

Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

**CALIBRAÇÃO E REPLANEJMENTO DE TAREFAS PARA UM
ROBÔ INDUSTRIAL EM AMBIENTES HOSTIS**

Autor:

Benito Fontes Monteiro Filho

Orientador:

Prof. Fernando César Lizarralde

Co-orientador:

Prof. Antônio Candea Leite

Examinador:

Examinador:

DEL

Fevereiro de 2009

AGRADECIMENTO

A Deus pela força e pela luz nos momentos de dificuldade.

Aos meus pais que sempre me apoiaram incondicionalmente. À minha irmã Tatiana e ao meu cunhado Gilberto pelo apoio quando foi preciso.

À minha noiva Ana Carolina e sua mãe dona Conceição pelo companheirismo e paciência.

Ao meu co-orientador Antônio C. Leite pelo apoio, paciência e dedicação.

Ao meu orientador Fernando Lizarralde por me orientar e ter acreditado em meu potencial.

RESUMO

Em sistemas robóticos avançados, capazes de operar em lugares hostis e de difícil acesso, o planejamento de uma trajetória de referência geralmente é realizado na superfície ou em um ambiente seguro e estruturado. Entretanto, quando a célula robótica é inserida no ambiente de trabalho pode apresentar incertezas em sua localização absoluta em relação à localização original adotada, caracterizando-se em um problema de replanejamento de trajetória. Neste contexto, o replanejamento pode ser realizado a partir da estimativa do erro de configuração (posição e orientação) entre as localizações já citadas, utilizando algoritmos de calibração, bem como a trajetória ideal previamente planejada.

A calibração é uma atividade essencial quando é necessário utilizar um sistema robótico para operar em ambientes remotos, como é o caso de aplicações submarinas em águas profundas. De fato, mesmo que a geometria do espaço de trabalho seja previamente conhecida, deve-se determinar a configuração do robô com respeito à estrutura submarina de interesse, a fim de permitir a execução de tarefas sobre a mesma com eficiência e precisão. Entretanto, em ambientes remotos pouco estruturados é difícil instalar dispositivos de medição externos ao robô, sobretudo em águas profundas. Então, a calibração deve, preferencialmente, ser realizada apenas com os sensores do próprio robô. Este problema não é de solução simples e tem sido objeto de pesquisa na área da robótica.

Neste trabalho, considera-se o problema de calibração remota e replanejamento de trajetórias para um robô industrial. Um método de calibração é apresentado para estimar a incerteza na configuração do robô a partir de uma grade de calibração conhecida usando sensores internos (encoders ou resolvers) e externos (câmeras). Resultados de simulação, obtidos a partir do modelo matemático de um robô manipulador de 6 graus de liberdade, ilustram a viabilidade do esquema de controle proposto. Uma ferramenta de realidade virtual será desenvolvida para visualizar e simular em laboratório a execução e o planejamento de tarefas em ambientes hostis.

Palavras-Chave: Calibração de robô, planejamento de trajetória, manipuladores robóticos.

ABSTRACT

In Advanced robotic systems, capable of operating in hostile places and difficult to access, planning a trajectory of reference is usually performed on the surface or in a structured and safe environment. However, when the robotic cell is inserted in the workplace can present uncertainties in its absolute location in relation to the original location adopted, characterizing themselves in an issue of the trajectory replanning. In this context, the replanning can be done from the estimate of error of configuration (position and orientation) among the sites already mentioned, using algorithms for calibration, and the ideal trajectory previously planned.

The calibration is an essential activity when it is necessary to use a robotic system to operate in remote environments, such as underwater applications in deep water. In fact, even when the geometry of the area of work is previously known, you should determine the configuration of the robot with respect to the underwater structure of interest, to allow the execution of work on it with efficiency and precision. However, in remote environments little structured is difficult to install the measuring devices outside the robot, especially in deep water. Then the calibration should preferably be conducted only with the sensors of the robot. This problem is not easy to solve and has been the object of research in the field of robotics.

In this work, it is considered the problem of remote calibration and trajectory for a robot industry. A method of calibration is presented to estimate the uncertainty in the configuration of the robot from a grid of known calibration using internal sensors (encoders or resolvers) and external (cameras). Results of simulation, obtained from the mathematical model of a robot manipulator of 6 degrees of freedom, illustrate the viability of the proposed scheme of control. A virtual reality tool will be developed to visualize and simulate in the lab planning and execution of tasks in hostile environments.

Keywords— Trajectory planning, Robot calibration, Robotic manipulators.

Sumário

Capítulo 1

Introdução

1.1 – Tema

Neste trabalho, considera-se o problema de calibração remota e replanejamento de trajetórias para um robô industrial em ambientes hostis.

Considerando a diferença entre o posicionamento do robô ao planejar e ao executar a tarefa, um método de calibração é apresentado para estimar a incerteza na configuração do robô a partir de uma grade de calibração conhecida usando sensores internos (encoders ou resolvers) e externos (câmeras). Uma estratégia de replanejamento de trajetórias no espaço de velocidades cartesianas é apresentada para solucionar o problema de planejamento de trajetória e redundância cinemática. Resultados de simulação, obtidos a partir do modelo matemático de um robô manipulador de 6 graus de liberdade, ilustram a viabilidade do esquema de controle proposto. Uma ferramenta de realidade virtual será desenvolvida para visualizar e simular em laboratório a execução e o planejamento de tarefas em ambientes hostis.

1.2 – Delimitação

Para a calibração considera-se que o robô seja capaz de tocar num número mínimo de três pontos da grade de calibração e que os pontos tocados tanto na base de planejamento quanto na base de execução da tarefa sejam tocados na mesma ordem.

O ambiente de realidade virtual será desenvolvido baseado no modelo de um robô de 6 graus de liberdade de juntas de revolução.

1.3 – Localização

Há tarefas a serem consideradas, em lugares em que a ação do homem é difícil e arriscada, como em unidades de sistemas submarinos de extração de petróleo em águas profundas ou em atividades de missões espaciais. Para executá-las, faz-se necessária a presença de máquinas teleoperadas (robôs), que as realizam, contornando os riscos que um ser humano poderia sofrer. Nesse contexto está inserido o tema deste trabalho.

1.4 - Justificativa

Em sistemas robóticos avançados, capazes de operar em lugares hostis e de difícil acesso, o planejamento de uma trajetória de referência geralmente é realizado na superfície ou em um ambiente seguro e estruturado. Entretanto, quando a célula robótica é inserida no ambiente de trabalho pode apresentar incertezas em sua localização absoluta em relação à localização original adotada, caracterizando-se em um problema de replanejamento de trajetória. Neste contexto, o replanejamento pode ser realizado a partir da estimativa do erro de configuração (posição e orientação) entre as localizações já citadas, utilizando algoritmos de calibração, bem como a trajetória ideal previamente planejada.

A calibração é uma atividade essencial quando é necessário utilizar um sistema robótico para operar em ambientes remotos, como é o caso de aplicações submarinas em águas profundas. De fato, mesmo que a geometria do espaço de trabalho seja previamente conhecida, deve-se determinar a configuração do robô com respeito à estrutura submarina de interesse, a fim de permitir a execução de tarefas sobre a mesma com eficiência e precisão.

1.5 - Objetivo

O objetivo geral é, então, propor um método para replanejamento de trajetória e um modelo para calibração utilizando uma grade de pontos para um robô manipulador industrial 6 DOF (degrees of freedom) de juntas de revolução. Uma ferramenta de realidade virtual será desenvolvida para visualizar e simular em laboratório a execução e o planejamento de tarefas em ambientes hostis.

1.6 - Metodologia

Para aquisição dos pontos será utilizado o robô Zebra Zero que deverá tocar os pontos da grade de calibração em várias configurações diferentes, variando-se a posição e rotação da sua base.

Para calibração serão utilizados conceitos de robótica, tais como: transformação homogênea, ângulos de Euler ZYZ, Quaternions unitário, técnicas de mínimos quadrados.

Para replanejamento serão utilizados algoritmos de cinemática inversa e técnicas de replanejamento baseadas na matriz transformação obtida na fase de calibração.

1.7 – Descrição

O capítulo 2 apresenta vários conceitos importantes para o planejamento de trajetórias, abordando transformação homogênea, ângulos de *Euler ZYZ*, quaternion unitário e parâmetros de *Denavit-Hartenberg*.

No capítulo 3 será abordado o problema de calibração utilizando sensores internos e externos, incluindo metodologias e resultados de simulação.

A questão de replanejamento de trajetórias será abordada no capítulo 4, que constará de uma parte de exposição do método proposto e em seguida de simulações realizadas.

O capítulo 5 abordará o ambiente de realidade virtual, que será utilizado para visualizar a execução e o resultado das etapas anteriores.

Finalmente, o capítulo 6 será formado pela conclusão do trabalho, explicitando os resultados obtidos durante as diversas fases do projeto.

Capítulo 2

Planejamento de trajetória

O planejamento de trajetórias consiste em gerar entradas de referência que assegurem que o manipulador robótico execute a tarefa planejada. Neste trabalho, será considerado que o robô, ao ser inserido no ambiente de execução da tarefa, poderá apresentar incerteza entre a posição onde será planejada a trajetória e a sua localização de execução da tarefa, logo sua trajetória deverá ser replanejada. Com isso, a execução da tarefa deverá seguir conforme o planejado.

Serão utilizados alguns conceitos para se determinar o planejamento de trajetórias, tais como: transformação homogênea, ângulos de Euler ZYZ e etc, que serão mostrados a seguir para um bom entendimento das ferramentas utilizadas na solução do problema de Planejamento de trajetórias.

2.1 - Transformação Homogênea

A transformação homogênea expressa a mudança de representação de vetores entre dois sistemas de coordenadas. A figura 2.1 representa um ponto P no espaço, o vetor p_0 de coordenadas deste ponto P em relação ao sistema de coordenadas de referência E_0 , o vetor p_1 de coordenadas do mesmo ponto P em relação ao sistema de coordenadas de referência E_1 e o vetor O_{01} representa o vetor entre as origens dos sistemas de coordenadas E_0 e E_1 .

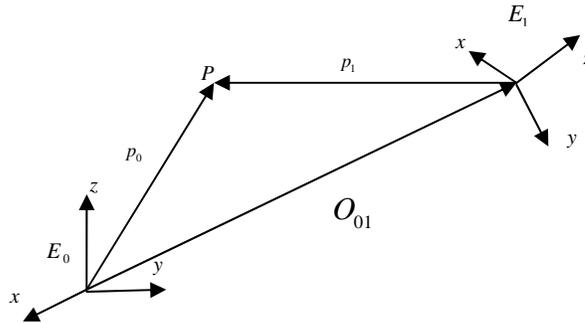


Figura 2.1 - Representação do ponto P nos sistemas de coordenadas E_0 e E_1 .

Conforme podemos verificar na figura acima:

$$p_0 = O_{01} + R_{01}p_1 \quad (2.1)$$

onde R_{01} representa a matriz de rotação de E_1 em relação a E_0 .

A relação descrita em (2.1) representa a transformação homogênea de coordenadas (translação e rotação) de um vetor entre os sistemas de coordenadas E_0 e E_1 .

Para uma representação compacta da relação entre os sistemas de coordenadas em dois sistemas de coordenadas diferentes, é válido introduzir uma quarta componente unitária no vetor do ponto p , resultando em \tilde{p} , conforme abaixo:

$$\tilde{p} = \begin{bmatrix} p \\ 1 \end{bmatrix} \quad (2.2)$$

Fazendo-se uso dessa representação, a equação descrita em (2.1) pode ser reescrita como:

$$\tilde{p}_0 = \begin{bmatrix} R_{01} & O_{01} \\ 0 & 1 \end{bmatrix} \tilde{p}_1 \quad (2.3)$$

ou da forma:

$$\tilde{p}_0 = T_{01}\tilde{p}_1 \quad (2.4)$$

onde T_{01} representa a matriz de transformação homogênea de um ponto no sistema de coordenadas E_1 para o sistema de coordenadas E_0 .

2.2 - Ângulos de Euler ZYZ

A representação mínima da orientação é constituída por três parâmetros independentes. Essa representação mínima pode ser obtida utilizando um conjunto de três ângulos de rotação $\Phi = [\varphi, \vartheta, \psi]^T$, considerando que a matriz em questão é expressa por rotações elementares e sucessivas como função de um único ângulo.

A rotação a partir da utilização do método de ângulos de *Euler ZYZ* é obtida da seguinte forma: inicialmente efetua-se uma rotação de φ em torno do eixo z, após é realizada uma rotação de ϑ em torno do eixo y obtido após a primeira rotação, e por último deve-se realizar uma rotação de ψ sobre o eixo z obtido após as duas rotações anteriores.

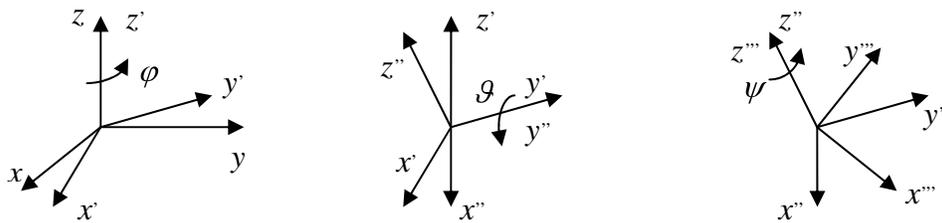


Figura 2.2 – Representação dos ângulos de *Euler ZYZ*

A composição das rotações pode ser representada por:

$$R(\Phi) = R_z(\varphi)R_y(\vartheta)R_z(\psi) \quad (2.5)$$

equivalentemente:

$$R(\Phi) = \begin{bmatrix} \cos \varphi \cos \vartheta \cos \psi - \text{sen} \varphi \text{sen} \psi & -\cos \varphi \cos \vartheta \text{sen} \psi - \text{sen} \varphi \cos \psi & \cos \varphi \text{sen} \vartheta \\ \text{sen} \varphi \cos \vartheta \cos \psi + \cos \varphi \text{sen} \psi & -\text{sen} \varphi \cos \vartheta \text{sen} \psi + \cos \varphi \cos \psi & \text{sen} \varphi \text{sen} \vartheta \\ -\text{sen} \vartheta \cos \varphi & \text{sen} \vartheta \text{sen} \psi & \cos \vartheta \end{bmatrix} \quad (2.6)$$

De uma forma simplificada, tem-se:

$$R(\Phi) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.7)$$

Os ângulos de rotação φ , ϑ e ψ podem ser obtidos a partir das seguintes relações dos termos da matriz de rotação $R(\Phi)$:

$$\varphi = \text{atan2}(r_{23}, r_{13}) \quad (2.8)$$

$$\vartheta = \text{atan2}(\sqrt{r_{13}^2 + r_{23}^2}, r_{33}) \quad (2.9)$$

$$\psi = \text{atan2}(r_{32}, -r_{31}) \quad (2.10)$$

Observa-se que se $r_{23} = r_{13} = 0$ há presença de singularidades, justamente o que acontece quando se tem uma rotação no eixo z representada pelos três ângulos de rotação. Sua representação será representada pelos ângulos φ e ψ , onde $\varphi + \psi = \Phi$. Como se pode verificar na forma da matriz $R(\Phi)$ quando se faz $\vartheta = 0$:

$$R(\Phi) = \begin{bmatrix} \cos \varphi \cos \psi - \text{sen} \varphi \text{sen} \psi & -\cos \varphi \text{sen} \psi - \text{sen} \varphi \cos \psi & 0 \\ \text{sen} \varphi \cos \psi + \cos \varphi \text{sen} \psi & -\text{sen} \varphi \text{sen} \psi + \cos \varphi \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

o que é igual a:

$$R(\Phi) = \begin{bmatrix} \cos(\varphi + \psi) & -\text{sen}(\varphi + \psi) & 0 \\ \text{sen}(\varphi + \psi) & \cos(\varphi + \psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} = R_z(\varphi + \psi) \quad (2.12)$$

A rotação em z, como se pode ver em (2.12), será representada por todas as combinações de soma das rotações em z representados pelos ângulos φ e ψ .

2.3 - Quaternion unitário

Existe uma outra forma muito comum de se representar orientação. Essa maneira chama-se quaternion unitário, que utiliza quatro parâmetros ao invés de três, conforme são utilizados na representação por ângulos de *Euler ZYZ*. Essa representação possui uma grande vantagem, pois permite a obtenção de soluções únicas, reduzindo assim o número de manipulações computacionais.

O quaternion unitário $Q(q_0, q_v)$ é representado por uma parte escalar q_0 e uma parte vetorial $q_v = [q_{v_x}, q_{v_y}, q_{v_z}]^T$. Para a formação dos parâmetros do quaternion unitário precisa-se definir um vetor unitário arbitrário $v = [v_x, v_y, v_z]^T$, que através do qual será feita uma rotação σ , como se pode observar na figura abaixo:

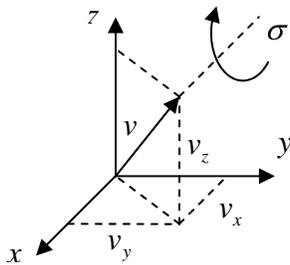


Figura 2.3 – representação da rotação σ sobre o vetor unitário v .

A parte escalar q_0 e vetorial q_v do quaternion unitário são definidas como:

$$q_0 = \cos \frac{\sigma}{2} \quad (2.13)$$

$$q_v = \text{sen} \frac{\sigma}{2} v \quad (2.14)$$

e estão sujeitas à seguinte restrição:

$$q_0^2 + q_{v_x}^2 + q_{v_y}^2 + q_{v_z}^2 = 1 \quad (2.15)$$

pois $q_0^2 = \cos^2\left(\frac{\sigma}{2}\right)$, $q_{v_x}^2 = \text{sen}^2\left(\frac{\sigma}{2}\right)v_x^2$, $q_{v_y}^2 = \text{sen}^2\left(\frac{\sigma}{2}\right)v_y^2$ e $q_{v_z}^2 = \text{sen}^2\left(\frac{\sigma}{2}\right)v_z^2$

assim, somando-se os quatro termos acima, tem-se:

$$q_0^2 + q_{v_x}^2 + q_{v_y}^2 + q_{v_z}^2 = \cos^2\left(\frac{\sigma}{2}\right) + \text{sen}^2\left(\frac{\sigma}{2}\right)[v_x^2 + v_y^2 + v_z^2] \quad (2.16)$$

Usando-se a relação trigonométrica fundamental e o fato de que o vetor v é unitário e por isso possui norma igual a um, comprova-se a restrição descrita em (2.15).

Em vista de (2.13), (2.14) e (2.15), a matriz de rotação do quaternion unitário assume a forma a seguir:

$$R(q_0, q_v) = \begin{bmatrix} 2(q_0^2 + q_{v_x}^2) - 1 & 2(q_{v_x}q_{v_y} - q_0q_{v_z}) & 2(q_{v_x}q_{v_z} + q_0q_{v_y}) \\ 2(q_{v_x}q_{v_y} + q_0q_{v_z}) & 2(q_0^2 + q_{v_y}^2) - 1 & 2(q_{v_y}q_{v_z} + q_0q_{v_x}) \\ 2(q_{v_x}q_{v_z} + q_0q_{v_y}) & 2(q_{v_y}q_{v_z} + q_0q_{v_x}) & 2(q_0^2 + q_{v_z}^2) - 1 \end{bmatrix} \quad (2.17)$$

ou de uma forma reduzida:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.18)$$

Logo pode-se escrever q_0 e q_v em função dos valores da matriz de rotação R descrita em (2.18):

$$q_0 = \frac{1}{2} \sqrt{r_{11} + r_{22} + r_{33} + 1} \quad (2.19)$$

$$q_v = \begin{bmatrix} \operatorname{sgn}(r_{32} - r_{23}) \sqrt{r_{11} - r_{22} - r_{33} + 1} \\ \operatorname{sgn}(r_{13} - r_{31}) \sqrt{r_{22} - r_{33} - r_{11} + 1} \\ \operatorname{sgn}(r_{21} - r_{12}) \sqrt{r_{33} - r_{11} - r_{22} + 1} \end{bmatrix} \quad (2.20)$$

onde $\operatorname{sgn}(x)$ é definida da seguinte forma:

$$\operatorname{sgn}(x) = \begin{cases} 1, & \text{para } x \geq 0 \\ 0, & \text{para } x < 0 \end{cases}$$

Note que é assumido em (2.19) que $q_0 \geq 0$, isto implica na restrição do domínio do ângulo $\sigma \in [-\pi, \pi]$. Percebe-se então a partir de (2.19) e (2.20) que as soluções para q_0 e q_v estão livres de singularidades.

O quaternion unitário que é extraído a partir de $R^{-1} = R^T$ é representado por $Q^{-1} = (q_0, -q_v)$.

O produto de quaternions é definido pelo operador $*$, sendo dado por:

$$Q_1 * Q_2 = (q_1 q_2 - q_{v_1}^T q_{v_2}, q_1 q_{v_2} + q_2 q_{v_1} + q_{v_1} \times q_{v_2}) \quad (2.21)$$

onde $Q_1 = (q_1, q_{v_1})$ e $Q_2 = (q_2, q_{v_2})$ são os quaternions correspondentes às matrizes de rotação R_1 e R_2 , respectivamente. Note que, a partir de (2.21), se $Q_2 = Q_1^{-1}$, tem-se que:

$$Q_1 * Q_1^{-1} = (1, [0 \ 0 \ 0]) \quad (2.22)$$

2.4 - Parâmetros de Denavit-Hartenberg

A convenção dos parâmetros adotada por *Denavit-Hartenberg* estabelece que a transformação homogênea $T_{i-1,i}$ entre os sistemas de coordenadas E_{i-1} e E_i referentes a dois elos consecutivos, através de uma cadeia cinemática do manipulador, composto de elos rígidos, separados por uma junta, pode ser escrita por até quatro matrizes de transformações homogêneas.

A notação de *Denavit-Hartenberg* baseia-se no fato de que para determinar a posição relativa de duas retas no espaço são necessários somente dois parâmetros. O primeiro parâmetro, representado pela letra a , é a distância medida ao longo da normal comum entre as duas retas e o segundo, representado pela letra grega α , é o ângulo de rotação em torno da normal comum, sobre a qual uma das retas deve girar para que fique paralela à outra.

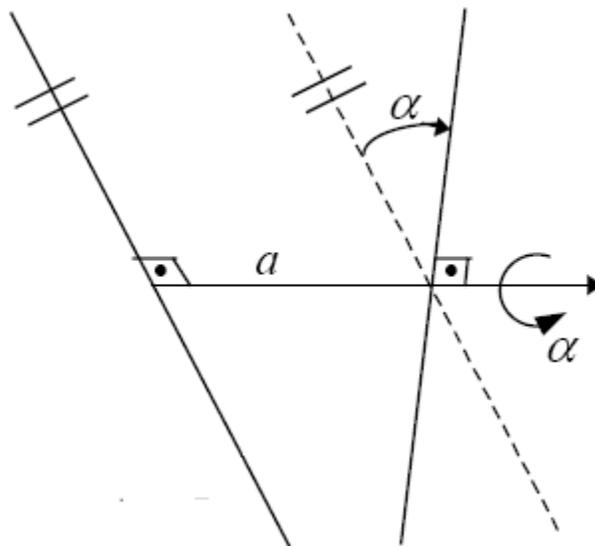


Figura 2.4 – Representação dos parâmetros a e α .

Se para definir a posição relativa de duas retas no espaço são necessários dois parâmetros, então para definir a posição relativa de dois sistemas de coordenadas tridimensionais são necessários quatro parâmetros. Isto decorre do fato de que um sistema de coordenadas é definido por três eixos (retas), sendo que conhecendo-se dois deles o terceiro estará definido pelas regras de ortogonalidade e da mão

direita. Portanto, a partir da definição de posição relativa entre dois eixos de dois sistemas de coordenadas, pode-se descrever a posição relativa entre dois sistemas de coordenadas.

