Desenvolvimento e implementação de sistema de calibração de ferramenta de robô industrial

Vasco Daniel Quelhas de Sousa Marques Branco

Dissertação de Mestrado

Orientadores:

Prof. Paulo Augusto Ferreira de Abreu

Prof. António Manuel Ferreira Mendes Lopes



Mestrado Integrado em Engenharia Mecânica

Opção de Automação

Porto, julho de 2015

Informação pessoal



Nome: Vasco Daniel Quelhas de Sousa Marques Branco

Número de aluno: 201005339

Ano da primeira inscrição: 2010

E-mail institucional: em10099@fe.up.pt

E-mail alternativo: vascodani@sapo.pt

"... eu quero aprender sempre mais, não por vaidade, nem só para agradar aos meus familiares e professores, mas para ser útil a mim mesmo, à minha família, à sociedade e à minha Pátria..."

Excerto da oração Auxílio aos Estudantes a Santa Catarina de Alexandria.

Resumo

Em manipuladores robóticos a correta definição do referencial da ferramenta é fundamental para o sucesso dos processos. Os sistemas de calibração de ferramenta visam adquirir essa informação e transmiti-la ao controlador do robô.

Este trabalho inicia-se com o levantamento do estado da arte de dispositivos de calibração de robôs industriais, seguindo-se uma recolha de informação sobre os fundamentos teóricos comuns aos procedimentos de calibração da posição do referencial da ferramenta. É também feita uma comparação da performance dos modelos matemáticos de duas metodologias que seguem o mesmo procedimento de calibração. O teste consiste em calcular o erro na determinação da posição do referencial da ferramenta, quando introduzida uma perturbação nos pontos recolhidos para calibração.

Depois, reveem-se os procedimentos de calibração disponíveis numa célula robótica, e desenvolvem-se estratégias para aumento da confiabilidade nos mesmos. Além disso, exploram-se novos métodos, adaptados de sugestões de outros autores. Os procedimentos de calibração da posição do referencial da ferramenta que são abordados não recorrem a dispositivos de medição ou localização externos, ou seja, usam o próprio robô como instrumento de medição. Na execução dos procedimentos de calibração, é instalada uma esfera na ferramenta do robô, para contacto com um ponto ou uma superfície de referência. Para a implementação destes procedimentos o robô é controlado quer em posição quer em força. Os procedimentos de calibração são analisados tendo em conta o seu modelo teórico, sendo depois programados e simulados. Posteriormente, os procedimentos são testados experimentalmente na célula robótica disponível, sendo utilizadas duas ferramentas distintas.

Com a análise dos resultados obtidos através das simulações e experimentalmente, os procedimentos são comparados quantitativa e qualitativamente, sendo identificado o mais expedito. Por último, este procedimento é integrado num sistema semiautomático de calibração de ferramenta, que tira partido da disponibilidade de controlo de força do robô. Este sistema inclui ainda um procedimento de determinação da massa e centro de massa da ferramenta e um procedimento para obtenção da orientação do referencial da ferramenta.

Development and implementation of an industrial robot tool calibration system

Abstract

The right definition of the tool frame is critical to the success of the processes when using robot manipulators. Calibration procedures are used to get this information and transmit it to the robot controller.

This work begins with a research on the state of the art, followed by collecting theoretical information on the most common calibration procedures. It is also conducted a sensibility comparison of different methodologies, when introduced a disturbance in the points collected for calibration, regarding the error in determining the tool frame.

Then the existing calibration procedures in the available robotic cell are checked so that some strategies can be developed in order to increase its reliability. New methods suggested by other authors were also explored. The studied procedures don't use external measuring devices, they use the robot itself as a measuring instrument instead. This implies the use of surfaces or reference points and coupling a sphere in the tools during the execution of the procedures. Some procedures use force control which is available in the robotic cell. All procedures are collected and analysed in its theoretical basis and then programmed and simulated using appropriated software. At the end of this phase, the procedures are tested experimentally on the robotic cell using two different tools.

With the analysis of the results from the simulations and experiments, the procedures are compared so that those who best meet the cell's needs can be selected. Finally, a semi-automatic tool calibration system, comprising one of the studied calibration procedures with force control is implemented.

Agradecimentos

Agradeço aos professores, Paulo Abreu e António Mendes Lopes, orientadores deste trabalho, pela dedicação e acompanhamento ao desafio que me propuseram, pelo tempo que despenderam para discussão de ideias, pela disponibilidade que tiveram em atender às dificuldades que fui encontrado e pelo apoio que deram na resolução dos problemas que foram surgindo, com a partilha de conhecimentos sem os quais certamente seria mais difícil alcançar o sucesso neste projeto.

Ao professor Francisco Freitas, responsável pela unidade curricular de Dissertação em Automação, agradeço não só as iniciativas que deram a conhecer o mercado de trabalho, nomeadamente as visitas de estudo e palestras que se fizeram e se revelaram de extrema importância, mas também a sua preocupação em manter um ambiente cooperativo de trabalho entre os alunos e as críticas construtivas que ajudaram a melhorar os pontos menos positivos.

Ao senhor Joaquim Silva, assistente técnico do laboratório de óleo hidráulica, um sincero obrigado por estar sempre disposto a emprestar um pouco da sua arte aos trabalhos de oficina que este projeto envolveu.

A todos os colegas finalistas e em especial aos de Automação que partilharam comigo o espaço de trabalho pela entreajuda, dicas e companheirismo.

Aos meus pais, por investirem na minha formação, pois querem o meu bem e preocupam-se com o meu futuro, por me apoiarem nas minhas decisões e aconselharem ao longo da vida e pela paciência durante todo o meu percurso académico.

Aos meus e avós, porque são um exemplo de vida para mim, pela atenção e interesse no andamento do curso, pela espectativa, incentivos e orgulho que foram importantes para manter o ânimo nos momentos mais difíceis.

Aos meus irmãos e amigos, pelos também importantes momentos de descontração.

Índice de Conteúdos

Capítulo 1 – Introdução 1
1.1 Contextualização e motivação 1
1.2 Objetivos 2
1.3 Informação prévia para compreensão do tema 3
1.3.1 Sistemas de coordenadas em manipuladores robóticos 3
1.3.2 Referencial da ferramenta 4
1.3.3 Calibração da ferramenta 5
1.4 Estrutura da dissertação 6
Capítulo 2 - Dispositivos de calibração comerciais para robôs industriais 7
2.1 Sistemas de teste e monitorização 7
2.1.1 Sistemas com contacto 7
2.1.2 Sistemas sem contacto 9
2.2 Sistemas dedicados à calibração da ferramenta 11
2.2.1 Sistemas sem contacto 11
2.2.2 Sistema com contacto 15
2.3 Síntese dos sistemas existentes 16
Capítulo 3 – Ferramentas matemáticas de apoio à calibração do TCP 17
3.1 Cinemática direta e algoritmo de Denavit-Hartenberg 17
3.2 Calibração com geometrias de referência 20
3.2.1 Calibração com um ponto de referência 20
3.2.2 Calibração com um plano de referência 27
Capítulo 4 - Comparação de metodologias por perturbação dos pontos de calibração 31
4.1 Análise da influência da amplitude da perturbação 32
4.2 Análise da sensibilidade das metodologias à perturbação 33
4.3 Análise da influência da posição relativa dos pontos adquiridos 34
4.4 Análise do número de pontos adquiridos 35
Capítulo 5 – Análise e implementação de procedimentos de calibração do TCP 37
5.1 Análise dos procedimentos disponíveis 37
5.1.1 Método de introdução direta 37
5.1.2 Medição geométrica com o robô 38
5.1.3 Método da ABB 39
5.1.4 Aumento da performance dos procedimentos disponíveis 40
5.2 Implementação de procedimentos alternativos 42

5.2.1 Componentes utilizados 42
5.2.2 Simulações virtuais e desenvolvimento de *software* 45
5.2.2.1 RobotStudio 45
5.2.2.2 Modelo virtual da estação de trabalho 46
5.2.3 Programação e simulação de procedimentos alternativos 47
5.2.3.1 Ponto de referência – Procedimento PNT_1 48
5.2.3.2 Plano de referência horizontal – Procedimento PLN_1 49
5.2.3.3 Plano de referência vertical – Procedimento PLN_2 50
5.2.4 Problemas encontrados 51
5.3 Testes e resultados 52
5.3.1 Medição geométrica com o robô 52
5.3.2 Método da ABB 54
5.3.3 Procedimento PNT_1 55
5.3.4 Procedimento PLN_1 56
5.3.5 Procedimento PLN_2 57
5.4 Avaliação quantitativa e qualitativa dos procedimentos 58
5.5 Aplicação de sistema de calibração a um *spindle* 60
Capítulo 6 – Conclusões e perspetivas de trabalho futuro 67
Referências 71
Anexos 73
Anexo 1 – Programa em RAPID para aquisição e conversão dos pontos de calibração 73
Anexo 2 – Programa gerado em Matlab para comparação das metodologias de minimização do erro e cálculo da superfície esférica 77
Anexo 3 – Síntese dos tipos de variáveis e instruções mais relevantes para o trabalho de programação desenvolvido 83
Anexo 4 – Características do controlador virtual utilizado no RobotStudio 85
Anexo 5 – Programas gerados em RAPID para simulação e implementação dos procedimentos de calibração do TCP 87
Anexo 6 – Indicações para contornar mensagens de erro 97
Anexo 7 – Programação em RAPID de sistema de calibração de ferramenta integrando o controlo de força e o procedimento PNT_1 99

Índice de Figuras

Figura 1 – Sistemas de coordenadas do robô [3] 3
Figura 2 – Trajetória descrita pelo TCP [3] 4
Figura 3 – RENISHAW Ballbar [4] 7
Figura 4 – ABB Navigator [5] 8
Figura 5 – LEICA Absolute Laser Tracker [6] 9
Figura 6 – FARO Laser Tracker [7] 9
Figura 7 – WIEST AG GripLAB [8] 10
Figura 8 – WIEST AG LaserLAB [8] 11
Figura 9 – ABB Bulls Eyes [20] 12
Figura 10 – LEONI Advintec [9] 13
Figura 11 – DYNALOG AutoCal [10] 13
Figura 12 – WIEST AG RotoLAB [8] 14
Figura 13 – ABB Force Control Calibration [5] 15
Figura 14 – Calibração com um ponto conhecido [15] 21
Figura 15 – Calibração com um ponto desconhecido [14] 22
Figura 16 – Calibração com um plano de referência cuja orientação é conhecida 27
Figura 17 – Calibração com um plano de referência (posição e orientação são conhecidas) 28
Figura 18 – Patente US 5910719 [15] 29
Figura 19 – Influência da amplitude da perturbação no método de minimização do erro 32
Figura 20 – Influência da amplitude da perturbação no método da superfície esférica 32
Figura 21 – Sensibilidade das duas metodologias à mesma perturbação 33
Figura 22 – Erro no cálculo do TCP com pontos próximos 34
Figura 23 – Erro no cálculo do TCP com pontos afastados 34
Figura 24 – Influência do número de pontos para o método de minimização do erro 35
Figura 25 – Influência do número de pontos para o método da superfície esférica 36
Figura 26 – Interface para definição do TCP pelo método de introdução direta [23] 38
Figura 27 – Medição geométrica com o robô 39
Figura 28 – Método da ABB [18] 39
Figura 29 – Interface gráfica para o método da ABB 40
Figura 30 – Esquema de acomodação e reorientação da ferramenta [6] 41
Figura 31 – Garra PZN100 42
Figura 32 – *Spindle* XLC-070 43
Figura 33 – Superfícies e objetos de referência 43

Figura 34 – Contacto da ferramenta com uma superfície 44
Figura 35 – Caracterização da esfera 45
Figura 36 – Aspeto real da estação de trabalho 46
Figura 37 – Modelo virtual da estação de trabalho 46
Figura 38 – Deteção de proximidade e colisão entre objetos no espaço virtual 47
Figura 39 – Contacto da ferramenta com o ponto de referência 48
Figura 40 – Contacto da ferramenta com a superfície horizontal de referência 49
Figura 41 – Contacto da ferramenta com o plano vertical de referência 50
Figura 42 – Interface para configuração de juntas 52
Figura 43 – Sequência de execução do procedimento - Medição geométrica com o robô 53
Figura 44 – Recolha das coordenadas de posição de Tool0 53
Figura 45 – Sequência de execução do procedimento - Método da ABB 54
Figura 46 – Resultado do método da ABB para a garra (esquerda) e *spindle* (direita) 54
Figura 47 – Sequência de execução do procedimento - PNT_1 55
Figura 48 – Resultado do procedimento PNT_1 para *spindle* 55
Figura 49 – Sequência de execução do procedimento - PLN_1 56
Figura 50 – Resultado do procedimento PLN_ 1 para a garra (esquerda) e *spindle* (direita).. 56
Figura 51 – Sequência de execução do procedimento - PLN_2 57
Figura 52 – Resultado do procedimento PLN_2 para a garra (esquerda) e *spindle* (direita)... 57
Figura 53 – Posição do suporte de assento da esfera sobre a mesa de trabalho..... 61
Figura 54 – Modelo virtual da Garra 61
Figura 55 – Interface criada para a consola de programação..... 62
Figura 56 – Cálculo da massa e centro de gravidade da garra 62
Figura 57 – Cálculo do TCP da garra no modelo virtual (esquerda) e real (direita) 63
Figura 58 – Atribuição de TCP e orientação a uma ferramenta..... 63
Figura 59 – Suportes de fixação da ferramenta do *spindle* 64
Figura 60 – *Spindle* sobre o ponto de referência no modelo virtual (esquerda) e real (direita)64
Figura 61 – Procedimento de determinação da orientação do eixo de rotação..... 65
Figura 62 – Medição das distâncias D1 e D2 e fixação da ferramenta 66
Figura 63 – Cálculo da massa e centro de gravidade do *spindle* 66

Índice de Tabelas

Tabela 1 - Síntese dos sistemas existentes 16
Tabela 2 - Coordenadas do TCP resultantes dos procedimentos ensaiados 58
Tabela 3 - Avaliação qualitativa dos procedimentos 59

Capítulo 1 – Introdução

Este capítulo é dedicado à contextualização do tema relativo à calibração de ferramentas em manipuladores robóticos. Apresenta a motivação que levou ao desenvolvimento deste trabalho e especifica os principais objetivos a atingir, que incluem o desenvolvimento de um sistema de calibração. É ainda apresentada informação relativa a dispositivos comerciais utilizados para calibração de robôs industriais e suas ferramentas.

1.1 Contextualização e motivação

Em qualquer operação envolvendo manipuladores robóticos, a correta definição do referencial da ferramenta é fundamental para o sucesso do processo. Em operações de maquinagem, onde a tolerância dimensional das peças a produzir é muito apertada, este problema adquire maior importância.

Um dos objetivos do projeto europeu COMET [1] (*COmponents and METhods for adaptive control of industrial robots*) é desenvolver sistemas robóticos que possam competir com máquinas ferramenta de cinco eixos. O problema dos robôs no desempenho de operações de maquinagem está associado à sua falta de precisão comparativamente às máquinas CNC. As imprecisões devem-se sobretudo à existência de flexibilidade mecânica devido à menor rigidez estrutural dos manipuladores robóticos, aos efeitos de contração e dilatação térmica dos materiais, às folgas nos elementos de transmissão e a imprecisão na determinação do referencial da ferramenta ou seja, da posição e orientação da ferramenta de trabalho.

Em trabalhos desenvolvidos na área das operações de maquinagem com robôs industriais, verificou-se a existência de defeitos nas peças produzidas, que foram atribuídos a imprecisões na determinação do referencial da ferramenta [2]. Isto significa que há necessidade não só de fazer uma revisão aos procedimentos de calibração disponíveis, no sentido de desenvolver estratégias que possam aumentar a confiabilidade nos mesmos, mas também de explorar novos métodos e procedimentos que possam ser expeditos e que apresentem bons resultados. Este trabalho visa pois contribuir para os desenvolvimentos efetuados relativos a operações de maquinagem com robôs industriais, focando-se na calibração de ferramenta de um robô.

1.2 Objetivos

Como referido, a correta definição do referencial da ferramenta é fundamental para o sucesso do processo de maquinagem e foi enumerada como um dos obstáculos no contexto dos trabalhos desenvolvidos na área de maquinagem com robôs industriais. De acordo com a contextualização feita e as motivações apresentadas, o objetivo deste projeto passa por implementar e testar diferentes procedimentos de calibração da posição do referencial da ferramenta, para depois conceber e desenvolver um sistema de calibração que permita a identificação do referencial da ferramenta de uma forma precisa e expedita.

Tal desenvolvimento terá que passar por uma fase inicial de levantamento do estado da arte, seguindo-se a pesquisa e análise de procedimentos de calibração existentes. Dessa análise serão selecionados e adaptados os procedimentos que melhor se enquadrem no contexto deste trabalho. Seguir-se-á a implementação dos procedimentos selecionados com o desenvolvimento de *software* necessário à simulação virtual e à construção de *hardware* para a verificação experimental e teste dos mesmos. Tal verificação será desenvolvida numa célula robótica equipada com o robô (ABB IRB2400/16). Trata-se de um manipulador de configuração série, dispondo de seis eixos de rotação. Tem um alcance máximo de 1.5m, uma capacidade de carga de 16Kg e uma repetibilidade de posicionamento de 0.07mm. No seu elemento terminal tem acoplado o sensor de força (ATI Delta IP60) e o sistema de mudança automática de ferramenta (SCHUNK SWS-011). Na célula, está montado sobre um pedestal junto à mesa posicionadora (ABB IRBP 500C) e é controlado pelo controlador (ABB IRC5).

Finalmente, com a avaliação qualitativa e quantitativa dos procedimentos abordados será definido um sistema para calibração de ferramenta que deverá:

- **Ser dedicado à célula robótica disponível.**

  Importa calibrar as ferramentas da célula robótica no contexto dos trabalhos desenvolvidos.

- **Dispensar a utilização de dispositivos de medição ou localização externos.**

  Os sistemas de calibração que usam dispositivos externos recorrem a tecnologias dispendiosas. Em alternativa pode usar-se o próprio robô como instrumento de medição o que implica levar a ferramenta a contactar com superfícies ou pontos de referência.

- **Permitir a identificação do referencial da ferramenta de uma forma expedita tirando partido da disponibilidade do sistema de controlo de força.**

  O robô da célula robótica disponível possui controlo de força. Trata-se de uma funcionalidade que permite a movimentação manual do robô executada diretamente pelo operador, o que facilita a execução de tarefas que envolvem contacto do robô com superfícies de referência.

1.3 Informação prévia para compreensão do tema

Antes de aprofundar os sistemas de calibração propriamente ditos, é necessário referir os sistemas de coordenadas utilizados nos manipuladores robóticos, o que é o *Tool Center Point* e porque é importante a sua definição.

1.3.1 Sistemas de coordenadas em manipuladores robóticos

Os robôs são controlados em posição e velocidade. Para exprimir a posição do robô, da sua ferramenta, ou do objeto e da mesa de trabalho, no espaço tridimensional, são utilizados referenciais cartesianos. Para a calibração da ferramenta importa referir três destes referenciais: o referencial de base, o referencial da flange de montagem da ferramenta e o referencial da ferramenta, representados na Figura 1.

Nos robôs da ABB, o referencial de base é predefinido na base do robô, estando o eixo Z na vertical. Sobre este referencial podem ser definidos outros referenciais obedecendo a uma estrutura hierárquica, onde a origem de cada referencial é definida relativamente a um referencial seu antecessor. O referencial da flange, localizado no centro da flange do elemento terminal, tem o nome de Tool0. Define a posição e orientação do punho do robô relativamente ao referencial de base. O eixo Z deste referencial aponta para fora da flange, segundo a perpendicular à superfície da mesma. O referencial da ferramenta define como a ferramenta está montada na flange.

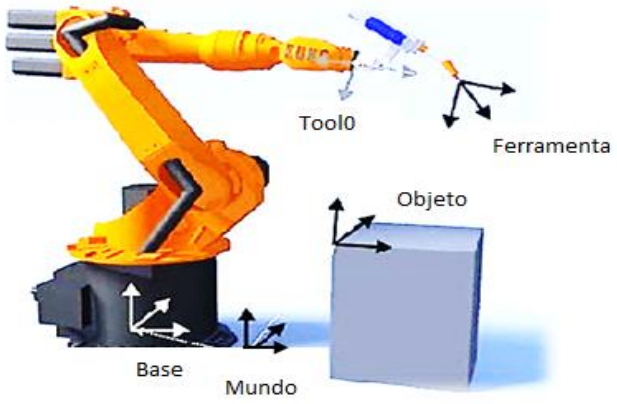


Figura 1 – Sistemas de coordenadas do robô [3]

1.3.2 Referencial da ferramenta

Um manipulador robótico é equipado com ferramentas específicas adequadas às operações que desempenha. Estas ferramentas são fixas à flange do elemento terminal do manipulador. São exemplos as tochas de soldadura, pistolas para deposição de cola ou tinta, *spindle* para operações de maquinagem e garras para manipulação de objetos.

Como exemplificado na Figura 2, o ponto da ferramenta que é comandado para seguir as trajetórias e atingir localizações no espaço de trabalho quando o robô executa uma tarefa, é conhecido por *Tool Center Point* ou TCP. Trata-se do ponto da ferramenta de trabalho que o operador define como sendo o ponto ideal para descrever as trajetórias programadas. Regra geral é o ponto central e da extremidade da ferramenta. A título de exemplo, se a ferramenta for uma tocha de soldadura, o TCP deverá ser definido na ponta do fio, junto ao bocal da tocha. Se for uma garra, o TCP deverá ser definido no centro da zona de preensão dos dedos da mesma.

Associado ao TCP existe um referencial da ferramenta cuja origem coincide com o TCP. Este referencial de ferramenta é definido relativamente ao referencial Tool0. Se não for definido nenhum referencial de ferramenta, o controlador toma para TCP a origem do referencial Tool0. A exatidão das operações efetuadas pelo robô depende diretamente do facto de o controlador saber com precisão a posição e a orientação do referencial da ferramenta relativamente ao elemento terminal do robô. O controlador pode guardar a informação do referencial da ferramenta de várias ferramentas, mas apenas uma ferramenta pode estar ativa a cada momento.

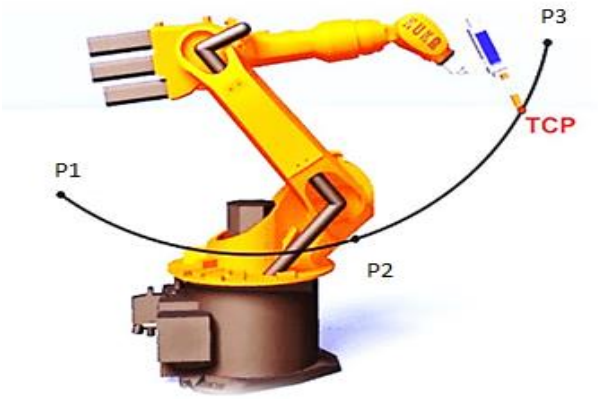


Figura 2 – Trajetória descrita pelo TCP [3]

1.3.3 Calibração da ferramenta

Do ponto de vista metrológico, a calibração é um conjunto de operações realizadas com o intuito de comparar os valores indicados por um instrumento de medição, com os valores representados por uma medida materializada de referência, a fim de lhe corrigir erros de graduação. Um instrumento diz-se calibrado quando indica dimensões concordantes com o objeto padrão, ou estas se encontrarem dentro de limites estabelecidos como admissíveis. Do ponto de vista mecânico, calibrar é afinar uma peça de um mecanismo, de modo a obter uma total homogeneidade de funcionamento das suas partes com vista ao fim para que foi criado, retificando eventuais desajustes ou irregularidades. Nos procedimentos de calibração em robótica industrial, importa olhar a calibração tanto do ponto de vista metrológico como mecânico, pois os procedimentos visam avaliar e aumentar o grau de confiabilidade nos valores lidos e nas operações realizadas pelos equipamentos.

O controlador de um manipulador série utiliza um modelo matemático para mover o último referencial da cadeia cinemática para a localização solicitada. Se não for definido um referencial de ferramenta, este último referencial é o da flange de acoplamento da ferramenta ou Tool0. Quando o operador fixa uma ferramenta necessária para uma tarefa específica, o controlador deve ser atualizado de acordo com a informação da posição e orientação do referencial da ferramenta. Só assim será capaz de levar a mesma às localizações programadas. Se essa informação não for atualizada, os movimentos serão conduzidos segundo o Tool0, como se a ferramenta não existisse, ou seja, o operador define todos os pontos de interesse e as trajetórias para a flange. Isso é possível com o método de ensino e confirmação visual e admitindo que a ferramenta tem sempre a mesma dimensão, mas não é recomendado. Se a ferramenta for alterada, o controlador não será capaz de retificar as configurações do robô para garantir que é seguida a mesma trajetória, ou seja, todo o caminho terá que ser reprogramado.

Para que ao mudar de ferramenta, o controlador altere a configuração do robô, a fim de manter as trajetórias programadas, é fundamental executar um procedimento de calibração. A calibração de uma dada ferramenta, passa por definir a posição e orientação do referencial da ferramenta em relação ao referencial Tool0. Após a execução de um procedimento de calibração, o ponto central da ferramenta passa a ser o ponto cuja localização em relação à estrutura de base é resolvida pelas equações cinemáticas do controlador.

1.4 Estrutura da dissertação

Neste primeiro capítulo, encontra-se a contextualização do tema relativo à calibração de ferramentas em manipuladores robóticos, com referência ao interesse e motivação que levou ao seu desenvolvimento. É feita a descrição dos objetivos do trabalho, que passa por desenvolver um sistema de calibração, e apresentada informação prévia necessária à compreensão do tema e da relevância deste tipo de dispositivos.

No Capítulo 2 são apresentadas algumas soluções de sistemas de calibração para robôs industriais disponíveis no mercado. São dispositivos que recorrem a tecnologias avançadas e representam o estado da arte.

O Capítulo 3 é dedicado à síntese das ferramentas matemáticas necessária à modelação dos procedimentos que suportam os sistemas de calibração. São abordadas soluções patenteadas e que constituem a base teórica de todo o trabalho desenvolvido.

No Capítulo 4 é feita uma comparação da performance de um procedimento quando usados modelos matemáticos distintos de processamento dos mesmos dados. O teste consiste em calcular o erro na determinação da posição do referencial da ferramenta, quando introduzida uma perturbação nos pontos recolhidos para calibração.

Segue-se o Capítulo 5 onde se apresenta a análise e implementação de diversos procedimentos de calibração da posição do referencial da ferramenta, o que inclui o desenvolvimento de *software*, não só para simulação, mas também para verificação experimental dos procedimentos. É feita uma comparação das diversas metodologias com a avaliação quantitativa e qualitativa dos procedimentos e referência às dificuldades encontradas. Por último é apresentada uma proposta para o sistema de calibração.

Finalmente no Capítulo 6 são apresentadas as conclusões e feitas recomendações para desenvolvimento de trabalhos futuros.

Capítulo 2 - Dispositivos de calibração comerciais para robôs industriais

Dada a enorme importância da exatidão de um manipulador robótico, existem no mercado diversas soluções para avaliação e calibração de robôs industriais. Apresentam-se seguidamente não só o estado da arte dos sistemas de calibração de ferramenta, mas também alguns equipamentos para calibração de robôs e componentes de células robóticas.

2.1 Sistemas de teste e monitorização

Os procedimentos e dispositivos de calibração do robô e de referenciais de objetos não são alvo de estudo deste trabalho, contudo os métodos usados recorrem a tecnologias também aplicáveis à calibração da ferramenta, pelo que se referem seguidamente alguns destes equipamentos. Destacam-se dois grupos, os sistemas que envolvem contacto físico com uma referência e os que não estabelecem contacto físico.

2.1.1 Sistemas com contacto

Ballbar

O sistema Ballbar QC20-W da Renishaw [4] é um *kit* de teste à precisão no posicionamento de máquinas ferramenta e manipuladores robóticos. É composto por um sensor telescópico linear com esferas de precisão nos topos, e dois suportes magnéticos de assento das mesmas. Um dos suportes é montado na mesa de trabalho da máquina, na posição fixa, e o outro na ferramenta. Isto permite fazer o teste exemplificado na Figura 3.

A máquina é programada a descrever uma trajetória circular em torno do suporte fixo à mesa, e o sensor identifica a variação do raio da trajetória. Os dados recolhidos são enviados via *bluetooth* para um computador, o que permite fazer a análise numérica dos mesmos. Os dados podem ainda ser visualizados de forma gráfica.

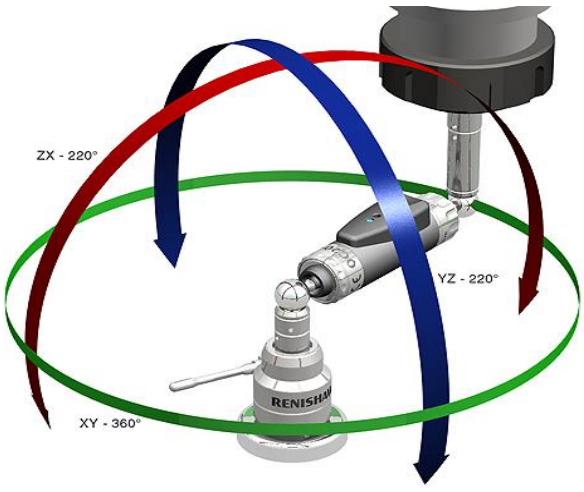


Figura 3 – RENISHAW Ballbar [4]

Navigator

Tipicamente, o processo de calibração de referenciais de objetos passa por usar o robô como sistema de medição, para fazer localização de pontos específicos sobre os objetos. Essa localização é feita por um operador qualificado com *jogging* manual do manipulador. O Navigator [5] é um sistema desenvolvido pela ABB que permite a execução automática da localização desses pontos. O robô está equipado com uma ferramenta de prova e a célula robótica é preparada com furos para montagem de três pinos com pequenas esferas. Os centros destas esferas serão os pontos específicos a determinar.

A calibração é realizada, permitindo que o robô localize as esferas dos pinos. Assim, a etapa manual de apontar posições de referência para a calibração, passa a ter métodos de pesquisa automatizados. Este sistema só funciona se a posição das esferas for de alguma forma conhecida com uma incerteza de alguns milímetros. O robô obedece a instruções de busca linear em direção à referência carregada para o centro da esfera, tal como exemplificado na Figura 4. O contacto mecânico entre a prova e as esferas é determinado com o estabelecimento de contacto elétrico. Este sinal faz com que o robô pare e memorize uma posição, que será usada para determinar a localização do centro da esfera. Desta forma, a posição exata da esfera é atualizada para o sistema real. O reconhecimento da localização das três esferas é usado para gerar a posição e orientação do referencial do objeto.

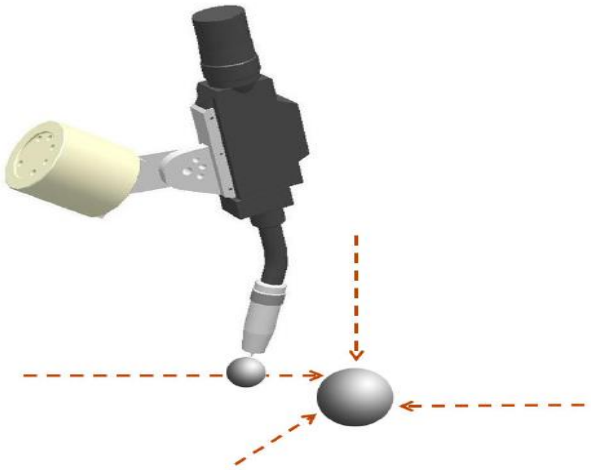


Figura 4 – ABB Navigator [5]

2.1.2 Sistemas sem contacto

Laser Tracker

Os sistemas LEICA Absolute Laser Tracker [6] e FARO Laser Tracker [7] representados respetivamente na Figura 5 e Figura 6, são soluções equivalentes de fabricantes diferentes, que usam uma tecnologia baseada em laser. O *laser tracker* é um dispositivo composto por uma cabeça rotativa ativa com um emissor e recetor de feixe laser. Na ferramenta é acoplado um alvo refletor. Assim que a cabeça faz o reconhecimento do alvo, o seu controlador mantém contacto visual da mesma com o refletor, independentemente dos movimentos que este tome. Ao fixar o refletor, o dispositivo permite identificar a posição do mesmo no espaço de trabalho. Admitindo que o refletor se encontra fixo à ferramenta, este sistema é capaz de monitorizar constantemente a posição do TCP e compará-la com a posição teórica carregada no controlador. Em caso de desvio, é possível desencadear um processo de compensarão do mesmo corrigindo a posição do TCP.

