

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
MINAS GERAIS - *CAMPUS* BETIM  
BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Douglas Viana Dornelas

**SIMULAÇÃO, PROGRAMAÇÃO *OFFLINE* E COMISSIONAMENTO  
VIRTUAL DE UMA CÉLULA ROBOTIZADA**

Betim  
2024

DOUGLAS VIANA DORNELAS

**SIMULAÇÃO, PROGRAMAÇÃO *OFFLINE* E COMISSIONAMENTO  
VIRTUAL DE UMA CÉLULA ROBOTIZADA**

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus* Betim, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

**Orientador:** Prof. Dr. Arthur Hermano Rezende Rosa

Betim  
2024

## FICHA CATALOGRÁFICA

D696s Dornelas, Douglas Viana  
Simulação, programação offline e comissionamento virtual de  
uma célula robotizada / Douglas Viana Dornelas. – 2024.

72 f. : il.

Trabalho de conclusão de curso (Bacharelado em  
Engenharia de Controle e Automação) - Instituto Federal de  
Educação, Ciência e Tecnologia de Minas Gerais, Câmpus Betim,  
2024.

Orientador: Prof. Dr. Arthur Hermano Rezende Rosa

1. Robótica. 2. Comissionamento virtual. 3. Tecnomatix OLP. 4.  
Engenharia de Controle e Automação. I. Dornelas, Douglas Viana. II.  
Título.

CDU: 681.5

Douglas Viana Dornelas

## **SIMULAÇÃO, PROGRAMAÇÃO OFFLINE E COMISSIONAMENTO VIRTUAL DE UMA CÉLULA ROBOTIZADA**

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus* Betim, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Aprovado em: 18 / 09 / 2024 pela banca examinadora:



---

Prof. Dr. Arthur Hermanno Rezende Rosa (Orientador) - IFMG *Campus* Betim



---

Prof. Me. Herbert Ribeiro de Sá - IFMG *Campus* Betim



Documento assinado digitalmente

REGINALDO VAGNER FERREIRA

Data: 24/09/2024 17:55:12-0300

Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Reginaldo Vagner Ferreira - IFMG *Campus* Betim

## RESUMO

O ambiente industrial atual é altamente competitivo, exigindo das empresas foco em qualidade, agilidade e eficiência para atender às demandas dos consumidores por produtos personalizados e inovadores. A crescente complexidade e dinamismo do ambiente produtivo impõem a necessidade de processos industriais flexíveis, econômicos e inteligentes, com elevado nível de digitalização e automação. Especialmente na indústria automotiva, a velocidade de lançamento de novos produtos é fundamental para a competitividade. As tecnologias modernas, como a simulação digital de processos, a programação *offline* de robôs e o comissionamento virtual, permitem que as empresas realizem mudanças e adaptações rápidas e eficazes, otimizando a transição entre diferentes modelos de produção e identificando problemas antes da implementação física. Dessa forma, o objetivo deste trabalho de conclusão de curso consiste em desenvolver a simulação, a programação *offline* e o comissionamento virtual de uma célula robotizada no processo de fabricação automotivo, utilizando como meios os softwares Tecnomatix Process Simulate e TIA Portal. Sua principal contribuição baseia-se em demonstrar todas as etapas desse processo em ambiente digital. Os resultados confirmam a possibilidade de validar um processo automatizado com uma representação virtual precisa dos equipamentos reais. Além disso, a verificação prévia em software permitiu identificar interferências mecânicas e testar a lógica de comunicação entre robôs e CLPs, reduzindo o tempo de programação em campo e prevenindo problemas durante o *startup* da linha produtiva.

**Palavras-chave:** Simulação Robótica; Programação *Offline*; Comissionamento Virtual; Manufatura Digital; Tecnomatix Process Simulate.

## ABSTRACT

The current industrial environment is highly competitive, requiring companies to prioritize quality, agility, and efficiency to meet consumer demands for personalized and innovative products. The increasing complexity and dynamism of the production environment necessitate flexible, cost-effective, and intelligent industrial processes with a high level of digitization and automation. In the automotive industry, in particular, the speed of new product launches is crucial for maintaining competitiveness. Modern technologies, such as digital process simulation, offline robot programming, and virtual commissioning, enable companies to implement quick and effective changes and adaptations, optimizing the transition between different production models and identifying issues before physical implementation. Therefore, the objective of this work is to develop the simulation, offline programming, and virtual commissioning of a robotic cell in the automotive manufacturing process, using Tecnomatix Process Simulate and TIA Portal software. The main contribution lies in demonstrating all stages of this process in a digital environment. The results confirm the feasibility of validating an automated process with an accurate virtual representation of the actual equipment. Furthermore, preliminary software verification allowed for the identification of mechanical interferences and the testing of communication logic between robots and PLCs, reducing on-site programming time and preventing issues during the production line startup.

**Keywords:** Robotic Simulation; Offline Programming; Virtual Commissioning; Digital Manufacturing; Tecnomatix Process Simulate.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Substituição da ferramenta de manipulação de um robô em ambiente simulado.	12
Figura 2 – Configurações de manipuladores robóticos e seus respectivos envelopes.	14
Figura 3 – Robô articulado de 6 eixos.	15
Figura 4 – Movimento de robôs em relação a <i>frames</i> de referência.	16
Figura 5 – <i>Teach Pendant</i> do fabricante Comau.	17
Figura 6 – Ambiente de Programação <i>Offline</i> da empresa RoboDK.	18
Figura 7 – Configurações de comissionamento aplicáveis na indústria.	19
Figura 8 – Comissionamento <i>Hardware-in-the-loop</i> e <i>Software-in-the-loop</i> .	20
Figura 9 – Ambiente de trabalho no <i>software</i> Process Simulate.	21
Figura 10 – Ambiente de trabalho no <i>software</i> Tia Portal.	22
Figura 11 – Integração entre Tecnomatix Process Simulate e Tia Portal.	23
Figura 12 – Fluxo de processos para simulação virtual.	24
Figura 13 – Produto cujo processo de manufatura foi simulado.	25
Figura 14 – Sub conjuntos que formam o <i>Spider</i> .	25
Figura 15 – Pontos de solda do produto: Geometria e <i>Respot</i> .	26
Figura 16 – <i>Layout</i> inicial com definição das operações.	27
Figura 17 – Elementos da primeira estação de abastecimento: OP05.	27
Figura 18 – Elementos da segunda estação de abastecimento: OP10.	28
Figura 19 – Sensores de presença de peça das estações de abastecimento: OP05 e OP10.	29
Figura 20 – Dispositivo de solda de geometria da OP20.	29
Figura 21 – Exemplo de cinemática criada para os grampos e pinos de localização.	30
Figura 22 – Garras de manipulação utilizadas na simulação.	30
Figura 23 – Pinça de solda da OP20 e sua cinemática.	31
Figura 24 – Modelos de robôs Fanuc utilizados na simulação.	32
Figura 25 – Robô 05R01 em diferentes posições: repouso, pega e depósito.	32
Figura 26 – Atributos de movimentos do robô 05R01.	33
Figura 27 – Nuvem de pinça do robô 20R01.	34
Figura 28 – Atributos de movimentos no programa de solda do robô 020R01.	34
Figura 29 – Ciclo dos robôs 30R01 e 30R02.	35
Figura 30 – Organização, divisão e nomenclatura de programas dos robôs 10R01 e 20R01.	36
Figura 31 – Robô 20R01 e equipamento utilizado para fresagem e troca de eletrodos da pinça de solda.	37
Figura 32 – Sinais de entrada referentes aos sensores de presença de peça.	38
Figura 33 – Sinais para controle do dispositivo de geometria da OP20.	38
Figura 34 – Bloco lógico para controle do dispositivo de geometria da OP20.	39
Figura 35 – Sinais do robô 20R02: sinais padrão em verde e sinais de processo em azul.	39
Figura 36 – Portões de acesso da estação automática.	40
Figura 37 – Áreas monitoradas nas estações de abastecimento.	40

Figura 38 – Configuração do CLP no <i>software</i> TIA Portal. . . . .	41
Figura 39 – Estrutura de programação no <i>software</i> TIA Portal. . . . .	41
Figura 40 – Comunicação entre os <i>softwares</i> Tecnomatix Process Simulate e TIA Portal. . . . .	42
Figura 41 – <i>Layout</i> final da célula robotizada. . . . .	43
Figura 42 – Trajetórias dos robôs apresentadas em sequência. . . . .	44
Figura 43 – Interferência mecânica entre eletrodo da pinça de solda do robô 20R02 e dispositivo de geometria. . . . .	45
Figura 44 – Verificação de colisão entre robôs através de seus volumes de trabalho. . . . .	45
Figura 45 – Exemplo de programa do robô obtido através do <i>software</i> de simulação. . . . .	46
Figura 46 – <i>QR Code</i> para acesso ao vídeo de funcionamento da célula robotizada. . . . .	47

## **LISTA DE ABREVIATURAS E SIGLAS**

OLP	<i>Offline Programming</i>
VC	<i>Virtual Commissioning</i>
CLP	Controlador Lógico Programável
ISO	<i>International Organization for Standardization</i>
TC	<i>Technical Committee</i>
PLM	<i>Product Lifecycle Management</i>
TIA	<i>Totally Integrated Automation</i>
SD	<i>Start Distance</i>
ED	<i>End Distance</i>
CO-OP	<i>Co-operative</i>
GM	<i>General Motors</i>
TD	<i>Tip Dresser</i>
CC	<i>Cap Changer</i>
CPU	<i>Central Processing Unit</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
1.1	Justificativa	12
1.2	Objetivos	13
1.2.1	<i>Objetivo geral</i>	13
1.2.2	<i>Objetivos específicos</i>	13
1.3	Organização do Texto	13
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>14</b>
2.1	Robótica industrial	14
2.1.1	<i>Classificação de robôs industriais</i>	14
2.1.2	<i>Localização no espaço: Frames de referência</i>	15
2.1.3	<i>Métodos de programação de robôs</i>	17
2.2	Comissionamento Virtual	19
2.2.1	<i>Requisitos para o comissionamento virtual</i>	20
2.3	Softwares de simulação e Comissionamento Virtual	21
2.3.1	<i>Tecnomatix Process Simulate</i>	21
2.3.2	<i>TIA Portal</i>	21
2.3.3	<i>Integração entre os softwares</i>	22
<b>3</b>	<b>METODOLOGIA</b>	<b>24</b>
3.1	Produto e processo de fabricação	25
3.2	Layout e fluxo dos processos	26
3.3	Simulação	28
3.3.1	<i>Equipamentos simulados: criação de sensores e cinemática</i>	28
3.3.2	<i>Robôs utilizados na simulação</i>	31
3.3.3	<i>Trajetórias dos robôs</i>	32
3.4	Programação Offline dos robôs	35
3.5	Comissionamento Virtual	37
3.5.1	<i>Criação de sinais e blocos lógicos</i>	37
3.5.2	<i>Dispositivos de segurança</i>	39
3.5.3	<i>Programação do CLP</i>	40

<b>4</b>	<b>RESULTADOS</b> . . . . .	<b>43</b>
<b>4.1</b>	<b>Layout</b> . . . . .	<b>43</b>
<b>4.2</b>	<b>Simulação</b> . . . . .	<b>43</b>
<b>4.3</b>	<b>Programação <i>Offline</i> dos robôs</b> . . . . .	<b>46</b>
<b>4.4</b>	<b>Comissionamento Virtual</b> . . . . .	<b>46</b>
<b>5</b>	<b>CONCLUSÃO</b> . . . . .	<b>48</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>49</b>
	<b>APÊNDICE A – PROGRAMAS DOS ROBÔS</b> . . . . .	<b>51</b>
	<b>APÊNDICE B – PROGRAMA DO CLP</b> . . . . .	<b>62</b>

# 1 INTRODUÇÃO

No cenário atual, o ambiente de manufatura tornou-se altamente competitivo, tornando imprescindível que as empresas priorizem a qualidade dos produtos, a agilidade e a eficiência em suas operações para garantir a sobrevivência no mercado. Os consumidores exigem cada vez mais produtos personalizados e inovadores, acompanhados de uma resposta rápida por parte das empresas (BHATIA; KUMAR, 2020). Paralelamente, o ambiente produtivo tem se tornado mais complexo e dinâmico, o que impõe grandes desafios às organizações. Para superar esses desafios, é fundamental que os processos e as estratégias da indústria sejam não apenas flexíveis e economicamente eficientes, mas também inteligentes, capazes de operar de forma autônoma e adaptativa. Isso requer um alto nível de digitalização, automação e integração entre diferentes ferramentas de engenharia (PAPULOVA; GAZOVA; SUFLIARSKY, 2022).

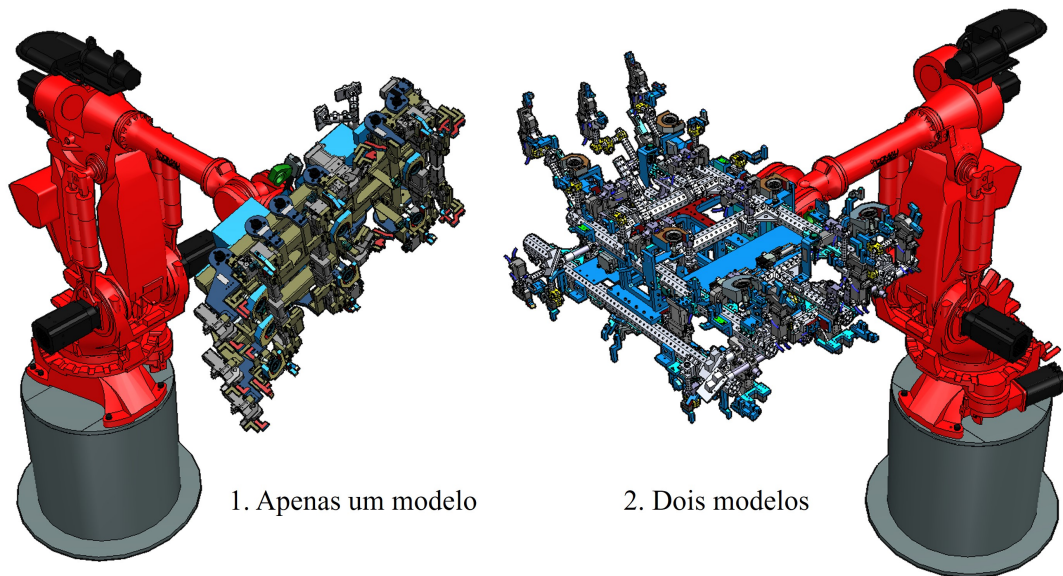
Durante cada etapa do desenvolvimento de um produto, mudanças acontecem no *design* inicial, colocando as empresas em uma posição em que devem decidir se implementam ou não as alterações especificadas. Sabendo exatamente o impacto de cada mudança, os engenheiros não precisam esperar por estudos complexos e podem tomar decisões com muito mais rapidez. Ter uma representação digital do processo real e da fábrica é de grande valia na detecção de problemas e nos testes para identificar qual cenário ou solução é melhor implementar (PIROMALIS; KANTAROS, 2022).

Com as tecnologias atuais é possível fazer grandes mudanças e adaptar a automação de processos simultaneamente a tomada de decisões. A indústria automotiva é um bom exemplo, uma vez que um novo modelo de carro é lançado a cada um ou dois anos. A mesma linha de produção pode ser usada na transição de um modelo para o novo e, em pouco tempo, dois ou mais modelos podem ser fabricados nas mesmas células robotizadas (PAPULOVA; GAZOVA; SUFLIARSKY, 2022). A Figura 1 apresenta em exemplo do que foi mencionado, mostrando a substituição de uma ferramenta de manipulação utilizada para apenas um modelo de veículo por uma nova ferramenta, capaz de manipular peças de dois modelos distintos.

Diante disso, destacam-se abordagens de engenharia que envolvem a simulação de processos em ambiente digital, a programação *offline* (*Offline Programming* - OLP) de manipuladores robóticos e o comissionamento virtual (*Virtual Commissioning* - VC) de sistemas automáticos (DUMITRASCU; DINCA; PREDINCEA, 2017). A utilização dessas ferramentas em um projeto viabiliza a validação de equipamentos utilizando seu modelo 3D, permite que robôs sejam programados sem a presença do robô físico e possibilita que todo processo seja comissionado com um controlador lógico programável (CLP) antes de sua implementação física na planta produtiva (PAN *et al.*, 2012).

Nas duas seções seguintes, a justificativa e os objetivos deste trabalho são abordados, esclarecendo a escolha pelo desenvolvimento da simulação, programação *offline* e comissionamento virtual de uma célula robotizada.

Figura 1 – Substituição da ferramenta de manipulação de um robô em ambiente simulado.



Fonte: Elaborado pelo autor, 2021.

## 1.1 Justificativa

Na atualidade, um dos aspectos que impulsiona o mercado competitivo da indústria automotiva é ser o primeiro a lançar um produto no mercado (GASTROW, 2012). Para que o tempo de lançamento seja curto é importante acompanhar todas as fases do processo, desde o projeto inicial até a instalação física da planta. Além disso, devido ao fato dos processos de fabricação tenderem a ser extremamente complexos, lidando com centenas de diferentes dispositivos em constante comunicação entre si, é de extrema importância o uso de uma tecnologia que permita alta flexibilidade e grande assertividade (LLOPIS-ALBERT; RUBIO; VALERO, 2021).

Ao analisar grande parte das ferramentas convencionais que são utilizadas no âmbito da manufatura digital, como por exemplo o *software* FactoryCad, percebe-se que elas não trazem nenhum benefício além de apresentar o modelo 3D da planta (COHEN *et al.*, 2019). O comissionamento virtual, por outro lado, permite que um programador identifique qualquer erro no código do controlador lógico programável antes mesmo do *startup* da planta.

Em processos automatizados que utilizam sistemas robóticos, o comissionamento virtual está diretamente relacionado à programação *offline* dos manipuladores robóticos. De forma geral, a programação *offline* de robôs consiste em programá-los fora do ambiente de produção, eliminando o tempo de inatividade causado pela programação em ambiente fabril. Essas duas ferramentas se complementam e fazem parte da simulação de processos robotizados em ambiente digital (DUMITRASCU; DINCA; PREDINCEA, 2017). Diante disso, se torna possível estudar vários cenários de uma célula robotizada antes mesmo de construí-la.

## 1.2 Objetivos

### 1.2.1 *Objetivo geral*

O presente trabalho apresenta como objetivo geral desenvolver a simulação, a programação *offline* e o comissionamento virtual de uma célula robotizada.

### 1.2.2 *Objetivos específicos*

- Simular o processo de fabricação robotizado de uma indústria automotiva através do *software* Tecnomatix Process Simulate.
- Criar cinemáticas para os equipamentos utilizados e desenvolver as trajetórias dos robôs.
- Programar os robôs utilizando seu controlador virtual.
- Comissionar a célula robotizada em ambiente digital, por meio da integração entre os *softwares* Tecnomatix Process Simulate e Tia Portal.

## 1.3 Organização do Texto

No Capítulo 2 deste trabalho é apresentada uma breve revisão bibliográfica, abordando algumas definições relacionadas a robótica industrial, métodos de programação de robôs, comissionamento virtual e *softwares* pertinentes ao tema. O Capítulo 3 evidencia a metodologia utilizada para o desenvolvimento do trabalho, descrevendo a sequência de atividades realizadas para a elaboração da simulação da célula robotizada. No Capítulo 4 os resultados obtidos por meio da simulação são apresentados e, no Capítulo 5, são abordados os principais pontos de conclusão do trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Robótica industrial

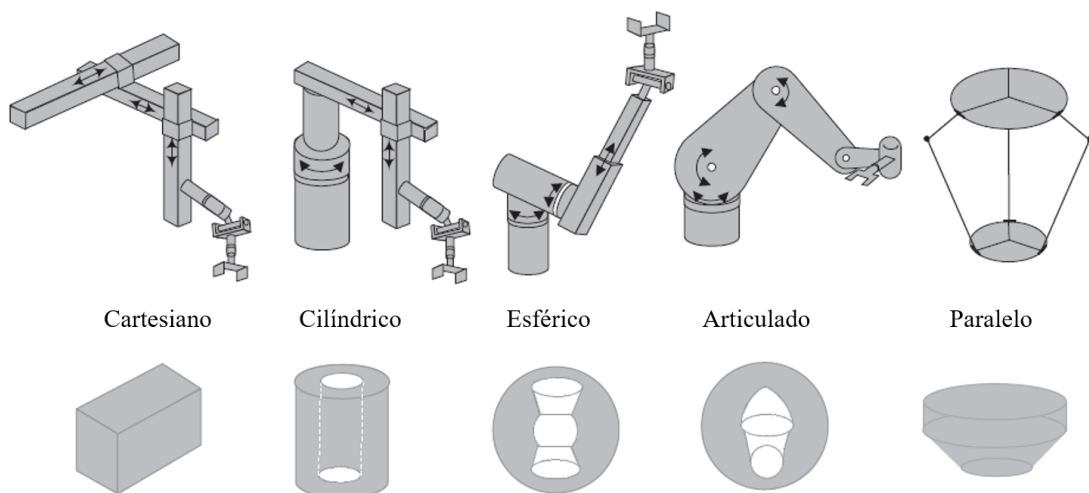
A robótica pode ser caracterizada como a arte, a base de conhecimento e a capacidade de projetar e aplicar robôs em tarefas e esforços humanos. Os sistemas robóticos não se limitam apenas a robôs, mas se estendem a outros dispositivos e sistemas utilizados em conjunto com os mesmos. Os robôs podem ser usados em ambientes de manufatura, na exploração subaquática e espacial, para auxiliar pessoas com deficiência, ou mesmo para entretenimento. A robótica é um assunto interdisciplinar que se beneficia da engenharia mecânica, engenharia elétrica e eletrônica, ciência da computação, ciências cognitivas, biologia e muitas outras disciplinas (NIKU, 2020).

Apesar de todos os ambientes citados anteriormente, o presente trabalho concentra seu interesse apenas na robótica industrial. De acordo com (ISO/TC 299, 2016), considera-se um robô industrial o “manipulador multiuso reprogramável, controlado automaticamente, programável em três ou mais eixos, que pode ser fixo ou móvel para uso em aplicações de automação industrial”.

#### 2.1.1 Classificação de robôs industriais

Uma parte importante de qualquer estudo de aplicações de robôs é a compreensão do processo de seleção dos mesmos. Embora tenham sido feitas tentativas para introduzir procedimentos de seleção sistemática, na maioria dos casos, a seleção de robôs é subjetiva. Independentemente de o engenheiro de aplicação chegar a uma identificação racional de um robô adequado para uma tarefa seguindo um procedimento rígido, ou se decide mais livremente, é importante que o mesmo esteja familiarizado com os vários meios de classificação desses manipuladores (WILLIAMS; APPLETON, 1987).

Figura 2 – Configurações de manipuladores robóticos e seus respectivos envelopes.



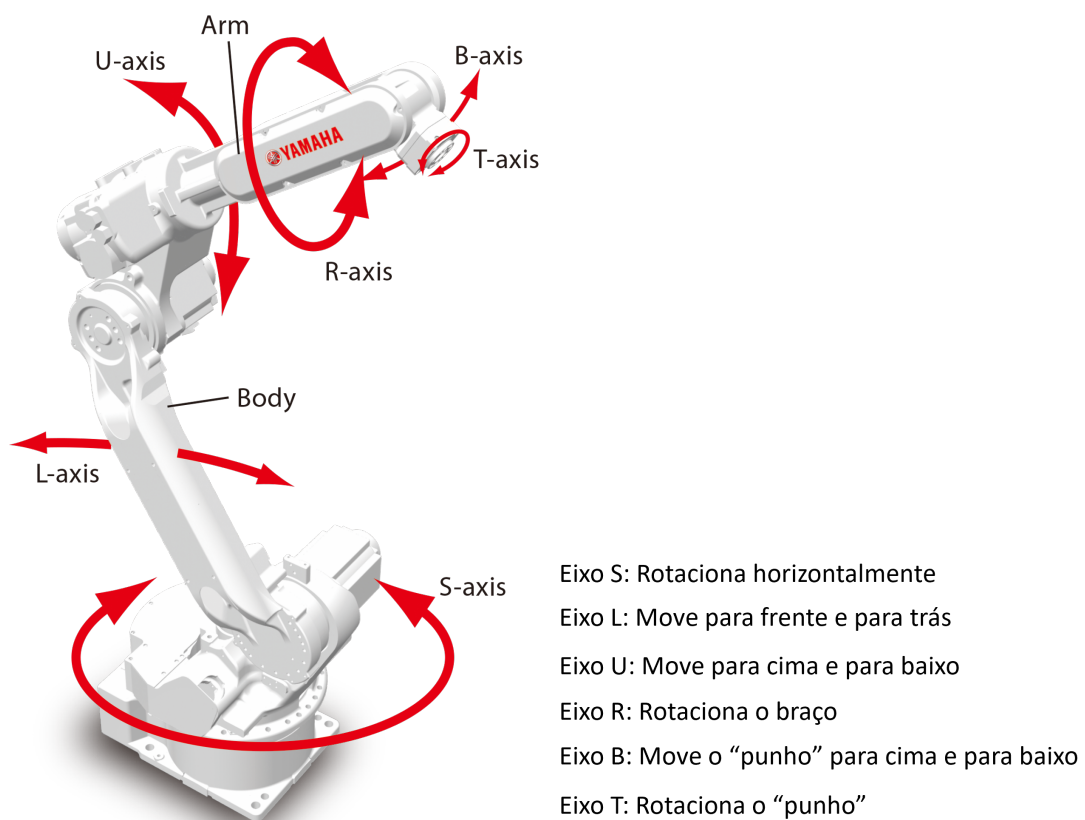
Fonte: Adaptado de NIKU, 2020.

