modelagem em SED permite estruturar um processo de forma independente do tempo, e dependente apenas das próprias características atuais do sistema.

Com essa abordagem, é possível utilizar-se de controladores lógico-programáveis (CLPs) para realizar o controle e automação de processos a partir de sensores e componentes que indicam estados do sistema. A elaboração dessa configuração física é essencial para consolidar o modelo estrutural de um problema de engenharia. Nessa frente, é importante definir quais são as entradas e saídas necessárias para entender e controlar um processo sendo modelado, ou seja, é necessário entender quais variáveis são responsáveis por descrever completamente cada estado desse processo e quais comandos são necessários para ser capaz de realizar todas as atividades previstas. Com essa informação em mente, é possível indicar o I/O do controlador, ou seja, os sinais que o controlador lógico-programável vai receber e os sinais que ele irá emitir. Os sinais recebidos serão vindos de sensores e os emitidos irão ativar ou desativar componentes como motores e válvulas, entre outros. A dinâmica de transições entre atividades causadas por sinais enviados pelo CLP e mudanças de estado causadas por sinais recebidos pelo CLP é o que chama-se de controle de um Sistema a Eventos Discretos.

Abaixo segue um esquema simplificado do controlador lógico-programável (CLP) cuja função é comunicar para o sistema a lógica de transições e atividades durante a execução do processo. Esse componente é a peça chave para possibilitar o controle e a automação do processo de descarregamento em questão. Circulado em amarelo está o foco desta imagem, as entradas e saídas do controlador, por meio das quais toda a lógica é estruturada. Veja a seguir uma breve explicação das entradas (setas em vermelho) e saídas (setas em verde) para um controlador:

- Entradas providas do Dispositivo de Comando são sinais recebidos pelo controlador, capazes de ativar transições, e enviados por um agente externo ao sistema, normalmente o operador responsável pelo processo;
- Entradas providas do Dispositivo de Detecção são sinais, recebidos pelo controlador, cujo papel é ativar transições em função de estados do sistema detectados automaticamente por sensores;
- Saídas enviadas para os dispositivos de monitoração são sinais que ativam componentes indicadores de algum estado do sistema, como por exemplo Leds que indicam que alguma máquina está em operação. Estes sinais são importantes para os operadores responsáveis pelo andamento do processo.
- Saídas enviadas para os dispositivos de atuação são os sinais responsáveis por controlar as atividades sendo realizadas pelas instalações e máquinas do processo

Figura 2 - Arquitetura de CLP para o modelo estrutural



Fonte: Autoria própria.

Observe que, para um problema de alta complexidade, é difícil definir o processo e entender como controlá-lo sem ter sua estrutura claramente disposta, e suas dependências explícitas. Uma forma de fazer isso é por meio da máquina de estados, no entanto, esse método se torna progressivamente menos útil e claro conforme adicionamos atividades que devem ser realizadas em paralelo. A máquina de estados também falha em fornecer uma visão explícita do processo modelado, pois é uma representação visual pouco inteligível. Como alternativa, com o objetivo de modelar atividades e transições discretas, podemos usar as Redes de Petri, inventadas por Carl Petri. Essas redes, se encaixam na metodologia de projeto utilizada nas aulas de PMR 3305, a qual envolve um procedimento de ponta-a-ponta para construir o sistema de controle de um processo sendo descrito. Neste trabalho, essa será a metodologia utilizada. As principais etapas da metodologia são:

2.1 PFS (Production Flow Schema)

Utiliza-se desta ferramenta para identificar as atividades que compõem um processo maior sendo modelado e organizá-las para permitir a posterior tradução para uma Rede de Petri RdP organizada. Aqui são identificadas etapas concorrentes, etapas paralelas, e a ordem das etapas. São representados, também, os buffers de transição entre essas etapas e processos. O resultado disso é a organização lógica do sistema modelado e suas atividades.

2.2 RdP (Rede de Petri):

A rede de Petri é extraída a partir da última etapa do PFS, com ela representam-se todos os lugares, indicativos de estados onde sinais podem estar sendo enviados, e transições, que marcam a dinâmica do processo e passagem de um estado para outro. Além disso, aqui aplicamos as marcações que representarão a passagem de algum objeto do processo pelas suas diferentes etapas. É importante ressaltar que durante esta parte da metodologia são definidos e representados os intertravamentos que garantem o pleno funcionamento do modelo, sendo eles intertravamentos de partida, referentes às condições de inicialização do processo; de funcionamento, referentes à garantia de que o processo está obedecendo uma série de condições de funcionamento essenciais; e de processo, referentes à própria produção da RdP.

2.3 Norma IEC 61131-3 para programação

A norma IEC 61131-3 padroniza a programação de CLPs, fornecendo diretrizes que facilitam o desenvolvimento e manutenção de sistemas de controle industrial. Ela permite estruturar tarefas em

blocos funcionais, introduz variáveis globais e locais para maior organização e define linguagens padronizadas, como Ladder Diagram (LD) e Sequential Function Chart (SFC), as quais foram utilizadas durante o andamento da disciplina PMR 3305, entre outras, amplamente usadas em automação. Neste artigo, a linguagem "Ladder Diagram" foi a utilizada para o desenvolvimento do controle do CLP. Sua elaboração se dá por uma representação gráfica baseada em diagramas de relés, amplamente usada em automação industrial, e será aprofundada mais adiante.

Linguagens como essa simplificam a tradução de modelos como Redes de Petri (RdP) para programas executáveis em CLPs, alinhando-se ao processo metodológico descrito até aqui. A padronização, exigida pela norma, reduz erros, melhora a documentação e garante a integração eficiente de hardware e software no controle de sistemas a eventos discretos. Assim, a IEC 61131-3 conecta a modelagem teórica ao controle prático, viabilizando a automação descrita no trabalho.

2.4 Transcrição RdP para linguagem de programação

O objetivo desta etapa é converter a Rede de Petri em um formato que possa ser implementado por CLPs. Para isso, os elementos da RdP, como Lugares, Transições, Entradas e Saídas (Inputs e Outputs), são nomeados como variáveis do sistema (por exemplo, L1, L2, L3; T1, T2, T3; I1, I2, I3; Q1, Q2, Q3). Essas variáveis podem assumir os valores binários 0 ou 1 e são registradas pelo processador em uma tabela de entradas.

Com essas definições estabelecidas, a aplicação do Ladder segue uma sequência específica de passos que devem ser rigorosamente respeitados, já que a lógica Ladder é processada por varredura, de cima para baixo e da esquerda para a direita. O acionamento das saídas (outputs) depende diretamente da ativação dos lugares, que, por sua vez, é condicionado ao disparo das transições. Este encadeamento garante o funcionamento correto e sincronizado do sistema automatizado.

Seguem os passos a serem seguidos:

A Implementação das Transições

Para cada uma das transições, são mapeadas as condições de entrada dessa transição. O Ladder então é montado de forma que, para a transição ser ativada ($T_x = 1$), todas as condições de entrada devem ser satisfeitas (Uso do AND lógico entre as variáveis de entrada).

B Implementação dos Lugares

Aqui, são definidos as instruções "SETS" e "RESETS" de cada um dos lugares da rede, que indicam que, caso todas as condições de entrada "SET" sejam cumpridas em uma varredura, a saída permanece ligada mesmo se as condições de entrada deixem de ser satisfeitas nas próximas varreduras, permanecendo ligada até que as condições de entrada "RESET" sejam cumpridas. Os "SETS" definem todos os elementos de entrada que precisam ser satisfeitos para o lugar poder receber uma marcação. Já os "RESETS" definem todos os elementos que precisam ser satisfeitos para que a marcação seja retirada do lugar.

C Implementação dos Outputs

Para cada um dos outputs, são definidos quais são os lugares que fazem com que esse output esteja ativo (OU lógico entre as variáveis dos lugares).

Dada essa representação em Ladder, os CLPs são capazes de diretamente interpretar essa programação em Ladder para executar a automação desejada, no qual é feita a leitura do Ladder de cima para baixo, da esquerda para direita, e em varreduras, e é por esse motivo que a ordem dos processos anteriores é relevante. Sendo assim, esse é um processo metódico que sempre é capaz de gerar um código para CLP a partir de uma RdP.