Agora será demonstrada uma forma de representação de um sistema de coordenadas E_{i-1} a partir de E_i .

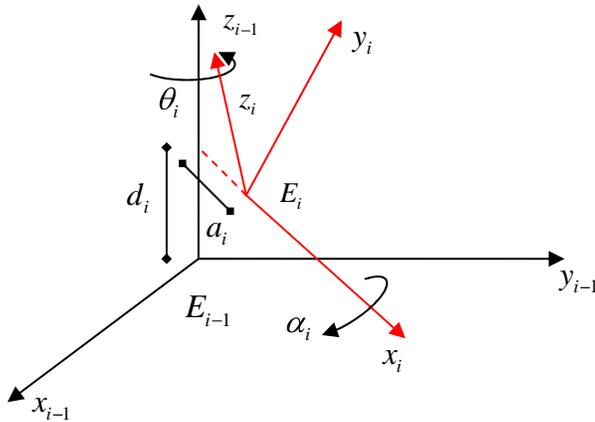


Figura 2.5 – Representação do sistema de coordenadas E_{i-1} a partir de E_i .

Primeiramente consideram-se os dois sistemas de coordenadas como coincidentes:

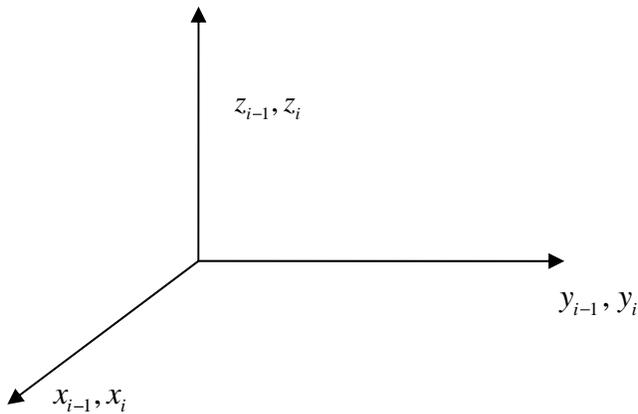


Figura 2.6 – Representação dos sistemas de coordenadas E_{i-1} e E_i de forma coincidente.

Logo, a matriz de transformação homogênea que relaciona esses dois sistemas de coordenadas é:

$$T_{i-1 i} = I_{4 \times 4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.23)$$

Em seguida, deve-se rotacionar o sistema E_i de θ_i em relação ao eixo Z_{i-1} .

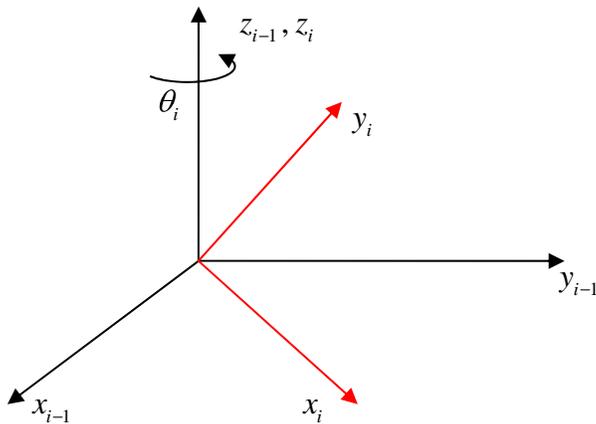


Figura 2.7 – Sistema de coordenada E_i rotacionado de θ_i em relação ao eixo Z_{i-1} .

A matriz transformação ficará da forma:

$$T_{i-1 i} = \begin{bmatrix} \cos \theta_i & -\text{sen} \theta_i & 0 & 0 \\ \text{sen} \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.24)$$

A seguir, deve-se transladar o sistema de coordenadas E_i em d_i unidades ao longo do eixo z_i :

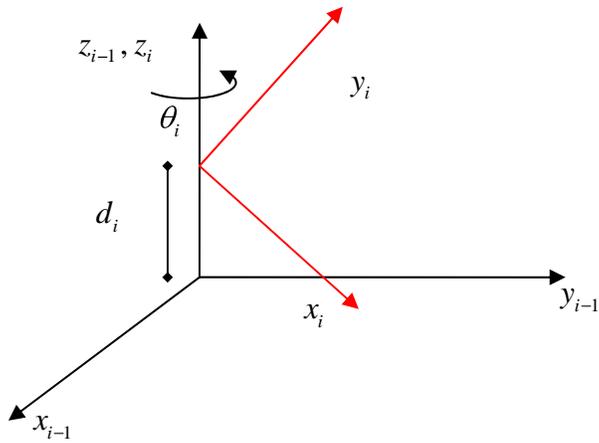


Figura 2.8 – Sistema de coordenada E_i transladado de d_i unidades ao longo do eixo z_i .

Com isso, a matriz transformação ficará da forma:

$$T_{i-1 i} = \begin{bmatrix} \cos \theta_i & -\text{sen} \theta_i & 0 & 0 \\ \text{sen} \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.25)$$

Novamente deve-se fazer uma translação, mas desta vez ao longo do eixo x_i de a_i unidades :

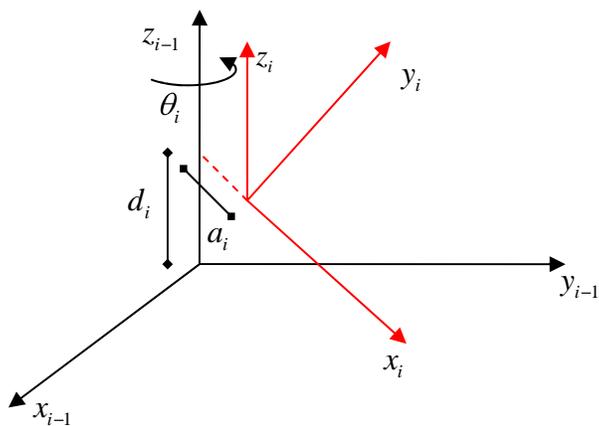


Figura 2.9 – Sistema de coordenada E_i transladado de a_i unidades ao longo do eixo x_i .

Desta forma, a matriz transformação se tornará:

$$T_{i-1 i} = \begin{bmatrix} \cos \theta_i & -\text{sen} \theta_i & 0 & a_i \cos \theta_i \\ \text{sen} \theta_i & \cos \theta_i & 0 & a_i \text{sen} \theta_i \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.26)$$

Por último, deve-se fazer uma rotação de α_i em torno do eixo x_i :

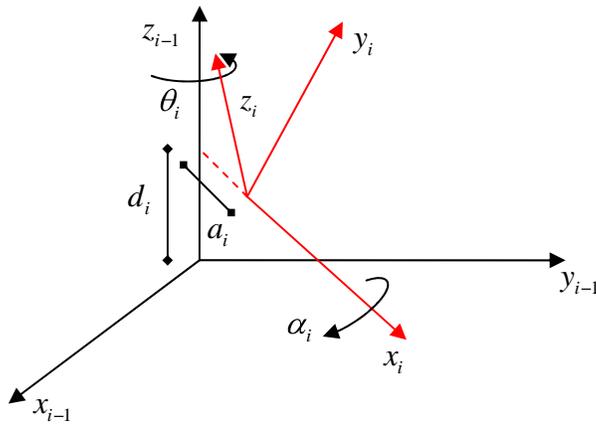


Figura 2.10 – Sistema de coordenada E_i rotacionado de α_i em torno do eixo x_i .

Desta forma, a matriz transformação se tornará:

$$T_{i-1 i} = \begin{bmatrix} \cos \theta_i & -\text{sen} \theta_i & 0 & 0 \\ \text{sen} \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\text{sen} \alpha_i & 0 \\ 0 & \text{sen} \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.27)$$

que é igual a:

$$T_{i-1 i} = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \text{sen} \theta_i & \text{sen} \alpha_i & a_i \cos \theta_i \\ \text{sen} \theta_i & \cos \alpha_i \cos \theta_i & -\text{sen} \alpha_i \cos \theta_i & a_i \text{sen} \theta_i \\ 0 & \text{sen} \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.28)$$

A partir dessas operações, observa-se que a posição relativa entre dois sistemas de coordenadas consecutivos é completamente determinada pelas posições relativas entre os eixos x_{i-1} e x_i e entre os eixos z_{i-1} e z_i , que foram definidas pelos quatro parâmetros:

θ_i : é o ângulo entre os eixos x_{i-1} e x_i , medido em torno do eixo z_{i-1} , segundo a regra da mão direita.

a_i : é a distância entre z_{i-1} e z_i , medida ao longo do eixo x_i , que é a normal comum entre z_{i-1} e z_i ;

α_i : é o ângulo entre o eixo z_{i-1} e z_i , medido em torno do eixo x_i , segundo a regra da mão direita.

d_i : é a distância entre os eixos x_{i-1} e x_i , medida sobre o eixo z_{i-1} .

Esses quatro parâmetros são então definidos como parâmetros de *Denavit-Hartenberg* e são utilizados em robótica para representar a matriz transformação de um elo do robô em relação ao elo anterior.

Para representação de um manipulador, os parâmetros a_i e α_i de cada elo são constantes. Note que se a junta entre os elos for prismática, tem-se que d_i é variável enquanto θ_i é constante, mas se a junta for de revolução, tem-se que θ_i é variável enquanto d_i é constante.

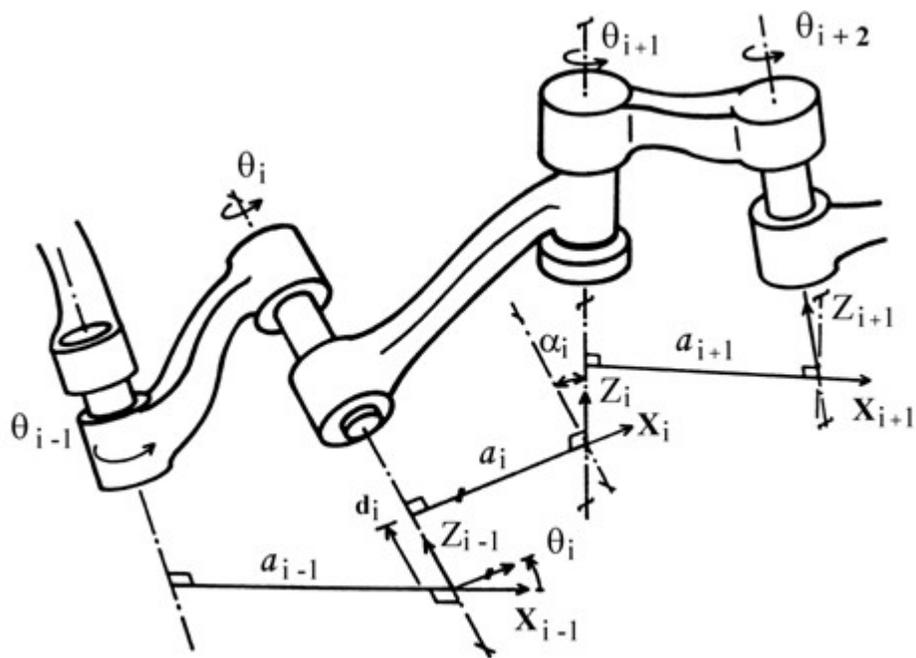


Figura 2.11 – Representação dos parâmetros de *Denavit-Hartenberg* para diversas juntas.

Capítulo 3

Calibração

A calibração pode ser entendida como a obtenção de uma lei matemática para compensação dos erros em um manipulador robótico como efeito de correção dos desvios de sua posição e orientação.

A necessidade da utilização de métodos de calibração neste trabalho é devido à incerteza na localização absoluta do manipulador robótico em relação à sua localização original assumida para solucionar o problema de planejamento de trajetória.

Neste trabalho considera-se essa incerteza como desconhecida, sendo proposto para sua estimativa a utilização de um algoritmo baseado no método de mínimos quadrados, que utiliza informações de sensores internos (encoders) e externos (câmera de vídeo). Em vista disso, são abordados dois métodos de calibração para obtenção do erro de posicionamento entre os sistemas de coordenadas original E_b e atual E_{b1} para a base do manipulador robótico.

Primeiro método de calibração

No primeiro método de calibração utilizam-se sensores internos (*encoders*) para obtenção das medidas de posicionamento do efetuador do robô, que consiste em medir os pontos da grade de calibração na localização considerada como original E_b e, após isso, assumindo que houve um deslocamento em relação ao sistema de coordenadas inercial, deve-se medir os pontos da grade de calibração em relação ao novo sistema de coordenadas da base, considerada como atual E_{b1} . Após a determinação das medidas dos pontos de calibração da grade em relação aos dois sistemas de coordenadas da base, estima-se o erro de posicionamento através de um algoritmo de mínimos quadrados.

Segundo método de calibração

No segundo método utilizam-se sensores externos, no caso uma câmera de vídeo fixa no espaço de trabalho do manipulador. Vale ressaltar que este método é complementar ao primeiro, pois os pontos de calibração medidos com respeito às localizações original e atual são obtidos do primeiro método, contudo, esses pontos agora são vistos no ambiente da câmera. Devido à necessidade que existe em ambientes remotos e hostis, onde a ação do homem é complicada e perigosa ou até mesmo impossível, o uso desse método torna-se de extrema importância, já que assim pode-se visualizar a execução da tarefa.

3.1 - Estimação do erro de posicionamento com sensor interno

Conforme mencionado, a estimação do erro de posicionamento com sensor interno é obtida com a medição das posições dos pontos da grade de calibração em relação a sua localização original E_b e sua localização atual E_{b1} . Após isso, calcula-se a incerteza de posicionamento entre as duas localizações utilizando um algoritmo baseado em mínimos quadrados, essa incerteza é representada através da transformação homogênea T_{bb1} . O número mínimo de pontos necessários para realizar a calibração são três, pois estamos levando em consideração apenas o erro de posição. Entretanto, como as medidas dos sensores apresentam ruído, é medido um número de pontos de calibração superior para uma melhor exatidão na obtenção dos parâmetros do erro de posição.

A parte translacional da matriz de transformação homogênea T_{bb1} é composta pelo vetor posição: $\left[P_{bb1_x} P_{bb1_y} P_{bb1_z} \right]^T$.

A parte rotacional da matriz T_{bb1} utiliza a representação ZYZ dos ângulos de Euler. Sendo assim, ela é dada por uma seqüência de três rotações em torno dos eixos z, y, e z novamente e expressa pelo produto $R(\Phi) = R_z(\varphi)R_y(\vartheta)R_z(\psi)$. Logo, os parâmetros do erro de posicionamento P_{bb1_x} , P_{bb1_y} , P_{bb1_z} , φ, ϑ, ψ são funções do sistema geométrico, representado pela notação vetorial $x_{bb1} = \left[P_{bb1} \Phi_{bb1} \right]^T$, onde:

$$P_{bb1} = \begin{bmatrix} P_{bb1_x} & P_{bb1_y} & P_{bb1_z} \end{bmatrix}^T \quad (3.1)$$

$$\Phi_{bb1} = \begin{bmatrix} \varphi & \mathcal{G} & \psi \end{bmatrix}^T \quad (3.2)$$

Para se calcular a matriz x_{bb1} deve-se tomar então as medidas dos pontos de calibração $\lambda_1, \lambda_2, \dots, \lambda_n$ da grade nas duas bases, onde n representa o número de pontos a serem tocados. Primeiramente, o manipulador deve estar na base original E_b , onde tocará os pontos da grade obtendo o conjunto de pontos $P_{b\lambda_1}, P_{b\lambda_2}, \dots, P_{b\lambda_n} \in \mathbb{R}^3$. Logo após, deve ser efetuado um deslocamento na sua configuração de posicionamento, encontrando-se então na base atual E_b tocará os pontos da grade novamente obtendo o conjunto de pontos: $P_{b1\lambda_1}, P_{b1\lambda_2}, \dots, P_{b1\lambda_n} \in \mathbb{R}^3$. Basta então resolver o sistema de equações para determinar P_{bb1} e $R_{bb1} \in \mathbb{R}^{3 \times 3}$, matriz rotação entre E_b e E_{b1} :

$$\begin{aligned} P_{b\lambda_1} &= P_{bb1} + R_{bb1} P_{b1\lambda_1}, \\ P_{b\lambda_2} &= P_{bb1} + R_{bb1} P_{b1\lambda_2}, \\ &\cdot \\ &\cdot \\ &\cdot \\ P_{b\lambda_n} &= P_{bb1} + R_{bb1} P_{b1\lambda_n} \end{aligned} \quad (3.3)$$

Note que a representação da posição e orientação dos pontos de calibração através da matriz transformação T_{bb1} pode ser dada por:

$$T_{bb1} = T_{b\lambda_i} T_{b1\lambda_i}^{-1}, \quad i = 1, 2, \dots, n \quad (3.4)$$

Como se deseja encontrar T_{bb1} utilizando um algoritmo baseado em mínimos quadrados, deve-se definir uma função objetivo $f(x)$ que descreva o sistema em termos dos seus parâmetros de posição e orientação. O problema de minimização consiste em minimizar a função objetivo $f(x)$. A função objetivo $\in \mathbb{R}^{3n}$ é dada por:

$$f = \begin{bmatrix} P_{b\lambda 1} - P_{bb1} - R_{bb1} P_{b1\lambda 1} \\ P_{b\lambda 2} - P_{bb1} - R_{bb1} P_{b1\lambda 2} \\ \vdots \\ P_{b\lambda n} - P_{bb1} - R_{bb1} P_{b1\lambda n} \end{bmatrix} \quad (3.5)$$

Para minimizar essa função deve-se utilizar o seu Jacobiano, que são as derivadas parciais de primeira ordem em relação aos parâmetros P_{bb1} e Φ_{bb1} , ou seja:

$$J = \nabla f = \begin{bmatrix} \frac{\partial f}{\partial P_{bb1}} & \frac{\partial f}{\partial \Phi_{bb1}} \end{bmatrix}^T \quad (3.6)$$

Note que esse jacobiano apresenta uma forma que simplifica os cálculos, pois $\frac{\partial f}{\partial P_{bb1}} = I_{3 \times 3}$.

Portanto, basta calcular o Jacobiano de f em relação a Φ_{bb1} :

$$\frac{\partial f}{\partial \Phi_{bb1}} = \begin{bmatrix} -\frac{\partial R_{bb1}}{\partial \varphi} P_{b1\lambda 1} & -\frac{\partial R_{bb1}}{\partial \vartheta} P_{b1\lambda 1} & -\frac{\partial R_{bb1}}{\partial \psi} P_{b1\lambda 1} \\ \vdots & \vdots & \vdots \\ -\frac{\partial R_{bb1}}{\partial \varphi} P_{b1\lambda n} & -\frac{\partial R_{bb1}}{\partial \vartheta} P_{b1\lambda n} & -\frac{\partial R_{bb1}}{\partial \psi} P_{b1\lambda n} \end{bmatrix} \quad (3.7)$$

onde $\frac{\partial R_{bb1}}{\partial \varphi}$, $\frac{\partial R_{bb1}}{\partial \vartheta}$, $\frac{\partial R_{bb1}}{\partial \psi} \in \mathbb{R}^{3 \times 3}$ e $\frac{\partial f}{\partial \Phi_{bb1}} \in \mathbb{R}^{3n \times 3}$.

Então a forma do jacobiano J é dada por:

$$\begin{bmatrix} -I_{3 \times 3} - \frac{\partial R_{bb1}}{\partial \varphi} P_{b1\lambda1} & -\frac{\partial R_{bb1}}{\partial \mathcal{G}} P_{b1\lambda1} & -\frac{\partial R_{bb1}}{\partial \psi} P_{b1\lambda1} \\ \vdots & \vdots & \vdots \\ -I_{3 \times 3} - \frac{\partial R_{bb1}}{\partial \varphi} P_{b1\lambda n} & -\frac{\partial R_{bb1}}{\partial \mathcal{G}} P_{b1\lambda n} & -\frac{\partial R_{bb1}}{\partial \psi} P_{b1\lambda n} \end{bmatrix} \quad (3.8)$$

Nota-se que $J \in \mathbb{R}^{3n \times 6}$ e que de acordo com o método de *Newton-Raphson* será invertida para determinar x_{bb1} , para esta operação utiliza-se a pseudo-inversa à esquerda, pois o posto da matriz J é dado por coluna. Logo a correção nos valores de x_{bb1} é realizada a cada iteração k , da seguinte forma:

$$x_{k+1} = x_k - \alpha \frac{f(k)}{f'(k)} \quad (3.9)$$

ou melhor:

$$x_{k+1} = x_k - \alpha J^\dagger(k) f(k) \quad (3.10)$$

onde $\alpha > 0$ e $J^\dagger(k) = (J^T(k)J(k))^{-1} J^T$ denota a pseudo-inversa da matriz J .

Existem alguns cuidados que devem ser tomados ao se utilizar o método de calibração proposto:

- A grade de calibração deve ser perfeitamente conhecida, pois a escolha de um ponto fora da área de trabalho do manipulador pode levar a divergências.
- A determinação de x_{bb1} utiliza o método de *Newton-Raphson* para determinar as soluções do sistema, esse método utiliza uma condição inicial que deve garantir a convergência para um mínimo global, pois se esta condição não for satisfeita a função não irá convergir para a solução.

3.1.1 - Simulação

Nesta seção apresentam-se os resultados das simulações utilizando o *Matlab* (Mathworks, Inc) para a calibração do manipulador utilizando sensores internos.

As simulações foram separadas em duas partes:

- A partir de pontos originados por um algoritmo, onde foram feitos dois tipos de simulação, uma sem ruído (caso ideal) e outra utilizando a função *rand(N)* do Matlab para gerar um ruído aleatório. Esta parte faz-se necessária para verificar a validade do algoritmo proposto nesse trabalho.
- A partir de pontos obtidos pelos sensores internos através de uma grade de calibração localizada no ambiente do laboratório. Nesta parte foi utilizado o robô *Zebra Zero* (IMI Inc.) para a tomada dos dados.

3.1.1.1 - Calibração utilizando pontos originados pelo algoritmo

3.1.1.1.1 - Simulação sem ruído

Para testar o correto funcionamento do algoritmo proposto, foi criada uma matriz de pontos no *Matlab* (vide tabela 3.1) para a base E_b . A partir dessa matriz utilizou-se uma matriz de erro de posicionamento x_{bb_1} conhecida (vide tabela 3.2) para assim se determinar os pontos na base E_{b_1} . Tendo-se agora as duas matrizes de pontos, tanto na base E_b quanto na base E_{b_1} , precisa-se voltar a determinar a matriz de erro de posicionamento utilizando o método proposto. Para verificar a sua validade, os parâmetros da matriz transformação obtida devem ser iguais aos parâmetros da matriz transformação conhecida utilizada para gerar os pontos na base E_{b_1} .

Ponto(mm)	X	y	z
λ_1	228,6	0	379,4
λ_2	211,2352	98,48078	379,4
λ_3	193,8704	196,9616	379,4
λ_4	228,6	0	279,4
λ_5	211,2352	98,48078	279,4
λ_6	193,8704	196,9616	279,4
λ_7	228,6	0	179,4
λ_8	211,2352	98,48078	179,4
λ_9	193,8704	196,9616	179,4
λ_{10}	193,8704	-196,962	379,4
λ_{11}	211,2352	-98,4808	379,4
λ_{12}	228,6	0	379,4
λ_{13}	193,8704	-196,962	279,4
λ_{14}	211,2352	-98,4808	279,4
λ_{15}	228,6	0	279,4
λ_{16}	193,8704	-196,962	179,4
λ_{17}	211,2352	-98,4808	179,4
λ_{18}	228,6	0	179,4

Tabela 3.1 - Pontos na base E_b

Pârametros de x_{bb1}	valor ideal	Valor estimado
P_{bb1_x} (mm)	50	50.000
P_{bb1_y} (mm)	100	100.000
P_{bb1_z} (mm)	150	150.000
φ (rad)	-0.7835	-0.7835
ϑ (rad)	0.1233	0.1233
ψ (rad)	0.7873	0.7873

Tabela 3.2 – Matriz de erro de posicionamento.

