VI Simpósio de Iniciação Científica 2023
Ciência como ferramenta de transformação da sociedade

PROGRAMAÇÃO E TESTE DE LEIS DE CONTROLE EM QUADRICÓPTERO DE 3 GDL

Guilherme de Paula Pinheiro[1] (IC), Marcelo Santiago de Sousa[1] (PQ), Leandro Diniz de Jesus[1] (PQ) e Yohan Alí Díaz Méndez[1] (PQ)
[1]*Universidade Federal de Itajubá (UNIFEI).*

Palavras-chave: Arduino, Controle Tolerante a Falhas, PID, Quadricóptero, SMC.

Introdução

Veículos Aéreos Não Tripulados (VANTs) têm atraído, constantemente, investimentos a nível mundial devido à crescente demanda dos mesmos e à enorme aplicabilidade que têm na solução de problemas em diversas áreas. Dentre as aplicações mais recentes e promissoras para o futuro encontra-se o transporte de mercadorias, entregas e inclusive mobilidade urbana de pessoas, fato que reduz a distância dessas dessas aeronaves com as pessoas aumentando na mesma proporção a probabilidade de acidentes e ferimentos.

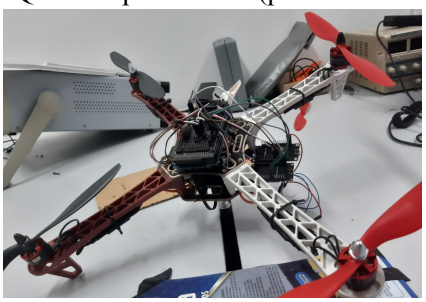
Falhas em VANTs, também conduzem a perdas financeiras irreparáveis. Mesmo quando estas falhas são "pequenas" levam a danos catastróficos, principalmente em drones do tipo quadrirotor onde a redundância de atuadores é mínima. Devido à complexidade destes dispositivos, uma falha pode ter origem em diferentes componentes, em atuadores, sensores, estrutura (colisões) ou inclusive a "delays" devido a problemas de comunicação ou processamento nas placas controladoras. Portanto, torna-se necessário que haja a constante melhoria tecnológica desses veículos de modo a aprimorar a sua eficiência, segurança e garantir a confiabilidade na presença de falhas. Leis de Controle Tolerantes a Falhas (FTC, Fault Tolerant Control) têm sido desenvolvidas nos últimos anos com o intuito de solucionar esses problemas e são atrativas devido ao fato de não precisarem de redundância de hardware. FTC passivas (PFTC) aproveitam a capacidade de controladores robustos e adaptativos para lidar com as falhas considerando as mesmas como perturbações do sistema, por outro lado, FTC ativas (AFTC), fazem uso de sistemas de Detecção e Identificação de Falhas (FDI, Fault Detection and Identification) com o intuito de identificar e classificar, em tempo hábil, o tipo de falha e sua magnitude com o intuito de reconfigurar o controlador, lidando dessa forma com falhas mais "graves" em relação a PFTC.

As plataformas de drones quadrirotores para programação e teste de leis de controle elaboradas como são as baseadas em FTC são cada vez mais escassas e as poucas disponíveis no mercado são extremamente custosas. Na Universidade Federal de Itajubá, uma plataforma de 3 GDL (graus de liberdade) de drone quadrirotor de baixo custo foi construída e adaptada para permitir a programação de qualquer lei de controle usando programação de baixo nível, tendo acesso direto a parâmetros do drone como ângulos, taxas e acelerações e a capacidade de gerar sinais PWM adequados para correção de manobras pré selecionadas e programadas (sinais de referência). No presente trabalho de Iniciação Científica (IC) realizou-se a programação de duas leis de controle nessa plataforma, o clássico PID (Proporcional Integral Derivativo) e uma lei de controle robusta, SMC (Sliding Mode Control) com o intuito de demonstrar a sua eficiência na presença de perturbações. As perturbações são como saturação do sinal de controle (imitando falha no atuador). Diversos testes foram realizados e os dados coletados e graficados para fins de comparação. Os resultados confirmam a utilidade da plataforma e revelam a capacidade de ambos controladores para lidarem com falhas. Muitos desafios foram encontrados, principalmente com a calibração dos sensores, a programação da placa e configuração dos parâmetros, assim como a programação de um sistema para sintonização dos ganhos dos controladores, em tempo real.

Metodologia

A plataforma utilizada para a implementação e teste dos controladores de voo PFTC, SMC e PID foi um quadricóptero F450 da DJI com motores, hélices,, controlador eletrônico de velocidade (ESCs), placa de desenvolvimento Arduino Mega 2560, o resto da eletrônica embarcada será descrita em breve. Os testes de voo consistem na injeção de uma falha em um dos motores com os controladores de atitude ligados. A falha é inserida saturando o sinal PWM para um valor máximo limitando o desempenho do motor. Os controladores de atitude realizam o controle de trajetória angular do drone, neste trabalho, a trajetória desejada é um doublet de 5° de ângulo de rolagem *(roll*), enquanto os ângulos de arfagem (*pitch*) e guinada (*yaw*) são mantidos nulos (0°). As leis de controle implementadas e testadas são, PID e SMC.

Figura 1 – Quadricóptero F450 (plataforma de teste).



