

UNIVERSIDADE DE SÃO PAULO

JOÃO VICTOR ROCHA SOARES

Análise e implementação de técnicas de robótica colaborativa para tarefa de
montagem de mobiliário aeronáutico

São Carlos

2017

JOÃO VICTOR ROCHA SOARES

Análise e implementação de técnicas de robótica colaborativa para tarefa de
montagem de mobiliário aeronáutico

Trabalho de conclusão de curso
apresentado ao Departamento de
Engenharia Elétrica e de Computação
para obtenção do título de Bacharelado
em Engenharia Elétrica

Área de Concentração: Robótica
Colaborativa

Orientador: Prof. Dr. Glauco Augusto de
Paula Caurin

São Carlos

2017

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Rocha Soares, João Victor
RR672a Análise e implementação de técnicas de robótica
colaborativa para tarefa de montagem de mobiliário
aeronáutico / João Victor Rocha Soares; orientador
Glauco Augusto de Paula Caurin. São Carlos, 2017.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Sistemas de Energia e Automação) -- Escola de
Engenharia de São Carlos da Universidade de São Paulo,
2017.

1. Aprendizagem por reforço. 2. Aprendizagem por
demonstração. 3. Robôs manipuladores. I. Título.

FOLHA DE APROVAÇÃO

Nome: João Victor Rocha Soares

Título: “Análise e implementação de técnicas de robótica colaborativa para tarefa de montagem de mobiliário aeronáutico”

Trabalho de Conclusão de Curso defendido e aprovado

em 28 / 11 / 2017,

com NOTA Def (10 , 0), pela Comissão Julgadora:

**Prof. Titular Glauco Augusto de Paula Caurin - Orientador -
SEM/EESC/USP**

Prof. Dr. Valdir Grassi Júnior - SEL/EESC/USP

Mestre Gustavo José Giardini Lahr - Doutorando - SEM/EESC/USP

**Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino**

Resumo

Este projeto estuda novas formas de automação com robôs aplicados à aeronáutica. A hipótese a ser testada sugere que, com a inserção de métodos de aprendizagem por reforço e demonstração, os robôs passam a se comportar de forma mais versátil, reconfigurável e eficaz. O sistema desenvolvido neste projeto, trabalha com aplicações que envolvem de forma inerente o contato entre máquinas, em especial tarefas nas quais robôs entram em contato com os usuários e com o ambiente de trabalho. Devido a constante interação com usuário e ambiente, técnicas de segurança foram adicionadas de forma a viabilizar e assegurar a realização da tarefa. Para ilustrar os conceitos adotamos como exemplo a montagem de gavetas com placas material *HoneyComb*. O sistema proposto é composto de diversos dispositivos que devem se comunicar e por se tratar de um sistema em contato direto com pessoas foi preciso assegurar a qualidade de comunicação. Após obter uma garantia perante a qualidade das conexões, passou-se a fase de estudos e revisão bibliográfica que resultaram no desenvolvimento de algoritmos que facilitem a programação do manipulador robótico Kuka KR16. Foram utilizados dois enfoques de aprendizagem por reforço e por demonstração, com os algoritmos *Q-learning* e *Dynamic movement primitive*. Com a implementação dos métodos foi possível realizar a montagem completa de uma gaveta formada por cinco placas de *HoneyComb*, com apenas duas demonstrações do usuário e posições das ferramentas e placas, o sistema foi capaz de atender toda a demanda do processo de montagem.

Palavras-chave: Aprendizagem por reforço; Aprendizagem por demonstração; robôs manipuladores; mobiliário aeronáutica; *Q-learning*; *Dynamic Movement Primitive*; segurança.

Abstract

This project studies new forms of automation with robots applied to aeronautics. The hypothesis to be tested suggests that, with the introduction of machine learning methods such as reinforcement and demonstration, the robots will behave in a more versatile, reconfigurable and effective way. The system developed in this project works with applications that inherently involve contact between machines, especially tasks in which robots come into contact with users and the work environment. Due to the constant interaction with user and environment, safety techniques were added in order to make feasible and ensure the accomplishment of the task. To illustrate the concepts we take as an example the assembly of drawers with HoneyComb material plates. The proposed system is composed of several devices that must communicate and because it is a system in direct contact with people, it was necessary to ensure the quality of communication. After obtaining a guarantee regarding the quality of the connections, the phase of studies and bibliographical revision was carried out, which resulted in the development of algorithms that facilitate the programming of the robotic manipulator Kuka KR16. Two learning approaches were used for reinforcement and demonstration, with Q-learning and Dynamic movement primitive algorithms. With the implementation of the methods it was possible to complete the assembly of a drawer consisting of five HoneyComb boards, with only two user demonstrations and positions of the tools and boards, the system was able to meet all the demands of the assembly process.

Keywords: Reinforcement learning; Demonstration learning; robot manipulators; aeronautical furniture; Q-learning; Dynamic Movement Primitive; safety.

Lista de Figuras

Figura 1 - Projeções de mercado da indústria robótica japonesa	10
Figura 2 - Trajetória demonstrada e calculada utilizando DMP	16
Figura 3 – Equipamentos de Segurança aplicados a célula robóticas	18
Figura 4 – Demonstração de contraste entre robótica tradicional e colaborativa.....	19
Figura 5 – Classificação hierárquica por níveis das normas de segurança.....	20
Figura 6 – Adaptação do Fluxograma para apreciação e mitigação de riscos da ISO 12100	20
Figura 7 - Diagrama do algoritmo de execução de DMP com admitância em tempo real.....	24
Figura 8 - Dados comparativos de uma placa HoneyComb e uma placa de sólida de metal.....	25
Figura 9 – Sensores	26
Figura 10 - Manipulador robótico Kuka KR16 no Laboratório de Mecatrônica da EESC-USP ..	27
Figura 11 - Gráfico da força aplicada em função do ponto de pega.....	28
Figura 12 - Esquemático 3D da ferramenta.....	28
Figura 13 - Placas de <i>HoneyComb</i> unidas	30
Figura 14 - Desenho renderizado 3D da utilização da Prensa de madeira em uma gaveta.....	30
Figura 15 - Experimento para comparação da posição do braço medida pelo sensor inercial e manipulador robótico.....	32
Figura 16 - Controle de admitância	33
Figura 17 - Fluxograma do processo de execução de trajetória	35
Figura 18 - Diagrama para execução de DMP em tempo real com admitância	37
Figura 19 - Diagrama do fluxo de dados e informações entre o robô, sensor de força, usuário e unidade de processamento e memória	38
Figura 20 - Diagrama de operação do Software de controle	41
Figura 21 - Placas <i>HoneyComb</i> unidas com fita crepe para auxiliar a montagem	41
Figura 22 – Modelos mecânicos utilizados na simulação	42
Figura 23 - Classificações de tarefas colaborativas.....	45
Figura 24 - Interface Homem-Máquina na plataforma Android versão 1	47
Figura 25 - Interface Homem-Máquina na plataforma Android versão 2	48
Figura 26 - Avisos de erro	48
Figura 27 - Ambiente para realização do <i>pick-and-place</i>	49
Figura 28 - Tempo de execução do percurso da tarefa em função da tarefa atual	50
Figura 29 - Deslocamento no espaço entre os pontos A e B	51
Figura 30 - Deslocamento no espaço entre os pontos B e A	51
Figura 31 - Comparaçao entre primeira tentativa de ida variando a quantidade de estados	52
Figura 32 - Comparaçao entre primeira tentativa de retorno variando a quantidade de estados...	52
Figura 33 - Comparaçao entre décima tentativa de ida variando a quantidade de estados	53
Figura 34 - Comparaçao entre décima tentativa de retorno variando a quantidade de estados.....	53

Figura 35 - Comparação do número de iterações necessárias para a realização de uma tarefa completa com diversas quantidades de estados.....	54
Figura 36 - Teste verificação de erro posição final da IMU.....	55
Figura 37 - Ferramenta de pega.....	56
Figura 38 - Prensa de Madeira.....	56
Figura 39 - Prensa de madeira pressionada por grampo de aperto rápido em uma gaveta de <i>HoneyComb</i>	57
Figura 40 - Calculo da força não linear da DMP.....	58
Figura 41 - Trajetórias obtidas via DMP	59
Figura 42 - Comparação entre trajetórias com e sem distúrbio	59
Figura 43 - Inversão no sentido da trajetória no eixo Z.....	60
Figura 44 - Proporcionalidade em toda a trajetória com a alteração de pontos finais.....	61
Figura 45 - Sistema de três coordenadas utilizando DMP	61
Figura 46 - Funções Gaussianas antes do processo de aprendizagem.....	62
Figura 47 - Gaussianas Ponderadas e Forças Atrativas Desejada e Calculada	63
Figura 48 - Trajetória calculada para pega das duas placas	63
Figura 49 - Sequência de fotos do experimento com aprendizagem por demonstração utilizando o método DMP	64
Figura 50 – Aba <i>Configuration</i> da Interface Homem Máquina em plataforma Windows.....	65
Figura 51 – Aba <i>Information</i> da Interface Homem Máquina em plataforma Windows	66
Figura 52 – Aba <i>Emergency Control</i> da Interface Homem Máquina em plataforma Windows ...	67
Figura 53 - Processo para pegar e trazer a fita crepe.....	68
Figura 54 - Sequência de ações para a realizar a pega da fita crepe.....	68
Figura 55 - Distribuição das placas 2 e 3 na bancada.....	69
Figura 56 - Trajetórias geradas pelo DMP de pega para atingir os pontos de pega das placas <i>HoneyComb</i> e fita	69
Figura 57 - Trajetórias geradas pelo DMP de trazer as placas <i>HoneyComb</i>	70
Figura 58 - Trajetória do eixo de rotação	70
Figura 59 - Sequência do processo de Ajuste Fino	71
Figura 60 - Movimento no eixo rotativo C para obter a perpendicularidade	71
Figura 61 - Correções realizada pelo controle de admitância para o processo de ajuste fino	71
Figura 62 - Sequência de imagens para colocar a prensa na etapa final do processo	72
Figura 63 – Matriz de relação entre tarefas colaborativas e riscos observados.....	72
Figura 64 – Ambiente Colaborativo simulado proposto.	75
Figura 65 - Fluxograma dos Algoritmos dos modos de segurança colaborativa.....	76
Figura 66 - Gráfico do modo de segurança colaborativa condução manual	76
Figura 67 - Gráfico do modo de segurança colaborativa força e potência limitada	77

Lista de Tabelas

Tabela 1 - Metodologia HRN para cálculo da Severidade do Dano considerado	22
Tabela 2 - Metodologia HRN para cálculo da Frequência de Exposição ao Risco.....	23
Tabela 3 - Metodologia HRN para cálculo da Probabilidade de Ocorrência do Dano	23
Tabela 4 – Metodologia HRN para cálculo do número de pessoas expostas	23
Tabela 5 – Classificação do Risco (CI)	24
Tabela 6 – Tamanho das gavetas utilizadas no projeto	29
Tabela 7 - Comparação dentre formas de execução de DMP.....	57
Tabela 8 – Analise HRN do Risco 1	73
Tabela 9 – Analise HRN do Risco 2	73
Tabela 10 – Analise HRN do Risco 3.....	73
Tabela 11 – Analise HRN do Risco 4.....	74
Tabela 12 – Analise HRN do Risco 5.....	74
Tabela 13 – Analise HRN do Risco 1 após modos de segurança e ambiente colaborativo	78
Tabela 14 – Analise HRN do Risco 2 após modos de segurança e ambiente colaborativo	78
Tabela 15 – Analise HRN do Risco 3 após modos de segurança e ambiente colaborativo	78
Tabela 16 – Analise HRN do Risco 4 após modos de segurança e ambiente colaborativo	79
Tabela 17 – Analise HRN do Risco 5 após modos de segurança e ambiente colaborativo	79

Sumário

1.	Introdução.....	10
1.1.	Motivação.....	10
1.2.	Objetivos	13
2.	Revisão bibliográfica.....	14
2.1.	Aprendizagem por reforço	14
2.2.	Q-Learning	14
2.3.	Aprendizagem por demonstração cinestésica em robótica.....	15
2.4.	Divisão do ambiente em grades de ocupação.....	16
2.5.	Normas e Padrões de Segurança	17
2.6.	Controle de Admitância	24
2.7.	Mobiliário aeronáutico de Material <i>HoneyComb</i>	25
3.	Materiais e métodos	26
3.1.	Equipamentos.....	26
3.2.	Biblioteca RSI e comunicação robô-computador.....	31
3.3.	Obtenção da posição da mão do usuário	31
3.4.	Implementação do controle de admitância.....	32
3.5.	Compensação de gravidade.....	33
3.6.	Desenvolvimento do algoritmo de aprendizagem por reforço	34
3.7.	Aplicação e reprodução do <i>Dynamic Movement Primitive</i>	35
3.8.	Interfaces homem-máquina.....	37
3.9.	Planejamento e desenvolvimento do <i>Software</i> de controle	38
3.10.	Aplicação do <i>software</i> na montagem de gavetas.....	39
3.11.	Ambiente Simulado.....	42
3.12.	Adequação do software a especificação técnica ISO 15066	43
4.	Resultados	46
4.1.	Comunicação robô e computador.....	46
4.2.	IHM Android.....	46
4.3.	Experimento <i>pick and place</i> utilizando algoritmo <i>occupancy grids</i> com <i>Q-learning</i>	49
4.4.	Experimento de aquisição e comparação de dados dos sensores inerciais.....	54
4.5.	Projeto da ferramenta de pega	55
4.6.	Elaboração da Prensa de Madeira	56
4.7.	Comparação dentre as formas de execução do algoritmo DMP.....	57
4.8.	Observações para alteração de pontos finais e iniciais da DMP	59
4.9.	Experimento de pega de duas placas <i>HoneyComb</i> utilizando DMP <i>offline</i>	61
4.10.	IHM Windows.....	64
4.11.	Experimento de Montagem Completa da Gaveta.....	67
4.12.	Apreciação de Riscos	72
4.13.	Ambiente de Colaboração Simulado Proposto.....	74
4.14.	Modos de Segurança Colaborativa.....	75
4.15.	Apreciação de Riscos após modos de segurança e ambiente colaborativo	77
5.	Conclusão	80

6. Referências bibliográficas	82
Apêndice A – Pacote XML adaptado do Manipulador KR16 para computador.....	85
Apêndice B – Pacote XML adaptado do computador para o Manipulador KR16.....	86
Apêndice C – Desenho Técnico da Ferramenta de Pega.....	87
Apêndice D – Desenho Base de Contato da Prensa de Madeira	88
Apêndice E – Desenho Tabua de Ajuste da Prensa de Madeira.....	89
Apêndice F – Desenho Junção de Tabuas da Prensa de Madeira.....	90
Apêndice G – Pacote XML da Comunicação Computador e Celular	91
Anexo A – ISO/TS 15066 classificação de forças de contato conforme resistência do corpo humano (ROBOTIQ, 2016).....	92

1. Introdução

1.1. Motivação

A utilização de robôs tem crescido e se tornado mais presente no ambiente fabril. Conforme se pode ver na Figura 1, o investimento no setor robótico japonês deve ter um aumento de quase seis vezes em 20 anos. Isso demonstra a necessidade de se incorporar técnicas de programação facilitada, permitindo que usuários com menos horas de treinamento estejam capacitados a utilizar tais equipamentos. Essas técnicas visam também facilitar a definição de tarefas complexas com menor necessidade de tempo de programação, como a aplicação de inteligência artificial com um enfoque de auxiliar na programação da rotina ou tarefa.

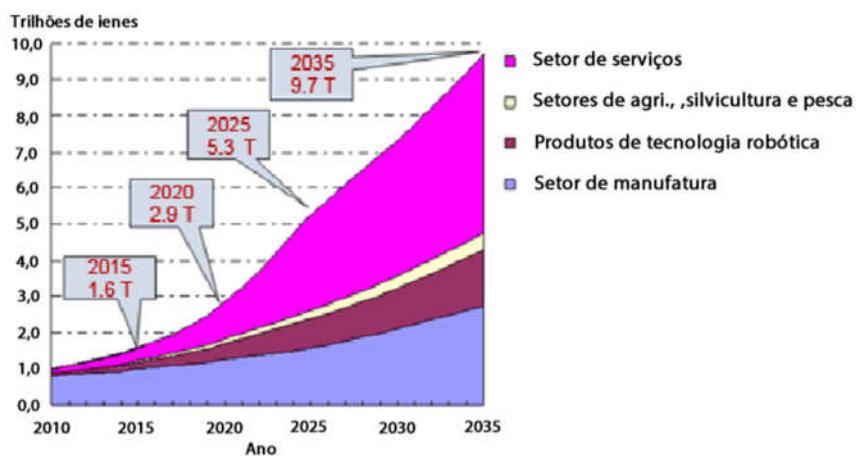


Figura 1 - Projeções de mercado da indústria robótica japonesa (adaptado de TAJI et al., 2013)

Aprendizado de máquina envolve diversos tipos de aprendizagem, tais como demonstração e reforço, esse é uma vertente importante dentro da inteligência artificial. O aprendizado por demonstração através de ações do usuário permite que o robô se adapte e aprenda como exercer sua tarefa no ambiente, por meio de uma interface de comunicação entre o usuário e a máquina (ARGALL et al., 2009). Destacamos, também neste contexto, os métodos de aprendizagem por reforço, os quais possibilitam que um sistema possa melhorar seu desempenho utilizando experiências anteriores de falhas e sucessos (KAELBLING; LITTMAN; MOORE, 1996)

A fim de compreender o comportamento do ser humano, como professor, em um sistema de aprendizagem por reforço, Thomaz e Breazeal (2006) utilizaram o método *Q-learning* em um ambiente virtual. Os autores desenvolveram duas interfaces: na primeira só é possível fornecer recompensas, enquanto que na segunda é possível guiar, de modo cooperativo, e recompensar o

robô. O processo com guiamento obteve 12,1% menos de tentativas falhas em relação a outra interface, sugerindo que um maior envolvimento do usuário leva a melhoria do desempenho.

Essa interação com o usuário do equipamento, denomina-se uma operação colaborativa, a qual consiste em tarefas nas quais o ser humano tem contato direto com o robô, como toque, proximidade e auxílio nas decisões, durante essas atividades a máquina continua a se movimentar e desenvolver suas tarefas (OMRON ELECTRONICS LLC, 2015a). Diversas normas de segurança de segurança foram desenvolvidas a fim de tornar esse ambiente estável e seguro, a ISO/TS 15066 é focada nas aplicações de robótica colaborativa, vindo a complementar a ISO 10218, ambas são internacionais (ABB, 2014).

A robótica, no Brasil, é contemplada pela NR 12, Segurança no Trabalho em Máquinas e Equipamentos. Entretanto, esta norma acaba limitando a atuação da robótica colaborativa, devido a diversos fatores, tornando-se defasada ao desenvolvimento atual. Por exemplo, o item 12.41 e 12.39 que obrigam a isolação do equipamento por barreiras físicas fixas ou móveis, impedindo o acesso enquanto a máquina estiver em movimento. Com essas restrições, a robótica colaborativa acaba sendo inviabilizada nacionalmente, diminuindo a competitividade da indústria nacional (MTE, 2011).

A implementação das técnicas de colaboração ainda não é uma realidade presente em todos os processos fabris, que empregam largamente programação ponto-a-ponto em robôs industriais, demandando longos períodos de programação para tarefas relativamente complexas. Tal modelo de programação não é flexível, pois caso exista uma alteração no ambiente ou nas peças do processo, toda a tarefa pode ser comprometida de maneira pontual ou catastrófica, podendo chegar a perda de peças, ferramentas ou paralização de linhas de produção. Caso ainda deseje-se reconfigurar o sistema, um novo tempo de máquina parada será necessário para recrutar equipe técnica responsável para efetuar reprogramação. A implementação de aprendizado de máquina permite ao robô se ajustar às novas condições, sendo treinado enquanto o usuário coopera com ele.

Como desvantagem adicional da abordagem atual de programação de robôs, podemos citar ainda a necessidade de ocupar especialistas em programação de manipuladores, devido à dificuldade de integração do sistema em seu ambiente de trabalho. Como uma alternativa para tornar esse processo mais intuitivo ao usuário, Fernandes (2014) utilizou três sensores iniciais acoplados ao braço humano, permitindo obter dados de posição do centro da ferramenta, que é a própria mão do usuário, por meio da cinemática direta. Ao obter esse ponto, Fernandes (2014)

controlou um robô SCARA IBM7545, de forma a espelhar o movimento do braço na área de trabalho do robô. Esta interface torna mais intuitiva a movimentação e reprodução de uma tarefa designada pelo usuário.

No caso específico da fabricação de mobiliários aeronáuticos, não há uma constância na produção, uma vez que existem diversos modelos, cujos componentes variam de acordo com a demanda existente. Esta variabilidade desfavorece o uso de manipuladores robóticos convencionais. Por outro lado, o processo atual de fabricação de mobiliários ainda é preponderantemente manual, e necessita do emprego de diversas pessoas e a qualidade final do produto não é garantida, pois está sujeito a erros de medição, posicionamento espacial, além de possíveis variações entre peças do mesmo modelo. É nesse contexto que este trabalho se insere.

Para simplificar o problema em questão, este projeto foca os estudos no caso de auxiliar a automação na produção de gavetas a partir de chapas de material *HoneyComb* com a implementação de técnicas de colaboração em um robô, permitindo ao operário trabalhar junto com o robô. O emprego de um manipulador robótico com um algoritmo de aprendizagem por reforço e demonstração acelera o processo unindo a força, agilidade e repetibilidade da máquina com o critério de decisão e condução do ser humano, provendo assim ao sistema, caso necessário, a propriedade de configurar-se com as necessidades impostas pelo processo.

A capacidade de inserir métodos de aprendizagem e controle em um robô convencional Kuka KR16, demonstra as possibilidades de implementação de algoritmos, buscando otimizar o sistema e os reais limites de tais manipuladores, os quais em alguns casos, são trocados por robôs mais modernos, sendo que os já presentes são capazes de atender as necessidades e serem mais cooperativos e flexíveis.

1.2. Objetivos

1.2.1. Objetivo geral

Elaborar exemplos reais de procedimentos de automação para processos de montagem de gavetas com estrutura *HoneyComb*, com a utilização de um manipulador robótico com métodos de aprendizagem de máquina e cooperação com o usuário.

1.2.2. Objetivos específicos

- Elaborar uma comunicação estável e segura utilizando protocolo UDP entre computador e robô.
- Desenvolver um *software* em linguagem C# para controle do manipulador Kuka KR16 através de um computador, de forma colaborativa. O computador também será responsável por todo o processamento e manipulação dos dados do sistema composto pelos sensores iniciais, sensor de força e manipulador.
- Estudar e desenvolver formas intuitivas de programação baseadas na teoria de aprendizagem por demonstração, utilizando os métodos desenvolvidos por Schaal (2003) e Ijspeert (2013).
- Implementar os métodos de aprendizagem desenvolvidos ao *software* de controle, a fim de tornar o sistema flexível e reduzir a complexidade de programação exigida.
- Aplicar metodologias de segurança ao sistema desenvolvido, de forma a assegurar a integridade de todas as pessoas e equipamentos envolvidos.
- Reproduzir um ambiente real de montagem de gavetas e utilizar o manipulador de forma colaborativa com algoritmo de aprendizagem para dar apoio e suporte ao trabalhador na fase de montagem da produção.

2. Revisão bibliográfica

2.1. Aprendizagem por reforço

Aprendizagem por reforço é proveniente da psicologia, baseado na metodologia de Skinner (DELPRATO; MIDGLEY, 1992), e é fundamentado na seguinte lógica: toda ação de um agente gera uma reação do ambiente, caso a reação seja positiva a ação tende a se repetir, caso negativa, a se extinguir. Nos casos averiguados por Skinner (DELPRATO; MIDGLEY, 1992), esse agente pode ser, por exemplo, um macaco que recebe gotas de suco de laranja como recompensa por realizar uma ação desejável ou um aluno que estuda e recebe uma nota alta. Sendo assim, a área de inteligência artificial se apropria dos estudos de Skinner para desenvolver algoritmos capazes de reproduzir esta forma de aprendizado, como o *Q-Learning*.

Para a implementação do aprendizado por reforço, é necessário que o meio de trabalho seja modelado como uma Cadeia de Markov: um espaço conhecido e mapeado em uma quantidade finita de estados discretos com ações definidas. Define-se uma política, a qual orienta o agente dentre os estados e uma matriz de probabilidades de transição, relacionando o estado anterior, a ação tomada e o novo estado. O sistema, por meio de tentativas de realizar uma ação, recebe uma recompensa negativa ou positiva referente ao desempenho (GOSAVI, 2008). Um conjunto de recompensas positivas podem vir a se tornar um aprendizado, o qual é utilizado caso o problema ou a situação se repetir. A utilização de soluções antigas favorece o desempenho do robô, formando uma memória de práticas para resolver problemas conhecidos por meio das experiências.

2.2. Q-Learning

O algoritmo *Q-Learning* é baseado em valores conhecidos como Q , que são atualizados a cada etapa do processo de acordo com a equação (1) (WATKINS, 1989): α corresponde ao fator de aprendizagem que varia de zero à um e determina a velocidade de aprendizado, sendo que, para valores próximos de um, o aprendizado é potencializado e próximos de zero dificultam o processo. Valores mal definidos de α podem fazer com que o sistema não encontre a solução, logo é preciso determinar um valor correto e condizente com o processo; γ é o fator de desconto, relacionado ao novo estado (s') em que o agente se encontra após exercer uma ação, um fator essencial para processos que possuem uma sequência para encontrar a solução; a letra r é a recompensa, varia de acordo com a intenção e da forma que o sistema foi implementado. A função Q é dependente do estado atual (s) e ação (a), então para cada ação presente em cada estado há um valor referente.

$$Q[s, a] \leftarrow Q[s, a] + \alpha(r + \gamma \max_{a'} Q[s', a']) - Q[s, a]) \quad (1)$$

No processo existe uma política que estabelece padrões para a seleção de ações em cada estado, pode variar de acordo com a necessidade do sistema. Entretanto, para todos os algoritmos de aprendizado o valor Q age diretamente na escolha das ações, possuindo uma grande influência na política de seleção. Watkins e Dayan (1992) provaram que o algoritmo apresenta convergência para um desempenho ótimo, entretanto o processo de seleção da recompensa correta é essencial para atingir esse desempenho, como a punição dos custos desejados por exemplo.

2.3. Aprendizagem por demonstração cinestésica em robótica

Um modo de tornar a programação do robô mais intuitiva pode ser obtido com a inserção de algoritmos de aprendizagem por demonstração, onde o usuário é capaz de mostrar ao sistema o que ele deseja, isto pode ser realizado de diversas maneiras como teleoperação, observação, sensores acoplados ao usuário, imitação e cinestésica (ARGALL et al., 2009). Todas as formas possuem vantagens e desvantagens, logo é necessário conhecer a situação e o ambiente, onde a tarefa é executada.

A demonstração cinestésica possui pontos positivos, como o *feedback* visual e de tato direto ao usuário no momento da demonstração, mas possui como negativo a necessidade da proximidade ao robô. Como exemplo de aspecto contrário no uso da teleoperação destacamos as eventuais dificuldades para fornecer um *feedback* em tempo real ao usuário. Por outro lado a distância entre usuário e robô inerente, a essa técnica pode ser entendida com um aspecto vantajoso. Para realizar a demonstração cinestésica é necessário algum controle por força, seja este um controle mais simples, como o complacente (proporcional) ou mais complexo como, o de admitância (ARGALL et al., 2009).

Os dados provenientes de uma demonstração devem ser processados, a fim de obter parâmetros que sejam suficientes para a repetição de tal movimento ou ação. Dada esta necessidade, Schaal (2003) desenvolveu um algoritmo capaz de aprender uma trajetória independente do tempo e das posições iniciais e finais, conhecido como *Dynamic Movement Primitive* (DMP).

2.3.1. Dynamic movement primitive

Um DMP consiste em dois sistemas diferenciais, o canônico e o de transformação. Estes são responsáveis pela reprodução da demonstração, o canônico consiste em um sistema que inicia em zero e converge para um, indicando ao sistema, como uma transição temporal, da posição inicial, representada pelo zero, até a posição final, um, agindo de forma similar ao tempo. O sistema de transformação consiste em equações diferenciais de um modelo dinâmico com uma força ponderada não linear, resultando em uma aceleração ao ser solucionado, a partir dessa é possível encontrar a velocidade e enfim a trajetória de posição, por meio de integração discreta.

A robustez perante a alteração dos pontos iniciais e finais fornecem ao sistema a capacidade de aprender o intuito de uma ação, já que estas posições podem conter peças, objetos ou pontos estratégicos, por exemplo, em Schaal (2003), como prova de conceito, foi realizado uma demonstração de como bater em uma bolinha de tênis, esta habilidade pôde ser reproduzida diversas vezes, mesmo com a alteração da posição da bolinha. Na Figura 2 foi demonstrado ao sistema como desenhar uma rota semelhante a letra “a” e alterando o ponto final, o formato é mantido na reprodução.

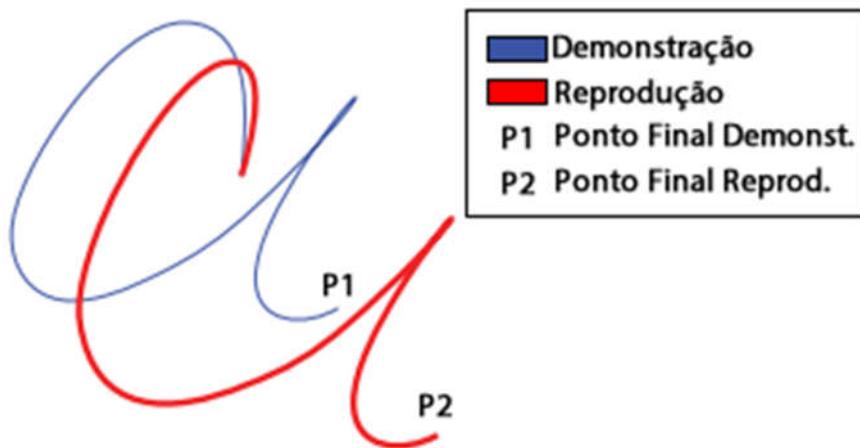


Figura 2 - Trajetória demonstrada e calculada utilizando DMP (adaptado de IJSPEERT et al., 2013)

2.4. Divisão do ambiente em grades de ocupação

Grades de ocupação (*occupancy grids*) consiste em discretizar o espaço do ambiente em que o robô se encontra (ELFES, 1989). A região escolhida é então dividida em várias células com dimensões menores, este número pode variar dependendo da aplicação.

Esta divisão reduz a complexidade de movimentação, pois, a grande quantidade de posições possíveis, passam a ser uma área ou volume do espaço. As células possuem um valor de ocupação, o qual é definido pelo próprio sistema. Sensores auxiliam na determinação da situação de cada célula, podendo ser binária, ou uma probabilidade, de ocupação.