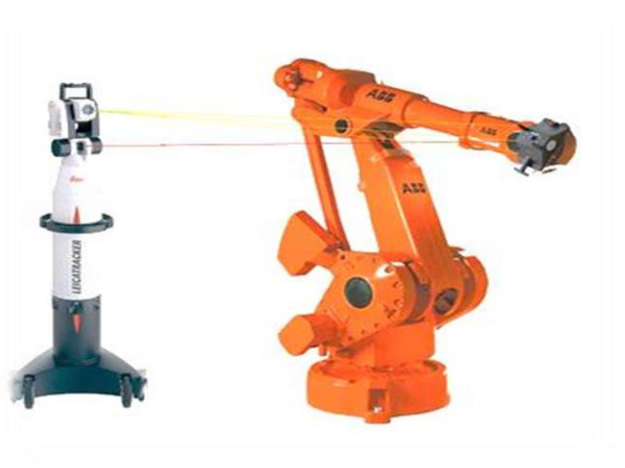


Figura 5 – LEICA Absolute Laser Tracker [6]

O facto de a monitorização ser constante permite compensar não só desvios devido a colisão ou desgaste da ferramenta, mas também a deformação estrutural do manipulador causada pela carga transportada, os efeitos de contração e dilatação térmica e as folgas nos elementos de transmissão.



Figura 6 – FARO Laser Tracker [7]

GripLAB

O GripLAB da Wiest AG [8] é um sistema desenvolvido para detetar e compensar desvios das peças sobre preensão de garras, reduzindo o risco de colisão. Permite medir diretamente uma peça fixa na garra através de um sistema de medição compacto baseado em tecnologia laser. Como visível na Figura 7, o GripLAB é fixo na estação de trabalho do robô e a medição é executada periodicamente. Se o teste verificar que a peça está fora de um determinado limite de tolerância, o sistema desencadeia uma ação de ajuste aos dados relativos ao referencial da ferramenta. Ao fazer a medição diretamente sobre a peça, o dispositivo acaba também por compensar o desgaste que possa ocorrer na garra que suporta a peça.

O sistema é controlado por um programa instalado no controlador do próprio robô, que troca informação com a estação de medição através de uma comunicação RS232, *Ethernet* ou *Fieldbus*. A Wiest AG disponibiliza as versões do programa de controlo do sistema nas linguagens nativas consoante o fabricante do robô.

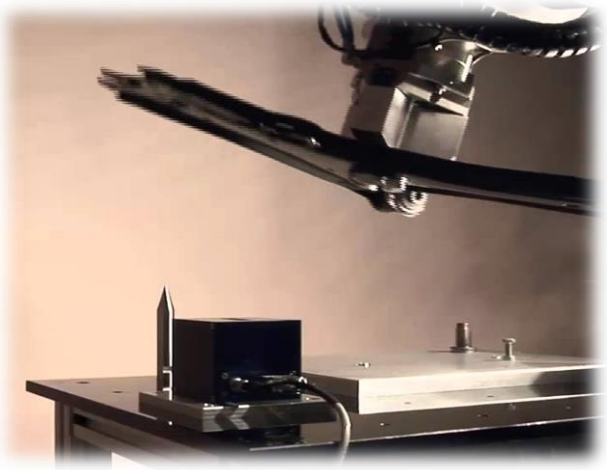


Figura 7 – WIEST AG GripLAB [8]

2.2 Sistemas dedicados à calibração da ferramenta

Para a determinação da posição e orientação de uma ferramenta em relação à flange do robô vários sistemas podem ser usados. Destacam-se nesta secção alguns dos sistemas e dispositivos de calibração de ferramenta, também agrupados consoante estabelecem ou não contacto físico com uma referência.

2.2.1 Sistemas sem contacto

LaserLAB

O LaserLAB [8] é uma solução integrada de calibração automática dos equipamentos de uma célula robótica. Permite a calibração de manipuladores robóticos de fabricantes diversos, de referenciais de objetos, de ferramentas, e até de eixos adicionais de posicionadores que a célula robótica integre. É por isso uma tecnologia de calibração muito poderosa. Na sua configuração mais simples, apresentada na Figura 8, é utilizada uma interface de medição diretamente acoplada à base do robô e uma esfera padrão fixa ao elemento terminal.

A interface de medição é constituída por cinco sensores laser posicionados numa base pentagonal e alinhados de forma a que os feixes se intersectem num único ponto. Ao posicionar a esfera no dispositivo laser são medidas distâncias de cada um dos sensores à superfície da esfera. Isto permite determinar coordenadas tridimensionais de pontos da superfície esférica com os quais a posição do centro da esfera pode ser calculada.

Antes da primeira utilização, este sistema requer um procedimento de configuração. A calibração é depois feita de forma automática por inicialização do programa de calibração. O LaserLAB não opera diretamente sobre o controlador do robô. Os dados das medições efetuadas são processados por um computador com o *software* de medição LoopIn, que assume o controlo do robô e move os eixos para posições memorizadas efetuando várias medições tridimensionais.

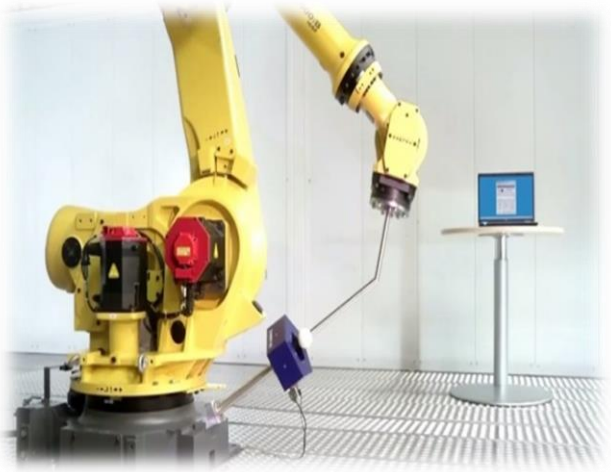


Figura 8 – WIEST AG LaserLAB [8]

Fork Sensor

O ABB BullsEyes [20] é um dispositivo de calibração capaz de determinar a posição do TCP e a orientação da ferramenta. Utiliza um sensor de forquilha para localizar as arestas da ferramenta no sistema de coordenadas do robô. O sensor de forquilha é uma peça metálica em forma de "U", dotada de um sensor de barreira ótico. Num dos braços encontra-se um emissor de feixe laser, no outro um recetor. Se este feixe for interrompido porque uma ferramenta se colocou entre o emissor e o recetor, o sensor comuta o estado digital da sua saída.

O método utiliza o robô como dispositivo de medição. Isto significa que a precisão é dependente da repetibilidade do robô. O modelo comercial é composto pela unidade de calibração de feixe único (sensor de forquilha), que deve ser instalada na estação de trabalho tal como indicado na Figura 9. Esta unidade possui um sistema de controlo próprio e independente do controlador do robô.

Durante a execução do procedimento de calibração, o robô efetua uma série de movimentos com a ferramenta cortando o feixe do sensor. Com a passagem da ferramenta pelo feixe, é possível determinar o diâmetro nominal da mesma. Se esta passagem for repetida em diferentes secções ao longo do comprimento da ferramenta, o eixo da ferramenta pode também ser determinado o que permite atribuir-lhe uma orientação. Conhecenda a orientação do eixo, a ferramenta pode ser deslocada perpendicularmente ao feixe o que possibilita a determinação da sua extremidade.



Figura 9 – ABB Bulls Eyes [20]

O ABB BullsEyes foi um sistema especificamente desenvolvido para a calibração de tochas de soldadura e só pode ser usado em ferramentas simétricas em torno do seu eixo. Contudo existem no mercado soluções, que também usam um sensor do tipo forquilha, mas que permitem que a posição e a orientação do ponto central da ferramenta sejam adquiridas para ferramentas não simétricas em torno do seu eixo.

O sistema Advintec da LEONI [9], ilustrado na Figura 10, é um desses casos. Em vez de usar apenas um único feixe, utiliza duas barreiras de luz que se cruzam entre si, o que também torna todo o procedimento de calibração mais rápido. O sistema AutoCal, representado na Figura 11, é comercializado pela DYNALOG [10] e segue o mesmo princípio de funcionamento.



Figura 10 – LEONI Advintec [9]

Importa referir que estes sistemas envolvem procedimentos automáticos e destinam-se principalmente à recuperação do TCP e da orientação da ferramenta. Isto significa que o robô se desloca periodicamente ao sensor para verificação, seguida de correção em caso de desvio. Para que o robô se mova para o interior da unidade antes da primeira utilização, é necessário carregar uma estimativa prévia do referencial da ferramenta no controlador do robô.



Figura 11 – DYNALOG AutoCal [10]

RotoLAB

O RotoLAB da Wiest AG [8] é um dispositivo que segue o princípio de funcionamento anteriormente apresentado. Contudo, nesta solução o feixe move-se e o robô mantém a ferramenta imóvel no interior do dispositivo tal como ilustrado na Figura 12. Possui uma unidade de medição de coordenadas bidimensional, que executa um varrimento de 360 graus à ponta da ferramenta. É também um dispositivo criado para recuperar a posição e orientação de tochas de soldadura mas pode ser aplicado a outro tipo de ferramentas.

O robô move a ferramenta periodicamente para o RotoLAB que determina se o TCP saiu ou não da zona de tolerância. Estas inspeções periódicas asseguram que a ferramenta está sempre na posição correta, tornando o processo mais eficiente e prevenindo desvios de trajetória devido a colisões que alterem a posição da ferramenta. Se a inspeção determinar que a ferramenta está fora da zona de tolerância, inicia-se o procedimento de recuperação com medição completa. As novas coordenadas são depois transferidas automaticamente para o controlador do robô. Em caso de mudança de ferramenta, o novo TCP pode também ser calculado.

A interface de medição deve ser colocada na célula robótica em localização alcançável pelo robô. Esta unidade é controlada por um programa que corre no controlador do robô e estabelece uma comunicação série com o mesmo. A Wiest AG disponibiliza o *software* de controlo compatível com controladores de diferentes fabricantes de robôs industriais.



Figura 12 – WIEST AG RotoLAB [8]

2.2.2 Sistema com contacto

FC Calibration

Force Control Calibration [5] é um sistema de calibração de ferramenta desenvolvido pela ABB para os seus robôs dotados de sistema de controlo de força. A ABB disponibiliza esta opção de controlo em força para robôs dedicados a operações de rebarbagem e polimento. Nos robôs com controlo de força, tipicamente é usado um sensor de força montado no elemento terminal do robô, sendo disponibilizados dois modos distintos de controlo de força. No modo *speed change,* o controlador pode alterar a velocidade dos percursos programados em função da força medida. No modo de *force control*, o controlador pode alterar a posição do elemento terminal de forma a garantir uma determinada força de contacto.

Este sistema usa o sensor de força para detetar o contacto com o bloco de calibração representado na Figura 13, que é especificamente desenvolvido para esta aplicação, e segue um princípio de funcionamento idêntico ao do Navigator. Na prática, o sistema é capaz de calibrar a posição e orientação de uma ferramenta de forma automática, por deteção do contacto entre a ferramenta e o bloco de calibração e usando a informação do sensor de força. A posição é medida pelo próprio robô. Adicionalmente, o sistema pode também definir um referencial de um objeto pela definição de três pontos estacionários.

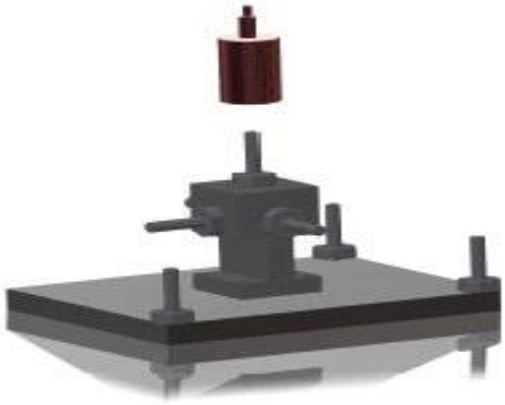


Figura 13 – ABB Force Control Calibration [5]

2.3 Síntese dos sistemas existentes

Para melhor visualizar a oferta e potencialidades dos sistemas de calibração utilizados em robôs industriais, procurou-se sintetizar e agrupar as suas principais características na Tabela 1.

Tabela 1 - Síntese dos sistemas existentes

Sistema	Fabricante	Tecnologia	Com contacto	Calibração de:			Teste	Aplicação		Usa robô para medição
				Ferramenta	Objeto	Robô		Contínua	Periódica	
Ballbar	Renishaw	LVDT	X				X		X	
Navigator	ABB	Contacto elétrico	X		X				X	X
Absolute Tracker	LEICA	Laser Tracker		X	X	X	X	X		
FARO Laser Tracker	FARO	Laser Tracker		X	X	X	X	X		
GripLAB	WIEST AG	Laser scanner		X					X	
LaserLAB	WIEST AG	Laser scanner		X	X	X	X		X	
Bulls Eyes	ABB	1 feixe laser		X					X	X
Advintec	LEONI	2 feixes laser		X					X	X
RotoLAB	WIEST AG	1 feixe laser rotativo		X					X	X
FC Calibration	ABB	Controlo de força	X	X	X				X	X

Verifica-se que a maioria dos sistemas de calibração comerciais não estabelece contacto físico e são baseados em tecnologias laser. Dos sistemas analisados, o LaserLAB é o que oferece maiores possibilidades de calibração. Os sistemas baseados em *Laser Tracker* são os únicos que fazem a monitorização contínua. Apenas os sistemas que usam sensor de forquilha, o Navigator e o FC Calibration usam o robô como instrumento de medição.

Capítulo 3 – Ferramentas matemáticas de apoio à calibração do TCP

Para a análise dos procedimentos envolvidos na calibração da ferramenta de um robô, importa referir os métodos matemáticos utilizados para a representação da posição e orientação dos referenciais associados quer ao elemento terminal do robô quer à ferramenta.

3.1 Cinemática direta e algoritmo de Denavit-Hartenberg

A cinemática direta de um manipulador permite determinar a posição e orientação do elemento terminal no espaço de trabalho a partir dos valores de posição dos elos, tendo em consideração a estrutura cinemática do manipulador.

O algoritmo de Denavit-Hartenberg [21], é um método que permite a obtenção da cinemática direta para manipuladores de estrutura série de uma forma expedita. Apenas é necessário atribuir um referencial, respeitando determinadas regras, a cada um dos *n* elos do manipulador e especificar quatro parâmetros geométricos (a_i, d_i, α_i e θ_i), onde *i* representa cada um dos elos. Os parâmetros a_i e α_i são constantes, d_i é variável apenas se a junta for prismática e θ_i é variável se a junta for rotativa. Se o manipulador for constituído apenas por juntas rotativas, então apenas os parâmetros θ_i são variáveis. Com estes parâmetros é possível obter uma representação matricial que permite relacionar a posição e orientação de dois referenciais consecutivos. Trata-se de uma matriz de transformação homogénea e tem a forma apresentada na Equação 3.1.

$$ {}^{i-1}\mathbf{A}_i = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^{i-1}\mathbf{R}_i & {}^{i-1}\mathbf{p}_i \\ 0 & 1 \end{bmatrix} \tag{3.1} $$

Onde,

${}^{i-1}\mathbf{R}_i$ representa a orientação do referencial do elo *i* relativamente ao do elo *i*-1.

${}^{i-1}\mathbf{p}_i$ representa a posição da origem do referencial do elo *i* relativamente ao do elo *i*-1.

O vetor de zeros da quarta linha da matriz representa o efeito de perspetiva.

O elemento (4,4) corresponde ao fator de escala.

Os valores indicados correspondem aos normalmente usados na modelação cinemática de robôs.

As matrizes de transformação homogéneas podem ser multiplicadas, permitindo estabelecer uma relação entre diferentes referenciais. Por exemplo, a matriz que relaciona o referencial do elo 5 com o do elo 2 é dada pela matriz ${}^{2}\mathbf{T}_{5}$ representada na Equação 3.2.

$$ {}^{2}\mathbf{T}_{5} = {}^{2}\mathbf{A}_{3}\,{}^{3}\mathbf{A}_{4}\,{}^{4}\mathbf{A}_{5} \tag{3.2} $$

De igual modo, a matriz que relaciona o referencial do elemento terminal com o referencial de base, para um manipulador com *n* elos, é a matriz ${}^{0}\mathbf{T}_{n}$ dada na Equação 3.3.

$$ {}^{0}\mathbf{T}_{n} = {}^{0}\mathbf{A}_{1}\,{}^{1}\mathbf{A}_{2}\ldots{}^{n-1}\mathbf{A}_{n} = \begin{bmatrix} {}^{0}\mathbf{R}_{n} & {}^{0}\mathbf{p}_{n} \\ 0 & 1 \end{bmatrix} \tag{3.3} $$

Onde,

${}^{0}\mathbf{R}_{n}$ representa a matriz de rotação que fornece a orientação do referencial do elemento terminal relativamente ao de base do manipulador.

${}^{0}\mathbf{p}_{n}$ representa o vetor de posição que fornece a posição da origem do referencial do elemento terminal relativamente ao de base do manipulador.

Esta matriz define a posição e orientação do elemento terminal do robô em relação ao referencial de base, que é fixo, em função da posição angular θ_i de cada junta e da geometria do manipulador (a_i, α_i, d_i). Isto significa que a matriz ${}^{0}\mathbf{T}_{n}$ muda consoante a configuração do robô. A mesma matriz pode ainda ser representada numa forma alternativa como,

$$ {}^{0}\mathbf{T}_{n} = \begin{bmatrix} {}^{0}\mathbf{R}_{n} & {}^{0}\mathbf{p}_{n} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{n} & \mathbf{s} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4} $$

A matriz ${}^{0}\mathbf{T}_{n}$ de dimensão [4x4] engloba uma matriz de rotação [3x3] contendo os versores de cada um dos eixos do referencial do elemento terminal em relação ao referencial de base (**n**, **s**, **a**), e um vetor **p** [3x1], contendo a posição da origem do referencial do elemento terminal em relação ao de base.

A matriz ${}^{0}\mathbf{T}_{n}$ pode ser usada como matriz de transformação de um outro referencial, cuja posição da origem e orientação estão definidas em relação ao referencial do elemento terminal, para o referencial da base.

Admita-se que é acoplada uma ferramenta ao elemento terminal. Será certamente mais conveniente estabelecer a posição do TCP da ferramenta em relação ao referencial do elemento terminal do robô do que em relação ao referencial de base.

Note-se que a ferramenta vê fixo o referencial do elemento terminal e móvel o referencial da base. Isto traduz-se em que as coordenadas da posição do TCP resultem apenas do conhecimento das dimensões da ferramenta. Exprimir o TCP no referencial de base envolve usar a matriz ${}^{0}\mathbf{T}_{n}$ como matriz de transformação.

$$ {}^{0}\mathbf{T}_{T} = {}^{0}\mathbf{T}_{n}\,{}^{n}\mathbf{T}_{T} \tag{3.5} $$

Onde,

${}^{0}\mathbf{T}_{T}$ representa a posição e orientação da ferramenta em relação ao referencial da base.

${}^{0}\mathbf{T}_{n}$ é a matriz de transformação que contém a posição e orientação do referencial do elemento terminal em relação ao de base.

${}^{n}\mathbf{T}_{T}$ representa a posição e orientação da ferramenta em relação ao referencial do elemento terminal.

Obtida a transformação, o controlador do robô utiliza a matriz ${}^{0}\mathbf{T}_{T}$, ou seja, é o TCP e não a origem do referencial do elemento terminal que passa a ser considerado para posicionar o robô numa determinada localização (*target*).

Por outro lado, um *target* é muito mais fácil de referenciar relativamente à base, que vê fixa, do que em relação ao elemento terminal, que vê móvel. Pode então estabelecer-se a relação expressa na Equação 3.6.

$$ {}^{0}\mathbf{T}_{L} = {}^{0}\mathbf{T}_{n}\,{}^{n}\mathbf{T}_{T} \tag{3.6} $$

Onde,

${}^{0}\mathbf{T}_{L}$ representa o *target*, ou seja, a posição e orientação pretendida para a ferramenta atingir em relação ao referencial de base do robô.

3.2 Calibração com geometrias de referência

Existem diversos métodos para calibração de manipuladores robóticos e suas ferramentas. De entre esses distintos métodos, Zhongxue e Qing [18] referem soluções gerais de calibração de ferramenta que usam um *target* de calibração com uma restrição geométrica. Segundo os autores essa restrição pode ser um ponto, uma linha, um plano ou uma superfície esférica.

Para os procedimentos de calibração do TCP, optou-se por usar um ponto e um plano como elemento de referência. Esta escolha deve-se ao facto de estas referências geométricas serem as usadas em algumas soluções patenteadas que também foram analisadas. Todos estes métodos são aplicáveis aos requisitos estabelecidos para este trabalho, nomeadamente no que diz respeito a efetuarem a calibração sem recorrer a um dispositivo de medição externo.

3.2.1 Calibração com um ponto de referência

Segundo a patente US 5194792 [16]

Na Secção 3.1 viu-se que é possível estabelecer a matriz de transformação ${}^{0}\mathbf{T}_{n}$ para qualquer configuração do robô, pois esta é função dos valores das juntas do manipulador. Atendendo à Equação (3.6), considerando apenas posição e sabendo que a matriz ${}^{0}\mathbf{T}_{n}$ pode ser usada como matriz de transformação, podem estabelecer-se as seguintes relações,

$$ {}^{0}\mathbf{p}_{p} = {}^{0}\mathbf{T}_{n} \cdot {}^{n}\mathbf{p}_{TCP} \tag{3.7} $$

$$ {}^{n}\mathbf{p}_{TCP} = {}^{0}\mathbf{T}_{n}^{-1} \cdot {}^{0}\mathbf{p}_{p} \tag{3.8} $$

Essencialmente a Equação (3.8) mostra que se forem conhecidas as coordenadas de um ponto alcançável pelo robô no referencial de base, ${}^{0}\mathbf{p}_{p}$, é possível determinar as coordenadas da ferramenta no referencial do elemento terminal (Tool0), ${}^{n}\mathbf{p}_{TCP}$, através da inversa da matriz de transformação correspondente à configuração do robô quando é estabelecido contacto entre a extremidade da ferramenta e o ponto conhecido. Note-se que no momento do contacto, o ponto fixo e a extremidade da ferramenta são coincidentes.

Esta patente é baseada exatamente neste pressuposto teórico, ou seja, propõe a calibração do TCP levando a ferramenta a um ponto de calibração, que é um ponto do espaço de trabalho cujas coordenadas relativamente ao referencial de base do robô são conhecidas.

Durante a fase de configuração, uma ferramenta é montada na flange do manipulador e é estabelecido um ponto fixo do espaço de trabalho para ser o ponto de calibração. A calibração da ferramenta usada durante a fase de configuração é feita através de um outro método, ou seja, admite-se conhecida a sua posição e orientação no referencial do elemento terminal. O reconhecimento da posição do ponto fixo, que permite torná-lo o ponto de calibração, é feito levando a ferramenta calibrada a um ponto fixo do espaço de trabalho e aplicando a Equação (3.7). Memorizando a posição do ponto de calibração, desde que a sua posição não seja alterada, é possível calibrar qualquer ferramenta montada na flange do elemento terminal do robô por aplicação da Equação (3.8) e levando a ferramenta a calibrar a contactar com o ponto de calibração. Contudo, se o ponto de calibração por algum motivo se desviar, deixa de poder ser usado como ponto calibrador. A patente sugere que após o procedimento de definição da posição do ponto de calibração se faça a calibração de um ponto do próprio manipulador tal como representado na Figura 14 no número 36. Assim, passa a ser possível usar esse ponto do manipulador para localizar o ponto de calibração caso este mude de posição no espaço de trabalho, ou definir um qualquer outro ponto do espaço de trabalho como ponto de calibração.

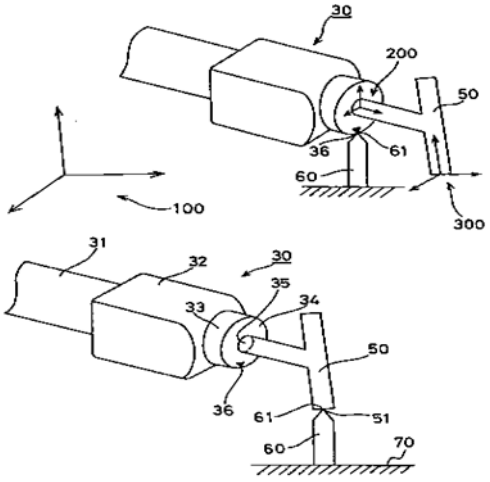


Figura 14 – Calibração com um ponto conhecido [15]

Por minimização do erro

O procedimento representado esquematicamente na Figura 15 passa por montar na flange do robô uma ferramenta (t) e fixar um ponto (p), algures no espaço de trabalho do robô. Levar a extremidade da ferramenta (t) ou o ponto que se pretende para TCP, à vizinhança do ponto (p) consecutivamente e segundo orientações diferentes fazendo o *jogging* do robô. Gravar a posição e orientação do robô, matriz ${}^{0}\mathbf{T}_{n_i}$, em cada uma das orientações atingidas para aproximar a extremidade da ferramenta (t) ao ponto (p), sendo tipicamente usadas quatro orientações distintas.

A determinação do TCP é feita calculando a posição da extremidade da ferramenta em relação referencial Tool0 com a formulação de um problema de otimização segundo o método dos mínimos quadrados.

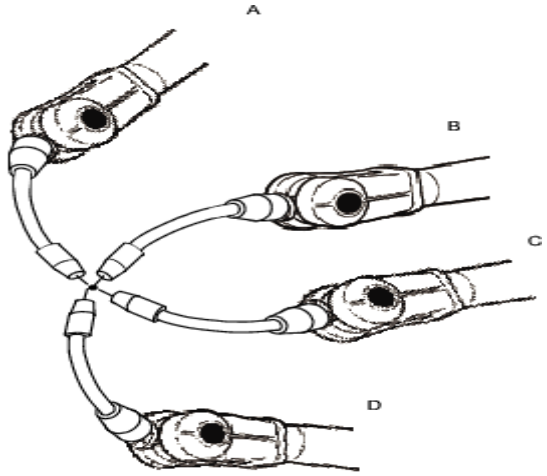


Figura 15 – Calibração com um ponto desconhecido [14]

Atendendo a que a matriz ${}^{0}\mathbf{T}_{n}$ está expressa em coordenadas homogéneas, é possível reescrever a Equação (3.7) dividindo a matriz ${}^{0}\mathbf{T}_{n}$ nas suas submatrizes como,

$$ {}^{0}\mathbf{p}_{p} = {}^{0}\mathbf{R}_{n} \cdot {}^{n}\mathbf{p}_{t} + {}^{0}\mathbf{p}_{n} \tag{3.9} $$

Atendendo à sua dimensão, os vetores e matrizes têm que ser manipulados na forma que se segue,

$$ \begin{Bmatrix} {}^{0}\mathbf{p}_{p} \\ 1 \end{Bmatrix} = \begin{bmatrix} {}^{0}\mathbf{R}_{n} & {}^{0}\mathbf{p}_{n} \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} {}^{n}\mathbf{p}_{t} \\ 1 \end{Bmatrix} = \begin{Bmatrix} {}^{0}\mathbf{R}_{n} \cdot {}^{n}\mathbf{p}_{t} \\ 0 \end{Bmatrix} + \begin{Bmatrix} {}^{0}\mathbf{p}_{n} \\ 1 \end{Bmatrix} \tag{3.10} $$

Onde,

${}^{0}\mathbf{p}_{p}$ e ${}^{n}\mathbf{p}_{t}$ não são conhecidos e representam a posição do ponto fixo a atingir (p) no referencial de base do robô e a posição da ferramenta (t) no referencial da flange do elemento terminal.

${}^{0}\mathbf{T}_{n}$ é conhecida porque se dispõe da cinemática do manipulador e os valores das juntas do manipulador são obtidos a partir dos sensores das juntas. Esta matriz representa, como já foi visto, a posição e orientação do referencial da flange do elemento terminal em relação ao referencial de base do robô.

A aplicação do método dos mínimos quadrados para a obtenção da posição do TCP da ferramenta é feita do modo seguinte. É definido o vetor de variação da posição do ponto fixo (p) obtido a partir da matriz ${}^{0}\mathbf{T}_{n_i}$ e expresso pela Equação (3.11) como,

$$ \Delta {}^{0}\mathbf{p}_{p_i} = ({}^{0}\mathbf{R}_{n_i} - {}^{0}\mathbf{R}_{n_0}) \cdot {}^{n}\mathbf{p}_{t} + ({}^{0}\mathbf{p}_{n_i} - {}^{0}\mathbf{p}_{n_0}) \tag{3.11} $$

$$ \Delta {}^{0}\mathbf{p}_{p_i} = \Delta {}^{0}\mathbf{R}_{n_i} \cdot {}^{n}\mathbf{p}_{t} + \Delta {}^{0}\mathbf{p}_{n_i} $$

$$ \text{Para } i = 1....m $$

Na Equação (3.12) define-se uma função custo como sendo a soma dos quadrados das diferenças entre os vetores de posição do ponto fixo em relação ao referencial de base para as diferentes orientações usadas.

$$C({}^{n}\mathbf{p}_{t}) = \sum (\Delta^{0}\mathbf{p}_{p_i})^{2} = \sum (\Delta^{0}\mathbf{R}_{n_i} \cdot {}^{n}\mathbf{p}_{t} + \Delta^{0}\mathbf{p}_{n_i})^{2} \tag{3.12}$$

Ao derivar a função C e igualar a zero, é possível determinar o valor de ${}^{n}\mathbf{p}_{t}$ que minimiza a função, ou seja, o valor que minimiza os desvios e que representa a melhor aproximação à localização da extremidade da ferramenta.

$$\partial C({}^{n}\mathbf{p}_{t}) = 2\sum \left(\Delta^{0}\mathbf{R}_{n_i}{}^{\mathrm{T}} \Delta^{0}\mathbf{R}_{n_i} \cdot {}^{n}\mathbf{p}_{t_{opt}} + \Delta^{0}\mathbf{R}_{n_i}{}^{\mathrm{T}} \Delta^{0}\mathbf{p}_{n_i} \right) = 0 \tag{3.13}$$

$${}^{n}\mathbf{p}_{t_{opt}} = -\left[\sum \left(\Delta^{0}\mathbf{R}_{n_i}{}^{\mathrm{T}} \Delta^{0}\mathbf{R}_{n_i} \right) \right]^{-1} \sum \left(\Delta^{0}\mathbf{R}_{n_i}{}^{\mathrm{T}} \Delta^{0}\mathbf{p}_{n_i} \right) \tag{3.14}$$

No procedimento anteriormente apresentado admitiu-se que durante a fase de configuração se conhecida a posição e orientação de uma ferramenta no referencial do elemento terminal pois esta teria sido calibrada por outro método. Este é um possível método de calibração dessa ferramenta pois não necessita do conhecimento prévio das coordenadas de um ponto de calibração no referencial de base.

Adicionalmente, estando o ponto fixo e a extremidade da ferramenta em contacto, ao substituir ${}^{n}\mathbf{p}_{t_{opt}}$ na Equação (3.7) vem,

$${}^{0}\mathbf{p}_{p_{opt}} = {}^{0}\mathbf{T}_{n} \cdot {}^{n}\mathbf{p}_{t_{opt}} \tag{3.15}$$

Ou seja, ${}^{0}\mathbf{p}_{p_{opt}}$ fica bem definido e, desde que a posição do ponto fixo (p) não mude de lugar no espaço de trabalho do robô, é sempre possível calibrar qualquer ferramenta levando a extremidade da mesma ao encontro de ${}^{0}\mathbf{p}_{p_{opt}}$ e aplicando a Equação (3.8) que resulta na seguinte relação.

$${}^{n}\mathbf{p}_{TCP} = {}^{0}\mathbf{T}_{n}{}^{-1} \cdot {}^{0}\mathbf{p}_{p_{opt}} \tag{3.16}$$

Através do cálculo de uma superfície esférica

O procedimento anteriormente visto, que utiliza um ponto fixo do espaço de trabalho e posiciona o robô e respetiva ferramenta com diferentes orientações nesse ponto, pode utilizar um método de cálculo do TCP distinto do anteriormente apresentado. Este outro método envolve a determinação da superfície esférica que melhor se ajusta aos pontos de calibração adquiridos. De acordo com o procedimento, qualquer que seja a orientação, se a ponta da ferramenta estiver em contacto com o mesmo ponto fixo, então o ponto adquirido para a posição de Tool0 encontra-se numa superfície esférica cujo centro é o ponto fixo e o raio é a distância do TCP à origem do referencial do elemento terminal, Tool0. Isto significa que é possível aplicar um algoritmo matemático para gerar a superfície esférica que passa pelos pontos adquiridos. Se os pontos estiverem definidos relativamente ao referencial de base, então, as coordenadas do centro da esfera são a posição do ponto usado para calibração, que é também o TCP no referencial de base do robô. Segundo a teoria apresentada na Secção 3.1 e usando a matriz de transformação relativa a uma das orientações é possível estabelecer as coordenadas do TCP em relação ao referencial Tool0.

