

Malha Fechada - Controlador Discreto

June 9, 2020

1 Modelo de Simulação

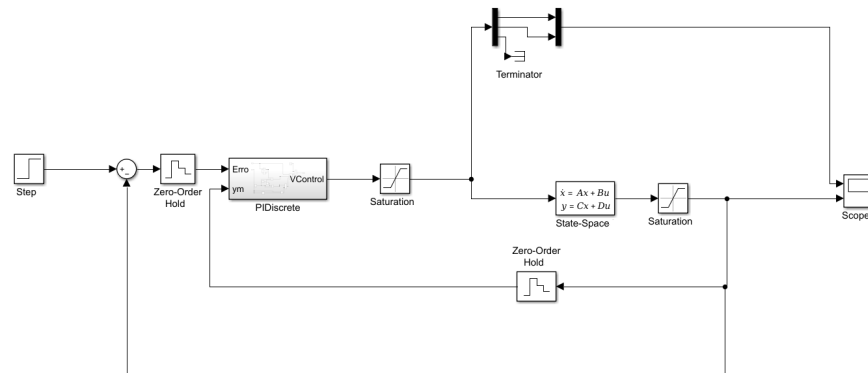


Figure 1: Diagrama de Blocos

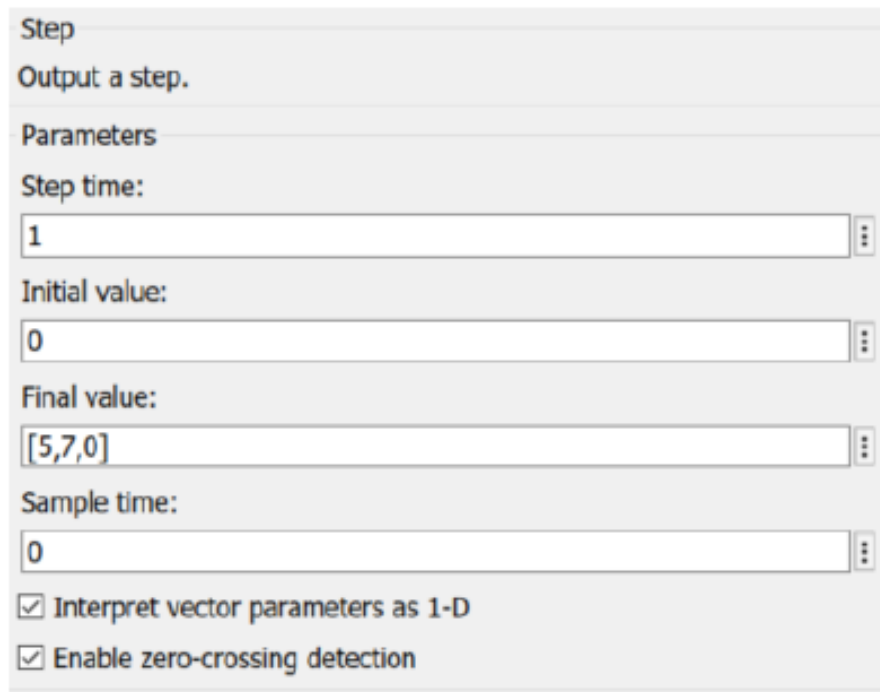
```
A1=5;
A2=5;
A3=5;
q11=5;
q12=2;
rh1=1;
rh2=1;
rh3=1;

A = [(-1/(A1*rh1)),0,0,(1/(A2*rh1)),(-1/(A2*rh2)),0,0,(1/(A3*rh2)),(-1/(A3*rh3))];
B=[1/A1,0,0,0,1/A2,0,0,0,0];
C=[1,0,0,0,1,0,0,0,1];
D=[0,0,0,0,0,0,0,0,0];

u=[q11;q12];
T=1;
taui=5; %ki=0.2;
taud=0.1; %kd=0;
kc=2;
```

Figure 2: Inicialização

Após a criação do diagrama de blocos e da definição dos valores iniciais para as matrizes e constantes, seguiu-se para o estabelecimento das referências e o design do PID usando o algoritmo da velocidade (figura 4):



Step

Output a step.

