Capítulo 4

Estudos de caso e análise de capabilidade

4.1 Introdução

Neste capítulo serão apresentados quatro estudos de caso de experimentos feitos com o robô ASEA, permitindo avaliar em cada um deles o comportamento e desempenho após a finalização do *retrofitting*, com o objetivo de quantificar e conhecer parâmetros que possibilitem estimar se é factível a integração com um processo industrial em sub-esquemas de operações específicas. Avaliar o desempenho do robô permite verificar as capacidades de desenvolver tarefas definidas segundo a aplicação (ROMANO, 2002). Alguns dos parâmetros que serão apresentados neste capítulo por meio dos diferentes estudos de caso estão relacionados com uma medida para quantificar o desempenho de um manipulador industrial, denominado *exatidão* dos movimentos que o robô pode realizar a partir de um processo predeterminado. Parâmetros como a repetibilidade e capabilidade do manipulador em condições não variáveis de montagem, meio ambiente, instrumentos de medição, configuração do sistema de controle e tipos de trajetórias a serem executadas pelo robô, garantem a avaliação desejada. 2

O manipulador deve ter a capacidade de repetir a mesma tarefa repentinas vezes, e para quantificar isso é necessário o parâmetro de avaliação do desempenho do robô chamado *repetibilidade*, o que pode definir como a habilidade do manipulador tem em atingir de forma consistente um ponto específico (ROMANO, 2002). Outros parâmetros que serão apresentados ao longo do presente capítulo, afim de conhecer o desempenho do robô é a capabilidade do processo ou capacidade do processo para produzir produtos ou serviços livres de defeito de forma controlada, permitindo conhecer se o robô esta em capacidade de fazer um processo específico com requisitos de qualidade definidos. Uma das técnicas que permite saber se o processo pode falhar dentro de limites específicos, é a análise de capabilidade(STERN, 2016).

4.2 Validação retrofitting baseado em programação por *teach-in*

No presente estudo de caso é empregada uma técnica de programação on-line denominada programação *teach-in*, a qual possibilita a introdução de um programa específico no controlador de um manipulador industrial. A ideia principal desse tipo de programação é deslocar manualmente o efetuador do robô por meio de uma sequência desejada de pontos sob a supervisão do operador (NOF, 1999). Nesse caso, a sequência de pontos específicos fez com que o robô conseguisse trasladar uma peça tipo *LEGO* de um local para o outro, ressaltando que o robô tem a capacidade de atingir as instruções do controlador *LinuxCNC* baseadas na estratégia de programação *teach-in*.

O módulo *teach-in* responsável por gravar ou adquirir os pontos desejados em termos de coordenadas cartesianas da área de trabalho do robô (X,Y,Z,A,B), foi gerado por meio de um aplicativo desenvolvido em *Python*. Os pontos são guardados em um arquivo de edição, possibilitando a modificação segundo a aplicação final. Nesse caso foi editado o arquivo de saída do módulo *teach-in* para obter dois programas NC compatíveis com o interpretador *LinuxCNC* da norma RS-274. A Figura 4.1 apresenta os passos para obter a programação do robô com o módulo *teach-in*:

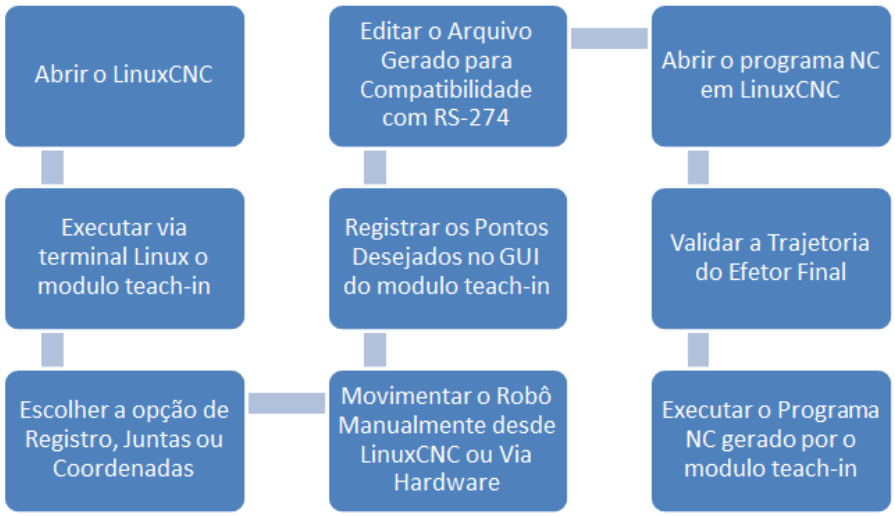

Figura 4.1: Processo de obtenção de pontos com o modulo *teach-in* em *LinuxCNC*

Para a captura de pontos é necessário abrir a interface GUI do *LinuxCNC* e ao mesmo tempo executar o módulo *teach-in* a partir do terminal do Linux, possibilitando desta forma ter um registro dos pontos finais desejados para uma aplicação específica. Inicialmente na Figura 4.2 é apresentada a interface GUI no módulo *teach-in*:

O módulo teach-in tem a opção de fazer o monitoramento dos pontos finais desejados ou armazenar os pontos em um arquivo para edição. Também tem a alternativa de conhecer o deslocamento em graus de cada uma das juntas do robô, assim como os valores das coordenadas cartesianas da área de trabalho. O processo de registro dos pontos deve ser feito de forma manual com um botão disposto para essa tarefa. O detalhe das opções descritas anteriormente são apresentadas na Figura 4.3

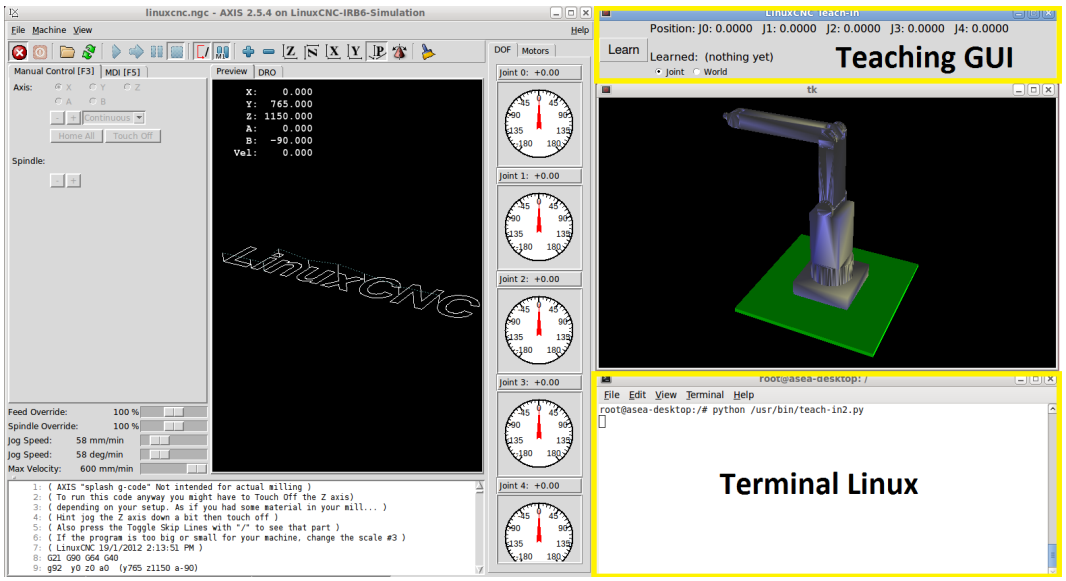

Figura 4.2: GUI inicial *teach-in* com *LinuxCNC*

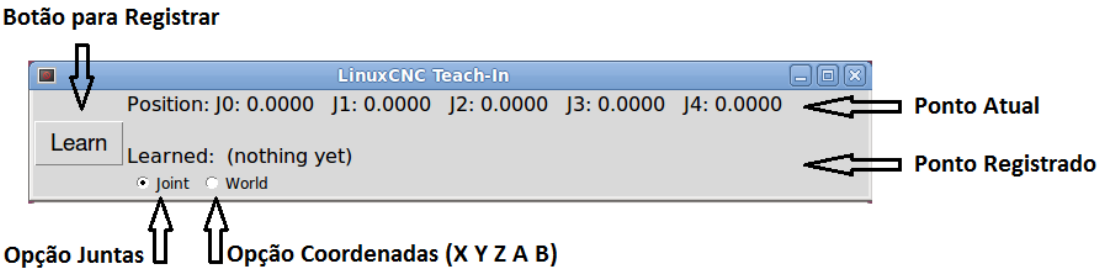

Figura 4.3: GUI do módulo *teach-in*

Através da GUI do *LinuxCNC* ou via hardware, é necessário movimentar o robô até os pontos desejados segundo a aplicação final. Nesse caso, o manipulador ASEA recebeu o comando com um *pendant* P4-S do fornecedor VistaCNC, no qual pode-se observar na Figura 4.4. O uso desse dispositivo facilita o controle do robô por qualquer usuário, garantindo ergonomia e flexibilidade no momento de localizar os pontos desejados para o efetuador do robô, além de ter compatibilidade com *LinuxCNC* por meio do padrão USB.

O arquivo de edição gerado pelo módulo *teach-in* tem a informação dos pontos desejados, segundo a opção escolhida na interface GUI do módulo. Se é *Joint*, o arquivo de edição terá a informação de cada junta do manipulador em graus, mas se a opção escolhida é *World* o arquivo gerado vai ter as variáveis cartesianas da área do trabalho do robô (X, Y, Z, A, B).

A Figura 4.5 apresenta a comparação do arquivo editável com os pontos adquiridos, com um exemplo de como são registrados os valores das variáveis articulares do robô, ou as juntas e as coordenadas cartesianas da áreas de trabalho (X, Y, Z, A, B), gerado por meio do módulo *teach-in*, e da mesma forma é apresentado o formato do arquivo compatível com a norma RS-274.

