

Memorial do Projeto De Um Dispositivo Mecânico Autocontrolado Por *Encoder Óptico* e Controlado Remotamente Via *Bluetooth*

Caio Garcia Cancian, Gustavo Alencar Bisinotto, Luis Felipe Gomes de Oliveira

Resumo – Neste trabalho são descritas as etapas de desenvolvimento de um dispositivo proposto como projeto da disciplina de Introdução ao Projeto de Sistemas Mecânicos (PMR3202) da Escola Politécnica da Universidade de São Paulo (EPUSP). O dispositivo deveria recolher o maior número de objetos de uma região delimitada e transportá-los até um local estipulado durante determinado período de tempo, devendo realizar parte do circuito de forma autônoma. A máquina foi concebida de forma a atender uma série de requisitos, identificados a partir das restrições do projeto. Para sua execução, ao longo do projeto, fez-se: a avaliação das soluções elaboradas com o uso de uma Matriz de Decisão, dimensionamento dos componentes mecânicos através de formulações, projeto do sistema para desempenho autônomo da máquina, programação do microcontrolador para comando dos sensores e atuadores, fabricação e montagem completa. Pelos resultados obtidos foi possível concluir que a máquina atendeu de maneira aceitável aos requisitos e também identificar aspectos que poderiam ser aperfeiçoados para melhorar o desempenho do dispositivo.

Palavras-chave – Memorial de Projeto; Metodologia de Projeto Arduino UNO; *Encoder Óptico*; *Bluetooth*;

1 Introdução

O presente memorial descritivo expõe de forma detalhada todo o desenvolvimento de um dispositivo mecatrônico empregado para a execução das tarefas do projeto “Coleta de Recicláveis”, realizado na disciplina PMR3202, do 3º semestre do curso de Engenharia Mecatrônica da EPUSP. De modo que a partir das informações e da documentação fornecida, seja possível executar o projeto plenamente de maneira adequada.

São apresentadas todas as fases do ciclo de vida do projeto, desde a concepção, onde foram elaborados requisitos e possíveis soluções, até o planejamento, execução e finalização, sendo que nessa última etapa buscou-se aperfeiçoar determinados aspectos do projeto que não foram englobados ou identificados durante as etapas anteriores.

2 Objetivo e Restrições

O principal objetivo, conforme o enunciado do projeto, consistia na construção de um dispositivo capaz de recolher o maior número de objetos de uma região delimitada e transportá-los até um local determinado em um período de 180 segundos. A máquina deveria ser acionada eletricamente e comandada pelo microcontrolador embarcado na máquina, sendo que nos primeiros 60 segundos de cada rodada da competição o dispositivo deveria comportar-se de forma autônoma.

O projeto deveria atender às restrições relacionadas a materiais, dimensões e outros aspectos como a alimentação e sistema de controle do dispositivo. A máquina deveria ser construída utilizando-se apenas os materiais indicados na lista de materiais contida no enunciado do projeto, além de elementos de fixação específicos como parafusos, porcas, arruelas, e de rebites.

Quanto às restrições dimensionais, ao início de cada partida da competição, o dispositivo poderia ter no máximo as seguintes medidas de comprimento, largura e altura, respectivamente: 300 mm x 250 mm x 300 mm. Além disso, o peso máximo permitido era de 10 kgf, incluindo todos os componentes da máquina, sobretudo a placa eletrônica e fonte de alimentação elétrica.

A máquina deveria receber apenas energia elétrica, fornecida por uma bateria embarcada na máquina, com tensão máxima de 12 V e capacidade máxima de 1,3Ah. E em relação ao sistema de controle, era obrigatório o uso do microcontrolador (Arduino UNO[®]) em conjunto com a plataforma de acionamento e comunicação baseada em Arduino (PACA) desenvolvida pelo Departamento de Mecatrônica e Sistemas Mecânicos (PMR) da instituição, ambos fornecidos pela comissão organizadora da disciplina, sendo que um microcomputador deveria ser utilizado para comando do dispositivo estabelecendo comunicação serial com o microcontrolador via *Bluetooth*.

Fora essas restrições, o uso de motores DC (corrente contínua) era restrito a no máximo duas unidades, com corrente máxima de 1A para cada motor e não eram permitidas alterações químicas nos materiais.

3 Materiais e Métodos

Para o desenvolvimento sistemático do projeto, adotou-se a metodologia baseada nos conceitos de Pahl et al. (2007), de tal forma que foram após definidos o propósito do projeto e suas restrições, foram levantados uma série de parâmetros para o dispositivo, e os requisitos aos quais eles deveriam se adequar, a fim de, assim, obtermos um conjunto de soluções para o problema proposto [1]. Finalmente, utilizou-se o método da Matriz de Decisão para obter a melhor relação entre os requisitos e as soluções apresentadas.

3.1 Parâmetros e Pesos Relativos

Com base nos objetivos, restrições e demais informações do regulamento, foram adotados os seguintes parâmetros para o desenvolvimento do projeto:

- Peso;
- Velocidade;
- Estabilidade;
- Capacidade de carga;
- Velocidade de descarga;
- Segurança de descarga;
- Facilidade de manobrar;
- Facilidade de manutenção;
- Facilidade de montagem;
- Facilidade de fabricação.

Os pesos relativos foram dados, dois a dois, de acordo com a importância de um dos parâmetros quando comparado a outro. Assim, distribuíram-se os pesos de acordo com a seguinte escala:

- 1 a 3: pouco mais importante;
- 3 a 6: mais importante;
- 7 a 9: muito mais importante.

Com uma faixa de 3 pontos por escala para tentar captar eventuais nuances dentro da própria categoria escolhida.

