

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE DIPLOMAÇÃO – ENG04029

GUSTAVO GRAMS TEIXEIRA

USO DO MOUSE 3D NO ENSINO DE ROBÓTICA

Porto Alegre

2013

GUSTAVO GRAMS TEIXEIRA

USO DO MOUSE 3D EM ENSINO DE ROBÓTICA

Projeto de Diplomação submetido ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como requisito à obtenção do Diploma de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Renato Ventura Bayan Henriques

Porto Alegre

2013

GUSTAVO GRAMS TEIXEIRA

USO DO MOUSE 3D EM ENSINO DE ROBÓTICA

Esse Projeto de Diplomação foi analisado e julgado adequado para a obtenção do título de Bacharel em Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora designada pelo Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul.

Prof. Dr. Renato Ventura Bayan Henriques

Prof. Dr. Altamiro Amadeu Susin

Aprovado em ____/____/____

BANCA EXAMINADORA

Prof. Dr. Marcelo Götz – UFRGS – _____

Prof. Dr. Valner João Brusamarello – UFRGS – _____

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema virtual para movimentação de um robô através de um mouse 3D. A integração do mouse tem como função aproximar o usuário da realidade virtual, pois estarão ao alcance de sua mão os principais movimentos encontrados em um robô industrial. As etapas desenvolvidas são: adquirir o movimento do mouse, interpretar qual movimento foi executado, transmitir para o software de simulação Workspace, movimentar o robô e criar uma interface de operação amigável. As etapas foram executadas de maneira que o leitor compreenda a sua finalidade e também qual o método utilizado para resolvê-la. Os resultados encontrados foram satisfatórios, pois todas as etapas foram resolvidas com êxito, no último capítulo há um teste em uma célula de solda para demonstrar a capacidade do projeto.

Palavras-chave: Robótica Industrial. Mouse 3D. Programação Off-Line.

ABSTRACT

This work presents the development of a virtual system for moving a robot through a 3D spatial device, the mouse integration has the function to approximate the user of virtual reality, because will be within reach of his hand the main movements found in an industrial robot. The development steps are: get mouse movement, which movement was executed interpret, send data to the Workspace software, move the robot and create a friendly operator interface. The steps were performed so that the reader understands its purpose and also which method used to solve it. The results were satisfactory, as all steps have been successfully resolved, and in the last chapter there is a test in a cell weld to demonstrate the ability of the project.

Keywords: Industrial Robotics. 3D Mouse. Off-Line Programming.

LISTAS DE ILUSTRAÇÕES

Figura 1 - Aplicações comuns de robôs Industriais	11
Figura 2 - Robô Cartesiano e robô antropomórfico	12
Figura 3 - Ponto Central da Ferramenta no flange do robô	13
Figura 4 - TCP de uma tocha fixada no flange do robô	13
Figura 5 - Esquema de um manipulador de seis eixos	14
Figura 6 - Configuração de um manipulador de seis eixos.....	15
Figura 7 - Aproximação do pulso do robô.....	15
Figura 8 - Numeração dos eixos de um robô.....	16
Figura 9 - Movimento de junta de um robô	17
Figura 10 - Coordenadas de base de um robô	18
Figura 11 - Movimento Linear de um robô.....	19
Figura 12 - Movimento Linear de um robô.....	19
Figura 13 - Sequência de movimentos de junta e linear.....	20
Figura 14 - Coordenadas do TCP do robô	21
Figura 15 - Sequência de movimentos de reorientação de um robô	21
Figura 16 - Célula de trabalho virtual.....	22
Figura 17 - Visualização do software RobotStudio	24
Figura 18 - TeachPendant virtual do software RobotStudio	25
Figura 19 - Visualização do software MotoSim EG	26
Figura 20 - TeachPendant virtual do software MotoSim EG-VRC.....	27
Figura 21 – Software de visualização do Robotics Suite	28
Figura 22 - TeachPendant virtual do Robotics Suite da Stäubli	29
Figura 23 - Visualização do software WorkSpace	30
Figura 24 - TeachPendant virtual do software WorkSpace	31
Figura 25 - Mouse 3D da empresa 3DCONNEXION.....	32
Figura 26 - Representação dos eixos do mouse 3D.....	33
Figura 27 - Aplicação típica de um mouse 3D	33
Figura 28 - Fluxograma da visão geral de funcionamento do projeto.....	34
Figura 29 - Fluxograma da interpretação da ação do usuário no mouse 3D.....	35
Figura 30 - Sistema de coordenadas do mouse 3D	36
Figura 31 - Fluxograma parcial do recebimento dos movimentos executados	37

Figura 32 - Fluxograma parcial da ação que o manipulador executará.....	38
Figura 33 - Movimento teste executado do tipo junta	40
Figura 34 - Movimento teste executado do tipo linear	42
Figura 35 - Movimento Linear.....	42
Figura 36 - Movimento Linear.....	43
Figura 37 - Movimento teste executado do tipo rotacional A.....	44
Figura 38 - Movimento teste executado do tipo rotacional B.....	44
Figura 39 - Movimento teste executado do tipo rotacional C.....	45
Figura 40 - Painel de Operação desenvolvido.....	46
Figura 41 - Sequência exibida no painel de operação.....	47
Figura 42 - Modos de Operação do painel de operação	47
Figura 43 - Sequência de exibição para informar ao usuário os movimentos disponíveis	48
Figura 44 - Sequência de exibição para outro modo de operação	48
Figura 45 - Exemplo do painel de operação	49
Figura 46 - Exemplo do modo tipo junta.....	50
Figura 47 - Posição padrão do robô para os testes.....	51
Figura 48 - Teste realizado na direção do eixo +1, modo junta.....	51
Figura 49 - Teste realizado na direção do eixo +2, modo junta.....	52
Figura 50 - Teste realizado na direção do eixo +X, modo linear	53
Figura 51 - Visualização dos valores individuais dos eixos	53
Figura 52 - Teste realizado na direção do eixo -Z, modo linear.....	54
Figura 53 - Teste realizado na direção do eixo -rX, modo rotacional	54
Figura 54 - Teste realizado na direção do eixo +rY, modo rotacional	55
Figura 55 - Dimensões de uma viga tipo "I".....	56
Figura 56 - Célula de solda virtual desenvolvida para testes	57
Figura 57 - Célula virtual de solda.....	57
Figura 58 - Aproximando a tocha	58
Figura 59 - Posição inicial para efetuar a solda.....	59
Figura 60 - Posição final da solda	59
Figura 61 - Outra visualização da posição final da solda	60
Figura 62 - Posição inicial para efetuar a solda.....	60
Figura 63 - Operação real do sistema	61

SUMÁRIO

1	INTRODUÇÃO	9
1.1	MOTIVAÇÃO.....	9
1.2	OBJETIVOS.....	10
1.3	ESTRUTURA DO TRABALHO.....	10
2	ROBÓTICA INDUSTRIAL	11
2.1	ESTRUTURA DE UM ROBÔ MANIPULADOR	12
2.2	ROBÔ ANTROPOMÓRFICO	14
2.3	TIPOS DE MOVIMENTO	15
2.3.1	Movimento de Junta	16
2.3.2	Movimento Linear	17
2.3.3	Movimento de Reorientação ou Rotacional	20
3	SOFTWARE DE DESENVOLVIMENTO	22
3.1	ABB – ROBOTSTUDIO.....	23
3.2	MOTOMAN – MOTOSIM	25
3.3	STÄUBLI – STÄUBLI ROBOTICS SUITE	27
3.4	WORKSPACE.....	29
3.5	CONSIDERAÇÕES.....	31
4	MOUSE 3D	32
5	METODOLOGIA	34
5.1	VISÃO GERAL.....	34
5.2	CAPTURE DOS SINAIS DO MOUSE 3D	35
5.3	SOFTWARE DE DESENVOLVIMENTO	37
5.3.1	Tratamento dos Sinais do Mouse	38
5.3.2	Movimentos de Junta	40
5.3.3	Movimentos Lineares	41
5.3.4	Movimentos Rotacionais	43
5.3.5	Painel de Operação	45
5.3.6	Operação e Testes	50
5.4	APLICAÇÃO EM UMA CÉLULA	55
6	CONCLUSÕES	62
	REFERÊNCIAS BIBLIOGRÁFICAS	63
	APÊNDICE A – Guia de Instalação	64

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

O mercado de robótica no Brasil está em plena ascensão. Mesmo durante a crise de 2008 a automação robótica não teve decréscimo significativo, e a partir da recuperação do mercado em 2010 o crescimento anual voltou a ser significativo, e acima do crescimento de outros setores. Segundo dados da Federação Internacional de Robótica, *International Federation of Robotics*(IFR), de 2011, o mercado brasileiro teve grande aumento no número de robôs industriais instalados como também aumentou sua participação no mercado global. (IFR, 2011).

Com o crescimento rápido do mercado e o aumento mais veloz ainda da produtividade nas últimas décadas os métodos para instalar e comissionar manipuladores industriais foi alterado. Não há mais espaço, ou tempo, para a indústria adquirir um robô, instalá-lo e somente após essas etapas um técnico começar a programá-lo. Desta maneira o desenvolvimento de softwares de programação off-line, ou seja, sem a necessidade da presença do operador junto ao manipulador, também sofreram grandes evoluções. Atualmente há uma grande quantidade de softwares que realizam tal operação e também simulam o movimento do robô e outros componentes presentes no ambiente industrial. (CRAIG, 2004 e ROMANO, 2002).

No contexto da simulação ou operação virtual de robôs a grande dificuldade encontrada, para o usuário inicial, é visualizar e interpretar o real movimento do manipulador, ou seja, o que uma certa ação executada no mundo virtual reflete na prática. Essa dificuldade tende a ser reduzida quando o usuário entende o movimento que ele está executando, as novas ferramentas disponíveis e em desenvolvimento ajudam neste entendimento. (BRUMSOM, 2009).

1.2 OBJETIVOS

Os objetivos do presente trabalho podem ser divididos em objetivos geral e específicos. O objetivo geral resume a ideia principal do trabalho, que é: Manipular um robô virtual com um Mouse 3D. E os objetivos específicos são: interpretar os movimentos realizados pelo usuário no mouse 3D; desenvolver uma interface simples e intuitiva no software de desenvolvimento para receber e utilizar os dados oriundos do Mouse 3D; movimentar o robô no modo junta; movimentar o robô no modo linear; movimentar o robô no modo rotacional; e, aplicar o sistema em uma célula de solda.

1.3 ESTRUTURA DO TRABALHO

Os capítulos foram divididos de forma que o leitor tenha um entendimento sequencial e consiga visualizar o projeto desenvolvido na sua totalidade. O capítulo dois introduz alguns conceitos na área de robótica industrial, focando no tipo de robô mais utilizado na indústria e também nos tipos de movimentos executados. O capítulo três expõe alguns softwares para programação off-line encontrados no mercado, suas aplicações e métodos de operação. O capítulo quatro demonstra o que é e onde é utilizado o mouse 3D. No capítulo cinco visualiza-se o desenvolvimento do projeto de modo a cumprir os objetivos por etapas, como por exemplo: identificar o movimento do mouse e movimentar o manipulador.

2 ROBÓTICA INDUSTRIAL

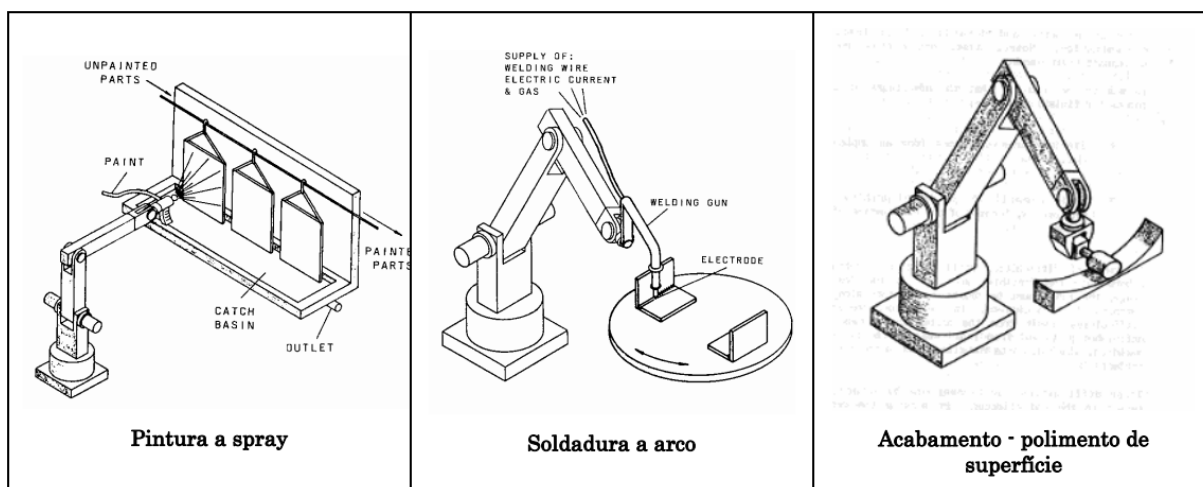
Robôs industriais, ou também chamados de manipuladores, são elementos muito importantes em indústrias de manufatura. O robô é uma máquina multifuncional reprogramável, projetada para movimentar materiais, partes, ferramentas ou peças especiais, através de diversos movimentos programados, para o desempenho de uma variedade de tarefas. (ROMANO, 2002).

As principais vantagens de se utilizar um robô industrial são (SANTOS, 2003; SHIMON,1986; SHAHINPOOR, 1987):

- Técnica: Versatilidade, custos, aumento de produtividade e aumento da qualidade;
- Humana: Tarefas inapropriadas para pessoas;
- Econômica: Um mesmo equipamento pode ter múltiplas funções e substituir vários equipamentos distintos, retorno do investimento em pouco tempo.

Na Figura 1 pode-se verificar três processos onde o robô tem vantagens em sua utilização comparado ao processo manual, devido as restrições de velocidade e precisão inerentes as limitações do ser humano. Um fato socioeconômico importante relacionado com a utilização de manipuladores na indústria, é que não houve aumento do desemprego nos países pesquisados e sim uma redução do número de desempregados mesmo com o aumento do número de robôs. (IFR,2011).

Figura 1 - Aplicações comuns de robôs Industriais



Fonte: SANTOS, 2004

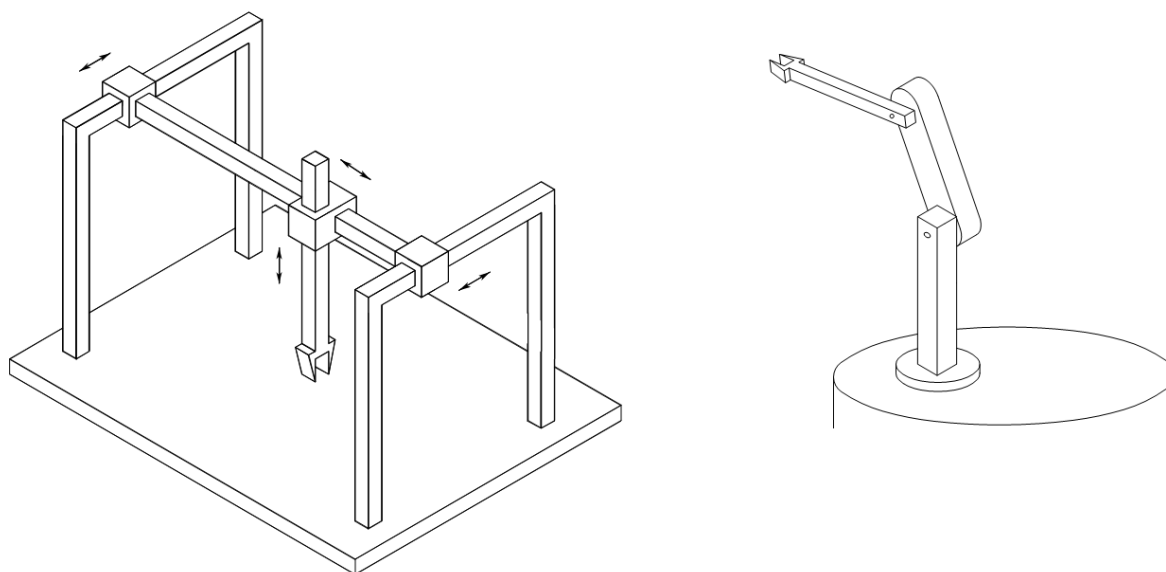
Nas próximas seções serão abordados alguns conceitos fundamentais para o entendimento do objetivo deste trabalho.

2.1 ESTRUTURA DE UM ROBÔ MANIPULADOR

A estrutura mecânica de robô manipulador consiste na combinação de elementos rígidos conectados entre si através de articulações, ou juntas. Um manipulador é caracterizado por um braço que garanta a sua mobilidade, um pulso que assegure a destreza e posicionamento e um elemento final que execute a função desejada.

A mobilidade do manipulador é garantida pela presença de juntas. As juntas realizam a tarefa de articular dois elementos rígidos consecutivos e as mais comuns são: a junta rotacional e a junta prismática. Nas cadeias cinemáticas abertas, geralmente a utilizada em robôs industriais, cada junta representa um grau de liberdade, GDL. Na Figura 2 observa-se, à esquerda, um manipulador com três juntas prismáticas e à direita um manipulador com três juntas rotacionais.

Figura 2 - Robô Cartesiano e robô antropomórfico



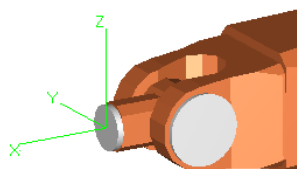
Fonte: SANTOS,2004

O tipo de junta mais utilizado em manipuladores industriais é a junta rotacional, basicamente devido ao seu tamanho e confiabilidade, com esse tipo de junta os robôs passam a ser denominados antropomórficos e geralmente possuem seis juntas.

Outra definição essencial para o aprendizado e desenvolvimento na área de robótica é o conceito de Ponto Central da Ferramenta, cuja sigla é TCP, segundo (ABB, 2004), “O TCP é o ponto em relação ao qual se define todo o posicionamento do robô. Em geral, o TCP é definido como sendo relativo a uma posição no flange do manipulador ou na ferramenta encaixada nele. O TCP também estabelece a origem do sistema de coordenadas da ferramenta.”

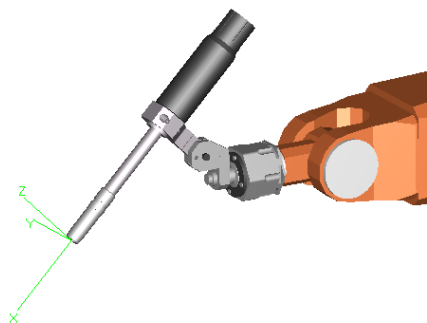
A Figura 3 ilustra a posição do TCP em um manipulador, esta posição pode ser alterada de acordo com a ferramenta que será fixada no flange, um exemplo de ferramenta fixada no flange é observado na Figura 4 e o novo TCP deste manipulador é dado pela origem do sistema de coordenadas.

Figura 3 - Ponto Central da Ferramenta no flange do robô



Fonte: Autor

Figura 4 - TCP de uma tocha fixada no flange do robô

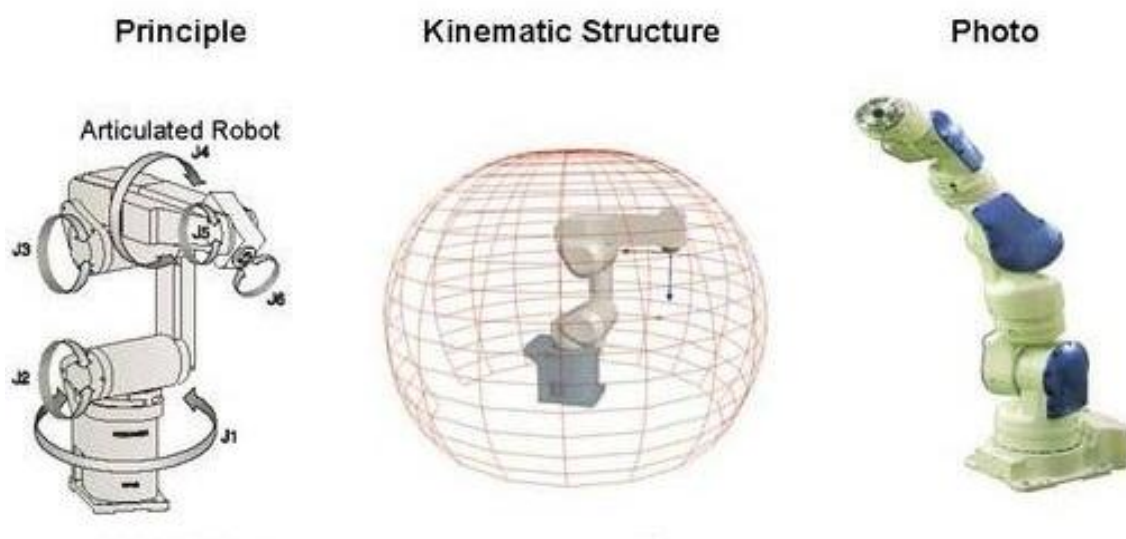


Fonte: Autor

2.2 ROBÔ ANTROPOMÓRFICO

Os robôs de 6 eixos articulados são um dos principais manipuladores utilizados na indústria, basicamente em função de sua flexibilidade do ponto de vista espacial, pois teoricamente alcança qualquer ponto no espaço dentro do seu volume, e também por sua robustez e redução de custos. (IFR, 2011). A Figura 5 demonstra um manipulador de seis eixos genérico, onde visualizam-se no lado esquerdo a indicação dos seus eixos de rotação, na parte central o volume de trabalho do robô, também chamado de envelope, e a direita uma foto desse tipo de manipulador.