Após ter os dois programas NC, foi possível executar-los com o controlador *LinuxCNC* para

Figura 4.4: Pendant P4S compatível com *LinuxCNC*
Fonte: VistaCNC (2016, p. 1)

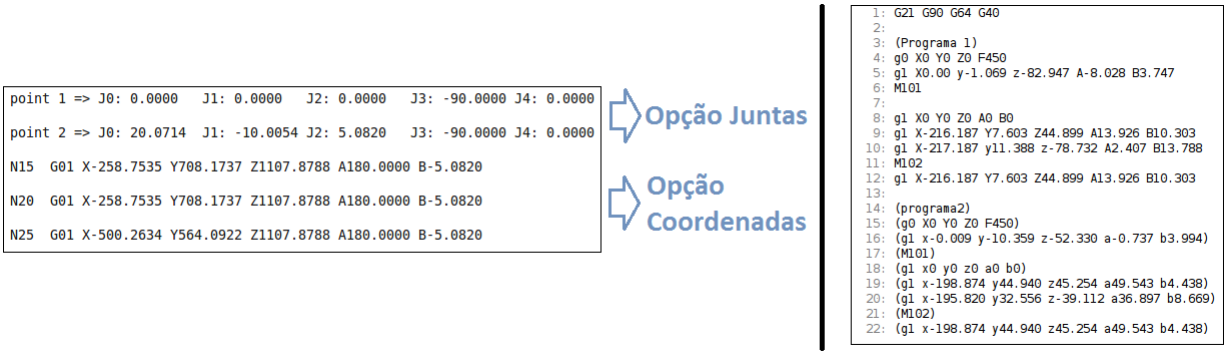

Figura 4.5: Arquivos gerados em *LinuxCNC* com a opção *teach-in*
a) Arquivo pontos adquiridos; b) Arquivo código G

movimentar o robô segundo os pontos finais obtidos pelo módulo *teach-in*, tendo como resultado final a movimentação de uma peça entre dois locais diferentes com o efetuador do robô.

É possível observar na Figura 4.6 a distribuição física disposta para garantir a movimentação de uma peça tipo *LEGO* de um ponto a outro, de acordo com as instruções interpretadas pelo controlador *LinuxCNC*, do programa NC gerado com a estratégia de programação *teach-in*.

A movimentação feita pelo robô para mover a peça *LEGO* tem um registro gráfico na interface GUI do *LinuxCNC* apresentando a trajetória do efetuador do robô. Com isso a garra pneumática fixada na junta final possibilita a execução da aplicação. A Figura 4.7 apresenta a trajetória do

Figura 4.6: Estudo de caso pick and place

efetuador que deverá atingir o manipulador para levar a peça lego segundo os pontos programados via *teach-in*.

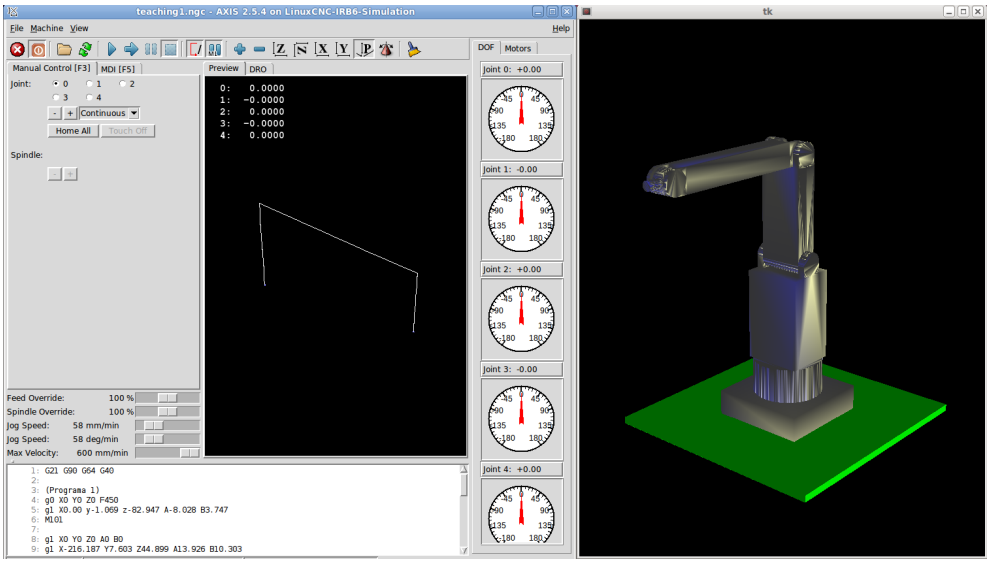

Figura 4.7: Trajetória desejada via *teach-in* programa NC numero 1

Após a execução do primeiro programa NC gerado pelo módulo *teach-in*, é possível validar a trajetória seguida pelo efetuador do manipulador na GUI do controlador *LinuxCNC*, assim como uma simulação básica do modelo do robô IRB6-S2, conforme apresentado na Figura 4.8.

Para acionar a garra pneumática foram gerados dois comandos compatíveis com a norma RS-274 para abrir e fechar a garra quando estivesse na posição adequada com a peça *LEGO*. Esses comandos são M101 (Abrir) e M102 (Fechar) adicionados ao programa NC.

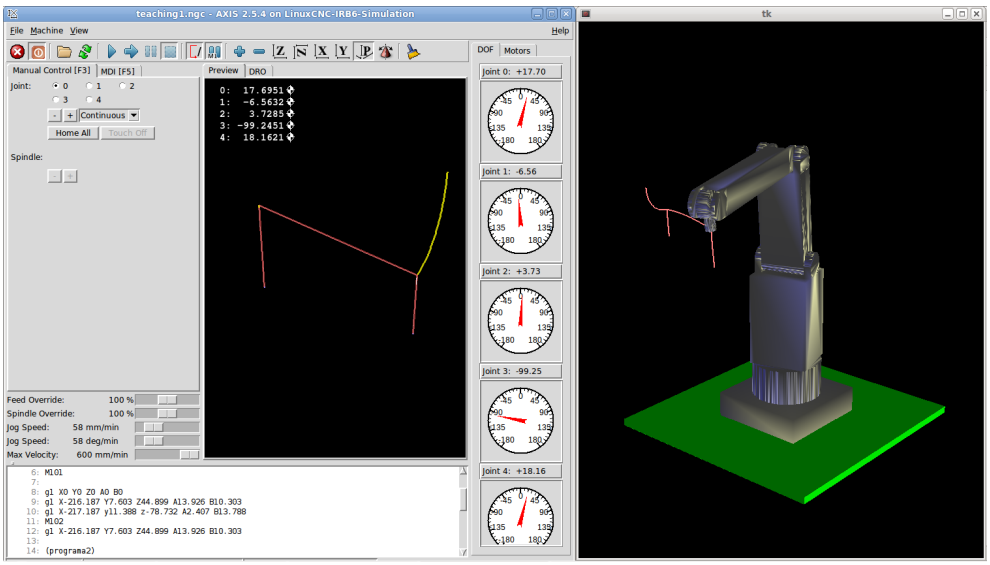

Figura 4.8: Simulação posicionamento pecã *LEGO* Programa NC numero 1

É possível observar na Figura 4.9 a segunda simulação antes de executar o segundo programa NC obtido a partir do módulo *teach-in*, permitindo verificar por meio do controlador *LinuxCNC* se existem linhas de código com erros que devam ser corrigidos possibilitando assim, que o robô movimente a peça entre pontos desejados.

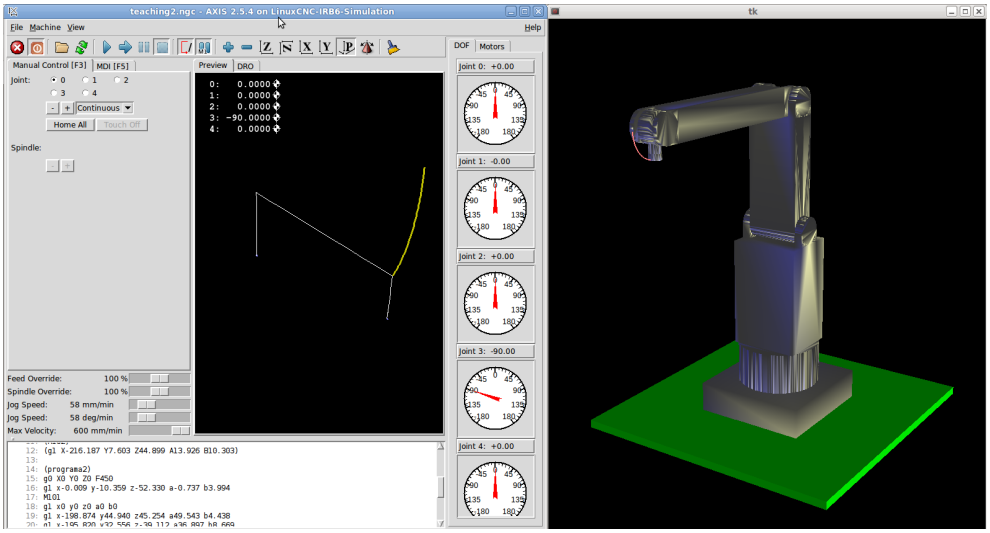

Figura 4.9: Trajetória desejada via *teach-in* programa NC numero 2

Finalmente pode-se observar na Figura 4.10, no GUI do controlador *LinuxCNC* a trajetória atingida pelo efetuador do manipulador do segundo programa NC gerado a partir do módulo *teach-in*, permitindo o deslocamento de uma peça tipo *LEGO* entre dois pontos estabelecidos pelo operário na programação *on-line* com o robô IRB6-S2.