A classificação com base na anatomia do manipulador é provavelmente a forma mais óbvia

e estabelecida de classificação de robôs. A maneira mais comum de agrupamento é considerar a forma do envelope que representa os limites de alcance do braço robótico, ou usar o sistema de coordenadas que descreve o volume contido neste envelope (WILLIAMS; APPLETON, 1987). A Figura 2 apresenta cinco configurações de manipuladores robóticos e seus respectivos envelopes de trabalho.

Das cinco categorias de robôs apresentadas na Figura 2, o articulado se destaca como a configuração mais comum de robô industrial, sendo composto inteiramente de juntas de rotação, similar a um braço humano (NIKU, 2020). Na Figura 3 é possível visualizar um robô articulado de seis eixos do fabricante Yamaha, seus eixos e direções dos movimentos são destacados.

Figura 3 – Robô articulado de 6 eixos.



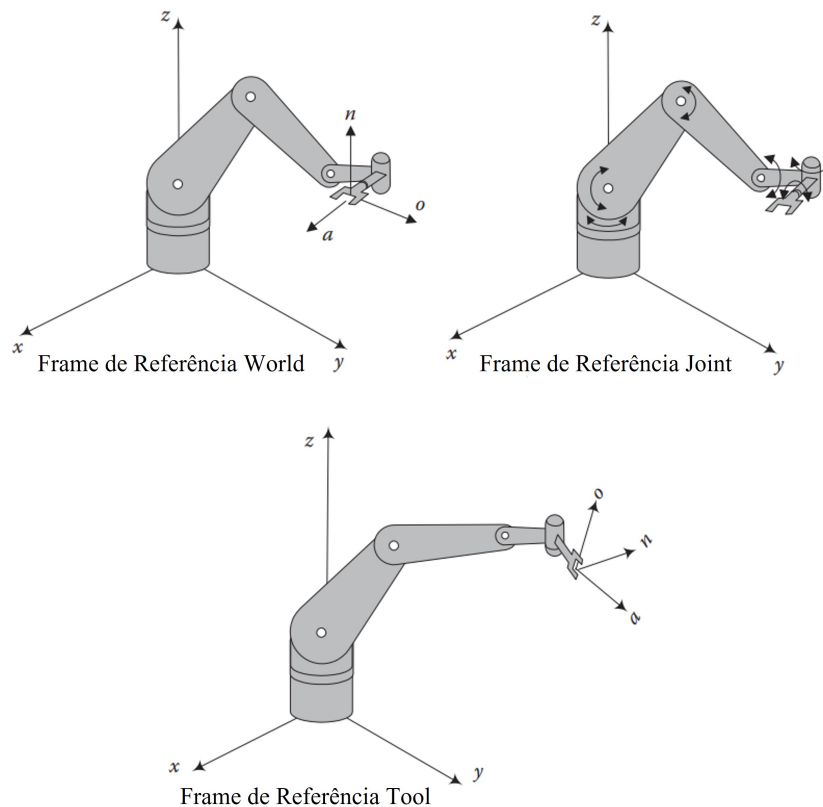
Fonte: Adaptado de YAMAHA, 2024.

### 2.1.2 Localização no espaço: *Frames* de referência

Sabe-se que, para localizar um ponto no espaço, é necessário especificar três coordenadas (como as coordenadas  $x$ ,  $y$  e  $z$  ao longo dos três eixos cartesianos). Três coordenadas são necessárias e adequadas para completamente definir a localização de um ponto. Embora diferentes sistemas de coordenadas possam ser usados para expressar essa informação, elas são sempre necessárias. No entanto, nem duas nem quatro serão possíveis; duas é inadequado para localizar um ponto no espaço e quatro é desnecessário, pois há informações demais (NIKU, 2020).

Para localizar um corpo rígido (um objeto tridimensional em vez de um ponto) no espaço, primeiro é preciso especificar a localização de um ponto selecionado nele e, portanto, exige-se três informações. Em seguida, também é necessário definir a orientação do objeto para especificá-lo totalmente no espaço. Isso significa que seis informações são necessárias para especificar totalmente a localização e orientação de um corpo rígido (NIKU, 2020). Os robôs podem se mover em relação a diferentes *frames* de referência, resultando em diferentes movimentos.

Figura 4 – Movimento de robôs em relação a *frames* de referência.



Fonte: Adaptado de NIKU, 2020.

A Figura 4 apresenta três *frames* de referência. O *frame World* é um sistema de coordenadas universal definido pelos eixos  $x$ ,  $y$  e  $z$ . As articulações do robô se movem simultaneamente de maneira coordenada para criar movimentos ao longo dos três eixos principais. Nesse *frame*, não importa onde o braço robótico esteja, um movimento positivo ao longo do eixo  $x$  é sempre na direção positiva do eixo  $x$  e assim por diante. O propósito do *frame World* é definir os movimentos do robô em relação a outros objetos, definir a posição de outras peças e máquinas com as quais o robô comunica, além de especificar as trajetórias de movimento (CRAIG, 2004).

O sistema de referência *Joint* é usado para especificar movimentos das articulações individuais do robô. Nesse caso, cada eixo é acessado e movido individualmente e, portanto, apenas um eixo se move por vez. O *frame Tool*, no entanto, especifica os movimentos do punho do robô em relação a uma ferramenta fixa no mesmo (CRAIG, 2004). Dessa forma, todos os movimentos são referentes aos eixos “o”, “n” e “a” presentes na Figura 4.

### 2.1.3 Métodos de programação de robôs

Atualmente existem duas categorias principais de programação de robôs: programação *online* e programação *offline* (OLP). A programação *online* tem sido convencionalmente desempenhada por programadores qualificados, guiando o robô pela trajetória desejada usando um *Teach Pendant*, apresentado pela Figura 5.

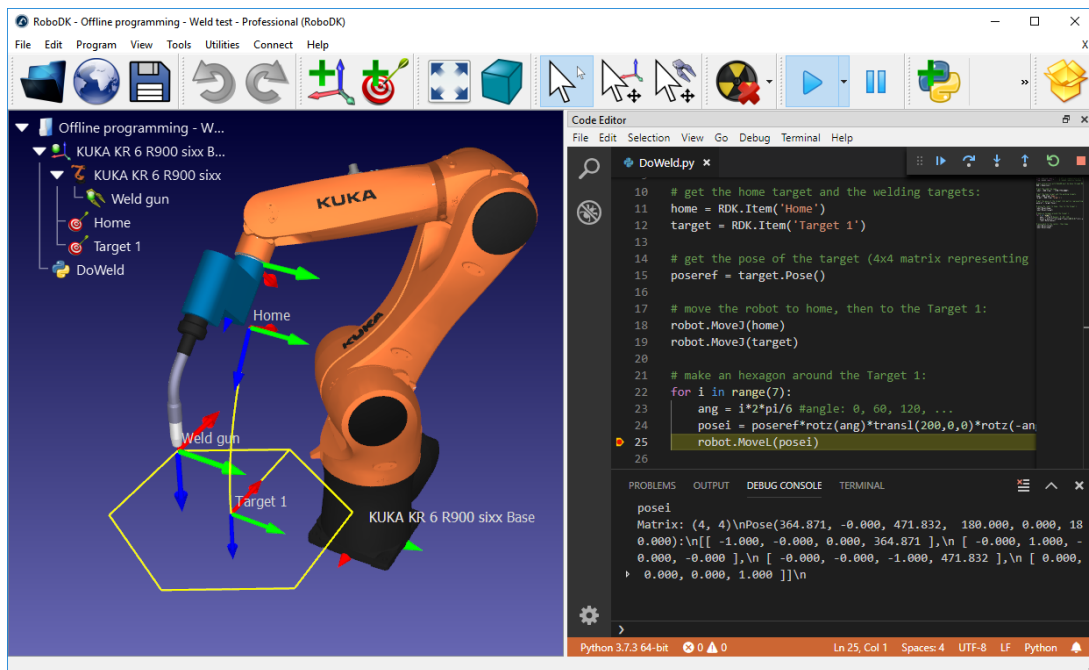
Figura 5 – *Teach Pendant* do fabricante Comau.



Fonte: COMAU, 2024.

Normalmente, esse método inclui as etapas de movimentar o robô pelo caminho desejado, registrando os pontos específicos no controlador do robô e utilizando as coordenadas registradas para criar comandos de movimento (HEIMANN; GUHL, 2020). Embora o método de programação *online* convencional seja simples e tenha sido amplamente utilizado, ele apresenta várias desvantagens. Primeiramente, movimentar um robô usando um *teach pendant* não é intuitivo, pois geralmente muitos sistemas de coordenadas são definidos em um manipulador robótico. O robotista deve sempre rastrear em qual *frame* de referência o robô está ao movimentá-lo (BOTTAZZI; FONSECA, 2007).

Guiar o robô pelo movimento desejado com precisão, sem nunca permitir uma colisão com um objeto na área de trabalho, é geralmente uma tarefa muito difícil e demorada, especialmente quando o produto tem uma geometria complexa ou o processo em si é muito complicado (PAN *et al.*, 2012). Além disso, quando um programa é gerado, muitos testes devem ser realizados antes que o mesmo apresente desempenho satisfatório, por motivos de confiabilidade e segurança. Finalmente, o programa do robô gerado usando tal método carece de flexibilidade e capacidade de reutilização. O complicado processo de programação deve ser repetido novamente para um produto com apenas uma pequena diferença (PAN *et al.*, 2012).

Figura 6 – Ambiente de Programação *Offline* da empresa RoboDK.

Fonte: ROBODK, 2024.

Define-se um sistema de programação *offline* como uma linguagem de programação de robô que foi suficientemente ampliada, geralmente por meio de computação gráfica, de forma que o desenvolvimento dos programas possa ocorrer sem acesso ao próprio robô (CRAIG, 2004). A Figura 6 apresenta um ambiente de programação *offline* de robôs da empresa RoboDK.

Sendo assim, a OLP utiliza modelos CAD 3D de uma determinada peça de trabalho e da célula robotizada para desenvolver e simular programas, sendo muito utilizada em sistemas de manufatura de larga escala. O usuário pode então testar a acessibilidade, ajustar com precisão os movimentos do robô e manipular informações relacionadas ao processo antes de gerar um programa, que posteriormente pode ser baixado para o manipulador robótico (PAN *et al.*, 2012).

A programação *offline* oferece muitas vantagens sobre o método *online*. Primeiramente, o processo de programação não requer o robô real, minimizando o tempo de inatividade do manipulador robótico presente em uma célula existente. Os programas do robô podem ser desenvolvidos mais cedo no ciclo de projeto / produção e a programação pode ser realizada em paralelo com a produção, em vez de em série com a mesma. Em segundo lugar, os programas gerados no processo *offline* são mais flexíveis do que utilizando o *teach pendant*. Mudanças no programa podem ser incorporadas rapidamente substituindo apenas a parte necessária do mesmo e rotinas previamente desenvolvidas podem ser facilmente incluídas em novos programas. Não somente, a simulação é geralmente incorporada ao método OLP (PAN *et al.*, 2012). Como resultado, os programas podem ser verificados previamente, confirmando assim os movimentos dos robôs, minimizando a chance de erro e, portanto, melhorando a produtividade e a segurança. Também existe uma maior possibilidade de otimização do *layout* do espaço de trabalho e do

planejamento das tarefas do manipulador.

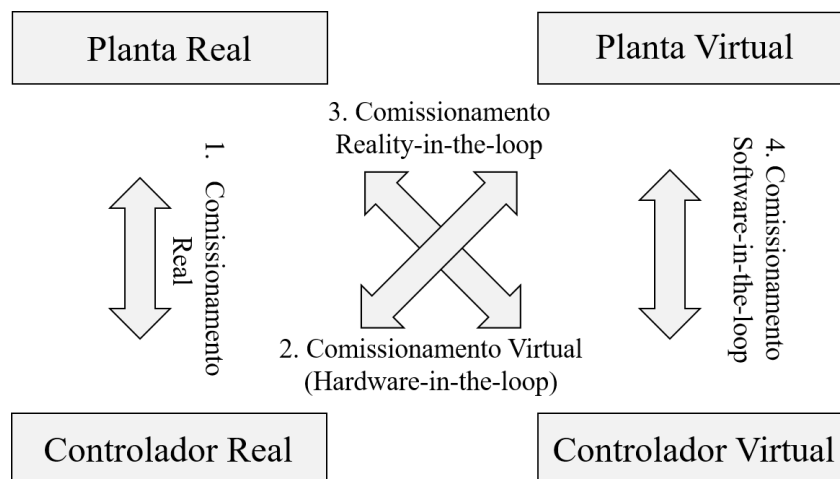
## 2.2 Comissionamento Virtual

O conceito de Comissionamento Virtual, ou *Virtual Commissioning* (VC), se tornou uma das fases mais essenciais no desenvolvimento e simulação de processos automatizados. Ao criar uma cópia digital do sistema de manufatura real, as indústrias passam a ter a capacidade de testar cenários mais complexos com sistemas robóticos automatizados, diminuindo consideravelmente o tempo de *startup* da planta e reduzindo o tempo de lançamento de um produto ao mercado, enquanto aumenta a qualidade geral simultaneamente (DUMITRASCU; DINCA; PREDINCEA, 2017).

O aspecto fundamental do comissionamento virtual é a digitalização de todos os dispositivos presentes na planta industrial, a fim de que o modelo virtual funcione exatamente igual ao real. Este também é um dos princípios básicos do que hoje é conhecido como Indústria 4.0, ter uma representação digital da planta e comissionar esse modelo antes e/ou paralelamente à planta real (LEE; PARK, 2014). Com os recursos atuais, é possível testar todo processo com softwares. Uma opção viável e comumente utilizada consiste em utilizar o modelo computacional da planta em um computador e usar um controlador lógico programável real para desenvolver o código e executar a simulação.

A Figura 7 apresenta quatro configurações de comissionamento aplicáveis na indústria, sendo a quarta diretamente relacionada ao presente trabalho.

Figura 7 – Configurações de comissionamento aplicáveis na indústria.

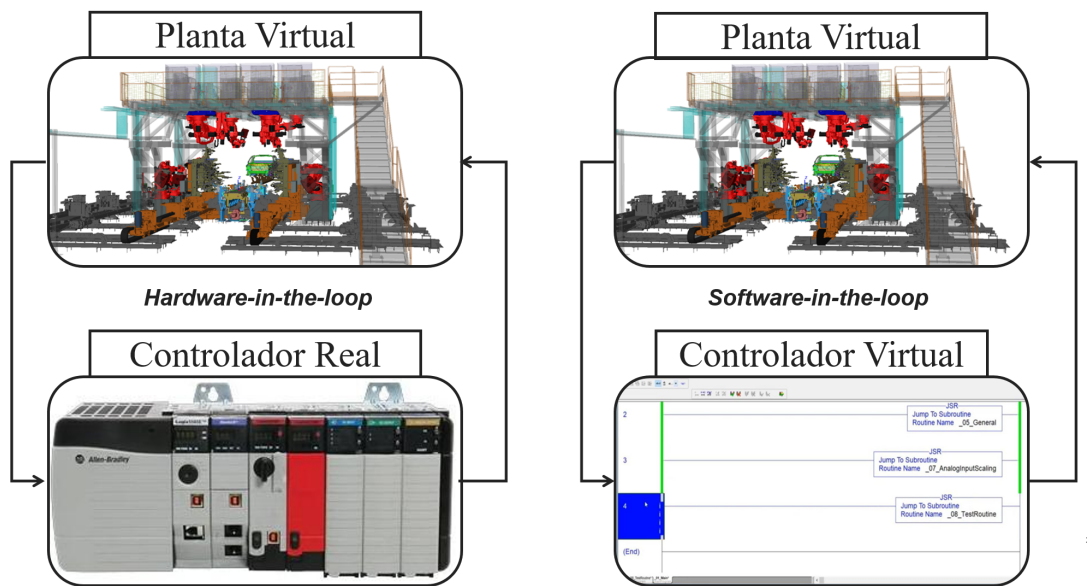


Fonte: Elaborado pelo autor, 2024.

O Comissionamento *software-in-the-loop* consiste em utilizar uma representação virtual do processo e do controlador (AYED; ZOUARI; ABID, 2017). Dessa maneira, a modelagem 3D da planta é controlada por um *software* de CLP emulado. Tal método tem a vantagem de ser

relativamente barato e rápido para implantar. Além disso, por não usar nenhum equipamento real, vários problemas relacionados à segurança são evitados completamente (como colisões de robôs ou ferimentos em humanos) (DUMITRASCU; DINCA; PREDINCEA, 2017). Assim como o método anterior, a simulação *hardware-in-the-loop* faz uso de um modelo 3D da planta (BACIC, 2005). No entanto, esse método se estende à utilização do CLP físico. Sendo assim, pode-se executar o código no modelo exato do CLP com a configuração de hardware especificada para o projeto. Os conceitos explorados anteriormente são apresentados pela Figura 8.

Figura 8 – Comissionamento *Hardware-in-the-loop* e *Software-in-the-loop*.



Fonte: Elaborado pelo autor, 2024.

### 2.2.1 Requisitos para o comissionamento virtual

Para implementar a fase de comissionamento virtual corretamente, é importante ter os seguintes pré-requisitos, tanto da engenharia mecânica quanto da engenharia de controle (LIU; SUCHOLD; DIEDRICH, 2012):

- Um *layout* detalhado da planta e uma descrição correta de todos os recursos utilizados.
- Simulação 3D abrangente para cada tipo de dispositivo e robôs que serão utilizados. Isso significa que é absolutamente necessário possuir um modelo 3D em extensão “.jt” com a geometria e cinemática corretas, os robôs devem ter todas as operações finalizadas com as zonas de interferência definidas e todas as demais medidas de segurança implementadas no modelo.
- A sequência de operações deve estar completa e definida da forma mais eficiente possível.
- Os sinais de entrada e saída devem ser mapeados e definidos corretamente.

- A configuração de hardware do CLP deve ser determinada e deve estar disponível para o engenheiro de controle, como dispositivo físico ou emulado.

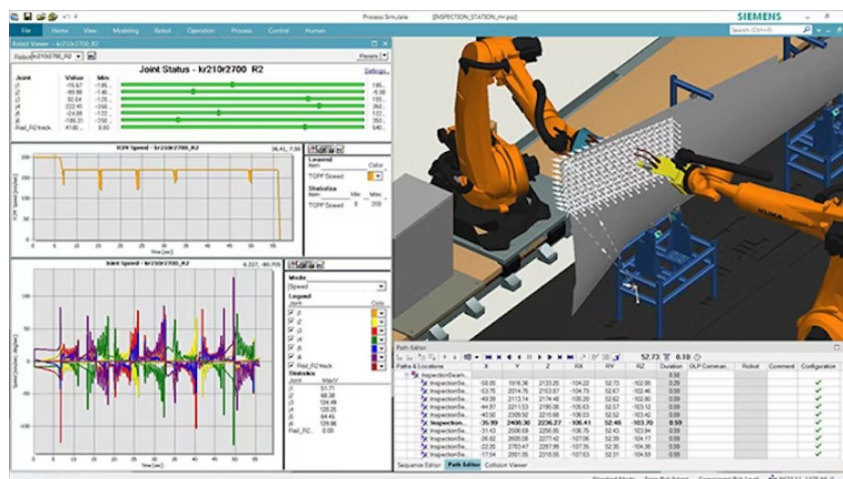
## 2.3 Softwares de simulação e Comissionamento Virtual

### 2.3.1 Tecnomatix Process Simulate

O Tecnomatix Process Simulate é um *software* da Siemens que faz parte da suíte de soluções PLM (*Product Lifecycle Management*). Este *software* é amplamente utilizado para a simulação de processos de manufatura e montagem, com foco em otimizar e validar processos antes da implementação física. Ele oferece ferramentas para simulação de robôs, ergonomia, processos de montagem e validação de tempo de ciclo, permitindo que os engenheiros detectem e corrijam problemas em fases iniciais do projeto (SIEMENS, 2024).

Uma das principais características do Process Simulate é sua capacidade de criar gêmeos digitais de linhas de produção. Isso significa que é possível simular e visualizar operações de produção em um ambiente virtual, identificando e solucionando problemas antes de construir fisicamente a linha de produção. Esse tipo de simulação permite a otimização do *layout* de fábrica, a validação de processos automatizados, e a análise de desempenho do sistema, o que resulta em uma redução significativa de custos e tempo (SIEMENS, 2024). A Figura 9 apresenta o ambiente de trabalho do Process Simulate.

Figura 9 – Ambiente de trabalho no *software* Process Simulate.



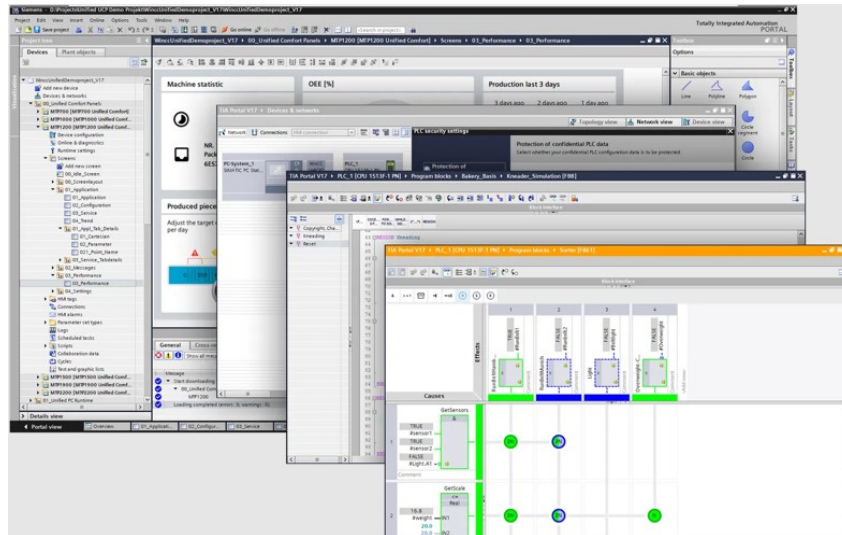
Fonte: SIEMENS, 2024.

### 2.3.2 TIA Portal

O TIA Portal (*Totally Integrated Automation Portal*) é outra ferramenta da Siemens, voltada para a automação industrial. Este ambiente de engenharia integrado permite o planejamento, a programação e o comissionamento de sistemas de automação, como controladores

lógicos programáveis, interfaces homem-máquina (IHMs) e drives. O TIA Portal é projetado para oferecer uma plataforma unificada onde todas as etapas do processo de automação possam ser realizadas, desde a configuração de hardware até a programação e a visualização (SIEMENS., 2024). A Figura 10 apresenta o ambiente de trabalho desse *software*.

Figura 10 – Ambiente de trabalho no *software* Tia Portal.



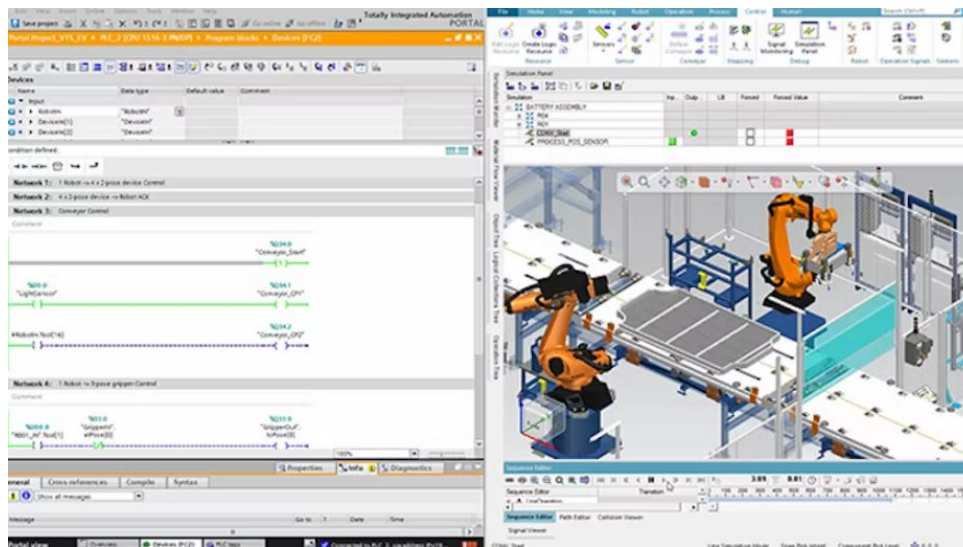
Fonte: SIEMENS., 2024.

Entre os principais benefícios do TIA Portal estão a facilidade de integração entre diferentes componentes de automação e a sua interface intuitiva, que reduz a complexidade do desenvolvimento de projetos.

### 2.3.3 Integração entre os *softwares*

No contexto do comissionamento virtual, o Tecnomatix Process Simulate pode ser usado para criar uma simulação detalhada da linha de produção, incluindo robôs, transportadores, e outras máquinas. Essa simulação pode então ser conectada ao TIA Portal, que fornece o programa de automação real que será usado no ambiente físico. Essa integração permite que os engenheiros testem o comportamento do sistema de automação em um ambiente virtual, identificando e corrigindo problemas antes do comissionamento físico (SIEMENS, 2024). A Figura 11 mostra os *softwares* em questão operando de forma integrada.

Figura 11 – Integração entre Tecnomatix Process Simulate e Tia Portal.