Os sensores funcionam constantemente, informando ao sistema as situações das células próximas e com isso é possível evitar falhas no deslocamento. Em Elfes (1989), este método é utilizado na robótica móvel para a locomoção em ambientes desconhecidos, através de um sensor de ultrassom o robô obtia informações sobre o ambiente e atualizava os valores de ocupação das grades, com base nos valores encontrados era possível atingir a posição desejada e desviar das células ocupadas.

2.5. Normas e Padrões de Segurança

O ambiente fabril, sem os devidos cuidados, pode se tornar um local inseguro aos colaboradores que ali desenvolvem suas funções. As máquinas e equipamentos precisam ser seguras, de forma a garantir a saúde e integridade física dos trabalhadores e de outros equipamentos, visando este tópico, diversas normas foram desenvolvidas por órgãos governamentais e não governamentais. No Brasil, o Ministério do Trabalho e Emprego (MTE) é o responsável por tais normas. Em 8 de julho de 1978, o MTE iniciou a implementação de 28 Normas Regulamentadoras (NR), hoje, as mesmas foram atualizadas e 8 normas foram adicionadas ao conjunto.

A norma vigente está associada a localização da aplicação da mesma, no âmbito nacional, utiliza-se a NR, entretanto existem outras instituições, como a ISO (*International Organization for Standardization*) e OSHA (*Occupational Safety and Health Administration*). Essas organizações buscam propor técnicas e metodologias para tornar e garantir a segurança dos operários, máquinas e equipamentos.

No ambiente nacional, a NR 12 regulamenta a grande maioria dos processos fabris, neste caso a robótica, a negligência desta pode ocasionar multas e punições, que podem até inviabilizar projetos. A NR 12 ainda não contempla de forma específica e suficiente a robótica colaborativa. No cenário internacional, a ISO/TS 15066, *Collaborative Robots*, foi elaborada para regulamentar processos colaborativos, incrementando a ISO 10218, *Robots and robotic devices – Safety requirements for industrial robots*. A ISO/TS 15066 está ainda em processo de desenvolvimento,

logo é denominada como uma especificação técnica (*Technical Specifications*), já a ISO 10218 é um Padrão Formal (*Formal Standard*) (ROBOTIQ, 2016).

Mesmo possuindo diferenças, as normas com foco em maquinário industrial, convergem para um objetivo comum, de que a máquina, após o acontecimento de uma falha ou problema de qualquer natureza, seja capaz de ir para um estado seguro, o qual varia, podendo ser a parada do fornecimento de energia, acionamento de freios mecânicos, entre outros.

2.5.1. Análise da NR12 aplicada a robótica colaborativa

O acesso ao espaço de trabalho do robô deve ser limitado e restrito por barreiras físicas quando o mesmo estiver operando, de acordo com os itens 12.41a para barreiras fixas e 12.41b para barreiras móveis e estes ainda são complementados pelos itens 12.45 e 12.46 que impõe a utilização de dispositivos de segurança intertravados e que só podem liberar o acesso ao espaço de trabalho do robô ou manipulador robótico quando as partes móveis estiverem desativadas impedindo o risco de lesões. Com isso, não é possível que usuário se aproxime ou conduza o manipulador.



a)



b)

Figura 3 – Equipamentos de Segurança aplicados a célula robóticas: a) Dispositivo de intertravamento (MELO, 2014); b) Célula robótica com componentes de segurança (PROCTER, 2017)

Ao invés das barreiras físicas é possível inserir sistemas de segurança, tais como sensores ópticos de presença, cortina de luz, tapete e *scanner* de área, contudo é necessário adequar e inseri-los conforme o anexo I da NR 12, o qual determina as distâncias mínimas entre o manipulador

robótico e o sistema. No caso de uma cortina de luz, por exemplo, a formula utilizada é composta pela distância do ponto de detecção da cortina de luz até a zona de perigo em milímetros (S), parâmetro de velocidade de aproximação do corpo (K), tempo necessário para realizar a parada do movimento (T) e a distância adicional até o momento em que a cortina detecta a intrusão a zona de perigo (C).

$$S = (K \cdot T) + C \quad (2)$$

Uma atividade colaborativa consiste na tarefa onde o robô possui interação direta, seja esta contato ou proximidade, com o operador humano (OMRON ELECTRONICS LLC, 2015b). Com base nos requisitos citados na acima, o ganho produtivo proposto pela aplicação é inviabilizado, já que não é possível realizar essa interação com o equipamento ligado ou em movimento. Além disso, com base no item 12.40 e 12.40.1 da NR12, todo momento em que ocorrer o contato do operador com o manipulador robótico é necessário realizar um rearne manual após a paralização do equipamento.



a)



b)

Figura 4 – Demonstração de contraste entre robótica tradicional e colaborativa: a) aplicação com robótica colaborativa (OMRON ELECTRONICS LLC, 2015a); b) aplicação com robótica tradicional (TECNO INGENIERIA,)

2.5.2. Cadeia Hierárquica das Normas e Referências

As normas são baseadas em uma cadeia hierárquica, dividida por níveis: A; B; e C. O nível mais geral é o A, onde é especificado os requisitos gerais para a segurança de máquina, a norma IEC 61508, *Functional Safety*, formalizada pela *International Electromechanical Comission*,

engloba todo e qualquer equipamento elétrico, eletrônico programável e não programável, os quais a falha pode acarretar em um dano ao ambiente ou a pessoas.

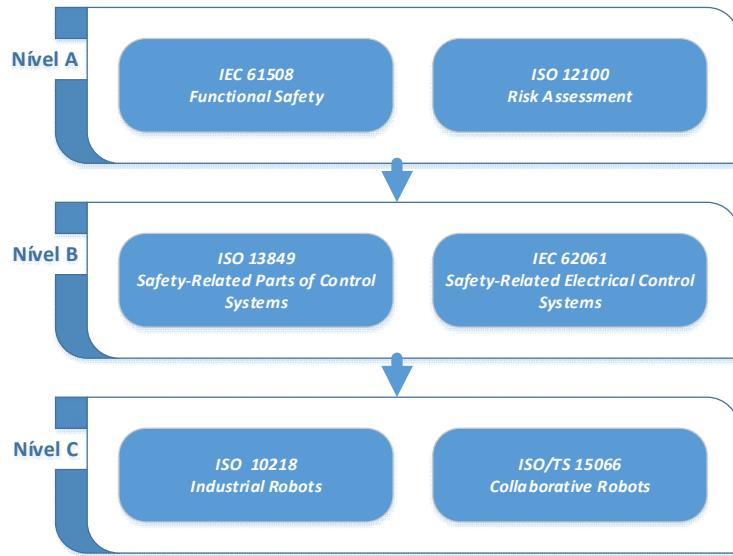


Figura 5 – Classificação hierárquica por níveis das normas de segurança

A ISO 12100, *Risk Assessment*, fornece base para realizar toda a análise e mitigação de riscos associados a máquinas em ambientes industriais. Conforme o fluxograma adaptado, a norma fornece ferramentas analíticas e quantitativas para realizar o processo de identificação e redução dos riscos.

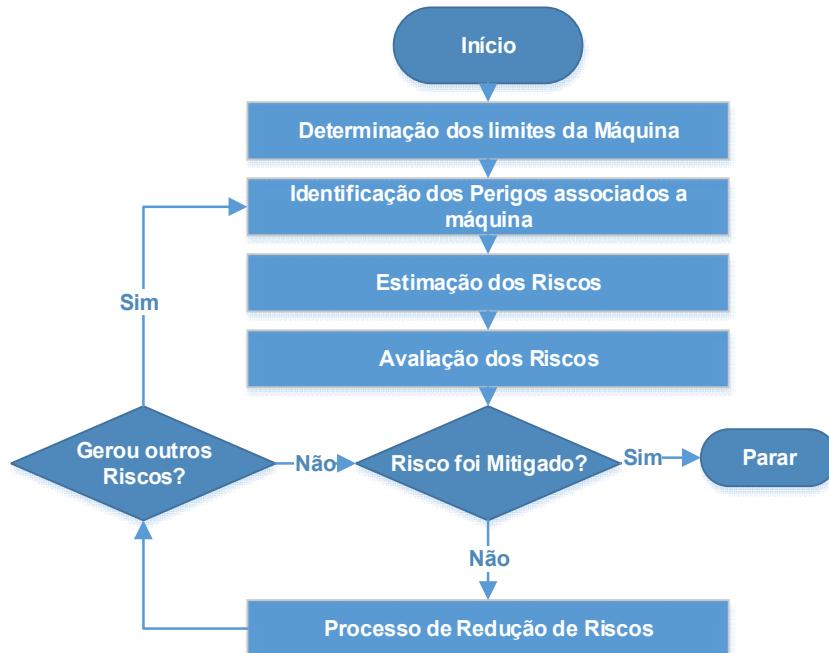


Figura 6 – Adaptação do Fluxograma para apreciação e mitigação de riscos da ISO 12100 (BECKER, 2015)

As normas do nível B (Figura 6), apresentam categorias e classificações para os equipamentos de segurança. A ISO 13849 possui duas formas de classificações: o nível de desempenho (*Performance Level – PL*), baseado no tempo médio para falhas perigosas (*Mean Time to Dangerous Failure – MTTF*), ou seja, na confiabilidade; e a categoria, em ordem crescente zero, um, dois, três e quatro, que condiz com a forma de ação arquitetura do sistema mediante a uma falha, por exemplo, a categoria mais alta, quatro, refere-se que quando ocorrer uma falta, o sistema sempre irá atingir o estado de segurança e a falha é detectada a tempo de prevenir a parada do funcionamento do sistema de segurança.

A IEC 62061, embora possua o mesmo objetivo da ISO 13849, a classificação é realizada com diferentes parâmetros: a probabilidade de falha perigosas por hora (*Probability of dangerous failures per hour – PFHd*); a arquitetura do sistema, redundância e diagnóstico de falhas; causas comuns de falhas (*Common cause failure – CCF*); falha na comunicação e transmissão do sinal; e Interferência Eletromagnética. A partir desses, estabelece o nível de integridade de segurança (*Safety Integrity Level – SIL*) (SCHNEIDER, 2015).

No nível C, as normas são mais específicas e buscam tratar de forma mais direta as aplicações, no caso da ISO 10218 tem o intuito de normatizar as atividades que envolvam um maquinário robótico. Entretanto esta ainda não supria as necessidades da robótica colaborativa, mediante a essa motivação, surgiu a demanda por algo mais relacionada a aplicação colaborativa com isso foi elaborada a ISO/TS 15066.

2.5.3. ISO/TS 15066

Essa especificação técnica busca padronizar e assegurar os meios de trabalho colaborativo, um dos ideais principais dessa metodologia consiste em: “Se um contato entre o robô e um humano é permitido e um contato acidental ocorra, então este contato não deve resultar em dano ou ferimento ao humano e/ou robô” (ROBOTIQ, 2016). A especificação técnica possui fornece informações importantes sobre as forças limitantes do contato entre o robô e o corpo humano, a velocidade e potência máxima permitida ao robô, critérios de desenvolvimento e explanações sobre as operações colaborativas.

As operações colaborativas são divididas em quatro grupos: parada monitorizada de segurança (*safety-rated monitored stop*); condução manual (*hand-guiding*); monitoramento de velocidade e distância (*speed and separation monitoring*); e Força e Potência limitada (*power and*

force limiting). Para cada grupo são fornecidos limites de operação, aonde acredita-se que quando estes são respeitados é possível assumir que se trata de uma aplicação segura ao ambiente, seres humanos e máquinas. Em todos os tipos de operação é necessário definir quais são os espaços colaborativos no espaço de trabalho do manipulador robótico para isso é preciso assegurar que não haja a interação com o operador fora desta área pré-estabelecida (ABB, 2015).

2.5.4. Número de Classificação de Risco (HRN)

Uma metodologia desenvolvida para realizar a apreciação de riscos é HRN (*Hazard Rating Number*) (CICHORSKI, 2015) este método busca classificar o tipo de risco, a fim de viabilizar a mitigação do mesmo. Essa abordagem é largamente utilizada e está alinhada com a norma ISO 12100.

O objetivo da HRN é encontrar a classe de Risco (CI). Está é obtida através da multiplicação de quatro índices (equação (3)), os quais possuem tabelas para realizar a quantificação. Os índices são: Severidade do Dano considerado (Se), Frequência de Exposição ao Risco (Fr), Probabilidade de Ocorrência do Dano (Pr) e Número de Pessoas Expostas ao Risco (NP). Após o cálculo é possível classificar o risco, determinando se o mesmo é adequada a aplicação proposta.

$$CI = Fr * Se * Pr * NP \quad (3)$$

Tabela 1 - Metodologia HRN para cálculo da Severidade do Dano considerado

Severidade do Dano (Se)	Valor
Morte	15
Perde de 2 membros/olhos ou doença grave (irreversível)	8
Perda de 1 membro/olho ou doença grave (temporária)	4
Fratura - Ossos importantes ou doença leve (Permanente)	2
Fratura - Ossos menores ou doença leve (Temporária)	1
Lacerção/Efeito leve na saúde	0.5
Arranhão/Contusão	0.1

Tabela 2 - Metodologia HRN para cálculo da Frequência de Exposição ao Risco

Frequência de Exposição ao Risco (Fr)	Valor
Constantemente	5
Poucas vezes ao dia	4
Diariamente	2.5
Semanalmente	1.5
Mensalmente	1
Anualmente	0.2
Raramente	0.1

Tabela 3 - Metodologia HRN para cálculo da Probabilidade de Ocorrência do Dano

Probabilidade de Ocorrência do Dano (Pr)	Valor
Certamente	10
Esperado	8
Provável	5
Alguma Chance	2
Não Esperado	1
Impossível	0.03

Tabela 4 – Metodologia HRN para cálculo do número de pessoas expostas

Número de Pessoas Expostas (NP)	Valor
Mais de 50 pessoas	12
16-50 pessoas	8
8-15 pessoas	4
3-7 pessoas	2
1-2 pessoas	1

Tabela 5 – Classificação do Risco (CI)

HRN Valor	Risco	Descrição	Ação
0 - 5	Insignificante	Oferece um risco muito baixo para a segurança e saúde	Nenhuma ação requerida
5 - 50	Baixo porém significativo	Contém riscos necessários para a implementação de medidas de controle de segurança	Melhoria recomendada
50 - 500	Alto	Oferece possíveis riscos, necessitam que sejam utilizadas medidas de controle de segurança urgentemente.	Necessária ação de melhoria
500 +	Inaceitável	É inaceitável manter a operação do equipamento na situação que se encontra	Necessária ação de melhoria

2.6. Controle de Admitância

A interação entre corpos com impedâncias mecânicas distintas, nem sempre é uma tarefa simples, além da possibilidade de danificar o ambiente ou o equipamento, pode ocorrer instabilidade, inviabilizando aplicações de contato e adaptação da trajetória com o ambiente. Para correção dos problemas citados, controles baseados na força de contato buscam estabilizá-la e ameniza-la, em Hogan (2011) foi introduzido o controle de impedância e admitância, os quais buscam relacionar posição, velocidade e força através da simulação de um sistema dinâmico mecânico de segunda ordem, o sistema simulado passa a ser utilizado como controlador da malha

Figura 7.

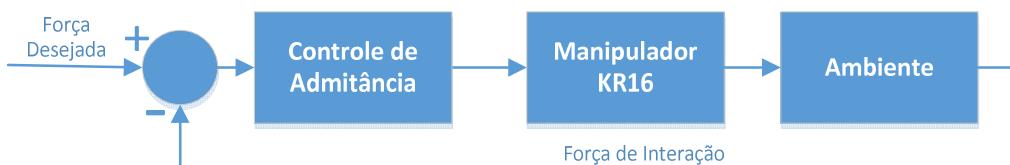


Figura 7 - Diagrama do algoritmo de execução de DMP com admitância em tempo real

O modelo mecânico utilizado no controlador é bem conhecido e aceito pela comunidade, corresponde ao modelo de um sistema massa, mola e amortecedor, A equação (4) corresponde ao sistema citado, onde K é a constante de mola, B a de amortecimento e M a inércia. Em Lahr et al. (2016), formas de implementação do controle de admitância foram discutidas e dois formatos foram apresentados o integrativo e o discretizado, ambos apresentaram desempenho similar para

as mesmas situações, demonstrando que o sistema foi capaz de se adaptar e atingir as forças de interação desejadas.

$$\frac{X(s)}{F(s)} = \frac{1}{Ms^2 + Bs + K} \quad (4)$$

2.7. Mobiliário aeronáutico de Material *HoneyComb*

Este tipo de material possui propriedades favoráveis a aplicações aeronáuticas, especialmente na produção de peças mobiliárias, como armários e gavetas, devido a sua alta rigidez e baixo peso, por outro lado, impõe dificuldades em termos de qualidade de usinagem e acabamento.

Devido ao seu formato interno, geometricamente, similar a uma colmeia de abelhas, as forças aplicadas a superfície são divididas entre as estruturas, na Figura 8 é possível observar uma relação média realizada pela empresa HEXCEL, a qual demonstra as propriedades favoráveis presentes nas placas de *HoneyComb*, mostrando o aumento mínimo de peso, entretanto um ganho positivo em propriedades mecânicas como força e rigidez (HEXCEL COMPOSITES, 1999).

	Placa sólida de Metal	Placa de HoneyComb 1	Placa de HoneyComb 2
<i>Rigidez Relativa</i>	100	700 <i>7 vezes mais rígida</i>	3700 <i>37 vezes mais rígida</i>
<i>Força Relativa</i>	100	350 <i>3,5 vezes mais forte</i>	925 <i>9,25 vezes mais forte</i>
<i>Peso Relativo</i>	100	103 <i>3% de aumento no peso</i>	106 <i>6% de aumento no peso</i>

Figura 8 - Dados comparativos de uma placa HoneyComb e uma placa de sólida de metal (HEXCEL COMPOSITES, 1999)

3. Materiais e métodos

3.1. Equipamentos

Foram utilizados um robô manipulador, um sensor de força, dois sensores inerciais, um *notebook*, uma garra pneumática, um celular, uma prensa de madeira e placas de material *HoneyComb*. A descrição de cada um segue nas seções de 3.1.1 a 3.1.7.

3.1.1. Sensor de forças e torques - 6 eixos DAQ F/T Delta SI-660-60

Instalado na flange após o punho do robô, este sensor da ATI Industrial Automation (Figura 9a) permite o conhecimento das forças de contato em tarefas que exigem manipulação do ambiente ou de uma peça. Este sensor possui uma resolução de $\frac{1}{8}$ N nas forças dos eixos X e Y e $\frac{1}{4}$ N em Z, e torques de 10/1333 Nm nos três eixos. Essa configuração é mais do que suficiente para a aplicação desejada, visto que o fator limitante do processo é a carga máxima suportada pelo Kuka KR16 que é 16 Kg, aproximadamente, 156.96 N, enquanto o sensor de força é limitado a um intervalo de 660 N nos eixos X e Y e 1980 N no Z (ATI INDUSTRIAL AUTOMATION,).

O sensor de força deve fazer parte do desenvolvimento, uma vez que é essencial em atividades que envolvem contato e interação com o ambiente, cenário que faz parte da proposta desse projeto na montagem de mobiliário aeronáutico. Dados de força são a forma de realimentar o sistema e assim tomar a atitude necessária para um correto cumprimento da tarefa, que faz parte do cenário de aprendizado e inteligência do robô.



a)



b)

Figura 9 – Sensores: a) Sensor de força DAQ F/T Delta; b) IMU Razor 9DOF

3.1.2. Manipulador robótico

O modelo KR16 da empresa Kuka *Robotics* GmbH (Figura 10) é um robô manipulador para baixas cargas, até 16 kg, desenvolvido para aplicações industriais e, portanto, com controle de posição validado. O robô oferece diferentes funcionalidades integradas para facilitar a sua implementação em diversas áreas como montagem, soldagem, entre outros. Possui 6 graus de liberdade com uma repetibilidade de $\pm 0,05$ mm de posicionamento do ponto de referência na extremidade (Tool Center Point – TCP) e um envelope de trabalho de até 1610 mm, descontando o alcance adicional da ferramenta a ser inserida (KUKA ROBOTER GMBH, 2004).



Figura 10 - Manipulador robótico Kuka KR16 no Laboratório de Mecatrônica da EESC-USP

Este robô possui controlador KR C2 com o *software* KSS 5.4 (*Kuka System Software*). Para interface este robô utiliza Windows XP *Embedded*, mas como *software* de tempo real, o VxWorks, da empresa americana Wind River.

3.1.3. Computador

O computador de desenvolvimento utilizado neste projeto é um *notebook* modelo Dell 15R 5537, com 8 GB de memória RAM, sistema operacional Microsoft Windows 10 e processador Intel Core I7-4 500U 3GHz.

3.1.4. Garra pneumática SMC MHZ2 25D

Uma garra paralela de dois dedos de dupla ação produzida pela SMC *Pneumatics* é a ferramenta utilizada nos experimentos. Para as tarefas de preensão em questão não há a necessidade de exercer grande força, já que o material *HoneyComb* é leve. A garra é capaz de exercer 65 N para pega externa e 104 N para a interna, neste caso utiliza-se a externa com pressão de 0,5 Mpa conforme a Figura 11b (SMC PNEUMATICS,).

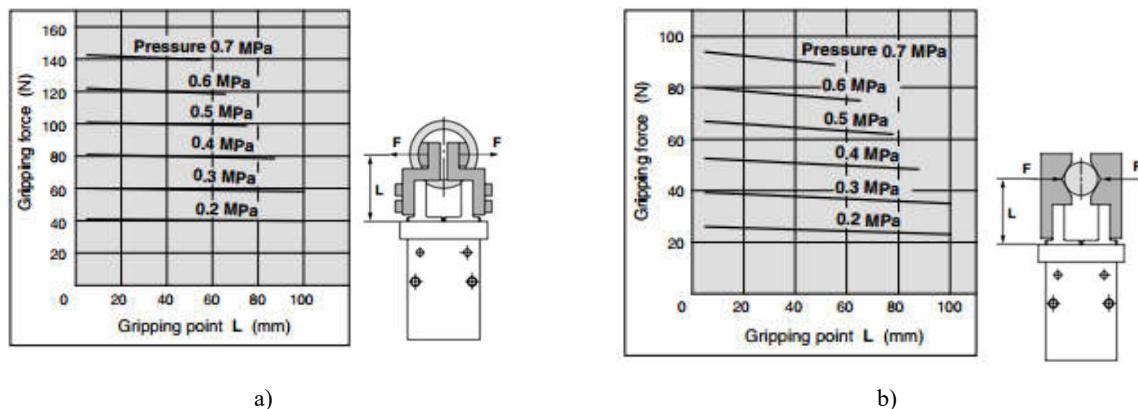


Figura 11 - Gráfico da força aplicada em função do ponto de pega: a) pega interna; b) pega externa (adaptado de SMC PNEUMATICS, [s.d.])

Com o intuito de melhorar o contato da garra com as gavetas nas tarefas realizadas, uma ferramenta adicional foi desenvolvida utilizando impressão 3D com plástico ABS. Esta ferramenta acoplada aos dois dedos da garra aumenta a superfície de contato. Com isso, é capaz de transferir com mais facilidade as forças e torques aplicados a placa de *HoneyComb*, além de evitar pontos de alta pressão no momento da pega. Na Figura 12 está o desenho do design proposto para a ferramenta de pega.

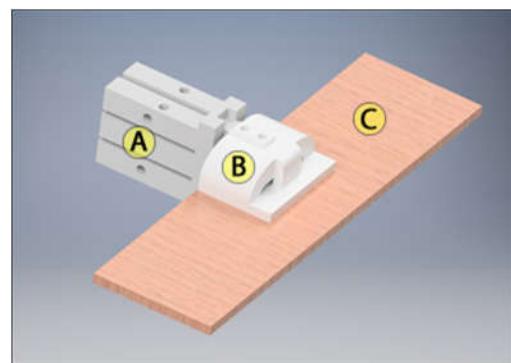


Figura 12 - Esquemático 3D da ferramenta: garra MHZ2 25D (A), ferramenta adicional (B) e esquemático da placa *HoneyComb* (C)

3.1.5. Sensor inercial (IMU – *Inertial Measurement Unit*) Razor 9DOF

Produzido pela empresa SparkFun, é composta por quatro componentes: magnetômetro de três eixos, acelerômetro de três eixos, giroscópio de três eixos e *chip* de processamento ATmega 328 de 8MHz com *bootloader* da plataforma Arduino. O sistema integra o processamento das nove entradas de dados provenientes dos sensores, que no fim do processo se fundem para fornecer três valores correspondentes aos ângulos de Euler (*Yaw*, *Pitch* e *Roll*) por comunicação serial, via *Bluetooth* ou *USB* (Figura 9b) (“9 Degrees of Freedom - Razor IMU”,).

3.1.6. Gavetas de material *HoneyComb*

Uma gaveta, neste caso, é constituída por três diferentes tamanhos de placas: base (1), lateral maior (2) e lateral menor (3), no total são utilizadas cinco placas: uma base, duas laterais menores e duas laterais maiores. Todos os cantos que são colados a outras placas possuem um chanfro de 45°, como pode ser observado na Figura 13b, este corte busca facilitar o processo de união perpendicular das placas, resultando 90° após finalizar a colagem. Para este projeto, dois tamanhos de gavetas serão utilizados (Tabela 6)

Tabela 6 – Tamanho das gavetas utilizadas no projeto

Gavetas	Comprimento	Largura	Altura
Gaveta Tipo 1	240 mm	240 mm	115 mm
Gaveta Tipo 2	300 mm	240 mm	115 mm

A Figura 13a é possível visualizar as placas que constituem uma gaveta do tipo 2, neste caso todas as placas possuem espessura de 5mm. A placa 1 é de 300x240mm, a 2 de 300x115mm e a 3 de 240x115mm. Para uma gaveta do tipo 1, a lateral maior e menor são iguais sendo placas de 240x115mm e a base 240x240mm. Ao lado (Figura 13b) está representado de forma amplificada as bordas com chanfro de 45° que irão auxiliar no processo de colagem e perpendicularidade entre as placas.

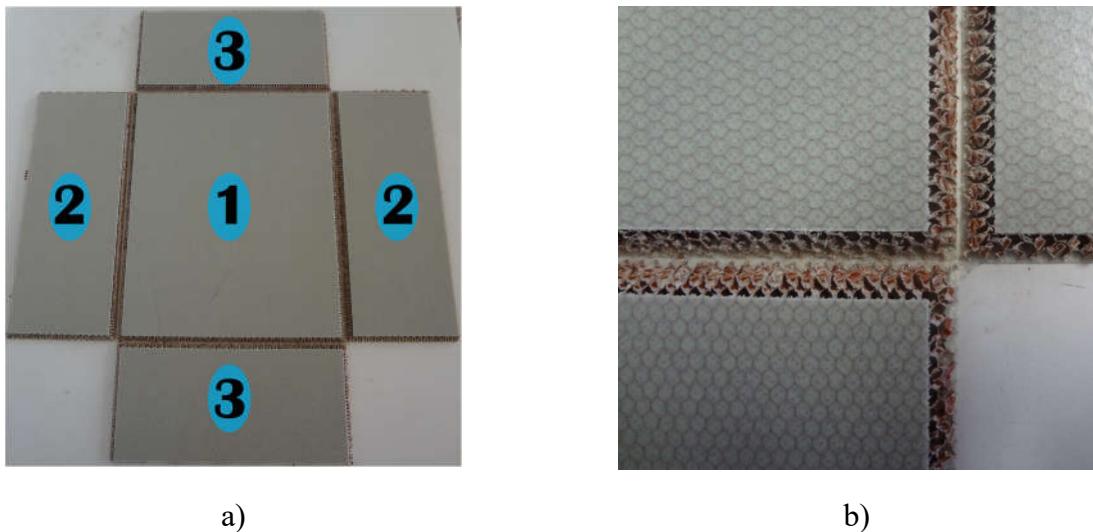


Figura 13 - Placas de *HoneyComb* unidas: a) Numeração das placas; b) Detalhe das bordas com chanfro 45°

3.1.7. Prensa de Madeira

O processo final de montagem necessita de uma ferramenta capaz de garantir o formato ao produto final, mediante a esta necessidade, uma prensa de madeira desenvolvida pelo autor, que utilizou como referência uma prensa para molduras (OFICINA DE CASA, 2014). Na Figura 14 a prensa é representada através de um desenho em plataforma CAD.

Esta é composta por três tipos de peças de madeira e um grampo de marceneiro de aperto rápido. No total são quatro tabuas de ajuste (Apêndice D), quatro bases de contato perpendicular (Apêndice E), duas junções de tabuas (Apêndice F), oito parafusos de fenda com rosca 3/16" e comprimento 1 1/2" de corpo cilíndrico e cabeça panela e oito porcas borboleta para apertos e ajustes manuais.

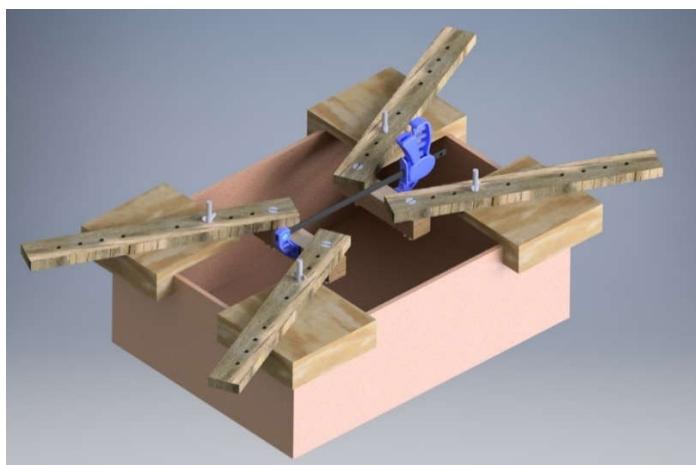


Figura 14 - Desenho renderizado 3D da utilização da Prensa de madeira em uma gaveta

3.2. Biblioteca RSI e comunicação robô-computador

A biblioteca RSI 2.3 (*Robot Sensor Interface*) é responsável pela comunicação de dispositivos externos com o sistema de controle do robô (KUKA ROBOTER GMBH, 2009). Esta biblioteca permite a comunicação do robô com o sensor de força e com dados enviados via rede Ethernet. Os dados do robô e sensor de força são compilados e comunicados em formato de texto XML (*eXtensible Markup Language*). Após o envio, o sistema do robô aguarda um pacote XML vindo do computador com os dados de comando, permitindo assim garantir a troca de informações entre computador e robô. Os pacotes XML fornecidos pela própria Kuka *Robotics* GmbH (KUKA ROBOTER GMBH, 2009) foram adaptados para o sistema proposto neste trabalho e estão representados no Apêndice A e Apêndice B.