Três pontos não definem uma esfera. Se forem não colineares irão definir um círculo pelo qual se podem fazer passar uma infinidade de esferas. Quatro pontos definem uma única esfera se e só se forem não complanares. Assim sendo, esta metodologia requer um número mínimo de quatro pontos para que possa ser aplicada. Na prática podem ser recolhidos mais do que quatro pontos e o algoritmo passa a determinar a superfície que melhor se ajusta aos pontos dados [21].

A equação de uma superfície esférica é dada por,

$$(x-a)^2+(y-b)^2+(z-c)^2=r^2 \tag{3.17}$$

Onde (a,b,c) são as coordenadas do centro e r é o raio.

Que é equivalente a,

$$x^2-2ax+a^2+y^2-2by+b^2+z^2-2cz+c^2-r^2=0 \tag{3.18}$$

Ou,

$$(x^2+y^2+z^2)-2ax-2by-2cz+a^2+b^2+c^2-r^2=0 \tag{3.19}$$

Seja,

$$\begin{aligned} c_1&=1\\ c_2&=-2a\\ c_3&=-2b\\ c_4&=-2c\\ c_5&=a^2+b^2+c^2-r^2 \end{aligned} \tag{3.20}$$

Então a mesma equação pode ser escrita como,

$$(x^2+y^2+z^2)\cdot c_1+x\cdot c_2+y\cdot c_3+z\cdot c_4+c_5=0 \tag{3.21}$$

$$x\cdot c_2+y\cdot c_3+z\cdot c_4+c_5=-(x^2+y^2+z^2)\cdot c_1 \tag{3.22}$$

$$x\cdot\frac{c_2}{c_1}+y\cdot\frac{c_3}{c_1}+z\cdot\frac{c_4}{c_1}+\frac{c_5}{c_1}=-(x^2+y^2+z^2) \tag{3.23}$$

Assim sendo, a equação da esfera que passa em *n* pontos não complanares, é dada pelo sistema de equações que se segue,

$$\begin{cases} x_i \cdot \frac{c_2}{c_1} + y_i \cdot \frac{c_3}{c_1} + z_i \cdot \frac{c_4}{c_1} + \frac{c_5}{c_1} = -(x_i^{\,2} + y_i^{\,2} + z_i^{\,2}) \\ x_{i+1} \cdot \frac{c_2}{c_1} + y_{i+1} \cdot \frac{c_3}{c_1} + z_{i+1} \cdot \frac{c_4}{c_1} + \frac{c_5}{c_1} = -(x_{i+1}^{\,2} + y_{i+1}^{\,2} + z_{i+1}^{\,2}) \\ \\ \cdots \\ \\ x_{n-1} \cdot \frac{c_2}{c_1} + y_{n-1} \cdot \frac{c_3}{c_1} + z_{n-1} \cdot \frac{c_4}{c_1} + \frac{c_5}{c_1} = -(x_{n-1}^{\,2} + y_{n-1}^{\,2} + z_{n-1}^{\,2}) \\ x_n \cdot \frac{c_2}{c_1} + y_n \cdot \frac{c_3}{c_1} + z_n \cdot \frac{c_4}{c_1} + \frac{c_5}{c_1} = -(x_n^{\,2} + y_n^{\,2} + z_n^{\,2}) \end{cases} \tag{3.24}$$

A título de exemplo, para os pontos,

$P_1 = (0, 3, 2)$, $P_2 = (1, -1, 1)$, $P_3 = (2, 1, 0)$ e $P_4 = (5, 1, 3)$ vem,

$$\begin{cases} 3 \cdot \frac{c_3}{c_1} + 2 \cdot \frac{c_4}{c_1} + \frac{c_5}{c_1} = -13 \\ \\ \frac{c_2}{c_1} - \frac{c_3}{c_1} + \frac{c_4}{c_1} + \frac{c_5}{c_1} = -3 \\ \\ 2 \cdot \frac{c_2}{c_1} + \frac{c_3}{c_1} + \frac{c_5}{c_1} = -5 \\ \\ 5 \cdot \frac{c_2}{c_1} + \frac{c_3}{c_1} + 3 \cdot \frac{c_4}{c_1} + \frac{c_5}{c_1} = -35 \end{cases} \tag{3.25}$$

Este sistema linear pode ser associado a uma matriz envolvendo os coeficientes numéricos e a parte literal e assim ser resolvido pelo método de eliminação de Gauss. Se $\mathbf{A}$ for a matriz dos coeficientes, de dimensão [*n*x*n*], e $\mathbf{b}$ um vetor com *n* componentes correspondentes à parte literal, $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ é a solução do sistema $\mathbf{Ax} = \mathbf{b}$. Esta operação pode ser feita em Matlab como $\mathbf{x} = [\mathbf{A} \setminus \mathbf{b}]$.

$$[\mathbf{A} \setminus \mathbf{b}] = \left|\begin{array}{cccc|c} 0 & 3 & 2 & 1 & -13 \\ 1 & -1 & 1 & 1 & -3 \\ 2 & 1 & 0 & 1 & -5 \\ 5 & 1 & 3 & 1 & -35 \end{array}\right| \tag{3.26}$$

A solução para este caso vem,

$$\begin{cases} \frac{c_2}{c_1} = -4 \\ \frac{c_3}{c_1} = -2 \\ \frac{c_4}{c_1} = -6 \\ \frac{c_5}{c_1} = 5 \end{cases} \tag{3.27}$$

As coordenadas do centro são então dadas por,

$$\begin{cases} a = \frac{c_2}{-2} = 2 \\ b = \frac{c_3}{-2} = 1 \\ c = \frac{c_4}{-2} = 3 \end{cases} \tag{3.28}$$

O raio é dado por,

$$r = \sqrt{a^2 + b^2 + c^2 - c_5} = 3 \tag{3.29}$$

3.2.2 Calibração com um plano de referência

Adquirindo a orientação do plano

A utilização de um ponto como restrição geométrica permite calibrar o TCP formulando um problema de otimização, em que as três incógnitas relativas à posição do TCP são simultaneamente determinadas com recurso ao método dos mínimos quadrados ou resolvendo um sistema de equações para determinar uma superfície esférica em que as coordenadas do centro são o TCP. Uma forma alternativa de abordar o problema é separando a obtenção das incógnitas a serem resolvidas. Cada uma passa a ser determinada numa medição independente.

Admita-se que se possui um plano para calibração, como representado na Figura 16, cuja espessura (e) é conhecida. Este plano é colocado na célula robótica, num local alcançável pelo robô. A posição e orientação do plano não são conhecidas. O procedimento passa por fazer o reconhecimento da orientação do plano, o que permite definir um vetor normal ao plano. Esse vetor é usado para gerar um referencial (R) através do qual são feitas medições sobre o plano. Segue-se, um por um, a orientação dos eixos do referencial Tool0 segundo o vetor normal ao plano. Para cada eixo alinhado ortogonalmente com o plano são executadas duas medições em direções opostas sobre o plano, por contacto da ferramenta com o mesmo, num procedimento idêntico ao representado na Figura 16. Em cada medição é guardada a posição (P*i*) da origem do referencial Tool0 em relação ao referencial (R).

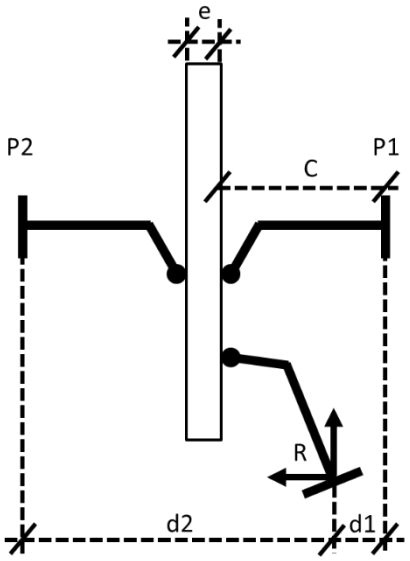


Figura 16 – Calibração com um plano de referência cuja orientação é conhecida

Em acordo com a Figura 16, para as medições associadas ao eixo que é colocado ortogonalmente ao plano, é possível estabelecer a relação,

$$C = \frac{\left|\left(d_1 - d_2\right)\right| - e}{2} \tag{3.30}$$

Note-se que para fazer reconhecimento da orientação do plano se podem usar as funções tipicamente incorporadas nos controladores dos robôs, especificamente desenvolvidas para a determinação de referenciais de objetos. Com estas funções, a orientação é determinada adquirindo pontos sobre o plano, fazendo contactar a ponta da ferramenta com o mesmo. A determinação da orientação do plano é independente do TCP desde que se mantenha a orientação da ferramenta durante o procedimento. Na prática, como o TCP ainda não é conhecido, os pontos adquiridos são as posições da origem do elemento terminal do robô (Tool0) quando a ferramenta contacta com o plano, ou seja, não são pontos do plano mas pontos de um plano virtual e paralelo.

Adquirindo a posição e orientação do plano

No procedimento anteriormente apresentado é apenas determinada a orientação do plano. Isto leva a que seja necessário contactar com as duas faces do plano para cada uma das três cotas a determinar. Em alternativa, é possível fazer-se a localização da posição do plano na primeira vez em que a ferramenta é levada às duas faces. Conhecendo a posição e orientação do plano, não é mais necessário fazer duas medições para cada uma das cotas a determinar, bastando contactar com uma das faces.

Atente-se na Figura 17. Três "toques" inicialmente executados determinam a orientação do plano. É gerado o referencial (R) com essa orientação. Um dos eixos de Tool0 é orientado segundo a normal ao plano em duas direções e são adquiridas as posições (P*i*). O ponto médio entre estas duas posições é depois usado para posicionar um referencial (R1), que contém a orientação de (R), mas está localizado no centro do plano, tal como indicado na Figura 17. É sobre este referencial que são depois feitas as medições.

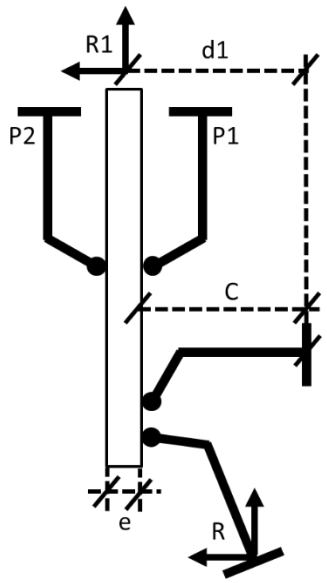


Figura 17 – Calibração com um plano de referência (posição e orientação são conhecidas)

Em acordo com a figura, a relação geométrica que permite determinar a cota C com a medição associada ao eixo que é colocado ortogonalmente ao plano é,

$$C = d_1 - \frac{e}{2} \tag{3.31}$$

Segundo a patente US 5910719 [15]

Considere-se que o TCP de uma dada ferramenta está corretamente definido. Após um determinado período de operação, é possível que a ferramenta se afaste da sua posição de calibração, ou seja, durante a execução das operações ocorreu um pequeno desvio do TCP.

Esta patente propõe uma forma de recuperar o TCP após um período de operação. O procedimento passa por posicionar o robô sobre um plano e gravar uma posição, movimentar o robô perpendicularmente e em direção ao plano e gravar uma nova posição quando a ferramenta contacta com o plano, depois, regressar à posição inicial. A diferença entre os dois pontos é a cota $L0$ indicada na Figura 18. Segue-se a rotação do robô em torno do TCP ativo de um ângulo conhecido θ^+ e nova movimentação em direção ao plano. Se o TCP tiver sofrido um desvio, então a diferença entre o ponto inicialmente gravado e o de contacto será a cota L^+. O procedimento repete-se agora com uma rotação θ^- para determinação da cota L^-.

Note-se que se o TCP não tiver sofrido um desvio, $L^+ = L^- = L0$.

Sendo $\mathbf{v}$ o vetor de posição do TCP real em relação ao suposto TCP, em acordo com a Figura 18, é possível estabelecer as seguintes relações geométricas.

$$L0 + v_y = L^+ - v_x \sin\theta + v_y \cos\theta = L^- + v_x \sin\theta + v_y \cos\theta \tag{3.32}$$

O que permite escrever as componentes v_x e v_y do vetor $\mathbf{v}$ em função das cotas medidas e do ângulo usado para a rotação.

$$v_x = \frac{L^+ - L^-}{2\sin\theta} \tag{3.33}$$

$$v_y = \frac{L^+ + L^- - 2L0}{2 - 2\cos\theta} \tag{3.34}$$

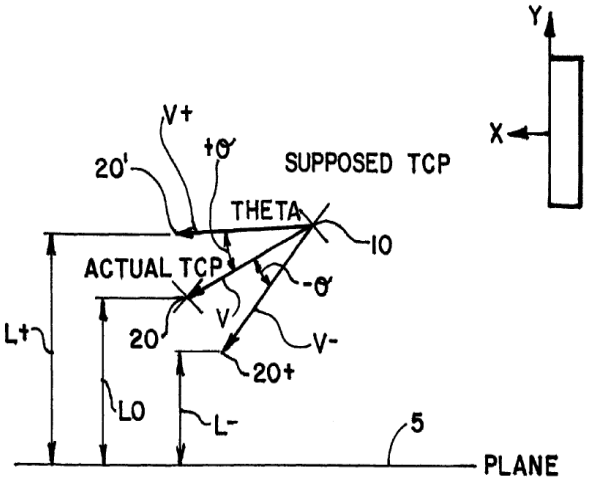


Figura 18 – Patente US 5910719 [15]

Apesar de apresentado na patente como uma forma de apenas revalidar o TCP, este procedimento pode ser estendido à determinação do TCP sem qualquer valor prévio. Para isso basta que o ponto em torno do qual se faz a rotação seja a origem de Tool0.

Capítulo 4 - Comparação de metodologias por perturbação dos pontos de calibração

No Capítulo 3 viu-se que para a calibração usando um ponto como referência geométrica, há duas metodologias matemáticas de processamento dos dados que podem ser utilizadas para o cálculo de TCP, o método dos mínimos quadrados e o cálculo da superfície esférica que melhor se ajusta aos pontos identificados. Contudo, o procedimento de calibração é o mesmo, envolvendo posicionar a ferramenta consecutivamente num ponto fixo com orientações diferentes, gravando a posição e orientação do referencial do elemento terminal, Tool0.

Sendo as metodologias diferentes mas o procedimento o mesmo, importa avaliar:

- se ambas as metodologias conduzem à mesma solução;
- se as duas metodologias conduzem à solução ótima;
- a influência da quantidade de pontos adquiridos;
- a influência da posição relativa dos pontos (mais próximos ou afastados);
- a influência da amplitude da perturbação na posição dos pontos medidos;
- qual a sua sensibilidade à perturbação nos pontos adquiridos para calibração.

Tal avaliação foi feita simulando o procedimento com um modelo virtual de um robô, usando uma aplicação informática de simulação/programação de estações de trabalho para robôs da ABB. Da simulação do procedimento foram recolhidas a posição e orientação de Tool0, pontos que depois foram processados em Matlab. Informação mais detalhada sobre o *software* de simulação, o RobotStudio, encontra-se na Secção 5.2.2.1.

O modelo virtual do robô (ABB IRB2400/16) foi equipado com uma ferramenta conhecida. Isto permite conhecer o TCP da ferramenta e utilizar o referencial da ferramenta para reposicionar o robô em torno do ponto fixo de forma teoricamente correta. Assim, são definidas configurações distintas que constituem os pontos de calibração a utilizar, podendo ser utilizados quatro ou oito pontos.

Para executar o cálculo do TCP em Matlab importa conhecer o vetor de posição de Tool0 expresso num referencial cartesiano e o vetor de orientação expresso em ângulos de Euler. Visto que o controlador do robô opera em quatérnios mas o cálculo em Matlab usa ângulos de Euler, foi criado um pequeno programa no *software* de simulação, que se encontra no Anexo 1, para permitir converter os dados obtidos a partir da simulação e facilitar o seu envio para o Matlab.

O TCP utilizado, os pontos de calibração definidos e o ponto fixo considerado são apresentados no Anexo 2, bem como o programa desenvolvido em Matlab que implementa as duas metodologias matemáticas. Este programa foi desenvolvido de forma a simular a incerteza na aquisição de pontos para o cálculo do TCP, que ocorre quando o robô é posicionado pelo operador num ponto fixo do espaço de trabalho e feita uma inspeção visual do seu posicionamento. Simular a incerteza introduzida pelo operador passa por perturbar os pontos amostrados de forma aleatória. A perturbação é executada *n* vezes, e os resultados para as coordenadas do TCP são apresentados em termos de erro na determinação das corretas coordenadas do TCP em gráficos tridimensionais.

4.1 Análise da influência da amplitude da perturbação

O primeiro teste efetuado avalia a influência na precisão dos resultados consoante a amplitude da perturbação introduzida. Consideraram-se duas incertezas de posicionamento que o operador consegue obter para posicionar a ferramenta no ponto fixo, ± 3mm e ± 0.3 mm.

Na Figura 19 são apresentados os resultados obtidos com a metodologia associada à minimização do erro. Na Figura 20 são apresentados os resultados obtidos com a metodologia da superfície esférica que melhor se ajusta aos pontos amostrados. O resultado para a perturbação maior encontra-se representado graficamente a azul e para a menor, a vermelho. Para o cálculo de cada valor do TCP foram utilizadas quatro orientações de ferramenta. O cálculo do TCP foi executado mil vezes, para cada perturbação considerada.

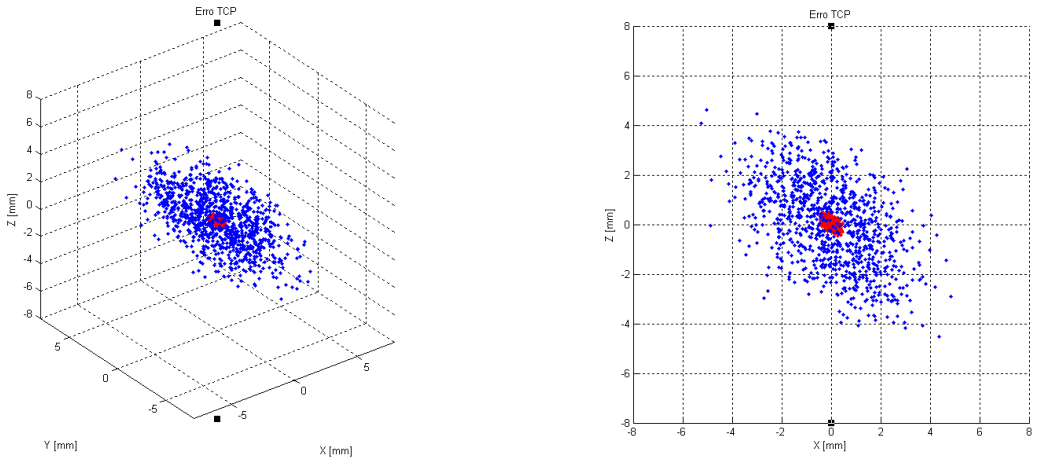


Figura 19 – Influência da amplitude da perturbação no método de minimização do erro (± 3mm a azul, ± 0.3mm a vermelho)

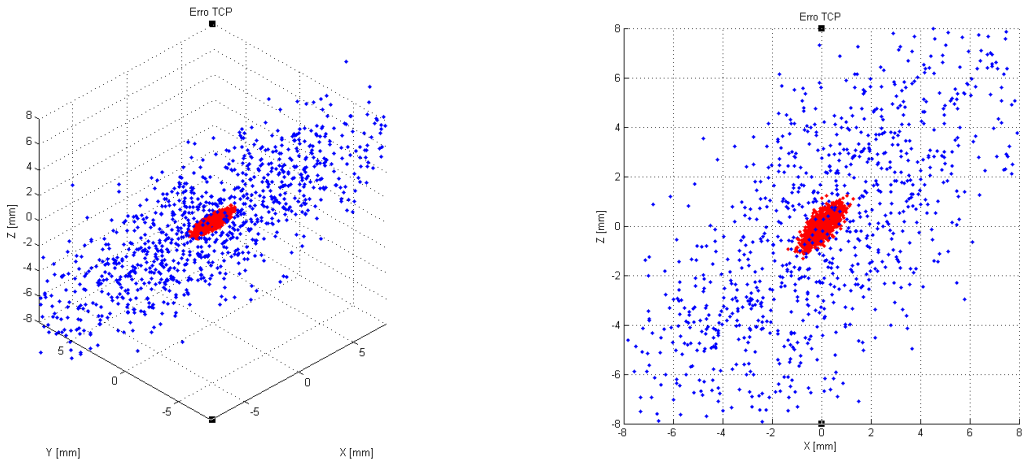


Figura 20 – Influência da amplitude da perturbação no método da superfície esférica (± 3mm a azul, ± 0.3mm a vermelho)

Verifica-se que, associado a uma redução da amplitude da perturbação está uma redução do erro na determinação das coordenadas do TCP. Em ambos os métodos a nuvem de pontos a vermelho, correspondente à perturbação menor, é significativamente mais reduzida do que a azul. Adicionalmente verifica-se que ambas as metodologias convergem para a solução ótima, uma vez que as nuvens de pontos têm centro em erro nulo.

4.2 Análise da sensibilidade das metodologias à perturbação

As duas metodologias foram também analisadas quando sujeitas à mesma perturbação nos pontos adquiridos para calibração, no sentido de perceber qual das duas é mais sensível a essa perturbação. Neste e nos seguintes testes, considerou-se uma amplitude da perturbação de ± 0.3mm. Na Figura 21 apresentam-se os resultados para o método de minimização do erro (a azul) e para o método da superfície esférica (a vermelho). As coordenadas do TCP foram calculadas 10000 vezes.

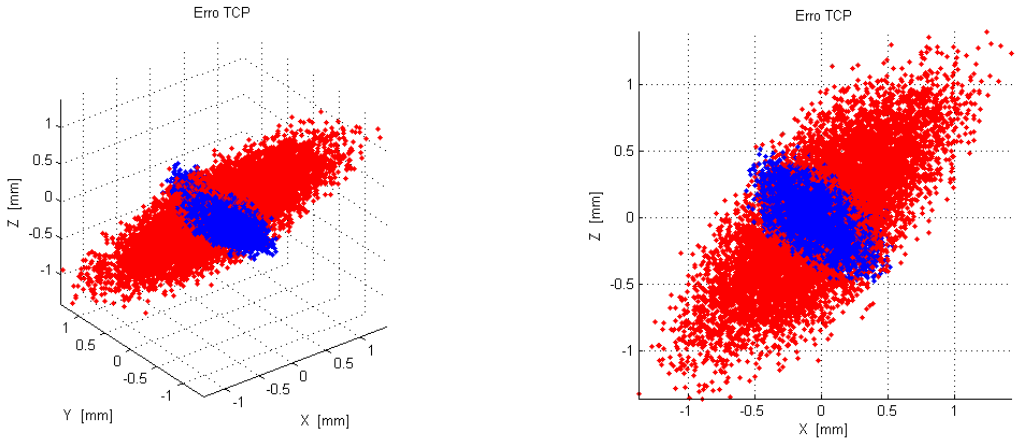


Figura 21 – Sensibilidade das duas metodologias à mesma perturbação
(minimização do erro a azul, superfície esférica a vermelho)

Este teste que as duas nuvens assumem a forma de um paraboloide. A dispersão dos pontos no método da superfície esférica, a vermelho na Figura 21, é maior do que no de minimização do erro, em que os pontos aparecem mais concentrados. Uma vez que a perturbação introduzida foi a mesma, atendendo à dispersão dos pontos, pode dizer-se que a sensibilidade à perturbação no método da superfície esférica é maior do que no de minimização do erro.

4.3 Análise da influência da posição relativa dos pontos adquiridos

Na execução do procedimento é indicado que as posições de Tool0 adquiridas devem ser o mais afastadas possível, isto é, as orientações que se usam para atingir o ponto fixo de referência devem ser bastante distintas. Isto implica um *jogging* acentuado do manipulador em torno do ponto de referência. Contudo, regra geral o utilizador opta por efetuar pequenos movimentos pois este tipo de *jogging* não é prático e pode ser de difícil execução consoante a geometria da ferramenta. Este teste submeteu as duas metodologias a dois grupos de pontos de calibração. Um em que as posições de Tool0 utilizadas são muito próximas e outro em que são bastante afastadas.

Na Figura 22 apresentam-se os resultados para o conjunto de pontos próximos e na Figura 23 para os mais afastados. A azul está representado o método de minimização do erro e a vermelho o método da superfície esférica. Uma vez mais o cálculo foi repetido 10000 vezes.

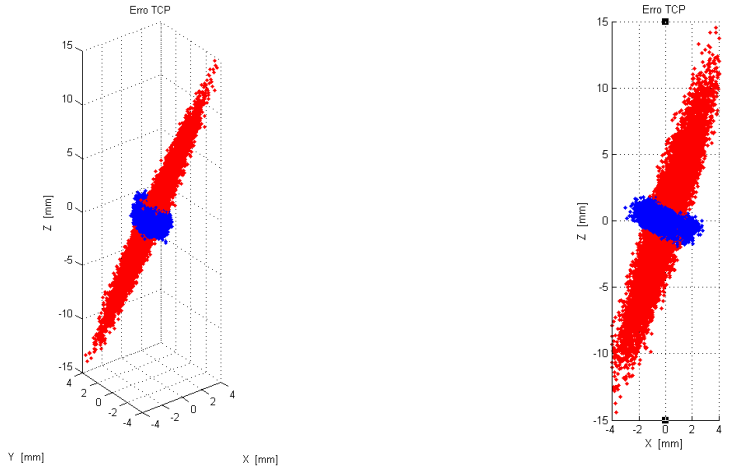


Figura 22 – Erro no cálculo do TCP com pontos próximos

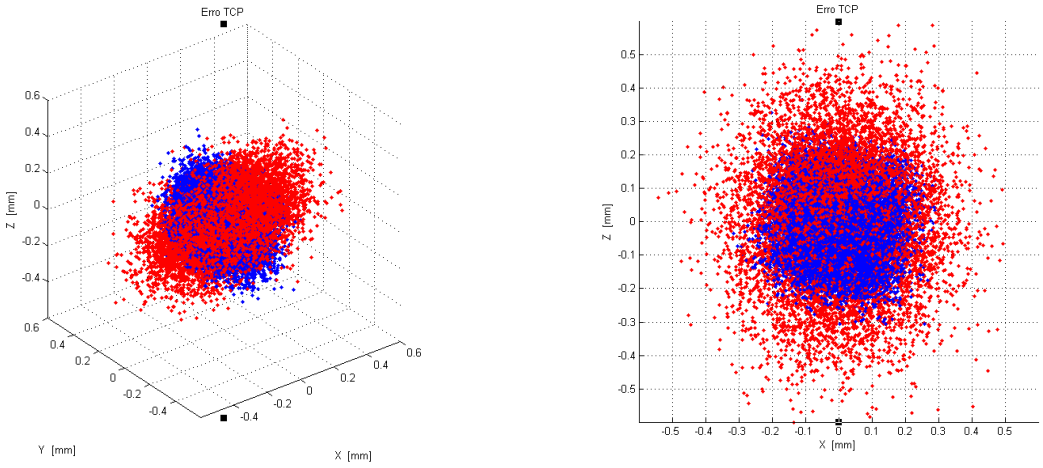


Figura 23 – Erro no cálculo do TCP com pontos afastados

O volume da nuvem de pontos relativa ao método da minimização do erro para pontos próximos, a azul na Figura 22, é idêntico ao da Figura 23, para pontos afastados. Isto indica que o método da minimização do erro não é sensível à posição relativa dos pontos adquiridos. Por outras palavras, é indiferente se o utilizador faz o *jogging* de forma a adquirir pontos afastados ou não. Por outro lado, o método da superfície esférica, a vermelho nas duas figuras, é muito sensível à posição relativa dos pontos de calibração. Na Figura 22, a dispersão dos pontos é significativamente maior do que a da Figura 23, em que o resultado é aproximadamente igual do método de minimização do erro.

4.4 Análise do número de pontos adquiridos

Outra recomendação para a execução do procedimento está relacionada com o número de pontos que são adquiridos para calibração. O método de minimização do erro recorre aos mínimos quadrados para achar a solução que minimiza o erro, o método da superfície esférica procura encontrar a esfera que melhor se ajusta aos pontos adquiridos. Assim é de esperar que quanto maior for o número de pontos adquiridos, melhores sejam os resultados. Neste teste as duas metodologias são executadas usando quatro e oito pontos de calibração. A Figura 24 apresenta os resultados da influência do número de pontos no cálculo do erro do TCP utilizando o método de minimização do erro. A Figura 25 apresenta os resultados da influência do número de pontos no cálculo do erro do TCP utilizando o método da superfície esférica.

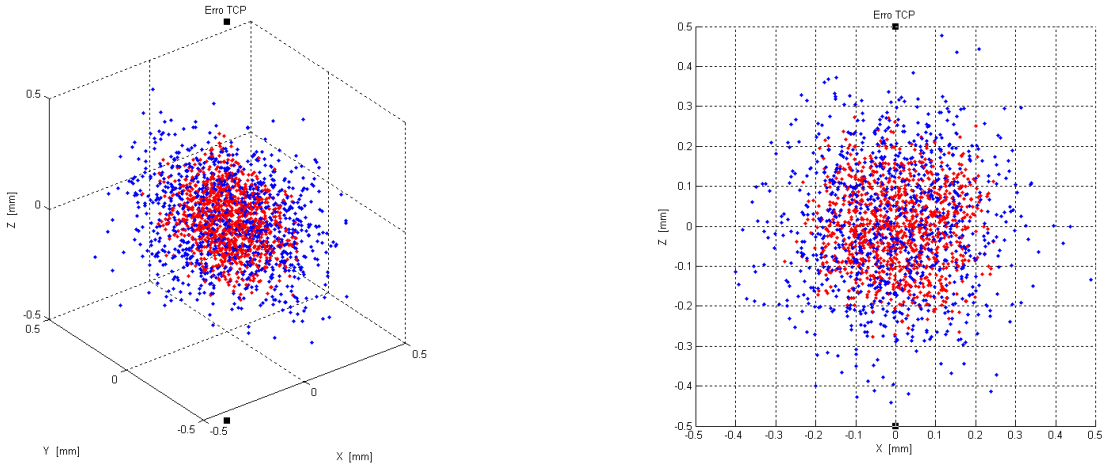


Figura 24 – Influência do número de pontos para o método de minimização do erro
(quatro pontos de calibração a azul, oito pontos de calibração a vermelho)

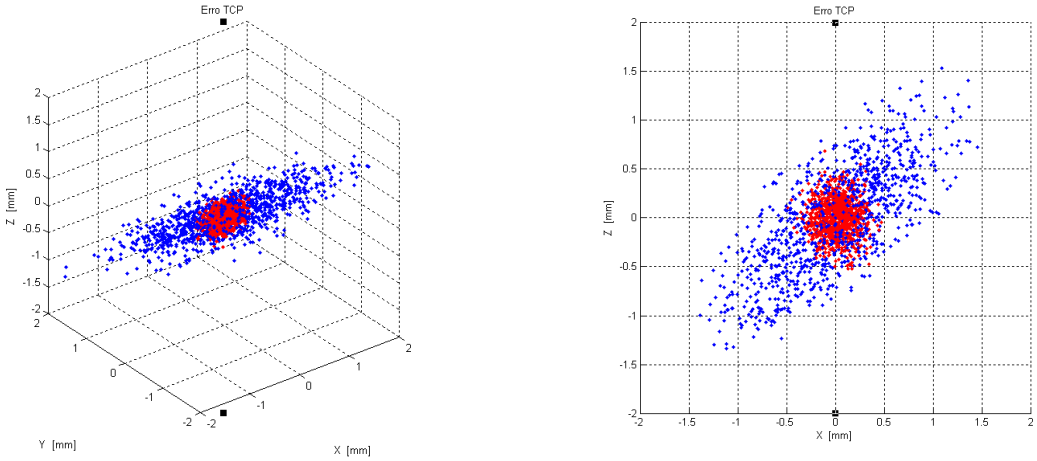


Figura 25 – Influência do número de pontos para o método da superfície esférica

(quatro pontos de calibração a azul, oito pontos de calibração a vermelho)

As nuvens a vermelho correspondem aos resultados obtidos usando o dobro do número de pontos. Como se pode verificar pela Figura 24 e Figura 25, as nuvens a azul são maiores do que as vermelhas pelo que efetivamente se constata que o aumento do número de pontos de calibração influencia positivamente os resultados.

Em suma, o método de minimização do erro é mais robusto que o da superfície esférica. É capaz de absorver mais facilmente perturbações nos pontos adquiridos e é imune ao facto de os pontos usados serem muito próximos. Adicionalmente, os resultados com este método são tanto melhores quanto maior for o número de pontos usados.

