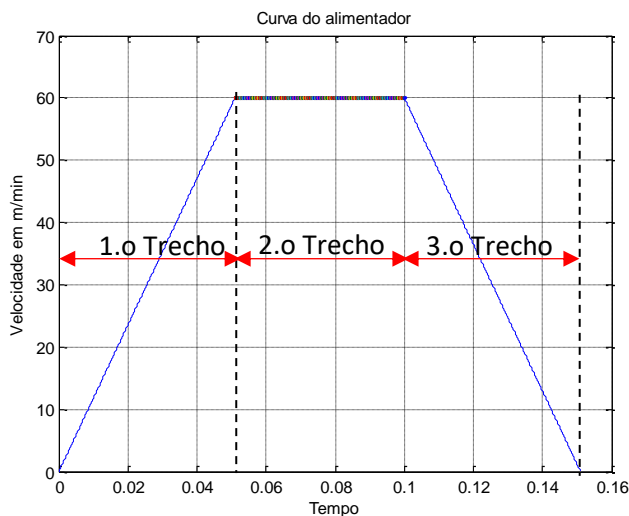


UM SIMPLIFICADO ALGORÍTMO PRÁTICO DE POSIÇÃO PARA UM EIXO.

Resumo: O processo para a realização do posicionamento através de um servomotor CA passa pelo sistema de acionamento e controle. Processos como alimentadores de prensas constituído por um alimentador por rolos e supervisionados por um sensor digital, normalmente um "encoder" tem desempenho diretamente relacionado com o algoritmo utilizado. Um sistema de controle cuja estrutura com duas malhas de controle conectadas sendo de velocidade e posição comumente escolhida para esse processo e implementadas ou com circuitos da área analógica ou da área digital. O algoritmo proposto é particionado em trechos ou segmentos de retas, para a aceleração inicial, velocidade constante e desaceleração final. Certos casos o processo é realizado em dois trechos sendo aceleração e desaceleração. O final do processo é inserido um controle da posição normalmente um PI para a posição final. O algoritmo é alimentado com parâmetros como avanço em mm, velocidade da alimentação em m/min, tempo de aceleração e desaceleração em segundos, rotação do motor em rpm, diâmetro do rolo alimentador em mm, comprimento da roda medidora em mm, tempo de subida e descida do servoacionamento e número de pulsos do encoder. Um limite do alimentador é avanço mínimo de 6mm.

Introdução: A figura a seguir mostra dois exemplos de alimentação sendo o primeiro com três trechos para a realização da alimentação e o segundo com dois trechos necessários para a posição. Observando o primeiro exemplo o trecho um é responsável pela aceleração do alimentador cuja inclinação depende do tempo de aceleração programado acrescido do tempo de subida do servoacionamento e da amplitude programada da velocidade da linha. O segundo trecho cuja velocidade é constante e igual à programada e o tempo desse trecho dependerá do avanço escolhido e do tempo programado para a desaceleração do alimentador, sendo esse tempo programado e acrescido do tempo de descida do servoacionamento.



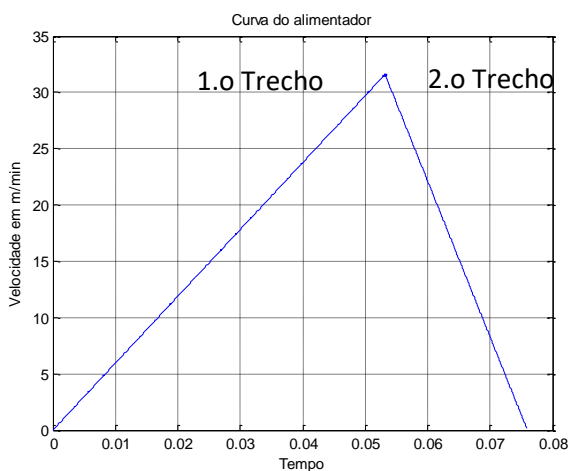
Dados de programação:

1. Avanço = 100mm.
2. Velocidade da linha = 60m/min.
3. Tempo de aceleração = 0.05s
4. Tempo de desaceleração = 0.05s
5. Redução = 10
6. Diâmetro do rolo alimentador = 80mm
7. Comprimento da roda = 400mm
8. Rotação do motor = 2000rpm
9. Velocidade da prensa = 30g/min
10. Tempo de subida = 0.001s
11. Tempo de descida = 0.001s

A figura acima é referente ao primeiro exemplo em três segmentos.