3.2 Requisitos

A seguir estão listados os principais requisitos formulados:

- Atender às restrições do projeto: vale observar que das restrições de peso máximo (10kgf) e volume (300 mm x 250 mm x 250 mm), definidas no enunciado, faz-se necessário buscar sua minimização, quanto ao peso, para otimizar o desempenho dos motores, e quanto ao vo-

- lume, quanto menor, maior será a facilidade de manobrar da máquina;
- Capacidade de carga mínima de 360g: valor base obtido a partir da aproximação do peso médio de 6 unidades dos itens bolas de tênis ou cubos de madeira, que se pretende transportar sobre a máquina ou em eventuais compartimentos, além dos objetos a serem empurrados, garantindo uma maior capacidade de carga e mobilidade, uma vez que a máquina se movimentará mais livremente tendo parte da carga sobre si, ao invés de unicamente empurrada;
 - Desempenho autônomo por 60s: com base nas regras da competição, a máquina deve ser capaz de se manter autonomamente durante os primeiros 60s de cada rodada. Estima-se completar nesse período um percurso que seja no mínimo da partida até o cesto 1, carregando os itens encontrados durante o trajeto;
 - Velocidade mínima de 5mm/s: valor base obtido a partir do percurso mínimo desejado para a parte autônoma, saindo do ponto de partida, passando pela região em “H” e chegando até o cesto 1, onde serão depositados os itens já recolhidos. Esse percurso teve base de tempo de 60s, o que implicaria numa pontuação razoável na parte autônoma devido ao seu peso nos pontos finais;
 - Montável com rebites, parafusos, ajustes e eventualmente adesivos: a facilidade de montagem é essencial para maior simplicidade da máquina, e sendo usados métodos de fixação não permanentes, será tanto melhor para corrigir pequenos erros durante a montagem, quanto para uma manutenção rápida durante a competição;
 - Fabricável com peças em formas geométricas elementares (como retângulos, triângulos e círculos): haja vista que a facilidade de fabricação implicará em maior precisão nas peças e conseqüentemente maior facilidade na montagem, além da possibilidade de se fazer peças de reposição para eventuais danos ocorridos durante a competição ou testes;
 - Máximo de 10s para depositar os itens recolhidos: tendo coletado pelo menos 3 itens sobre si, além daqueles empurrados, a máquina deve ser capaz de depositá-los todos no cesto e retornar à condição de movimento de percurso em até 10s, garantindo maior agilidade e tempo para o restante do trajeto.

3.3 Matriz de Decisão

Com base nos parâmetros e requisitos apresentados, foram elaboradas um total de 9 soluções distintas (apêndices de A a I), as quais foram comparadas e avaliadas através de uma Matriz de Decisão (apêndice J). Por fim, a solução escolhida foi a de número 5.

3.4 Justificativas das Notas das Soluções

Cada solução recebeu uma nota de 1 a 9 da seguinte forma: de acordo com as características desejadas em cada parâmetro, cada solução foi avaliada quanto ao seu possível desempenho; as soluções foram, então, ordenadas de acordo com esses desempenhos, de modo que soluções com mesmo desempenho em potencial receberam as mesmas notas, enquanto que as notas de soluções com desempenhos em potencial diferentes eram proporcionais a essa diferença.

Por exemplo, se fosse obtido $3 < 5 < 1 = 2 < 8$ (os números indicam as soluções). Isso significaria que a solução 3 tem um desempenho pior do que 5, que por sua vez é pior que 1 (que é igual a 2); e, por fim, 1 e 2 seriam piores do que 8.

Nessas condições a solução 8 receberia nota 2,0 quando comparada com as soluções 1 ou 2; receberia nota 3,0 se comparada a solução 5 e nota 4,0 se comparada a solução 3. A solução 1 receberia nota 1,0 se comparada à solução 2, mas teria um 0,5 se comparada à solução 8 e um 3,0 se comparada a solução 3.

Em suma, o procedimento de avaliação para cada parâmetro foi: analisar as soluções, escrevê-las em ordem crescente de desempenho naquele parâmetro e atribuir as notas de acordo com a regra anterior.

A 3.2.1 Velocidade

As notas de velocidade se basearam na possibilidade de travamento do dispositivo ao transportar os objetos coletados. Assim, soluções que empurravam os objetos obtiveram menores notas, enquanto que as soluções que carregavam efetivamente os objetos foram aquelas com maiores notas. Como as soluções 2 e 4 apenas carregavam os objetos, foram as que receberam maiores notas; em um nível abaixo ficaram as soluções 1 e 9 que mesclavam os dois tipos de transporte; mais distante e, assim, com menores notas, ficaram as soluções 3, 5, 6, 7 e 8, cujo mecanismo principal de transporte se baseava apenas em empurrar.

B 3.2.2 Peso

Como as soluções utilizaram, aproximadamente, os mesmos materiais, as notas de peso foram dadas principalmente de acordo com o modo com que se efetua o transporte (as que carregam os objetos tem o peso aumentado), mas, eventualmente, também foram considerados os números de componentes de cada solução (motores, rodas, servo-motores, etc). Por isso, as soluções 1, 2, 4, 6 e 9 receberam a pior nota: 2 e 4 porque carregam os objetos, 1 por ser muito parecido com o 2 e apresentar maior número de componentes, 6 e 9 pela elevada quantidade de componentes; em seguida, com nota 2,0 se comparadas às 5 anteriores, ficaram as soluções 3 e 7 que possuíam menos componentes; em segundo ficou a solução 8 e em primeiro a solução 5, devido a necessidade cada vez menor de materiais e componentes, bem como por só empurrarem os objetos.

C 3.2.3 Estabilidade

Basicamente, foi considerada a distribuição de massa (quanto maior a possibilidade de distribuí-la, maior a nota), bem como número de eixos (quanto mais, melhor) e a existência de movimentos de peças que poderiam causar o tombamento da solução. Em quarto lugar ficou a solução 5, uma vez que só possuía um eixo e a massa teria que ficar concentrada na traseira; em terceiro a solução 4, por ser uma solução com dimensão de altura elevada, o que facilitaria o tombamento; empatadas em segundo ficaram as soluções 3 e 8, que apesar de serem baixas e com mais de um eixo, possuíam concentração de massa na traseira; em primeiro ficaram as soluções 1, 2, 6, 7 e 9 que tinham mais de um eixo, eram baixas e com a massa distribuída de maneira relativamente uniforme.

D 3.2.4 Capacidade de carga

As soluções com as maiores notas foram aquelas que combinavam a possibilidade de armazenar de alguma forma os objetos, com a de empurrá-los. Em seguida, vieram soluções que apenas carregavam os objetos e, com as menores notas, aquelas que apenas os empurravam. Assim, em sexto lugar ficaram empatadas as soluções 6, 7 e 8 que apenas empurravam os objetos; as soluções 4 e 9 ficaram em quinto porque possuíam um complexo sistema dedicado ao transporte (e que poderia falhar); em quarto a solução 2, porque possuía um sistema de carga e poderia empurrar, mas sem proteção alguma; em terceiro a solução 3 pois possuía apenas um compartimento interno para empurrar os objetos em segurança; em segundo ficou a solução 1, pois consertava a falta de proteção da solução 2 e mantinha suas vantagens; em primeiro ficou a solução 5, uma vez que possuía um espaço interno para empurrar os objetos em segurança e um espaço externo com proteções laterais para minimizar as perdas (de modo a possuir o maior volume efetivo de capacidade de carga).