O protocolo selecionado para efetuar a comunicação com dispositivos externos e a RSI 2.3 é o UDP (*User Datagram Protocol*), trata-se de um padrão de comunicação da camada de transporte do modelo OSI (*Open System Interconnection*) adotado pelos meios de comunicação atuais, possuem algumas singularidades favoráveis para comunicação que exigem uma velocidade alta no fluxo de dados. Por não possuir uma forma de verificação se os dados fornecidos a rede foram recebidos pelo endereço, se torna uma forma mais rápida de comunicação, entretanto é necessário inserir uma forma de verificação (*handshaking*) para evitar as perdas de pacotes.

A RSI possui em sua configuração uma forma de verificar a qualidade de comunicação, tanto no tempo de envio e resposta, como na perda de pacotes, afim de prover uma segurança e estabilidade na comunicação.

O tempo para a troca de pacotes exigida pelo sistema de tempo real do controlador é de 12 ms e o limite de pacotes perdidos podem ser configurados pelo usuário. Para testar a qualidade da comunicação, a conexão com o robô é mantida por 1 minutos, totalizando em 5 mil trocas de pacotes, o número de pacotes perdidos será analisado.

3.3. Obtenção da posição da mão do usuário

O sinal bruto enviado pelo sensor possui descontinuidades, ocasionando em degraus quando os valores estão próximos de 360° , 0° , 180° e -180° , algo comum para o espaço numérico de ângulos. Para tornar este sinal suscetível ao projeto proposto, primeiramente, foi utilizado uma matriz de rotação em Z e Y para obter os valores referentes as coordenadas cartesianas (x, y e z), o ângulo de rotação em X (*roll*) foi desprezado, de forma simplificada, foi adotado um sistema

esférico, onde o ângulo entre X e Y é o $yaw(\theta)$ e X e Z o $pitch(\varphi)$ e módulo do vetor é o comprimento (L) do *link* (braço ou antebraço do usuário).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} L * \cos(\theta) * \cos(\varphi) \\ L * \sin(\theta) * \cos(\varphi) \\ L * \sin(\varphi) \end{bmatrix} \quad (5)$$

Dois sensores foram utilizados, um para o braço e outro para o antebraço. A equação (5) é aplicada uma vez a cada sensor, os valores de X, Y e Z provenientes dos dois sensores são somados para obter a posição da mão do usuário. Um experimento é necessário para validar a posição enviada pelos sensores junto ao braço. Para realização deste, utilizamos o próprio robô KUKA como referência. Nesse caso, o robô se desloca livremente com a aplicação de uma força, assim é possível para o usuário movimentá-lo junto ao seu braço (Figura 15), após o teste uma comparação entre as posições para determinar a qualidade e confiabilidade da posição obtida via sensores inerciais.

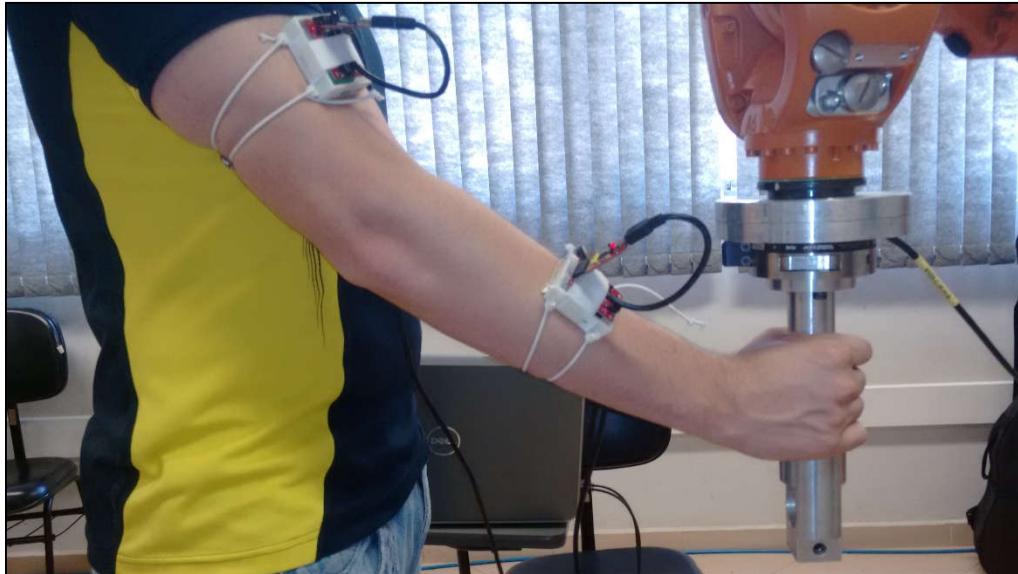


Figura 15 - Experimento para comparação da posição do braço medida pelo sensor inercial e manipulador robótico

3.4. Implementação do controle de admitância

Para este projeto foi implementado o modelo de admitância por integração citado por Lahr et al. (2016), a interação com o meio externo foi necessária ao projeto em diversas situações que serão melhores abordadas nas seções seguintes. Os ganhos M, B e K da equação (4) foram alterados de acordo com a necessidade, estes influenciam parâmetros importantes, como tempo de subida,

tempo de acomodação, coeficiente de amortecimento e frequência natural, a seleção dos parâmetros foram determinados por método iterativo e experiência do autor. A escolha dos ganhos foi realizada de forma empírica.

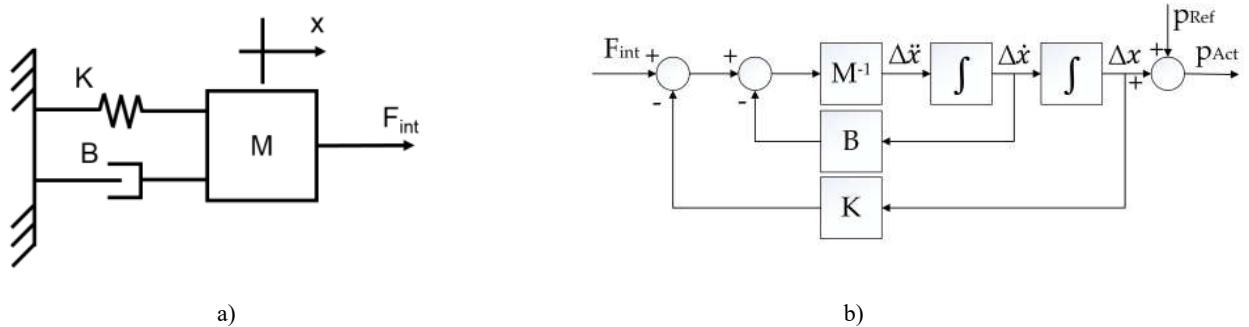


Figura 16 - Controle de admitância: a) Representação do modelo mecânico; b) Malha do controle no método integrativo

3.4.1. Controlador de admitância para a demonstração cinestésica

Todas as demonstrações realizadas nas seções seguintes utilizaram os mesmos ganhos para o processo de demonstração do usuário, através do sensor de força foi possível realizar o controle de admitância no TCP (*tool center point*) do manipulador robótico. Através de testes empíricos percebeu que é mais fácil conduzir o robô quando o coeficiente de amortecimento é unitário, sobressinal e oscilações ao final da condução dificultam o processo de posicionamento, pois é necessário esperar o tempo de acomodação. Os ganhos definidos foram 20Kg, 18KN/m e 1.2KNs/m para M , K e B respectivamente.

3.5. Compensação de gravidade

O sensor de força utilizado no projeto não é recomendado para situações com grandes alterações nos eixos rotacionais (A, B e C). A empresa aconselha que para situações como esta deve-se utilizar uma matriz de endereço correspondente a cada possível posição sistema. Mesmo utilizando o processo citado não foi possível realizar um algoritmo com grande precisão, pois uma matriz de posição é limitada a situações estáticas. O sistema de compensação ficou limitado a uma dimensão de rotação, que fica a critério da escolha do usuário. A compensação de gravidade é utilizada em conjunto com o controle de admitância, quando este é utilizado nos eixos rotacionais, como por exemplo na seção 4.10 ao realizar o ajuste fino.

3.6. Desenvolvimento do algoritmo de aprendizagem por reforço

Um ambiente discretizado é necessário para aplicar a técnica de Watkins (1989) obtendo desta forma uma Cadeia de Markov. O algoritmo desenvolvido por Elfes (1989), *occupancy grids*, é o modelo utilizado para realizar tal discretização. Uma parte da recompensa, referente à acessibilidade do estado que a ação resultou, é diretamente relacionada à variável de ocupação que, neste caso, adotou-se a binária.

Todo o processo de cálculo de trajetória é efetuado pelo algoritmo implementado em um PC que possui comunicação via rede direta com o robô. As dimensões de cada célula devem ser cuidadosamente selecionadas, sendo condizentes com as dimensões do experimento físico, há dois fatores limitantes, a velocidade de processamento do computador e o *workspace* do manipulador.

O sistema, ao receber um ponto no espaço, reconhece em qual estado o novo ponto se encontra e determina este como o estado objetivo, então o agente pode dar início ao processo de simulação de navegação entre os estados. Após um número de iterações definido, o processo de simulação é finalizado. Escolhe-se o percurso em que os estados possuem o maior valor Q , que não é a trajetória ótima, mas apenas uma candidata.

Para o primeiro cálculo do percurso, o algoritmo considera o ambiente sem a presença de qualquer corpo (obstáculo), isto é, todos os estados livres. Para um estado genérico s_i , em um espaço tridimensional, é possível tomar seis ações diferentes, representadas pelo movimento no sentido positivo e negativo nos eixos X, Y e Z. Ações diagonais não foram implementadas para agilizar a convergência do algoritmo.

Neste caso, o agente é um robô manipulador, que executa todos os comandos de posição enviados pelo algoritmo implementado. Logo, a trajetória candidata é executada sendo que, durante o percurso entre o estado atual e o próximo, o robô pode colidir com algum obstáculo no caminho. Caso a colisão ocorra, define-se o estado como inacessível.

Quando o impacto acontece, o robô retorna ao estado anterior e uma recompensa constante negativa é relacionada ao estado inacessível. Essa recompensa faz com que todas as ações que resultem nesse estado sejam recompensadas negativamente na próxima interação.

Ao retornar para o estado anterior, o robô abandona a trajetória inicial calculada, atualiza o ponto de partida. Assim, com o estado em que ocorreu a colisão com uma recompensa fortemente negativa, inicia-se um novo cálculo de trajetória com o mesmo ponto final, mas com o inicial atualizado.

O novo trajeto obtido, pela mesma lógica de seleção das ações com melhor desempenho, é executado novamente. Caso encontre mais obstáculos que gerem outros estados inacessíveis, este processo é repetido. As recompensas negativas constantes não são reinicializadas automaticamente, logo, um estado, uma vez definido como inacessível, sempre é evitado, a não ser que o usuário autorize essa reinicialização ao perceber mudanças no ambiente. A Figura 17 mostra o fluxograma que representa as etapas necessárias para o agente partir do estado inicial e atingir o estado final com a lógica desenvolvida neste projeto.

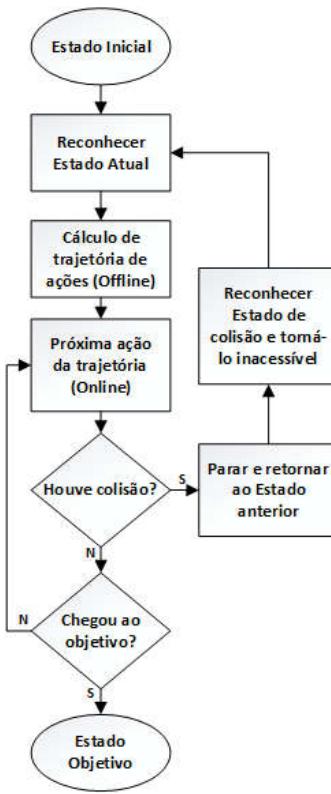


Figura 17 - Fluxograma do processo de execução de trajetória

3.7. Aplicação e reprodução do *Dynamic Movement Primitive*

O algoritmo possui um conjunto de variáveis que precisam ser definidas, antes mesmo do período de aprendizado, como pode ser observado nas equações 2, 3, 4, 5, 6 e 7. Na força de atração (f) é necessário definir o número de funções gaussianas, os centros (c_i) e covariâncias (h_i) das mesmas, de forma que se ocupe todo o espaço da fase (s).

As componentes de inercia do modelo dinâmico de massa, mola e amortecedor (3) devem respeitar apenas uma restrição que é ser um sistema criticamente amortecido, onde A_z equivale a componente de amortecimento, o produto de A_z com B_z a constante de mola e a massa é adotada

como unitária, em Hoffman (2009) recomenda-se B_z ser igual a um quarto de A_z para respeitar a restrição. Este sistema dinâmico possui como ponto de equilíbrio o objetivo (g), x como posição atual e x_0 como posição inicial. No sistema canônico o α determina a velocidade de convergência do zero ao um.

Existe uma variável que pode ser alterada após o aprendizado que é o τ , é necessária certa cautela, já que está relacionada com o tempo de execução, ou seja, um τ menor representa uma diminuição em escala da velocidade e τ maior que um representa um aumento, usualmente, utiliza-se o valor unitário para manter a mesma velocidade da demonstração

A força ponderada é responsável pela reprodução da trajetória demonstrada, ela é composta por funções de distribuição gaussianas multiplicada por pesos. Os pesos são obtidos, quando a trajetória da demonstração é processada por um método de regressão ponderada linear (LWR). O processo se resume em obter a posição, velocidade e aceleração da demonstração, integrar o sistema canônico para obter a fase com a equação (9), encontrar a força desejada (F_d) com a equação (7) e por fim os ganhos (ATKESON; MOORE; SCHAAAL, 1996).

$$f(s) = \frac{\sum_i \psi_i W_i}{\sum_i \psi_i} * s \quad (6)$$

$$\tau \ddot{x} = A_z * (B_z * (g - x) - \dot{x}) + f(s) * (g - x_0) \quad (7)$$

$$\psi_i = e^{-h_i(s - c_i)^2} \quad (8)$$

$$\tau \dot{s} = -\alpha s \quad (9)$$

$$\bar{\psi}_i = \left[\begin{pmatrix} \psi_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \psi_N \end{pmatrix} \right] \quad (10)$$

$$W_i = \frac{s^T * \bar{\psi}_i * F_d}{s^T * \bar{\psi}_i * s} \quad (11)$$

Após a fase de aprendizagem, obtenção dos pesos, para reproduzir a habilidade necessita-se apenas introduzir os pontos finais e iniciais pretendidos. A trajetória calculada similar a demonstrada pode ser obtida por um processo *offline*, quando o ponto final é constante ou *online*, podendo variar este durante o percurso. Como o sistema tende a reproduzir a trajetória desejada mesmo com distúrbios na posição, devido ao sistema dinâmico ser capaz de autocorrigir-se, é

possível adicionar ao modo *online* o controle de admitância para melhorar a interação e adaptação com o meio externo, para isso, utilizou-se a malha da Figura 18.

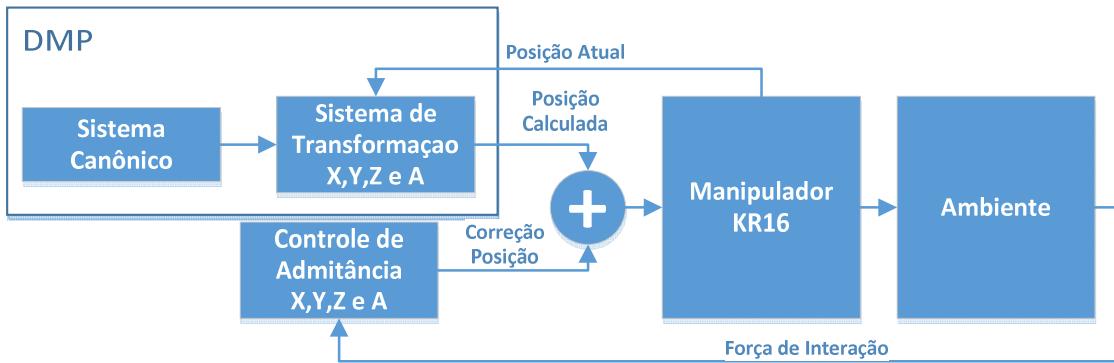


Figura 18 - Diagrama para execução de DMP em tempo real com admitância

Um experimento foi proposto para avaliar a eficiência do algoritmo para a situação proposta neste projeto. Este experimento foi dividido em duas etapas, uma de demonstração, onde o usuário mostra ao sistema como é feita a pega de uma placa de *HoneyComb* e a segunda é o processo de execução, aonde o robô pegou duas placas com posições espaciais distintas, utilizando os pesos obtidos através de apenas uma demonstração de como pegar e outra de como trazer.

3.8. Interfaces homem-máquina

O *software* desenvolvido não possui uma interface amigável e intuitiva como proposta nesse projeto, a fim de facilitar o contato, duas IHM (Interface Homem Máquina) foram desenvolvidas. Os algoritmos e códigos aqui desenvolvidos são mascarados por essas.

Cada IHM possui um escopo distinto, a primeira em plataforma Android é focada no controle remoto, onde o usuário envia comandos para o sistema identificar qual é sua próxima ação, já a segunda em Windows, o foco desta é prover um *feedback* dos parâmetros, estados e variáveis atuais, como também atuar em momentos que seja necessário emitir alertas ou interromper o ciclo de operação do robô devido alguma restrição de segurança, através dessa é possível realizar o *reset* manual.

Com o objeto de facilitar a comunicação com o operador, uma interface de comando compatível com o sistema operacional de dispositivos móveis Android 5.0 Lollipop é responsável pelo envio de comandos, sejam estes digitais, abrir ou fechar garra, ou complexos como desenvolver uma trajetória. O *software* para o desenvolvimento da interface é o Android Studio fornecido pela empresa Google.

Os dispositivos móveis atuais possuem diversas formas de contato direto com o usuário, para este aplicativo os requisitos básicos são: sistema de vibração, comunicação por rede sem fio, tela sensível ao toque. Altos índices de processamento e memória RAM não são necessários, já que não necessita de um rígido controle de tempo.

Neste projeto empregou-se um celular Sony Xperia Z2 D6503 que atende as exigências, com o Android 6.0 Marshmallow. O próprio celular serve como roteador para gerar a rede de comunicação sem fio entre o computador e celular. A interface da primeira versão do aplicativo está representada na Figura 24. O sistema de vibração será utilizado como dispositivo de *feedback* ao usuário, a frequência da vibração pode ser determinada em função de alguma variável desejada, como por exemplo a força medida pelo sensor de força.

3.9. Planejamento e desenvolvimento do *Software* de controle

Um *software* de processamento em linguagem C# (Figura 19) para o sistema operacional Windows é responsável pela comunicação, processamento e interpretação dos dados. O computador efetua todo o processamento necessário para o fluxo de dados. A plataforma com capacidade de comunicação UDP e serial é essencial para a comunicação de todos os dispositivos disponíveis e necessários, por este motivo o C# foi selecionado, já que possui bibliotecas para suprir as necessidades.

Por se tratar de um sistema composto por vários dispositivos, os quais são assíncronos entre si, se faz necessário a elaboração de linhas de encadeamento (*threads*), uma técnica para elaborar rotinas de programação que sejam executadas paralelamente.

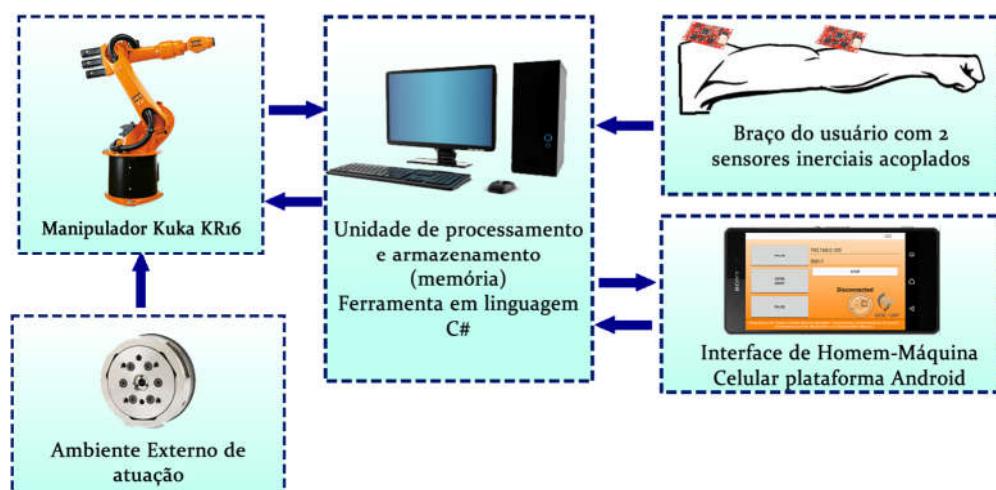


Figura 19 - Diagrama do fluxo de dados e informações entre o robô, sensor de força, usuário e unidade de processamento e memória

O *software* utiliza um método de aprendizagem por reforço para locomoção em ambientes desconhecidos mapeados em *occupancy grids* e outro por demonstração, DMP, para realizar tarefas especificadas pelo usuário. O início do processo é a parte de aprendizado e treinamento, o usuário demonstra as trajetórias bases que serão utilizadas posteriormente pelo DMP, desde a forma que o robô deve pegar uma placa *HoneyComb* e como pegar as ferramentas que são necessárias para a montagem.

Mesmo uma gaveta sendo composta por cinco placas de material *HoneyComb*, uma base e quatro laterais, duas demonstrações são suficientes, uma para a pega e outra para a entrega, já que a base da trajetória é a mesma, altera-se apenas o ponto inicial e final. Há também como demonstrar ao sistema se há um obstáculo e o próprio algoritmo elabora uma trajetória desviando do mesmo.

Para todo e qualquer deslocamento que não tenha sido ensinado, o algoritmo de movimentação com aprendizado por reforço é ativado, já que este é preparado para possíveis colisões, que acarretaram na atualização dos valores de ocupação do ambiente e o aprendizado por reforço evitará tal célula em situações futuras. O usuário tem controle das ações do robô, as quais são enviadas pela interface do celular com plataforma Android.

Para os objetivos propostos, é preciso desenvolver um sistema seguro e confiável, características quais são essenciais para robôs colaborativos e interativos. Um controle de velocidade no movimento nos seis eixos é necessário também, para evitar possíveis problemas, uma vez que aplicações com contato tendem a ser mais lentas. Além disso, o próprio manipulador Kuka KR16 possui dispositivos de segurança, como detector de torque excessivo nos motores, botão de emergência e limitador de altas velocidade, todos estes desativam os motores e freiam o sistema de forma rápida.

3.10. Aplicação do *software* na montagem de gavetas

Um ambiente de produção de gavetas reproduzido no laboratório. As placas de *HoneyComb* e materiais necessários dispostos no envelope de trabalho do robô. A fim de prover uma solução, a forma de produção foi estudada para testar e validar as técnicas empregadas no projeto.

Inicialmente, há a possibilidade de ensinar ou carregar ao sistema a demonstração da pega de objetos, neste caso os objetos são as placas de *HoneyComb* e a fita crepe, o algoritmo DMP irá generalizar a demonstração, para que o sistema seja capaz de reproduzir o aprendizado. Após esta fase, deve ser informado ao sistema as posições de pega de cada objeto, as quais são utilizadas

como objetivo (g) nas trajetórias geradas com o DMP, para informar é possível movimentar o manipulador até a posição ou carregar posições previamente salvas.

Com a fase de demonstração e aprendizagem concluídas, o processo de montagem é iniciado, o usuário utiliza a interface Android versão 2 para informar os comandos que ele deseja ao sistema, os comandos correspondem a: pegar fita crepe; pegar placa 2; pegar placa 3; pegar cola; e ajuste perpendicular fino.

As pegas de placas consistem na reprodução de duas DMP, uma para a pega e outra para trazer, na trajetória de pega, a posição inicial é a posição atual do manipulador, já a final consiste no ponto de pega referente a placa, para a trajetória de trazer a placa, a inicial é a posição do ponto de pega e a final corresponde ao ponto que foi ensinado ao fim de demonstração.

Ao pegar uma ferramenta o sistema utiliza o mesmo DMP de pega, com ponto inicial sendo a posição atual do manipulador e final o ponto de pega da ferramenta e ao retornar utiliza a posição da mão do usuário informado ao sistema através dos sensores iniciais acoplados ao braço do mesmo. Com a posição objetivo, o sistema utiliza o algoritmo *Q-learning* com *occupancy grids* para chegar em uma posição próxima ao braço do usuário. No momento de realizar a pega da fita, em especial, foi utilizado o controle de impedância para atingir uma força desejada e, assim, aumentar as chances de realizar uma pega bem-sucedida.

O ajuste fino compete ao sistema garantir a perpendicularidade da peça, para isso utiliza-se a compensação de gravidade em algum eixo rotativo, podendo ser A, B ou C, e inicia o processo de condução do manipulador até a placa desejada, então o usuário solicita para que o robô feche a garra na mesma, então altera-se os ganhos do controle de admitância para uma impedância mais rígida e começa a correção no ângulo selecionado anteriormente na compensação de gravidade.

A sequência de comandos é de responsabilidade do usuário, então o sistema após o aprendizado e demonstração, aguarda o envio de comandos. A sequência proposta é pegar a fita crepe, então pegar as placas 2 e 3, enquanto o robô executa a pega o usuário já pode utilizar a fitas para uni-las conforme a configuração na Figura 21, com todas as placas unidas, o usuário então utilizaria a cola, processo o qual não foi implementado ao sistema e, por fim, solicita o ajuste fino levando o manipulador até uma lateral maior e quando a placa atingir a posição perpendicular colocar a prensa de madeira concluindo a montagem de uma gaveta.

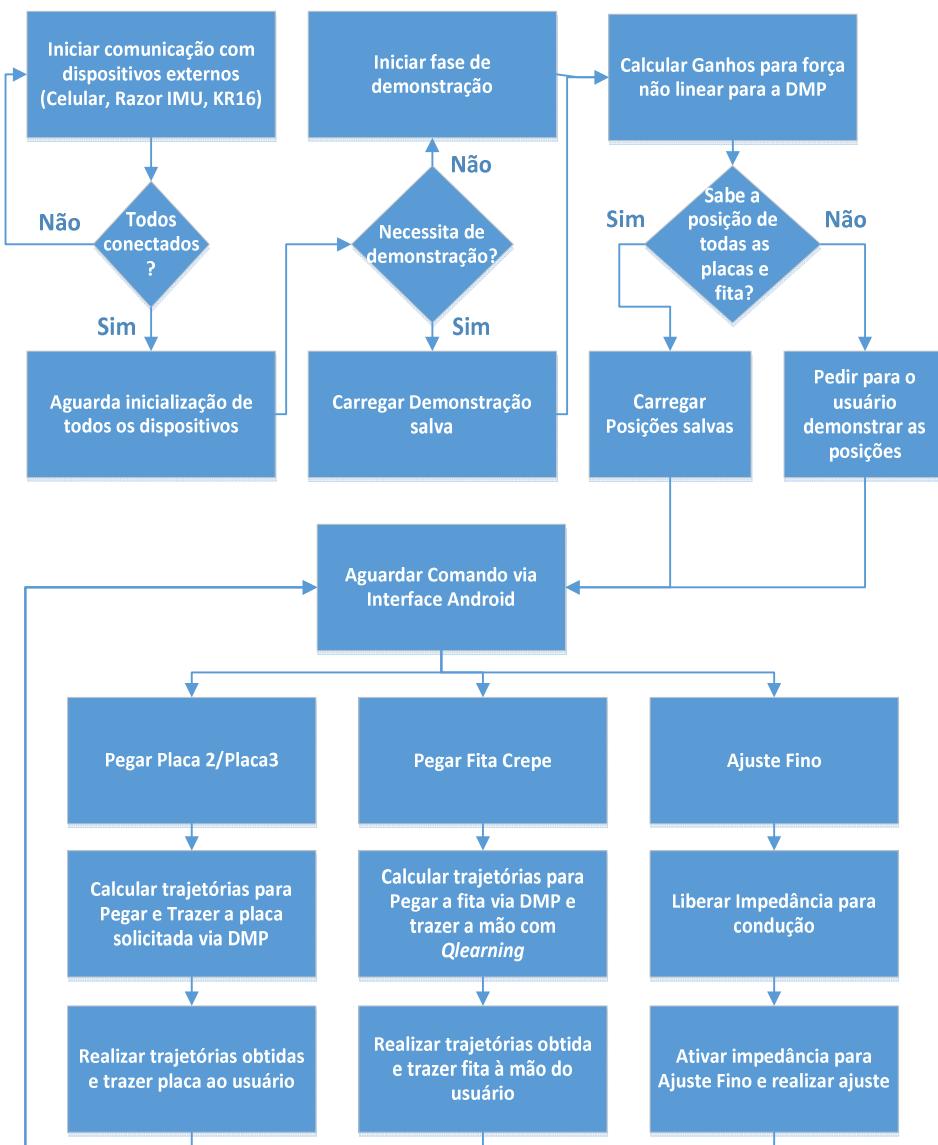


Figura 20 - Diagrama de operação do Software de controle



Figura 21 - Placas HoneyComb unidas com fita crepe para auxiliar a montagem

3.11. Ambiente Simulado

O *software* V-REP (*virtual robot experimentation platform*), desenvolvido pela Coppelia Robotics, foi utilizado como ambiente para desenvolver uma simulação. Devido ao *software* não possuir o modelo do manipulador robótico KR16, foi necessário desenvolvê-lo. Para isso utilizou-se um modelo CAD do KR16 (SOARES, 2012). Os parâmetros do manipulador foram obtidos utilizando o manual do mesmo (KUKA ROBOTER GMBH, 2004).