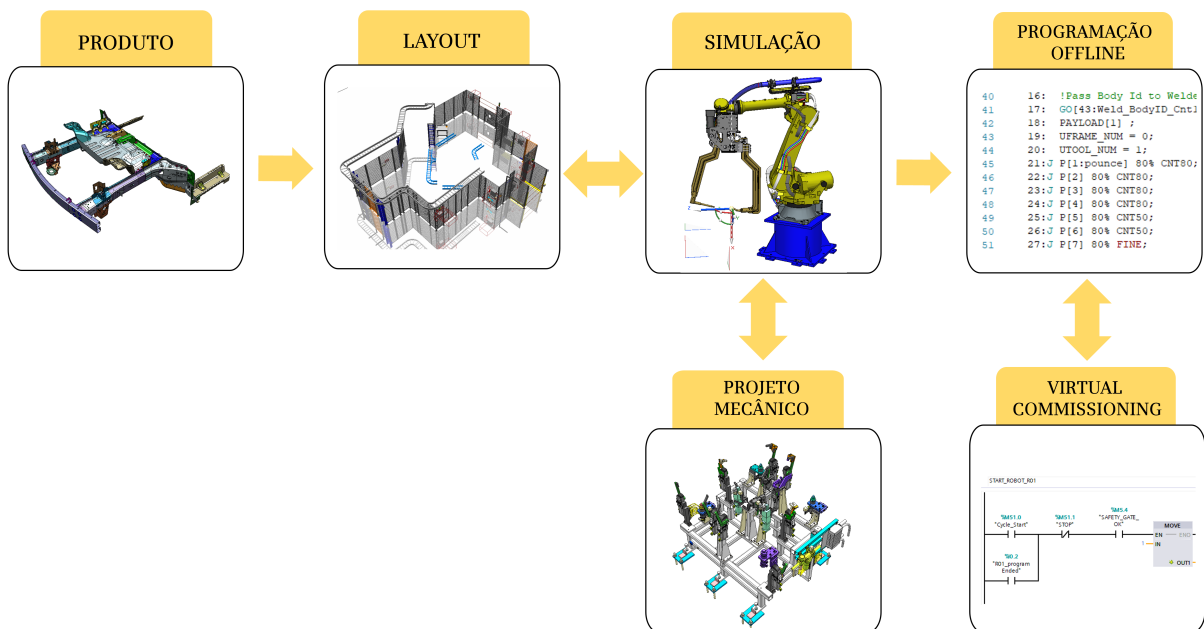


Fonte: SIEMENS, 2024.

### 3 METODOLOGIA

Levando em consideração os objetivos definidos para este trabalho, uma sequência de processos precisa ser definida e apresentada para a compreensão das atividades desenvolvidas. A elaboração do projeto de uma linha de fabricação automotiva demanda uma série de etapas tanto sequenciais quanto concomitantes, envolvendo equipes multidisciplinares de engenharia mecânica, elétrica e automação. A Figura 12 mostra um recorte do fluxo de trabalho típico envolvido no projeto e na concepção de uma linha de manufatura de veículos, nesse caso específico, com ênfase na simulação em ambiente virtual do processo de fabricação de carrocerias. Embora esses processos possam variar entre diferentes montadoras e integradoras, eles proporcionam uma boa compreensão das etapas envolvidas nesse trabalho.

Figura 12 – Fluxo de processos para simulação virtual.



Fonte: Elaborado pelo autor, 2024.

O desenvolvimento da simulação virtual inicia-se ao receber o modelo 3D do produto a ser fabricado, juntamente com a definição de quais processos estarão envolvidos em sua fabricação, tais como pontos de solda, cordões de cola e adesivo. A partir dessas informações, o *layout* industrial da linha de fabricação pode ser desenvolvido considerando limitações de espaço, estudos de tempo de ciclo, estimativa do número de robôs e outros equipamentos.

Esse *layout* preliminar é utilizado durante a etapa de simulação para definir a posição dos robôs e de outros equipamentos, possibilitando dessa maneira a realização de testes de alcance dos manipuladores robóticos. A troca de informações entre essas duas etapas é constante, almejando uma solução que seja factível em relação às limitações de instalação na planta produtiva e ao êxito de execução dos processos de fabricação automatizados. O processo de simulação também interage diretamente com a etapa de desenvolvimento do projeto mecânico. Isso é fundamental

pois o projeto das estações de abastecimento e solda, por exemplo, deve evitar interferências com as trajetórias dos robôs.

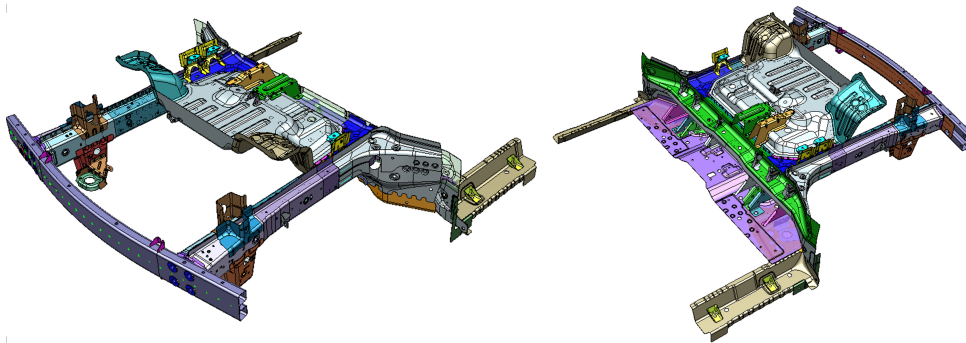
A fase de programação *offline* ocorre após a definição de todos os elementos físicos da simulação e se relaciona com o processo de comissionamento virtual. Dessa forma, a programação dos robôs é desenvolvida e a lógica validada através da integração do *software* de simulação com o CLP.

Nas seções seguintes esses processos serão apresentados com ênfase nas atividades realizadas para a elaboração do presente trabalho.

### 3.1 Produto e processo de fabricação

Na Figura 13 é possível visualizar o produto cujo processo de manufatura foi simulado. Este conjunto é chamado na indústria automotiva de *Spider*, caracterizando-se como uma sub parte de um conjunto maior, denominado de *Motor Compartment*.

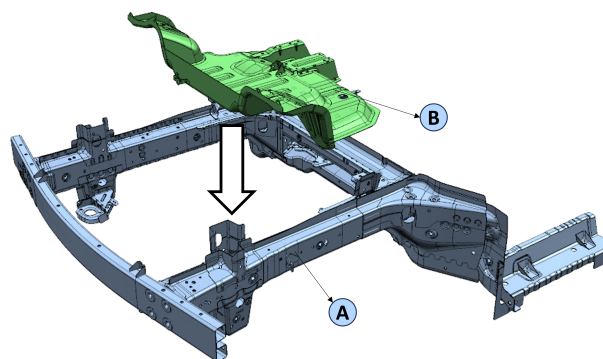
Figura 13 – Produto cujo processo de manufatura foi simulado.



Fonte: Elaborado pelo autor, 2024.

Embora o *Spider* seja formado por diversas peças individuais, a estação automática simulada é responsável por unir os dois sub-conjuntos A e B apresentados na Figura 14.

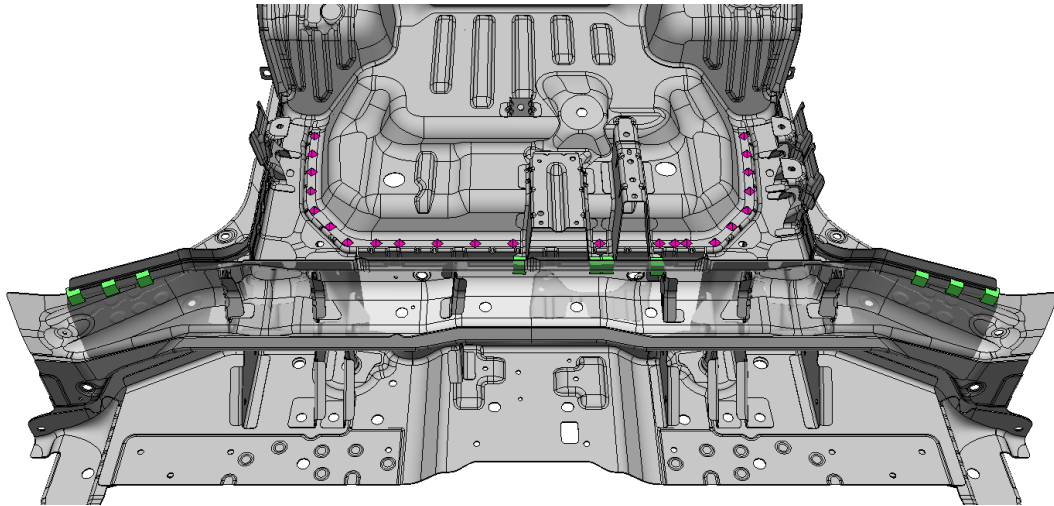
Figura 14 – Sub conjuntos que formam o *Spider*.



Fonte: Elaborado pelo autor, 2024.

A união das peças A e B é realizada através do processo de soldagem por resistência elétrica, utilizando-se manipuladores robóticos equipados com pinças de solda. Na Figura 15, os pontos de solda necessários para a junção das duas peças são indicados. As áreas destacadas na cor magenta são pontos de geometria. Isso significa que é necessário soldar as regiões indicadas - em uma mesma estação e com diversos pontos de fixação na peça - para garantir a estabilidade do conjunto, alinhamento e outras características específicas relacionadas a qualidade do produto. Em contrapartida, as regiões realçadas na cor verde são pontos de complementação (*respot*). Isto é, esses pontos podem ser soldados em estações diferentes dos demais e não requerem múltiplos pontos de fixação no produto, tendo em vista que a geometria da peça já foi estabelecida previamente (WILLIAMS; APPLETON, 1987).

Figura 15 – Pontos de solda do produto: Geometria e *Respot*.



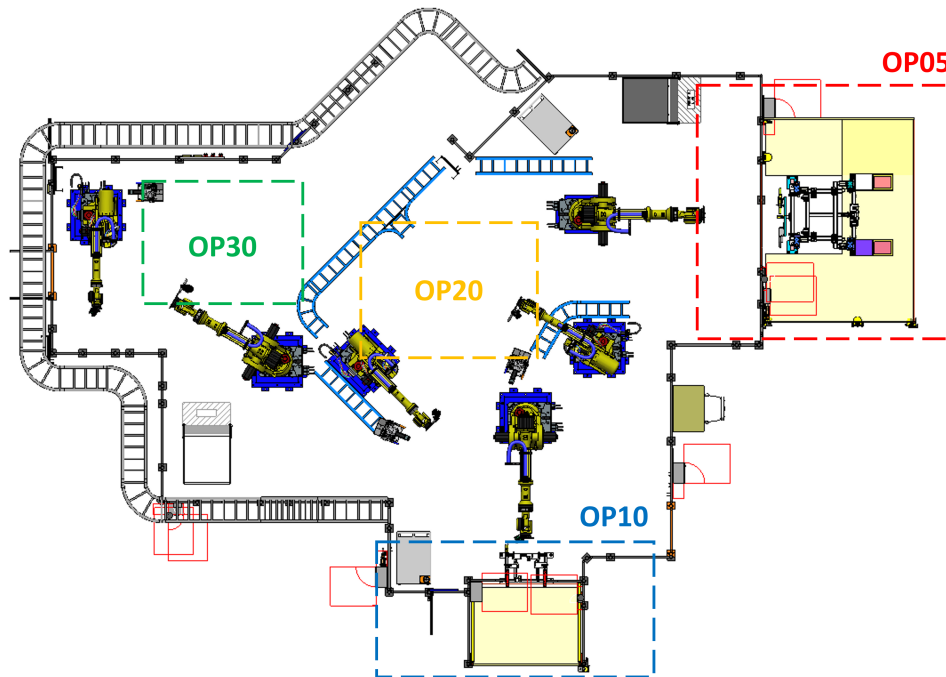
Fonte: Elaborado pelo autor, 2024.

### 3.2 *Layout* e fluxo dos processos

A simulação da estação automática de solda do presente trabalho também considerou o espaço físico da planta em que o produto é fabricado. É comum que o *layout* de uma célula robotizada represente cercas de proteção, eletrocalhas, portões de acesso, equipamentos de segurança, painéis elétricos, robôs e seus periféricos, tubulações e unidades de preparação de ar comprimido, entre outros. O escopo da simulação proposta não possui um *layout* tão rico em detalhes, se atendo apenas aos elementos importantes para a compreensão do processo. Na Figura 16 é possível visualizar o espaço disponível para o projeto e uma definição inicial das operações que compõe a linha automática.

A operação 05 (OP05) foi projetada para o abastecimento do conjunto A indicado na Figura 14, sendo composta por uma mesa para abastecimento do produto, *scanners* a laser e barreira de luz. Essa barreira tem como objetivo identificar fisicamente se o robô responsável pela coleta do produto está dentro da área de abastecimento, enquanto que os *scanners* a laser

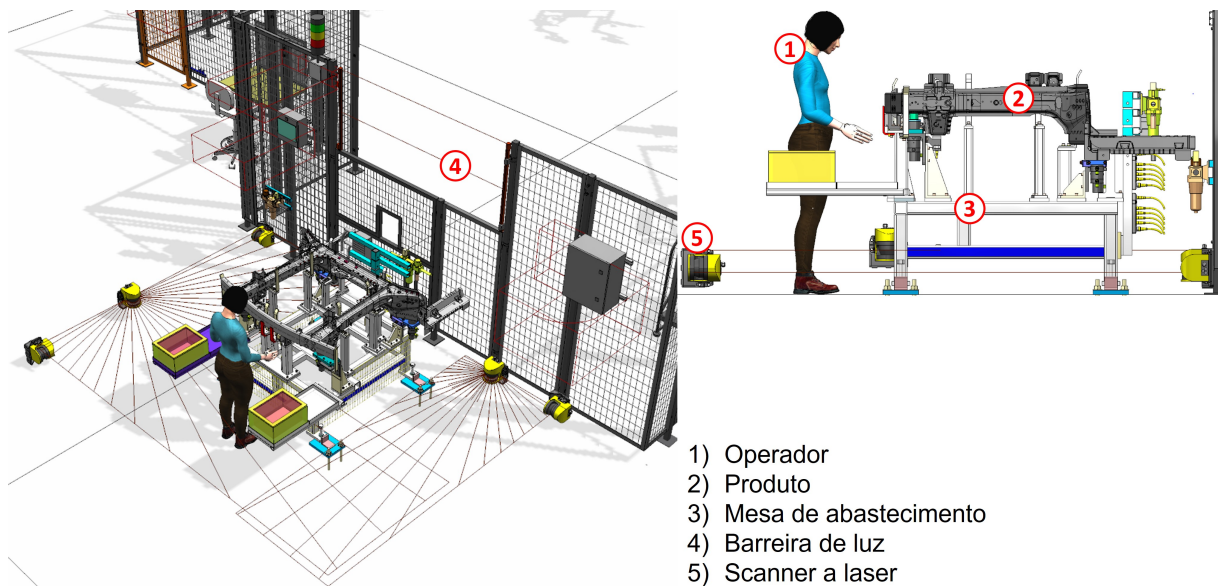
Figura 16 – *Layout* inicial com definição das operações.



Fonte: Elaborado pelo autor, 2024.

garantem o monitoramento de presença do operador nessa área. Os elementos mencionados estão indicados na Figura 17.

Figura 17 – Elementos da primeira estação de abastecimento: OP05.

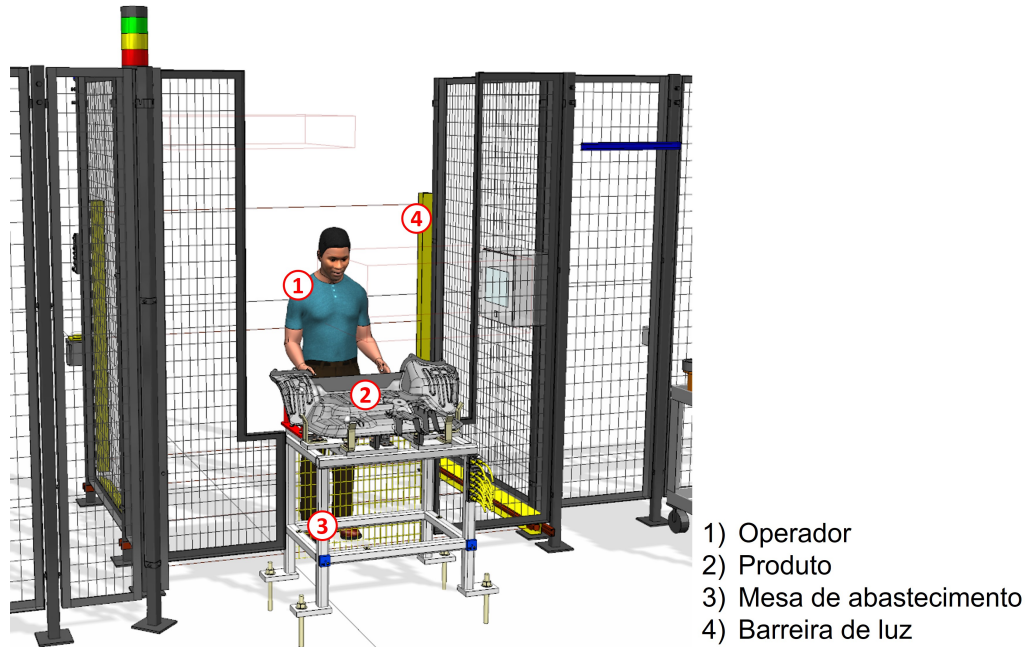


Fonte: Elaborado pelo autor, 2024.

Na Figura 18 é apresentada a segunda operação de abastecimento manual OP10, responsável pelo carregamento do conjunto B que compõe o *Spider*. Tal como na OP05, essa operação

possui uma mesa de abastecimento, porém dispõe apenas de barreiras de luz para garantir o monitoramento do operador.

Figura 18 – Elementos da segunda estação de abastecimento: OP10.



Fonte: Elaborado pelo autor, 2024.

Após o abastecimento das peças nas estações apresentadas anteriormente, elas seguem para a estação de solda OP20, onde os pontos de geometria são soldados por dois robôs. Posteriormente, o produto tem seus pontos de *respot* soldados na OP30 e logo em seguida é retirado da estação automática. Esses processos são abordados com mais abrangência nas próximas seções.

### 3.3 Simulação

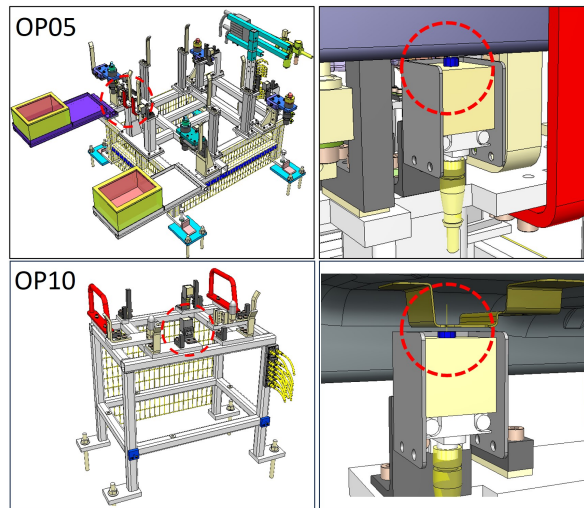
O processo de simulação da estação automática teve início após a definição do produto a ser produzido, processos de fabricação envolvidos e *layout* do espaço físico disponível na fábrica. Como indicado na Figura 12, as etapas de simulação e projeto mecânico ocorrem simultaneamente devido a constante necessidade de alterar o projeto para evitar interferências com as trajetórias dos robôs. Para que seja possível descrever as atividades realizadas neste trabalho de forma sequencial e sem exceder seu escopo, presume-se que o projeto mecânico dos dispositivos, pinças de solda, esteira e garras dos robôs esteja pronto e disponível para realizar as atividades de simulação.

#### 3.3.1 Equipamentos simulados: criação de sensores e cinemática

As estações de abastecimento OP05 e OP10 foram apresentadas anteriormente, assim como suas respectivas mesas de carregamento. Esses equipamentos não possuem partes móveis, como grampos pneumáticos ou guias lineares, pois sua finalidade consiste apenas em fornecer

suporte ao produto. Diante disso, a preparação desses componentes para simulação envolve exclusivamente criar os sensores de presença de peça e seus respectivos sinais, que serão utilizados futuramente como entradas no CLP. A Figura 19 mostra os dispositivos das operações OP05 e OP10 e os sensores correspondentes.

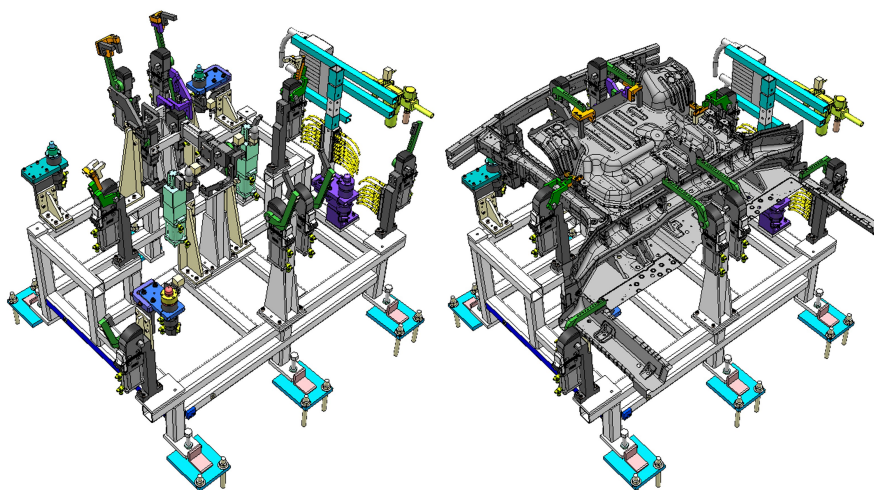
Figura 19 – Sensores de presença de peça das estações de abastecimento: OP05 e OP10.



Fonte: Elaborado pelo autor, 2024.

Conforme mencionado anteriormente, os conjuntos A e B, abastecidos nas estações OP05 e OP10, são soldados na OP20. Ao contrário das mesas da Figura 19, o equipamento que garante o posicionamento das peças durante a soldagem dos pontos de geometria apresenta elementos móveis. Dessa forma, além dos elementos de referimento estáticos, o dispositivo de geometria apresentado na Figura 20 possui 11 grampos pneumáticos e 2 pinos de indexação retráteis.

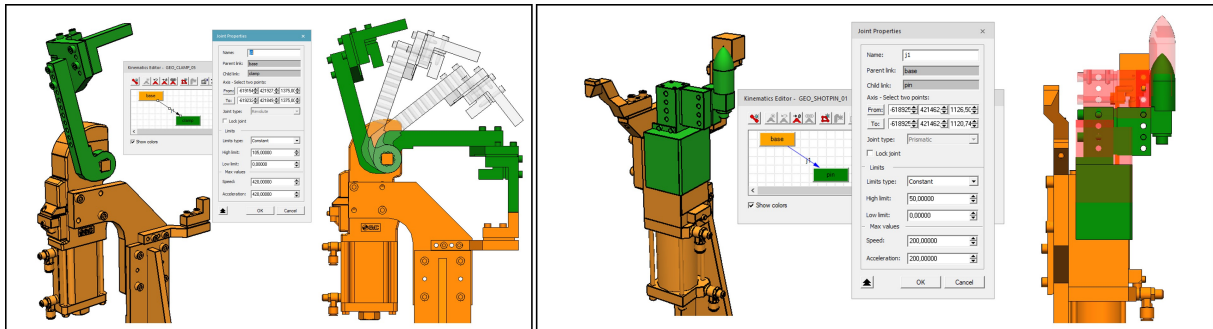
Figura 20 – Dispositivo de solda de geometria da OP20.



Fonte: Elaborado pelo autor, 2024.

Para cada um dos elementos móveis foi realizado o processo de criação de cinemática dentro do *software* de simulação, de forma que pudessem abrir, fechar, avançar e recuar. Na Figura 21 são apresentados dois exemplos de simulação de movimento, à esquerda um grampo com junta de revolução e à direita um pino de localização com junta prismática.

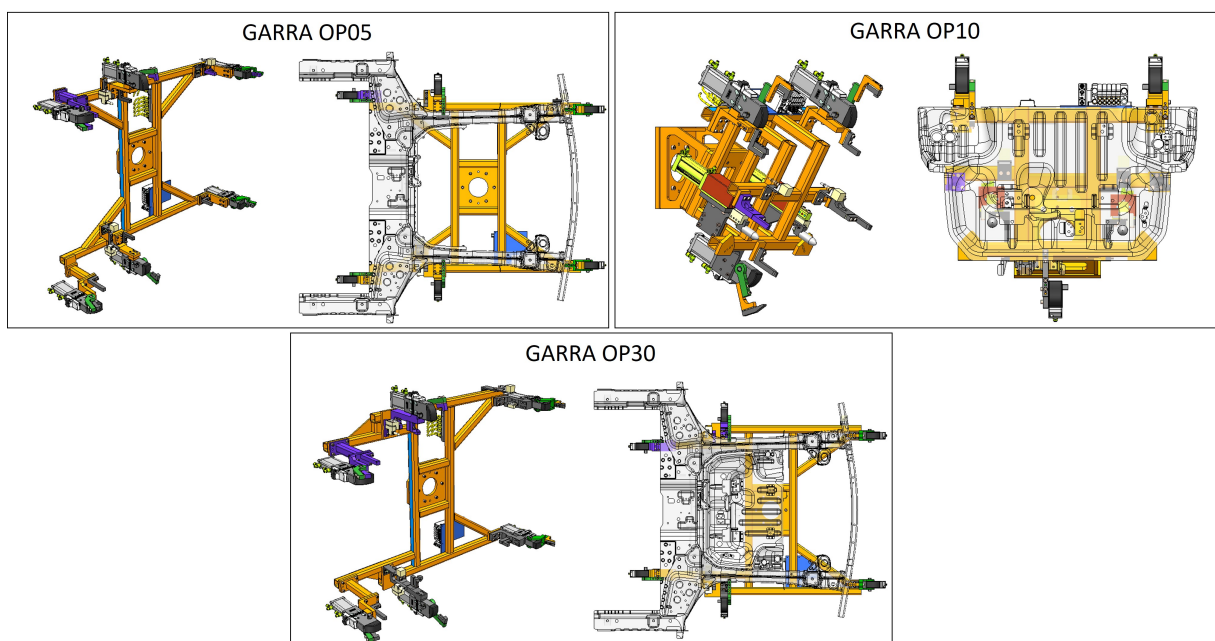
Figura 21 – Exemplo de cinemática criada para os grampos e pinos de localização.



