

*Edilson Hiroshi Tamai*

KARINE MORIYA

LUIZ FELIPE ALMEIDA SOUZA

*8,5 bits e mais*  
*KAM*

***SISTEMA DE CONTROLE DE UM MECANISMO DE  
POSICIONAMENTO NO PLANO HORIZONTAL***

*Projeto de Formatura  
apresentado à Escola Politécnica  
da Universidade de São Paulo –  
Departamento de Engenharia  
Mecânica – Hab. em Automação e  
Sistemas.*

*Orientador:  
Prof. Dr. Edilson Hiroshi Tamai*

*São Paulo  
1998*

<b>1. INTRODUÇÃO .....</b>	<b>4</b>
1.1 OBJETIVOS.....	4
1.2 MOTIVAÇÃO - APLICAÇÃO .....	5
1.3 LAYOUT DO SISTEMA .....	6
1.4 INTEGRAÇÃO DOS COMPONENTES .....	7
<b>2. METODOLOGIA.....</b>	<b>8</b>
<b>3. PROJETO DO MECANISMO.....</b>	<b>10</b>
3.1 SOLUÇÃO GEOMÉTRICA DO MECANISMO .....	12
3.1.1 Determinação do ângulo $\theta_2$ .....	13
3.1.2 Determinação do ângulo $\beta_2$ .....	15
<b>4. COMPONENTES DO SISTEMA .....</b>	<b>17</b>
4.1 PROTÓTIPO .....	17
4.2 MOTORES ELÉTRICOS.....	19
4.3 SENSORES DE POSIÇÃO .....	22
4.4 PLACA CONVERSORA A/D E D/A .....	25
4.5 CIRCUITO DE POTÊNCIA.....	27
<b>5. INTEGRAÇÃO FÍSICA .....</b>	<b>29</b>
<b>6. MODELAGEM MATEMÁTICA .....</b>	<b>30</b>
<b>7. TEORIA DE CONTROLE .....</b>	<b>35</b>
7.1 SISTEMAS DE SEGUNDA ORDEM.....	35
7.2 CONTROLADORES.....	37
7.3 CONTROLADOR DIGITAL .....	39
7.3.1 Método de Transformação Bilinear.....	40
7.3.2 Método do Casamento de Pólos e Zeros.....	41
7.4 EQUAÇÕES DE DIFERENÇAS.....	41

<b>8. SIMULAÇÃO NUMÉRICA.....</b>	<b>43</b>
8.1. RESPOSTA EM MALHA ABERTA .....	43
8.2 RESPOSTA EM MALHA FECHADA .....	44
<b>9. SOFTWARE DE CONTROLE.....</b>	<b>47</b>
9.1 PROGRAMA PRINCIPAL .....	47
9.2 PROCEDIMENTO QUE CALCULA OS DESLOCAMENTOS ANGULARES DAS HASTES .....	48
9.2.1. Diagrama do cálculo do ângulo $\delta$ .....	49
9.2.2. Diagrama do cálculo da posição final da articulação 2 ( $A_2$ ) .....	49
9.2.3. Diagrama do cálculo do ângulo $\gamma$ .....	50
9.3. PROCEDIMENTO QUE MOVE A HASTE 2 PARA A SUA POSIÇÃO INICIAL.....	51
9.4 PROCEDIMENTO QUE MOVE A HASTE 1 PARA A POSIÇÃO FINAL DESEJADA .....	51
9.5 PROCEDIMENTO QUE MOVE A HASTE 2 PARA A POSIÇÃO FINAL DESEJADA .....	53
<b>10. ANÁLISE DOS RESULTADOS .....</b>	<b>54</b>
<b>11. CONCLUSÕES.....</b>	<b>58</b>
<b>12. BIBLIOGRAFIA .....</b>	<b>60</b>
<b>13. ANEXOS .....</b>	<b>61</b>
13.1 LISTAGEM DO PROGRAMA PICO .....	61
13.2 DESENHOS DE FABRICAÇÃO .....	74

## **1. Introdução**

### **1.1 Objetivos**

O projeto "Sistema de Controle de um Mecanismo de Posicionamento no Plano Horizontal" tem como objetivo projetar e implementar um protótipo de um mecanismo composto por duas hastes articuladas, que realizam um movimento de rotação no plano horizontal.

O movimento deste braço mecânico, com dois graus de liberdade, é executado por dois motores elétricos de corrente contínua. Cada um deles é responsável pela rotação de cada uma das hastes.

Para o acionamento dos motores é necessário o projeto de um sistema de potência e de controle, de modo que os motores operem adequadamente.

O controle deste sistema é realizado com o auxílio de um microcomputador, que receberá a informação do posicionamento do mecanismo, dada por um sensor de posição, comparando-a com um valor de referência.

Na primeira fase do projeto, que ocorreu no primeiro semestre de 1998, o objetivo principal foi o projeto do mecanismo e de seus componentes, a modelagem matemática de motores elétricos de corrente contínua, a seleção dos componentes a serem utilizados na fase de implementação e o desenvolvimento de um algoritmo que calcula o ângulo de rotação de cada haste, necessário para atingir a posição final desejada.

Para a segunda fase do projeto, que ocorreu no segundo semestre de 1998, os objetivos foram: a construção do protótipo, o projeto e a implementação do sistema de potência, a simulação computacional do modelo e o desenvolvimento e implementação do software de controle, para a integração dos componentes do sistema.

## **1.2 Motivação - Aplicação**

A principal motivação inicial para este Trabalho de Formatura foi o desenvolvimento de um projeto que envolvesse a implementação prática de um sistema de controle.

Tendo isto em vista, decidimos optar pelo controle de motores elétricos. O fato da existência prévia de dois motores elétricos de corrente contínua no departamento nos deu subsídios para chegarmos a esta decisão.

Uma boa aplicação para motores elétricos seria a atuação destes sobre um mecanismo, onde poderíamos aplicar os conhecimentos adquiridos durante o curso. Sem mencionar o fato de que o projeto está intimamente ligado à área de Engenharia Mecatrônica.

A primeira aplicação idealizada foi a de um mecanismo que passasse por posições fixas e pré-definidas, visando a automação industrial de uma produção em série, como por exemplo: mover uma peça de uma esteira para uma bancada, onde seria realizado um processo de furação e, em seguida, posicionada em uma outra esteira. Porém, esta aplicação foi considerada limitada e resolveu-se estendê-la para uma área de posições possíveis.

O mecanismo agora idealizado é responsável por posicionamentos no plano horizontal dentro desta área previamente definida. A escolha pelo movimento no plano horizontal e não vertical, deu-se, principalmente, pois o objetivo não é levantar peças de uma posição para outra, mas apenas posicioná-las sem a influência da força gravitacional. Com isto, podemos utilizar os motores como atuadores responsáveis por estes deslocamentos angulares.

Uma possível aplicação ainda poderia ser o posicionamento de peças, ferramentas e acessórios, por exemplo, em uma linha de produção, porém com a diferença de existir uma área limitada de operação do mecanismo.

### 1.3 Layout do Sistema

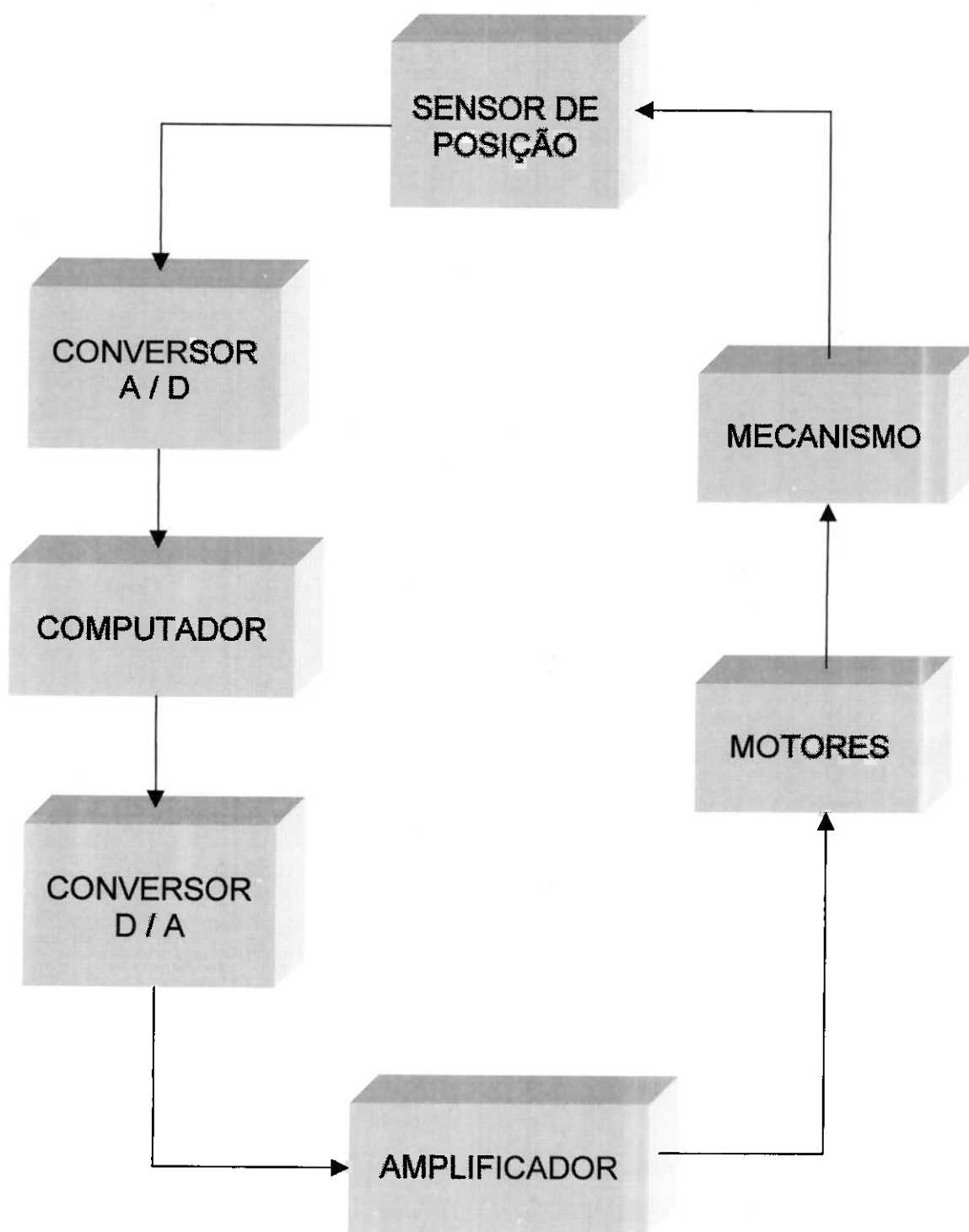


Figura 1: Layout do sistema

## **1.4 Integração dos Componentes**

- **MECANISMO**
  - ⇒ ENTRADA: Torque e rotação dos motores.
  - ⇒ SAÍDA: Posição angular.
  
- **SENSOR DE POSIÇÃO**
  - ⇒ ENTRADA: Posição angular do mecanismo.
  - ⇒ SAÍDA: Sinal de tensão (0V a 10V).
  
- **CONVERSOR A / D**
  - ⇒ ENTRADA: Sinal de tensão analógica.
  - ⇒ SAÍDA: Sinal de tensão digital para o computador.
  
- **COMPUTADOR**
  - ⇒ ENTRADA: Posição final desejada do mecanismo  
Posição atual do mecanismo
  - ⇒ SAÍDA: Sinal de tensão de controle digital ( $\pm 5V$ )
  
- **CONVERSOR D / A**
  - ⇒ ENTRADA: Sinal de tensão digital
  - ⇒ SAÍDA: Sinal de tensão analógica
  
- **AMPLIFICADOR**
  - ⇒ ENTRADA: Sinal de tensão do conversor
  - ⇒ SAÍDA: Sinal de corrente amplificado e tensão de controle
  
- **MOTORES**
  - ⇒ ENTRADA: Sinal amplificado
  - ⇒ SAÍDA: Torque e rotação

## **2. Metodologia**

Para a realização do projeto, fizemos uma breve descrição da metodologia utilizada.

- *Revisão Bibliográfica:* pesquisa e consulta a livros e/ou textos sobre motores elétricos de corrente contínua, robótica, controle de motores, projeto de mecanismos, catálogos de sensores, manual da placa conversora e eventuais temas que precisaram ser abordados no decorrer do projeto;
- *Modelagem Matemática:* realização do modelo matemático de motores de corrente contínua, baseado em informações coletadas da Revisão Bibliográfica, levando-se em consideração as características técnicas e construtivas dos motores utilizados;
- *Projeto do Mecanismo:* projeto geométrico do mecanismo que atende aos requisitos de posicionamento do projeto, baseado na teoria de díadas, bem como o dimensionamento, os desenhos de fabricação e seleção do material das peças componentes do mecanismo, tentando prever a sua montagem e desmontagem, as situações extremas de operação e a sua exequibilidade física;
- *Seleção dos Componentes:* seleção dos componentes necessários para a construção do hardware mecânico e elétrico e o funcionamento do protótipo como um todo, desde os motores até os sensores de posição;
- *Sistema de Controle:* projetar um controlador de modo que o movimento do mecanismo alcance o posicionamento final desejado, utilizando as técnicas de controle clássico baseado na família de controladores PID;
- *Simulação Numérica:* simulação, com o auxílio computacional (MATLAB), a resposta do sistema em malha aberta e malha fechada controlado,



prevendo o funcionamento do protótipo e realizando os ajustes necessários do controlador, de modo a garantir um bom funcionamento do sistema;

- *Construção do Protótipo*: fabricação e montagem das peças constituintes do protótipo, desde o mecanismo em si até a sua interface com o computador, atentando para os possíveis detalhes construtivos e operacionais não previstos anteriormente na fase de Projeto do Mecanismo;
- *Sistema de Potência*: implementação de um circuito de potência capaz de amplificar o sinal de corrente enviada pelo computador de modo a garantir o acionamento dos motores;
- *Software de Controle*: implementação digital do controle via software, com o desenvolvimento de algoritmos de controle e de interfaceamento entre o protótipo e a placa conversora, em uma linguagem de programação computacional (Pascal);
- *Integração*: realização da integração dos componentes do sistema, apresentado no layout, de modo a testar as entradas desejadas e verificar os resultados obtidos, solucionando os eventuais problemas.

### **3. Projeto do Mecanismo**

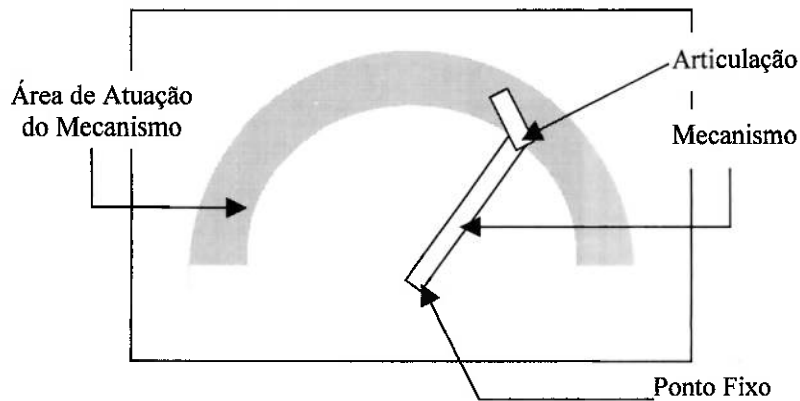
O mecanismo consiste basicamente de duas hastes, de movimentos angulares independentes, unidas por uma articulação.

Inicialmente, a idéia era posicionar o eixo de cada motor em uma extremidade da respectiva haste a ser movida, de modo que o movimento angular das hastes ocorresse com precisão. Porém, no caso da segunda haste, a inércia do motor ali instalado sobrecarregaria o outro motor, que teria que movimentá-lo também, e o peso do mesmo faria com que a primeira haste fletisse consideravelmente de modo a dificultar o movimento da mesma. Além disso, a sua fixação à haste seria um tanto quanto complicada. Tendo isto em vista, resolveu-se por optar em fixar os dois motores, com os eixos alinhados, e o uso de um sistema de transmissão de torque e rotação por correia e polias do segundo motor para a segunda haste [4]. Porém, esta solução aumenta um pouco a complexidade do projeto do mecanismo, pois temos que garantir o alinhamento dos dois eixos e pode diminuir a precisão do movimento, devido ao rendimento da transmissão utilizada.

Portanto, a montagem do mecanismo ficou, basicamente, da seguinte forma: uma das extremidades da primeira haste está articulada e acoplada a um motor. Este motor está fixo e permite uma rotação da haste de  $0^\circ$  a  $180^\circ$ . O outro motor, também fixo na mesma estrutura do primeiro motor, permite uma rotação de  $0^\circ$  a  $180^\circ$  da segunda haste, por meio da transmissão por correia. Assim, é possível definir uma área de atuação do mecanismo, limitada pelo movimento e comprimento das hastes.

O mecanismo parte de uma posição inicial conhecida e movimenta-se até uma posição final, dentro desta área, que deverá ser estabelecida pelo usuário através da interface com o computador.

A figura (2), a seguir, mostra um esboço do mecanismo, assim como o seu funcionamento:

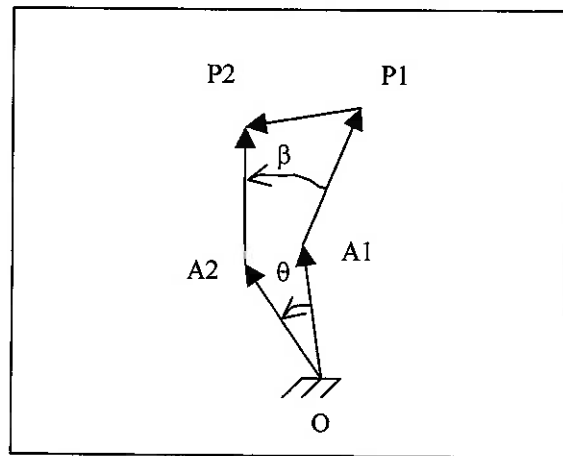


**Figura 2: Esboço do mecanismo**

Deve-se notar que a haste menor gira de  $0^\circ$  A  $180^\circ$  em relação à haste maior, porém existe ainda uma limitação deste movimento quando a haste maior encontra-se nas posições extremas ( $0^\circ$  e  $180^\circ$ ). Quando a haste maior se movimentar para o  $0^\circ$ , a haste menor girará entre  $90^\circ$  e  $180^\circ$  e quando ela se deslocar para  $180^\circ$ , a outra haste girará entre  $0^\circ$  e  $90^\circ$ . Esta limitação foi imposta para que o mecanismo não ultrapassasse a área de atuação definida.

O comprimento da haste maior será de 15 cm, enquanto que a haste menor será de 11 cm. Deste modo, o raio máximo de atuação da díada, (duas peças ligadas por uma articulação), será de 26 cm.

O esquema da figura (3) ilustra as posições inicial  $P_1$  e final  $P_2$  do mecanismo.



**Figura 3:** Representação esquemática do mecanismo

O primeiro motor (motor1) será responsável pela rotação  $\theta$  da haste maior (haste1) em relação a um referencial ligado à base, e o segundo motor (motor 2) pela rotação  $\beta$  da haste menor (haste2) em relação à haste maior. O desenvolvimento do cálculo destes ângulos, bem como o algoritmo que irá implementá-lo computacionalmente, serão mostrados no capítulo 7.

### **3.1 Solução Geométrica do Mecanismo**

Nesta etapa iremos determinar os ângulos que as hastes deverão se movimentar saindo de uma posição inicial a uma posição final. A haste 1 deverá movimentar-se segundo um ângulo  $\theta_2$  e a haste 2, segundo um ângulo  $\beta_2$ . A solução será implementada no software de controle e o algoritmo está apresentado do item 9.