E 3.2.5 Velocidade de descarga

As soluções que apenas empurram objetos receberam as maiores notas, enquanto que aquelas que deveriam utilizar algum mecanismo para descarregá-los receberam notas menores. Em quinto ficaram as soluções 9 e 4 que deveriam acionar um complexo mecanismo na descarga; em quarto ficou a solução 1, pois o mecanismo de descarga era menos complexo; em terceiro ficaram as soluções 3 e 5, pois exigiriam uma sequência de manobras para liberar os objetos de seu espaço interno; em segundo ficou a solução 1, já que necessitaria apenas de uma operação simples para reverter o sentido de rotação do motor; em primeiro ficaram as soluções 6, 7 e 8 que apenas deveriam empurrar os objetos nos cestos.

F 3.2.6 Segurança de descarga

Ao contrário da velocidade, soluções com mecanismos dedicados à descarga obtiveram maiores notas, quando comparadas as soluções que apenas empurram os objetos no cesto. Dessa vez, 6, 7 e 8 ficaram em quarto lugar; 3 e 5 ficaram em terceiro devido ao compartimento de contenção; em segundo, as soluções 4 e 9 porque possuíam mecanismos de transporte mais complexos do que as soluções 1 e 2, que, portanto, ficaram em primeiro lugar.

G 3.2.7 Facilidade de manobrar

A nota foi dada baseada na possibilidade de compactação da solução (quanto mais redutível seu tamanho, maior a nota) e em sua forma geométrica. Em quarto lugar ficou a solução 8, devido ao seu formato em Y que seria propício ao travamento em todas as orientações; em terceiro, a solução 9 devido às suas dimensões elevadas; em segundo ficaram empatadas as soluções 1, 3, 4, 6 e 7 devido ao formato semelhante, mas que não poderia sofrer muitas alterações; empatados em primeiro ficaram as soluções 2 e 5, que apresentavam formas mais compactadas e arredondadas, com menor possibilidade de travamento.

H 3.2.8 Facilidade de manutenção

As soluções que mais pontuaram foram aquelas com o menor número de mecanismos extras (além do que realiza a movimentação) ou, no caso de mesma quantia de mecanismos extras, aquelas com menor complexidade e refinamento das peças desses mecanismos. Em quinto lugar ficaram as soluções 4 e 9, devido aos seus mecanismos complexos; em quarto vieram 1 e 2 que possuíam mecanismos, mas relativamente mais simples; em terceiro ficaram as soluções 3 e 5 que necessitavam dos mecanismos mais simples e que eram facilmente reparáveis em caso de falha; em segundo, ficou a solução 7, porque a única grande fonte de falha seria seu rolo; em primeiro ficaram empatados 6 e 8 porque não possuíam mecanismo algum e, portanto, os reparos seriam imediatos.

I 3.2.9 Facilidade de montagem

Esse parâmetro foi avaliado pela forma com que a solução funcionava de maneira global, isto é, aquelas com menor número de mecanismos e as com ideias mais simples receberam notas maiores. De forma análoga ao critério anterior: 4 e 9 ficaram em quarto; 1 e 2 ficaram em terceiro; 3, 5 e 7 ficaram em segundo (dessa vez o 7 ficou empatado pois a complexidade de montar o rolo é a mesma dos mecanismos com servo das outras duas soluções); e em primeiro as soluções 6 e 8 que não possuíam qualquer mecanismo.

J 3.2.10 Facilidade de fabricação

Também foram analisadas questões de complexidade, mas em uma escala das peças que compõe cada mecanismo. Assim, soluções cujas peças exigissem menos processos, ou processos mais simples, receberam notas mais elevadas. Mais uma vez: 9 e 4 ficaram em quinto, por apresentarem alta complexidade nas peças; 8 ficou em quarto, devido à dificuldade de se fabricar a parte mecânica que transporta os objetos, assim como era o problema da solução 6, mas que, por ser um pouco menos complexo, ficou em terceiro; em segundo ficaram as soluções 1 e 2 e em primeiro as soluções 3, 5 e 7, pois possuíam apenas peças com formas geométricas básicas em todos os seus mecanismos.

3.5 Formulações

Abaixo são indicadas as formulações matemáticas mais relevantes que consideradas para um adequado reconhecimento e planejamento da máquina (o desenvolvimento dessas formulações é apresentado mais adiante):

- Torque de cada motor DC: valor significativo para que possa ser realizada a avaliação entre

peso e desempenho. Além disso, como a máquina funcionará basicamente a baixas rotações é importante que o motor possua seus picos de torque nesses valores para que o dispositivo se movimente com eficiência, mesmo carregado;

- Torque do servomotor: quantificação essencial para a capacidade de carga que se espera ser possível para o trajeto até cada um dos cestos, assim sendo, como o servo estará condicionado ao mecanismo de carregamento da máquina, que varia conforme as soluções, inclusive a quantidade de servos, quanto maior for seu torque, maior a quantidade de carga que a máquina poderá transportar.

4 Resultados

4.1 Descrição geral e evolução do dispositivo

A ideia inicial da solução, como é possível observar do desenho (apêndice E), era a construção de uma máquina que seria capaz de armazenar objetos em um compartimento interno, além de empurrá-los. Seriam utilizados dois servomotores para o acionamento de uma “pá” que permitiria a coleta de objetos no compartimento. O veículo se movimentaria apenas com duas rodas, de forma que a parte frontal ficasse apoiada nos dois anteparos laterais. A parte autônoma seria feita com a utilização de três sensores ópticos de reflexão (usados normalmente em seguidores de linha), dos quais dois deles seriam usados nas rodas e serviriam como um tipo de encoder óptico (o funcionamento dessa parte do dispositivo será mais bem explicitado adiante) e um deles serviria para leitura de linha na superfície sobre a qual o veículo se movimentaria.