Fonte: Elaborado pelo autor, 2024.

As peças abastecidas nas estações OP05 e OP10 são transportadas para a estação de geometria OP20 por meio de dois robôs distintos, cada um equipado com sua respectiva garra de manipulação. Não somente, após a solda dos pontos de geometria, um terceiro robô de manipulação realiza a pega do *Spider* e dá continuidade ao processo. Sendo assim, a simulação da estação automática possui três garras de manipulação, apresentadas na Figura 22. Esses equipamentos possuem grampos pneumáticos e pinos de localização retráteis e, portanto, também passaram pelo processo de cinemática.

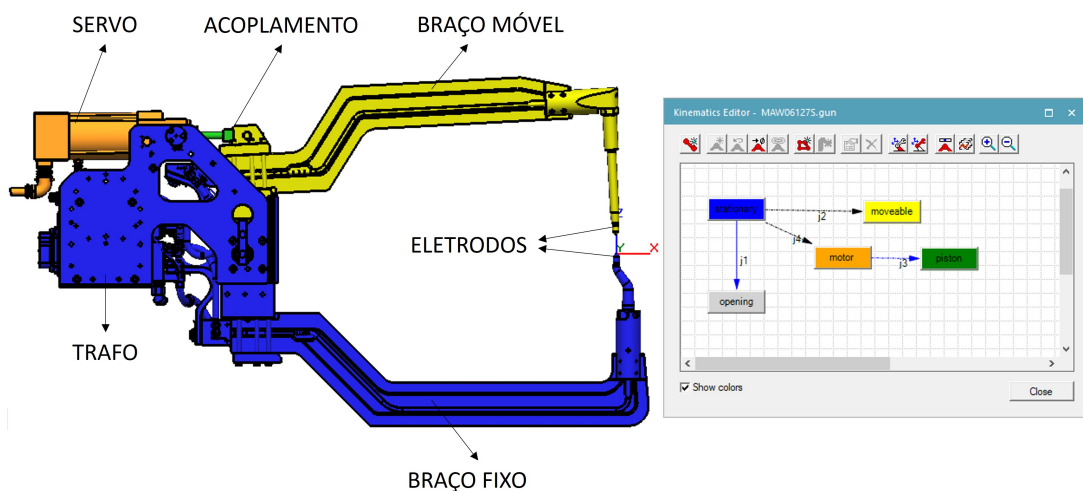
Figura 22 – Garras de manipulação utilizadas na simulação.



Fonte: Elaborado pelo autor, 2024.

As ferramentas dos robôs desse projeto não se limitam às garras, mas englobam também as pinças de solda. Na estação de solda de geometria são utilizadas duas pinças iguais, enquanto que na operação de solda *respot* apenas uma pinça de solda menor se faz presente. De forma resumida, as pinças de solda possuem um braço fixo e um braço móvel com um eletrodo na extremidade de cada braço. O conjunto móvel é acoplado a um servo motor que permite um controle milimétrico da abertura da pinça (WILLIAMS; APPLETON, 1987). De forma similar aos dispositivos apresentados até o momento, a cinemática das pinças de solda também foi criada e é apresentada na Figura 23.

Figura 23 – Pinça de solda da OP20 e sua cinemática.



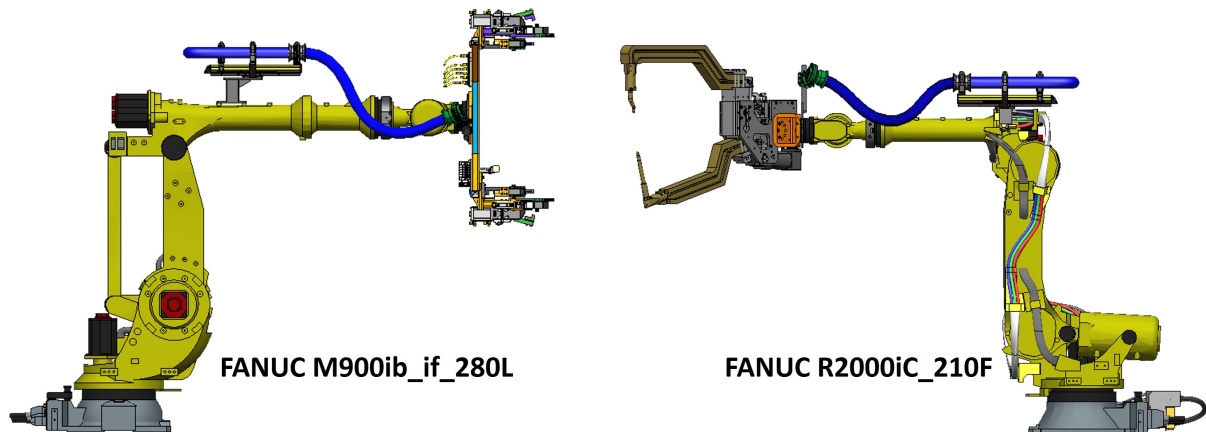
Fonte: Elaborado pelo autor, 2024.

### 3.3.2 Robôs utilizados na simulação

Os manipuladores robóticos são elementos centrais da simulação, uma vez que almeja-se por meio dela a validação de acessos e desenvolvimento das trajetórias dos robôs. Ao contrário dos equipamentos projetados para a estação automática, tais como dispositivos de solda, pinças e garras, o modelo 3D cinematizado dos robôs é fornecido pelo fabricante, isso garante que não exista discrepâncias entre o modelo virtual e o robô real. Nesse projeto, foram utilizados 6 robôs do fabricante Fanuc, sendo três deles para aplicação de solda e outros três para manipulação.

Na Figura 24 é possível visualizar, à esquerda, o modelo de robô utilizado para realizar a manipulação do produto, levando-o de uma operação à outra entre os processos. À direita, por sua vez, é apresentado o robô aplicado nas estações de solda de geometria e de *respot*. A definição do modelo de robô necessário para cada aplicação depende principalmente da carga máxima suportada e alcance dos robôs (CRAIG, 2004).

Figura 24 – Modelos de robôs Fanuc utilizados na simulação.

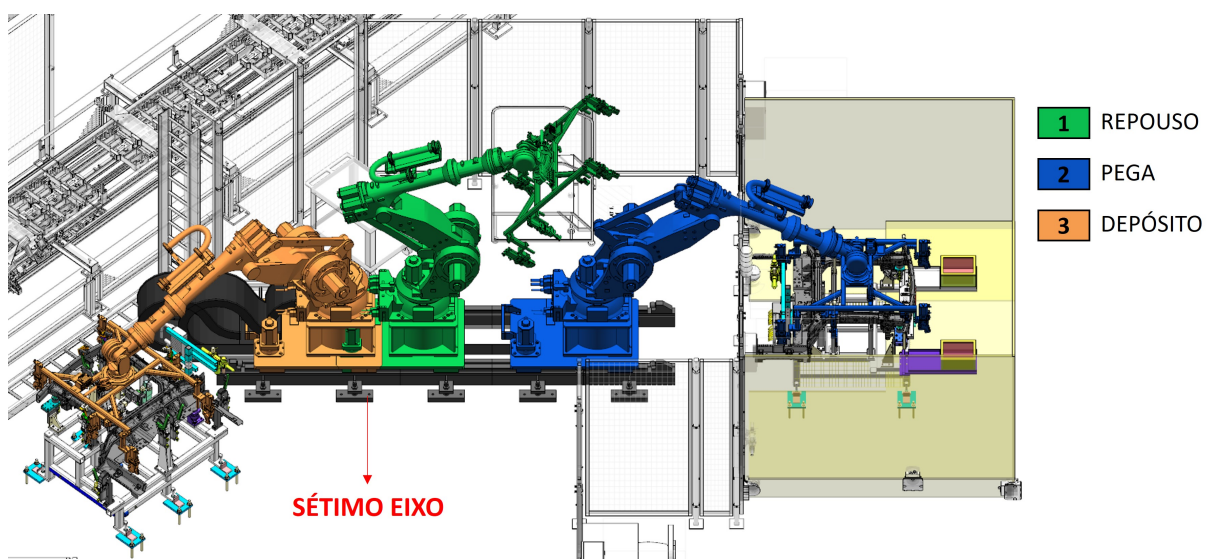


Fonte: Elaborado pelo autor, 2024.

### 3.3.3 Trajetórias dos robôs

O desenvolvimento das trajetórias dos robôs foi realizado após a definição de posicionamento dos dispositivos de cada operação e montagem das ferramentas nos manipuladores. Entende-se como trajetória, nesse contexto, os movimentos que o manipulador robótico realiza para sair de uma posição inicial até atingir uma posição final (NIKU, 2020). Normalmente define-se uma posição de repouso para o robô livre de qualquer interferência com outro manipulador robótico, chamada de *home* ou *start*. A partir dessa posição outros pontos necessários foram criados de acordo com a aplicação de cada robô. Para os robôs de manipulação, além da posição de repouso, são necessárias duas outras posições: pega e depósito.

Figura 25 – Robô 05R01 em diferentes posições: repouso, pega e depósito.



Fonte: Elaborado pelo autor, 2024.

A Figura 25 ilustra o robô da OP05 nas três posições citadas anteriormente. Utilizando

essas três posições como referência, foram criadas posições intermediárias para evitar movimentos extensos e colisões durante a pega/depósito do produto, além de desviar do trajeto de robôs próximos. Durante os testes de alcance identificou-se que o manipulador não seria capaz de alcançar as estações de abastecimento e de geometria, por esse motivo o robô 05R01 foi montado sobre o sétimo eixo de 3 metros indicado na Figura 25. Esse equipamento possibilitou o aumento do alcance e do envelope de trabalho do robô.

Cada um dos movimentos criados possui atributos a eles associados, tais como nome, tipo de movimento (linear ou *joint*), velocidade, *frames* de referência para a movimentação, configurações de eixo e precisão. Algumas dessas características influenciam diretamente no tempo que o robô despense para executar uma atividade. Na Figura 26 é possível visualizar os atributos da trajetória de pega do robô 05R01 dentro do *software* de simulação.

Figura 26 – Atributos de movimentos do robô 05R01.

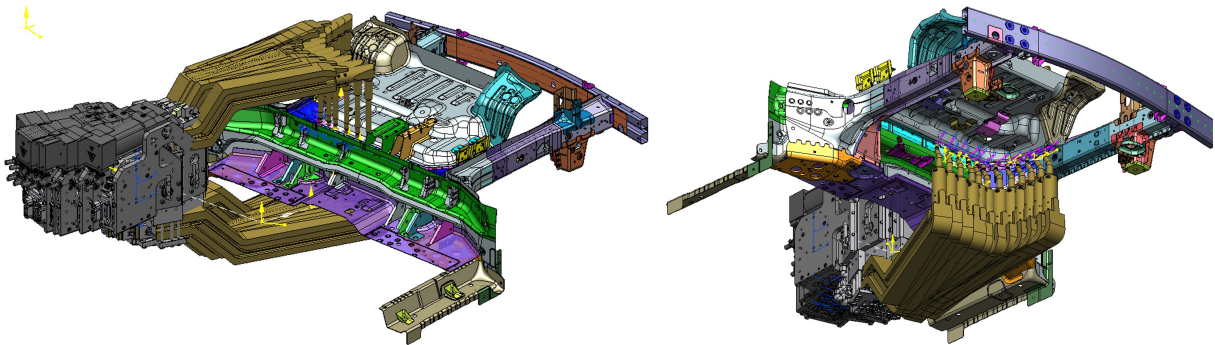
Paths & Locations	Comment	Config	Utool	Uframe	Motion Type	Speed	Term type	Def Type
s01pick1	s01pick1					0.00	-	
└─ pounce		N U T,0,0,0	1tcp_pin	0-WorldFrame	Joint	100 %	FINE	Cartesian
└─ via2		N U T,0,0,0	1tcp_pin	0-WorldFrame	Joint	100 %	CNT80	Cartesian
└─ via3		N U T,0,0,0	1tcp_pin	0-WorldFrame	Joint	80 %	CNT80	Cartesian
└─ pick1_app	pick1_app	N U T,0,0,0	1tcp_pin	0-WorldFrame	Linear	400 mm/sec	FINE	Cartesian
└─ pick1	pick1	N U T,0,0,0	1tcp_pin	0-WorldFrame	Linear	400 mm/sec	FINE	Cartesian
└─ pick1_dep	pick1_dep	N U T,0,0,0	1tcp_pin	0-WorldFrame	Linear	400 mm/sec	CNT50	Cartesian
└─ via4		N U T,0,0,0	1tcp_pin	0-WorldFrame	Joint	80 %	CNT80	Cartesian
└─ via5		N U T,0,0,0	1tcp_pin	0-WorldFrame	Joint	100 %	CNT80	Cartesian
└─ via6		N U T,0,0,0	1tcp_pin	0-WorldFrame	Joint	100 %	CNT80	Cartesian
└─ via7		N U T,0,0,0	1tcp_pin	0-WorldFrame	Joint	100 %	FINE	Cartesian
└─ via8		N U T,0,0,0	1tcp_pin	0-WorldFrame	Joint	100 %	CNT80	Cartesian
└─ via9		N U T,0,0,0	1tcp_pin	0-WorldFrame	Joint	80 %	FINE	Cartesian

Fonte: Elaborado pelo autor, 2024.

O desenvolvimento de trajetórias para os robôs de solda se difere do anterior, uma vez que não possui posições de pega e depósito. Para essa aplicação, as trajetórias foram criadas utilizando os pontos de solda como posições de referência. O robô 20R01, por exemplo, possui 12 pontos de solda e sua nuvem de pinça é apresentada na Figura 27.

A nuvem de pinça possibilita visualizar a ferramenta de solda do robô em todos os locais do produto que são soldados, a partir dessas posições movimentos intermediários foram criados possibilitando o manipulador robótico acessar todos os pontos de solda sem colidir com o produto ou com o dispositivo de geometria. As *vias* das trajetórias de solda possuem outros atributos específicos para essa aplicação, indicados na Figura 28. Em cada um dos movimentos foi necessário declarar o valor de abertura da pinça, para os robôs Fanuc esse valor é negativo e medido em mm. Nos pontos de solda também foi necessário incluir os valores SD (*Start distance*

Figura 27 – Nuvem de pinça do robô 20R01.



Fonte: Elaborado pelo autor, 2024.

- Distância inicial) e ED (*End distance* - Distância final), que representam respectivamente a distância de aproximação e saída do eletrodo no local de solda. O tempo necessário para soldar cada ponto, medido em segundos, foi estipulado a partir da experiência do autor e inserido como atributo dos pontos de solda no *software* de simulação.

Figura 28 – Atributos de movimentos no programa de solda do robô 020R01.

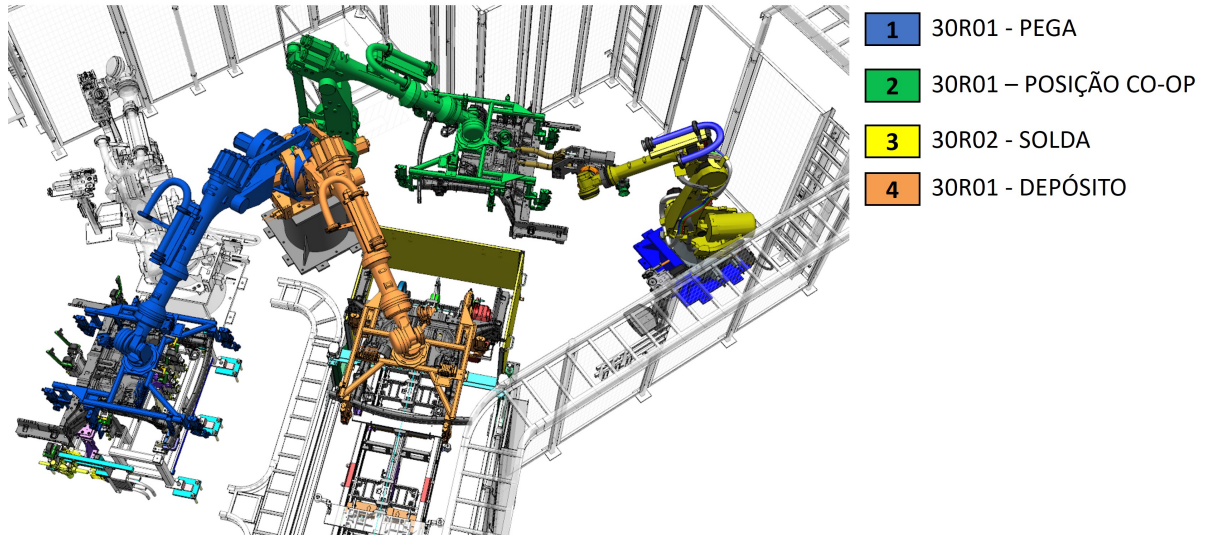
Path Editor - 20R01												
Paths & Locations	Comment	Servo Value	SD	ED	Weld Time	Config	Utool	Uframe	Motion Type	Speed	Tem type	Def Type
s01proc1	s01proc1											
pounce	pounce	-200 mm				F U T,0.0,0	1tcp_gun	0-WorldFrame	Joint	80 %	CNT80	Cartesian
via3		-250 mm				F U T,0.0,0	1tcp_gun	0-WorldFrame	Joint	80 %	CNT80	Cartesian
via4		-250 mm				F U T,0.0,0	1tcp_gun	0-WorldFrame	Joint	80 %	CNT80	Cartesian
via5		-250 mm				F U T,0.0,0	1tcp_gun	0-WorldFrame	Joint	80 %	CNT80	Cartesian
via6		-250 mm				F U T,0.0,0	1tcp_gun	0-WorldFrame	Joint	80 %	CNT50	Cartesian
via7		-250 mm				F U T,0.0,0	1tcp_gun	0-WorldFrame	Joint	80 %	CNT50	Cartesian
via8	enter_izone1	-250 mm				F U T,0.0,0	1tcp_gun	0-WorldFrame	Joint	80 %	FINE	Cartesian
via9		-250 mm				F U T,0.0,0	1tcp_gun	0-WorldFrame	Joint	80 %	CNT80	Cartesian
via10		-250 mm				F U T,0.0,0	1tcp_gun	0-WorldFrame	Joint	80 %	CNT80	Cartesian
via11		-150 mm				F U T,0.0,0	1tcp_gun	0-WorldFrame	Joint	80 %	CNT80	Cartesian
via12		-150 mm				F U T,0.0,0	1tcp_gun	0-WorldFrame	Joint	80 %	CNT80	Cartesian
12A1200	12A1200	-30 mm	15 (15+15=30)	15 (15+15=30)	0.48	F U T,0.0,0	1tcp_gun	0-WorldFrame	Linear	800 mm/sec	CNT100	Cartesian
12A1201	12A1201	-30 mm	15 (15+15=30)	15 (15+15=30)	0.48	F U T,0.0,0	1tcp_gun	0-WorldFrame	Linear	800 mm/sec	CNT100	Cartesian
via10		-150 mm				F U T,0.0,0	1tcp_gun	0-WorldFrame	Joint	80 %	CNT80	Cartesian
via11	exit_izone1	-150 mm				F U T,0.0,0	1tcp_gun	0-WorldFrame	Linear	1600 mm/...	FINE	Cartesian
12A1202	12A1202	-30 mm	15 (15+15=30)	15 (15+15=30)	0.48	F U T,0.0,0	1tcp_gun	0-WorldFrame	Linear	800 mm/sec	CNT100	Cartesian
12A1203	12A1203	-30 mm	15 (15+15=30)	15 (15+15=30)	0.48	F U T,0.0,0	1tcp_gun	0-WorldFrame	Linear	800 mm/sec	CNT100	Cartesian

Fonte: Elaborado pelo autor, 2024.

A operação OP30, por sua vez, é composta por um robô de manipulação (30R01) e um robô de solda (30R02). Nesta estação os pontos de *respot* apresentados na Figura 15 são soldados, porém ao contrário da OP20, sem a presença de um dispositivo no piso para realizar o apoio e a fixação do produto. O processo realizado por esses robôs é comumente chamado na indústria de *CO-OP*, pois baseia-se na cooperação entre dois robôs para realizar uma determinada tarefa. Na Figura 29 é possível visualizar que o ciclo do robô de manipulação consiste em coletar o produto na estação de geometria e direcionar-se para a posição de soldagem. Após esse posicionamento, o robô 30R02 realiza a solda dos pontos de complementação enquanto o robô 30R01 segura e

estabiliza o *Spider* com sua garra. Posteriormente, a peça é depositada em uma esteira e segue para outra estação automática de solda.

Figura 29 – Ciclo dos robôs 30R01 e 30R02.



Fonte: Elaborado pelo autor, 2024.

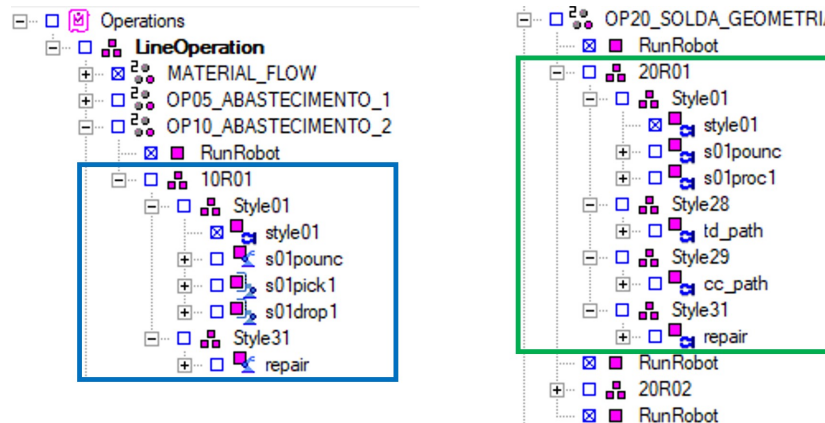
### 3.4 Programação *Offline* dos robôs

As trajetórias criadas para cada robô são divididas em programas de acordo com o padrão de programação adotado por cada montadora. No presente trabalho, adotou-se o padrão de programação para robôs Fanuc utilizado em todas as fábricas da *General Motors*, denominado *GM-global*. Ainda seguindo essa estrutura, os programas foram divididos em *styles* (modelos). Os *styles* podem representar diferentes números de modelos de carro que são produzidos em uma determinada linha, além disso, existem também *styles* reservados para programas de manutenção, fresagem e troca de eletrodo da pinça de solda. Para a peça fabricada na estação simulada, utilizou-se o código 1 e conseqüentemente a nomenclatura dos programas de trabalho do robô inclui o prefixo "s01" (*style 1*).

A Figura 30 foi obtida do Tecnomatix Process Simulate e mostra como as trajetórias dos robôs 10R01 (manipulação) e 20R01 (solda) foram divididas em programas, organizadas e nomeadas. Ambos os robôs possuem os programas *style01*, *s01pounc* e *repair*, que podem ser definidos como se segue:

- **Programa *style01***: recebe do CLP o código de modelo a ser fabricado e realiza a chamada dos outros programas de acordo com o ciclo definido para o robô.
- **Programa *s01pounc***: o robô se desloca da posição de repouso para uma posição estratégica, sem interferir com outros componentes da célula e próxima à estação de trabalho. Em seguida, aguarda um sinal do CLP para iniciar o ciclo.

Figura 30 – Organização, divisão e nomenclatura de programas dos robôs 10R01 e 20R01.



Fonte: Elaborado pelo autor, 2024.

- **Programa *repair***: programa de manutenção. Ao receber do CLP o código 31 os robôs são direcionados para uma posição ergonômica e fora de interferência com outros equipamentos, permitindo que um profissional de manutenção realize posteriormente reparos ou ajustes nas ferramentas do robôs.

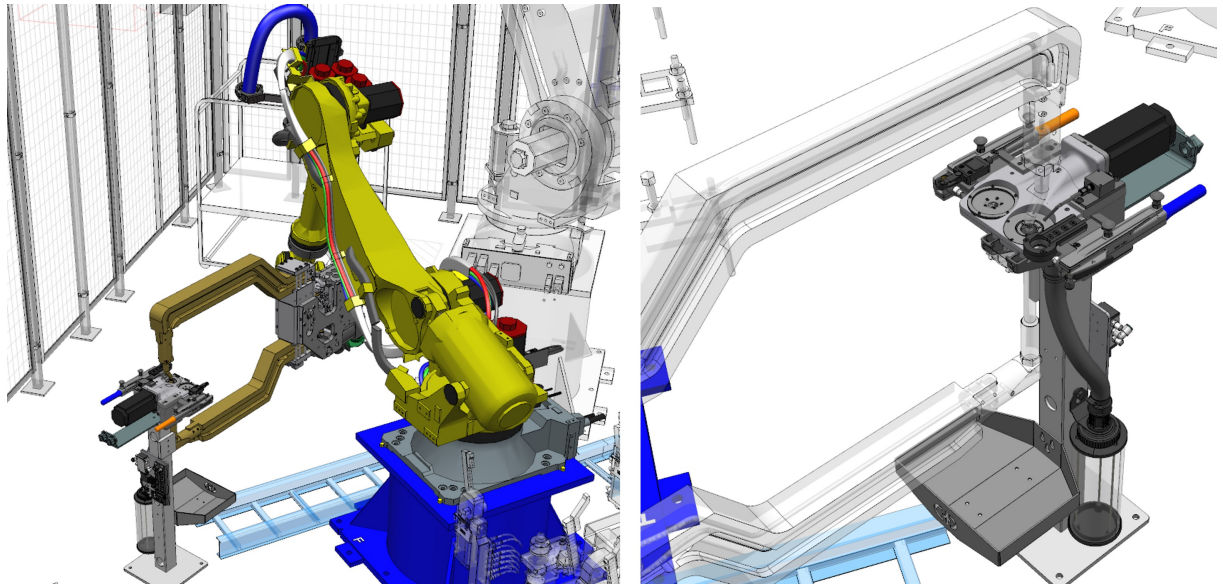
Existem, contudo, programas que são específicos para cada aplicação. A seguir, são definidos os programas específicos para os robôs de manipulação.

