



BRAÇO ROBÓTICO PARA POSICIONAMENTO DE UM SISTEMA DE VISÃO ARTIFICIAL PARA MONITORAR DESGASTE DE FLANCO DA FERRAMENTA DE USINAGEM DE UM TORNO CONVENCIONAL

ROBOTIC ARM FOR POSITIONING AN ARTIFICIAL VISION SYSTEM FOR TOOL WEAR MONITORING IN THE TURNING MACHINING

J. K. KHOURY JUNIOR^{1*}, R. N. F. VIDAL¹, I. G. ALMEIDA², E. O. LUCAS¹ and G. P. VILLIBOR¹

¹ Universidade Federal de Viçosa, Departamento de Engenharia de Produção e Mecânica, Viçosa, MG, Brasil

² Universidade Federal de Viçosa, Departamento de Engenharia Elétrica, Viçosa, MG, Brasil

*Joseph Kalil Khoury Junior. Universidade Federal de Viçosa, Departamento de Engenharia de Produção e Mecânica, Viçosa, MG, Brasil, Phone:+55 31. 988851815

e-mail address: kalil@ufv.br (J. K. Khoury Junior).

ARTICLE INFO

Article history:

Received 2019-12-03

Accepted 2020-02-20

Available online 2020-02-20

palavras-chave

Manipulador robótico

Microcontrolador

Placa única

keywords

Robot

Microcontroller

Arduino

ABSTRACT

The tool wear and damage of the insert is one of the factors that directly influence the part manufactured, dimensional tolerances and finish. The flank wear monitoring can mainly prevent the loss of parts and early replacement of the tool. The aim was to build an automated robotic arm to flank insert image acquisition on turning machining. The robotic arm image can be utilized with any image processing software to identify and measure flank wear. In this work, mechanical, electronic and software parts of the robotic manipulator were developed for being able to position the digital microscope to acquire images with high repeatability between each monitoring. The system developed was able to improve the lost accuracy from stiffness and clearances on mechanical parts and acquire images with high precision, whose correlation coefficient was 97% between the target and automatically acquired image.

RESUMO

O desgaste e avarias da ferramenta de corte (inserto) durante o processo de usinagem é um dos fatores que influenciam diretamente na fabricação de uma peça, na tolerância dimensional e seu acabamento. O monitoramento do inserto pode evitar a perda de peça, bem como a reposição da ferramenta antes da real necessidade. Deste modo, objetivou-se construir um braço robótico para automatizar o processo de aquisição de imagens do flanco da ferramenta de corte em um torno mecânico. O manipulador robótico desenvolvido adquire imagem para fins de monitorar e medir desgaste do flanco com o auxílio de algum software de processamento de imagens. Nesse trabalho foi desenvolvida a parte mecânica, eletrônica e de programação do manipulador robótico de modo que ele seja capaz de posicionar o microscópio digital para aquisição de imagens com alta repetitividade entre um monitoramento e outro. Pode-se concluir que o sistema com fechamento da malha permitiu corrigir as folgas dos componentes mecânicos e adquirir imagens com precisão e acurácia altas, cujo coeficiente de correlação foi de 97% entre as imagens alvos e adquiridas automaticamente.

1. INTRODUÇÃO

O fim da vida da ferramenta de corte, ou a perda efetiva da capacidade de corte, é detectável quando ocorrem mudanças perceptíveis nas variáveis de processo. Numa operação de desbaste, tais mudanças são perceptíveis no aumento da força de usinagem, mudanças no ruído, na forma do cavaco, vibrações, qualidade da superfície usinada e a exatidão dimensional (HOPKINS et al. 2019; PRADO et al. 2017).

O maior obstáculo ao aumento da produtividade é a parcela de tempo improdutivo gerado pela interrupção do processo (parada de máquina) para substituir uma ferramenta de corte por falha prematura. Isto torna a vida da ferramenta de corte uma das importantes considerações econômicas na usinagem (HESSER et al. 2019; LUCAS, 2003).

Com o desenvolvimento da indústria nacional de alta tecnologia, os processos de fabricação se tornaram mais importantes no cotidiano das empresas. Com isso, se faz necessário o desenvolvimento de novas tecnologias que possibilitem melhorar a qualidade das peças fabricadas e o aumento da produtividade (CHRYSSOLOURIS, 2006).

Uma dessas novidades é o monitoramento em tempo real da integridade das ferramentas de usinagem por meio de um software que é capaz de analisar diferentes imagens e definir se a ferramenta de corte está apropriada para uso (GARCÍA-ORDÁS et al., 2017).

Nesse contexto, ferramentas de cortes frequentemente sofrem avarias em suas extremidades e para que a operação de usinagem de peças seja eficaz, é necessário que o inserto esteja com desgastes de flanco inferiores a 0,15 mm no acabamento ou 0.40 mm no desbaste, conforme a norma ISO3685. Fora disto, podem ocorrer danos à peça usinada, bem como problemas no inserto e no porta ferramenta. A Figura 1 mostra as avarias que podem ocorrer nos flancos e faces de uma ferramenta de corte.

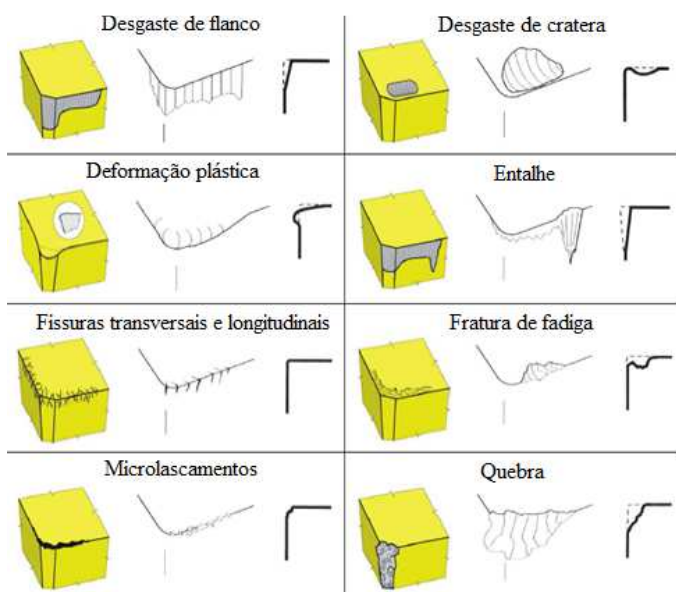


Figura 1- Principais regiões de desgaste de uma ferramenta de corte. Fonte: SANDVIK, 1994

O problema do controle do desgaste das ferramentas de corte é a dificuldade do operador visualizar e mensurar

visualmente o nível do desgaste em tempo hábil. Por isso, existem métodos indiretos para tal função como detectar vibração da ferramenta, qualidade do acabamento da peça, ondas sonoras, bem como sistemas de visão artificial. Um sistema capaz de automatizar esse processo de monitoramento do desgaste vem sendo estudado por vários pesquisadores há vários anos (BOMBONATO et al. 2008, FAVARETTO et al. 2009, GARCÍA-ORDÁS, 2017).