Da figura Velocidade x Tempo onde a área do trapézio é equivalente ao espaço percorrido pode ser calculado como: $(0,15 + 0,05) \times 1000/2 = 100\text{mm}$, onde $60\text{m/min} = 1000\text{mm/s}$.

O segundo exemplo a seguir têm somente dois trechos.



Dados de programação:

1. Avanço = 20mm.
2. Velocidade da linha = 60m/min.
3. Tempo de aceleração = 0.1s
4. Tempo de desaceleração = 0.1s
5. Redução = 10
6. Diâmetro do rolo alimentador = 80mm
7. Comprimento da roda = 400mm
8. Rotação do motor = 2000rpm
9. Velocidade da prensa = 30g/min
10. Tempo de subida = 0.001s
11. Tempo de descida = 0.001s

Da figura Velocidade x Tempo onde a área do triângulo é equivalente ao espaço percorrido pode ser calculado como: $0,075 \times 533,33/2 = 20\text{mm}$, onde $32\text{m/min} = 533,33\text{mm/s}$.

Fórmulas aplicadas no algoritmo do alimentador.

1. Rotação de saída $N2 = N1/R1$, onde $N1$ = Rotação do motor e $R1$ = redução do redutor;
2. Velocidade da linha em (mm) $VL = \text{PI} \times D1/1000 \times N2$, onde $D1$ = diâmetro do rolo alimentador e $N2$ rotação de saída do redutor;
3. $C1 = \text{PI} \times D2$, onde $D2$ igual ao diâmetro da roda medidora e $C1$ igual ao comprimento da roda medidora;
4. Resolução = $C1/N3$ - onde $N3$ igual ao número de pulsos do encoder da linha;
5. Tempo da 1.a e 2.a rampas $TA1 = TA1 + Tr$ e $TA2 = TA2 + Tf$;

6. Velocidade da linha em mm/s $\Rightarrow V_x = VL \cdot 1000 / 60$;
7. Aceleração do trecho 1 $\Rightarrow A_4 = V_x / TA_1$;
8. Mínimo passo = 6mm $\Rightarrow s = \frac{1}{2} \times A_4 \times TN_1^2$ e $TN_1 = \sqrt{2 \cdot (A_1 - 6) / A_4}$, onde TN_1 é o tempo do trecho 1 e 6mm é o ponto de início da desaceleração;
9. Velocidade máxima do trecho 1 $\Rightarrow CN_1 = A_4 \times TN_1$;
10. Espaço percorrido do trecho 1 $\Rightarrow SN_1 = A_1 - 6$, onde A_1 é o avanço;
11. Espaço percorrido no trecho constante S_3 e espaço percorrido no trecho de descida S_2 ;
12. Tempo percorrido no trecho subida, constante e descida TN_1 , T_c e TA_2 ;
13. Tempo percorrido $S_2 \Rightarrow S_2 = 6\text{mm} = V_x \times TA_2 / 2$ e $TA_2 = 2 \times 6 / V_x = 12 / V_x$ e A_5 a aceleração $V_x = A_5 \times TA_2$ e;
14. Se ($V_x > CN_1$)
 $V_x = CN_1$
 $TA_1 = TN_1$;
 $S_3 = 0$;
 $A_4 = V_x / TA_1$;
 $S_1 = SN_1$;
 $T_c = 0$;
 $TA_2 = 12 / V_x$;
 $A_5 = V_x / TA_2$;
 $S_2 = A_5 / 2 \times TA_2^2$;
 $TA_3 = TA_1 + TA_2$;

- Se ($V_x \geq CN_1$)
 $S_1 = A_4 / 2 \times TA_1^2$;
 $A_5 = V_x / TA_2$;
 $S_2 = A_5 / 2 \times TA_2^2$;
 $S_3 = A_1 - S_1 - S_2$;
 $S_3 = \text{abs}(S_3)$;
 $T_c = S_3 / V_x$;
 $TA_3 = TA_1 + TA_2 + T_c$;

15. Número de pulsos por segmento para o motor;
16. $NP_1 = S_1 / \text{Resol}$; % Rampa 1 aceleração;
17. $NP_2 = S_2 / \text{Resol}$; % Rampa 2 desaceleração;
18. $NP_3 = S_3 / \text{Resol}$; % Velocidade constante;
19. Frequência dos pulsos por segmento;
20. $f_1 = NP_1 / TA_1$; % frequência na rampa 1;
21. $f_2 = NP_2 / TA_2$; % frequência na rampa 2;