2.5 Visão geral da metodologia

Em suma, a metodologia apresentada fornece um caminho estruturado para a modelagem, automação e controle de variados processos industriais. A metodologia engloba desde a concepção inicial do sistema, até a sua implementação em CLPs. A abordagem em Redes de Petri, consolidada por Cardoso e Valette (1997) mostra-se uma referência robusta, capaz de lidar com processos discretos complexos. A aplicação da norma IEC 61131-3 e o uso da linguagem Ladder reforçam a aplicabilidade prática desses conceitos, permitindo a implementação direta em controladores usados na indústria.

3 Aplicação de Automação Mecatrônica

Para idealizar o método de controle para a automação do processo de descarregamento de um navio cargueiro deve-se pensar de antemão na dificuldade logística do processo considerando sua escala e questões de segurança. O planejamento de uma infraestrutura capaz de comportar as exigências complexas que a automação deste processo demanda é uma etapa essencial do processo de modelagem do problema. Assim, seguem algumas das questões a serem consideradas no projeto de controle do descarregamento de um navio cargueiro e segmentação de contêineres por cores (representativas das marcas):

- Embarcações de grande porte utilizam um sistema de guindastes para retirar os contêineres por seções do barco, como pode ser visto na Figura 1, o que torna o processo mais eficiente. Por outro lado, para a automação de um procedimento assim, se fosse utilizado um sistema de guindastes com braços capazes de realizar movimentos rotativos ou exceder de alguma forma os limites espaciais da sua seção, de tal maneira que fosse possível ocorrer um choque entre duas garras, a modelagem da Rede de Petri deveria levar em conta essa possibilidade. Assim, seria necessário modelar um tipo de célula na rede, cuja ocupação por uma garra impediria a movimentação de outra, e, portanto, não ocorreriam choques. No entanto, uma maneira mais interessante de contornar esse problema mantendo a eficiência é criar uma infraestrutura física na qual não seja possível uma garra exceder os limites espaciais da seção pela qual é responsável. Assim foi idealizado o sistema de trilhos deste trabalho, baseando-o em sistemas de trilhos com garras já existentes mas expandido seu funcionamento para dimensões maiores e procedimentos mais robustos.
- Modelar a diferenciação de containers pelas marcas que aparecem neles é um problema complexo mas que pode ser resolvido por visão computacional e premissas e métodos muito similares aos utilizados no projeto apresentado aqui. Assim, ao representar a diferenciação de marcas por meio da diferenciação de cores, tem-se como objetivo fazer com que o problema se torne mais tangível e compreensível.
- O processo de descarregamento pode ser considerado um subprocesso dentro da etapa de parada de um navio cargueiro em um porto. Em vista disso, entende-se que existem dois processos auxiliares ao principal sendo modelado neste trabalho, são eles o processo de atracamento e o de desatracamento. No processo de atracamento temos a chegada do barco ao porto, a conexão dos equipamentos de descarga e a inspeção de segurança, essas 3 atividades são condições para o início do processo de descarregamento, que é o principal sendo modelado, cujo fim é sucedido pelo processo de desatracamento, composto de uma etapa de desconexão e uma de saída da embarcação do porto. Os processos auxiliares não terão aprofundamento neste trabalho.

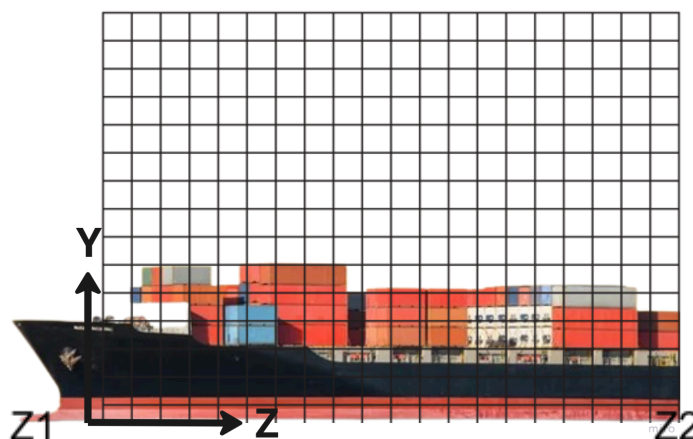
- O descarregamento ideal de um navio cargueiro deve ocorrer de tal forma que a estrutura resultante da retirada gradual de containers seja a mais estável possível. Buscar essa estabilidade é uma tarefa essencial no processo de design do controle ao passo que, sem ela, a queda de containers se torna um risco grave. Para assegurar que a estrutura esteja sempre estável, o ideal é remover os containers em camadas, da mais alta para a mais baixa, terminando sempre a camada superior completa antes de seguir para a próxima inferior.

Sendo assim, o processo de automação mecatrônica proposto para o descarregamento de navios cargueiros de grande porte utiliza uma abordagem modular e sequencial para garantir eficiência e precisão. A seguir, detalha-se o método e os recursos empregados.

3.1 Descrição do sistema que será automatizado (modelo esquemático)

Inicialmente, o sistema considera a visão do perfil do navio cargueiro do ponto de vista dos trilhos responsáveis por movimentar a garra que descarrega os containers. Cada garra opera em uma seção específica, delimitada no eixo Z (entre Z1 à esquerda e Z2 à direita), e seu trilho está equipado com sistemas de visão computacional que identificam a posição e a altura do container mais alto na seção e com coordenada Z mais próxima de Z1. Essa identificação inicial permite calcular as coordenadas Y e Z do container alvo a ser descarregado. A Figura 3 mostra um exemplo da visão lateral do navio que permitiria a localização das coordenadas Y e Z do container mais alto.

Figura 3 - Vista lateral da seção de navio pela qual uma garra seria responsável



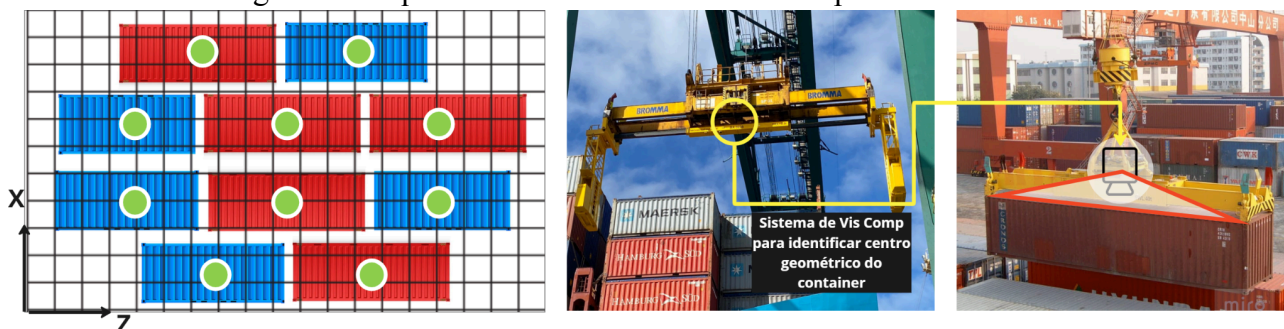
Fonte: Autoria própria.

Uma vez definidas a posição Z e a altura Y, o sistema move a garra automaticamente ao longo do eixo Z, utilizando um deslocamento controlado por tempo e velocidade constante. Ao atingir a posição correta na coordenada Z, a garra para e, assim, a movimentação no eixo X pode ser iniciada - sendo X o eixo com sentido entrando no papel na Figura 3. Vale ressaltar que essa movimentação é possibilitada pelo fato de que o sistema de trilhos imaginado passa por cima do navio cargueiro, permitindo o posicionamento da garra diretamente sobre os containers.