Os dois programas NC gerados a partir da estratégia de programação de robôs *teach-in*, foram verificados antes de ser executados para determinar possíveis erros nas linhas de código através

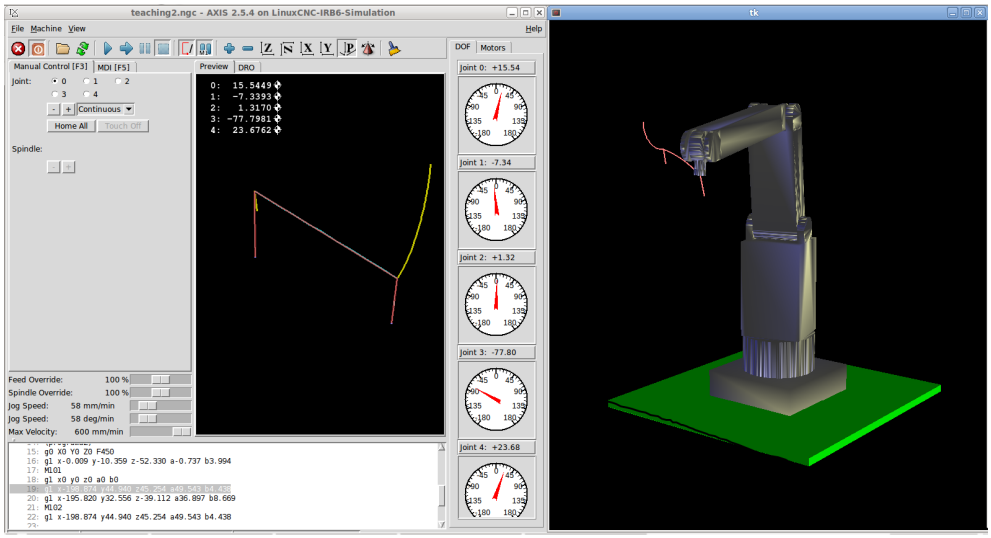

Figura 4.10: Simulação posicionamento pecã *LEGO* programa NC numero 2

do *LinuxCNC*, e posteriormente executados e simulados em tempo real e permitindo validar que o manipulador após o *retrofitting* tem a capacidade de se deslocar manualmente com a supervisão de um operador para permitir a programação *on-line* do módulo *teach-in*, além de atingir pontos desejados registrados e interpretados de um arquivo compatível com a norma RS-274.

Os programas NC usados para movimentar uma peça de um ponto inicial até um ponto final estão disponíveis no apêndice A. Os vídeos da movimentação do robô para os dois programas NC estão disponíveis para consulta no (ARENAS, 2016).

4.3 Validação *retrofitting* baseado em programa NC gerado por CAD/CAM

Com o programa exemplo do *LinuxCNC*, gerado por CAD/CAM compatível com a norma RS-274, foi possível validar a cinemática do robô através da movimentação do efetuador, desenhando em uma folha de papel a figura interpretada pelo controlador com ajuda de uma caneta fixada na junta final do manipulador. Abaixo é apresentada na Figura 4.11 a interface GUI padrão do *LinuxCNC*, assim como a trajetória realizada pela caneta fixada ao manipulador IRB6-S2.

Terminada a execução do programa exemplo no controlador *LinuxCNC*, na Figura 4.12 é apresentada a simulação da trajetória atingida pela ferramenta na interface GUI do controlador.A simulação que foi executada em tempo real junto com a movimentação física do manipulador para que a caneta, como o efetuador do presente experimento conseguisse desenhar sobre a folha de papel o traço determinado pelo programa NC carregado e interpretado pelo controlador *LinuxCNC*.

Paralelamente a simulação no controlador *LinuxCNC*, o robô fez a movimentação de cada uma das juntas segundo a cinemática integrada própria do manipulador, esperando ter ao final da execução da rotina um resultado semelhante ao mostrado pela simulação. É possível observar

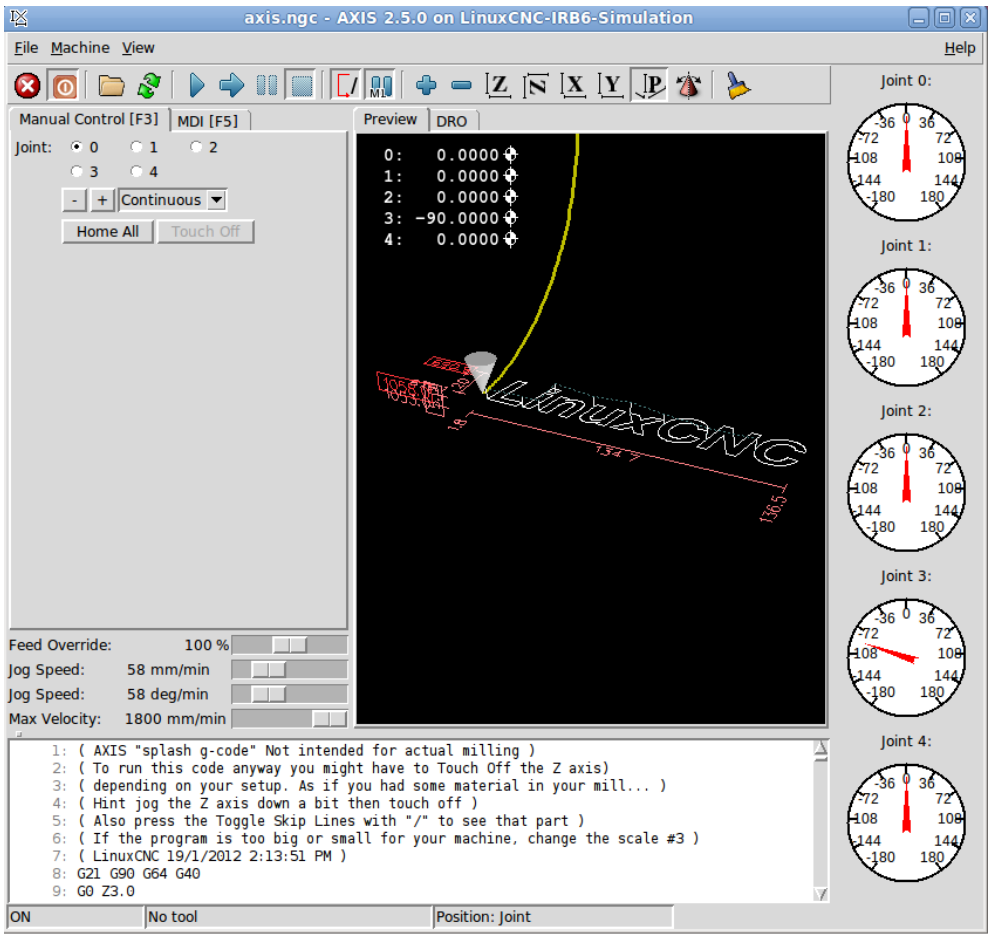

Figura 4.11: Trajetória a ser atingida pelo manipulador

na fig. 4.13 a operação do manipulador, tal como o acondicionamento feito para possibilitar a validação da cinemática do manipulador através do desenho feito em uma folha de papel com uma caneta comum.

Terminada a execução da rotina foi possível observar graficamente, na folha de papel se o resultado gerado pelo robô é semelhante as instruções geradas pelo controlador baseado no programa NC. O resultado é apresentado na fig. 4.14

Graficamente é possível contrastar o resultado obtido no desenho da trajetória feita pela caneta fixada ao robô, e garantir que a cinemática implementada no controlador -nesse estudo de caso- está funcionando virtualmente e fisicamente com a configuração do robô ASEA. O vídeo da execução do presente estudo de caso está disponível para consulta em (ARENAS, 2016).