Sistemas de visão artificial, bem como análise de desgaste de ferramenta por imagem, deve constituir as seguintes etapas segundo GONZALES, 2009: aquisição, processamento e análise de imagens. Porém neste trabalho foi desenvolvido somente a parte de aquisição da imagem e será utilizado um *software* previamente desenvolvido para as etapas subsequentes de medição do desgaste. Com isto, objetivou-se desenvolver e testar a acurácia de um braço robótico para aquisição de imagens automaticamente para fins de desgaste de flanco de ferramenta de usinagem em um torno mecânico convencional. Para isto será projetado um mecanismo de barras articuladas automaticamente, como uma câmera microscópica, e controlado por placa única. Bem como manter a acurácia e repetitividade das imagens durante um possível acompanhamento do desgaste.

2. MATERIAL E MÉTODOS

O mecanismo robótico para realização do posicionamento da câmera partiu do conceito da luminária localizada no torno mecânico que tem barras articuladas que facilitam o posicionamento adequado para visualização da ferramenta de corte e da peça usinada. Deste modo, a Figura 2 mostra o desenho conceitual do braço robótico e a peça de fixação dos motores de passo à base à guia linear. As peças H e I foram projetadas de material em aço, tendo em vista que não afetam no cálculo do momento fletor nas barras e torque nos motores. As demais, que sofrem movimento, foram de alumínio, devido ao baixo peso. As demais, que sofrem movimento, foram de alumínio, devido ao baixo peso.

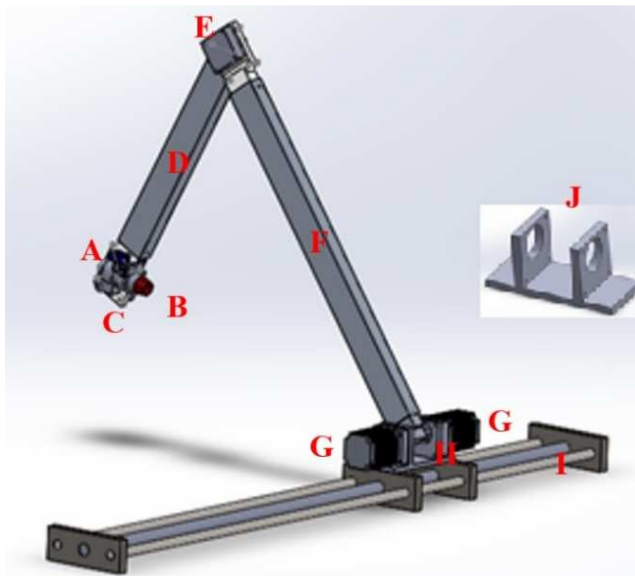
2.1 Estrutura Metálica

O dimensionamento das barras estruturais do braço robótico foi feito considerando a condição estática e as tensões devido à flexão, desconsiderando torção e força cortante, tendo em vista que o momento fletor rege o valor das forças nas barras. Deste modo, foram escolhidos como elementos estruturais tubos de alumínio com tensão admissível de escoamento (σ_{adm}) de 35 MPa. Assim, escolheu-se um perfil de alumínio que possui valor superior ou igual ao seu momento resistente W (I/C), no qual W foi calculado pela relação entre o momento fletor máximo (M) e a tensão admissível $W = M / \sigma_{adm}$ (BEER & JOHNSTON, 2006).

2.2 Motores de Passo

Ao analisar o dimensionamento dos motores, considerou-se que as barras estavam posicionadas na horizontal e engastadas em uma das extremidades, pois essa é a pior condição de trabalho. Estes diagramas serviram tanto para cálculo do torque necessário para escolha do motor, quanto para o cálculo estrutural das barras de alumínio. Neste caso, o torque do motor causa momento fletor nas barras (BEER & JOHNSTON, 2006).

O peso considerado para o perfil de alumínio utilizado foi 10N/m, o peso da câmera na extremidade do braço foi 3N e o peso dos motores foi 12N. A seguir, a Figura 3 mostra o diagrama de corpo livre do braço robótico engastada na extremidade.



Legenda: **A** - Motor do microscópio (Motor 4); **B** - Microscópio digital; **C** - Barra 3; **D** - Barra 2; **E** - Motor de ligação entre as Barras 1 e 2 (Motor 3); **F** - Barra 1; **G** - Motores da base (Motor 2); **H** - Peça de suporte dos dois motores da base; **I** - Guia linear por fuso; e **J** - Detalhe da peça H,

Figura 2- Esquema do conceito do braço robótico para posicionamento da câmera em um torno mecânico.

Fez-se um somatório de momentos para uma situação estática (BEER & JOHNSTON, 2006), obtendo a reação de momento do ponto de engaste M_1 , M_2 e M_3 necessárias de 28,55 N.m., 7,5N.m e 0,35 N.m, respectivamente. As peças de ligação entre barras e motores de passo foram construídas por usinagem e manufatura aditiva no Departamento de Engenharia de Produção e Mecânica (DEP).

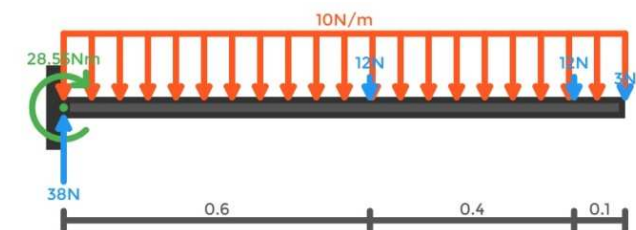


Figura 3- Diagrama de corpo livre das Barras B1, B2 e B3 na posição de maior demanda de torque dos motores. As forças de 12 N são referentes ao peso dos motores de passo, já a de 3 N é referente à câmera e 10 N/m é o peso das próprias barras.

Para escolher os motores a serem utilizados foram usados os seguintes critérios: Torque mínimo com um fator de segurança igual a dois, ou seja, 57,1 N m, 15 N m e 0,7 N; Compatibilidade dos modelos disponíveis no mercado; Compatibilidade de tensão e corrente elétrica; e peso dos motores.

2.3 Programação e controle do braço

O sistema de malha fechada do manipulador robótico foi dividido em três algoritmos. O primeiro desses códigos é responsável por atuar diretamente sobre os sistemas eletrônicos do braço, ou seja, executar a tarefa de enviar pulsos e outros comandos diretamente para drivers *Leadshine M542*. Esse tem a função de controlar os motores de passo do mecanismo em questão, além de ler os sensores de fim de curso e tratar os comandos de movimento, tanto do usuário quanto os que são relacionados ao fechamento da malha. Para controle do braço robótico, utilizou-se botões sem trava (“pushdown”) e chaves alavanca pulsantes instalados em uma base metálica. Tal escolha foi justificada pela simplicidade e facilidade para o operador identificar os controles (HU & CHEN, 2012). Enquanto os demais controles foram realizadas na própria plataforma de programação do Arduino baseada em linguagem C. conhecida como ‘*Integrated Development Environment*’, IDE (Arduino, 2020). Ainda sobre o primeiro código, foi executado na placa única ArduinoMega 2560 e armazenado na sua própria memória flash.

Já o segundo código, trata-se de uma função desenvolvida na plataforma de programação Matlab. O objetivo foi o tratamento em tempo real de cada frame do vídeo capturado na câmera durante todo o processo, que retorna os dados e permitem identificar o posicionamento com base em um ponto. Este ponto foi definido previamente no alvo da imagem, neste caso o flanco da ferramenta de corte.

