

PROPOSTA DE ARQUITETURA DE CONTROLE DE UM ROBÔ ABB IRB 2000 UTILIZANDO LINUXCNC

Giulia Filippi Giannetti
Carlos Gustavo Muniz Simões
Alberto José Alvares
Guilherme Caribé de Carvalho

Universidade de Brasília - UnB - Faculdade de Tecnologia - Campus Universitário Darcy Ribeiro - Brasília, DF, 70910-900
giulia.filippi@aluno.unb.br, carlos.muniz@aluno.unb.br, alvares@alvarestech.com, gcarval@unb.br

Resumo: Este trabalho apresenta uma proposta de arquitetura de controle do Robô ABB IRB 2000 utilizando o software de Comando Numérico LinuxCNC para a implementação de uma solução de arquitetura de controle aberta e aderente à indústria 4.0, utilizando um Computador Pessoal com uma distribuição Linux com RTAI (Real Time) conectado a uma placa MESA 7i95 responsável por enviar sinais de trem de pulso do controle do LinuxCNC aos servoconversores WEG SCA06 (drive) os quais controlam os motores em cada uma das seis juntas do robô. A estrutura mecânica original do manipulador, que se encontra atualmente no laboratório do Grupo de Inovação em Automação Industrial da Universidade de Brasília, está em ótimo estado e, portanto, pode ser reutilizada quase integralmente. O gabinete de controle original também está em ótimo estado e pode ser facilmente modificado para abrigar o novo sistema de controle. A cinemática do robô, além de compor a lógica de controle, foi utilizada para estruturar um Digital Twin do Robô, a partir de um arquivo escrito em linguagem Python, para criar um modelo de simulação digital do manipulador em tempo real em 3D, como realidade virtual do ativo. Para o desenvolvimento do Digital Twin foi utilizada a norma ISO 23247, Modelo de Referência de Digital Twin para Manufatura. A nova arquitetura de controle do manipulador ABB IRB 2000 desenvolvida neste trabalho permite sua utilização em diversos cenários diferentes, trazendo soluções modernas de controle para um robô de mais de 20 anos e capacitando-o para o uso de tecnologias atuais e que podem ser facilmente modificadas para possibilitar sua integração em soluções de manufatura integrada da indústria 4.0.

Palavras-chave: LinuxCNC, Retrofitting, Digital Twin, Robô ABB IRB 2000

1. INTRODUÇÃO

Com a chegada da indústria 4.0, soluções de modelos ciberfísicos e aplicações de Internet das Coisas (IoT) em sistemas de manufatura vem sendo cada vez mais requisitadas pelas empresas/fábricas que desejam aderir suas plantas aos mais modernos modelos de produção. Nos últimos anos, indústrias e fábricas de todos os portes vêm tentando se adequar e, cada vez mais, se imergir nestas novas soluções tecnológicas (Arenas, 2016). Uma adversidade encontrada neste processo é o alto custo das máquinas que oferecem de fábrica os novos recursos da quarta revolução industrial. Uma solução para este problema em crescente é o uso misto de tecnologias mais modernas com tecnologias já datadas, e o mecanismo de engenharia que viabiliza a integração sinérgica destes sistemas é chamado de Retrofitting (Kopacek *et al*, 2021).

O processo de Retrofitting consiste na integração de tecnologias novas em sistemas antigos ou ultrapassados, permitindo uma modernização ou adequação do processo sem a necessidade de um aporte financeiro mais expressivo para a aquisição de um maquinário completamente novo. A ideia é se utilizar das peças/máquinas já inseridas no sistema e realizar atualizações ou renovações pontuais para permiti-lo às novas aplicações desejadas, sejam elas de software ou hardware (Lins e Oliveira, 2020). Com a demanda crescente de inserção da indústria 4.0 e de seus meios nas fábricas, soluções de Retrofitting serão cada vez mais comuns em sistemas e máquinas que não proporcionam, por exemplo, comunicação digital e infraestrutura com capacidade de IoT (Kopacek *et al*, 2021).

Uma das características mais marcantes da indústria 4.0 é a integração do meio físico com o virtual, e uma das formas de fazer isso em uma máquina/robô é implementando uma réplica virtualmente funcional do robô em um software de simulação. Este tipo de réplica virtual é normalmente chamado de Digital Twin, e é parte extremamente importante do Retrofitting de uma máquina para inseri-la no contexto moderno da indústria 4.0 (Ayani *et al.*, 2018).

No presente trabalho é descrito o processo de criação do Digital Twin para um manipulador robótico modelo ABB IRB 2000 M90A, bem como sua integração no software de controle livre LinuxCNC, além de um projeto de controle inteiramente novo para o manipulador. O manipulador está localizado no laboratório do Grupo de Inovação em Automação Industrial (GIAI) na Universidade de Brasília (UNB), mostrado na Fig. 1. Ele possui 6 graus de liberdade,

acionados por motores síncronos de corrente alternada e é recomendado para uso em diversos cenários, como soldagem a arco, colagem/selagem, montagem, corte à jato d'água e corte a laser (ABB Robotics Products AB, 1993). O sistema robótico original completo consiste no conjunto do manipulador e do gabinete de controle, que permitem apenas algumas funcionalidades muito restritas quando comparadas às competências de um software de controle livre. Para o controle do Digital Twin do robô foi escolhido trabalhar com o LinuxCNC, exatamente por ser um software de controle livre e código aberto, que permite a implementação das mais diversas configurações cabíveis ao robô (LinuxCNC, 2022b).