Capítulo 5 – Análise e implementação de procedimentos de calibração do TCP

Este capítulo apresenta a implementação de diversos procedimentos de calibração do TCP da ferramenta. Os passos de execução dos procedimentos são programados e simulados. Alguns procedimentos são verificados experimentalmente com o robô da célula robótica. Com base nos resultados das simulações e nos obtidos experimentalmente, é feita uma comparação das diversas metodologias apresentando dificuldades encontradas e avaliando-as quantitativa e qualitativamente. Por último é apresentada uma proposta para o sistema de calibração.

5.1 Análise dos procedimentos disponíveis

São apresentados três procedimentos de calibração de ferramenta que são utilizados na célula robótica disponível. Estes procedimentos são analisados no sentido de identificar as vantagens e os inconvenientes e explorar formas de obter melhores resultados com os mesmos.

5.1.1 Método de introdução direta

A informação da posição e orientação da ferramenta tem que ser carregada no controlador do robô. Uma possibilidade é designada "*direct entry method"* ou método de introdução direta. Neste método, a informação da ferramenta é diretamente escrita no controlador sem passar por um procedimento de calibração havendo três formas de o fazer. A obtenção dos valores a utilizar pode usar as dimensões do ficheiro de CAD da ferramenta. Nos *softwares* de programação *off-line* é necessário definir um TCP virtual. Este valor pode ser transportado para a estação real. Outra hipótese, comum em operações de maquinagem, é através do *software* CAD/CAM do fabricante em que o utilizador tem que preencher um formulário idêntico ao da Figura 26, onde insere as características da ferramenta. O próprio *software* calcula o TCP com essa informação. Se nenhuma das vias anteriores estiver disponível, é ainda possível medir fisicamente as coordenadas do TCP em relação à flange e usar esse valor.

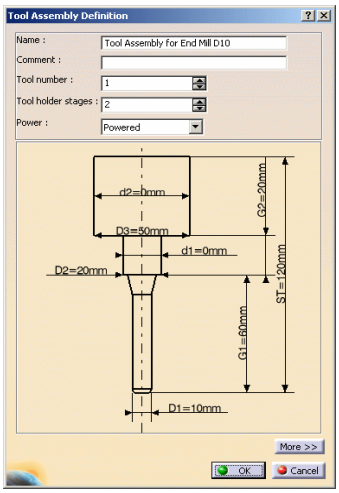


Figura 26 – Interface para definição do TCP pelo método de introdução direta [23]

Este parece ser um método rápido, prático e de fácil execução. Contudo, nem sempre se conhecem as dimensões da ferramenta com rigor. Podem existir incertezas na medição ou até em relação ao fabrico da ferramenta. Além disso, mesmo que se conheçam as dimensões da ferramenta com rigor, e se introduza essa informação diretamente no controlador, podem ocorrer colisões que modifiquem a posição da ferramenta pelo que são por vezes realizadas verificações periódicas, no sentido de identificar alterações que eventualmente possam ser compensadas com um reajuste automático do TCP.

5.1.2 Medição geométrica com o robô

Este é um método eficaz para determinar o TCP de ferramentas que fiquem definidas com apenas uma cota na direção Z de Tool0. A execução do procedimento passa por orientar o robô de forma a que o eixo Z do referencial Tool0 fique perpendicular a um plano horizontal. Segue-se a recolha das coordenadas de posição de Tool0 quando a extremidade da ferramenta contacta com o plano na posição (P1), e quando a flange se encontra ao nível do plano na posição (P2), tal como indicado na Figura 27.

O plano a usar deve ser uma superfície horizontal alcançável pelo robô. Assumindo que essa superfície é paralela a dois dos eixos do referencial de base, é possível orientar a ferramenta perpendicularmente comandando diretamente as juntas do manipulador robótico. A ferramenta é facilmente colocada nas posições de calibração referidas com *jogging* cartesiano do robô. A consola de programação disponibiliza a posição de Tool0 em relação ao referencial de base do robô. Subtraindo as coordenadas de Tool0 na direção Z, que é perpendicular à superfície, é possível obter a cota (C) que representa a posição do TCP no referencial Tool0.

$$C_z = P_{1z} - P_{2z} \tag{5.1}$$

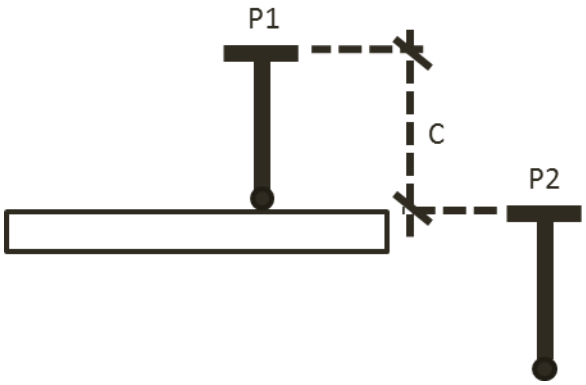


Figura 27 – Medição geométrica com o robô

Este procedimento apresenta duas limitações. Uma prende-se com o facto de ser apenas aplicável a ferramentas cujo TCP pertença ao eixo Z de Tool0. A outra deve-se ao facto de não se garantir a perpendicularidade entre a ferramenta e a superfície horizontal uma vez que, na prática não é feito o reconhecimento da orientação da superfície. Note-se que se a ferramenta contactar a superfície de forma enviesada, não será possível achar com precisão a cota C_z.

5.1.3 Método da ABB

Em alternativa ao método de introdução direta, a ABB e outros fabricantes de robôs industriais disponibilizam um procedimento de calibração incorporado no controlador dos seus robôs, capaz de determinar a posição e orientação da ferramenta. Para determinar a posição, o procedimento consiste em levar o ponto central da ferramenta consecutivamente ao mesmo ponto de referência, segundo pelo menos três orientações diferentes, tal como representado na Figura 28. De cada vez que o robô é reorientado e a ferramenta é posta em contacto com o ponto de referência por *jogging* manual do manipulador, é registada a posição e orientação de Tool0. O TCP é calculado por aplicação das equações apresentadas na Secção 3.2.1 para a calibração de ferramenta usando um ponto de referência.

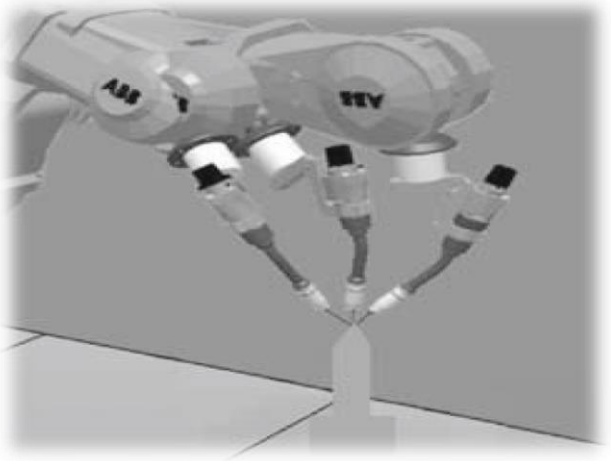


Figura 28 – Método da ABB [18]

Nos robôs da ABB, o procedimento é lançado acedendo à aplicação de calibração de ferramenta representada na Figura 29, através da consola de programação e seguindo os passos que vão sendo indicados. A aplicação permite a execução do procedimento usando três a nove pontos. É indicado que a um maior número de pontos está associada uma melhor calibração, ainda que o tempo despendido pelo utilizador para reorientar, posicionar e inspecionar visualmente o contacto entre a extremidade da ferramenta e o ponto de referência seja maior. O menu da aplicação disponibiliza a possibilidade de editar o TCP de uma ferramenta já existente ou criar e atribuir um TCP a uma nova ferramenta. Pode ainda escolher-se atribuir por defeito a orientação do referencial Tool0 ao referencial da ferramenta ou definir uma orientação específica.

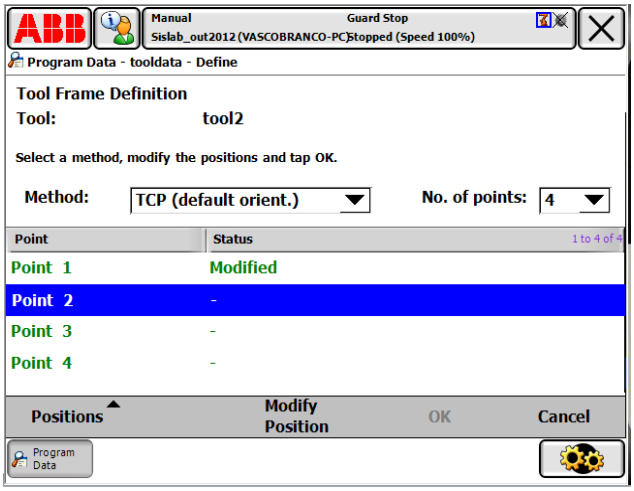


Figura 29 – Interface gráfica para o método da ABB

O principal problema deste procedimento é que os resultados dependem da experiência do operador que executa a calibração, nem sendo sempre fácil reposicionar o robô no mesmo ponto fixo, tendo em atenção a geometria da ferramenta. Note-se que é o operador que fazendo *jogging* do robô, o orienta e leva ao ponto de calibração inspecionando visualmente o posicionamento da extremidade da ferramenta no ponto fixo.

5.1.4 Aumento da performance dos procedimentos disponíveis

Para o método de introdução direta não foi encontrada nenhuma estratégia que permita melhorar os resultados. Será sempre dependente da qualidade dos desenhos ou das medições feitas à ferramenta.

Para a medição geométrica sugere-se fazer um reconhecimento do plano, que depois será usado para fazer a calibração. Os robôs possuem normalmente funções específicas para gerar referenciais com pontos adquiridos. Adquirindo pontos sobre o plano, é possível gerar um referencial contendo a orientação do plano que poderá ser usado para orientar a ferramenta perpendicularmente ao mesmo. Elimina-se assim a incerteza quanto à orientação da ferramenta em relação ao plano.

Para o método da ABB verificou-se que, fixando uma esfera na ferramenta e usando uma superfície que acomodasse a mesma (suporte para assento da esfera), seria possível atingir o mesmo ponto fixo de forma rigorosa, independentemente da orientação, tal como se exemplifica na Figura 30. Desta forma a incerteza introduzida pela experiência do operador calibrador é também minimizada.

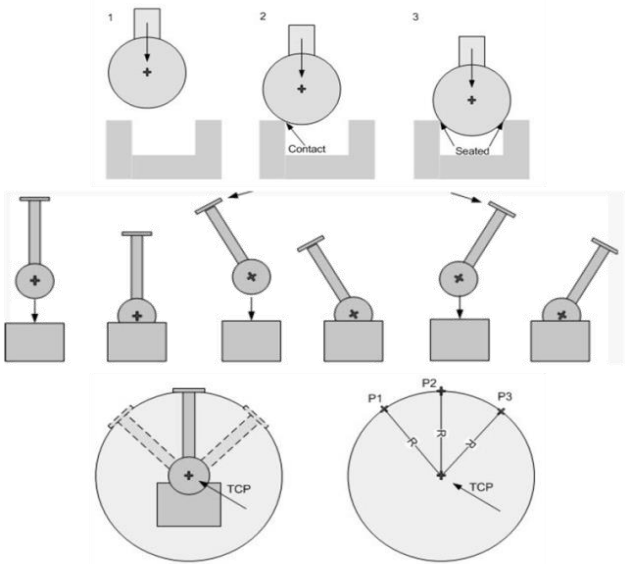


Figura 30 – Esquema de acomodação e reorientação da ferramenta [6]

5.2 Implementação de procedimentos alternativos

No sentido de avaliar outras possibilidades de calibração do TCP, procedeu-se à implementação de procedimentos de calibração de ferramenta baseados nas metodologias apresentadas no Capítulo 3. Nomeadamente, dois procedimentos com recurso a um plano de referência, designados por PLN_1 e PLN_2, e baseados respetivamente na solução patenteada US 5910719 e na proposta para a calibração com um plano de referência adquirindo a posição e orientação do plano, e um procedimento que usa um ponto de referência, designado por PNT_1, baseado na patente US 5194792.

5.2.1 Componentes utilizados

Nesta secção encontra-se uma breve descrição das ferramentas selecionadas para calibração do TCP bem como das superfícies e objetos de referência essenciais à execução dos procedimentos de calibração.

Para validar os procedimentos de calibração foram selecionadas duas ferramentas. Estas ferramentas foram escolhidas, não só por serem as habitualmente usadas na célula disponível mas também porque possuem características muito distintas. A primeira ferramenta selecionada foi a garra PZN100 representada na Figura 31. O TCP deve ser calibrado de forma a que fique no centro dos dedos da garra e à face dos mesmos. A forma com que a ferramenta é fixa à flange, faz com que o TCP pertença ao eixo Z de Tool0, pelo que as suas coordenadas se resumem a uma cota na direção Z. Além disso, a orientação pretendida coincide com a orientação de Tool0 pelo que pode ser automaticamente atribuída sem que se use um procedimento de calibração da orientação.

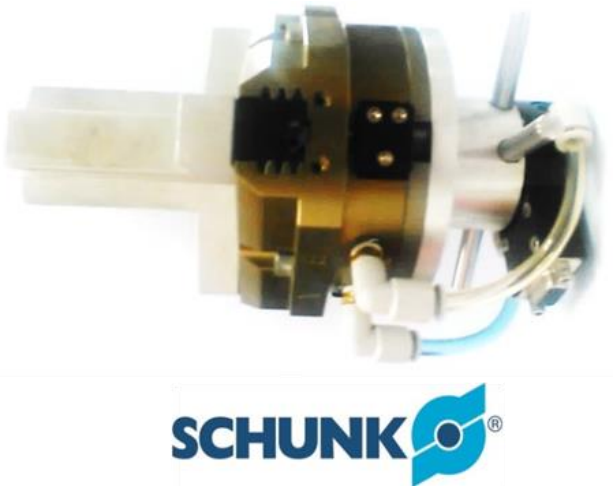


Figura 31 – Garra PZN100

A segunda ferramenta escolhida foi o *spindle* que se encontra representado na Figura 32. Trata-se de um motor-árvore dotado de um sistema automático de mudança de ferramenta, utilizado em operações de maquinagem.

O TCP deve ser calibrado de forma a que fique no centro e extremidade da ferramenta de corte a usar. Devido à geometria do *spindle* o TCP fica definido como três cotas não nulas em relação a Tool0. A orientação não coincide com a do referencial Tool0 pelo que é necessário executar um procedimento de calibração da orientação que faça o eixo Z do referencial da ferramenta coincidente com o eixo de rotação do motor-árvore.

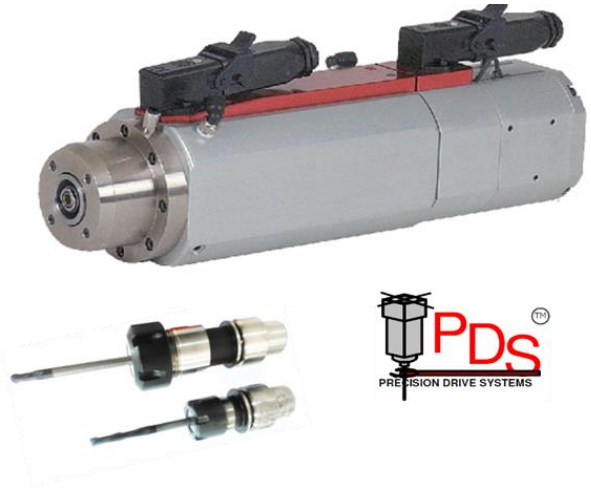


Figura 32 – *Spindle* XLC-070

Para testar experimentalmente os procedimentos, foi necessário criar as superfícies e pontos de referência. Como visto, a medição geométrica com o robô e o procedimento baseado na patente US 5910719 necessitam de um plano horizontal. A mesa posicionadora da célula robótica, tem montada uma superfície de trabalho com uma placa de alumínio horizontal, que é usada nestes dois procedimentos. O método da ABB e o procedimento baseado na patente US 5194792, usam um ponto de calibração. Viu-se que o uso de uma esfera fixa à ferramenta e de um suporte para assento da mesma, facilita o posicionamento no ponto de referência. Para este suporte, cuja função é acomodar a esfera, usou-se um bloco maciço de alumínio com furo passante de diâmetro inferior ao da esfera. A calibração com um plano de referência com aquisição da posição e orientação do plano, usa um plano que tem que ser acessível de ambos os lados. Para isso, fixou-se uma pequena placa de aço num torno sobre a mesa posicionadora. Os objetos enumerados podem ser observados na Figura 33.

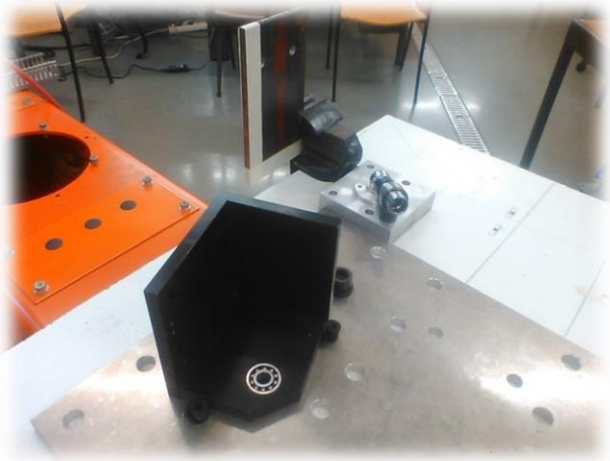


Figura 33 – Superfícies e objetos de referência

Os procedimentos de calibração abordados envolvem o contacto da ferramenta com uma superfície de referência segundo orientações diversas. Pela Figura 34 percebe-se que, quer para a garra quer para o *spindle*, não é possível garantir que o ponto da ferramenta que contacta com a superfície é sempre o mesmo, o que é fundamental para que os procedimentos sejam válidos. Assim sendo, foi necessário criar uma forma de assegurar que o ponto a converter em TCP fique sempre à mesma distância da superfície de calibração. A solução encontrada, foi usar uma esfera fixa na ferramenta durante o procedimento de calibração, para estabelecer o contacto com a superfície de referência, tal como o já sugerido para melhorar os resultados conseguidos com o método da ABB.

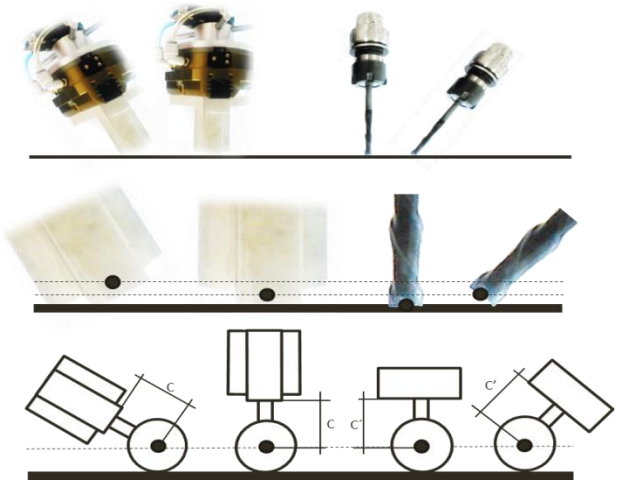


Figura 34 – Contacto da ferramenta com uma superfície

As esferas usadas em calibração de máquinas são fabricadas com tolerâncias muito apertadas sendo comum encontrar esferas com uma variação de raio de 500nm. O rigor exigido a estes equipamentos faz com que o seu preço de venda seja elevado. Para validar experimentalmente os procedimentos foi usada uma esfera de baixo custo. A esfera selecionada foi caracterizada quanto ao seu diâmetro e excentricidade, no sentido de aferir em que medida é que pode influenciar os resultados dos testes.

Inicialmente, com um micrómetro, mediu-se o seu diâmetro e registou-se 28mm. Montou-se seguidamente o exercício da Figura 35 para avaliar a variação do raio da superfície esférica. Fez-se assentar a esfera no suporte já referido para acomodar a posição da esfera no método da ABB e colocou-se um comparador sobre a mesma. Rodou-se a esfera em diferentes direções e mediu-se com o comparador uma variação de raio de $\pm 0.03\,\text{mm}$. Depois, com a esfera montada no suporte de ferramenta do *spindle*, verificou-se que apresenta uma excentricidade de ± 0.4 mm que se deve ao sistema de fixação da esfera no suporte.



Figura 35 – Caracterização da esfera

5.2.2 Simulações virtuais e desenvolvimento de *software*

Os procedimentos implementados foram previamente programados e simulados em ambiente virtual. Foi utilizado o *software* de programação *off-line* de robôs da ABB, o RobotStudio.

5.2.2.1 RobotStudio

O RobotStudio é uma aplicação informática de simulação de estações de trabalho para robôs da ABB [12]. Permite a programação *off-line* de tarefas, e a visualização do comportamento do robô na sua execução. A construção do ambiente virtual é feita criando ou importando equipamentos de uma biblioteca de equipamentos da ABB disponíveis. No *software* estão implementados os modelos virtuais dos controladores dos robôs reais e da consola de programação. Desta forma, o utilizador pode programar o robô virtual tal como se estivesse a programar *on-line* com o robô real.

Os programas desenvolvidos com este *software* podem ser diretamente utilizados no controlador físico do robô, não sendo necessária nenhuma conversão ou pós-processamento. Os ficheiros criados podem ser transferidos do computador para o controlador do robô com ligação *Ethernet* por cabo ou transportando os ficheiros numa *pen* USB.

O RobotStudio e os robôs da ABB usam a linguagem RAPID. Trata-se de uma linguagem de alto nível com instruções dedicadas ao controlo de robôs industriais. No Anexo 3 apresentam-se alguns tipos de variáveis e instruções disponíveis na linguagem RAPID que são importantes para a compreensão do trabalho de programação desenvolvido.

5.2.2.2 Modelo virtual da estação de trabalho

Para poder simular os procedimentos de calibração em ambiente virtual foi necessário recriar no RobotStudio, a estação de trabalho física representada na Figura 36, com as ferramentas e as peças para calibração. O modelo do robô, da mesa posicionadora e dos elementos de segurança foram importados dos equipamentos disponíveis na biblioteca de equipamentos. Os blocos e a esfera de calibração foram criados com o *software* de modelação tridimensional SolidWorks e depois importados para o RobotStudio. O aspeto final da célula virtual pode ser visualizado na Figura 37.



Figura 36 – Aspeto real da estação de trabalho

Foi também necessário selecionar e ajustar o controlador virtual às características do controlador físico, ativando funcionalidades como o *force control* e o *machining tool pack*. Note-se que, apesar de não se poder simular o controlo de força, é necessário instalar o pacote de funções do controlo de força para que sejam reconhecidas na programação. O sistema usado encontra-se no Anexo 4.

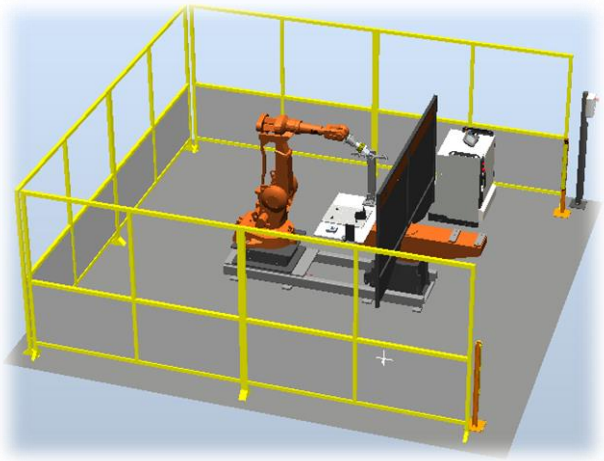


Figura 37 – Modelo virtual da estação de trabalho

Uma vez que nos procedimentos há contacto da ferramenta com as superfícies de referência, utilizou-se a funcionalidade de deteção de proximidade e colisão entre objetos da estação virtual. Na Figura 38 representa-se a aproximação da ferramenta à referência geométrica com *jogging* cartesiano e reorientação onde é clara a importância da ativação desta funcionalidade.

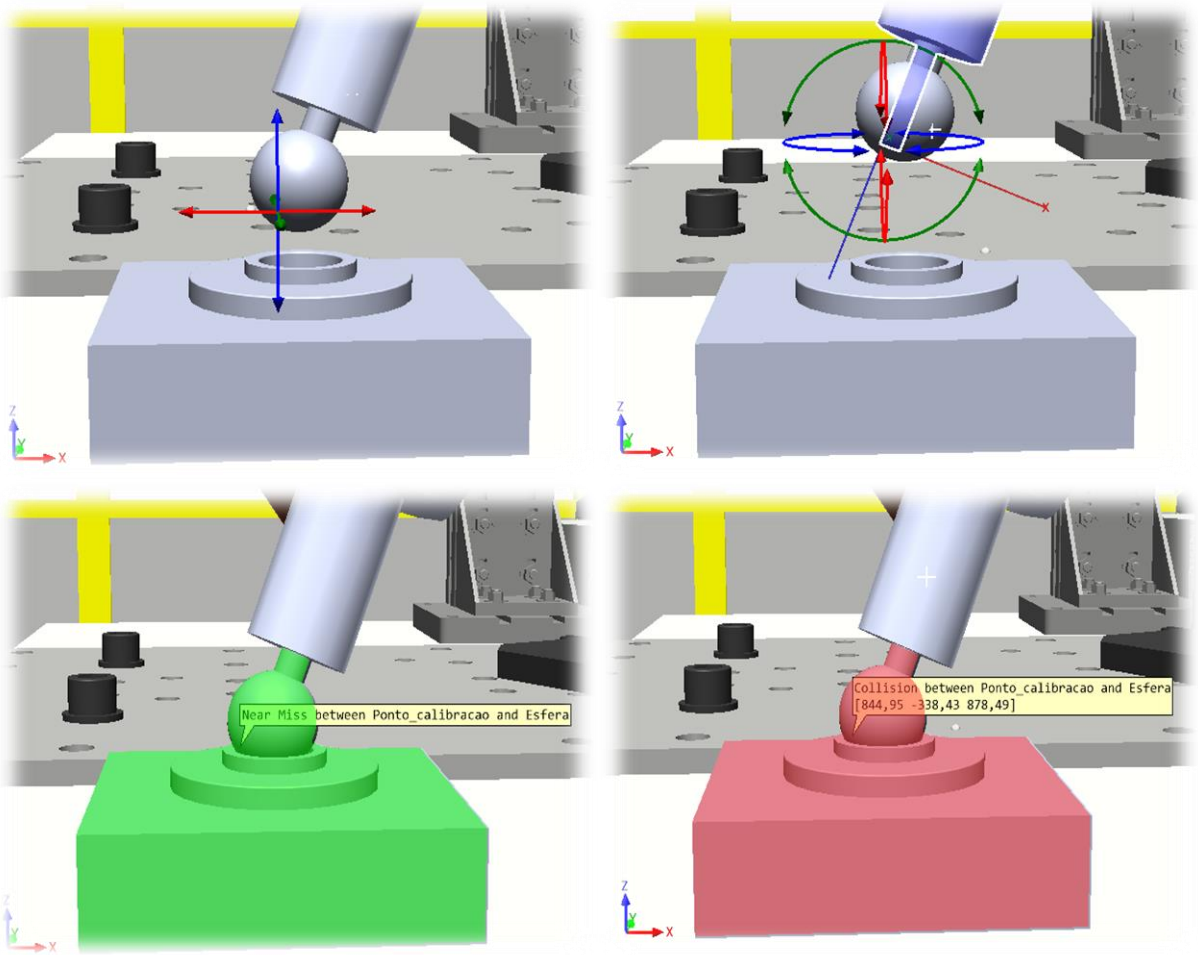


Figura 38 – Deteção de proximidade e colisão entre objetos no espaço virtual

5.2.3 Programação e simulação de procedimentos alternativos

Foi desenvolvido *software* de programação do robô que permite implementar os procedimentos de calibração correspondentes aos métodos alternativos. A simulação permitiu identificar algumas limitações. Os programas desenvolvidos tiram partido da consola de programação do robô, disponibilizando uma interface que facilita a realização dos procedimentos de calibração. No Anexo 5 são apresentados todos os programas desenvolvidos em maior detalhe.

Seguem-se as etapas explicativas dos passos de execução dos procedimentos que foram fundamentais para a programação dos mesmos, com algumas imagens do ambiente virtual recolhidas das simulações efetuadas.

5.2.3.1 Ponto de referência - Procedimento PNT_1

Este procedimento é baseado na patente US 5194792.

Configuração:

1. Montar uma ferramenta no manipulador;
2. Calibrar a ferramenta com o método da ABB (determinação do TCP da ferramenta);
   2.1. Levar a ponta da ferramenta consecutivamente ao mesmo ponto mas com orientações diferentes;
3. Usar essa ferramenta para localizar um ponto do espaço de trabalho (ponto de calibração);
   3.1. Levar a ferramenta ao ponto de calibração;
4. Registar as coordenadas do ponto de calibração no referencial de base do robô;

Calibração:

1. Montar uma ferramenta no manipulador;
2. Levar a ferramenta ao ponto de calibração, ver Figura 39;
3. Lançar o programa PNT_1;
   Seguir as instruções do programa:
   3.1. Introduzir as coordenadas do ponto de calibração;
   3.2. Ao validar, o programa aplica as equações apresentadas na Secção 3.2.1 para o cálculo do TCP com um ponto de referência segundo a patente US 5194792, e apresenta o resultado na consola de programação;

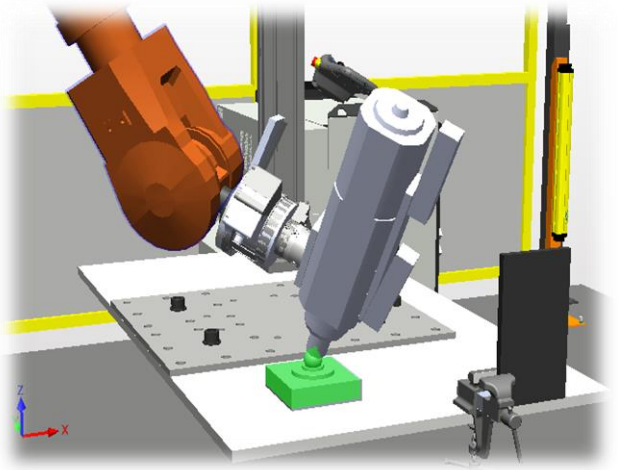


Figura 39 – Contacto da ferramenta com o ponto de referência

5.2.3.2 Plano de referência horizontal – Procedimento PLN_1

Este procedimento é baseado na patente US 5910719 e não necessita de uma etapa de configuração.

Calibração:

1. Montar uma ferramenta no manipulador;
2. Lançar o programa PLN_1;
   Seguir as instruções do programa:
   2.1. Mover o robô para a posição Home;
   2.2. Levar o robô a uma zona sobre a superfície horizontal de referência, a usar para reorientação automática;
   2.3. Levar o robô a contactar a superfície horizontal com a ferramenta como representado na Figura 40;
   2.4. Permitir o regresso à zona de reorientação e a reorientação automática da ferramenta;
   2.5. Levar a ferramenta novamente à superfície de referência e memorizar posição de Tool0 no referencial de base;
   2.6. Repetir os passos 2.4 e 2.5 três vezes;
   2.7. Ao validar, o programa calcula o TCP com aplicação das equações apresentadas na Secção 3.2.2 para a calibração com um plano de referência segundo a patente US 5910719, e apresenta o resultado na consola de programação;

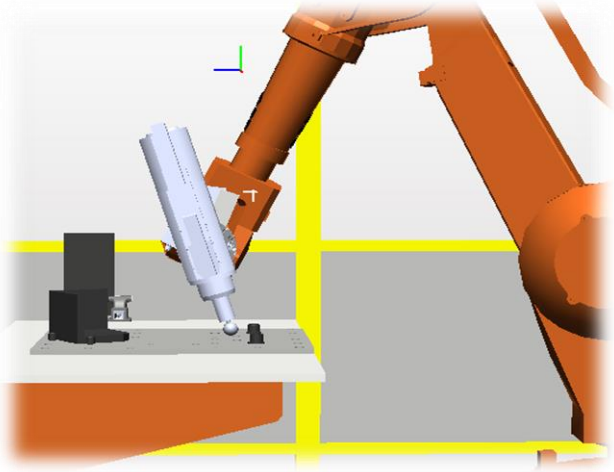


Figura 40 – Contacto da ferramenta com a superfície horizontal de referência

5.2.3.3 Plano de referência vertical – Procedimento PLN_2

Este procedimento é baseado na estratégia proposta para a calibração com um plano de referência adquirindo a posição e orientação do plano.