Após diversas avaliações, foram realizadas algumas modificações nessa ideia inicial do projeto que representaram melhorias consideráveis no planejamento do dispositivo, as principais são apresentadas a seguir: primeiramente verificou-se que a utilização de apenas um servomotor seria suficiente para atender às necessidades do projeto; foram adicionadas duas rodas frontais, que evitariam possíveis problemas de travamento da máquina durante a locomoção; a disposição dos componentes foi ajustada de forma a distribuir de maneira mais uniforme a massa no veículo; os motores passariam a serem fixados na parte inferior do chassi e foram criadas peças ou encaixes para fixação de todos os componentes da máquina, de forma que o chassi foi redesenhado para comportar os furos e encaixes. Os apêndices K (perspectiva isométrica do dispositivo) e L (desenho de conjunto) mostram com maiores detalhes as alterações mencionadas e permitem observar com detalhes o resultado obtido na fase de planejamento.

4.2 Equacionamento

De acordo com as formulações especificadas anteriormente, foi realizado o cálculo do torque necessário para o motor e o servomotor, e com base nisso esses itens foram adquiridos.

A 4.2.1 Torque do motor

Utilizou-se o esquema a seguir para realização do cálculo do torque necessário para o motor de corrente contínua. Nesse caso, levou-se em consideração a situação em que o veículo se encontra em uma rampa, pois é nessa situação em que se é necessário maior torque.

Estimando o peso máximo permitido a massa M do dispositivo, será em torno de 10kg. Considere α como sendo o ângulo formado entre a horizontal e a rampa da pista de competição que possui a maior elevação. Aplicando o Teorema do Momento Angular obtém-se a seguinte expressão para o torque T :

Erro! Não é possível criar objetos a partir de códigos de campo de edição.

(1)

A partir das informações sobre a pista, chega-se ao valor de α . Substituindo os demais valores conclui-se que cada motor deveria fornecer um torque de aproximadamente 4 kgf.

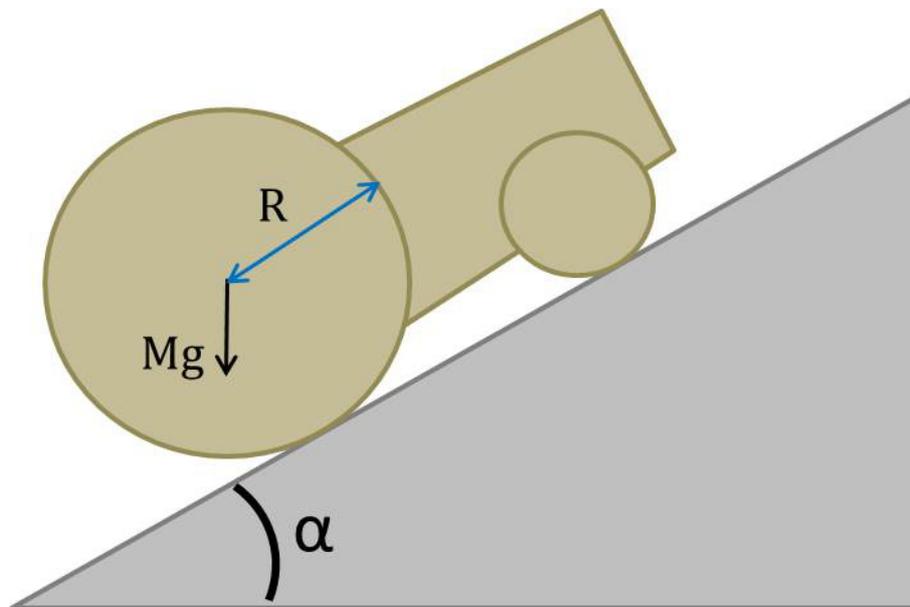


Figura 1 Esquema do dispositivo em uma rampa

Assim, aliando as informações obtidas devido ao equacionamento e a disponibilidade em mercado, utilizou-se um par de motores DC com as seguintes especificações:

Tabela I Especificações dos motores DC

Tensão Nominal	12V	Velocidade sem carga	84rpm
Corrente máx. Rendimento	0,30A	Torque máx. Rendimento	6kgf.cm
Potência	7W	Caixa de Redução	1:50

B 4.2.2 Torque do servomotor

Utilizou-se o esquema a seguir para realização do cálculo do torque necessário para o servomotor.

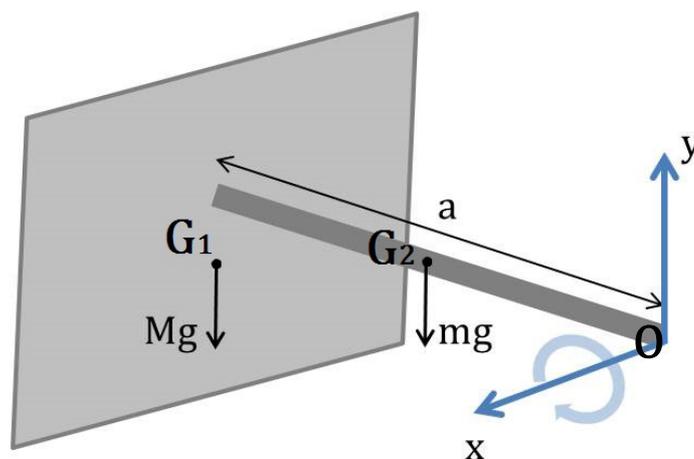


Figura 2 Esquema do mecanismo ligado ao servo

Seja M a massa da Pá e m a massa da chapa que liga a Pá ao servo (Braço). Considerando a como sendo o comprimento do Braço, o sistema de coordenadas $Oxyz$ solidário ao sistema “braço-

pá” e sabendo que o servo está vinculado ao mecanismo no ponto O, aplicou-se o Teorema do Momento Angular em relação ao polo O, desprezando o atrito no vínculo, o que resulta na seguinte expressão para o torque T :

Erro! Não é possível criar objetos a partir de códigos de campo de edição.

(2)

É preciso ressaltar que essa expressão foi obtida considerando a aceleração angular desprezível. Estimando as massas a partir das densidades dos materiais e considerando $g = 10 \text{ m/s}^2$ obtém-se aproximadamente $T = 0,23 \text{ N}\cdot\text{m}$. Assim, atendendo aos requisitos calculados e dada a facilidade de obtenção no mercado, optou-se por utilizar o servo MG995 [2].

4.3 Fabricação

Em conformidade com um dos requisitos do projeto relacionado à facilidade de fabricação do dispositivo, não foram necessários processos de fabricações demasiadamente complexos, de maneira que foram utilizadas somente as ferramentas disponíveis nas oficinas do departamento, como torno, furadeira de bancada, guilhotina, serra de fita e lixadeira de cinta.