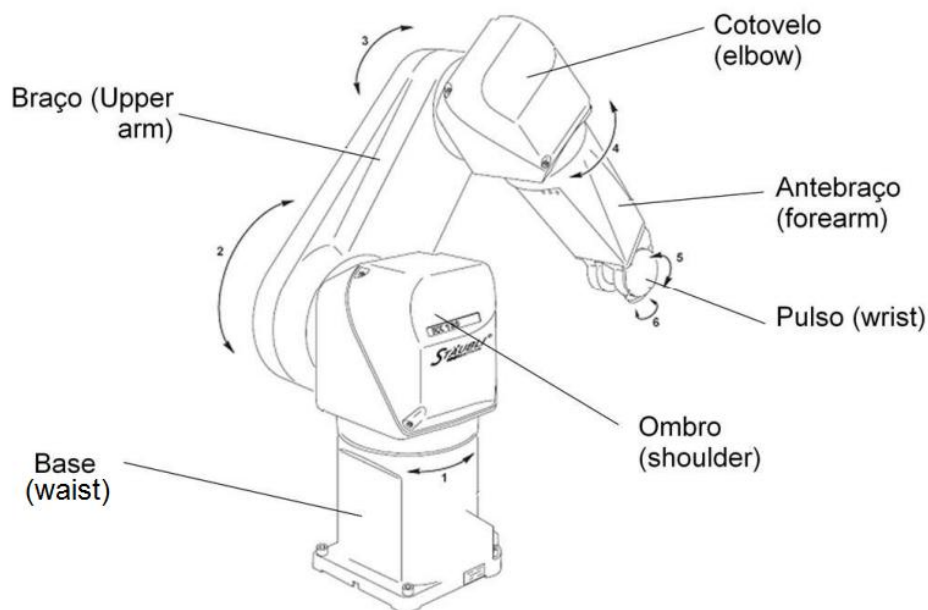
Figura 5 - Esquema de um manipulador de seis eixos



Fonte: IFR, 2013

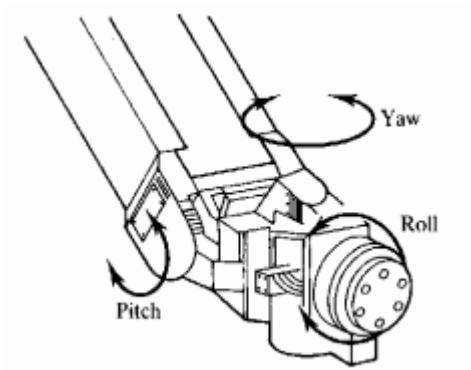
O número de juntas de um robô determina a capacidade de posicionamento do mesmo, e um manipulador de 6 eixos é capaz de se mover para qualquer lugar dentro do alcance do robô. Estes eixos podem ser divididos em duas classes: de posicionamento e de orientação. Na Figura 6 pode-se observar que: os eixos denominados de waist(1), shoulder(2) e elbow(3) definem o posicionamento dentro do ambiente de trabalho do robô e na Figura 7 os eixos denominados de pitch(5), yaw(4) e roll(6), definem a orientação da extremidade do robô. (SANTOS, 2004). A sequência de numeração dos eixos varia conforme o fabricante.

Figura 6 - Configuração de um manipulador de seis eixos



Fonte: Stäubli, 2010

Figura 7 - Aproximação do pulso do robô



Fonte: SANTOS, 2004

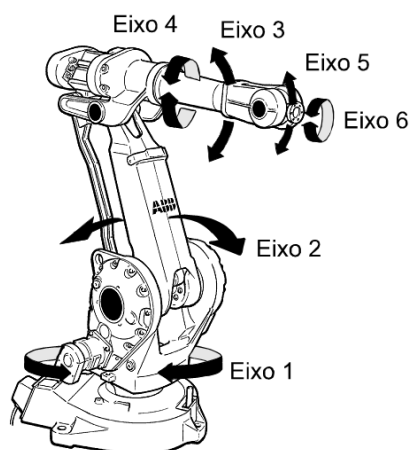
2.3 TIPOS DE MOVIMENTO

Os movimentos manuais em um robô são caracterizados em três tipos: junta, linear ou reorientação. Todos os movimentos são de igual importância na utilização do robô, por isso na sequência serão abordados estes três movimentos e suas respectivas funções.

2.3.1 Movimento de Junta

O movimento de junta consiste em alterar a posição de cada junta individualmente, esse movimento é importante para que o programador ou usuário possa planejar uma trajetória que seja executada no menor tempo possível. Outro fator importante do movimento de junta é o posicionamento do manipulador, pois há posições em que os outros movimentos (linear e reorientação) não são realizáveis. (ABB, 2004). A Figura 8 ilustra os eixos ou as juntas que podem ser ajustadas individualmente.

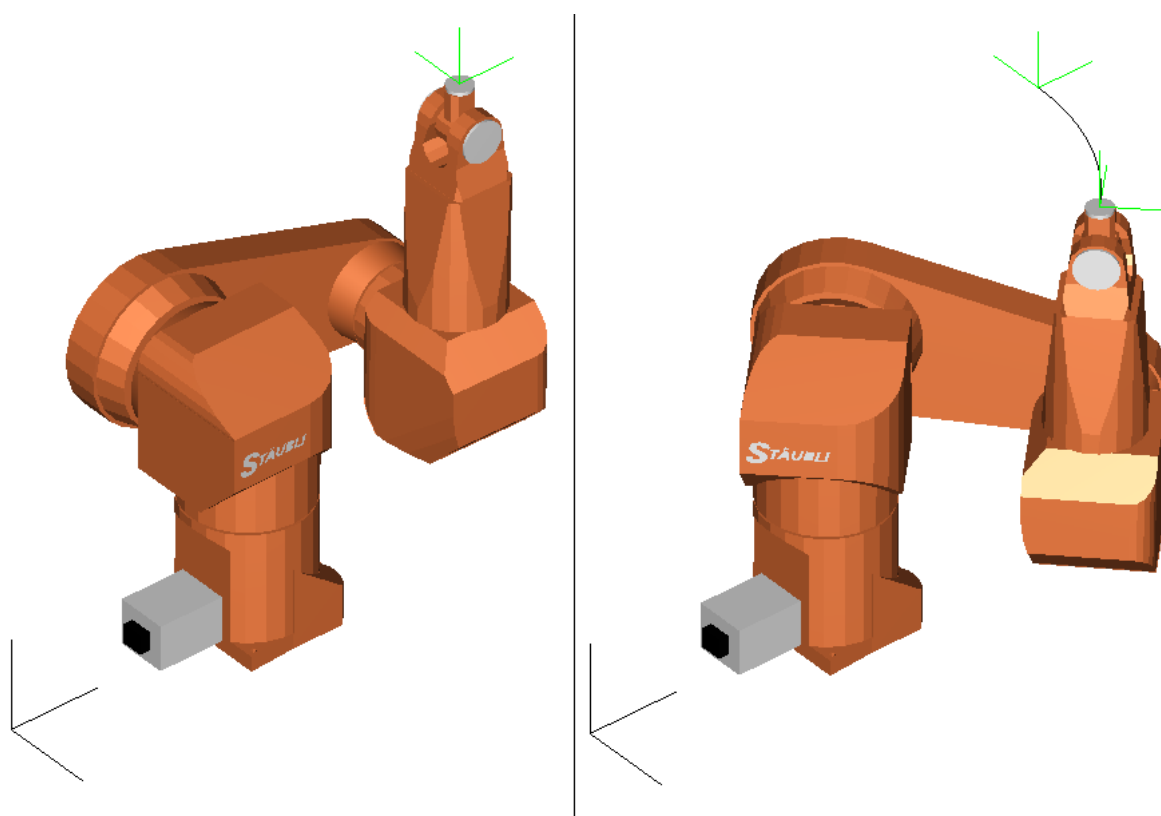
Figura 8 - Numeração dos eixos de um robô



Fonte: ABB, 2004

Para ilustrar o movimento de junta a Figura 9 é apresentada, no lado esquerdo há um manipulador em certa posição, e no lado direito o eixo 1 do manipulador é deslocado em $+45^\circ$ e também é exibida a trajetória do TCP, um item verificado é que os outros eixos permaneceram com os mesmo valores, esta é a característica principal do movimento de junta, pois os outros eixos não são afetados quando é desejado movimentar somente um eixo. É possível movimentar mais eixos ao mesmo tempo, e um não influenciará no outro.

Figura 9 - Movimento de junta de um robô

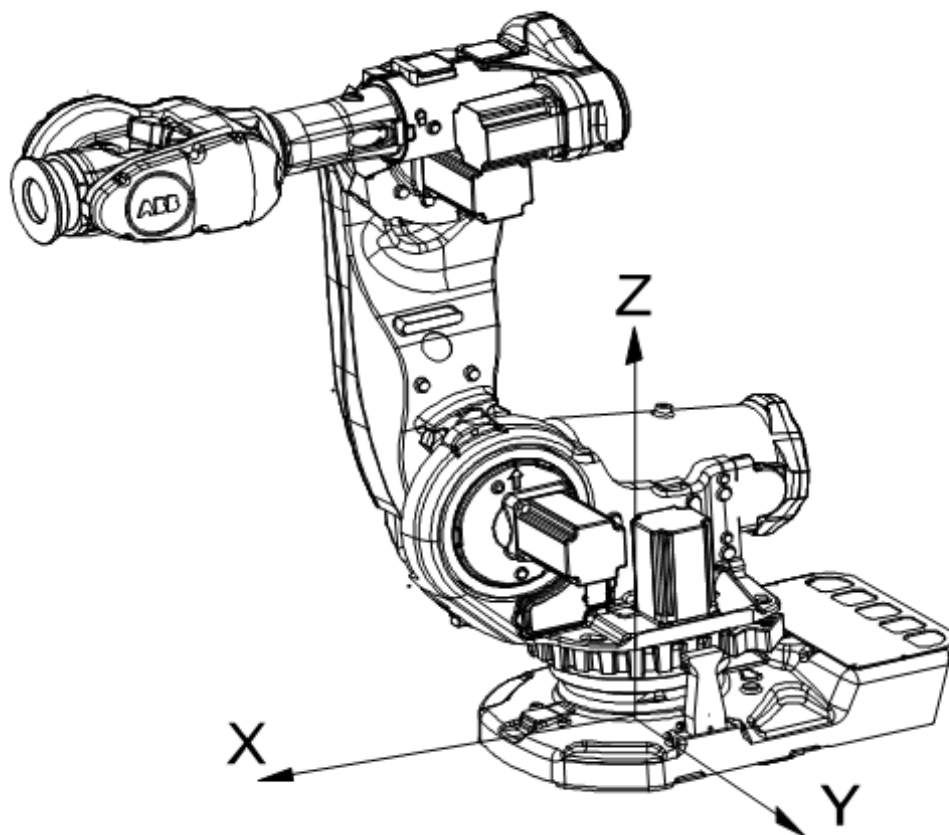


Fonte: Autor

2.3.2 Movimento Linear

O movimento linear é essencial na operação e na programação de um robô. Com ele tem-se o movimento do TCP em um plano cartesiano cuja base é a base do próprio manipulador, ou seja, consegue-se movimentar em uma das direções cartesianas: X, Y ou Z. Neste movimento o usuário decide mover o manipulador 10mm na direção X e o manipulador moverá as juntas necessárias para que o TCP seja deslocado 10mm na direção e ao longo do eixo X da base. A Figura 10 ilustra um manipulador e sua base de coordenadas.

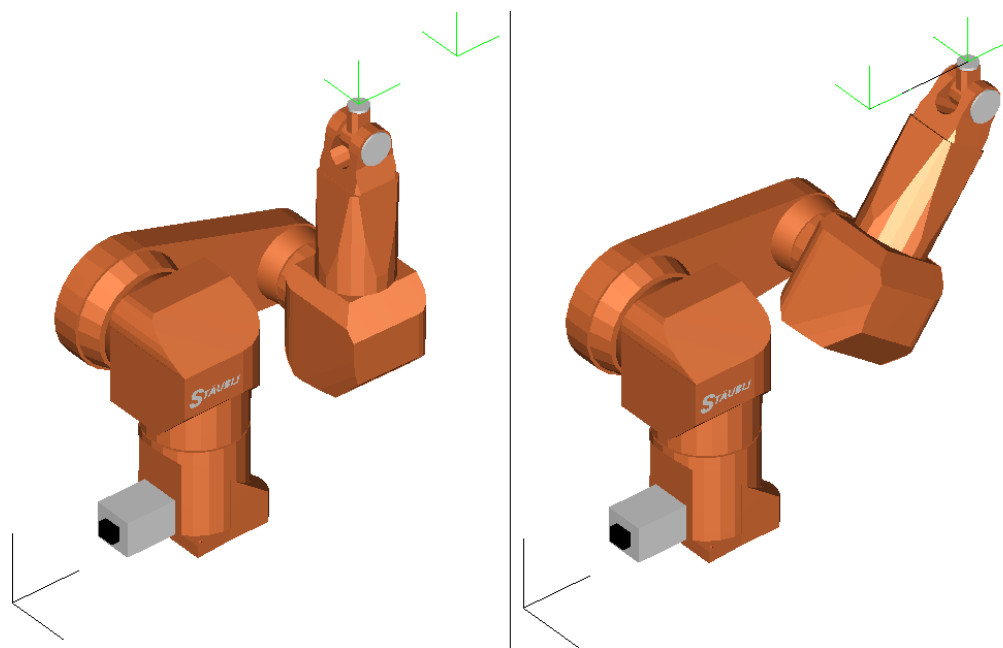
Figura 10 - Coordenadas de base de um robô



Fonte: ABB, 2004

Basicamente, quando o robô está operando em modo automático, se for assim definido, o movimento linear ocorrerá entre dois pontos no espaço, e quando o movimento é manual pode-se somente movimentar em cima de algum eixo pré-definido. Esse movimento respeita a base do sistema, geralmente a base é definida pela posição inicial do robô. Na Figura 11, ilustra-se um exemplo de movimento linear na direção do eixo X. Onde no lado esquerdo observa-se o robô na posição inicial e também as coordenadas de base, e do lado direito o robô é movimentado +250mm.

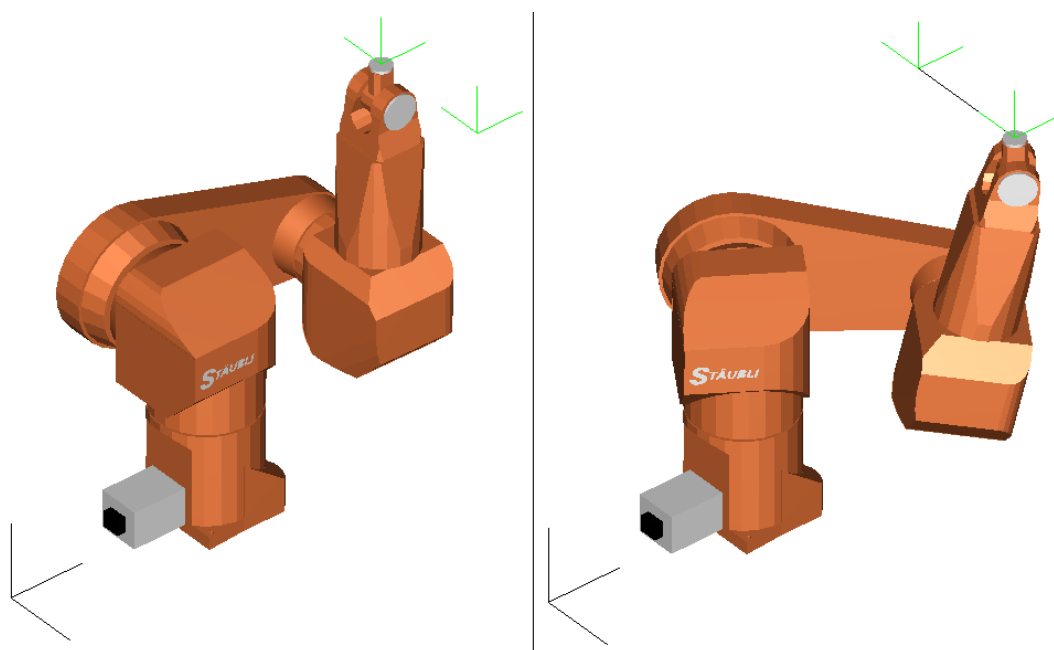
Figura 11 - Movimento Linear de um robô



Fonte: Autor

Na Figura 12 demonstra-se um movimento no eixo Y, do lado esquerdo o robô encontra-se na posição inicial e no lado direito o robô está deslocado 300mm na direção Y+, também observa-se a trajetória executada.

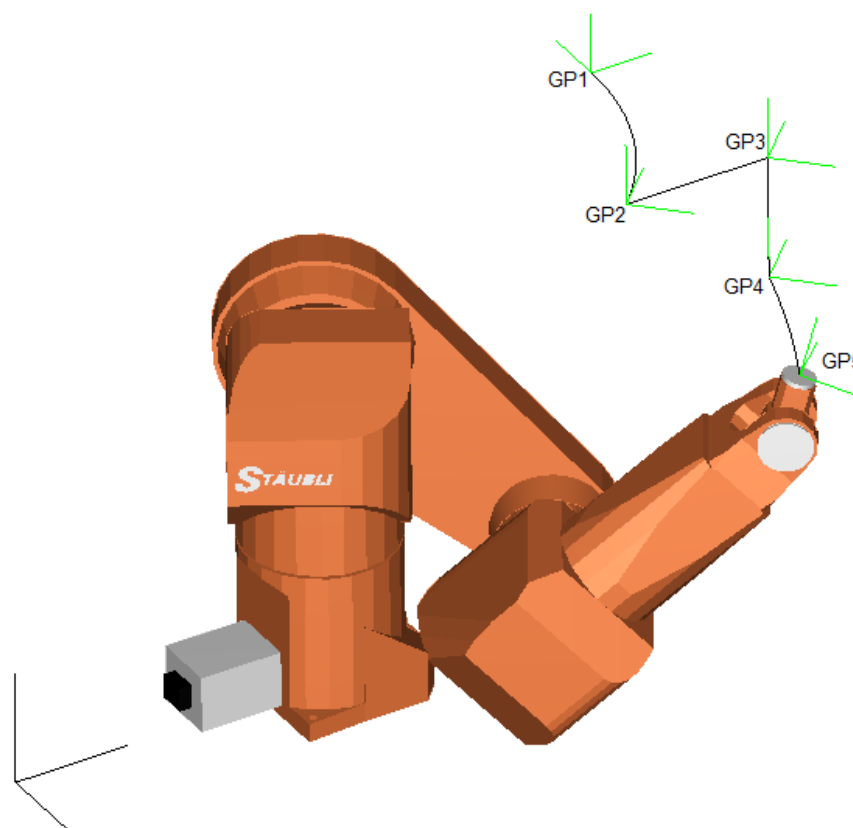
Figura 12 - Movimento Linear de um robô



Fonte: Autor

A Figura 13 exibe uma trajetória de movimentos executados. Nessa trajetória observam-se alguns pontos numerados de 1 a 5: entre os pontos 1 e 2 é executado um movimento de junta, entre 2, 3 e 4 são executados movimentos lineares e entre 4 e 5 é executado movimento de junta novamente. Nota-se que os movimentos de junta a trajetória é curva e nos movimentos lineares é uma reta.

Figura 13 - Sequência de movimentos de junta e linear

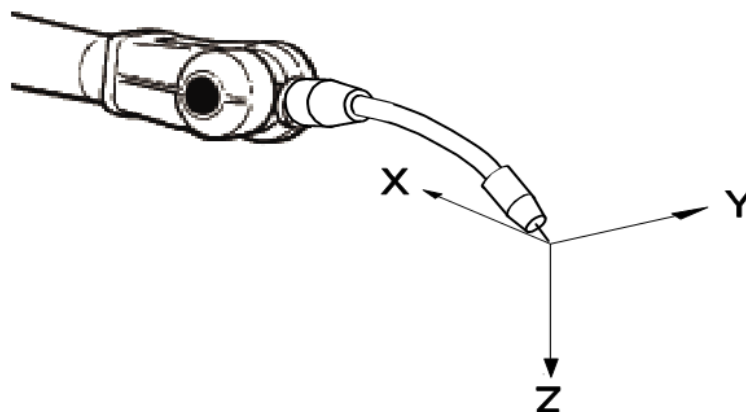


Fonte: Autor

2.3.3 Movimento de Reorientação ou Rotacional

O movimento de reorientação tem como objetivo rotacionar o manipulador em torno do TCP. A principal funcionalidade desse movimento é alterar ângulos de aproximação para efetuar uma solda ou para realizar um furo, por exemplo. A Figura 14 ilustra uma ferramenta fixada no manipulador, e conseqüentemente a nova posição do TCP.