Como esperado, a matriz dos parâmetros de erro de posicionamento x_{bb1} obtida utilizando o algoritmo proposto apresentou valores idênticos aos da matriz de erro conhecida.

Pode-se verificar na figura 3.1 como os parâmetros de erro de posicionamento convergem para

seus respectivos valores a partir do 4º passo de iteração.

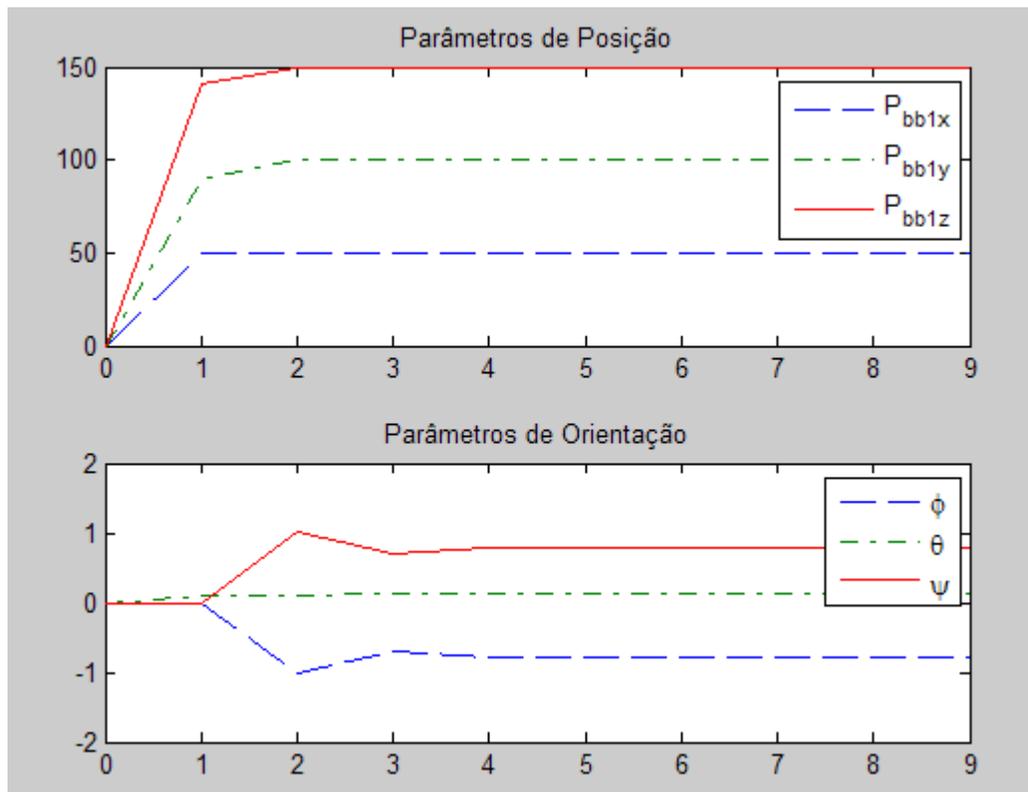


Figura 3.1 – Convergência dos parâmetros de x_{bb1}

3.1.1.1.2 - Simulação com ruído

Após verificada a validade do algoritmo, foram feitas simulações para medir a sua robustez ao ruído de tomada de dados, pois no mundo real a presença de ruído é inevitável.

Foi utilizada a função $rand(N)$ do *Matlab*, que retorna uma matriz quadrada de ordem N cujos termos possuem valores aleatórios entre 0 e 1, para adicionar ruído aos pontos da base E_{b1} .

Para esta simulação foi utilizado o mesmo método adotado na simulação sem ruído.

Parâmetros de x_{bb1}	valor ideal	valor estimado	erro absoluto
P_{bb1_x} (mm)	50	48.9967	1.0033
P_{bb1_y} (mm)	100	101.8268	1.8268
P_{bb1_z} (mm)	150	152.3185	2.3185
φ (rad)	-0.7835	-0.7530	0.0305
ϑ (rad)	0.1233	0.1332	0.0098
ψ (rad)	0.7873	0.7490	0.0383

Tabela 3.3 - Erro de posicionamento estimado com 18 pontos de calibração.

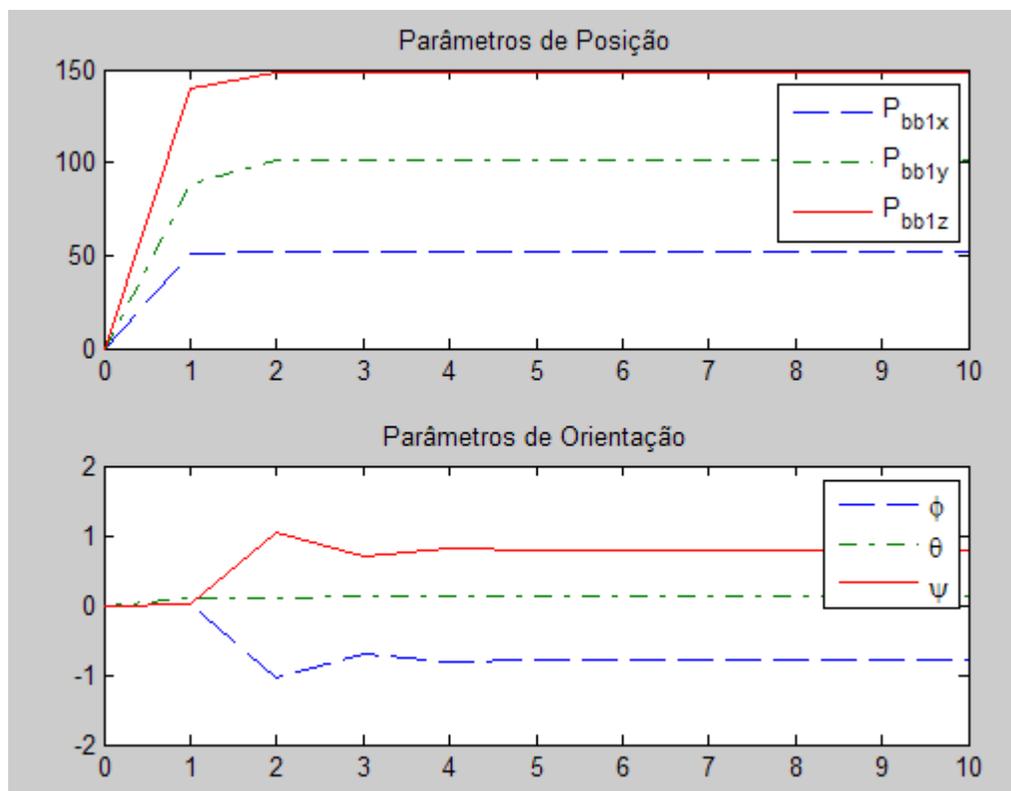


Figura 3.2 - Convergência dos parâmetros de x_{bb1} para N=18.

Parâmetros de x_{bb1}	valor ideal	valor estimado	erro absoluto
P_{bb1_x} (mm)	50	46.9038	3.0962
P_{bb1_y} (mm)	100	101.4816	1.4816
P_{bb1_z} (mm)	150	152.5121	2.5121
φ (rad)	-0.7835	-0.7375	0.0460
ϑ (rad)	0.1233	0.1422	0.0188
ψ (rad)	0.7873	0.7293	0.0580

Tabela 3.4 - Erro de posicionamento estimado com 14 pontos de calibração.

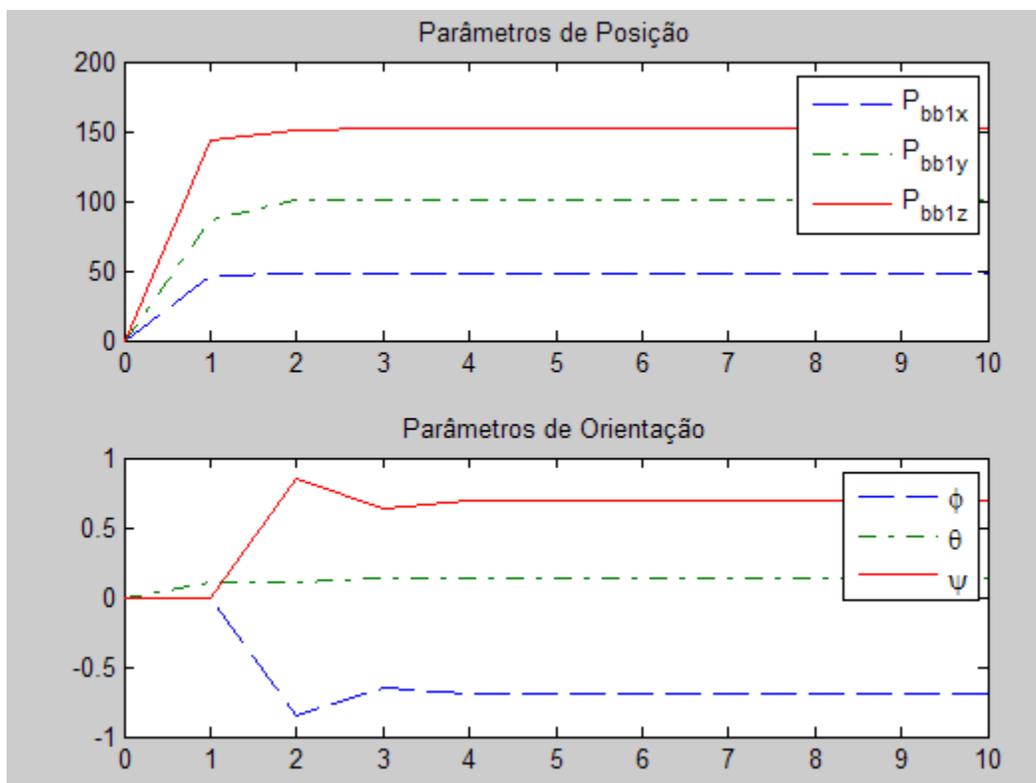


Figura 3.3 - Convergência dos parâmetros de x_{bb1} para N=14.

Parâmetros de x_{bb1}	valor ideal	valor estimado	erro absoluto
P_{bb1_x} (mm)	50	49.6399	1.3266
P_{bb1_y} (mm)	100	99.8571	0.2602
P_{bb1_z} (mm)	150	148.7975	1.6634
φ (rad)	-0.7835	-0.7481	0.0591
ϑ (rad)	0.1233	0.1221	0.0159
ψ (rad)	0.7873	0.7506	0.0568

Tabela 3.5 - Erro de posicionamento estimado com 10 pontos de calibração.

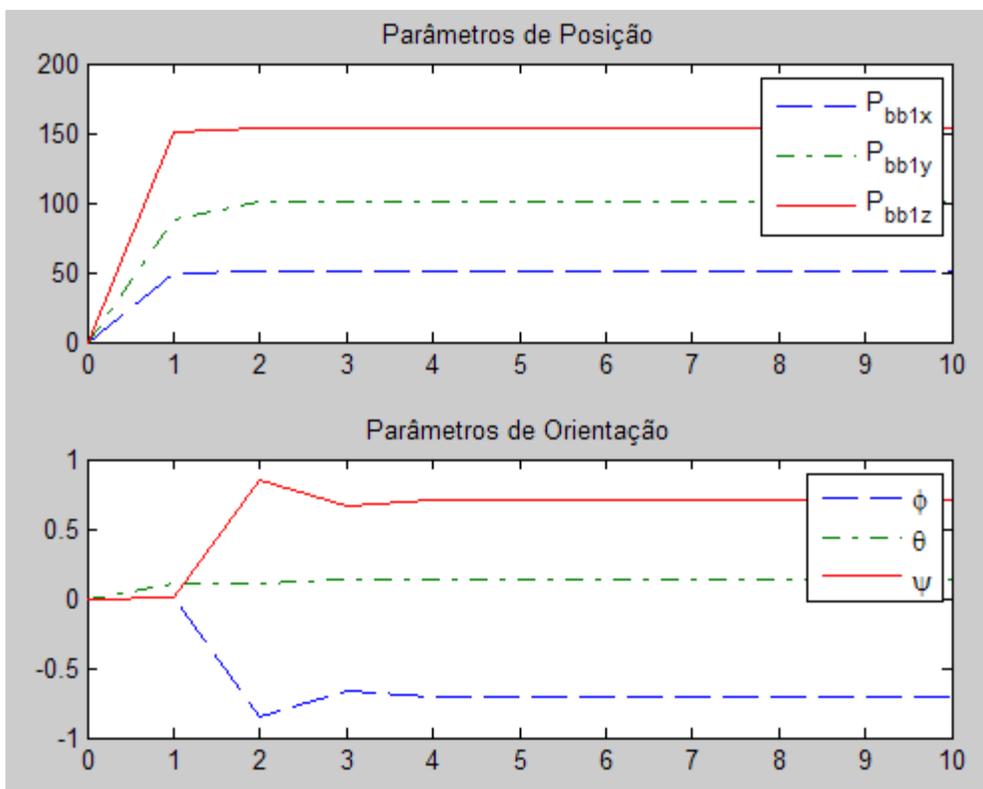


Figura 3.4 - Convergência dos parâmetros de x_{bb1} para N=10.

Parâmetros de x_{bb1}	valor ideal	valor estimado	erro absoluto
P_{bb1_x} (mm)	50	52.8508	2.8508
P_{bb1_y} (mm)	100	105.3172	5.3172
P_{bb1_z} (mm)	150	147.8371	2.1629
φ (rad)	-0.7835	-0.9774	0.1939
ϑ (rad)	0.1233	0.1186	0.0047
ψ (rad)	0.7873	0.9749	0.1876

Tabela 3.6 - Erro de posicionamento estimado com 5 pontos de calibração.

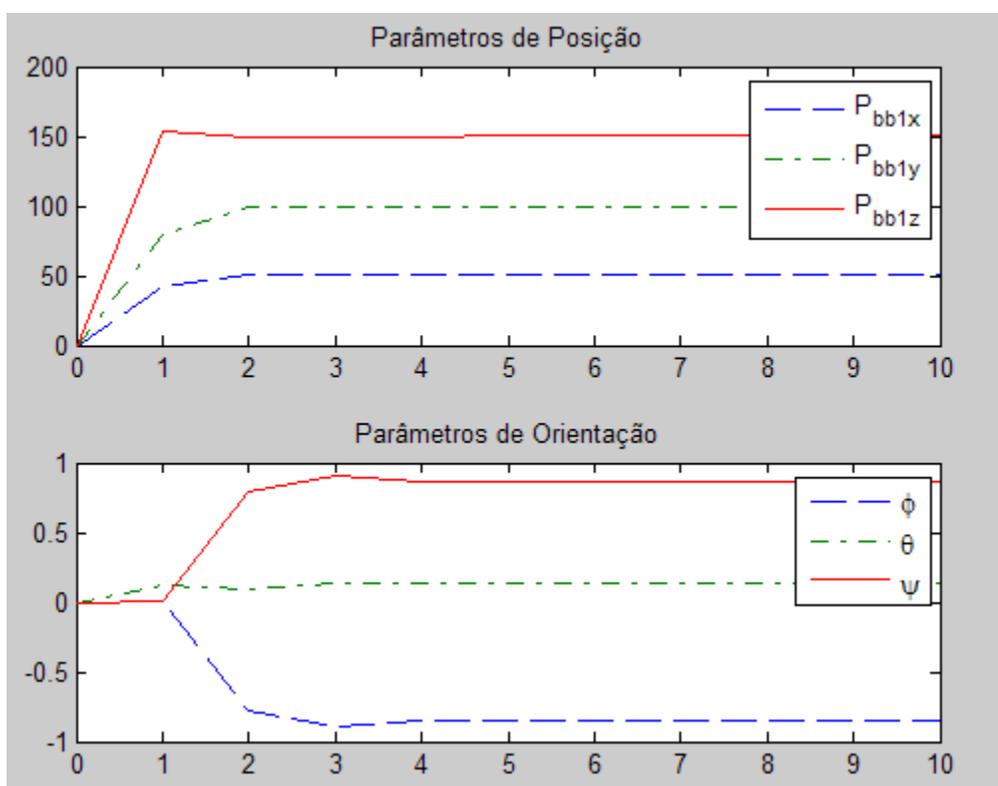


Figura 3.5 - Convergência dos parâmetros de x_{bb1} para N=5.

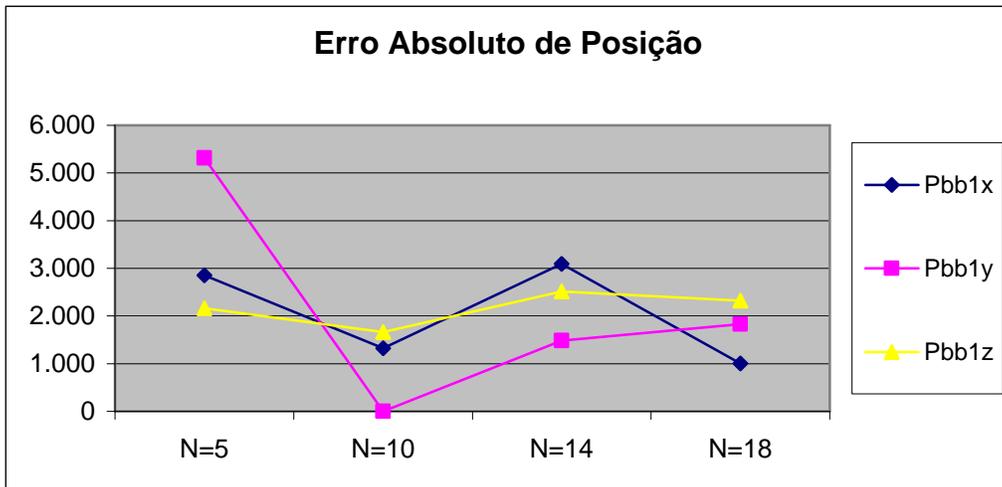


Figura 3.6 – Comparação dos erros absolutos de posição.

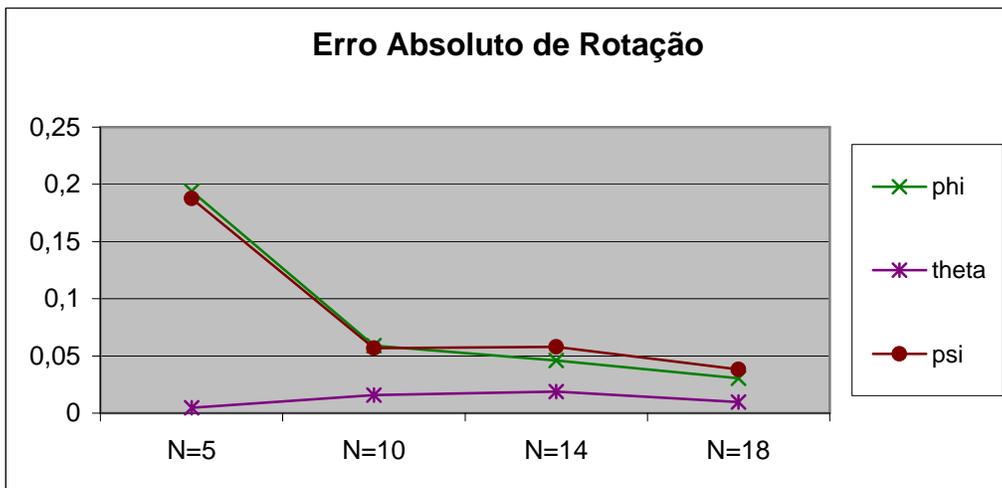


Figura 3.7 – Comparação dos erros absolutos de rotação.

As figuras 3.6 e 3.7 mostram dois gráficos comparativos entre os erros absolutos dos parâmetros do erro de posicionamento obtidos nas simulações com 5, 10, 14 e 18 pontos. O melhor resultado obtido, como se pode observar, foi com a simulação utilizando 10 pontos de calibração e o pior para N=5.

3.1.1.2 - Calibração utilizando pontos da grade

Nesta etapa de experiências de simulação utiliza-se a grade contendo os pontos a serem tocados pelo robô. O manipulador utilizado para obtenção das medidas dos parâmetros dos pontos foi o *Zebra Zero*, que é um robô de seis graus de liberdade de juntas rotacionais equipado com uma plataforma

móvel de três graus de liberdade, a qual em conjunto com o manipulador produzem nove graus de liberdade. O modelo cinemático desse robô, segundo a convenção *standard* de *Denavit-Hartenberg* é mostrado na tabela 3.7 a seguir:

Junta i	α_i (rad)	a_i (mm)	θ_i (rad)	d_i (mm)
1	$\frac{\pi}{2}$	0	0	0
2	0	l_1	0	0
3	$-\frac{\pi}{2}$	0	0	0
4	$\frac{\pi}{2}$	0	l_2	0
5	$-\frac{\pi}{2}$	0	0	0
6	0	0	0	0

Tabela 3.7 – Parâmetros de *Denavit-Hartenberg* para o robô *Zebra Zero*

A grade de calibração é composta por 50 pontos em três planos diferentes, como pode ser observado na visualização dos pontos medidos utilizando os sensores internos do manipulador na figura 3.8 a seguir:

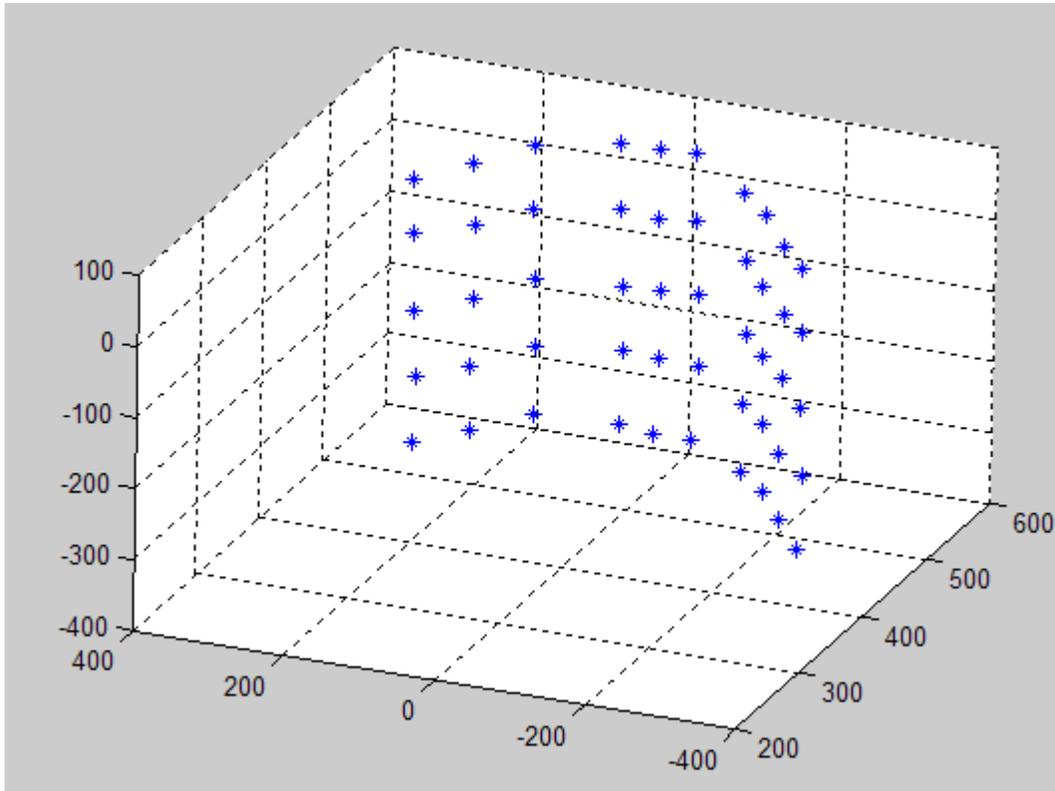


Figura 3.8 – Pontos da grade de calibração

Para simular a calibração, primeiramente deve-se tomar as medidas dos pontos da grade de calibração utilizando os sensores internos do robô na base original E_b . Em seguida, deve-se deslocar o robô para uma outra posição, base atual E_{b1} , e fazer novamente as medidas dos pontos utilizando a mesma grade de calibração.