Inicialmente posicionada no ponto mais alto do trilho, a garra avança no eixo X até identificar um contêiner que esteja a uma altura correspondente à altura do contêiner mais alto da pilha identificado anteriormente pelo sistema de visão computacional. Essa identificação é feita por meio de um sensor de altura instalado na garra, garantindo a captura do contêiner na altura desejada. Simultaneamente, um sistema de visão computacional embarcado na garra monitora a posição

relativa entre a garra e o centro geométrico do contêiner, presumido como o centro de massa. A movimentação no eixo X é interrompida somente quando forem atendidas as condições de altura desejada e alinhamento centralizado com o contêiner identificado. Abaixo, na Figura 4, é mostrada a visão da garra, bem como imagens que exemplificam o sistema físico estudado. Leve em consideração que os pontos verdes são apenas indicações da posição com a qual o sistema de visão computacional deve alinhar o centro da garra e não existem no modelo físico real.

Figura 4 - Esquema ilustrativo do método de captura do container



Fonte: Autoria própria.

Note que essa sistematização funciona de maneira a estruturar um processo de descarregamento no qual, em ordem de prioridade, os containers mais altos, mais próximos da origem Z da seção e, por fim, mais próximos da origem em X do barco, são descarregados primeiro. Assim é mantida a estrutura estável da pilha de contêineres no navio cargueiro ao longo do processo gradual de retirada deles.

Após o posicionamento correto da garra sobre o container alvo, o sistema autoriza a descida da garra até que sua base toque o container. Durante a captura, considerando o funcionamento de uma garra Spreader, sensores de pressão verificam se o container está firmemente preso, enquanto sensores ópticos identificam a cor do container, classificando-o entre quatro categorias que representam marcas específicas. Essa classificação permite a segmentação realizada no armazenamento subsequente.

Para facilitar o descarregamento dos contêineres em suas respectivas cores, é assumida uma distribuição dos depósitos que é mostrada nas Figuras 5 e 6. Sendo assim, uma vez capturado o container, a garra retorna até a origem do barco, com a garra em sua posição mais alta e então segue para a origem do devido local de armazenamento para a cor detectada do container, avançando no sentido -X até que chegue nessa origem desejada. Observe que cada seção de descarregamento, em função da modularização do processo, deve ter suas 4 próprias seções de depósito alinhadas.

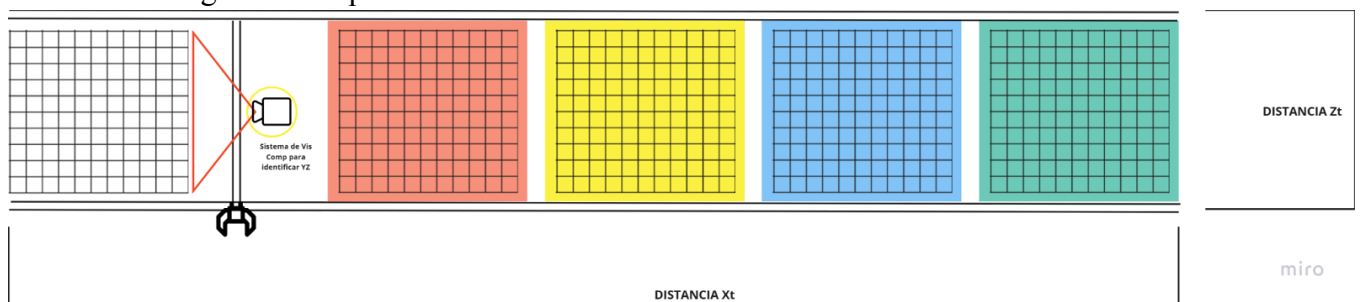
Figura 5 - Esquema ilustrativo da tarefa sendo realizada

Funcionamento por um sistema de trilhos para movimentação em X e Z e descida e subida da garra em Y



Fonte: Autoria própria.

Figura 6 - Esquema ilustrativo da infraestrutura física de trilhos modelada



Fonte: Autoria própria.

Para a alocação no depósito, sistemas externos de visão computacional identificam as posições disponíveis nas áreas designadas para cada cor. A descida do container é monitorada por sensores de peso, que detectam a liberação da carga e encerram a operação no momento correto. O processo garante estabilidade estrutural ao priorizar a construção de bases sólidas antes de avançar para níveis superiores.

Além disso, o método assume premissas simplificadoras, como o alinhamento longitudinal dos contêineres no navio, eliminando a necessidade de movimentos rotacionais da garra e restringindo as operações a movimentos translacionais. Presume-se, também, uma folga entre a garra e o container cuidadosamente dimensionada para compensar variações mínimas no posicionamento e espessura das laterais da garra spreader.

A modularidade do sistema, com trilhos e garras independentes operando em seções específicas, permite escalar a automação para embarcações maiores sem comprometer a precisão ou a segurança. Cada área de depósito possui um sistema de visão computacional próprio para evitar imprecisões decorrentes de um controle centralizado.

O ciclo operacional é repetido até que todos os containers sejam descarregados e alocados nas áreas correspondentes, assegurando uma logística eficiente e segura. Este modelo propõe uma solução robusta para automatizar uma etapa crítica no transporte marítimo, otimizando recursos e minimizando os riscos associados à intervenção humana.

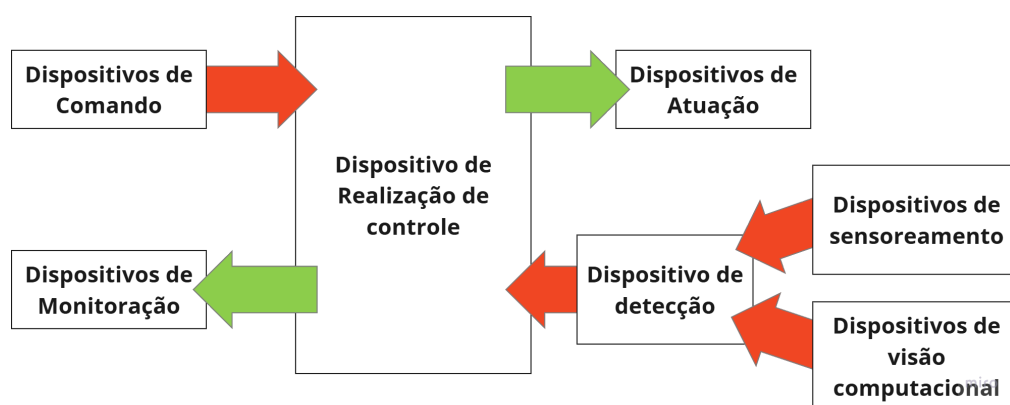
3.2 Modelo estrutural do sistema

O modelo estrutural de controle, como comentado na seção de Revisão Bibliográfica, é uma etapa crucial da elaboração do controle e automação de um sistema. Nesta subseção estará disposta a

arquitetura de controle desenhada para se integrar com a infraestrutura física que possibilita a automatização do processo de descarregamento e segmentação de containers. Para isso, a descrição e posicionamento de componentes de sensoriamento e visão computacional, essenciais para o funcionamento do processo idealizado, será explicitada, possibilitando o entendimento da comunicação entre as diferentes partes do sistema.

Na modelagem deste processo, é importante ressaltar a utilização de dispositivos de visão computacional. Estes componentes são imprescindíveis para múltiplas das atividades a serem realizadas no descarregamento. Por meio deles é que são definidas as transições que permitem o posicionamento correto da garra sobre os containers para sua captura e também dos containers na posição em que serão depositados. Com isso em mente, segue abaixo um esquema de como esses sistemas se integram na arquitetura de CLP exposta na Figura 2.

Figura 7 - Esquema final de dispositivos para o modelo estrutural utilizado



Fonte: Autoria própria.

Com isso definido, pode-se entender os sinais esperados de cada um desses dispositivos. A seguir se encontram uma série de tabelas onde estão descritos os múltiplos sinais de input e output que irão compor o conjunto I/O do modelo estrutural.

Começando pelos outputs a serem enviados para os dispositivos de atuação do sistema, temos, na tabela, as colunas: “# output”, que se refere ao nome do sinal empregado na transcrição do modelo para Ladder; “Sinais”, que indica o nome, propositalmente intuitivo, do sinal sendo emitido; e “Descrição da atuação”, que aborda a atividade resultante da emissão do sinal pelo CLP.