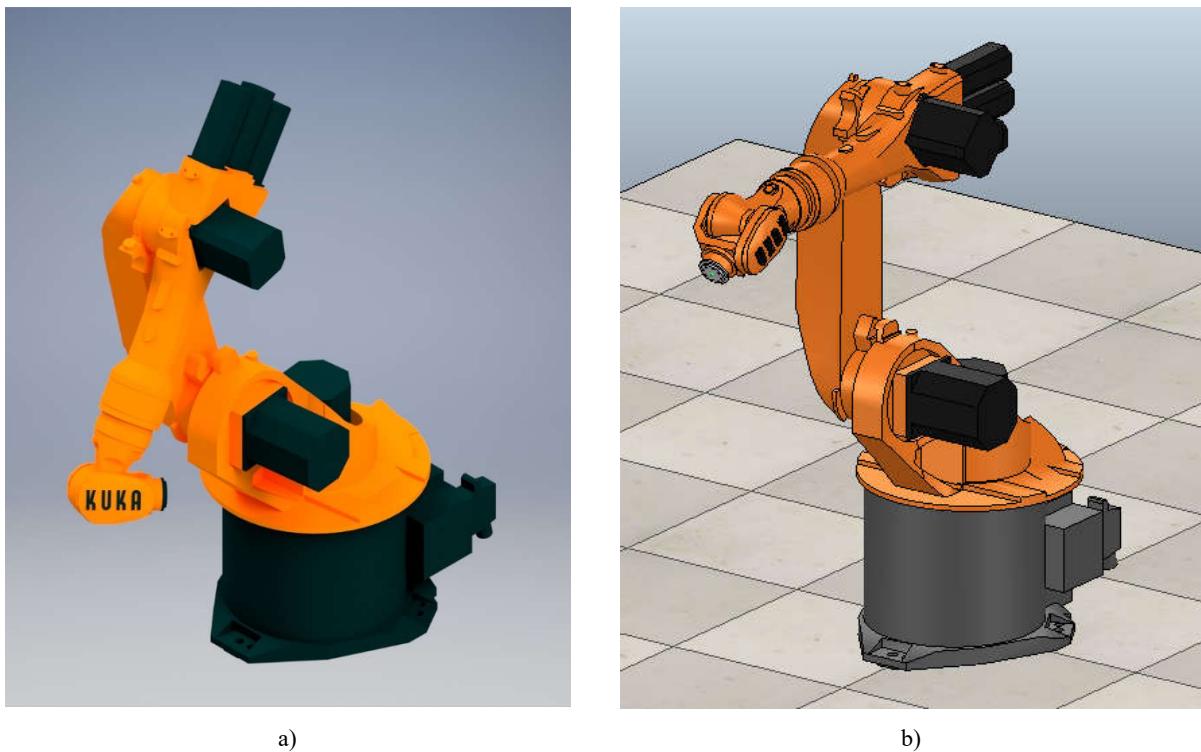


Figura 22 – Modelos mecânicos utilizados na simulação: a) Modelo em plataforma CAD Inventor; b) Modelo em plataforma VREP

O V-REP também possui uma plataforma de programação que é integrada ao ambiente visual, no âmbito de programação e comunicação foi necessário elaborar uma comunicação UDP, a fim de que o *software* de controle desenvolvido neste trabalho seja capaz de se comunicar e comandar a simulação. O protocolo e o pacote de dados de comunicação utilizado possuem o mesmo formato que o manipulador real utiliza, com isso é possível tornar a simulação próxima da realidade. Com esses quesitos, a simulação foi utilizada, como validação inicial das técnicas de segurança com referência na ISO 15066.

3.12. Adequação do software a especificação técnica ISO 15066

Para cada uma das quatro tarefa atribuída na Figura 20 foi necessário classificá-la e adequá-la a um grupo, conforme os citados no último parágrafo da 2.5.2. Após a classificação, iniciou-se a fase de apreciação de riscos conforme estabelece a norma ISO 12100, além disso, também precisou-se adaptar as tarefas aos limites fornecidos pela ISO/TS 15066.

Sabendo-se que não é possível validar a implementação, devido as normas solicitarem equipamentos específicos para as funções de segurança, os quais não estão dispostos neste manipulador. Esse estudo buscou compreender e encontrar formas de implementar e adequar as tarefas aqui desenvolvidas. Duas abordagens foram utilizadas, testes simulados com um modelo do KR16 e o *software* VREP e testes reais com o manipulador KR16. A utilização do sensor de força foi crucial para tal desenvolvimento, já que o mesmo supri de certa forma as necessidades de uma tarefa colaborativa, entretanto o manipulador em questão, não é colaborativo e não possui sistemas de medição necessários para torná-lo em um, como sensores de força e torque em todas as juntas, capacidade de controlar torque e entre outros.

3.12.1. Estados de Segurança

Na rotina de operação de uma máquina ou equipamento que possa oferecer a possibilidade de riscos a humanos, equipamentos ou a si mesmo, é necessário que seja elaborado diferentes formas de funcionamento. Essas formas, no contexto de segurança, são conhecidos como estados de segurança: desligado; em operação; alerta; e estado de segurança.

Desligado é quando o equipamento não está alimentado ou ainda não foi acionado, já em operação refere-se ao momento onde a máquina está funcionando de modo regular e todos os recursos estão disponíveis, usualmente, pode-se considerar que é o estado em que o equipamento possui maior performance. O estado de alerta possui como objetivo, informar o operador que o equipamento está próximo a uma situação perigosa ou adversa, se aproximando a um estado de segurança, no qual o equipamento pode paralisar sua operação ou retornar para uma posição/estado seguramente estável em que não haja riscos ao ambiente, operador, pessoas e ativos.

Conforme o tópico 12.40 da NR12 (MTE, 2011), toda situação que seja adversa a segurança do operador ou colaborador, ativo e ambiente, o equipamento deve possuir uma forma de paralisar sua movimentação e atingir um estado de segurança. Neste projeto assumi-se como estado de

segurança o robô com movimentação paralizada, mantém-se a parte elétrica ativa para sustentar o manipulador em uma posição estável.

Além de possuir um estado seguro, também, o equipamento necessita de um *reset* manual do operador para que esse possa retornar ao estado de operação, neste caso ao modo de trabalho contínuo. Para viabilizar a aplicação de um estado seguro, esse foi implementado no *software* desenvolvido e o usuário interage com o equipamento utilizando a interface citada no próximo parágrafo.

3.12.2. Modos de Segurança Colaborativa

Conforme a ISO/TS 15066, atividades colaborativas podem ser divididas em quatro modos, conforme a Figura 23. Cada modo possui suas singularidades também na parte de segurança, um ponto em comum é que todos necessitam de uma categoria 3, PLd, a qual não foi possível aplicar neste projeto devido a restrições de verbas para aquisição de uma estrutura de segurança independente, como relés, contatores e CLP (Controlador Lógico Programável) que atendam a essa categoria.

Mediante a esta limitação, foi desenvolvido no *software* algoritmos que sejam compatíveis com as necessidades solicitadas na norma ISO/TS 15066. Uma adaptação foi realizada para que esses algoritmos sejam compatíveis com os equipamentos utilizados. No total, são dois algoritmos, Força e Potência limitada e Condução Manual.

Os dois modos citados no parágrafo anterior são acionados conforme a tarefa que o manipulador venha a desenvolver. Todos possuem a capacidade de fazer com que o sistema passe do estado de operação para alerta, como também, de alerta ou operação para segurança. O tópico 12.40 da NR12 também foi respeitado, forçando o *reset* manual como a única forma de fazer com que o sistema retorne ao estado de operação após entrar em estado de segurança.

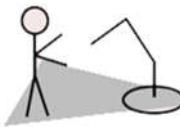
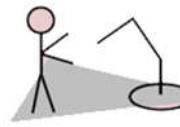
Modos de Colaboração	Propósito	Seguranças requeridas
Condução manual	 <p>Conduzir o manipulador robótico a partir do comando manual inserido pelo operador</p>	<p>Botão de acionamento e controle de velocidade com dispositivos na categoria 3, PLd</p> <p>Fácil acesso ao botão de emergência</p>
Monitoramento de velocidade e distância	 <p>Manipulador robótico apresenta velocidade reduzida quando a pessoa está próxima</p>	<p>Sensor de posição do operador e controle de velocidade com dispositivos na categoria 3, PLd.</p> <p>Sem <i>reset</i> automático, tem que ser <i>reset</i> manual.</p>
Parada monitorizada de segurança	 <p>Manipulador robótico pausa sua movimentação quando a pessoa se aproxima</p>	<p>Sensor de posição do operador e controle de velocidade com dispositivos na categoria 3, PLd.</p> <p>Sem <i>reset</i> automático, tem que ser <i>reset</i> manual.</p>
Força e Potência limitada	 <p>Manipulador robótico paralisa sua operação quando ocorre um contato crítico</p>	<p>Respeitar as forças limites e críticas conforme anexo A</p> <p>Sensor de força/contato e controle de velocidade com dispositivos na categoria 3, PLd.</p>

Figura 23 - Classificações de tarefas colaborativas (adaptado de UMBREIT, 2016)

4. Resultados

4.1. Comunicação robô e computador

O controlador do Manipulador Kuka KR16 com a biblioteca RSI possui um protocolo de comunicação UDP e TCP. Neste projeto adotou o UDP por recomendação do próprio fabricante, os dados são enviados em formato de uma estrutura de dados XML de comunicação.

No sentido do manipulador robótico ao computador (Apêndice A), o pacote contém informações do estado atual e dados de conexão. Os termos *delay* e IPOC são formas de conferir a reciprocidade e segurança da conexão, o primeiro realiza uma contagem de pacotes perdidos e o segundo é um número para identificação de cada pacote, por se tratar de uma comunicação *real-time* é obrigatório realizar comunicação com um período máximo de 12ms. Sobre a situação atual, dados como posição, tanto nas coordenadas X, Y e Z (RIst e RSol) como os ângulos de cada junta (EIst), força (FTC) e saídas digitais (DigOut).

Como resposta o computador deve enviar um pacote XML (Apêndice B) com comandos de correção na posição (Rkorr e EKorr), entradas digitais (DigIn) e IPOC. Correções e informações na corrente de cada motor estão representadas no pacote, mas não foram implementadas neste manipulador, logo só é possível controlar a posição do sistema.

A fim de validar a comunicação elaborada entre computador e o manipulador, a conexão foi estabelecida por um período de tempo necessário para realizar a troca de 5000 pacotes, ou seja, com duração de 1 minuto. Nesta análise, constatou que apenas dois pacotes foram perdidos durante a comunicação, resultando em 0,04% do total, demonstrando que a comunicação é eficaz e suficiente para a prática proposta, entretanto deve-se atentar aos processos do computador já que o mesmo não é recomendado para realizar tarefas *real-time*.

4.2. IHM Android

O *software* Android Studio utiliza a linguagem orientada a objeto Java de programação, esse auxilia no desenvolvimento do aplicativo. Ao final da compilação do projeto, gera-se um arquivo instalável no sistema operacional com extensão apk.

O design do *layout*, também, foi realizado no *software*, diversos componentes são disponibilizados e a configuração de posição no espaço da tela é bem simples, sendo possível arrastá-los ou criar tabelas ou grades de *layout*. Na área de codificação é necessário identificar os

componentes pelo nome de identificação (ID) e o próprio sistema já fornece quais os possíveis métodos e variáveis.

A integração do *layout* com o código é simples, entretanto deve-se lembrar que o dispositivo móvel não é um sistema projetado para operações *real-time*, assim é necessário simplificar e evitar grandes esforços computacionais. A comunicação com o computador foi elaborada utilizando o protocolo UDP, pelos mesmos fatores já citados anteriormente, o usuário necessita apenas inserir qual o endereço de IP do computador e a porta desejada para a comunicação.

Os pacotes de dados são enviados em ambos sentidos entre o computador e celular, na primeira versão o celular é capaz de enviar três entradas digitais, o valor de orientação do dispositivo por meio dos ângulos *yaw*, *pitch* e *roll* e a numeração do pacote. O computador envia uma frequência para atuar no motor de vibração e a numeração do pacote. Todos os dados ainda são enviados em um formato *string* e os valores são separados por pelo espaçamento *tab*.

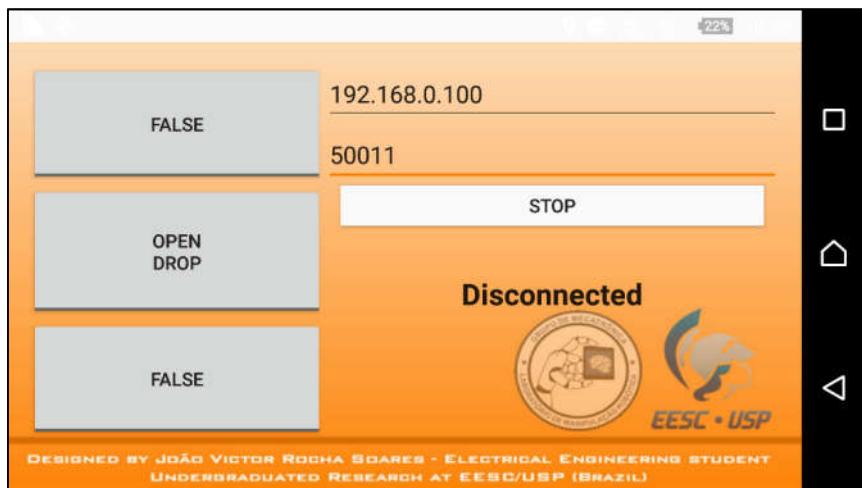


Figura 24 - Interface Homem-Máquina na plataforma Android versão 1

Uma segunda versão foi desenvolvida para aprimorar a interação do usuário com o sistema, já que na primeira não é possível selecionar o tipo de material e a única forma de atuar era através de três sinais digitais, previamente programados na plataforma C#. Nesta nova versão além da adição de sinais, foi incorporado o sistema de *TextToSpeech*, que é capaz de ler *string* e reproduzir em forma sonora, utilizando frases programadas para informar a fase do processo ou se o sistema está executando alguma tarefa, facilitando ainda mais a comunicação com o usuário, por exemplo, na fase de aprendizagem o sistema diz: “*Can you teach me how to pick the Board?*”.

Além da melhoria na comunicação, agora o usuário é capaz de selecionar qual ferramenta ou tipo de placa de *HoneyComb* ele deseja. No total é possível solicitar dois tipos de placas, a cola

ou a fita crepe. Ao selecionar um material, o campo Comando do pacote xml (Apêndice G) repassa o comando solicitado ao *software* em C#. Com a utilização percebeu a necessidade de inserir dois avisos de erro, um quando o usuário esquece de informar o endereço de IP e a porta e outro quando o sistema ainda não finalizou o comando solicitado (Figura 26b), também foi implementado um *feedback* de estado de comunicação através da cor do botão de início da conexão (Figura 25).

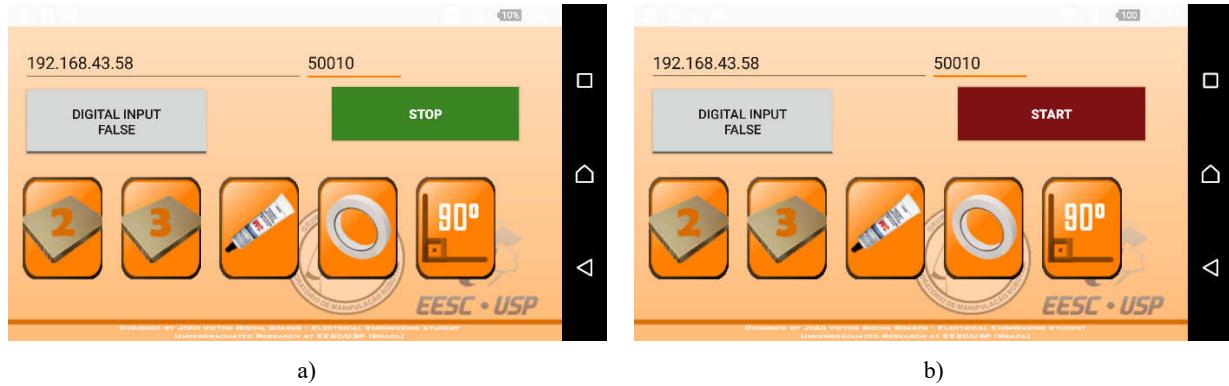


Figura 25 - Interface Homem-Máquina na plataforma Android versão 2: a) Conectado; b) Desconectado

No pacote de comunicação no sentido do computador ao celular (Apêndice G) contém os dados de força e torque nos seis eixos na seção “ForTor”, a ativação da vibração como *feedback* haptico em “*Haptic*”, número do pacote em “Pacote”, uma seção para inserção de texto para o sistema pronunciar “Speech” e um booleano para avisar se o comando já foi executado pelo sistema “Rdy”. No outro sentido o pacote possui a informação da orientação do celular em ângulos de Euler na seção “YPR”, o comando solicitado pelo usuário em “Com”, número do pacote em “Pacote”, valor atual do botão digital em “DigIn”.

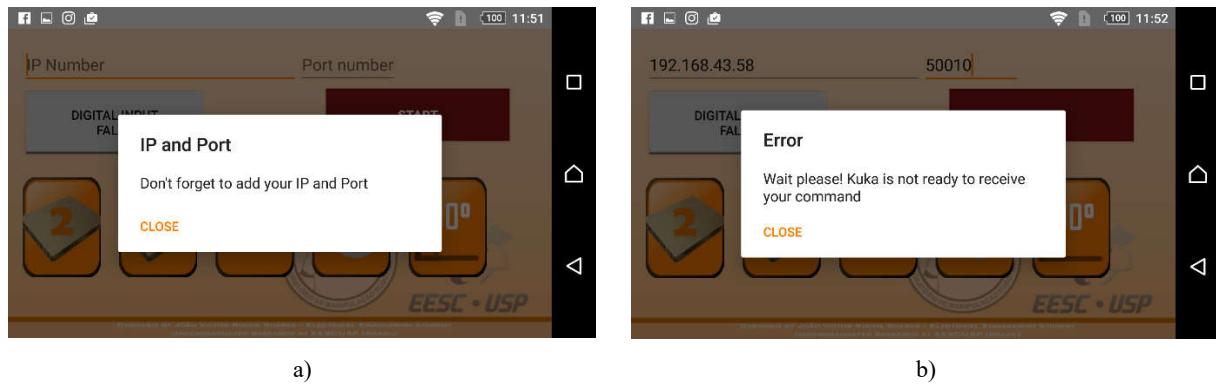


Figura 26 - Avisos de erro: a) Sem endereço e porta de comunicação; b) Aguardar o sistema finalizar o comando

4.3. Experimento *pick and place* utilizando algoritmo *occupancy grids* com *Q-learning*

Para a tarefa de *pick and place*, é utilizado um cenário conforme apresentado na Figura 27. Neste caso, deseja-se pegar a peça no ponto A descartá-la em B e retornar a A. Dá-se o nome de “tarefa” a um circuito completo de ida e volta entre A e B. Foram fixados dois obstáculos, apresentados nas regiões C (obstáculo dois) e D (obstáculo um), para que ocorresse a obstrução de um caminho possível.

Esse cenário permite o deslocamento apenas no espaço XYZ. Conforme descrito, o ambiente é mapeado por grades de ocupação, conforme representado pela Figura 27a, por questões gráficas está representado por um plano para facilitar visualização, esta é apenas uma ideia do tipo de discretização, uma vez que a quantidade de estados e dimensões deste podem variar.

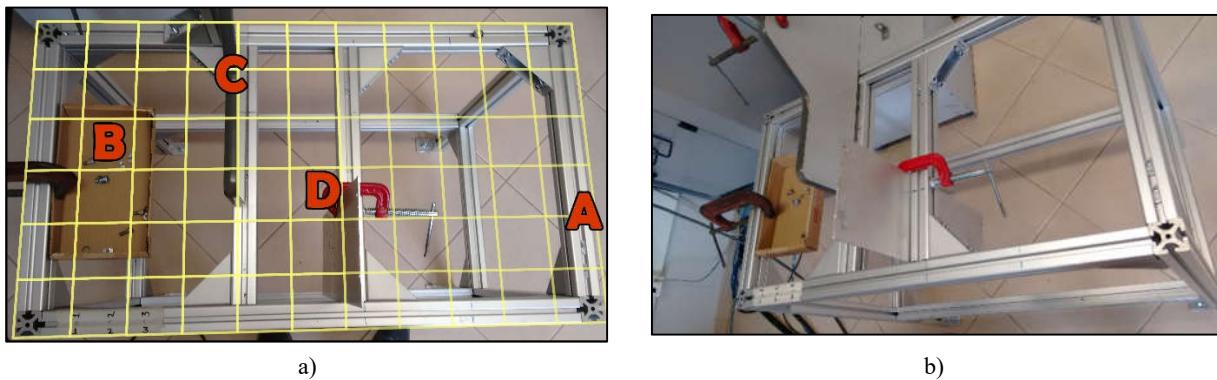


Figura 27 - Ambiente para realização do *pick-and-place*: a) Ponto inicial e final, obstáculos e representação plana de divisão do ambiente em *occupancy grids*; b) Fotografia com foco nos obstáculos inseridos no ambiente

Para o manipulador desenvolver o experimento proposto, foi selecionada um espaço de 1000x500x500mm mapeada em um quadrado onde o eixo X possui 20 estados (células), o Y com 10 e por fim o Z com 10 resultando em 2000 estados com um tamanho de 50x50x50mm. A escolha da granularidade da discretização é essencial para o desempenho do algoritmo, poucos estados desfavorecem a exploração do agente, entretanto agiliza a convergência. O critério de seleção foi escolher de forma que o menor obstáculo ocupe por volta de quatro estados.

Uma fase anterior para seleção dos fatores de aprendizagem e de desconto da equação (1) foi necessária, ambos influenciam diretamente no resultado, positiva e negativamente. Esse teste foi elaborado de forma empírica, testando-se vários valores, e foram selecionados os que

apresentaram melhor desempenho perante o número de iterações necessárias. Desta maneira, tanto α quanto γ possuem valor igual a 0,8.

A tarefa de *pick and place* será repetida 10 vezes e este experimento busca demonstrar a melhora do desempenho com as experiências obtidas das tarefas anteriores. Pode ser observado na Figura 28 o tempo que o manipulador levou para completar tal ação, iniciando com valores próximos a 43,376s para ida e 37,073s para o retorno e ambos caindo para, respectivamente, 22,025s e 22,889s, resultando em uma melhoria de 49% e 38%. Esse tempo leva em conta também a quantidade de iterações que o PC executou.

Este tempo também é relativo à velocidade do robô que, como prova de conceito, limitou-se a 58,3 mm/s, uma vez que altas velocidades não são recomendadas para experimentos com contato.

A Figura 28, também, busca representar o aumento do desempenho por meio da experiência, é possível constatar que o processo de cálculo apresentou um bom desempenho. A convergência para a parte de ida e volta pode ser observada, a partir da terceira tentativa os valores tendem a manter um padrão, demonstrando que o sistema já conhece o ambiente.

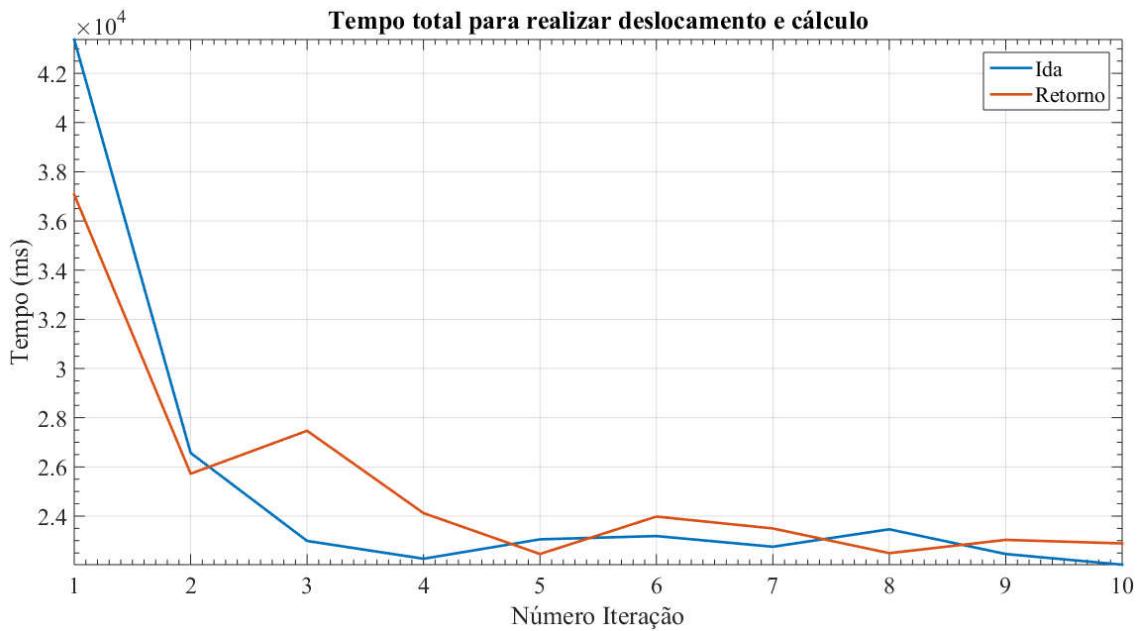


Figura 28 - Tempo de execução do percurso da tarefa em função da tarefa atual

Esta divisão foi suficiente para a tarefa executada. Nas Figura 29 e Figura 30 representa amostras de deslocamentos do manipulador no espaço. No gráfico (Figura 29) referente ao processo

de ida, observa-se que o robô colidiu com os obstáculos oito vezes na primeira tentativa e 1 na segunda. A partir da terceira, este não apresentou mais colisões, outra observação, é que o manipulador foi capaz de completar a tarefa em todas as tentativas.

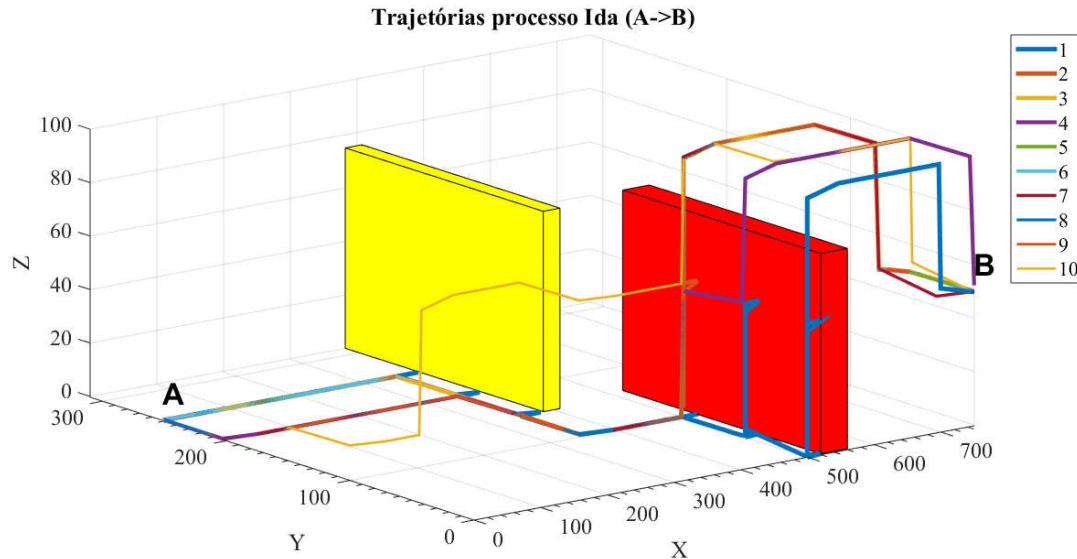


Figura 29 - Deslocamento no espaço entre os pontos A e B

Já no processo de volta (Figura 30), o sistema apresentou seis colisões na primeira tentativa e uma na terceira, um número menor quando comparado ao processo de ida, pois o agente conhece uma parte do ambiente, então alguns possíveis choques foram evitados.

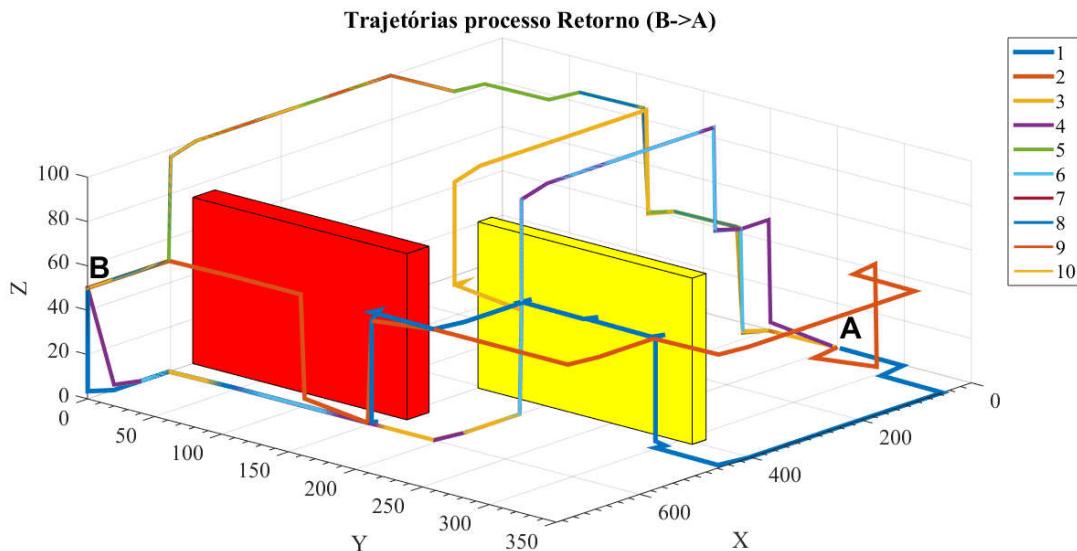


Figura 30 - Deslocamento no espaço entre os pontos B e A

Com o intuito de efetuar uma comparação com o experimento anterior, o espaço de trabalho para realizar a mesma tarefa foi mapeado em uma quantidade diferente de estados, os quais apresentam um volume maior no primeiro caso e menor no segundo. No primeiro caso são 40x20x20 estados com 25mm cada lado do cubo e no segundo 10x5x5 estados com 100mm.