### 3.1.1 Determinação do ângulo $\theta_2$

O ângulo  $\theta_2$  foi calculado a partir da seguinte análise geométrica:

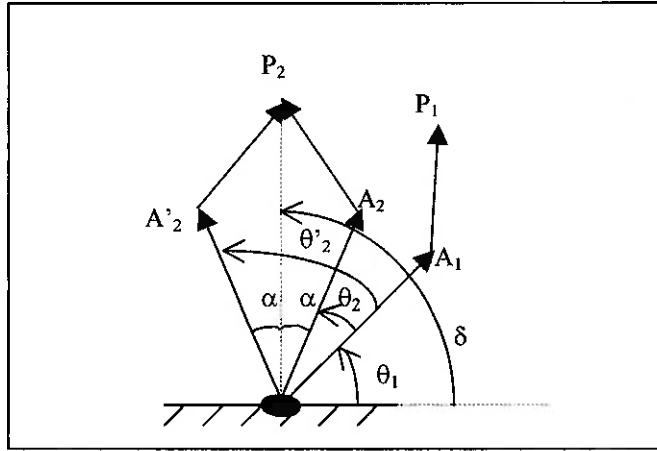


Figura 4: Representação esquemática do mecanismo

Sendo:

$$|z_1| = \text{comprimento da haste 1}$$

$$|z_2| = \text{comprimento da haste 2}$$

$$|z_3| = |(P2 - O)|$$

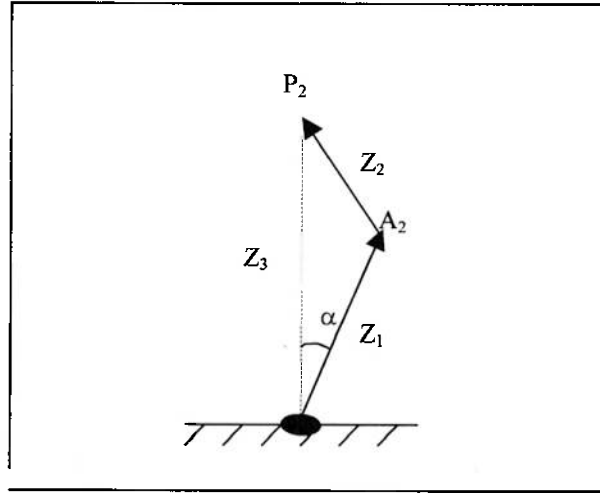
O ângulo  $\theta_2$  será obtido através das seguintes relações angulares:

$$\delta = \theta_1 + \theta_2 + \alpha \Rightarrow \theta_2 = \delta - \theta_1 - \alpha \quad (1)$$

$$\delta + \alpha = \theta_1 + \theta'_2 \Rightarrow \theta'_2 = \delta - \theta_1 + \alpha \quad (2)$$

Através das relações (1) e (2) obtém-se dois valores de ângulos possíveis:  $\theta_2$  e  $\theta'_2$ . Em seguida, optamos por escolher o menor ângulo entre os dois, o que resulta no menor e mais rápido caminho entre as posições inicial e final.

Para resolvermos (1) e (2), temos que calcular os ângulos  $\alpha$  e  $\delta$ , e utilizar o ângulo  $\theta_1$ , que é dado pela posição inicial.



**Figura 5:** Representação esquemática para obter o ângulo  $\alpha$

O ângulo  $\alpha$  é calculado a partir da lei dos cossenos:

$$z_2^2 = z_1^2 + z_3^2 - 2 \cdot z_1 \cdot z_3 \cdot \cos \alpha \quad \Leftrightarrow \quad \alpha = \arccos \left( \frac{z_1^2 + z_3^2 - z_2^2}{2 \cdot z_1 \cdot z_3} \right)$$

Como pode ser visto na figura 1, o ângulo  $\delta$  é a inclinação do vetor  $z_3$ :

$$\operatorname{tg} \delta = \left( \frac{P_{2y} - O_y}{P_{2x} - O_x} \right) \quad \Leftrightarrow \quad \delta = \operatorname{arctg} \left( \frac{P_{2y}}{P_{2x}} \right)$$

Antes de calcular  $\delta$ , deve-se verificar se  $P_{2x} = 0$ . Caso  $P_{2x}$  for igual a zero, assume-se que  $\delta$  vale  $90^\circ$ . Se  $P_{2x}$  for menor que zero, tem-se que  $\delta = \delta + 180^\circ$ . Isso porque, ao calcular a tangente de um número negativo, o compilador assume o ângulo negativo.

### 3.1.2 Determinação do ângulo $\beta_2$

Determinado o menor ângulo  $\theta_2$ , o ângulo  $\beta_2$  é obtido através da seguinte relação angular, como pode ser visto na figura abaixo:

$$\gamma = \beta_1 + \beta_2 \quad (3)$$

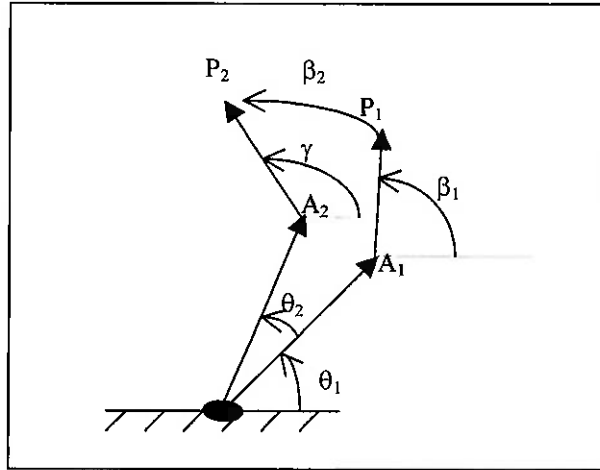


Figura 6: Representação esquemática da haste 2

Para obter o ângulo  $\gamma$ , é necessário conhecer a posição  $A_2 = (A_{2x} ; A_{2y})$ , que é determinada através do seguinte sistema:

$$\begin{aligned} A_{2x}^2 + A_{2y}^2 &= |(A_2 - O)|^2 \\ \operatorname{tg}(\theta_1 + \theta_2) &= \frac{A_{2y} - O_y}{A_{2x} - O_x} = \frac{A_{2y}}{A_{2x}} \quad (4) \end{aligned}$$

Se  $(\theta_1 + \theta_2)$  for igual a zero,  $A_{2x} = z_1$  (comprimento da haste 1) e  $A_{2y} = 0$ .

Se  $(\theta_1 + \theta_2)$  for igual a  $90^\circ$ ,  $A_{2x} = 0$  e  $A_{2y} = z_1$ .

O ângulo  $\gamma$  é dado pela inclinação da haste 2 na segunda posição (posição final):

$$\operatorname{tg} \gamma = \frac{P_{2y} - A_{2y}}{P_{2x} - A_{2x}} \quad \Leftrightarrow \quad \gamma = \operatorname{arctg} \left( \frac{P_{2y} - A_{2y}}{P_{2x} - A_{2x}} \right)$$

Antes do cálculo através da equação acima, deve-se verificar os seguintes casos:

- Caso 1: Se  $P_{2x} = A_{2x}$  então  $\gamma = 90^\circ$
- Caso 2: Se  $P_{2x} < A_{2x}$  então  $\gamma = \gamma + 180^\circ$

Através da relação (3), obtém-se o ângulo  $\beta_2$  através da equação:

$$\beta_2 = \gamma - \beta_1$$

Após calcular os ângulos de rotação  $\theta_2$  e  $\beta_2$ , o mecanismo executará o movimento para chegar à posição desejada. Feito isso, caso o usuário deseje uma nova posição, o programa executa todos os passos anteriores, assumindo que:

$$\theta_1 = \theta_1 + \theta_2$$

$$\beta_1 = \beta_1 + \beta_2$$



## **4. COMPONENTES DO SISTEMA**

### **4.1 Protótipo**

O protótipo foi projetado de modo a atender o funcionamento desejado do mecanismo proposto, levando em conta a exequibilidade física, os processos de usinagem e fabricação, e adotando soluções para os eventuais problemas de montagem.

As peças projetadas que compõem o protótipo são as seguintes:

- *Base para o motor 1:* Esta peça tem como função a fixação do motor inferior e da base de suporte do potenciômetro da haste 1. O material utilizado é uma chapa de alumínio de 1/8" de espessura.
- *Base para o motor 2:* Peça que tem a função de fixar o motor superior, que deverá ter o seu eixo alinhado com o eixo do motor inferior. O material utilizado é o mesmo da base para o motor1.
- *Haste 1:* O material utilizado nesta haste é uma barra de alumínio de 1/8" de espessura. Em uma das extremidades há um furo rosqueado para a sua fixação ao eixo do motor, na outra extremidade o furo não é rosqueado, de modo a permitir a rotação da haste 2 sem causar momento.
- *Haste 2:* O material desta haste é uma barra de alumínio de 13mm de espessura. A peça tem uma forma conveniente de forma que possa ser articulada com a haste 1.
- *Eixos:* Foram projetados dois eixos de aço 1020, um com a função de articular as duas hastes, fixar uma das polias e o potenciômetro da haste 2; e o outro para fixar a outra polia.
- *Base para o potenciômetro:* Foram projetadas duas destas peças com o intuito de fixar os potenciômetros a uma referência. Material: chapa de alumínio.

Os desenhos de fabricação de cada uma das peças e do conjunto encontram-se em anexo.

Para a fixação das peças foram utilizados parafusos, porcas, arruelas lisas e de pressão.

Algumas soluções foram adotadas de forma a viabilizar a idéia do mecanismo. As mais importantes são as seguintes:

- Para a transmissão do torque do motor 2 para movimentar a haste 2, com relação de transmissão 1:1, foi utilizado uma par de polias de alumínio, com diâmetro nominal de 24.5 cm e uma correia dentada padronizada para uma distância entre centros de 15 cm.
- Para a leitura da posição angular da haste 1, foi utilizado um par de engrenagens de mesmo diâmetro nominal 34.5 cm e mesmo número de dentes, para transmitir a rotação do eixo do motor 1 para o eixo do potenciômetro 1, com relação de transmissão 1:1. Esta solução foi adotada, pois os eixos dos dois motores precisariam estar os mais próximos possíveis, para diminuir o momento causado pela força de tração que surge quando o eixo do motor 2 gira. Portanto, não daria para fixar o potenciômetro na ponta do eixo do motor 1.
- Uma luva de naylor foi fabricada com a função de alinhar o eixo de rotação do potenciômetro 2 com o eixo que articula as duas hastes.

A fabricação e a montagem da todas as peças foi realizada no Laboratório de Máquinas do Departamento de Engenharia Mecânica.

## **4.2 Motores Elétricos**

Os motores elétricos utilizados neste projeto foram 2 Motorredutores de corrente contínua. Entretanto, deve-se ressaltar que a escolha destes motores ocorreu devido a sua pré-disponibilidade no Departamento de Engenharia Mecânica, ou seja, não partimos de uma análise sobre qual seria o tipo de motor mais indicado para este tipo de aplicação e sua posterior seleção.

Os tipos de motores mais adequados para atender ao requisito de rígido controle de posição seriam os motores de passo e servomotores.

Os motores de passo possuem a vantagem de oferecer uma interface digital que facilita o envio de sinais entre ele e o controlador. Quanto às suas características cinemáticas e dinâmicas, este tipo de motor possui uma faixa de velocidades baixa que é função do torque, que por sua vez também é baixo. No caso deste projeto, o motor não será submetido a uma carga alta, visto que eles terão que movimentar hastes de pequenas dimensões e pequeno peso (devido ao material ser de alumínio) e, no pior caso, teriam que mover a polia na transmissão do torque. Consequentemente não teríamos o problema de perda de passos. Os motores de passo, entretanto, possuem uma limitação de resolução, pois apresentam um número finito e não contínuo de posições possíveis, o que poderia diminuir a precisão da posição a ser atingida.

Os servomotores apresentam um ótimo controle de velocidade, torques relativamente elevados e alta eficiência, portanto são ideais para aplicações de controle. Este tipo de motor é amplamente utilizado em dispositivos controlados de pequenas e médias dimensões.

Os motores utilizados são motorredutores DC do tipo CHP da Bosch e possuem as seguintes características técnicas:

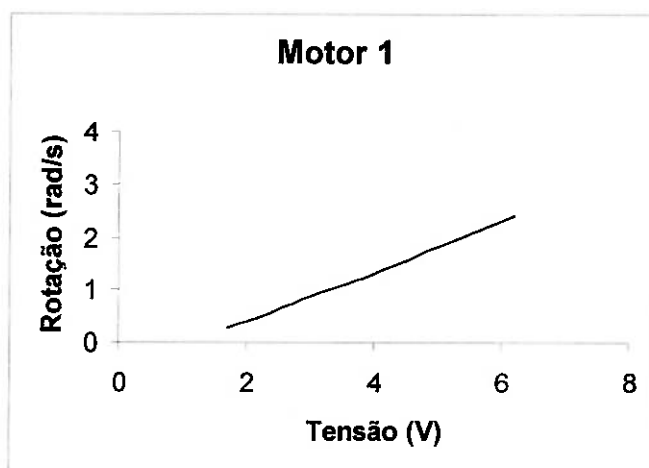
- Tensão: 12V ou 24 V;
- Corrente máxima: 15 A;

- Torque: 7 a 13 Nm;
- Sentido de rotação: horário e anti-horário;
- Rotação máxima: 50 rpm.

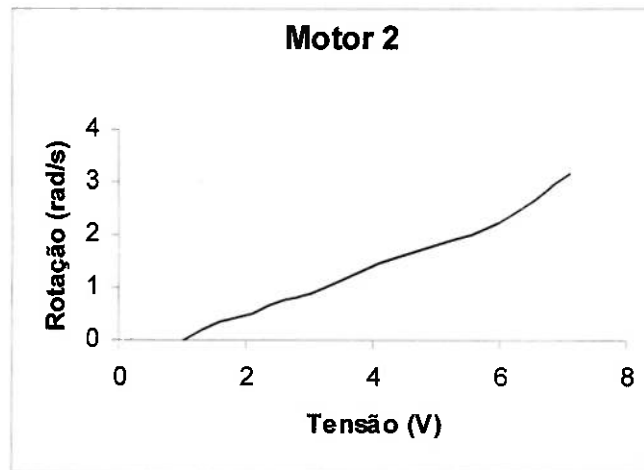
Em anexo, segue um "folder" deste motor.

Foram realizados experimentos com os motores para determinar uma curva que relaciona a tensão aplicada aos terminais dos mesmos e a velocidade de rotação do eixo. Estes experimentos levaram em consideração a aplicação de uma tensão constante ao motor por um certo intervalo de tempo, que garantisse o movimento em regime, para realizar a medição da rotação relativa à tensão aplicada. Em seguida, as medições foram repetidas para novos valores de tensão.

Estas curvas estão apresentadas nas figuras 7 e 8, sendo que a figura 7 mostra a curva do motor 1 e a figura 8, a curva do motor 2.



**Figura 7: Curva rotação x tensão para o motor 1**



**Figura 8: Curva rotação x tensão para o motor 2**

Analisando-se as duas curvas nota-se que os dois motores apresentam desempenhos praticamente iguais. Isto era esperado, pois os dois motores movem cargas bastantes parecidas. Pode-se considerar também, que a velocidade de rotação do eixo varia linearmente com a tensão de alimentação dos motores.

### **4.3 Sensores de Posição**

Neste projeto utiliza-se dois sensores de posição, um para cada grau de liberdade do mecanismo. A escolha destes sensores ocorreu de forma diferente da escolha dos motores. O procedimento para a escolha foi baseado nos requisitos e nas limitações encontradas no projeto.

Para o projeto, estaríamos interessados em sensores que medissem deslocamentos angulares, inicialmente, com uma boa precisão, de forma a garantir um movimento preciso do mecanismo.

Para esta aplicação, seria mais indicada a utilização de encoders óticos, que além de possuírem boa precisão (obviamente, dependendo do modelo e do preço), possuem uma saída digital, o que elimina o conversor de tensão analógico-digital. Porém, não havia a disponibilidade deste tipo de sensor no Departamento e o custo de um encoder ótico é elevado para os fins deste projeto.

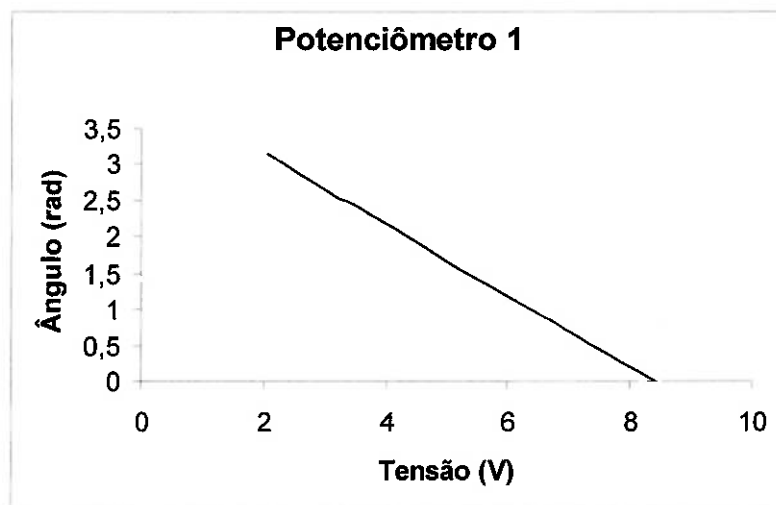
Foi ainda cogitada a fabricação de um encoder, porém a dupla concluiu que isto estaria fora do escopo do projeto.

Portanto, a solução mais viável encontrada foi a utilização de potenciômetros, visto que, normalmente, estes dispositivos apresentam boa linearidade e baixo custo, além da facilidade de operação e manuseio. Porém, a precisão na leitura da posição angular ficaria um pouco prejudicada, se comparada ao encoder, e nota-se ainda a necessidade da utilização de um conversor A/D.

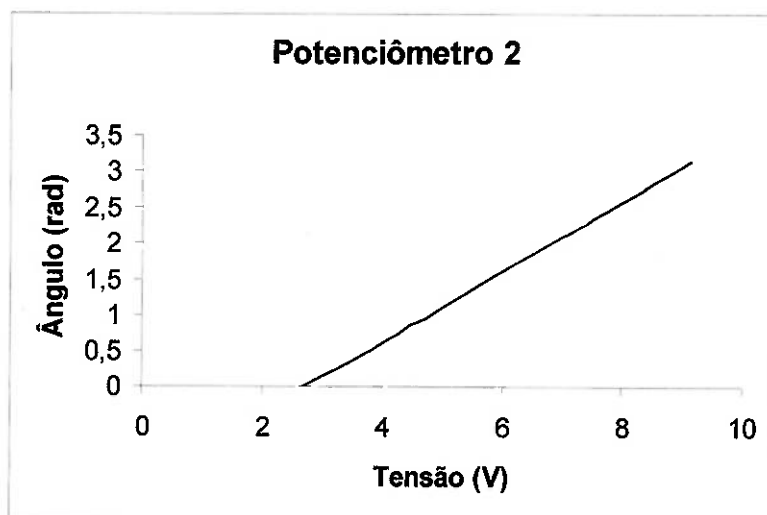
Como cada haste do mecanismo se movimenta entre  $0^\circ$  e  $180^\circ$  em torno do seu referencial, os potenciômetros utilizados não precisam completar um volta completa, ou seja, podem ser utilizados potenciômetros comuns com o curso de  $360^\circ$ .

Os potenciômetros devem possuir uma resistência interna compatível com a corrente envolvida. Segundo o manual da placa conversora CAD 12/36, a faixa de corrente consumida varia de 50 mA a 650 mA, e portanto a resistência dos potenciômetros deve estar numa faixa de 8 a 50 ohms. Tendo isto em vista, foram utilizados dois potenciômetros Hitech de 20 ohms cada.