Parameters

Step time:

1

Initial value:

0

Final value:

[5,7,0]

Sample time:

0

☒ Interpret vector parameters as 1-D

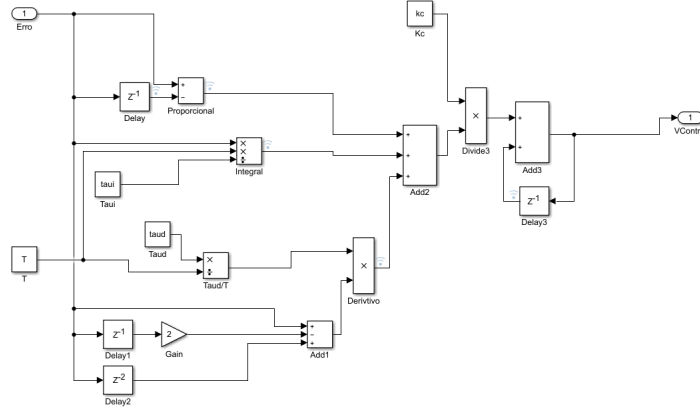
☒ Enable zero-crossing detection

Figure 3: Estabelecimento das Referencias

2 Simulação do PID

Procedeu-se com a mesma metodologia do PID contínuo, realizaram-se primeiro testes ao comportamento do controlador puramente proporcional. Inicialmente com $K_p=1$ (figura 5), tal como no PID contínuo, as entradas $q1$ e $q2$ começam com valores muito elevados, que apenas têm um limite devido a um bloco de saturação usado para simular as limitações físicas do sistema. Mas, em ambiente discreto isto apresenta um problema conhecido: verificou-se a presença de **saltos derivativos**.

Seguidamente verifica-se que, tal como o PID contínuo, com apenas ganho proporcional o sistema é incapaz de alcançar as referências.



$$c(n) = c(n-1) + K_c \left[(e(n) - e(n-1)) + \frac{T}{\tau_i} e(n) + \tau_d \frac{e(n) - 2e(n-1) + e(n-2)}{T} \right]$$

Figure 4: PIDVelocidade

Após a alteração progressiva dos ganhos (k_p , k_i e k_d) reparou-se que o sistema, como era de esperar, comporta-se da mesma forma que o contínuo e para evitar alguma redundância na informação, passar-se-á para os resultados mais notáveis.

Também se verificou que não havia diferença notável entre usar o algoritmo de posição, de velocidade e de posição modificado (figura ??). O controlo ótimo foi encontrado no algoritmo de velocidade modificado.

A implementação de uma componente integral também permite a correção dos saltos derivativos pois permite o uso de uma versão modificada do algoritmo da velocidade (figura 6). Isto é um detalhe importante **o algoritmo da velocidade modificado não funciona sem componente integral** pois se se analisar a equação na figura 6 verifica-se que a componente integral é a única que é influenciada pelo erro, todas as outras utilizam a medição da variável anterior que, se o k_i for 0, será sempre 0 logo sem componente integral o algoritmo modificado não tem resposta.