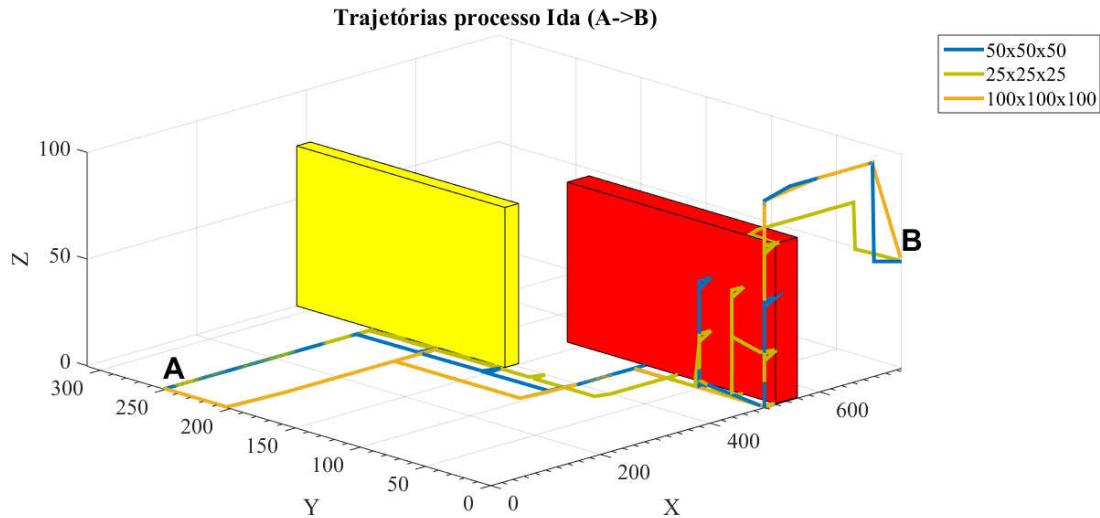


Figura 31 - Comparação entre primeira tentativa de ida variando a quantidade de estados

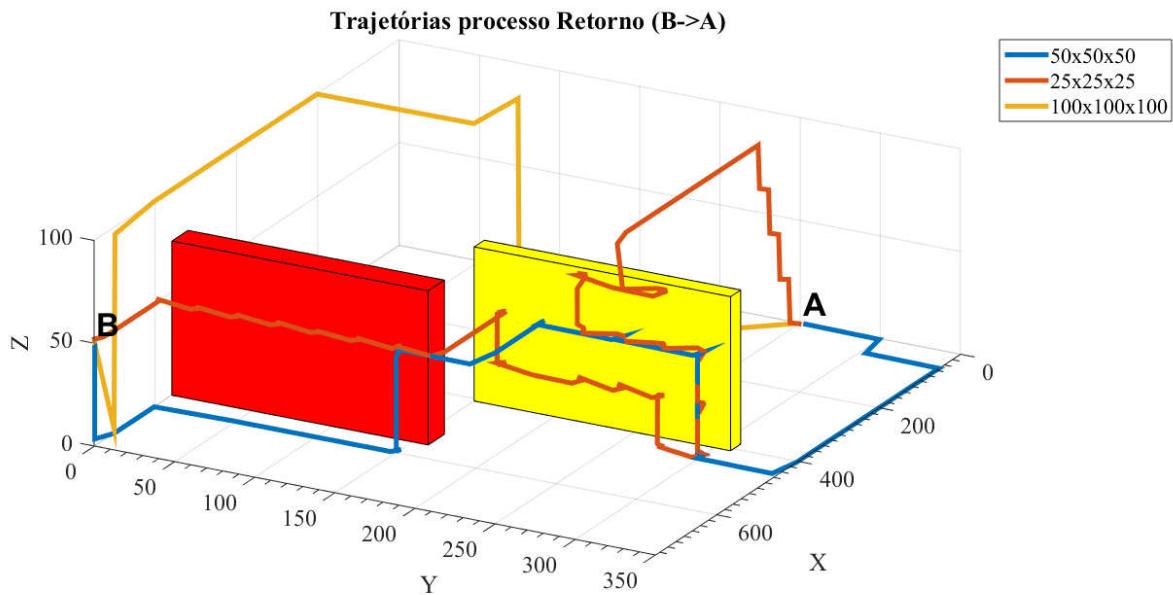


Figura 32 - Comparação entre primeira tentativa de retorno variando a quantidade de estados

Ao diminuir o número de estados, primeiro caso, o processo apresentou uma velocidade maior e um número menor de iterações. Com isso, o sistema perde a resolução, e pode ocorrer de um estado ser inacessível, mas dentro desse existe a solução ou um caminho menor, mas o sistema não consegue acessar. Como pode ser observado na Figura 34, na décima tentativa do percurso de retorno com 20x10x10 estados (50x50x50mm), o manipulador percorreu 114,76 cm, com 40x20x20 (25x25x25mm) estados, 109,83 cm e com 10x5x5 estados (100x100x100mm), 124,26 cm. Neste caso, a distância percorrida com 1600 estados foi 4,29 % menor comparada a 2000, já no caso de 250 foi 8,28 % maior comparada a 2000.

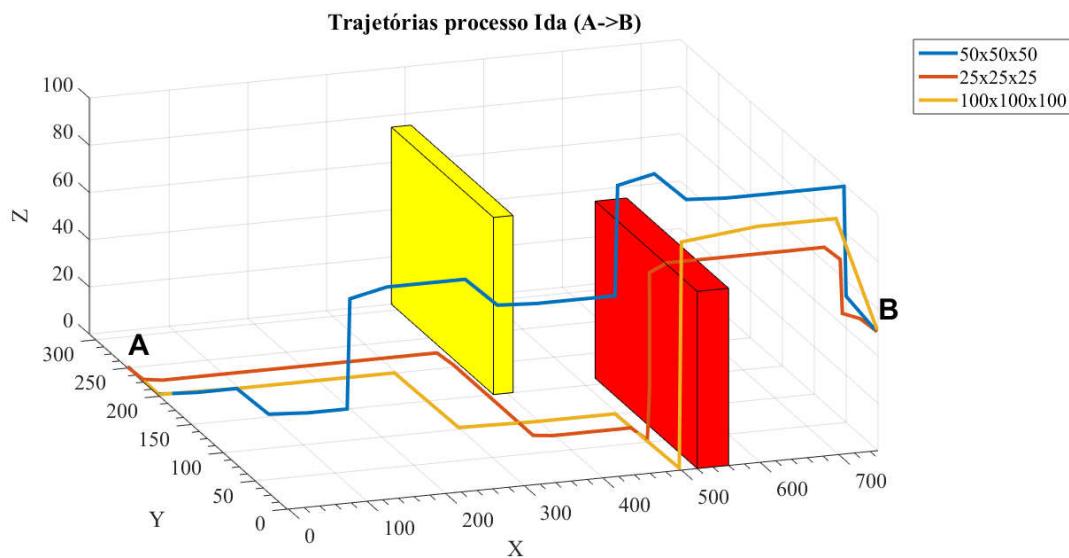


Figura 33 - Comparação entre décima tentativa de ida variando a quantidade de estados

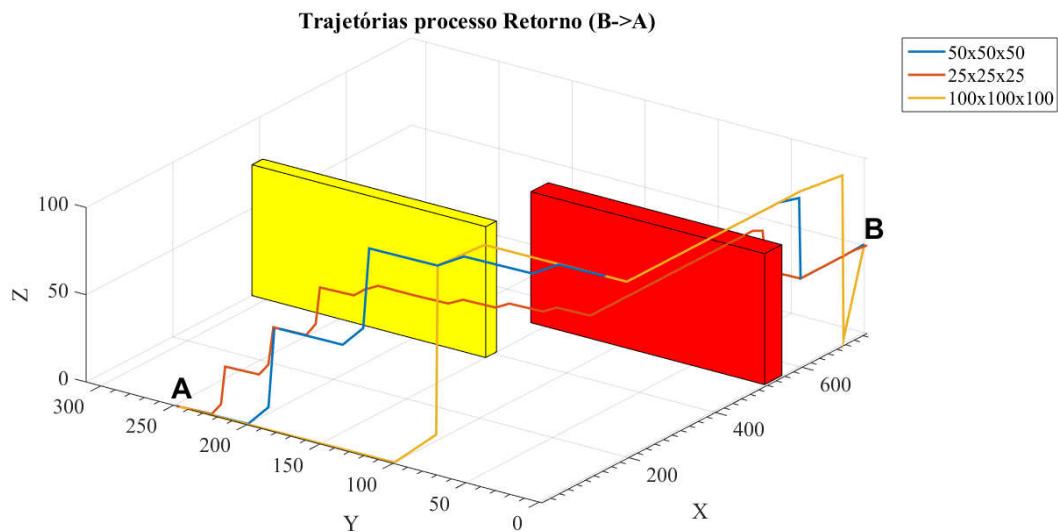


Figura 34 - Comparação entre décima tentativa de retorno variando a quantidade de estados

Para o caso de aumentar o número de estados, o sistema apresentou um tempo maior de execução devido ao aumento do número iterações necessárias para o cálculo de trajetória. A cada colisão, um novo processo de cálculo é iniciado e, neste caso, o número de colisões são maiores, visto que o obstáculo ocupa um número maior de estados. A Figura 35 demonstra o aumento do período necessário a cada caso.

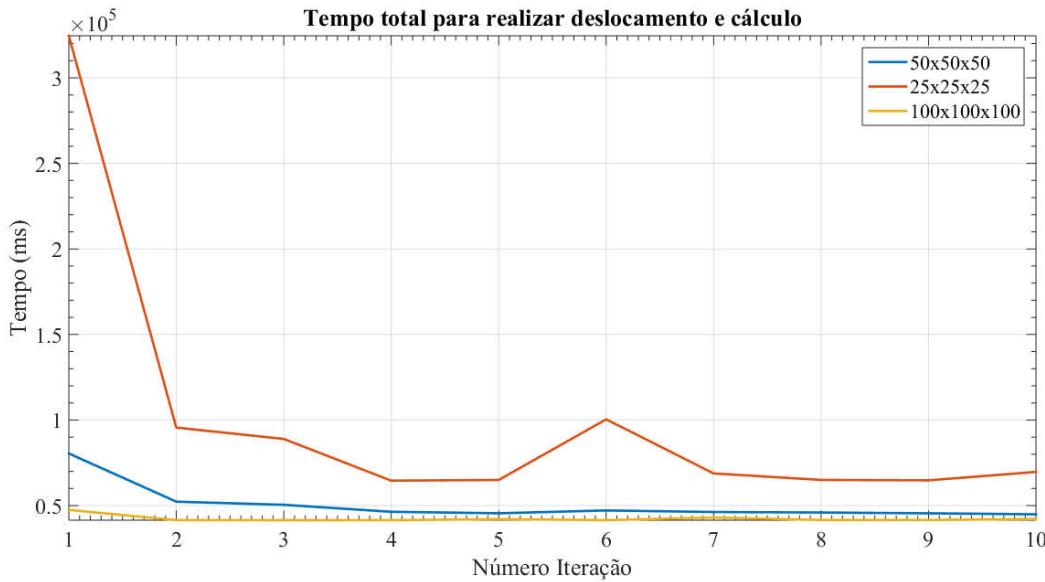


Figura 35 - Comparaçāo do número de iterações necessárias para a realização de uma tarefa completa com diversas quantidades de estados

A exigência de um esforço computacional maior passa a diminuir com o aprendizado, então é necessária uma análise para escolher de forma correta a divisão de estados que seja suficiente e eficiente.

4.4. Experimento de aquisição e comparação de dados dos sensores iniciais

Para mensurar os erros foi proposto um experimento utilizando o manipulador robótico KUKA e o sensor de força com um controle de impedância (LAHR et al., 2016), B igual a 1200Nm/s, K a 18000N/m e M de 20Kg, com isso o usuário pode move-lo livremente. Os dados de posição da posição da mão do usuário com a ferramenta do robô foram comparados (Figura 36) e o máximo desvio em X foi de 97.7505mm, em Y de 76.5932mm e em Z de 127.8656mm.

Os sensores iniciais nos testes demonstraram certa imprecisão na medição, devido ao deslocamento do ombro e coluna do usuário serem necessários e involuntários para certas posições, então não é possível ter uma referência estática. Mediante aos erros observados, o dado proveniente

do sensor não pode ser considerado confiável, entretanto é um dado útil e pode ser levado em consideração para definir uma área próxima à mão do usuário, mas o usuário deve evitar o deslocamento do ombro e saber que a posição do braço possui como referência a posição inicial do usuário. Logo os dados dos sensores serão utilizados para que o robô possa trazer ferramentas até o usuário. Exemplos poderiam incluir o transporte de cola e prendedores, agilizando o processo de montagem, evitando o deslocamento do operador.

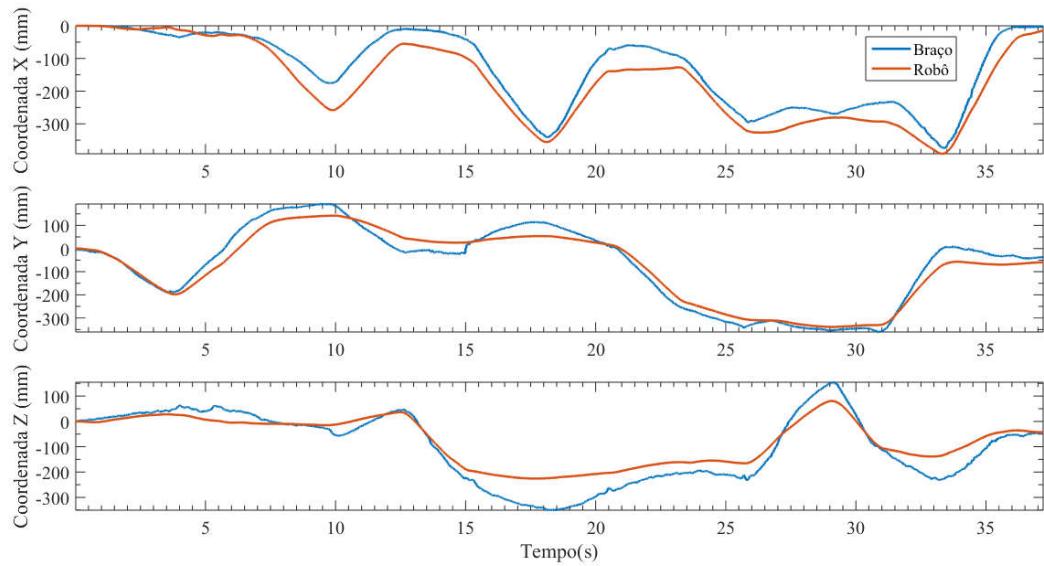


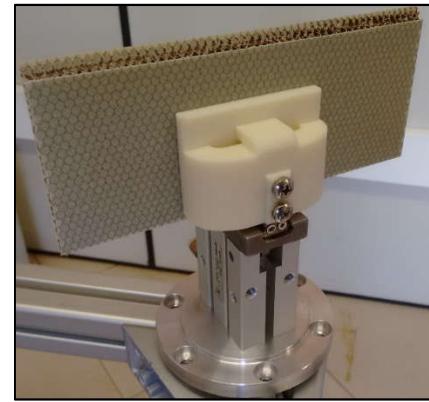
Figura 36 - Teste verificação de erro posição final da IMU

4.5. Projeto da ferramenta de pega

Com o apoio do *software* CAD Autodesk Inventor, foi possível desenhar e dimensionar a garra e gerar o arquivo necessário para realizar a impressão 3D. O intuito deste formato (Apêndice C) foi melhorar a transferência de forças da placa *HoneyComb* para o sensor de força acoplado ao *End-Effecter* do manipulador, como também aumentar a superfície de contato no momento da pega.



a)

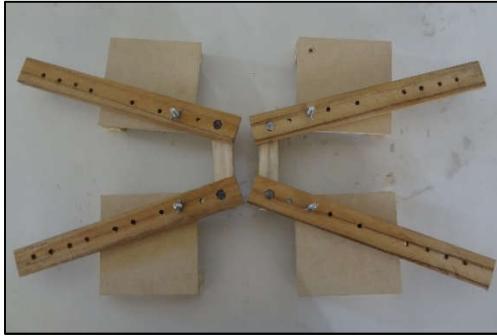


b)

Figura 37 - Ferramenta de pega: a) acoplada no *End-Effector*; b) foto demonstrativa

4.6. Elaboração da Prensa de Madeira

Com base nos desenhos elaborados (Figura 14) foi desenvolvida a prensa, utilizando as mesmas medidas projetadas. Nas quatro tabusas de ajuste foram utilizadas madeiras de eucalipto, as duas junções de tábuas e bordas da base de contato madeira pinus e para as quatro bases de contato madeira MDF.



a)



b)

Figura 38 - Prensa de Madeira: a) Visão superior; b) Visão Inferior

Um grampo de aperto rápido da marca Lufkin, com 6" de trilho e 220Kg de pressão máxima, foi utilizado para fixar e pressionar os modelos da Figura 38 junto as gavetas. A Figura 39 retrata o último processo de montagem, onde todas as placas já foram unidas e é colocado a prensa para fixar na posição desejada e garantir um bom processo de colagem das placas.



Figura 39 - Prensa de madeira pressionada por grampo de aperto rápido em uma gaveta de *HoneyComb*

4.7. Comparação dentre as formas de execução do algoritmo DMP

Três formas de reprodução da trajetória demonstrada foram implementadas no software: o offline; tempo real (online); e tempo real com controle de admitância (RTAd). Como pode ser analisado na Tabela 7, entre estas formas de execução, há pontos positivos e negativos, uma vez que para as trajetórias geradas em tempo real não é possível prever os pontos necessários para reprodução do aprendizado, diferente da offline, mas o objetivo não pode ser atualizado durante o processo de execução.

Tabela 7 - Comparação dentre formas de execução de DMP

Característica	Offline	Online	RTAd
Visualizar trajetória antes de executá-la	Sim	Não	Não
Mudar o objetivo durante o processo de execução	Não	Sim	Sim
Capacidade de interagir e se adaptar ao ambiente externo	Não	Não	Sim
Garantia que ao fim da execução do trajeto encontra o objetivo	Sim	Sim	Não
Desviar de obstáculos	Sim	Sim	Sim

Para demonstrar as diferenças entre estas formas de execução, foi realizado uma demonstração e a mesma foi reproduzida pelos três modos, mantendo os mesmos ganhos e pontos

finals e iniciais. Na configuração da DMP utilizou-se A_z igual a 25 e seguiu-se a configuração proposta por Hoffman (2009), então B_z igual 6,25 , a força não linear é composta por 300 funções gaussianas nesse caso e foram utilizados três sistemas de transformação, correspondentes ao eixo X, Y e Z. Os ganhos do controlador de admitância para o processo de execução foi de 20Kg, 18KN/m e 1.2KNs/m para M, K e B respectivamente. Abaixo é possível observar a força desejada e a calculada via regressão LWR para o caso citado, apenas 20 das 300 gaussianas ponderadas estão representadas.

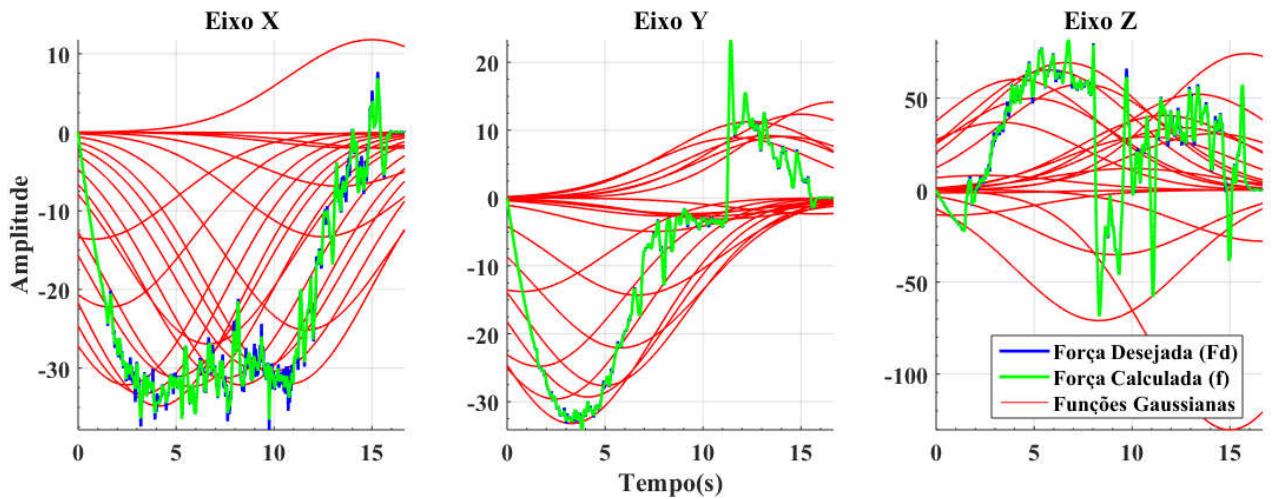
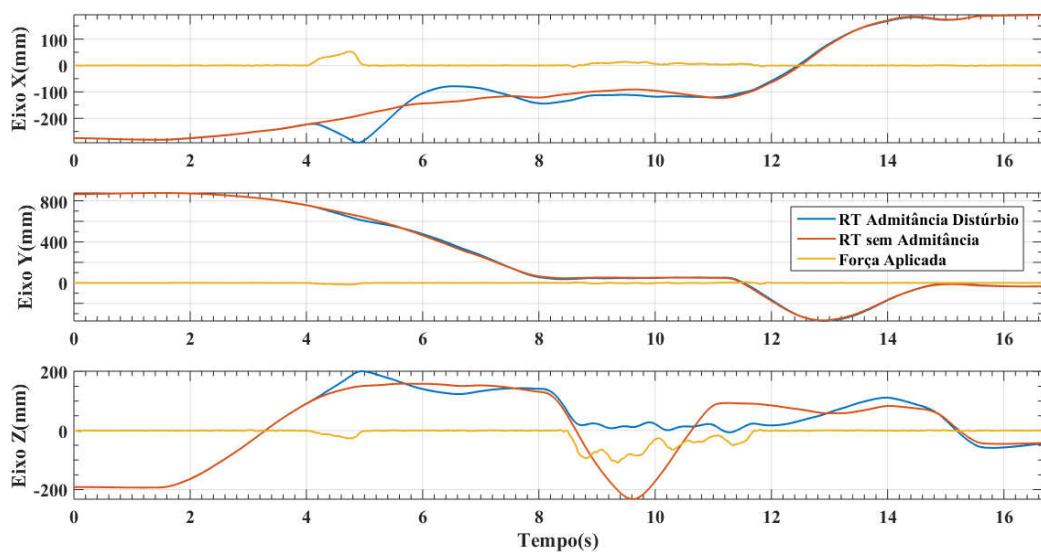
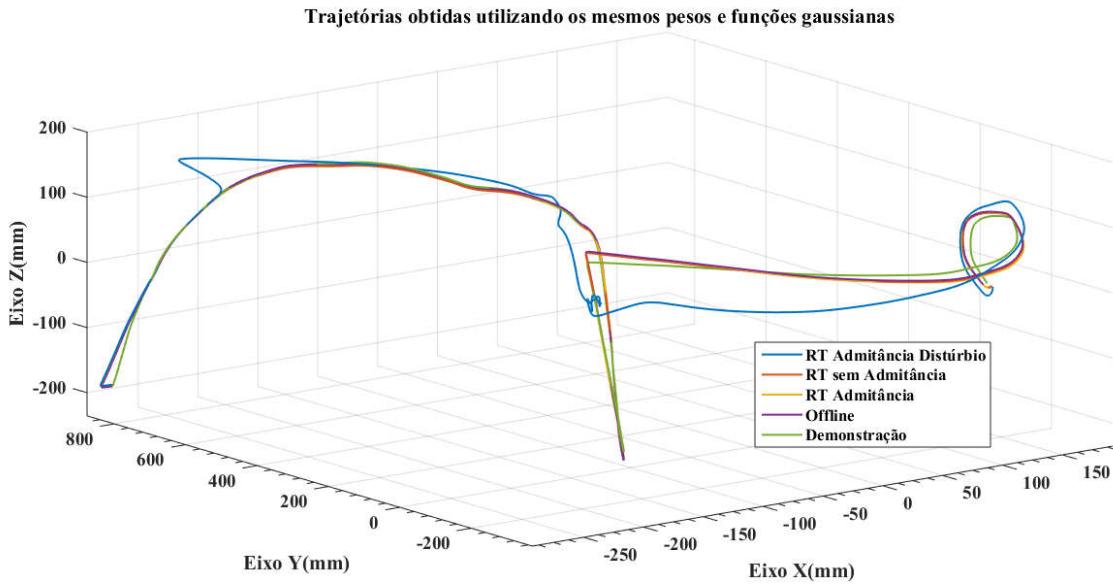


Figura 40 - Calculo da força não linear da DMP

Na Figura 41 pode-se observar que todas reproduziram a demonstração de forma similar, a tempo real com admitância sem distúrbio (RT Admitância), tempo real (RT sem Admitância) e Offline ficaram bem próximas, assim o decidir dentre destas, deve-se utilizar a Tabela 7 para determinar qual a necessidade, percebe-se, também, que a trajetória em tempo real com admitância com distúrbio (RT Admitância distúrbio), distúrbio o qual foi realizado pelo próprio usuário do sistema com a mão, impedindo o robô de atingir posições indesejadas, é possível analisar a distorção gerada na trajetória na Figura 42, onde está representada a força aplicada no robô.



4.8. Observações para alteração de pontos finais e iniciais da DMP

Pela equação (7) é possível perceber que a força não linear multiplicada pela diferença entre o objetivo e a posição inicial é um termo de proporcionalidade, isto indica que de certo modo a força inserida ao sistema pode alterar o seu sentido, caso a subtração resulte em um valor com sinal diferente da demonstração ou ainda amplificar ruídos.

As características citadas foram observadas durante as práticas que utilizaram o DMP, embora seja um algoritmo que se mostrou muito eficaz, estas podem prejudicar no momento da

execução, principalmente, em modos online onde o usuário não pode visualizar a trajetória antes de sua execução. Um experimento foi realizado para demonstrar essas características, utilizou-se as mesmas configurações citadas na seção 4.7.

A troca do sentido em casos específicos, como nos eixos X, Y, A, B e C, pode ser benéfica, entretanto no eixo Z pode representar um certo perigo, por exemplo no caso que o manipulador realize a pega de um objeto sobre uma superfície, é esperado que o mesmo venha de cima para baixo, entretanto pode acontecer de inverter a força não linear em Z, ocasionado em um movimento inverso, com a chance do robô colidir com a superfície onde o objeto está posicionado.

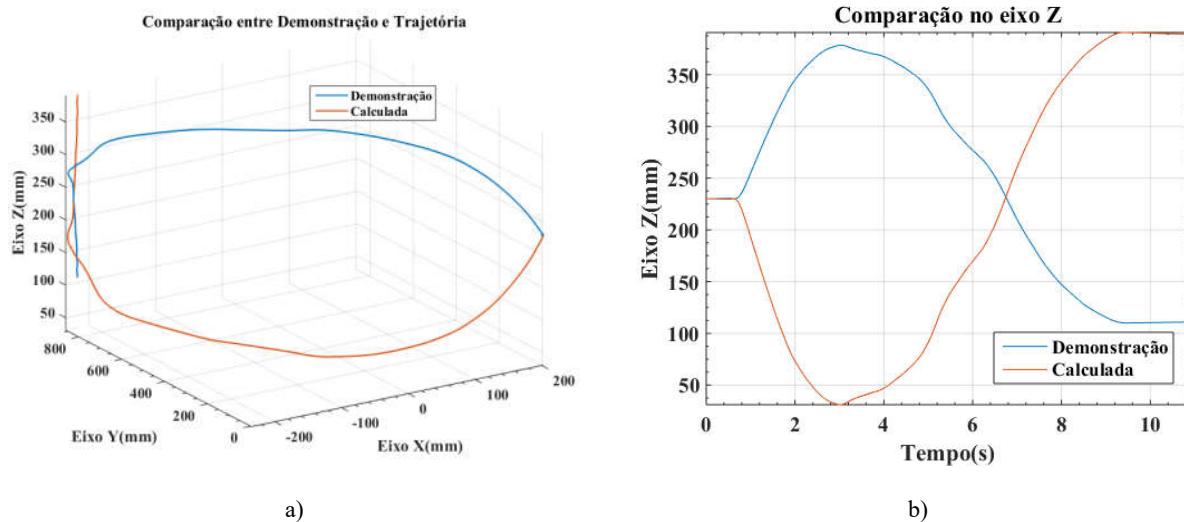


Figura 43 - Inversão no sentido da trajetória no eixo Z: a) Trajetória 3D; b) Eixo Z

Amplificar ruídos, oscilações e a trajetória em si é também outro fator, isto ocorre quando a diferença entre o ponto inicial e final inseridos para cálculo da trajetória são maiores do que a diferença entre os mesmos obtidos durante a demonstração, recomenda-se que a demonstração sempre seja a maior trajetória possível, porque a redução implica em reduzir esta característica, mas caso não for possível, deve-se atentar que existe um sistema de transformação para cada dimensão. Na Figura 44 aumentou a posição em Z do objetivo em 100mm, percebe que esta alteração ocasionou em um aumento proporcional de toda a trajetória no eixo Z com um pico de diferença máxima de 427mm.

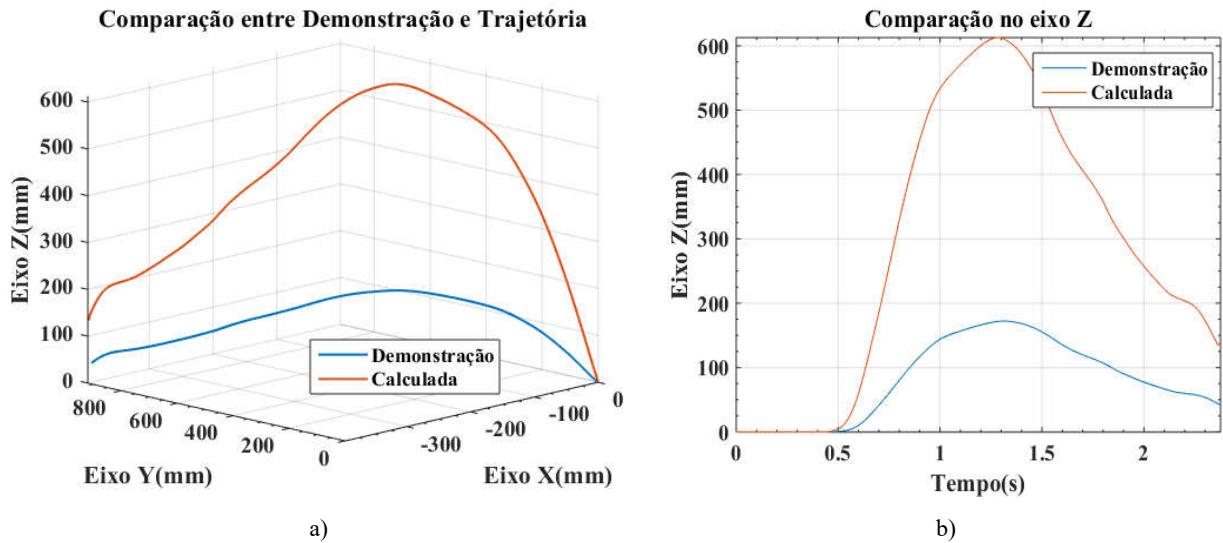


Figura 44 - Proporcionalidade em toda a trajetória com a alteração de pontos finais: a) Trajetória 3D; b) Eixo Z

4.9. Experimento de pega de duas placas *HoneyComb* utilizando DMP *offline*

De acordo com a literatura, um sistema de transformação deve ser utilizado por coordenada no método DMP. Neste caso optou-se por apenas três dimensões, não interferindo nas orientações, caso as orientações forem adicionadas seriam necessários seis sistemas de transformação ao invés de três (Figura 45).