Ademais, o terceiro código, que também foi executado no Matlab, utiliza a função do segundo código e tem a finalidade de atuar como um elo entre o processamento da imagem da câmera, feita através da função discutida e o ArduinoMega que controla os drivers do braço. Para isso, foi utilizado uma segunda placa única Arduino UNO que recebe os comandos do Matlab pela porta serial (USB) e os repassa ao ArduinoMega.

A Figura 4 demonstra uma simplificação dessas interações entre os dispositivos no momento em que o sistema inicia o processo de autocorreção da posição da câmera em malha fechada, vale salientar que esse ciclo é quebrado quando a posição alvo é obtida e a imagem é armazenada, neste caso, para análise de desgaste de flanco da ferramenta.

2.4. Avaliação do protótipo

Define-se como exatidão, no sentido particular aplicado em robótica, como sendo o quão próximo se obtém a posição-orientação real do efetuator do valor predito pelo controlador do robô dentro do volume de trabalho. Por outro lado, a repetitividade pode ser definida como a capacidade do efetuator de retornar a uma posição-orientação que já tinha sido atingida repetidas vezes (VERGEEST & TANGELDER, 1996).

Para quantificar a exatidão, é necessário definir a exatidão ou acurácia unilateral de posicionamento (AP), que corresponde à diferença entre a posição comandada e o centroide do conjunto de pontos alcançados. Logo, quanto maior AP menor a exatidão do sistema.

Por outro lado, para quantificar a repetitividade, é necessário definir a repetitividade unilateral de posicionamento (RP), que corresponde ao raio do círculo centrado no centroide do conjunto de pontos alcançados. Assim, quanto maior RP

menor a repetitividade do sistema. A Figura 5 mostra a diferença entre repetitividade e exatidão.

A fim de avaliar o desempenho do braço sem o sistema de malha fechada, foram feitas nove repetições, em que todas elas são baseadas em uma única imagem de referência. Para esse teste, o flanco da ferramenta de corte foi mantido estático e, por meio do braço controlado por um joystick, manualmente posicionava a câmera até atingir o foco de um ponto verde no flanco da ferramenta. Utilizando o programa desenvolvido para repetir a imagem alvo inicial foram realizadas nove vezes automaticamente. Os resultados das coordenadas do centroide da imagem foram registrados para comparação com a imagem alvo obtida inicialmente.

No posicionamento automático por retroalimentação, os códigos criados para esta função utilizaram da imagem alvo inicial, o ponto verde no flanco. Para aquisição de uma nova imagem e com o mesmo foco, inicialmente criava um quadrado circunscrito no ponto verde localizado no flanco (alvo), no qual serviu de ponto de referência na imagem. Assim era salva a próxima imagem somente quando os movimentos dos braços e a guia linear atingissem novamente o centroide definido na imagem alvo.

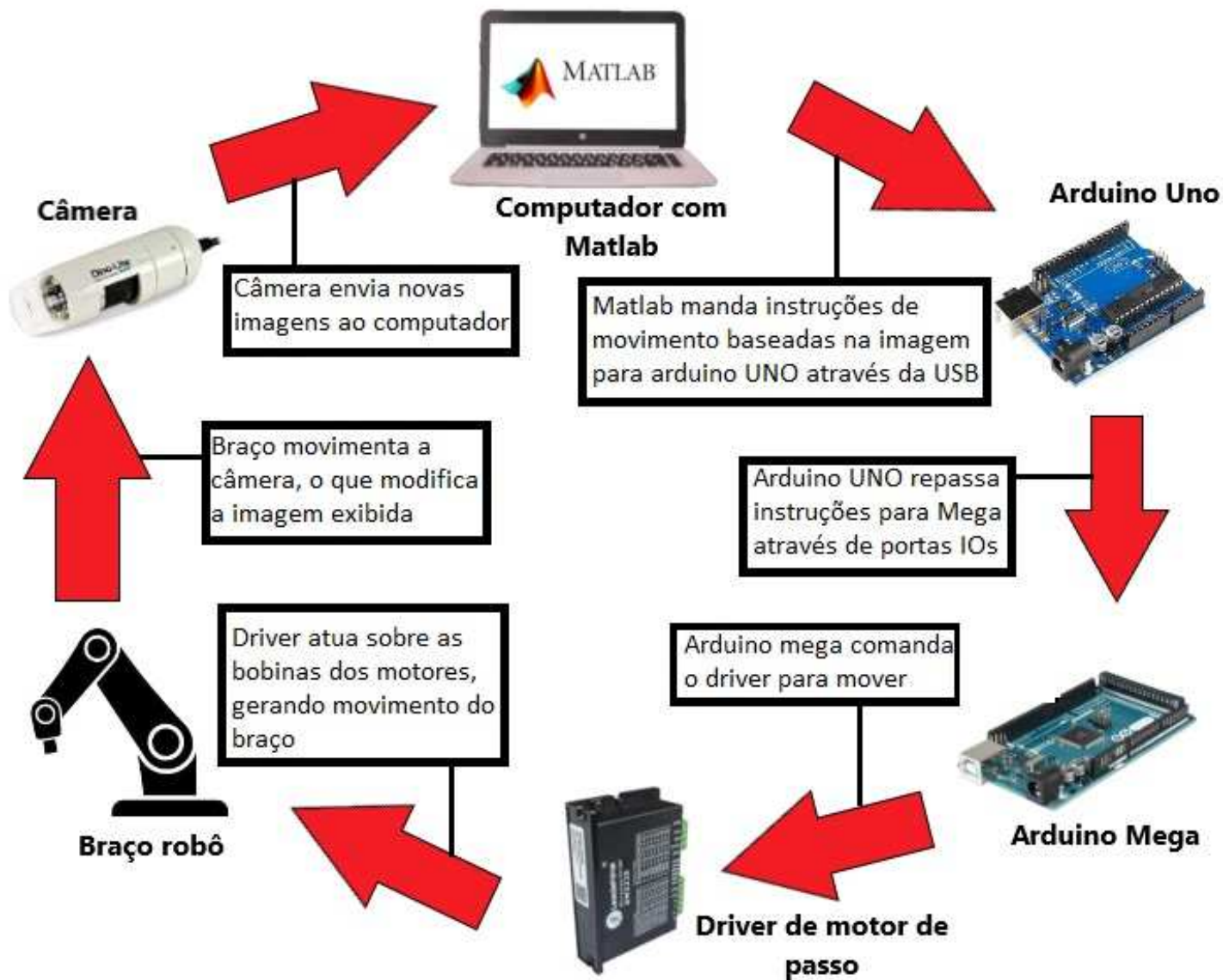


Figura 4 – Esquema do funcionamento do braço robótico na etapa de autocorreção em malha fechada e os componentes utilizados no sistema.

Em suma, o grande interesse é saber o quão eficaz é essa técnica de autocorreção por retroalimentação da posição da câmera, que utiliza o centroide e as fronteiras da região para manter a próxima imagem no mesmo foco. Uma forma de fazer a comparação entre antes e depois de fechar a malha de controle é utilizando o coeficiente de correlação entre cada uma das repetições e a imagem alvo. Nesse ínterim, o coeficiente de correlação 2D é um número que representa a similaridade entre duas imagens levando em consideração as respectivas intensidades dos pixels (KIM, 2011) e sua expressão é usada pela rotina do Matlab 'corr2'. Para esse teste de correlação, o flanco da ferramenta foi preservado sem alterar o desgaste e as condições de iluminação foram mantidas constantes durante a aquisição das imagens.