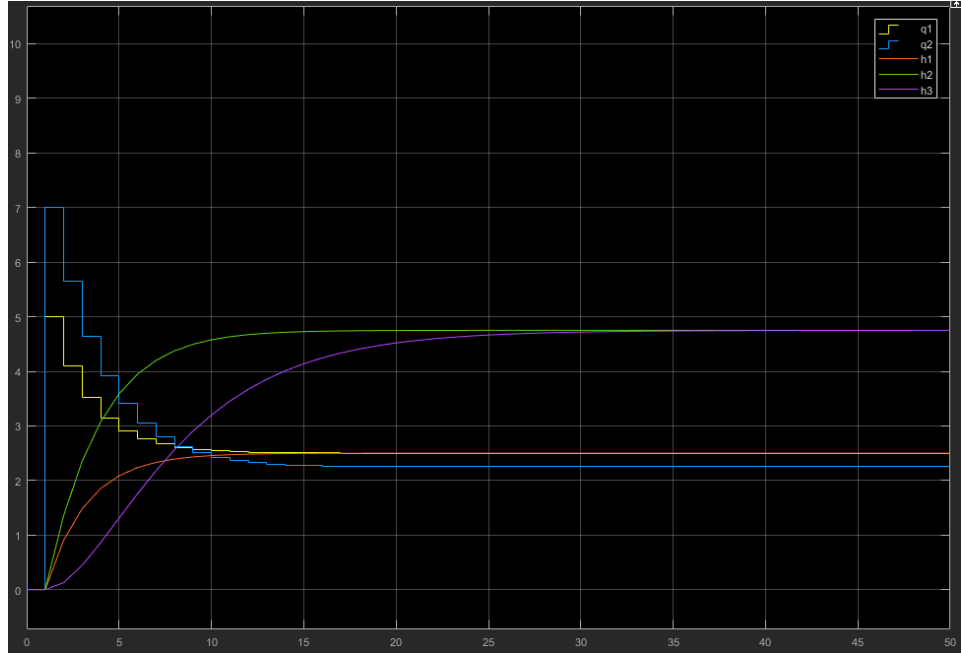
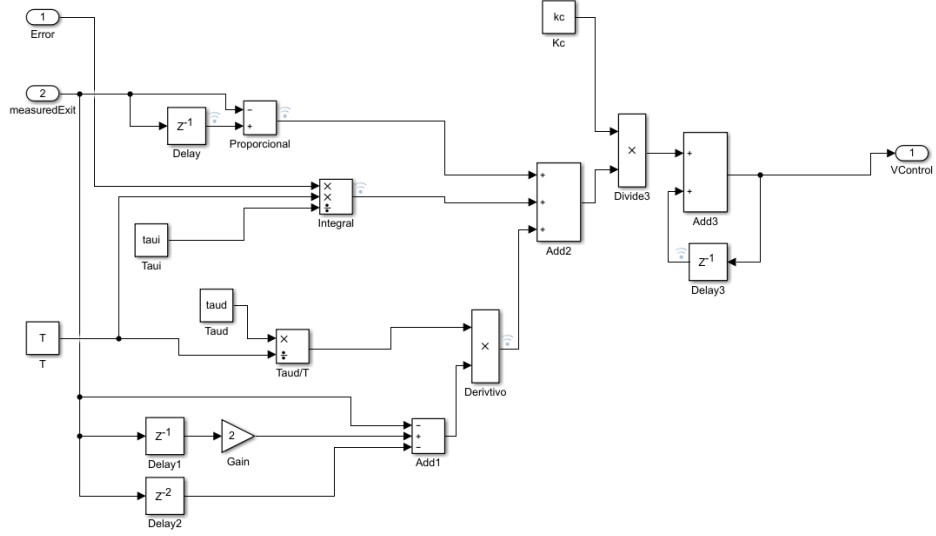


Figure 5: $K_p=1$



$$c(n) = c(n-1) + K_c \left[(y_m(n-1) - y_m(n)) + \frac{T}{\tau_i} e(n) + \tau_d \frac{-y_m(n) + 2y_m(n-1) - y_m(n-2)}{T} \right]$$

Figure 6: Algoritmo de Velocidade Modificado

Estabelecendo agora $K_i=0.2, K_p=2, k_d=0.1$ e um sample time de 1s:

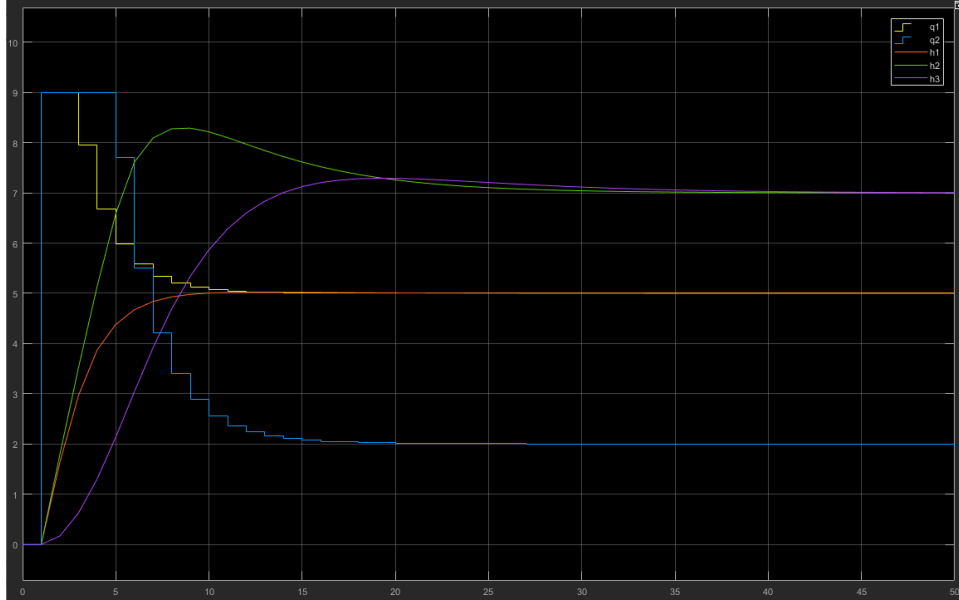


Figure 7: Resposta dos Algoritmos de Velocidade, Posição e Posição Modificado

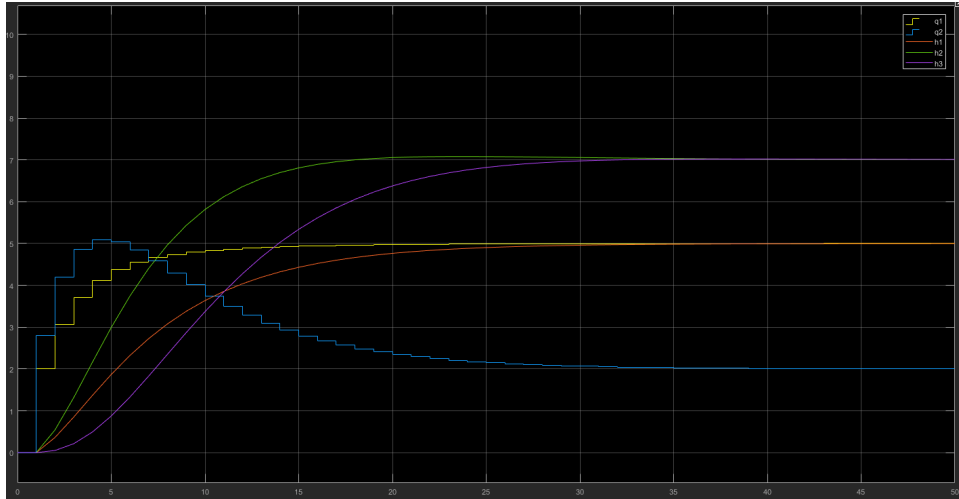


Figure 8: Resposta do Algoritmo de Velocidade Modificado

Na figura 7 com as novas componentes implementadas repara-se que os

saltos derivativos mantiveram-se e nota-se ainda um overshoot substancial que poderá ser consequência desses saltos juntamente com o efeito a componente integral, de qualquer das formas os saltos derivativos apresentam um problema para o sistema e devem ser corrigidos usando o algoritmo da velocidade modificado.

O caso ótimo do PID contínuo (figura 8) é melhorado pelo algoritmo do PID discreto. Os saltos derivativos foram eliminados: as saídas do controlador vão aumentando progressivamente em vez de começarem imediatamente no valor máximo (ação que pode causar estragos ao sistema) ainda que ao custo da diminuição da velocidade do sistema. É de salientar que apesar da velocidade ter diminuído o sistema foi capaz de controlar as variáveis sem overshoot ou oscilações notáveis.

3 Alteração do Sample Time

A escolha do período de amostragem é um dos passos essenciais para um bom controle em ambiente discreto. Para este sistema nota-se que até 1s de T (figura 8) o controle não apresenta riscos para o sistema sendo que as entradas q_1 e q_2 não são forçadas a alcançar valores altos subitamente e as alturas dos tanques atingem a referência. A partir de $T=2$ (figura 9) nota-se que as entradas já têm saltos possivelmente perigosos mas ainda não apresenta problemas em termos de alcançar a referência.

Para $T=3$ (figura 10) os saltos das entradas pioram e nota-se alguma dificuldade no controle das alturas, embora ainda alcancem as referências e se mantenham lá embora com alguma oscilação. Para $T=4$ (figura 11) já se nota grandes dificuldades no controle, o sistema é incapaz de se manter nas referências e as entradas apresentam saltos de grande magnitude, sendo por isso um perigo para o sistema.