A seguir podemos observar na tabela 3.8 as coordenadas dos pontos da grade de calibração medidas na base E_b .

Ponto(mm)	x	Y	z	Ponto(mm)	x	Y	Z
λ_1	399.2050	-311.5750	-319.2950	λ_{26}	534.5700	-14.4630	114.8850
λ_2	436.1050	256.3000	317.6790	λ_{27}	530.1450	32.2490	115.0890
λ_3	468.8510	207.8550	312.9060	λ_{28}	507.1340	129.4180	102.4440
λ_4	498.4190	154.3720	318.4430	λ_{29}	471.4910	181.5530	111.5430
λ_5	534.2820	-56.0430	318.7590	λ_{30}	437.2400	232.7650	109.0820
λ_6	533.3880	-8.9120	316.5030	λ_{31}	436.9360	232.0660	-0.4210
λ_7	535.7040	38.9260	313.3900	λ_{32}	471.0510	180.9800	-8.3100
λ_8	507.4750	129.8530	292.6660	λ_{33}	504.2400	132.3510	-2.9270
λ_9	471.7680	183.0650	297.6120	λ_{34}	535.7820	41.0430	-11.5730
λ_{10}	440.4050	233.8630	298.1850	λ_{35}	534.7770	-9.9460	-15.4420
λ_{11}	440.6490	229.4490	203.1380	λ_{36}	535.4010	-59.1790	-11.0920
λ_{12}	472.6970	184.9410	206.9520	λ_{37}	497.4690	157.0340	-18.8050
λ_{13}	508.2530	129.7320	199.0300	λ_{38}	465.8960	204.7890	-22.3390
λ_{14}	539.1910	37.7710	213.2940	λ_{39}	427.3720	266.1720	-21.3140
λ_{15}	535.3890	-12.2560	211.4220	λ_{40}	401.6470	310.8660	-16.5330
λ_{16}	533.3150	-67.7560	211.7610	λ_{41}	402.9940	308.9880	70.0800
λ_{17}	495.6580	157.3620	219.3270	λ_{42}	433.3950	260.8730	70.5640
λ_{18}	463.7350	209.3620	214.0030	λ_{43}	465.3780	208.1830	78.0990
λ_{19}	432.0100	258.5870	220.1620	λ_{44}	497.1930	154.3600	76.6170
λ_{20}	398.2030	317.7170	215.1650	λ_{45}	534.9470	-57.9660	85.5920
λ_{21}	398.2740	315.1180	119.5250	λ_{46}	535.5040	-8.6120	81.4410
λ_{22}	433.6860	259.0510	116.4330	λ_{47}	536.4980	44.6870	79.3610
λ_{23}	463.3970	209.4550	118.6340	λ_{48}	503.3840	128.2890	87.3530
λ_{24}	496.0700	160.8470	120.2190	λ_{49}	469.4000	183.8400	80.4600
λ_{25}	535.0970	-63.4070	112.3880	λ_{50}	436.5280	234.2760	75.8090

Tabela 3.8 – Pontos da grade de calibração medidos na base E_b

3.1.1.2.1 – Simulação com erro de posição

Primeiramente foi realizada a simulação do algoritmo para apenas translação da posição do manipulador, consistindo em um deslocamento de 100mm no eixo y, resultando nos seguintes pontos na base atual E_{b1} :

Ponto(mm)	x	Y	z	Ponto(mm)	X	Y	z
λ_1	397.1700	-	-322.6530	λ_{26}	533.8080	97.2180	-
λ_2	432.7190	-	-	λ_{27}	534.4570	147.6020	-
λ_3	459.9580	-	-	λ_{28}	504.2640	238.2720	-
λ_4	496.8190	-	-	λ_{29}	470.0290	287.9660	-
λ_5	528.3730	-	-	λ_{30}	438.8370	340.9070	-
λ_6	535.5810	-	-	λ_{31}	436.7730	342.0210	-
λ_7	537.6190	-	-	λ_{32}	468.4360	289.5770	-
λ_8	505.5520	-	-	λ_{33}	502.9510	235.6640	-
λ_9	472.0850	-	-	λ_{34}	539.8560	146.7710	-
λ_{10}	440.0490	-	-	λ_{35}	535.1870	100.3110	-
λ_{11}	437.0530	-	-	λ_{36}	535.0210	45.0880	-
λ_{12}	476.1130	-	-	λ_{37}	494.7840	-46.8710	-
λ_{13}	507.4330	-	-	λ_{38}	458.3920	102.0810	-
λ_{14}	534.2830	-	-	λ_{39}	431.9260	150.8220	-
λ_{15}	534.4960	-	-	λ_{40}	394.3820	206.8870	-
λ_{16}	534.0410	-	-	λ_{41}	397.7170	205.0350	-
λ_{17}	491.6400	-	-	λ_{42}	431.7320	152.3780	-
λ_{18}	457.9780	-	-	λ_{43}	464.3010	-97.8470	-
λ_{19}	426.0560	-	-	λ_{44}	494.9420	-51.3050	-
λ_{20}	394.2370	-	-	λ_{45}	534.9970	51.1500	-
λ_{21}	396.0190	-	-	λ_{46}	534.2560	101.2100	-
λ_{22}	426.6440	-	-	λ_{47}	534.2090	150.7750	-
λ_{23}	463.6460	-	-	λ_{48}	502.6880	240.7820	-

		101.2730	115.7530				
λ_{24}	494.4210	-50.3410	120.1660	λ_{49}	469.6670	292.8370	78.6350
λ_{25}	533.9500	40.7690	115.6960	λ_{50}	436.3420	341.6620	74.6030

Tabela 3.9 – Pontos da grade de calibração na base E_{b1} .

A matriz de erro de posicionamento x_{bb1} fica da forma:

Pârametros de x_{bb1}	Valor ideal	valor estimado
P_{bb1_x} (mm)	0	1.8808
P_{bb1_y} (mm)	-100	-110.5450
P_{bb1_z} (mm)	0	2.2211
φ (rad)	-A	-4.9682
ϑ (rad)	0	-0.0074
ψ (rad)	A	4.9746

Tabela 3.10 - Matriz de erro de posicionamento x_{bb1}

Vale ressaltar que quando $\vartheta=0$, há presença de singularidades, e por isso o ângulo de rotação Φ é determinado por todas as combinações de soma dos ângulos de rotação φ e ψ no eixo z.

No caso dessa simulação os ângulos φ e ψ devem se anular para caracterizar a ausência de rotação do erro de posicionamento. Pode-se observar a partir dos valores estimados que $\vartheta \approx 0$ e $\varphi + \psi = 0,0064$, praticamente igual a zero também.

Pode-se verificar na figura 3.9 como os parâmetros de erro de posicionamento convergem para seus respectivos valores a partir do 10º passo de iteração.

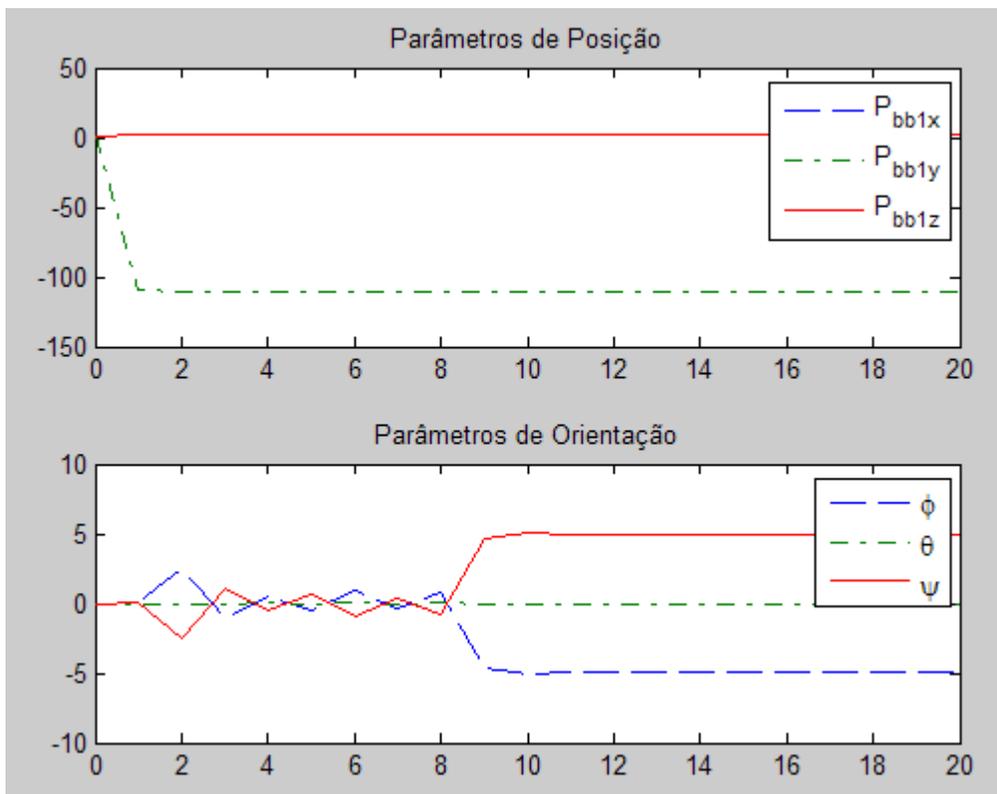


Figura 3.9 - Convergência dos parâmetros de x_{bb1} .

3.1.1.2.2 – Simulação com erro de posição e orientação

Após concluída a simulação anterior, foi realizada a simulação de calibração por sensor interno considerando translação e rotação da base original E_b para a base atual E_{b1} , consistindo em um deslocamento no eixo x de 10 mm e de 60mm no eixo y e uma rotação em z de $\frac{\pi}{6} rad$ ou 30° , como resultado tem-se os seguintes pontos na base E_{b1} :

Ponto(mm)	x	Y	Z	Ponto(mm)	X	Y	Z
λ_1	286.1340	-394.9420	-318.7790	λ_{26}	524.9790	-165.0410	-111.0530
λ_2	338.1610	-356.8650	-321.5550	λ_{27}	544.2470	-117.3360	-112.2910
λ_3	388.0520	-320.2260	-317.1160	λ_{28}	551.9640	-23.1500	-99.0820
λ_4	434.5020	-288.7880	-318.2940	λ_{29}	539.0120	43.4260	105.4420
λ_5	504.1370	-210.2160	-315.2920	λ_{30}	524.8170	99.0290	113.0230
λ_6	522.4430	-165.3790	-318.9950	λ_{31}	525.5380	99.4260	-11.2450
λ_7	546.0640	-118.9850	-308.1190	λ_{32}	537.4280	37.7040	-9.2250
λ_8	547.0620	-26.7950	-305.3740	λ_{33}	550.1590	-20.1170	-3.0290
λ_9	539.9660	32.0540	-293.9270	λ_{34}	545.8880	112.5480	-14.6660
λ_{10}	524.9750	91.1850	-302.4850	λ_{35}	526.9140	159.9810	-13.6780
λ_{11}	526.5690	96.8170	-210.8790	λ_{36}	506.1550	209.6320	-14.8890
λ_{12}	538.5920	30.3630	-209.6860	λ_{37}	430.5070	285.4580	-15.9260
λ_{13}	551.6750	-29.1540	-201.4600	λ_{38}	357.6670	320.7200	-27.5050
λ_{14}	545.7340	-115.5480	-216.6840	λ_{39}	332.8030	358.4220	-21.0710
λ_{15}	520.4370	-170.6640	-217.9080	λ_{40}	280.7350	389.5470	-19.8930
λ_{16}	499.6460	-217.1780	-214.4310	λ_{41}	294.3650	389.5840	77.5950
λ_{17}	428.4110	-290.2250	-223.0390	λ_{42}	338.9320	358.0520	75.6770
λ_{18}	383.7450	-321.7860	-219.4380	λ_{43}	386.2370	321.3870	73.1130
λ_{19}	338.0010	-354.6000	-221.6990	λ_{44}	433.8100	285.3540	73.6070
λ_{20}	282.7350	-392.0630	-218.2510	λ_{45}	507.1520	209.3980	73.4960
λ_{21}	283.1840	-392.2230	-119.4810	λ_{46}	526.7710	162.9690	72.3050
λ_{22}	334.2660	-359.9000	-115.2810	λ_{47}	543.0860	116.3600	78.6520
λ_{23}	386.0000	-317.9930	-117.5290	λ_{48}	547.0710	-17.3850	79.5310
λ_{24}	433.7800	-284.3770	-120.6210	λ_{49}	535.5960	41.3320	83.2630
λ_{25}	508.2230	-211.3060	-105.6290	λ_{50}	525.0020	96.1410	78.7560

Tabela 3.11 – Pontos da grade de calibração na base E_{b1} .

A matriz transformação x_{bb1} fica da forma:

Pârametros de x_{bb1}	Valor ideal	valor estimado
P_{bb1_x} (mm)	-10	-11.7885
P_{bb1_y} (mm)	-60	-57.2759
P_{bb1_z} (mm)	0	2.5679
φ (rad)	-A	5.1285
ϑ (rad)	0	0.0058
ψ (rad)	A+0.5236	-4.7400

Tabela 3.12 - Matriz de erro de posicionamento x_{bb1}

Pode-se observar, a partir dos valores estimados, que $\vartheta \approx 0$ e $\varphi + \psi = 0,3885 \text{ rad}$ ou $22^\circ 15'$.

Pode-se verificar na figura 3.10 como os parâmetros de erro de posicionamento convergem para seus respectivos valores a partir do 4º passo de iteração.

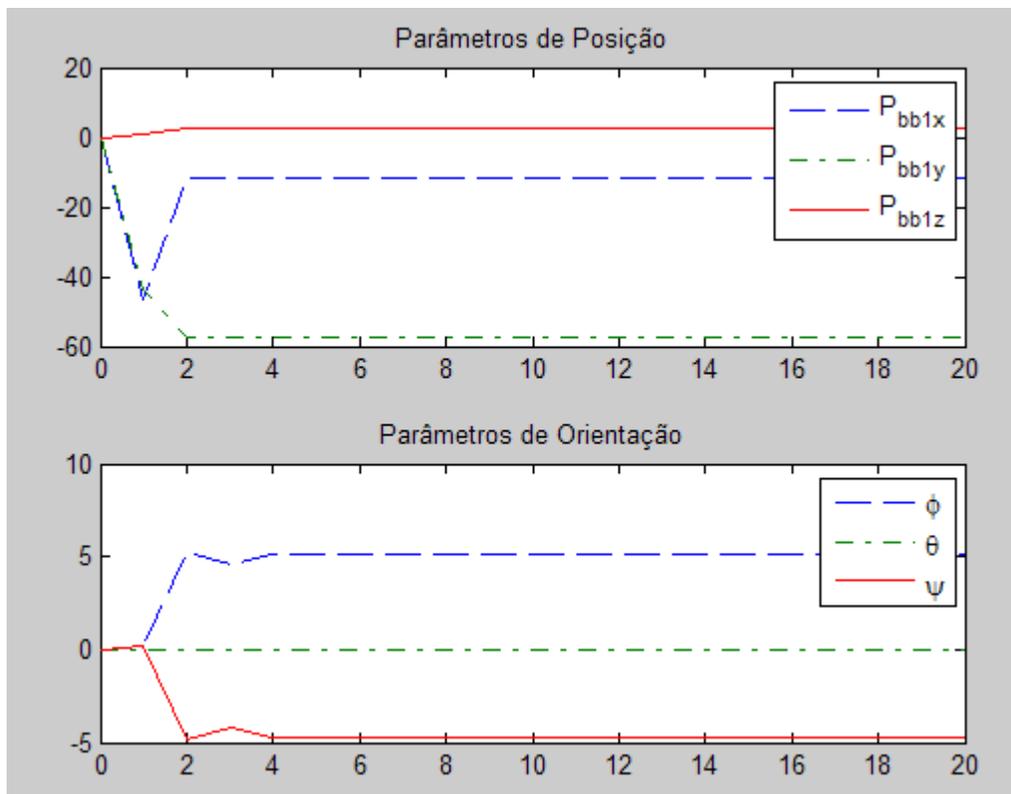


Figura 3.9 - Convergência dos parâmetros de x_{bb1} .

Foi verificado que a ordem de medida e o número de pontos da grade de calibração devem ser os mesmos tanto na base original E_b quanto na base atual E_{b1} , caracterizando restrições ao processo de tomada de dados.

3.2 - Estimação do erro de posicionamento com sensor externo

Na estimação do erro de posicionamento com sensor externo, o erro de configuração do manipulador é estimado através da utilização de uma câmera de vídeo fixa no espaço de trabalho, apresentando papel bastante importante na realização de tarefas de robôs em ambientes hostis e remotos.

Da mesma forma que no método de calibração por sensor interno, o robô deverá tocar os pontos da grade de calibração estando na base original E_b . Em seguida deverá tocar esses mesmos pontos estando na base atual E_{b1} . Entretanto, a tomada de medidas será vista no ambiente da câmera. Como pode ser observado, a medida dos pontos é feita conforme o método de estimação do erro de posicionamento com sensor interno.

Vale lembrar que o método de calibração por sensor externo é complementar ao método por sensor interno, e não concorrente a ele, pois as medidas de orientação dos pontos da grade de calibração tanto na base E_b quanto na base E_{b1} são tomados através dos *encoders*.

Seja $P_{c\lambda_i}$ o ponto λ_i no plano de trabalho expresso em relação ao sistema de coordenada da câmera, R_{cb} a matriz de rotação do sistema de coordenadas E_b em relação ao sistema de coordenadas da câmera E_c , $P_{b\lambda_i}$ o ponto λ_i expresso em relação ao sistema de coordenada na base E_b e P_{bc} o vetor translação entre o sistema de coordenada do manipulador e o sistema de coordenada da câmera, tem-se que:

$$P_{b\lambda_i} = R_{cb}^T P_{c\lambda_i} + P_{bc} \quad (3.11)$$

Equivalentemente:

$$P_{c\lambda_i} = R_{cb} (P_{b\lambda_i} - P_{bc}) \quad (3.12)$$

ou em termos de coordenadas cartesianas:

$$\begin{bmatrix} x_{c\lambda i} \\ y_{c\lambda i} \\ z_{c\lambda i} \end{bmatrix} = R_{cb} \left(\begin{bmatrix} x_{b\lambda i} \\ y_{b\lambda i} \\ z_{b\lambda i} \end{bmatrix} - \begin{bmatrix} x_{bc} \\ y_{bc} \\ z_{bc} \end{bmatrix} \right) \quad (3.13)$$

As coordenadas de projeção no plano imagem são dadas por:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \frac{f}{z_{c\lambda i}} \begin{bmatrix} x_{c\lambda i} \\ y_{c\lambda i} \end{bmatrix} \quad (3.14)$$

onde f é a distância focal da câmera de vídeo.

Considerando a utilização de uma câmera CCD digital, deve-se fazer uma correção nas coordenadas de projeção no plano imagem:

$$\begin{bmatrix} x_{im} \\ y_{im} \end{bmatrix} = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix}, \quad (3.15)$$

onde α_1 e α_2 são os fatores de escala da câmera de vídeo [pixels/mm].

Logo, a partir das relações descritas em (3.14) e (3.15), podemos escrever:

$$P_{im_i} = \begin{bmatrix} x_{im} \\ y_{im} \end{bmatrix} = \frac{f}{z_{c\lambda i}} \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} \begin{bmatrix} x_{c\lambda i} \\ y_{c\lambda i} \end{bmatrix} \quad (3.16)$$

Utilizando um conjunto de n pontos da grade de calibração 3D $[x_{c\lambda i} y_{c\lambda i} z_{c\lambda i}]^T$ e o correspondente 2D $[x_{im_i} y_{im_i}]^T$, onde $i=1, \dots, n$ serão estimados os parâmetros de orientação e posição do erro de posicionamento entre as localizações original e atual.

A partir de (3.16) e de (3.13) tem-se que:

$$P_{im_i} = \frac{f}{z_{c\lambda i}} \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} (R_{cb})_{2D} (P_{b\lambda i})_{2D} - \frac{f}{z_{c\lambda i}} \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} (R_{cb})_{2D} (P_{bc})_{2D}, \quad (3.17)$$

onde $(R_{cb})_{2D}$ é a forma bidimensional da matriz rotação entre a base do manipulador e a da câmera de vídeo, $(P_{b\lambda i})_{2D}$ é a representação bidimensional dos pontos λ_i em relação a base do manipulador e $(P_{bc})_{2D}$ é a representação bidimensional da parte translacional da transformação entre a base do manipulador e a da câmera de vídeo.

ou de uma forma mais resumida:

$$P_{im_i} = K_{p_i} (P_{b\lambda i})_{2D} - P_{c0}, \quad (3.18)$$

onde $K_{p_i} = \frac{f}{z_{c\lambda i}} \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} (R_{cb})_{2D}$ é a matriz de transformação do espaço operacional para o espaço da câmera e $P_{c0} = \frac{f}{z_{c\lambda i}} \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} (R_{cb})_{2D} (P_{bc})_{2D}$ é um termo constante que depende da posição do sistema de coordenadas da câmera em relação ao sistema de coordenadas do manipulador.

Em geral, P_{bc} é constante e não-homogênea. Entretanto, sem perda de generalidade, pois em etapas posteriores essa equação passará por processo de derivação, pode-se assumir que as origens dos sistemas de coordenadas da câmera de vídeo e do manipulador são coincidentes, assim $P_{c0}=0$ e P_{im_i} resume-se a:

$$P_{im_i} = K_{p_i} (P_{b\lambda i})_{2D}. \quad (3.19)$$

Observa-se que através dessa equação pode-se obter apenas uma forma bidimensional dos pontos de calibração na base E_b . Entretanto, para conseguir representar tridimensionalmente a posição do ponto na base E_b , basta acrescentar a coordenada $z_{c\lambda_i}$ ao vetor $(P_{b\lambda_i})_{2D}$, que ficará na forma:

$$(P_{b\lambda_i})_{3D} = [x_{b\lambda_i} y_{b\lambda_i} z_{c\lambda_i}]^T \quad (3.20)$$

Assim, as posições do ponto λ_i em relação aos sistemas de coordenadas E_b e E_{b1} tornam-se, respectivamente:

$$P_{b\lambda_{i_{novo}}} = R_{cb}^T (P_{b\lambda_i})_{3D} + P_{bc} \quad (3.21)$$

$$P_{b_1\lambda_{i_{novo}}} = R_{cb}^T (P_{b_1\lambda_i})_{3D} + P_{bc} \quad (3.22)$$

Logo, com $P_{b\lambda_{i_{novo}}}$ e $P_{b_1\lambda_{i_{novo}}}$ pode-se utilizar o mesmo procedimento proposto na seção de estimação de erro de posicionamento por sensor interno para se determinar os parâmetros da matriz T_{bb1} .

3.2.1 - Simulação

A partir do método apresentado na seção de calibração por sensor externo, observa-se que sua simulação tem de verificar a validade dos pontos $P_{b\lambda_{i_{novo}}}$ e $P_{b_1\lambda_{i_{novo}}}$ estimados a partir dos pontos medidos pela câmera para encontrar a matriz x_{bb1} .

Os parâmetros da câmera encontram-se na tabela 3.13 a seguir:

Variável	Valor	Unidade
f	6.0	mm
α_1	100.0	mm
α_2	100.0	Mm

Tabela 3.13 – Parâmetros da câmera de vídeo

As simulações desta seção serão separadas em duas partes. A primeira utiliza pontos gerados computacionalmente e a segunda utiliza pontos obtidos a partir da grade de calibração. Vale ressaltar que esses pontos foram medidos previamente utilizando os sensores internos do robô.