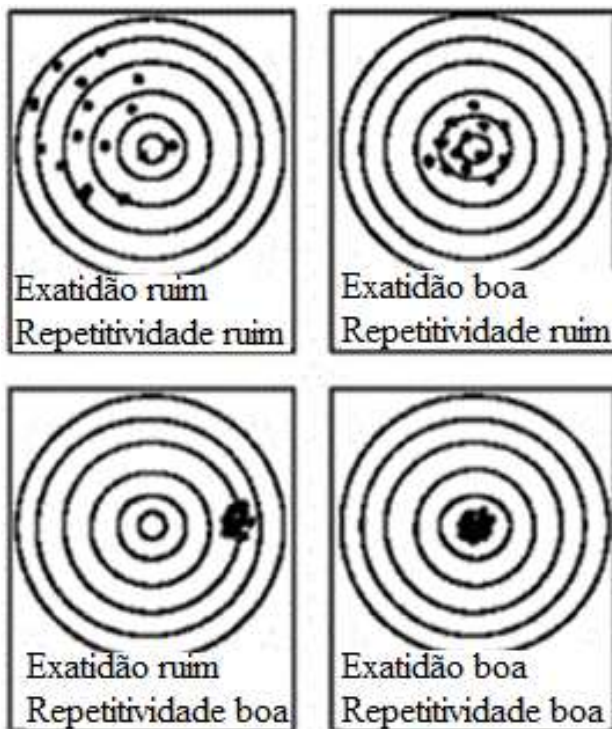


Figura 5 - Repetitividade versus Exatidão (acurácia). Fonte: GROOVER, 1989

3. RESULTADOS E DISCUSSÃO

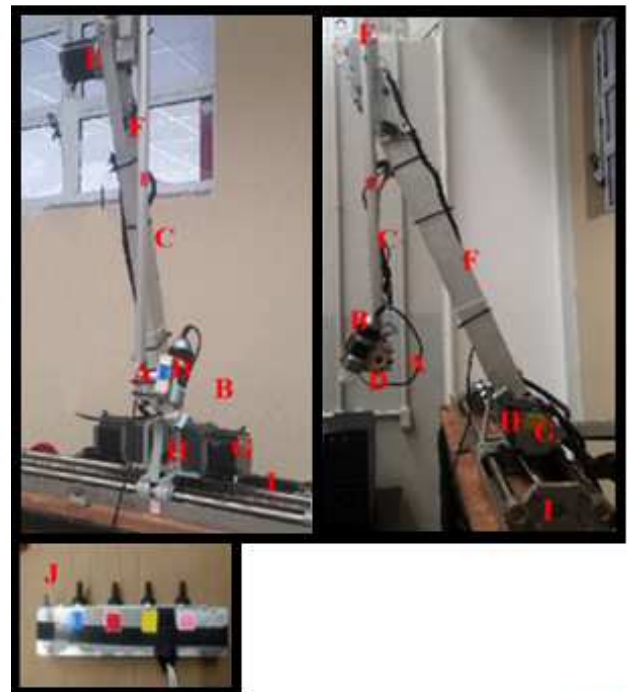
O perfil de alumínio escolhido para fabricação da estrutura do braço foi a Liga 1100-H11 (99% Al), que possui os limites de tensão de escoamento de 70 Mpa, com $W \geq 323,6 \text{ mm}^3$ e uma seção retangular de 23 por 24,5 mm e 1mm de espessura. As peças de ligação entre barras e motores de passo foram construídas por usinagem e manufatura aditiva no Departamento de Engenharia de Produção e Mecânica (DEP).

Visando a fixação dos motores de passo à base, foi projetada uma peça de aço ABNT 1020, mostrado na Figura 6, que possui um furo central para passagem do eixo do motor e quatro furos nas extremidades para fixação dos motores.

As ligações dos motores da base com a barra um e do motor dois e três com a barra dois foram usinados em liga de alumínio 6061-T6. Já a conexão entre a barra dois e o motor quatro, bem como a barra três (corpo do microscópio digital) foram construídos por manufatura aditiva em plástico ABS. As demais peças como parafusos, porcas, arruelas e abraçadeiras são todas comerciais.

Os motores foram definidos da seguinte forma:: para movimentar a barra três, que necessitava de ao menos 0,7N.m, foi escolhido um de motor de Passo NEMA 23 de 1,5Nm / 2,1 A; para movimentar a barra dois, que necessitava de ao menos 15 Nm, foi escolhido um motor de passo com caixa de redução NEMA 23 de 15 Nm/ 2,1 A; para o barra um, que necessitava de ao menos 56,1 N.m, foram escolhidos dois motores de passo com caixa de redução NEMA 23 de 30 Nm/ 2,1 A. Além disso, os motores de passo utilizados neste trabalho são controlados com o auxílio de cinco drivers M542 (um para cada motor). Tanto os motores quanto os drivers são da marca Neoyoma.

A Figura 6 ilustra a montagem final do protótipo em duas vistas e o detalhe do controle manual.



Legenda: **A** -Motor do microscópio (Motor 4); **B** - Microscópio digital; **C** - Barra 3; **D** - Barra 2; **E** - Motor de ligação entre as Barras 1 e 2 (Motor 3); **F** - Barra 1; **G** - Motores da base (Motor 2); **H** - Peça de suporte dos dois motores da base; **I** - Guia linear por fuso; e **J** - Detalhe Joystick de controle, as marcas coloridas correspondem às barras do braço.

Figura 6- Visão geral do protótipo em duas vistas, o microscópio digital na ponta do manipulador, a estrutura metálica formada por barras de alumínio e a guia que permite movimento linear por fuso. O detalhe J, o Joystick e as cores representando cada barra para controle manual dos motores.

Para controle manual do braço robótico, utilizou-se botões sem trava (“pushdown”) e chaves alavanca pulsantes instalados em uma base metálica conforme Figura 7 J. Tal escolha foi justificada pela simplicidade e facilidade para o operador identificar os controles (HU & CHEN, 2012), enquanto os demais controles automáticos como posicionamento até encontrar o foco e retorno da câmera para a posição Home são feitos automaticamente na própria plataforma de programação IDE do Arduino.

A Figura 7 mostra o fluxograma do código que é executado no ArduinoMega, ou seja, trata-se da lógica de funcionamento que interage diretamente com os drivers de motores de passo e recebe os comandos do usuário manualmente.

Como o objetivo deste trabalho foi utilizar o ajuste por retroalimentação a fim de melhorar a repetibilidade, o programa inicia a tarefa de retroalimentação da posição por meio de processamento do foco na imagem alvo. Sendo assim, após esse retorno até a posição gravada considerando somente o número de passos, o ArduinoMega envia um sinal ao Arduino Uno que é transferido ao Matlab, indicando o momento de iniciar o procedimento de autocorreção em malha fechada. Nesse sentido, o ArduinoMega recebe sinais provenientes do UNO2 em seis de suas portas que indicam se a câmera deve mover-se para esquerda, direita, para cima, para baixo, se aproximar ou se afastar.