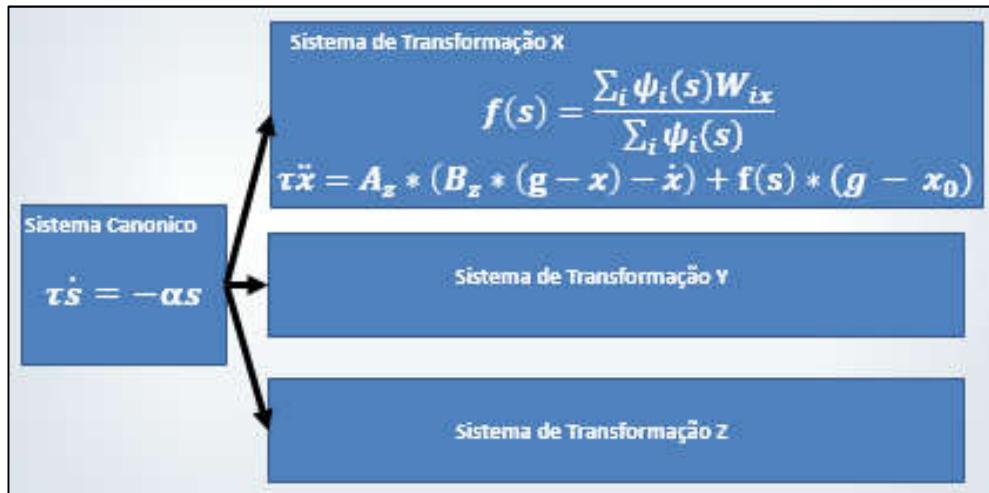


Figura 45 - Sistema de três coordenadas utilizando DMP

Inicialmente deve-se escolher os valores referentes as constantes do sistema, para o sistema dinâmico determinou que A_z é igual a 14 resultando em um B_z de 3.5 e α de 7, 200 funções gaussianas com centros distribuídos de forma equidistante ocupando todo o espaço da fase, foi

utilizado as equações 8 e 9 em função do tempo total necessário para realizar a demonstração em milisegundos, na Figura 46 estão representadas 40 funções das 200 para facilitar a visualização e a ponderação ainda não foi aplicada. O valor das variáveis devem ser mantidos durante todo o processo, apenas o τ pode ser alterado, entretanto neste experimento foi mantido o valor unitário.

$$c_i = \frac{i * TempoTotal}{(N - 1)} \quad (12)$$

$$h_i = \frac{0.5 * TempoTotal}{N} \quad (13)$$

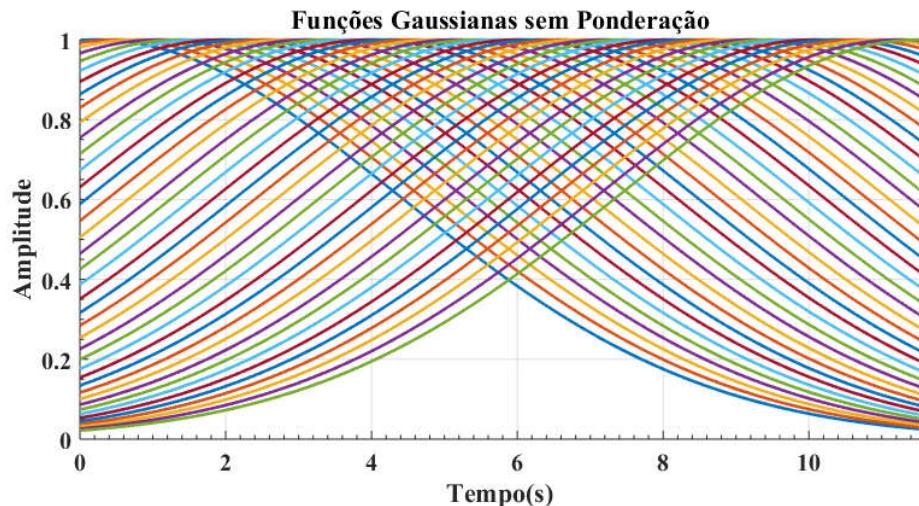


Figura 46 - Funções Gaussianas antes do processo de aprendizagem

O processo de aprendizagem consiste em duas trajetórias: o movimento de buscar a placa e o de trazer a placa até o usuário, logo foram duas demonstrações, uma para cada movimento. Após este processo, os ganhos referentes a cada demonstração foi obtido com o uso do LWR. Para cada trajetória obteve-se 600 pesos, 200 cada coordenada. Na Figura 47 apenas 40 funções gaussianas estão representadas e já foi aplicado o peso respectivo, a linha azul é a força desejada para que o sistema seja capaz de repetir a trajetória da demonstração e a verde é a calculada utilizando a equação (6).

Com os pesos definidos, inicia-se a fase de execução onde é necessário passar a posição para realizar a pega das placas, neste experimento assumiu que o sistema já conhecia os pontos de pega. A posição da pega será utilizada como ponto final no movimento de buscar e ponto inicial no de trazer. Entaõ, as trajetórias para trazer três placas até o operário foram calculadas e obtidas, antes do robô realiza-las, as mesmas foram apresentadas ao usuário utilizando a biblioteca que

integra o MATLAB ao C# e com isso foi possível usufruir da plataforma de gráficos (Figura 48) e somente após a autorização do usuário elas foram executadas.

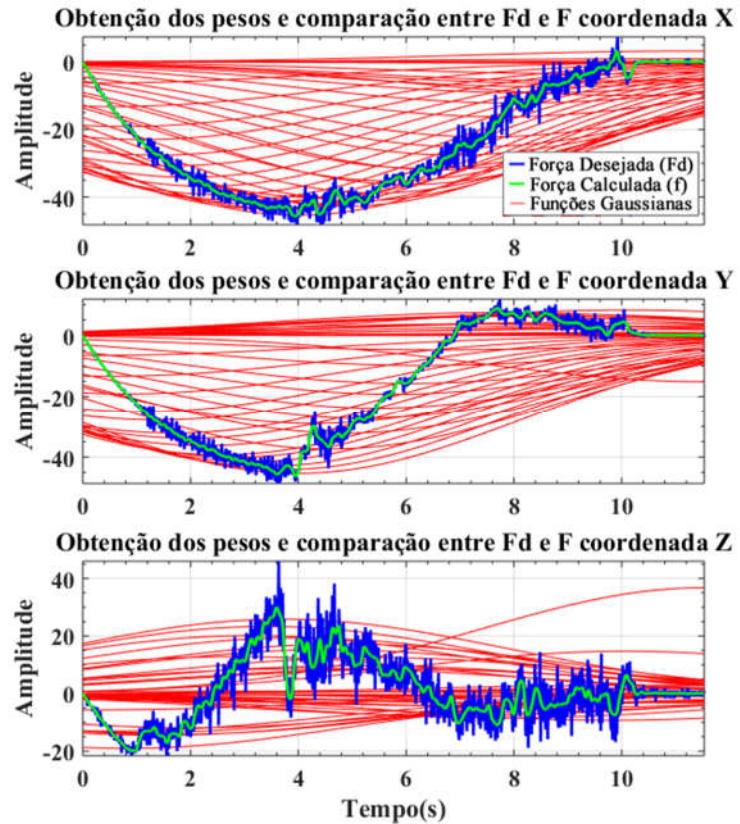


Figura 47 - Gaussianas Ponderadas e Forças Atrativas Desejada e Calculada

A Figura 49 busca retratar o experimento com uma sequência de imagens. As imagens de um a cinco, correspondem ao processo de aprendizagem, onde o usuário faz uma demonstração cinestésica ensinando o sistema a como realizar a pega de uma placa, já nas de seis a doze, o robô executa a pega das duas gavetas utilizando o algoritmo proposto.

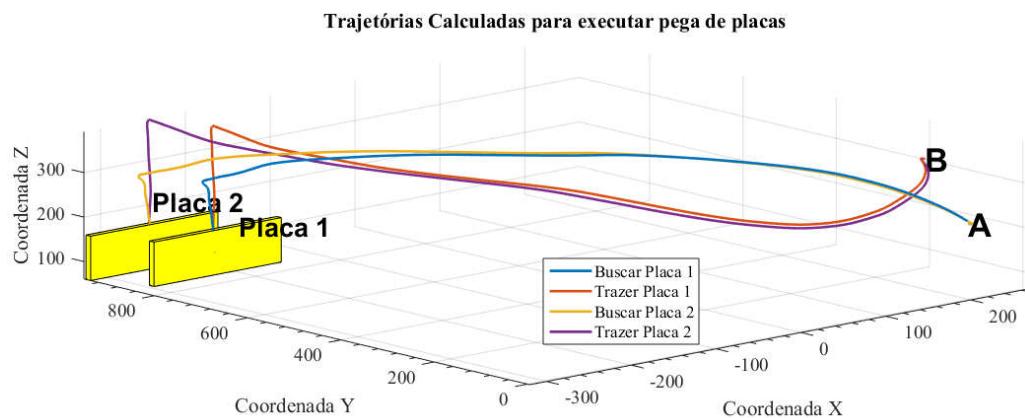


Figura 48 - Trajetória calculada para pega das duas placas

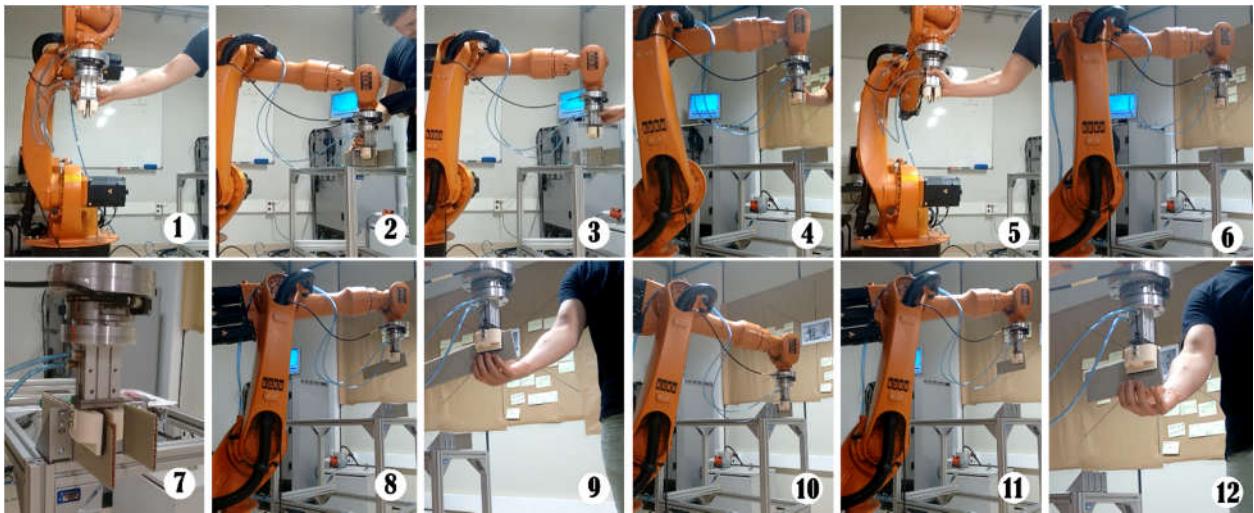


Figura 49 - Sequência de fotos do experimento com aprendizagem por demonstração utilizando o método DMP

4.10. IHM Windows

Com base em um dos objetivos principais desse projeto, que é desenvolver um sistema intuitivo, foi necessário elaborar uma interface homem máquina que pudesse mascarar e auxiliar no processo geral. A interface é dividida em três abas: *configuration*; *Information*; e *Emergency Control*.

A primeira aba, *Configuration*, o usuário configura os parâmetros do programa. Na primeira caixa, *Limit*, informa-se a porta que será utilizada para realizar a comunicação com o Manipulador KR16; os limites de velocidade, tanto linear (x, y e z) e angular; e por fim os limites máximos do espaço de trabalho para cada dimensão. Ainda nessa aba, configura-se também os valores de gatilho

A comunicação com os dispositivos externos, necessitam ser configurados, para o celular necessita da porta de comunicação configurada no próprio aplicativo (Figura 25), os sensores inerciais contém dois parâmetros, *Link Length*, refere-se ao tamanho do antebraço ou braço do usuário, conforme a distribuição das IMU (Figura 15) e por se tratar de uma conexão serial precisa identificar quais portas o computador dedicou para a comunicação, por este motivo há o botão *Check Serial Ports*, após o clique o sistema informa no painel de texto se há alguma porta dedicada e o nome da mesma. Para iniciar a próxima aba, *Information*, primeiro inicia-se a comunicação com os dispositivos ao pressionar *Start Devices*, o sistema escreverá no painel quando o processo de inicialização for concluído e então o botão *Start Program* é liberado e ao clicar nesse, inicia-se o manipulador robótico.

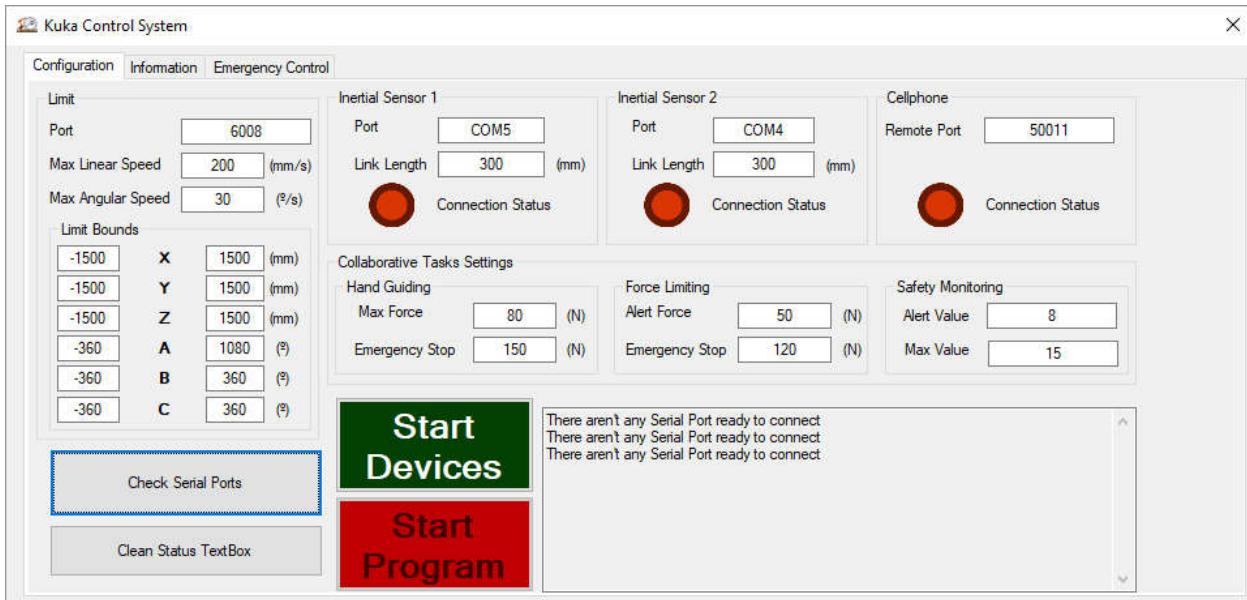


Figura 50 – Aba *Configuration* da Interface Homem Máquina em plataforma Windows

Após a inicialização do KR16 e dos dispositivos externos, a interface muda de um escopo de configuração para trazer informações sobre o estado atual do sistema. Nos campos a esquerda informa todos os valores de posições, tanto do manipulador quanto do braço do usuário e as forças e torques aplicados aplicadas. Na área central, o sistema informa como está a processo, na parte superior, ele indica o acionamento dos modos colaborativos de segurança, abaixo informa se está na etapa de aprendizagem, execução ou emergência (estado de segurança) e na parte inferior um botão de emergência simulado que aciona o estado de segurança ao ser pressionado. No canto esquerdo superior, uma caixa de texto para informar os alertas e mensagens do sistema e na área inferior, um relógio analógico no modelo Gauge que informa a força e fornece ao usuário informações sobre os valores de gatilho do modo de segurança que estiver ativo.

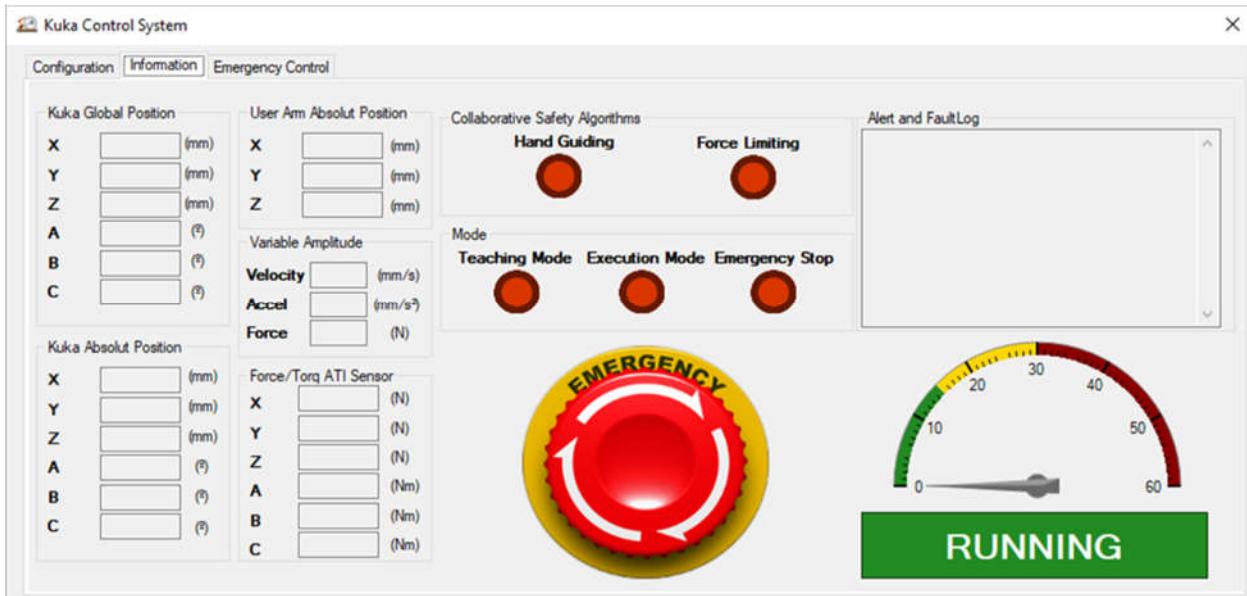


Figura 51 – Aba *Information* da Interface Homem Máquina em plataforma Windows

A última aba, não possui livre acesso e só é acionada, quando o sistema entra em modo de segurança, podendo ser por um gatilho do modo de segurança colaborativo ou ao pressionar o botão de emergência. Na *Emergency Control* há diversas formas manuais e não automáticas de controlar a posição do manipulador, fornecendo ao operador a liberdade de movimentá-lo para realizar o *reset manual*, duas formas principais de movimentação, passo a passo, onde o usuário informa o valor de cada passo e pressiona o botão de avanço ou retorno e o envio de uma posição absoluta. O *reset manual* é realizado através dos dois botões, *Clear Fault* e *Reset*, primeiro deve pressionar o *Clear Fault* informando ao sistema que o operador está ciente das falhas e alertas e aí o botão do *reset* é liberado.

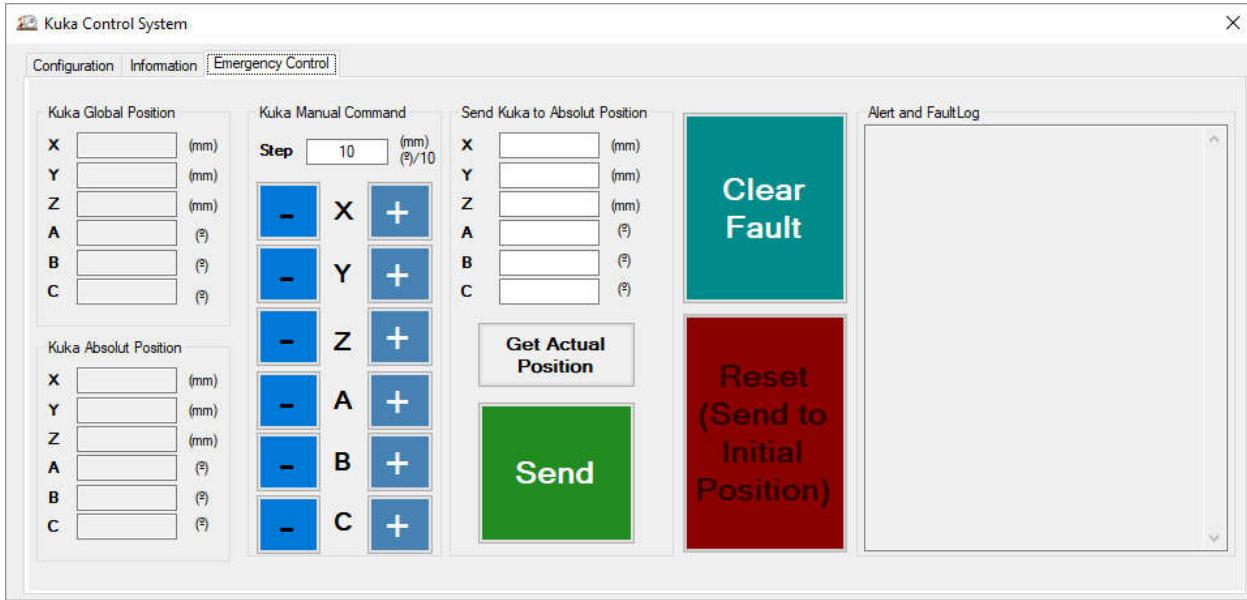


Figura 52 – Aba *Emergency Control* da Interface Homem Máquina em plataforma Windows

4.11. Experimento de Montagem Completa da Gaveta

Inicia-se o processo com a fase de demonstração do usuário ao manipulador robótico, pode ser dividida em três etapas, uma primeira demonstração para o robô chegar até a posição de pega da peça (DMP de pega), outra para trazer a peça (DMP de trazer) e por fim conduzir o manipulador até as posições das placas e ferramentas. Para possibilitar a condução através de demonstração cinestésica foi utilizado com o controle de admitância com M igual a 20Kg, B de 1.2KNs/m e K de 18KN/m. Com todas as posições e as duas trajetórias de aprendizado do DMP demonstradas, o sistema informa ao usuário através do comando de voz do aplicativo Android que está pronto para iniciar a fase de execução.

O primeiro comando é a pega da fita crepe, o *software* irá calcular uma trajetória através da DMP de pega para chegar até a posição da fita, na fase de execução, as trajetórias geradas por meio do DMP utilizaram-se o modo tempo real com impedância com ganhos de 1000Kg, 400KN/m e 80KNs/m para M, K e B respectivamente. Após atingir o objetivo com a garra fechada, insere 20N como força desejada ao controle de admitância no eixo Z e aguarda o sistema estabilizar a força (Figura 53b), a fita crepe repousa em uma base de madeira com espuma embaixo para facilitar a ação do controle de admitância, então a garra é aberta e realiza uma pega externa (Figura 54). Depois de realizar a pega, através do *Q-learning* o sistema retorna a uma área próxima à mão do usuário, a qual foi informada pelos sensores inerciais.

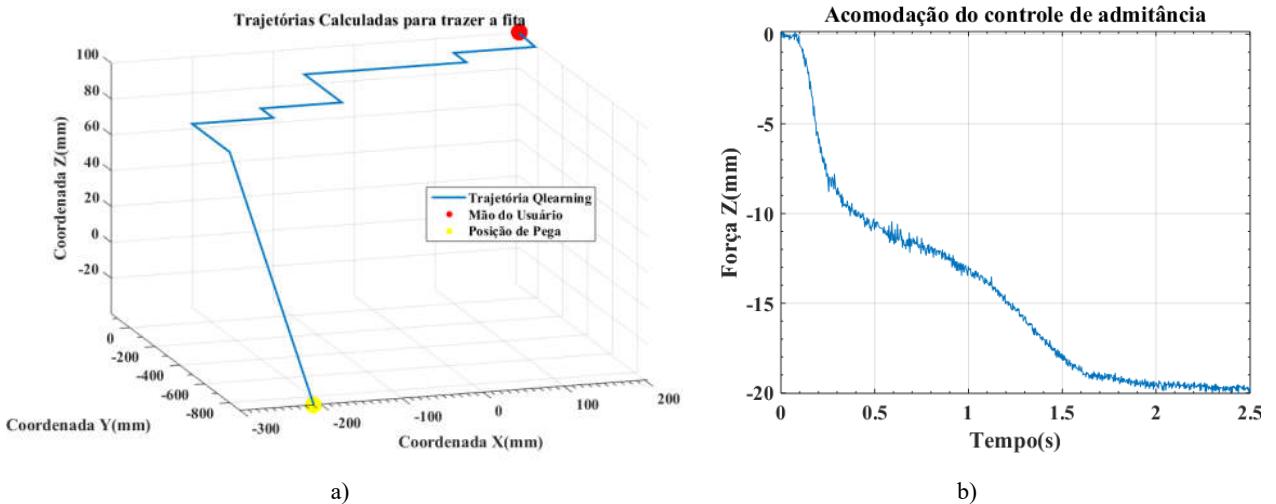


Figura 53 - Processo para pegar e trazer a fita crepe: a) trajetória gerada pelo *Qlearning*; b) Acomodação da força em Z

Com a entrega da fita crepe ao usuário concluída, então inicia o processo de pega de placas de *HoneyComb*, o usuário já possui em sua proximidade a placa base da gaveta, então é necessário pedir ao sistema as quatro placas restantes, neste intervalo, onde o sistema realiza a pega de cada placa, o usuário já pode colocar fita crepe nas bordas da placa base. Percebe-se pela Figura 55 que para realizar a pega das placas, além das coordenadas cartesianas, utilizou-se uma quarta coordenada, a rotação no eixo Z ou coordenada A, isso significa que a DMP foi composta por quatro sistema de transformação, diferente da seção 4.9 que foram utilizadas apenas três coordenadas para a pega das duas placas.

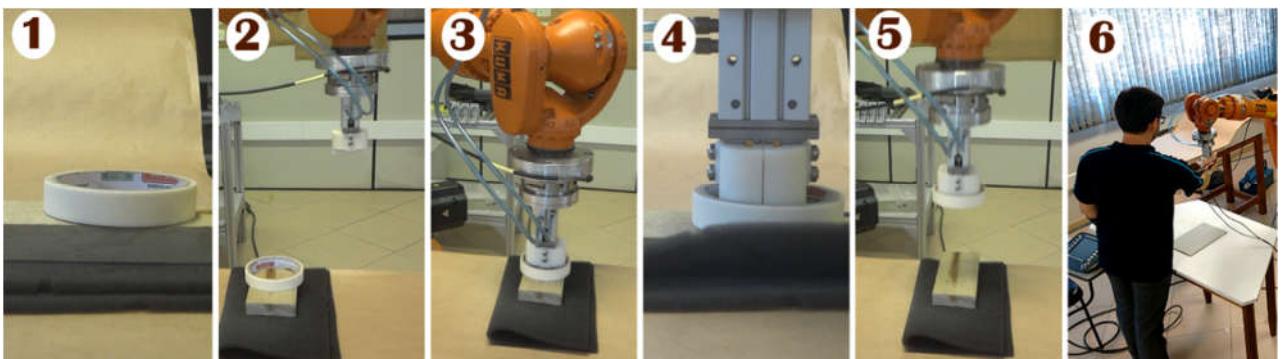


Figura 54 - Sequência de ações para a realizar a pega da fita crepe

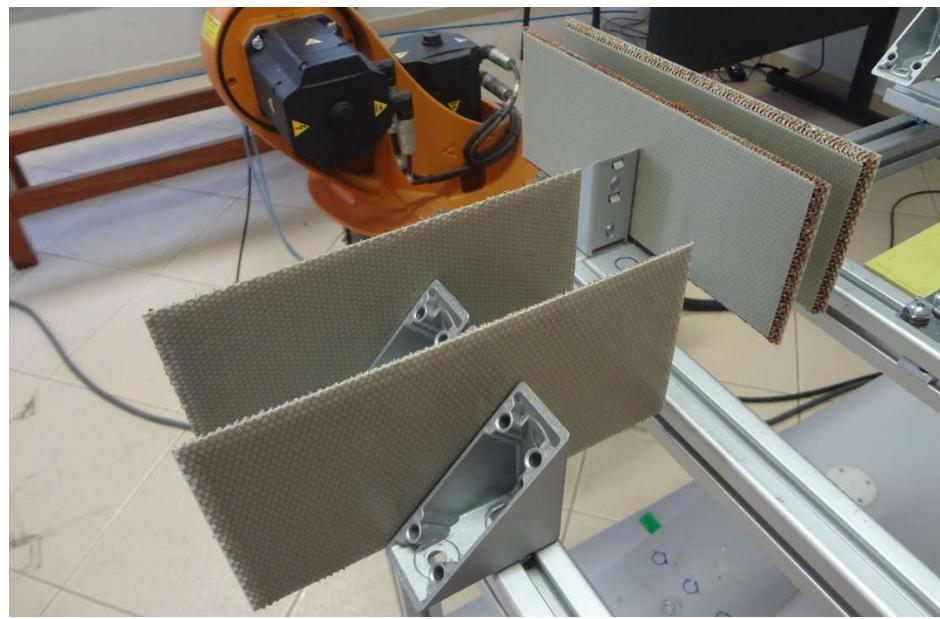


Figura 55 - Distribuição das placas 2 e 3 na bancada

As Figura 56, Figura 57 e Figura 58 estão representadas as trajetórias desenvolvidas pelo manipulador robótico, no movimento de pega das placas (paralelepípedo amarelos) e fita crepe (paralelepípedo vermelho) e para o movimento de trazer as quatro placas. Em todo o processo, foi utilizado apenas duas demonstrações, uma para a pega e outra para trazer, na demonstração foi ensinado a como realizar a pega e trazer a placa quatro, devido a este motivo, em ambas as figuras os trajetos de demonstração e execução estão sobrepostos nesta placa.

