

EXPERIÊNCIA 5
PROJETO DE CONTROLADOR EM TEMPO
DISCRETO

1 DESCRIÇÃO DA PLANTA

Nessa experiência, será projetado um controlador para o pouso de um módulo lunar. A descrição da planta é baseada na experiência 04 do curso AMME3500 de Stefan Williams. O objetivo é garantir que a nave chegue na posição $(X, Y) = (0, 0)$ com velocidade zero gastando o mínimo de combustível possível. A Figura 1 apresenta o esquemático do problema.

A dinâmica do sistema é descrita por:

$$\ddot{x} = (F_l \cos\theta - F_t \sin\theta)/m \quad (1.1)$$

$$\ddot{y} = (F_l \sin\theta + F_t \cos\theta)/m \quad (1.2)$$

$$\ddot{\theta} = 4F_l/J \quad (1.3)$$

Para mais detalhes da modelagem, veja Assignment 4 Lunar Lander Solution.

Para a experiência, a dinâmica da planta foi descrita nos arquivos *DinamicaCompleta.slx* e *Dinamica Modulo.slx*.

O arquivo *Dinamica Modulo.slx* implementa as equações dinâmicas do sistema. Os blocos estão exibidos na Figura 2.

Já o arquivo *DinamicaCompleta.slx* faz as integrações necessárias para dinâmica e inclui o feedback de θ . Esse módulo é exibido na Figura 3.

Os módulos estão em função dos parâmetros dos problemas, descritos na tabela 1 e podem ser inicializados com o arquivo *InitialParams.m*.

Note que para esse problema, as variáveis controláveis são as forças de propulsão vertical e lateral e existem 3 variáveis livres x, y, θ acopladas pela dinâmica do módulo. Além disso, a planta apresenta não-linearidades, como a dependência de seno e cosseno

Figura 1: Modelo 2D do módulo lunar

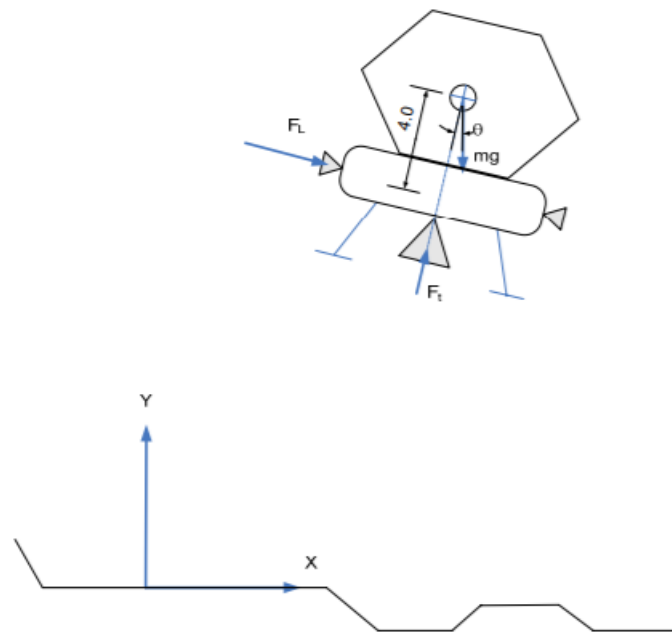


Tabela 1: Parâmetros físicos do modelo

Propriedade	Valor
Massa(M)	15.000kg
Massa de Combustível(MFuel)	8.000kg
Momento de Inércia (J)	100.000kg.m ²
Máxima propulsão horizontal(F_t)	44kN
Máxima propulsão vertical(F_l)	0, 5kN
Impulso específico do foguete (power)	3.0kNs/kg
Constante da gravidade (g)	1.6kg.m/s ²

de θ .