Os desenhos de fabricação encontram-se disponíveis nos apêndices M a Y, e foram executados com auxílio do *software* Inventor da Autodesk®.

4.4 Eletrônica

A parte eletrônica do dispositivo desempenhou um papel fundamental em seu funcionamento, afinal era responsável por permitir o funcionamento correto dos sensores (cujos dados eram utilizados para o funcionamento autônomo), motores de corrente contínua (que permitiam a movimentação do dispositivo) e do servomotor (responsável pela movimentação do mecanismo de armazenamento de objetos).

A descrição do circuito elétrico que foi implementado no dispositivo pode ser melhor compreendida através do diagrama de ligações esquemático, gerado pelo *software* Fritzing®, com algumas modificações:

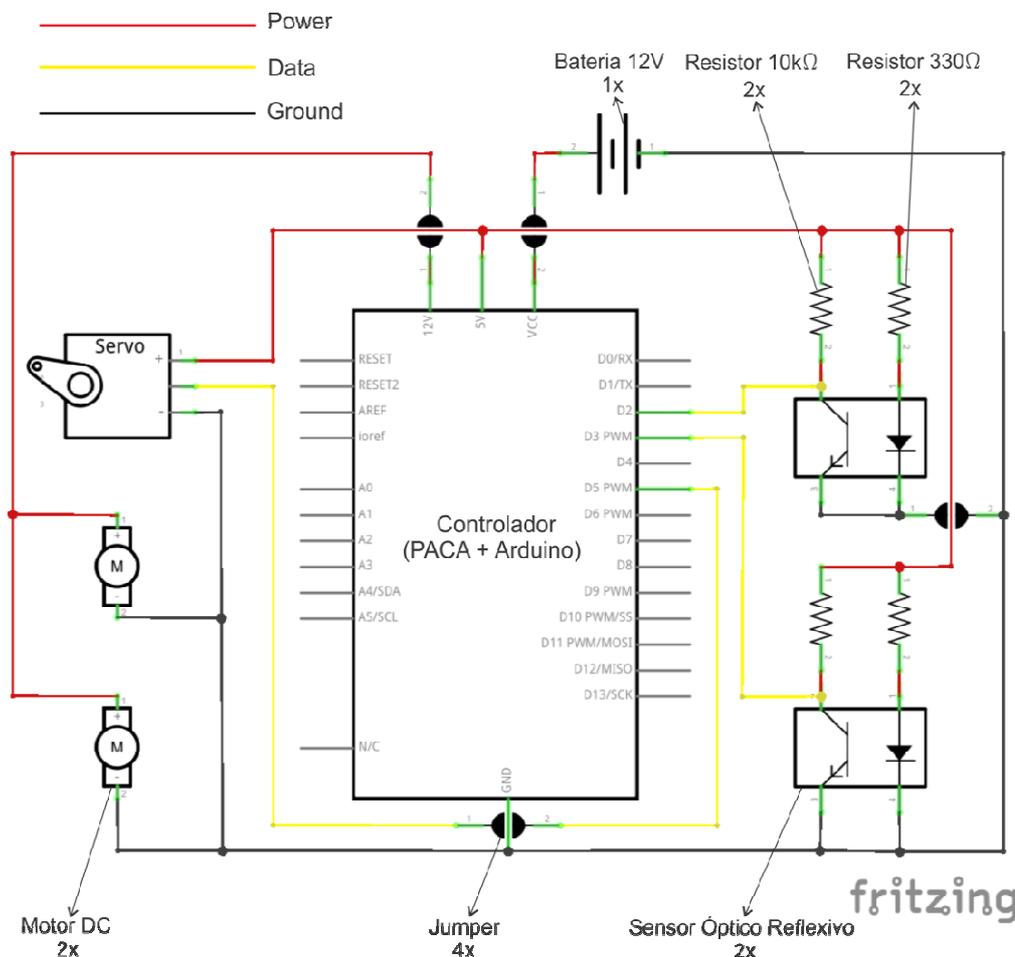


Figura 3 Circuito implementado no dispositivo

Basicamente, foram utilizados: uma bateria 12V, responsável por alimentar 2 motores de mesma tensão e a PACA (em conjunto com o microcontrolador Arduino); um servo de 5V, ligado de modo a receber corrente pelo retificador da PACA (e não diretamente do Arduino); e dois sensores ópticos reflexivos [3], montados em um circuito com resistores de 330 e 10.000 Ω .

Os motores de corrente contínua deveriam ser ligados corretamente nas saídas de 12V existentes na PACA e indicadas por *Motor Power*, de modo a garantir seu funcionamento adequado. De modo nativo, essas saídas da PACA estão ligadas aos pinos 9 e 10, para um motor, e 7 e 8 para o outro; sendo que os pinos 10 e 8 devem ser utilizados para o controle do sentido de rotação do motor (saída digital), enquanto que os pinos 9 e 7 controlam a velocidade propriamente dita. É importante notar que apenas o pino 9 possui PWM nativo no Arduino, de modo que é preciso utilizar algumas bibliotecas extras na programação (*SoftPWM*), para transformar o pino 7 em PWM também.

O servo, além de ser ligado, como já mencionado, na saída de 5V da própria PACA, também deveria ser ligado em uma saída digital (no caso, foi utilizado o pino número 5) para que as informações sobre a sua angulação pudessem ser transmitidas. Evidentemente, o servo também deveria ser aterrado.

Por último, foi montado o circuito dos sensores. Para que os sensores pudessem enviar os dados lidos, cada um deles precisou ser conectado em um pino digital (foram utilizados os pinos 2 e 3). Além disso, foi necessário, além de conectá-los em entradas de 5V e *ground*, montar um pequeno circuito com alguns resistores convenientes, de modo a atender as especificações de corrente e tensão desses sensores.

4.5 Controle

Para o controle autônomo do dispositivo durante o período de 60 segundos, foram utilizados sensores ópticos reflexivos que funcionavam, essencialmente, captando mudanças na luminosidade que recebiam, ou seja, detectavam se a uma pequena distância a sua frente havia algum objeto escuro ou claro. Eles foram utilizados de modo a implementar um sistema de odometria para o movimento autônomo: a parte interna das rodas do dispositivo foi pintada com faixas brancas e pretas alternadas, enquanto que os sensores foram posicionados apontando diretamente para essas faixas, de modo que liam as mudanças de cor, assim os sensores trabalhavam com dois *encoders* ópticos. Dessa forma, era possível saber o quanto a roda havia girado e, portanto, a distância percorrida pelo dispositivo.