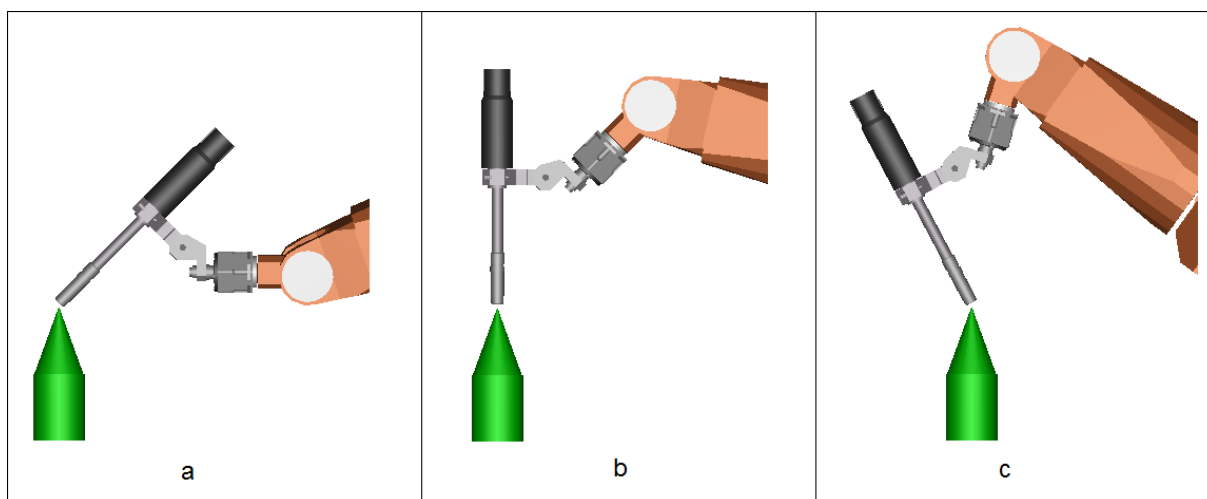
Figura 14 - Coordenadas do TCP do robô



Fonte: ABB, 2004

Com o movimento de reorientação consegue-se mover em torno da origem do TCP, a Figura 15 exibe um manipulador com uma ferramenta acoplada e um sólido qualquer, na qual observa-se que o ângulo da ferramenta é diferente nas posições a, b e c, mas a origem do TCP não é alterada.

Figura 15 - Sequência de movimentos de reorientação de um robô

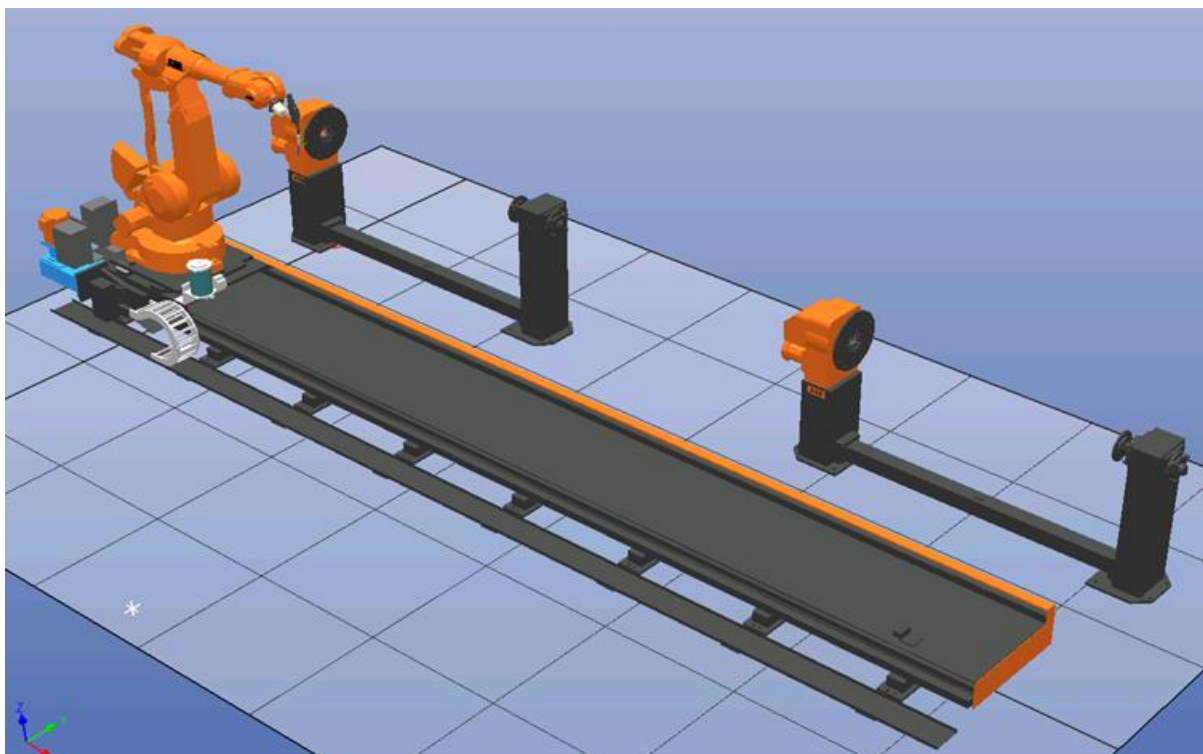


Fonte: Autor

3 SOFTWARE DE DESENVOLVIMENTO

Os softwares de desenvolvimento são interfaces gráficas destinadas à programação off-line de robôs. Nos quais geralmente é possível programar e simular um ambiente industrial. Essa simulação é desenvolvida através de ambientes gráficos 3D. Na Figura 16 visualiza-se um exemplo desse tipo de interface. A diferença básica entre programar robôs no método on-line e off-line é que a primeira exige a presença do manipulador físico, e na segunda é que somente um computador é necessário. (ROMANO, 2002).

Figura 16 - Célula de trabalho virtual



Fonte: ABB, 2013

Em um ambiente industrial, quando há sistemas que possuem lotes pequenos ou médios a serem produzidos e que seja necessária uma reprogramação do robô, o tempo que a célula de manufatura ficará inoperante é crucial, ou seja, em uma visão global o tempo de produção de uma peça será aumentado e com isso a produtividade cairá. Este tempo será muito maior quando o robô for reprogramado no método on-line, pois a célula ficará parada durante toda a reprogramação. Isto não acontece quando é utilizado o método off-line, porque o programa será alterado

em um computador, e somente quando estiver pronto e simulado será atualizado o robô e neste período a célula ficará parada, mas com um tempo muito inferior.

As vantagens do uso de softwares de programação off-line são várias se comparadas com a programação on-line, as principais vantagens são: (ROMANO,2002).

- Redução do tempo ocioso: O robô pode continuar trabalhando enquanto o programa é alterado;
- Ambientes potencialmente perigosos: Redução do tempo do técnico no ambiente, reduzindo também o risco do mesmo;
- Depuração de programas: Verificação de rotas de colisão, facilidade na verificação de erros de software;

As desvantagens do uso de ferramentas de programação off-line são poucas, basicamente tem-se o custo da licença do software e a precisão de pontos específicos(ponto de uma solda ou onde pegar uma peça), para corrigir este erro, geralmente, o técnico precisa ajustar manualmente estes pontos.

Segundo Brumson (2009) "*Estudantes gostam de ver resultados imediatos e isso é possível através de um robô virtual. Eles são igualmente estimulados usando um robô virtual ou usando um robô real.*" E também afirmando que o processo de programação e aprendizado em robótica virtual precede o uso do robô real.

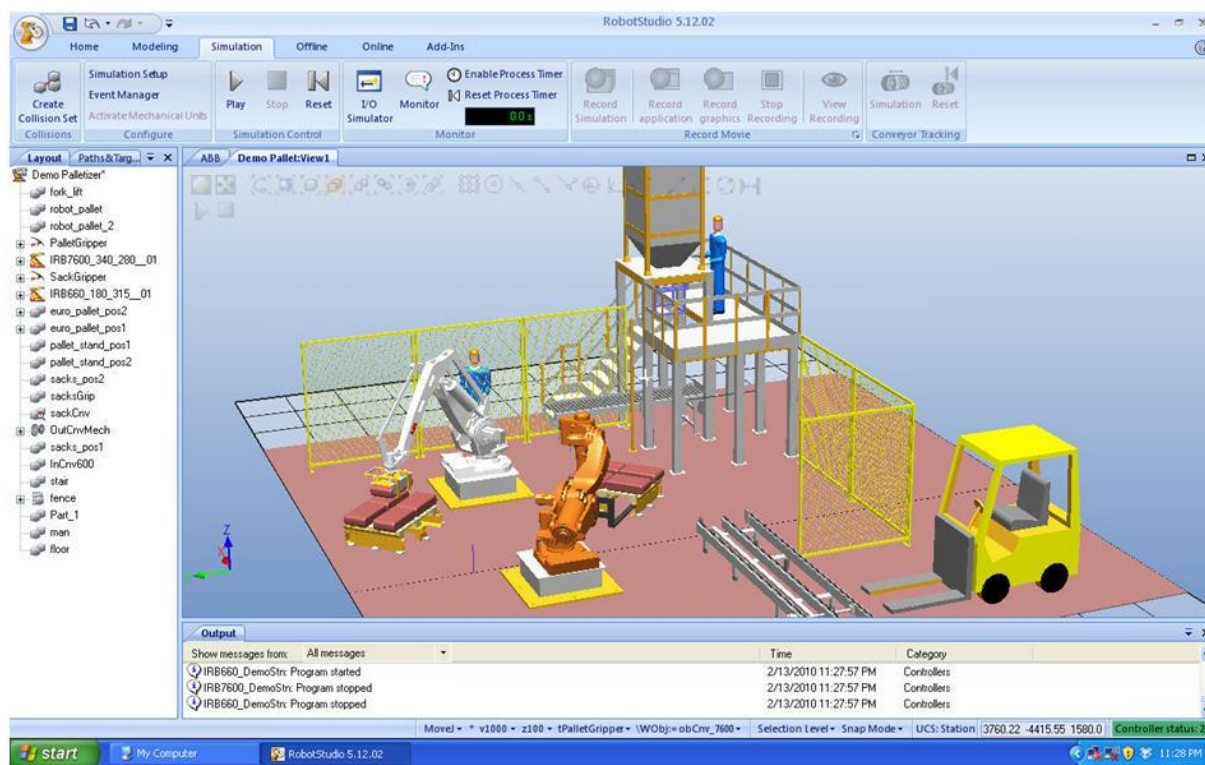
Nas próximas seções serão abordados alguns dos principais softwares de desenvolvimento que dominam o mercado de robótica industrial, com um foco nas características de operação e na facilidade de interpretação gráfica do usuário.

3.1 ABB – ROBOTSTUDIO

O software de desenvolvimento, ou de programação off-line, da Asea Brown Boveri, ABB, é um software proprietário, destinado à programação de equipamentos robóticos da própria empresa cujo nome é RobotStudio. Neste software consegue-se programar e simular manipuladores e posicionadores, conjuntamente com entradas e saídas digitais(IO digitais) e também importar sistemas ou máquinas oriundos de outros softwares CAD. Na Figura 17 tem-se a visualização do software RobotStudio

com dois manipuladores e outros componentes de uma célula de manufatura. (ABB, 2013)

Figura 17 - Visualização do software RobotStudio

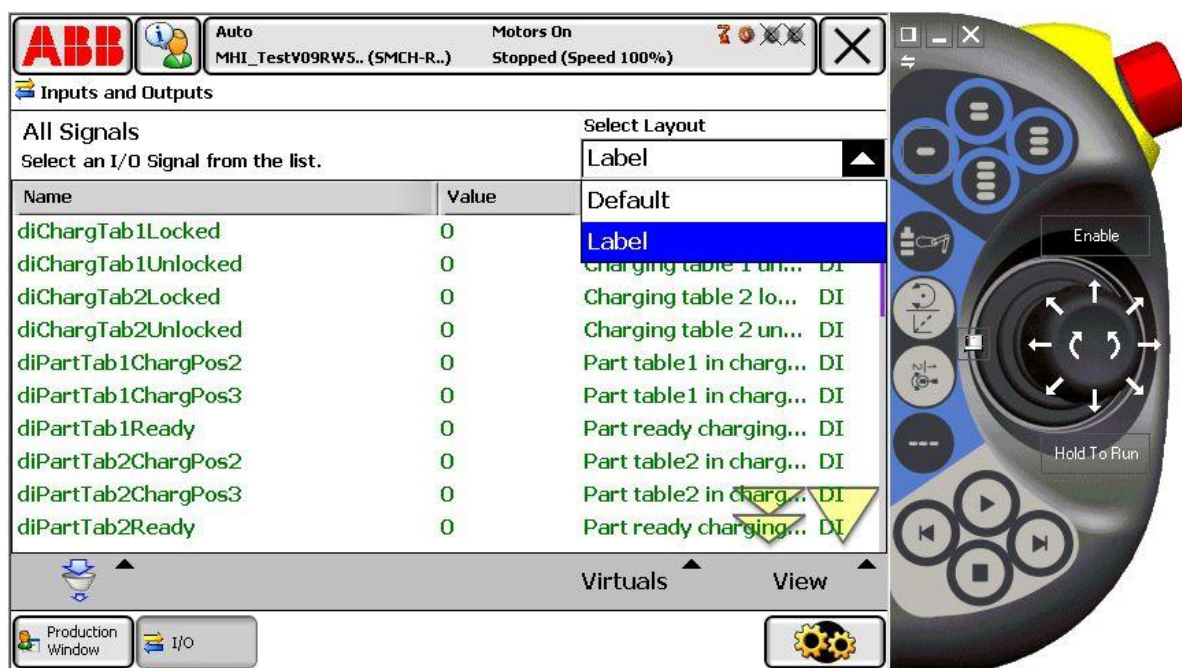


Fonte: Autor

No software da ABB todas as funções executadas estão dentro de um único software, facilitando o seu uso para um profissional experiente e dificultando para um usuário com pouca ou nenhuma experiência. Este software pode ser obtido diretamente da página web da ABB e pode ser utilizado de maneira irrestrita por trinta dias, após esse período somente pode ser utilizado para programação on-line.

Na operação do software tem-se a opção de manipular o robô virtualmente, para efetuar essa operação é utilizado um TeachPendant virtual no próprio RobotStudio. Com ele o usuário consegue operar o robô e simular operações complexas. A Figura 18 ilustra o TeachPendant virtual da ABB, e no lado direito dele encontram-se os botões para movimentação. Um fato importante a ser ressaltado é que as operações de movimentação realizadas pelo pendant são simples, intuitivas e não exigem muita experiência do usuário.

Figura 18 - TeachPendant virtual do software RobotStudio



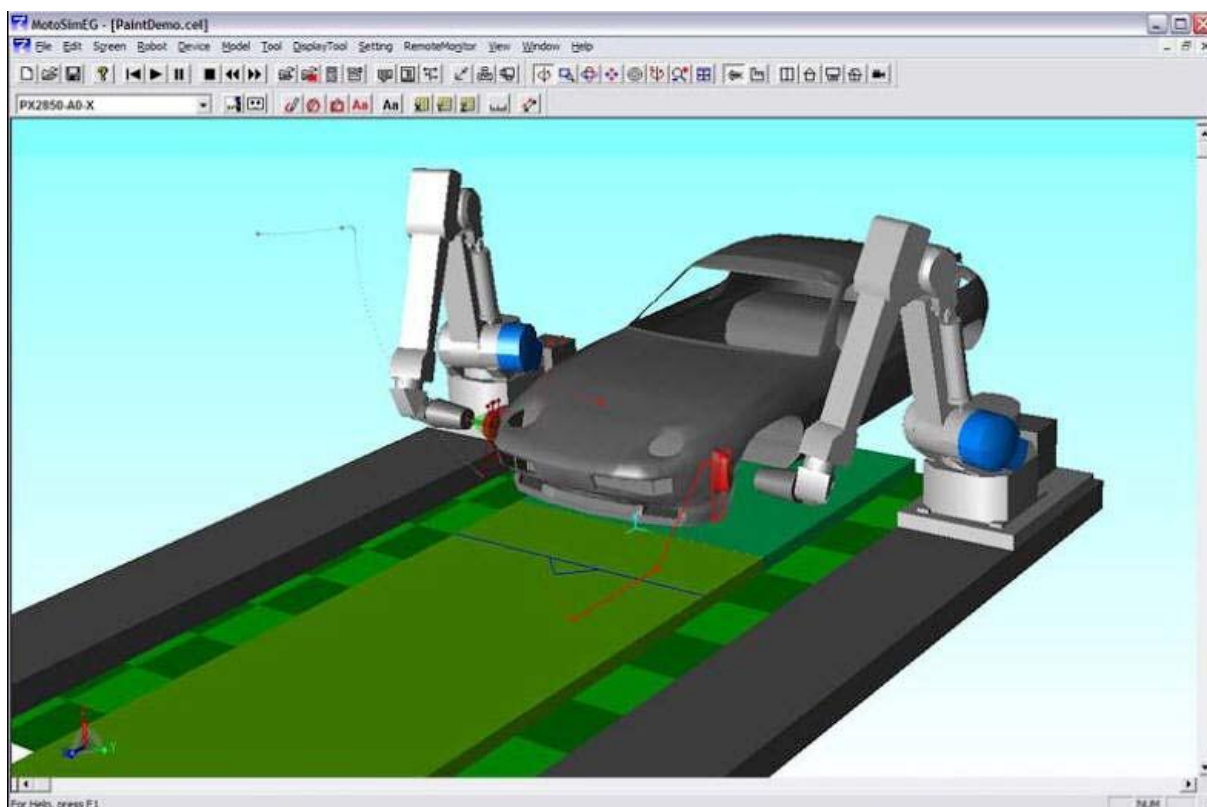
Fonte: ABB, 2013

3.2 MOTOMAN – MOTOSIM

Os softwares de desenvolvimento, da Yaskawa, são destinados à programação de equipamentos robóticos da própria empresa cujos nomes são: MotoSim EG e MotoSim EG-VRC. A principal diferença entre os modelos é o uso do Teach Pendant, no software VRC há um Pendant virtual, já no outro modelo não há o Pendant.

Neste software consegue-se programar e simular manipuladores e posicionadores, conjuntamente com entradas e saídas digitais (IO digitais) e também importar sistemas ou máquinas oriundos de outros softwares CAD. Na Figura 19 tem-se a visualização do software MotoSim EG com dois manipuladores e outros componentes de uma célula de solda automotiva. (Motoman, 2013).

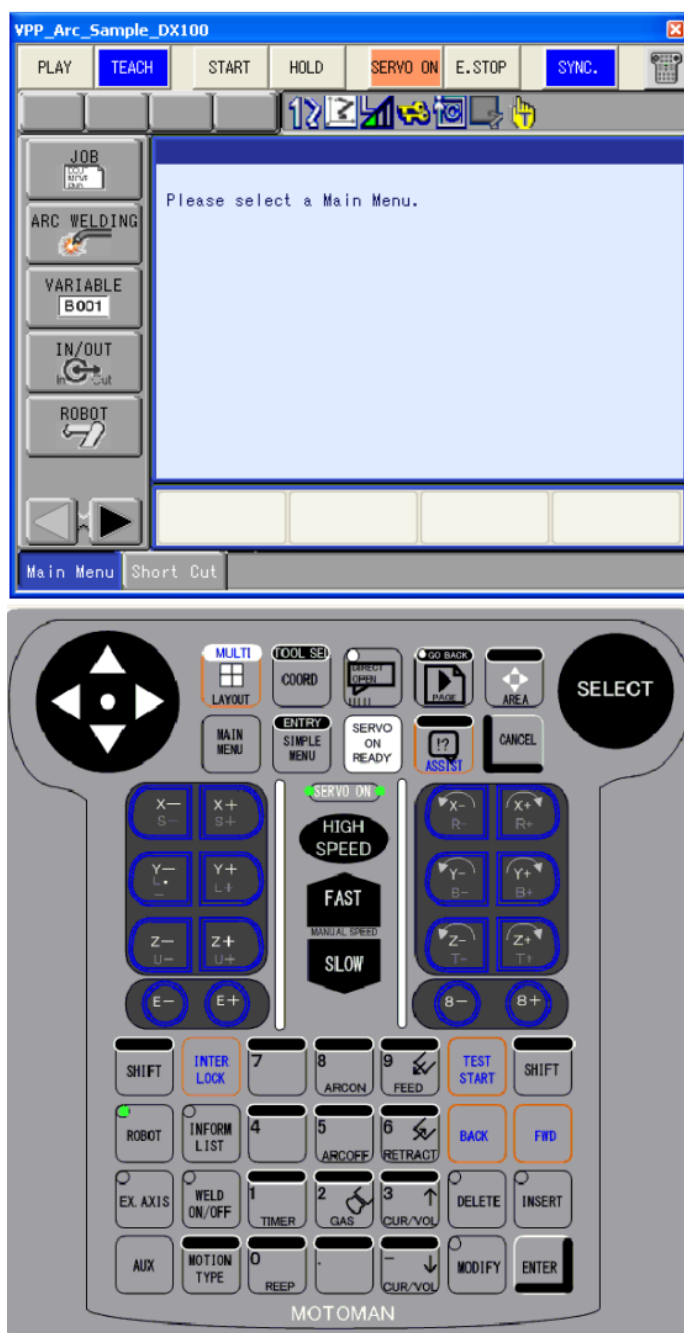
Figura 19 - Visualização do software MotoSim EG



Fonte: Motoman, 2008

Na Figura 20 tem-se o TeachPendant que é utilizado na versão VRC do software da Motoman. Observa-se que há uma quantidade de botões muito grande ao comparar-se com o pendant da ABB, a operação deste pendant é mais complexa e exige uma experiência ou explicações prévias para obter sucesso na simulação.