Figura 1. Gabinete (à esquerda) e Manipulador Robótico (à direita) do ABB IRB 2000.

2. REVISÃO DE LITERATURA

A arquitetura de controle original do ABB IRB 2000 é fechada e restrita ao gabinete de controle original, onde são permitidas poucas funcionalidades, como a especificação de trajetórias de referência, calibração de parâmetros do robô e utilização de programas pré-existentes (e Silva *et al.*, 2010). Para solucionar este problema foram propostas diferentes soluções em Arenas (2016) e em Silva *et al.* (2010), onde os autores desenvolveram novas arquiteturas de controle abertas para os robôs ASEA IRB6-S2 e ABB IRB 2000, respectivamente. Em Arenas (2016), o autor utilizou o software LinuxCNC para a implementação do Gêmeo Virtual do robô ASEA IRB6-S2 e para a implementação da arquitetura de controle completa do robô, já em Silva *et al.* (2010), os autores implementaram o algoritmo de controle e interface gráfica via o software Matlab/Simulink para o robô ABB IRB 2000.

Por ser um software mais completo e melhor condizente com os propósitos deste projeto, foi escolhido utilizar o LinuxCNC em detrimento ao Matlab, trilhando os mesmos passos do autor para o desenvolver o Digital Twin do ASEA IRB6-S2, mas realizando as devidas alterações para o manipulador ABB, visto que o manipulador ASEA possui 5 graus de liberdade enquanto o IRB 2000 possui 6. A utilização do LinuxCNC se mostra interessante por ser uma solução que permite a implementação ponta a ponta do retrofit do robô, com simulação do Digital Twin em tempo real, de implementação descrita neste documento, e sendo cabível para implementação da arquitetura de controle livre e completa do manipulador, baseada em linguagem de programação de Comando Numérico, em uma atualização futura.

Por motivos de simplificação e em função do desenvolvimento deste projeto ocorrer no mesmo laboratório em que foi desenvolvida a solução utilizada de base (Arenas, 2016), foi proporcionado um computador pessoal que é um clone do computador utilizado pelo autor para desenvolver seu trabalho com o robô ASEA. A versão do LinuxCNC a ser utilizada é a 2.8, em função de sua compatibilidade nativa com a placa Mesa 7i95/7i97, especificada para integração com os servoconversores, e que é capaz de atuar em tempo real. O LinuxCNC possui código aberto e suas configurações são todas descritas em documentos de texto/arquivos que são acessíveis e editáveis pelo usuário.

3. METODOLOGIA

Ao realizar uma inspeção no manipulador e no gabinete, foi notado que as partes mecânicas do manipulador se encontram em ótimo estado e não necessitam ser substituídas, já no gabinete de controle foi descoberto que algumas partes do circuito interno estão queimadas e, por ser um modelo que não é mais fabricado, não existem peças de reposição no mercado, e portanto deverá ser aplicada uma nova arquitetura de controle para o manipulador. A estratégia escolhida é similar à desenvolvida em Silva *et al.* (2010), mas para outro software de controle. Para fazer a integração com o LinuxCNC serão herdadas algumas das soluções em Arenas (2016). A solução para a arquitetura de controle

escolhida usa como base o uso de uma placa Mesa 7i95/97, conectada a um computador central por meio de um cabo Ethernet para fazer o link de comando entre a CPU e os drivers dos motores do manipulador. A placa Mesa 7i95 possuem um conjunto de 24 entradas e 6 saídas de uso geral, mais suporte para controle de até seis eixos por meio de sinais step/dir, com feedback por sinal de encoder, e por isso é a placa de preferência para a implementação da arquitetura de controle (Mesa, 2022). Em contrapartida, a placa 7i97 possui saídas analógicas de +-10V em vez das saídas por step/dir.

Os servoconversores WEG SCA06 possibilitam controlar motores de corrente alternada trifásicos com realimentação por resolver, o que permite o uso para controle nos seis motores síncronos do manipulador. Esses drivers podem ser controlados por meio de entradas digitais, no entanto, não possuem função de encoder. Como a placa de interface requer sinais de encoder, pode-se utilizar o módulo de expansão EAN1, também da fabricante WEG, que realiza a simulação de encoder para comunicação entre esses componentes. Outra vantagem da utilização deste módulo de expansão é que além de simular sinal de encoder, expande a quantidade de portas I/O, possibilitando a implementação de um sistema de segurança mais robusto.

Por fim, na Fig. 2, é possível observar graficamente todos os pontos levantados acima para construção da arquitetura de controle proposta. Vale mencionar que os sistemas de segurança e de proteção não estão representados na figura, mas foram levados em consideração na elaboração da arquitetura bem como na escolha dos componentes, visto que tanto a placa de interface MESA quanto os servoconversores SCA06 acoplados com o módulo EAN1 possuem entradas do tipo I/O para a plena implementação destes sistemas.