- **Programa *s01pick1***: Programa de coleta do produto. O robô desloca-se da posição *pounce* para a posição de coleta da peça, finalizando sua trajetória em uma posição fora de interferência com outros equipamentos, mas próxima ao local de depósito.
- **Programa *s01drop1***: programa de depósito do produto. O robô inicia sua trajetória a partir da última posição definida no programa de pega, deposita a peça e retorna à posição de repouso.

Os robôs de solda diferenciam-se dos robôs de manipulação por possuírem os programas *s01proc1*, *td\_path* e *cc\_path*.

- **Programa *s01proc1***: programa de soldagem. O robô inicia a trajetória a partir da posição de *pounce*, solda o produto e retorna à posição de repouso.
- **Programa *td\_path***: programa de fresagem de eletrodo (*tip dresser*). Ao receber do CLP o código 28 o robô realiza o processo de fresagem de eletrodos da pinça em um equipamento dedicado, conforme ilustrada na Figura 31.
- **Programa *cc\_path***: programa de troca de eletrodos (*cap changer*). Ao receber do CLP o código 29 o robô realiza o processo de troca de eletrodos da pinça em um equipamento dedicado, conforme ilustrado Figura 31.

Figura 31 – Robô 20R01 e equipamento utilizado para fresagem e troca de eletrodos da pinça de solda.



Fonte: Elaborado pelo autor, 2024.

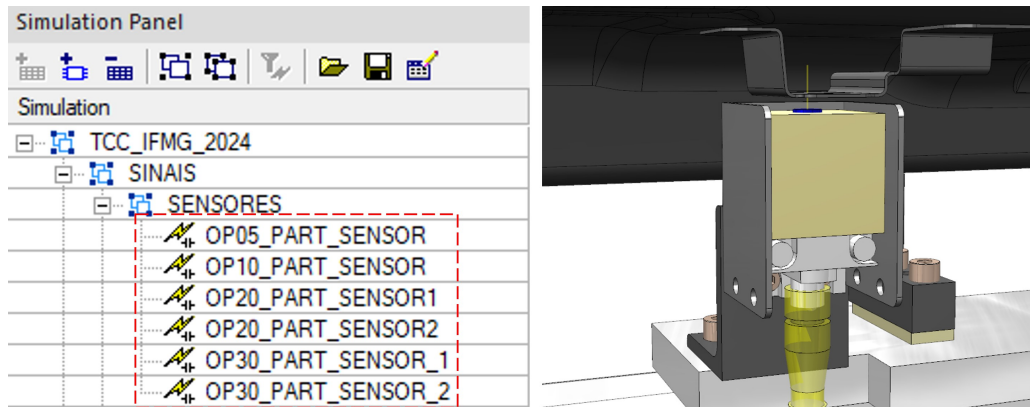
Os programas mencionados são compostos por atributos específicos de cada movimento (definidos durante a fase de simulação), coordenadas cartesianas (que variam conforme o *frame* de referência utilizado para criar as trajetórias) e instruções de programação que são inseridas ao longo dos movimentos dos robôs.

## 3.5 Comissionamento Virtual

### 3.5.1 Criação de sinais e blocos lógicos

Após a finalização da etapa de programação dos robôs, foram criados os sinais essenciais para viabilizar o desenvolvimento da programação do CLP responsável pelo controle da estação automática. O processo simulado inclui 6 sensores de presença de peça, que são utilizados nas mesas de abastecimento e de solda. Não são considerados os sensores das garras de manipulação, pois os mesmos são monitorados pelo controlador do robô. Os 5 sinais de entrada referentes aos sensores são apresentados na Figura 32.

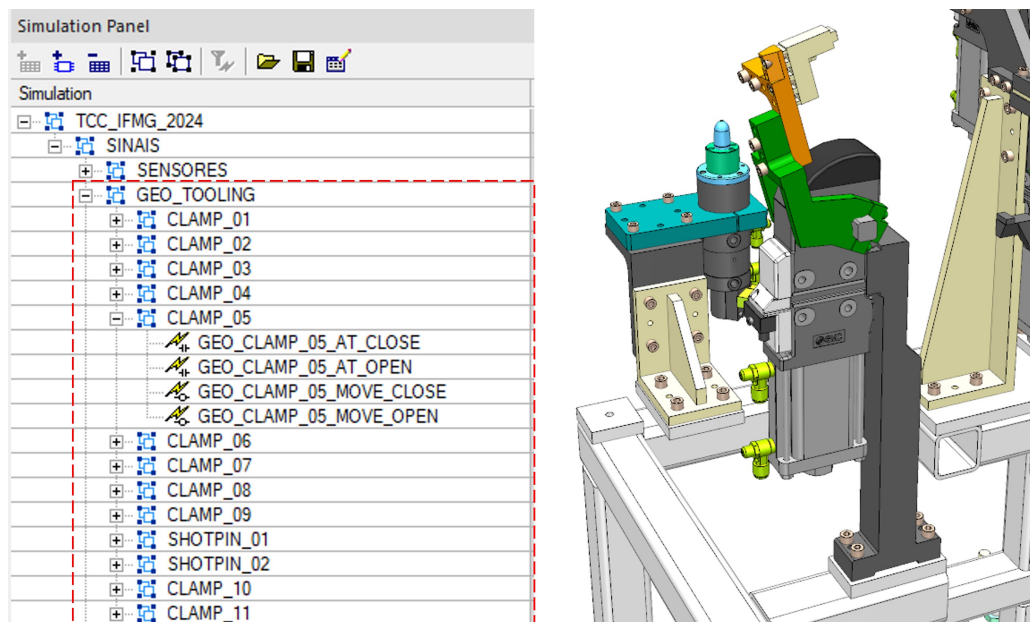
Figura 32 – Sinais de entrada referentes aos sensores de presença de peça.



Fonte: Elaborado pelo autor, 2024.

No que se refere ao dispositivo de geometria da OP20, foram criados 4 sinais para cada parte móvel, sendo dois sinais de entrada e outros dois sinais de saída. Na Figura 33, os sinais de entrada possuem o sufixo *AT\_OPEN* e *AT\_CLOSE*. Eles são provenientes de sensores internos dos grampos e guias lineares, responsáveis por monitorar os estados "aberto/fechado" e "avançado/recuado". Já os sinais de saída são utilizados pelo CLP para comandar as válvulas eletropneumáticas que realizam o acionamento dos grampos.

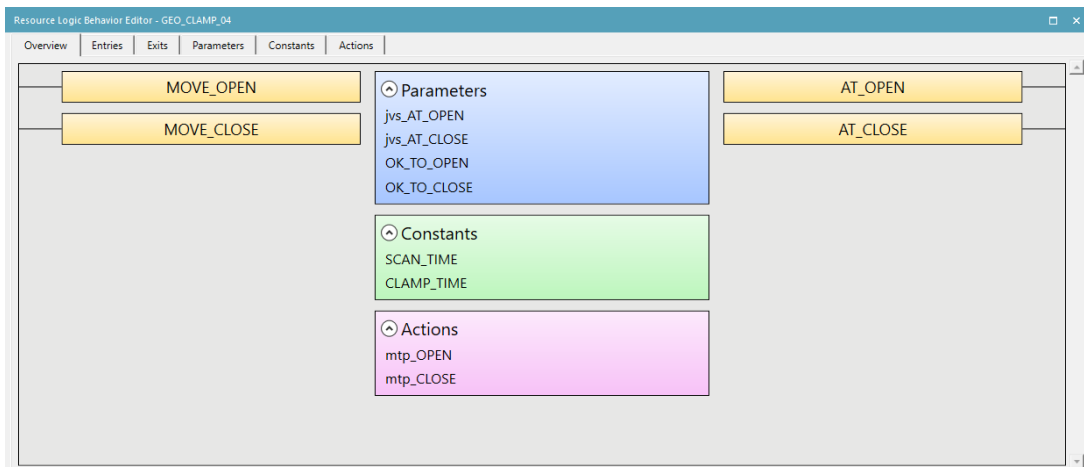
Figura 33 – Sinais para controle do dispositivo de geometria da OP20.



Fonte: Elaborado pelo autor, 2024.

Para que seja possível movimentar os dispositivos na simulação a partir dos sinais do CLP, também foi necessário criar um bloco lógico para cada grampo pneumático, apresentado na Figura 34. Nele são definidos os sensores mencionados anteriormente e a relação de sinais.

Figura 34 – Bloco lógico para controle do dispositivo de geometria da OP20.



Fonte: Elaborado pelo autor, 2024.

Para cada um dos robôs simulados foram criados dois conjuntos de sinais. O primeiro conjunto, destacado na Figura 35 pelo retângulo verde, refere-se aos sinais comuns para todos os robôs e necessários para o funcionamento da simulação dentro do *software* Tecnomatix Process Simulate. O segundo conjunto, destacado em azul, é composto por sinais criados de acordo com a necessidade de cada processo.

Figura 35 – Sinais do robô 20R02: sinais padrão em verde e sinais de processo em azul.

PLC Signal Name	Robot Signal Name	I/O	Signal Function	HW Type	Address	External Connection
OP20R02_startProgram	startProgram	Q	Starting Program	BOOL	No Address	PLC_1
OP20R02_robotReady	robotReady	I	Robot Ready	BOOL	No Address	PLC_1
OP20R02_programPause	programPause	Q	Program Pause	BOOL	No Address	PLC_1
OP20R02_programNumber	programNumber	Q	Program Number	INT	No Address	PLC_1
OP20R02_emergencyStop	emergencyStop	Q	Program Emergency Stop	BOOL	No Address	PLC_1
OP20R02_at_start	start	I	Pose Signal	BOOL	No Address	PLC_1
OP20R02_mirrorProgramNumber	mirrorProgramNumber	I	Mirror Program Number	INT	No Address	PLC_1
OP20R02_emorProgramNumber	errorProgramNumber	I	Error Program Number	BOOL	No Address	PLC_1
OP20R02_programEnded	programEnded	I	Ending Program	BOOL	No Address	PLC_1
OP20R02_Seg63_InPathSeg	OP20R02_Seg63_InPathSeg	I		BOOL	No Address	PLC_1
OP20R02_Seg62_InPathSeg	OP20R02_Seg62_InPathSeg	I		BOOL	No Address	PLC_1
OP20R02_Seg50_InPathSeg	OP20R02_Seg50_InPathSeg	I		BOOL	No Address	PLC_1
OP20R02_Seg1_ContOK	OP20R02_Seg1_ContOK	Q		BOOL	No Address	PLC_1
OP20R02_RequestCont_Clamp10_...	OP20R02_RequestCont_Clamp10_OK	Q		BOOL	No Address	PLC_1
OP20R02_RequestCont_Clamp10	OP20R02_RequestCont_Clamp1	I		BOOL	No Address	PLC_1
OP20R02_Req_Zone1_OK	OP20R02_Req_Zone1_OK	Q		BOOL	No Address	PLC_1
OP20R02_Req_Zone1	OP20R02_Req_Zone1	I		BOOL	No Address	PLC_1

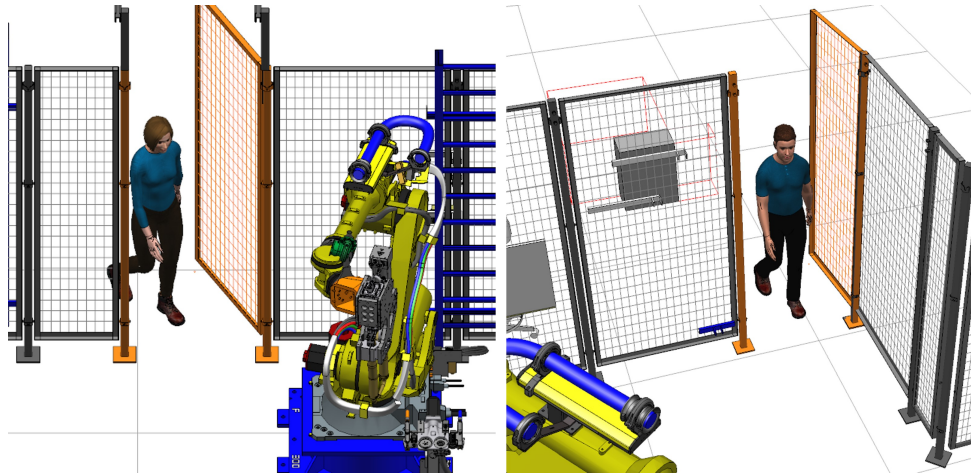
Fonte: Elaborado pelo autor, 2024.

### 3.5.2 Dispositivos de segurança

O comissionamento virtual desenvolvido no presente trabalho também se propôs a monitorar os portões de acesso da estação automática, apresentados na Figura 36. A célula possui dois portões que permitem o acesso de profissionais de manutenção aos equipamentos e robôs. O monitoramento desses portões envolve a verificação de seus sinais para garantir que este-

jam fechados. Caso algum dos portões seja aberto, o processo de fabricação é imediatamente interrompido.

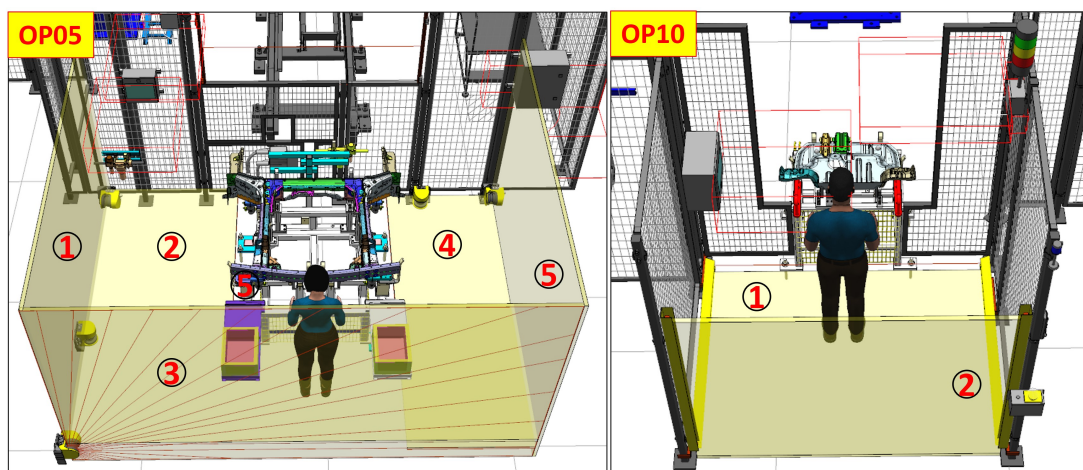
Figura 36 – Portões de acesso da estação automática.



Fonte: Elaborado pelo autor, 2024.

Além da segurança dos portões, áreas de monitoramento de presença do operador nas estações de abastecimento foram configuradas no *software* de simulação. Cada uma dessas áreas, ilustradas na Figura 37, possui um sinal associado. Sempre que o operador está posicionado em uma dessas áreas, o movimento dos robôs em direção ao abastecimento é impedido automaticamente.

Figura 37 – Áreas monitoradas nas estações de abastecimento.



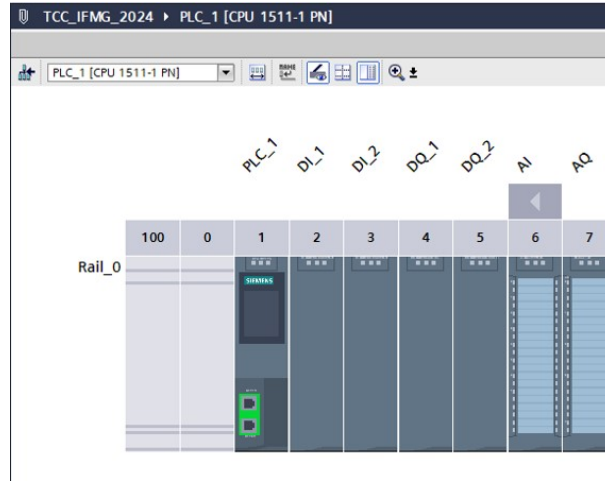
Fonte: Elaborado pelo autor, 2024.

### 3.5.3 Programação do CLP

Para o desenvolvimento do comissionamento virtual optou-se pela abordagem *software-in-the-loop*. Dessa forma, foi realizada a configuração do CLP emulado no *software* TIA Portal. Na Figura 38 é possível visualizar o controlador Siemens Simatic S7-1500, composto pela CPU

1511-1 PN, 2 módulos de entrada digital, 2 módulos de saída digital, 1 módulo de entrada analógica e 1 módulo de saída analógica.

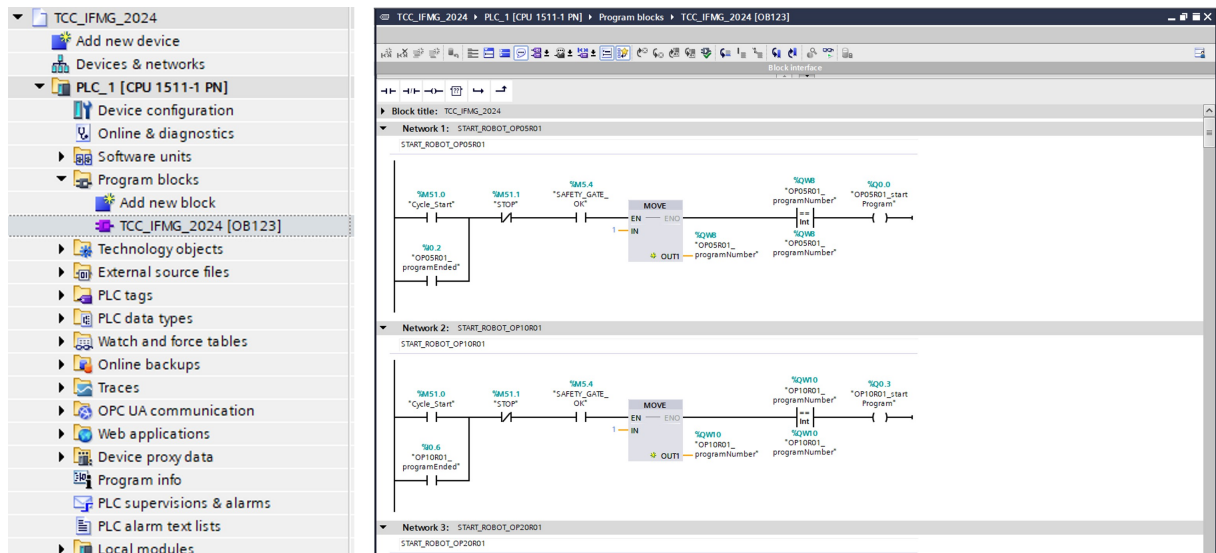
Figura 38 – Configuração do CLP no *software* TIA Portal.



Fonte: Elaborado pelo autor, 2024.

Todos os sinais criados dentro do *software* de simulação foram mapeados no TIA Portal e o programa do CLP desenvolvido na linguagem *Ladder*, utilizando-se apenas um bloco de programação principal subdividido em diferentes *Networks*. A Figura 39 mostra a estrutura mencionada.

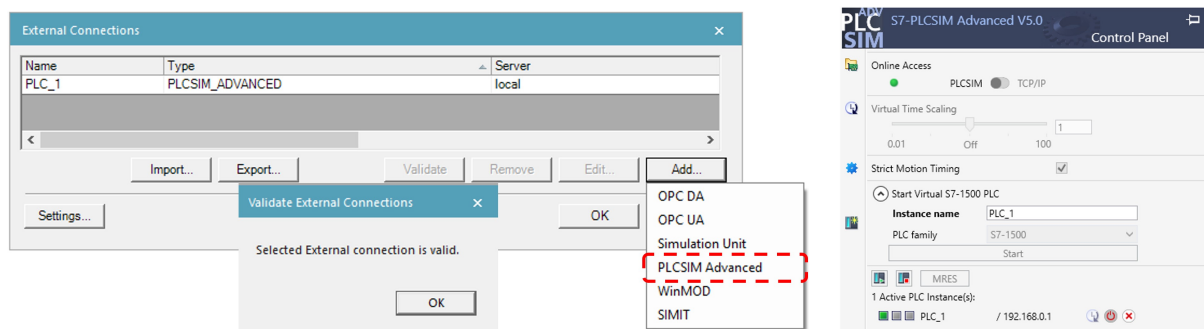
Figura 39 – Estrutura de programação no *software* TIA Portal.



Fonte: Elaborado pelo autor, 2024.

A comunicação entre os dois *softwares* utilizados para o desenvolvimento do projeto, apresentada na Figura 40, foi estabelecida utilizando-se a ferramenta Siemens S7-PLCSIM Advanced V5.0. Isso possibilitou uma conexão direta entre o Tecnomatix Process Simulate e TIA Portal, eliminando a necessidade de protocolos de comunicação como OPC UA ou OPC DA.

Figura 40 – Comunicação entre os *softwares* Tecnomatix Process Simulate e TIA Portal.



Fonte: Elaborado pelo autor, 2024.

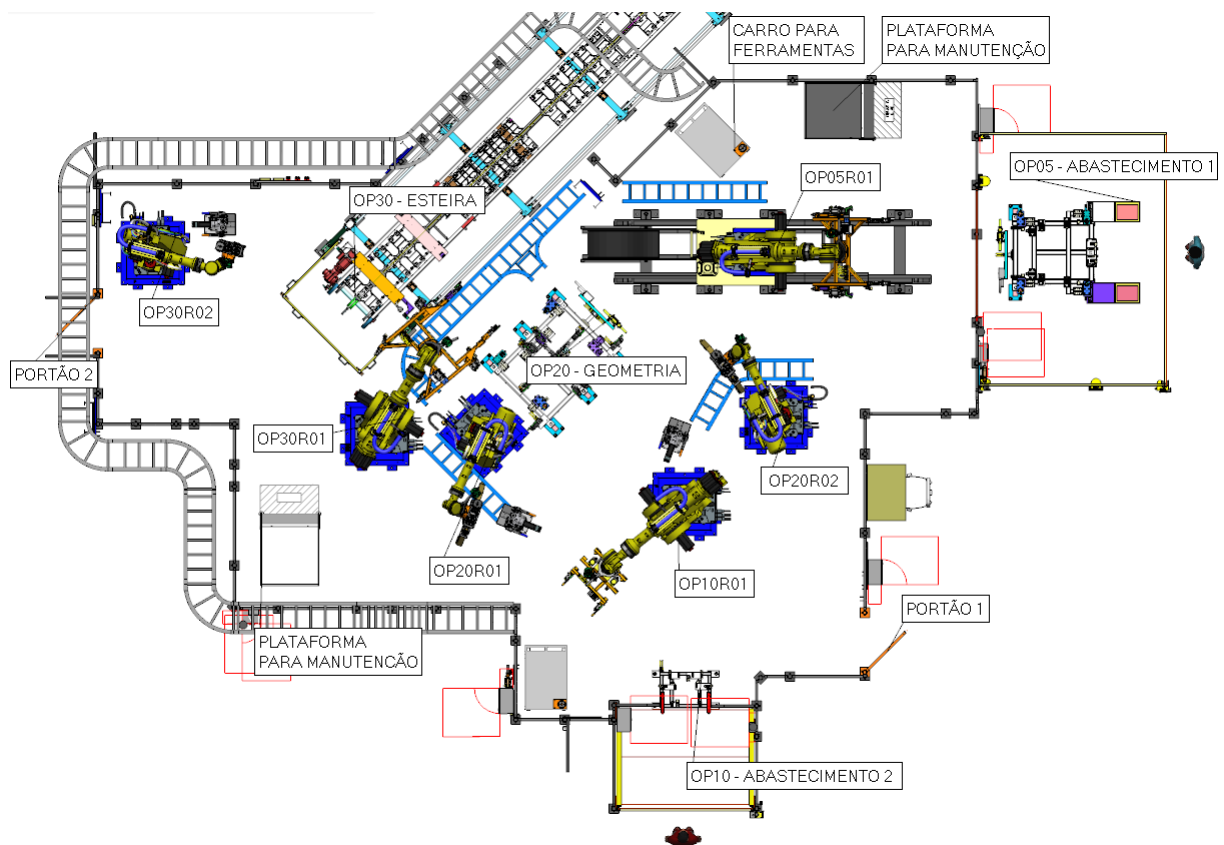
## 4 RESULTADOS

Neste capítulo são apresentados os resultados do desenvolvimento da simulação, programação *offline* e comissionamento virtual da célula robotizada.

### 4.1 Layout

A Figura 41 apresenta o *layout* final da estação automática, obtido após todas as validações de acesso, posicionamento dos robôs e de outros equipamentos. Além dos aspectos relacionados ao processo de manufatura robotizada, percebe-se que elementos relativos à ergonomia dos operadores e profissionais de manutenção também foram considerados, tais como portões de acesso, plataformas de manutenção e espaço para transitar dentro da estação.

Figura 41 – *Layout* final da célula robotizada.