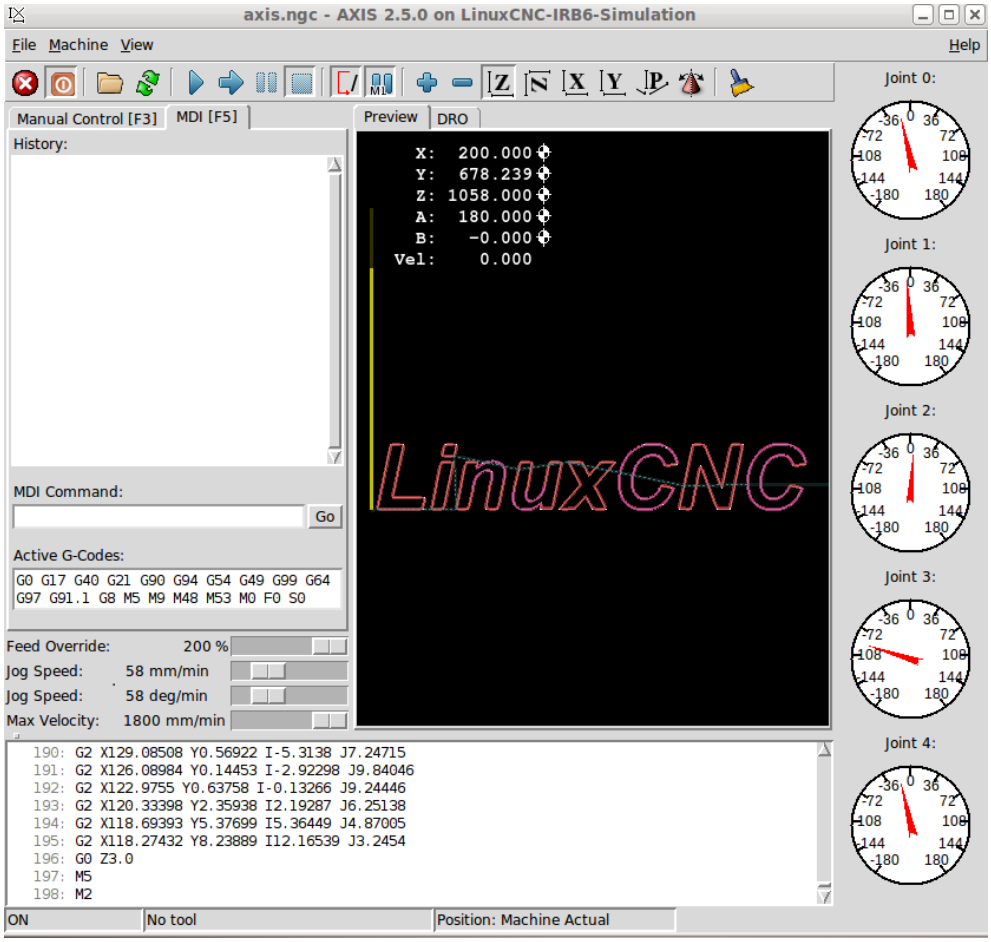

Figura 4.12: Simulação da trajetória gerada pelo programa NC

4.4 Estudo de repetibilidade com peça de geometria complexo em 2D gerada por CAD/CAM

Para esse experimento mede-se a diferença entre a espessura do traço gerado por várias percursos feitos pelo efetuador e a mesma trajetória desenhada em um único percurso, tudo comandado através do *LinuxCNC*, possibilitando estimar a diferença entre as espessuras das linhas feitas pela caneta e assim permitindo avaliar a *repetibilidade* como uma estimativa da distância máxima esperada da diferença entre as posições atingidas pelo manipulador de um único ponto programado previamente. Neste caso, Para o presente experimento foi usada uma distribuição de *T-Student* para extrair o número de amostras a ser executadas pelo robô, nesse caso foram escolhidas 10 repetições para um t = 3,69.

O programa NC gerado para este estudo de caso foi escolhido de um exemplo do controlador *LinuxCNC* onde foram removidas algumas linhas de código, visando diminuir o tempo da execução total dos testes. É possível observar na Figura 4.15 a simulação da trajetória gerada pelo efetuador do manipulador, nesse caso foi usada uma caneta fixada na última junta. O programa NC pode ser consultado no apêndice B.

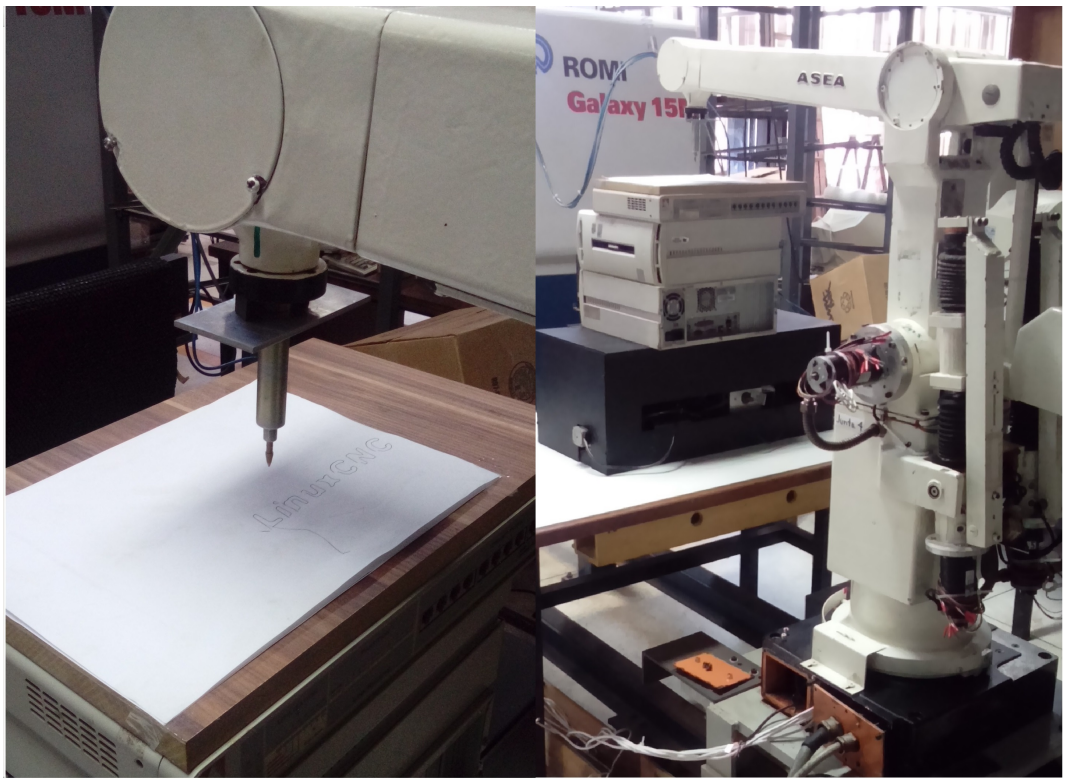

Figura 4.13: Robô em operação executando o programa NC

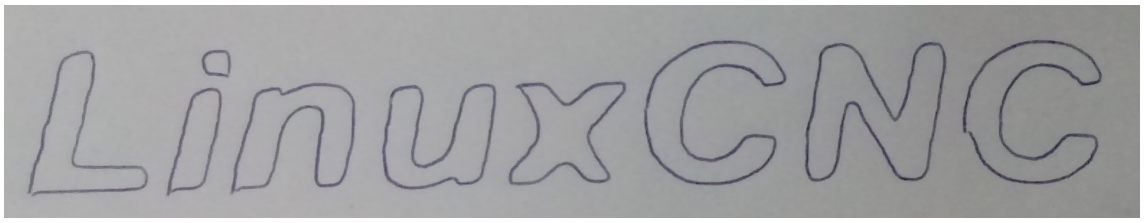

Figura 4.14: Resultado do traço feito pela caneta como efetuador do robô

Após verificar que o controlador *LinuxCNC* conseguiu interpretar o programa NC é possível executar as 10 rotinas, cada uma sobre o mesmo plano de trabalho, evitando movimentar a folha de papel onde a caneta fez o determinado desenho. Na Fig. 4.16 se apresenta a primeira e algumas outras corridas posteriores ao inicio deste estudo de caso.

Finalizando as 10 corridas se apresenta na Figura 4.17 o desenho feito pela caneta sempre foi constante em qualquer uma das amostras feitas, garantindo desta forma que o robô pode fazer a mesma atividade programada n vezes, reduzindo a incerteza do processo no qual pode haver falhas na operação se as condições externas pudessem mudar durante o processo de marcação da folha de papel.

Para quantificar o erro na repetibilidade do processo com o robô ASEA foi feita uma última corrida na mesma folha de papel usada para os 10 procedimentos. A Figura 4.18 esta apresentando o resultado geral ao momento de fazer a marcação das figuras desenhadas pelo robô.

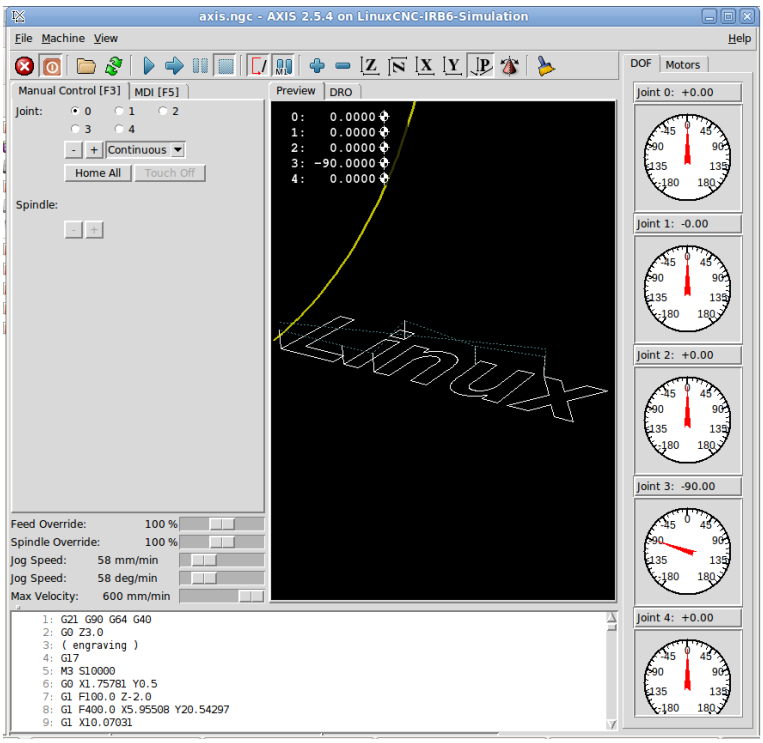

Figura 4.15: Simulação inicial da trajetória do efetuador

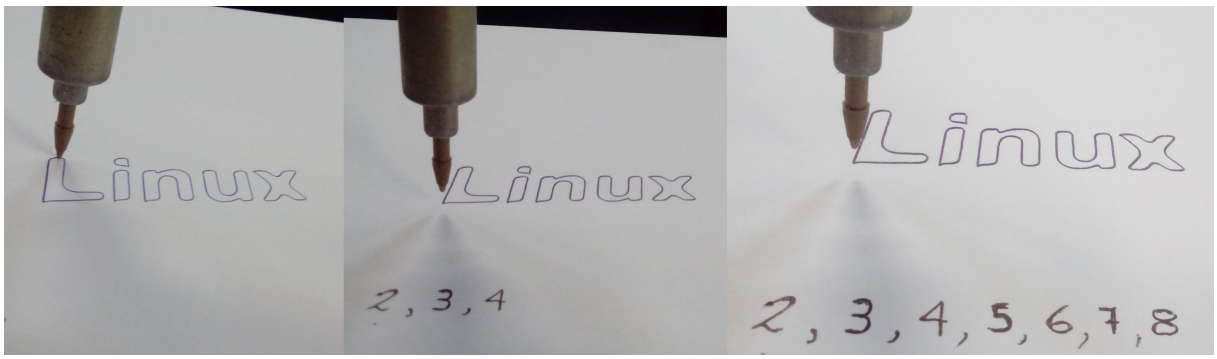

Figura 4.16: Registro digital da repetibilidade

Para estimar o erro médio das espessuras do traço feito nas 10 repetições em comparação a um traço feito com a mesma trajetória, foram medidos 50 pontos diferentes para cada um dos dois traços, levando em conta que para uma distribuição normal de dados é necessário ter mais de 30 amostras (WALPOLE; MYERS; MYERS, 1999), e assim definir um intervalo de confiança de 95% para este experimento de *repetibilidade*.