Os potenciômetros foram submetidos a uma calibração para a determinação da relação entre a posição angular e a tensão medida por este. O experimento foi realizado com os potenciômetros já montados no protótipo, procedendo-se da seguinte maneira: para a haste maior, fixou-se uma referência com ângulos marcados no centro de rotação, solidário à base, e mediu-se para cada posição a tensão correspondente nos terminais do potenciômetro; para haste menor, o procedimento foi o mesmo, porém a referência estava solidária a haste maior. Em ambos os casos, a leitura dos potenciômetros deveria estar entre 0 V e +10 V, pois o modo de conversão da placa conversora está "setado" para converter valores de tensão dentro desta faixa. Isto será visto com maiores detalhes no próximo item. Então os potenciômetros foram alimentados com uma tensão de 12 V e os ângulos de 0° a 180° ficaram situados em uma faixa de no mínimo 0 V a no máximo 10 V. As curvas que relacionam ângulo e tensão estão apresentadas nas figuras 9, para o potenciômetro da haste 1, e 10, para o da haste 2.



**Figura 9: Curva ângulo x tensão - potenciômetro 1**



**Figura 9: Curva ângulo x tensão - potenciômetro 2**

Analisando-se as duas curvas, nota-se que os potenciômetros apresentam um comportamento linear e bastante similar, com a diferença que a leitura do potenciômetro 1 cresce no sentido horário de rotação e a do potenciômetro 2, no sentido anti-horário, devido à utilização do par de engrenagens, que transmite a rotação da haste 1 para o eixo do potenciômetro 1.



De posse destes gráficos, foram feitas aproximações por retas, chegando-se às seguintes relações para o potenciômetro 1 e 2, respectivamente:

$$A_1 = -0.52017 \cdot V_1 + 4.0104$$

$$A_2 = 0.50346 \cdot V_2 - 1.3274$$

#### **4.4 Placa Conversora A/D e D/A**

Como foi visto no layout do sistema, estava prevista a utilização de uma placa conversora analógico-digital, para converter o sinal de tensão analógico (entrada) proveniente dos potenciômetros em uma tensão digital a ser utilizada no software de controle. Em seguida, o sinal de tensão de controle necessita ser convertido de digital para analógico (saída), para o acionamento controlado dos motores.

A placa utilizada é a CAD 12/36 da Lynx Tecnologia Eletrônica Ltda. Ela possui 15 entradas analógicas e quatro saídas analógicas sendo que o interfaceamento com os sinais externos é feito por meio de um conector do tipo DB-37.

Ao ler um sinal analógico em volts, a placa o converte em um sinal digital com resolução de 12 bits, que é apresentado na forma de dois bytes (byte A e byte B). Os 12 bits estão alinhados à esquerda, sendo que os 4 primeiros bits são iguais a zero. Os 12 bits variam de 0000.0000.0000 a 1111.1111.1111. Assim, os dados formam números que variam de 0000.0000.0000.0000 a 1111.1111.1111.0000.

A interpretação destes números depende do formato escolhido. No caso deste projeto, a placa foi ajustada para converter os sinais no modo unipolar, com ganho igual a 1 e formato binário. O modo unipolar indica que o conversor converte um sinal de 0 V a +10/ ganho Volts, ou seja, o sinal a ser convertido deverá estar entre 0 V e +10V.

A conversão da tensão de entrada está representada na tabela 1.

*Tabela 1: Conversão da tensão de entrada*

Byte B								Byte A								Tensão Entrada	Código
D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0000 V	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0.0049 V	16
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	9.9902 V	65504
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	9.9951 V	65520

Desta forma, os sinais são convertidos segundo a relação:

$$T = \frac{10 \cdot L}{65536}$$

onde:

T= tensão na entrada;

L= leitura do conversor.

Esta tensão de entrada será utilizada no programa para o cálculo da tensão que deverá ser enviada ao motor. Esta tensão de saída deverá ser convertida na forma de um número, que será escrito na porta de saída da placa, que irá transformá-lo em um sinal de tensão analógico. Para tanto, a placa foi ajustada para converter no modo binário, bipolar com ganho igual a 2, ou seja, a tensão na saída será de -5 V a +5V.

A tabela 2 mostra melhor como é feita a conversão.

*Tabela 2: Conversão digital-analógica*

Byte B								Byte A								Tensão
D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0	Saída
0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	-5.0000 V
0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	-4.9976 V
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	0.0000 V
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	1	1	1	1	1	1	1	1	1	1	1	X	X	X	X	4.9976V

Portanto, a relação entre o número a ser escrito na placa (N) e a tensão a ser enviada (V) é a seguinte:

$$N = 4096 \cdot \left( 1 - \frac{5 - V}{10} \right)$$

#### 4.5 Circuito de Potência

A função do circuito de potência é amplificar a corrente enviada, da placa conversora, aos motores, de modo a efetuar o acionamento dos mesmos. Porém, a diferença de potencial na entrada deve se manter a mesma na saída, visto que esta é a tensão de controle.

Para o projeto, foi utilizado um circuito bastante simples, composto por um amplificador operacional LM 12 e dois resistores de 10 kΩ, para cada motor.

A figura 11 mostra a representação esquemática do circuito implementado.

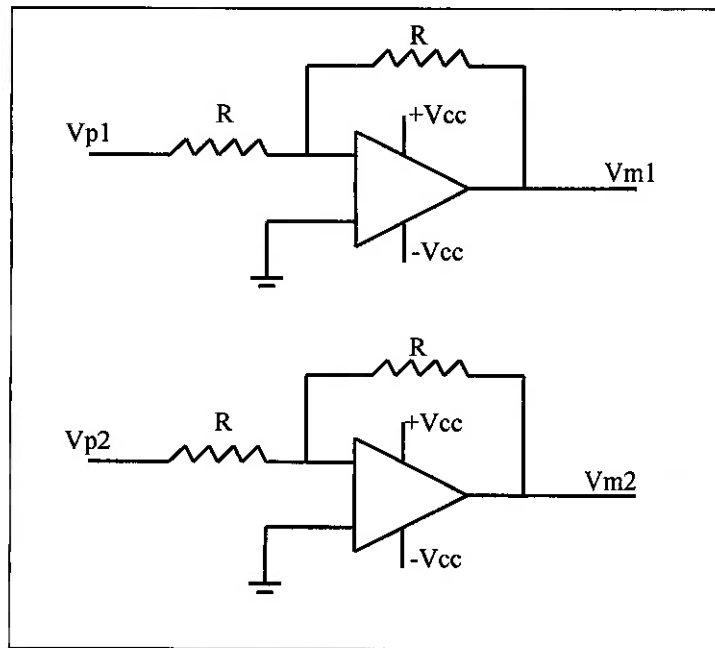


Figura 101: Representação do circuito amplificador

Para se obter a equação do circuito amplificador, vamos analisar o circuito de alimentação do motor 1. A corrente  $i_1$  é a corrente de entrada do circuito.

$$i_1 = \frac{V_{P1} - V_T}{R_1}$$

$$i_2 = \frac{V_T - V_{M1}}{R_2}$$

A corrente que flui para o amplificador ( $i_2$ ) é muito pequena e pode ser desprezada. Assim:

$$i_1 = i_2$$

$$V_{M1} = -\frac{R_2}{R_1} \cdot V_{P1}$$

Como as resistências são iguais, o ganho do amplificador é igual a  $-1$ . Portanto,  $V_{m1}$  é igual a  $-V_{p1}$ .

## **5. Integração Física**

Este item tem como objetivo fornecer uma melhor visualização de como os componentes do sistema se relacionam fisicamente.

Como foi visto no layout do sistema (item 1.3), os motores atuam no mecanismo fazendo-os movimentar. Os deslocamentos angulares das hastes são lidos pelos potenciômetros acoplados aos eixos de rotação na forma de uma diferença de potencial. Estes sinais de tensão, que variam entre 0 V e 10 V, entram na placa conversora A/D através do conector DB-37. Após a conversão do sinal digital para analógico (-5 V a +5 V), a placa o envia para a entrada do circuito amplificador de potência, também através do conector DB-37. O circuito amplifica a corrente que vem da placa (da ordem de 30 mA) e conduz aos terminais dos motores (da ordem de 1.5 A), sem alterar o valor da tensão, que movem as hastes, fechando o ciclo.

Duas fontes de tensão são utilizadas neste projeto: uma tem a função de alimentar o circuito de potência com +Vcc igual a +12 V e -Vcc igual a -12V, além de fornecer o terra de referência; e a outra alimenta os potenciômetros com tensão entre 0 V e +12 V.

Já a interface entre o usuário e o mecanismo é feita através do software implementado no computador, que será comentado mais adiante.

## 6. Modelagem Matemática

Para modelar os motores, considerou-se que os mesmos são independentes entre si e, portanto, a modelagem é a mesma para os dois motores. Adotou-se o seguinte circuito mostrado na figura 13 [2] :

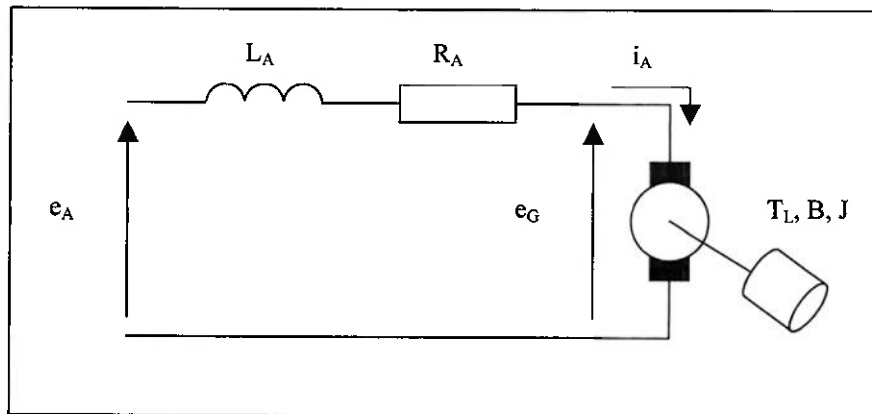


Figura 12: Circuito equivalente do motor CC com carga

Deste modelo obteve-se a seguinte equação diferencial:

$$e_A = e_G + R_A \cdot i_A + L_A \cdot \frac{di_A}{dt} \quad (1)$$

onde:

$e_A$  = tensão aplicada à armadura da máquina;

$e_G$  = força contra eletromotriz gerada;

$R_A$  = resistência de armadura;

$L_A$  = indutância da armadura.

A força contra eletromotriz é proporcional ao fluxo gerado pelo circuito de excitação  $\phi$  e à rotação  $n$ :

$$e_G = C \cdot \phi \cdot n \quad (2)$$

O torque eletromagnético desenvolvido pelo motor é proporcional ao produto da corrente de armadura  $i_A$  pelo fluxo  $\phi$ :

$$T_e = K \cdot \phi \cdot i_A \quad (3)$$

Ao se realizar o balanço de torque, leva-se em consideração o torque da carga  $T_L$ , o torque devido ao atrito e ao momento de inércia:

$$T_e = T_L + B \cdot n + J \cdot \frac{dn}{dt} \quad (4)$$

onde:

$B$  = constante de atrito;

$J$  = momento de inércia.

Deve-se notar que é de nosso interesse representar as equações em função da posição angular  $\theta$ . Foi considerado que o fluxo eletromagnético é constante, obtendo-se das equações (2), (3), (4) acima:

$$e_G = C \cdot \frac{d\theta}{dt} \quad (5)$$

$$T_e = K \cdot i_A \quad (6)$$

$$T_e = T_L + B \cdot \frac{d\theta}{dt} + J \cdot \frac{d^2\theta}{dt^2} \quad (7)$$

Supondo-se condições iniciais nulas e aplicando-se a Transformada de Laplace nas equações (1), (5), (6) e (7), obtém-se:

$$E_A(s) = E_G(s) + R_A \cdot I_A(s) + L_A \cdot s \cdot I_A(s) \quad (8)$$

$$E_G(s) = C \cdot s \cdot \theta(s) \quad (9)$$

$$T_E(s) = K \cdot I_A(s) \quad (10)$$

$$T_E(s) = T_L(s) + B \cdot s \cdot \theta(s) + J \cdot s^2 \cdot \theta(s) \quad (11)$$

Manipulando-se a equação (8), chega-se à seguinte expressão:

$$I_A(s) = \frac{E_A(s) - E_G(s)}{R_A + s \cdot L_A} = \frac{\frac{E_A(s) - E_G(s)}{R_A}}{1 + \tau_e \cdot s} \quad (12)$$

onde:

$$\tau_e = \frac{L_A}{R_A}, \text{ constante de tempo elétrica do motor.}$$

Da equação (11), tem-se:

$$\theta(s) = \frac{T_E(s) - T_L(s)}{B \cdot s + s^2 \cdot J} = \frac{\frac{T_E(s) - T_L(s)}{B}}{s \cdot (1 + \tau_m \cdot s)} \quad (13)$$

onde:

$$\tau_m = \frac{J}{B}, \text{ constante de tempo mecânica do sistema.}$$

Através destas relações é possível obter o seguinte diagrama de blocos do modelo do motor, mostrado na figura 13:



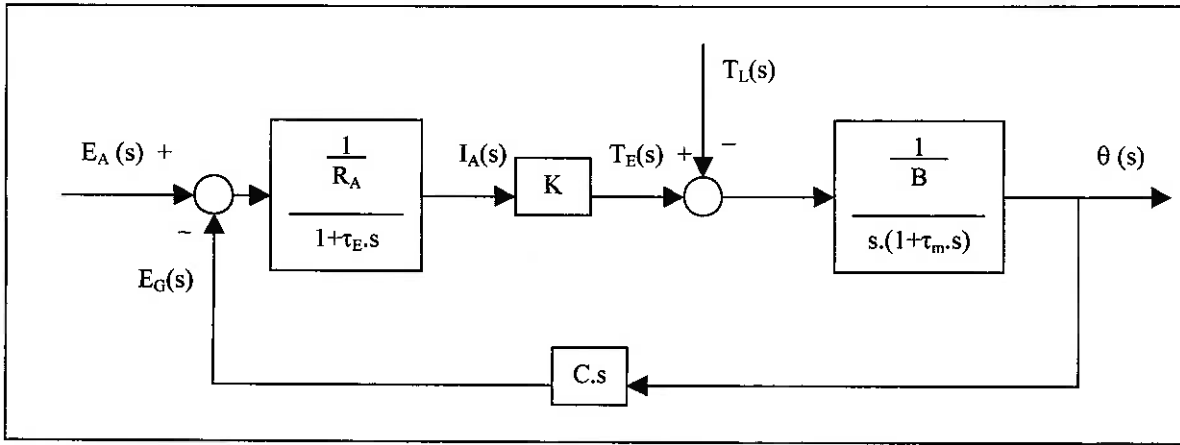


Figura 13: Diagrama de blocos do motor

Simplificando-se o diagrama de blocos e considerando-se o torque de carga nulo, a relação entre a posição angular e a tensão aplicada à armadura fica:

$$\frac{\theta(s)}{E_A(s)} = \frac{K}{s \cdot [R_A \cdot B \cdot (1 + \tau_e \cdot s) \cdot (1 + \tau_m \cdot s) + K \cdot C]}$$

Como na maioria dos casos  $\tau_e \ll \tau_m$ ,  $\tau_e$  pode ser desprezado sem erros apreciáveis [1], logo:

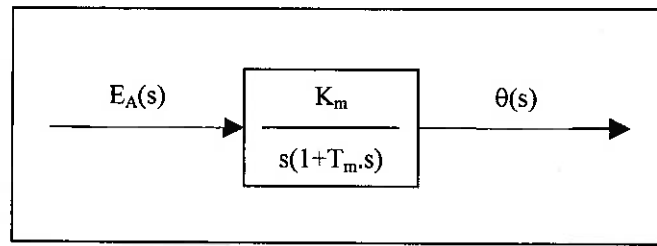
$$\frac{\theta(s)}{E_A(s)} = \frac{K}{s \cdot (R_A \cdot B \cdot \tau_m \cdot s + R_A \cdot B + K \cdot C)} = \frac{K_m}{s \cdot (1 + T_m \cdot s)} \quad (14)$$

onde:

$$K_m = \frac{K}{R_A \cdot B + K \cdot C}$$

$$T_m = \frac{R_A \cdot B}{R_A \cdot B + K \cdot C} \cdot \tau_m$$

Com isso, o diagrama simplificado da planta fica:



**Figura 114: Diagrama simplificado do sistema**

Para esta modelagem, foi considerado um motor de corrente contínua genérico. Os valores de  $K_m$  e  $T_m$  são obtidos pelas características construtivas e elétricas dos motores utilizados.

## 7. Teoria de Controle

### 7.1 Sistemas de Segunda Ordem

Nesta parte será feita uma breve apresentação da teoria de controle utilizada para o projeto de um controlador.

O modelo matemático do motor elétrico de corrente contínua, apresentado no item anterior, revela que este se caracteriza por ser um sistema de segunda ordem.

Um sistema de segunda ordem pode ser descrito pela seguinte função de transferência genérica:

$$\frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

Onde:

$Y(s)$  é a saída do sistema;

$U(s)$  é a entrada do sistema;

$\omega_n$  é a frequência natural do sistema;

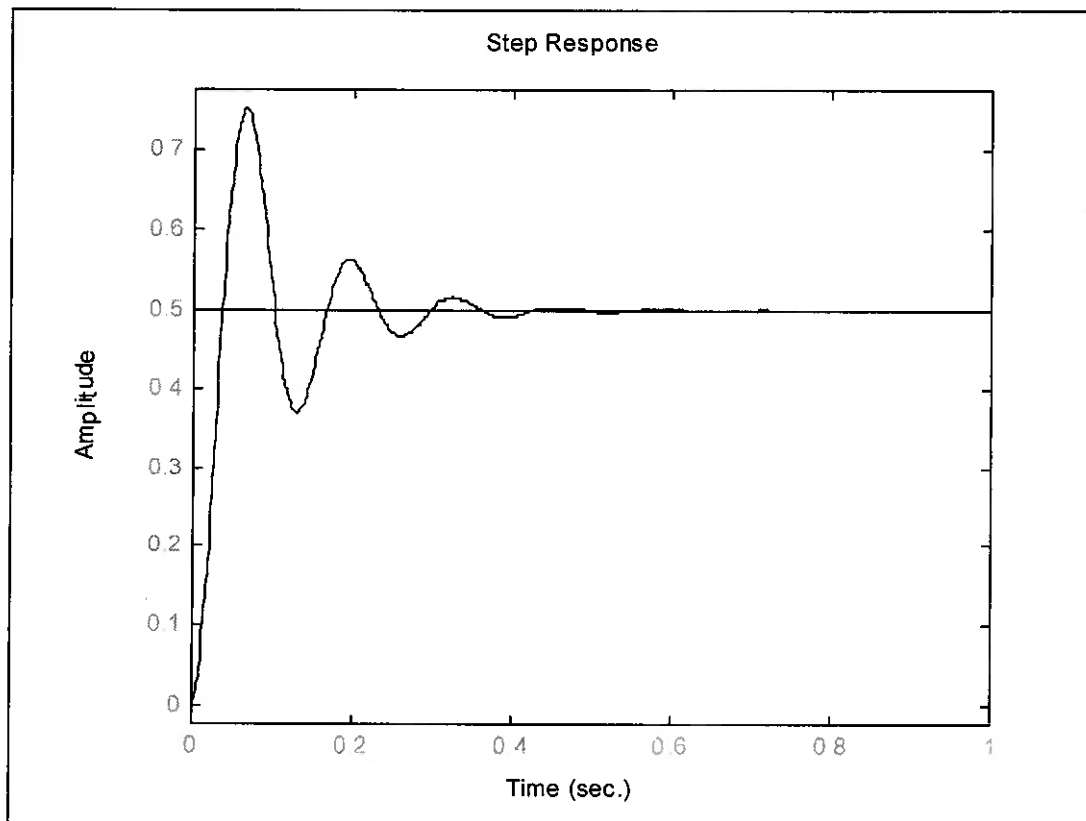
$\xi$  é o grau de amortecimento.