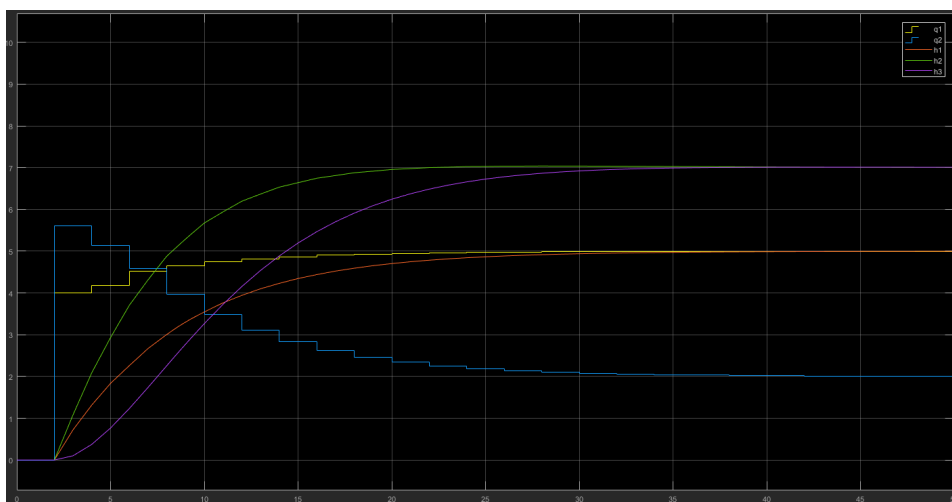


Figure 9: Resposta do Algoritmo de Velocidade Modificado com $T=2$

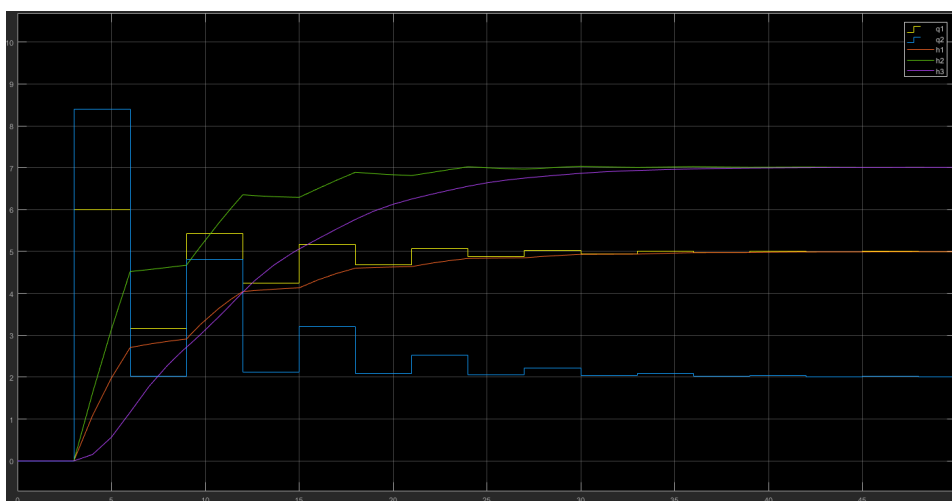


Figure 10: Resposta do Algoritmo de Velocidade Modificado com $T=3$

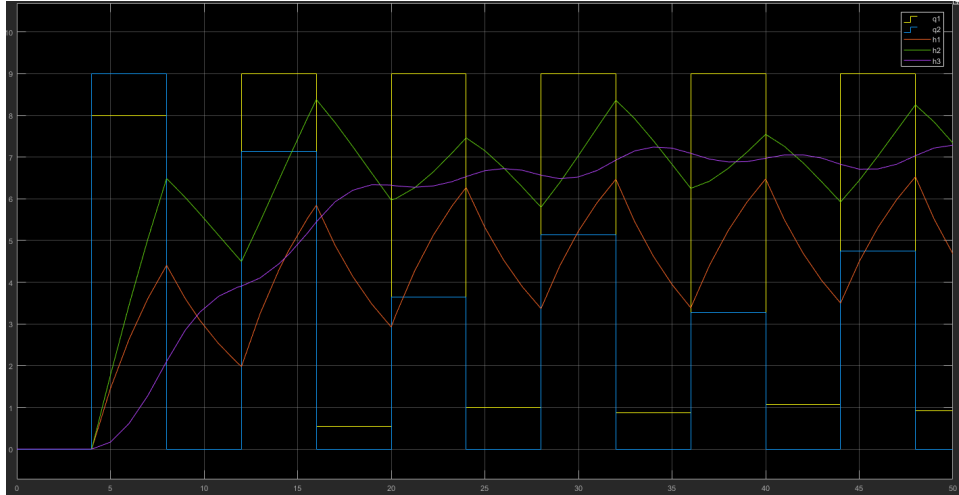


Figure 11: Resposta do Algoritmo de Velocidade Modificado com $T=4$

4 Conclusões

- A simulação ideal do PID contínuo foi melhorada pelo PID discreto, a correção dos saltos das entradas no sistema diminuem a possibilidade de estragos neste.
- Neste sistema os algoritmos de posição, velocidade e posição modificado apresentam respostas sem grandes diferenças umas em relação às outras.
- Neste sistema o algoritmo ideal é o de velocidade modificado pois resolve o problema das entradas q_1 e q_2 terem uma variação de caudal muito grande em muito pouco tempo e assim diminui riscos de segurança.
- O período de amostragem não pode ser 4 ou superior pois com esse T não há controlo e o sistema comporta-se melhor com T menor ou igual a 1.