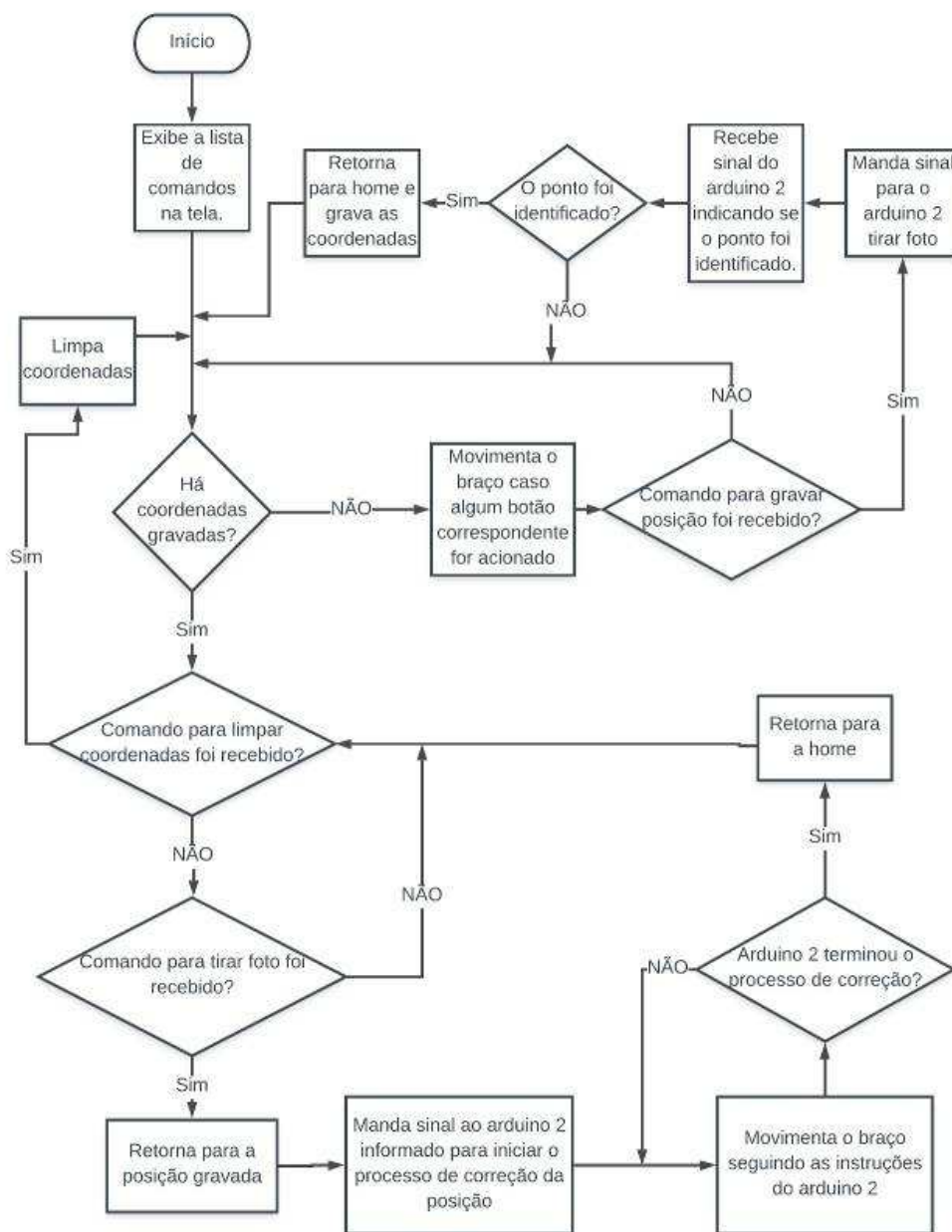


Figura7 - Fluxograma do código um que é executado no ArduinoMega com objetivo de comandar drivers dos motores de passo, receber e tratar comandos do usuário ou sinais do Arduino Uno 2 (executa comando enviados do posicionamento automático).

Após o Arduino Uno sinalizar que o processo de correção terminou, ele envia um comando para o ArduinoMega retornar à posição home e preservar as coordenadas previamente gravadas. Por fim, o usuário pode consultar a imagem do novo estado da peça, que estará salva e pronta para iniciar o programa de monitoramento do desgaste. A Figura 8 mostra o

fluxograma relativo ao segundo código, que tem o objetivo de tratar as imagens do vídeo da câmera em tempo real. Inicialmente, a função recebe como entrada um *frame* (imagem do vídeo) a ser processado naquele instante, obtido da câmera pela função “*getsnapshot*”, além de coordenadas referentes a posição alvo.