Figura 20 - TeachPendant virtual do software MotoSim EG-VRC



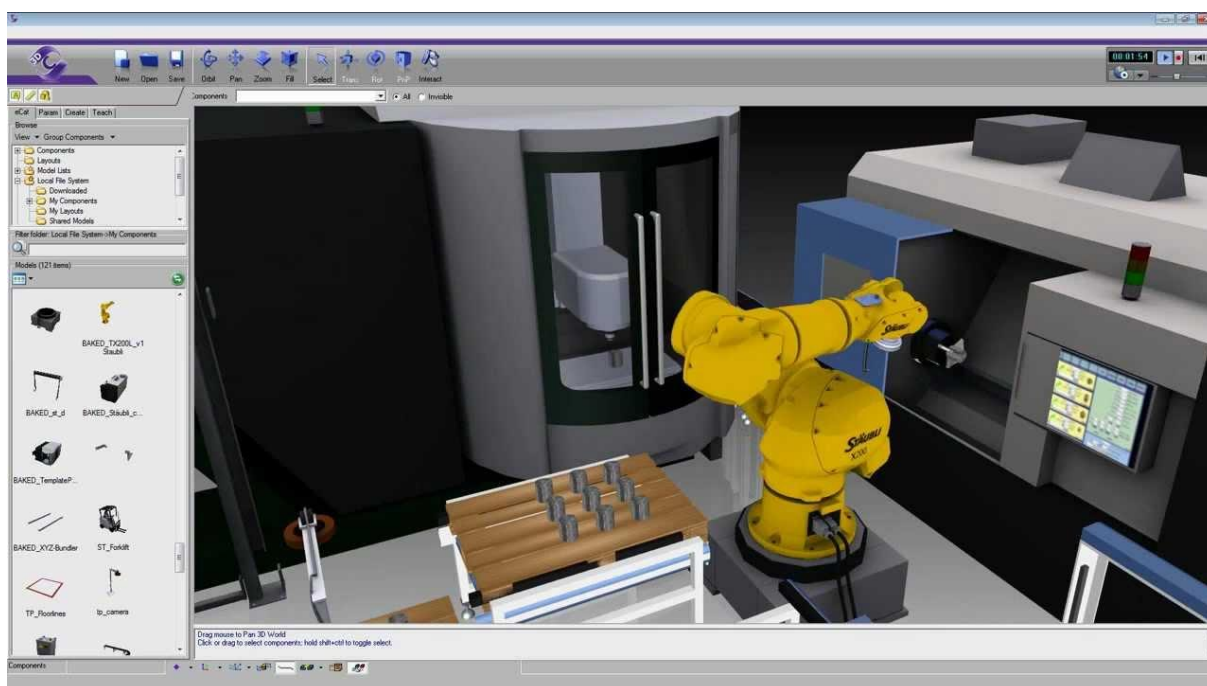
Fonte: Motoman, 2008

3.3 STÄUBLI – STÄUBLI ROBOTICS SUITE

O software de desenvolvimento da Stäubli possui praticamente todas as funções que os softwares anteriores, mas com a diferença que este software é

subdividido. Inicialmente tem-se um menu principal, cujo nome é Stäubli Robotics Suite, e após aberto esse menu pode optar-se por editar a célula de trabalho, editar o software do robô, emular o sistema, visualizar os movimentos do manipulador, transferir os programas do computador para o controlador e vice-versa e ainda há a opção de visualizar e controlar o robô de modo remoto. A Figura 21 ilustra um exemplo de uma célula desenvolvida no pacote da Stäubli. (STÄUBLI, 2010).

Figura 21 – Software de visualização do Robotics Suite

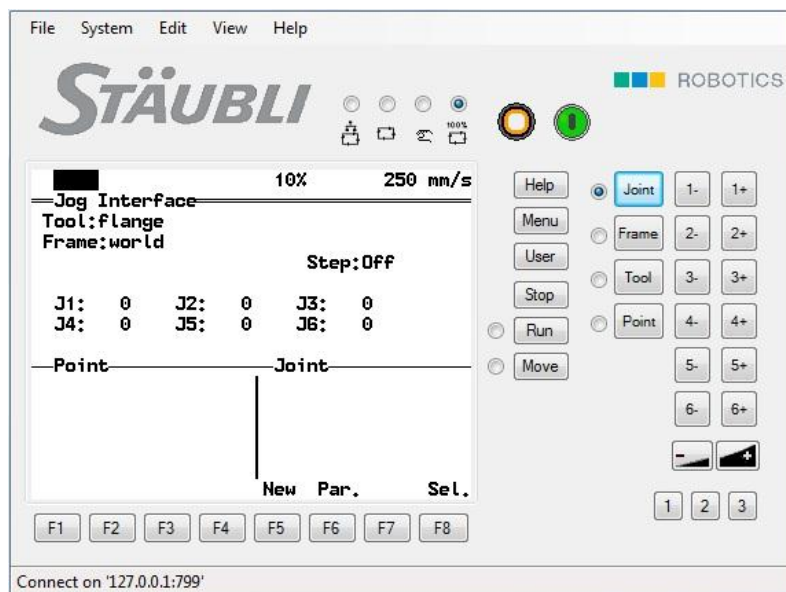


Fonte: Stäubli, 2010

O método projetado para a utilização do software da Stäubli, com um menu principal, é mais organizado e de simples visualização, sendo mais fácil para o usuário iniciante a sua utilização. Este software quando não licenciado pode executar várias funções off-line, basicamente simular a operação do robô juntamente com o seu teach pendant.

Na Figura 22 visualiza-se o TeachPendant virtual da Stäubli, a sua interface é prática e também é intuitiva, observa-se que no lado direito são os botões utilizados para movimentar o robô virtual.

Figura 22 - TeachPendant virtual do Robotics Suite da Stäubli



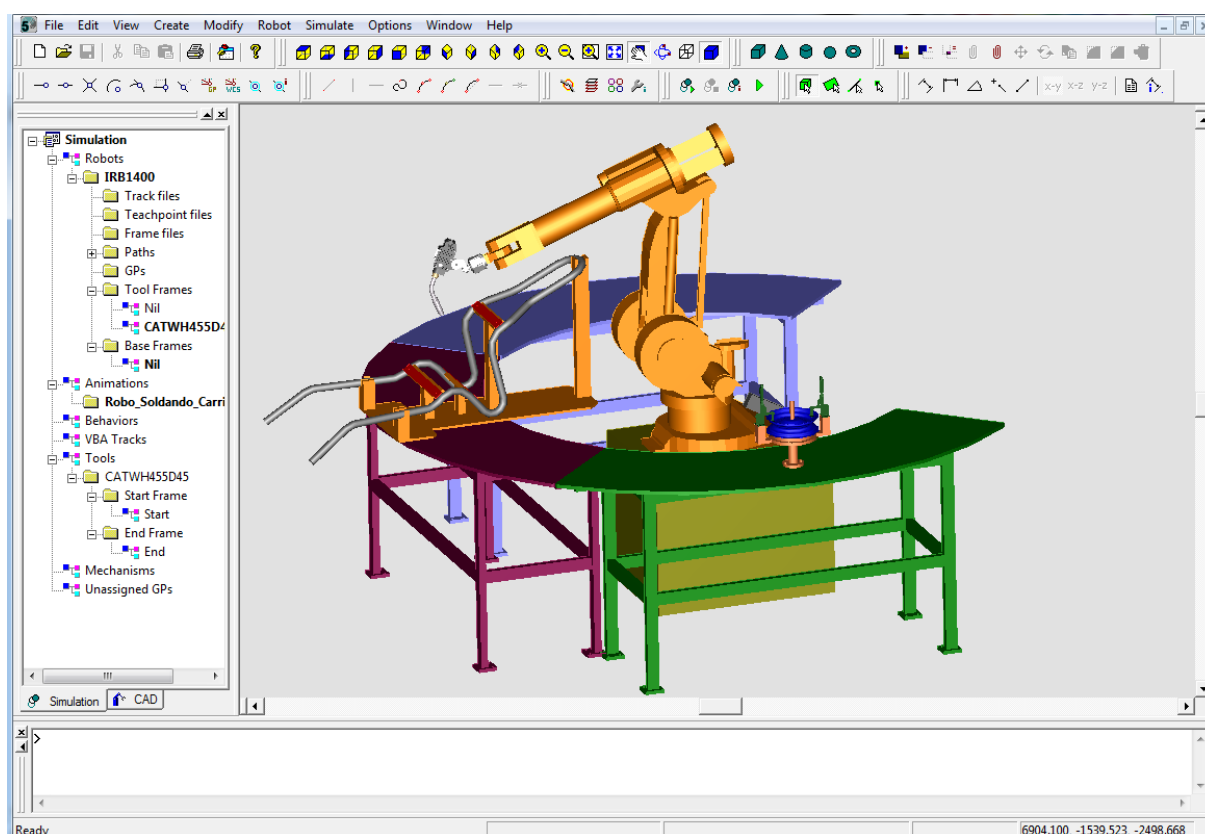
Fonte: Stäubli, 2010

3.4 WORKSPACE

O software WorkSpace da empresa WAT Solutions, difere dos citados anteriormente em três aspectos. São eles: o software não foi desenvolvido por uma empresa fabricante de robôs; pode-se programá-lo em várias linguagens de programação; e o software é para uso exclusivo de programação off-line. Esse software é oferecido com duas configurações, uma versão profissional e uma versão educacional. Ambas as versões possuem um ambiente gráfico sofisticado, com possibilidade de adicionar vários manipuladores, importar máquinas de outros softwares CAD e importar ferramentas. As versões também possuem a ferramenta de poder programar um determinado robô com uma linguagem diferente da linguagem nativa, e se o usuário desejar pode convertê-la.

A Figura 23 ilustra uma pequena célula de solda desenvolvida no software WorkSpace. Nela observa-se um manipulador, uma ferramenta acoplada no flange e uma bancada onde está fixada a peça a ser soldada. Neste software também é possível a integração entre robôs usando sinais elétricos simulados e criar dispositivos para auxiliar na simulação.

Figura 23 - Visualização do software WorkSpace

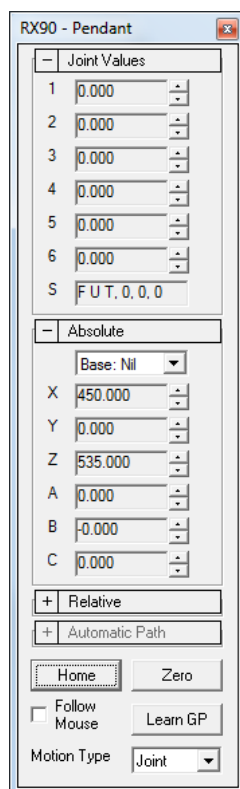


Fonte: Autor

O software possui as duas versões citadas anteriormente e também possui uma versão demo, na qual se consegue criar, importar, programar e simular uma célula, só não é possível salvar o arquivo durante o uso. Este software do ponto de vista acadêmico é excelente, pois permite trabalhar com várias linguagens de programação e ainda há o recurso de poder criar interfaces diversas com o auxílio do Visual Basic, que está incluso no pacote. Já do ponto de vista industrial, é útil se na indústria há vários robôs de diferentes fabricantes, tornando inviável a aquisição dos softwares de cada fabricante.

O TeachPendant do software da WorkSpace é visualizado na Figura 24. Nota-se que este pendant é mais simples que os anteriores tanto na questão visual quanto na questão operacional, com este pendant o usuário pode movimentar o robô virtual e marcar pontos. Estes aspectos são assim definidos porque o pendant deve atender uma gama grande de fabricantes e, conseqüentemente, funcionar com todos, há algumas versões e atualizações disponíveis que simulam o pendant do próprio fabricante.

Figura 24 - TeachPendant virtual do software WorkSpace



Fonte: Autor

3.5 CONSIDERAÇÕES

Os softwares de desenvolvimento abordados nos itens anteriores possuem características muito semelhantes entre si, por exemplo: qualidade gráfica, importação de objetos CAD, programação off-line, simulação de movimentos, integração entre equipamentos distintos e simulação de sinais digitais e analógicos. Há outro ponto em comum: a dificuldade no uso; os próprios fabricantes disponibilizam cursos de operação e uso dos seus softwares em virtude da quantidade de opções e configurações disponíveis. Mas o software definido para o uso neste projeto, WorkSpace, é o mais simples em termos de operação, atende vários fabricantes de robôs e também possui uma integração com outros softwares mais amigável através de macros em Visual Basic.

4 MOUSE 3D

O mouse 3D é um periférico de computador desenvolvido essencialmente para ser utilizado em softwares que trabalham com ambientes tridimensionais, nesses ambientes podem ser desenvolvidos, por exemplo: uma máquina, um edifício, uma animação, etc. Mas geralmente a função do mouse 3D é somente para alterar a visualização dos objetos dentro do ambiente tridimensional, ou seja, quando deseja-se aproximar, rotacionar ou deslocar o objeto na tela o mouse poupa tempo e aumenta a produtividade do usuário. (3DCONNEXION, 2013)

A Figura 25 ilustra a imagem de um mouse 3D. Nele observa-se algumas diferenças para o mouse comum, a principal é que não há deslocamento no plano, por exemplo uma mesa comum, a outra diferença é que o mouse comum trabalha com dois eixos, usualmente X e Y, e o mouse 3D trabalha com seis eixos: três de posição e três de rotação.

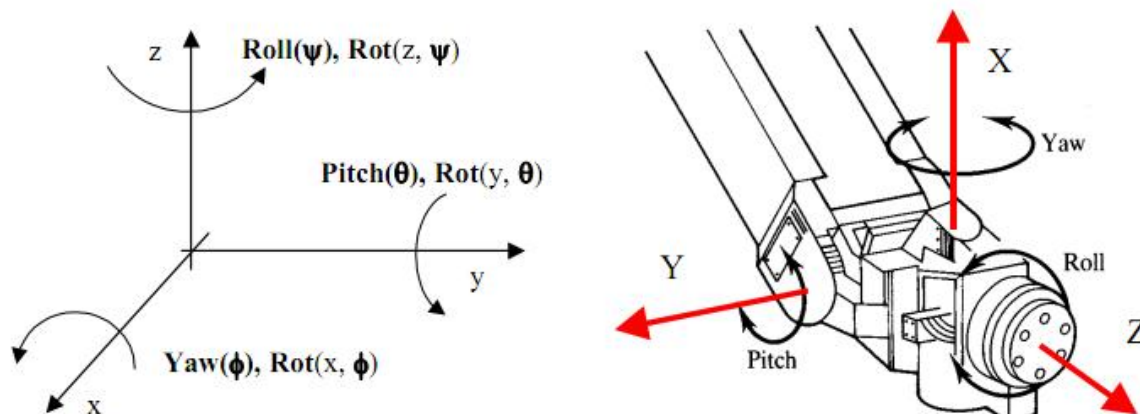
Figura 25 - Mouse 3D da empresa 3DCONNEXION



Fonte: 3DCONNEXION, 2013

Na Figura 26 encontram-se os eixos do mouse 3D. Basicamente há três eixos que definem o deslocamento, são eles: X, Y e Z. E também há os que definem a rotação, são eles: θ_x , θ_y e θ_z . Em um mouse comum só há as componentes X e Y por exemplo. Este mouse também é conhecido como mouse de seis graus de liberdade, pois pode-se operar em qualquer um dos eixos e também é possível trabalhar com um ponto qualquer no espaço, ou seja, utilizar eixos combinados.

Figura 26 - Representação dos eixos do mouse 3D



Fonte: SANTOS, 2004

A aplicação usual do mouse 3D, como citado anteriormente, é para visualização de objetos tridimensionais. Na Figura 27 demonstra-se a ação aplicada no mouse e sua consequência em um software 3D, as ações executadas na primeira linha da Figura são as de deslocamento, com isso a visualização do sólido seria deslocada, nas ações da segunda linha o objeto seria rotacionado conforme ilustrado.

Figura 27 - Aplicação típica de um mouse 3D



Fonte: 3DCONNEXION, 2013

O mouse 3D oferece uma quantidade enorme de possibilidades de uso e o próprio fabricante disponibiliza ferramentas de desenvolvimento para aumentar a gama de aplicações. Também há fóruns para que usuários e programadores troquem informações sobre aplicações.

5 METODOLOGIA

5.1 VISÃO GERAL

O sistema desenvolvido deverá ser capaz de interpretar os sinais oriundos do mouse 3D e movimentar o manipulador virtual conforme configurado, ou seja, quando o usuário movimentar o eixo X+ do mouse com uma certa intensidade o robô deverá ser movimentado de igual maneira. A Figura 28 demonstra o fluxograma básico de funcionamento.

Figura 28 - Fluxograma da visão geral de funcionamento do projeto



Fonte: Autor

No fluxograma supracitado utiliza-se o conceito de eventos. Nesse caso um evento caracteriza-se pela alteração da posição do mouse 3D, por isso o fluxograma só será executado quando o usuário utilizar o mouse 3D. Observa-se que há dois sub-processos, o primeiro tem a utilidade de verificar o tipo de movimento requerido pelo usuário e o segundo tem a função de executar o movimento desejado no manipulador em uso.

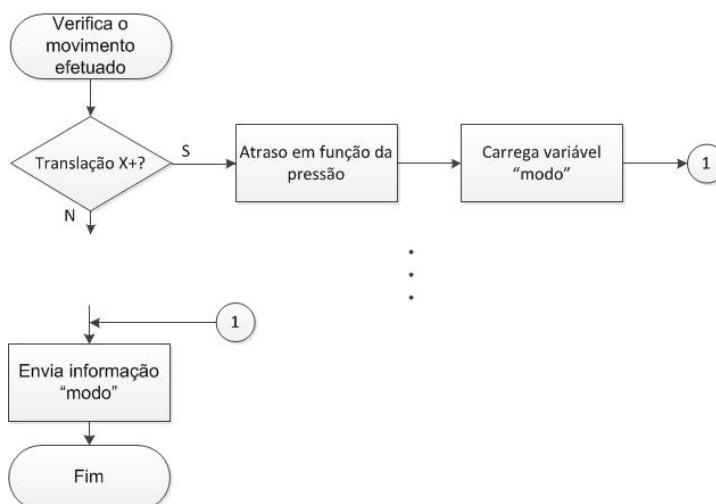
5.2 CAPTURA DOS SINAIS DO MOUSE 3D

A captura dos sinais do Mouse 3D é essencial para o desenvolvimento do trabalho, nesta etapa será verificado o movimento executado pelo usuário e assim definir o movimento e a intensidade do mesmo. Para executar essa tarefa o próprio fabricante disponibiliza uma biblioteca na qual é possível ler as informações oriundas do mouse 3D. Esta biblioteca está no arquivo “TDx.TDxInput.dll”.

O programa utilizado para ler e tratar as informações do mouse 3D é o Visual C# Express 2008, que é um software gratuito e com documentação disponível. (MICROSOFT, 2013b).

Conforme citado anteriormente, essa etapa é um sub-processo de um fluxograma principal. Na Figura 29 é exibido o fluxograma parcial deste sub-processo. Observa-se que após ocorrer a ação do usuário são executadas tomadas de decisão para cada tipo de movimento. Também há um atraso e após e enviado a informação do tipo de movimento através da variável “modo”.

Figura 29 - Fluxograma da interpretação da ação do usuário no mouse 3D



Fonte: Autor

Na verificação do tipo de movimento e força aplicada, é necessário atentar para três variáveis. A primeira variável é a *translation*, esta indica qual eixo sofreu uma translação. Para o caso de uma rotação, a variável *rotation* indica o sentido e sobre qual eixo ocorreu a rotação. A terceira variável é a força aplicada pelo usuário. Ela varia de 1 a 200 e a unidade destes valores é de força.