Configuração:

1. Montar uma ferramenta no manipulador;
2. Lançar o programa PLN_2;
   Seguindo as instruções do programa:
   2.1. Localizar a orientação do plano;
      2.1.1. Levar a ponta da ferramenta com qualquer orientação a três alvos assinalados sobre o plano mantendo a mesma orientação;
      2.1.2. O programa memoriza a posição dos alvos e com essa informação calcula um referencial contendo a orientação do plano;
   2.2. Localizar a posição do plano;
      2.2.1. Levar o robô a uma zona a usar para reorientação automática;
      2.2.2. O programa orienta um dos eixos de Tool0 perpendicularmente ao plano;
      2.2.3. Com essa orientação o utilizador leva a ferramenta a contactar com as duas faces do plano;
      2.2.4. O programa memoriza os pontos de contacto e usa essa informação para localizar o plano.

Calibração:

1. Com o programa em execução levar o robô à zona de reorientação;
2. Permitir a reorientação automática do eixo X de Tool0 perpendicularmente ao plano;
3. Com essa orientação levar a ferramenta a contactar com o plano, ver Figura 41;
4. Repetir os passos 1 a 3 para os eixos Y e Z de Tool0;
5. O programa memoriza os pontos de contacto e usa essa informação para calcular o TCP de acordo com as equações apresentadas na Secção 3.2.2 para a calibração com um plano de referência adquirindo a posição e orientação do plano. O resultado é apresentado na consola de programação;

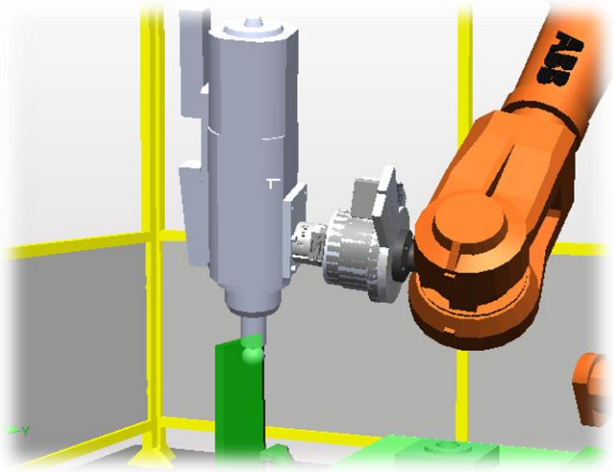


Figura 41 – Contacto da ferramenta com o plano vertical de referência

5.2.4 Problemas encontrados

O *software* do controlador físico do robô é a versão 5.12. Há funcionalidades em versões mais recentes que não são reconhecidas pela versão 5.12. Assim sendo, apesar do *software* usado no RobotStudio ser a versão 6.0, é importante usar a versão 5.12 do *Robotware* de forma a garantir compatibilidade de versões. O mesmo se aplica ao *Secreenmaker*, ferramenta usada para desenvolver aplicações para controlo dos programas a partir da consola de programação, que deverá ser também a versão 5.12, caso contrário não será possível transferir os programas para a consola e controlador físico do robô.

Os robôs possuem dois modos de funcionamento, o manual e o automático. O modo manual é usado para fazer programação *on-line.* Neste modo o utilizador ensina localizações ao robô recorrendo a *jogging* manual. O programa que contém esse conjunto de localizações com diversas instruções associadas, representa uma tarefa. Em modo automático, esse programa é carregado no controlador do robô e quando lançado, o robô executa a tarefa programada automaticamente. Ou seja, em modo automático, um programa é executado e o robô não pode ser movido manualmente pelo que os procedimentos têm que ser executados em modo manual. Durante as simulações verificou-se uma dificuldade em articular as instruções de movimentação automáticas, invocadas no programa de calibração, com as de movimentação manual com *joystick* da consola. Este problema também afetou as funcionalidades dos menus criados para a consola com o *Screenmaker*. No Anexo 6 encontram-se indicações para contornar estes problemas encontrados.

5.3 Testes e resultados

Os procedimentos de calibração do TCP disponíveis e programados foram validados experimentalmente na célula robótica com o robô (ABB IRB2400/16) e usando as duas ferramentas selecionadas, garra e *spindle*.

5.3.1 Medição geométrica com o robô

Este procedimento inicia-se ajustando os valores das juntas do manipulador para que este assuma uma configuração em que o eixo da ferramenta fica paralelo ao eixo Z do referencial de base do robô. O ajuste individual da posição de cada uma das juntas pode ser feito através da consola de programação como indicado na Figura 42.

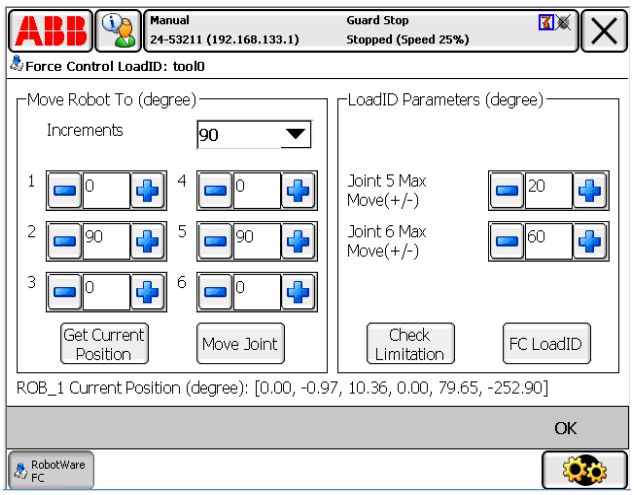


Figura 42 – Interface para configuração de juntas

Segue-se a ativação do controlo de força para movimentação manual. Note-se que ao fazer *jogging* cartesiano manual em controlo de força, a orientação da ferramenta não se perde tal como representado na Figura 43 (2). O utilizador leva a ponta da ferramenta a contactar com uma superfície de referência tal como representado na Figura 43 (1). Neste momento acede ao menu *jogging* da consola e anota as coordenadas de posição do robô no referencial de base tendo como ferramenta ativa o Tool0. Depois leva a flange à superfície de referência, tal como ilustrado na Figura 43 (3), e anota novamente a posição de Tool0. Na Figura 43 (4) o utilizador verifica com um esquadro se a flange de montagem da ferramenta se encontra na posição pretendida.

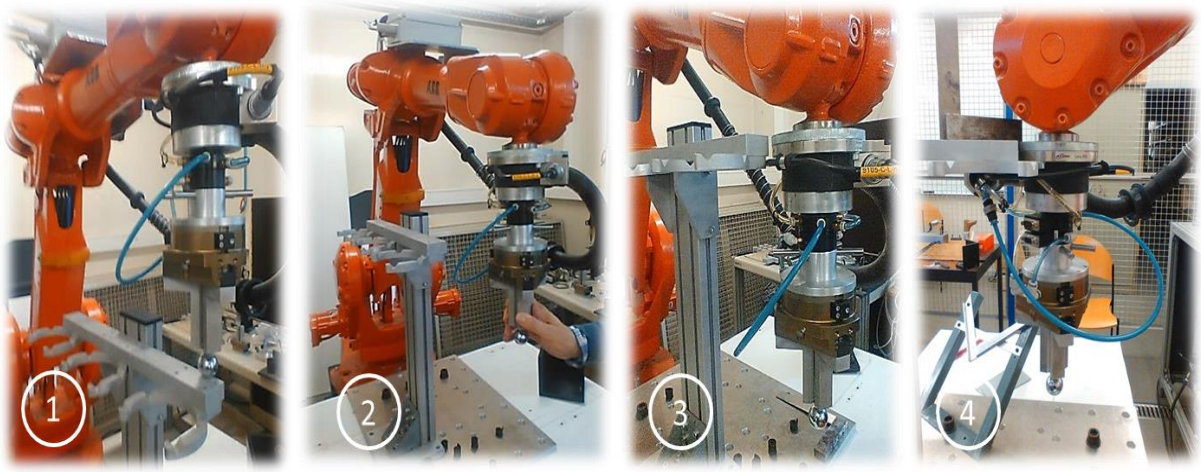


Figura 43 – Sequência de execução do procedimento - Medição geométrica com o robô

O aspeto gráfico da consola nas posições (1) e (3) da Figura 43 e as coordenadas de posição de Tool0 nessas posições encontram-se na Figura 44.

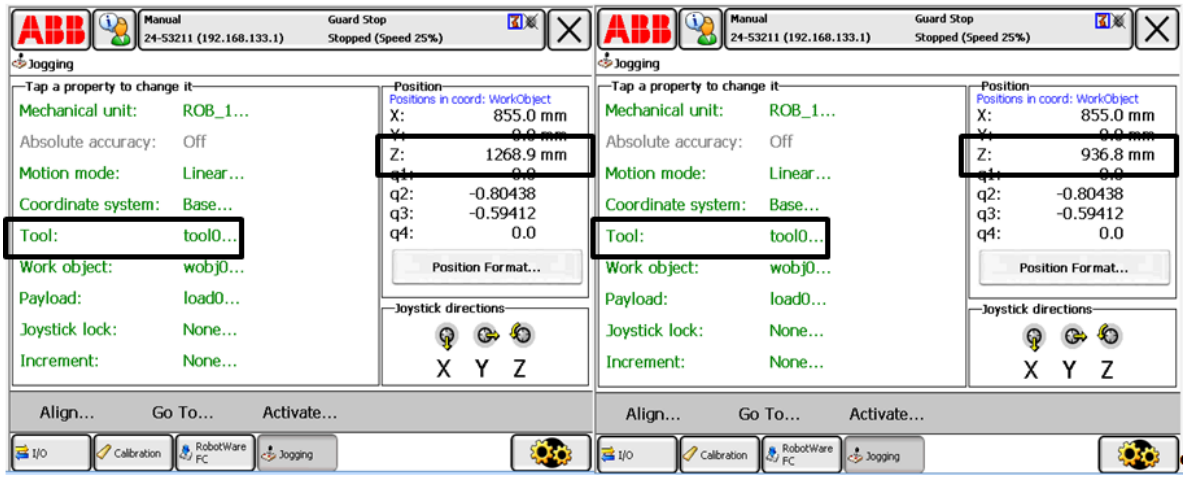


Figura 44 – Recolha das coordenadas de posição de Tool0

O TCP para o centro da esfera é calculado manualmente com a diferença da coordenada Z de Tool0 na posição (1) pela da posição (3) e subtraindo o raio de 14mm da esfera. Assume-se que a ferramenta está perfeitamente alinhada logo o valor das coordenadas X e Y do TCP são zero. Assim sendo, as coordenadas do TCP da garra são [0,0; 0,0; 318,1] mm. Este procedimento foi apenas testado com a garra porque só consegue determinar o TCP de ferramentas cuja extremidade pertença ao eixo Z de Tool0, o que não acontece com o *spindle*.

5.3.2 Método da ABB

Este procedimento é lançado a partir da consola de programação, como visto anteriormente, e permite calcular o TCP de uma ferramenta levando a extremidade da mesma a um ponto fixo do espaço de trabalho com pelo menos três orientações diferentes. O procedimento está exemplificado sequencialmente na Figura 45 para a garra (1) e (2) e para o *spindle* (3) e (4). Note-se o uso da esfera e do respetivo suporte durante a execução do mesmo tal como sugerido na Secção 5.1.4.

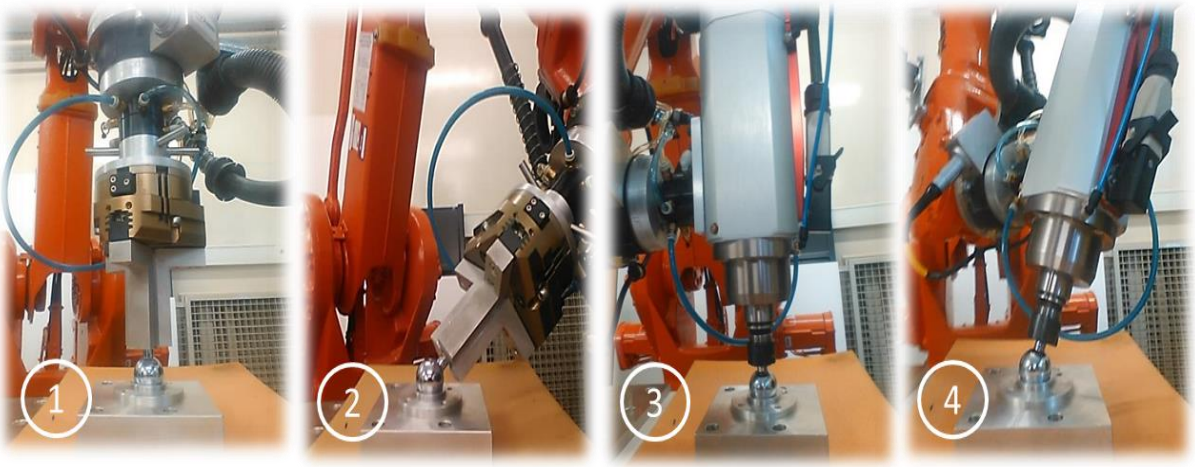


Figura 45 – Sequência de execução do procedimento - Método da ABB

O resultado obtido para a posição do TCP da garra foi de [0,57; -0,47; 317,01] mm e do *spindle* [-65,52; -154,07; 174,64] mm. Estes valores são apresentados na consola de programação como ilustrado na Figura 46. Se forem validados pelo utilizador, os valores ficam armazenados na memória do controlador.

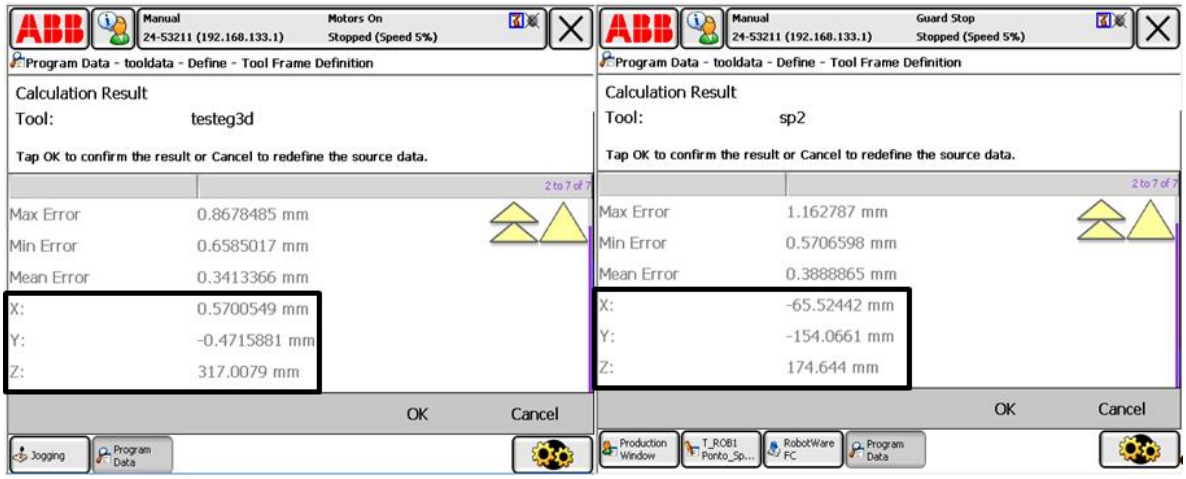


Figura 46 – Resultado do método da ABB para a garra (esquerda) e *spindle* (direita)

5.3.3 Procedimento PNT_1

Este procedimento foi executado apenas para determinar as coordenadas do TCP do *spindle*. Calibrou-se a garra com o método da ABB para fazer a localização do ponto de calibração no espaço de trabalho. Para isso, levou-se a garra à base de assento da esfera manualmente em controlo de força tal como representado na Figura 47 (1), e definiu-se esse ponto como o ponto de calibração, anotando as suas coordenadas no referencial de base do robô.

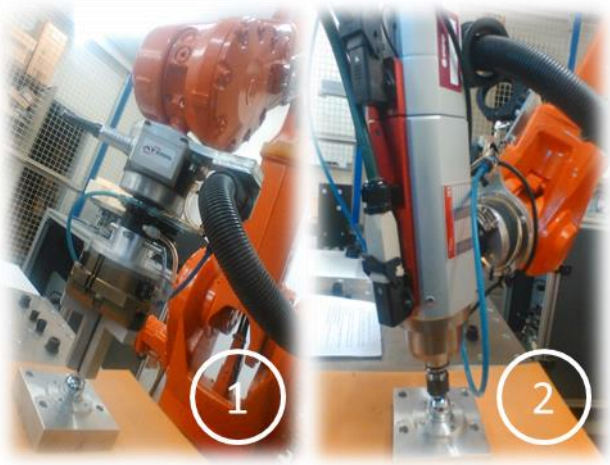


Figura 47 – Sequência de execução do procedimento - PNT_1

Depois montou-se o *spindle* com a esfera no manipulador e levou-se ao ponto de calibração tal como representado na Figura 47 (2). Nesta posição executou-se o programa desenvolvido que calcula o TCP baseando-se na posição e orientação de Tool0 e nas coordenadas do ponto de calibração localizado segundo a metodologia apresentada na Secção 3.2.1 para o cálculo do TCP com um ponto de referência segundo a patente US 5194792. O resultado obtido para o *spindle* foi de [-63,67; -155,28; 173,14] mm e pode ser visualizado a na Figura 48.

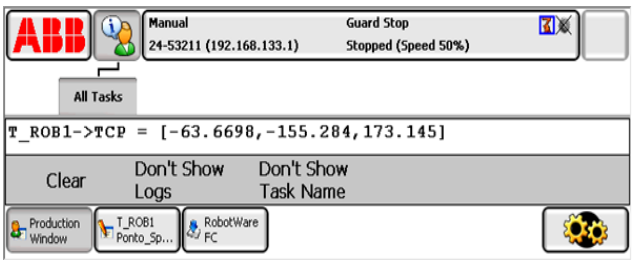


Figura 48 – Resultado do procedimento PNT_1 para *spindle*

5.3.4 Procedimento PLN_1

Este procedimento, representado na Figura 49 para o *spindle* consiste em levar a ferramenta a uma superfície horizontal com orientações diferentes. A sequência das etapas deste procedimento estão assinaladas sequencialmente de (1) a (6).

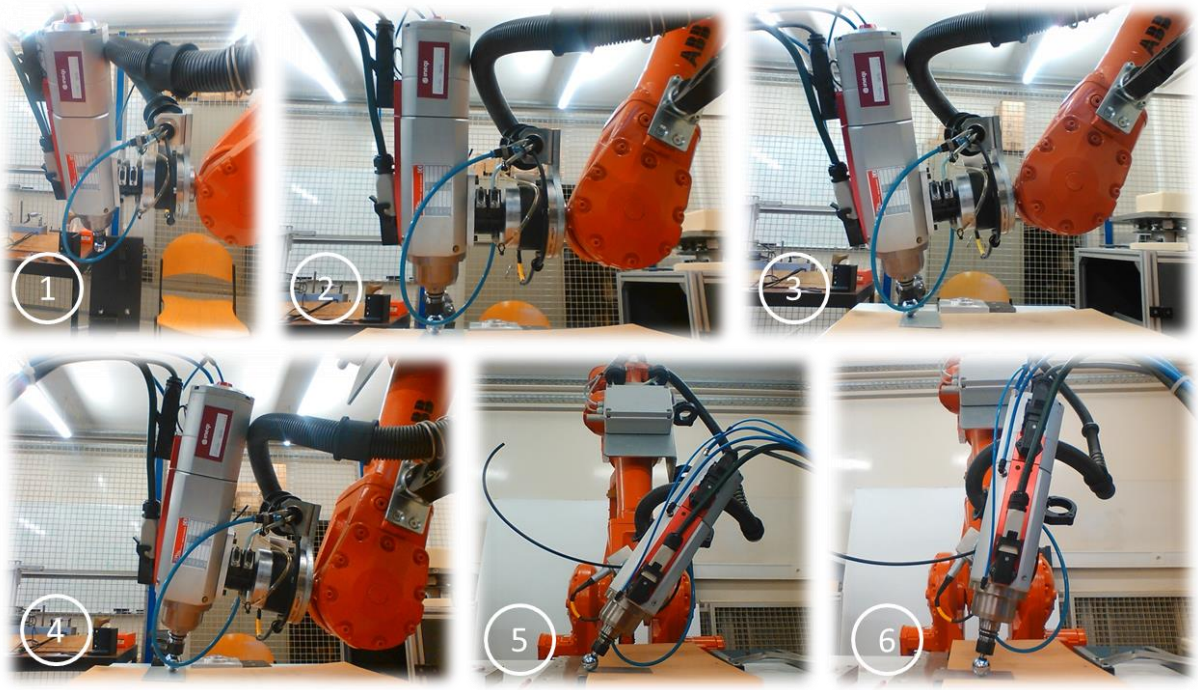


Figura 49 – Sequência de execução do procedimento - PLN_1

O TCP é calculado com os pontos memorizados com aplicação das equações apresentadas na Secção 3.2.2 para a calibração com um plano de referência segundo a patente US 5910719. Os resultados obtidos experimentalmente para o TCP da garra e do *spindle* foram de [-0,657; 0,546; 317,924] mm e de [-64,217; -153,546; 172,568] mm, respetivamente. A Figura 50 ilustra os referidos valores obtidos.



Figura 50 – Resultado do procedimento PLN_ 1 para a garra (esquerda) e *spindle* (direita)

5.3.5 Procedimento PLN_2

Neste procedimento é feito o reconhecimento da posição e orientação de um plano fixo, que é depois usado para servir de referência ao cálculo das coordenadas do TCP de uma ferramenta. A orientação é determinada levando a ferramenta a três alvos circulares marcados sobre o plano como ilustrado na Figura 51 (1), (2) e (3). A posição fica definida levando a ferramenta a contactar com as duas faces do plano tal como representado na Figura 51 (4) e (5). Conhecida a posição e orientação, o TCP é calculado levando a ferramenta a contactar perpendicularmente com o plano como se pode ver na Figura 51 (6) e de acordo com as equações apresentadas na Secção 3.2.2 para a calibração com um plano de referência.

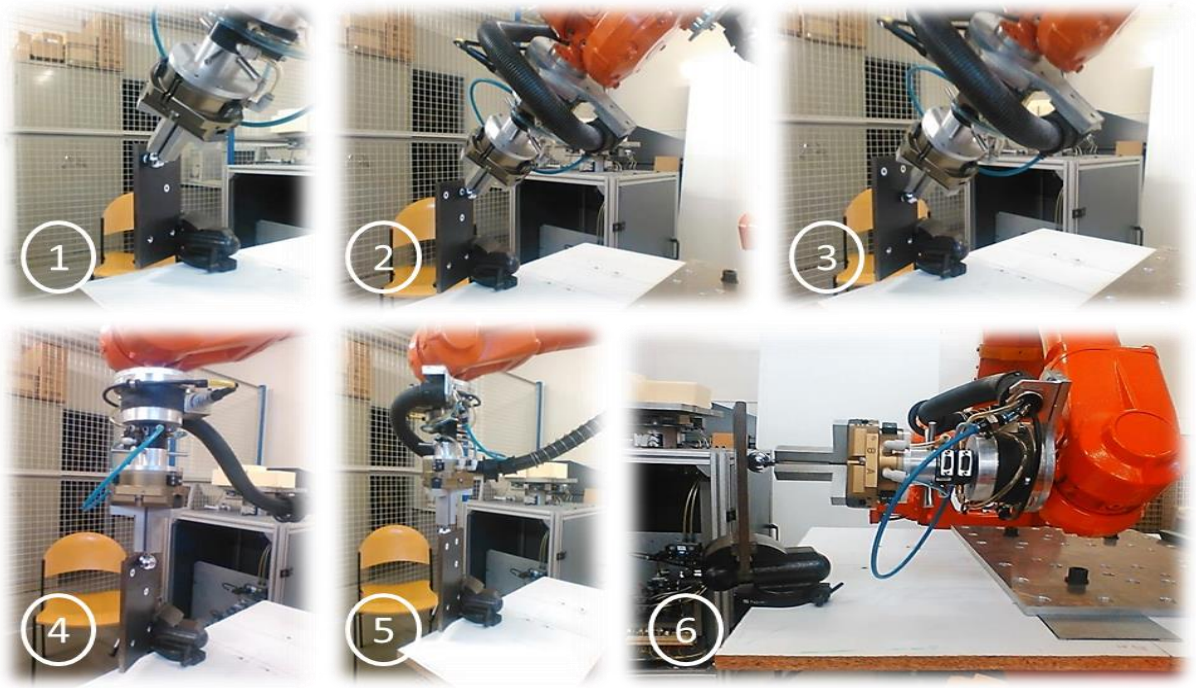


Figura 51 – Sequência de execução do procedimento - PLN_2

O resultado obtido encontra-se na Figura 52 tendo-se registado [-0,986; 0,468; 317,432] mm para o TCP da garra e [-65,8224; -154,46; 171,427] mm para o TCP do *spindle*.

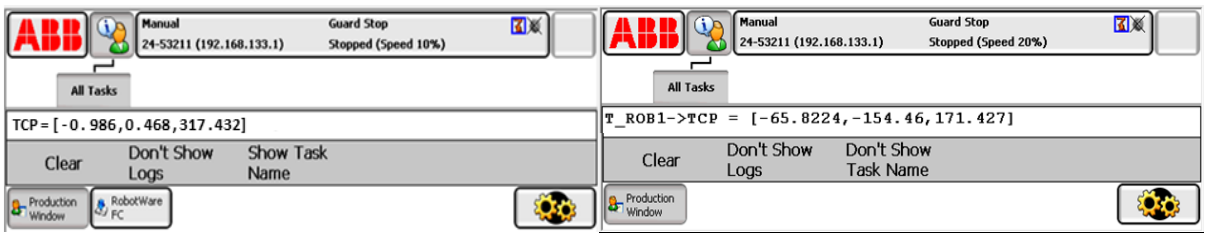


Figura 52 – Resultado do procedimento PLN_2 para a garra (esquerda) e *spindle* (direita)

5.4 Avaliação quantitativa e qualitativa dos procedimentos

A Tabela 2 sumariza os resultados da determinação das coordenadas do TCP da garra e do *spindle* utilizando os diferentes procedimentos.

Tabela 2 - Coordenadas do TCP resultantes dos procedimentos ensaiados

Procedimento	**TCP para a Garra [mm]**				**TCP para o *Spindle* [mm]**		
	X	Y	Z		X	Y	Z
Medição geométrica com o robô	0.00	0.00	318.10				
Método da ABB	0.57	-0.47	317.01		-65.52	-154.07	174.64
PNT_1					-63.67	-155.28	173.14
PLN_1	-0.66	0.55	317.92		-64.22	-153.55	172.57
PLN_2	-0.99	0.47	317.43		-65.82	-154.46	171.43

Com se pode ver na Tabela 2, os valores obtidos para as coordenadas do TCP de cada uma das ferramentas são semelhantes. Para validar os resultados, os valores obtidos foram usados para reorientar as ferramentas em torno de um ponto fixo. Verificou-se que durante a reorientação a ponta das ferramentas se mantinha sobre o ponto fixo. Isto permite concluir que todos os procedimentos testados foram bem sucedidos na determinação das coordenadas do TCP. As diferenças estão relacionadas com o facto de os procedimentos terem associadas metodologias distintas de cálculo das coordenadas do TCP. Admite-se que em ambiente controlado, e com uma execução mais rigorosa, seria possível aproximar ainda mais os valores.

Os ensaios efetuados permitem também fazer uma avaliação qualitativa dos procedimentos no sentido de selecionar o mais expedito. Na Tabela 3 encontra-se a síntese das principais características dos procedimentos, o que permitirá fazer tal avaliação.

Tabela 3 - Avaliação qualitativa dos procedimentos

Procedimento	Referencia Geométrica	*Jogging*	Reorientação	Recolha de Dados	Cálculo	Configuração inicial	Tempo execução
Método de introdução direta	NA	NA	NA	Semiautomático	Automático	Não	Curto
Medição geométrica com o robô	Plano	Manual	Manual	Semiautomático	Manual	Não	Médio
Método da ABB	Ponto	Manual	Manual	Automática	Automático	Sim	Longo
PNT_1	Ponto	Manual	NA	Automática	Automático	Sim	Curto
PLN_1	Plano	Manual	Automática	Automática	Automático	Não	Médio
PLN_2	Plano	Manual	Automática	Automática	Automático	Sim	Longo

(NA-Não se aplica)

Concluiu-se que dos procedimentos ensaiados, o mais expedito é o PNT_1 baseado na patente US 5194792, porque, apesar de necessitar de uma configuração inicial, o procedimento associado a essa configuração é apenas executado uma vez antes da primeira utilização do procedimento de calibração. Não necessita de reorientação da ferramenta, a recolha de dados para a calibração e o cálculo do TCP são processos completamente automáticos, o que faz com que este procedimento tenha associado também um tempo de execução reduzido. Além disso, o *jogging* manual pode ser feito ativando o controlo de força.

5.5 Aplicação de sistema de calibração a um *spindle*

Os procedimentos de calibração vistos, apenas permitem determinar o TCP da ferramenta. Na prática, para se poder usar uma ferramenta no manipulador robótico, é necessária uma definição mais completa, que inclui a orientação do referencial, bem como a sua massa e centro de gravidade. Assim sendo, um sistema de calibração deve ser visto como um conjunto de etapas, que permitem a definição completa de qualquer ferramenta.

Dos procedimentos de determinação do TCP avaliados experimentalmente, viu-se que a solução PNT_1 é a mais expedita. Apresenta-se nesta secção um sistema de calibração que procura integrar este procedimento tirando partido do controlo de força disponível no controlador do robô real, e integrando os desenvolvimentos efetuados e equipamentos auxiliares disponíveis, nomeadamente o uso da esfera de calibração e respetivo suporte para assento e acomodação da sua posição, de forma a ir ao encontro dos requisitos inicialmente estabelecidos.

O sistema de calibração de ferramenta envolve a realização das seguintes etapas:

- Calibração de uma ferramenta padrão com o método da ABB;
- Configuração do sistema usando a ferramenta padrão para identificação de um ponto para calibração;
- Determinação de massa e centro de gravidade de uma nova ferramenta;
- Cálculo das coordenadas do TCP segundo PNT_1;
- Determinação da orientação do referencial da ferramenta;
- Verificação do resultado da calibração.

Os procedimentos envolvidos neste sistema já foram abordados em detalhe ao longo do trabalho, com exceção para o procedimento de determinação da orientação do referencial da ferramenta, que se refere aqui com maior ênfase. Assim sendo, optou-se por apresentar o sistema de calibração com o exemplo prático da calibração do *spindle* com o robô da célula robótica disponível. O sistema foi desenvolvido e simulado com o *software* de programação RobotStudio, e dispõe de uma aplicação produzida através da funcionalidade *Screenmaker*, para a consola de programação, que serve de auxílio ao utilizador durante as etapas de calibração.

Sendo a primeira vez que se usa este sistema, é necessário fazer a sua configuração. No espaço de trabalho é colocado o suporte de assento da esfera, por hipótese sobre a mesa posicionadora da célula robótica tal como representado na Figura 53.

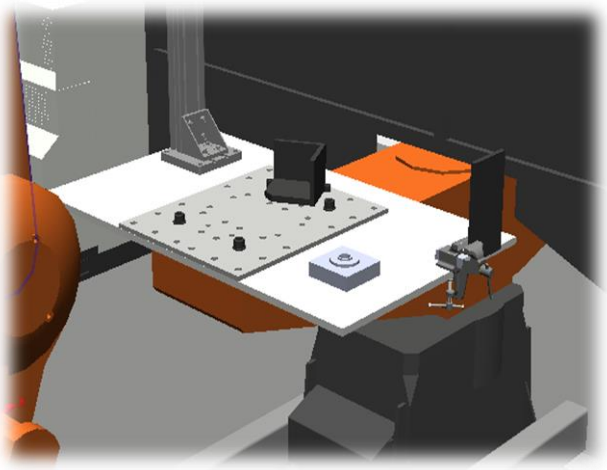


Figura 53 – Posição do suporte de assento da esfera sobre a mesa de trabalho

Para fazer o reconhecimento da localização do suporte de assento, que será o ponto de calibração, deve usar-se uma ferramenta padrão. Idealmente a ferramenta padrão é uma ferramenta específica para proceder à identificação da posição do ponto de calibração, com uma geometria simples e cujo TCP seria bem conhecido. Na ausência de uma ferramenta apenas para efeitos de configuração do sistema, pode ser usado um qualquer outro objeto que possa ser fixo à flange de montagem de ferramenta do robô. Admita-se que o robô tem montada a garra, à qual se fixa a esfera, e que esta será usada como ferramenta padrão, ver Figura 54.