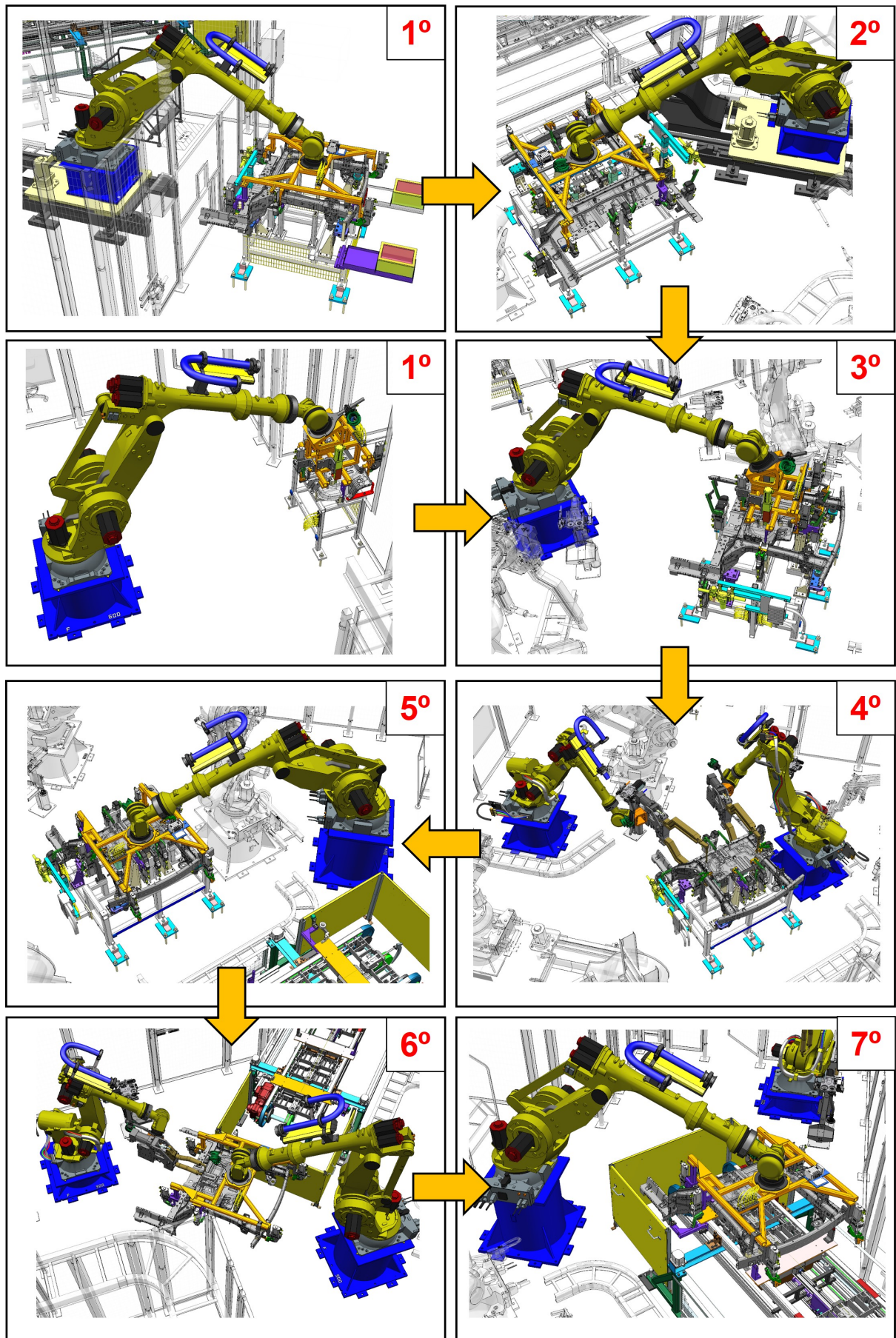


Fonte: Elaborado pelo autor, 2024.

### 4.2 Simulação

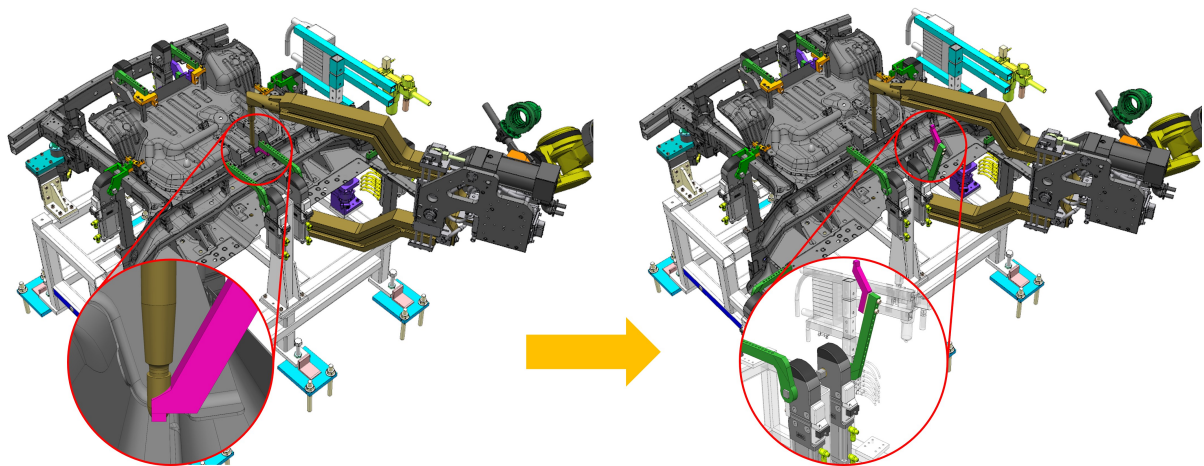
A simulação em ambiente virtual da estação automática possibilitou o desenvolvimento das trajetórias dos robôs de manipulação e de solda, apresentadas em sequência na Figura 42.

Figura 42 – Trajetórias dos robôs apresentadas em sequência.



A criação da cinemática dos dispositivos desempenhou um papel importante nesse processo, possibilitando visualizar o comportamento dos grampos pneumáticos e pinos retráteis em cada processo. Dessa forma, obteve-se movimentos livres de interferências entre os manipuladores robóticos e outros equipamentos. A Figura 43 apresenta um exemplo do que foi mencionado, onde verificou-se que em um determinado ponto soldado pelo robô 20R02 o eletrodo colidia com um dos elementos de estabilização do produto. Diante disso, foi possível inserir na programação do robô uma requisição para abrir o grampo pneumático antes de soldar essa região.

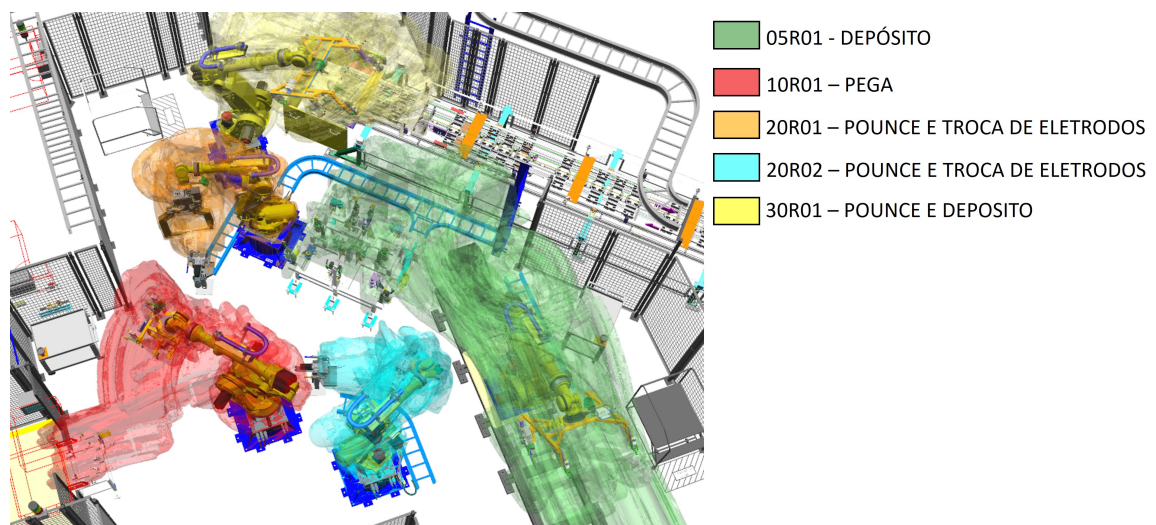
Figura 43 – Interferência mecânica entre eletrodo da pinça de solda do robô 20R02 e dispositivo de geometria.



Fonte: Elaborado pelo autor, 2024.

Além da verificação de interferência com os dispositivos, também foi possível validar as distâncias entre os volumes de trabalho dos robôs e garantir que as trajetórias estivessem livres de colisões, como ilustrado na Figura 44.

Figura 44 – Verificação de colisão entre robôs através de seus volumes de trabalho.



Fonte: Elaborado pelo autor, 2024.

### 4.3 Programação *Offline* dos robôs

O *software* Tecnomatix Process Simulate possibilitou o desenvolvimento da programação dos robôs utilizando um controlador virtual e desconectado do robô físico. Obteve-se dessa etapa todos os programas dos robôs importantes para o processo, considerando seus atributos de movimento, coordenadas cartesianas e instruções de lógica contendo os sinais necessários para os ciclos dos manipuladores robóticos. A Figura 45 apresenta um exemplo desses programas.

Figura 45 – Exemplo de programa do robô obtido através do *software* de simulação.

<pre> 18: !ASSIGN PATH SEGMENT VALUES ; 19: SET SEGMENT(10) ; 20: ; 21: Set MH Tool(1) ; 22: ; 23: !Check for EOAT In Correct ; 24: !State Prepare to Pick(X) ; 25: Prepare to Pick(0) ; 26: ; 27: !MOVE TO TOOL X PICKUP ; 28: ; 29: UFRAME_NUM = 0; 30: UTOOL_NUM = 1; 31: J P[1] 100% FINE; 32: J P[2] 100% CNT80; 33: J P[3] 80% CNT80; 34: L P[4:pickl_app] 400mm/sec FINE; 35: L P[5:pickl] 400mm/sec FINE; 36: Grip Part(1) ; 37: ; 38: PAYLOAD[1] ; 39: ; 40: !MOVE ABOVE PICK POS ; 41: !TO CHECK NO PART IN TOOL ; 42: L P[6:pickl_dep] 400mm/sec CNT50; 43: !CHECK PART IN EOAT ; 44: Part Present(0) ; 45: ; 46: REQUEST CONTINUE ; 47: SET SEGMENT(11) ; 48: ; </pre>	<pre> P[2] {   GP1 :     UF : 0, UT : 1,   CONFIG : 'N U T,0,0,0',     X = 4796.954 mm,  Y = 523.801 mm,   Z = 859.804 mm,     W = -178.101 deg, P = 24.891 deg,   R = -179.726 deg }; P[3] {   GP1 :     UF : 0, UT : 1,   CONFIG : 'N U T,0,0,0',     X = 5352.238 mm,  Y = 489.429 mm,   Z = -197.509 mm,     W = 180.000 deg,  P = 0.000 deg,   R = 180.000 deg }; P[4:"pickl_app"] {   GP1 :     UF : 0, UT : 1,   CONFIG : 'N U T,0,0,0',     X = 5352.238 mm,  Y = 489.429 mm,   Z = -397.509 mm,     W = 180.000 deg,  P = 0.000 deg,   R = 180.000 deg }; P[5:"pickl"] {   GP1 :     UF : 0, UT : 1,   CONFIG : 'N U T,0,0,0',     X = 5352.207 mm,  Y = 489.431 mm,   Z = -597.520 mm,     W = 180.000 deg,  P = 0.000 deg,   R = 180.000 deg }; P[6:"pickl_dep"] {   GP1 :     UF : 0, UT : 1,   CONFIG : 'N U T,0,0,0',     X = 5352.238 mm,  Y = 489.429 mm,   Z = -397.509 mm,     W = 180.000 deg,  P = 0.000 deg,   R = 180.000 deg }; </pre>
--	---

Fonte: Elaborado pelo autor, 2024.

No Apêndice A é possível visualizar os principais programas dos robôs OP05R01 e OP20R02.

### 4.4 Comissionamento Virtual

A partir da comunicação entre o *software* de simulação e o CLP obteve-se a validação da lógica de programação dos robôs e da célula de forma geral. Foi possível iniciar o ciclo da estação pelo TIA Portal e acompanhar o produto avançando em cada uma das operações.

Os aspectos relativos à segurança da célula também foram evidenciados. Durante o funcionamento da estação a abertura dos portões interrompeu a movimentação dos robôs e, ao fechá-los, o processo de produção foi retomado. A presença do operador nas áreas de abastecimento durante os programas de pega ocasionou a parada dos robôs referentes a cada estação de abastecimento, como previsto durante o desenvolvimento do programa do CLP.

Como resultado final deste trabalho, destaca-se o vídeo de funcionamento da célula, onde todos aspectos mencionados até o momento são apresentados de maneira visual e dinâmica. Este vídeo pode ser acessado através do *QR Code* da Figura 46.

Figura 46 – *QR Code* para acesso ao vídeo de funcionamento da célula robotizada.



Fonte: Elaborado pelo autor, 2024.

O programa do CLP referente ao comissionamento virtual da célula robotizada está disponível no Apêndice B.

## 5 CONCLUSÃO

Este trabalho propõe o desenvolvimento da simulação, programação *offline* e comissionamento virtual de uma célula robotizada na indústria automotiva. Esses conceitos, embora já consolidados e amplamente utilizados na indústria, podem ainda ser pouco familiares no contexto educacional. Nesse sentido, sua principal contribuição consistiu em demonstrar todas as etapas envolvidas no processo de simulação em ambiente digital, utilizando os *softwares* Tecnomatix Process Simulate e TIA Portal como ferramentas de execução.

Os resultados obtidos mostram que é possível validar um processo de fabricação automatizado desde que uma representação digital dos elementos envolvidos esteja disponível e condizente com os equipamentos reais.

A realização deste trabalho permitiu evidenciar que verificar o funcionamento de um dispositivo em um *software* de simulação, considerando seu ciclo de funcionamento e cinemática, possibilita que interferências mecânicas com outros equipamentos sejam identificadas ainda na fase de projeto, evitando complicações durante o *startup* da linha. Além disso, através da programação *offline* dos robôs e comissionamento virtual, toda lógica de comunicação entre manipuladores robóticos e CLPs pode ser testada previamente, reduzindo dessa maneira o tempo de programação em campo.

## REFERÊNCIAS

- AYED, M. B.; ZOUARI, L.; ABID, M. Software in the loop simulation for robot manipulators. **Engineering, Technology Applied Science Research**, p. 7(5), 2017.
- BACIC, M. On hardware-in-the-loop simulation. **In Proceedings of the 44th IEEE Conference on Decision and Control**, p. 3194–3198, 2005.
- BHATIA, M. S.; KUMAR, S. Critical success factors of industry 4.0 in automotive manufacturing industry. **IEEE Transactions on Engineering Management**, p. 2439–2453, 2020.
- BOTTAZZI, V.; FONSECA, J. Off-line programming industrial robots based in the information extracted from neutral files generated by the commercial cad tools. **Advanced Robotic Systems International**, p. 349–364, 2007.
- COHEN, Y.; FACCIO, M.; PILATI, F.; YAO, X. Design and management of digital manufacturing and assembly systems in the industry 4.0 era. **The International Journal of Advanced Manufacturing Technology**, p. 3565–3577, 2019.
- COMAU. Auxiliary equipment: Teach pendant. **Comau Products and Solutions**, 2024. Disponível em: <<https://www.comau.com/pt-br/competencies/products-solutions/controls-auxiliary-equipment/teach-pendant/>>. Acesso em: 01 jul. 2024.
- CRAIG, J. J. **Introduction to Robotics: Mechanics and Control**. 3. ed. New Jersey: Pearson Prentice Hall, 2004.
- DUMITRASCU, N.-A.; DINCA, A.; PREDINCEA, N. Virtual commissioning of a robotic cell using tecnomatix process simulate. **Annals of the Academy of Romanian Scientists Series on Engineering Sciences**, p. 45–60, 2017.
- GASTROW, M. A review of trends in the global automotive manufacturing industry and implications for developing countries. **African Journal of Business Management**, p. 5895–5905, 2012.
- HEIMANN, O.; GUHL, J. Industrial robot programming methods: A scoping review. **25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)**, p. 696–700, 2020.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/TC 299 Robotics**. [S.l.], 2016.
- LEE, C. G.; PARK, S. C. Survey on the virtual commissioning of manufacturing systems. **Journal of Computational Design and Engineering**, p. 213–222, 2014.
- LIU, Z.; SUCHOLD, N.; DIEDRICH, C. Virtual commissioning of automated systems. **IntechOpen - Automation**, p. 131–148, 2012.
- LLOPIS-ALBERT, C.; RUBIO, F.; VALERO, F. Impact of digital transformation on the automotive industry. **Technological Forecasting and Social Change**, p. 1–8, 2021.
- NIKU, S. B. **Introduction to Robotics: Analysis, Control, Applications**. 3. ed. EUA: John Wiley Sons, 2020.

PAN, Z.; POLDEN, J.; LARKIN, N.; DUIN, S. V.; NORRISH, J. Recent progress on programming methods for industrial robots. **Robotics and Computer-Integrated Manufacturing**, p. 87–94, 2012.

PAPULOVA, Z.; GAZOVA, A.; SUFLIARSKY, L. Implementation of automation technologies of industry 4.0 in automotive manufacturing companies. **Procedia Computer Science**, p. 1488–1497, 2022.

PIROMALIS, D.; KANTAROS, A. Digital twins in the automotive industry: The road toward physical-digital convergence. **Applied System Innovation**, p. 1–12, 2022.

ROBODK. Robodk api for python. **Python Package Index**, 2024. Disponível em: <<https://pypi.org/project/robodk/>>. Acesso em: 01 jul. 2024.

SIEMENS. Tecnomatix process simulate. **Produtos Siemens**, 2024. Disponível em: <<https://plm.sw.siemens.com/en-US/tecnomatix/products/process-simulate-software/>>. Acesso em: 01 jul. 2024.

SIEMENS. Tia portal: Portal de automação totalmente integrado. **Produtos Siemens**, 2024. Disponível em: <<https://www.siemens.com/br/pt/produtos/software/industria/automacao/tia-portal.html/>>. Acesso em: 01 jul. 2024.

WILLIAMS, D.; APPLETON, E. **Industrial Robot Applications**. 1. ed. EUA: John Wiley And Sons, 1987.

YAMAHA. Ya series 6-axis. **Yamaha Robotics**, 2024. Disponível em: <<https://www.yamaha-usa-robotics.com/robotics/articulated/ya-series-6-axis>>. Acesso em: 01 jul. 2024.

# APÊNDICE A - PROGRAMAS DOS ROBÔS

PROGRAMAS PRINCIPAIS DO ROBÔ 05R01  
(style01, s01pounc, s01drop1 e s01pick1)

```

/PROG style01
/MN
 1: !***** ;
 2: !TCC - IFMG - 2024 ;
 3: !ROBOT: OP05R01 ;
 4: !STYLE: S01 ;
 5: !ASSEMBLY: SPIDER ;
 6: !***** ;
 7: !PROCESS SIMULATE: v16.1 ;
 8: !TEACH: Fanuc RJ v3.7.1 ;
 9: !XML : v8.3_R30iB_GM_global4 ;
10: !STUDY: TCC_IFMG_2024 ;
11: !USER : Douglas Viana;
12: !DATE : 2024-08-18 ;
13: !***** ;
14: ! TEMPLATE 5 ;
15: !STYLE S01 MH ONLY ;
16: !***** ;
17: ;
18: !ECHO STYLE AND OPTION ;
19: GO[1:ManualStyle] = 1 ;
20: ECHO OPTION ;
21: ;
22: CALL s01pounc ;
23: ;
24: CALL s01pick1 ;
25: ;
26: REQUEST CONTINUE ;
27: ;
28: CALL s01drop1 ;
29: ;
30: MOVE TO HOME ;
/POS
/END

```

Fonte: Elaborado pelo autor, 2024.

```

/PROG s01pounc
/MN
  1: !***** ;
  2: !TCC - IFMG - 2024 ;
  3: !ROBOT: OP05R01 ;
  4: !STYLE: S01 ;
  5: !ASSEMBLY: SPIDER ;
  6: !***** ;
  7: !PROCESS SIMULATE: v16.1 ;
  8: !TEACH: Fanuc RJ v3.7.1 ;
  9: !XML : v8.3_R30iB_GM_global4 ;
 10: !STUDY: TCC_IFMG_2024;
 11: !USER : Douglas Viana;
 12: !DATE : 2024-08-18 ;
 13: !***** ;
 14: !PATH SEGMENT - MOVE TO POUNCE ;
 15: SET SEGMENT(1) ;
 16: PAYLOAD[1] ;
 17: UFRAME_NUM = 0;
 18: UTOOL_NUM = 1;
 19: J P[1:home] 100% FINE;
 20: J P[2:pounce] 100% FINE;
 21: AT POUNCE ;
/POS
P[1:"home"] {
  GP1 :
    UF : 0, UT : 1, CONFIG : 'N U T,0,0,0',
    X = 2233.933 mm, Y = 498.487 mm, Z = 836.422 mm,
    W = 180.000 deg, P = 50.000 deg, R = 180.000 deg,
    E1 = 1000.000 mm
};
P[2:"pounce"] {
  GP1 :
    UF : 0, UT : 1, CONFIG : 'N U T,0,0,0',
    X = 2352.561 mm, Y = 512.370 mm, Z = 888.532 mm,
    W = -178.235 deg, P = 31.890 deg, R = -179.789 deg,
    E1 = 1000.000 mm
};
/END

```

Fonte: Elaborado pelo autor, 2024.

**/PROG s01pick1**

/MN

```
1:  !***** ;
2:  !TCC - IFMG - 2024 ;
3:  !ROBOT: OP05R01 ;
4:  !STYLE: S01 ;
5:  !ASSEMBLY: SPIDER ;
6:  !***** ;
7:  !PROCESS SIMULATE: v16.1 ;
8:  !TEACH: Fanuc RJ v3.7.1 ;
9:  !XML   : v8.3_R30Ib_GM_global4 ;
10: !STUDY: TCC_IFMG_2024 ;
11: !USER  : Douglas Viana;
12: !DATE  : 2024-08-18 ;
13: !***** ;
14:      ;
15: !PICKUP PART FROM TOOL X ;
16: !***** ;
17:      ;
18: !ASSIGN PATH SEGMENT VALUES ;
19: SET SEGMENT(10) ;
20:      ;
21: Set MH Tool(1) ;
22:      ;
23: Prepare to Pick(0) ;
24:      ;
25: !MOVE TO TOOL X PICKUP ;
26:      ;
27: PAYLOAD[1] ;
28: UFRAME_NUM = 0;
29: UTOOL_NUM = 1;
30: J P[1] 100% FINE;
31: J P[2] 100% CNT80;
32: J P[3] 80% CNT80;
33: L P[4:pick1_app] 400mm/sec FINE;
34: L P[5:pick1] 400mm/sec FINE;
35: Grip Part(1) ;
36:      ;
37: PAYLOAD[2] ;
38:      ;
39: !MOVE ABOVE PICK POS ;
40: !TO CHECK NO PART IN TOOL ;
41: L P[6:pick1_dep] 400mm/sec CNT50;
42: !CHECK PART IN EOAT ;
43: Part Present(0) ;
44:      ;
45: SET SEGMENT(11) ;
46:      ;
47: J P[7] 80% CNT80;
48: J P[8] 100% CNT80;
49: J P[9] 100% CNT80;
50: J P[10] 100% FINE;
51: !CLEAR OF TOOL X PICKUP ;
52:      ;
53: SET SEGMENT(12) ;
54:      ;
55: !MOVE TOWARDS DROP ;
56: J P[11] 100% CNT80;
```

```

57:J P[12] 80% FINE;
/POS
P[1] {
  GP1 :
    UF : 0, UT : 1,    CONFIG : `N U T,0,0,0',
    X = 2352.561 mm,   Y = 512.370 mm,    Z = 888.532 mm,
    W = -178.235 deg, P = 31.890 deg,    R = -179.789 deg,
    E1 = 1000.000 mm
};

                                . . .

};
P[4:"pick1_app"] {
  GP1 :
    UF : 0, UT : 1,    CONFIG : `N U T,0,0,0',
    X = 5352.238 mm,   Y = 489.429 mm,    Z = -397.509 mm,
    W = 180.000 deg,  P = 0.000 deg,    R = 180.000 deg,
    E1 = 2900.000 mm
};
P[5:"pick1"] {
  GP1 :
    UF : 0, UT : 1,    CONFIG : `N U T,0,0,0',
    X = 5352.207 mm,   Y = 489.431 mm,    Z = -597.520 mm,
    W = 180.000 deg,  P = 0.000 deg,    R = 180.000 deg,
    E1 = 2900.000 mm
};
P[6:"pick1_dep"] {
  GP1 :
    UF : 0, UT : 1,    CONFIG : `N U T,0,0,0',
    X = 5352.238 mm,   Y = 489.429 mm,    Z = -397.509 mm,
    W = 180.000 deg,  P = 0.000 deg,    R = 180.000 deg,
    E1 = 2900.000 mm
};

                                . . .

P[12] {
  GP1 :
    UF : 0, UT : 1,    CONFIG : `N U T,0,0,0',
    X = -1077.226 mm,  Y = -1237.452 mm,   Z = 1932.275 mm,
    W = 140.277 deg,  P = -1.216 deg,    R = -44.365 deg,
    E1 = 100.000 mm
};
/END

```

Fonte: Elaborado pelo autor, 2024.