Figura 2: Equações Dinâmicas

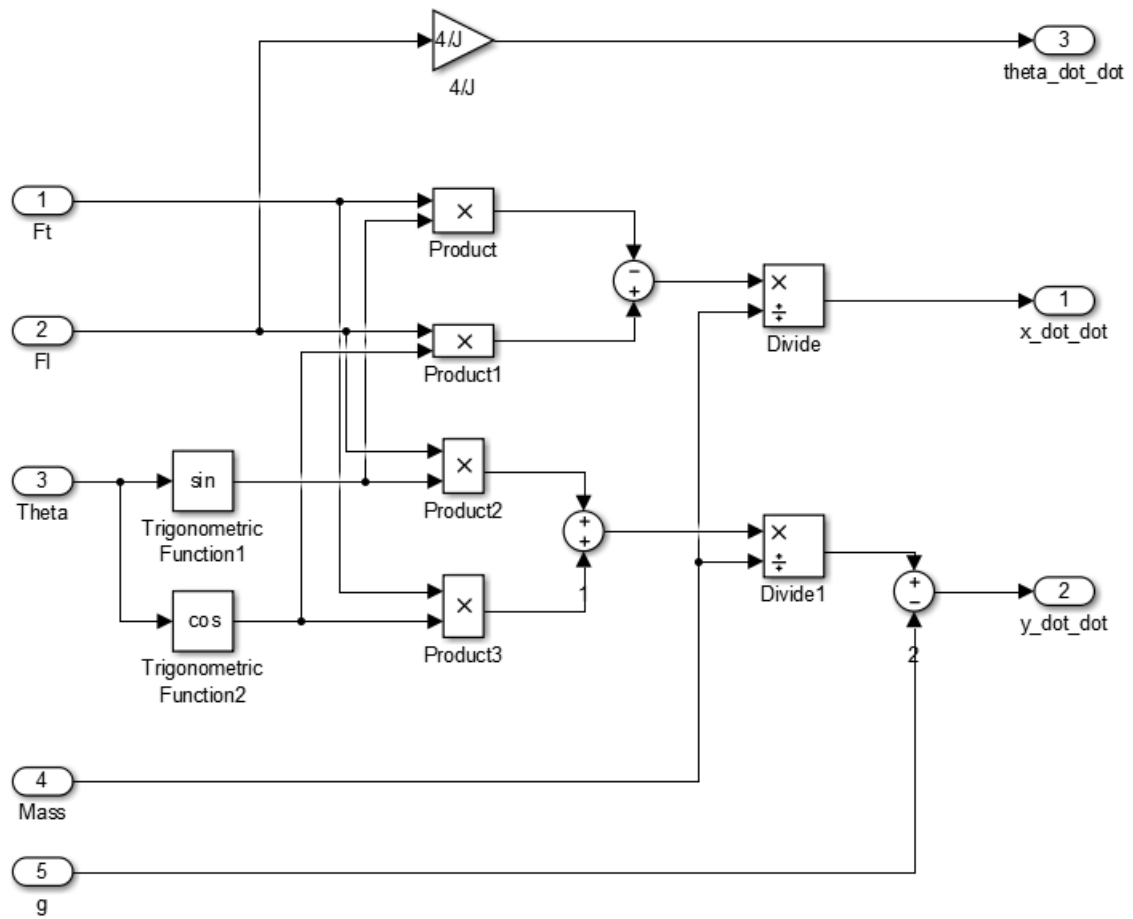
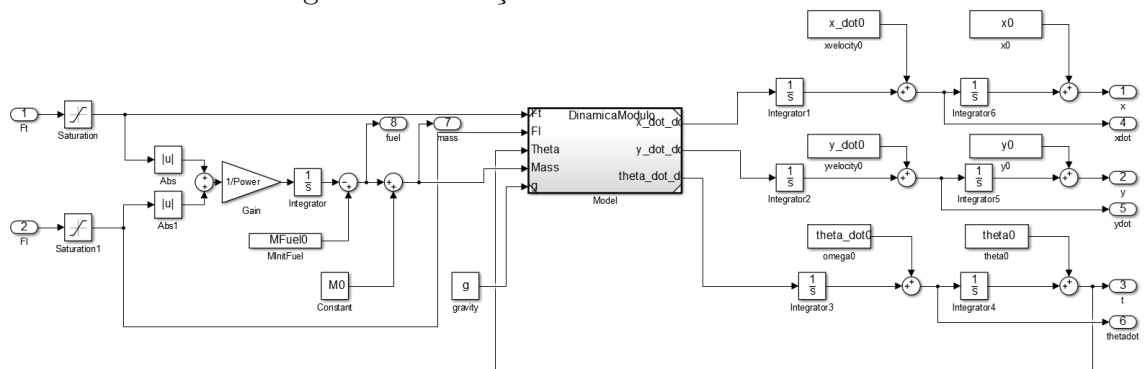