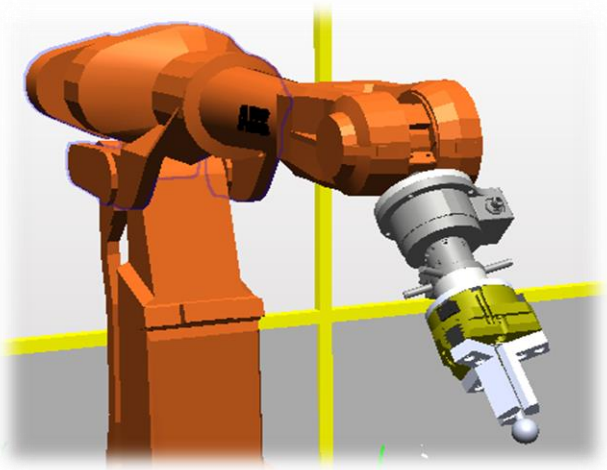


Figura 54 – Modelo virtual da Garra

Este sistema integra o uso do controlo de força para movimentação manual do robô, o que implica uma definição completa da ferramenta padrão (massa, centro de gravidade, TCP e orientação). Não sendo conhecida esta informação para a garra, executa-se inicialmente a determinação da massa e centro de massa acedendo ao menu Configuração>Carga representado na Figura 55.

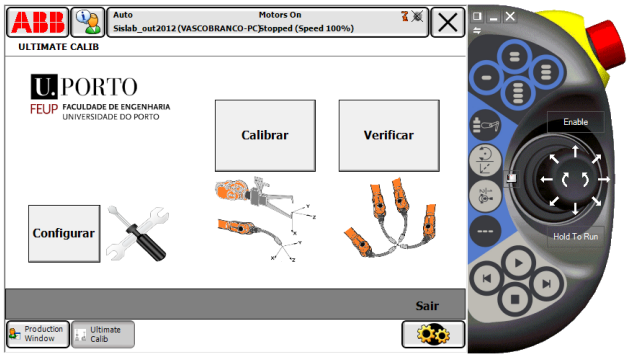


Figura 55 – Interface criada para a consola de programação

O utilizador é informado de que o robô irá iniciar uma série de movimentos em torno das juntas 5 e 6 para determinar a massa e o centro de gravidade da ferramenta com o sensor de força. Esta funcionalidade é disponibilizada pelo *software* do controlador do robô. O resultado para a garra encontra-se na Figura 56.

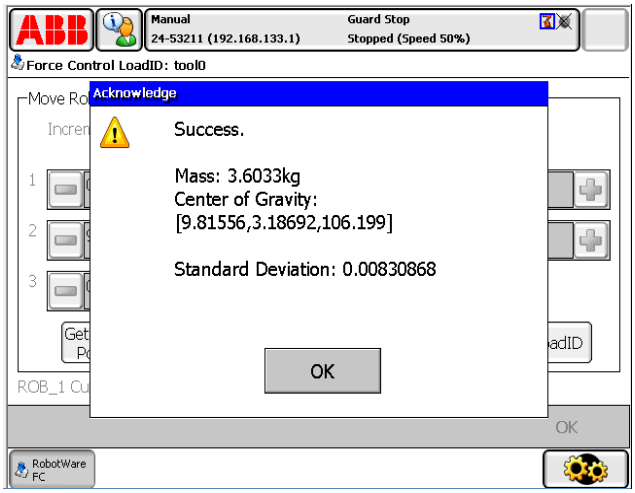


Figura 56 – Cálculo da massa e centro de gravidade da garra

Seguidamente aplica-se o método da ABB para fazer o reconhecimento do TCP, aproveitando-se a disponibilidade da esfera e suporte de assento para auxílio na reorientação em torno do ponto fixo de referência. No menu configurar da consola, acedendo a TCP, o utilizador lança o procedimento de cálculo do TCP. No sentido de facilitar a calibração, o robô reorienta-se automaticamente e o utilizador apenas tem que levá-lo manualmente ao ponto fixo (suporte de assento da esfera) como indicado na Figura 57.

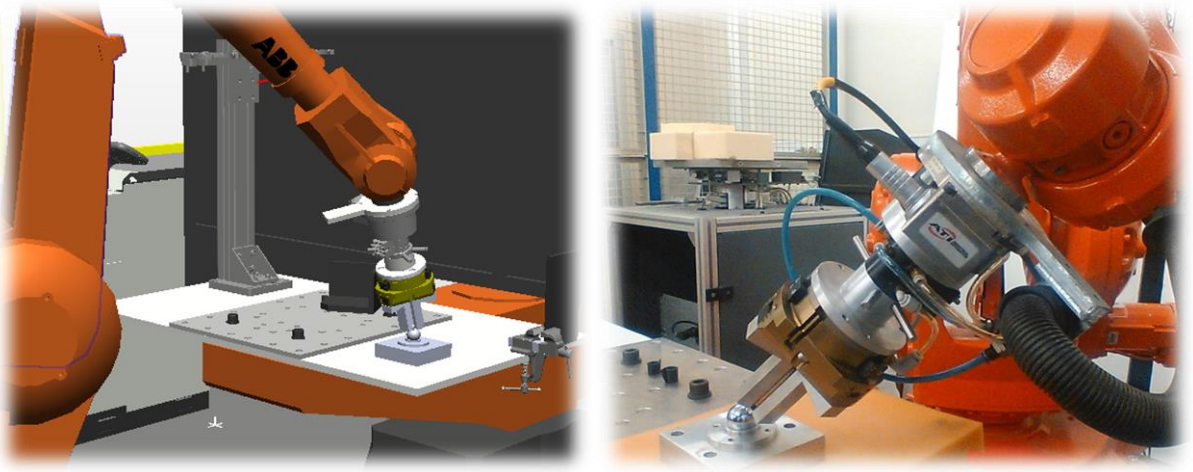


Figura 57 – Cálculo do TCP da garra no modelo virtual (esquerda) e real (direita)

Determinado o TCP da garra, o robô pode ser usado como instrumento de medição de coordenadas. Assim sendo, levando a esfera ao suporte de assento com controlo de força, é possível determinar as coordenadas de posição do suporte no referencial de base do robô, e definir o centro do mesmo como o ponto de calibração. A este ponto é associada uma orientação predefinida.

Está assim concluída a configuração do sistema. Agora, para determinar o TCP de qualquer ferramenta, basta acoplar a esfera à ferramenta, levar a ferramenta ao ponto de calibração com controlo de força e chamar, através da consola, o programa que aplica as equações de cálculo do TCP acedendo ao menu Calibrar>TCP. No mesmo menu pode também definir-se a massa e centro de massa da ferramenta e a orientação.

A orientação predefinida para o ponto de calibração é atribuída ao TCP na posição em que a ferramenta se encontra no contacto com o ponto de calibração. Assim sendo, é importante que ao levar a ferramenta ao ponto de calibração, se posicione a mesma de forma a que a orientação predefinida coincida com a orientação pretendida para a ferramenta, tal como representado na Figura 58. Alternativamente o utilizador pode optar por atribuir diretamente a orientação de Tool0 à ferramenta.

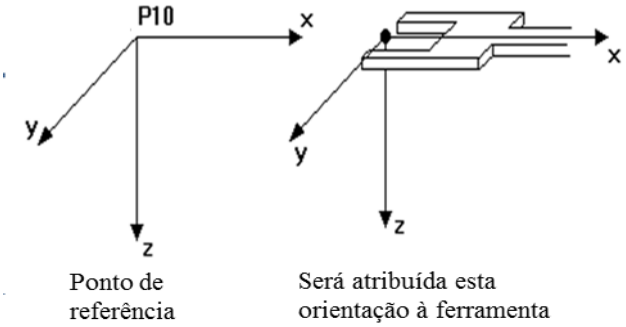


Figura 58 – Atribuição de TCP e orientação a uma ferramenta

Por vezes, pode não ser prático posicionar a ferramenta de forma a que a orientação predefinida seja corretamente atribuída, e a orientação do referencial Tool0 pode não ser a desejada. O *spindle* é um exemplo de uma ferramenta nesta situação. Para o *spindle* pretende-se que o eixo Z fique coincidente com o eixo de rotação do motor-árvore e apontando em sentido contrário ao do suporte de fixação da ferramenta de corte. O *spindle* da célula robótica possui dois tamanhos de suporte de fixação da ferramenta, ver Figura 59.



Figura 59 – Suportes de fixação da ferramenta do *spindle*

A calibração da orientação do *spindle* passa então por aceder ao menu Calibrar>Orientação, e seguir as instruções indicadas que incluem, montar a esfera no suporte pequeno, levar a ferramenta ao ponto de calibração, e determinara as coordenadas do TCP, ou seja, do centro da esfera como $\mathbf{p}_{TCP}$, tal como indicado na Figura 60. Depois, repetir o procedimento mas alterando o comprimento da ferramenta ou seja, montando a esfera no suporte grande. O TCP nesta situação é $\mathbf{p}_{TCPext}$.

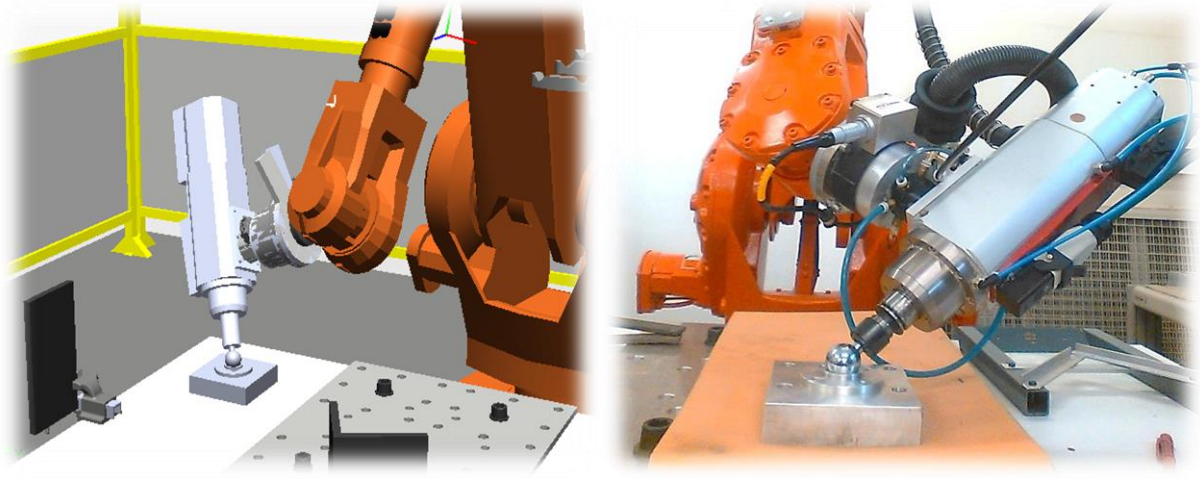


Figura 60 – *Spindle* sobre o ponto de referência no modelo virtual (esquerda) e real (direita)

O eixo de rotação da ferramenta é obtido com a informação dos dois pontos pela diferença dos vetores $\mathbf{p}_{TCPext}$ e $\mathbf{p}_{TCP}$. O versor do eixo de rotação da ferramenta vem então dado pela Equação 5.2 e faz-se com que seja o eixo Z do referencial da ferramenta.

$$\begin{bmatrix} TCPext_x \\ TCPext_y \\ TCPext_z \end{bmatrix} - \begin{bmatrix} TCP_x \\ TCP_y \\ TCP_z \end{bmatrix} = \begin{bmatrix} z_x \\ z_y \\ z_z \end{bmatrix} \tag{5.2}$$

Os restantes dois eixos são obtidos fazendo com que $\mathbf{y}$ seja paralelo à flange, impondo as seguintes condições,

$$\begin{cases} y_z = 0 \\ z_x \times y_x + z_y \times y_y = 0 \\ {y_x}^2 + {y_y}^2 = 1 \end{cases} \tag{5.3}$$

O eixo $\mathbf{x}$ é dado pelo produto vetorial dos versores dos eixos $\mathbf{y}$ e $\mathbf{z}$.

$$\mathbf{x} = \mathbf{y} \times \mathbf{z} \tag{5.4}$$

Conhecida a orientação, o TCP para o *spindle* é calculado lançando o programa de calibração com base na distância D1, e D2 indicadas na Figura 61.

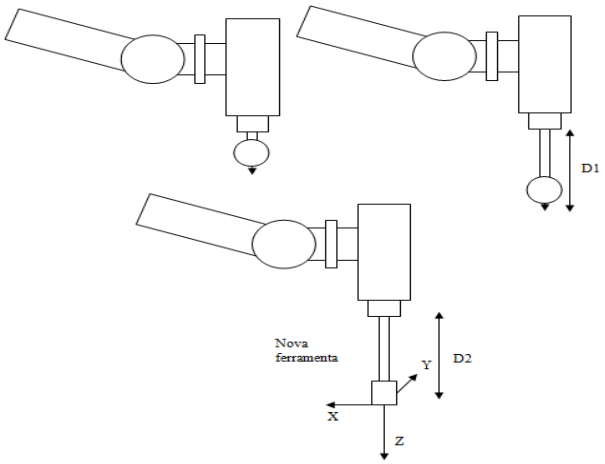


Figura 61 – Procedimento de determinação da orientação do eixo de rotação

Se a orientação do *spindle* ficar guardada, bem como a distância D1, a calibração da ferramenta é feita apenas indicando D2 e executando o programa de calibração. Isto significa que, o procedimento de determinação da orientação do *spindle* pode também considerar-se uma configuração. Na Figura 62 encontra-se a medição das distâncias D1 e D2 e fixação da ferramenta de corte, no suporte de ferramenta, à distância D2.

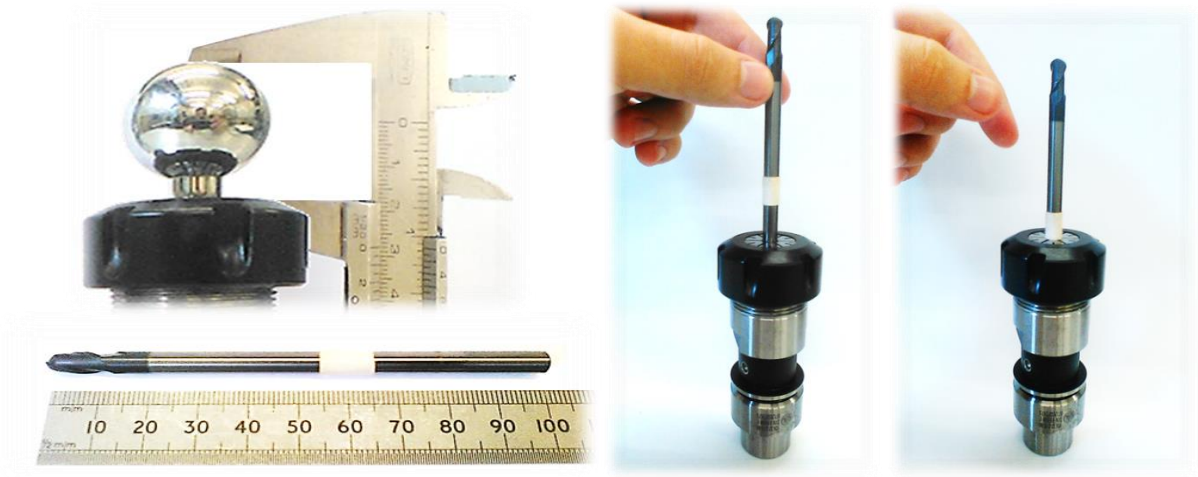


Figura 62 – Medição das distâncias D1 e D2 e fixação da ferramenta

A massa e centro de gravidade para o *spindle* também foram calculados acedendo ao menu Calibração>Carga. O resultado encontra-se na Figura 63.

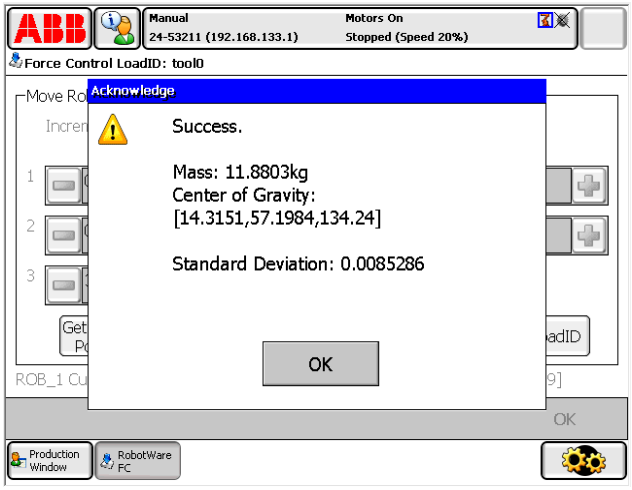


Figura 63 – Cálculo da massa e centro de gravidade do *spindle*

O procedimento de verificação da posição do TCP só pode ser executado se a ferramenta já estiver completamente definida. Destina-se a avaliar se ocorreu um desvio em relação à posição inicialmente carregada. Para fazer a verificação, o utilizador deve aceder ao menu Verificação através da consola de programação. O robô executa uma movimentação automática em torno de um ponto fixo. O utilizador verifica se a ponta da ferramenta se desvia ou não desse ponto. Desvios significativos indicam necessidade de recalibração.

No Anexo 7 encontra-se a versão integral do código RAPID gerado para a implementação deste sistema de calibração.

Capítulo 6 – Conclusões e perspetivas de trabalho futuro

A realização deste trabalho permite tecer algumas conclusões relativamente a sistemas de calibração de ferramenta para manipuladores robóticos, que usam o robô como instrumento de medição, e objetos de referência de auxílio à calibração.

Os manipuladores robóticos são preparados para trabalhar com diversas ferramentas. Cada ferramenta é definida no controlador, como um referencial cartesiano, cuja origem é o ponto central da extremidade da ferramenta, designado por TCP. Este é o ponto comandado a seguir as trajetórias e atingir localizações do espaço de trabalho quando o robô executa uma tarefa. Os sistemas de calibração de ferramenta fazem o reconhecimento da posição da extremidade da ferramenta em relação à flange de montagem, atribuem-lhe um referencial com uma dada orientação, e escrevem essa informação no controlador do robô. Assim sendo, conclui-se que os sistemas de calibração de ferramenta são dispositivos de extrema relevância para manipuladores robóticos, pois o sucesso das operações é dependente da correta definição da ferramenta no controlador.

As soluções de sistemas de calibração para robôs industriais disponíveis no mercado, são dispositivos que recorrem a tecnologias avançadas. Da análise deste tipo de sistemas comerciais, concluiu-se que a maioria usa dispositivos de medição externos baseados em tecnologias laser. Contudo, há soluções que se servem das funcionalidades do próprio robô para fazer a calibração, usando-o como máquina de medição de coordenadas. De entre estes dispositivos destacam-se os que estabelecem contacto físico com uma referência para cálculo do TCP. Trata-se de um *target* de calibração, com uma restrição geométrica, que pode ser um ponto, uma linha, um plano ou uma superfície esférica.

Da comparação da performance de um procedimento de calibração usando um ponto como referência geométrica, quando usados modelos matemáticos distintos de processamento dos mesmos dados, concluiu-se que um mesmo procedimento pode ter associadas metodologias de cálculo diferentes, mas isso condiciona os resultados obtidos com o procedimento em questão. Os testes realizados indicaram que o erro na determinação da posição do referencial da ferramenta, é função do número de pontos recolhidos, da posição relativa desses pontos e da amplitude de perturbação introduzida. Há metodologias capazes de absorver mais facilmente perturbações nos pontos adquiridos e imunes ao facto de os pontos usados serem muito próximos. Dos métodos analisados usando um ponto com referência geométrica, concluiu-se que o de minimização do erro é mais robusto que o da superfície esférica que melhor se ajusta aos pontos de calibração adquiridos. Adicionalmente, os resultados com estes métodos são tanto melhores quanto maior for o número de pontos usados.

Com a revisão dos procedimentos de calibração usados na célula robótica disponível, tendo em conta o contexto dos trabalhos desenvolvidos, o tipo de ferramentas utilizadas e os períodos de operação, concluiu-se que é possível estabelecer estratégias não só no sentido de tornar mais prática a execução dos procedimentos aplicados, mas também de aumento da confiabilidade nos resultados proporcionados pelos mesmos. Verificou-se por exemplo, que usar uma esfera fixa à ferramenta durante a execução dos procedimentos que usam superfícies de referência, facilita o contacto da ferramenta com a superfície e melhora os resultados obtidos. Adicionalmente, o robô da célula em estudo está equipado com controlo de força. Trata-se de uma funcionalidade que permite a movimentação manual do robô. Concluiu-se que a disponibilidade de movimentação manual do robô com o controlo de força torna também a execução do procedimento mais fácil do que com a movimentação com o *joystick* da consola de programação, e simplifica o uso do próprio robô como instrumento de medição.

A análise dos fundamentos teóricos que dizem respeito à calibração de ferramenta, como por exemplo, saber que a cinemática direta permite localizar o referencial da ferramenta no espaço de trabalho a partir dos valores de posição das juntas tendo em conta a estrutura do manipulador, revelou ser comum e aplicável à maioria dos procedimentos de calibração da posição do referencial da ferramenta. O domínio destes conceitos matemáticos permitiu analisar diversos procedimentos e basear novas soluções em métodos existentes.

A exploração de metodologias alternativas incluiu o desenvolvimento de *software*, não só para simulação, mas também para verificação experimental dos procedimentos. Os valores obtidos dos ensaios realizados para determinar as coordenadas do TCP para a mesma ferramenta, usando diferentes procedimentos de calibração, foram próximos. Isto permite concluir quanto à precisão dos procedimentos avaliados. Ao carregar os valores obtidos no controlador do robô, verificou-se que durante a reorientação, a ponta das ferramentas se mantinha sobre um ponto fixo. Isto permite concluir também quanto à validação dos procedimentos.

As ligeiras diferenças foram atribuídas ao facto de os procedimentos terem associadas metodologias distintas de cálculo das coordenadas do TCP. Admite-se que em ambiente controlado, e com uma execução mais rigorosa, seria possível aproximar ainda mais os valores. Apesar de todos os procedimentos examinados serem capazes de adquirir a informação do TCP, alguns revelaram ser mais expeditos.

O uso de um ponto conhecido como referência geométrica é, dos procedimentos ensaiados, o mais expedito. Apesar de necessitar de uma configuração inicial, o procedimento associado a essa configuração é apenas executado uma vez antes da primeira utilização como procedimento de calibração. Não necessita de reorientação da ferramenta, a recolha de dados para a calibração e o cálculo do TCP são processos completamente automáticos, o que faz com que este procedimento tenha associado também um curto tempo de execução. Além disso, o *jogging* manual pode ser feito ativando o controlo de força.

Concluiu-se que, recorrendo a um *software* de programação *off-line* de robôs, é possível implementar procedimentos de calibração do TCP alternativos aos incorporados nos modelos dos robôs. Estes procedimentos alternativos podem ser baseados em sugestões propostas na literatura. Isto motivou o desenvolvimento de um sistema de calibração que procura integrar o procedimento selecionado como mais expedito, tirando partido do controlo de força disponível no controlador do robô real, e integrando os desenvolvimentos efetuados e equipamentos auxiliares disponíveis. Nomeadamente o uso da esfera de calibração e respetivo suporte de assento fixo no espaço de trabalho, que se revelou ser um fator determinante para a qualidade final da calibração.

Os procedimentos de calibração avaliados experimentalmente, apenas permitem determinar o TCP da ferramenta. Na prática, para poder usar uma ferramenta no manipulador robótico, é necessária uma definição mais completa, que inclui a orientação do referencial da ferramenta bem como a sua massa e centro de gravidade. Assim sendo, um sistema de calibração deve ser visto como um conjunto de etapas, que permitem a definição completa de qualquer ferramenta.

Concluiu-se que um sistema de calibração em controlo de força é composto por um procedimento de determinação da massa e centro de gravidade da ferramenta, por um procedimento de cálculo do TCP e outro de cálculo da orientação do referencial da ferramenta, sendo o procedimento de determinação das coordenadas do TCP mais exigente do que o de determinação da orientação.

Tendo este projeto criado uma base sólida de conhecimentos quanto à calibração de ferramenta de manipuladores robóticos, é possível propor a execução de outros estudos de modo a aperfeiçoar a solução implementada, por exemplo, desenvolver mais a aplicação criada com o *Screenmaker*, no sentido de obter uma interface mais apelativa e com menus de seleção e edição da ferramenta.

Como desenvolvimentos futuros adicionais, sugere-se a exploração de sistemas de calibração de ferramenta, que não usando dispositivos de medição externos, recorrem a uma unidade externa de deteção da posição e orientação da ferramenta. Os sistemas que usam um sensor de forquilha, por exemplo, são aceites como uma solução à medida para o problema da calibração de ferramenta no meio industrial.

A célula robótica usada, realiza trabalhos variados por projeto, ou seja, não está inserida numa linha de produção em série, em que é fundamental fazer verificações periódicas automáticas à posição do referencial da ferramenta, com correção imediata em caso de desvio. Ainda que neste trabalho o objetivo fosse integrar o controlo de força para movimentação manual, o que restringe os sistemas de calibração a sistemas semiautomáticos, seria igualmente interessante explorar soluções de calibração e verificação completamente automáticas, usando por exemplo o controlo de força para a deteção do contacto entre a ferramenta e um objeto.

Referências

[1] COMET. *"Plug-and-produce COmponents and METhods for adaptive control of industrial robots enabling cost effective, high precision manufacturing in factories of the future"* (s.d). Disponível online em: www.cometproject.eu. Acedido em Março de 2015.

[2] Oliveira, Luis. (2013) *"Maquinagem de superfícies complexas com recurso a sistema robótico"*. Porto: Faculdade de Engenharia da Universidade do Porto.

[3] Robot welding. *"Robot welding, The essential guide"* (2015). Disponível online em: www.robotwelding.co.uk. Acedido em Abril de 2015.

[4] ABB Developer Center. *"Introduction to RobotStudio, Terms and concepts, Coordinate systems"*. (s.d). Disponível online em: www.developercenter.robotstudio.com. Acedido em Março de 2015.

[5] Renishaw. *"Telescoping ballbar systems"* (s.d). Disponível online em: www.renishaw.com. Acedido em Maio de 2015.

[6] Bergstöm, Gustav (2011). *"Method for calibration of off-line generated robot program".* Sweden: Chalmers University of Technology.

[7] Leica. *"Laser Tracker Systems"* (s.d). Disponível online em: www.leica-geosystems.com. Acedido em Junho de 2015.

[8] Faro. *"Advanced, accurate, and versatile laser trackers"* (s.d). Disponível online em: www.faro.com. Acedido em Março de 2015.

[9] Wiest AG. *"Calibration Systems"* (s.d). Disponível online em: www.wiest-ag.com. Acedido em Março de 2015.

[10] Leoni. *"Tool calibration advintec TCP"* (s.d) Disponível online em: www.leoni-industrial-solutions.com. Acedido em Maio de 2015.

[11] Dynalog. *"Precision Robotic Applications".* (s.d) Disponível online em: www.dynalog-us.com. Acedido em Junho de 2015.

[12] Abreu, Paulo. (2013). *"Manual de Utilização do RobotStudio Parte 1 – Introdução".* Porto: Faculdade de Engenharia da Universidade do Porto.

[13] ABB Developer Center. *"Programming and testing – Tools Defining the tool frame"* (s.d). Disponível online em: www.developercenter.robotstudio.com. Acedido em Março de 2015.

[14] Gunnarsson, Kristjan and Hemmerle, James. (1998). *"Tool center point calibration apparatus and method"*. US 5929584.

[15] Thorne, Henry F.(1996). *"Tool center point calibration for spot welding guns".* US 5910719.

[16] Hara, Ryuichi. (1993). *"Tool center point setting method in a robot"*. US 5194792.

[17] Brantmark, Håkan. (1997). *"The system module utility"*

[18] Gan, Zhongxue and Tang, Qing. (2011). "*Visual Sensing and its Applications".* Tsinghua University, Beijing.

[19] ABB Robotics. (2012). *"Application manual – Navigator".* Sweden.

[20] ABB Robotics. (2004). *"User's Guide – BullsEye".* Sweden.

[21] Lopes, António Mendes. (2011). *"Algoritmo de Denavit-Hartenberg (D-H) – Aulas práticas".* Porto: Faculdade de Engenharia da Universidade do Porto.

[22] Watfa, Simon. (1993). "Finding a Sphere From Four Points". Disponível online em: www.steve.hollasch.net. Acedido em Abril de 2015.

[23] CATIADOC. *"Edit a Tool Assembly in the Resource List*". (s.d). Disponível online em: www.catiadoc.free.fr/online/mfgug_C2/mfgugbt0211.htm. Acedido em Março de 2015.

Anexos

Anexo 1 – Programa em RAPID para aquisição e conversão dos pontos de calibração

Através do RAPID, o programa invoca inicialmente uma instrução em que o controlador virtual lê os valores das juntas do manipulador e aplica um algoritmo de cinemática direta. Isto permite obter a posição e orientação do referencial Tool0, posicionado na flange ou elemento terminal do robô, em relação ao referencial de base.

O formato em que a informação fica guardada é o seguinte:

Ponto_1:=[[905.8,-12.6,454.9],[0.092,0.671,-0.691,-0.251],

[-1,2,4,0],[0,9E+09,9E+09,9E+09,9E+09,9E+09]];

O primeiro vetor contém a posição da origem do referencial Tool0 em relação ao referencial de base. O segundo contém a orientação expressa em quatérnios. O terceiro está associado à configuração assumida pelo robô para atingir essa posição com essa orientação. O quarto diz respeito a eixos externos ao robô caso existam. Estes dois últimos são irrelevantes para o cálculo do TCP.

Para transformar o vetor de ângulos de Euler na matriz de rotação que relaciona os referenciais é necessário entender que:

Os ângulos de Euler são um conjunto de três ângulos que representam a orientação de um referencial como uma sequência de rotações de eixos partindo de uma posição inicial. Admita-se um referencial 0 numa dada posição. Roda-se agora esse referencial em torno do eixo x de X0 obtendo-se o referencial 1. Seguidamente, roda-se o referencial 1 em torno do seu eixo y de Y1 e obtém-se o referencial 2. Finalmente roda-se o referencial 2 em torno do seu eixo z de Z2 e obtém-se o referencial 3. Os ângulos de Euler representam a orientação do referencial 3 expressa no referencial 0.

Convencionalmente a sequência de rotações é X0, Y1, Z2, contudo, os robôs da ABB usam a sequência Z0, Y1, X2.

Considerando que os ângulos de Euler são (ϕ, θ, ψ).

A representação da orientação em termos de ângulos de Euler corresponde à seguinte sequência de rotações:

- 1º - Rotação de um ângulo ψ segundo o eixo *OZ* ($\mathbf{R}_{z,\psi}$);
- 2º - Rotação de um ângulo θ segundo o eixo rodado *OY*, isto é, *OY1* ($\mathbf{R}_{y,\theta}$);
- 3º - Rotação de um ângulo ϕ segundo o eixo rodado *OX*, isto é, *OX2*($\mathbf{R}_{x,\phi}$).

A matriz de rotação equivalente vem dada por:

$$\mathrm{R}_{\psi,\theta,\phi} = \mathrm{R}_{z,\psi}\mathrm{R}_{y,\theta}\mathrm{R}_{x,\phi} = \begin{bmatrix} C\psi & -S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & -S\phi \\ 0 & S\phi & C\phi \end{bmatrix}$$

$$= \begin{bmatrix} C\psi C\theta & C\psi S\theta S\phi - S\psi C\phi & C\psi S\theta C\phi + S\psi S\phi \\ S\psi C\theta & S\psi S\theta S\phi + C\psi C\phi & S\psi S\theta C\phi - C\psi S\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix}$$

```
VAR iodev file;
VAR robtarget ferramentas;

VAR robtarget pose_i;
VAR pos pos_i{100};
VAR pos rot_i{100};
VAR pos ferramenta_pos;

VAR num angle_x;
VAR num angle_y;
VAR num angle_z;

VAR num n_pontos:=8;
TASK PERS tooldata tool2:=[TRUE,[[0,0,0],[1,0,0,0]],[-1,[0,0,0],[1,0,0,0],0,0,0]];

PROC Main()
   ferramentas:=CRobT(\Tool:=AW_Gun\WObj:=wobj0);
   ferramenta_pos:=ferramentas.trans;

   for i from 1 to n_pontos DO

      pose_i:=CRobT(\Tool:=tool0\WObj:=wobj0);
      pos_i{i}:=pose_i.trans;
      angle_x:=EulerZYX(\X,pose_i.rot);
      angle_y:=EulerZYX(\Y,pose_i.rot);
      angle_z:=EulerZYX(\Z,pose_i.rot);
      rot_i{i}:=[angle_x,angle_y,angle_z];
      Stop;

   ENDFOR
```

```
    open "home:"\file:="file1.txt",file;
    write file,"Referência: "\pos:=ferramenta_pos;
    Write file,"";
    Write file,"";

    for i from 1 to n_pontos DO

      write file,"P"\Num:=i;
      write file,"="\pos:=pos_i{i};
      write file,";";
      write file,"O"\Num:=i;
      Write file,"="\pos:=rot_i{i};
      Write file,";";

    ENDFOR

    close file;

  ENDPROC

ENDMODULE
```

Anexo 2 – Programa gerado em Matlab para comparação das metodologias de minimização do erro e cálculo da superfície esférica

Neste anexo encontra-se o programa desenvolvido em Matlab para processar os dados obtidos a partir do RobotStudio segundo as duas metodologias.