4.6 Programação

O programa utilizado no controle do dispositivo, escrito diretamente no IDE do Arduino, segue a estruturação de qualquer programa desenvolvido para esse microcontrolador, contando com três blocos: declaração de variáveis e importação de bibliotecas; inicialização de algumas dessas variáveis (o *setup()*); e o código de controle propriamente dito (o *loop()*), que roda enquanto o Arduino UNO não é reiniciado. O código completo está disponível no anexo XXVI.

A 4.5.1 Importações e definições iniciais

Em primeiro lugar foram feitas importações e declarações globais para o funcionamento do código: importação da biblioteca para enviar e receber comandos via *bluetooth*, bem como de uma biblioteca que transforma pinos digitais em PWM e uma terceira para a utilização do servo. Também foram definidas variáveis booleanas para o controle do sentido do carrinho (“fwd”, “bwd” e “sentido”).

```
//Bibliotecas e definicoes gerais
#include <SoftwareSerial.h> //comunicacao bluetooth
#include <SoftPWM.h> //biblioteca que transforma um pino digital em PWM
#include <Servo.h> //biblioteca do servo
SoftwareSerial bluetooth(11, 12); //reserva os pinos 11 e 12 para o bluetooth
char comando; //char que vai armazenar os comandos recebidos
int sentido = 0; //variavel booleana para alterar o sentido dos motores
#define fwd 0 //definicoes de sentido para frente (fwd) e para tras (bwd)
#define bwd 1
```

Figura 4 Bibliotecas Importadas e Definições

Em seguida, foram declaradas variáveis relativas ao servo e aos dois motores de corrente contínua, sendo importante notar que para os dois motores foram utilizados pinos via *SoftPWM* (mesmo que o 9 possuísse PWM nativo) para que as diferenças de velocidade fossem minimizadas. Além disso, as designações “esquerda” e “direita” que acompanham cada motor foram feitas levando-se em conta a vista traseira do dispositivo.

```
//Variaveis do servo
Servo myservo;

//Variaveis do motor 1 (esquerda)
int velocidadePerc1 = 0; //velocidade em porcentagem
int sentido1 = 10; //controle do sentido

//Variaveis do motor 2 (direita)
int velocidadePerc2 = 0; //velocidade em porcentagem
int sentido2 = 8; //controle do sentido
```

Figura 5 Declarações das variáveis do servomotor e dos motores DC

Por último, foram feitas as declarações relativas aos sensores de reflexivos utilizados na parte autônoma:

```
//Variaveis do sensor
int sensorPin1 = 2; //le da roda da esquerda
int sensorPin2 = 3; //le da roda da direita
int estado; //guarda o estado do sensor
int aux; //variaveis auxiliares para a
int antes; //interpretacao dos dados dos sensores
int firsttime[] = {1, 1, 1, 1, 1, 1, 1}; //indica se e a 1 vez que leu do sensor
float deslocamento[] = {0, 0, 0, 0, 0, 0, 0}; //armazena o quanto andou em cada parte do trajeto
int flag[] = {0, 0, 0, 0, 0, 0, 0}; //vetor que indica em qual parte do trajeto o loop deve entrar
float D = (2*3.14159*40)/4; //incremento de distancia de cada transicao na roda
float P[] = {300, 3.14159*102, 700, 3.14159*102, 185, 3.14159*101, 840}; //distancias de cada parte
int pista = 1; //pista azul = 1; pista vermelha = 2
```

Figura 6 Declaração das demais variáveis

Assim, foram declarados: os pinos digitais em que estariam os sensores (2 e 3); variáveis para armazenar as leituras do sensor; bem como algumas variáveis e vetores auxiliares para o controle da parte autônoma (e cujas funções ficará mais clara quando for mostrado o código de controle).

B 4.5.2 Setup()

Esse bloco do código é dedicado às inicializações de algumas variáveis e definições de pinos como sendo de entrada ou saída de informações (*INPUT/OUTPUT*). Assim, estabeleceu-se a velocidade de comunicação por *bluetooth* e foram inicializados os pinos dos motores DC transformando-os em PWM digital, através da biblioteca *SoftPWM*. Além disso, os pinos de sentido foram definidos como sendo de saída de informação e os pinos dos sensores como de entrada. Por último ini-

cializou-se o servo e sua angulação inicial.

```
void setup() {
  //inicializacao da comunicacao bluetooth
  bluetooth.begin(9600);

  //Inicializacao do softPWM nos dois pinos em que os motores estao ligados
  SoftPWMBegin();
  SoftPWMSet(9,0);
  SoftPWMSet(7, 0);

  //Deve-se declarar os pinos de sentido como OUTPUT
  pinMode(sentido1, OUTPUT);
  pinMode(sentido2, OUTPUT);

  //Pinos dos sensores devem enviar sinais (INPUT)
  pinMode(sensorPin1, INPUT);
  pinMode(sensorPin2, INPUT);

  //Inicializacao do pino e angulo inicial do servo
  myservo.attach(5);
  myservo.write(90);
}
```

Figura 7 Inicialização das Variáveis

C 4.5.3 Loop()

O *loop* começa sempre checando se há alguma informação sendo passada via *bluetooth* e, se for o caso, armazena essa informação na variável “comando” que é do tipo “char”. Dependendo do conteúdo de “comando” uma ação seria realizada: controle do servo, controle dos motores DC ou inicialização da parte autônoma.