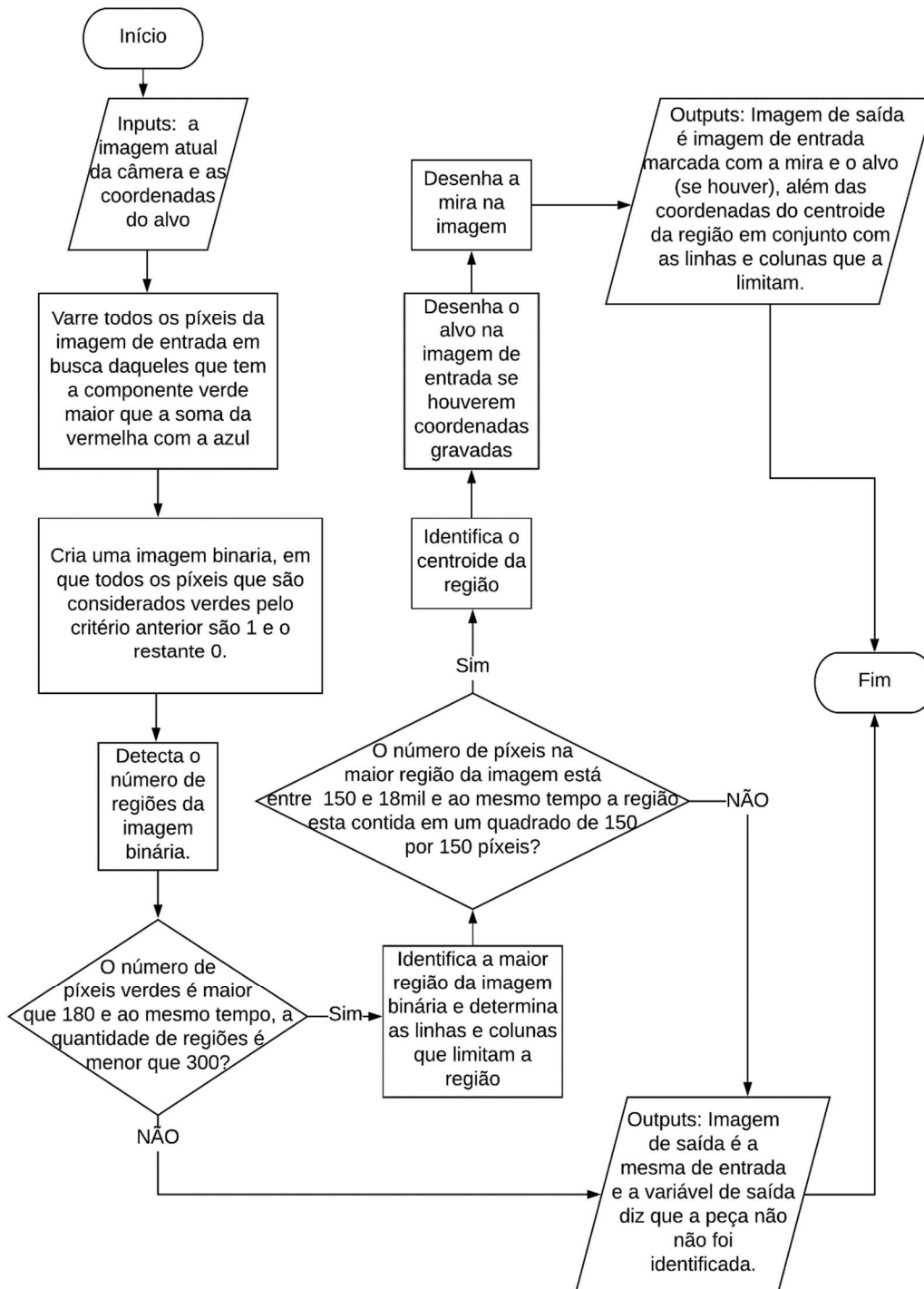


Figura 8 - Fluxograma do código dois, referente a função desenvolvida para processamento de imagens da câmera em tempo real com objetivo de identificar a presença de um ponto na imagem a fim de orientar o processo de autocorreção.

Em seguida, a etapa de identificação de um ponto verde, usado de alvo, presente no flanco da ferramenta. Esta etapa identifica o ponto e segmenta o ponto em uma imagem binária, na qual todas as posições referentes aos pixels que possuem componente verde maior que a soma da componente vermelha com a azul, são atribuídos valores 1 (branco), enquanto nas demais posições, preenche-se com zeros. Tal critério de classificação foi decidido com base em observação a priori de imagens da ferramenta marcada com o ponto verde (Figura 9).

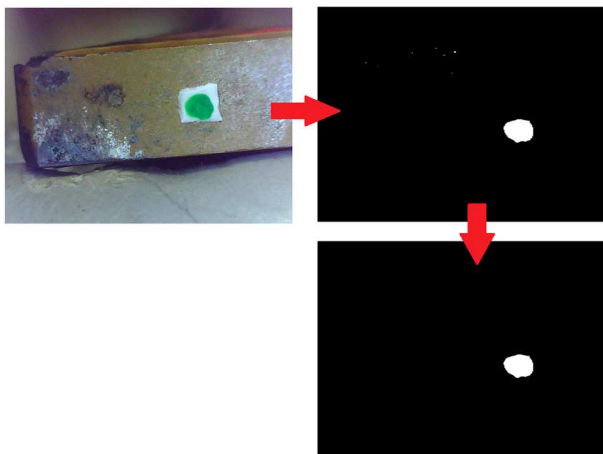


Figura 9 – Resultados da sequência do processamento feito pela função para detecção do ponto, primeiro a imagem é convertida em binária e depois o ruído é eliminado.

Em sequência, cria-se uma região de pixels conexos e adjacência horizontal e vertical para vizinhança-4 ou diagonais (vizinhança-8) (GONZALEZ, 2009). Neste processo utilizou técnicas para eliminar ruídos como definindo um número de pixels mínimos verdes identificados para ser considerado um objeto. Esta região agrupada pelos vizinhos verdes mais próximos e inscrita em um quadrado de 150 por 150 pixels, que servirá de alvo inicial. Após identificado o ponto para o ajuste das próximas imagens, utiliza-se o seu centroide, que foi calculado usando a função “*regionprops*” na imagem originalmente configurada pelo usuário como alvo. A função “*regionprops*” retorna propriedades das regiões de uma imagem e ao receber as informações da região junto com a “*stringCentroid*” retorna a coordenada do centroide do quadrado verde.

Para finalizar, a função retorna a imagem marcada para que ela seja exibida na tela de modo que o usuário possa acompanhar em tempo real a identificação do ponto e o processo de correção. Em adição, a função retorna também as coordenadas do centroide do ponto na imagem e as limitações laterais e verticais da região do ponto, informações que serão usadas pelo terceiro código para determinar para onde a câmera deve mover-se (Figura 10).

O terceiro código, que tem o objetivo de comunicação por meio da porta USB entre o *Matlab* e o *Arduino Uno2* é realizada por meio das funções do “*MATLAB SupportPackage for Arduino Hardware*”, que é um pacote que contém as funções utilizadas como, “*writeDigitalPin* e *readDigitalPin*” responsáveis respectivamente por escrita e leitura do estado lógico dos pinos digitais do *Arduino Uno2*.

Assim que o comando de gravar vindo do *ArduinoMega* é recebido pelo *Arduino Uno2* e transferido ao *Matlab*, esse programa determina se o ponto está ou não presente no frame atual e caso esteja presente, o frame é salvo como imagem de referência e as informações de coordenadas e limitações do ponto na peça são salvas em variáveis. Em seguida, o código determina que seja enviado um sinal ao *ArduinoMega* informando que a imagem alvo foi adquirida e executa o retorno para posição home. Se o ponto não for identificado no momento que o usuário registrar a imagem, o sinal de retornar para home. Uma vez que a imagem de referência foi adquirida e as informações sobre ela foram armazenadas, o programa permanece atualizando a imagem exibida na tela e as variáveis do ponto, acessando a função do código 2. Então, um loop continua executar na plataforma do *arduinoMega* e sinalize que o usuário requisitou a limpeza das coordenadas ou solicitou uma nova imagem nas coordenadas já gravadas.

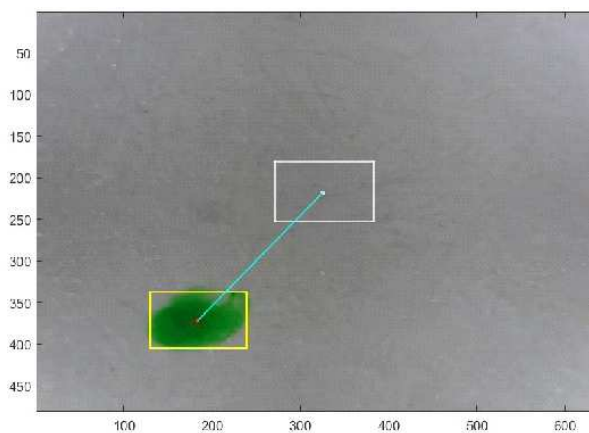


Figura 10 - Exemplo de reposicionamento do braço robótico até a mira em amarelo para encontrar o alvo inicial, neste caso o quadro branco, o programa encontra o reposicionamento inicial quando os quadros se sobrepõem, coordenadas em pixels.