O atraso é fundamental no projeto desenvolvido, pois o mesmo dará a velocidade de movimentação do robô, ele é realizado nessa etapa porque não é possível de ser executado diretamente na etapa de movimentação do robô. O Quadro 1 demonstra uma parte do software utilizado.

Quadro 1 - Fração do programa de interpretação da ação do usuário no mouse 3D

```

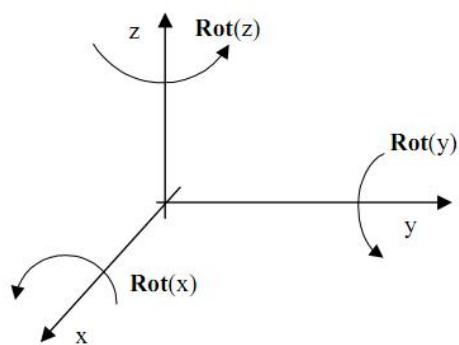
if (translation.X > 1)
{
    tempo =Convert.ToInt16(210 - Math.Abs(translation.X)) * 3;
    Thread.Sleep(tempo);
    modo = "X+";
}
else if (rotation.X == 1)
{
    tempo =Convert.ToInt16(210 - Math.Abs(rotation.Angle)) * 3;
    Thread.Sleep(tempo);
    modo="rX+";
}
.....
Envia_dado();

```

Fonte: Autor

O sistema de coordenadas utilizado é visualizado na Figura 30, nesta imagem à esquerda está o mouse 3D e a direita os eixos.

Figura 30 - Sistema de coordenadas do mouse 3D



Fonte: Autor

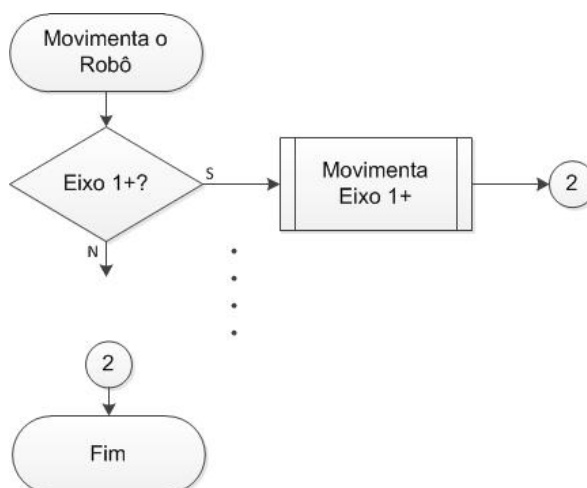
O método utilizado para envio dos dados é feito por meio de uma porta serial virtual. Isto se deve ao fato do software do robô apresentar alguns problemas ao utilizar meios mais refinados para esta troca de dados.

5.3 SOFTWARE DE DESENVOLVIMENTO

O software utilizado para simulação e movimentação do robô virtual será o WorkSpace 5. Este software possui uma ferramenta que abre muitas possibilidades para o usuário, que é o Visual Basic, que já está incorporado no programa. Através desta ferramenta o robô será movimentado e também ocorrerá a interface entre software e usuário.

No capítulo anterior fora visto um sub-processo do fluxograma principal, que era a interpretação e envio do sinal do mouse 3D, no presente capítulo o robô será efetivamente movimentado. Para executar esta tarefa o fluxograma parcial da Figura 31 é apresentado, nele visualiza-se as tomadas de decisão dos dados recebidos do tratamento do mouse 3D, e após há o sub-processo correspondente a cada eixo movimentado pelo usuário.

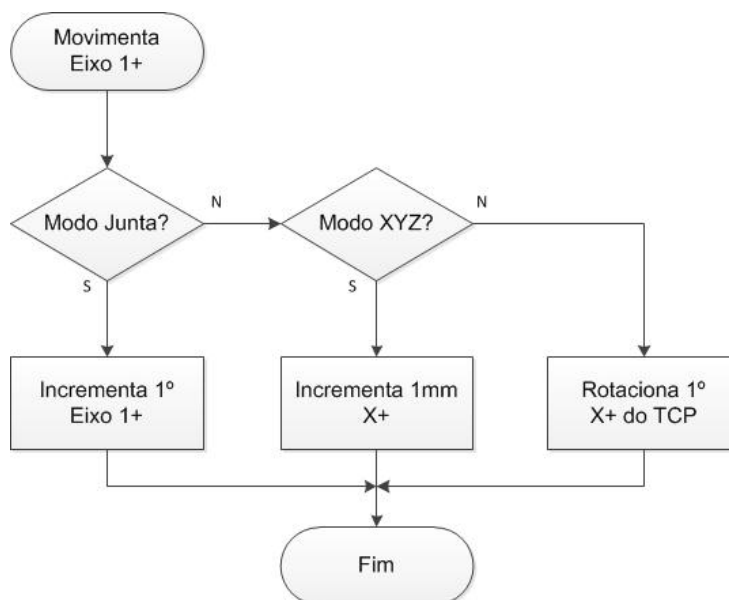
Figura 31 - Fluxograma parcial do recebimento dos movimentos executados



Fonte: Autor

Na Figura 32 apresenta-se o fluxograma que contém a ação de movimentar o eixo específico quando solicitado pelo usuário. Nas próximas subseções serão abordados cada item e visualizados o seu procedimento.

Figura 32 - Fluxograma parcial da ação que o manipulador executará



Fonte: Autor

5.3.1 Tratamento dos Sinais do Mouse

Os sinais oriundos do mouse 3D são tratados pelo software desenvolvido em Visual C# e após enviados via porta serial virtual, conforme citado anteriormente, a partir deste ponto é necessário receber esse sinal e definir o eixo selecionado. Para executar essas funções é necessária a utilização de macros em Visual Basic, o software Visual Basic, da mesma maneira que o Visual C# trabalha com eventos, ou seja, o software fica inativo até algo acontecer, por exemplo: o usuário clicar em um botão, ser recebido um dado pela porta serial ou mesmo pressionar alguma tecla, essa técnica facilita a programação. Conforme o fluxograma da Figura 28 e o funcionamento do software, quando recebido um dado pela porta serial um evento será chamado, que no caso do fluxograma é “Movimenta o Robô”, neste evento será verificado esse dado e após é chamada a sub-rotina pertinente. O Quadro 2 exhibe uma parte essencial deste software. (MICROSOFT, 2013a)

Quadro 2 - Trecho do programa que define a ação que o robô executará

```

Private Sub MSComm1_OnComm()

    Dim direcao As String

    direcao = MSComm1.Input

    TextBox1.Text = "Direção = " & direcao

    If direcao = "X+" Then
        If (Modo = 1) Or (Modo = 4) Then
            but_X_mais_Click
        ElseIf Modo = 3 Then
            Modo = 1
            but_X_mais_Click
            Modo = 3
        End If

    ElseIf direcao = "X-" Then
        If (Modo = 1) Or (Modo = 4) Then
            but_X_menos_Click
        ElseIf Modo = 3 Then
            Modo = 1
            but_X_menos_Click
            Modo = 3
        End If
    ElseIf direção = "rX+" Then
        .....
    End Sub

```

` Repete para todos
` os movimentos

Fonte: Autor

No trecho do software supracitado é exibido o evento que recebe o dado via porta serial através do procedimento “MSComm1_OnComm()”, após é carregado o dado do buffer da porta serial na variável “direção”, também é exibido este valor na tela principal e depois são efetuadas as decisões com a função “If”. Dois detalhes importantes são visualizados dentro da estrutura “If”. O primeiro é o uso de outras tomadas de decisão para verificar o modo de operação, e o outro é a chamada da sub-rotina específica de movimento, por exemplo “but_X_mais_Click”, ambos serão melhor abordados na sequência.

5.3.2 Movimentos de Junta

Os movimentos de junta são definidos, conforme citado anteriormente, em movimentos independentes de uma junta ou eixo. Executados de modo que se some ou subtraia um valor de ângulo. Para realizar esta ação é utilizado o código do Quadro 3. Nele observa-se que os movimentos de junta são bem simples de ser implementados. O código executa apenas a soma ou subtração de um valor de ângulo, em graus, e após enviar o manipulador para a nova posição.

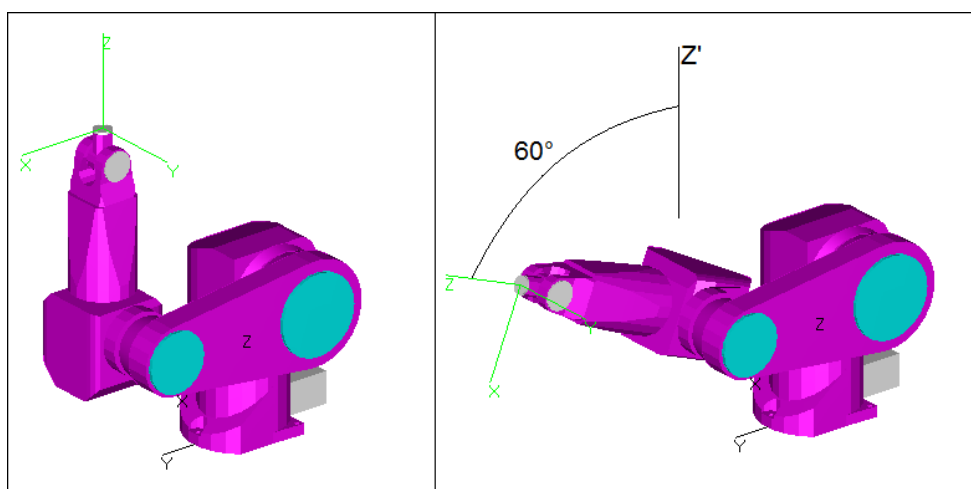
Quadro 3 - Trecho do programa que executa o movimento do tipo junta

```
Joint_Default(0) = Joint_Default(0) + velocidade           \ Eixo 1
SetImmediateJointPosition rRobot, Joint_Default
```

Fonte: Autor

Na Figura 33 observa-se o resultado de um teste realizado com o código supracitado. Neste teste o trecho de código é executado 60 vezes, o que corresponde a 60° de alteração no eixo 3, o manipulador do lado esquerdo está na posição inicial e o do lado direito está com o eixo 3 avançado em 60°. É importante observar que somente o eixo 3 foi modificado, caracterizando um movimento de junta.

Figura 33 - Movimento teste executado do tipo junta



Fonte: Autor

5.3.3 Movimentos Lineares

Os movimentos lineares são caracterizados por variações retilíneas ao longo de algum eixo específico ou sobre uma reta no plano. A diferença entre esse tipo de movimento e o de junta é que o deslocamento do TCP ocorre em uma reta. Outra diferença é que a orientação dos eixos do TCP não alteram-se, e sim a sua posição no plano, isto pode ser visualizado na Figura 11.

Para executar esses movimentos no Workspace é necessário uma série de comandos executados em Visual Basic, para ilustrar o código implementado o Quadro 4 será utilizado. Na primeira parte do código altera-se o valor de algum eixo. No caso deste programa o eixo X é alterado, somando a ele o valor de 2mm. Os outros valores permanecem iguais. Após os novos valores a definição do novo ponto é executada através da função “SetCartesianTeachpoint” e “MoveTo”, a partir deste ponto é realizada uma operação para calcular os valores de ângulo de cada junta, para executar tal função o comando principal é o “RCSJointPosToWsJointPos” depois de calculado os valores das juntas o comando “SetImmediateJointPosition” posiciona o robô na posição desejada.

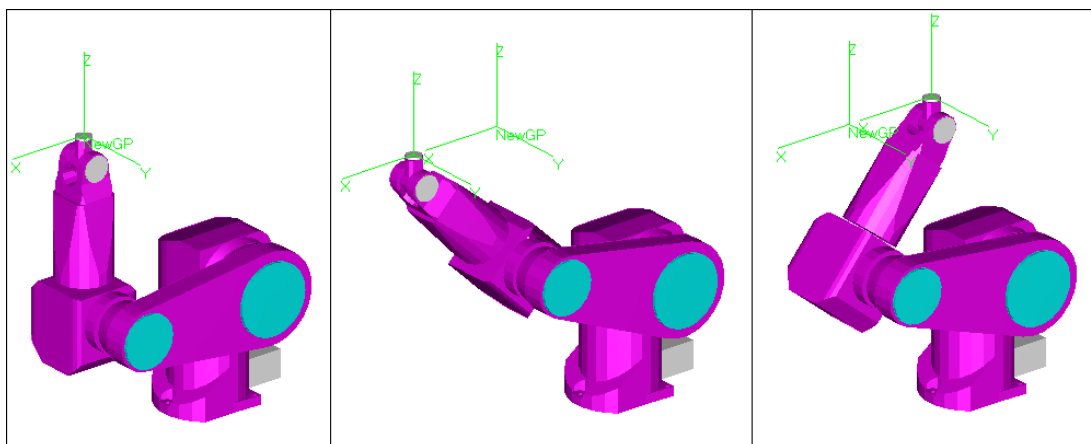
Quadro 4 - Fração do programa que executa o movimento linear

```
x = Pos_ant(0) + velocidade
y = Pos_ant(1)
z = Pos_ant(2)
a = Pos_ant(3)
b = Pos_ant(4)
c = Pos_ant(5)
SetCartesianTeachpoint rRobot, TP1, x, y, z, a, b, c, "LNDB"
MoveTo rRobot, TP1
  With TP1
    RCSJointPosToWsJointPos rRobot, rRobot.RRS.JointPos, .JointPos
    .IsJointTeachpoint = True
  End With
  For j = 0 To rRobot.NumMainAxes - 1
    Joint_Default(j) = Joint_Default_temp(j) + Joint_temp(j)
  Next j
SetImmediateJointPosition rRobot, Joint_Default
```

Fonte: Autor

Na Figura 34 apresenta-se o resultado da aplicação do procedimento no eixo X. Na imagem da esquerda o manipulador está na posição inicial, na imagem do centro a rotina de software é executado incrementando o valor de X e na imagem da direita a rotina é executada decrementando o valor de X, pode-se observar que o TCP do robô percorre uma linha reta.

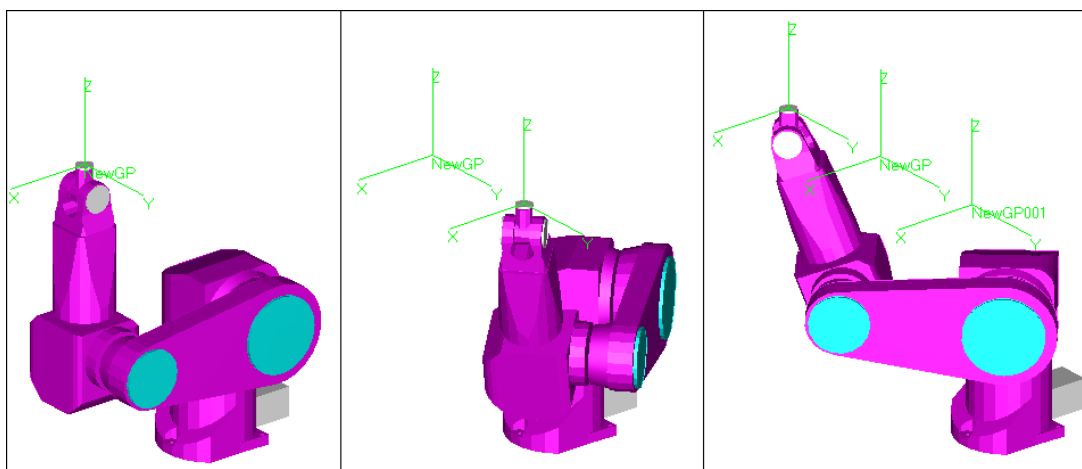
Figura 34 - Movimento teste executado do tipo linear



Fonte: Autor

Na Figura 35 apresenta-se o resultado da aplicação do procedimento no eixo Y. Na imagem da esquerda o manipulador está na posição inicial, na imagem do centro a rotina de software é executado incrementando o valor de Y e na imagem da direita a rotina é executada decrementando o valor de Y, pode-se observar que o TCP do robô está percorrendo uma linha reta.

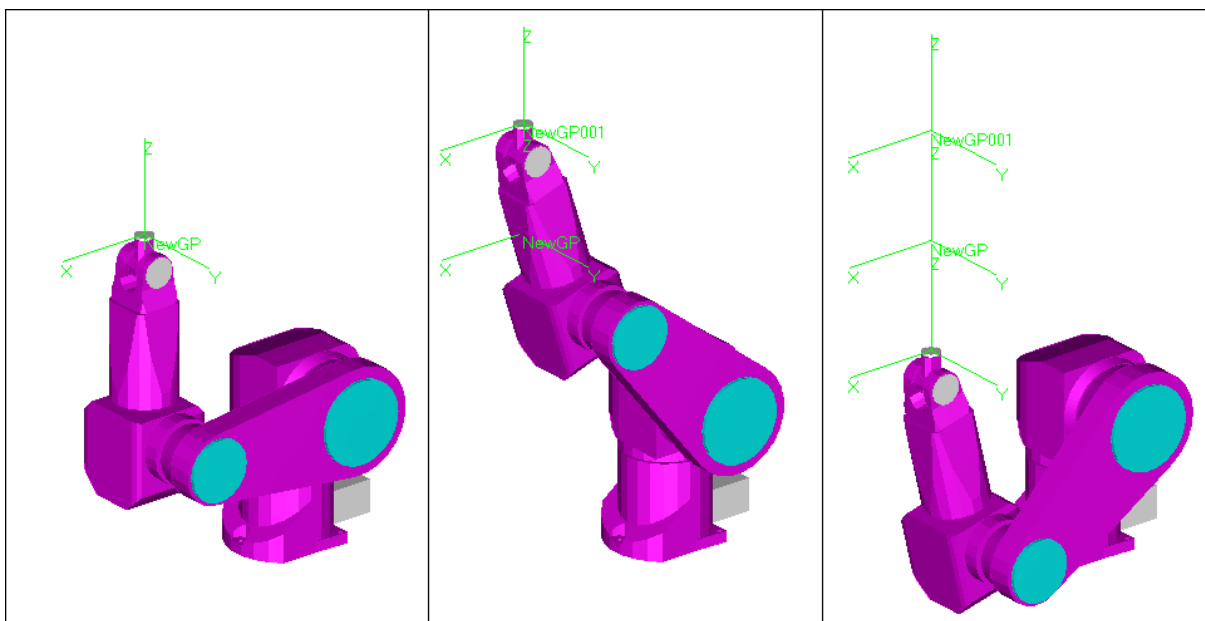
Figura 35 - Movimento Linear



Fonte: Autor

Na Figura 36 demonstra-se o teste realizado com o eixo Z, na imagem da esquerda o manipulador está na posição inicial, na imagem do centro a rotina de software é executado incrementando o valor de Z e na imagem da direita a rotina é executada decrementando o valor de Z, pode-se observar que o TCP do robô está percorrendo uma linha reta.

Figura 36 - Movimento Linear



Fonte: Autor

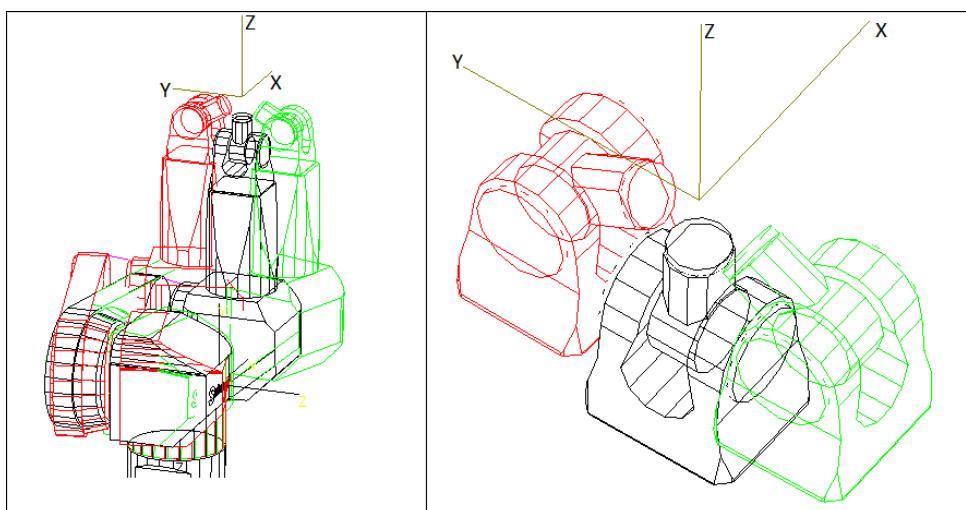
5.3.4 Movimentos Rotacionais

Os movimentos rotacionais são utilizados para rotacionar o manipulador em torno do TCP. Ou seja, diferentemente do movimento linear, no qual o interesse é deslocar a origem do TCP, este movimento é executado quando há a necessidade de alterar a orientação do TCP. Sendo que a sua posição no espaço não será alterada.

Para executar esse tipo de movimento o código utilizado é o mesmo que para o movimento linear só que ao invés de incrementar/ decrementar as variáveis X, Y e Z, são alteradas as variáveis A, B e C, essas variáveis correspondem a rotação em torno do eixo X, Y e Z, respectivamente.