22. $f_3 = NP_3/T_c$; % frequência na velocidade constante;
23. Tempo de uma batida da prensa dividida por 2;
24. $TC_1 = V_2/120$; % Velocidade da prensa em segundos dividida por 2 ciclos subida e descida;
25. Angulo de carregamento do alimentador;
26. $ang = 180 \cdot T_3/TC_1$; % $T_3 =$ Tempo total sobre o tempo de um ciclo x 180 graus;
27. $NT = NP_1+NP_2+NP_3$; % Total de pulsos gerados para o motor;
28. Primeira projeção da rampa de subida de 0 a TA_1 ;
29. $x = 0:0.0001:TA_1$; % Faixa da rampa 1;
30. $VT_1 = A_4 \cdot x'$; % Valor da velocidade de saída em mm/s;
31. Segunda projeção do segmento constante TA_1 a $TA_1 + T_c$ (tempo de veloc. cte);
32. $x_1 = TA_1:0.0001:TA_1+T_c$; % Faixa da parte constante;
33. $VT_2 = V_x$; % Velocidade constante
34. Terceiro trecho da rampa de descida de T_1+T_c a T_3 ;
35. $x_2 = TA_1+T_c:0.0001:TA_3$; % Faixa na rampa 2;
36. $x_3 = x_2 - (TA_1+T_c)$; % troca de variável para decrescente;
37. $VT_3 = VT_2 - A_5 \cdot x_3$; % Velocidade de saída em mm/s;
38. $VT_3 = \text{abs}(VT_3)$;

Programação com microcontrolador SP32 como controlador do alimentador de prensa.

A implementação usando SP32 na plataforma do Arduino e verificação através do plotter e monitor serial, conforme a seguir.

```
// Projeto alimentador de prensa ESP32
// Criado em 12/01/2022 - Lucapel Comercial Ltda
// Projeto elaborado por MSc prof. Luís Caldas
// Parâmetros do alimentador - Programação normal
// 1. Passo 2. Velocidade trabalho 3. Tolerância
// Parâmetros de máquina do alimentador
// 1. Tempo de aceleração T1 2. Ponto de desaceleração T2 3. Rotação máxima do motor
N1
// 4. Redução R1 5. Diâmetro do rolo alimentador D1 6. Diâmetro da roda D2 7. Velocidade
da prensa VL
// 8. Tempo de subida do conjunto Tr 9. Tempo de descida do conjunto Tf
```

```
// 10. Tensão máxima de referência Up 11. Número de pulsos do encoder da linha N3
// Fórmulas aplicadas
// Rotação de saída N2 = N1/R1, onde N1 = Rotação do motor e R1 = redução do redutor
// Velocidade da linha em (mm) VL = PI x D1/1000 x N2, onde D1 = diâmetro do rolo
alimentador e N3 rotação de saída do redutor
// C1 = PI x D2, onde D2 igual ao diâmetro da roda medidora e C1 igual ao comprimento da
roda medidora
// Resolução = C1/N3 - onde N3 igual ao número de pulsos do encoder da linha
// Tempo da 1.a e 2.a rampas TA1 = TA1 + Tr e TA2 = TA2 + Tf
// Velocidade da linha em mm/s Vx = VL*1000/60
// Aceleração do trecho 1 A4 = Vx/TA1
// Mínimo passo = 6mm s = 1/2 A4 TN1^2 e TN1 = sqrt (2*(A1 - 6)/A4), onde TN1 é o
tempo do trecho 1 e 6mm ponto de desaceleração
// Velocidade máxima do trecho 1 CN1 = A4 x TN1
// Espaço percorrido do trecho 1 SN1 = A1 - 6
// Espaço percorrido no trecho constante S3 e espaço percorrido no trecho de descida
S2
// Tempo percorrido nos trechos subida, constante e descida TN1, Tc e TA2
// Tempo percorrido S2 S2 = 6mm = Vx x TA2/2 e TA2 = 2 x 6/Vx = 12/Vx e A5 a
aceleração Vx = A5 x TA2 e
// Se (Vx > CN1) então Vx = CN1 TA1 = TN1 S3 = 0 A4 = Vx/TA1 S1 = SN1 Tc = 0 TA2 =
12/Vx A5 = Vx/TA2 S2 = A5/2 * TA2^2 TA3 = TA1 + TA2;
// Se (Vx >=CN1) então S1 = A4/2 * TA1^2 A5 = Vx/TA2 S2 = A5/2 * TA2^2 S3 = A1 -
S1 -S2, S3 = abs(S3) Tc = S3/Vx TA3 = TA1 + TA2 + Tc;
// % Número de pulsos por segmento para o motor
// NP1 = S1/Resol; % Rampa 1 aceleração
// NP2 = S2/Resol; % Rampa 2 desaceleração
// NP3 = S3/Resol; % Velocidade constante
// Frequência dos pulsos por segmento
// f1 = NP1/TA1; % frequência na rampa 1
// f2 = NP2/TA2; % frequência na rampa 2
// f3 = NP3/Tc; % frequência na velocidade constante

// Tempo de uma batida da prensa dividida por 2.
// TC1 = V2/120; % Velocidade da prensa em segundos dividida por 2 ciclos subida e
descida

// Angulo de carregamento do alimentador
// ang = 180*T3/TC1; % T3 = Tempo total sobre o tempo de um ciclo x 180 graus
```

```
// NT = NP1+NP2+NP3; %Total de pulsos gerados para o motor
```

```
// Primeira projeção da rampa de subida de 0 a TA1
```

```
// x = 0:0.0001:TA1; % Faixa da rampa 1
```

```
// VT1 = A4*x'; % Valor da velocidade de saída em mm/s
```

```
// Segunda projeção do segmento constante TA1 a TA1 + Tc (tempo de veloc. cte)
```

```
// x1 = TA1:0.0001:TA1+Tc; % Faixa da parte constante
```

```
// VT2 = Vx; % Velocidade constante
```