No plano complexo, pode-se determinar os pólos do sistema de segunda ordem e obter algumas de suas características como, por exemplo, a frequência natural e o coeficiente de amortecimento.

- grau de amortecimento:  $\xi = \cos(\beta) = \frac{\sigma}{\omega_n}$
- $\beta = \arccos\left(\frac{\sigma}{\omega_n}\right)$

As características de desempenho desse tipo de sistema são geralmente especificadas em termo de resposta transitória para uma entrada em degrau unitário,

devido a facilidade de sua geração e análise matemática. A resposta típica de um sistema de segunda ordem a um degrau unitário é mostrada na figura 15.



**Figura 125: Resposta de um sistema de segunda ordem.**

Através da figura 15 pode-se determinar algumas características importantes relacionadas ao desempenho do sistema, a saber:

- tempo de assentamento 2%:  $t_s \cong \frac{4}{\xi\omega_n}$
- tempo de subida:  $t_r = \frac{\pi - \beta}{\omega_d}$
- máximo sobressinal:  $M_p = \exp\left(\frac{-\pi\xi}{\sqrt{1-\xi^2}}\right)$

Analisando-se a figura 15, nota-se que o tempo de assentamento de 2% é o tempo que a saída do sistema leva para chegar a 98% do valor de referência desejado para uma entrada degrau unitário. O tempo de subida é o tempo necessário para a saída cruzar pela primeira vez o valor de referência, no caso de sistemas com sobressinal ou o tempo que leva para atingir 90% do mesmo valor, no caso de sistemas sobre-amortecidos. O máximo sobressinal é a diferença percentual entre o máximo valor atingido pela saída e o valor de referência.

## 7.2 Controladores

Dentre as várias táticas de controle existentes, vamos aqui abordar o tipo de controle mais normalmente utilizado: o controlador do tipo PID.

Os controladores da classe dos PID são consagradamente utilizados nesta aplicação de controle de motores elétricos de corrente contínua.

Um controlador do tipo proporcional (P) é caracterizado por representar um ganho puro (K) que multiplica a função de transferência da planta. É o tipo de controlador mais simples, porém apresenta a característica de não eliminar o erro em regime. Sua função de transferência é dada por:

$$U(s) = K$$

Um compensador tipo proporcional-derivativo (PD) é dado pela seguinte equação:

$$U(s) = K(1 + T_d s)E(s)$$

onde:

K é o ganho do controlador

T<sub>d</sub> é o tempo de derivação

U(s) é a entrada do sistema controlado

E(s) é o erro entre o valor desejado e o valor instantâneo da saída do sistema

Uma característica da adição do termo derivativo é a diminuição do sobressinal e do tempo de assentamento do sistema.

Um compensador do tipo proporcional-integral (PI) é dado pela seguinte equação:

$$U(s) = K\left(1 + \frac{1}{T_i s}\right)E(s)$$

onde, T<sub>i</sub> é o tempo de integração.

A adição do termo integrativo elimina o erro em regime do sistema, porém tende a aumentar o sobressinal.

Um compensador tipo proporcional-derivativo-integral (PID) pode ser representado por:

$$U(s) = K\left(1 + T_d s + \frac{1}{T_i s}\right)E(s)$$

Este controlador é o mais completo e apresenta o melhor desempenho no sistema, reunindo as características de todos os termos. Porém, dependendo dos requisitos do sistema e do próprio sistema, consegue-se encontrar um controlador mais simples que satisfaça os requisitos.

O projeto de um compensador tipo PID pode ser realizado teoricamente por dois métodos: Lugar das Raízes ou no domínio em frequência, analisando-se os Diagramas de Bode. Porém, neste projeto, optou-se por um projeto mais prático utilizando o programa MATLAB, onde pode-se ajustar os valores dos ganhos de cada termo do PID até conseguir encontrar um controlador que satisfaça os requisitos de projeto.

### **7.3 Controlador Digital**

Até este ponto o sistema foi tratado no domínio contínuo ou em  $s$ . Entretanto, para a implementação do controlador via software, faz-se necessário passar do domínio contínuo para o domínio discreto.

O controlador digital é obtido a partir da “discretização” do controlador analógico projetado. Existem vários métodos para se achar o equivalente discreto do controlador analógico, porém é desejável utilizar o método que melhor preserva as características de resposta transitória e resposta em frequência. Deve-se lembrar que a frequência de amostragem do sinal também é um fator que influi para a obtenção de um bom controlador discreto.

Os métodos que serão descritos são: Método de transformação bilinear e o Método do casamento de pólos e zeros. Estes métodos baseiam-se na transformação da função de transferência do controlador de  $s$  para  $z$ .

### **7.3.1 Método de Transformação Bilinear**

Também chamado de Método da Integração Trapezoidal ou de Tustin, este método utiliza a substituição da variável  $s$  por uma função na variável  $z$  dada por:

$$s = \frac{2}{T_a} \left( \frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

onde  $T_a$  é o período de amostragem.



### 7.3.2 Método do Casamento de Pólos e Zeros

Este método corresponde a realizar um mapeamento dos pólos e zeros da função de transferência contínua  $G(s)$  para o plano complexo discreto  $z$ . Para tanto, os zeros e os pólos em  $s$  são mapeados em  $z = e^{T_a s}$ .

No caso de um controlador PI, tem-se:

O zero de  $G(s)$  em  $s = -1/T_i$  é mapeado em  $z = e^{\frac{-T_a}{T_i}}$  em  $G(z)$ ;

O pólo de  $G(s)$  em  $s = 0$  é mapeado em  $z = 1$ , ou seja, na origem de  $G(z)$ .

Para se ajustar o ganho  $K_z$  de  $G(z)$  deve-se fazer com que a resposta a um impulso do controlador digital tenha o mesmo valor final da resposta do controlador analógico, considerando-se também que se trata de um sistema do tipo 1, ou seja:

$$\boxed{T_a \lim_{s \rightarrow 0} [sG(s)] = \lim_{z \rightarrow 1} [(z-1)G(z)]}$$

Portanto,  $G(z)$  para um controlador PI seria da forma:

$$\boxed{G(z) = K \frac{z - e^{\frac{-T_a}{T_i}}}{z - 1}}$$

### 7.4 Equações de Diferenças

No domínio do tempo discreto, o equivalente no tempo a uma função de transferência na variável  $z$ , é uma equação de diferenças, que é obtida da seguinte forma. Seja a seguinte função de transferência de ordem  $n$ :

$$\boxed{\frac{Y(z)}{U(z)} = G(z) = \frac{b_m z^m + b_{m-1} z^{m-1} + \dots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0}}$$

Deve-se colocar  $G(z)$  em função de potências negativas de  $z$  e multiplicar  $Y(z)$  pelo denominador de  $G(z)$  e  $U(z)$  pelo numerador de  $G(z)$ . Feito isso, aparecerão

fatores do tipo  $z^n Y(z)$ , que no plano  $z$  significa no tempo a variável  $y(kT_a)$  atrasada de  $nT_a$ , onde  $kT_a$  é o tempo corrido.

Deste modo obtém-se a seguinte equação de diferenças:

$$y(kT_a) = -a_{n-1}y(kT_a - T_a) - \dots - a_1y(kT_a - (n-1)T_a) - a_0y(kT_a - nT_a) + \\ + b_mu(kT_a - (n-m)T_a) + b_{m-1}u(kT_a - (n+1-m)T_a) + \dots + b_1u(kT_a - (n-1)T_a) + b_0u(kT_a - nT_a)$$

Esta equação representa uma fórmula de recorrência, onde conhecendo-se os valores passados de  $y(kT_a)$  e os valores de entrada  $u(kT_a)$  pode-se avançar no tempo discretamente.

## 8. Simulação Numérica

A simulação de um sistema, numericamente, tem a função de prever o comportamento do sistema teoricamente. Estas simulações foram realizadas com o auxílio do programa MATLAB/SIMULINK.

Para realizar o controle dos motores, assumiu-se que os mesmos são independentes entre si. Esta simplificação não implica grandes erros, como visto anteriormente. A simulação numérica foi realizada a partir dos dados apresentados na tabela 3.

Tabela 3: Parâmetros dos motores

MOTOR	J (kg.m <sup>2</sup> )	B (N.m.s)	Ke=Kt (N.m/a)	R (Ω)
1	1.07x10 <sup>-3</sup>	3.5077x10 <sup>-6</sup>	0.3857	6.0
2	1.13 x10 <sup>-3</sup>	3.5077x10 <sup>-6</sup>	0.3857	5.5

A função de transferência obtida em (14) fica:

$$\frac{\theta(s)}{E_A(s)} = \frac{K}{R \cdot J \cdot s^2 + R \cdot B \cdot s + K^2 \cdot s}$$

### 8.1. Resposta em malha aberta

O sistema em malha aberta apresenta a seguinte estrutura:

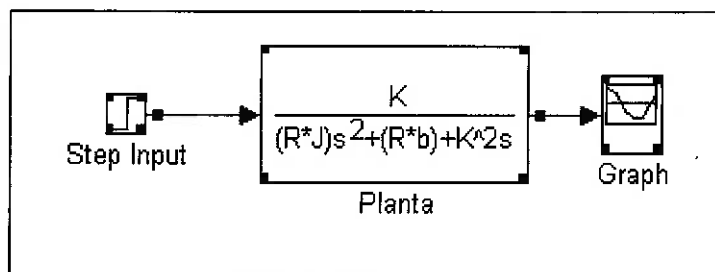
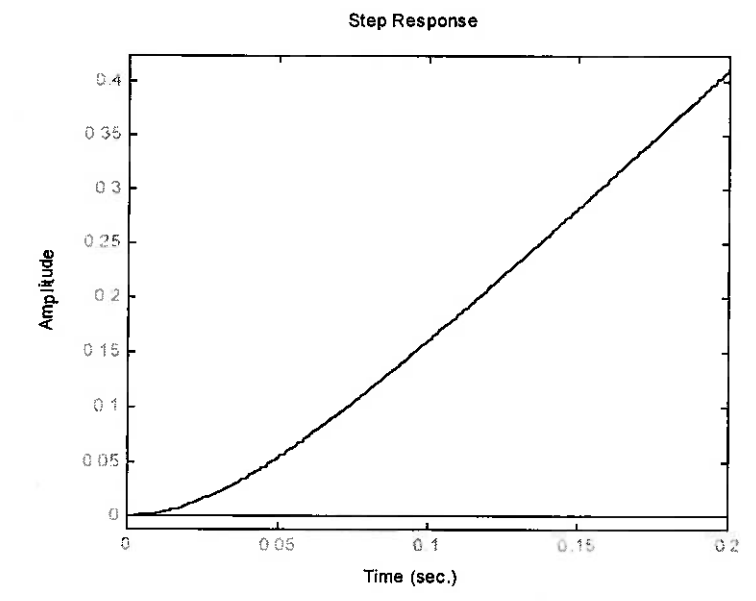


Figura 136: Sistema em malha aberta

Aplicando-se uma entrada degrau de 1 Volt ao sistema em malha aberta, obteve-se os seguintes gráficos para os motores:

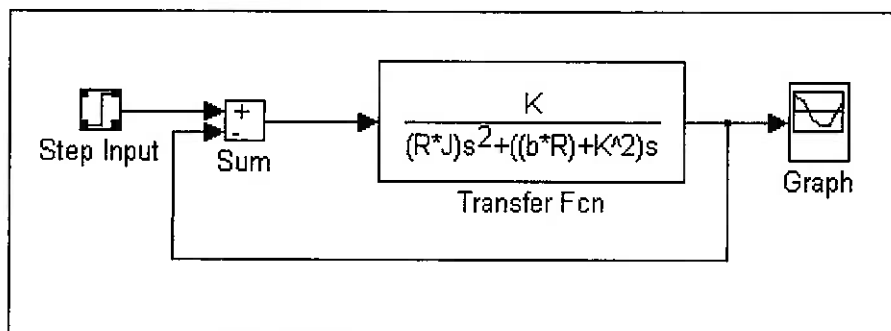


**Figura 17 - Resposta do sistema em malha aberta - motor 1**

Pela figura 17 percebe-se que o motor entra em regime rapidamente.

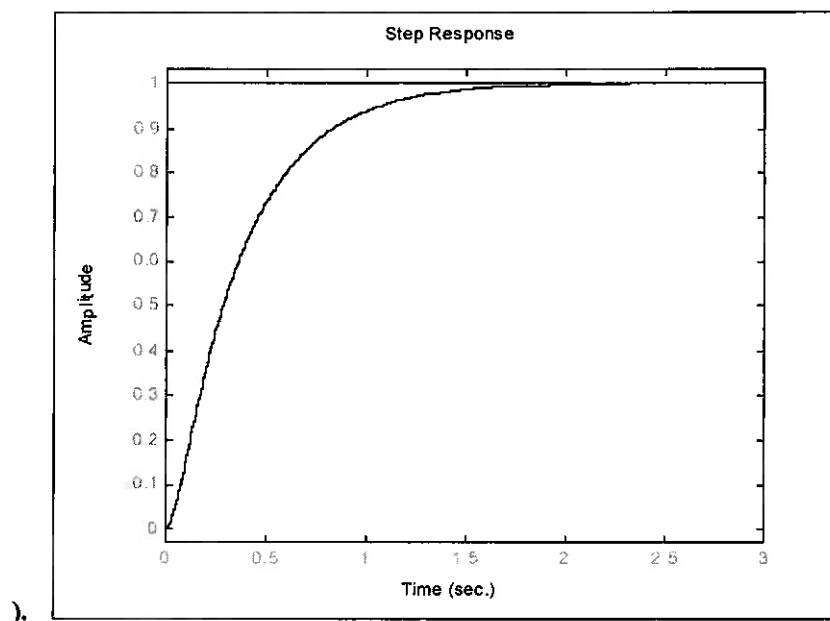
## 8.2 Resposta em malha fechada

O sistema em malha fechada sem controlador, apresenta a seguinte estrutura:



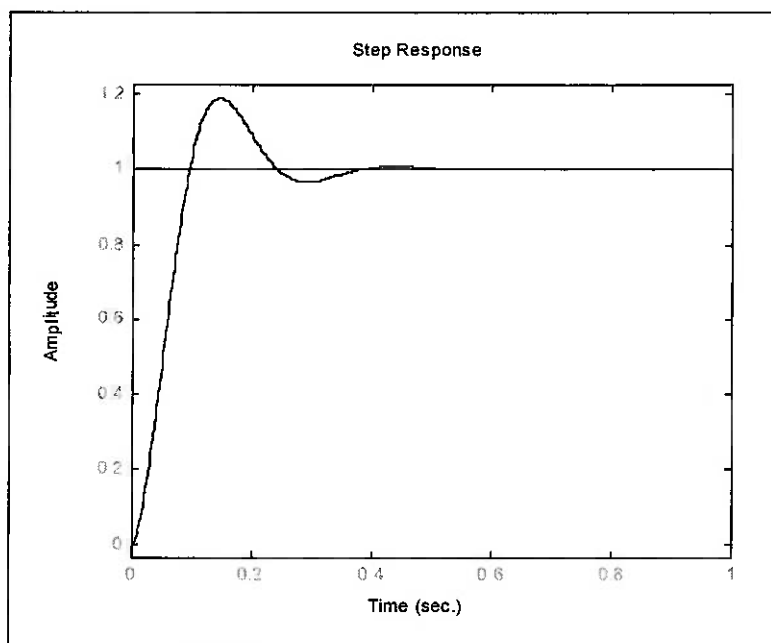
**Figura 18: Sistema em malha fechada**

A resposta à entrada degrau do sistema em malha fechada sem controlador está apresentada na figura 19.



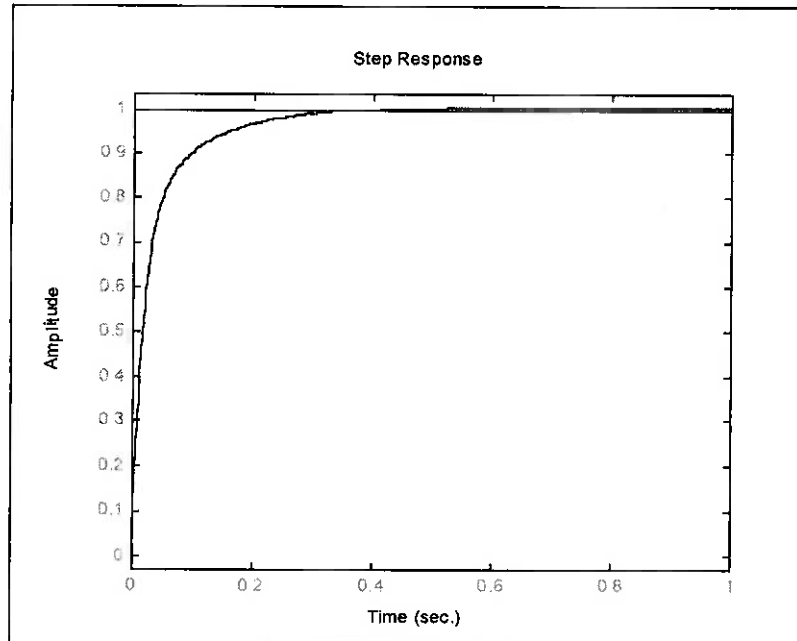
**Figura 149: Resposta em malha fechada**

O sistema em malha fechada sem controlador apresenta uma resposta à entrada degrau insatisfatória, uma vez que o tempo de acomodamento é muito alto. Para diminuir o tempo de acomodamento, foi implementado um controlador proporcional, com  $K_p=10$ , obtendo-se a seguinte resposta à entrada degrau unitário:



**Figura 15: Resposta com o controlador proporcional -  $K_p = 10$**

O controlador proporcional implementado, como previsto, apesar de ter atingido um tempo de acomodamento da resposta do sistema satisfatório, apresentou um sobressinal muito grande. Para minimizar este sobressinal, foi implementado um controlador PID com os seguintes parâmetros:  $K_p=10$ ;  $K_i=1.5$ ;  $K_d=0.75$ .



**Figura 16: Resposta com um controlador PID -  $K_p = 10$   $K_i = 1.5$   $K_d = 0.75$**

Como pode ser visto pelo gráfico acima, o controlador PID apresentou tempo de acomodamento e sobressinal satisfatórios. O sistema controlado responde à entrada degrau unitário em menos de 0.35s e não há sobressinal.

O projeto do controlador teórico do motor 2 é análogo ao primeiro motor, uma vez que apresentam parâmetros similares.

Após projetar o controlador, deve-se discretizá-lo, de modo a ser possível a sua implementação via software.