3.2.1.1 - Simulação a partir dos pontos gerados computacionalmente

3.2.1.1.1 - Simulação sem ruído

Os valores encontrados para $P_{b_{novo}}$ a partir do algoritmo foram:

Ponto(mm)	X	y	Z
λ_1	228.6000	0	379.4000
λ_2	211.2352	98.4808	379.4000
λ_3	193.8704	196.9616	379.4000
λ_4	228.6000	0	279.4000
λ_5	211.2352	98.4808	279.4000
λ_6	193.8704	196.9616	279.4000
λ_7	228.6000	0	179.4000
λ_8	211.2352	98.4808	179.4000
λ_9	193.8704	196.9616	179.4000
λ_{10}	193.8704	-196.9616	379.4000
λ_{11}	211.2352	-98.4808	379.4000
λ_{12}	228.6000	0	379.4000
λ_{13}	193.8704	-196.9616	279.4000
λ_{14}	211.2352	-98.4808	279.4000
λ_{15}	228.6000	0	279.4000
λ_{16}	193.8704	-196.9616	179.4000
λ_{17}	211.2352	-98.4808	179.4000
λ_{18}	228.6000	0	179.4000

Tabela 3.14 – Pontos na base E_b

Enquanto que os valores encontrados para $P_{b_{novo}}$ foram:

Ponto(mm)	x	Y	z
λ_1	157.2433	-79.6259	251.9059
λ_2	140.6926	18.4801	241.8419
λ_3	124.1420	116.5861	231.7780
λ_4	165.9257	-88.3415	152.6655
λ_5	149.3751	9.7645	142.6015
λ_6	132.8244	107.8705	132.5376
λ_7	174.6081	-97.0571	53.4251
λ_8	158.0575	1.0489	43.3612
λ_9	141.5068	99.1550	33.2972
λ_{10}	121.1497	-275.8380	265.9800
λ_{11}	139.1965	-177.7320	258.9429
λ_{12}	157.2433	-79.6259	251.9059
λ_{13}	129.8321	-284.5536	166.7396
λ_{14}	147.8789	-186.4475	159.7025
λ_{15}	165.9257	-88.3415	152.6655
λ_{16}	138.5145	-293.2691	67.4992
λ_{17}	156.5613	-195.1631	60.4622
λ_{18}	174.6081	-97.0571	53.4251

Tabela 3.15 – Pontos na base E_{b1}

Conforme esperado, os valores obtidos pra os parâmetros da matriz x_{bb1} são idênticos aos valores ideais, verificando assim a validade do método proposto para obtenção da matriz transformação a partir do método de calibração por sensor externo.

Pârametros de x_{bb1}	valor ideal	valor obtido
P_{bb1_x} (mm)	50	50.0000
P_{bb1_y} (mm)	100	100.0000
P_{bb1_z} (mm)	150	150.0000
φ (rad)	-0.7835	-0.7835
\mathcal{G} (rad)	0.1233	0.1233
ψ (rad)	0.7873	0.7873

Tabela 3.16 – Matriz de erro de posicionamento x_{bb1}

Pode-se observar abaixo como os parâmetros de x_{bb1} convergem para os valores desejados na figura 3.20. Nota-se que os valores são alcançados a partir do 3º passo de iteração.

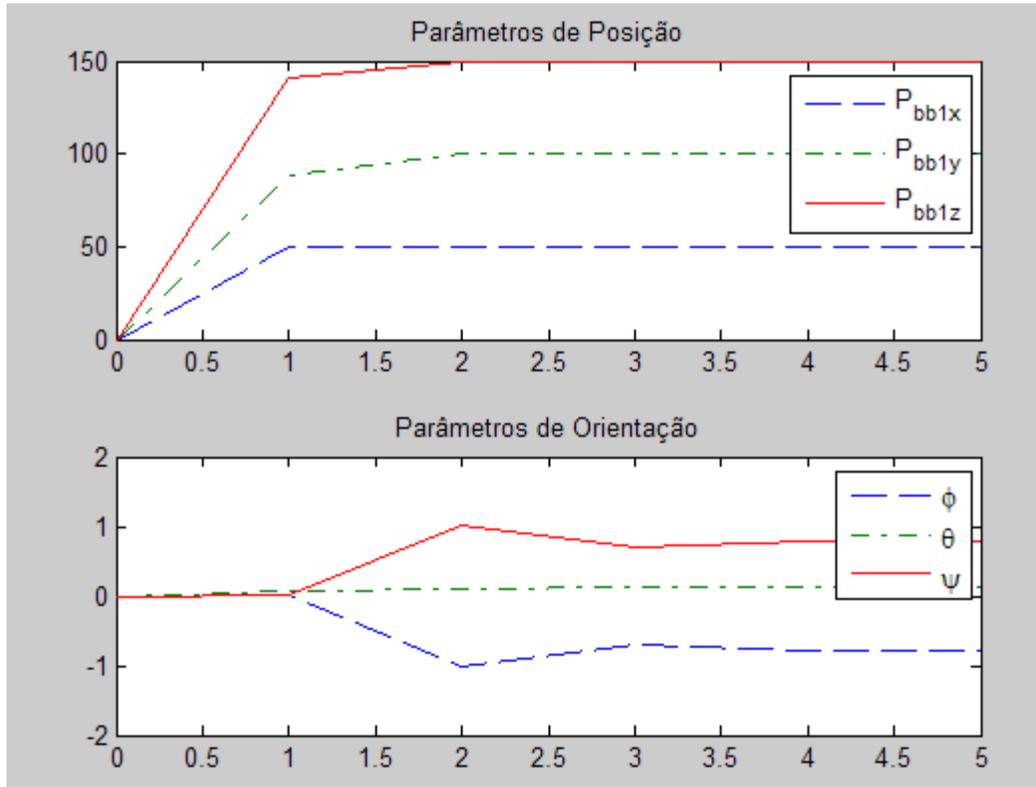


Figura 3.10 – Convergência dos parâmetros de x_{bb1} .

3.2.1.1.2 - Simulação com ruído

Para simular a tomada de dados com ruído, utilizou-se a função *rand()* do Matlab, da mesma forma que no método de calibração por sensor interno.

Os valores encontrados para P_{b1novo} a partir do algoritmo foram:

Ponto(mm)	x	Y	z
λ_1	158.1155	-79.5626	251.9059
λ_2	139.9735	18.8964	241.8419
λ_3	124.9257	115.4727	231.7780
λ_4	166.2025	-88.5255	152.6655
λ_5	149.8482	9.7685	142.6015
λ_6	133.1020	107.7763	132.5376
λ_7	174.5037	-97.2229	53.4251
λ_8	157.9245	1.1279	43.3612
λ_9	141.4411	99.1688	33.2972
λ_{10}	120.2211	-275.3116	265.9800
λ_{11}	138.8816	-176.7997	258.9429
λ_{12}	158.1342	-79.3903	251.9059
λ_{13}	129.8264	-283.8870	166.7396
λ_{14}	148.3926	-186.2161	159.7025
λ_{15}	166.4112	-88.0969	152.6655
λ_{16}	138.4078	-293.4111	67.4992
λ_{17}	156.4653	-195.1425	60.4622
λ_{18}	174.7295	-97.1590	53.4251

Tabela 3.17 – Pontos na base E_{b1} .

Pâmetros de x_{bb1}	valor ideal	valor obtido	erro absoluto
P_{bb1_x} (mm)	50	49.8635	0.1365
P_{bb1_y} (mm)	100	99.7843	0.2157
P_{bb1_z} (mm)	150	149.9238	0.0762
φ (rad)	-0.7835	-0.7899	0.0064
\mathcal{G} (rad)	0.1233	0.1231	0.0002
ψ (rad)	0.7873	0.7949	0.0076

Tabela 3.18 - Erro de posicionamento estimado com 18 pontos de calibração.

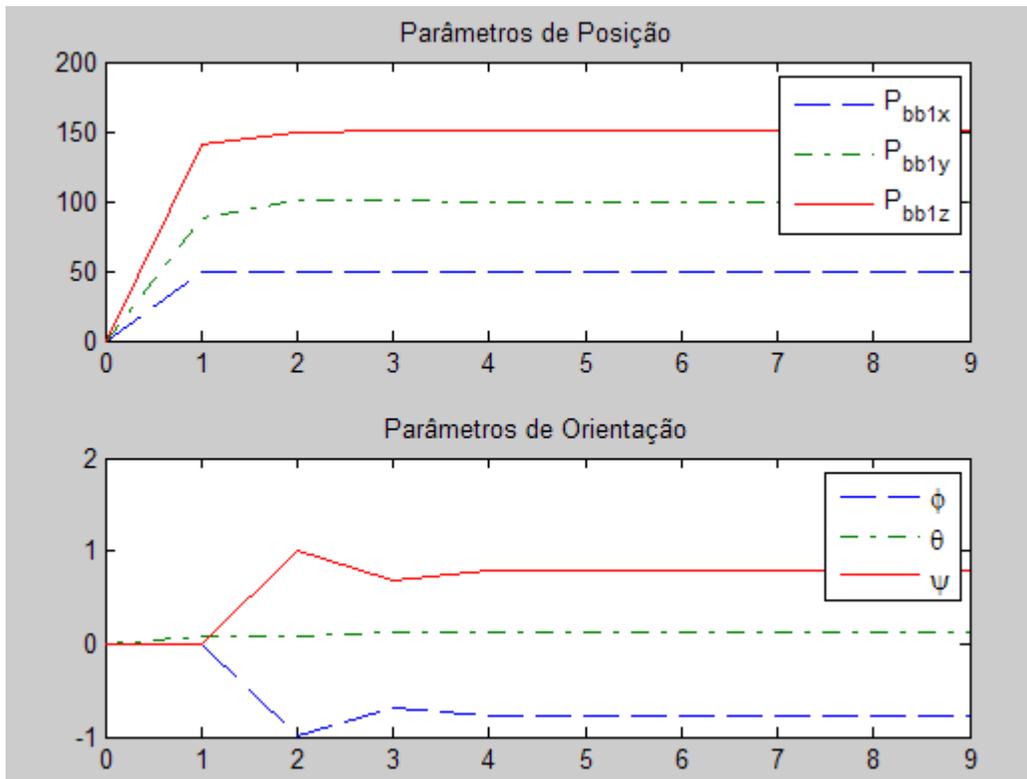


Figura 3.11 - Convergência dos parâmetros de x_{bb1} para N=18.

A partir da figura 3.11, observa-se a convergência dos parâmetros de erro de posicionamento a partir do 4º passo de iteração.

Pârametros de x_{bb1}	valor ideal	valor obtido	erro absoluto
P_{bb1_x} (mm)	50	50.0653	0.0653
P_{bb1_y} (mm)	100	99.6472	0.3528
P_{bb1_z} (mm)	150	149.8008	0.1992
φ (rad)	-0.7835	-0.7846	0.0011
ϑ (rad)	0.1233	0.1223	0.0010
ψ (rad)	0.7873	0.7896	0.0023

Tabela 3.19 - Erro de posicionamento estimado com 14 pontos de calibração.

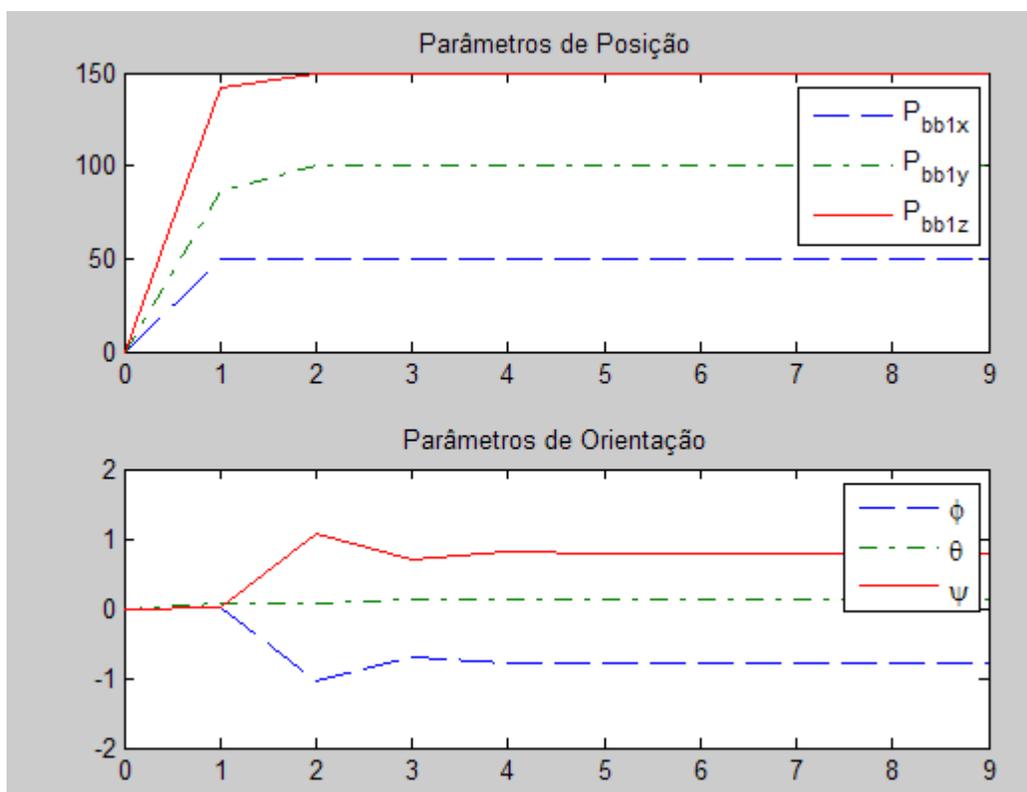


Figura 3.12 - Convergência dos parâmetros de x_{bb1} para N=14.

A partir da figura 3.12, observa-se a convergência dos parâmetros de erro de posicionamento a partir do 4º passo de iteração.

Parâmetros de x_{bb1}	valor ideal	valor obtido	erro absoluto
P_{bb1_x} (mm)	50	49.9498	0.0502
P_{bb1_y} (mm)	100	100.3433	0.3433
P_{bb1_z} (mm)	150	149.9200	0.0800
φ (rad)	-0.7835	-0.7935	0.0100
ϑ (rad)	0.1233	0.1233	0.0000
ψ (rad)	0.7873	0.7959	0.0086

Tabela 3.20 - Erro de posicionamento estimado com 10 pontos de calibração.

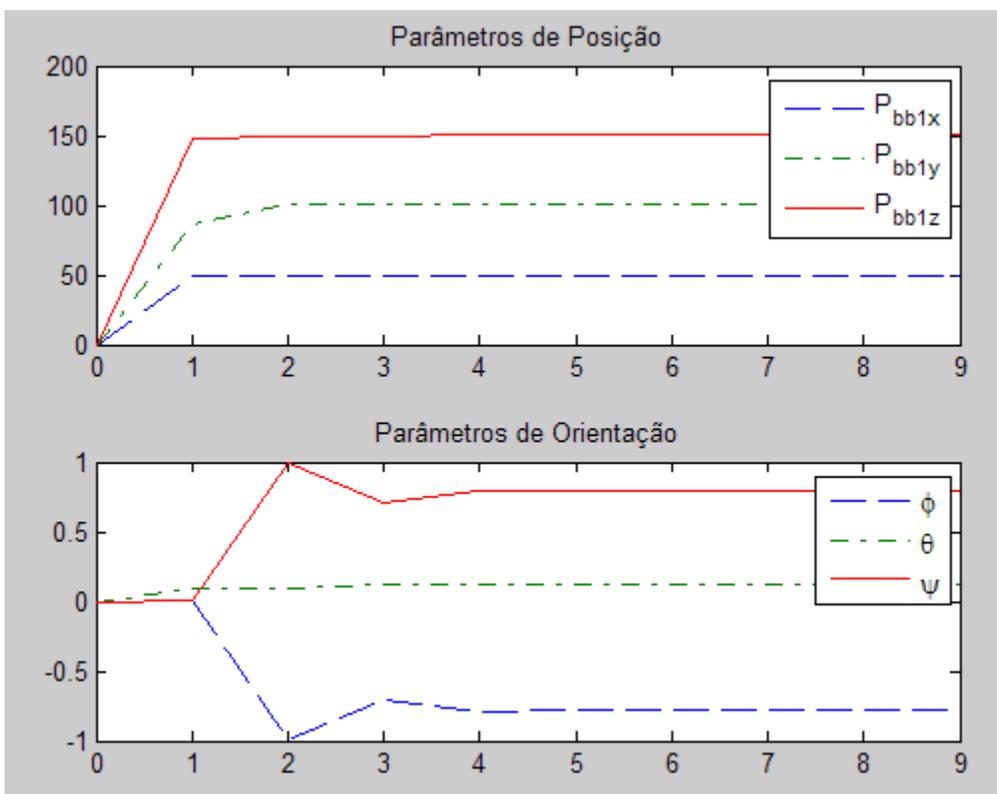


Figura 3.13 - Convergência dos parâmetros de x_{bb1} para $N=10$.

A partir da figura 3.13, observa-se a convergência dos parâmetros de erro de posicionamento a partir do 4º passo de iteração.

Pâmetros de x_{bb1}	valor ideal	valor obtido	erro absoluto
P_{bb1_x} (mm)	50	50.3016	0.3016
P_{bb1_y} (mm)	100	100.0876	0.0876
P_{bb1_z} (mm)	150	150.0080	0.0080
φ (rad)	-0.7835	-0.7869	0.0034
ϑ (rad)	0.1233	0.1240	0.0007
ψ (rad)	0.7873	0.7928	0.0055

Tabela 3.21 - Erro de posicionamento estimado com 5 pontos de calibração.

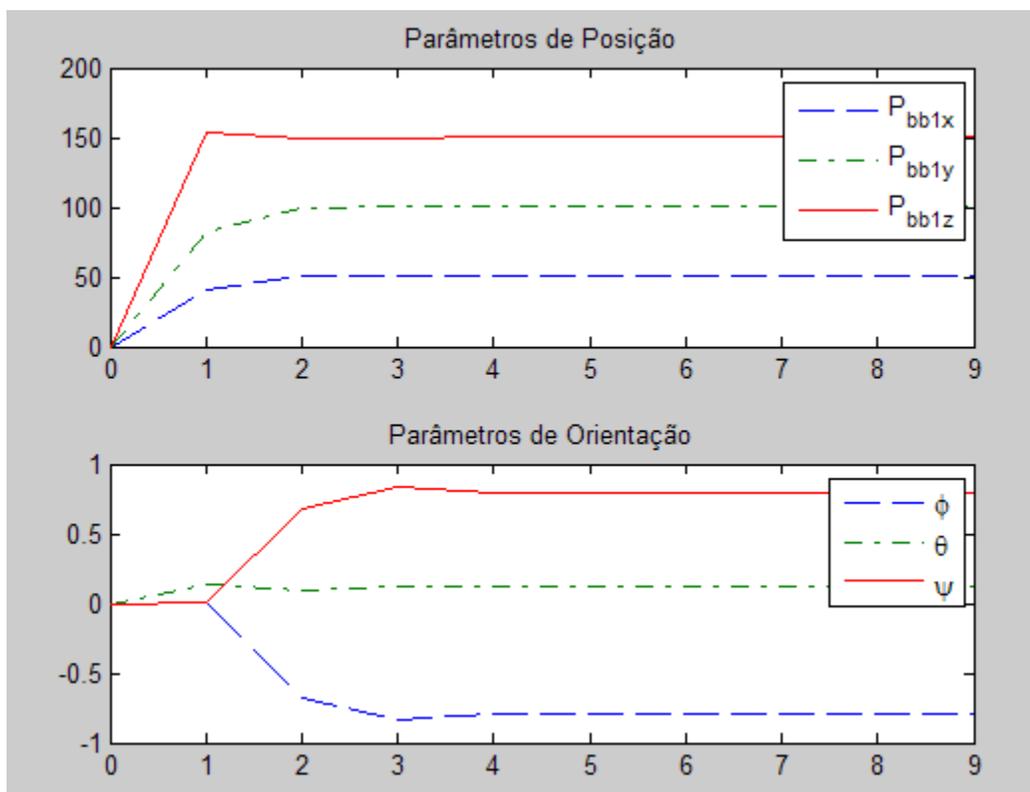


Figura 3.14 - Convergência dos parâmetros de x_{bb1} para $N=5$.

A partir da figura 3.14, observa-se a convergência dos parâmetros de erro de posicionamento a partir do 4º passo de iteração.

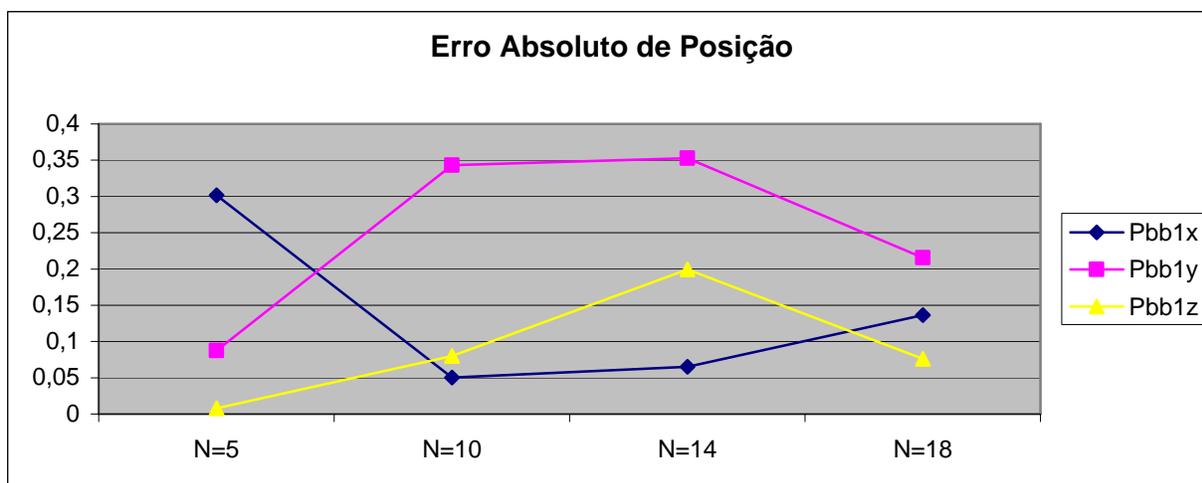


Figura 3.15 – Comparação dos parâmetros de erro absoluto de posição

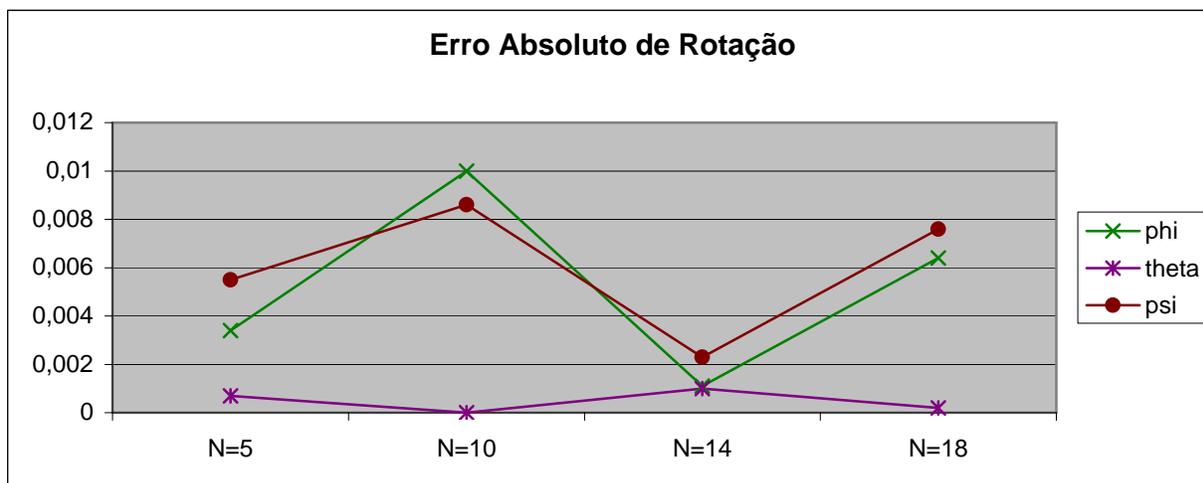


Figura 3.16 – Comparação dos parâmetros de erro absoluto de rotação

A partir das figuras 3.15 e 3.16, observa-se que para N=18 foram obtidos os menores erros de posição, enquanto que para N=14 foi apresentado o maior erro de posição. E para N=14 foram obtidos os menores erros de rotação, enquanto que para N=10 foi apresentado um maior erro de rotação.