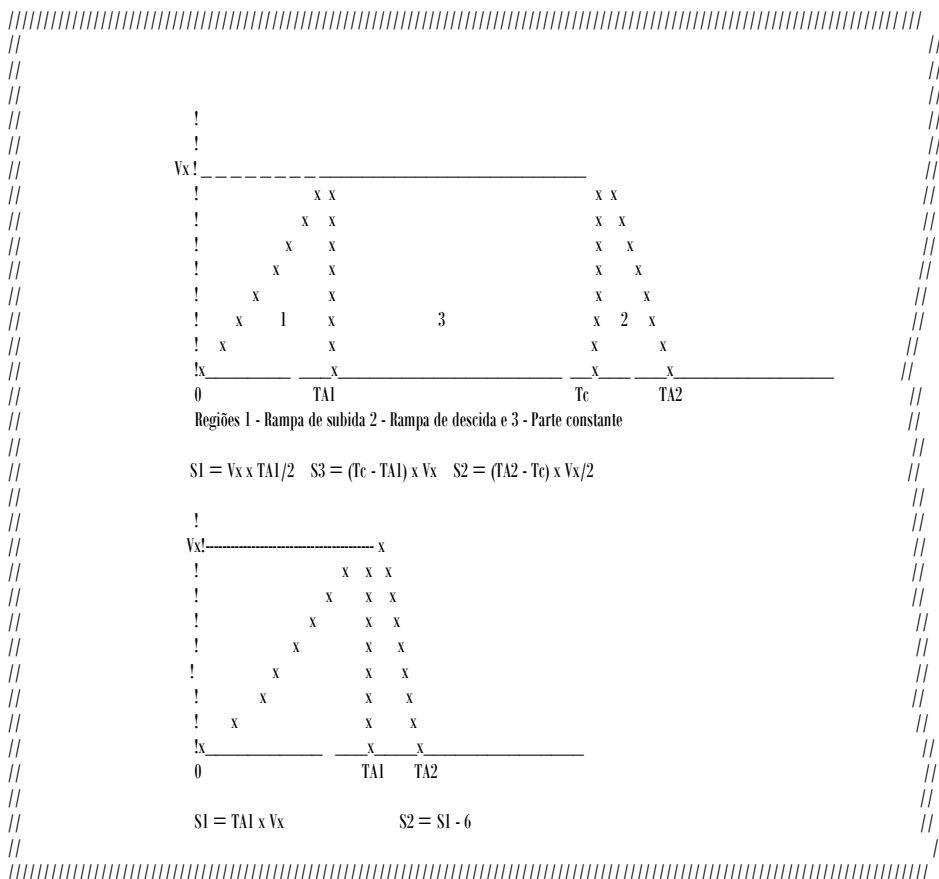
```
// Terceiro trecho da rampa de descida de T+Tc a T3
```

```
// x2 = TA1+Tc:0.0001:TA3; % Faixa na rampa 2
```

```
// x3 = x2 - (TA1+Tc); %troca de variável para decrescente
```

```
// VT3 = VT2 - A5*x3; % Velocidade de saída em mm/s
```

```
// VT3 = abs(VT3);
```



```
#include <SPI.h>
#include <Ethernet.h>           // incluir a biblioteca para comunicação I2C

// Define saída analógica
#define DAC1 25 // pino de saída analógica 1
boolean run = false;
boolean manual;
boolean producao;
boolean t;
//Declaração das variáveis globais
float ang;
int z;
int i;
int cod_event;
int atraso;
int for_loop;
char recebido[32];
float VA[100]; // Vetores para montagem das senóides
float VT[10000];
//float VT1[1000];
//float VT2[1000];
//float VT3[1000];

int c[32]; // = {0,0,0,0};
int Corr;
int Corr1;
double x = 0.00000;
float x1 = 0.00000;
float x2 = 0.00000;
float x3 = 0.00000;
float Tr = 0.0010;
float Tf = 0.0010;
float TA1 = 0.00000;
float TA2 = 0.00000;
float TA3 = 0.00000;
float Tc = 0.00000;
float TC1 = 0.00000;
float TNP;
float TN1;
```

```
float CN1;  
float SN1 = 0.00000;  
float S1 = 0.00000;  
float S2 = 0.00000;  
float S3 = 0.00000;
```

```
float Vx;  
float V1;  
float V2;  
float A1;  
float Ac4;  
float Ac5;  
float Ac6;
```

```
float NP1;  
float NP2;  
float NP3;  
float NT;
```

```
float Resol;
```

```
float f1;  
float f2;  
float f3;
```

```
// Definição dos pinos da comunicação serial TXD2 pin o 17 e RXD2 pino 16  
#define TXD2 17  
#define RXD2 16
```

```
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600);          // inicia o Serial Monitor, necessário se quiser depurar este  
  sketch  
  Serial2.begin(4800, SERIAL_8N1,RXD2,TXD2); //Serial Rx1 = 8 e Tx1 = 10 pinos  
  // Start the software serial port, to another Master  
  Serial2.flush(); // esvazia buffer comunicação  
  //Serial.println(x);  
  //delay(1000);
```



```
run = true;
t = false;
} // Fim do setup

void loop() {
  if (t == 0) {
    processo();
  }
}

void processo () {
  TA1 = 0.05;
  //Serial.println(TA1);
  TA2 = 0.05;
  V1 = 60;
  A1 = 100;
  Tr = 0.001;
  Tf = 0.001;
  TA1 = TA1 + Tr;
  //Serial.println(TA1);
  //delay(1000);
  TA2 = TA2 + Tf;
  Vx = V1*1000/60; // Velocidade da linha em (mm/s)
  //Serial.println(Vx);
  //delay(1000);
  Ac4 = Vx/TA1; // Aceleração A4 e Delta t = T1
  //Serial.println(Ac4);
  //delay(1000);
  TNP = 2*(A1 - 6)/Ac4; //Mínimo passo = 6mm - Triangulo e TNP é ao quadrado (s = 1/2
  gama t^2)
  TN1 = sqrt(TNP); // Tempo total percorrido no trecho 1 do triangulo
  //Serial.println(TN1);
  //delay(1000);
  CN1 = Ac4*TN1; // Velocidade final no trecho 1
  //Serial.println(CN1);
  //delay(1000);
  SN1 = A1 - 6; // Area faltante para terminar o passo.

  // Decisão se vai ser triangulo ou trapézio
  // Se Vx > CN1 => Triangulo
```

```
// Se não  $Vx < CN1 \Rightarrow$  Trapézio
// Para gerar triangulo
if ( $Vx > CN1$ ){
     $Vx = CN1$ ;
    //Serial.println( $Vx$ );
    //delay(1000);
     $TA1 = TN1$ ;
    //Serial.println( $TA1$ );
    //delay(1000);
     $S3 = 0$ ; // Velocidade do trecho 2