Figura 3: Condições iniciais e feedback de θ



2 ROTEIRO PARA PROJETO DO CONTROLADOR

O projeto de controladores em tempo discreto pode ser feito de duas formas:

- Projetar o controlador em tempo contínuo e aproximá-lo para um controlador em tempo discreto.
- Projetar diretamente em Z e implementar a equação de diferenças para o controlador.

Cada método tem seus pontos positivos e negativos. Na parte teórica da disciplina eles serão vistos em detalhes. Na experiência, adotará-se a primeira técnica.

Devido a dificuldade da planta, será descrito um roteiro para projeto dos três controladores.

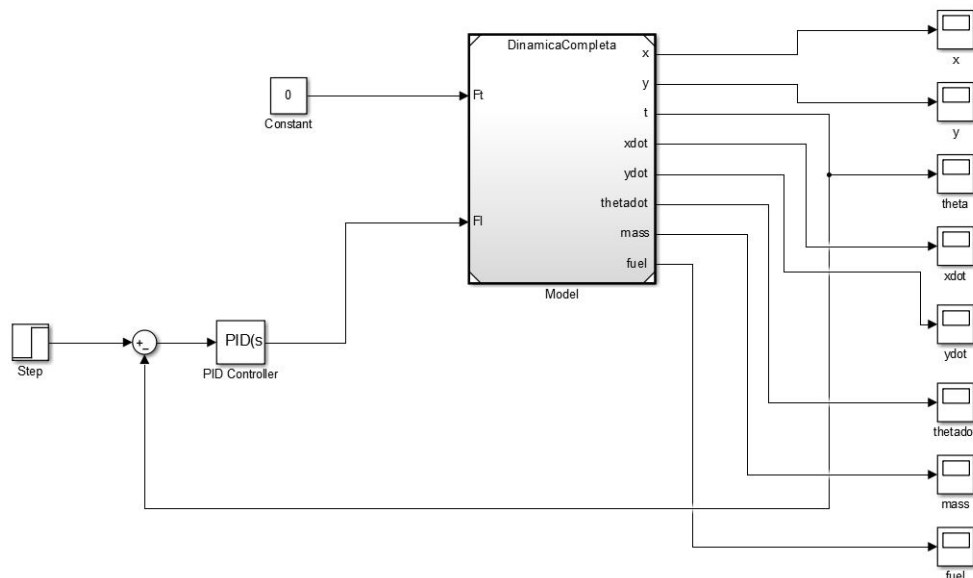
Contudo, fazendo-se uma análise preliminar, o sistema é puramente inercial. Assim, para estabilizá-lo será necessário de no mínimo um controlador derivativo. Como o objetivo da experiência é mais exercitar a passagem dos controladores em tempo contínuo para tempo discreto e não o projeto em si, recomenda-se que se restrinja aos controladores PD.

2.1 Controle contínuo de θ

Observando a equação 1.3, nota-se que apenas F_t influencia a dinâmica da planta. Projete um controlador PD que leve o sistema até $\theta = 0$. Para isso, utilize a planta do arquivo *simulation.slx*. Essa planta mantém a entrada F_t constante e igual a zero. A figura 4 exibe a planta para projeto do controle.

Para projetar o controle, utiliza-se o bloco PID do simulink. Esse bloco recebe quatro parâmetros:

Figura 4: Planta para projeto de θ



- K - Constante proporcional
- I - Constante integral
- D - Constante Derivativa
- N - Filtro para constante derivativa

O parâmetro N existe pois um controlador puramente derivativo é não-causal.