Com ajuda do medidor/projetor de perfil *Mitutoyo PJ-A3000* foi possível fazer as medições das espessuras do traco em pontos específicos de cada uma das letras desenhadas pela caneta. O projetor permite ter uma imagem em escala do traço permitindo obter uma boa estimação do valor das espessuras. Na Figura 4.19 é possível observar o equipamento usado no momento da tomada das amostras na folha de papel que foi usado no experimento.

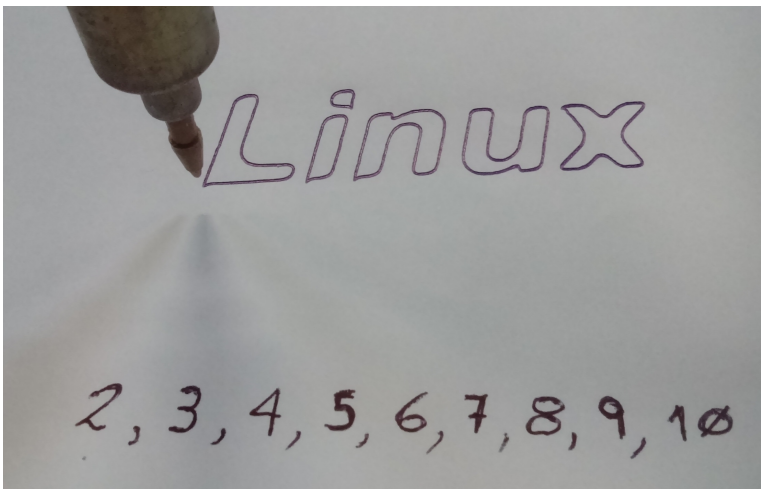

Figura 4.17: Resultado final da repetibilidade do programa NC

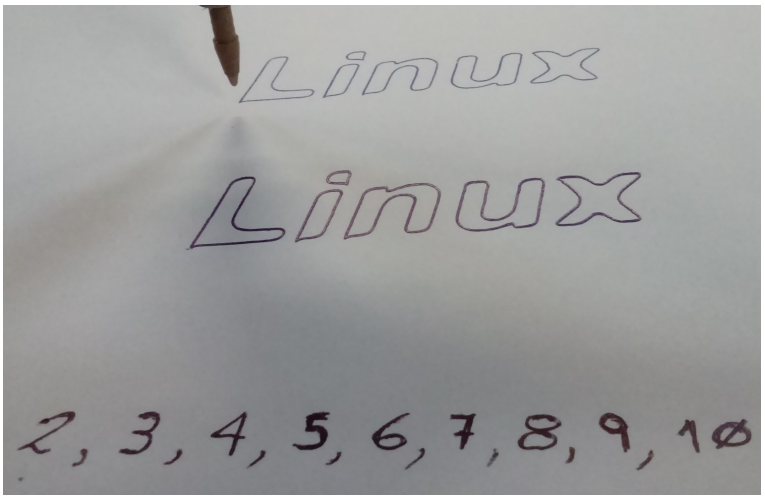

Figura 4.18: Resultado gráfico para medição das espessuras

O projetor de perfil usado para o experimento tem uma interface para facilitar a medição de coordenadas 2D de forma manual, sendo assim, foi usada a função chamada "*line to point*", a qual permite obter a distância entre um ponto e uma linha através da normal, neste caso, para achar a espessura do traço. Na Figura 4.20 é apresentada a função e a interface de usuário disposta para fazer as medições.

É possível observar na Tabela 4.1 os valores das espessuras em pontos diferentes do desenho feita em uma repetição, para cada uma das letras geradas a traves do programa NC. O medidor/projetor de perfil *Mitutoyo PJ-A3000* apresenta uma incerteza de medição de $\pm 0.15\%$ do valor obtido em cada amostra (MITUTOYO, 2011).

Para analisar as amostras obtidas na medição dos pontos, verifica-se o comportamento dos dados em uma curva normal, como mostra a Figura 4.21, observando dois tipos de resultados da qualidade do processo feito. Neste caso a repetição das marcas registradas na folha de papel.

Figura 4.19: Projetor de perfil *Mitutoyo PJ-A3000*

Tabela 4.1: Medição das espessuras do traço único

Letra	1 (mm)	2 (mm)	3 (mm)	4 (mm)	5 (mm)	6 (mm)	7 (mm)	8 (mm)	9 (mm)	10 (mm)
L	0.435	0.413	0.441	0.408	0.434	0.391	0.432	0.390	0.391	0.433
I	0.426	0.425	0.463	0.475	0.414	0.328	0.326	0.334	0.353	0.460
N	0.434	0.345	0.462	0.361	0.412	0.410	0.347	0.395	0.416	0.429
U	0.401	0.452	0.396	0.331	0.371	0.421	0.413	0.463	0.345	0.367
X	0.403	0.438	0.424	0.381	0.413	0.393	0.391	0.420	0.397	0.385

Um destes tipos de análises leva em conta um grupos de dados, ou seja, grupos de amostras do mesmo experimento chamado *Overall Capability*, entretanto a seguinte análises denominado *Potential Capability* gera resultados da capabilidade de fazer um processo com condições inicias específicas de todos os dados coletados, sem importar os subgrupos.

Visando o análises do experimento desenvolvido pelo manipulador, no presente estudo de caso foi usado e analisado valores relacionados com os resultados que agrupam todos os dados como único conjunto, ou seja, (*Potential Capability*). A Figura 4.21 foi gerada pelo software de estatística *Minitab17*.

Com o analises de capabilidade feito pelo software de estatística *Minitab*, é possível saber se o processo desenvolvido pelo robô tem a capacidade de produzir falhas controladas, com os indicadores gerados pela análise, como Cp e Cpk. Estas duas variáveis podem ser usadas juntas para saber o quão capaz é o processo, além de validar que o processo seja centralizado (STAROVESKI

Figura 4.20: Opção "*line to point*"da interface de usuário do projetor de perfil

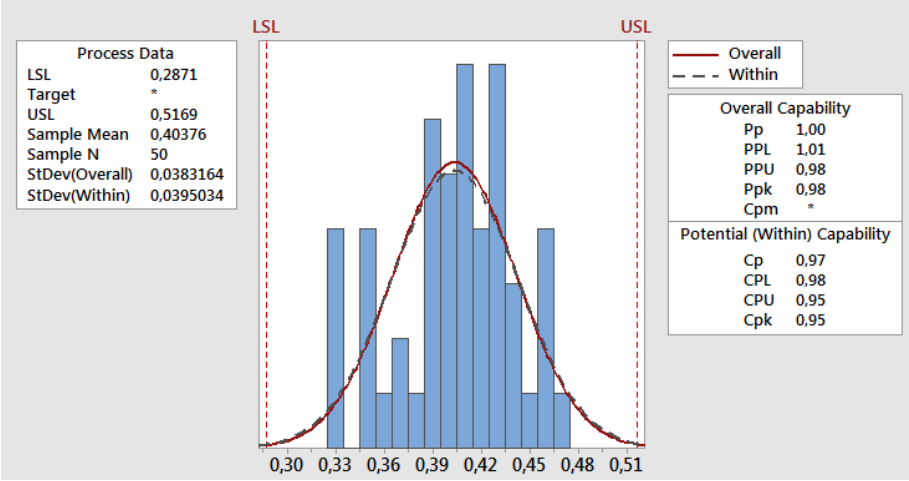

Figura 4.21: Curva normal das espessuras medidas do traço único

et al., 2009). Cp é a capabilidade potencial do processo e define se o processo é capaz de cumprir especificações, enquanto o Cpk é o índice de capabildiade, que permite conhecer se o processo esta centrado segundo os limites ou requerimentos. Um processo com capacidade de produzir

produtos nos limites de desenho não necessariamente pode ser aceito devido aos resultados não serem centralizados aos limites do processo, o que geraria alternativas para melhorar os resultados e garantir a capabilidade do processo.

Se o indicador de capabilidade Cp é medido, deve haver um valor igual ou superior a 1, sendo que em manufatura frequentemente o valor nominal deve ser igual o maior a 1,33, para que o processo seja considerado estável. Desta forma com os resultados apresentados na Figura 4.21, o Cp tem um valor de 0,97, segundo os dados coletados, e é possível inferir que o processo esta perto de ser estável para um Cp igual a 1, alem disso o indicador Cpk é de 0.95, o que indica que o processo esta centralizado segundo os limites de tolerância.