3.2.1.2 - Simulação utilizando pontos obtidos a partir da grade de calibração

Nesta etapa da simulação foi utilizado o robô *Zebra Zero*, da mesma forma que no método de calibração por sensor interno. Também foram utilizados a mesma grade de calibração e os mesmos procedimentos para tomada de dados que no outro método.

A seguir podem-se observar as coordenadas dos pontos da grade de calibração medidas na base

E_b :

Ponto(mm)	x	Y	z	Ponto(mm)	x	y	z
λ_1	399.2050	-	-	λ_{26}	534.5700	-14.4630	-
λ_2	436.1050	256.3000	317.6790	λ_{27}	530.1450	32.2490	-
λ_3	468.8510	207.8550	312.9060	λ_{28}	507.1340	129.4180	-
λ_4	498.4190	154.3720	318.4430	λ_{29}	471.4910	181.5530	-
λ_5	534.2820	-56.0430	318.7590	λ_{30}	437.2400	232.7650	-
λ_6	533.3880	-8.9120	316.5030	λ_{31}	436.9360	232.0660	-
λ_7	535.7040	38.9260	313.3900	λ_{32}	471.0510	180.9800	-
λ_8	507.4750	129.8530	292.6660	λ_{33}	504.2400	132.3510	-
λ_9	471.7680	183.0650	297.6120	λ_{34}	535.7820	41.0430	-
λ_{10}	440.4050	233.8630	298.1850	λ_{35}	534.7770	-9.9460	-
λ_{11}	440.6490	229.4490	203.1380	λ_{36}	535.4010	-59.1790	-
λ_{12}	472.6970	184.9410	206.9520	λ_{37}	497.4690	157.0340	-
λ_{13}	508.2530	129.7320	199.0300	λ_{38}	465.8960	204.7890	-
λ_{14}	539.1910	37.7710	213.2940	λ_{39}	427.3720	266.1720	-
λ_{15}	535.3890	-12.2560	211.4220	λ_{40}	401.6470	310.8660	-
λ_{16}	533.3150	-67.7560	211.7610	λ_{41}	402.9940	308.9880	-
λ_{17}	495.6580	157.3620	219.3270	λ_{42}	433.3950	260.8730	-
λ_{18}	463.7350	209.3620	214.0030	λ_{43}	465.3780	208.1830	-
λ_{19}	432.0100	258.5870	220.1620	λ_{44}	497.1930	154.3600	-
λ_{20}	398.2030	317.7170	215.1650	λ_{45}	534.9470	-57.9660	-
λ_{21}	398.2740	315.1180	119.5250	λ_{46}	535.5040	-8.6120	-
λ_{22}	433.6860	259.0510	116.4330	λ_{47}	536.4980	44.6870	-
λ_{23}	463.3970	209.4550	118.6340	λ_{48}	503.3840	128.2890	-
λ_{24}	496.0700	160.8470	120.2190	λ_{49}	469.4000	183.8400	-
λ_{25}	535.0970	-63.4070	112.3880	λ_{50}	436.5280	234.2760	-

Tabela 3.22 – Pontos na base E_b

3.2.1.2.1 – Simulação com erro de posição

Primeiramente foi realizada a simulação do algoritmo para apenas translação da posição do manipulador, consistindo em um deslocamento de 100mm no eixo y, resultando nos seguintes pontos na base atual E_{b1} :

Ponto(mm)	x	Y	Z	Ponto(mm)	x	y	z
λ_1	397.1700	-203.8710	-	λ_{26}	533.8080	97.2180	-
λ_2	432.7190	-154.9780	-	λ_{27}	534.4570	147.6020	-
λ_3	459.9580	-104.4850	-	λ_{28}	504.2640	238.2720	-
λ_4	496.8190	-50.8520	-	λ_{29}	470.0290	287.9660	-
λ_5	528.3730	41.0450	-	λ_{30}	438.8370	340.9070	-
λ_6	535.5810	95.1570	-	λ_{31}	436.7730	342.0210	-
λ_7	537.6190	148.1030	-	λ_{32}	468.4360	289.5770	-
λ_8	505.5520	238.2000	-	λ_{33}	502.9510	235.6640	-
λ_9	472.0850	286.0620	-	λ_{34}	539.8560	146.7710	-
λ_{10}	440.0490	338.4860	-	λ_{35}	535.1870	100.3110	-
λ_{11}	437.0530	337.9020	-	λ_{36}	535.0210	45.0880	-
λ_{12}	476.1130	280.7160	-	λ_{37}	494.7840	-46.8710	-
λ_{13}	507.4330	234.8110	-	λ_{38}	458.3920	-102.0810	-
λ_{14}	534.2830	146.2470	-	λ_{39}	431.9260	-150.8220	-
λ_{15}	534.4960	91.8960	-	λ_{40}	394.3820	-206.8870	-
λ_{16}	534.0410	41.0710	-	λ_{41}	397.7170	-205.0350	-
λ_{17}	491.6400	-49.5800	-	λ_{42}	431.7320	-152.3780	-
λ_{18}	457.9780	-107.7400	-	λ_{43}	464.3010	-97.8470	-
λ_{19}	426.0560	-159.4300	-	λ_{44}	494.9420	-51.3050	-
λ_{20}	394.2370	-207.4750	-	λ_{45}	534.9970	51.1500	-
λ_{21}	396.0190	-205.6640	-	λ_{46}	534.2560	101.2100	-
λ_{22}	426.6440	-157.5680	-	λ_{47}	534.2090	150.7750	-
λ_{23}	463.6460	-101.2730	-	λ_{48}	502.6880	240.7820	-
λ_{24}	494.4210	-50.3410	-	λ_{49}	469.6670	292.8370	-
λ_{25}	533.9500	40.7690	-	λ_{50}	436.3420	341.6620	-

Tabela 3.23 – Pontos na base E_{b1} .

A matriz de erro de posicionamento x_{bb1} fica da forma:

Pârametros de x_{bb1}	valor ideal	valor obtido
P_{bb1_x} (mm)	0	1.8630
P_{bb1_y} (mm)	-100	-110.6772
P_{bb1_z} (mm)	0	1.9286
φ (rad)	-A	1.2678
ϑ (rad)	0	-0.0079
ψ (rad)	A	-1.2612

Tabela 3.24 – parâmetros de x_{bb1} .

No caso dessa simulação os ângulos φ e ψ devem se anular, já que $\vartheta \approx 0$, para caracterizar a ausência de rotação do erro de posicionamento. Pode-se observar a partir dos valores estimados que $\varphi + \psi = 0,0067$, praticamente igual a zero também.

Pode-se verificar na figura 3.17 como os parâmetros de erro de posicionamento convergem para seus respectivos valores a partir do 5º passo de iteração.

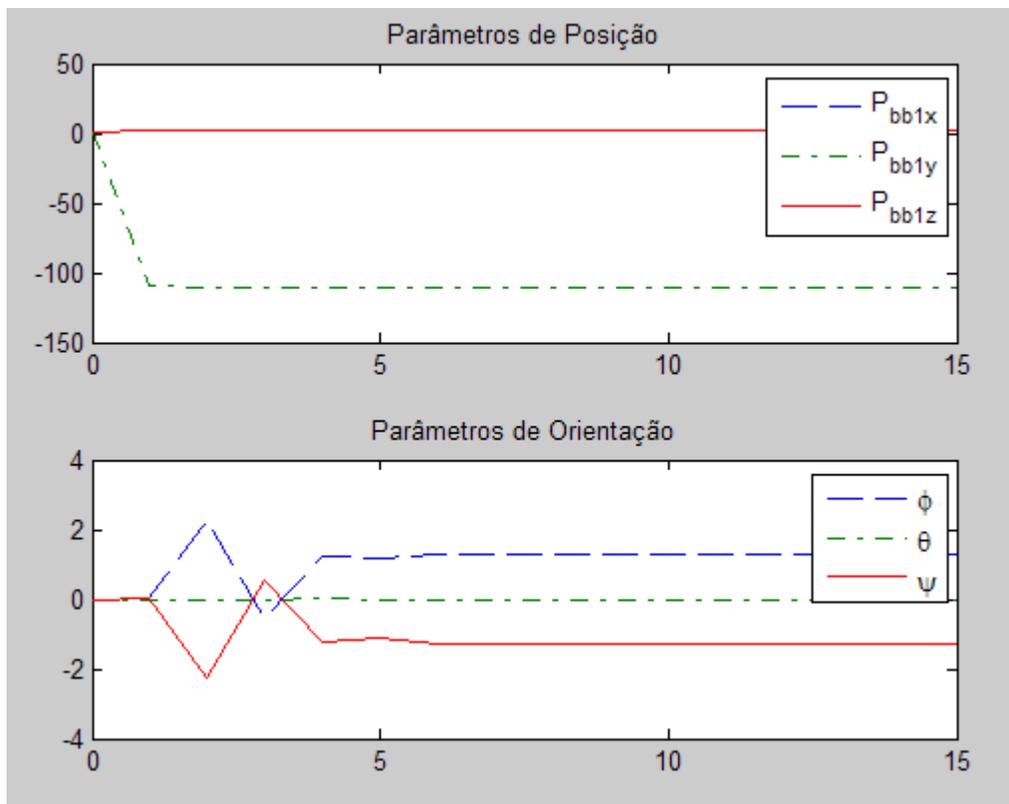


Figura 3.17 – Convergência dos parâmetros de x_{bb1} .

3.2.1.2.2 – Simulação com erro de posição e rotação

Após concluída a simulação anterior, foi realizada a simulação de calibração por sensor interno considerando translação e rotação da base original E_b para a base atual E_{b1} , consistindo em um deslocamento no eixo x de 10 mm e de 60mm no eixo y e uma rotação em z de $\frac{\pi}{6}$ rad ou 30°, como resultado tem-se os seguintes pontos na base E_{b1} :

Ponto(mm)	x	Y	z	Ponto(mm)	x	y	z
λ_1	286.1340	-394.9420	-318.7790	λ_{26}	524.9790	-165.0410	-111.0530
λ_2	338.1610	-356.8650	-321.5550	λ_{27}	544.2470	-117.3360	-112.2910
λ_3	388.0520	-320.2260	-317.1160	λ_{28}	551.9640	-23.1500	-99.0820
λ_4	434.5020	-288.7880	-318.2940	λ_{29}	539.0120	43.4260	-105.4420
λ_5	504.1370	-210.2160	-315.2920	λ_{30}	524.8170	99.0290	-113.0230
λ_6	522.4430	-165.3790	-318.9950	λ_{31}	525.5380	99.4260	-11.2450
λ_7	546.0640	-118.9850	-308.1190	λ_{32}	537.4280	37.7040	-9.2250
λ_8	547.0620	-26.7950	-305.3740	λ_{33}	550.1590	-20.1170	-3.0290
λ_9	539.9660	32.0540	-293.9270	λ_{34}	545.8880	-112.5480	-14.6660
λ_{10}	524.9750	91.1850	-302.4850	λ_{35}	526.9140	-159.9810	-13.6780
λ_{11}	526.5690	96.8170	-210.8790	λ_{36}	506.1550	-209.6320	-14.8890
λ_{12}	538.5920	30.3630	-209.6860	λ_{37}	430.5070	-285.4580	-15.9260
λ_{13}	551.6750	-29.1540	-201.4600	λ_{38}	357.6670	-320.7200	-27.5050
λ_{14}	545.7340	-115.5480	-216.6840	λ_{39}	332.8030	-358.4220	-21.0710
λ_{15}	520.4370	-170.6640	-217.9080	λ_{40}	280.7350	-389.5470	-19.8930
λ_{16}	499.6460	-217.1780	-214.4310	λ_{41}	294.3650	-389.5840	77.5950
λ_{17}	428.4110	-290.2250	-223.0390	λ_{42}	338.9320	-358.0520	75.6770
λ_{18}	383.7450	-321.7860	-219.4380	λ_{43}	386.2370	-321.3870	73.1130
λ_{19}	338.0010	-354.6000	-221.6990	λ_{44}	433.8100	-285.3540	73.6070
λ_{20}	282.7350	-392.0630	-218.2510	λ_{45}	507.1520	-209.3980	73.4960
λ_{21}	283.1840	-392.2230	-119.4810	λ_{46}	526.7710	-162.9690	72.3050
λ_{22}	334.2660	-359.9000	-115.2810	λ_{47}	543.0860	-116.3600	78.6520
λ_{23}	386.0000	-317.9930	-117.5290	λ_{48}	547.0710	-17.3850	79.5310
λ_{24}	433.7800	-284.3770	-120.6210	λ_{49}	535.5960	41.3320	83.2630
λ_{25}	508.2230	-211.3060	-105.6290	λ_{50}	525.0020	96.1410	78.7560

Tabela 3.25 – Pontos da grade de calibração na base E_{b1} .

A matriz transformação x_{bb1} fica da forma:

Pârametros de x_{bb1}	Valor ideal	valor obtido
P_{bb1_x} (mm)	-10	-11.7708
P_{bb1_y} (mm)	-60	-57.5568
P_{bb1_z} (mm)	0	2.3322
φ (rad)	-A	-7.5093
ϑ (rad)	0	0.0054
ψ (rad)	A+0.5236	7.8982

Tabela 3.26 – Parâmetros de x_{bb1}

Pode-se observar, a partir dos valores estimados, que $\vartheta \approx 0$ e $\varphi + \psi = 0,3890 \text{ rad}$ ou $22^\circ 16'48''$.

Pode-se verificar na figura 3.18 como os parâmetros de erro de posicionamento convergem para seus respectivos valores a partir do 4º passo de iteração.

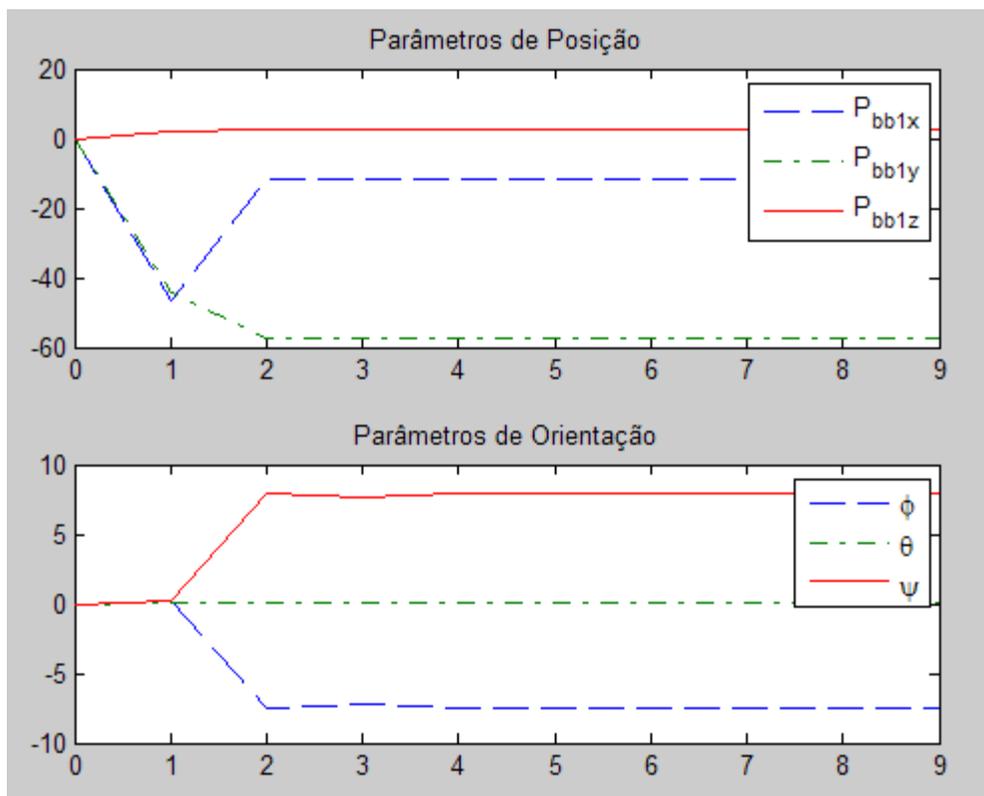


Figura 3.18 – Convergência dos parâmetros de x_{bb1} .

Também para este método de calibração foi verificado que a ordem de medida e o número de pontos da grade de calibração devem ser os mesmos tanto na base original E_b quanto na base atual E_{b1} , caracterizando restrições ao processo de tomada de dados.

Capítulo 4

Replanejamento de trajetória

O replanejamento de trajetória consiste em obter a mesma trajetória previamente programada na etapa de planejamento na base E_b , mas nesse momento o manipulador deve encontrar-se na base E_{b1} , deslocado em relação à primeira base. Para isto, devem-se utilizar as informações do erro de posicionamento contidas na matriz T_{bb1} obtida na etapa de calibração.

Considerando o manipulador em sua localização original E_b , a sua trajetória ideal é dada por:

$$x_{bd} = k(\theta_{bd}), \text{ representação na cinemática direta.} \quad (4.1)$$

$$\theta_{bd} = k^{-1}(x_{bd}), \text{ representação na cinemática inversa.} \quad (4.2)$$

onde $k(\cdot)$ indica a função cinemática direta.

De acordo com a cinemática diferencial, a relação descrita em (4.1) pode ser escrita como:

$$\dot{x}_{bd} = J(\theta_{bd})\dot{\theta}_{bd} \quad (4.3)$$

Considerando um deslocamento do manipulador em relação à localização original E_b , isto é, encontrando-se na base atual E_{b1} , a nova trajetória desejada é dada por:

$$x_{b_1d} = T_{bb1}^{-1}x_{bd} \quad (4.4)$$

e

$$x_{b_1d} = k(\theta_{b_1d}), \text{ representação na cinemática direta.} \quad (4.5)$$

$$\theta_{b_1d} = k^{-1}(x_{b_1d}), \text{ representação na cinemática inversa.} \quad (4.6)$$

Derivando (4.5), tem-se:

$$\dot{x}_{b_1d} = T_{bb_1}^{-1} \dot{x}_{bd} \quad (4.7)$$

Substituindo (4.3) em (4.7), tem-se:

$$\dot{x}_{b_1d} = T_{bb_1}^{-1} J(\theta_{bd}) \dot{\theta}_{bd} \quad (4.8)$$

equivalentemente:

$$J(\theta_{b_1d}) \dot{\theta}_{b_1d} = T_{bb_1}^{-1} J(\theta_{bd}) \dot{\theta}_{bd} \quad (4.9)$$

Consequentemente a velocidade das juntas para a trajetória desejada na localização atual E_{b_1} é dada por:

$$\dot{\theta}_{b_1d} = J^{-1}(\theta_{b_1d}) T_{bb_1}^{-1} J(\theta_{bd}) \dot{\theta}_{bd} \quad (4.10)$$

4.1 – Simulação

Para simular o replanejamento foi utilizada uma trajetória circular dada através das seguintes equações:

$$x = r \sin(wt)$$

$$y = 800$$

$$z = r \cos(wt)$$

onde r é o raio da circunferência, w é a frequência angular da trajetória e t é o tempo em segundos.

Para efeito das simulações foram utilizados os seguintes valores:

Variável	Valor	Unidade
r	450.0	mm
w	$2\pi / 10$	rad/s

Tabela 4.1 – Valores utilizados para simulação.

Para efeito de testes de validade do algoritmo proposto foram realizados três tipos de simulação:

- Simulação utilizando a matriz x_{bb1} obtida através dos pontos gerados computacionalmente sem ruído.
- Simulação utilizando a matriz x_{bb1} obtida através dos pontos da grade com erro apenas de posição.
- Simulação utilizando a matriz x_{bb1} obtida através dos pontos da grade com erro de posição e rotação.

4.1.1 - Simulação utilizando a matriz x_{bb1} obtida através dos pontos gerados computacionalmente sem ruído

Dados da matriz x_{bb1} :

Parâmetros de x_{bb1}	valor
P_{bb1_x} (mm)	50
P_{bb1_y} (mm)	100
P_{bb1_z} (mm)	150
φ (rad)	-0.7835
ϑ (rad)	0
ψ (rad)	0.0019

Tabela 4.2 – Parâmetros de x_{bb1} .

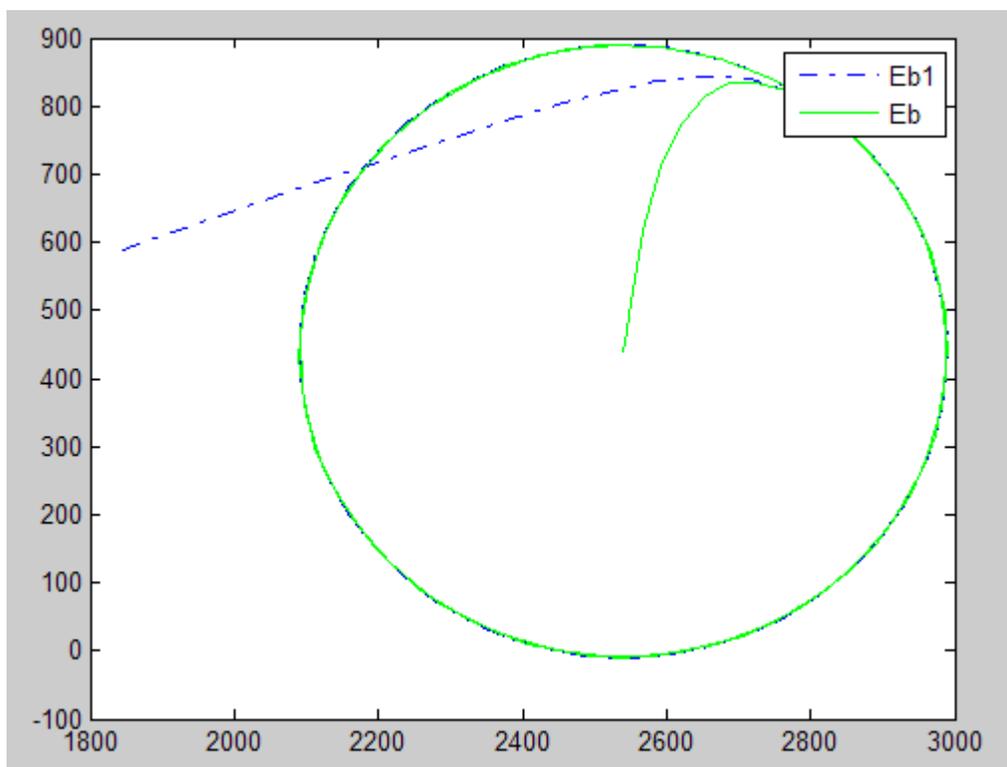


Figura 4.1 – Trajetórias nas bases E_b e E_{b1} .

4.1.2 - Simulação utilizando a matriz x_{bb1} obtida através dos pontos da grade com erro de posição

Dados da matriz x_{bb1} :

Pârametros de x_{bb1}	valor
P_{bb1_x} (mm)	1.8808
P_{bb1_y} (mm)	-110.5450
P_{bb1_z} (mm)	2.2211
φ (rad)	-4.9682
\mathcal{J} (rad)	-0.0074
ψ (rad)	4.9746

Tabela 4.3 – Parâmetros de x_{bb1} .