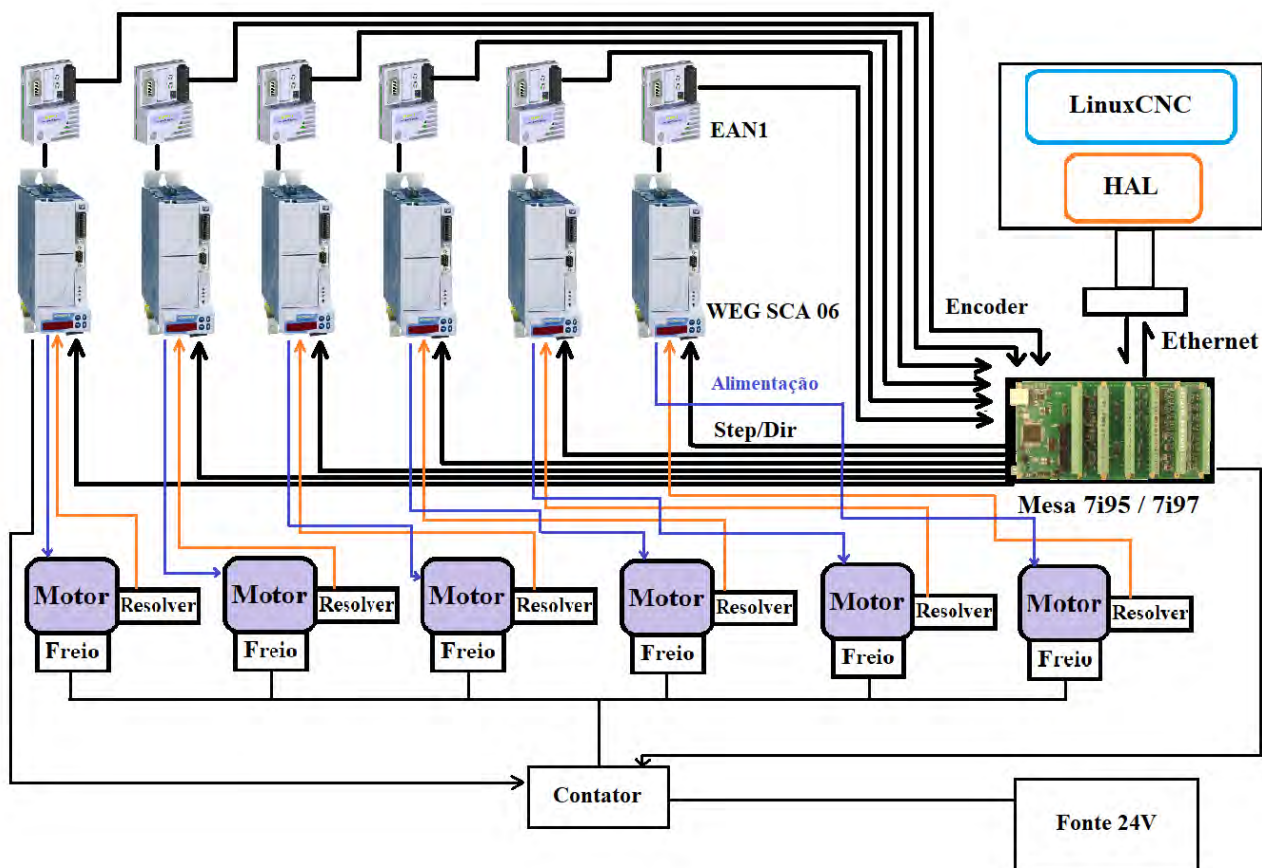


Figura 2. Arquitetura de controle a ser utilizada

É necessário enfatizar que, ainda que bem resolvida, esta arquitetura ainda não pôde ser implementada devido ao processo de compra das peças ainda estar em trâmite. Ainda assim, neste documento estão descritos todos os procedimentos paralelos à inserção das peças mecânicas para o pronto funcionamento do sistema, assim que forem adquiridas e instaladas. Há expectativa que todos os componentes cheguem por volta do mês de julho de 2023, mas esta data ainda não é final.

Tabela 1. Especificação de componentes requeridos para arquitetura de controle

Especificação	Quantidade	Custo Unitário	Custo Total	Justificativa
SERVOCONVERSOR CA SCA06B05P0D2C3P6 - NCM: 9032.89.89	5	5.000,00	R\$ 25.000,00	Controle servomotor Robô
SERVOCONVERSOR CA SCA06C08P	1	6.000,00	R\$ 6.000,00	Controle servomotor Robô
MODULO EXPANSAO FUNCAO EAN1 PARA SERVOCONVERSOR - NCM: 9032.90.10	6	800,00	R\$ 4.800,00	Converte Sinal Resolver em Encoder
MINICONTATOR AZ CW07-10-30V25 - NCM: 8536.49.00	12	50	R\$ 600,00	Proteção
SINALEIRO CEW-SM1-D23 MONOBLOCO VERMELHO 22 mm DE RESINA TERMOPLASTICA 220V 50/60Hz-CC - NCM: 8531.80.00	12	50	R\$ 600,00	Proteção
CONVERSOR DE CORRENTE CONTINUA BASEADO EM TECNICA DIGITAL FONTE ALIMENTACAO	2	1000,00	R\$ 2.000,00	Fonte 24 Volts Freios Robô
Placa Mesa 7i95 (importado R\$ 5,00 o Dólar, incluindo importação)	1	3.000,00	R\$ 3.000,00	Interface LinuxCNC com Servoconversor
Placa Mesa 7i97 (importado R\$ 5,00 o Dólar, incluindo importação)	1	3.000,00	R\$ 3.000,00	Interface LinuxCNC com Servoconversor
Fita isolante, conectores, painel, trilhos, outros necessários para montagem	1 (pacote com vários consumíveis)	2.500,00	R\$ 2.500,00	Montagem painel proteção servoconversores