O controle do servo consistiu basicamente em alterar a angulação, decrementando 60 graus de sua posição original, ou retornando à angulação para a inicial (90 graus). As linhas com esses comandos foram:

```
//Servo
if (comando == 'k') {
  myservo.write(30);
}
if (comando == 'j') {
  myservo.write(90);
}
```

Figura 8 Comando do servomotor

A parte autônoma também possuía um comando dedicado à sua inicialização. Quando esse comando ocorria, o programa entrava em 7 trechos diferentes de código uma única vez (sendo essas entradas controladas pelo vetor *flag[]*). Cada um desses trechos correspondia à movimentação do dispositivo em 7 trechos específicos da pista, que foram escolhidos para que fossem percorridos durante o movimento autônomo. As distâncias que o dispositivo deveria percorrer em cada trecho foram armazenadas no vetor “P[]”, enquanto que o monitoramento da distância que já havia sido percorrida era feito através do incremento da quantia “D” no vetor “deslocamento[]”. Abaixo, trechos do código de controle nas duas primeiras partes do movimento autônomo:

```

while (flag[0] == 0) {

    if (firsttime[0] == 1) {
        estado = digitalRead(sensorPin2);
        if (estado == LOW) antes = 0;
        else antes = 1;
        firsttime[0] = 0;
    }

    velocidadePerc1 = 43;
    SoftPWMSetPercent(9, velocidadePerc1);
    velocidadePerc2 = 40;
    SoftPWMSetPercent(7, velocidadePerc2);

    estado = digitalRead(sensorPin2);
    if (estado == LOW) aux = 0;
    else if (estado == HIGH) aux = 1;

    if (aux != antes) {
        deslocamento[0] += D;
        antes = aux;
    }

    if (deslocamento[0] >= P[0]) {
        velocidadePerc1 = 0;
        SoftPWMSetPercent(9, velocidadePerc1);
        velocidadePerc2 = 0;
        SoftPWMSetPercent(7, velocidadePerc2);
        flag[0] = 1;
    }
}

while (flag[1] == 0) {

    if (pista == 1) {
        if (firsttime[1] == 1) {
            estado = digitalRead(sensorPin1);
            if (estado == LOW) antes = 0;
            else antes = 1;
            firsttime[1] = 0;
        }

        velocidadePerc1 = 40;
        SoftPWMSetPercent(9, velocidadePerc1);

        estado = digitalRead(sensorPin1);
        if (estado == LOW) aux = 0;
        else if (estado == HIGH) aux = 1;

        if (aux != antes) {
            deslocamento[1] += D;
            antes = aux;
        }

        if (deslocamento[1] >= P[1]) {
            velocidadePerc1 = 0;
            SoftPWMSetPercent(9, velocidadePerc1);
            flag[1] = 1;
        }
    }
}

```

Figura 9 Código do Controle Autônoma

O vetor *firsttime[]* servia para checar se era a primeira vez que o dispositivo estava em determinado trecho da parte autônoma. Se fosse o caso, o sensor era lido uma vez e a cor da pintura da roda era determinada naquele instante inicial. Depois disso, iniciava-se o movimento dos motores (ou de um só, se fosse uma curva) e, a cada momento, lia-se o sensor.

Quando havia uma mudança no estado de leitura, isto é, quando havia uma transição na pintura das rodas havia um incremento na posição do vetor “deslocamento[]” correspondente ao trecho percorrido. Por fim, quando a distância percorrida atingia o valor desejado, os motores eram parados e o vetor *flag[]* modificado para indicar que o dispositivo havia passado para o próximo trecho autônomo.

Além disso, é importante destacar que em trechos em que eram realizadas curvas era preciso acrescentar uma checagem de qual pista o dispositivo estava (que deveria ser especificada antes de rodar o programa) para decidir para qual lado virar e, conseqüentemente, de qual sensor tomar os dados da medição de distância percorrida.

Por fim, a última parte desse bloco está relacionada ao controle direto dos motores de corrente contínua. Seu controle consistia em basicamente responder aos comandos lidos pelo *bluetooth* incrementando ou decrementando gradativamente o PWM dos dois motores (com 3% a mais em um dos motores, de modo a otimizar a linearidade do seu avanço) e, inclusive, invertendo o sentido do movimento sempre que, mantendo o comando passado via *bluetooth*, a velocidade chegava a zero.

```

// Trecho de código para andar em linha reta
if (comando == 'w' && sentido == fwd) {
  velocidadePerc1 = velocidadePerc1 + 15;
  velocidadePerc2 = velocidadePerc2 + 15;
}

if (comando == 's' && sentido == fwd) {
  velocidadePerc1 = velocidadePerc1 - 15;
  velocidadePerc2 = velocidadePerc2 - 15;
}

if (comando == 'w' && sentido == bwd) {
  velocidadePerc1 = velocidadePerc1 - 15;
  velocidadePerc2 = velocidadePerc2 - 15;
}

if (comando == 's' && sentido == bwd){
  velocidadePerc1 = velocidadePerc1 + 15;
  velocidadePerc2 = velocidadePerc2 + 15;
}

if (velocidadePerc1<=0 && sentido == fwd) {
  velocidadePerc1 = 0;
  velocidadePerc2 = 0;
  sentido = bwd;
}

if (velocidadePerc1<=0 && sentido == bwd) {
  velocidadePerc1 = 0;
  velocidadePerc2 = 0;
  sentido = fwd;
}

// Trecho do código do controle de curvas
if (comando == 'd') {
  velocidadePerc2 = 0;
  SoftPWMSetPercent(7, velocidadePerc2);
  velocidadePerc1 = 60;
  SoftPWMSetPercent(9, velocidadePerc1);
  delay(500);
  velocidadePerc1 = 0;
  SoftPWMSetPercent(9, velocidadePerc1);
}

if (comando == 'a') {
  velocidadePerc1 = 0;
  SoftPWMSetPercent(9, velocidadePerc1);
  velocidadePerc2 = 60;
  SoftPWMSetPercent(7, velocidadePerc2);
  delay(500);
  velocidadePerc2 = 0;
  SoftPWMSetPercent(7, velocidadePerc2);
}

//Troca o sentido
if (sentido == bwd) {
  digitalWrite(10, HIGH);
  digitalWrite(8, HIGH);
}

if (sentido == fwd) {
  digitalWrite(10, LOW);
  digitalWrite(8, LOW);
}

//Manda as novas velocidades para o Arduino
SoftPWMSetPercent(9, velocidadePerc1+3);
SoftPWMSetPercent(7, velocidadePerc2);

```

Figura 10 Código do Controle Remoto

5 Discussão

Tomando como base os requisitos propostos, o desempenho apresentado pelo dispositivo foi bastante coerente com a grande maioria dos parâmetros, especialmente quanto à facilidade de fabricação e montagem, a máquina foi composta de peça simples de serem usinadas, e podendo ser unidas na sua quase totalidade por parafusos e porcas; por ser facilmente montável e desmontável, a manutenção de eventuais problemas também se mostrou simples.