     $Ac4 = Vx/TA1$ ; // Aceleração trecho 1
    //Serial.println( $Ac4$ );
    //delay(1000);
     $S1 = SN1$ ; // Espaço percorrido trecho 1
     $Tc = 0$ ; // Tempo percorrido no trecho 2
     $TA2 = 12/Vx$ ; // Do triangulo Area =  $6mm$ , tempo de desacelerar =  $TA2$  e velocidade de
desaceleração  $Vx \Rightarrow s = Vx TA2$  e base =  $TA2$ , altura  $Vx$  dividido por 2
     $Ac5 = Vx/TA2$ ; // Aceleração Delta  $V/\Delta T$ 
     $S2 = Ac5/2 * sq(TA2)$ ; //Espaço percorrido 1/2 de gama  $t^2$ 
     $TA3 = TA1 + TA2$ ; // Tempo total  $TA3$ 
}
// Para gerar trapézio
else {
     $S1 = Ac4/2 * sq(TA1)$ ;
    //Serial.println( $Ac4$ );
    //delay(1000);
    //Serial.println( $S1$ );
     $Ac5 = Vx/TA2$ ;
    //Serial.println( $Ac5$ );
     $S2 = Ac5/2 * sq(TA2)$ ;
     $S3 = A1 - S1 - S2$ ;
     $S3 = abs(S3)$ ;
    //Serial.println( $S3$ );
     $Tc = S3/Vx$ ;
     $TA3 = TA1 + TA2 + Tc$ ;
    //Serial.println( $TA3$ );
    //delay(5000);
}
```

```
// Número de pulsos por segmento para o motor
t = 1;
// Cálculo for for_loop

for_loop = TA3/0.0001;

NP1 = S1/Resol; // Rampa 1 aceleração
NP2 = S2/Resol; // Rampa 2 desaceleração
NP3 = S3/Resol; // Velocidade constante

NT = NP1+NP2+NP3; //Total de pulsos gerados para o motor

// frequencia dos pulsos por segmento

f1 = NP1/TA1; // frequencia na rampa 1
f2 = NP2/TA2; // frequencia na rampa 2
f3 = NP3/Tc; // frequencia na velocidade constante

//Tempo de uma batida da prensa dividida po 2.
TC1 = V2/120; // Velocidade da prensa em segundos dividida por 2 ciclos subida e descida

// Angulo de carregamento do alimentador

ang = 180*TA3/TC1; // TA3 = Tempo total sobre o tempo de um ciclo x 180 graus
//Serial.println("Inicio");
//Serial.println(TA1);
//Serial.println(Ac4);
Serial.println("Primeiro trecho");
delay(1000);
Corr = abs(10000*TA1);
Serial.println(Corr);
// Primeira projeção da rampa de subida de 0 a TA1
for (i = 1; i <= 10000*TA1; i = i + 1){ // Faixa da rampa 1
    x = x + 1;
    VT[i] = Ac4*x; // Valor da velocidade de saída em mm/s
    Serial.println(VT[i]/10000);
    //delay(10);
    //Serial.println(i);
    //delay(1000);
```

```
}// Fim do for

// Segunda projeção do segmento constante TA1 a TA1 + Tc (tempo de veloc. cte)
Serial.println("Segundo trecho");
//delay(1000);
Corr1 = abs(10000*(TA1 + Tc));
for (i = 10000*TA1; i <= 10000*(TA1+Tc); i++){ // Faixa da parte constante
  x1 = x1 + 1;
  VT[i] = Vx; // Velocidade constante
  Serial.println(VT[i]);
  //delay(1000);
  //Serial.println(i);
}

// Fim do for
// Terceiro trecho da rampa de descida de TA1+Tc a TA3

Serial.println("Terceiro trecho");
//Serial.println(Vx);
//Serial.println(Ac5);
//delay(1000);
for (i = 10000*(TA1+Tc); i <= 10000*TA3; i++){ // Faixa na rampa 2
  x2 = i + 1;
  x3 = x2 - 10000*(TA1+Tc); //troca de variável para decrescente
  //Serial.println(x3);
  VT[i] = 10000*Vx - Ac5*x3; // Velocidade de saída em mm/s
  //Serial.println(VT[i]);
  //delay(50);
  VT[i] = abs(VT[i]);
  VT[i] = VT[i]/10000;
  Serial.println(VT[i]);
  //delay (200);
  //Serial.println(i);
}

//Fim do for
t = 1; i = 1; x = 0; x1 = 0; x2 = 0; x3 = 0;
//VT = VT1 + VT2 + VT3
}

// Fim processo
```