Na Figura 37 demonstra-se o teste realizado com a rotação A, na imagem da esquerda tem-se o robô em três posições distintas e sobrepostas com ângulos de rotação diferentes. Na imagem da direita tem-se a aproximação para melhor observar a rotação em torno do eixo X. Nota-se que o TCP foi deslocado na direção Z para melhorar a visualização do movimento.

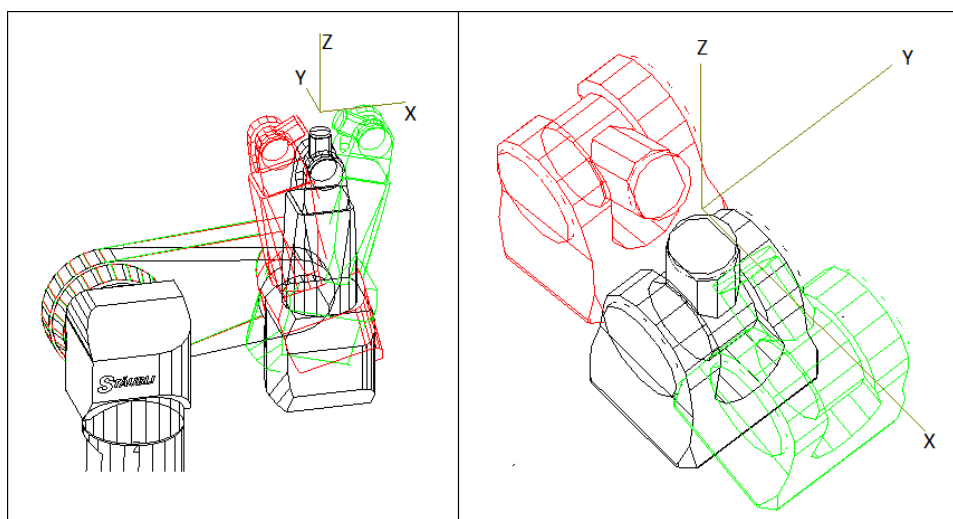
Figura 37 - Movimento teste executado do tipo rotacional A



Fonte: Autor

Na Figura 38 demonstra-se o teste realizado com a rotação B, na imagem da esquerda tem-se o robô em três posições distintas e sobrepostas com ângulos de rotação diferentes. Na imagem da direita tem-se a aproximação para melhor observar a rotação em torno do eixo Y. Nota-se que o TCP foi deslocado na direção Z para melhorar a visualização do movimento.

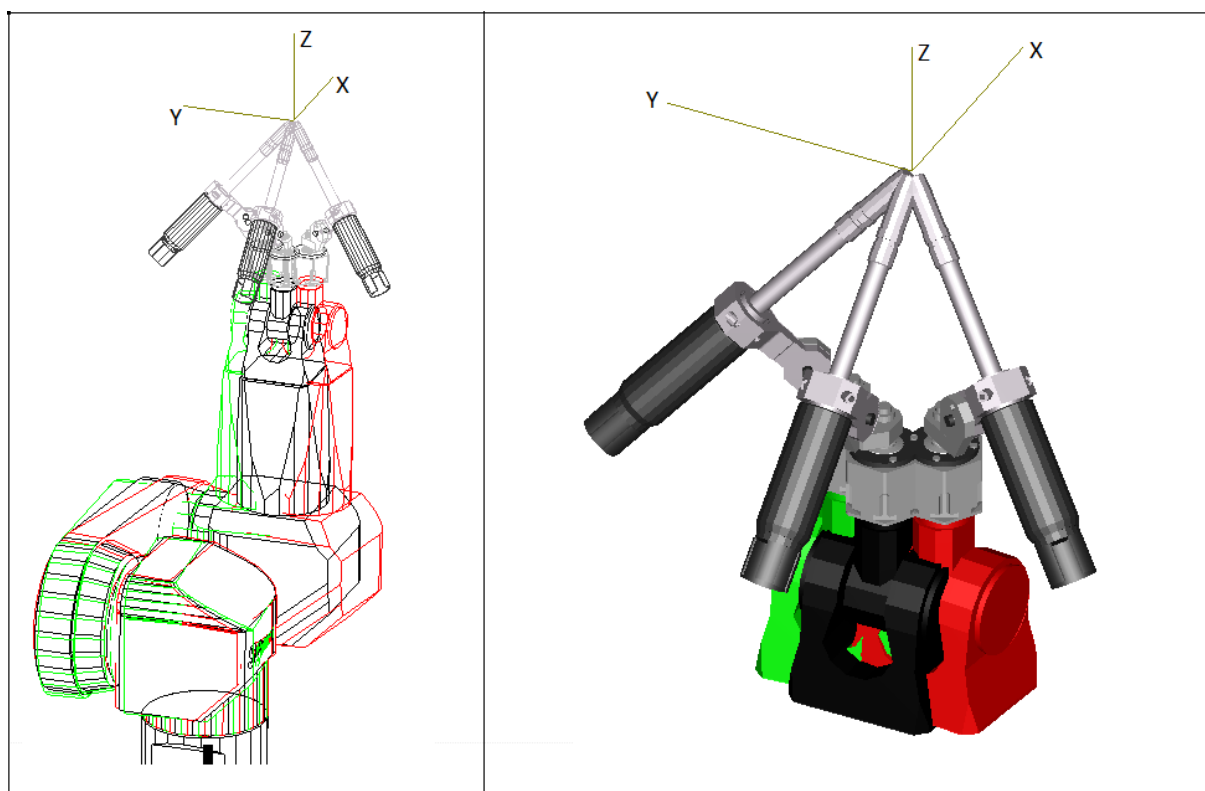
Figura 38 - Movimento teste executado do tipo rotacional B



Fonte: Autor

Na Figura 39 demonstra-se o teste realizado com a rotação C, na imagem da esquerda tem-se o robô em três posições distintas e sobrepostas com ângulos de rotação diferentes. Na imagem da direita tem-se a aproximação para melhor observar a rotação em torno do eixo Z. Nota-se que neste teste uma ferramenta é acoplada no manipulador para visualizar melhor a rotação em Z.

Figura 39 - Movimento teste executado do tipo rotacional C



Fonte: Autor

5.3.5 Painel de Operação

O Painel de Operação é necessário para o usuário selecionar o tipo de movimento a ser executado. Se irá movimentar o manipulador no modo manual, ou seja, clicando, ou se irá utilizá-lo através do mouse 3D. O leiaute deste painel foi pensado de modo a tornar a operação simples e intuitiva, limitando a quantidade de botões disponíveis e exibindo somente informações necessárias para a movimentação.

A Figura 40 exibe a tela inicial quando o usuário inicia o sistema, nesta tela observa-se na parte superior uma tela, onde o software informa o que fazer, quando não inicializado o modo, e também informa a direção do movimento executado pelo usuário no mouse 3D. Na região central encontra-se o Modo de Operação, nela o usuário define o tipo de movimento a ser executado e se utilizará ou não o mouse 3D. No lado direito estão os botões para inicializar e finalizar os movimentos e finalmente, na parte inferior estão os botões para operação em manual, no caso desta Figura como não há nenhum modo selecionado também não aparece nenhum botão.

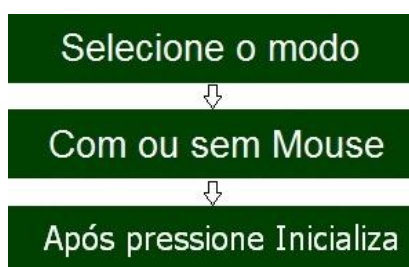
Figura 40 - Painel de Operação desenvolvido



Fonte: Autor

Na tela de visualização enquanto o usuário não inicializa o modo de operação, uma sequência de imagens fica sendo exibida com as possíveis ações que o mesmo poderá executar, a Figura 41 exibe esta sequência, após a última imagem a sequência é reiniciada.

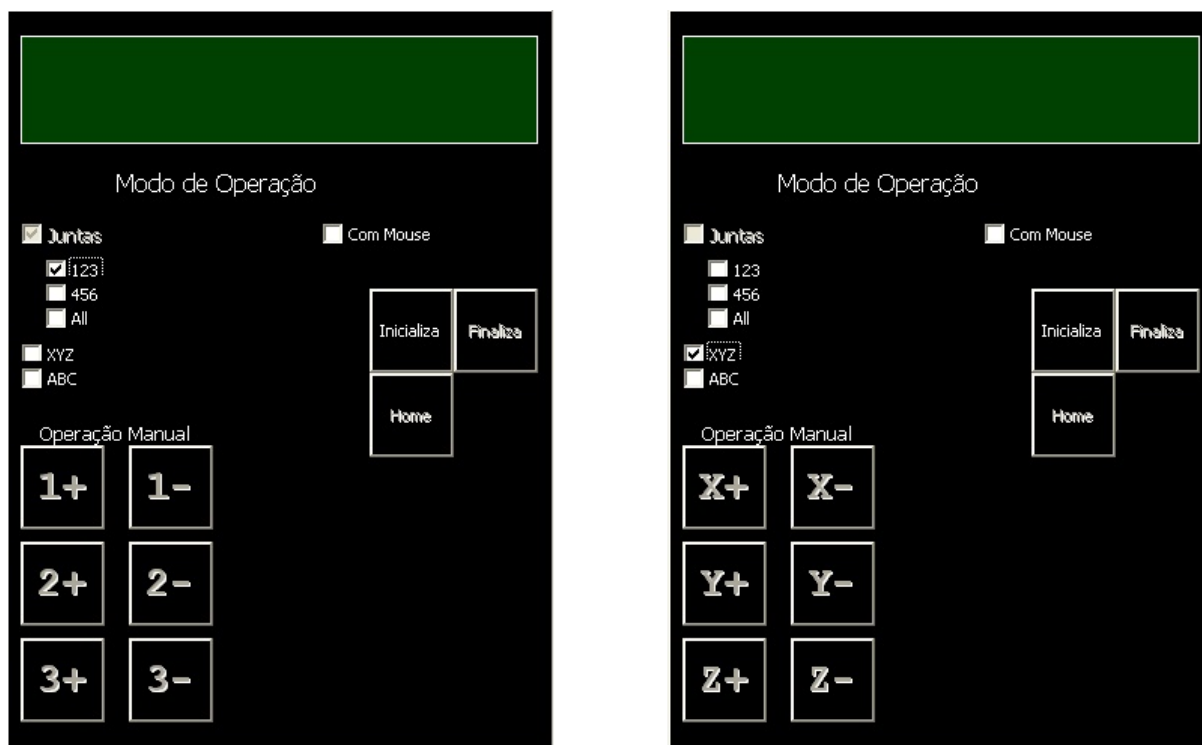
Figura 41 - Sequência exibida no painel de operação



Fonte: Autor

Após o usuário selecionar o modo de operação, serão exibidos na parte inferior os eixos correspondentes para a operação manual. Um detalhe é que os botões estarão visíveis, mas não operacionais, já que somente quando inicializado o modo, o usuário poderá movimentar o manipulador. A Figura 42 ilustra o que foi citado anteriormente, do lado esquerdo o modo Junta eixos 1, 2 e 3 foi selecionado e os botões disponíveis correspondem ao incremento ou decremento de cada eixo, já na imagem do lado direito o modo selecionado foi XYZ e desta vez os botões exibidos são correspondentes aos eixos X, Y e Z.

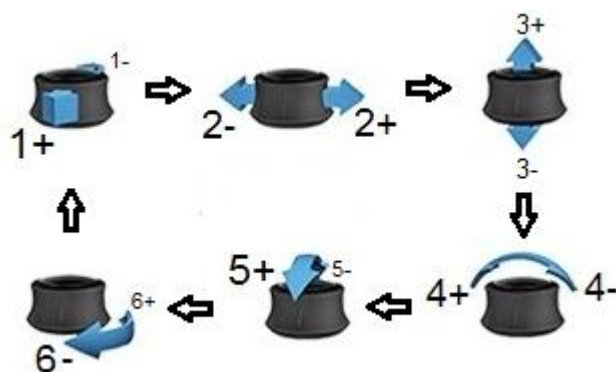
Figura 42 - Modos de Operação do painel de operação



Fonte: Autor

Quando o usuário seleciona a operação com o Mouse 3D é exibida também uma sequência de imagens correspondentes ao modo selecionado para que o mesmo saiba o efeito de sua ação no mouse, na Figura 43 é ilustrada a sequência de imagens quando o modo todos os eixos(All) é selecionado.

Figura 43 - Sequência de exibição para informar ao usuário os movimentos disponíveis



Fonte: Autor

Na Figura 44 exibe-se a sequência para o modo XYZ.

Figura 44 - Sequência de exibição para outro modo de operação



Fonte: Autor

A imagem do painel de operação, quando o mouse está ativo, é visualizada na Figura 45, neste exemplo o modo selecionado é o Junta, eixos 4, 5 e 6, e é observado que no centro da figura há a demonstração do movimento que o usuário deverá executar no mouse para que o robô altere o eixo 4.

Figura 45 - Exemplo do painel de operação



Fonte: Autor

Após o usuário selecionar o modo, optar pelo uso do mouse 3D e clicar no botão Inicializa, o movimento do robô será liberado, neste momento o painel de operação sofrerá algumas mudanças, são elas: os botões para operação manual serão habilitados e no visor serão exibidos os movimentos oriundos do mouse 3D, este último está relacionado com os eixos da Figura 30.

A Figura 46 demonstra as mudanças ocorridas quando a movimentação está permitida, o primeiro fato a ser observado é que o visor está exibindo a direção aplicada no mouse, que no caso é uma rotação em torno do eixo Y. O outro fato é que os botões estão habilitados.

Figura 46 - Exemplo do modo tipo junta



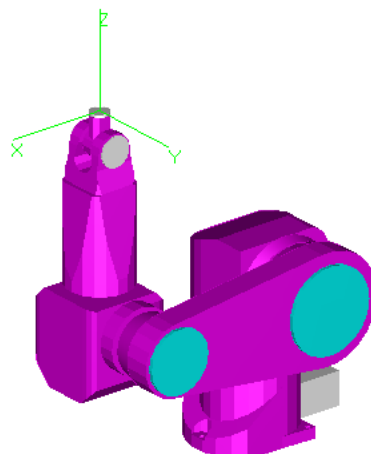
Fonte: Autor

5.3.6 Operação e Testes

Na etapa de operação e testes serão demonstrados os movimentos executados pelo robô conjuntamente com a visualização do painel de operação. Serão explicitados cada movimento a partir de uma posição fixa do manipulador, todos os movimentos executados pelo manipulador utilizarão o código implementado em Visual Basic e também será utilizado o mouse 3D, já que a operação manual deriva da operação com o mouse.

O primeiro teste será com o movimento tipo junta e serão executados testes com alguns eixos. Todos os resultados dos testes serão visualizados a partir de uma posição fixa do robô. A posição fixa é observada na Figura 47. Um detalhe importante nos testes é que o TCP do manipulador está centralizado no flange do mesmo e possui os eixos visíveis, outro fato é que os movimentos de junta e linear neste projeto não seguem o sistema de coordenadas do TCP e sim as coordenadas da base, mas o movimento rotacional utiliza o TCP como base para seus movimentos.

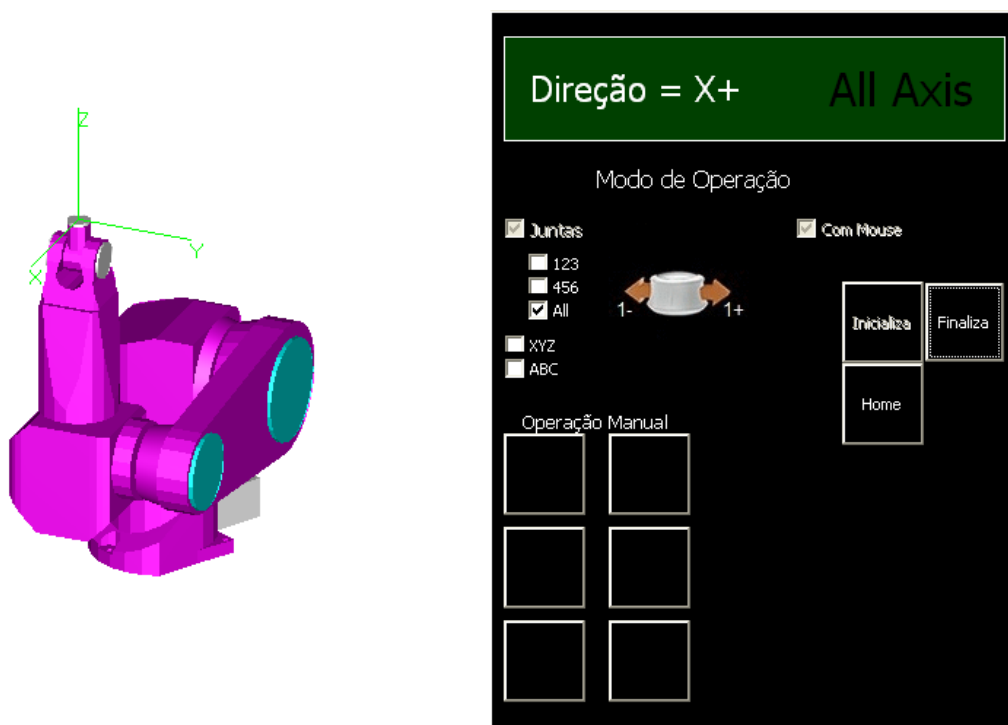
Figura 47 - Posição padrão do robô para os testes



Fonte: Autor

Definido o elemento padrão para os testes inicia-se movimentando o eixo 1, conforme visto na seção anterior ou observando o exemplo no painel de operação, a Figura 48 ilustra a ação executada pelo usuário no mouse 3D. No lado esquerdo visualiza-se o manipulador com o eixo 1 alterado em $+45^\circ$ e no lado direito pode-se observar que o sinal oriundo do mouse 3D é “X+” confirmando o funcionamento adequado.

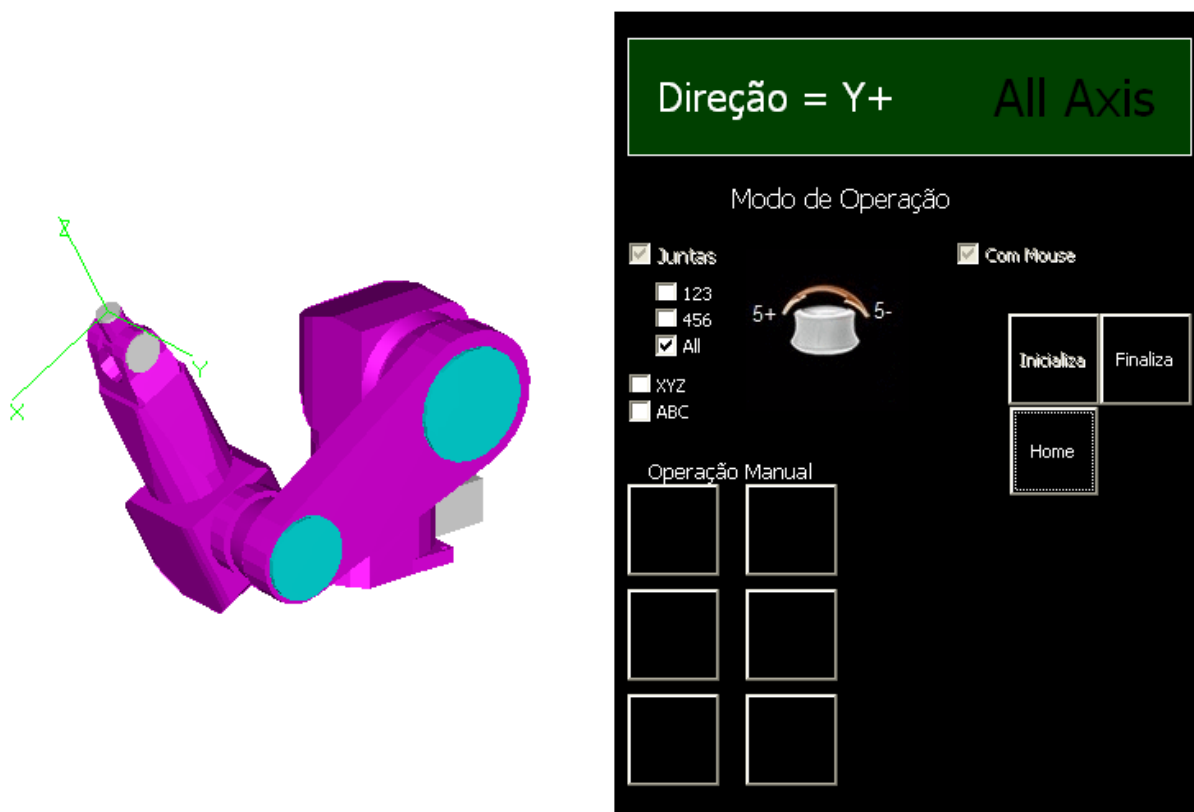
Figura 48 - Teste realizado na direção do eixo +1, modo junta



Fonte: Autor

Na Figura 49 exibe-se o teste realizado com o eixo 2. Neste caso o ângulo foi alterado em $+40^\circ$, pode-se observar que o sistema de coordenadas do TCP varia conforme se alteram os ângulos dos eixos, também nota-se que a direção é “Y+” caracterizando assim o incremento de ângulo do eixo 2 .