**/PROG s01drop1**

/MN

```

1:  !***** ;
2:  !TCC - IFMG - 2024 ;
3:  !ROBOT: OP05R01 ;
4:  !STYLE: S01 ;
5:  !ASSEMBLY: SPIDER ;
6:  !***** ;
7:  !PROCESS SIMULATE: v16.1 ;
8:  !TEACH: Fanuc RJ v3.7.1 ;
9:  !XML : v8.3_R30iB_GM_global4 ;
10: !STUDY: TCC_IFMG_2024 ;
11: !USER : Douglas Viana;
12: !DATE : 2024-08-18 ;
13: !***** ;
14: ;
15: !DROPOFF PART TO TOOL X ;
16: !***** ;
17: ;
18: !ASSIGN PATH SEGMENT VALUES ;
19: SET SEGMENT(30) ;
20: ;
21: Set MH Tool(1) ;
22: ;
23: !MOVE TO DROPOFF TOOL X ;
24: ;
25: PAYLOAD[2] ;
26: UFRAME_NUM = 0;
27: UTOOL_NUM = 1;
28:J P[1] 80% FINE;
29:J P[2] 80% CNT80;
30:J P[3] 80% CNT80;
31:J P[4] 80% CNT80;
32:L P[5:drop1_app] 400mm/sec FINE;
33:L P[6:drop1] 400mm/sec FINE;
34: Release Part(1) ;
35: ;
36: ;
37: PAYLOAD[1] ;
38: ;
39: !MOVE ABOVE DROP POS ;
40: !TO CHECK NO PART IN TOOL ;
41:L P[7:drop1_dep] 400mm/sec CNT50;
42: !CHECK NOPART IN EOAT ;
43: Check No Part(0) ;
44: ;
45: SET SEGMENT(31) ;
46: ;
47:J P[8] 80% CNT80;
48:J P[9] 80% CNT80;
49:J P[10] 80% CNT80;
50:J P[11] 100% FINE;
51: !CLEAR OF TOOL X DROPOFF ;
52: ;
53: SET SEGMENT(32) ;
54: ;
55: !MOVE TOWARDS HOME ;
56:J P[12] 100% CNT80;

```

```

57: !PATH SEGMENT - ALL CLEAR ;
58: SET SEGMENT(63) ;
59:J P[13:home] 100% FINE;
/POS

P[1] {
  GP1 :
    UF : 0, UT : 1,      CONFIG : 'N U T,0,0,0',
    X = -1077.226 mm,    Y = -1237.452 mm,      Z = 1932.275 mm,
    W = 140.277 deg,    P = -1.216 deg,      R = -44.365 deg,
    E1 = 100.000 mm
};

                                . . .

P[5:"drop1_app"] {
  GP1 :
    UF : 0, UT : 1,      CONFIG : 'N U T,0,0,0',
    X = -2417.298 mm,    Y = -2217.220 mm,      Z = -367.479 mm,
    W = 180.000 deg,    P = 0.000 deg,      R = -45.000 deg,
    E1 = 100.000 mm
};

P[6:"drop1"] {
  GP1 :
    UF : 0, UT : 1,      CONFIG : 'N U T,0,0,0',
    X = -2417.294 mm,    Y = -2217.248 mm,      Z = -567.491 mm,
    W = 180.000 deg,    P = 0.000 deg,      R = -45.000 deg,
    E1 = 100.000 mm
};

P[7:"drop1_dep"] {
  GP1 :
    UF : 0, UT : 1,      CONFIG : 'N U T,0,0,0',
    X = -2417.298 mm,    Y = -2217.220 mm,      Z = -367.479 mm,
    W = 180.000 deg,    P = 0.000 deg,      R = -45.000 deg,
    E1 = 100.000 mm
};

                                . . .

P[13:"home"] {
  GP1 :
    UF : 0, UT : 1,      CONFIG : 'N U T,0,0,0',
    X = 2233.933 mm,     Y = 498.487 mm,      Z = 836.422 mm,
    W = 180.000 deg,    P = 50.000 deg,      R = 180.000 deg,
    E1 = 1000.000 mm
};
/END

```

Fonte: Elaborado pelo autor, 2024.

PROGRAMAS PRINCIPAIS DO ROBÔ 20R02  
(Style01, s01pounc, s01procl)

```

/PROG style01
/MN
  1: !***** ;
  2: !TCC - IFMG - 2024 ;
  3: !ROBOT: OP20R02 ;
  4: !STYLE: S01 ;
  5: !ASSEMBLY: SPIDER ;
  6: !***** ;
  7: !PROCESS SIMULATE: v16.1 ;
  8: !TEACH: Fanuc RJ v3.7.1 ;
  9: !XML : v8.3_R30iB_GM_global4 ;
 10: !STUDY: TCC_IFMG_2024 ;
 11: !USER : Douglas Viana ;
 12: !DATE : 2024-08-28 ;
 13: !***** ;
 14: ! TEMPLATE 2 ;
 15: !STYLE 01 CARRIED WELDER ;
 16: !Servo Gun;
 17: !***** ;
 18: ;
 19: !ECHO STYLE AND OPTION ;
 20: GO[1:ManualStyle] = 1 ;
 21: ECHO OPTION ;
 22: MAINT_PROG[1]=MOV_REPR ;
 23: ;
 24: CALL s01pounc ;
 25: ;
 26: CALL STPR_CHK ;
 27: ;
 28: CALL s01procl ;
 29: ;
 30: RUN CAP_WEAR ;
 31: ;
 32: MOVE TO HOME ;
 33: ;
 34: WAIT (F[1:Capwear Complete]) ;
/POS
/END

```

Fonte: Elaborado pelo autor, 2024.

```

/PROG s01pounc
/APPL
  SPOT : TRUE ;
  SPOT WELDING EQUIPMENT NUMBER : 1 ;
/MN
  1: !***** ;
  2: !TCC - IFMG - 2024 ;
  3: !ROBOT: OP20R02 ;
  4: !STYLE: S01 ;
  5: !ASSEMBLY: SPIDER ;
  6: !***** ;
  7: !PROCESS SIMULATE: v16.1 ;
  8: !TEACH: Fanuc RJ v3.7.1 ;
  9: !XML : v8.3_R30iB_GM_global4 ;
 10: !STUDY: TCC_IFMG_2024 ;
 11: !USER : Douglas Viana ;
 12: !DATE : 2024-08-28 ;
 13: !***** ;
 14: !PATH SEGMENT - MOVE TO POUNCE ;
 15: SET SEGMENT(1) ;
 16: PAYLOAD[1] ;
 17: UFRAME_NUM = 0;
 18: UTOOL_NUM = 1;
 19:J P[1:home] 100% FINE;
 20:J P[2:pounce] 100% FINE;
 21: AT POUNCE ;
/POS
P[1:"home"] {
  GP1 :
    UF : 0, UT : 1,      CONFIG : 'F U T,0,0,0',
    X = 814.107 mm,     Y = -740.394 mm,      Z = -358.893 mm,
    W = 63.028 deg,     P = 87.747 deg,      R = 11.601 deg
  GP2 :
    UF : 0, UT : 1,
    J1 = -30.000 mm
};
P[2:"pounce"] {
  GP1 :
    UF : 0, UT : 1,      CONFIG : 'F U T,0,0,0',
    X = 1852.876 mm,     Y = -845.921 mm,      Z = -323.314 mm,
    W = 0.000 deg,      P = 0.000 deg,      R = 0.008 deg
  GP2 :
    UF : 0, UT : 1,
    J1 = -30.000 mm
};
/END

```

Fonte: Elaborado pelo autor, 2024.

```

/PROG s01procl
/APPL
  SPOT : TRUE ;
  SPOT WELDING EQUIPMENT NUMBER : 1 ;
/MN
  1: !***** ;
  2: !TCC - IFMG - 2024 ;
  3: !ROBOT: OP20R02 ;
  4: !STYLE: S01 ;
  5: !ASSEMBLY: SPIDER ;
  6: !***** ;
  7: !PROCESS SIMULATE: v16.1 ;
  8: !TEACH: Fanuc RJ v3.7.1 ;
  9: !XML : v8.3_R30iB_GM_global4 ;
 10: !STUDY: TCC_IFMG_2024 ;
 11: !USER : Douglas Viana ;
 12: !DATE : 2024-08-28 ;
 13: !***** ;
 14: !PATH SEGMENT - BEGIN PROCESS ;
 15: SET SEGMENT(50) ;
 16: !Pass Body Id to Welder Control ;
 17: GO[43:Weld_BodyID_Cnt1]=(GO[1:ManualStyle]) ;
 18: PAYLOAD[1] ;
 19: UFRAME_NUM = 0;
 20: UTOOL_NUM = 1;
 21:J P[1:pounce] 80% FINE;
 22:J P[2] 80% CNT80;
 23:L P[3] 1600mm/sec CNT80;
 24:L P[4] 1600mm/sec CNT80;
 25:L P[5] 1600mm/sec CNT80;
 26:L P[6] 1600mm/sec CNT80;
 27:L P[7] 1600mm/sec CNT80;
 28:L P[8:12A1206] 800mm/sec CNT100 SPOT[SD=15,P=1,t=8.0,S=1,ED=15];
 29:L P[9:12A1207] 800mm/sec CNT100 SPOT[SD=15,P=1,t=8.0,S=1,ED=15];
 30:L P[10:12A1208] 800mm/sec CNT100 SPOT[SD=15,P=1,t=8.0,S=1,ED=15];
 31:L P[11:12A1623] 800mm/sec CNT100 SPOT[SD=15,P=1,t=8.0,S=1,ED=15];
 32:L P[12:12A1611] 800mm/sec CNT100 SPOT[SD=15,P=1,t=8.0,S=1,ED=15];
 33:L P[13:12A1610] 800mm/sec CNT100 SPOT[SD=15,P=1,t=8.0,S=1,ED=15];
 34:L P[14:12A1603] 800mm/sec CNT100 SPOT[SD=15,P=1,t=8.0,S=1,ED=15];
 35:L P[15:12A1601] 800mm/sec CNT100 SPOT[SD=15,P=1,t=8.0,S=1,ED=15];
 36:L P[16:12A1589] 800mm/sec CNT100 SPOT[SD=15,P=1,t=8.0,S=1,ED=15];
 37:J P[17] 80% CNT80;
 38:L P[18] 1600mm/sec CNT80;
 39:L P[19:12A1588] 800mm/sec CNT100 SPOT[SD=15,P=1,t=8.0,S=1,ED=15];
 40:J P[20] 80% CNT80;
 41:L P[21] 1600mm/sec CNT80;
 42:J P[22] 80% CNT80;
 43:J P[23] 80% CNT80;
 44:J P[24] 80% CNT80;
 45:L P[25:enter_izone1] 1600mm/sec FINE;
 46: ! IZONE with [ OP20R01] ;
 47: ENTER I-ZONE(1) ;
 48:J P[26] 80% CNT80;
 49:L P[27:12A1205] 800mm/sec CNT100 SPOT[SD=15,P=1,t=8.0,S=1,ED=15];
 50: CALL PARTCOMP(1) ;

```

```

51:J P[28] 80% CNT80;
52:L P[29] 1600mm/sec CNT80;
53:J P[30] 80% CNT80;
54:J P[31:exit_zone1] 80% FINE;
55: ! IZONE with [ OP20R01] ;
56: EXIT I-ZONE(1) ;
57: ! ;
58: !PATH SEGMENT - TOOL CLEAR ;
59: SET SEGMENT(62) ;
60:J P[32] 80% CNT80;
61:J P[33] 80% FINE;
62: !PATH SEGMENT - ALL CLEAR ;
63: SET SEGMENT(63) ;
64:J P[34] 80% FINE;
/POS
P[1:"pounce"] {
  GP1 :
    UF : 0, UT : 1,      CONFIG : 'F U T,0,0,0',
    X = 1852.876 mm,    Y = -845.921 mm,    Z = -323.314 mm,
    W = 0.000 deg,    P = 0.000 deg,    R = 0.008 deg
  GP2 :
    UF : 0, UT : 1,
    J1 = -30.000 mm
};
P[2] {
  GP1 :
    UF : 0, UT : 1,      CONFIG : 'F U T,0,0,0',
    X = 2489.585 mm,    Y = -537.678 mm,    Z = -359.056 mm,
    W = -2.127 deg,    P = 6.943 deg,    R = 18.423 deg
  GP2 :
    UF : 0, UT : 1,
    J1 = -250.000 mm
};
P[3] {
  GP1 :
    UF : 0, UT : 1,      CONFIG : 'F U T,0,0,0',
    X = 3076.122 mm,    Y = -922.963 mm,    Z = -374.265 mm,
    W = -6.293 deg,    P = 3.628 deg,    R = -24.647 deg
  GP2 :
    UF : 0, UT : 1,
    J1 = -250.000 mm
};
. . .
P[34] {
  GP1 :
    UF : 0, UT : 1,      CONFIG : 'F U T,0,0,0',
    X = 814.107 mm,    Y = -740.394 mm,    Z = -358.893 mm,
    W = 63.028 deg,    P = 87.747 deg,    R = 11.601 deg
  GP2 :
    UF : 0, UT : 1,
    J1 = -30.000 mm
};
/END

```

## TCC\_IFMG\_2024\_DOUGLAS\_VIANA / PLC\_1 [CPU 1511-1 PN] / Program blocks

### Main [OB123]

#### Main Properties

##### General

<b>Name</b>	Main	<b>Number</b>	123	<b>Type</b>	OB
<b>Language</b>	LAD	<b>Numbering</b>	Automatic		

##### Information

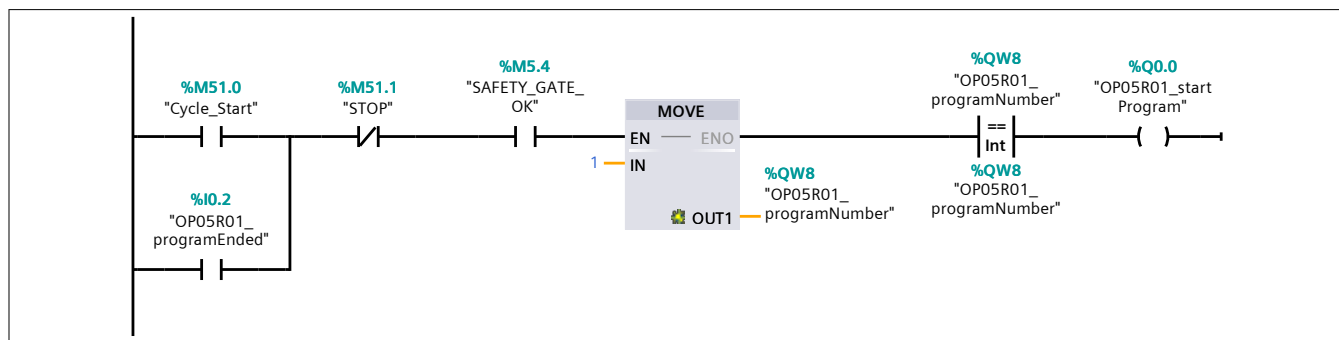
<b>Title</b>	"CICLO DE FUNCIONAMENTO - ESTAÇÃO ROBOTIZADA"	<b>Author</b>		<b>Comment</b>	Instituição: Instituto Federal de Minas Gerais - IFMG - Campus Betim Monografia: Simulação, Programação Offline e Virtual Commissioning de uma célula robotiza Autor: Douglas Viana Dornelas Orientador: Prof. Dr. Arthur Hermano Rezende Rosa
<b>Family</b>		<b>Version</b>	0.1	<b>User-defined ID</b>	

#### Main

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

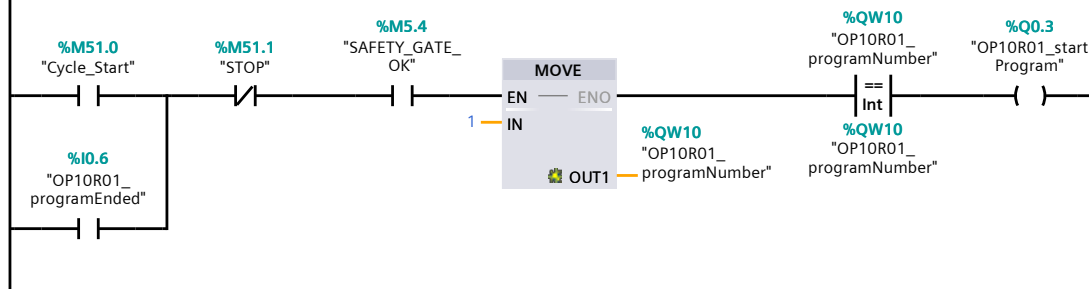
#### Network 1: START ROBO - OP05R01

Envia o código de programa para o robô 05R01 e o mantém em ciclo

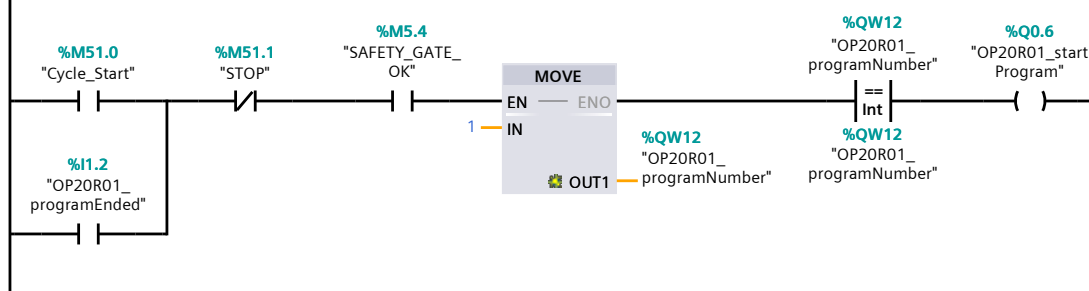


#### Network 2: START ROBO - OP10R01

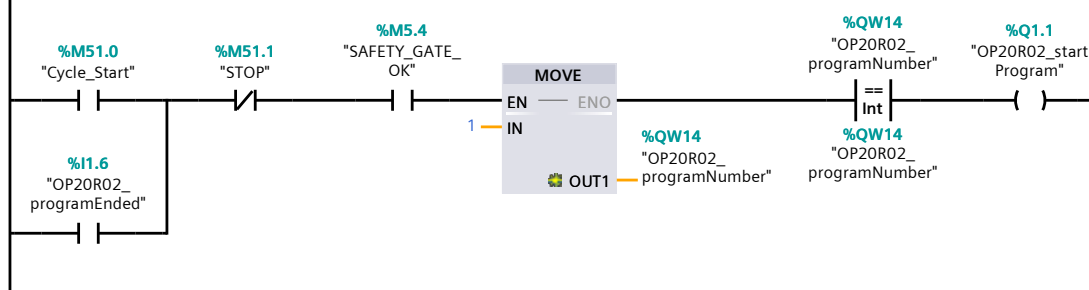
Envia o código de programa para o robô 10R01 e o mantém em ciclo

**Network 3: START ROBO - OP20R01**

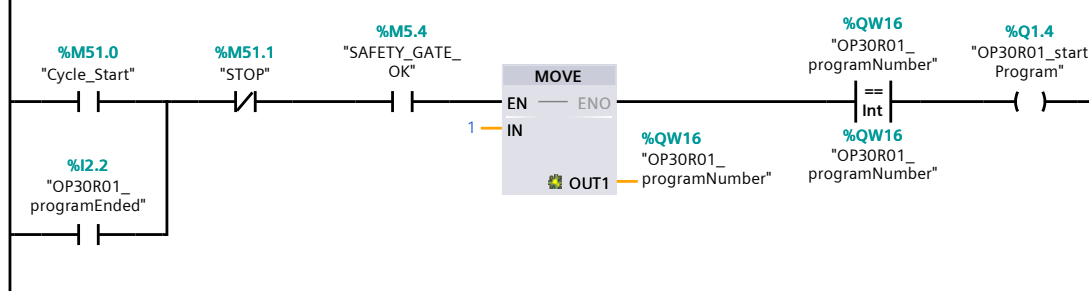
Envia o código de programa para o robô 020R01 e o mantém em ciclo

**Network 4: START ROBO - OP20R02**

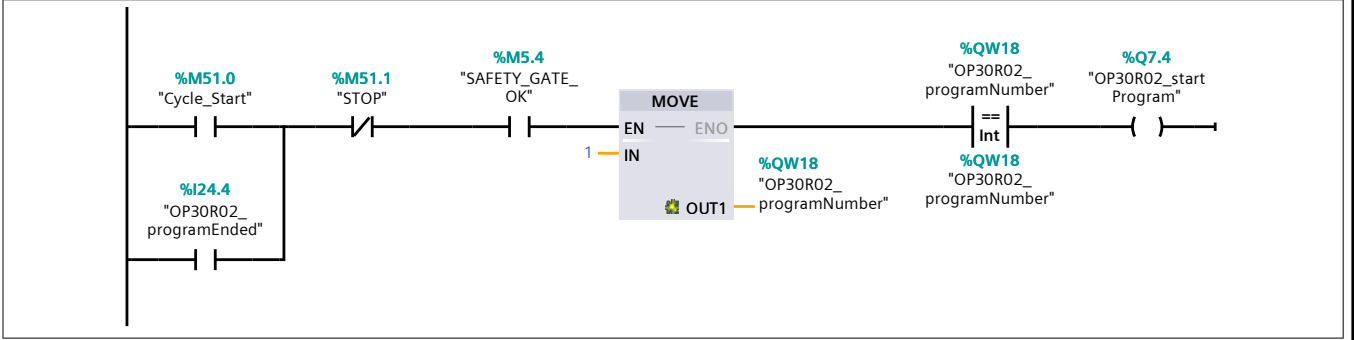
Envia o código de programa para o robô 020R02 e o mantém em ciclo