Programação com MATLAB

```
%Alimentador de prensa versão 1.1
% Prof. Luís Caldas
% 01/01/2022
%Geração do trapézio ou triangulo
% Rotação do motor, Tempo de aceleração T1 e desaceleração T2, Passo do
% alimentador, Redução do motor, número de pulsos do encoder, diâmetro da
% roda medidora, diâmetro do rolo alimentador e tempo de subida e descida
% do conversor, velocidade da prensa, frequência dos pulsos do conversor e
% angulo de carregamento.
clc;
clear all;
close all;
A1 = input('Entrar com o passo em milímetros (mm) = ');
V1 = input('A velocidade de trabalho em (m/min)= ');
T1 = input('Tempo de aceleração em (seg) = ');
T2 = input('Tempo de desaceleração em (seg) = ');
R1 = input('Rotação do motor em (rpm) = ');
R2 = input('Redução do redutor = ');
D1 = input('Diâmetro do cilindro ou rolo alimentador em (mm) = ');
C2 = input('Comprimento da roda medidora em (mm) = ');
N1 = input('Número de pulsos do encoder da linha em (pps) = ');
V2 = input('Velocidade da prensa em (golpes/min) = ');
Tr = input('Tempo de subida em (seg) do conjunto = ');
Tf = input('Tempo de descida em (seg) do conjunto = ');

%Processo 1
%Cálculo da rotação de saída do redutor
RT = R1/R2;
%fprintf(' A rotação de saída do redutor é % em rpm => RT = ',RT);
%Cálculo da velocidade da linha
VL = 3.14 * D1/1000 * RT; % VL = PI X D x N - Velocidade em m/min
%fprintf(' A velocidade da linha em (m/min) = ',VL);

%Resolução do passo
Resol = C2/N1; %

% Cálculo do comprimento da rampa em (mm) Quanto que andou o alimentador
```