Tabela I – Saídas para dispositivos de atuação

# output	Sinais	Descrição da atuação	Entidade relacionada
Q.01	LowerSpreader	Desce verticalmente a garra spreader	Garra
Q.02	SpreaderClose	Fecha a garra spreader	Garra
Q.03	RaiseSpreader	Levanta verticalmente a garra spreader	Garra
Q.04 Q.05	XForward, XBackward	Movimenta garra em X nos trilhos	Garra
Q.06 Q.07	ZForward, ZBackward	Movimenta garra em Z nos trilhos	Garra

Além dos outputs para atuação, também temos os outputs de monitoramento. A descrição das colunas da tabela pode ser considerada a mesma que a para a tabela anterior. Note que ambos os sinais de monitoramento ativam Leds indicativos de condições específicas do sistema.

Tabela II – Saídas para dispositivos de monitoramento

# output	Sinais	Descrição da atuação
Q.08	LedOcup	Indica que a garra N está carregando um container
Q.09	LedAlarm	Indica que houve um erro ou foi dado o comando de override

Agora, pensando nos sinais de entrada, os inputs, temos uma quantidade maior de informações e estados a serem levados em consideração. A tabela a seguir contém a informação detalhada de como cada input exigido é captado e qual sua função para o sistema. As colunas nas tabelas possuem as seguintes informações.

- Coluna “# input”: se refere ao nome do sinal empregado na transcrição do modelo para Ladder;
- Coluna “Sinais”: indica o nome, propositalmente intuitivo, do sinal sendo emitido;
- Coluna “Descrição da atuação”: descreve de onde vem o sinal, o método de funcionamento para a emissão do sinal e o papel desse sinal recebido para o controle do processo como um todo;
- Coluna “Componente”: Detalha quais os componentes físicos utilizados para permitir o envio do sinal, indicando que tipo de sensor deve ser utilizado e, caso não seja um sensor, quais os constituintes do dispositivo de detecção que envia esse sinal.
- Coluna “Posição física”: Indica qual o posicionamento idealizado para o componente descrito na coluna ‘Componente’ para que a sua efetividade em realizar a detecção e envio de sinal descritos na coluna ‘Descrição da atuação’ seja máxima. O posicionamento desses componentes da forma correta é crucial para que desempenhem corretamente suas funções.
- Coluna “ID”: Considerando que múltiplos sinais diferentes podem ser enviados pelo mesmo sensor ou sistema de visão computacional, o papel desta coluna é identificar qual sensor ou sistema está realizando o envio do sinal. Dessa forma é possível visualizar a disposição de componentes sabendo todos os sinais pelos quais cada um deles será responsável. Um exemplo disso é o fato de que o mesmo sensor LiDar de ID igual a “LiDar1_Sens” envia os sinais “HeightOK” e “HeightTouchOK” para o CLP em duas circunstâncias diferentes.

Tabela III – Entradas de dispositivos de detecção

# input	Sinais	Descrição da atuação	Componente	Posição física	ID
---------	--------	----------------------	------------	----------------	----

I.01	ZAlignOK	Sistema de visão computacional emite um sinal que diz que o alinhamento está correto no eixo Z (no processo de captura de container)	Câmera e Raspberry Pi com modelo de visão computacional	De frente para a lateral barco com visão em YZ e vista completa dos containers na seção pela qual a garra correspondente é responsável	VisComp1_Sist
I.02	SpreaderCenteredOK	Sistema de visão computacional emite um sinal que diz que o alinhamento está correto no plano XZ (no processo de captura de container)	Câmera e Raspberry Pi com modelo de visão computacional	Centralizado em um compartimento na garra apontando para baixo com vista de XZ com visão completa da seção	VisComp2_Sist
I.03	HeightOK	Sensor de altura acusa distância da garra em relação ao container (LiDAR) para que seja comparado com a altura identificada pelo sistema de visão computacional indicando HeightOK se for compatível	Sensor LiDAR centralizado no teto da garra	Topo da garra spreader	LiDar1_Sens
I.04	HeightTouchOK	Sensor que indica fim de curso na descida da garra	Sensor LiDAR centralizado no teto da garra	Sensor de altura na garra	LiDar1_Sens
I.05	PressOK	Sensor de pressão da garra acusa pressão necessária para garantir acoplamento da garra ao container sem risco de queda e retorna OK se sim	Célula de Carga de Compressão	Laterais da garra spreader	PressCel1_Sens
I.06 I.07 I.08 I.09	ColorGreen, ColorRed, ColorYellow, ColorBlue	Sensor óptico capaz de captar a cor do container	Sensor óptico de cor TCS34725 com LED Integrado para iluminar mesmo quase em contato com a superfície	Topo da garra spreader	Optic1_Sens
I.10	LetGoOK	Sensor de peso capaz de identificar quando há uma redução de carga equivalente a apoiar o container sobre um piso ou outro container	Célula de Carga Tipo Tension Link (Dinamômetro de Tensão)	Na conexão entre o braço e garra	WeightCel1_Sens

I.11	UnloadOK	Sistema de visão computacional emite um sinal que diz que ainda há containers na seção pertinente à garra em questão	Câmera e Raspberry Pi com modelo de visão computacional	De frente para a lateral barco com visão em YZ e vista completa dos containers na seção pela qual a garra correspondente é responsável	VisComp1_Sist
I.12	SpreaderTop	Sensor que indica que o spreader está na posição mais alta	Sensor de posição	No suporte superior da garra que a conecta com os trilhos	Pos1_Sens
I.13 I.14 I.15 I.16	OriginRedOK, OriginBlueOK, OriginYellowOK, OriginGreenOK	Sensor identifica containers na origem de sua respectiva cor	Sensores de posição	No trilho de extensão em X, na posição relativa à origem em X para cada cor	Pos2_Sens, Pos3_Sens, Pos4_Sens, Pos5_Sens
I.17	XSectionOriginOK	Sensor identifica o reposicionamento da garra na posição original (origem barco).	Sensor de posição	No trilho de extensão em X, na posição relativa à origem em X para a seção	Pos6_Sens
I.18	ZOriginOK	Sensor identifica o reposicionamento da garra na origem Z	Sensor de posição	No trilho de extensão em Z, na posição relativa à origem em Z para qualquer sistema de coordenadas	Pos7_Sens
I.19 I.20 I.21 I.22	XRedAlignOK, XBlueAlignOK, XYellowAlignOK, XGreenAlignOK	Sistema de visão computacional emite um sinal que diz que a garra está alinhada em X com a posição livre onde o container será depositado(cada área de cor possui um sistema de monitoramento próprio que emite esse sinal)	Câmera e Raspberry Pi com modelo de visão computacional	Origem de cada cor	VisComp3_Sist
I.23 I.24 I.25 I.26	ZRedAlignOK, ZBlueAlignOK, ZYellowAlignOK, ZGreenAlignOK	Sistema de visão computacional emite um sinal que diz que a garra está alinhada em Z com a posição livre onde o container será depositado(cada área de cor possui um sistema de monitoramento próprio que emite esse	Câmera e Raspberry Pi com modelo de visão computacional	Origem de cada cor	VisComp3_Sist

		sinal)			
I.27	SpreaderOpenOK	Sensor que identifica se a garra está aberta	Sensor de Posição Linear ou chave de limite	No mecanismo da garra de forma a identificar o estado dela	OpenClose1_Sens

Por fim temos a tabela com os sinais de entrada provindos dos dispositivos de comando. Estes sinais apresentam uma função interessante relativa a processos auxiliares ao de descarregamento, principalmente o de atracamento do navio, e também à segurança do sistema. O sinal “InspecOK”, é enviado por meio da confirmação do operador responsável pelo processo de atracamento. Essa confirmação se dá ao apertar um botão de “SET” que indica que o procedimento de inspeção das condições de atracamento, processo que ocorre durante a etapa auxiliar prévia ao descarregamento, foi concluído com êxito e nenhum problema foi encontrado. O sinal “EquipConnectOK” é similar mas indica que a conexão dos equipamentos, necessários para o descarregamento, à embarcação foi realizada com sucesso. Finalmente, o sinal “Override” é um método de parada forçada do sistema como forma de prevenção contra acidentes e correção de problemas no processo sendo controlado.