4. IMPLEMENTAÇÃO E RESULTADOS

A implementação de um digital twin funcional no caso do uso do LinuxCNC é dependente do kernel utilizado. O software é disponível para download em algumas versões de sistema operacional Linux, mas para que haja coordenação em tempo real entre o modelo e o robô é necessário um kernel RTAI, ou pela distribuição Wheezy para arquiteturas em 32 bits ou pela distribuição Buster ou Ubuntu para arquiteturas 64 bits (LinuxCNC, 2022b). O funcionamento do digital twin em conjunto com o manipulador real é declarado principalmente nos arquivos de configuração: o arquivo .ini define principalmente a aparência do controlador para o usuário, e aspectos importantes da movimentação do robô, como valores máximos e mínimos, velocidade, aceleração e posição do home; o arquivo .hal define parâmetros da interface entre componentes do sistema, principalmente drivers e placas de interface, como é o caso das placas Mesa. Além destes, para um robô com cinemática não trivial, pode ser necessário incluir no sistema novos arquivos que descrevem a cinemática direta e inversa do manipulador. No caso do robô ABB IRB 2000, como este possui seis graus de liberdade, é possível utilizar arquivos de cinemática já disponíveis como base para a construção das equações. Para a implementação, foi utilizada a cinemática da configuração gensekings, que necessita apenas da configuração dos parâmetros DH modificados no arquivo de configuração HAL (LinuxCNC, 2022b).

4.1. CINEMÁTICA DO MANIPULADOR ABB IRB 2000

A implementação da cinemática realizada em software LinuxCNC tem como base a notação Denavit-Hartenberg para os parâmetros das juntas do robô. Esses parâmetros definem a relação entre as juntas, e a partir de transformações

lineares, é possível adquirir a posição cartesiana do end-effector. Da mesma forma, a transformação linear inversa permite adquirir o ângulo das juntas a partir do valor cartesiano do end-effector. Os parâmetros DH modificados do robô são dados na Tab. 2.

Tabela 2. Parâmetros DH do robô ABB IRB 2000

	α_{n-1}	a_{n-1}	d_n	θ_n
Junta 1	0	0	750	θ_1
Junta 2	$-\frac{\pi}{2}$	0	0	θ_2
Junta 3	0	710	0	θ_3
Junta 4	$-\frac{\pi}{2}$	125	850	θ_4
Junta 5	$\frac{\pi}{2}$	0	0	θ_5
Junta 6	$-\frac{\pi}{2}$	0	100	θ_6

Para a configuração desses parâmetros considerando a cinemática gensekings, é necessário considerar a posição inicial do robô a mostrada na Fig. 5. Isso é feito pois há uma rotação equivalente a 90° em torno do eixo z entre a segunda e quarta junta, o que implicaria em um ângulo θ com offset diferente de zero, que não é implementado na cinemática (LinuxCNC, 2022c).

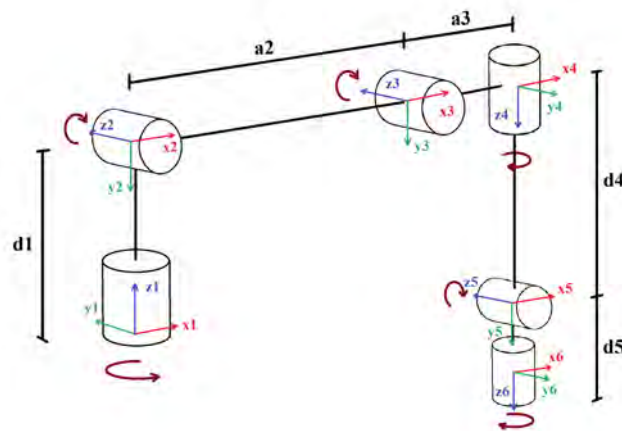


Figura 3. Configuração inicial considerada para a determinação dos parâmetros DH modificados.

4.2. CRIAR E CONFIGURAR UM NOVO PROJETO NO LINUXCNC

Para adicionar o modelo do robô dentro do LinuxCNC deve-se começar uma nova configuração (projeto) dentro do software, associado ao hardware que será utilizado e para isso utiliza-se o assistente “Stepconf”. Esse assistente é especificado para a configuração do LinuxCNC associado ao hardware especificado, permitindo apenas configurações de no máximo 4 eixos (LinuxCNC, 2022b), o que não atende otimamente neste caso, porém pode-se utilizá-lo para gerar a pasta com os arquivos de configuração e posteriormente complementa-los manualmente com configuração adicional para atender as necessidades do projeto, que foi a estratégia escolhida.

Cabe destacar que as placas Mesa não estão disponíveis, bem como os demais módulos de hardware servoconversores Weg, e outros, e por este motivo trabalha-se como uma interface baseada em Porta Paralela que irá permitir uma configuração dos pinos de sinais de I/O necessário para simular o Digital Twin do Robô ABB IRB 2000. Após validação da cinemática do Robô e a simulação da usinagem nas GUI LinuxCNC, dada a certeza que a cinemática direta e inversa estão trabalhando de forma correta, a configuração do LinuxCNC para o Robô ABB IRB 2000 estará devidamente validada. Após a aquisição do hardware Mesa e Servoconversores Weg a integração lógica e física será

realizada, neste momento o foco é a validação da cinemática sem atualização do hardware, apenas configuração LinuxCNC e cinemática não trivial do Robô.

Como se utiliza o Stepconf apenas para gerar a pasta de arquivos corretamente, basta configurar o nome da máquina na primeira tela de configuração do Stepconf e passar por todas as próximas telas sem fazer outras modificações, o nome escolhido para o projeto foi “abbirb”. Se necessário, posteriormente é possível acessar o projeto finalizado pelo Stepconf e realizar as modificações que forem necessárias manualmente. Ao finalizar o procedimento, é criada uma pasta de configuração de uma máquina nova no diretório “linuxcnc/configs” chamada de “abbirb”. Dentro da pasta da nova configuração podemos encontrar os arquivos “*abbirb.ini*” e “*abbirb.hal*” que são os arquivos de texto que deve-se editar para as configurações do robô; são arquivos de texto comum e podem ser acessados com qualquer editor de texto simples.