2.1 Definição de TCP, ponto fixo e pontos de calibração

```
Ponto fixo = [960.948,0,832.719];
TCP = [126.925343398,0,389.787472308];

Pontos de calibração em posição relativa próxima
(Posição expressa no referencial de base do robô e orientação expressa em
ângulos de Euler)

P1=[797.276,-48.99,1205.35];
O1=[160.838,35.9215,148.171];
P2=[855.034,-48.99,1225.69];
O2=[166.003,28.4773,150.93];
P3=[855.034,-111.348,1212.76];
O3=[157.2,23.7148,147.049];
P4=[798.163,-111.348,1192.09];
O4=[151.632,30.9935,144.483];
P5=[798.163,-17.4845,1208.54];
O5=[166.796,38.5755,153.188];
P6=[905.54,-17.4845,1238.51];
O6=[174.469,24.3486,157.228];
P7=[960.097,48.1835,1239.81];
O7=[-173.266,20.3026,162.531];
P8=[960.097,-54.3812,1239.03];
O8=[172.513,15.4398,158.143];

Pontos de calibração em posição relativa afastada

P1=[970.698,-65.3541,1237.29];
O1=[170.523,16.0428,174.711];
P2=[970.698,-332.418,1072.4];
O2=[125.227,7.62144,164.904];
P3=[635.448,-68.1679,1072.4];
O3=[125.227,7.6215,85.0527];
P4=[1217.27,-308.259,918.283];
O4=[-153.961,-56.7997,104.479];
P5=[685.737,-174.11,583.743];
O5=[55.9513,-20.6429,114.938];
P6=[727.617,227.184,583.743];
O6=[55.9513,-20.643,38.3837];
P7=[1243.1,35.4847,1127.98];
O7=[140.59,-2.53176,-112.349];
P8=[554.137,35.4847,868.642];
O8=[-89.1869,18.9201,-77.6223];
```

2.2 Programa desenvolvido

```
clear;
clc;
close all;

%Valores teóricos de referência

ref_wobj0=[960.9480,0,832.719];
ref_tool0=[126.925343398,0,389.787472308];
ref_raio=sqrt(ref_tool0(1,1)^2+ref_tool0(1,2)^2+ref_tool0(1,3)^2);

%Posição e orientação dos pontos de calibração

%Próximos
P1=[797.276,-48.99,1205.35];
O1=[160.838,35.9215,148.171];
P2=[855.034,-48.99,1225.69];
O2=[166.003,28.4773,150.93];
P3=[855.034,-111.348,1212.76];
O3=[157.2,23.7148,147.049];
P4=[798.163,-111.348,1192.09];
O4=[151.632,30.9935,144.483];
P5=[798.163,-17.4845,1208.54];
O5=[166.796,38.5755,153.188];
P6=[905.54,-17.4845,1238.51];
O6=[174.469,24.3486,157.228];
P7=[960.097,48.1835,1239.81];
O7=[-173.266,20.3026,162.531];
P8=[960.097,-54.3812,1239.03];
O8=[172.513,15.4398,158.143];

%Afastados
P1=[970.698,-65.3541,1237.29];
O1=[170.523,16.0428,174.711];
P2=[970.698,-332.418,1072.4];
O2=[125.227,7.62144,164.904];
P3=[635.448,-68.1679,1072.4];
O3=[125.227,7.6215,85.0527];
P4=[1217.27,-308.259,918.283];
O4=[-153.961,-56.7997,104.479];
P5=[685.737,-174.11,583.743];
O5=[55.9513,-20.6429,114.938];
P6=[727.617,227.184,583.743];
O6=[55.9513,-20.643,38.3837];
P7=[1243.1,35.4847,1127.98];
O7=[140.59,-2.53176,-112.349];
P8=[554.137,35.4847,868.642];
O8=[-89.1869,18.9201,-77.6223];
```

```
%Amplitude da perturbação
min=-3;
max=3;

%Divisor de amplitude da perturbação
d=10;

%Número de iterações
np=10000;

%Sin Cos

sth1=sin(O1(1,2)*pi/180);
cth1=cos(O1(1,2)*pi/180);
sth2=sin(O2(1,2)*pi/180);
cth2=cos(O2(1,2)*pi/180);
sth3=sin(O3(1,2)*pi/180);
cth3=cos(O3(1,2)*pi/180);
sth4=sin(O4(1,2)*pi/180);
cth4=cos(O4(1,2)*pi/180);

sfi1=sin(O1(1,1)*pi/180);
cfi1=cos(O1(1,1)*pi/180);
sfi2=sin(O2(1,1)*pi/180);
cfi2=cos(O2(1,1)*pi/180);
sfi3=sin(O3(1,1)*pi/180);
cfi3=cos(O3(1,1)*pi/180);
sfi4=sin(O4(1,1)*pi/180);
cfi4=cos(O4(1,1)*pi/180);

scs1=sin(O1(1,3)*pi/180);
ccs1=cos(O1(1,3)*pi/180);
scs2=sin(O2(1,3)*pi/180);
ccs2=cos(O2(1,3)*pi/180);
scs3=sin(O3(1,3)*pi/180);
ccs3=cos(O3(1,3)*pi/180);
scs4=sin(O4(1,3)*pi/180);
ccs4=cos(O4(1,3)*pi/180);

%Matrizes de Rotação

R1=[ccs1*cth1,ccs1*sth1*sfi1-scs1*cfi1,ccs1*sth1*cfi1+scs1*sfi1
    scs1*cth1,scs1*sth1*sfi1+ccs1*cfi1,scs1*sth1*cfi1-ccs1*sfi1
    -sth1, cth1*sfi1, cth1*cfi1];
R2=[ccs2*cth2,ccs2*sth2*sfi2-scs2*cfi2,ccs2*sth2*cfi2+scs2*sfi2
    scs2*cth2,scs2*sth2*sfi2+ccs2*cfi2,scs2*sth2*cfi2-ccs2*sfi2
    -sth2, cth2*sfi2, cth2*cfi2];
R3=[ccs3*cth3,ccs3*sth3*sfi3-scs3*cfi3,ccs3*sth3*cfi3+scs3*sfi3
    scs3*cth3,scs3*sth3*sfi3+ccs3*cfi3,scs3*sth3*cfi3-ccs3*sfi3
    -sth3, cth3*sfi3, cth3*cfi3];
R4=[ccs4*cth4,ccs4*sth4*sfi4-scs4*cfi4,ccs4*sth4*cfi4+scs4*sfi4
    scs4*cth4,scs4*sth4*sfi4+ccs4*cfi4,scs4*sth4*cfi4-ccs4*sfi4
    -sth4, cth4*sfi4, cth4*cfi4];

%Vetores de Variação

deltaR21=R2-R1;
deltaR31=R3-R1;
deltaR41=R4-R1;
```

```
%Matriz de transformação homogénea

R=R3;
P=P3;
T=[R,P';0,0,0,1];

for i=1:np,

 %Perturbação dos pontos de calibração

 P1_i=[(P1(1,1)+((max-min)*rand + min)/d),(P1(1,2)+((max-min)*rand +
min)/d),(P1(1,3)+((max-min)*rand + min)/d)]';
 P2_i=[(P2(1,1)+((max-min)*rand + min)/d),(P2(1,2)+((max-min)*rand +
min)/d),(P2(1,3)+((max-min)*rand + min)/d)]';
 P3_i=[(P3(1,1)+((max-min)*rand + min)/d),(P3(1,2)+((max-min)*rand +
min)/d),(P3(1,3)+((max-min)*rand + min)/d)]';
 P4_i=[(P4(1,1)+((max-min)*rand + min)/d),(P4(1,2)+((max-min)*rand +
min)/d),(P4(1,3)+((max-min)*rand + min)/d)]';

 %Segundo Minimização do Erro

 deltaP21=P2_i-P1_i;
 deltaP31=P3_i-P1_i;
 deltaP41=P4_i-P1_i;

 sum1=deltaR21'*deltaP21+deltaR31'*deltaP31+deltaR41'*deltaP41;
 sum2=deltaR21'*deltaR21+deltaR31'*deltaR31+deltaR41'*deltaR41;

 TCP_tool0_US=-inv(sum2)*sum1;

 raio_US=sqrt(TCP_tool0_US(1,1)^2+TCP_tool0_US(2,1)^2+TCP_tool0_US(3,1)^2);
 r_US(i)=raio_US;
 raio_US_erro=raio_US-ref_raio;
 r_US_erro(i)=raio_US_erro;

 TCP_tool0_US_1=[TCP_tool0_US;1];
 TCP_wobj0_US_1=T*TCP_tool0_US_1;
 TCP_wobj0_US_1(4,:)=[];

 TCP_wobj0_US=TCP_wobj0_US_1;

 erro_wobj0_US=TCP_wobj0_US-ref_wobj0';
 erro_tool0_US=TCP_tool0_US-ref_tool0';

 mat(i,:)=erro_tool0_US;
 %mat(i,:)=erro_wobj0_US;

 end

 x1=mat(:,1);
 y1=mat(:,2);
 z1=mat(:,3);
```

```
%Segundo Cálculo da Superfície Esférica

for i=1:np,

 P1_i=[(P1(1,1)+((max-min)*rand + min)/d),(P1(1,2)+((max-min)*rand +
min)/d),(P1(1,3)+((max-min)*rand + min)/d)];
 P2_i=[(P2(1,1)+((max-min)*rand + min)/d),(P2(1,2)+((max-min)*rand +
min)/d),(P2(1,3)+((max-min)*rand + min)/d)];
 P3_i=[(P3(1,1)+((max-min)*rand + min)/d),(P3(1,2)+((max-min)*rand +
min)/d),(P3(1,3)+((max-min)*rand + min)/d)];
 P4_i=[(P4(1,1)+((max-min)*rand + min)/d),(P4(1,2)+((max-min)*rand +
min)/d),(P4(1,3)+((max-min)*rand + min)/d)];

 x=[P1_i
     P2_i
     P3_i
     P4_i];

 z = x(:,3);
 y = x(:,2);
 x = x(:,1);

 % solve linear system of normal equations
 A = [x y z ones(size(x))];
 b = -(x.^2 + y.^2 + z.^2);
 a = A \ b;
 % A\B is roughly the same as inv(A)*B, except it is computed in a
different way.
 % If A is an n-by-n matrix and B is a column vector with n components,
then X = A\B is the solution
 % to the equation AX = B computed by Gaussian elimination.

 TCP_wobj0_SE = -a(1:3)./2;

 raio_SE = sqrt(sum(TCP_wobj0_SE.^2)-a(4));
 r_SE(i)=raio_SE;
 raio_SE_erro=raio_SE-ref_raio;
 r_SE_erro(i)=raio_SE_erro;

 erro_wobj0_SE=TCP_wobj0_SE-ref_wobj0';

 %Tinv=inv(T);
 TCP_wobj0_SE=[TCP_wobj0_SE;1];
 TCP_tool0_SE=T\TCP_wobj0_SE;
 TCP_tool0_SE(4,:)=[];

 erro_tool0_SE=TCP_tool0_SE-ref_tool0';

matriz(i,:)=erro_tool0_SE;
%matriz(i,:)=erro_wobj0_SE;
end

z2=matriz(:,3);
y2=matriz(:,2);
x2=matriz(:,1);
```

```
%Apresentação dos resultados

vermelho = [255,0,0]/255;
azul = [0,0,255]/255;

%Histograma Erro (raio)
histfit(r_SE_erro,100);
hold on;
histfit(r_US_erro,100);
hLines = findobj('Type','Line');
set(hLines(1),'Color','k');
set(hLines(2),'Color','k');
g=findobj(gca,'Type','patch');
set(g(1),'FaceColor','b','EdgeColor','b');
set(g(2),'FaceColor','r','EdgeColor','r');
set(gca,'Fontsize',12,'Fontname','euclid');
title('Histograma');
xlabel('Erro Raio [mm]');

%Gráfico Erro (raio)
 plot(r_US_erro);
 hold on;
 plot(r_SE_erro);
 hLines = findobj('Type','Line');
 set(hLines(1),'Color','b');
 set(hLines(2),'Color','r');
 title('Erro raio [mm]');
 xlabel('Número de pontos');
 ylabel('Erro');

%Gráfico Erro (TCP 3D ou 2D)
 plot3(x1,y1,z1,'.','color',vermelho);
 hold on;
 plot3(x2,y2,z2,'.','color',azul);
 axis equal;
 grid;
 title('Erro TCP');
 xlabel('X  [mm]');
 ylabel('Y  [mm]');
 zlabel('Z  [mm]');
 view(0,0);

 set(gcf,'Color',[1,1,1]);
```

Anexo 3 – Síntese dos tipos de variáveis e instruções mais relevantes para o trabalho de programação desenvolvido

Apresentam-se neste anexo alguns tipos de variáveis e instruções usadas na programação dos procedimentos de calibração com referência à sua função.

num - número
bool – variável lógica (booleana)
pos – vetor de posição
orient – quaternião (indica orientação)
pose – constituída por um vetor e uma orientação
robtarget - define uma localização a atingir pelo robô
wobjdata – define um referencial
tool - define uma ferramenta
Confdata – define a configuração do robô
loaddata - Load data

2.21 CJointT - Lê o valor dos ângulos das juntas
2.26 CRobT - Lê o valor da posição do TCP ativo
2.34 DefFrame - Define um referencial
2.40 EulerZYX – Transforma um vetor de quatérnios num de ângulos de Euler
2.78 Offs – Faz um *offcet* a um target
2.103 RelTool – Gera um target que resulta de um *offcet* da posição da ferramenta
2.152 VectMagn – Devolve a norma de um vetor

1.4 AliasIO – Associa uma variável digital externa a uma interna
1.95 MoveJ – Move o robô segundo uma interpolação no espaço de juntas
1.98 MoveL - Move o robô segundo uma interpolação linear
1.102 MToolTCPCalib – Aplica o algoritmo de cálculo do TCP
1.148 SearchL – Executa um movimento linear de busca
1.183 Stop – Para a execução do programa
1.200 TPErase – Apaga a informação escrita na consola
1.202 TPReadFK – Lê um botão da consola
1.205 TPWrite – Escreve na consola

8.1.1 FCAct – Ativa o controlo de força
8.1.2 FCCalib – Calibra o sensor de força
8.1.10 FCDeact – Desativa o sensor de força
8.2.4 FCLoadID – Identifica a carga que o sensor sente

Anexo 4 – Características do controlador virtual utilizado no RobotStudio

Neste anexo encontram-se as características do controlador virtual usado para a simulação dos procedimentos de calibração no RobotStudio. Este controlador virtual possui as características do controlador real.

System Name: Sislab_out2012 Serial Number: 24-53211 Using Media: Media: Name: RobotWare Version: 5.12.3023.03 Media: Name: 3HEA-024-53211 Version: 1.00.1330.01 Media: Name: RWMachiningFCGUI Version: 2.03.0000.00 Control Module: Key: sfRXNNOXEolDFVNNOPNlQe Key Signature: 109 Additional Languages: 644-6 Portuguese 645-1 French Options: RW Control module key RobotWare OS and English 644-6 Portuguese - 645-1 French 709-x DeviceNet - 709-1 Master/Slave Single 608-1 World Zones 609-1 Fixed Position Events 617-1 FlexPendant Interface 618-1 Fieldbus Command Interface 621-1 Logical Cross Connections 626-1 Advanced RAPID 877-1 Machining FC 661-2 Assembly FC	Drive Module: Key: Hgw636wCdmm7G5 Key Signature: 106 Manipulator: ABB standard manipulator Add axes IRB/drive module 1600/2400/260 IRB 2400/16 Type B Options: RW Drive module 1 key ABB standard manipulator - IRB 2400/16 Type B - 2400/16 Type B Add axes IRB/drive module 1600/2400/260 - R2C4 Add drive 3 x U or prepared - 753-2 Drive T in pos Z4 - No drive in pos Y4 Additional Options 3HEA-024-53211 Key: X7txYKas9xNHAorv-n.xvt8 Options: 3HEA-024-53211.01 RWMACHININGFCGUI Key: b8a9.HJUwBPssOev8v7N6NRuR Options: RWMachiningFCGUI

Anexo 5 - Programas gerados em RAPID para simulação e implementação dos procedimentos de calibração do TCP

Apresentam-se neste anexo pequenas porções do código, gerado para a simulação e teste dos procedimentos de calibração, com referências às funcionalidades que implementam.

Anexo 5.1 - Programação do procedimento PNT_1

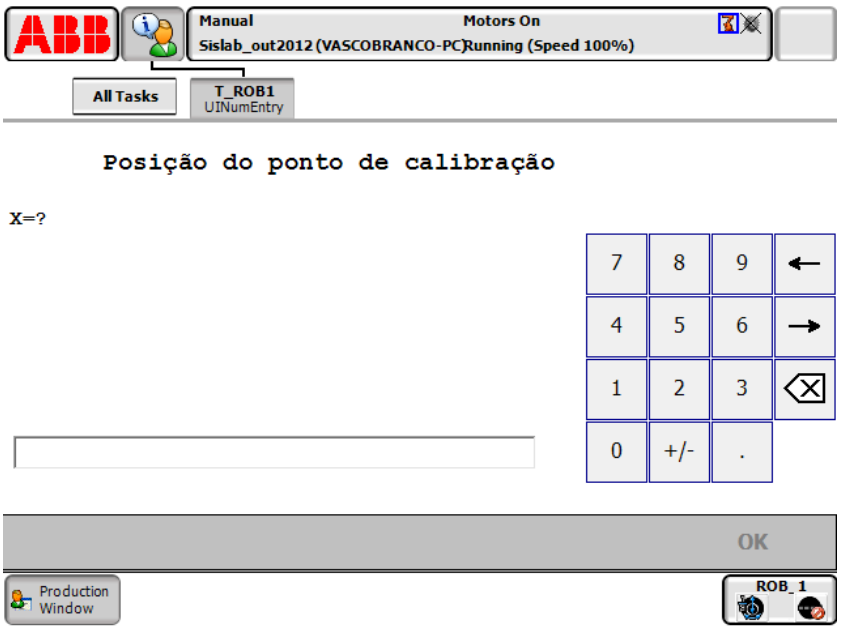
Antes de lançar o programa, o utilizador tem que conhecer a localização do ponto de calibração. O reconhecimento desta localização é feito com uma ferramenta pré calibrada de acordo com um outro procedimento de calibração. Depois move a ferramenta a calibrar para o ponto de calibração fazendo *jogging* do robô. Com a ferramenta no ponto de calibração, executa o programa que calcula a posição do TCP no referencial Tool0.

```
MODULE PNT_1
    VAR pos TCP;
    VAR robtarget T0;
    VAR pose T0_pose;
    VAR pos T0_pos;
    VAR orient T0_orient;
    VAR pose T0_l;
    VAR num x;
    VAR num Y;
    VAR num z;
```

Declaração de variáveis

```
PROC Main()
    x:=UINumEntry(\Header:="Posição do ponto de calibração"\Message:="X=?");
    y:=UINumEntry(\Header:="Posição do ponto de calibração"\Message:="Y=?");
    z:=UINumEntry(\Header:="Posição do ponto de calibração"\Message:="Z=?");
```

O utilizador é questionado sobre as coordenadas de posição do ponto de calibração



```
        T0:=Crobt(\Tool:=tool0\WObj:=wobj0);
        T0_pos:=T0.trans;
        T0_orient:=T0.rot;
```

Aquisição da posição e orientação de Tool0

```
        Mtrans:=[T0_pos,T0_orient];
        Inv_Mtrans:=PoseInv(Mtrans);
```

Matriz de transformação e sua inversa

```
        TCP:=PoseVect(Inv_Mtrans,[x,y,z]);
        TPErase;
        TPWrite "TCP = "\Pos:=TCP;
ENDPROC

ENDMODULE
```

O TCP é dado pelo produto da inversa da matriz de transformação pelo ponto de calibração tal como o exposto na Secção 3.2.1

Não tem um procedimento de configuração associado pelo que o utilizador deve lançar o programa e seguir as instruções que vão sendo dadas.

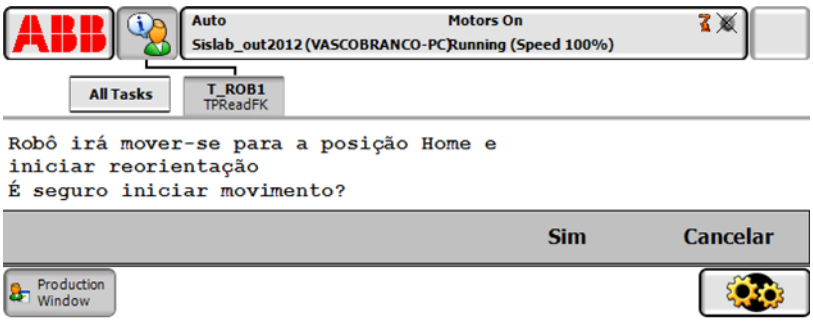
Anexo 5.2 - Programação do procedimento PLN_1

```
MODULE PLN_1
        CONST jointtarget Home:=[[0,0,0,0,30,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        VAR num lp;
        (…)
        VAR num vx;
        VAR pos TCP;

PROC Main()
        TPErase;
        TPWrite "Robô irá mover-se para Home e iniciar reorientação";
        TPReadFK r_ZDR,"É seguro iniciar movimento?","","","","","Sim","Cancelar";
```

Declaração de variáveis

O utilizador é questionado a autorizar a movimentação automática do robô




```
        IF r_ZDR=5 THEN
        ExitCycle;
        ENDIF
```

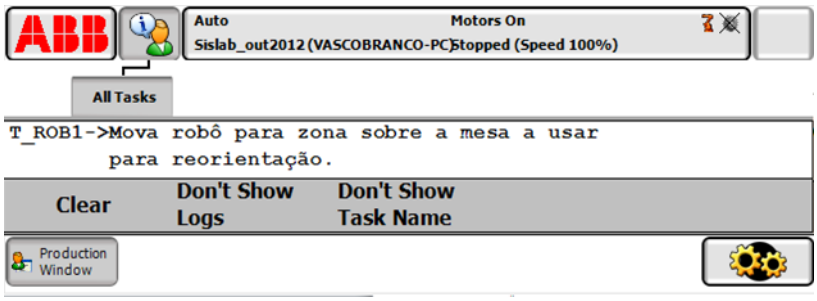
Quebra o ciclo e volta ao início do programa

```
IF r_ZDR=4 THEN
ConfJ\Off;
MoveabsJ Home,v10,fine,tool0\WObj:=wobj0;
orientar_junta6:=CJointT();
orientar_junta6.robax.rax_6:=90;
MoveAbsJ orientar_junta6,v10,Fine,Tool0\WObj:=wobj0;
orientar_junta5:=CJointT();
orientar_junta5.robax.rax_5:=0;
MoveAbsJ orientar_junta5,v10,Fine,Tool0\WObj:=wobj0;
ENDIF
```

Se for dada a autorização, o robô move-se para uma posição que facilita a aquisição de pontos sobre o plano

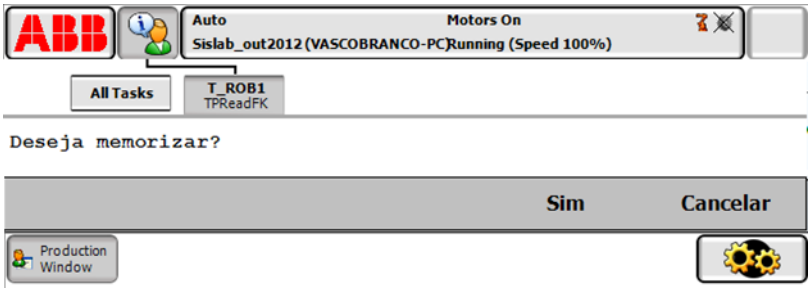
```
linha_132:
TPErase;
TPWrite "Mova robô para zona sobre a mesa a usar para reorientação.";
Stop;
```

O utilizador é informado de que deve mover manualmente o robô para um ponto de aproximação ao plano de calibração (mesa), que será usado para posterior reorientação da ferramenta




```
TPErase;
TPReadFK r_ip,"Deseja memorizar?","","","","Sim","Cancelar";
```

O utilizador tem que confirmar a memorização desse ponto




```
IF r_ip=5 THEN
ExitCycle;
ENDIF

IF r_ip=4 THEN
TPErase;
zx_i:=Crobt(\Tool:=tool0\WObj:=wobj0);
zx_i_pos:=zx_i.trans;
zx_i_z:=zx_i_pos.z;
ENDIF
```

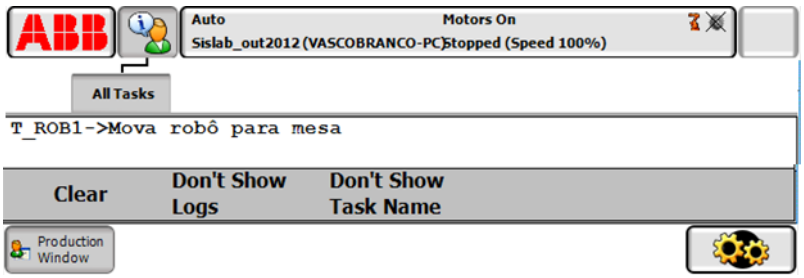
Grava a posição de Tool0 no referencial de base e recolhe a coordenada z

```
!guardar primeiro ponto sobre o plano

linha_247:
TPErase;
TPWrite "Mova robô para mesa";
Stop;
```

O utilizador é informado de que deve mover manualmente o robô até que a ferramenta contacte com a mesa

(…)

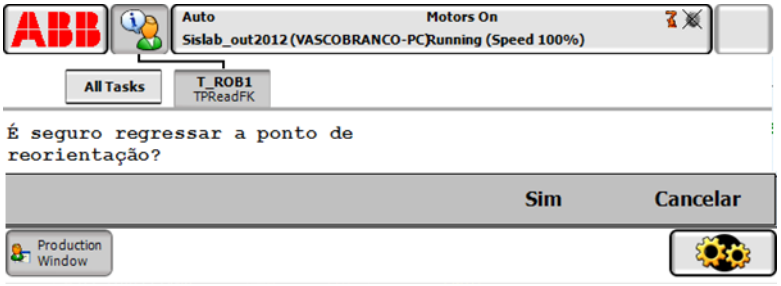


Quando a ferramenta está em contacto com a mesa o utilizador tem que confirmar a memorização desse ponto. O código é semelhante ao anteriormente apresentado para memorização do ponto de reorientação.

```
!regresso a ponto de partida

TPErase;
TPReadFK r_rip,"É seguro regressar a ponto de reorientação?","","","","Sim","Cancelar";
```

O utilizador é questionado a autorizar a movimentação automática do robô para o ponto de reorientação




```
IF r_rip=5 THEN
ExitCycle;
ENDIF

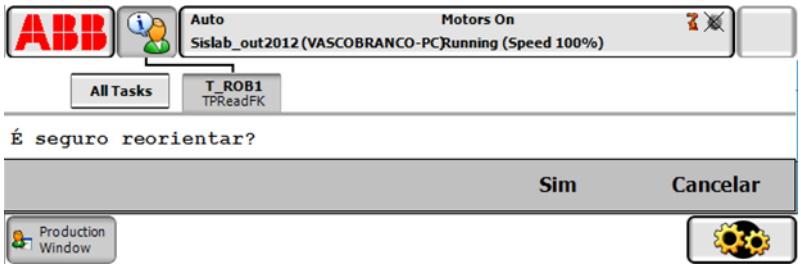
IF r_rip=4 THEN
MoveJ zx_i,v10,fine,tool0\WObj:=wobj0;
ENDIF
```

→ Robô inicia a movimentação automática para o ponto de reorientação

```
!rodar tool0 em torno de ponto de partida positivo

TPErase;
TPReadFK r_reorp,"É seguro reorientar?","","","","Sim","Cancelar";
```

O utilizador é questionado a autorizar a reorientação automática do robô




```
IF r_reorp=5 THEN
ExitCycle;
ENDIF

IF r_reorp=4 THEN
r_zx_p:=RelTool(zx_i,0,0,0,\Rx:=-10,\Ry:=0,\Rz:=0);
MoveJ r_zx_p,v10,fine,tool0\WObj:=wobj0;
ENDIF

(…)
```

→ É gerado um target baseado na posição e orientação de Tool0 e rodado de 10 graus em torno do seu eixo x. O move-se para esse target, o que se traduz na reorientação automática.

A execução do programa continua com a memorização de mais pontos sobre a mesa, sempre seguidos de regresso ao ponto de partida e reorientação da ferramenta rodando o punho do robô em torno de Tool0. O código associado é semelhante ao anteriormente apresentado.

```
l0:=zx_i_z-p_zx_i_z;

lp_zx:=zx_i_z-p_zx_n_z;
ln_zx:=zx_i_z-p_zx_p_z;
lp_zy:=zx_i_z-p_zy_n_z;
ln_zy:=zx_i_z-p_zy_p_z;

vx_zy:=-(lp_zy-ln_zy)/(2*sin(10));
!X
vy_zy:=-(lp_zy+ln_zy-2*l0)/(2-2*Cos(10));
!Z
vx_zx:=-(lp_zx-ln_zx)/(2*sin(10));
!Y
vy_zx:=-(lp_zx+ln_zx-2*l0)/(2-2*Cos(10));
!Z
TCP:=[vx_zy,vy_zy,vx_zx];
TPWrite "TCP="\Pos:=TCP;

ENDPROC

ENDMODULE
```

Com as coordenadas dos pontos memorizados são calculadas as cotas L0, LP e LN

Segue-se o cálculo das coordenadas do TCP com a aplicação das equações apresentadas na Secção 3.2.2.