Em outro fator relevante, que diz respeito à carga, o dispositivo apresentava uma capacidade de carregar simultaneamente um grande quantidade de objetos, sejam eles armazenados em um compartimento interno, sejam apenas empurrados até o destino desejado, contudo, uma maior capacidade de carga implicava em maiores dimensões para a máquina (quanto maior, mas espaço para guardar itens), dessa forma, foi obtida uma altura elevada para o dispositivo, necessitando de uma boa distribuição de peso para que o mesmo não tombasse em rampas. Entretanto o eixo motor se encontrava na região anterior, e uma distribuição de massa fazia com que o peso não se concentrasse sobre tal eixo, ocasionando um eventual deslizamento da roda, para tanto, o projeto foi modificado garantindo todo o peso sobre o eixo de tração, o que permitiu um movimento com menor deslizamento das rodas, porém impossibilitando a subida de rampas íngremes (rampa de entrada e saída do mar de isopor), devido ao tombamento.

Quanto à segurança na descarga, o projeto de proteções laterais no formato de “esquis” (para evitar atrito com o chão ou quinas) se mostrou uma boa opção para armazenamento, pois gerava uma área interna, juntamente com uma placa de alumínio acionada por um servo, onde se podia guardar objetos, especialmente com formato oval ou redondo, e uma área, guiada pela extremidade dos esquis, que podia empurrar com segurança os itens, especialmente perfis de formato reto (cubos).

No requisito velocidade, o dispositivo se mostrou coerente com os parâmetros de projeto, entretanto, foi-se observado que comparativamente com as máquinas de outros grupos tal quesito apresentava uma defasagem, devido à opção por um diâmetro de roda que não fosse muito grande, gerando perdas em velocidade, mas ganhos em estabilidade (necessário devido à altura elevada do

dispositivo).

Outros pontos genéricos observados que merecem destaque são o formato do dispositivo, com cantos arredondados para evitar quinas em curvas, o que se mostrou eficiente, contudo foi prejudicado por parafusos que se sobressaiam ao dispositivo e ficavam presos nas quinas. Além disso, a fixação dos motores apresentava graus de liberdade desnecessários, e no caso gerando movimentos não desejados (possibilidade de uma translação e rotação em torno do próprio eixo).

Além disso, o comportamento autônomo do dispositivo se mostrou eficiente, a utilização da odometria por meio de sensores reflexivos atuando como *encoders* forneceu resultados robustos, apesar de estarem intimamente condicionados a uma elevada precisão no alinhamento da máquina no ponto inicial e aos deslizamentos do motor em relação ao seu mancal e das rodas em relação à pista.

6 Conclusão

De uma forma geral a parte mecânica do dispositivo apresentou um desempenho adequado a proposta do projeto, salvo a questão da fixação dos motores e a questão dos parafusos sobressalentes que reduziam a manobrabilidade e o tombamento em rampas íngremes. Já no que diz respeito à parte elétrica, a utilização de dois sensores ópticos reflexivos juntamente com a pintura das rodas em bandas pretas e brancas, para funcionarem com *encoders* ópticos, foi uma alternativa interessante para a parte autônoma do percurso e que apresentava resposta bastante favorável.

E assim, pode-se salientar que o projeto como um todo apresentou um resultado bastante positivo e conforme a metodologia aplicada.

AGRADECIMENTOS

Agradecemos a todos os professores da disciplina que se mostraram disponíveis mesmo em horários fora da aula, também aos técnicos que foram muito prestativos com a ajuda na fabricação e com sugestões para o projeto. Em especial, gostaríamos de agradecer ao Prof. Dr. Rafael Traldi Moura, ao Prof. Dr. Thiago Castro Martins, e ao graduando Victor Pacheco Bartholomeu, do 5º ano da Engenharia Mecatrônica, pela contribuição essencial para o bom funcionamento do dispositivo.

REFERÊNCIAS

- [1] Pahl, Gerhard; Beitz, Wolfgang; Feldhusen, Jörg; Grote, Karl-Heinrich. Engineering Design: A Systematic Approach. 3ª edição. Springer, 2007.
- [2] Datasheet Tower Pro MG955 Metal Gear Servo. Disponível em: <http://www.electronicoscaldas.com/datasheet/MG995_Tower-Pro.pdf> Acesso em: 12 de março de 2016.
- [3] Datasheet Vishay Reflective Optical Sensor with Transistor Output (TCRT5000). Disponível em: <<http://www.vishay.com/docs/83760/tcrt5000.pdf>> Acesso em: 12 de março de 2016.

Project Memorial of a Mechanical Device Self Controlled by Optical Encoders and Remotely Controlled via Bluetooth.

Abstract – On this paper are described the stages of development of a device proposed as project for the discipline Introduction to Mechanical Systems Design (PMR3202) of the Polytechnic School of the University of São Paulo (EPUSP). The device should collect the largest number of objects as possible in a defined region and transport them to a specified location during a certain period. Part of the circuit should be performed autonomously. The machine was designed in order to fulfill a number of requirements, identified from the project's constraints. For its implementation throughout the project, several measures were taken: evaluation of solutions formulated using a Decision Matrix, design of mechanical components through formulations, design of the system for machine's autonomous performance, microcontroller programming for control of sensors and actuators, manufacturing of parts and complete assembly. From the results, it was concluded that the device answered acceptably the requirements, and also identify aspects that could be improved to enhance the device's performance.

Keywords – Project Memorial, Design Methodology, Arduino Uno, Optical Encoder, Bluetooth

Autores

Caio Garcia Cancian, estudante de graduação do curso de Engenharia Mecatrônica da Escola Politécnica da Universidade de São Paulo. Bolsista do Programa de Educação Tutorial, PET Automação e Sistemas da Escola Politécnica. Colaborador no Laboratório de Sistemas de Automação (LSA) do Departamento de Mecatrônica e Sistemas Mecânicos (PMR) da Escola Politécnica.

Gustavo Alencar Bisinotto, estudante de graduação do curso de Engenharia Mecatrônica da Escola Politécnica da Universidade de São Paulo. Bolsista do Programa de Educação Tutorial, PET Automação e Sistemas da Escola Politécnica. Colaborador no Laboratório de Sistemas de Automação (LSA) do Departamento de Mecatrônica e Sistemas Mecânicos (PMR) da Escola Politécnica.

Luis Felipe Gomes de Oliveira, estudante de graduação do curso de Engenharia Mecatrônica da Escola Politécnica da Universidade de São Paulo. Bolsista do Programa de Educação Tutorial, PET Automação e Sistemas da Escola Politécnica. Colaborador no Laboratório de Sistemas de Automação (LSA) do Departamento de Mecatrônica e Sistemas Mecânicos (PMR) da Escola Politécnica.