Com os valores apresentados anteriormente é possível conhecer o valor médio das espessuras feitas no desenho de referência, assim como o desvio padrão (σ). Tendo que o valor médio ou media, é o promédio numérico de um número determinado de dados em uma amostra, enquanto σ pode-se definir como a variação ou dispersão em relação a média ($\bar{x}$) (WALPOLE; MYERS; MYERS, 1999). O resultado da média para os valores das espessuras apresentadas na Tabela 4.1 esta detalhado na Eq. 4.1, enquanto o desvio padrão se apresenta na Eq. 4.2. A espessura de um único traço tem valor nominal de 0,403 mm, ou seja, este é o valor do traço esperado quando o efetuador do Robô com caneta executa uma marcação no papel. Este valor será usado como referência pra estimar o erro sistemático de múltiplos traços gerados pelo efetuador. Quanto melhor a precisão de posicionamento do Robô na marcação do traço sobre o traço já existente, melhor será a repetibilidade estimada pro Robô. Assim este valor de traço será utilizado como valor nominal pra estimar a parcela de erro sistemático da avaliação de desempenho usando o conceito/métrica de repetibilidade.

$$\bar{x}_1 = \frac{x_1 + x_2 + ... + x_n}{n} \quad = 0.403mm \tag{4.1}$$

$$\sigma_1 = 0.039mm \tag{4.2}$$

Foi necessário medir em pontos parecidos em cada letra desenhada sobre o papel, para conhecer a espessura dos traços no experimento de *repetibilidade* com as 10 repetições do mesmo programa NC. Os valores são apresentados na Tabela 4.1.

Tabela 4.2: Medição das espessuras do traço com 10 repetições

Letra	1 (mm)	2 (mm)	3 (mm)	4 (mm)	5 (mm)	6 (mm)	7 (mm)	8 (mm)	9 (mm)	10 (mm)
L	0.517	0.493	0.503	0.632	0.621	0.478	0.643	0.456	0.504	0.739
I	0.508	0.502	0.491	0.518	0.595	0.501	0.493	0.475	0.516	0.663
N	0.507	0.508	0.502	0.518	0.593	0.516	0.479	0.423	0.446	0.640
U	0.465	0.498	0.595	0.464	0.448	0.511	0.511	0.607	0.472	0.473
X	0.549	0.616	0.627	0.565	0.526	0.509	0.522	0.620	0.552	0.515

Novamente o uso do *Minitab17* para obter o análise de capabilidade e o gráfico dos valores normalizados obtidos na medição é indispensável, com o objetivo de continuar o analises dos resultados no presente estudo de caso, podendo-se observar na Figura 4.22:

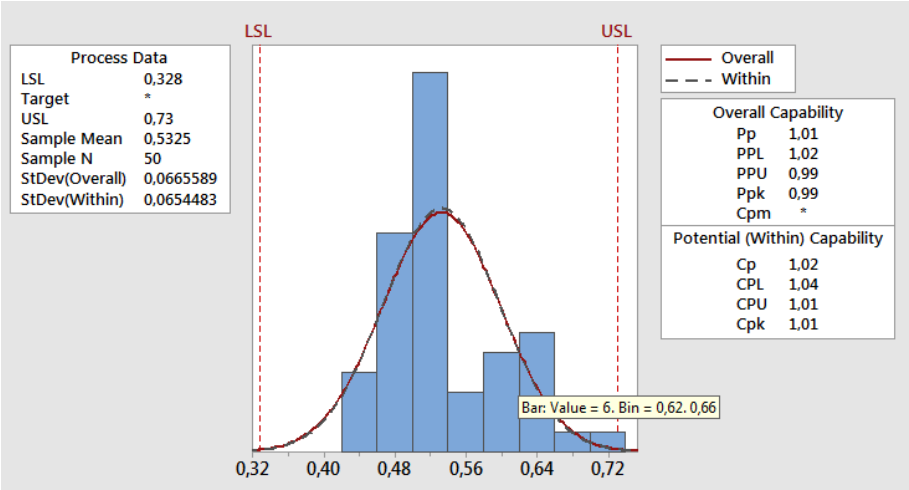

Figura 4.22: Curva normal das espessuras medidas com repetição

Da curva normal apresentada na Figura 4.22 é possível inferir que os dados coletados do traço feito em 10 repetições tem valores fora dos limites de uma da função gaussiana, ou seja, dados afastados do valor da espessura do traço desejado quando o robô faz várias repetições do programa escolhido. Além disso, novamente como no analise de capabilidade para um único traço os valores do limite superior e inferior de engenharia foram calculados para um Cp igual a 1, o que indica que o processo é estável. A Figura 4.22 apresenta o valor Cpk próximo a Cp, por tanto o processo é centralizado.

A partir da análise do Capabilidade Potencial (*Potencial Capability*), é possível conhecer a média $\bar{x}_2$ e o desvio padrão σ_2 para o experimento de *repetibilidade* nas seguintes equações:

$$\begin{aligned} \bar{x}_2 &= \frac{x_1 + x_2 + ... + x_n}{n} \\ \bar{x}_2 &= 0.532mm \end{aligned} \tag{4.3}$$

$$\sigma_2 = 0.065mm \tag{4.4}$$

Com os valores da média e desvio padrão calculados obtidos do analise de capabilidade nas Eq. (4.1), (4.2), (4.3) e (4.4) é possível avaliar e estimar a repetibilidade da tarefa executada varias vezes em referencia ao programa executado uma vez a traves da Eq. (4.5), para um intervalo de confiança igual a 95%.

$$ErroSistemático(\bar{x}_2, \bar{x}_1) = \bar{x}_2 - \bar{x}_1 = 0.129mm \tag{4.5a}$$

$$\sigma_{repetibilidade} = \sqrt{\sigma_2^2 + \sigma_1^2} = 0.076mm \tag{4.5b}$$

$$Repetibilidade(IC de 95\%) = ErroSistemático(\bar{x}_2, \bar{x}_1) \pm 2\sigma_{repetibilidade} \tag{4.5c}$$

$$Repetibilidade = 0.129 \pm 0.151 \tag{4.5d}$$

$$Repetibilidade_{min} = -0.02mm \tag{4.5e}$$

$$Repetibilidade_{max} = 0.28mm \tag{4.5f}$$

O valor negativo de estimação de repetibilidade não faz sentido físico, pois não há traço da linha negativo. Assim sendo, o valor estimado para a repetibilidade avaliada em relação ao traço de uma única linha é de 0,28 mm, ou seja, quando vários posicionamentos são repetidos n vezes o erro associado a repetibilidade do efetuador do robô que executa a marcação do traço é de 0,28 mm, mostrando um grau de precisão/variabilidade bom, comparável a robôs comerciais novos e modernos como KUKA KR6/2 que é de 0,1 mm (ABREU, 2002) e o ABB 6660 que tem repetibilidade de 0,07 mm (OLIVEIRA, 2013). Considerando tratar-se de um robô de 1975 modernizado, o valor estimado pra repetibilidade surpreende positivamente, pois segundo a ABB seus Robôs evoluíram significativamente desde de 1974 (Asea IRB 6) até os dias atuais, quando apresentavam uma precisão de posicionamento de 1 mm e hoje está na ordem de 10 mícrons (ABB, 2014).

O vídeo da execução do presente estudo de caso esta disponível para consulta em (ARENAS, 2016).

4.5 Ensaios geométricos com peça padrão moldura: erro de retilineidade e erro de perpendicularidade

Geometricamente é possível analisar os resultados do manipulador em uma determinada tarefa visando um feedback para melhorar o desempenho em futuros testes o aplicações específicas. Desta forma, na presente secção verifica-se através de um programa NC executado pelo manipulador o parâmetro geométrico denominado retilineidade para um plano de trabalho 2D, baseado numa peça moldura definida para avaliar o parâmetro nos eixos X e Y do robô IRB6-S2. Lá retilineidade pode ser definida como uma representação qualitativa de uma superfície em termos de variação/desvio da sua geometria em referência a uma linha reta pré definida (BEWOOR, 2009). A desenho feito por uma caneta fixada na última junta do manipulador pode-se observar na Figura 4.23.

Com o quadro desenhado numa folha de papel pela caneta fixada na última junta do robô, é possível fazer medições do comprimento em diferentes posições em referência aos eixos X e Y. As amostras obtidas foram medidas com um calibrador digital *SHAN*, com uma precisão de $\pm 0.03mm$. O instrumento usado pode ser observado na Figura 4.24.

Foram definidos 14 pontos de referência para cada eixo (X e Y) e desta forma obter uma boa precisão das amostras medidas. A medição de cada comprimento foi uma média de 10 medições