A fim de possibilitar o uso desta configuração como base para outros projetos futuros ou versões diferentes desta mesma configuração, esta pasta criada pelo assistente deve ser copiada para o diretório fonte do LinuxCNC e, após isso, o LinuxCNC deve ser reinstalado para finalizar o processo de criação do template de configuração. Para instalar o LinuxCNC a partir do código fonte, basta acessar o arquivo README no diretório fonte, nos conteúdos deste arquivo é possível encontrar o caminho para o arquivo INSTALL, que contém todas as informações necessárias para realizar a instalação do LinuxCNC.

4.4. CONFIGURAÇÃO DO ARQUIVO INI

No arquivo .ini estão contidos todos os parâmetros de atuação dos eixos do manipulador, além da indicação do nome dos arquivos auxiliares e outros parâmetros necessários para a simulação do Digital Twin. Como explicado, o arquivo “*abbirb.ini*” que foi criado pelo Stepconf não estava completo e precisou ser editado para concordar com o modelo do nosso robô. Para ficar mais entendível ao usuário, o arquivo ini é dividido em seções e a seguir indica-se onde foram feitas as modificações.

EMC: Apresenta configurações gerais do arquivo. Versão do arquivo, para o editor poder manter documentada suas edições, nome da máquina e um código de DEBUG para gerar mensagens durante a execução da máquina.

DISPLAY: Apresenta opções de interface gráfica do usuário(GUI). Foram mantidas as predefinições geradas pelo Stepconf.

RS274NGC: Nesta seção é definido o arquivo para tradução de instruções da norma RS724 para funções padrões da máquina.

ECMOT: Nesta seção são definidos os parâmetros de controle em tempo real. Não foram feitas modificações.

HAL: Nesta seção estão indicados os arquivos necessários para a simulação em tempo real do Digital Twin. Nesta etapa foram feitas apenas edições nos nomes dos arquivos para que ficassem condizentes com o nosso projeto.

TRAJ: Aqui são indicados parâmetros gerais dos eixos do robô, como número de juntas, definição de unidades de trabalho e definições de velocidade e aceleração.

AXES: Nesta seção são descritos parâmetros específicos de cada eixo do robô, como configurações de limite de velocidade e aceleração, limites angulares das juntas e definições de zeramento do eixo (função HOME). Na Fig. 13 pode ser vista a configuração feita para o eixo 0 do robô. O mesmo foi feito para todos os outros eixos, adequando os parâmetros de configuração corretamente para cada um deles.

Na documentação (LinuxCNC, 2022b) é possível encontrar mais informações sobre os parâmetros de cada seção do arquivo .ini e instruções para o cálculo do valor de escalas de step, encoder, e de outros parâmetros.

4.4. CONFIGURAÇÃO DO ARQUIVO HAL

O arquivo HAL permite a conexão do meio físico, ou seja, dos componentes do manipulador, com o meio computacional, fazendo o link de controle do software com o robô. Da mesma forma que o arquivo ini, o arquivo hal gerado pelo Stepconf também precisa ser editado para condizer com o modelo do robô, e para auxiliar nesta etapa foi utilizado como base o arquivo hal do robô puma560, disponível na biblioteca fonte do LinuxCNC, como também os tutoriais disponíveis na página do LinuxCNC (LinuxCNC, 2022b). O arquivo hal também é responsável por carregar o modelo virtual do robô, no caso chamado de abbgui, que é configurado por um arquivo em python descrito na seção seguinte.

4.5. CONFIGURAÇÃO DE SIMULAÇÃO DO MODELO VIRTUAL

Com as configurações dos arquivos `.ini` e `.hal` já bem definidas, é possível realizar a simulação do Digital Twin do manipulador dentro do LinuxCNC utilizando um pacote denominado Vismach (Virtual Machine). O módulo Vismach consiste de uma janela separada do LinuxCNC que é aberta assim que o arquivo de configuração é iniciado, desde que especificado no arquivo `hal`, conforme a Fig. 10 por meio do comando `loadusr`. Para a simulação, todas as peças do robô foram desenhadas separadamente em um sistema de CAD e importadas (formato `.obj`) para a pasta de configuração da máquina, que foi criada pelo Stepconf e copiada para o diretório fonte do LinuxCNC. O Vismach atua diretamente com o HAL, de forma que as animações das peças são feitas conforme os valores dos pinos HAL associados a elas são alterados. O modelo da máquina virtual do Vismach é montado a partir de um script em Python que foi criado seguindo as indicações mostradas na documentação do Vismach (LinuxCNC, 2022a) e foi copiado para o diretório `usr/bin` que é onde os códigos Vismach do LinuxCNC ficam dispostos.

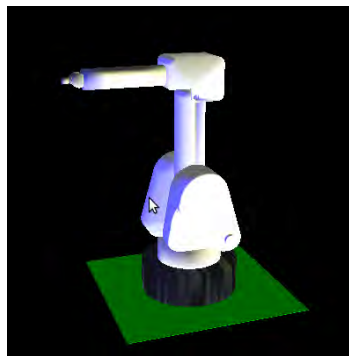


Figura 4. Modelo virtual simplificado do robô.