**Network 5: START ROBO - OP30R01**

Envia o código de programa para o robô 030R01 e o mantém em ciclo

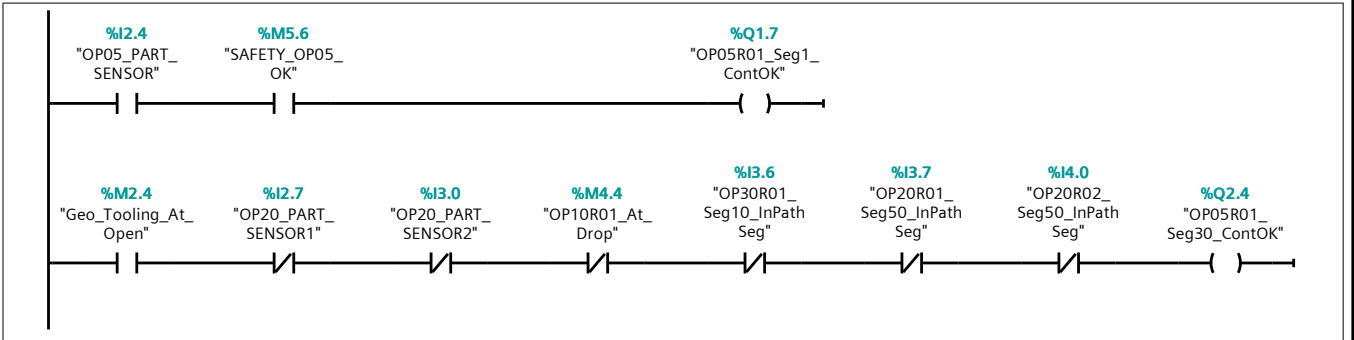
**Network 6: START ROBO - OP30R02**

Envia o código de programa para o robô 030R02 e o mantém em ciclo



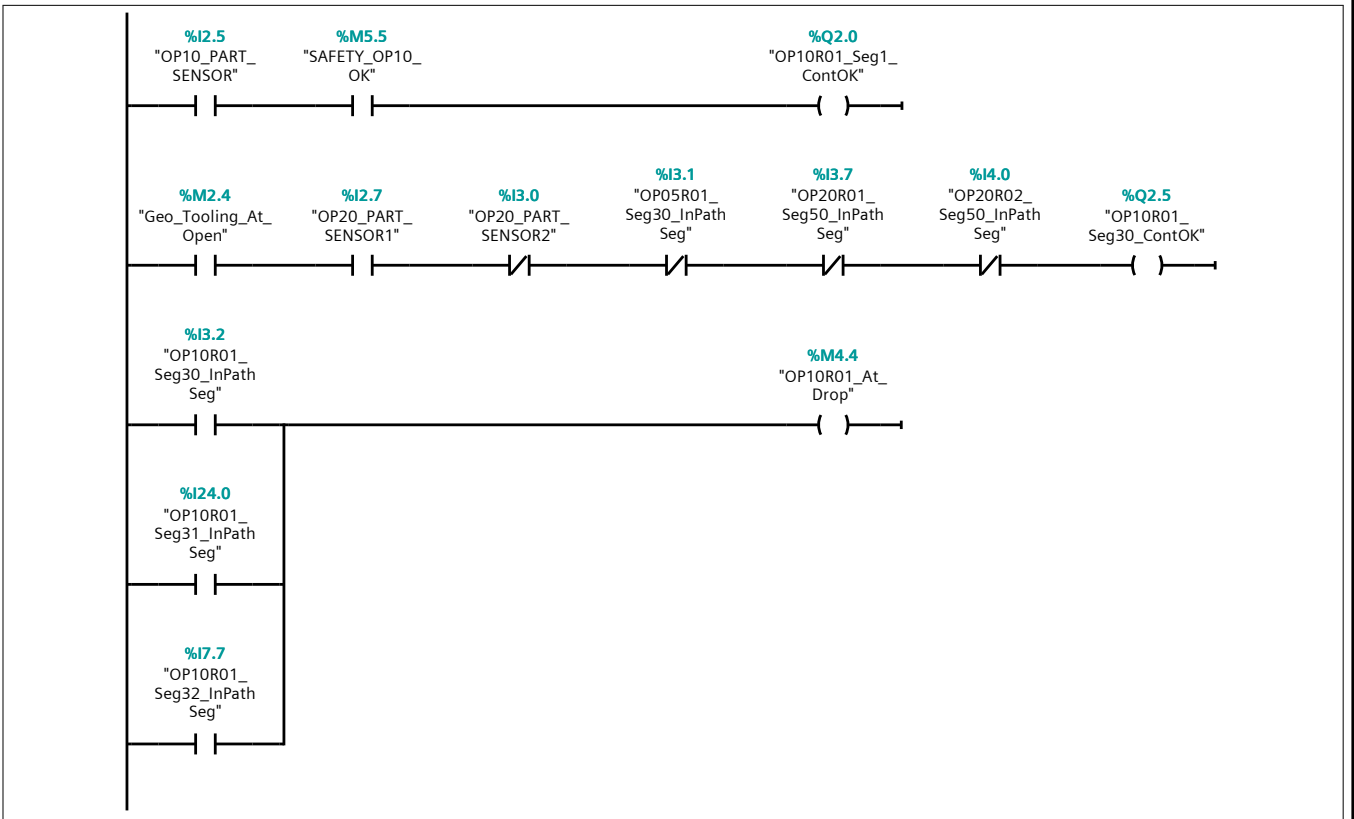
### Network 7: CICLO DO ROBÔ 05R01

Permissão para coleta e depósito da peça A



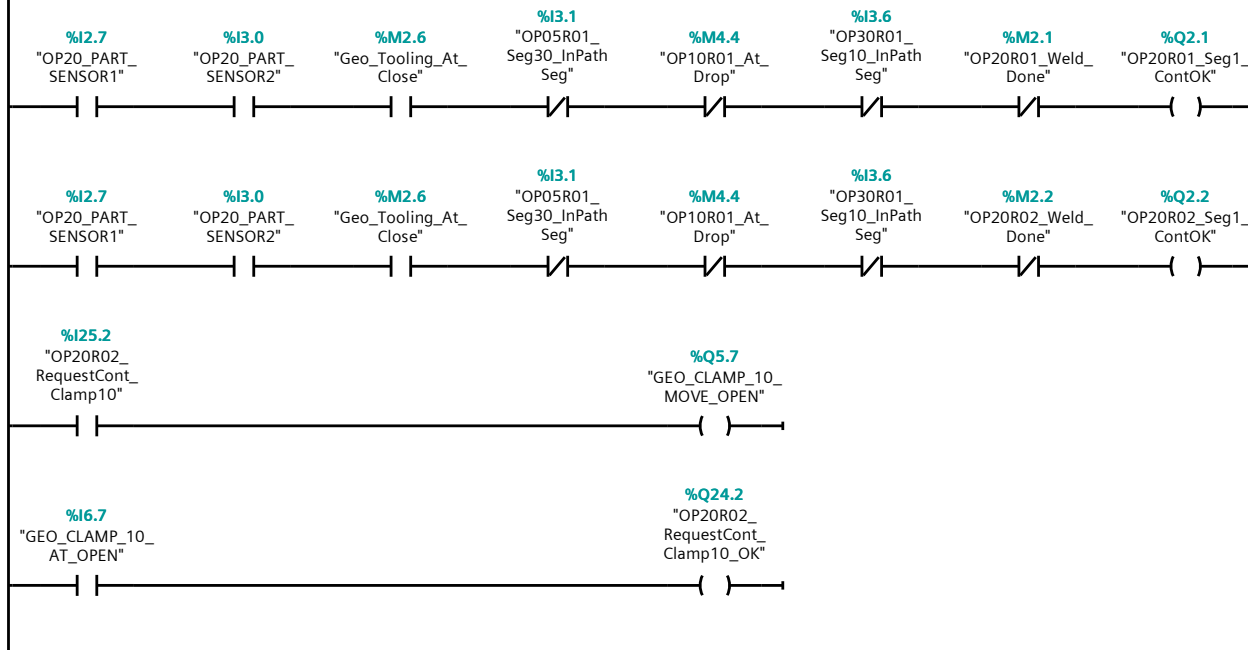
### Network 8: CICLO DO ROBÔ 10R01

Permissão para coleta e depósito da peça B



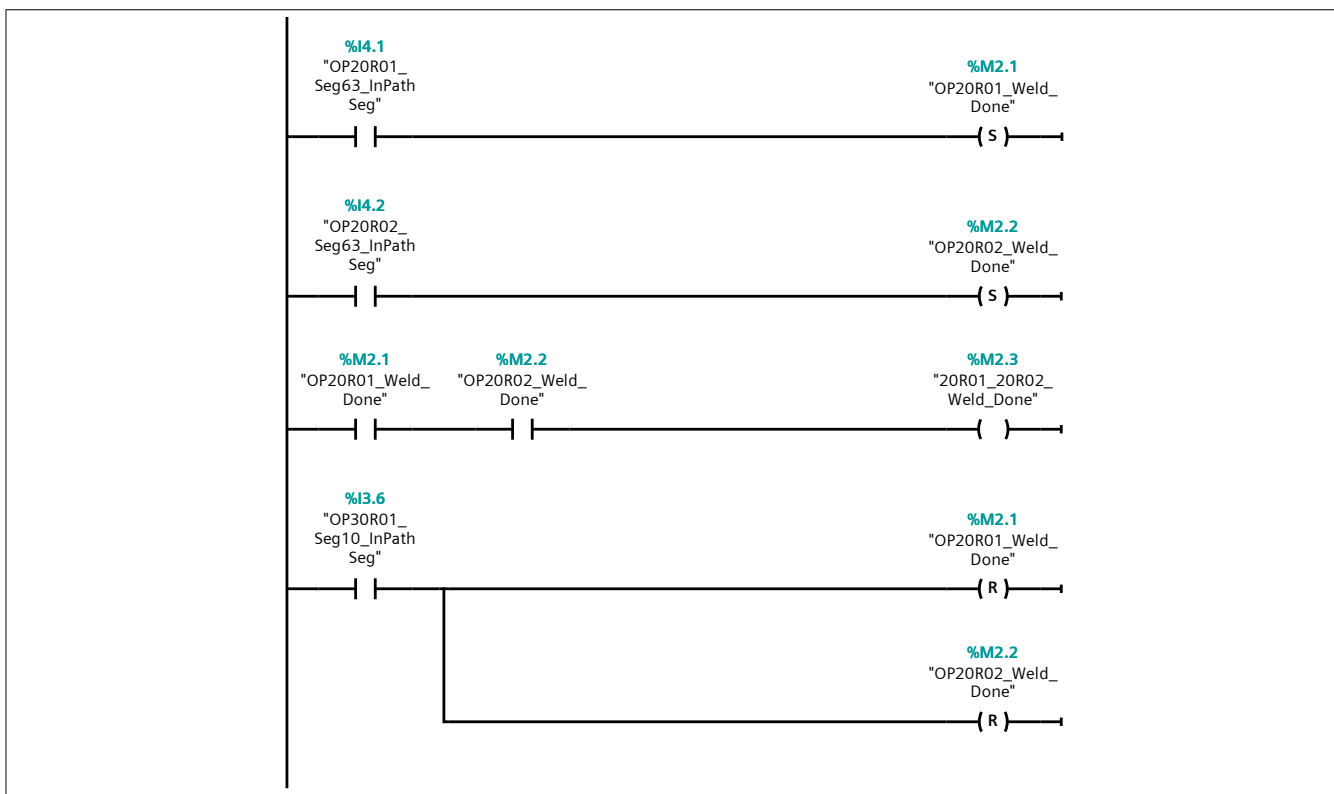
### Network 9: CICLO DOS ROBÔS 20R01 E 20R02

Permissão para soldar na mesa de geometria e abrir um dos grampos no último ponto de solda do robô 20R02



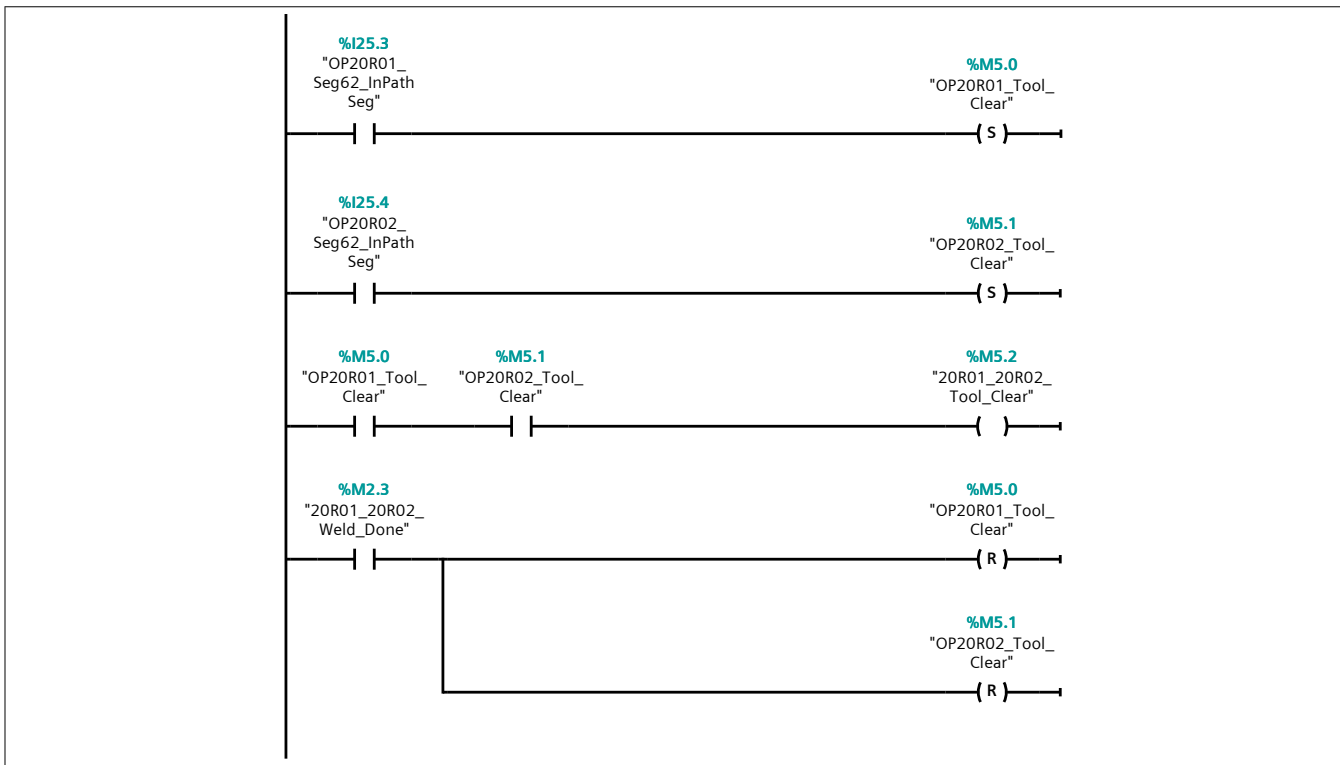
**Network 10: MONITORAMENTO DE FIM DE SOLDA NA OP20**

Lógica utilizada para verificar se a solda na OP20 foi concluída



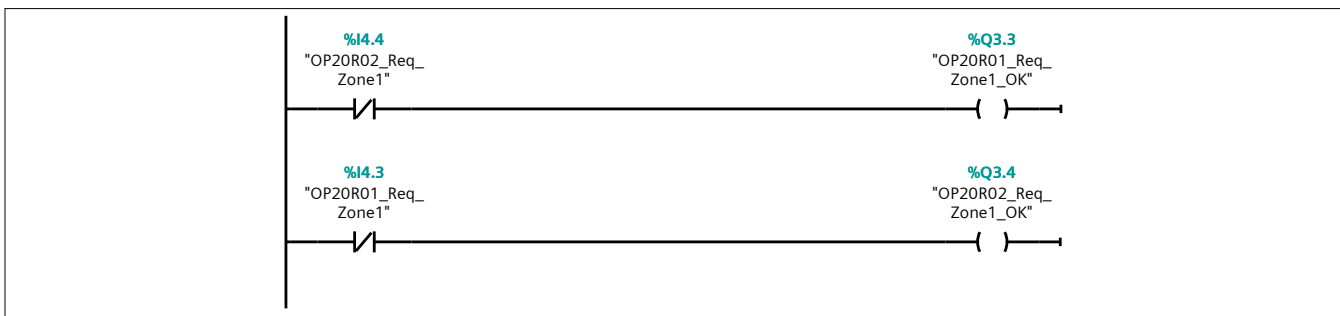
**Network 11: MONITORAMENTO DE FORA DE INTERFERÊNCIA NA OP20**

Lógica utilizada para verificar se os robôs estão fora do dispositivo de geometria da OP20



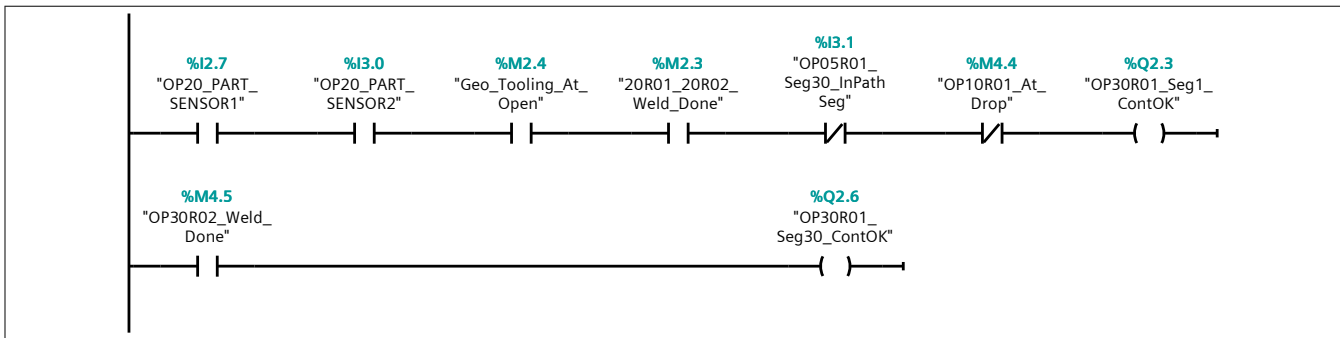
**Network 12: INTERLOCK ENTRE OS ROBÔS 20R01 E 20R02**

Lógica de intertravamento entre os robôs de solda, impedindo colisão quando soldam na mesma área



**Network 13: CICLO DO ROBÔ 30R01**

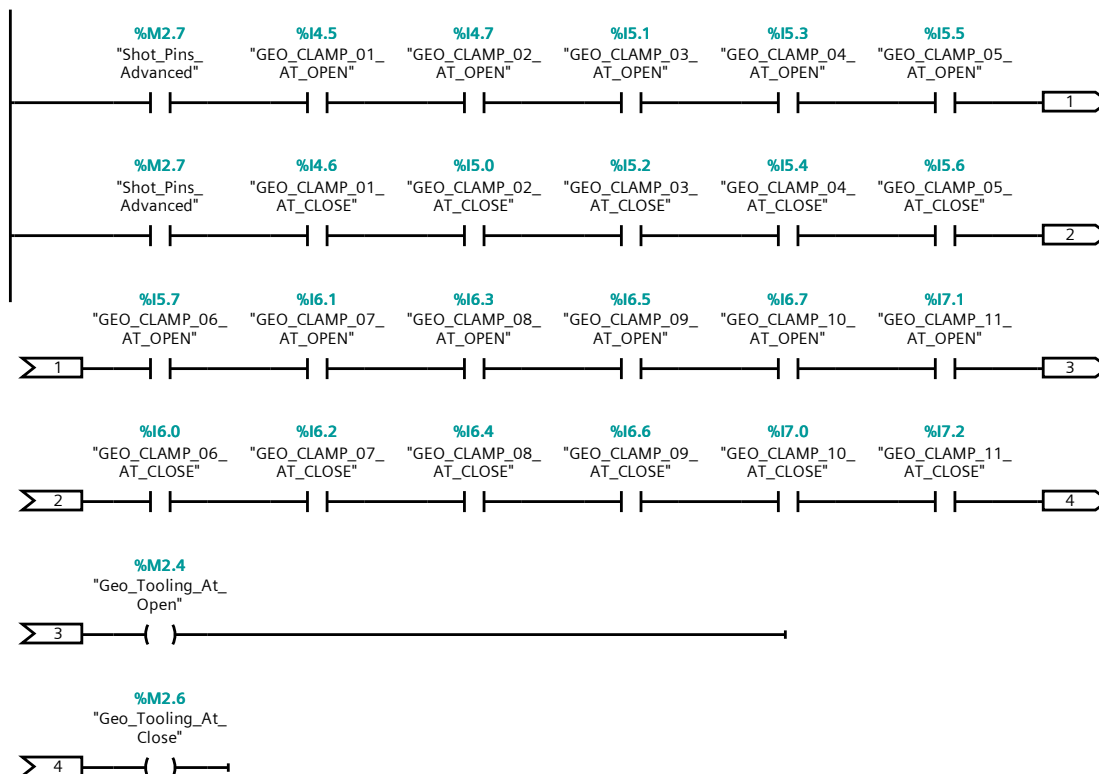
Permissão para coleta e depósito do Spider



**Network 14: CICLO DO ROBÔ 30R02**

Permissão para soldar o Spider na garra do robô 30R01 e identificar o fim de solda





### Network 17: FECHA DISPOSITIVO DE GEOMETRIA DA OP20

O dispositivo de geometria é fechado após a liberação da área pelo robô 10R01

**%I7.7**  
"OP10R01\_  
Seg32\_InPath  
Seg"

**%Q3.6**  
"GEO\_CLAMP\_01\_  
MOVE\_CLOSE"

**%Q4.0**  
"GEO\_CLAMP\_02\_  
MOVE\_CLOSE"

**%Q4.2**  
"GEO\_CLAMP\_03\_  
MOVE\_CLOSE"

**%Q4.4**  
"GEO\_CLAMP\_04\_  
MOVE\_CLOSE"

**%Q4.6**  
"GEO\_CLAMP\_05\_  
MOVE\_CLOSE"

**%Q5.0**  
"GEO\_CLAMP\_06\_  
MOVE\_CLOSE"

**%Q5.2**  
"GEO\_CLAMP\_07\_  
MOVE\_CLOSE"

**%Q5.4**  
"GEO\_CLAMP\_08\_  
MOVE\_CLOSE"

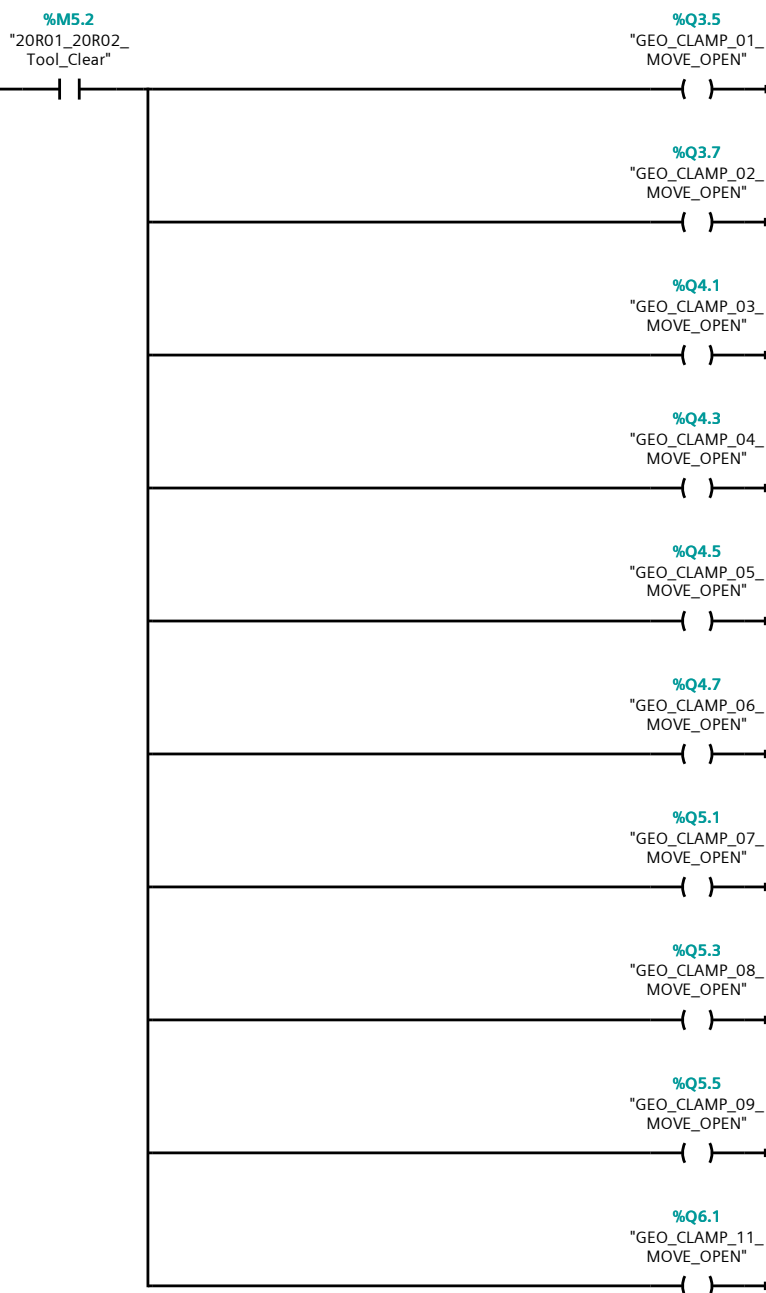
**%Q5.6**  
"GEO\_CLAMP\_09\_  
MOVE\_CLOSE"

**%Q6.0**  
"GEO\_CLAMP\_10\_  
MOVE\_CLOSE"

**%Q6.2**  
"GEO\_CLAMP\_11\_  
MOVE\_CLOSE"

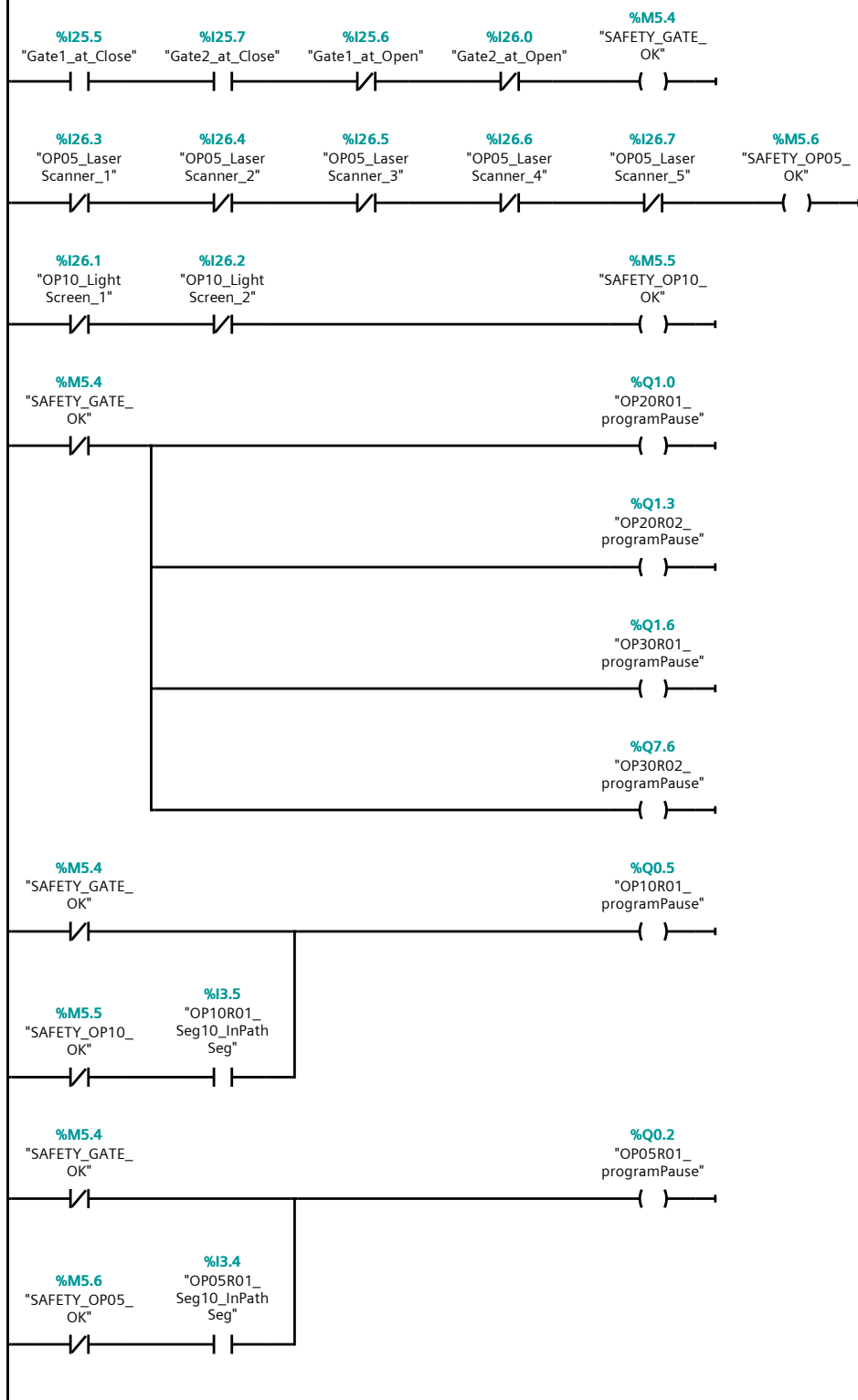
### Network 18: ABRE DISPOSITIVO DE GEOMETRIA DA OP20

O dispositivo é aberto após os robôs de solda liberarem a área



### Network 19: SEGURANÇA DA ESTAÇÃO

Monitoramento dos portões e áreas de abastecimento



## Network 20: CRIAÇÃO DE PEÇAS E FLUXO DA ESTEIRA

Lógica utilizada para criar as peças na simulação e realizar o translado na esteira