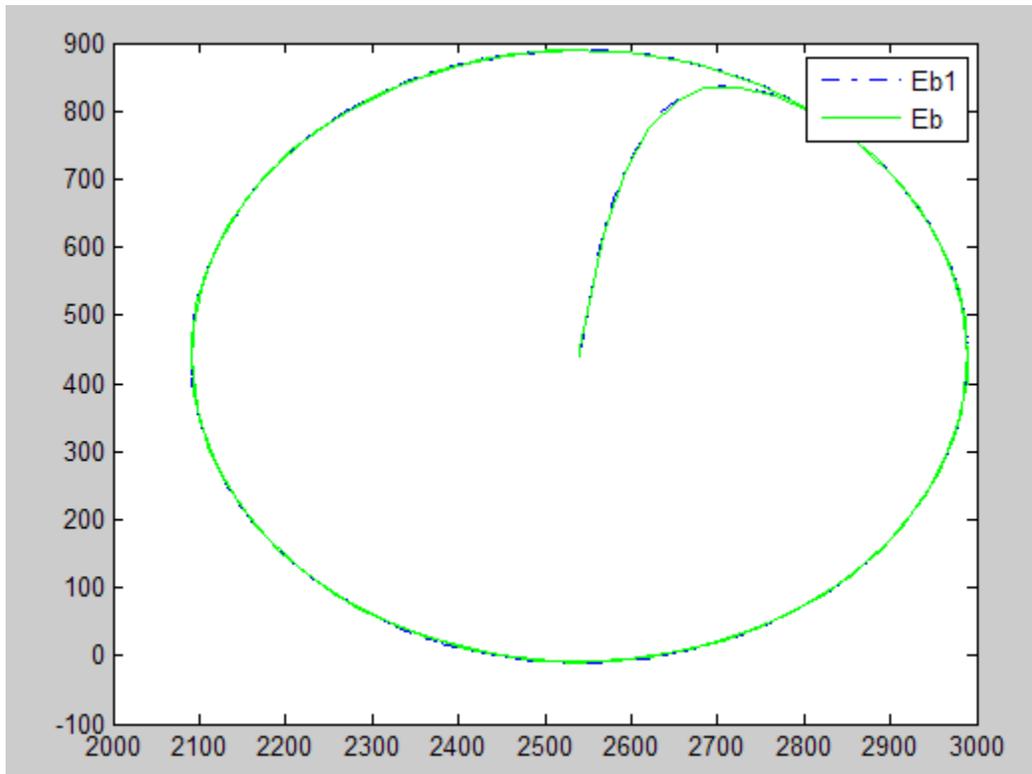


Figura 4.2 – Trajetórias nas bases E_b e E_{b1} .

4.1.3 - Simulação utilizando a matriz x_{bb1} obtida através dos pontos da grade com erro de posição e rotação

Dados da matriz x_{bb1} :

Pârametros de x_{bb1}	valor
P_{bb1_x} (mm)	-11.7885
P_{bb1_y} (mm)	-57.2759
P_{bb1_z} (mm)	2.5679
φ (rad)	5.1285
ϑ (rad)	0.0058
ψ (rad)	-4.7400

Tabela 4.4 – Parâmetros de x_{bb1} .

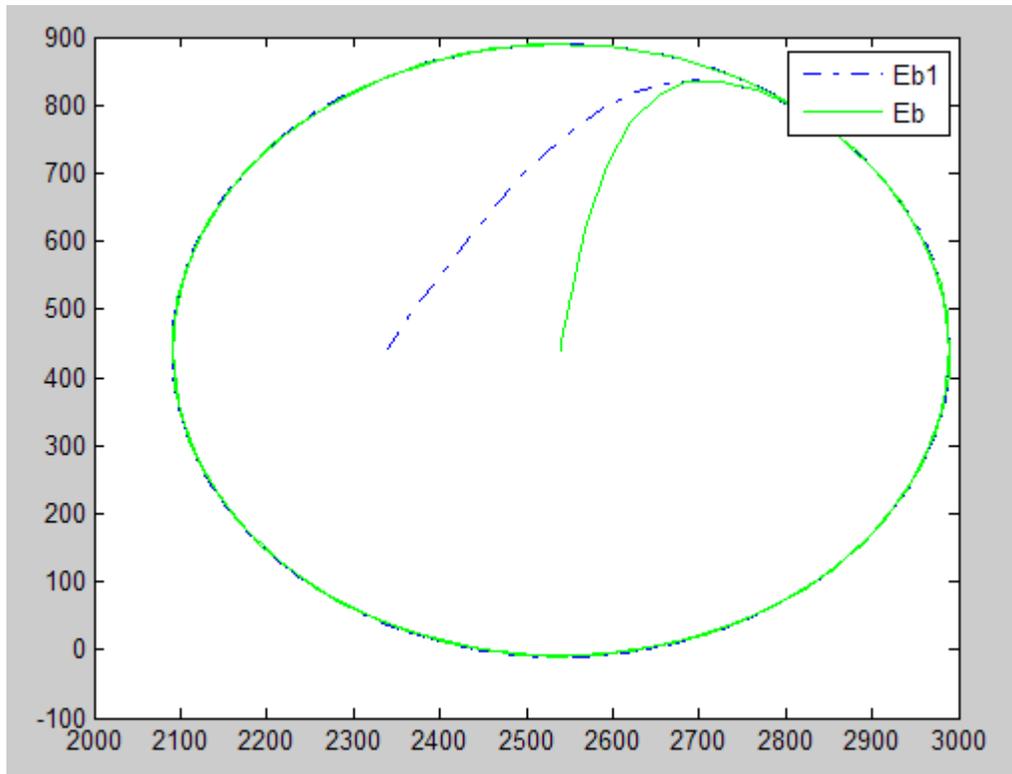


Figura 4.3 – Trajetórias nas bases E_b e E_{b1} .

De acordo com as simulações realizadas, nota-se que o modelo gerado computacionalmente com rotação de $\pi/4$ sobre o eixo z e com maior erro de posição, indicado em 4.1.1, foi o que demorou mais a rastrear a trajetória. Já o modelo com apenas erro de posição, indicado em 4.1.2, foi o que conseguiu rastrear a trajetória mais rapidamente.

Capítulo 5

Ambiente de realidade virtual

5.1 - Apresentação do modelo

O modelo utilizado como ambiente de realidade virtual será baseado nos arquivos do pacote *puma3dfiles* colocado a disposição no site da *Math Works* através do link: <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=14932&objectType=File>. O ambiente será utilizado para visualizar e simular em laboratório as seguintes etapas do projeto: planejamento, calibração e replanejamento de tarefas em ambientes hostis.

O modelo do robô foi desenvolvido utilizando um aplicativo de CAD (*Computer-Aided Design*) e o deslocamento dos elos e das juntas do robô foi desenvolvido através do *script* em *Matlab*: *puma3d.m*, que possui código aberto, com isso torna-se possível adicionar, remover e alterar as funcionalidades a seu ambiente.

Modelo antes das alterações:

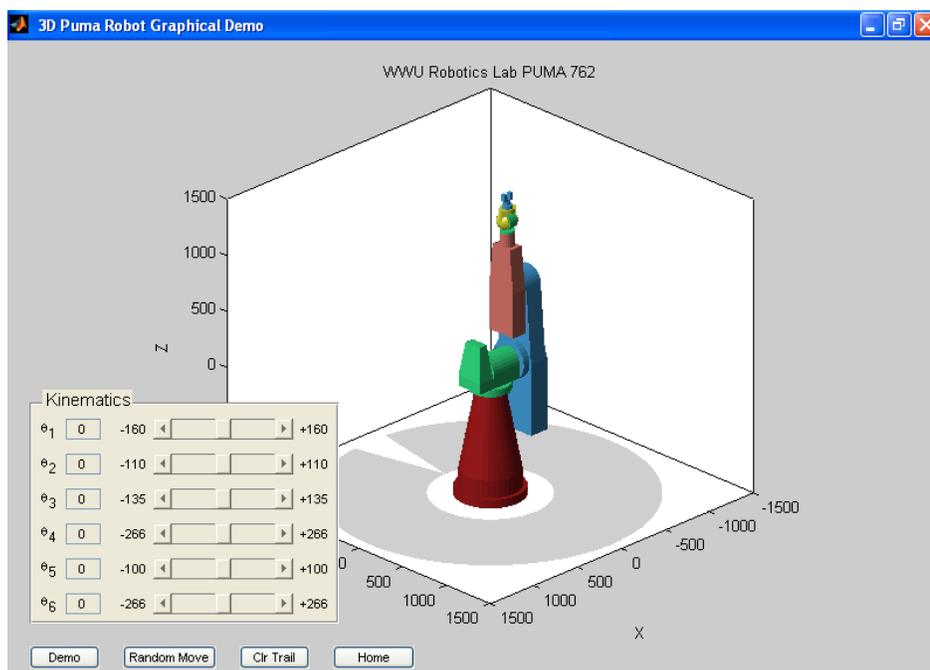


Figura5.1 - robô em sua posição inicial

FUNÇÕES:

- **Demo**
Apresenta na tela uma trajetória de demonstração em espiral do efetuador.
- **Random Move**
Apresenta na tela um movimento aleatório do robô.
- **Clr Trail**
Apaga a trajetória do efetuador.
- **Home**
Leva todos os elos do robô para as suas respectivas posições iniciais.
- **Kinematics**
Painel com o controle das juntas do robô.

As informações dos vértices e das faces de cada elo do robô estão contidas no arquivo *linksdata.mat* em forma de matrizes numéricas. Logo, para alterar as suas configurações basta modificar esse arquivo.

Para alterar as configurações dos elos ou mesmo criar novos objetos para serem inseridos no ambiente de realidade virtual, pode-se utilizar um aplicativo de *CAD*, neste projeto foi adotado o *AutoCAD2008*. Antes de exportar os elos para o *AutoCAD*, deve-se salvar as informações que constam no arquivo *linksdata.mat* em arquivos com extensão *stl*, que é uma extensão em formato de texto. Cada elo necessita ser armazenado em um arquivo *.stl* diferente. Ao importar os arquivos para o ambiente do *AutoCAD*, foi utilizado um *plug-in* em sua versão *trial*, chamado *STL Import for AutoCAD version 1.0*, desenvolvido pela *Sycode*.

A seguir estão apresentados todos os elos do robô exportados para o *AutoCAD2008*.

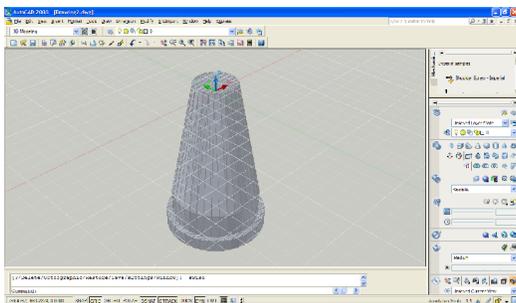


Figura 5.2 - Elo 1.

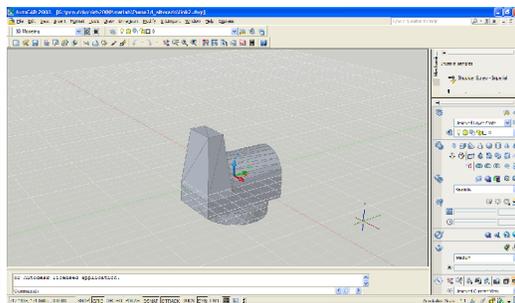


Figura 5.3 - Elo2.

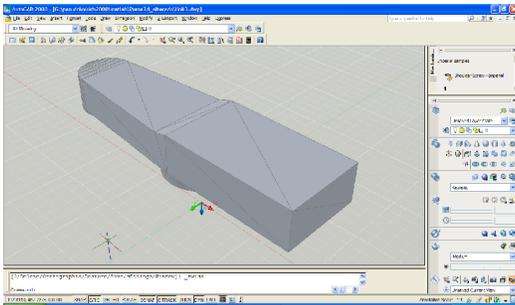


Figura 5.4 - Elo 3

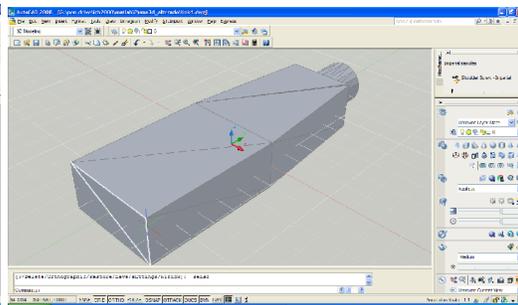


Figura 5.5 - Elo 4

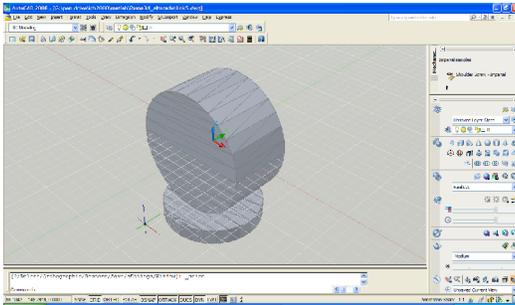


Figura 5.6 - Elo 5.

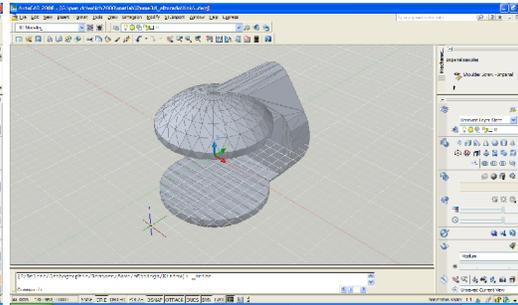


Figura 5.7 - Elo 6.

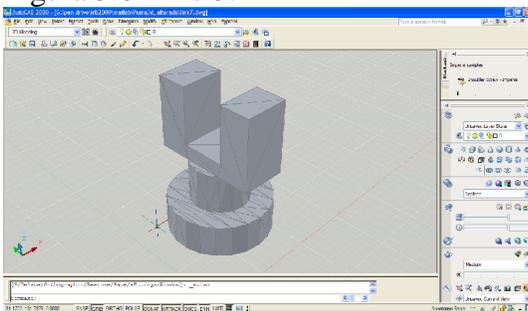


Figura 5.8 - Elo 7.

A extensão dos arquivos dentro do ambiente do *AutoCAD* é *dwg*, que é um formato nativo de diversas linguagens de CAD. Para exportar os modelos alterados ou criados dentro do ambiente de CAD para o ambiente do *Matlab*, deve-se utilizar o comando *stlout*. Estando novamente no ambiente do *Matlab*, deve-se agora gravar as informações dos arquivos *stl* para o arquivo *linksdata.mat*, para ser usado no script *puma3d.m*.

5.2 – Simulação

Nesta parte do projeto, foi utilizado o ambiente de realidade virtual para visualizar a calibração do manipulador através da grade de calibração e também o replanejamento da trajetória.

5.2.1 - Visualização da calibração através do ambiente de realidade virtual

Foi adicionado o botão: *calibration*, que permite a visualização do manipulador tocando os pontos da grade de calibração.

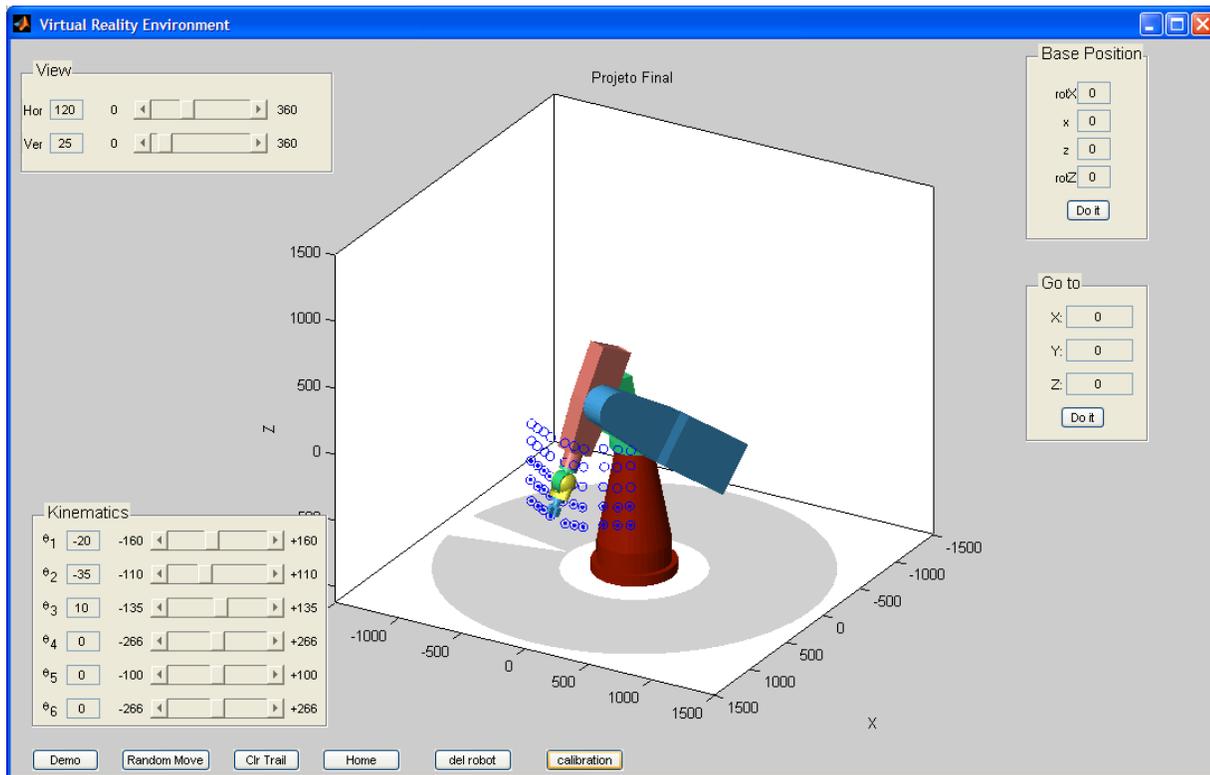


Figura 5.9 - Visualização do robô tocando os pontos na base E_b .

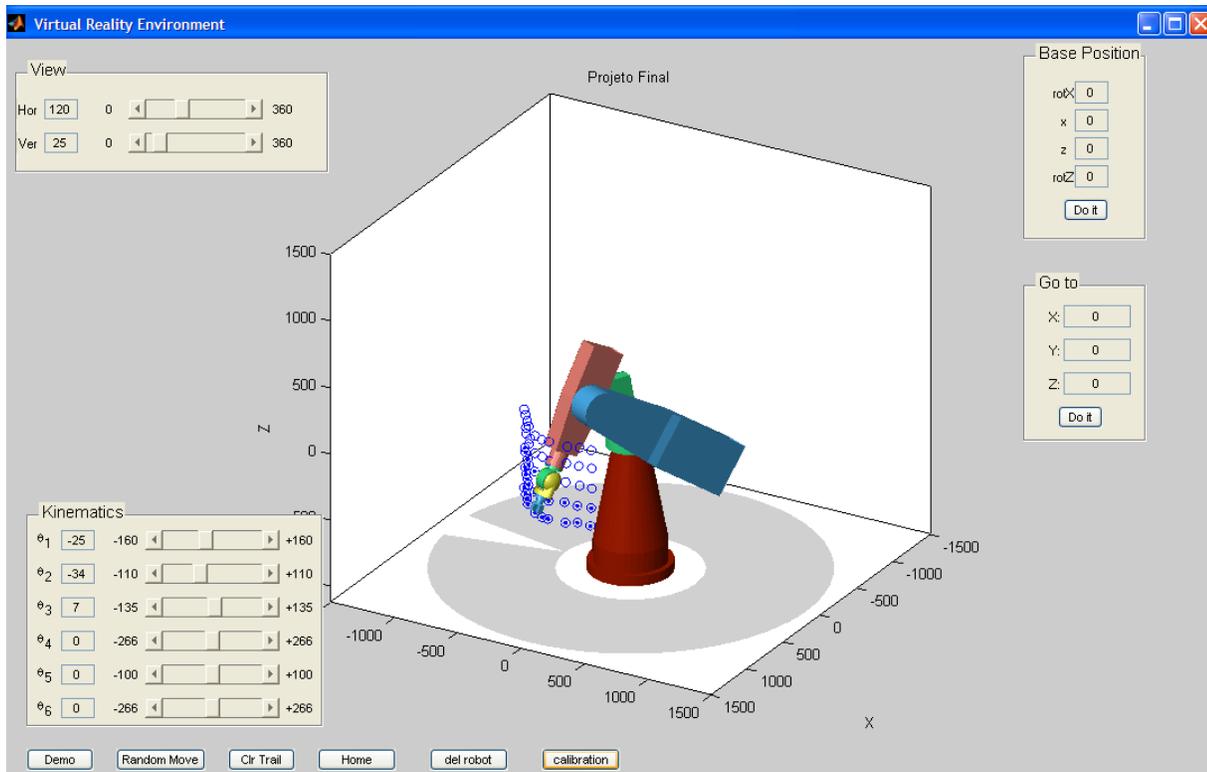


Figura 5.10 - Visualização do robô tocando os pontos na base E_{b1} .

Foram feitas algumas modificações no ambiente de realidade virtual, a fim de auxiliar nos testes de simulação:

- *View*:

O painel *View* tem a função de possibilitar a mudança dos parâmetros horizontal e vertical de visualização do sistema. Sendo utilizados os valores *default* 120 e 25 para os parâmetros horizontal e vertical, respectivamente.

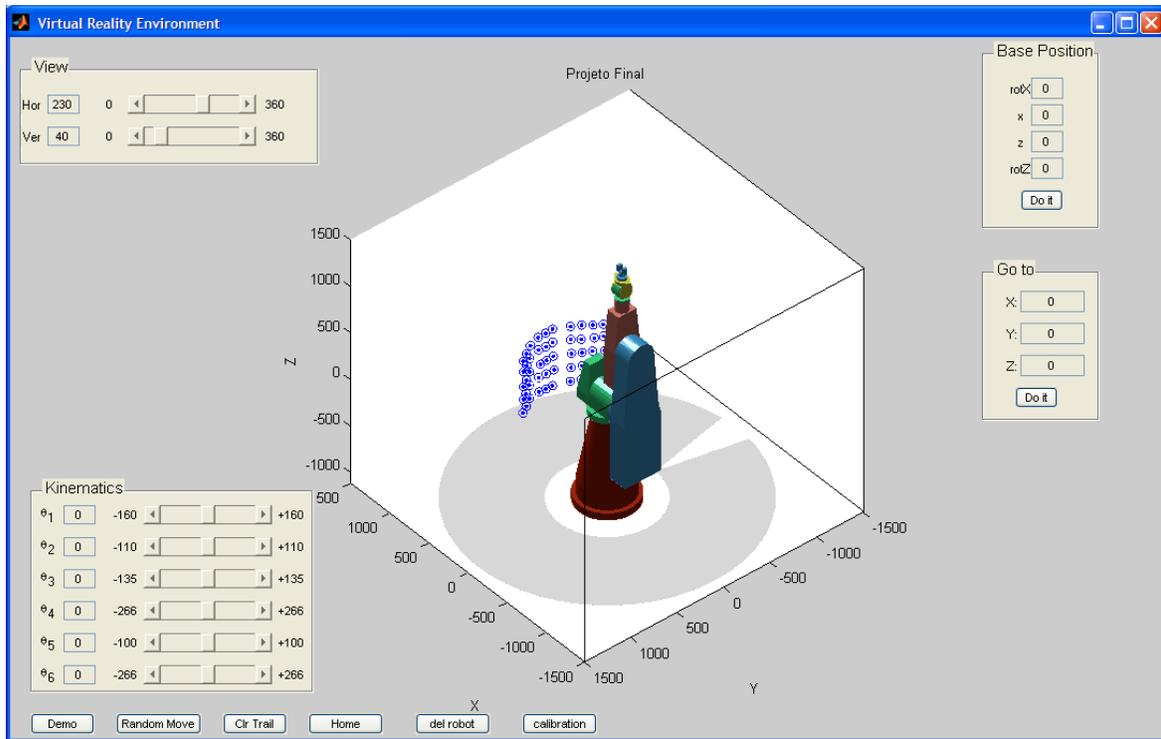


Figura 5.11 - Parâmetros da *View* alterados para Hor = 230 e Ver = 40.