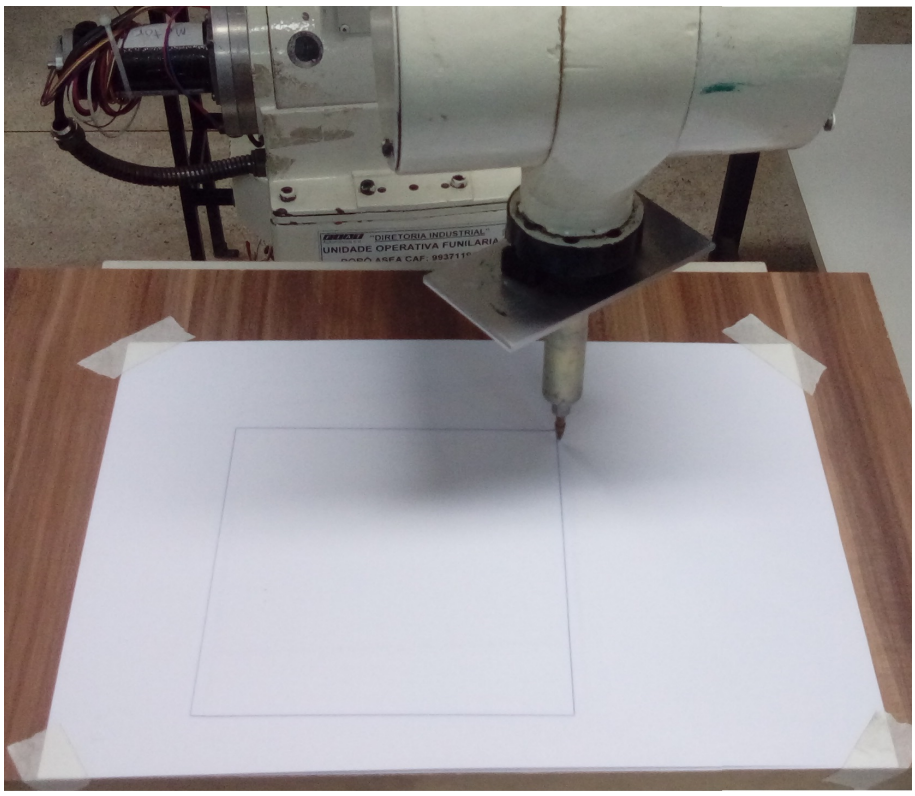

Figura 4.23: Marcação da caneta na área de trabalho

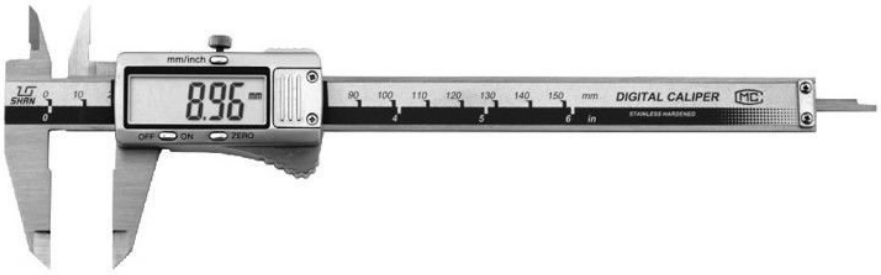

Figura 4.24: Paquímetro digital usado para as medições

para cada um dos pontos, com uma aproximação ao ponto do meio do traço feito pela caneta, garantindo uniformidade do conjunto de valores. É possível observar na Figura 4.25 como foi feita a distribuição dos pontos de referencia para medir cada um dos comprimentos de cada eixo.

Para gerar um gráfico de regressão linear dos dados obtidos na medição do comprimento ao longo dos eixos X e Y do quadro desenhado pelo robô, é necessário calcular o erro de cada um dos valores médios medidos com o calibrador digital. É apresentada a Eq. 4.6, a qual permite obter o valor do erro para cada amostra. O valor nominal programado no arquivo NC foi de 140 mm.

$$Erro_{x,y} = ValorNominal_{x,y} - ValorMedido_{x,y} \tag{4.6}$$

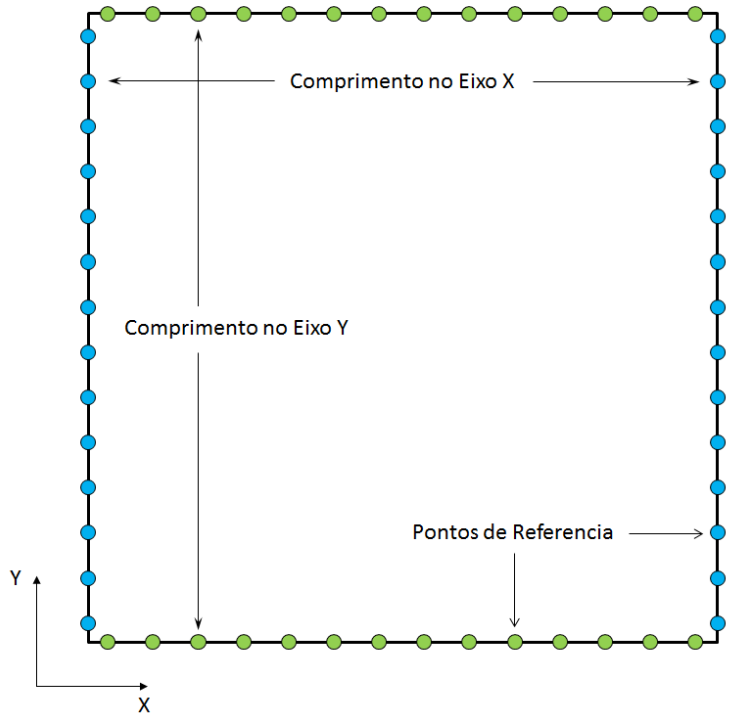

Figura 4.25: Pontos de referências para as medições

Com os valores obtidos de cada uma das medições obtidas com o calibrador digital, foi possível calcular os erros de retilineidade para os eixos X e Y. Na Tabela 4.3 são apresentados os valores para cada um dos eixos analisados no presente experimento, assim como os erros em referência ao Valor Nominal projetado para a movimentação do efetuador do robô.

Tabela 4.3: Medição do comprimento ao longo dos eixos X e Y

Posição	X (mm)	Y (mm)	Ex (mm)	Ey (mm)
1	140,86	140,04	-0,86	-0,04
2	140,81	140,11	-0,81	-0,11
3	140,79	140,12	-0,79	-0,12
4	140,74	140,19	-0,74	-0,19
5	140,66	140,20	-0,66	-0,20
6	140,54	140,21	-0,54	-0,21
7	140,36	140,23	-0,36	-0,23
8	140,30	140,26	-0,30	-0,26
9	140,16	140,37	-0,16	-0,37
10	140,08	140,39	-0,08	-0,39
11	140,07	140,41	-0,07	-0,41
12	139,98	140,42	0,02	-0,42
13	139,97	140,42	0,03	-0,42
14	139,97	140,43	0,03	-0,43

Com os valores dos erros foi possível fazer a análise da regressão lineal de cada um dos eixos

X e Y, permitindo conhecer uma aproximação da magnitude do erro de retilineidade que atualmente tem o manipulador. Na Figura 4.26 é apresentada a linha de regressão para os valores correspondentes ao eixo X.

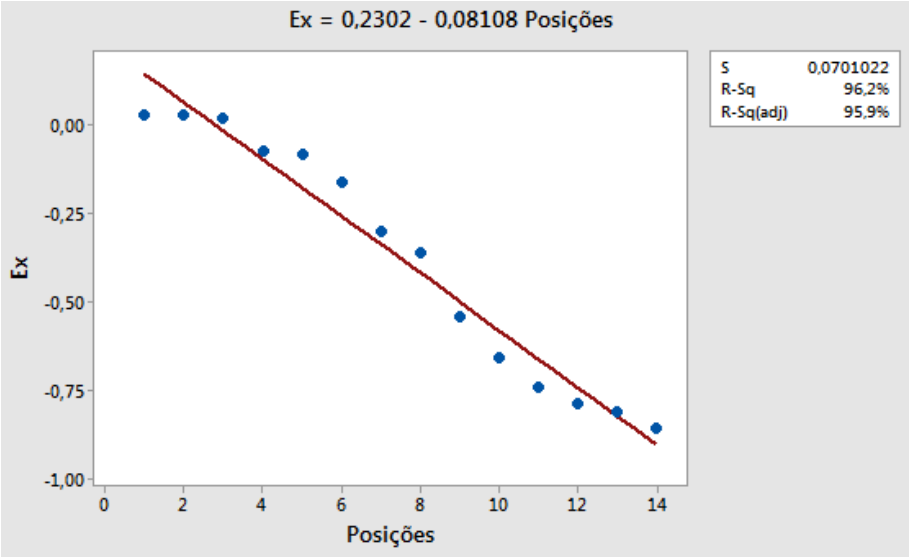

Figura 4.26: Regressão lineal do erro de retilineidade em X

Segundo os resultados da Figura 4.26, é possível observar que os valores das posições medidas estão perto da linha de regressão, ou seja, que o Erro Padrão da Regressão (S) tem um valor pequeno, representando desta forma a distância promédia dos valores observados em referência à linha de regressão. Enquanto o valor do coeficiente de determinação, R^2, com um valor de 95.9% (perto de 100%), indica que os dados se ajustam ao modelo de regressão lineal calculado através dos dados obtidos (SCHUMACKER, 2014).

Na Figura 4.27 apresenta-se a analise de regressão para o eixo Y, com base nas amostras coletadas no desenho feito pelo manipulador na folha de papel.

Em comparação com o resultado do eixo X, na Figura 4.27 apresenta pouca dispersão dos dados em referência ao modelo de regressão linear, o que se traduz um valor de 95.2% para R^2. Por outro lado, os dados apresentam pouca distância em relação a linha de regressão, justificando esta observação desde o valor de (S), menor ao valor calculado para o eixo X.