Caso o usuário solicite a nova imagem, o *arduinoMega* primeiramente irá voltar em malha aberta até dar o número de passos armazenados em variáveis e então enviar um sinal ao *Arduino Uno2*. Assim, O código do fluxograma da Figura 11 vai receber o comando para iniciar a correção da posição. Após receber o comando, o processo de correção se inicia através da comparação das coordenadas do centroide do ponto na imagem alvo com as coordenadas do ponto na

Durante todo o procedimento, novas imagens são coletadas na câmera atualizando as variáveis de posição atual, até que coincidam com o alvo. Uma vez que o centroide está na posição, os limites da região verde de ambos a imagem alvo e a atual são comparados. O afastamento da câmera é corrigido até que os limites dessa região coincidam, pois caso a câmera esteja muito próxima, o ponto parecerá maior e os limites verticais e horizontais da região demarcariam uma área maior na imagem do que deveria e vice-versa. Finalmente, quando o centroide e os limites estão alinhados com o alvo, uma nova imagem é adquirida e armazenada,

Deve-se ressaltar que foi feito o uso de duas placas Arduino ao invés de uma, utilizando um Arduino Uno2 para repassar informações provenientes do Matlab ao ArduinoMega ao invés de transferir diretamente do Matlab ao ArduinoMega. Tal decisão é motivada em razão de que uma das melhorias propostas para o braço foi de aumentar a precisão no movimento, fazendo com que uma volta do motor seja dividida em um número maior de passos, para aproveitar todo o potencial da tecnologia de micropasso que os drivers oferecem. Isso permitiu uma maior suavidade nos movimentos, dividindo uma volta em até 25 mil passos. Para isso, faz-se necessário que o Arduino enviasse 25 mil pulsos, um para cada passo para completar uma volta no motor, que ainda é acoplado em uma caixa de redução de 10x.

Logo, foi necessário otimizar ao máximo o código no ArduinoMega. Como esse microcontrolador que executava o código 2, não era interrompido a cada pulso para receber informações pela porta USB. Ao invés disso, preferiu-se o uso de uma placa Arduino Uno 2 dedicada exclusivamente a tarefa de interagir com o Matlab e deixar a informação de movimento diretamente em múltiplas IOs (Pinos de entrada e saída) do ArduinoMega. Acresce que a operação para ler as "IOs" foi constatado como consideravelmente mais rápida que receber informações do Matlab pela porta serial (USB). Todas as rotinas estão na forma de fluxograma nas Figuras 7, 8 e 11.

Sendo assim, a divisão das tarefas entre dois arduinos distintos foi a melhor maneira encontrada para otimizar, de modo que o Arduino Uno2 seja responsável apenas por esperar informações pela serial e as transmitir de volta sem ter a função de enviar milhares de pulsos aos drivers.

Nesse sentido, a Figura 12 mostra o resultado do funcionamento do braço em suas etapas. Na imagem assinalada com (1), ainda não há nenhuma posição gravada, o que possibilita o usuário mover o braço a fim de achar o posicionamento mais conveniente na imagem para visualização do desgaste e posterior medição do mesmo.

Além disso, esse usuário pode constatar o momento em que o ponto verde foi devidamente enquadrado em razão da mira que é desenhada pela função do segundo código sobre o ponto verde. Uma vez que isso ocorre, significa que está liberada a gravação daquela imagem como referência ou alvo definido pelo usuário. Assim que o usuário definir a imagem alvo, as coordenadas correspondentes são armazenadas e essa imagem é salva como referência para as demais imagens que serão adquiridas posteriormente, mantendo a repetitividade do foco.

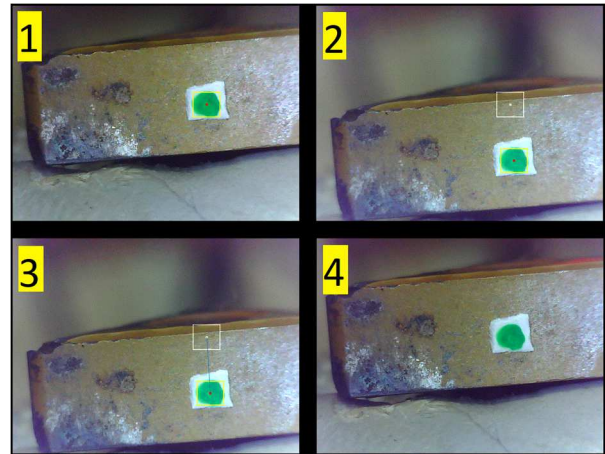


Figura 12 - Etapas do funcionamento do sistema de retroalimentação: (1)-ponto identificado e imagem salva; (2)-tentativa de retorno em malha aberta a posição gravada; (3)-Início do processo de autocorreção; (4)-imagem salva como resultado do processo de autocorreção por retroalimentação da posição.

A Figura 12-2 ilustra o momento após o usuário acionar a aquisição de uma nova imagem na mesma posição da etapa (1). Na etapa 2, se não houvesse autocorreção o braço adquiriria a imagem posicionada diferentemente da alvo, neste caso abaixo da referência, ainda em malha aberta. Nesse instante é possível notar que houve um erro entre a posição alvo e a posição obtida, a posição alvo é indicada pelo quadrado branco (alvo). A posição atual do ponto é indicada pelo quadrado amarelo com centroide marcado em vermelho (mira). Esse erro obtido foi justamente a motivação do trabalho no sistema de malha fechada.

Seguindo para a terceira etapa (Figura 12-3), observa-se o desenho de uma linha azul unindo o centroide do alvo com a mira. Quando essa linha aparece significa que o sistema iniciou o processo de autocorreção da posição e está operando em malha fechada. Tal marcação foi inserida na imagem com o objetivo de permitir que o usuário seja capaz de acompanhar em tempo real em que etapa a execução do código se encontra.

Por fim, a etapa (4) da Figura 12-4 mostra o exemplo do resultado do processo de autocorreção, ou seja, a imagem que foi salva, juntamente com a imagem de referência tirada na etapa (1). Observa-se que o posicionamento após a retroalimentação está nitidamente mais próximo da imagem de referência (Figura 12-1) do que antes da autocorreção (Figura 12-2).

Para validar a repetitividade e eficiência do manipulador robótico, foi feito um experimento no qual foram gravadas as posições do centroide da imagem em quatro situações diferentes.

Assim, a Tabela 1 mostra as coordenadas obtidas em cada uma das repetições com o manipulador robótico trabalhando em malha aberta. As coordenadas da imagem 1 (Tabela 1) é a imagem de referência, ou seja, é a posição alvo definida pelo usuário que se busca repetir. Já as demais são as coordenadas das imagens subsequentes antes da retroalimentação a ser executada.

Tabela 1 - Coordenadas em pixels dos centroides do ponto verde nas fotos sem feedback e sem distúrbio.

Imagem	1	2	3	4	5	6	7	8	9	10
Linha	221	342	338	336	337	337	342	342	343	346
Coluna	434	442	436	442	437	442	436	442	437	442

Com esses resultados pode-se observar que existiram diferenças consideráveis entre a imagem alvo (1) e as imagens adquiridas posteriormente (2 a 9).

A Tabela 2 mostra os coeficientes de correlação utilizados para avaliar a repetibilidade e exatidão do posicionamento da câmera comparando o sistema em malha aberta (Tabela 1) e malha fechada, observa-se que as imagens são em torno de três vezes mais bem correlacionadas no sistema de malha fechada.

Tabela 2 - Tabela comparativa para os coeficientes de correlação entre as repetições e a imagem 1 de referência, antes e depois do feedback para o caso sem distúrbio da peça.