Tabela IV – Entradas de dispositivos de comando

# input	Sinais	Descrição da atuação
I.28	Override	Para o sistema de imediato
I.29	InspecOK	Indica que a etapa de inspeção no processo anterior foi bem sucedida
I.30	EquipConnectOk	Indica que o processo anterior foi concluído e os equipamentos estão conectados e prontos para realizar a descarga

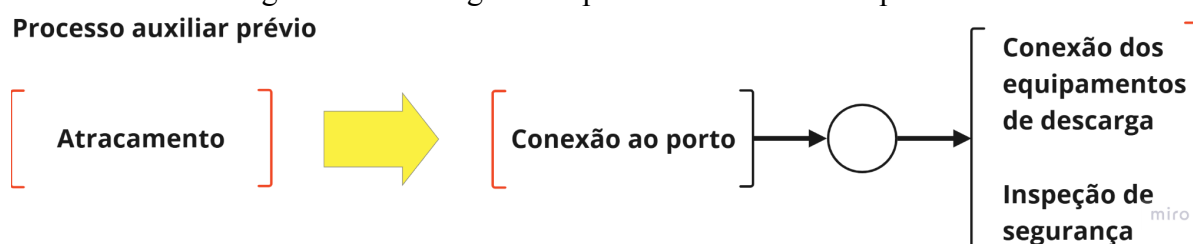
3.3 Aplicação da metodologia de modelagem

Uma vez definido o sistema e seu modelo estrutural, pode-se partir para a aplicação da metodologia de modelagem apresentada anteriormente, passando pelos passos do PFS, RdP e transcrição para Ladder. Cada uma dessas etapas será detalhada a seguir.

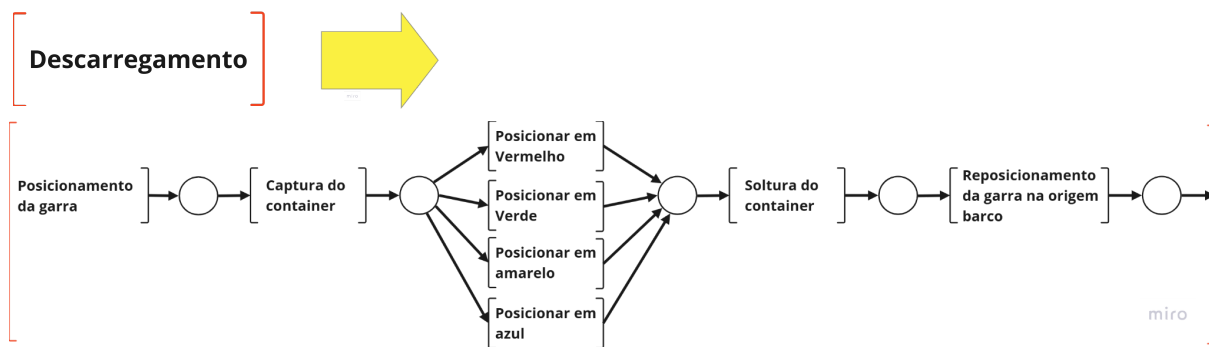
A PFS (Production Flow Schema)

Como citado anteriormente, o sistema analisado pode ser dividido em 3 processos diferentes, sendo eles o atracamento do navio, o seu descarregamento e por último o desatracamento. Cada um desses processos apresenta diferentes atividades associadas. A Figura 8 mostra a representação em PFS de cada um dos 3 processos, bem como o detalhamento de suas funções indicado pela seta amarela.

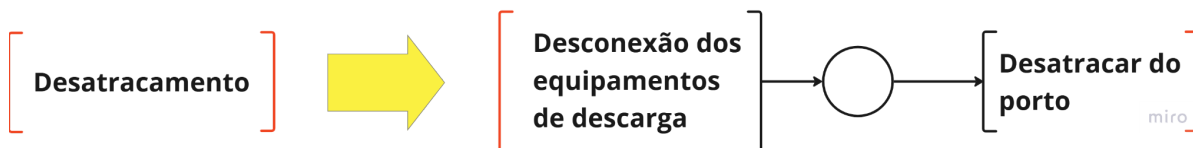
Figura 8 - Modelagem dos processos em PFS - etapas 1 e 2



Processo Principal Modelado



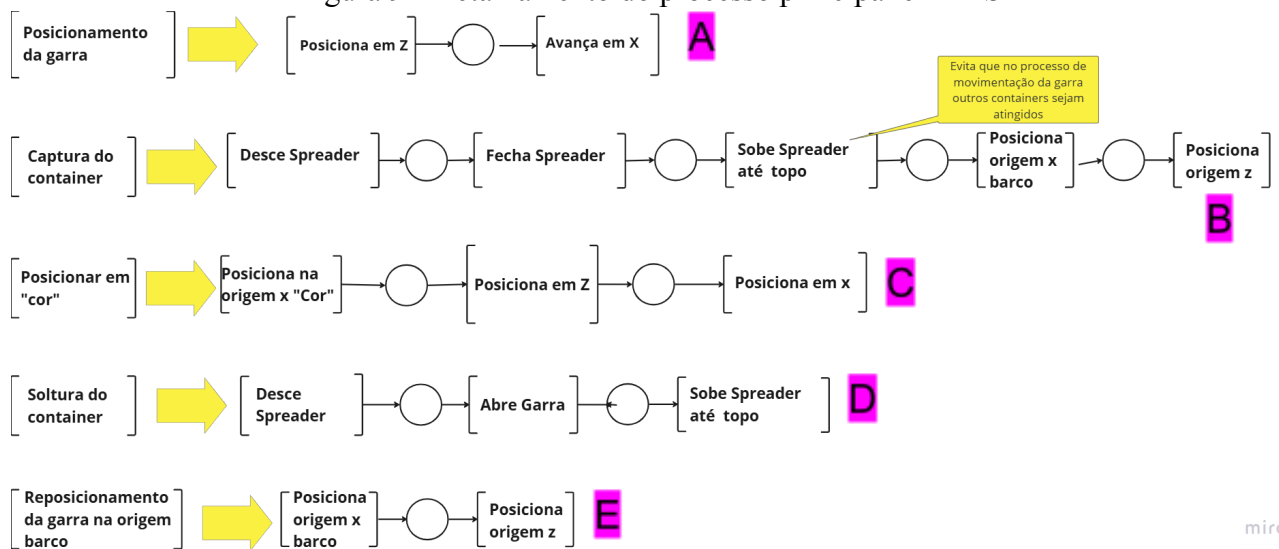
Processo auxiliar posterior



Fonte: Autoria própria.

Dentro do processo principal modelado, podemos detalhar ainda mais as funções separando-as em subatividades, como é mostrado na Figura 9. Essa figura identifica os ramos para facilitar o entendimento posterior da Figura 11.

Figura 9 - Detalhamento do processo principal em PFS



Fonte: Autoria própria.

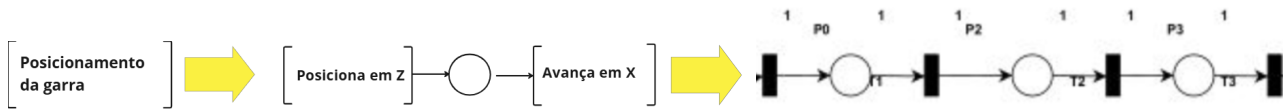
Essa imagem marca o final da etapa de PFS. Nos próximos passos serão apresentadas apenas modelagens referentes ao processo principal de descarregamento.

B RdP (Rede de Petri)

Tendo em mãos o PFS finalizado, é imediata a criação de uma RdP inicial por meio da troca dos

colchetes por transições e das atividades por lugares. Um exemplo dessa introdução de elementos Rdp pode ser visto na Figura 10.