Utilizando-se o MATLAB e o método de casamentos de pólos e zeros, encontramos o seguinte controlador digital:

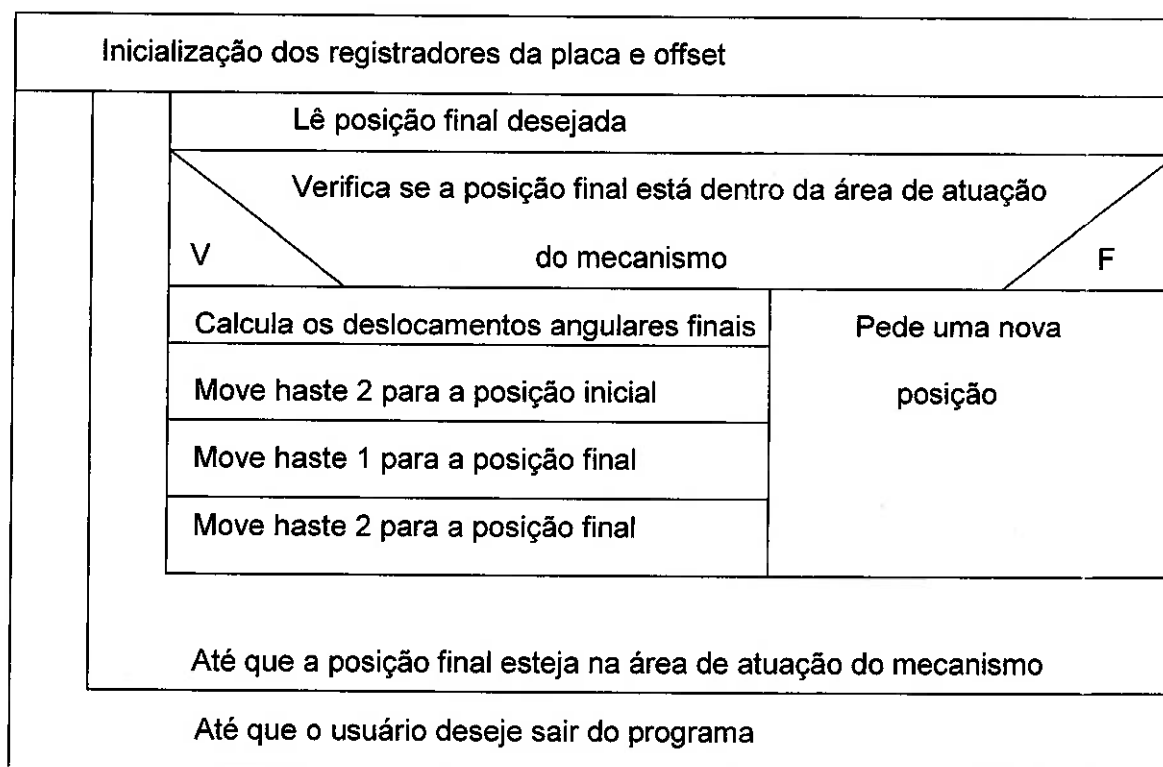
$$G_D(z) = \frac{(4.3845 \cdot z^2 - 8.7109 \cdot z + 4.3264) \cdot 10^{-2}}{z^3 - 2.9335 \cdot z^2 - 0.9340}$$

## 9. Software de Controle

O controle do sistema é efetivamente realizado pelo software de controle, implementado no computador, em linguagem Pascal. A listagem do programa (Program Pico) encontra-se em anexo, porém faz-se necessária a explicação do funcionamento do mesmo, através de Diagramas de Nassi-Schneidermann.

### 9.1 Programa Principal

O programa principal desenvolvido para o controle do mecanismo de posição apresenta a seguinte estrutura:



O programa é inicializado mandando um sinal de offset para os motores, de modo que os mesmos permanecerão parados. A placa é inicializada programando-se o modo 0; os canais de entrada analógica para ganho 1 e modo unipolar ; auto-

calibrando-se o conversor A/D; esvaziando-se a FIFO e, por fim, programando-se o modo da placa: DMA e BURST desligado, sem timers ou interrupções e modo polling.

## **9.2 Procedimento que calcula os deslocamentos angulares das hastes**

Para realizar o cálculo dos deslocamentos angulares das hastes, é necessário conhecer os ângulos iniciais das mesmas. Os ângulos iniciais são obtidos através do procedimento "Lê\_entradas" que faz a leitura das tensões dos potenciômetros, que é convertida para radianos.

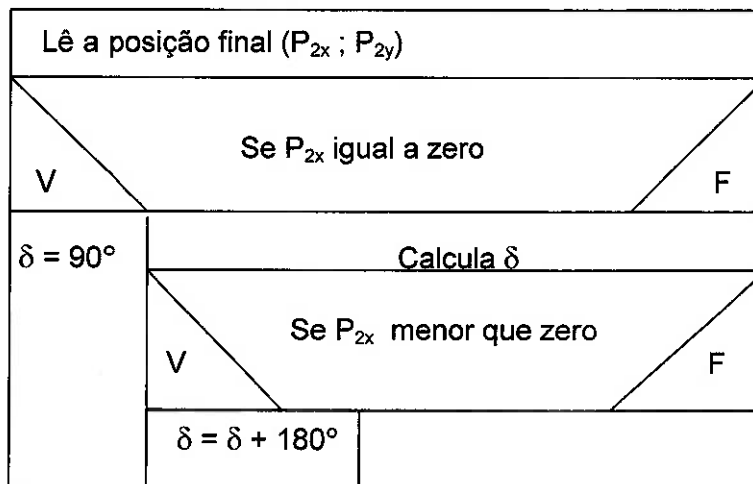
O procedimento abaixo, que realiza o cálculo dos deslocamentos angulares das hastes, é a implementação do algoritmo já apresentado:

Calcula distância entre a articulação O e a posição final ( $z_3$ )
Calcula o ângulo $\alpha$ formado por $z_3$ e a haste 1
Calcula o ângulo $\delta$ da inclinação de $z_3$
Calcula os ângulos $\theta_2$ e $\theta'_2$ e seleciona o menor entre eles
Calcula a posição final da articulação 2 ( $A_2$ )
Calcula o ângulo $\gamma$ formado pela haste 2 e o eixo x
Calcula o ângulo $\beta_2$ que a haste 2 movimentará
Calcula os ângulos finais das hastes



### 9.2.1. Diagrama do cálculo do ângulo $\delta$

Observando-se as condições para o cálculo do ângulo  $\delta$ , obtém-se o seguinte diagrama de Nassi-Schneidermann:



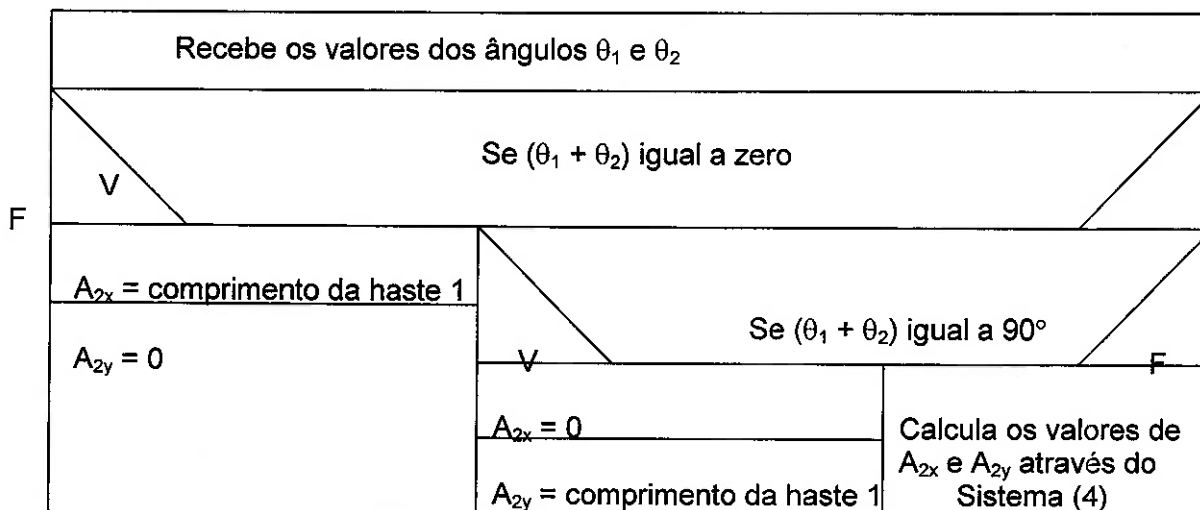
### 9.2.2. Diagrama do cálculo da posição final da articulação 2 ( $A_2$ )

Para determinar a posição  $A_2$ , é necessário o cálculo da tangente da soma dos ângulos  $\theta_1$  e  $\beta_2$ , observando-se as condições da seção 3.1.2.

Se  $(\theta_1 + \theta_2)$  for igual a zero,  $A_{2x} = z_1$  (comprimento da haste 1) e  $A_{2y} = 0$ .

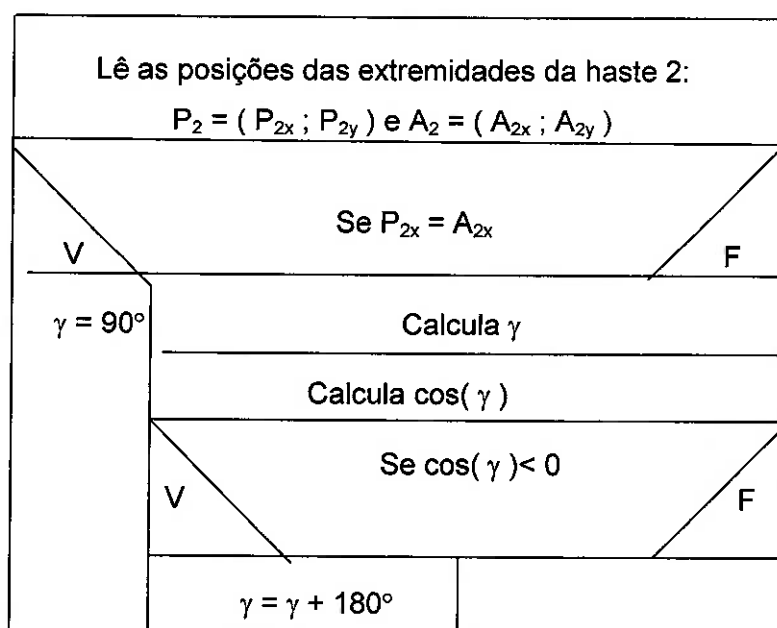
Se  $(\theta_1 + \theta_2)$  for igual a  $90^\circ$ ,  $A_{2x} = 0$  e  $A_{2y} = z_1$ .

Para outros valores, calcula-se  $A_{2x}$  e  $A_{2y}$  pelo sistema (4).



### 9.2.3. Diagrama do cálculo do ângulo $\gamma$

Como nos casos anteriores, o cálculo do ângulo  $\gamma$ , analogamente a  $\delta$ , implica na análise das condições impostas em 3.1.2.



### 9.3. Procedimento que move a haste 2 para a sua posição inicial

O protótipo foi construído de modo que o ângulo de atuação da haste 2 fosse de  $180^\circ$ . Para que a haste 1 movimente-se livremente, sem que a haste 2 chegue nos seus ângulos extremos ( limitados pela construção do protótipo ), deve-se realizar uma inicialização da haste 2. Esta inicialização é realizada analisando o sentido de rotação que a haste 1 realizará para chegar na posição final desejada.

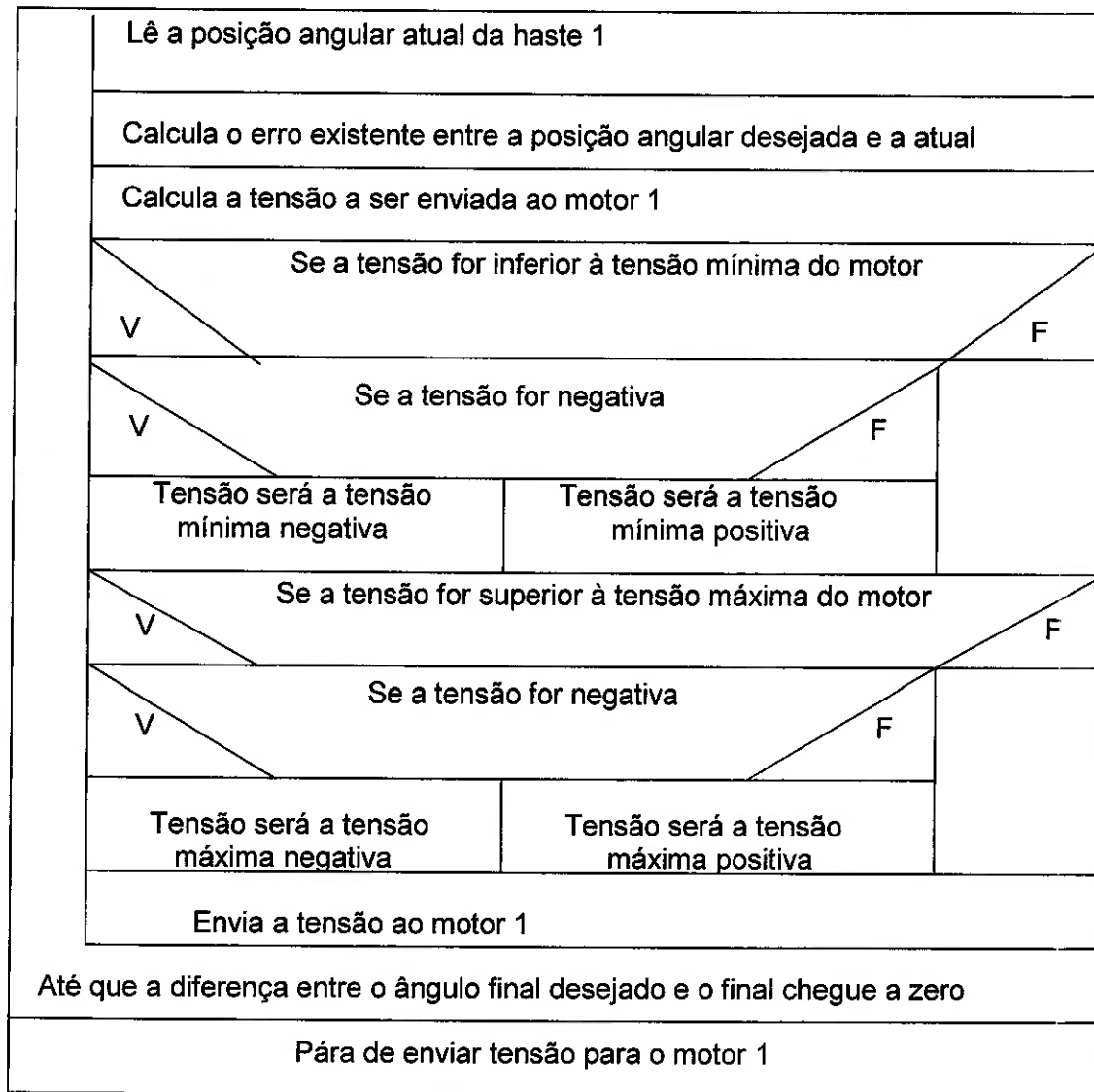
Haste 1 movimentará em sentido anti-horário	
V	F
Ângulo final da haste 2 vale $180^\circ$	Ângulo final da haste 2 vale $0^\circ$
Converte em volts o ângulo final da haste 2	Converte em volts o ângulo final da haste 2
Lê posição angular atual da haste 2	Lê posição angular atual da haste 2
Envia tensão para o motor 2	Envia tensão para o motor 2
Até que a posição angular da haste 2 ultrapasse $180^\circ$	Até que a posição angular da haste 2 ultrapasse $0^\circ$
Pára de enviar tensão para o motor 2	

### 9.4 Procedimento que move a haste 1 para a posição final desejada

Após inicializar a haste 2, a haste 1 poderá atingir a posição angular final livremente, sem que a haste 2 atrapalhe o movimento. O controle do posicionamento da haste 1 é realizado neste procedimento através do cálculo da diferença entre o

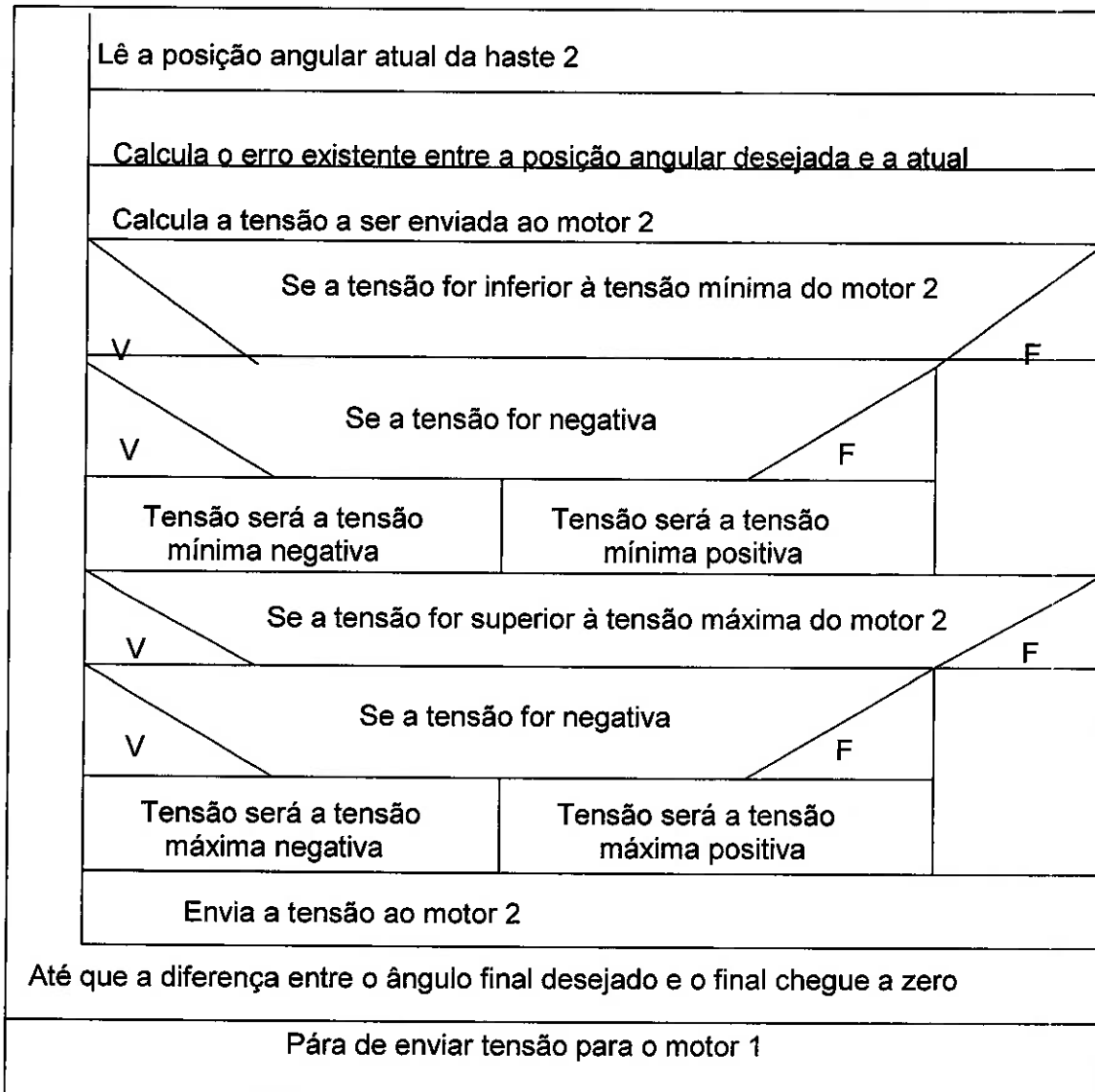
ângulo final desejado e o ângulo atual, enviando-se um sinal apropriado para o motor

1. Devido às limitações de tensão mínima e tensão máxima do motor, a tensão enviada ao mesmo deve situar-se neste intervalo.



## 9.5 Procedimento que move a haste 2 para a posição final desejada

A estrutura do procedimento que posiciona a haste 2 é análoga à da haste 1:



## 10. Análise dos Resultados

Nesta etapa vamos analisar os resultados obtidos com a integração de todo o sistema.

Ao se rodar o programa para movimentar o mecanismo, quatro arquivos de saída foram gerados. Em dois deles constam os valores da diferença entre as posições final e atual ao longo do percurso, um para cada motor. Os outros dois, mostram as leituras dos potenciômetros.