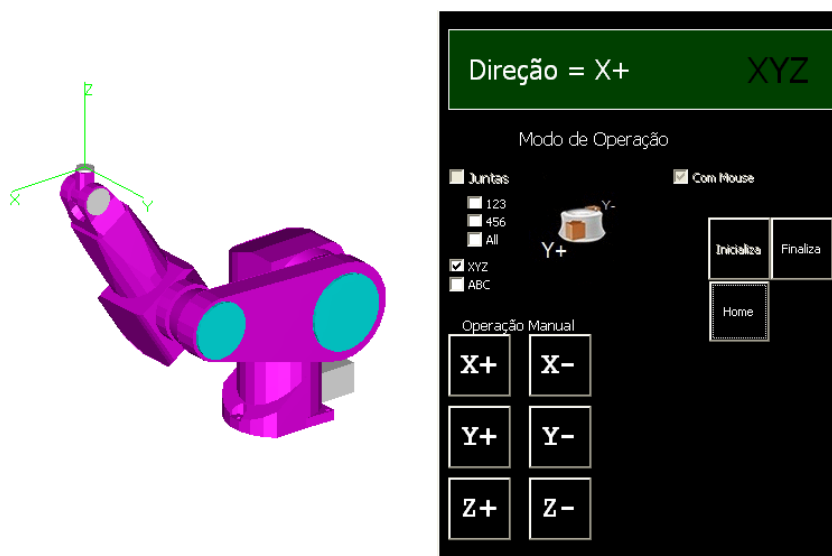
Figura 49 - Teste realizado na direção do eixo +2, modo junta



Fonte: Autor

O teste realizado com o movimento linear segue o mesmo raciocínio do teste anterior, será utilizada a mesma posição padrão para comparação, na Figura 50 é exibido o movimento na direção X+ executado pelo usuário no mouse 3D, no lado esquerdo está o manipulador na sua nova posição, para comprovar que o mesmo foi deslocado no modo linear pode ser observado o sistema de coordenadas do TCP, que está na mesma orientação e só foi deslocado na direção +X. Outro detalhe é que vários eixos foram alterados durante o trecho, o que pode ser visualizado na Figura 51.

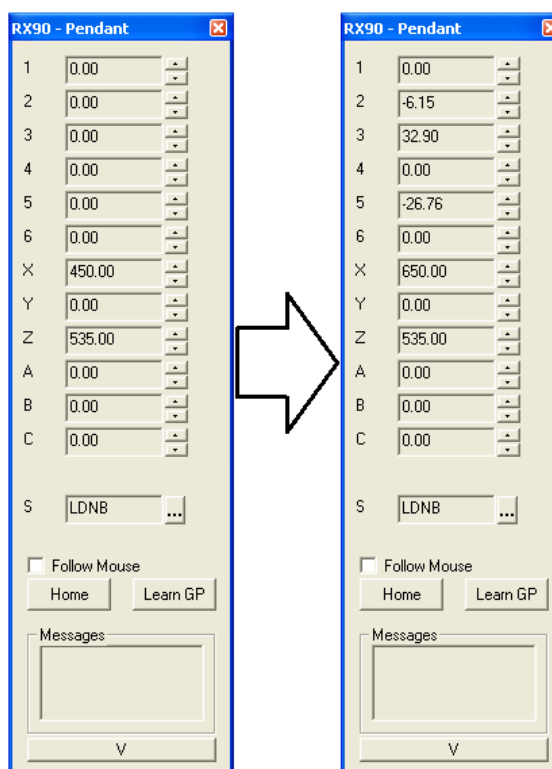
Figura 50 - Teste realizado na direção do eixo +X, modo linear



Fonte: Autor

Na Figura 51 visualiza-se a mudança ocasionada pelo deslocamento no eixo X, no lado esquerdo da imagem estão os valores dos eixos disponibilizados pelo próprio software WorkSpace, é salientado que os ângulos de todos os eixos são iguais a zero, após o deslocamento os novos valores são observados no lado direito, a alteração de três eixos foram necessários para que o movimento ocorresse.

Figura 51 - Visualização dos valores individuais dos eixos



Fonte: Autor

O movimento na direção $-Z$ é semelhante ao visualizado quando alterado o eixo 2, no primeiro momento pode ser confundido os dois movimentos, mas ao olhar-se com detalhe é observado que neste caso a orientação dos eixos do TCP permanece inalterado, a Figura 52 exemplifica o que foi citado anteriormente.

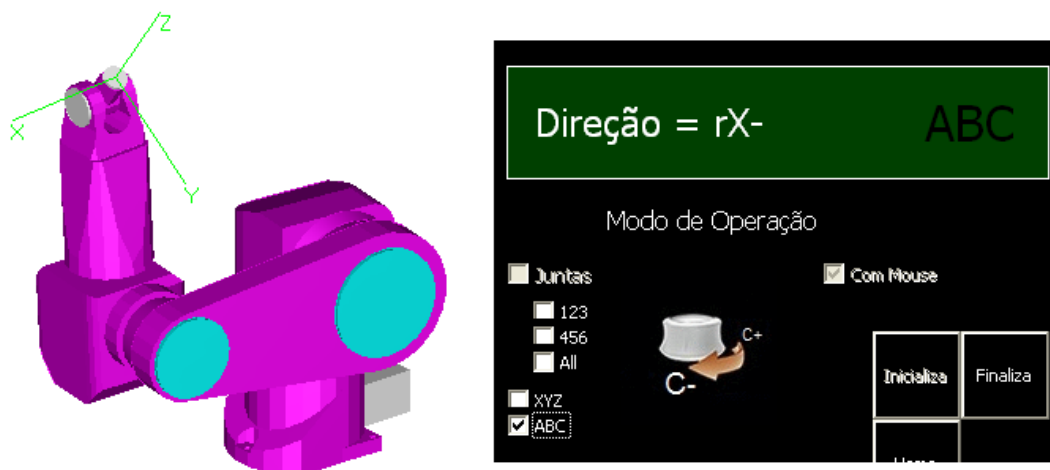
Figura 52 - Teste realizado na direção do eixo $-Z$, modo linear



Fonte: Autor

Os últimos testes realizados na movimentação são os do tipo rotacional, que conforme citado anteriormente são utilizados para girar algum eixo do TCP, nos testes correspondentes o TCP está no flange do manipulador. Na Figura 53 é visualizado o movimento de rotação em torno do eixo X, ao comparar-se o robô com a posição padrão é visto que o eixo X permanece na mesma posição e os outros eixos, Y e Z são deslocados em um certo ângulo.

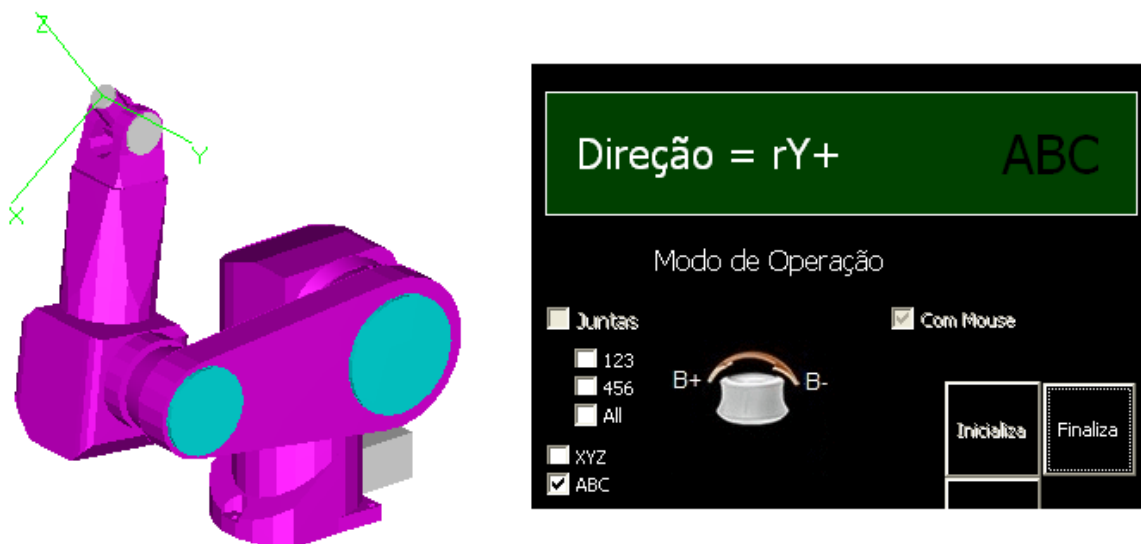
Figura 53 - Teste realizado na direção do eixo $-rX$, modo rotacional



Fonte: Autor

Na Figura 54 o movimento de rotação observado é em torno do eixo Y, já que o eixo Y permanece inalterado e os eixos X e Z são deslocados em um certo ângulo em torno do eixo Y.

Figura 54 - Teste realizado na direção do eixo +rY, modo rotacional



Fonte: Autor

Todos os testes realizados demonstram que as funções estão trabalhando de maneira correta. Não foram ilustrados nesta seção todos os testes realizados, pois há uma quantidade enorme de possíveis testes e todos são originados a partir dos testes demonstrados.

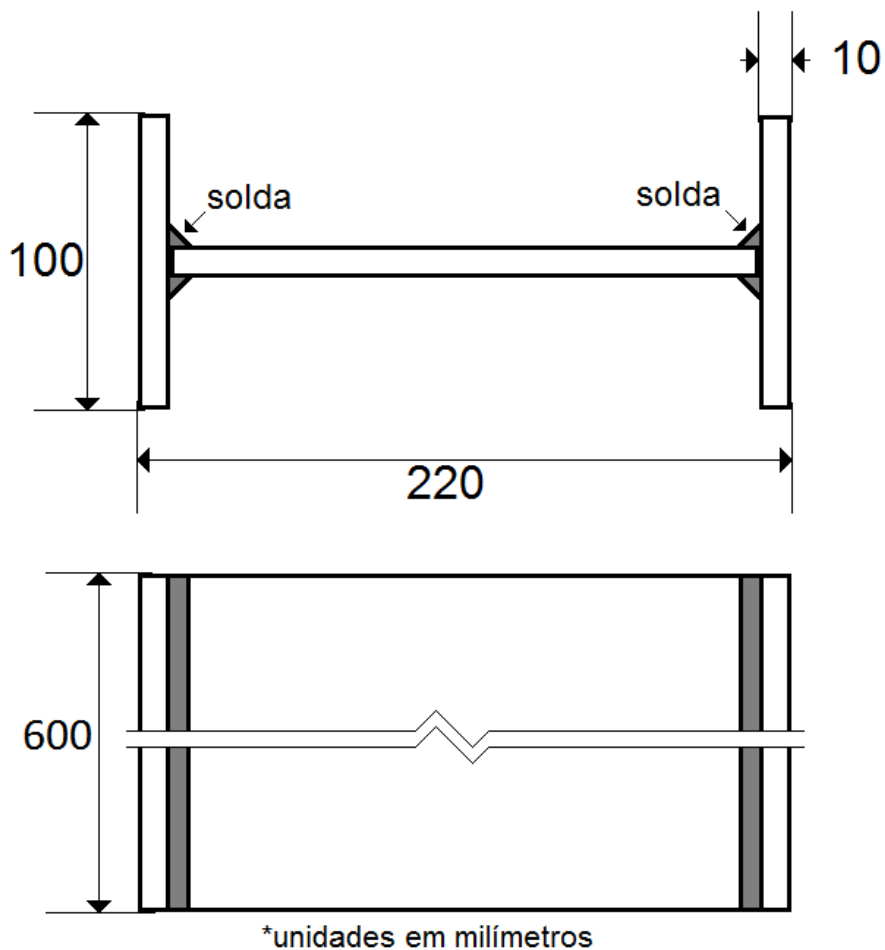
5.4 APLICAÇÃO EM UMA CÉLULA

Até o presente momento só foram visualizados testes com o manipulador em um espaço vazio. Nesta seção será visualizado o robô em um ambiente de trabalho. Para isso será integrado ao robô uma ferramenta e uma pequena estação de trabalho.

O objetivo desta estação de trabalho será a solda de uma viga “I”, esta viga é representada na Figura 55, como o objetivo desta solda é demonstração do funcionamento do robô não serão vistos detalhes da solda, como raiz, velocidade,

tensão ou corrente da fonte. Um detalhe será observado na figura como critério para a solda que é o ângulo de aproximação da tocha de aproximadamente 45° .

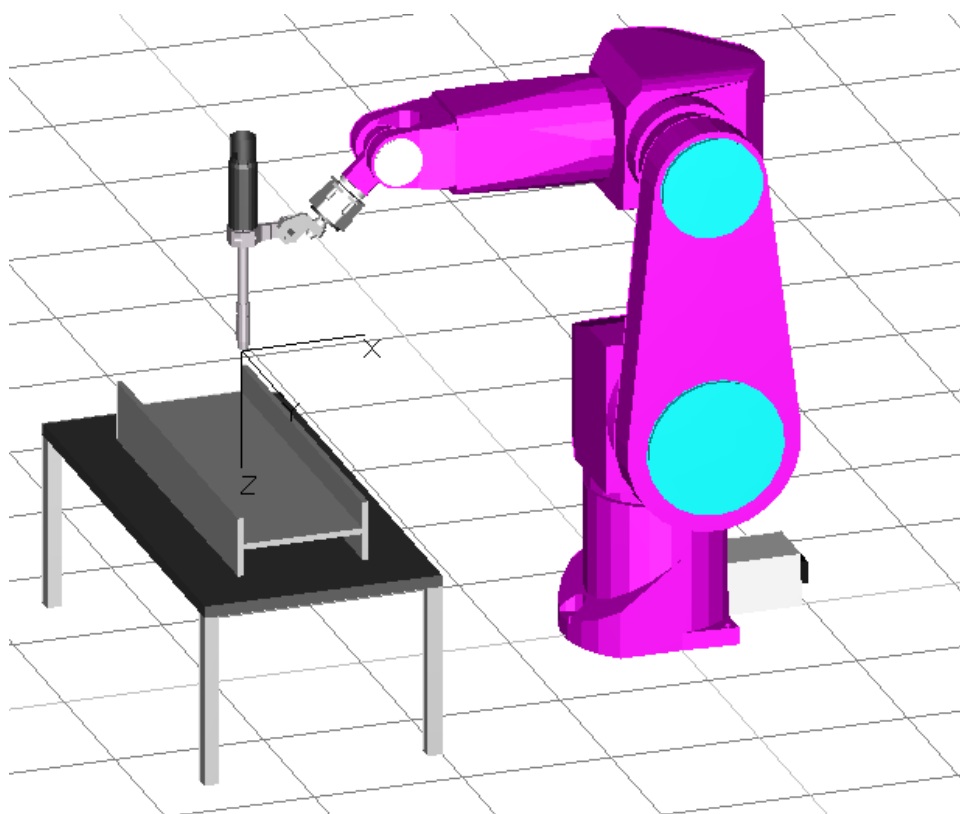
Figura 55 - Dimensões de uma viga tipo "I"



Fonte: Autor

A célula de trabalho é visualizada na Figura 56, a diferença deste manipulador para os observados nas seções anteriores é que o mesmo possui uma ferramenta acoplada no flange. Esta ferramenta é uma tocha de solda. Também é notado que a viga "I" está disposta na horizontal sobre uma mesa simples, conforme citado anteriormente os detalhes dos dispositivos para a solda não serão abordados.

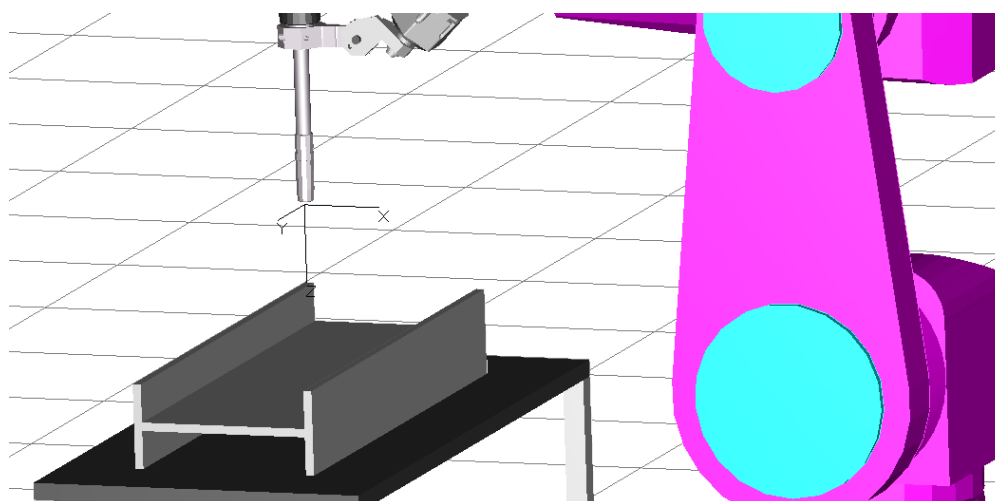
Figura 56 - Célula de solda virtual desenvolvida para testes



Fonte: Autor

Na Figura 57 exibe-se a célula em outra posição. É perceptível que robô encontra-se em uma posição acima da viga e que a ferramenta está ortogonal a viga.

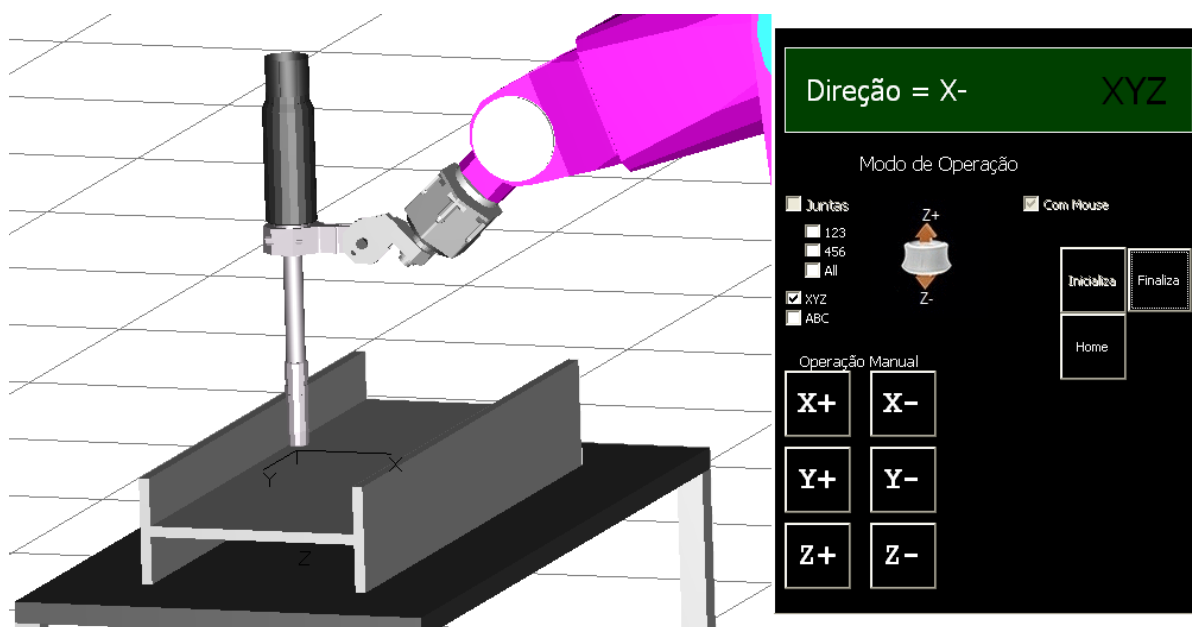
Figura 57 - Célula virtual de solda



Fonte: Autor

Para efetuar a solda é necessário aproximar a ferramenta até uma extremidade do futuro cordão de solda com um ângulo de 45° . Inicialmente são utilizados os movimentos lineares para aproximar a tocha, a Figura 58 exhibe esta aproximação, na qual foram deslocados o eixo Z, para baixo, e o eixo X, para frente. Na posição em que o robô está, falta o deslocamento na direção Y e também a rotação em torno do eixo Y.