Anexo 5.3 - Programação do procedimento PLN_2

```
MODULE PLN_2

VAR robtarget Plano_C_tool0;
VAR robtarget Plano_X_tool0;
(…)
VAR tooldata sphere;
VAR num esp:=10;

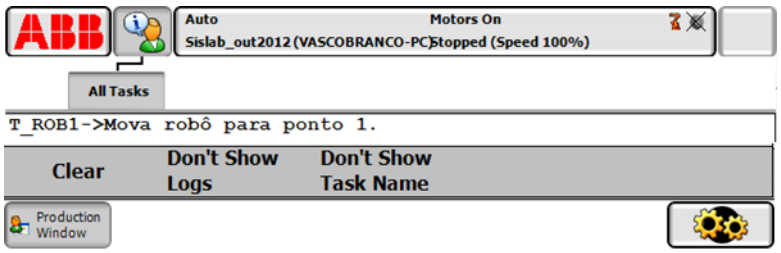
PROC Main()

!localização do plano

linha_93:
TPErase;
TPWrite "Mova robô para ponto 1.";
Stop;
```

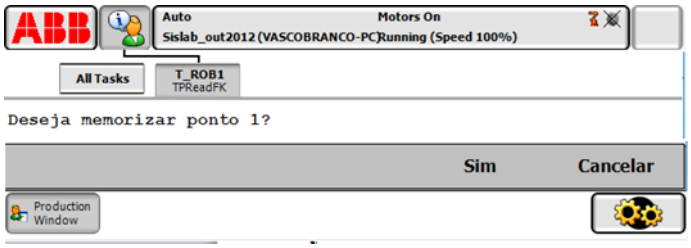
Declaração de variáveis

O utilizador é informado de que deve mover manualmente o robô para o ponto 1, que é um ponto assinalado no plano e usado para o reconhecimento da orientação do mesmo




```
TPErase;
TPReadFK r_o,"Deseja memorizar ponto 1?","","","","Sim","Cancelar";
```

O utilizador é questionado a autorizar a memorização do ponto




```
IF r_o=5 THEN
ExitCycle;
ENDIF

IF r_o=4 THEN
TPErase;
Plano_C_tool0:=Crobt(\Tool:=tool0\WObj:=wobj0);
Plano_C_tool0_pos:=Plano_C_tool0.trans;
TPWrite "Origem = "\Pos:=Plano_C_tool0_pos;
TPReadFK r_go,"","","","","Guardar","Editar","Cancelar";
TPErase;
```

A posição de Tool0 no referencial de base é adquirida e o utilizador pode visualizar o seu valor




```
IF r_go=5 THEN
ExitCycle;
ENDIF

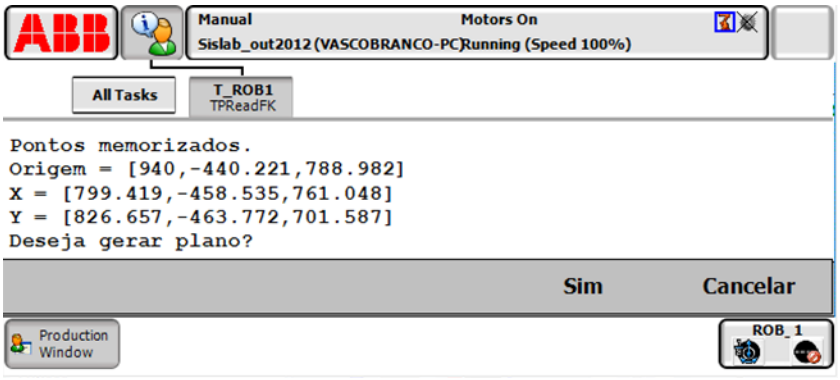
IF r_go=4 THEN
GOTO linha_93;
ENDIF

IF r_go=3 THEN
linha_123:
TPErase;
TPWrite "Mova robô para ponto 2.";
Stop;
(…)
```

Os passos enumerados repetem-se para a aquisição de mais dois pontos sobre o plano, necessários à determinação de um referencial contendo a orientação do plano.

```
TPErase;
TPWrite "Pontos memorizados.";
TPWrite "Origem = "\Pos:=Plano_C_tool0_pos;
TPWrite "X = "\Pos:=Plano_X_tool0_pos;
TPWrite "Y = "\Pos:=Plano_Y_tool0_pos;
TPWrite "Deseja gerar plano?";
TPReadFK r_gP,"","","","","Sim","Cancelar";
```

Ao guardar o terceiro ponto o utilizador pode visualizar os três pontos adquiridos e optar por gerar o plano ou repetir todo o procedimento




```
IF r_gP=5 THEN
ExitCycle;
ENDIF

IF r_gP=4 THEN
frame6:=DefFrame(Plano_C_tool0,Plano_X_tool0,Plano_Y_tool0);
frame6_pos:=frame6.trans;
frame6_orient:=frame6.rot;
frame6_wobj:=[false,true,"",frame6,pp];
```

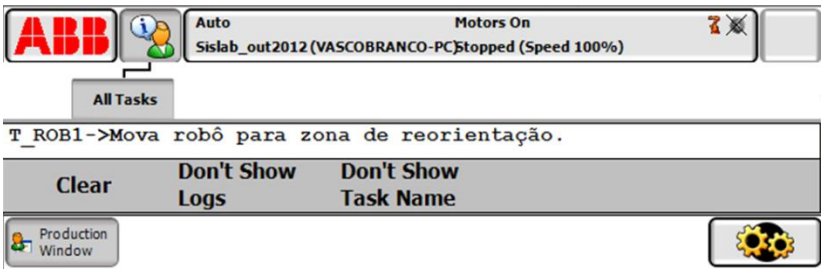
Gera um referencial com os pontos adquiridos sobre o plano

```
                                        ENDIF
                                    ENDIF
                                ENDIF
                            ENDIF
                        ENDIF
                    ENDIF
ENDIF

!orientar z de tool0 perpendicularmente ao plano

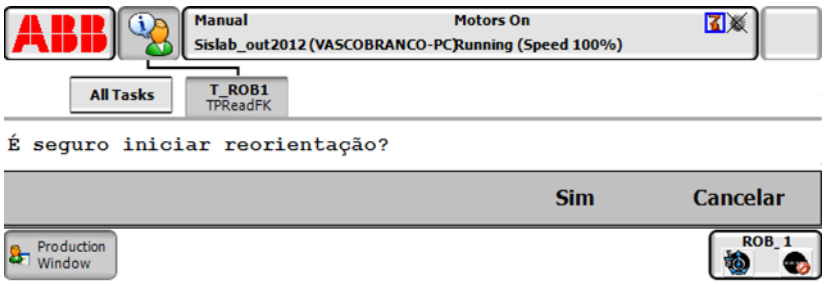
TPErase;
TPWrite "Mova robô para zona de reorientação.";
Stop;
```

O utilizador é informado de que deve movimentar o robô para uma zona que permita a sua reorientação automática



```
TPErase;
TPReadFK r_ZDR,"É seguro iniciar reorientação?","","","","Sim","Cancelar";
```

Quando o robô já está na zona de reorientação, o utilizador tem que autorizar a sua movimentação




```
IF r_ZDR=5 THEN
ExitCycle;
ENDIF

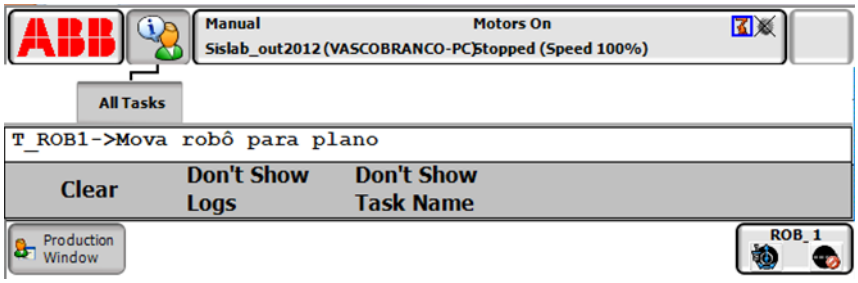
IF r_ZDR=4 THEN
ZDR:=Crobt(\Tool:=tool0\WObj:=wobj0);
ZDR_pos:=ZDR.trans;
ConfJ\Off;
face_ZD:=[ZDR_pos,frame6_orient,[1,0,0,0],[0,0,0,0,0,0]];
MoveJ face_ZD,v10,fine,tool0\WObj:=wobj0;
ENDIF
```

Gera um target cuja posição é a posição do referencial Tool0 no ponto de reorientação e a orientação é a do plano. Em seguida o robô move-se para esse target orientando o eixo z de Tool0 perpendicularmente ao plano com os pontos adquiridos sobre o plano

```
!posicionar ferramenta junto ao plano Z

linha_132:
TPErase;
TPWrite "Mova robô para plano";
Stop;
(…)
```

O utilizador é informado de que deve movimentar o robô para o plano



Assim que a ferramenta contacta com o plano segue-se a memorização do ponto. A execução do programa contínua de forma semelhante para a aquisição dos restantes pontos necessários à calibração.

```
medio:=(XD_tool0_pos+XE_tool0_pos);
ref:=[medio.x/2,medio.y/2,medio.z/2];
TPErase;
ENDIF
ENDIF

!calcular TCP
TPErase;

XD_tool0_pos:=XD_tool0.trans;
YD_tool0_pos:=YD_tool0.trans;
ZD_tool0_pos:=ZD_tool0.trans;

D_XD:=XD_tool0_pos.z;
D_YD:=YD_tool0_pos.z;
D_ZD:=ZD_tool0_pos.z;

ref_z:=ref.z;
DZ:=ref_z-D_ZD-(esp+28)/2;
DY:=ref_z-D_YD-(esp+28)/2;
DX:=ref_z-D_XD-(esp+28)/2;

TCP:=[DX,-DY,DZ];

TPErase;
TPWrite "TCP = "\Pos:=TCP;
ENDPROC

ENDMODULE
```

Calcula a posição do plano

Adquire as coordenadas dos pontos na direção ortogonal ao plano

Calcula as coordenadas do TCP por aplicação da Equação 3.31

Anexo 6 – Indicações para contornar mensagens de erro

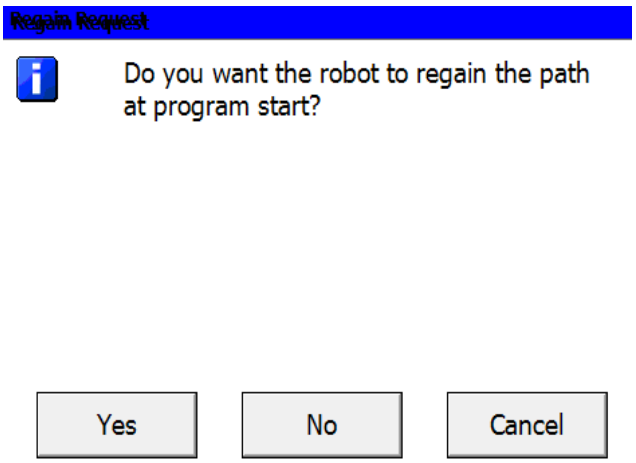
Durante a execução dos programas desenvolvidos, foi necessário usar a instrução *stop* para aceder à movimentação manual do robô. Esta instrução provoca a paragem da execução do programa até que a tecla *start* da consola de programação seja premida, retomando-se aí a execução do programa. Ao retomar o programa surgem mensagens do tipo *Regain Request* que devem ser respondidas de forma cuidadosa. Neste anexo encontram-se indicações para correr o programa e ajudar a responder a cada uma das mensagens.

1º - Correr programa em modo manual e contínuo com *motors on*.

Na figura apresenta-se três configurações do painel principal do controlador. No da esquerda os motores estão desligados e o modo automático está selecionado, no do centro alterou-se o modo para manual, no da direita ligaram-se os motores.



2º - A resposta à mensagem que se segue deve ser *NO*.

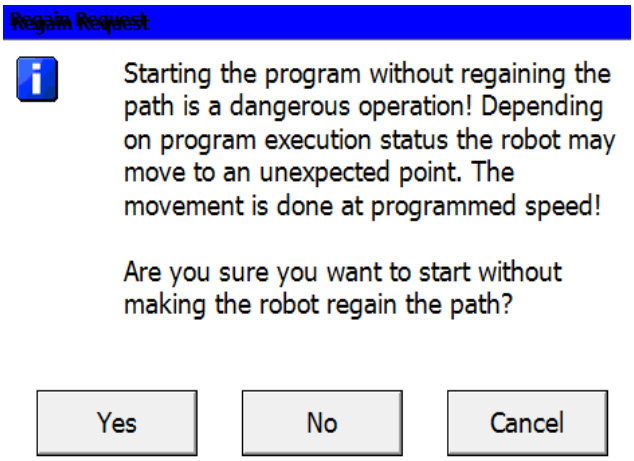


Yes - Robô move-se aparentemente para o último ponto memorizado.

No - Robô não se move e aparece nova mensagem.

Cancel - Não faz nada, se carregar em *start* a mensagem reaparece.

3º - A resposta à mensagem que se segue deve ser *YES*.



Yes - O risco indicado não existe porque o robô não se move.

No - Robô move-se aparentemente para o último ponto memorizado.

Cancel - Não faz nada, se carregar em *start* a mensagem reaparece.

4º - Após as mensagens é sempre necessário carregar novamente *start* a partir de *production window.*

Caso contrário surge a mensagem, *"The courent active view is not a valid test view".*

OK- É única resposta possível. Para retomar o programa há que carregar em *start* partindo de *production window*.

Anexo 7 - Programação em RAPID de sistema de calibração de ferramenta integrando o controlo de força e o procedimento PNT_1

Neste anexo encontra-se a versão integral do código RAPID gerado para a implementação do sistema de calibração integrando o controlo ativo de força para movimentação manual do robô e o procedimento PNT_1.

```
MODULE Ultimatecalib
   VAR robtarget p_calib;
   CONST jointtarget JointTarget_3:=[[-10.766984703,30.855318104,44.032665466,-
27.449546727,78.980537298,-45.997079097],[9E9,9E9,9E9,9E9,9E9,9E9]];
   CONST jointtarget JointTarget_4:=[[-
35.31951267,6.515474971,64.390488249,125.382561235,40.448004604,-
149.722117928],[9E9,9E9,9E9,9E9,9E9,9E9]];
   CONST jointtarget JointTarget_5:=[[-10.748334018,4.021557754,59.203943122,-
116.055582754,30.122223655,87.349684201],[9E9,9E9,9E9,9E9,9E9,9E9]];
   CONST jointtarget JointTarget_6:=[[-
26.622680611,7.325726224,46.14773258,22.223181597,29.659959867,-
40.851043822],[9E9,9E9,9E9,9E9,9E9,9E9]];
   CONST robtarget Target_270:=[[794.725431923,-
247.814714771,694.212064868],[0.501923111,0.763021293,-0.360320126,0.189844947],[-1,-
1,3,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
   CONST robtarget Target_280:=[[782.917532442,-258.264776198,719.93345779],[0.501923032,0.76302113,-
0.360320454,0.189845187],[-1,-1,3,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
   CONST robtarget Target_290:=[[769.363619653,-
268.998555796,743.817175284],[0.501922842,0.763021084,-0.360320654,0.189845498],[-1,-
1,3,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
   VAR num fkey1;
   CONST string but:="But first press Ok";
   CONST string press_ok:="Press Ok to finish";
   VAR robtarget pnt;
   VAR pose pse2:=[[0,0,0],[1,0,0,0]];
   VAR pose pse3:=[[0,0,0],[1,0,0,0]];
   VAR pose pse5:=[[0,0,0],[1,0,0,0]];
   VAR pose JointTarget_3_pose:=[[0,0,0],[1,0,0,0]];
   VAR pose JointTarget_4_pose:=[[0,0,0],[1,0,0,0]];
   VAR pose JointTarget_5_pose:=[[0,0,0],[1,0,0,0]];
   VAR pose JointTarget_6_pose:=[[0,0,0],[1,0,0,0]];
   VAR num DistanceLong:=0;
   VAR num DistanceNew:=0;
   VAR num max_err;
   VAR num mean_err;
   VAR pose psa:=[[0,0,0],[1,0,0,0]];
   VAR pose psb:=[[0,0,0],[1,0,0,0]];
   VAR num d{3,3};
   VAR num q{4};
   VAR num temp;
   PERS tooldata tool2:=[TRUE,[[97.4,0,223],[1,0,0,0]],[5,[23,0,75],[1,0,0,0],0,0,0]];
   PERS loaddata padrao_carga:=[10,[0,0,0.001],[1,0,0,0],0,0,0];
   PERS loaddata calculatedtool_carga:=[10,[0,0,0.001],[1,0,0,0],0,0,0];
   CONST jointtarget Home:=[[0,0,0,0,30,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
   VAR pos TCP;
   VAR robtarget zx_i;
   VAR robtarget p_zx_i;
   VAR robtarget p_zx_p;
   VAR robtarget p_zx_n;
   VAR robtarget p_zy_p;
```

```
VAR robtarget p_zy_n;
VAR robtarget r_zx_p;
VAR robtarget r_zx_n;
VAR robtarget r_zy_p;
VAR robtarget r_zy_n;
VAR jointtarget orientar_junta5;
VAR jointtarget orientar_junta6;
VAR pos zx_i_pos;
VAR num zx_i_z;
VAR pos p_zx_i_pos;
VAR num p_zx_i_z;
VAR pos p_zx_p_pos;
VAR num p_zx_p_z;
VAR pos p_zx_n_pos;
VAR num p_zx_n_z;
VAR pos p_zy_p_pos;
VAR num p_zy_p_z;
VAR pos p_zy_n_pos;
VAR num p_zy_n_z;
VAR num lp_zx;
VAR num ln_zx;
VAR num lp_zy;
VAR num ln_zy;
VAR num l0;
VAR num vx_zx;
VAR num vy_zx;
VAR num vx_zy;
VAR num vy_zy;
VAR num vx;
VAR num vy;
VAR num r_ip;
VAR num r_gip;
VAR num r_pip;
VAR num r_gpip;
VAR num r_rip;
VAR num r_ripp;
VAR num r_reorp;
VAR num r_reorn;
VAR num r_pipp;
VAR num r_gpipp;
VAR num r_pipn;
VAR num r_gpipn;
VAR num r_ZDR;
VAR num r_py;
VAR num r_ny;
VAR num r_reor_py;
VAR num r_reor_ny;
VAR num r_ripn;
VAR num r_rin;
VAR num CP;
VAR robtarget ip;
VAR robtarget pip;
VAR robtarget ipp;
VAR robtarget pipp;
VAR robtarget ipn;
VAR robtarget pipn;
VAR pos ip_pos;
VAR pos pip_pos;
VAR pos ipp_pos;
VAR pos pipp_pos;
VAR pos ipn_pos;
VAR pos pipn_pos;
```

```
VAR num ip_pos_z;
VAR num pip_pos_z;
VAR num ipp_pos_z;
VAR num pipp_pos_z;
VAR num ipn_pos_z;
VAR num pipn_pos_z;

PROC main()
ENDPROC

PROC calib_garra()

   !memorizar ponto de partida

   linha_132:
   TPErase;
   FCCalib padrao_carga;
   FCAct padrao;
   TPErase;
   TPWrite "Mova o robô para zona sobre a mesa";
   TPWrite "a usar para reorientação.";
   TPWrite "Depois, pressione Memorizar para definir";
   TPReadFK r_ip,"essa zona como a zona de reorientação.","","","","","Memorizar","Cancelar";

   IF r_ip=5 THEN
      ExitCycle;
   ENDIF

   IF r_ip=4 THEN
      TPErase;
      zx_i:=Crobt(\Tool:=tool0\WObj:=wobj0);
      zx_i_pos:=zx_i.trans;
      zx_i_z:=zx_i_pos.z;
   ENDIF

   !guardar primeiro ponto sobre o plano

   linha_247:
   TPErase;
   TPWrite "Leve a ferramenta ao bloco de";
   TPWrite "calibração.";
   TPWrite "Depois, pressione Memorizar para gravar";
   TPReadFK r_pip,"o primeiro ponto de calibração?","","","","","Memorizar","Cancelar";

   IF r_pip=5 THEN
      ExitCycle;
   ENDIF

   IF r_pip=4 THEN
      TPErase;
      FCDeact;
      p_zx_i:=Crobt(\Tool:=tool0\WObj:=wobj0);
      p_zx_i_pos:=p_zx_i.trans;
      p_zx_i_z:=p_zx_i_pos.z;

   ENDIF

   !regresso a ponto de partida

   TPErase;
   FCDeact;
   TPWrite "O robô irá mover-se automaticamente para";
```

```
    TPWrite "a zona de reorientação.";
    TPReadFK r_rip,"É seguro iniciar o movimento?","","","","Sim","Cancelar";

    IF r_rip=5 THEN
       ExitCycle;
    ENDIF

    IF r_rip=4 THEN
       MoveJ zx_i,v10,fine,tool0\WObj:=wobj0;
    ENDIF

    !rodar tool0 em torno de ponto de partida positivo

    TPErase;
    TPWrite "O robô irá reorientar a ferramenta";
    TPWrite "automaticamente.";
    TPReadFK r_reorp,"É seguro iniciar a reorientação?","","","","Sim","Cancelar";

    IF r_reorp=5 THEN
       ExitCycle;
    ENDIF

    IF r_reorp=4 THEN
       r_zx_p:=RelTool(zx_i,0,0,0,\Rx:=-10,\Ry:=0,\Rz:=0);
       MoveJ r_zx_p,v10,fine,tool0\WObj:=wobj0;
    ENDIF

    !guardar segundo ponto sobre o plano

    linha_130:
    FCCalib padrao_carga;
    FCAct padrao;
    TPErase;
    TPWrite "Leve a ferramenta ao bloco de";
    TPWrite "calibração.";
    TPWrite "Depois, pressione Memorizar para gravar";
    TPReadFK r_pipp,"o segundo ponto para calibração?","","","","Memorizar","Cancelar";

    IF r_pipp=5 THEN
       ExitCycle;
    ENDIF

    IF r_pipp=4 THEN
       TPErase;
       p_zx_p:=Crobt(\Tool:=tool0\WObj:=wobj0);
       p_zx_p_pos:=p_zx_p.trans;
       p_zx_p_z:=p_zx_p_pos.z;
    ENDIF

    !regresso a ponto de partida

    TPErase;
    FCDeact;
    TPWrite "O robô irá mover-se automaticamente para";
    TPWrite "a zona de reorientação.";
    TPReadFK r_ripp,"É seguro iniciar o movimento?","","","","Sim","Cancelar";

    IF r_ripp=5 THEN
       ExitCycle;
    ENDIF
```

```
IF r_ripp=4 THEN
   MoveJ zx_i,v10,fine,tool0\WObj:=wobj0;
ENDIF

!rodar tool0 em torno de ponto de partida negativo

TPErase;
TPWrite "O robô irá reorientar a ferramenta";
TPWrite "automaticamente.";
TPReadFK r_reorn,"É seguro iniciar a reorientação?","","","","Sim","Cancelar";

IF r_reorn=5 THEN
   ExitCycle;
ENDIF

IF r_reorn=4 THEN
   r_zx_n:=RelTool(zx_i,0,0,0,\Rx:=10,\Ry:=0,\Rz:=0);
   MoveJ r_zx_n,v10,fine,tool0\WObj:=wobj0;
ENDIF

!guardar terceiro ponto sobre o plano

linha_190:
FCCalib padrao_carga;
FCAct padrao;
TPErase;
TPWrite "Leve a ferramenta ao bloco de";
TPWrite "calibração.";
TPWrite "Depois, pressione Memorizar para gravar";
TPReadFK r_pipn,"o terceiro ponto de calibração?","","","","Memorizar","Cancelar";

IF r_pipn=5 THEN
   ExitCycle;
ENDIF

IF r_pipn=4 THEN
   TPErase;
   p_zx_n:=Crobt(\Tool:=tool0\WObj:=wobj0);
   p_zx_n_pos:=p_zx_n.trans;
   p_zx_n_z:=p_zx_n_pos.z;
ENDIF

!regresso a ponto de partida

TPErase;
FCDeact;
TPWrite "O robô irá mover-se automaticamente para";
TPWrite "a zona de reorientação.";
TPReadFK r_rin,"É seguro iniciar o movimento?","","","","Sim","Cancelar";

IF r_rin=5 THEN
   ExitCycle;
ENDIF

IF r_rin=4 THEN
   MoveJ zx_i,v10,fine,tool0\WObj:=wobj0;
ENDIF

!rodar tool0 em torno de ponto de partida negativo

TPErase;
TPWrite "O robô irá reorientar a ferramenta";
```

```
    TPWrite "automaticamente.";
    TPReadFK r_reor_py,"É seguro iniciar a reorientação?","","","","Sim","Cancelar";

    IF r_reor_py=5 THEN
       ExitCycle;
    ENDIF

    IF r_reor_py=4 THEN
       r_zy_p:=RelTool(zx_i,0,0,0,\Rx:=0,\Ry:=10,\Rz:=0);
       MoveJ r_zy_p,v10,fine,tool0\WObj:=wobj0;
    ENDIF

    !guardar terceiro ponto sobre o plano

    linha_237:
    FCCalib padrao_carga;
    FCAct padrao;
    TPErase;
    TPWrite "Leve a ferramenta ao bloco de";
    TPWrite "calibração.";
    TPWrite "Depois, pressione Memorizar para gravar";
    TPReadFK r_py,"o quarto ponto de calibração?","","","","Memorizar","Cancelar";

    IF r_py=5 THEN
       ExitCycle;
    ENDIF

    IF r_py=4 THEN
       TPErase;
       p_zy_p:=Crobt(\Tool:=tool0\WObj:=wobj0);
       p_zy_p_pos:=p_zy_p.trans;
       p_zy_p_z:=p_zy_p_pos.z;
    ENDIF

    !regresso a ponto de partida

    TPErase;
    FCDeact;
    TPWrite "O robô irá mover-se automaticamente para";
    TPWrite "a zona de reorientação.";
    TPReadFK r_ripn,"É seguro iniciar o movimento?","","","","Sim","Cancelar";

    IF r_ripn=5 THEN
       ExitCycle;
    ENDIF

    IF r_ripn=4 THEN
       MoveJ zx_i,v10,fine,tool0\WObj:=wobj0;
    ENDIF

    MoveAbsJ JointTarget_3,v100,fine,tool0\WObj:=wobj0;
    MoveAbsJ JointTarget_4,v100,fine,tool0\WObj:=wobj0;
    MoveAbsJ JointTarget_5,v100,fine,tool0\WObj:=wobj0;
    MoveAbsJ JointTarget_6,v100,fine,tool0\WObj:=wobj0;
    MToolTCPCalib JointTarget_3,JointTarget_4,JointTarget_5,JointTarget_6,padrao,max_err,mean_err;

  ENDPROC

  PROC verificar()
    MoveAbsJ JointTarget_3,v100,fine,CalculatedTool\WObj:=wobj0;
    MoveAbsJ JointTarget_4,v100,fine,CalculatedTool\WObj:=wobj0;
    MoveAbsJ JointTarget_5,v100,fine,CalculatedTool\WObj:=wobj0;
```

```
    MoveAbsJ JointTarget_6,v100,fine,CalculatedTool\WObj:=wobj0;
ENDPROC

PROC carga_padrao()
    TPErase;
    TPWrite "Robô irá mover-se para a posição Home e iniciar reorientação.";
    TPReadFK r_ZDR,"É seguro iniciar movimento?","","","","Sim","Cancelar";

    IF r_ZDR=5 THEN
        ExitCycle;
    ENDIF

    IF r_ZDR=4 THEN
        ConfJ\Off;
        MoveabsJ Home,v100,fine,tool0\WObj:=wobj0;
        orientar_junta6:=CJointT();
        orientar_junta6.robax.rax_6:=0;
        MoveAbsJ orientar_junta6,v10,Fine,Tool0\WObj:=wobj0;
        orientar_junta5:=CJointT();
        orientar_junta5.robax.rax_5:=45;
        MoveAbsJ orientar_junta5,v10,Fine,Tool0\WObj:=wobj0;
        TPErase;
    ENDIF

    padrao_carga:=FCLoadID();
ENDPROC

PROC carga_calculatedtool()
    TPErase;
    TPWrite "Robô irá mover-se para a posição Home e iniciar reorientação.";
    TPReadFK r_ZDR,"É seguro iniciar movimento?","","","","Sim","Cancelar";

    IF r_ZDR=5 THEN
        ExitCycle;
    ENDIF

    IF r_ZDR=4 THEN
        ConfJ\Off;
        MoveabsJ Home,v100,fine,tool0\WObj:=wobj0;
        orientar_junta6:=CJointT();
        orientar_junta6.robax.rax_6:=0;
        MoveAbsJ orientar_junta6,v10,Fine,Tool0\WObj:=wobj0;
        orientar_junta5:=CJointT();
        orientar_junta5.robax.rax_5:=45;
        MoveAbsJ orientar_junta5,v10,Fine,Tool0\WObj:=wobj0;
        TPErase;
    ENDIF

    calculatedtool_carga:=FCLoadID();
ENDPROC

PROC posicao_bloco()
    FCCalib padrao_carga;
    FCAct padrao;
    TPErase;
    TPWrite "Leve a ferramenta padrão ao bloco de";
    TPWrite "calibração.";
    TPWrite "Depois, pressione Memorizar para gravar";
    TPReadFK r_py,"a posição do bloco de calibração?","","","","Memorizar","Cancelar";

    IF r_py=5 THEN
        ExitCycle;
```

```
    ENDIF

    IF r_py=4 THEN
      TPErase;
      p_calib:=Crobt(\Tool:=padrao\WObj:=wobj0);
    ENDIF
    FCDeact;
    MoveAbsJ JointTarget_5,v100,fine,tool0\WObj:=wobj0;
  ENDPROC

  PROC calibrar()
    FCCalib padrao_carga;
    FCAct padrao;
    TPErase;
    TPWrite "Leve a ferramenta ao bloco de";
    TPWrite "calibração.";
    TPWrite "Depois, pressione OK";
    TPReadFK r_py,"a posição do bloco de calibração?","","","","OK","Cancelar";

    IF r_py=5 THEN
      ExitCycle;
    ENDIF

    IF r_py=4 THEN
      TPErase;
      FCDeact;
      MoveJ Target_270,v100,fine,tool0\WObj:=wobj0;
      UTL1_toolDefPoint;
    ENDIF


    FCCalib padrao_carga;
    FCAct padrao;
    TPErase;
    TPWrite "Leve a ferramenta ao bloco de";
    TPWrite "calibração.";
    TPWrite "Depois, pressione OK";
    TPReadFK r_py,"a posição do bloco de calibração?","","","","OK","Cancelar";

    IF r_py=5 THEN
      ExitCycle;
    ENDIF

    IF r_py=4 THEN
      TPErase;
      FCDeact;
      MoveJ Target_280,v100,fine,tool0\WObj:=wobj0;
      UTL2_toolDefPoint;
    ENDIF
  ENDPROC

  PROC f_tool()

    DistanceLong:=UINumEntry(\Header:="Offset"\Message:="Dlong=?");
    DistanceNew:=UINumEntry(\Header:="Offset"\Message:="DNew=?");
    !First calculate a rotation matrix where d11,d21,d31 represents the x-axis,
    !d12,d22,d32 represents the y-axis and d13,d23, d33 the z-axis.

    !The z-axis is pointing from the tip of the short pin to the tip of the long pin
    d{1,3}:=PinLong.tframe.trans.x-PinShort.tframe.trans.x;
    d{2,3}:=PinLong.tframe.trans.y-PinShort.tframe.trans.y;
    d{3,3}:=PinLong.tframe.trans.z-PinShort.tframe.trans.z;
```

```
  temp:=SQRT(d{1,3}*d{1,3}+d{2,3}*d{2,3}+d{3,3}*d{3,3});
  d{1,3}:=d{1,3}/temp;
  d{2,3}:=d{2,3}/temp;
  d{3,3}:=d{3,3}/temp;

  !The y-axis is perpendicular to the z-axis and in the x-y-plane of tool0
  temp:=SQRT(d{1,3}*d{1,3}+d{2,3}*d{2,3});
  d{1,2}:=-d{2,3}/temp;
  d{2,2}:=d{1,3}/temp;
  d{3,2}:=0;

  !The x-axis is perpendiculat to both the z-axis and the y-axis
  d{1,1}:=d{2,2}*d{3,3}-d{3,2}*d{2,3};
  d{2,1}:=d{3,2}*d{1,3}-d{1,2}*d{3,3};
  d{3,1}:=d{1,2}*d{2,3}-d{2,2}*d{1,3};

  !Now calculate the quaternion values from the rotation matrix
  q{1}:=SQRT(d{1,1}+d{2,2}+d{3,3}+1)/2;
  q{2}:=SQRT(d{1,1}-d{2,2}-d{3,3}+1)/2;
  q{3}:=SQRT(-d{1,1}+d{2,2}-d{3,3}+1)/2;
  q{4}:=SQRT(-d{1,1}-d{2,2}+d{3,3}+1)/2;

  IF d{3,2}<d{2,3}q{2}:=-q{2};
  IF d{1,3}<d{3,1}q{3}:=-q{3};
  IF d{2,1}<d{1,2}q{4}:=-q{4};

  !Now the new tcp-value shall be calculated from the difference of the long pin and the new tool
  !length. This is done with a pose multiplication.
  psa:=PinLong.tframe;
  psa.rot.q1:=q{1};
  psa.rot.q2:=q{2};
  psa.rot.q3:=q{3};
  psa.rot.q4:=q{4};

  psb.trans.z:=DistanceNew-DistanceLong;

  CalculatedTool:=PinLong;
  CalculatedTool.tframe:=PoseMult(psa,psb);


ENDPROC

PROC UTL1_toolDefPoint()
  TPErase;
  TPWrite "Change u_tool and pnt in ";
  TPWrite "dt u_tool, pnt; to the tool you";
  TPWrite "want to define and to the reference";
  TPWrite "point you are using.";
  TPWrite "then_press";
  TPReadFK fkey1,but,"OK","","","","";
  Stop;
  dt PinShort,p_calib;
  TPErase;
  TPWrite "Ready!!";
  TPReadFK fkey1,press_ok,"OK","","","","";
  Stop;
ENDPROC

PROC UTL2_toolDefPoint()
  TPErase;
  TPWrite "Change u_tool and pnt in ";
```

```
    TPWrite "dt u_tool, pnt; to the tool you";
    TPWrite "want to define and to the reference";
    TPWrite "point you are using.";
    TPWrite "then_press";
    TPReadFK fkey1,but,"OK","","","","";
    Stop;
    dt PinLong,p_calib;
    TPErase;
    TPWrite "Ready!!";
    TPReadFK fkey1,press_ok,"OK","","","","";
    Stop;
  ENDPROC

  LOCAL PROC dt(
  PERS tooldata ttool,
  robtarget rp)

    VAR pose pse2:=[[0,0,0],[1,0,0,0]];
    VAR pose pse3:=[[0,0,0],[1,0,0,0]];
    VAR pose pse5:=[[0,0,0],[1,0,0,0]];

    pnt:=CRobT(\Tool:=tool0\Wobj:=wobj0);
    pse2.trans:=rp.trans;
    pse2.rot:=rp.rot;
    pse3.trans:=pnt.trans;
    pse3.rot:=pnt.rot;
    pse5:=PoseMult(PoseInv(pse3),pse2);
    TPErase;
    TPReadFK fkey1,"What do you want to define?","Tool","TCP","Orient","","";
    IF fkey1=1 THEN
      ttool.tframe:=pse5;
    ELSEIF fkey1=2 THEN
      ttool.tframe.trans:=pse5.trans;
    ELSE
      ttool.tframe.rot:=pse5.rot;
    ENDIF
  ENDPROC


ENDMODULE
```