Os dados são referentes ao deslocamento da posição (26, 0) até (-26, 0).

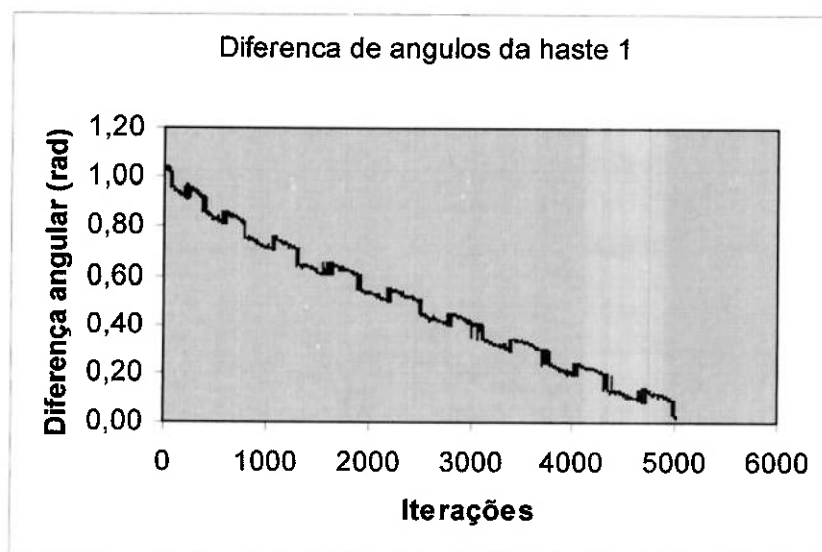


Figura 17: Diferença entre a posição angular final e atual ao longo do tempo – haste 1

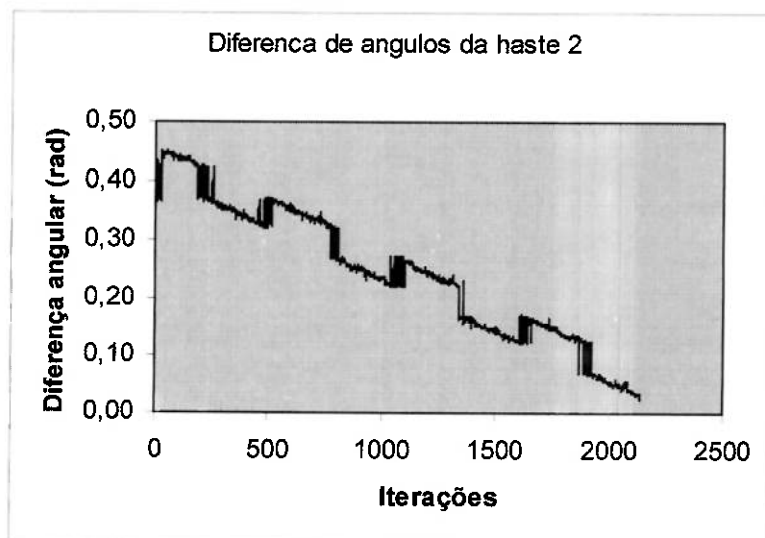
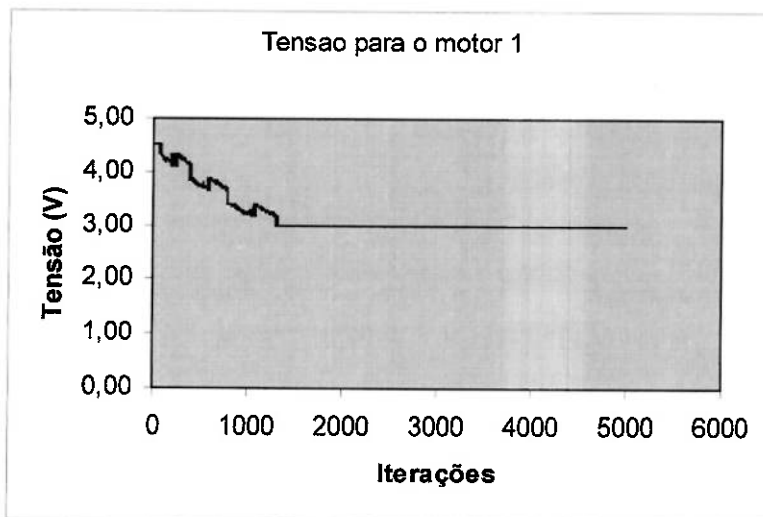
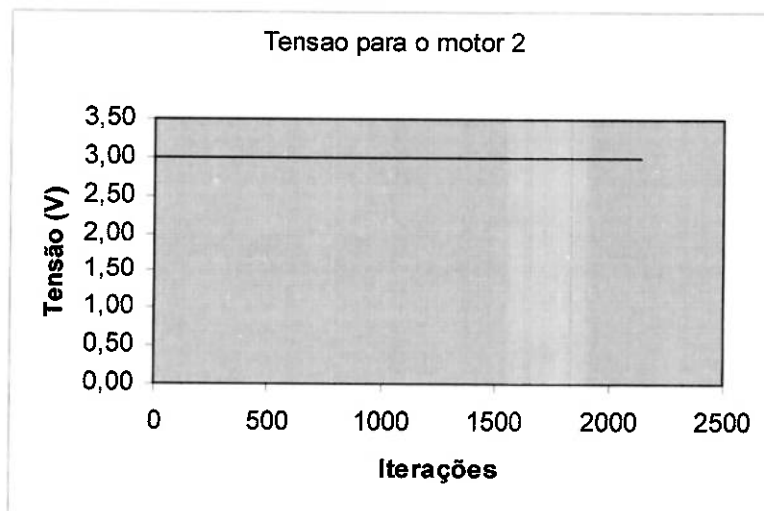


Figura 18: Diferença entre a posição angular final e atual ao longo do tempo - haste 2

Analisando-se os gráficos referentes aos valores da diferença de ângulos, percebe-se que essa diferença decresce, aproximadamente, linearmente, porém apresentam ruído. Nota-se também que a haste 1 realizou um número maior de iterações do que a haste 2 e a diferença não chega a zero, por causa da precisão setada no programa.

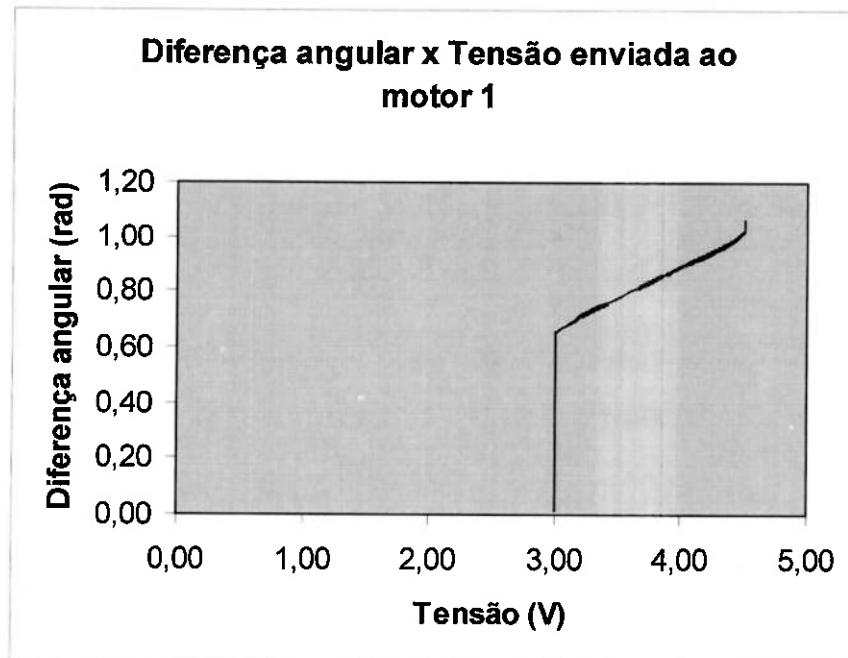


**Figura 19: Tensão enviada ao motor 1 durante o movimento**



**Figura 20: Tensão enviada ao motor 2 durante o movimento**

Observando-se o gráfico da figura 24, percebe-se que a tensão decresce até um certo ponto onde ele atinge o valor mínimo para o acionamento do motor e se mantém constante até o fim do movimento. A figura 25 mostra que o motor 2 é acionado, neste movimento, com a tensão mínima.



**Figura 21: Curva diferença angular x tensão enviada ao motor 1**

O gráfico da figura 26 mostra que a tensão enviada ao motor decresce com o decréscimo da diferença angular entre a posição final e a atual e se mantém constante até o término do movimento.

Observando-se o desempenho do mecanismo na prática, verificou-se que ele se movimentou de forma satisfatória, porém, de vez em quando, apresentava um erro maior de posicionamento da haste2, em relação à haste 1, principalmente em posições extremas, como a posição de coordenadas (26, 0) e (-26, 0). Isto se deve a vários fatores, principalmente de ordem física.

O principal problema detectado foi a folga da correia dentada, causada pelo pequeno desalinhamento entre o eixo do motor 2 e do motor 1. Na posição (26, 0), a correia apresentava uma folga bastante grande, pois a distância entre o eixo do motor



e o eixo da polia solidária à haste 2 era menor. Já na posição oposta (-26, 0), a correia apresentava-se bastante tracionada, pois a distância entre os eixos era maior.

Outro problema constatado, foi uma pequena imprecisão na leitura dos potenciômetros devido à fácil descalibração dos mesmos. Essa descalibração ocorre principalmente na posição de referência (coeficiente linear da curva ângulo x tensão), que é de extrema importância para os cálculos dos ângulos. Já em relação ao coeficiente angular, não houveram variações significativas. A descalibração ocorre, principalmente, devido à eventuais choques entre as hastes e as bases dos motores, na fase de testes; ao rendimento da transmissão por polias, para o potenciômetro 2; e ao rendimento da transmissão por engrenagens, para o potenciômetro 1.

Além de problemas físicos, foram detectados problemas de software, principalmente, em relação à satisfação das condições de igualdade nos loops do tipo repeat ... until. Isto depende da precisão adotada no programa.

Foi verificado que, para uma precisão menor que 0.02 rad, o programa não detectava a posição final desejada e entrava em loop infinito. Portanto, a precisão adotada foi de 0.02 rad para a leitura do potenciômetro 1 e 0.025 rad para o potenciômetro 2.

Outra possível causa de erros é o sinal de tensão na saída do circuito amplificador, que apresenta ruídos, os quais puderam ser observados com um osciloscópio. Este ruído pode se propagar ao longo de todo o processo de leitura, cálculo e envio de sinais de tensão, afetando o desempenho do sistema.

Ao final do item 10 deve-se acrescentar a seguinte análise:

No item 8, onde foi realizada a simulação teórica do sistema, chegou-se à conclusão que o controlador que apresentava o melhor desempenho era um PID, porém na prática verificou-se que um controlador proporcional já apresentava um desempenho satisfatório e, por ser de implementação mais simples, foi então escolhido como o controlador usado no protótipo.

Na simulação pode-se notar que o controlador proporcional não diminui o sobressinal da resposta do sistema, porém na prática o protótipo não apresentou sobressinal. Tal fato pode ter ocorrido devido ao atrito real ser maior do que o utilizado na simulação.

## **11. Conclusões**

Este projeto cumpriu o seu papel como um trabalho de formatura, visto que pudemos integrar os conhecimentos adquiridos, ao longo do curso, nas diversas áreas englobadas pela Engenharia Mecatrônica.

O projeto se mostrou bastante satisfatório em relação aos objetivos propostos no início deste trabalho e à simplicidade dos componentes utilizados.

Na construção do protótipo, pudemos vivenciar todas as etapas, desde o layout e desenhos de fabricação, até a fabricação e montagem das peças desenvolvidas, além do solucionamento de problemas de exequibilidade física e propostas de novas soluções. Em relação a este tópico, verificamos a dificuldade em garantir a precisão e as tolerâncias, tanto geométricas quanto dimensionais e de acabamento, como por exemplo, alinhamento entre centros de furação, perpendicularidade entre peças, acabamento em áreas onde se necessita boa planicidade, devido ao fato das máquinas utilizadas na fabricação das peças não serem muito precisas e à falta de recursos disponíveis.

A utilização de potenciômetros como sensores de posição não é a mais recomendada para este tipo de aplicação, porém o seu desempenho mostrou-se satisfatório.

O circuito de potência serviu para aplicarmos alguns conceitos de eletricidade e eletrônica e cumpriu o seu objetivo de amplificar a corrente, apesar do sinal de saída apresentar um certo ruído.

A implementação do controle também mostrou-se importante na aplicação dos conceitos da teoria de controle e também na implementação do software por uma linguagem de programação.

Por fim, gostaríamos de salientar as novas possibilidades de desenvolvimento que este projeto apresenta, como por exemplo, o controle de trajetória, a adição de mais um grau de liberdade, outros meios de transmissão de torque e outros tipos de

aplicação do mecanismo, bem como a utilização de sensores de posição mais precisos, motores mais adequados para esta aplicação e circuitos de amplificação mais confiáveis.