% durante a aceleração.

% Aceleração = (Velocidade final - velocidade inicial)/Tempo final - Tempo inicial

% Cálculo da velocidade real do alimentador

% Restrição para o tempo de aceleração x passo e velocidade do alimentador

% Ponto de desaceleração igual a 6mm

% Se $S1 \geq$ Passo - 10mm, calcular a velocidade de trabalho e tempo de aceleração

% Teste de S1 Mínimo passo 10mm

$T1 = T1 + Tr$

$T2 = T2 + Tf$

$Vx = V1 * 1000 / 60$; % Velocidade da linha em (mm/s)

$A4 = Vx / T1$; % Aceleração A4 e Delta t = T1

$TNP = 2 * (A1 - 6) / A4$; % 6mm ponto de mínimo do passo

$TN1 = \text{sqrt}(TNP)$;

$CN1 = A4 * TN1$;

$SN1 = A1 - 6$;

if ($Vx > CN1$)

$Vx = CN1$;

$T1 = TN1$;

$S3 = 0$;

$A4 = Vx / T1$;

$S1 = SN1$;

$Tc = 0$;

$T2 = 12 / Vx$;

$A5 = Vx / T2$;

$S2 = A5 / 2 * T2^2$;

$T3 = T1 + T2$;

else

$S1 = A4 / 2 * T1^2$;

$A5 = Vx / T2$;

$S2 = A5 / 2 * T2^2$;

$S3 = A1 - S1 - S2$;

$S3 = \text{abs}(S3)$;

$Tc = S3 / Vx$;