Dado as não linearidades da planta, o controle em malha fechada facilmente leva o sistema a instabilidade. Por conta disso, para o seu projeto parta de:

- $K = 1000$
- $I = 0$
- $D = 5000$
- $N = 100$

Então, simule a planta para valores diferentes e responda (pelo menos 3 controladores variando K, D):

- Quais os resultados de se aumentar o ganho K para essa planta ?

- Quais os efeitos de se alterar o ganho derivativo ?
- Dentre os controladores testados, qual o mais adequado para o projeto ? Avalie tempo de estabilização, esforço de controle, overshooting e outras coisas que julgar pertinente.

2.2 Controle para x

Observando a equação 1.1, nota-se que a força dominante em x é F_l , já que ela é multiplicada pelo cosseno do ângulo e, é de se esperar, que θ não se desvie muito de zero (Verifique nas simulações estáveis).

Semelhante ao controle em θ a figura 5 exhibe a planta para se avaliar os parâmetros do controlador. Nessa escolha de representação, o controlador PD foi representado pelo feedback dos sinais de velocidade e posição. Note que, ambos os controladores afetam a mesma variável de controle. Então, um controlador muito agressivo em θ pode afetar o desempenho em x . O arquivo *simulation_pdxslx* contém a planta a ser simulada.

Partindo de $K = 1$ e $D = 50$, realize as mesmas análises propostas para o controlador em θ ¹:

- Quais os resultados de se aumentar o ganho K para essa planta ?
- Quais os efeitos de se alterar o ganho derivativo ?
- Dentre os controladores testados, qual o mais adequado para o projeto ? Avalie tempo de estabilização, esforço de controle, overshooting e outras coisas que julgar pertinente.

2.3 Controle para y

O projeto de controle para y segue na mesma linhas dos anteriores. Observando a equação 1.2, a força dominante é F_t . A única observação a se fazer é que além do objetivo da posição, para esse controlador é indispensável que a velocidade de chegada ao solo seja baixa.

A figura ?? apresenta a planta com o controlador a ser projetado.

¹Caso encontre problemas de estabilidade nesse projeto, recomenda-se que se utilize o controlador padrão para θ e reavalie posteriormente a escolha.