## **12. Bibliografia**

- [1] Ogata, K. - "Engenharia de Controle Moderno", Prentice/Hall do Brasil, segunda edição, Rio de Janeiro, 1993.
- [2] Simões, M. G. - "Aplicações de Motores Elétricos em Engenharia Mecânica", Apostila de PMC 501 – EPUSP, janeiro 1998.
- [3] Coelho, T. - "Síntese de Mecanismos", Notas de Aula – EPUSP, 1997.
- [4] Rodrigues, A. L. T. e Barbosa F<sup>o</sup>, H. J. H. - "Sistema de Controle de Posicionamento de Telescópios" – EPUSP, 1996.
- [5] "Control Tutorials for MATLAB" - Carnegie Mellon - University of Michigan - <http://rcisgi.eng.ohio-state.edu>.
- [6] Departamento de Documentação e Engenharia de Hardware – "Conversor Analógico Digital CAD 12/36 – Manual do Usuário e de Referência"- versão 1.2 - Nov/1993.
- [7] Asada, H. e Slotine, J. - "Robot Analysis and Control"- MIT - Wiley Interscience Publication - 1986.
- [8] Bento, C.R. - "Sistema de Controle - Teoria e Projetos" - Mc. Graw Hill, Rio de Janeiro, 1975.
- [9] Ogata, K. - "Discrete-Time Control Systems", Prentice/Hall , segunda edição, New Jersey, 1995.



```
{*****}
**}
{
}
{
    INICIALIZACAO DA PLACA
}
{
}
{*****}
**}

procedure inicializa(base:word);
var
    i,
    lixo:byte;          { variavel auxiliar para esvaziar a FIFO
}
begin

{***** Programacao do modo 0 da placa
*****}

    port[base+4]:=03;    { direciona ponteiro para escrever na palavra
modo }
    port[base+5]:=00;    { programa modo zero
}

{***** Programacao da memoria de canais
*****}
{    programa os 8 primeiros canais de entrada analogica para
}
{***** ganho 1 e modo unipolar
*****}

    for i:=0 to 7 do
        begin
            port[base+4]:=01;    { direciona ponteiro para escrever no
}
                                { ponteiro da memoria de canais
}
            port[base+5]:=i;    { coloca o ponteiro para o i-esimo
}
                                { endereco da memoria de canais
}
            port[base+4]:=04;    { direciona ponteiro para escrever na
}
                                { memoria de canais
}
            port[base+5]:=112+i { programa posicao da memoria para o
}
                                { i-esimo canal com ganho 1 e modo
unipolar}
        end;

{***** Auto-calibracao do conversor A/D
*****}
```

```
port[base+4]:=06;      { direciona ponteiro para posicao de
}
                        { auto-calibracao
}
port[base+5]:=255;    { escreve qualquer dado na posicao de
}
                        { auto-calibracao
}
Delay(2000);

{***** Esvazia a FIFO da placa A/D
*****}

writeln(port[base+3]);

for i:=0 to 15 do      { limpa as 16 posicoes da FIFO
}
begin
  lixo:=port[base+4]; { le byte A da posicao atual da FIFO
}
  lixo:=port[base+5] { le byte B da posicao atual da FIFO
}
end;

writeln(port[base+3]);

{***** Programacao do modo da placa desejado
*****}
{      programa modo DMA desligado, sem timers ou interrupcoes
}
{***** modo pooling, BURST desligado
*****}

port[base+4]:=03;      { direciona ponteiro para escrever na
}
                        { palavra de modo
}
port[base+5]:=00;      { programa BURST desligado, sem DMA e
}
                        { sem timers
}
end;

{*****
}
{
}
{      LE ENTRADAS ANALOGICAS DOS POTENCIOMETROS
}
{
}
{*****
}

procedure le_entradas(base:word;var E:array of real);
```



```
var
  status:byte;
  i:integer;
  aux:word;
  r:real;

begin
  for i:=0 to 1 do
    begin
      port[base+4]:=02;    { direciona ponteiro para o endereco de
    }                      { conversao
    }
      port[base+5]:=i;    { canal de conversao i
    }

    {***** Espera FIFO ficar com pelo menos 1 dado
    *****}

    repeat
      status:=port[base+3];
    until
      (status and 1)>0;

    {***** Converte valor da entrada para volts
    *****}

    aux:=256*port[base+5]+port[base+4];
    r:=(10-(10/65536)*(65536-aux));
    E[i]:=r;
  end;
end;

{*****}
{
{
{          CALCULO DOS ANGULOS FINAIS
{
{
{*****}
}

procedure calcula_angulos(var teta1, teta2a, beta2, zeta, gama: real;
x2,y2: real);
var
  z3:real;                { modulo da posicao final a articulacao 1
}

  alfa,                    { angulo entre a haste 1 e z3 em rad
}
  senalfa, cosalfa,
  delta,                    { angulo entre o eixo x e z3 em rad
}
  betal,                    { angulo inicial da haste 2 em rad
}
```

```
teta2b,          { angulo final da haste 1 em rad
}
timao1,          { angulo final da haste 1 em graus
}
timao2,          { angulo final da haste 2 em graus
}
cosgama,         { cosseno do angulo da haste 2 em relacao
}
                { ao eixo x
}

A2x, A2y: real;   { posicao final da articulacao 2
}

i,j:integer;

begin

{***** Calculo do modulo da posicao final a articulacao 1
*****}

    z3:=sqrt(x2*x2+y2*y2);

{***** Calculo do angulo entre a haste 1 e z3 (rad)
*****}

    cosalfa:=(haste1*haste1+z3*z3-haste2*haste2)/(2*haste1*z3);
    senalfa:=sqrt(1-cosalfa*cosalfa);
    alfa:=arctan(senalfa/cosalfa);

{***** Calculo do angulo entre o eixo x e z3 (rad)
*****}

    if x2=0 then
        delta:=pi/2
    else
        begin
            delta:=arctan(y2/x2);
            if x2<0 then
                delta:=delta+pi;
            end;
        end;

{***** Tensao dos potenciometros (volt)
*****}

    for i:=0 to Passos-1 do
        begin
            le_entradas(896,E);
            for j:=0 to Last_canal do
                soma[j] := soma[j] + E[j];
            end;
            for j:=0 to Last_canal do
                soma[j]:=soma[j]/Passos;
            writeln('Leitura dos potenciometros',soma[0],soma[1]);
        end;

{***** Calculo do angulo inicial da haste 1 (rad)
*****}

    teta1:=soma[0]*inclinacao1 + coef_linear1;
```

```
{*** Calculo dos possiveis deslocamentos angulares da haste 1 (rad)
**}

teta2a:=delta-tetal-alfa;
teta2b:=delta-tetal+alfa;
if teta2b<teta2a then          { seleciona-se o menor deslocamento
}
    teta2a:=teta2b;           { angular
}

{***** Calculo da posicao final da articulacao 2 (rad)
*****}

if abs(tetal+teta2a)<=eps then
begin
    A2x:=haste1;
    A2y:=0;
end
else
if abs(tetal+teta2a-pi/2)<=eps then
begin
    A2x:=0;
    A2y:=haste1;
end
else
begin

A2x:=haste1*sqrt(1/(1+sqr((sin(tetal+teta2a)/cos(tetal+teta2a)))));
    A2y:=A2x*(sin(tetal+teta2a)/cos(tetal+teta2a));
end;

{***** Calculo do angulo entre a haste 2 e o eixo x (rad)
*****}

if x2=A2x then
    gama:=pi/2
else
begin
    gama:=arctan((y2-A2y)/(x2-A2x));
    cosgama:=(x2-A2x)/haste2;
    if cosgama<0 then
        gama:=pi+gama;
    end;

{***** Calculo do angulo inicial da haste 2 (rad)
*****}

    betal:=soma[1]*inclinacao2 + coef_linear2;

{***** Calculo do deslocamento angular da haste 2 (rad)
*****}

    beta2:=gama-betal;

writeln('');
writeln('');
```



```
end
else          { haste 1 move em sentido horario
}
begin
  beta:=0;
  volt:=-volt;
  pot:=(beta-coef_linear2)/inclinacao2;
  repeat
    le_entradas(896,inicial);
    motor:=round(4096*(1-(5-volt)/10));
    port[base+10]:=Lo(motor shl 4); { motor 2 move-se com uma
tensao}
    port[base+11]:=Hi(motor shl 4); { de 'volt' ate que a haste 2
}
    until inicial[1]<=pot;          { chegue em beta
}
  end;

  port[base+10]:=Lo(2048 shl 4);    { haste 2 para de se mover
}
  port[base+11]:=Hi(2048 shl 4);
  writeln('Haste 2 ja inicializada');
end;

{*****
}
{
}
{          POSICIONAMENTO DA HASTE 1
}
{
}
{*****
}

procedure move_haste1(teta2a, zeta:real);
var
  ang_atual1,          { angulo atual da haste 1 em rad
}
  errol1,              { diferenca entre o angulo final desejado
}
                      { e o angulo atual da haste 1 em rad
}
  volt1:real;          { tensao recebida pelo motor 1 em volts
}
  motor1:word;         { tensao recebida pelo motor 1 em decimal
}

begin

  assign(dif1,'d:\pikk\dif1.out');
  rewrite(dif1);
  writeln(dif1,'Diferenca de angulos da haste 1');
```

```
writeln(dif1, '');

assign(saida1, 'd:\pikk\saida1.out');
rewrite(saida1);
writeln(saida1, 'Tensao para o motor 1');
writeln(saida1, '');

repeat
  le_entradas(896, E);    { leitura da posicao angular da haste 1
}                          { pelo potenciometro 1 em volts
}

  ang_atual1:=inclinacao1*E[0]+coef_linear1;

  erro1:=(zeta-ang_atual1)/pi;

  writeln(dif1, erro1);

  volt1:=vmax*erro1;      { tensao enviada ao motor 1 conforme a
}                          { diferenca entre o angulo desejado e o
}                          { atual
}

  if abs(volt1)<=eps then
    volt1:=0
  else
    if abs(volt1)<vmin then { tensao minima necessaria para o      }
                          { acionamento do motor 1              }
      if volt1<0 then
        volt1:=-vmin
      else
        volt1:=vmin;

    if abs(volt1)>=vmax then { tensao maxima que o motor 1        }
                          { suporta                                }
      if volt1<0 then
        volt1:=-vmax
      else
        volt1:=vmax;

  writeln(saida1, volt1);

  motor1:=round(4096*(1-(5-volt1)/10));
  port[base+8]:=Lo(motor1 shl 4);    { motor 1 recebe a tensao
}
  port[base+9]:=Hi(motor1 shl 4);    { calculada ate que a haste 1
}
until abs(erro1)<=eps;               { chegue na posicao final
}

close(dif1);
close(saida1);

if teta2a<0 then                    { haste 1 move-se em sentido anti-horario }
begin
```

```
volt1:=-vmin;
motor1:=round(4096*(1-(5-volt1)/10));
port[base+8]:=Lo(motor1 shl 4); { motor 1 move-se por um tempo
}
port[base+9]:=Hi(motor1 shl 4); { para compensar o sentido de
}
delay (200); { rotacao do eixo
}
end;
port[base+8]:=Lo(2048 shl 4); { motor 1 para de se mover
}
port[base+9]:=Hi(2048 shl 4);
writeln('Haste 1 posicionada!');
end;
```

```
{*****}
{
}
{
{
POSICIONAMENTO DA HASTE 2
}
{
}
{*****}
}
```

```
procedure move_haste2(gama, tetat, teta2a:real);
var
  ang_atual2, { angulo atual da haste 2 em rad
}
  erro2, { diferenca entre o angulo final desejado
} { e o angulo atual da haste 2 em rad
}
  volt2:real; { tensao recebida pelo motor 2 em volts
}
  motor2:word; { tensao recebida pelo motor 2 em decimal
}
```

```
begin
  assign(dif2,'d:\pikk\dif2.out');
  rewrite(dif2);
  writeln(dif2,'Diferenca de angulos da haste 2');
  writeln(dif2,'');
  assign(saida2,'d:\pikk\saida2.out');
  rewrite(saida2);
  writeln(saida2,'Tensao para o motor 2');
  writeln(saida2,'');
  repeat
    le_entradas(896,E); { leitura da posicao angular da haste 2
  }
```

```
        { pelo potenciometro 2 em volts
    }

    ang_atual2:=inclinacao2*E[1]+coef_linear2;

    erro2:=(gama+(pi/2)-(tetal+teta2a)-ang_atual2)/pi;

    writeln(dif2,erro2);

    volt2:=vmax*erro2;    { tensao enviada ao motor 2 conforme a
    }                    { diferenca entre o angulo desejado e o
    }                    { atual
    }

    if abs(volt2)<=eps1 then
        volt2:=0
    else
        if abs(volt2)<vmin then { tensao minima necessaria para o      }
                                { acionamento do motor 2              }
            if volt2<0 then
                volt2:=-vmin
            else
                volt2:=vmin;
        if abs(volt2)>=vmax then { tensao maxima que o motor 2      }
                                { suporta                             }
            if volt2<0 then
                volt2:=-vmax
            else
                volt2:=vmax;

        writeln(saida2,volt2);

        motor2:=round(4096*(1-(5-volt2)/10));
        port[base+10]:=Lo(motor2 shl 4); { motor 2 recebe a tensao
    }
        port[base+11]:=Hi(motor2 shl 4); { calculada ate que a haste 2
    }
    until abs(erro2)<=eps1;              { chegue na posicao final
    }

    close(dif2);
    close(saida2);

    port[base+10]:=Lo(2048 shl 4);      { motor 2 para de se mover
    }
    port[base+11]:=Hi(2048 shl 4);
    writeln('Haste 2 posicionada!!');
end;

{ ****
}
{ ****
}
{
    PROGRAMA PRINCIPAL
}
```



```
{ *****
}
{ *****
}

var
  i,j,
  posicao_valida:integer; { parametro para verificar se a posicao
}
                        { final esta dentro da area de atuacao do
}
                        { mecanismo
}

  teta1,                { angulo inicial da haste 1 em rad
}
  teta2a,               { deslocamento angular da haste 1 em rad
}
  beta2,               { deslocamento angular da haste 2 em rad
}
  zeta,                { angulo final da haste 1 em rad
}
  gama,               { angulo final da haste 2 em rad
}

  x2_ant, y2_ant,      { posicao anterior (atual)
}
  x2,y2:real;         { posicao final desejada
}

  nova_posicao:char;    { parametro para verificar se o mecanismo
}
                        { ira para uma nova posicao
}

begin

  clrscr;

  writeln('*****
***');
  writeln('          MECANISMO POSICIONADOR HORIZONTAL "PIKK.COM"
');

  writeln('*****
***');
  delay(3000);
  inicializa(896);
  OFFSET;

  x2:=26;                { posicao inicial do mecanismo
}
  y2:=0;

  repeat
  nova_posicao:='n';
  writeln('Entre com a posicao final desejada:');

  repeat
    x2_ant:=x2;
```

```
y2_ant:=y2;
posicao_valida:=0;
write('x:');
readln(x2);
write('y:');
readln(y2);

{***** Verifica se a posicao final desejada esta
*****}
{***** na area de atuacao do mecanismo
*****}

  if (x2*x2+y2*y2 > (haste1+haste2)*(haste1+haste2)) or
    (y2<0) or ((x2*x2+y2*y2)< haste1*haste1 ) then
    writeln('Posicao Invalida')
  else
    if (x2=x2_ant) and (y2=y2_ant) then
      writeln('Posicao atual')
    else
      posicao_valida:=1;
until posicao_valida=1;

calcula_angulos(tetal,teta2a,beta2,zeta,gama,x2,y2);

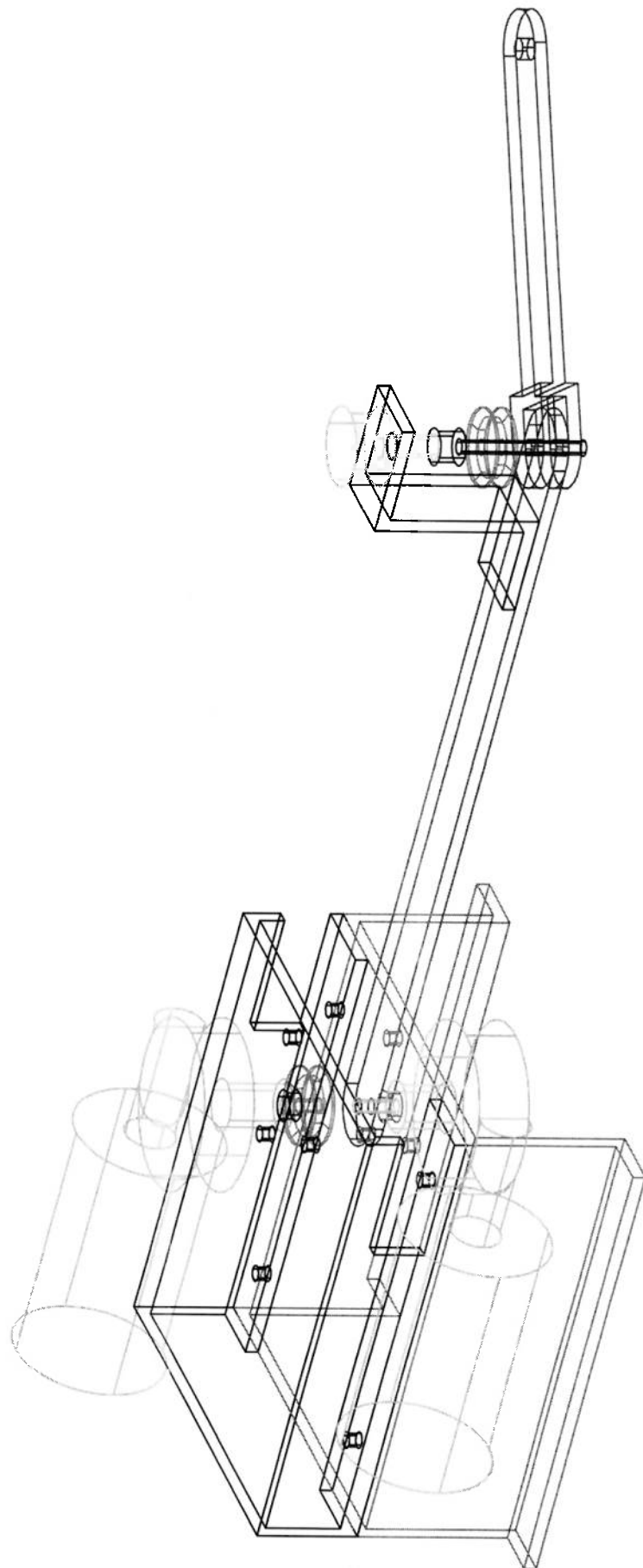
inicializa_haste2(teta2a);

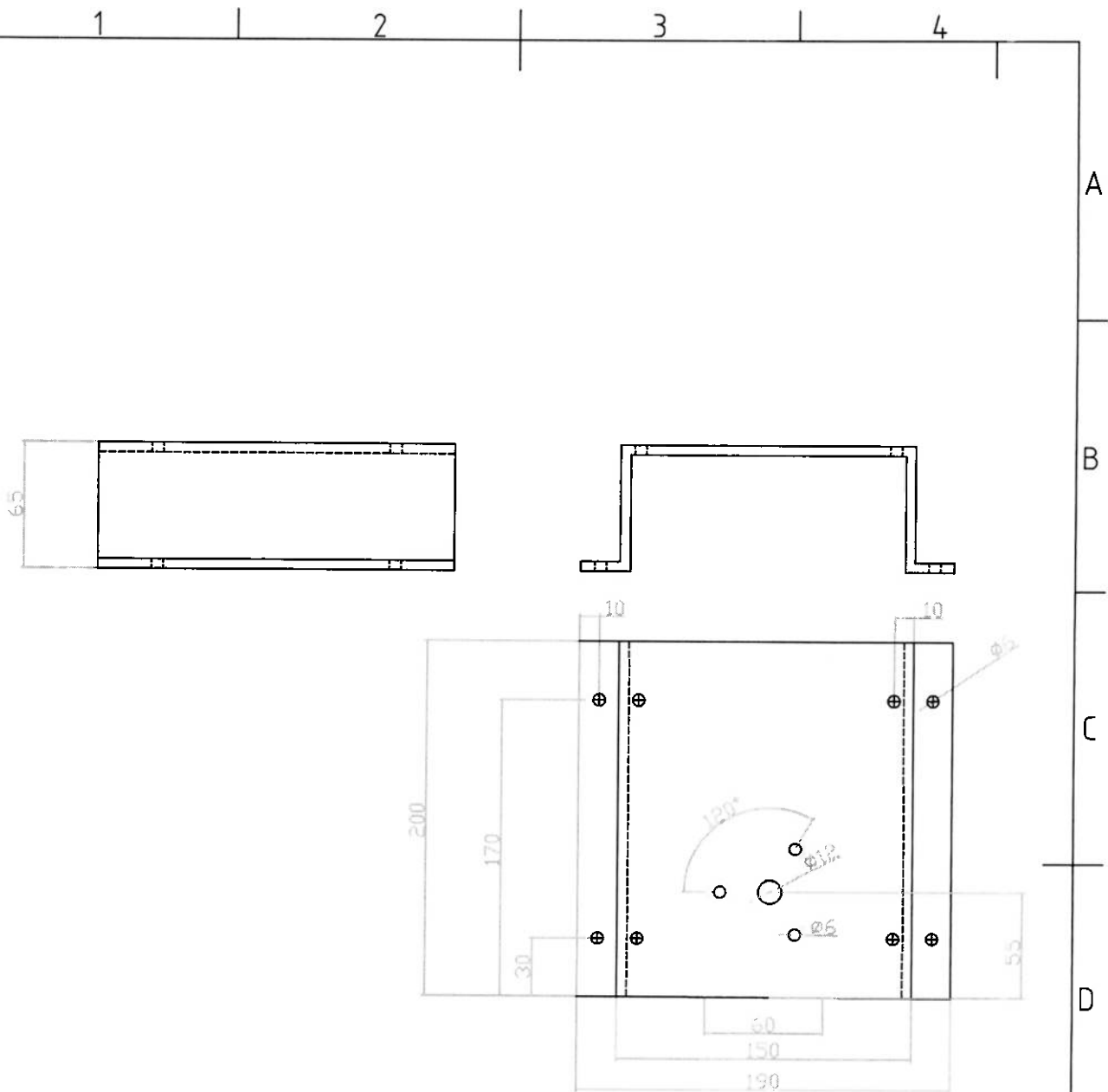
move_haste1(teta2a,zeta);

move_haste2(gama, tetal, teta2a);

writeln('Deseja mover para uma nova posicao? (s/n):');
readln(nova_posicao);
until nova_posicao='n';
end.
```

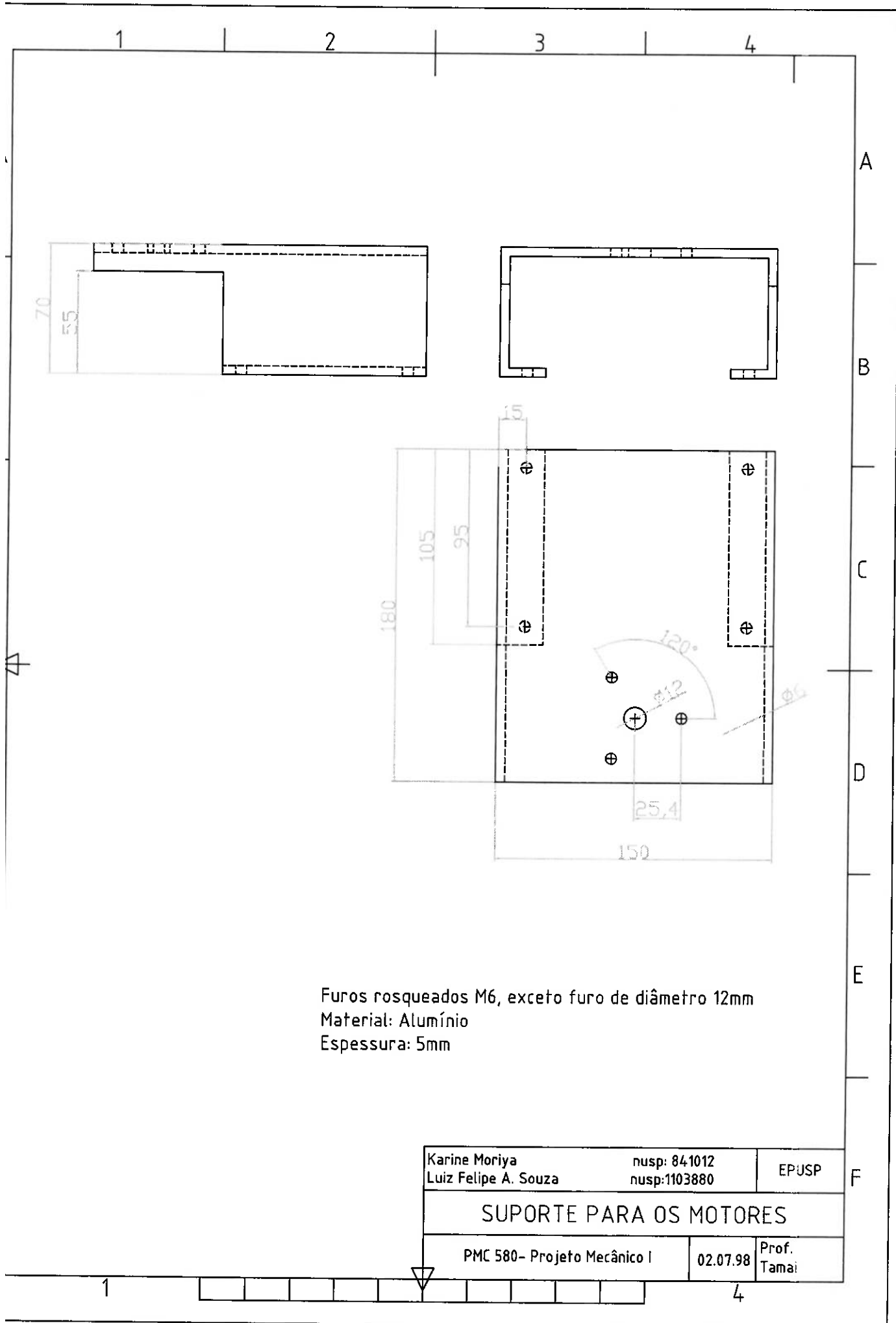
## **13.2 Desenhos de Fabricação**





Furos rosqueados M6  
Material: Alumínio  
Espessura: 1/8"