Figura 10 - Passagem de PFS para Rdp



Fonte: Autoria própria.

Com a Rede de Petri construída, é possível integrar os recursos modelados, que neste caso incluem a garra e o comando de fechamento. Embora o comando de fechamento não seja considerado um recurso no modelo físico, ele pode ser tratado dessa forma para garantir o correto funcionamento da rede. Essa abordagem permite representar de forma mais clara e eficaz as etapas do processo em que o comando deve ser emitido, desde a captura do contêiner até sua liberação, facilitando a visualização e o entendimento dessa característica do sistema. A figura abaixo mostra a rede já com os recursos representados e destaca também os ramos do PFS.

Figura 11 - Rdp com recursos



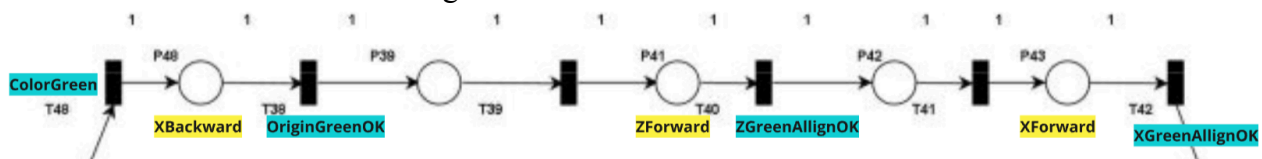
Fonte: Autoria própria.

Após a finalização da rede mostrada na Figura 11, ela é simulada no software PIPE de modo a garantir que as transições e lugares se comportam como esperado, validando o que foi feito até essa etapa. Em seguida, pode-se introduzir os sinais de controle que regem as atividades,

Após a conclusão do Ramo B, a garra encontra-se na posição mais alta e alinhada à origem do barco, pronta para se movimentar em direção a um dos armazéns, dependendo da cor do container. O Ramo C da Figura 14 aborda o caso específico do posicionamento de um container de cor verde, de modo que os sinais de controle utilizam o termo "Green" em seus nomes. O processo para outras cores segue a mesma lógica, alterando-se apenas o nome dos sinais para refletir a respectiva cor.

A atividade tem início com o recuo da garra no eixo X até alcançar a origem associada à cor verde, o que é indicado pelo sinal OriginGreenOK. Em seguida, a garra realiza movimentos nos eixos Z e X até receber os sinais de parada enviados pelo sistema de visão computacional. Esse sistema é responsável por identificar os espaços livres disponíveis em cada armazém, permitindo o posicionamento preciso do container.

Figura 14 - Detalhamento ramo C

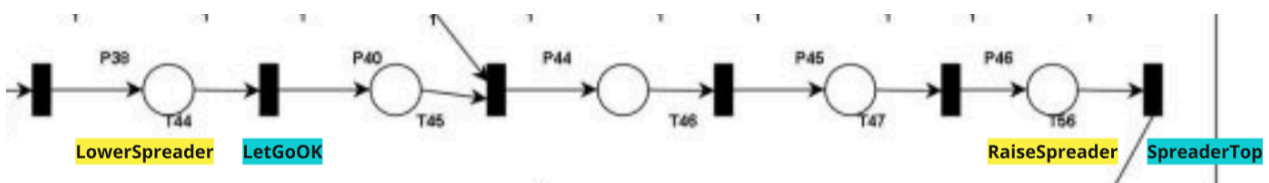


Fonte: Autoria própria.

Com o conjunto garra e container posicionado corretamente, independente da cor, inicia-se a atividade de soltura do container, representada pelo ramo D (Figura 15). Esse processo começa com o abaixamento da garra até o recebimento do sinal LetGoOK, que confirma que o container está apoiado no chão ou sobre outro container.

Na sequência, ocorre a abertura da garra, consumindo o recurso "Fecha Garra", como indicado pela seta que desce para a transição T45. Por fim, a garra recebe o comando de subida, prosseguindo até atingir a posição de topo, marcada pelo sinal SpreaderTop.

Figura 15 - Detalhamento ramo D

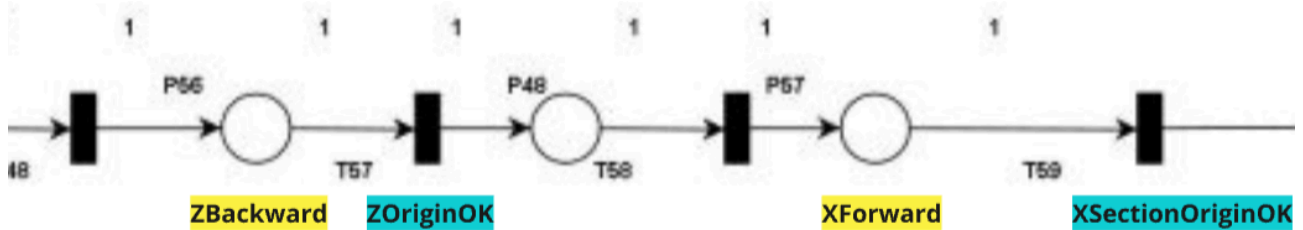


Fonte: Autoria própria.

A última atividade do processo é o reposicionamento da garra para a origem do barco, representada pelo ramo E na Figura 16. Essa etapa é executada em duas fases: primeiro, a garra realiza o recuo no eixo Z até atingir a origem de Z. Em seguida, ocorre o avanço no eixo X até que a garra alcance a origem do barco.

Ao chegar na posição de origem, o recurso "Garra" é liberado, permitindo que o processo seja reiniciado caso as condições iniciais do ramo A sejam atendidas.

Figura 16 - Detalhamento ramo E



Fonte: Autoria própria.

Com a rede completa, pode-se identificar intertravamentos de partida e realizar a criação de intertravamentos de funcionamento.

Para que o sistema possa iniciar o processo, é necessário que os intertravamentos de partida sejam atendidos. Essas condições incluem a confirmação de que o spreader está aberto (SpreaderOpenOK), a verificação de que ainda existem contêineres a serem descarregados (UnloadOK), a confirmação da conexão do equipamento (EquipConnectOK), o resultado satisfatório da inspeção (InspecOK), o alinhamento na origem em X (XSectionOriginOK), o alinhamento na origem em Z (ZOriginOK) e a indicação de que o spreader está na posição mais alta (SpreaderTop).

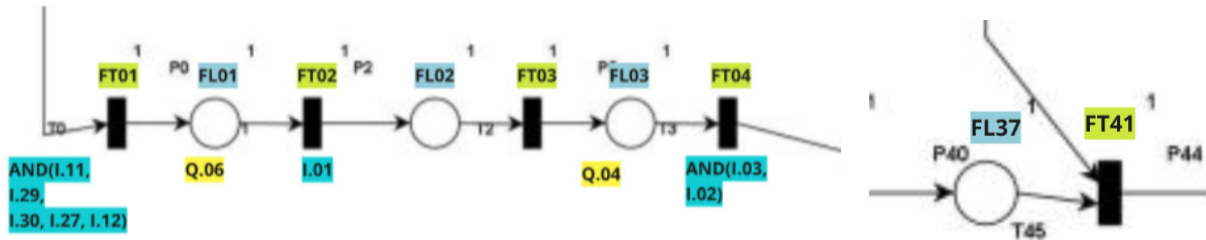
Durante o funcionamento do sistema, os intertravamentos de funcionamento atuam em todas as posições onde ocorrem ações, estabelecendo dois arcos concorrentes. O primeiro arco leva a uma transição acionada manualmente, por meio do botão de comando (override), permitindo intervenções diretas no processo. O segundo arco leva a uma transição acionada automaticamente após o tempo limite, cujo valor é calculado com base na velocidade média de movimentação da garra e nas distâncias totais em cada eixo. O tempo limite no eixo X é definido pela divisão da distância total no eixo X pela velocidade média, no eixo Z pela divisão da distância total no eixo Z pela mesma velocidade, e no eixo Y pela divisão da maior distância no eixo Y pela velocidade média. Além disso, o tempo limite para as operações de fechamento e abertura da garra é fixado em um minuto. Esses intertravamentos asseguram o controle seguro e eficiente do sistema, permitindo tanto operações manuais quanto automáticas de acordo com as condições do processo.