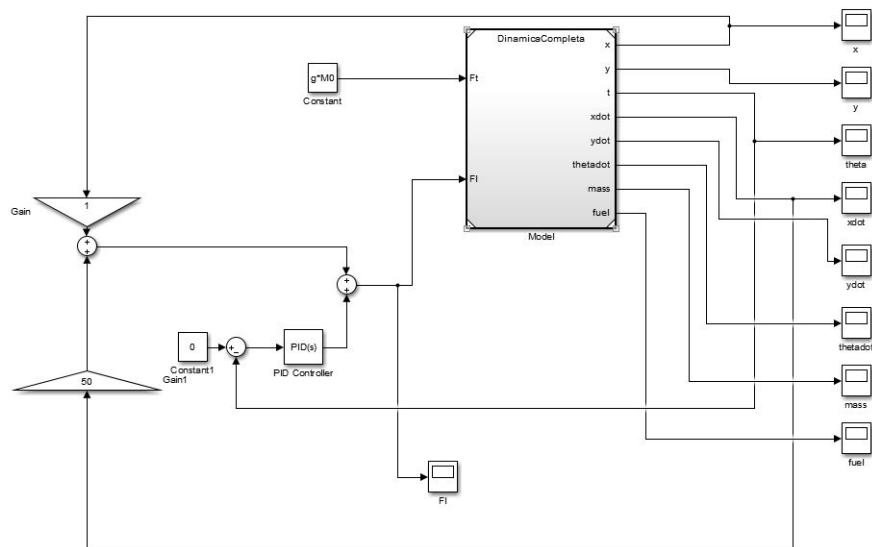


Figura 5: Planta para o projeto de x

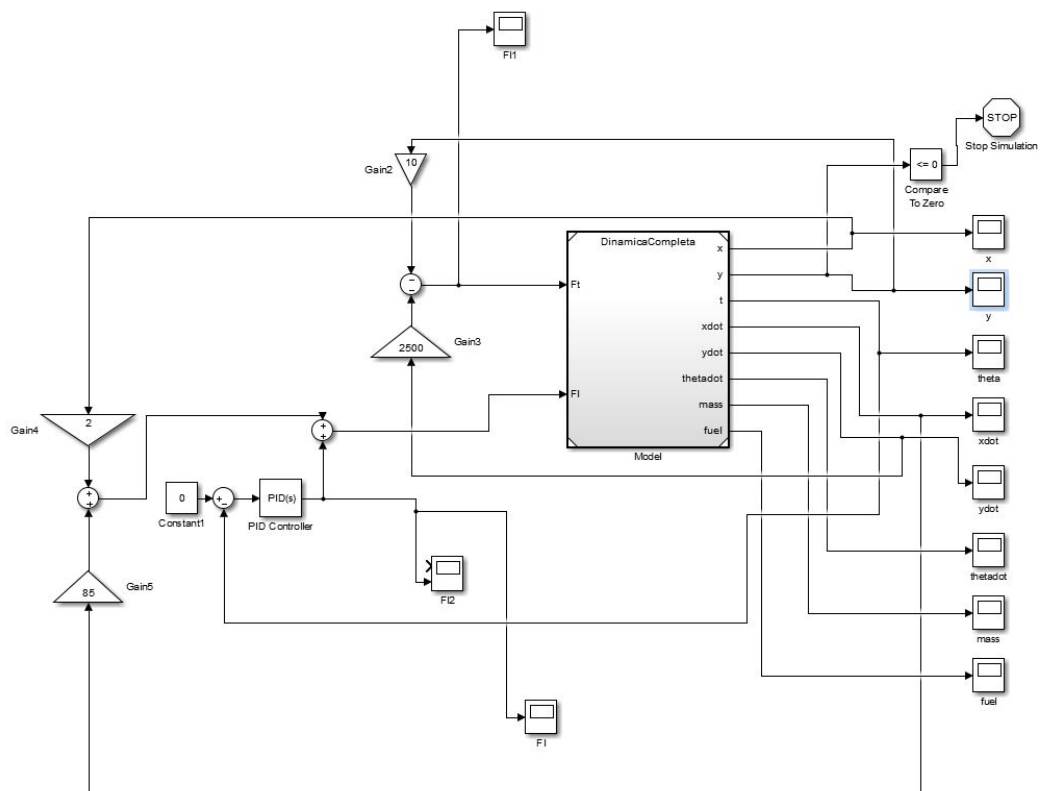
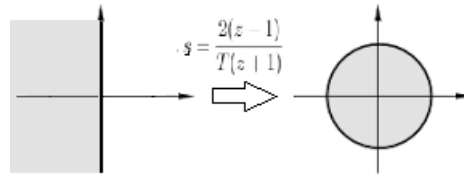


Figura 6: Planta para projeto do controlador em Y

Figura 7: Mapeamento pelo método de Tustin



Comece seus testes com $P = 10$ e $D = 2500$ para o controlador em y . Talvez seja necessário reajustar o ganho do controlador PD em x caso sejam encontradas instabilidades. O sistema em y é cerca de 5 vezes mais lento do que em x .

Descreva os controladores finais e os resultados encontrados.

2.4 Transformação de Filtros Analógicos em Digitais

A necessidade de se transformar os controladores analógicos projetados em digitais se deve ao fato de que essa transformação é necessária para se implementar o controlador em um computador digital. Atualmente, todos os controladores são implementados no computador digital.

Como nesse ponto já se tem os controladores, será apresentada uma forma de converter os controladores digitais em analógicos, utilizando as técnicas de manipulação simbólica do MATLAB.

2.4.1 Método de Tustin

A conversão pelo método de Tustin consiste em aproximar o filtro digital pelo filtro analógico calculado em $s = \frac{2(z-1)}{T(z+1)}$. Onde T é o período de amostragem. Essa transformação mapeia a área de estabilidade de s no círculo unitário em z , conforme a figura 7.

Esse mapeamento garante estabilidade para sistemas lineares. Contudo, existe uma distorção na resposta em frequência e períodos de amostragem muito longos mapeiam o sistema próximo a fronteira do círculo unitário, gerando respostas inadequadas.