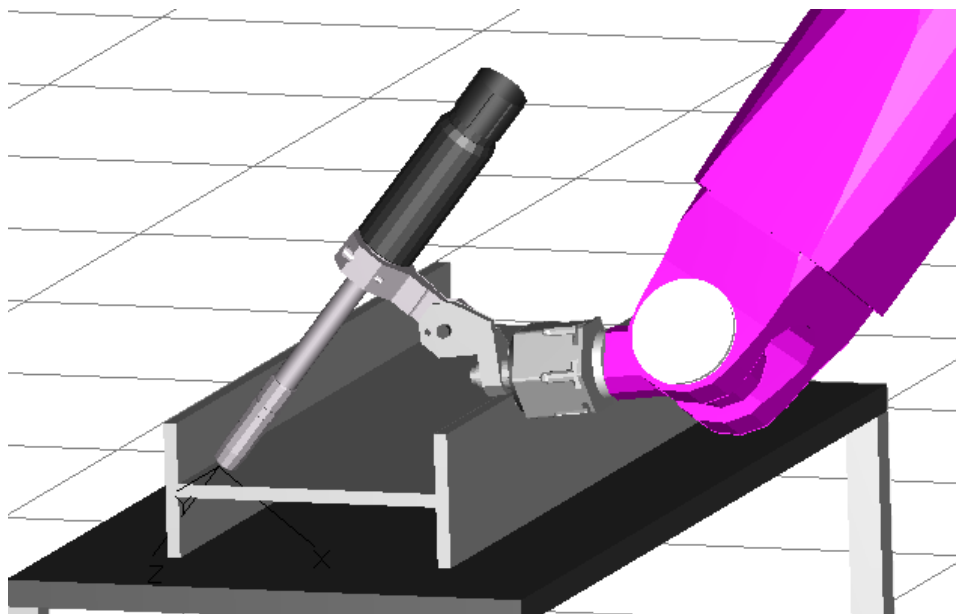
Figura 58 - Aproximando a tocha



Fonte: Autor

A Figura 59 exhibe os movimentos de deslocamento no eixo Y e a rotação em torno do eixo Y, está caracterizada pelo movimento B. Note que o robô está na posição inicial para efetuar a solda, a partir deste ponto o robô deverá percorrer uma linha reta na direção de $-Y$ para realizar a solda.

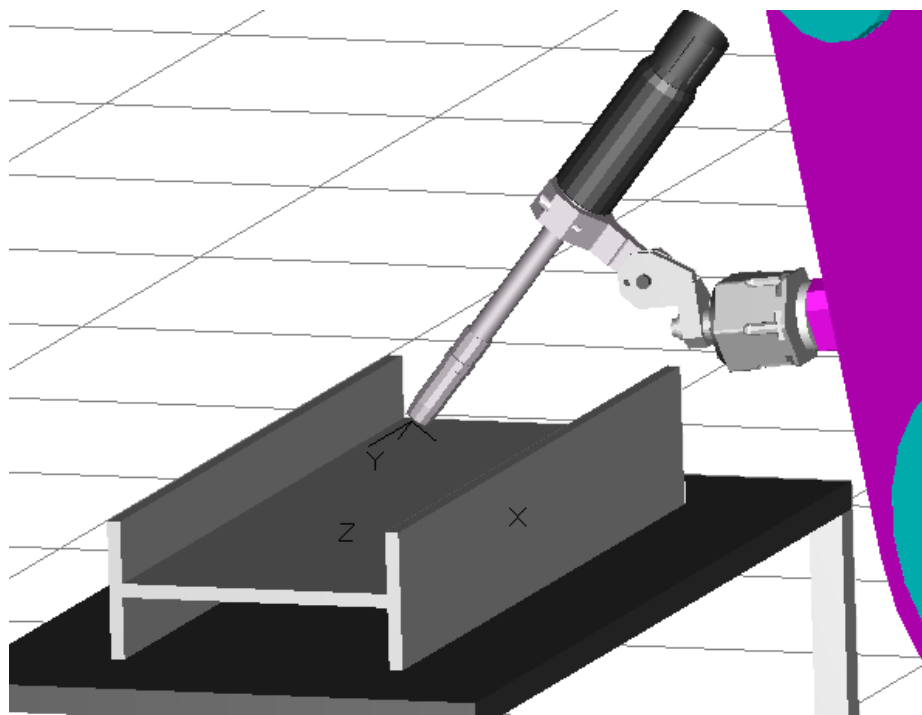
Figura 59 - Posição inicial para efetuar a solda



Fonte: Autor

A Figura 60 exibe o ponto final da execução da solda realizada. É importante frisar que para realizar esta solda só foi utilizado o movimento linear na direção $-Y$.

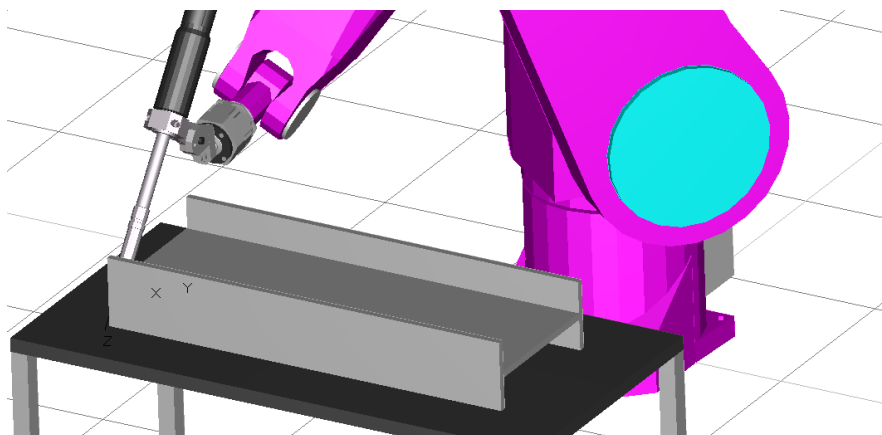
Figura 60 - Posição final da solda



Fonte: Autor

Na Figura 61 visualiza-se a posição anterior do robô só que de outro ângulo. É importante reparar a necessidade dos movimentos lineares e de rotação, que através destes foi possível simular a solda com facilidade e rapidez utilizando o mouse 3D.

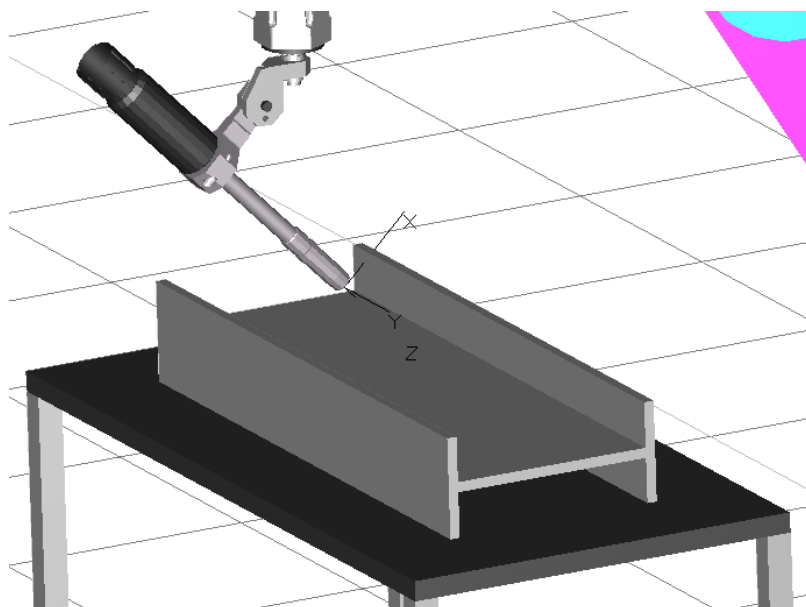
Figura 61 - Outra visualização da posição final da solda



Fonte: Autor

Utilizando movimentos lineares na direção de $-X$ e novamente o movimento rotacional em torno do eixo Y chega-se a nova posição para efetuar a solda, que é vista na Figura 62.

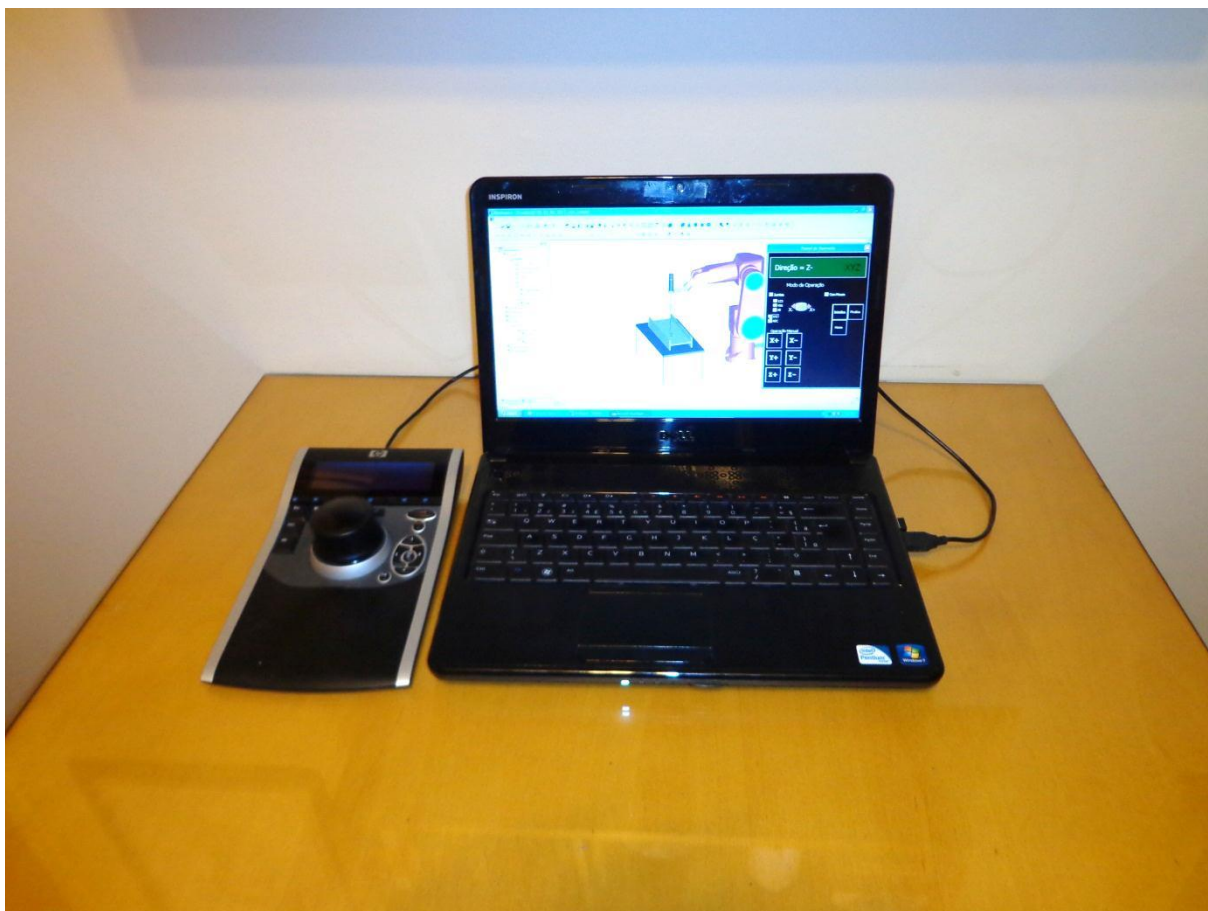
Figura 62 - Posição inicial para efetuar a solda



Fonte: Autor

Na Figura 63 tem-se uma fotografia da operação real do sistema anterior, onde observa-se o mouse 3D e a tela do computador, na tela é visualizado a célula de solda e o painel de operação.

Figura 63 - Operação real do sistema



Fonte: Autor

6 CONCLUSÕES

Os resultados encontrados estão de acordo com os objetivos propostos. O projeto foi capaz de interpretar o movimento executado pelo usuário no mouse 3D, desenvolver uma interface simples e intuitiva no software de desenvolvimento para receber e utilizar os dados oriundos do Mouse 3D, movimentar o robô no modo junta, movimentar o robô no modo linear, movimentar o robô no modo rotacional, aplicar o sistema em uma célula de solda.

Na primeira etapa o resultado da interpretação dos movimentos executados ocorre de maneira adequada, todos os movimentos executados foram coletados e transmitidos.

Os resultados obtidos com a interface de operação para o usuário também demonstram o funcionamento adequado, a tela é simples e com alguns recursos gráficos torna-se intuitivo para o usuário a operação, principalmente em virtude do mouse 3D, que pelo fato de possuir seis eixos o torna mais complexo para operar. Esses recursos desempenham a tarefa de indicar o que cada ação no mouse executará no manipulador.

A aplicação do robô em uma célula para a verificação dos movimentos do tipo junta, linear e rotacional, e também verificar a interface com o usuário demonstram o desempenho do sistema desenvolvido, nela pode-se notar que os movimentos são executados de maneira correta.

O projeto desenvolvido atende os objetivos previamente especificados, mas há itens que são sugeridos para uma futura versão, são eles: implementação de todo o projeto em uma plataforma gratuita, já que foi utilizado o Workspace que é um software proprietário; implementar movimentos lineares com coordenadas em função do TCP, pois atualmente são referenciados através das coordenadas do manipulador. Também seria importante testar o sistema como um todo com usuários iniciantes e partir deste ponto verificar se é viável ou não a sua utilização em ensino ou no treinamento na indústria.

REFERÊNCIAS BIBLIOGRÁFICAS

ABB. **IRC5 com FlexPendant**: Manual do Operador: 3HAC16590-6_Rev_A. Suécia, 2004

_____. **Robot studio**: Operating Manual, 3HAC032104-001. Suécia, 2013.

BRUMSON, Bennet. **Robotic Simulation and Off-line Programming**: From Academia to Industry. Disponível em < <http://www.robotics.org> >. Acesso em: 09 jun. 2013.

CRAIG, J. John. **Introduction to Robotics**. 3. ed. New Jersey: Prentice Hall, 2004.

INTERNATIONAL FEDERATION OF ROBOTICS. **Positive Impact of Industrial Robots on Employment**. London: Metra, 2011.

_____. **Industrial Robots**. Disponível e, <<http://www.ifr.org/>>. Acesso em: 18 maio 2013.

MICROSOFT. **Visual Basic 6**: Programmer's Guide. Disponível em <www.msdn.microsoft.com/en-us/library/>. Acesso em: 09 jun. 2013a.

_____. **Visual C#**: Programmer's Guide. Disponível em <www.msdn.microsoft.com/en-us/library/>. Acesso em: 09 jun. 2013b.

MOTOMAN. **MotoSim EG-VRC**: Operation Manual Ver.2.10. Japão,2008

ROMANO, Vitor Ferreira. **Robótica Industrial**. São Paulo: Edgard Blücher, 2002.

STÄUBLI. **Braço TX90**: Manual de Instruções. 2010

SANTOS, Vitor M. F.. **Robótica Industrial**. 2004. 166 p. Notas de Aula

SICILIANO, Bruno; KHATIB, Oussama. **Handbook of Robotics**. New York: Springer, 2008.

SHAHINPOOR, Mohsen. **A Robot Engineering Textbook**. New York: Harper & Row, 1987.

SHIMON, Y. Nof. **Handbook of Industrial Robotics**. New York: John Wiley & Sons, 1985.

WAT Solutions. **Workspace 5 Manual**. Ontario: 2009.

3DCONNEXION. **Technical Support**. Disponível em <<http://www.3dconnexion.com/service/technical-support.html>>. Acesso em: 09 jun. 2013.

APÊNDICE A – GUIA DE INSTALAÇÃO

Primeiro uso? Se sim execute todas as etapas, se não pule para a etapa 6.

1º) Descompacte o arquivo Projeto.zip na unidade C:\;

2º) Registre as bibliotecas mscomm32.ocx e gif89.dll do sistema operacional;
C:\Projeto\Drivers\

3º) Instale o driver do Mouse 3D, que encontra-se nos arquivos;
C:\Projeto\Drivers\Mouse

4º) Instale o software Virtual Serial Port, que encontra-se nos arquivos;
C:\Projeto\Programas\

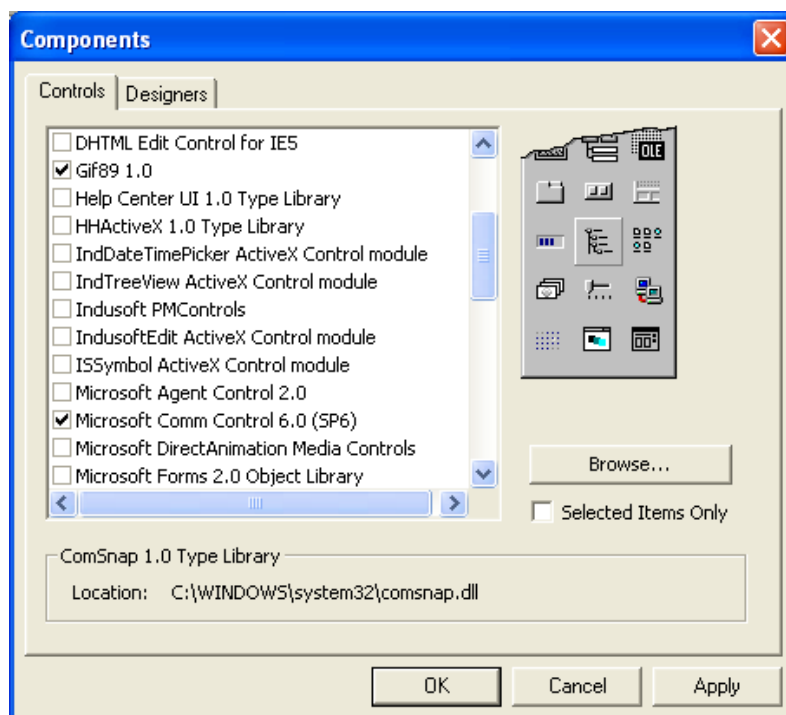
5º) Instale o Microsoft Visual C# 2008, que encontra-se nos arquivos;
C:\Projeto\Programas\

6º) Execute o arquivo Extra_serial_port.vspe, que encontra-se nos arquivos;
C:\Projeto\

7º) Execute o arquivo Envia_Sinais.exe;
C:\Projeto\

8º) Abra o software WorkSpace;

9º) Abra o Visual Basic, comando Ctrl+Shift+E, e após adicione os componentes Microsoft Comm Control e gif89 com o comando Ctrl+T;

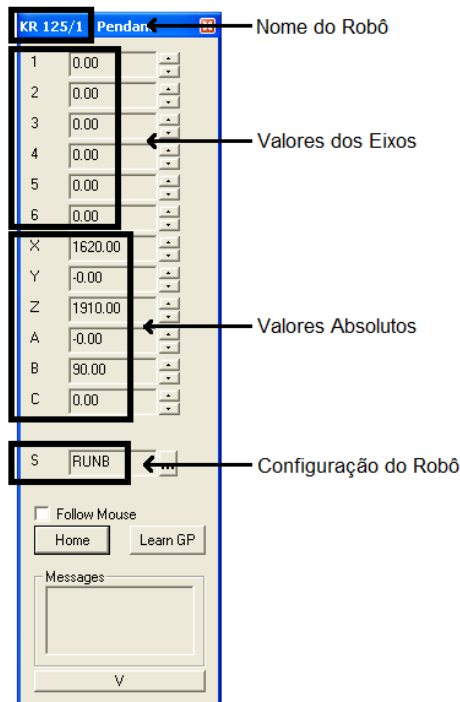


10º) Abra o arquivo padrão para o software WorkSpace Padrao.wsp;

C:\Projeto\Software\

11º) Adicione o robô de sua preferência e abra o pendant, tecla P;

12º) Anote o nome do robô, os valores absolutos e a configuração;



13º) Abra o Visual Basic e altere, a partir da linha 34 do Form Teach_Pendant, os valores anotados do item 12;

```

Rem      CONFIGURACOES INICIAIS
Rem Valores obtidos no pendant
Rem Valores das juntas
  Joint_Home(0) = 0           ' Valor do Eixo 1
  Joint_Home(1) = 0           ' Valor do Eixo 2
  Joint_Home(2) = 0           ' Valor do Eixo 3
  Joint_Home(3) = 0           ' Valor do Eixo 4
  Joint_Home(4) = 0           ' Valor do Eixo 5
  Joint_Home(5) = 0           ' Valor do Eixo 6

Rem Valores de posição
  Pos_ant(0) = 1620           ' Valor de X
  Pos_ant(1) = 0              ' Valor de Y
  Pos_ant(2) = 1910           ' Valor de Z
  Pos_ant(3) = 0              ' Valor de A
  Pos_ant(4) = 90             ' Valor de B
  Pos_ant(5) = 0              ' Valor de C

Rem Configurações do robô
  Robo_utilizado = "KR 125/1" ' ROBÔ UTILIZADO
  configuration = "LUNB"      ' Configuracao do robo

Rem Configurações de uso
  Velocidade = 3              ' DEFINE A VELOCIDADE DE OPERAÇÃO
  PortaSerial = 3             ' Porta Serial Utilizada
Rem      FIM DAS CONFIGURACOES INICIAIS

```

14º) Execute a macro TP.TP na tela do Workspace, Options/Macros/Run;

Obs.: Após criar a estação de trabalho os itens 8 a 13 não serão necessários.

O software WorkSpace é instalado no Windows XP.