C Transcrição RdP para ladder

O primeiro passo para a transcrição da RdP para Ladder é a definição das variáveis para cada um dos elementos da Rede de Petri. As tabelas III e IV contém a definição das variáveis de Input e Output. A definição das variáveis dos Lugares e Transições das redes foram feitas, e correspondem às variáveis FT1 - FT48, para as transições, e FL1 - FL45, para os lugares.

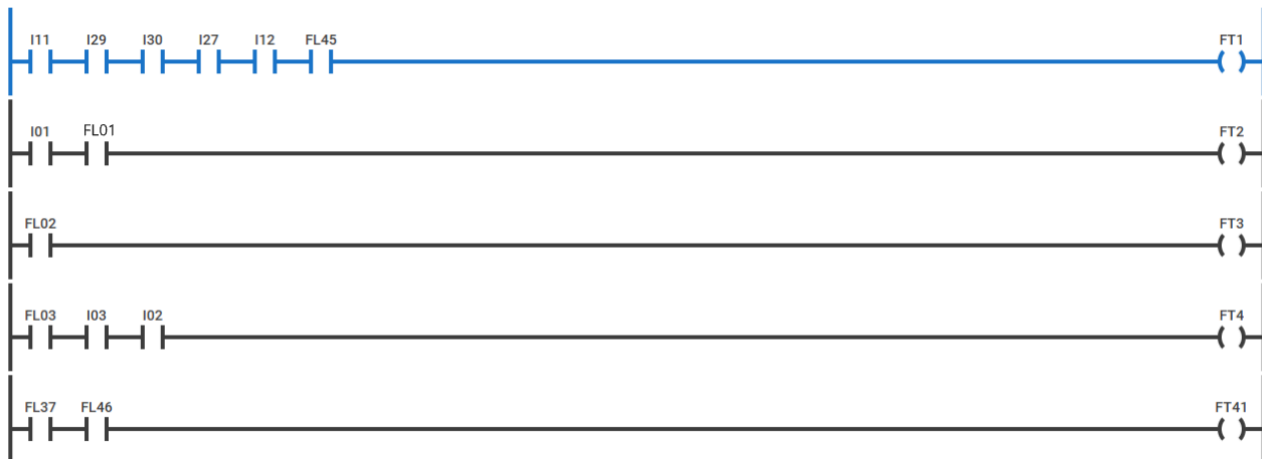
Com isso, o código Ladder pode ser produzido metodicamente. Em primeiro lugar, para cada uma das transições da RdP, foram mapeadas as variáveis de entrada, que precisam estar ativas para que essa transição seja ativada. As imagens a seguir mostram no formato RdP algumas transições de destaque dentre todas as que foram mapeadas, e em seguida observa-se essas mesmas transições em linguagem Ladder.

Figuras 17 e 18 - Transições em destaque na RdP



Fonte: Autoria própria.

Figura 19 - Transcrição das transições destacadas para o ladder

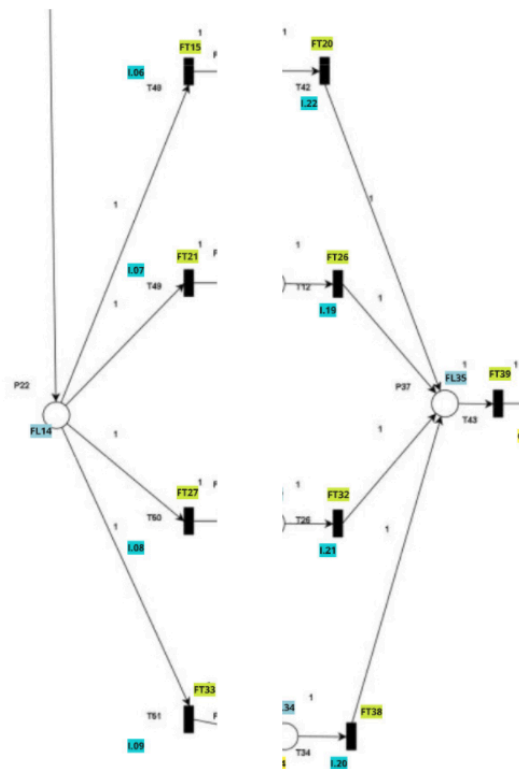


Fonte: Autoria própria.

No Ladder, todas as transições devem ser mapeadas, e as condições de entrada devem estar satisfeitas para que elas sejam ativadas. Por exemplo, para a transição FT02 ser ativada, é necessário que haja uma marca no lugar FL01 ($FL01=1$), e que o input I.01 também esteja ativado ($I.01=1$), se essas condições forem satisfeitas, após a varredura, FT02 assume o valor 1.

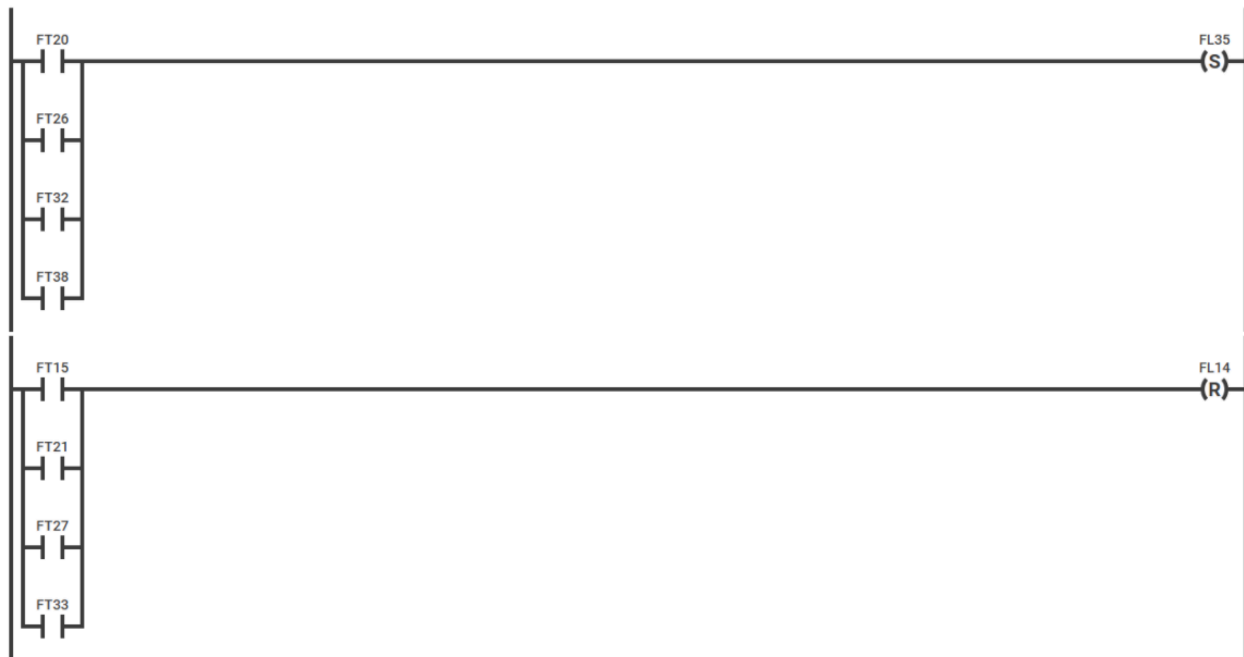
Em segundo lugar, as condições dos lugares são mapeadas utilizando os comandos "SET" e "RESET", de maneira semelhante ao mapeamento das transições. A Figura 20, apresentada a seguir, destaca os lugares relevantes na Rede de Petri e sua correspondente transcrição para a linguagem Ladder.

Figura 20 - Lugares em destaque na RdP



Fonte: Autoria própria.