Assim, a escolha do tempo de amostragem para essas transformações é um ponto crítico do projeto.

2.4.2 Manipulação simbólica no MATLAB

A transformação de controladores envolve um trabalho algébrico longo e "error-prone". Para minimizar a chance de erros, é possível utilizar as ferramentas de manipulação algébrica. Para exemplificar as técnicas, será mostrado o processo de conversão do filtro

$$G(s) = \frac{10(s+3)}{s(s+1)} \quad (2.1)$$

Primeiro, é necessário definir as variáveis relevantes de maneira simbólica com:

```
s = sym('s')
```

```
T = sym('T')
```

```
z = sym('z')
```

Após isso, define-se a função de transferência do filtro e a conversão de s para z:

```
s = 2*(z-1)/(T*(z+1))
```

```
Gs = 10*(s+3)/(s*(s+1))
```

Com isso, será exibido o controlador em função de T. Contudo, não estará na forma mais simplificada possível. Então, é possível simplificar a expressão com a função `simplifyFraction`

```
>> simplifyFraction(Gs)
```

```
ans =
```

```
(5*T*(z + 1)*(3*T + 2*z + 3*T*z - 2))/((z - 1)*(T + 2*z + T*z - 2))
```

Por fim, é necessário substituir o período de amostragem para obter a expressão final do controlador. Para isso, salve o resultado do passo anterior em uma variável auxiliar.

```
GsDiscreta = (5*T*(z + 1)*(3*T + 2*z + 3*T*z - 2))/  
((z - 1)*(T + 2*z + T*z - 2))
```

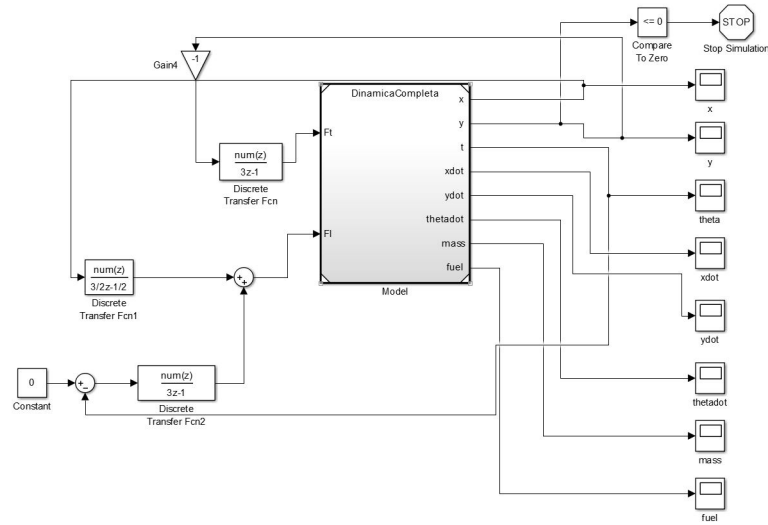
Por fim, é só atribuir diversos valores para T e utilizar a função `subs` para obter o controlador.

```
T = 0.1
```

```
subs(gsDiscreta)
```

É possível repetir esse processo para obter o controlador para diversos períodos de

Figura 8: Planta para validação final



amostragem.

2.4.3 Planta com controladores discretos

Converta os três controladores obtidos para a planta contínua e teste na planta do arquivo *simulation_discrete.slx* exibida na figura 8.

Para isso, teste a planta para diferentes tempos de amostragem e avalie o resultado. Caso tenha alguma dificuldade com a estabilidade do sistema discreto, pode-se utilizar outros arquivos de apoio para testar os controladores individualmente.

Resuma os seus resultados, escolhendo um período de amostragem e descrevendo o por que.

Não se esqueça que a representação do controlador PD projetado é na forma de:

$$G_s = P + I\frac{1}{s} + D\frac{N}{1 + N\frac{1}{s}} \quad (2.2)$$

Além disso, lembre-se de colocar o período de amostragem, T , correto na planta. Esse parâmetro deve ser colocado para cada uma das transformadas Z .

Faça uma avaliação global do projeto. Os resultados (tempo de pouso, posição final, combustível restante) são adequados ?