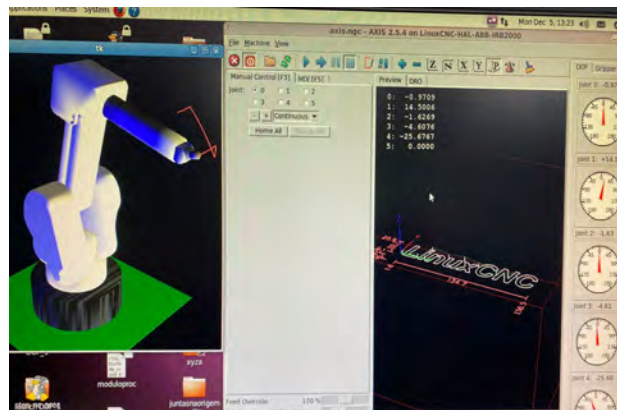


Figura 5. Aparência do modelo virtual em conjunto com Enhanced Machine Controller.

A configuração do módulo de controle foi realizada em uma versão prévia do software LinuxCNC (2.5.4), em função da máquina utilizada ser a mesma do projeto de retrofitting do robô IRB 6 (Arenas, 2016), que possui sistema operacional Ubuntu obsoleto, e portanto está sendo transferida para sistema Debian a fim de permitir a utilização da versão compatível com a placa de interface escolhida. A adição dos arquivos `.ini` e `.hal` na configuração adicionada permitiu verificar a movimentação das juntas do modelo virtual a partir da interface principal do software. Para o controle devido do robô, será necessário realizar a instalação dos servoconversores e a interconexão das partes a partir da placa Mesa e arquitetura especificada, uma vez que as peças forem adquiridas.

4.6. SOLUÇÃO PARA IMPLEMENTAÇÃO DO HARDWARE DO SISTEMA DE CONTROLE

O propósito desta seção é apresentar maiores detalhes sobre a integração dos equipamentos mostrados na Fig. 2. Ressalta-se que esta parte da implementação não foi desenvolvida em função da espera pela retificação do edital de financiamento para compra dos equipamentos e pelos extensos períodos de entrega das peças por falta de componentes por parte dos fabricantes.

No manipulador original, duas placas são as principais responsáveis pelo controle da unidade: a unidade de freio e a placa de medição serial (SMB). A placa de medição serial é responsável pelo envio das posições dos motores para a cabine de controle por meio de um único sinal serial no cabo X2. Os comandos, por sua vez, são enviados diretamente da cabine por meio do cabo X1. De forma semelhante, a unidade de freios é responsável pelo controle individual e

coletivo dos freios do manipulador, cujos sinais de comando são recebidos também pelo cabo X1 (ABB Robotics Products AB, 1993). A implementação da arquitetura de controle nova configura um bypass dessas duas unidades por meio do uso dos novos servoconversores WEG SCA-06, o que é benéfico para o sistema uma vez que a placa SMB requer uma bateria que deve ser trocada regularmente. Assim, os servoconversores ficam responsáveis por essas funções da mesma forma que o conjunto MESA 7i95- LinuxCNC substituem a cabine de controle.

Os servoconversores SCA-06 correspondem a uma solução completa para controle de motores pois são altamente configuráveis por meio da programação dos mais de 1200 parâmetros de funcionamento que cobrem desde a função das entradas e saídas analógicas a funções relacionadas à velocidade, posições limites e tempos de acionamento do motor. A configuração dos sinais de controle do servo conversor deve ser feita por meio desses parâmetros. Ademais, a conexão básica do servo conversor a um motor é mostrada na Fig. 6. A interface do botão de segurança ocorre próximo ao resistor de frenagem no servo conversor, e os freios são alimentados com 24 V (WEG, 2020).

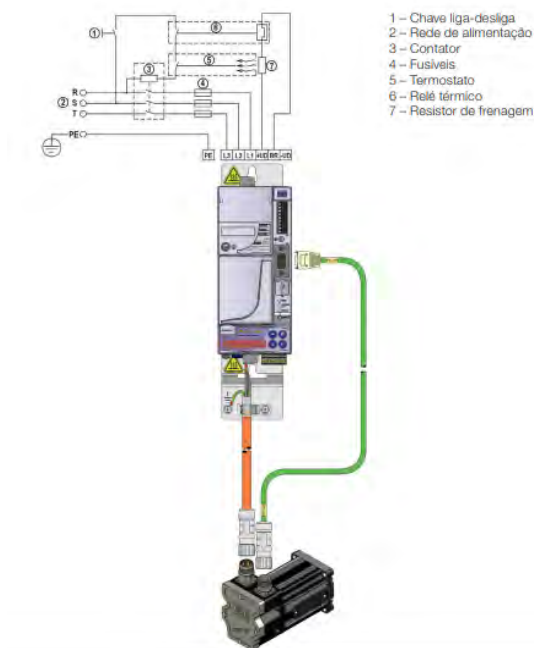


Figura 6. Conexões de potência e controle do servo conversor. (WEG, 2020)