- *Base Position:*

O painel *Base Position* possibilita alterar os parâmetros da base, como sua localização nos eixos x e z, bem como sua rotação em relação aos mesmos eixos.

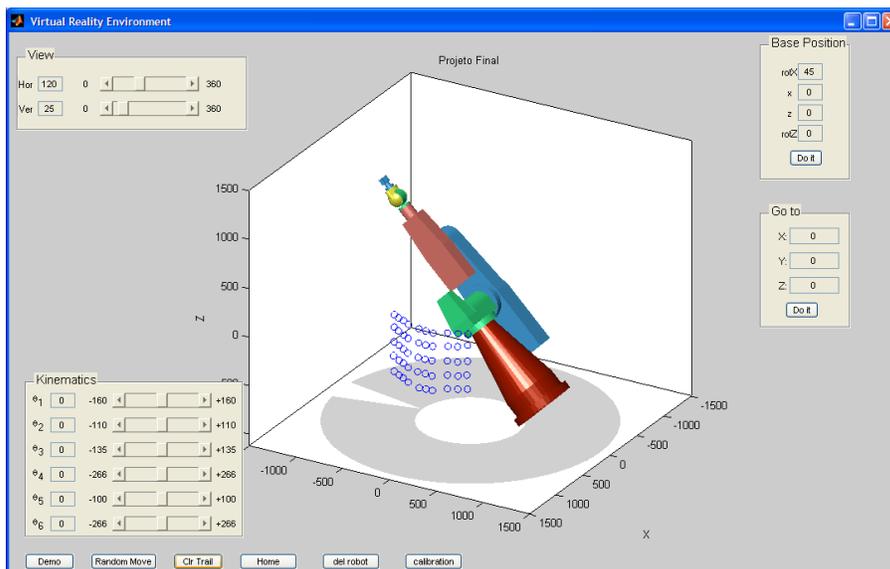


Figura 5.12 - Rotação de 45° do robô em relação ao eixo x.

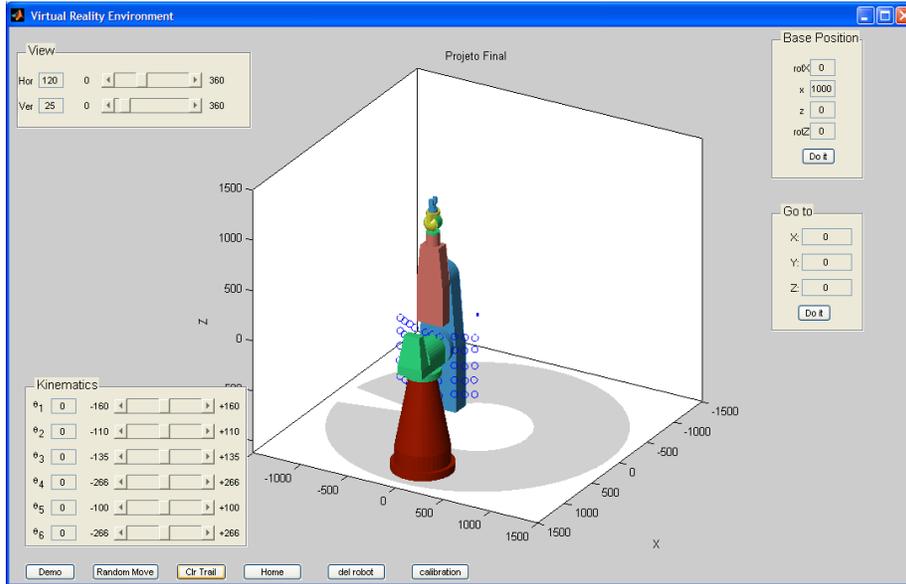


Figura 5.13 -Deslocamento do robô de 1000 unidades em relação ao eixo x.

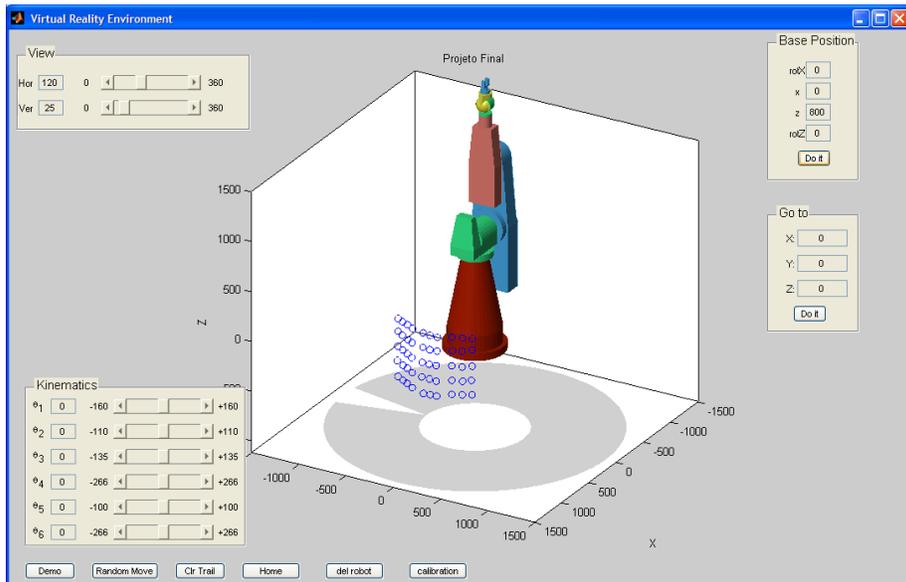


Figura 5.14 - Deslocamento do robô de 800 unidades em relação ao eixo z.

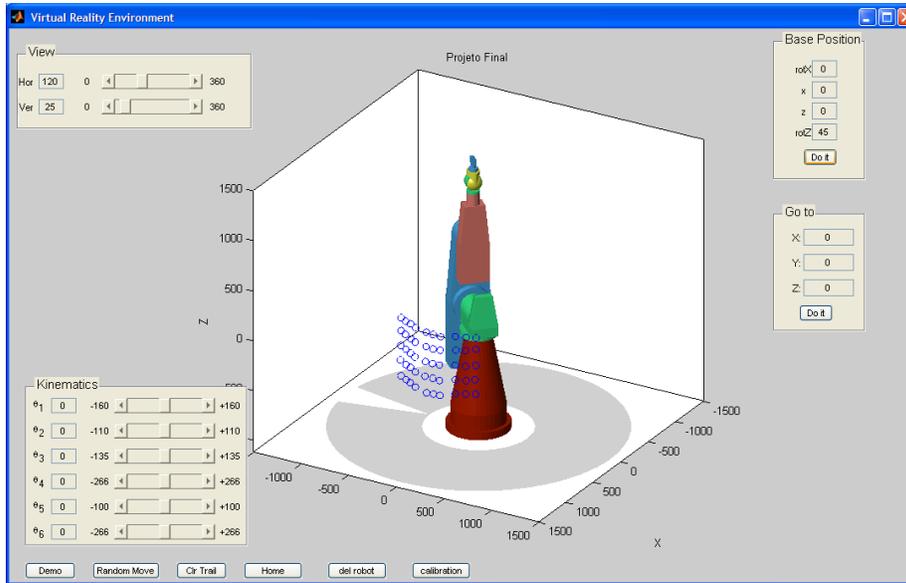


Figura 5.15 - Rotação de 45° do robô em relação ao eixo z.

- **GO TO:**

O painel *GO TO*, que possibilita movimentar o efetuador até a posição indicada pelos parâmetros de coordenadas nos eixos x, y e z, fazendo-se uso de cinemática inversa. Através desse painel também é possível realizar uma trajetória utilizando um vetor de pontos. Vale ressaltar que os pontos devem respeitar as restrições e limitações do sistema.

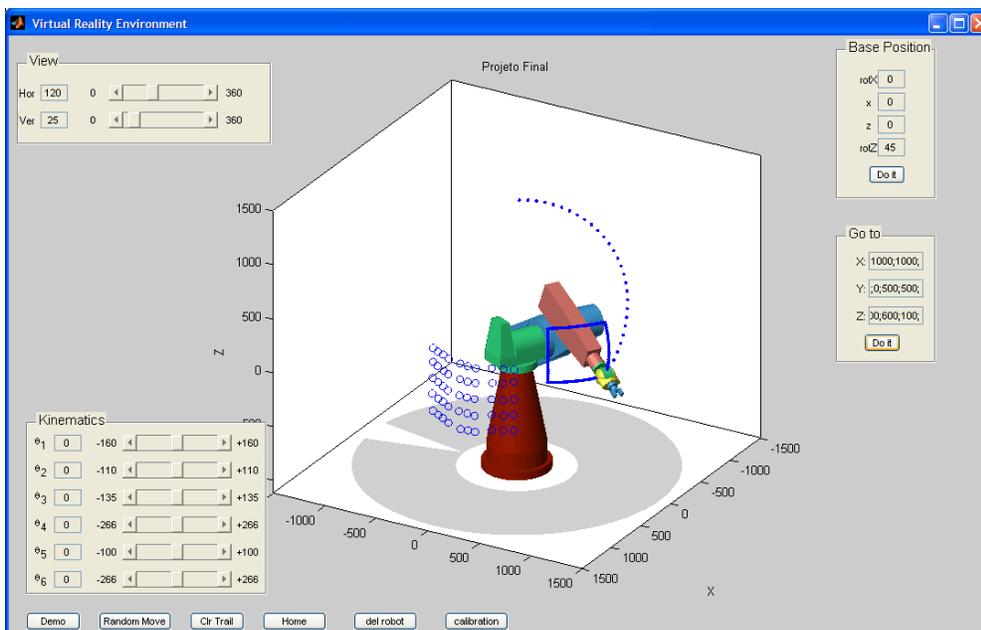


Figura 5.16 - Realização da trajetória passando pelos pontos: (1000,500,100), (1000,0,100), (1000,0,600), (1000,500,600) e finalmente retornando ao ponto (1000,500,100).

5.2.2 - Visualização do replanejamento através do ambiente de realidade virtual:

Primeiramente foi adicionado o botão *Plan* para apenas se ver a simulação do planejamento de uma trajetória.

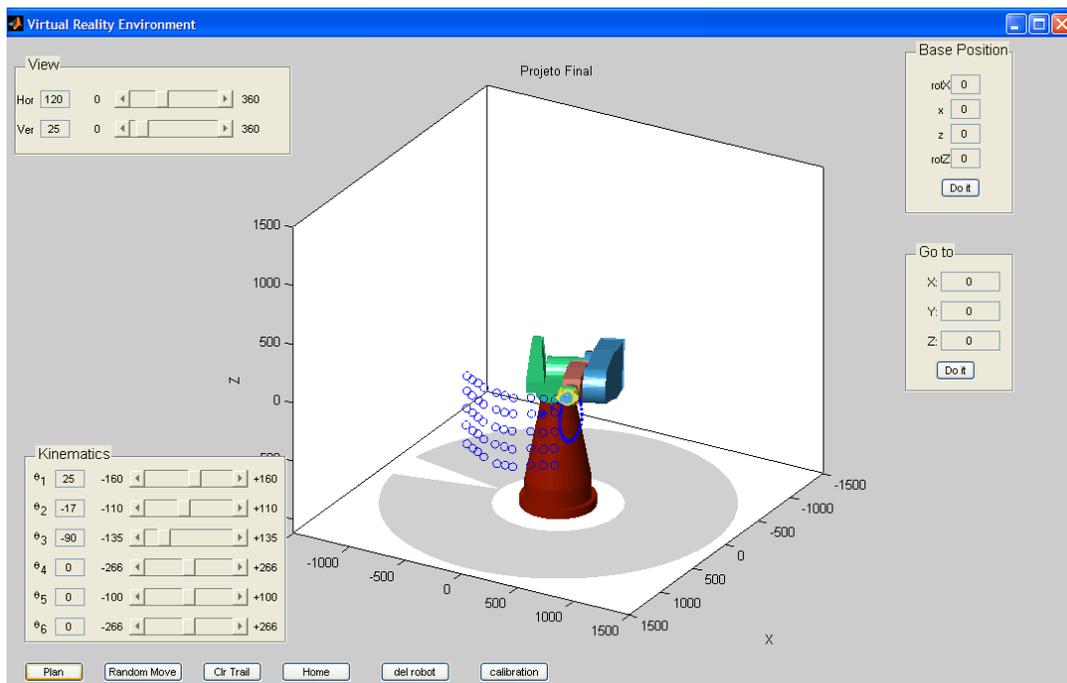


Figura 5.17 - Visualização do planejamento da trajetória na base E_b .

Depois de conseguir visualizar o planejamento de uma trajetória, foi substituído o botão *Plan* pelo botão *Plan/Replan*, que possibilita a visualização do planejamento e do replanejamento da trajetória.

Inicialmente o robô irá executar a trajetória na sua localização original E_b e posteriormente irá deslocar a sua base, de acordo com os parâmetros definidos na etapa de calibração, e deverá executar o replanejamento da trajetória na localização atual E_{b1} .

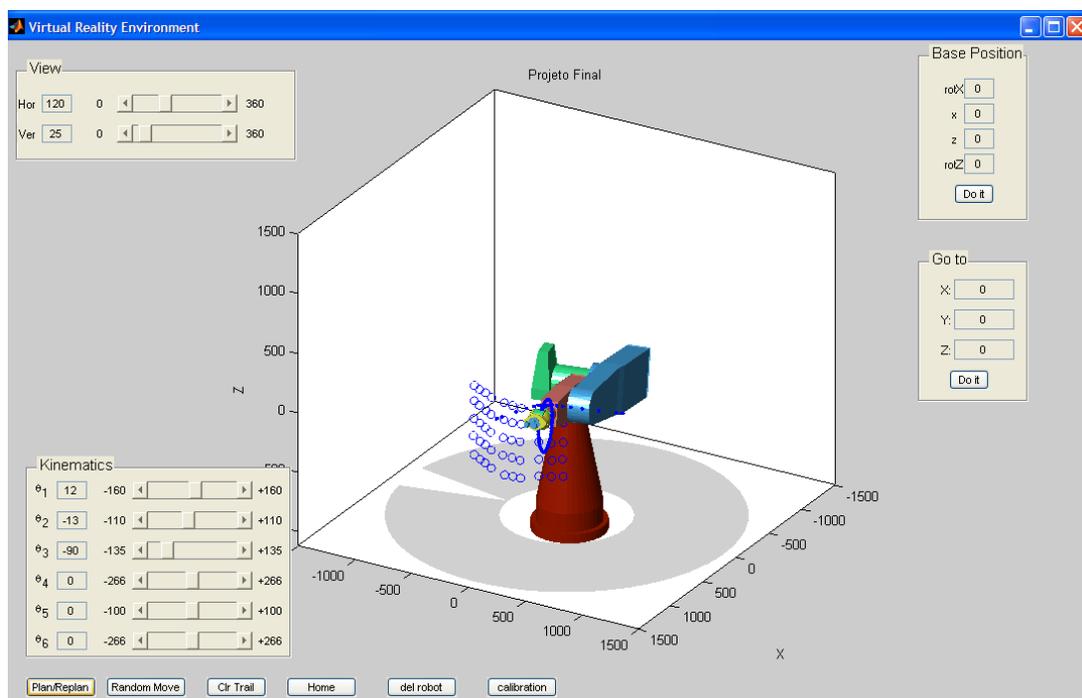


Figura 5.18 – Visualização da trajetória executadas nas bases E_b e E_{b1} .

Capítulo 6

Conclusão

Neste trabalho foram abordados dois métodos para calibração a fim de se obter a matriz transformação T_{bb1} através de processos de mínimos quadrados. Foi utilizada uma grade formada por pontos, que deveriam ser tocados pelo manipulador na localização original e atual.

Pode-se observar, a partir das simulações que foram realizadas, que o algoritmo proposto apresentou-se de forma satisfatória ao determinar os parâmetros de posição e orientação da matriz transformação entre as localizações já mencionadas. O método utilizado conseguiu obter de forma idêntica a matriz de transformação T_{bb1} gerada computacionalmente, como era de se esperar, a partir do 4º passo de iteração. Em relação ao ruído gerado computacionalmente, o método apresentou variações mínimas nos parâmetros desejados da matriz transformação, utilizando-se diversas quantidades de pontos da grade (5, 10, 14 e 18 pontos). Ao se utilizar o manipulador *Zebra Zero*, o método comportou-se também de forma satisfatória tanto para apenas erro de posição quanto para erro de posição + rotação.

Foi verificado que a ordem de medida e o número de pontos da grade de calibração devem ser os mesmos tanto na base original quanto na base atual, caracterizando restrições ao processo de tomada de dados.

A técnica utilizada para replanejamento de trajetória apresentou-se de forma satisfatória, como é possível de se ver na etapa de simulação, conseguindo rastrear a trajetória planejada.

O ambiente de realidade virtual foi utilizado para visualizar as tarefas realizadas pelo manipulador. Com ele foi possível ver a calibração sendo executada, assim como o planejamento e replanejamento de trajetórias. O modelo utilizado para realidade virtual foi baseado no *script puma3d.m* colocado a disposição no *site* do *Math Works*.

5.1 Propostas para trabalhos futuros

Criação de novos modelos de robôs utilizando ferramenta de *CAD* e exportá-los para o *Matlab* para serem utilizados no ambiente de realidade virtual.

Utilizar controle por joystick com realimentação de força ou controle híbrido (posição e força) para auxiliar na tomada de medidas dos pontos da grade de calibração nos ambientes pouco estruturados.

Bibliografia

- [1] CAMPOS, B.A.N., *Simulador para modelagem e calibração de robôs industriais*. M.Sc. dissertation, Universidade de Brasília, Dezembro 2006.
- [2] LEITE, A. C., MARQUES, G. C. & LIZARRALDE, F. C., *Calibração remota e replanejamento de trajetória para sistemas robóticos*. 2008.
- [3] MARQUES, G.C., *Calibração remota de sistemas robóticos utilizando sensores internos e externos*. M.Sc. dissertation, Universidade Federal do Rio de Janeiro, Julho 2005.
- [4] SCIAVICCO, L. & SICILIANO, B., *Modelling and Control of Robot Manipulators*. McGraw-Hill, 1996.
- [5] SEEREERAM, S. and WEN, J. T., *A global approach to path planning for redundant manipulators*, IEEE Transactions on Robotics and Automation, vol. 11, no. 1, pp. 152–160, 1995.

Apêndice A

Método de Newton-Raphson

O método de Newton-Raphson é utilizado para se determinar as raízes de uma função. Entretanto, esse método apresenta duas desvantagens:

- nem sempre converge
- precisa do cálculo da derivada da função, o que nem sempre é uma tarefa fácil.

Para desenvolver este método, utiliza-se a expansão de uma função em série de Taylor em torno do ponto x_0 . Ela é escrita como:

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2!} f''(x_0) + \dots \quad (\text{A.1})$$

Mantendo apenas os dois primeiros termos da série tem-se:

$$f(x) \approx f(x_0) + (x - x_0)f'(x_0) \quad (\text{A.2})$$

Esta é a equação de uma reta que passa pelo ponto $f(x_0)$ com inclinação $f'(x_0)$, isto é, ela é tangente à curva no ponto x_0 .

Supondo que a função $f(x)$ seja bem aproximada por uma reta, o ponto que essa reta cruza o eixo x , está próximo ao ponto que a função cruza o eixo x . Este ponto é dado por:

$$\begin{aligned} 0 &= f(x_0) + (x - x_0)f'(x_0) \\ -f(x_0) &= (x - x_0)f'(x_0) \\ x &= x_0 - \frac{f(x_0)}{f'(x_0)} \end{aligned} \quad (\text{A.3})$$

Esse ponto x é então usado no lugar de x_0 como um novo valor inicial melhorando assim a aproximação. A metodologia de se utilizar um valor para gerar um outro melhor é chamada de iteração. A figura abaixo ilustra a idéia do método.

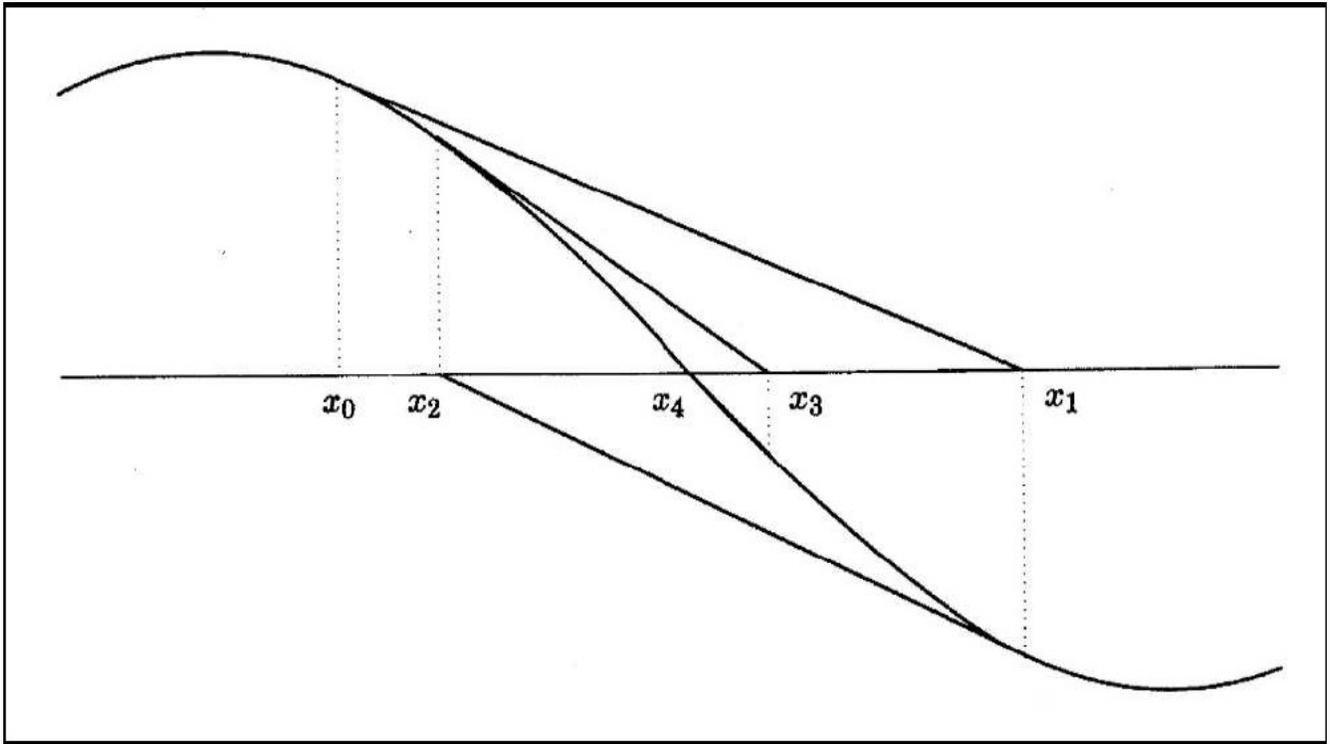


Figura A.1: Newton-Raphson em ação. Começando com x_0 , as sucessivas iterações aproximam-se do zero da função. A posição de x_4 e do zero real da função são indistinguíveis nesta escala.

Mudando ligeiramente a notação, podemos calcular o $(i + 1)$ ésimo valor usando o i -ésimo valor através da expressão:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \tag{A.4}$$

No Newton-Raphson atribui-se um valor inicial e dependendo deste valor nem sempre o método converge, pois podemos ter o caso em que a reta tangente à função no ponto inicial não representa bem a função naquele ponto.

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.