Foto	2	3	4	5	6	7	8	9	10	Média
Coef. de correlação antes do <i>feedback</i>	0.39	0.38	0.37	0.36	0.35	0.34	0.34	0.34	0.33	0.36
Coef. de correlação após <i>feedback</i>	0.99	0.99	0.98	0.93	0.96	0.96	0.94	0.92	0.96	0.96

Em seguida, a Tabela 3 mostra o mesmo caso da Tabela 1, mas com a diferença que dessa vez houve perturbação proposital por vibrações na peça, o que causa perda da precisão do braço.

Tabela 3 – Excitação do sistema para provocar vibrações e as coordenadas dos centroides do ponto de referência nas imagens sem e com retroalimentação.

Imagem	1	2	3	4	5	6	7	8	9	10
Linha	278	441	448	467	417	457	418	461	462	455
Coluna	449	550	384	401	273	445	419	444	446	556

Em adição, a Tabela 4 demonstrou como a atuação da autocorreção melhora a correlação entre as repetições sucessivas e a imagem alvo mesmo quando há distúrbio ou perturbações externas no braço, notando-se que a média antes de fechar a malha era de 21% e passou a 97,6% após a operação do sistema de retroalimentação. Acresce que as vibrações diminuíram o coeficiente de correlação médio.

Tabela 4 – Comparação dos coeficientes de correlação entre as repetições e a imagem de referência, antes e depois da retroalimentação para o caso com excitação de vibração na peça.

Foto	2	3	4	5	6	7	8	9	10	Média
Coef. de correlação antes do <i>feedback</i>	0.12	0.19	0.15	0.30	0.18	0.29	0.17	0.18	0.28	0.21
Coef. de correlação após <i>feedback</i>	0.97	0.97	0.98	0.97	0.98	0.97	0.98	0.97	0.96	0.97

Finalmente, verificou-se que o ponto alvo verde foi encontrado pela rotina de segmentação em todos os testes feitos. Para que esta localização do alvo seja possível, o erro do braço em malha aberta necessita localizar no alvo inicial no mínimo o ponto verde no flanco da ferramenta.

Portanto, uma melhoria futura possível seria a implementação de velocidade variável nos movimentos automáticos do braço durante a etapa sem feedback, de um modo que comece bem lentamente e aumente progressivamente a velocidade, mas que volte a reduzi-la quando estiver próximo a parar. Isso evita que o motor perca passos em razão de não conseguir vencer a própria inércia ou da oscilação no momento de parar. Isso aumentaria a certeza de encontrar a ferramenta de corte com o ponto verde antes da autocorreção.

4. CONCLUSÃO

Com o objetivo de construir um mecanismo controlado eletronicamente, equipado com uma câmera microscópica para aquisição de imagens da ferramenta de usinagem, foi desenvolvido um mecanismo nomeado de braço robótico. Conclui-se no seu desenvolvimento a necessidade de utilizar motores de passo nas articulações de barras estruturais em perfis de alumínio e placa única ou microcontroladores do tipo Arduino. Como o sistema em malha fechada do posicionamento da câmera obteve coeficiente de correlação de 97% entre as imagens alvos e adquiridas automaticamente. Pode-se concluir que o sistema com fechamento da malha permitiu corrigir as folgas dos componentes mecânicos e adquirir imagens com precisão e acurácia altas.

AGRADECIMENTOS

Os autores agradecem à Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) e Universidade Federal de Viçosa (UFV) ao apoio fornecido para a realização do presente trabalho.

REFERÊNCIAS

- ARDUINO GLOSSARY. Creative Commons, Mountain View, California, USA, Available at: <<https://www.arduino.cc/glossary/en/>>. Accessed on January 3rd, 2020.
- BEER, P. F.; JOHNSTON, E. R. Jr.; DEWOLF, J. T. Resistência dos Materiais, 4 ed. Editora Mc Graw Hill, 2006.
- BOMBONATO, S.G.; LUSTOSA, A.A., SILVA, M.B. Análise do desgaste de ferramentas via emissão acústica com aplicações de redes neurais. **Horizonte Científico**, vol2, n.2, 24 p., 2008.
- CHRYSSOLOURIS, G. Manufacturing systems –theory and practice, 2nd ed. New York: SpringerVerlag, 2006.
- FAVARETTO, A.S.; VALLE, P.D., CANGIOLIERI JUNIOR, O. O gerenciamento de ferramentas de corte na indústria automotiva: um estudo de casos na região metropolitana de Curitiba. **Produto & Produção**, vol. 10, n. 3, p. 45 - 60, 2009.
- GARCÍA-ORDÁS, M.T.; ALEGRE, E., GONZÁLEZ-CASTRO, V. et al. A computer vision approach to

- analyze and classify tool wear level in milling processes using shape descriptors and machine learning techniques. **The International Journal of Advanced Manufacturing Technology**, vol. 90, p. 1947-1962, 2017, <https://doi.org/10.1007/s00170-016-9541-0>.
- GONZALEZ, R. C.; WOODS, R. E. Processamento digital de imagens, 3 ed. Editora Pearson, 2009.
- GROOVER, M. P. Robótica, Tecnologia e Programação. McGraw-Hill, 1989.
- HESSER, D. F.; MARKERT, B. Tool wear monitoring of a retrofitted CNC milling machine using artificial neural networks **Manufacturing Letters**, Volume 19, p. 1-4, 2019.
- HOPKINS, C.; HOSSEINI, A. A Review of Developments in the Fields of the Design of Smart Cutting Tools, Wear Monitoring, and Sensor Innovation, **IFAC-PapersOnLine**, vol.52, Issue 10,p. 352-357, 2019.
- HU, Y. N. & CHEN, Y. H. Implementation of a Robot System for Sculptured Surface Cutting. Part 2. Finish Machining. **International Journal of Advanced Manufacturing Technology**, 2012.
- ISO3685, Ensaio de vida da ferramenta de ponta única para torneamento, 2017.
- KIM, T.; ADELI, H.; RAMOS, C.; KANG, B. Signal Processing, Image Processing and Pattern Recognition, 1 ed. Editora Springer, 2011.
- LUCAS, E. O. Desenvolvimento de um Dispositivo Quick-stop e sua aplicação no estudo da formação do cavaco na usinagem do ferro fundido nodular ferrítico. 2003. 151 P. Tese (Doutorado em Engenharia Mecânica) - Universidade Federal de Santa Catarina, Florianópolis, 2003.
- MatLab user's Guide, **The Math Works Inc.**, 2009.
- PRADO, A.; PEREIRA, J.A.; PÉREZ, T.G. Methodology for tool wear analysis by electrical measuring during milling of AISI H13 and its impact on surface morphology, **Procedia Manufacturing**. Vol. 13, p. 356-363, 2017.
- SANDVIK COROMANT. **Modern Metal Cutting – A Practical Handbook**.Suécia, 1 edition, 1994.
- VERGEEST, J. S., & TANGELDER, J. W. Robot machines rapid prototype. **Industrial Robot**, pp 17-20, 1996.
- YOUNG, H. D. & FREEDMAN, R. A. Física III: Eletromagnetismo, 12a. ed.Pearson, São Paulo, Brasil, 2009.