Em combinação com os servoconversores será necessária a utilização de módulos de expansão do modelo EAN1, da mesma fabricante. Estes módulos serão utilizados para viabilizar a conexão entre a placa FPGA MESA e os SCA06, visto que os servoconversores não possuem saída de encoder para realimentação. A instalação do módulo EAN1 é dada em uma entrada própria do servoconversor SCA06. O sinal de encoder é adquirido no módulo de expansão pelo conector XA133 para realizar a simulação de encoder. Pelo lado da placa MESA, as conexões de encoder e sinal STEP/DIR (padrão RS-422) estão bem definidas nas informações escritas na placa e em caso de dúvida podem ser consultados os manuais. Em função da grande variedade de formas de controle, no SCA06 é necessário configurar o modo posicionador relativo, que corresponde ao acionamento do motor em resposta a uma entrada digital e cujo comportamento também é configurável por outros parâmetros. Inicialmente, o parâmetro P00202 deve ser configurado como 3 para habilitar o modo posicionador, e o parâmetro P01070 deve ser configurado como 0 a fim de determinar que as três entradas digitais próprias do servoconversor serão utilizadas para controlar o movimento. Neste modo de operação, ao receber o comando para realizar um ciclo por meio dessas entradas, o tamanho do deslocamento realizado será determinado pelo número de voltas e frações de volta configurados nos parâmetros P01181 e P01171 respectivamente, e a resolução do conversor é de 16384 pulsos por volta, ou 0,02°. Essa resolução deve ser configurada no driver e no LinuxCNC, pois irá determinar o deslocamento de cada pulso do sinal de controle. Uma vez que a comunicação utilizada é STEP/DIR, o tipo de movimento 1 (P01151) deve ser igual a 3. Supondo a entrada digital 1, DI1 como a entrada STEP, o parâmetro P01081 deve ser 1, P01101 deve ser 1 e ambos P01102 e P01077 devem ser 0. Dessa forma, apenas um movimento será acionado na borda de subida do pulso. Para o sinal DIR, basta configurar o parâmetro P00301 como 7, de forma que o acionamento da entrada digital 2 inverte o sentido de giro do motor. Além destes, também pode-se utilizar outros parâmetros a fim de realizar o ajuste fino do acionamento dos motores, como por exemplo, o uso da entrada digital 3 para realizar a busca do zero máquina, uso de entradas digitais do módulo de expansão EAN1 para ocasionar o travamento do eixo do motor em caso de falha, configuração da velocidade máxima em rpm no pulso e configuração da aceleração desde a parada até a velocidade indicada no pulso (WEG, 2019). De modo geral, ocorre uma sobreposição de algumas funções do controle pelo software LinuxCNC, o que permite a formação de um sistema redundantemente seguro.

Por fim, a conexão da placa de interface MESA com o computador é dada por cabo Ethernet e, depois de concluídas as conexões, o sistema deverá ser parecido com o exemplificado na Fig. 2.

4.7. ANÁLISE DOS RESULTADOS OBTIDOS

Com base no que foi demonstrado nos parágrafos acima, a maior contribuição do projeto até o momento são o desenvolvimento do Digital Twin e da configuração dos parâmetros de controle no LinuxCNC. A configuração do robô ABB IRB 2000 não é nativa do LinuxCNC e, por ser um software aberto e livre, é possível compartilhar esta configuração com os demais usuários do software, permitindo que outros pesquisadores explorem o retrofit deste manipulador utilizando ou melhorando o trabalho desenvolvido neste projeto. Além disso, como citado anteriormente, o gabinete de controle original limita bastante o manipulador em questão de funcionalidades, portanto a utilização de um computador com o LinuxCNC para o controle do manipulador e drivers servoconversores configuráveis expande muito o horizonte de possibilidades de uso deste conjunto. Com a alta capacidade de comunicação e expansão de um computador moderno, a utilização deste manipulador em cenários e configurações de fábrica da indústria 4.0 fica extremamente mais acessível e factível que na disposição original do conjunto manipulador e gabinete.

Além desses fatores, também é possível ressaltar a proposta da arquitetura de controle, que apesar de não ter sido efetivamente implementada em favor dos motivos levantados anteriormente, está bem documentada neste trabalho e possui respaldo técnico de trabalhos anteriores e dos manuais das peças que a compõem.

5. CONCLUSÕES

Este trabalho apresenta os procedimentos de implementação de uma solução de retrofitting de um manipulador industrial, e em mais detalhes a criação de um modelo virtual, com integração e controle pelo software LinuxCNC. A implementação da configuração específica para o robô por meio de software LinuxCNC seguiu diretrizes já estabelecidas com base em projetos anteriores desenvolvidos pelo laboratório do Grupo de Inovação em Automação Industrial (GIAI) e outras fontes. A implementação do controle do robô está em processo de transição para uma máquina nova, com hardware e sistema operacional mais recente. Além disso, a solução desejada inclui um arquivo de cinemática exclusivo para o robô ABB IRB 2000, e portanto conta-se com a implementação de um arquivo customizado a fim de proporcionar maior contribuição para a biblioteca do LinuxCNC futuramente. Outro aspecto a ser otimizado é a implementação de sensores de posição no robô a fim de proporcionar uma sequência de homing bem definida e com isso, melhorar a exatidão das aplicações funcionais do manipulador. Finalmente, espera-se o maior desenvolvimento da solução posterior ao resultado do processo de solicitação de recursos, a partir do qual será possível adquirir o equipamento necessário para a concretização da arquitetura proposta.

6. AGRADECIMENTOS

Agradecemos o apoio da Fundação de Apoio à Pesquisa do DF (FAPDF) pela concessão de bolsas de Iniciação Científica e apoio, através de Projeto de Pesquisa, para compra dos servoconversores e demais equipamentos.