$T3 = T1 + T2 + Tc$;

```
end
% CÁLCULO DO for_loop
for_loop = T3/0.0001;

% Número de pulsos por segmento para o motor
NP1 = S1/Resol; % Rampa 1 aceleração
NP2 = S2/Resol; % Rampa 2 desaceleração
NP3 = S3/Resol; % Velocidade constante

NT = NP1+NP2+NP3; % Total de pulsos gerados para o motor

% frequência dos pulsos por segmento
f1 = NP1/T1; % frequência na rampa 1
f2 = NP2/T2; % frequência na rampa 2
f3 = NP3/Tc; % frequência na velocidade constante

% Tempo de uma batida da prensa dividida por 2.
TC1 = V2/120; % Velocidade da prensa em segundos dividida por 2 ciclos subida e
descida

% Angulo de carregamento do alimentador
ang = 180*T3/TC1; % T3 = Tempo total sobre o tempo de um ciclo x 180 graus

% Primeira projeção da rampa de subida de 0 a T1
x = 0:0.0001:T1; % Faixa da rampa 1
VT1 = A4*x; % Valor da velocidade de saída em mm/s
plot(x,VT1*0.06); % Velocidade em m/min
grid
hold on

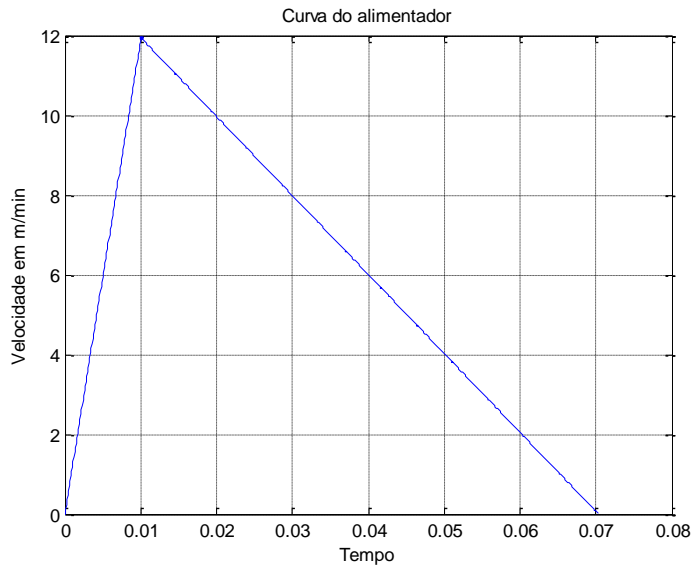
% Segunda projeção do segmento constante T1 a T1 + Tc (tempo de veloc. cte)
x1 = T1:0.0001:T1+Tc; % Faixa da parte constante
VT2 = Vx; % Velocidade constante
plot(x1,VT2*0.06);

% Terceiro trecho da rampa de descida de T1+Tc a T3
x2 = T1+Tc:0.0001:T3; % Faixa na rampa 2
x3 = x2 - (T1+Tc); % troca de variável para decrescente
VT3 = VT2 - A5*x3; % Velocidade de saída em mm/s
```

```

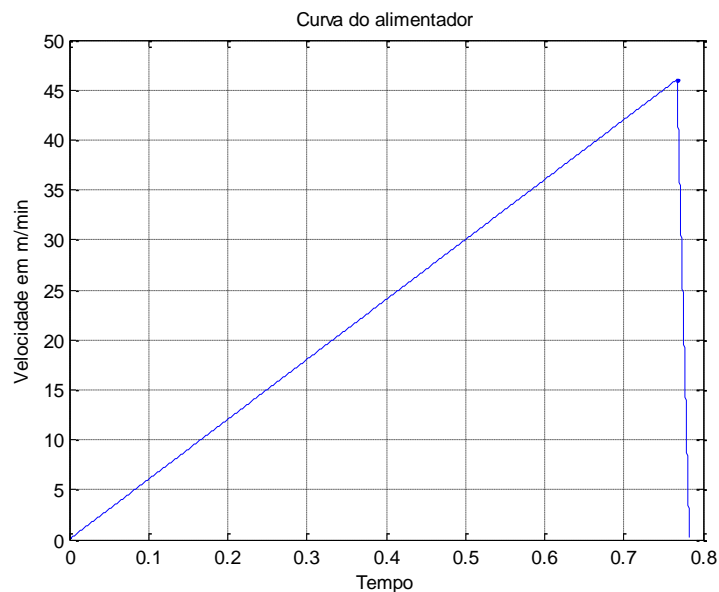
VT3 = abs(VT3);
plot(x2,VT3*0.06); % Velocidade de saída em m/min
xlabel('Tempo');
ylabel('Velocidade em m/min');
title('Curva do alimentador');
    
```

Exemplo: Um avanço de 7mm com velocidade de 120m/min e aceleração e desaceleração em 100ms.



Comprovação do evento

$$\text{Triângulo} = 0,07 \times 12.000/60 / 2 = 7\text{mm}$$



$$\text{Triângulo} = 0,78 \times 46.154/60 / 2 = 300\text{mm}$$