Na Eq. 4.7 é possível observar as equações das linhas de regressão para cada um dois eixos analisados na presente secção.

$$E_x = -0.08p + 0.2302 \tag{4.7a}$$

$$E_y = -0.03p + 0.0376 \tag{4.7b}$$

Com as equações das retas de regressão é possível achar os valores máximos de error de retilineidade para os eixos X e Y, com base no calculo de erro de resíduo, o qual é possível observar na

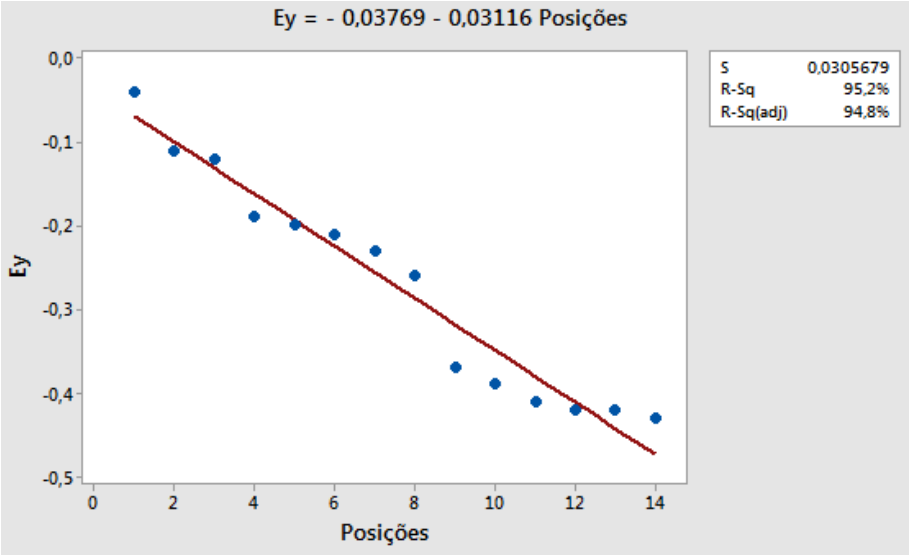

Figura 4.27: Regressão lineal do erro de retilineidade em Y

Eq. 4.8 como é possível calcular os valores máximo segundo os resíduos ou distancia entre o valor nominal e o valor da reta de regressão (BRUCE, 2015)

$$E_{x\text{-}máx} = \left|X_i - \bar{X}\right| = 0.119 \tag{4.8a}$$

$$E_{y\text{-}máx} = \left|Y_i - \bar{Y}\right| = 0.361 \tag{4.8b}$$

O grau de erro de retilineidade é aceitável para um robô que foi desenvolvido na década dos 80, em comparação a robôs modernos e com modelos de calibração avançados como o ABB IRB140 que apresenta um erro de repetibilidade de 0.115 mm (PAZIANI; GIACOMO; TSUNAKI, 2009),ou o manipulador IRB 6400 com valores que não excedem os 0.5 mm (YOUNG; PICKIN, 2000).

Com as equações de regressão apresentadas anteriormente (4.7), é possível conhecer a inclinação das retas de regressão em referência aos eixos X e Y, segundo o desempenho do manipulador no momento de fazer a peça moldura. Sendo m o coeficiente angular da reta, a Eq. 4.9 pode estimar o valor a inclinação da equação de regressão.

$$Y = mX + b \tag{4.9}$$

Com os valores $m_x = 0.08$ e $m_y = 0.03$ é possível estimar que existe baixa inclinação das retas de regressão em referência os eixos X e Y respetivamente, segundo o traço gerado na folha de papel da peça moldura planejada no programa NC.

Com as equações das linhas em relação ao referencial cartesiano de abcissas e ordenadas, é possível estimar se as dois retas são perpendiculares. Para isso é necessário calcular o angulo entre as linhas, ou seja, devem ter um ponto de interseção o que na literatura se denomina como

concorrência. Para que duas retas sejam perpendiculares deve cumprir com a condição que o angulo formado pelas duas linhas seja igual a 90° (RYAN, 1991). É necessário trocar na equação de regressão do erro em Y (E_y) os valores das coordenadas próprias das abscissa e ordenada, para que a referencia seja o eixo vertical e não o horizontal (ROHDE et al., 2012). A continuação na Figura 4.28 se apresenta a regressão para E_y com os valores trocados entre os eixos de referência.

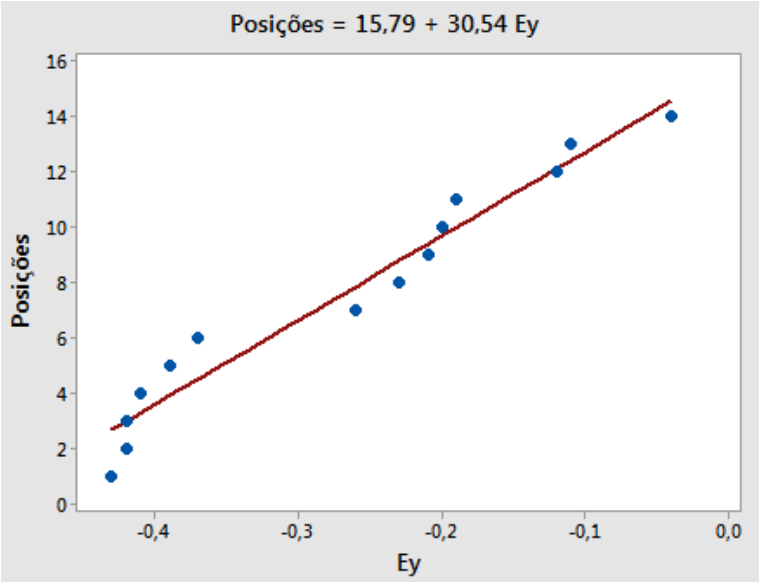

Figura 4.28: Regressão linear do erro em Y com eixos trocados

Para validar se as duas linhas que correspondem aos eixos X e Y do programa NC executado pelo manipulador IRB6-S2 são perpendiculares, é necessário obter o ângulo θ que formam as duas retas matematicamente descritas na Eq. 4.7. Para obter o ângulo é usada a Eq. 4.10 junto com coeficiente m_x e m_y de cada uma das retas de regressão descritas anteriormente (LARSON, 2010).

$$\tan\theta = \left| \frac{m_2 - m_1}{1 + m_1 \cdot m_2} \right| \tag{4.10}$$

Segundo os valores dos coeficientes das retas de regressão, $m_x = -0.08108$ e $m_y = 30.54$, é possível calcular o ângulo entre as retas como é apresentado a continuação na Eq. 4.11.

$$\theta = 87.24° \tag{4.11}$$

Desta forma pode-se assumir que os eixos X e Y na execução do programa NC a traves do robô IRB6-S2 tem um erro de perpendicularidade, definido como a diferencia entre o desvio das linhas de regressão e 90°, obtendo um valor de erro aproximado de 2.76°, assumindo assim a condição de perpendicularidade existente para a peça moldura desenhada para o presente experimento (STONE, 2012).

O vídeo da movimentação do robô para executar o programa NC da peça moldura está disponível para consulta no (ARENAS, 2016).

4.6 Análise dos resultados obtidos

Segundo os resultados obtidos em cada um dos experimentos é possível identificar as causas que afeitam o desempenho do robô após do retrofitting nas tarefas feitas e que permitiram obter os valores para as variáveis de desempenho identificadas ao longo do presente capitulo. A sínteses dos resultados são apresentados na Tabela 4.4.

Tabela 4.4: Síntese dos resultados obtidos

Parâmetro	Erro
Programação via *teach-in*	Foi gerado e executado um programa NC por meio da técnica de programação on-line *teach-in*
Programação via CAD/CAM	Foi executado um programa NC exemplo gerado através de CAD/CAM
Repetibilidade	Foi obtido um erro de $0.28mm$ após o *retrofitting*
Retilineidade	Foi estimado um erro de retilineidade para o eixo X de 0.119 e para Y de 0.361
Perpendicularidade	O erro de perpendicularidade estimado foi de $2.76°$

É possível fazer uma programação off-line do robô desde a metodologia de aprendizagem de maquinas *teach-in*, obtendo pontos desejados do elemento final do manipulador comandado em uma interface tipo *pendant*, permitindo que o operador possa obter um posicionamento exato devido á possibilidade de estar observando de perto os pontos específicos para o deslocamento do manipulador. O robô tem capacidade de se movimentar segundo o programa NC gerado através do módulo *teach-in* do controlador *LinuxCNC* aos pontos capturados na programação via *on-line*.

Baseado em um programa NC gerado por CAD/CAM e interpretado pelo controlador *LinuxCNC*, foi possível estimar a repetibilidade do manipulador depois de fazer mais de 10 traços da mesma trajetória, consegue ter um erro de repetibilidade aproximado a 0.28 mm. Sendo um valor aceitável para um robô que foi construído há mais de 40 anos e em operação continua numa fabrica automotiva da *FIAT* antes de ser doado á Universidade de Brasília, além de se comparar com robôs atuais com um erro de repetibilidade de 0.1 mm. Isto possibilita que o manipulador tenha a capacidade de fazer a mesma tarefa várias vezes com uma variável de desempenho estimada após o *retrofitting*, com um erro de posicionamento conhecido e aceitável.

Foi possível identificar variáveis que afetam o desempenho do robô no experimento de repetibilidade, como a instabilidade da mesa onde estava a folha de papel onde o robô fez a marcação em varias opções, assim como a geração de espessuras do taco longe da media calculada quando o robô atingia uma curva em cada uma das geometrias das letras planejadas no programa NC. ë possível melhorar os resultados, poderia se ter um número de amostras mínimo de 500 pontos para cada uma das repetições, segundo a norma ISO 9283 segundo os critérios de desempenho e métodos de testes correlatos para manipuladores industriais (SIRINTERLIKCI et al., 2009).

Mediante análises geométricas foi possível conhecer o erro de retilineidade entre a distância comandada e a distância executada pelo manipulador ao longo dos eixos X e Y. O erro de retilineidade esteve entre 0.2 e 0.4 milímetros aproximadamente, garantindo desta forma que o robô

IRB6-S2 após o procedimento de *retrofitting* está em capacidade de fazer linhas retas com um erro controlado e relativamente baixo em comparação a robôs industriais modernos e que tiveram métodos de calibração avançados. Outro parâmetro importante avaliado no robô após o processo de *retrofitting* foi a perpendicularidade entre duas retas desenhadas com a caneta, obtendo um erro baixo de aproximadamente 3 % em comparação com os 90° graus que deveria ter idealmente duas retas com este tipo de variável de desempenho avaliada.

Variáveis que afetavam este parâmetro de desempenho foram identificadas tais como a instabilidade da área de trabalho onde o robô fez a marcação e as limitações mecânicas do manipulador e que devido a movimentação das juntas serem mecanicamente dependentes entre si, permitindo compensar com valores aproximados o movimento das juntas de forma manual direitamente nas equações homogêneas do controlador *LinuxCNC*.