7. REFERÊNCIAS

- ABB Robotics Products AB, 1993. *Product Manual IRB 2000 M90*. Västerås, Suécia.
- Arenas, J.S.T., 2016. “*Retrofitting do robô ASEA IRB6-S2 baseado em tecnologias de comando numérico usando LinuxCNC*”. Tese de Mestrado. UnB - Universidade de Brasília. Brasília, Brasil.
- Ayani, M., Ganebäck, M. e Amos, H.C.Ng, 2018. *Digital Twin: Applying emulation for machine reconditioning*. Procedia CIRP. Vol 72. Pages 243-248. ISSN 2212-8271. <<https://doi.org/10.1016/j.procir.2018.03.139>>.
- Djuric, A.M. e ElMaraghy, W.H., 2010. “*Generalized Reconfigurable 6-Joint Robot Modelling*”. Intelligent Manufacturing Systems (IMS) Centre - University of Windsor. Windsor, Canadá.
- Kopacek, P., Bauer, J., Llaja, M., Fauda, S., Acosta, J., Algoter, R., e Cuello, F., 2021. “*Robot retrofitting by using LinuxCNC complemented with arduino/RaspberryPI*”. Proc. Int. Symp. Prod. Res., Lecture Notes in Mechanical Engineering. 2021. pp. 222–235.
- LinuxCNC, 2022a. *Vismach*. LinuxCNC Documentation. <<https://linuxcnc.org/docs/html/gui/vismach.html>>.
- LinuxCNC, 2022b. *LinuxCNC V2.8.2-23-g93a844ef5*. LinuxCNC Documentation. 01 Jan. 2022 <http://linuxcnc.org/docs/stable/pdf/LinuxCNC_Documentation.pdf>.
- LinuxCNC, 2022c. *Setting up "modified" Denavit-Hartenberg (DH) parameters for gensekins*. LinuxCNC Documentation. <<https://linuxcnc.org/docs/devel/html/motion/dh-parameters.html>>.
- Linx, T. e Oliveira, R.A.R., 2020. *Cyber-physical production systems retrofitting in context of industry 4.0*, Computers & Industrial Engineering. Vol 139. 2020. ISSN 0360-8352. <<https://doi.org/10.1016/j.cie.2019.106193>>.
- Mesa, 2022. *7i95 Ethernet Step/Dir + Encoder Motion Control Interface*. <<http://www.mesanet.com/pdf/parallel/7i95man.pdf>>

- du Preez, R. 2014. *3D 6-DOF SERIAL ARM ROBOT - Kinematics and Implementation in LinuxCNC*. SA-CNC-CLUB. 5 Feb. 2014.
- e Silva, A. L., Orenstein, L.P., Leite, F.L.A.C. e Gleizer, G. 2010. *Atualização de hardware e software de um robô industrial*. XVII Congresso de Automática. Bonito, Brasil.
- Tran, T. -A., Ruppert, T., Eigner, G. and Abonyi, J., 2022. *Retrofitting-Based Development of Brownfield Industry 4.0 and Industry 5.0 Solutions*. IEEE Access. vol. 10. pp. 64348-64374. 2022. doi: 10.1109/ACCESS.2022.3182491.
- WEG, 2020. *SCA06 - Manual do Usuário*. Servoconversor SCA06. <<https://static.weg.net/medias/downloadcenter/hdf/h28/WEG-sca06-manual-do-usuario-10000629636-manual-portugues-br.pdf>>
- WEG, 2019. *SCA06 - Manual de Programação*. Servoconversor SCA06. <<https://static.weg.net/medias/downloadcenter/hf2/h52/WEG-SCA06-manual-de-programacao-1000062686-pt.pdf>>.

8. RESPONSABILIDADE PELAS INFORMAÇÕES

Os autores são os únicos responsáveis pelas informações incluídas neste trabalho.

RETROFITTING OF AN ABB IRB 2000 ROBOT USING LINUXCNC

Giulia Filippi Giannetti

Carlos Gustavo Muniz Simões

Alberto José Alvares

Guilherme Caribé de Carvalho

Universidade de Brasília - UnB - Faculdade de Tecnologia - Campus Universitário Darcy Ribeiro - Brasília, DF, 70910-900
giulia.filippi@aluno.unb.br, carlos.muniz@aluno.unb.br, alvares@alvarestech.com, gccarval@unb.br

Abstract: *This work presents a proposal for the control architecture of the ABB IRB 2000 Robot using the Numerical Command software LinuxCNC for the implementation of an open control architecture solution that adheres to industry 4.0, using a Personal Computer with a Linux distribution with RTAI (Real Time) connected to a MESA 7i95 board responsible for sending pulse train signals from the LinuxCNC control to the WEG SCA06 servoconverters (drive) which control the motors in each of the six joints of the robot. The original mechanical structure of the manipulator, which is currently in the laboratory of the Industrial Automation Innovation Group at the University of Brasília, is in excellent condition and, therefore, can be reused almost entirely. The original control cabinet is also in very good condition and can be easily modified to house the new control system. The robot's kinematics, in addition to composing the control logic, were used to structure a Digital Twin of the Robot, from a file written in Python language, to create a digital simulation model of the manipulator in real time in 3D, as virtual reality of the robot. active. For the development of the Digital Twin, the ISO 23247 standard, Digital Twin Reference Model for Manufacturing, was used. The new control architecture of the ABB IRB 2000 manipulator developed in this work allows its use in several different scenarios, bringing modern control solutions to a robot that is more than 20 years old and enabling it to use current technologies that can be easily modified to enable its integration into industry 4.0 integrated manufacturing solutions.*

Keywords: *LinuxCNC, Retrofitting, Digital Twin, ABB IRB 2000 Robot*

RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.