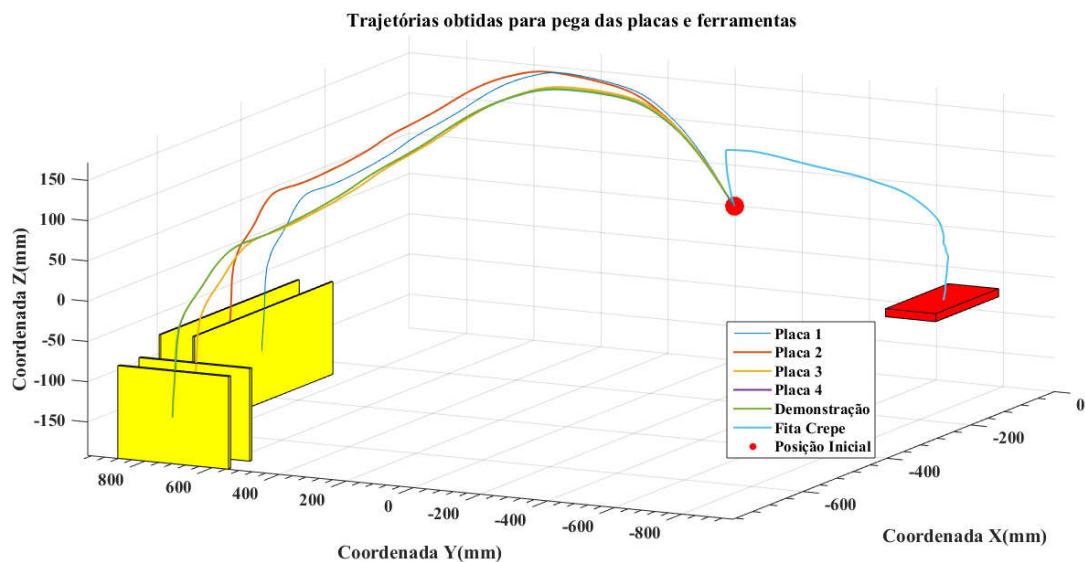


Figura 56 - Trajetórias geradas pelo DMP de pega para atingir os pontos de pega das placas *HoneyComb* e fita

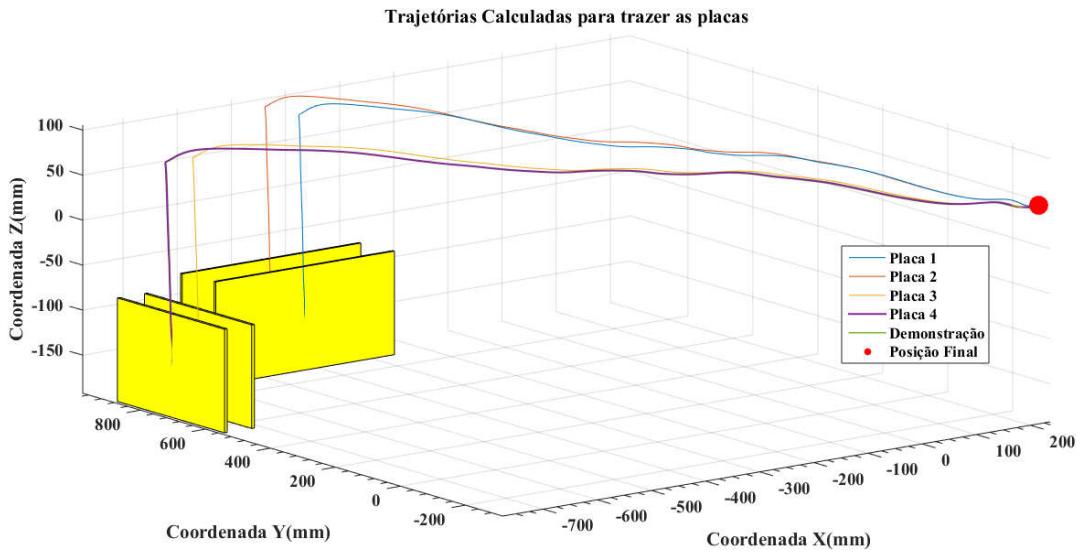


Figura 57 - Trajetórias geradas pelo DMP de trazer as placas *HoneyComb*

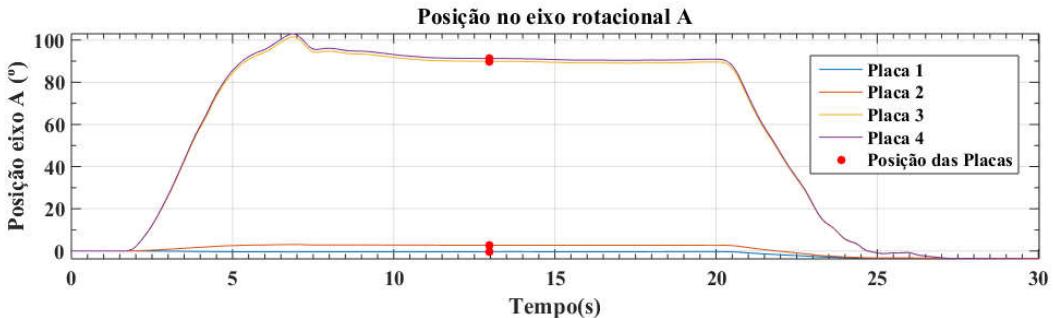


Figura 58 - Trajetória do eixo de rotação durante todo o processo de buscar e trazer as placas *HoneyComb*

Com as cinco placas em mãos e unidas conforme a configuração representada na Figura 21, é solicitado ao sistema o ajuste fino, que consiste em uma ação colaborativa junto ao usuário, no primeiro momento o robô pede ao usuário que o conduza com a garra aberta até a placa desejada, neste momento são ativos a compensação de gravidade e o controle de admitância com 20Kg, 18KN/m e 1.2KNs/m para M,K e B respectivamente, nas coordenadas X, Y, Z e mais um ângulo rotativo, B ou C, neste caso utilizou o ângulo C (Figura 59).

Ao conduzir o manipulador até a posição onde a garra é capaz de segurar a placa que deseja-se ser colocada na posição perpendicular à base, o usuário inicia a segunda fase desse processo de ajuste, os ganhos do controle de admitância aumentam para tornar o sistema mais rígido, a fim de melhorar a interação com a base e as placas, então M de 2000Kg, K de 800KN/m e B de 56KNs/m e uma força desejada de 25N em Z, fecha-se a garra e pede ao sistema que garanta a posição em 90° no eixo de rotação selecionado, devido ao controle de admitância, além de garantir a

perpendicularidade com a precisão do manipulador, o manipulador fica complacente a forças que atuam e corrige-se nos eixos X, Y e Z para que a placa não seja danificada (Figura 60 e Figura 61).

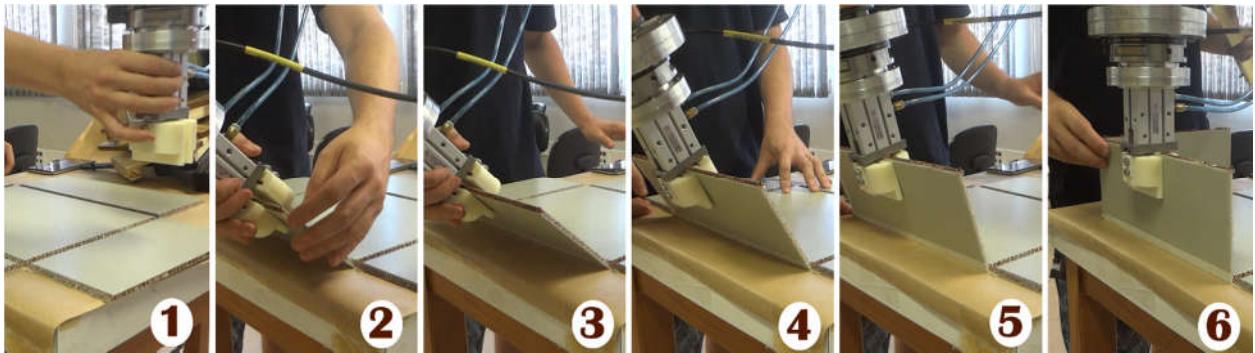


Figura 59 - Sequência do processo de Ajuste Fino

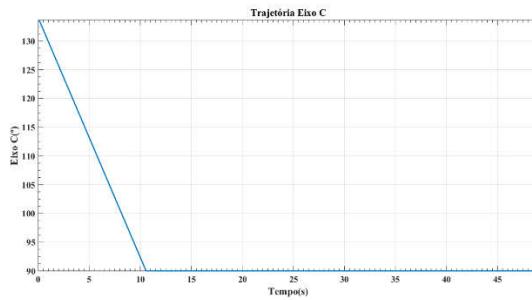


Figura 60 - Movimento no eixo rotativo C para obter a perpendicularidade

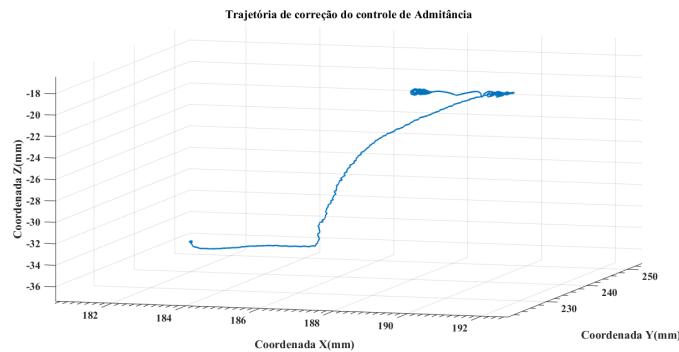


Figura 61 - Correções realizada pelo controle de admittance para o processo de ajuste fino

Ainda com a ajuda do robô segurando uma placa, o usuário coloca a prensa de madeira para fixar as placas nesta posição. A Figura 62 demonstra as etapas para se colocar a prensa, devido as bordas da base de contato perpendicular da prensa, no momento em que a prensa é colocada, esta já garante a perpendicularidade entre as placas. O grampo de aperto rápido é então colocado para forçar as superfícies e assim facilitar o processo de cola, este grampo garante até 2000N de força,

mas não necessita utilizar todo seu desempenho, precisa apenas pressioná-lo até o sistema ficar firme.

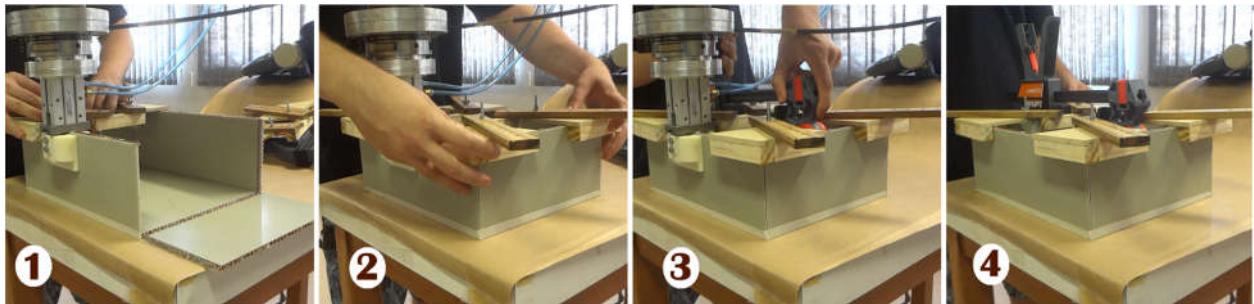


Figura 62 - Sequência de imagens para colocar a prensa na etapa final do processo

4.12. Apreciação de Riscos

Nessa seção realiza-se uma análise de riscos, conforme a norma ISO 12100 (BECKER, 2015), foram encontrados cinco riscos. Através de uma análise do processo total de montagem, buscou-se possibilidades e perigos em que o sistema pode oferecer ao manipulador robótico, operador humano e ao ambiente.

TAREFA	RISCO				
	Presença de pessoas no espaço de trabalho do Manipulador Robótico	Instabilidade do controle de impedância	Prensar corpo do operador a parede ou barreira	Prensar membro do operador a mesa	Colisão do Manipulador robótico com usuário
Demonstração de Trajetório ou Pontos de pega	X	X	X	X	X
Tarefa de pegar e trazer ferramentas ou placas	X				X
Ajuste Fino – Condução manual até a placa desejada	X	X	X	X	X
Ajuste Fino – Encontrar e ajustar a placa a normal do plano		X	X	X	X

Figura 63 – Matriz de relação entre tarefas colaborativas e riscos observados

Algum dos riscos já são mitigados pelo próprio sistema de controle do manipulador robótico, como a colisão do manipulador com ele mesmo e caso o mesmo venha perder a conexão

com o *software*. O controle de segurança de torque e velocidade do manipulador apresentam um bom funcionamento, entretanto os limites não são adequados para a aplicação desenvolvida.

Em cada risco foi realizado uma análise HRN (referenciaaaa), como forma de classificação e análise, além disso a mesma análise foi realizada após a implementação dos modos de segurança colaborativa baseada na ISO/TS 15066 como uma forma de comparação.

4.12.1. Risco 1 – Colisão do Manipulador robótico com usuário ou ambiente

Tabela 8 – Analise HRN do Risco 1

Índice	Classificação	Valor
Severidade do Dano considerado (Se)	Perda de 2 membros/olhos ou doença grave	8
Frequência de Exposição ao Risco (Fr)	Constantemente	5
Probabilidade de Ocorrência do Dano (Pr)	Provável	8
Número de Pessoas Expostas ao Risco (NP)	1 – 2 Pessoas	1
Classe de Risco	Alto	320

4.12.2. Risco 2 – Prensar membro do operador a mesa

Tabela 9 – Analise HRN do Risco 2

Índice	Classificação	Valor
Severidade do Dano considerado (Se)	Perda de 1 membro/olhos ou doença grave	8
Frequência de Exposição ao Risco (Fr)	Constantemente	5
Probabilidade de Ocorrência do Dano (Pr)	Esperado	10
Número de Pessoas Expostas ao Risco (NP)	1 – 2 Pessoas	1
Classe de Risco	Alto	400

4.12.3. Risco 3 – Prensar corpo do operador a parede ou barreira

Tabela 10 – Analise HRN do Risco 3

Índice	Classificação	Valor
Severidade do Dano considerado (Se)	Morte	15
Frequência de Exposição ao Risco (Fr)	Constantemente	5
Probabilidade de Ocorrência do Dano (Pr)	Provável	8
Número de Pessoas Expostas ao Risco (NP)	1 – 2 Pessoas	2
Classe de Risco	Inaceitável	600

4.12.4. Risco 4 – Instabilidade do controle de impedância

Tabela 11 – Analise HRN do Risco 4

Índice	Classificação	Valor
Severidade do Dano considerado (Se)	Perda de 2 membros/olhos ou doença grave	8
Frequência de Exposição ao Risco (Fr)	Constantemente	5
Probabilidade de Ocorrência do Dano (Pr)	Provável	8
Número de Pessoas Expostas ao Risco (NP)	1 – 2 Pessoas	1
Classe de Risco	Alto	320

4.12.5. Risco 5 – Presença de pessoas no espaço de trabalho do Manipulador Robótico

Tabela 12 – Analise HRN do Risco 5

Índice	Classificação	Valor
Severidade do Dano considerado (Se)	Morte	15
Frequência de Exposição ao Risco (Fr)	Constantemente	5
Probabilidade de Ocorrência do Dano (Pr)	Provável	8
Número de Pessoas Expostas ao Risco (NP)	1 – 2 Pessoas	1
Classe de Risco	Inaceitável	600

4.13. Ambiente de Colaboração Simulado Proposto

O projeto em questão busca implementar uma tarefa colaborativa, conforme a ISO/TS 15066, todo manipulador colaborativo possui um espaço de trabalho e esse pode ser dividido em áreas colaborativas, onde o operador possui contato direto com o manipulador e não colaborativa, na qual o manipulador possui interação ou contato apenas com o ambiente.

Com a ferramenta de simulação robótica, VREP, e o modelo do Kuka KR16 desenvolvido, foi possível propor um ambiente, na Figura 64a está representada a divisão, na área não colaborativa os modos de segurança colaborativa não estaram ativos, além disso é permitido que o manipulador possua uma velocidade superior ao momento em que este está na área colaborativa, por isso essa área deve ser inclausurada e o usuário não pode ter acesso enquanto o sistema não estiver em um estado seguro.

Neste ambiente, todo o trabalho de montagem de gavetas poderá ser realizado. A mesa interna a grade é para as ferramentas e utensílios, como o Grampo rápido e fita crepe, a esteira irá alimentar o processo com as peças de mobiliário aeronáutico, enfim na mesa próxima ao operador humano, é aonde este irá realizar a montagem com o auxílio do manipulador e das ferramentas.

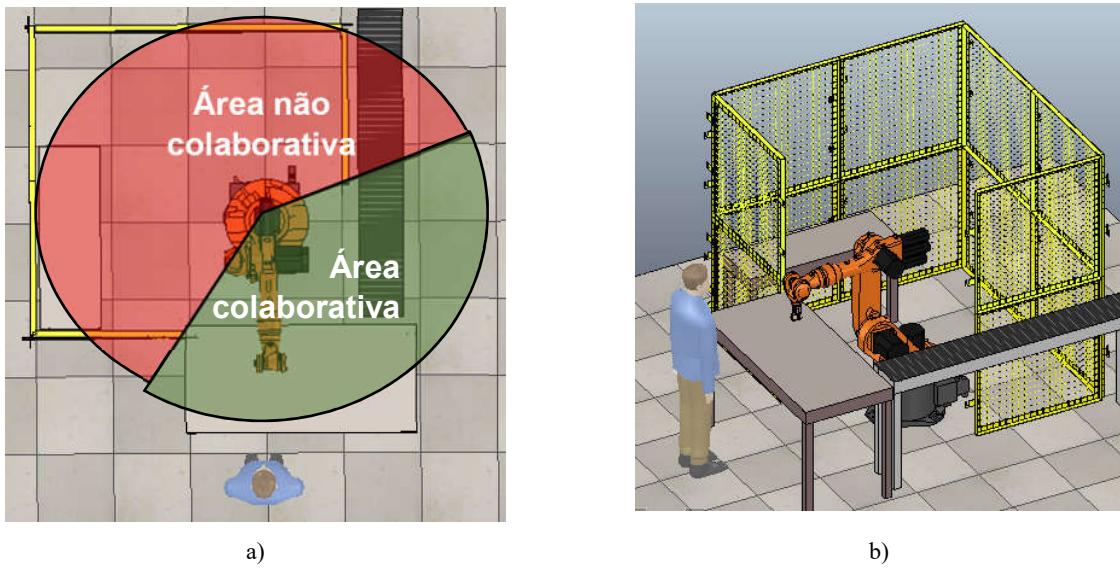


Figura 64 – Ambiente Colaborativo simulado proposto: a) Indicação da área colaborativa e não colaborativa; b) vista global do ambiente.

4.14. Modos de Segurança Colaborativa

Baseado nas divisões das tarefas colaborativas propostas na ISO/TS 15066, dois algoritmos foram desenvolvidos para que de tal forma esses possam ser responsáveis pela segurança de todo o sistema. Além disso, esses modos buscam aumentar a segurança e manter o desempenho do sistema como um todo. O sistema de segurança, bem como sua atuação e medições, são, independentes de todo o *software*, para isso foi desenvolvida uma *thread* dedicada com *loop* infinito, quando ativada, e tempo de leitura menor que o ciclo do KR16 (12ms).

De maneira geral, todos possuem dois valores, nível de alerta e de segurança, que funcionam como gatilhos para eventos, a partir do momento que uma variável ultrapassar esses valores o sistema muda seu estado. No primeiro valor esse gatilho escreve alarmes para usuários e ativa o estado de alerta, mas esse retorna ao estado de operação assim que a variável reduzir seu valor abaixo do nível de alerta. Entretanto, caso a variável eleve e supere o nível de segurança, nesse caso o sistema entra em Estado de Segurança e o mesmo só pode retornar a operação após o *reset* manual. O fluxograma na Figura 65 consiste no fluxograma desenvolvido e aplicado em ambos os modos.

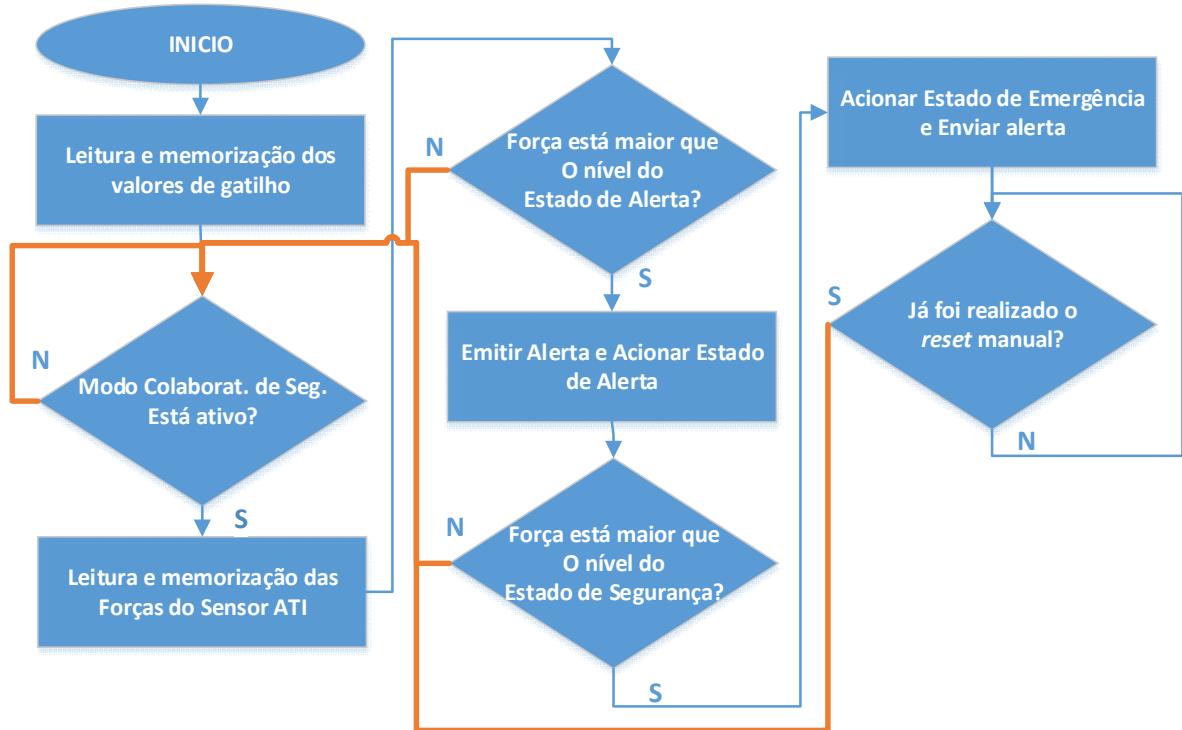


Figura 65 - Fluxograma dos Algoritmos dos modos de segurança colaborativa

4.14.1. Condução Manual

Na condução manual utiliza-se a força aplicada ao sensor de força da ATI acoplado entre a ferramenta e o manipulador robótico. A variável, neste caso a força, é utilizada como valor de gatilho. Neste caso, o primeiro nível, de alerta, foi configurada em 30N e o de segurança em 55N, através da Figura 66, percebe-se que o manipulador pausa ao atingir tal estado.

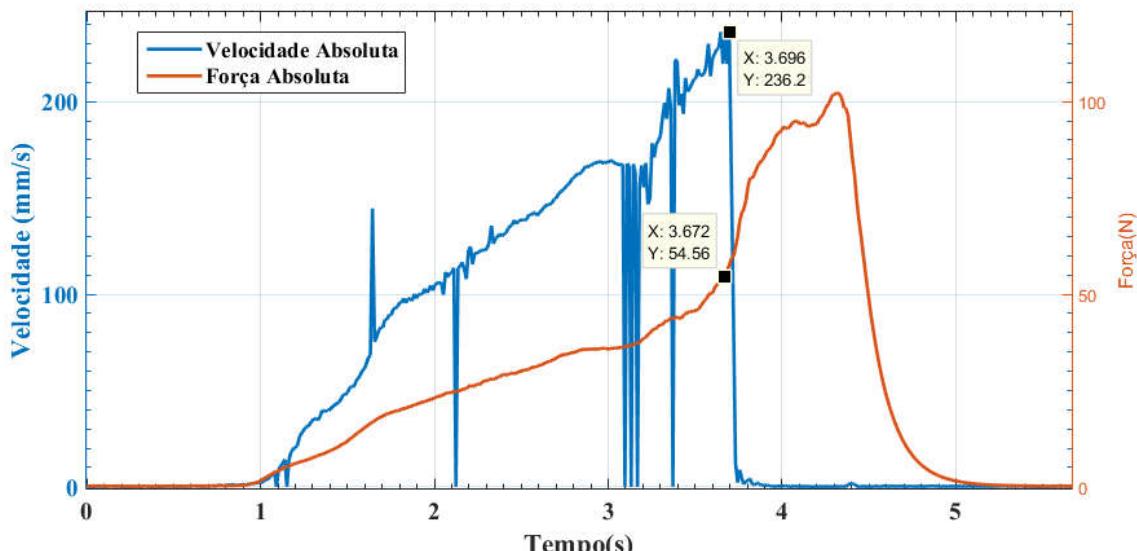


Figura 66 - Gráfico do modo de segurança colaborativa condução manual

4.14.2. Força e Potência Limitada

O modo de Força e Potência limitada usam a força aplicada ao sensor de força para realizar os disparos e diferente dos outros, no estado de alerta, a velocidade absoluta do manipulador é uma proporção de quatro vezes menor da nominal configurada pelo usuário. Na Figura 67, entre o instante 0 e 1s observa-se o estado de operação com uma velocidade limitada em 417mm/s, após a força atingir o primeiro nível, 45N, a velocidade reduz para 97.6mm/s e quando a força atinge o nível de 130N o manipulador é pausado, pois entrou em estado de segurança.

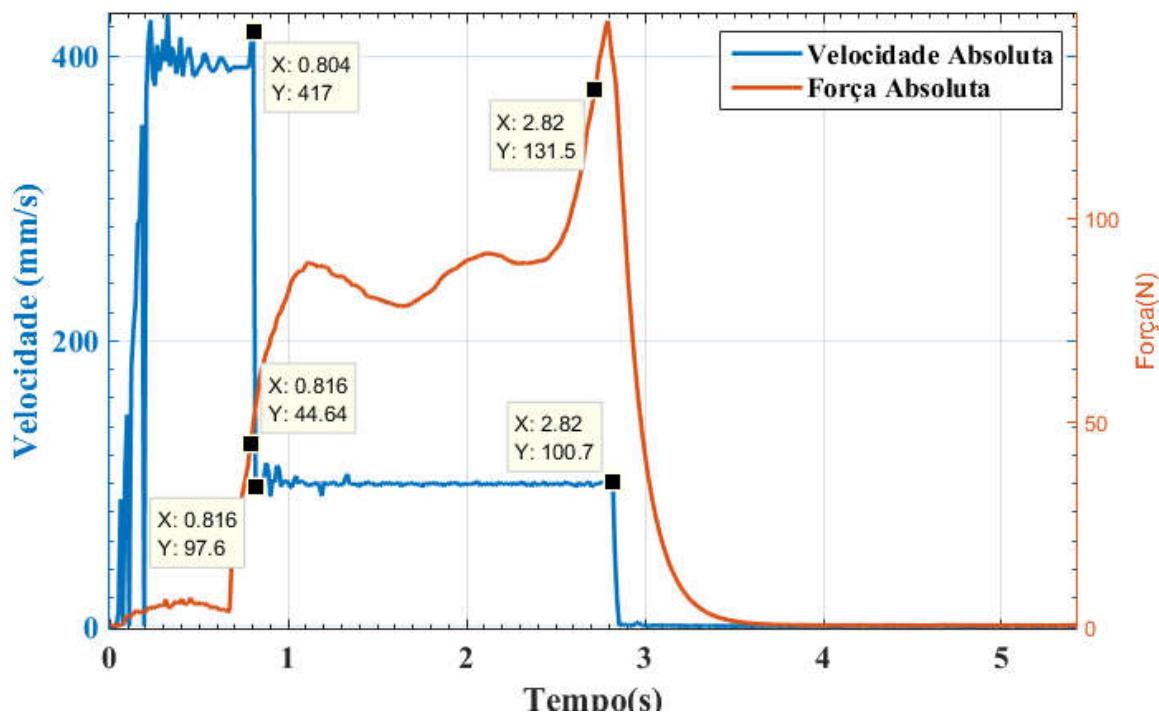


Figura 67 - Gráfico do modo de segurança colaborativa força e potência limitada

4.15. Apreciação de Riscos após modos de segurança e ambiente colaborativo

Uma nova análise HRN foi elaborada, a fim de comparar o resultado da aplicação dos métodos baseados na ISO/TS 15066. Antes da inserção das técnicas, três riscos apresentavam classe de risco alta e dois desses classe de risco inaceitável, essas classificações demonstram que era necessário implementar formas de mitigar e reduzir o impacto de tais riscos.

A severidade do dano considerado nos riscos 1,2,4 e 5 foram reduzidos devido aos modos de segurança colaborativa, que são capazes de mensurar a força e atuar na posição e velocidade do

manipulador robótico. Outra redução significativa foi a frequência de exposição nos riscos 3 e 5, ocorreu-se devido a separação de áreas entre colaborativa e não colaborativa, como também a inserção de grades que dificultam o acesso à parte não colaborativa.

Através das subseções a seguir, percebe-se que as técnicas foram suficientes para o objeto proposto nessa análise, quatro riscos tiveram sua classe reduzidas para o nível mais baixo, insignificante, e uma está com a classe baixo porém significativo.

4.15.1. Risco 1 – Colisão do Manipulador robótico com usuário ou ambiente

Tabela 13 – Analise HRN do Risco 1 após modos de segurança e ambiente colaborativo

Índice	Classificação	Valor
Severidade do Dano considerado (Se)	Laceração/Efeito leve na saúde	0,5
Frequência de Exposição ao Risco (Fr)	Constantemente	5
Probabilidade de Ocorrência do Dano (Pr)	Possível	2
Número de Pessoas Expostas ao Risco (NP)	1 – 2 Pessoas	1
Classe de Risco	Insignificante	5

4.15.2. Risco 2 – Prensar membro do operador a mesa

Tabela 14 – Analise HRN do Risco 2 após modos de segurança e ambiente colaborativo

Índice	Classificação	Valor
Severidade do Dano considerado (Se)	Fratura – ossos menores ou doença leve (temporária)	1
Frequência de Exposição ao Risco (Fr)	Constantemente	5
Probabilidade de Ocorrência do Dano (Pr)	Possível	2
Número de Pessoas Expostas ao Risco (NP)	1 – 2 Pessoas	1
Classe de Risco	Baixo porém significativo	10

4.15.3. Risco 3 – Prensar corpo do operador a parede ou barreira

Tabela 15 – Analise HRN do Risco 3 após modos de segurança e ambiente colaborativo

Índice	Classificação	Valor
Severidade do Dano considerado (Se)	Morte	15
Frequência de Exposição ao Risco (Fr)	Raramente	0,1
Probabilidade de Ocorrência do Dano (Pr)	Não esperado	1
Número de Pessoas Expostas ao Risco (NP)	1 – 2 Pessoas	2
Classe de Risco	Insignificante	3

4.15.4. Risco 4 – Instabilidade do controle de impedância

Tabela 16 – Analise HRN do Risco 4 após modos de segurança e ambiente colaborativo

Índice	Classificação	Valor
Severidade do Dano considerado (Se)	Laceração/Efeito leve na saúde	0,5
Frequência de Exposição ao Risco (Fr)	Constantemente	5
Probabilidade de Ocorrência do Dano (Pr)	Possível	2
Número de Pessoas Expostas ao Risco (NP)	1 – 2 Pessoas	1
Classe de Risco	Insignificativo	5

4.15.5. Risco 5 – Presença de pessoas no espaço de trabalho do Manipulador Robótico

Tabela 17 – Analise HRN do Risco 5 após modos de segurança e ambiente colaborativo

Índice	Classificação	Valor
Severidade do Dano considerado (Se)	Morte	15
Frequência de Exposição ao Risco (Fr)	Raramente	0,1
Probabilidade de Ocorrência do Dano (Pr)	Não esperado	1
Número de Pessoas Expostas ao Risco (NP)	1 – 2 Pessoas	2
Classe de Risco	Insignificante	3

5. Conclusão

Todos os objetivos propostos foram realizados, entretanto a validação dos métodos de segurança não puderam ser validados, devido a implementação do sistema de segurança ter sido realizada via software, pois a aquisição de componentes certificados foi inviabilizada por causa do alto custo.