Figura 21 - Transcrição dos lugares destacados para o ladder

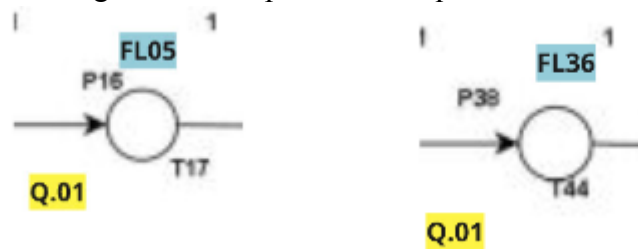


Fonte: Autoria própria.

No caso ilustrado, as condições de entrada, que no caso são as transições, são mostradas em paralelo no Ladder, já que basta alguma delas ser ativada para que o “SET” ou o “RESET” dos lugares seja ativo. Vale ressaltar que mesmo se as transições voltarem a valer zero após o “SET” ter sido realizado, o lugar permanecerá com a marca até que o “RESET” seja ativado.

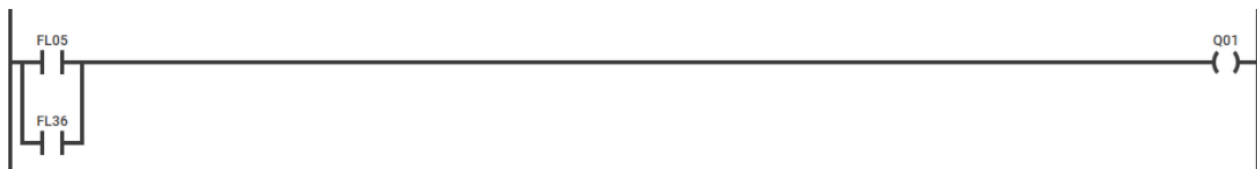
Por fim, são mapeados também de forma análoga os outputs usando a linguagem Ladder. A Figura 22 mostra um dos outputs do sistema que aparece em 2 lugares da RdP, enquanto a Figura 23 mostra sua transcrição para o Ladder.

Figura 22 - Output em destaque na RdP



Fonte: Autoria própria.

Figura 23 - Transcrição do output destacado para o ladder



Fonte: Autoria própria.

Nesse caso, as condições de entrada também estão em paralelo, já que caso qualquer uma delas esteja ativa, ou seja, possua marca, o output Q.01 estará ativo.

Por fim, junta-se o Ladder de todos os elementos da RdP, na ordem em que foram

construídos, para formar o Ladder final do sistema. Esse programa então pode ser inserido nos controladores para implementar a automação do sistema, completando o fluxo de implementação.

4 Observações finais

O presente trabalho propôs e desenvolveu uma solução de automação para o descarregamento de contêineres em navios cargueiros de grande porte, abordando questões críticas como segurança, eficiência e organização. Utilizando Sistemas a Eventos Discretos e Redes de Petri como base para a modelagem, foi possível estruturar um processo robusto, capaz de integrar sensores, sistemas de visão computacional e controladores lógicos programáveis, garantindo a execução automatizada e coordenada das operações.

Embora este trabalho tenha focado na modelagem do sistema, a aplicação prática de sistemas similares possuem o potencial de transformar significativamente as operações portuárias. A modularidade do modelo permite escalabilidade para embarcações maiores e adaptação para diferentes configurações portuárias, tornando-o uma solução versátil e economicamente viável. A implementação prática de um sistema como o descrito poderia levar a um aumento substancial na produtividade dos portos, além de promover maior segurança tanto para os trabalhadores quanto para a mercadoria.

Dada a complexidade crescente das operações logísticas globais, a automação de processos como o descarregamento de navios torna-se essencial para atender às demandas de escala e eficiência. Este projeto contribui não apenas com uma solução técnica, mas também com uma abordagem metodológica que pode ser replicada e aprimorada em outros contextos industriais. O impacto esperado de sua aplicação real vai além da eficiência operacional, abrangendo também a redução de custos, a minimização de desperdícios e o fortalecimento das cadeias de suprimentos globais.

Portanto, o trabalho aqui apresentado fornece uma base sólida para futuras pesquisas e implementações, consolidando-se como um exemplo do potencial transformador da automação mecatrônica no setor portuário. Espera-se que ele inspire novas iniciativas voltadas à modernização e à inovação, contribuindo para o avanço tecnológico e a sustentabilidade das operações logísticas em larga escala.

REFERÊNCIAS

- Franchi, C. M.; Camargo, V. L. A. Controladores Lógicos Programáveis - Sistemas Discretos. 1. edição, São Paulo, Érica, 2008.
- Cardoso, J.; Valette, R. Redes de Petri. Florianópolis, Editora da UFSC, 1997.
- Miyagi, P.E. Controle Programável - Fundamentos do Controle de Sistemas a Eventos Discretos. São Paulo, Ed. Edgard Blücher, 1996.

Title – Automation of Container Unloading from Large Cargo Ships and Color-Based Positioning Segmentation

Abstract – This work addresses, in the context of Discrete Event Systems, the development of a method for automation and control to tackle the issue of unloading large cargo ships and, additionally, the implementation of spatial segmentation of containers from different brands, a condition approximated by containers of different colors. The current unloading process heavily relies on human operators controlling the machinery that moves the containers. Human labor, in cases like this, poses a risk and vulnerability to the process, as workers are subject to fatigue, inattention, mistakes, and other conditions that can be very costly due to the large-scale loss of goods. Furthermore, automating the process is a means of increasing efficiency with significant potential to add value by saving time. These savings are reflected in the reduction of idle time for ships and in the decreased waiting time for other ships to dock.

Returning to the issue of spatial segmentation, it is understood that an important part of the unloading process is the ability to allocate containers in an organized manner to facilitate the transportation of goods out of the port. In this way, major brands can access their goods more easily, making the integration of the distribution network with the port less costly.

This work required the development of control for 39 signals, 30 of which were input signals. This was accomplished using the design methodology taught in the PMR 3305 course, whose steps include, in order: defining the physical architecture and structural model of the control system – its inputs and outputs (I/O); modeling the control behavior through PFS (Production Flow Schema) and RdP (Petri Nets) and defining interlocks; validating and simulating the model using PIPE 4.3; and applying isomorphism to program PLCs by converting the Petri Nets into Ladder Logic (LD). It is worth noting that the commissioning of the control system should mark the final stage of the project development, but the completion of this step was outside the proposed scope.

The conclusion of this project pointed to the real possibility of implementing safe automation systems for unloading cargo ships. The expected impact of the real implementation of a system similar to the one proposed here is to make large-scale freight transport networks more efficient, reducing costs and making the process safer, more organized, and more efficient.

Keywords – Automation, Control, Ships, Unloading, Container.



Mateus Matzkin Gurza, estudante de engenharia Mecatrônica pela Escola Politécnica da USP (Poli) com ensino médio concluído pelo Colégio Bandeirantes. Passagem pelo grupo de extensão: Poli Júnior, empresa júnior da Poli; tendo atuado na área de Inteligência de Dados com participação na Alta Gestão da empresa, gerenciando o núcleo de consultoria em Data Science e Engineering. Iniciação Científica em Visão Computacional voltada para análise de espaços de embedding e entropia da informação pelo núcleo Creativision, ligado ao Vision Research Group do IME-USP.



Thiago D. Velasquez, estudante de engenharia Mecatrônica pela Escola Politécnica da USP (Poli) com ensino médio concluído pelo Colégio Etapa. Passagem pelo grupo de extensão: Poli Júnior, empresa júnior da Poli, na área de Tecnologia e Desenvolvimento de Software com participação na Alta Gestão da empresa, gerenciando o núcleo de consultoria em projetos de Desenvolvimento Web.



Thiago Rosa Carvalho, estudante de engenharia Mecatrônica pela Escola Politécnica da USP (Poli) com ensino médio concluído pelo Colégio Bandeirantes. Diretor Financeiro e de Marketing da Equipe Poli Racing (EPR), trabalhando na prospecção de parcerias e comunicação, com passagem pela gestão da empresa. Iniciação Científica em navegação autônoma utilizando visão computacional pelo Laboratório de Percepção Avançada (LPA) da Poli USP