Karine Moriya Luiz Felipe A. Souza	nusp: 841012 nusp:1103880	EPUSP
BASE		
PMC 581- Projeto Mecânico II	02.07.98	Prof. Tamai



1

2

3

4

Ø10

M6

A

B

C

D

E

F

42

10

13

20

9

M5

Ø6

Material: Aço 1020

Karine Moriya  
Luiz Felipe A. Souzanusp: 841012  
nusp:1103880

EPUSP

## EIXO DA POLIA MOTORA

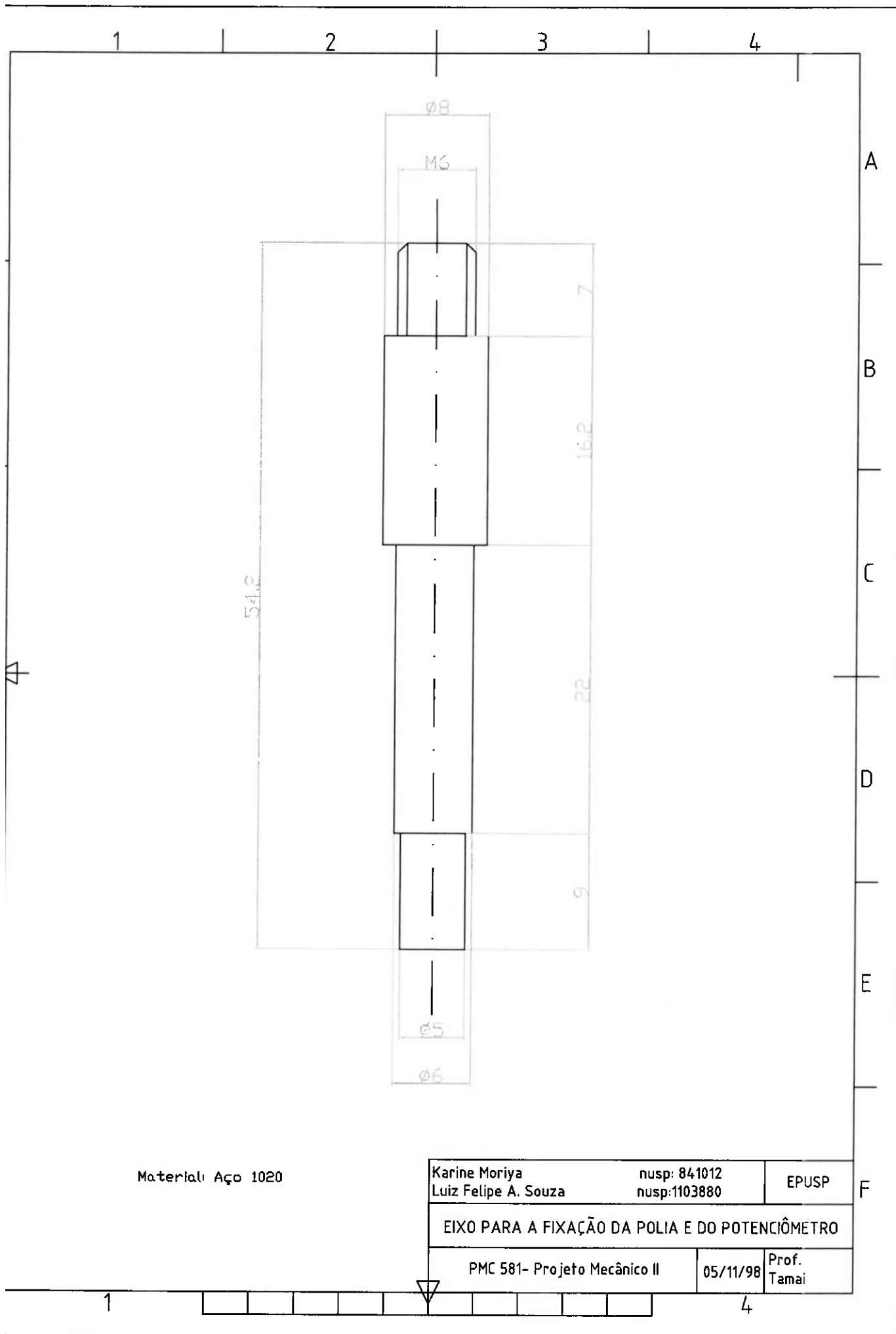
PMC 581- Projeto Mecânico II

05/11/98

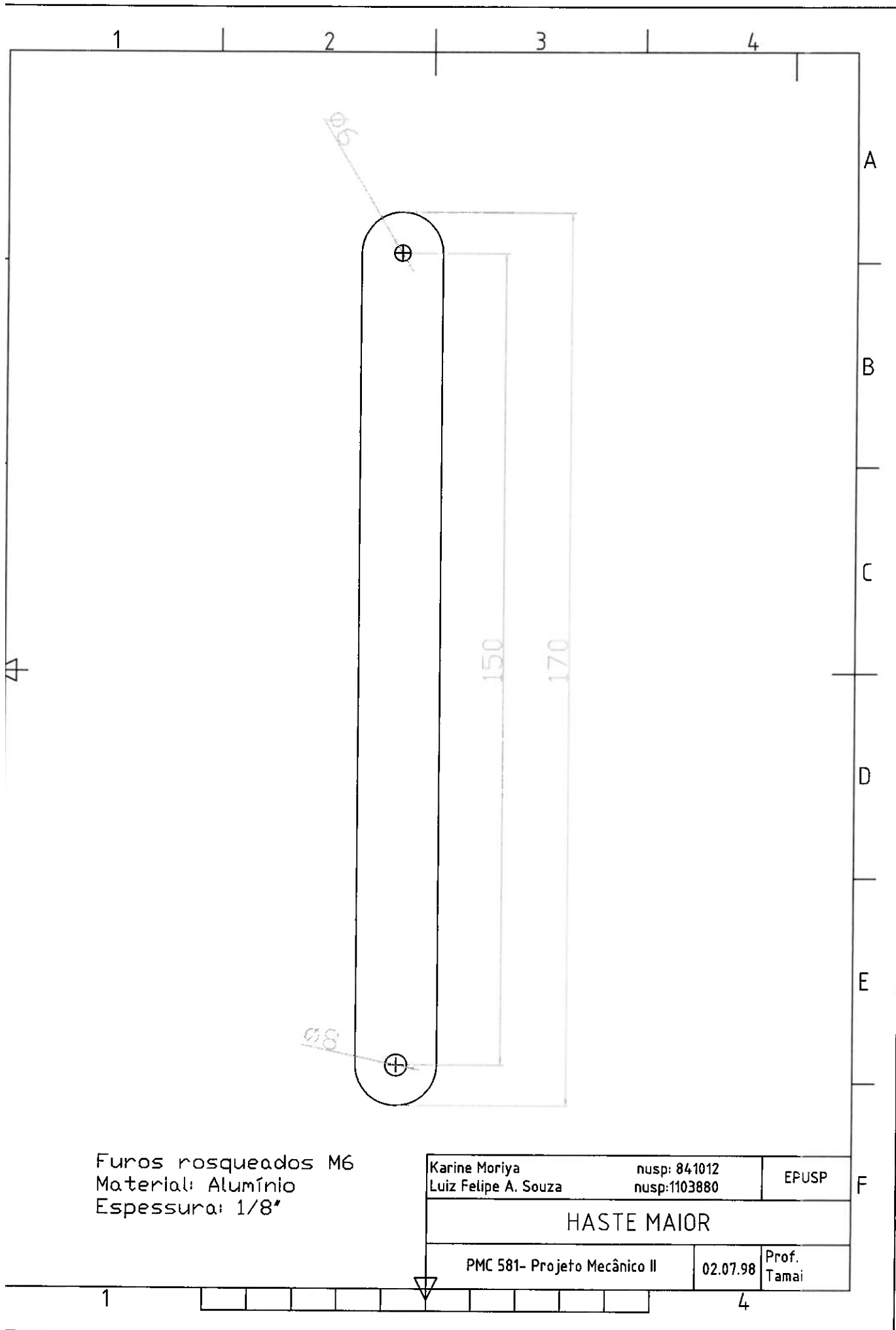
Prof.  
Tamai

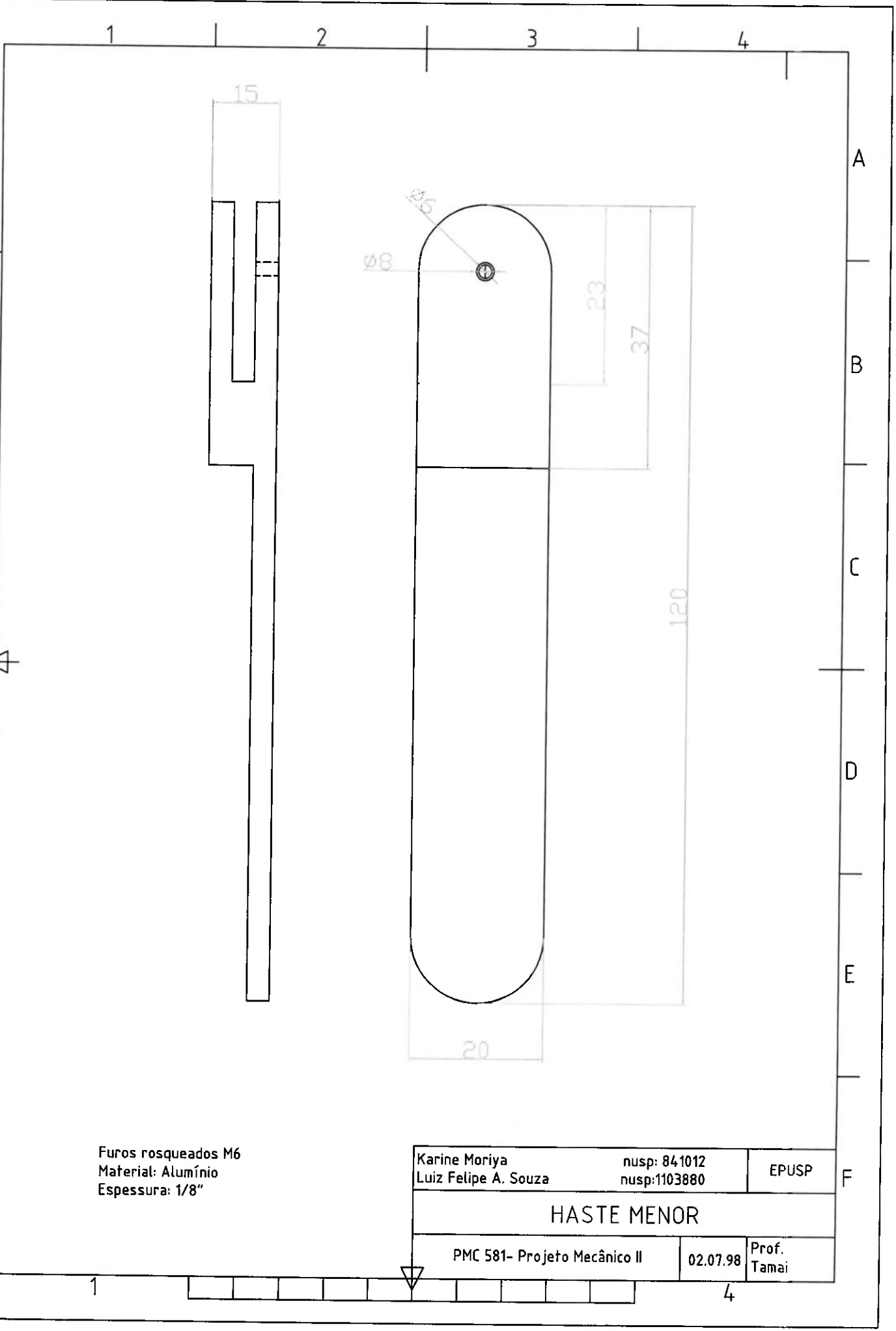
1

4









Furos rosqueados M6  
Material: Alumínio  
Espessura: 1/8"

Karine Moriya Luiz Felipe A. Souza	nusp: 841012 nusp:1103880	EPUSP
HASTE MENOR		
PMC 581- Projeto Mecânico II	02.07.98	Prof. Tamai

1 2 3 4

A

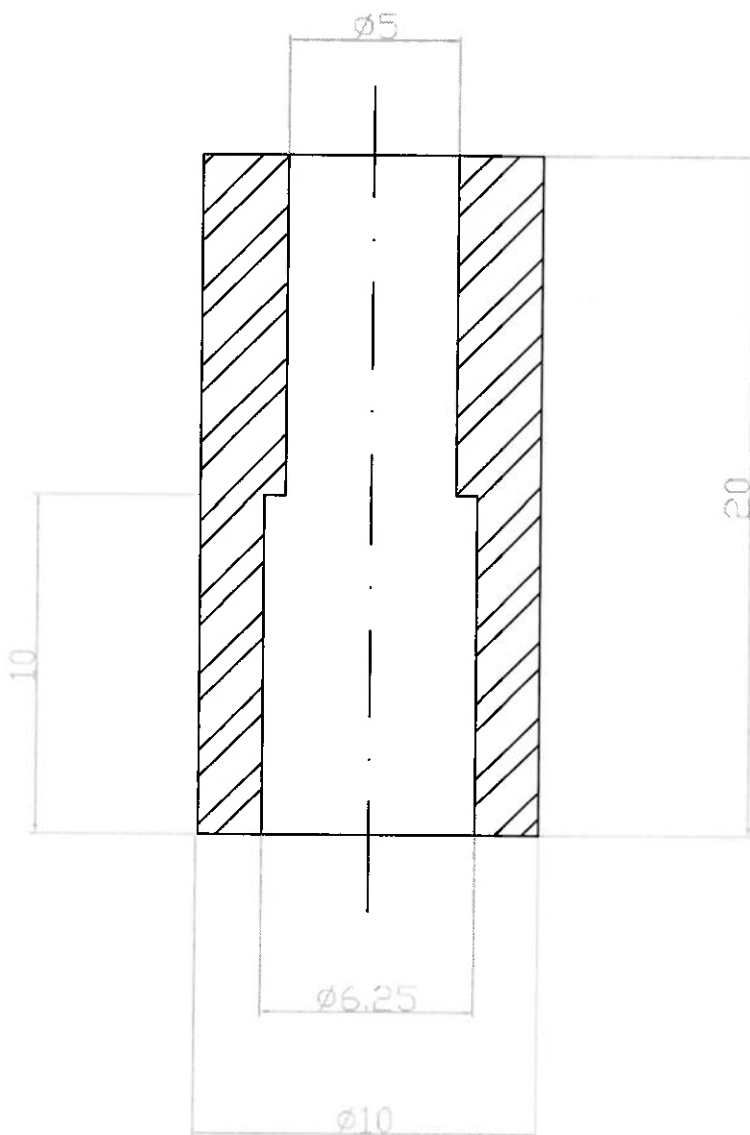
B

C

D

E

F



Material: Plástico

Karine Moriya  
Luiz Felipe A. Souza

nusp: 841012  
nusp:1103880

EPUSP

LUVA PARA FIXAÇÃO DO POTENCIÔMETRO NO EIXO

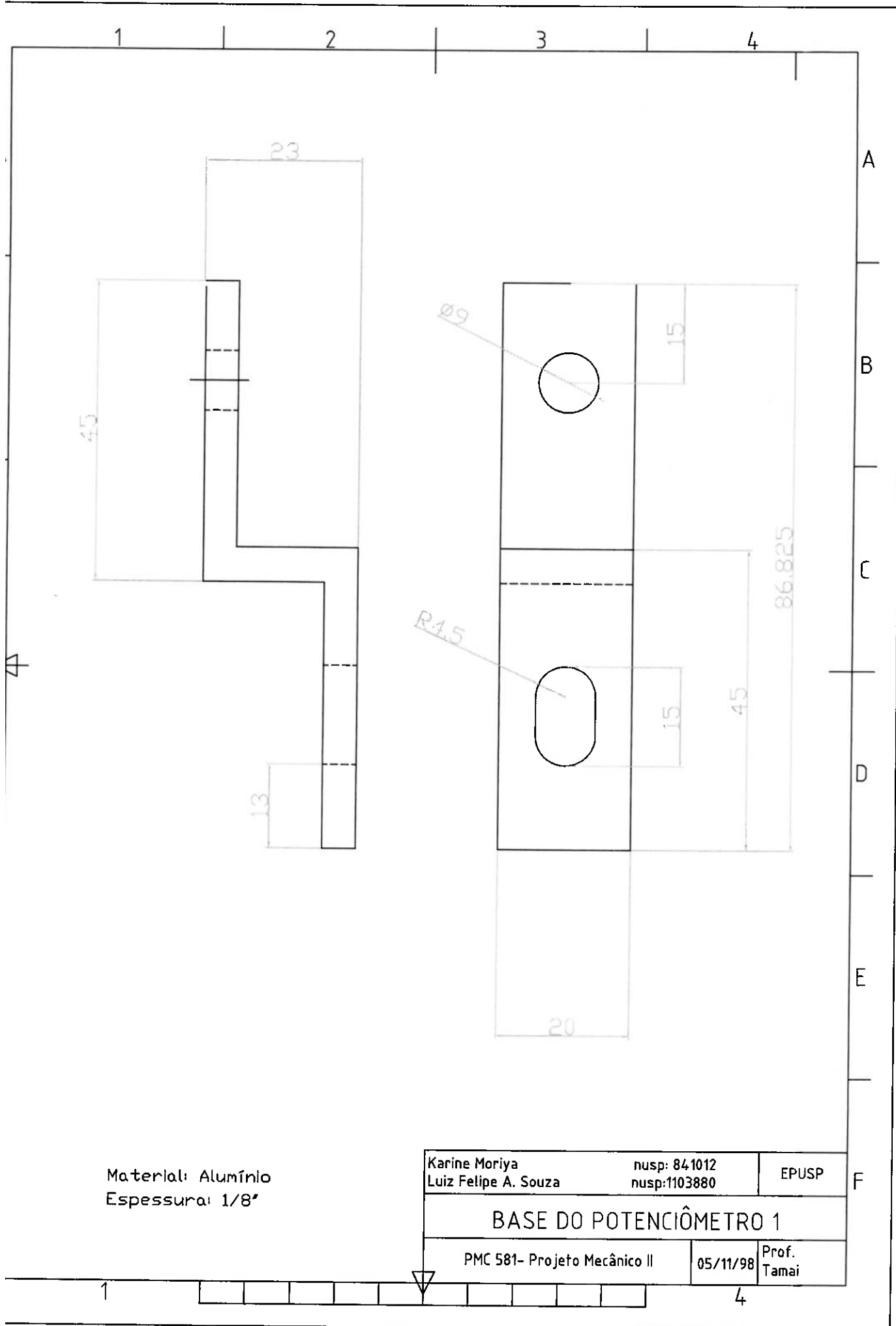
PMC 581- Projeto Mecânico II

05/11/98

Prof.  
Tama

1

4



Material: Alumínio  
Espessura: 1/8"

Karine Moriya Luiz Felipe A. Souza	nusp: 841012 nusp:1103880	EPUSP
BASE DO POTENCIÔMETRO 1		
PMC 581- Projeto Mecânico II	05/11/98	Prof. Tamai

1

2

3

4

A

B

C

D

E

F

72.825

38

17

17

 $\varnothing 9$  $\varnothing 8$ 

67

38

Material: Alumínio  
Espessura: 1/8"

Karine Moriya  
Luiz Felipe A. Souza

nusp: 841012  
nusp:1103880

EPUSP

BASE DO POTENCIÔMETRO 2

PMC 581- Projeto Mecânico II

05/11/98

Prof.  
Tamai

1

4