O manipulador robótico KR16 não é um robô colaborativo, como os modelos mais atuais, entretanto este estudo demonstrou que o mesmo foi adaptado e preparado para realizar atividades colaborativas através da inclusão do software de controle desenvolvido neste projeto. É importante, contudo ater-se aos limites do sistema.

Como o ambiente da proposta não é completamente estruturado, se faz necessário um planejamento de trajetória inteligente que seja capaz de realizar uma tarefa, mesmo com a presença de obstáculos próximos ou no caminho a ser percorrido. O aprendizado por reforço tem suprido tal necessidade. Os experimentos demonstraram que a capacidade de aprender características do ambiente e desviar de obstáculos proporcionou ao sistema uma forma segura e eficaz de se deslocar. Tal forma é essencial para atividades colaborativas como a montagem de gavetas aeronáuticas proposta.

De início, esperava-se que os sensores iniciais fossem fornecer informações mais precisas sobre a posição do braço do usuário, mas a cinemática do movimento do braço e do corpo humano demonstrou uma complexidade muito superior à esperada. A imposição que o usuário deve ficar parado e evitar movimentos bruscos com o ombro foi a forma de amenizar tais imprecisões, mas o movimento do ombro é inevitável. Devido a esta complexidade adicional, a posição obtida através de sensores iniciais passou a ser utilizada para definir apenas um volume de proximidade da mão do usuário, que é utilizada para receber ou entregar ferramentas do usuário.

A adição de um aplicativo de celular foi a alternativa encontrada para atender ao objetivo proposto de criação de um software de teleoperação. O aplicativo tem apresentado bons resultados e facilitou o desenvolvimento do processo, já que o usuário não necessita se locomover até um painel de controle do robô para introduzir instruções. O aplicativo foi atualizado para a segunda versão, tornando o sistema ainda mais intuitivo e simples para o próprio usuário.

Referente a área e técnicas de segurança aplicadas, conclui-se que o sistema atende aos requisitos básicos, tanto da NR12 e ISO/TS 15066, quando aplicado ao ambiente proposto via VREP. Devido a restrições de custo, não foi possível adquirir componentes certificados com as

categorias necessárias para a aplicação, entretanto os modos desenvolvidos via *software* foram elaborados com um viés de compreender e aprender o processo de validação e implementação de segurança. Mesmo não sendo possível aplicar as normas de forma completa, o processo de validação foi realizado, com a análise de riscos e HRN inicial, e por fim uma nova análise HRN foi gerada com o intuito de gerar uma comparação. O resultado demonstrou a efetividade das técnicas aqui empregadas.

O experimento de montagem completa de uma gaveta foi realizado com sucesso, com a aplicação do *software* desenvolvido, como motivação para projetos futuros busca-se aprimorar o algoritmo de DMP, pois há alguns problemas como a amplificação do ruído e troca no sentido, estas características não são desejadas em ambientes colaborativos, além disso aprimorar a aplicação de DMP com o controle de admitância e também encontrar meios alternativos para obter os ganhos para calcular a força não linear, como métodos de otimização e inteligência artificial. Com essas melhorias acredita-se que seja possível aumentar o alcance de aplicação do *software* de controle.

6. Referências bibliográficas

- 9 Degrees of Freedom - Razor IMU.** Disponível em: <<https://www.sparkfun.com/products/10736>>. Acesso em: 21 set. 2015.
- ABB. Industrial Safety Requirements for Collaborative Robots and Applications. **ERF 2014 – Workshop: Workspace Safety in Industrial Robotics: trends, integration and standards**, 2014.
- ABB. New safety standards for collaborative robots , ABB YuMi dual-arm robot. **IROS 2015, Workshop “Robotic co-workers - methods, challenges and industrial test cases”**, 2015.
- ARGALL, B. D.; ARGALL, B. D.; CHERNOVA, S.; CHERNOVA, S.; VELOSO, M.; VELOSO, M.; BROWNING, B.; BROWNING, B. A survey of robot learning from demonstration. **Robotics and Autonomous Systems**, v. 57, n. 5, p. 469–483, 2009. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0921889008001772>>.
- ATI INDUSTRIAL AUTOMATION. **F/T Sensor: Delta.** Disponível em: <http://www.ati-ia.com/products/ft/ft_models.aspx?ID=Delta>.
- ATKESON, C. G.; MOORE, A. W.; SCHAAAL, S. Locally Weighted Learning. 1996. Disponível em: <<http://www.cc.gatech.edu/faculty/Chris/Atkeson>>. Acesso em: 20 set. 2016.
- BECKER, A. C. Métodos de avaliação de risco e Ferramentas de estimativa de risco utilizados na Europa considerando Normativas Europeias e o caso brasileiro. 2015.
- CICHORSKI, G. **Estudo dos requisitos necessários de segurança em uma célula robotizada de soldagem de acordo com a norma nr-12.** 2015. 2015.
- DELPRATO, D.; MIDGLEY, B. Some fundamentals of BF Skinner's behaviorism. **American Psychologist**, 1992.
- ELFES, A. Using occupancy grids for mobile robot perception and navigation. **Computer**, v. 22, n. 6, p. 46–57, jun. 1989. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=30720>>. Acesso em: 16 out. 2015.
- HEXCEL COMPOSITES. **Honeycomb Attributes and Properties, A comprehensive guide to standard Hexcel honeycomb materials, configurations, and mechanical properties.** [s.l: s.n.]
- HOFFMANN, H.; PASTOR, P.; PARK, D.-H.; SCHAAAL, S. Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance. **2009 IEEE International Conference on Robotics and Automation**, p. 2587–2592, 2009.
- IJSPEERT, A. J.; NAKANISHI, J.; HOFFMANN, H.; PASTOR, P.; SCHAAAL, S. Dynamical movement primitives: learning attractor models for motor behaviors. **Neural computation**, v. 25, n. 2, p. 328–73, 2013. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3148415/>>. Acesso em: 1 set. 2016.
- KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement Learning: A Survey. **Journal of Artificial Intelligence Research**, v. 4, p. 237–285, 1996.
- KUKA ROBOTER GMBH. **KR 6 ; KR 16 ; KR 6 K ; KR 16 K Description.** 2004. 2004.
- KUKA ROBOTER GMBH. **Kuka.RobotSensorInterface 2.3.** V1. ed. [s.l: s.n.]

LAHR, G. J. G.; SOARES, J. V. R.; GARCIA, H. B.; SIQUEIRA, A. A. G.; CAURIN, G. A. P. Understanding the Implementation of Impedance Control in Industrial Robots. In: 2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR), *Anais...IEEE*, out. 2016. Disponível em: <<http://ieeexplore.ieee.org/document/7783538/>>. Acesso em: 26 nov. 2017.

MELO, M. P. **Apresentacao NR12**. Disponível em: <<https://pt.slideshare.net/paulolobo17/apresentacao-nr12>>. Acesso em: 22 out. 2017.

OMRON ELECTRONICS LLC. Introduction to Collaborative Robot Safet. **Collaborative Robots**, p. 1–2, 2015a. Disponível em: <<https://industrial.omron.us/en/misc/forms/download-robotics-article-1-introduction-to-collaborative-robot-safety>>.

OMRON ELECTRONICS LLC. Safe Power and Force. **Collaborative Robots**, p. 1–2, 2015b. Disponível em: <<https://industrial.omron.us/en/misc/forms/download-robotics-article-6-safe-power-and-force>>.

PROCTER. **Perimeter Guards Strong Gallery**. Disponível em: <<https://www.machinesafety.co.uk/gallery/satech-perimeter-guards-strong/52>>. Acesso em: 22 out. 2017.

ROBOTIQ. ISO / TS 15066 Explained. p. 2–7, 2016.

SCHAAL, S. Dynamic movement primitives---A framework for motor control in humans and humanoid robots. **Proc. Int. Symp. Adaptive Motion Animals**, 2003.

SCHAAL, S.; PETERS, J.; NAKANISHI, J.; IJSPEERT, A. Control, Planning, Learning, and Imitation with Dynamic Movement Primitives. **Workshop on Bilateral Paradigms on Humans and Humanoids, 2003 IEEE International Conference on Intelligent Robots and Systems IROS**, p. 1–21, 2003.

SCHNEIDER. How to choose between EN/ISO 13849 and EN/IEC 62061. **Safety Legislation and Standards**, 2015.

SMC PNEUMATICS. **Parallel Style Air Gripper**. [s.l: s.n.]

SOARES, D. **KUKA - KR16 - AutoCAD,STL,STEP / IGES - 3D CAD model - GrabCAD**. Disponível em: <<https://grabcad.com/library/kuka-kr16>>. Acesso em: 7 jun. 2017.

TAJTI, F.; SZAYER, G.; KOVACS, B.; DANIEL, B.; KORONDI, P. CRM TC covering paper - Robotics trends. **IECON Proceedings (Industrial Electronics Conference)**, p. 48–53, 2013.

TECNO INGENIERIA. **Robots Industriales: paletizado, manipulacion, soldadura y maquinaria industrial**. Disponível em: <http://www.tecnoing.com/p_robotica.aspx>. Acesso em: 7 jun. 2017.

THOMAZ, a. L.; BREAZEAL, C. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. **Aaai**, v. 6, p. 1000–1005, 2006. Disponível em: <<http://www.aaai.org/Papers/AAAI/2006/AAAI06-157.pdf>>.

UMBREIT, M. Industry 4 . 0 Using the Example of Human-Robot- Collaboration. **12th International TÜV Rheinland Symposium**, 2016.

WATKINS, C. J. C. H. **Learning from Delayed Rewards**. 1989. King's College, 1989. Disponível em: <http://www.cs.rhul.ac.uk/home/chrisw/new_thesis.pdf>. Acesso em: 20 set. 2016.

WATKINS, C. J. C. H.; DAYAN, P. Q-learning. **Machine Learning**, v. 8, n. 3–4, p. 279–292, 1992. Disponível em: <<http://link.springer.com/article/10.1007/BF00992698%5Cnhttp://link.springer.com/content/pdf/10.1007/BF00992698.pdf>>.

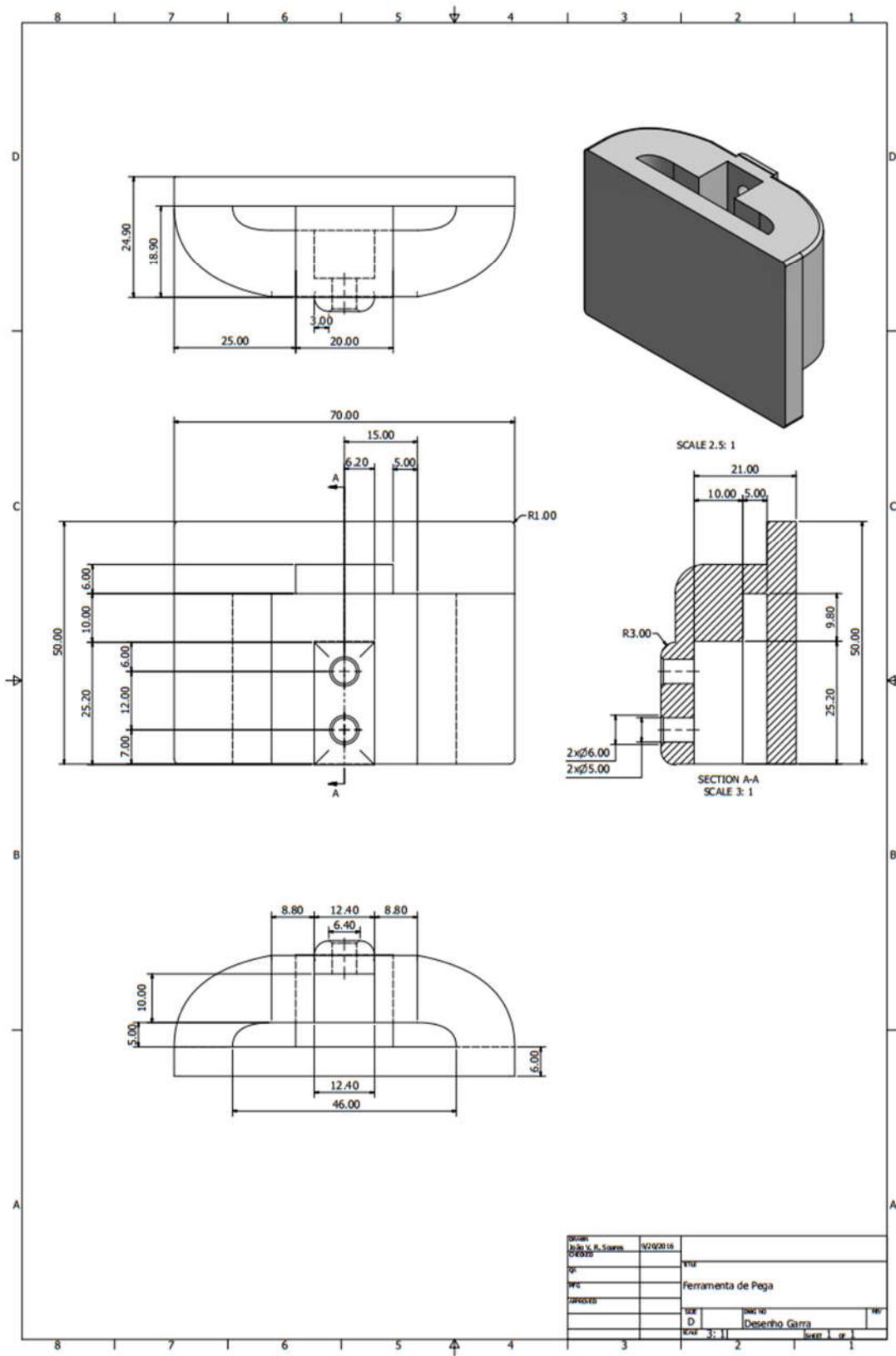
Apêndice A – Pacote XML adaptado do Manipulador KR16 para computador

```
<Rob Type="KUKA">  
<RIst X="930.0542" Y="0.1778" Z="971.7679" A="179.9992" B="0.0011" C="-180.0000"/>  
<RSol X="930.0553" Y="0.1778" Z="971.7679" A="179.9993" B="0.0013" C="180.0000"/>  
<AIPos A1="0.0000" A2="-90.0000" A3="90.0000" A4="0.0000" A5="90.0002" A6="-0.0001" />  
<ASPos A1="0.0000" A2="-90.0000" A3="90.0000" A4="0.0000" A5="90.0000" A6="0.0000"/>  
<EIPos E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000"/>  
<ESPos E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000"/>  
<MACur A1="0.0313" A2="0.9014" A3="-0.7944" A4="-0.0630" A5="0.0129" A6="-0.0430"/>  
<MECur E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000"/>  
<Delay D="0" />  
<FTC Fx="0.0148" Fy="0.0241" Fz="-0.0560" Mx="0.0035" My="-0.0032" Mz="-0.0034" />  
<IPOC>18006082</IPOC>  
</Rob>
```

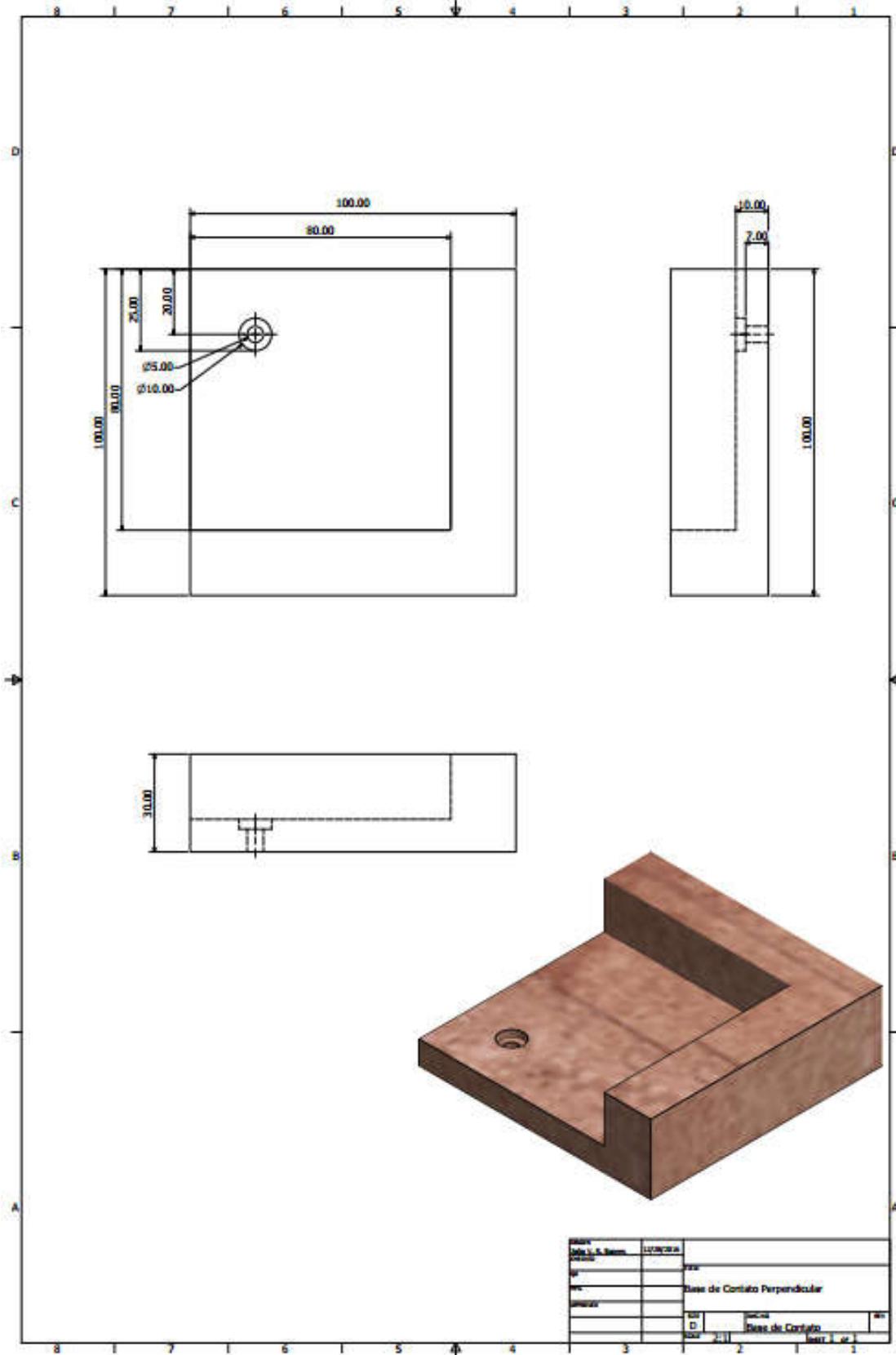
Apêndice B – Pacote XML adaptado do computador para o Manipulador KR16

```
<Sen Type="ImFree">
  <EStr>Laboratorio de Manipulacao Robotica EESC USP</EStr>
  <RKorr X="0.0000" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" A5="0.0000" A6="0.0000" />
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" E5="0.0000" E6="0.0000" />
  <DigIn D1="0" D2="1" D3="0" />
  <IPOC>18006088</IPOC>
</Sen>
```

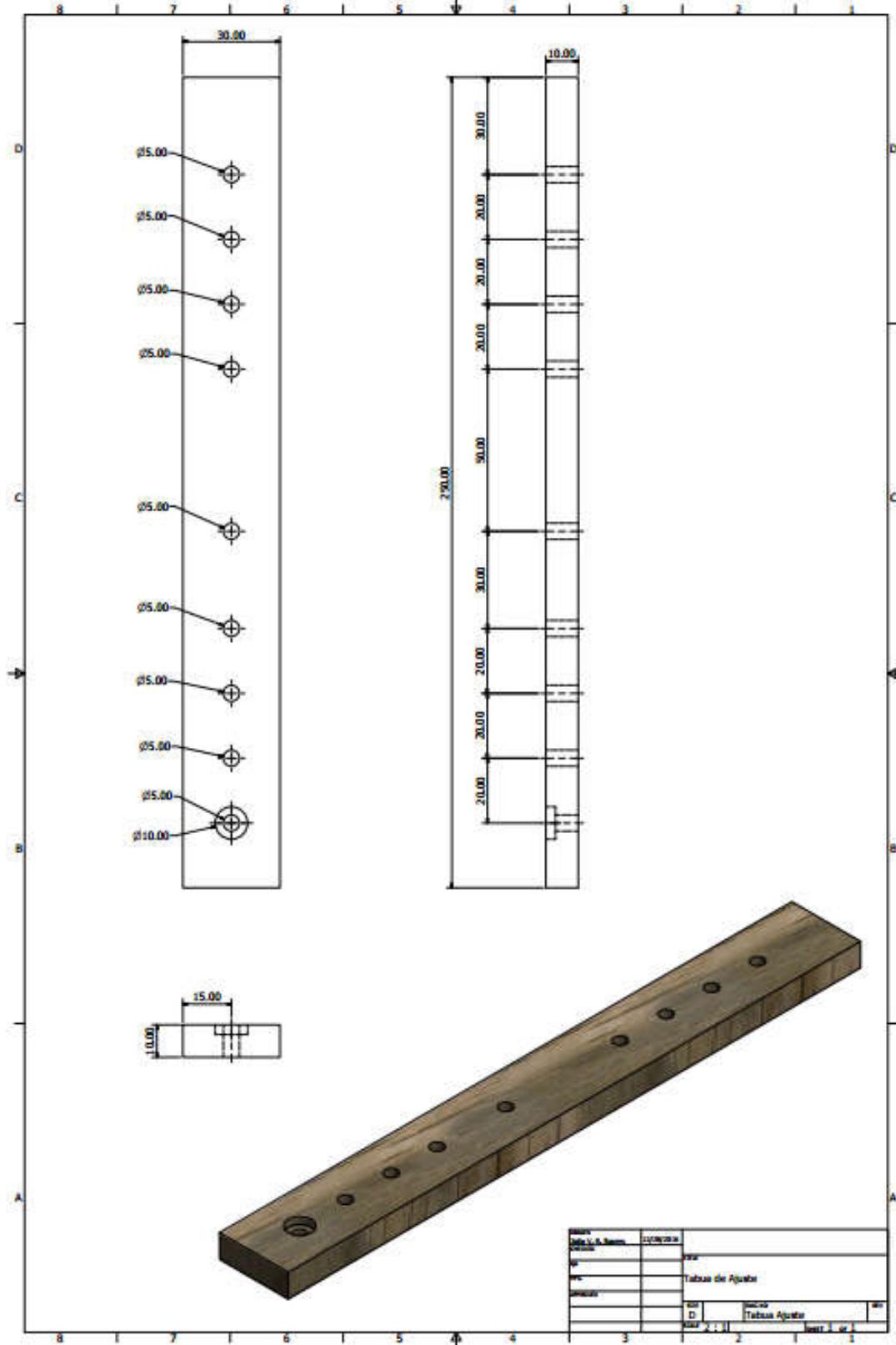
Apêndice C – Desenho Técnico da Ferramenta de Pega



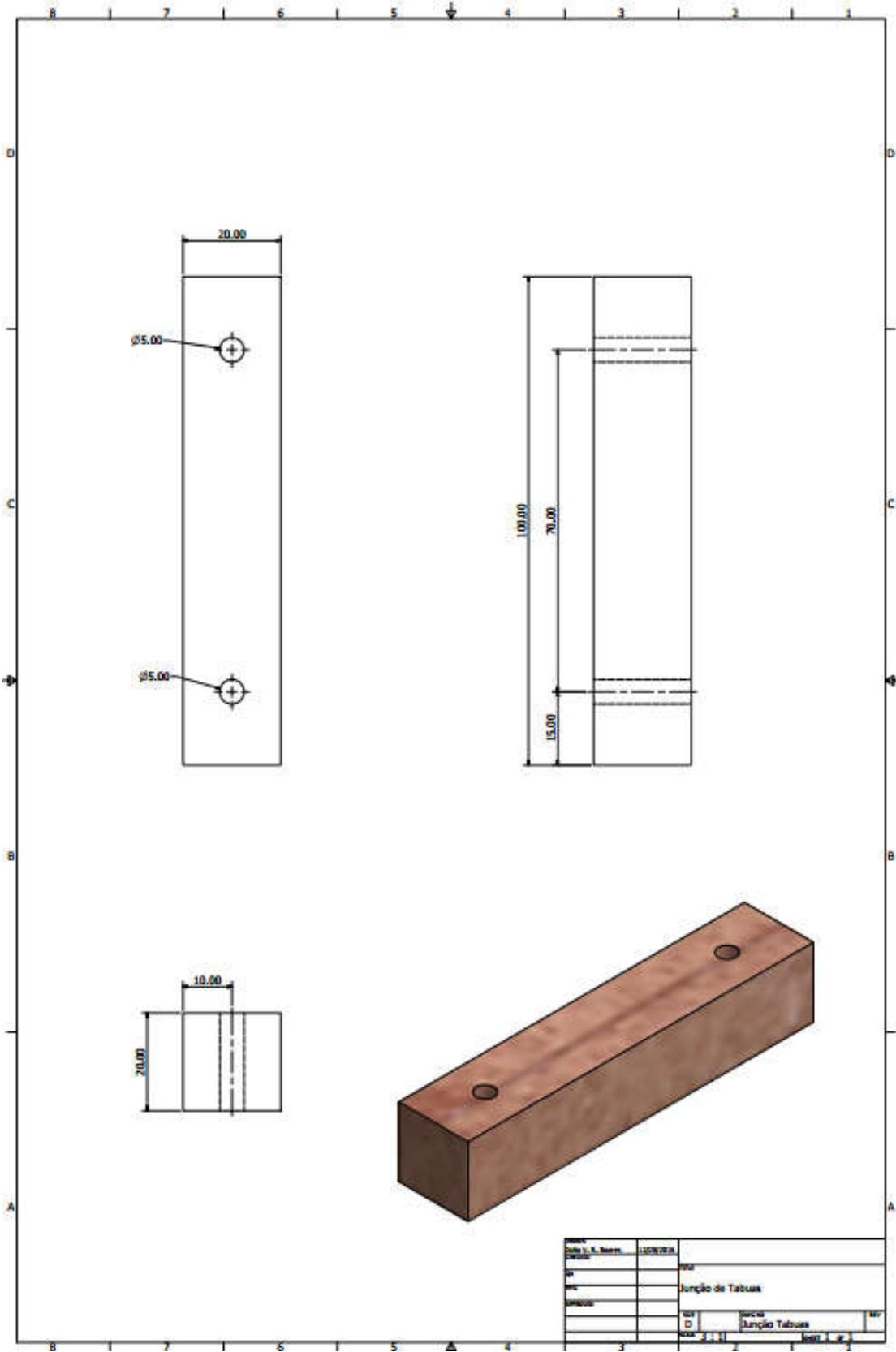
Apêndice D – Desenho Base de Contato da Prensa de Madeira



Apêndice E – Desenho Tabua de Ajuste da Prensa de Madeira



Apêndice F – Desenho Junção de Tabuas da Prensa de Madeira



Apêndice G – Pacote XML da Comunicação Computador e Celular

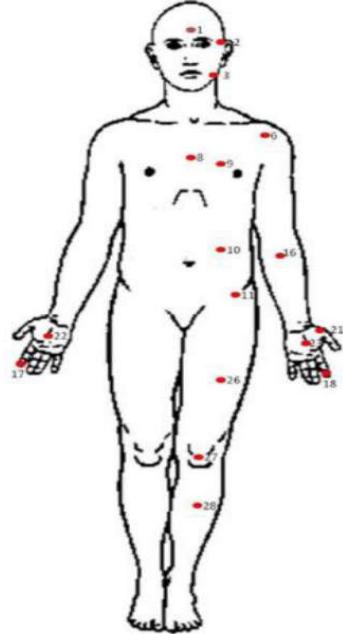
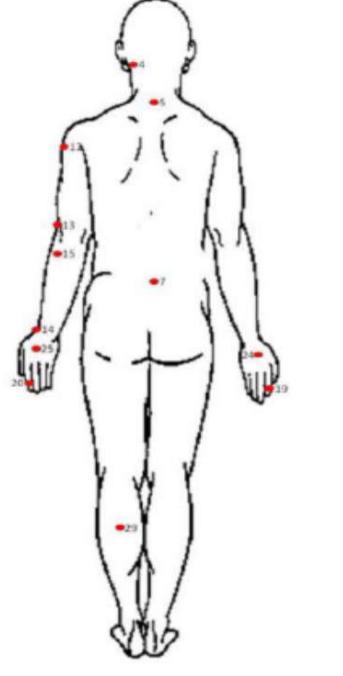
Pacote XML para comunicação UDP no sentido do Celular para o Computador:

```
<Cell2PC>
<YPR Y = "2.63" P = "-0.04" R = "-0.04"/>
<Com = "4"/>
<Pacote = "9144" />
<DigIn = "false" />
</Cell2PC>
```

Pacote XML para comunicação UDP no sentido do Computador para o Celular:

```
<PC2Cell>
<ForTor X = "0.4124" Y = "2.3842" Z = "-12.13" A = "2.1201" B = "0.0324" C = "9.4021" />
<Haptic = 0 />
<Pacote = "9144" />
<Speech = "Hello, I'm_ready_to_start!" />
<Rdy = 1/>
</PC2Cell>
```

Anexo A – ISO/TS 15066 classificação de forças de contato conforme resistência do corpo humano (ROBOTIQ, 2016)

Região do corpo	Área específica do Corpo	Contato Quase Estático	Contato Transiente	Corpo do Operador	
		Máxima Força permitida	Máxima Força permitida	Indicação das áreas específicas	
Cabeça e Fronte	1 Meio da frente	130	130		
	2 Temporal				
Face	3 Músculo Mastigatório	65	65		
Nuca	4 Músculo da Nuca	150	300		
	5 Sétima Vertebra				
Costa e Ombros	6 Junção do Ombro	210	420		
	Quinta vertebra				
Peitoral	7 lombar				
	8 Externo	140	280		
Abdômen	9 Músculo Peitoral				
	10 Músculo abdominal	110	220		
Pélvis	11 Osso da Pélvis				
Braço e cotovelo	12 Músculo Deltóide	150	300		
	13 Úmero				
Antebraço e pulso	14 Osso radial	160	320		
	Musculo frontal do braço				
Mão e dedo	15 Nervo do braço				
	16 Dedo da mão D				
Coxa e joelhos	17 Dedo da mão ND	140		280	
	18 Ponta do dedo da mão D				
Canela	19 Ponta do dedo da mão ND				
	20 Costa da mão D				
	21 Eminência Tenar				
	22 Palma D				
	23 Palma ND				
	24 Costa da mão ND				
	25 Costa da mão ND				
	26 Músculo da coxa	220		440	
	27 Patela				
	28 Meio da Tíbia	130		260	
	29 Panturilha				