O controlador PID possui a estrutura da equação a seguir, sendo k_p, k_i e k_d os ganhos proporcional, integral e derivativo respectivamente. o sinal de k_p pode ser escolhido de forma a fazer com que a saída do controlador aumente (ou diminua) à medida que o desvio (erro, *e*) aumenta. A ação de controle integral é muito usada porque ela apresenta uma importante característica prática, a eliminação do erro estacionário, já a ação de controle derivativa tem um caráter antecipatório, sendo sua função reagir antecipadamente ao comportamento futuro do sinal de erro com base na sua taxa de variação.

$$u = k_p.e + k_i \int e\,dt + k_d \frac{de}{dt}$$

O controlador SMC (*Sliding Mode Control*), trata-se de técnica de controle robusta que consiste em reduzir o problema de controle de um sistema genérico, descrito por equações não-lineares de ordem *n*, para um sistema de primeira ordem com incertezas nos parâmetros e/ou em sua própria estrutura matemática. A estrutura do SMC pode ser vista na equação a seguir, sendo *K* o ganho principal que limita a amplitude do sinal de controle, s é a superfície deslizante e k_1o ganho associado à velocidade com que a superfície deslizante converge para dentro da camada limite μ, a qual suaviza o sinal de controle reduzindo o efeito de chattering. Todos os ganhos foram sintonizados via tentativa-erro.

$$u = K.sat(s/\mu)$$
$$s = k_1 e + \frac{de}{dt}$$

O código implementado para o controle do drone utiliza a recepção dos sinais de um rádio controle o qual transfere os dados do ângulo desejado para uma placa controladora Arduino Mega 2560 que interpreta os dados recebidos calculando o erro entre o valor do ângulo medido por um sensor que coleta a aceleração e a velocidade angular sofrida no sensor nos eixos X,Y e Z, com isso se calcula o erro pela subtração entre o valor medido e o valor desejado para os ângulos Phi, Theta e Psi, dessa forma o valor do erro é uma das entradas para os controladores, juntamente dos parâmetros variáveis do controlador, dessa forma cada gera uma saída diferente que é direcionada para os ESCs os quais controlam a potência de cada motor.

Para a coleta dos dados dos ângulos foi utilizado um MPU6050 um sensor que é capaz de medir a aceleração e a velocidade angular sofrida nele, para transferir essas medidas para o arduino é necessário utilizar a comunicação I2C, onde o arduino solicita os dados e o sensor fornece-os de forma serial. Com os dados de aceleração e velocidade angular é necessário a conversão para o ângulo atual do drone onde utiliza-se um filtro de kalman para eliminação de ruídos e oscilações bruscas em ambas medições, além disso foi necessário converter os valores da aceleração nos eixos X,Y e Z para um valor de ângulo com a decomposição vetorial da aceleração porém apenas essa medida não é eficiente durante oscilações bruscas do drone, onde nesse caso é utilizado o somatório do giro em cada eixo para determinar o ângulo atual, esse valor se mostrou efetivo durante as oscilações mais bruscas do drone. Juntando os valores das acelerações em ambos os casos é possível determinar o ângulo do drone. Para a calibração destes sensores foram obtidos 2000 valores de rotação para cada eixo com o drone parado para que seja possível saber o erro e assim subtraí-lo do valor obtido pelo sensor, sendo a calibração feita antes de cada voo.

A fim de realizar os movimentos do drone são utilizados 4 motores brushless e 4 ESCs , para o controle dos motores é enviado um pulso PWM através do arduino, onde esse pulso é interpretado pelo ESC e convertido em potência para o motor. Cada ESC trabalha em um período de 4ms, onde o sinal interpretado por eles é um pulso que varia de largura, sendo o motor desligado com 1ms e ligado na potência máxima um pulso de largura de 2ms, qualquer valor entre estes reflete um uma faixa onde o motor está ligado porém não com velocidade máxima. Além disso os motores são distribuídos em X na estrutura do drone, onde para realização dos movimentos é necessário que para cada movimento todos os motores são utilizados, um exemplo é o movimento de roll para a direita, onde ambos motores do lado direito são desacelerados e os motores da esquerda acelerados, resultando na inclinação do drone. Os motores são acelerados e desacelerados com base na seguinte função,onde Uphi representa o ângulo de roll , Utheta o ângulo de pitch e Upsi o ângulo de yaw :

```
pwm[Prop::LF] = 115 + Uz + Uphi + Utheta + Upsi;

pwm[Prop::RF] = 115 + Uz - Uphi + Utheta - Upsi;
```

pwm[Prop::LB] = 115 + Uz + Uphi - Utheta - Upsi;

pwm[Prop::RB] = 115 + Uz - Uphi - Utheta + Upsi;

Os controladores foram implementados com o uso de objetos, com suas respectivas variáveis e funções, as variáveis do controlador PID foram os parâmetros do controlador "Kp", "Ki" e "Kd", os valores desejado e medido, respectivamente "x" e "xd", e a variação do tempo "dt". Já no controlador SMC foram utilizados os parâmetros k1,k2 e U, juntamente dos valores medido e desejado e da variação do tempo "x", "xd" e "dt". Para a implementação em C++ foi necessário substituir a integral por um somatório do erro*dt e a derivada por erro atual e a função de saturação foi implementada utilizando o condicional :

$$i = Ki.\sum(e.dt)$$
$$d = \frac{erro\ atual - erro\ anterior}{dt}$$

No código a funcionalidade de dublê foi implementada dentro do loop principal, onde a todo momento era verificado se o canal 6 foi acionado, e em caso positivo ele conta o código desde o acionamento do dublê e realiza os movimentos desejados. Para a implementação da falha durante o dublê era verificado a cada loop se o *stick* do controle teve alguma variação e em caso positivo a limitação do PWM é inserida utilizando um saturador, limitando os valores máximos para o PWM do motor RF. Já para o recebimentos dos dados foi utilizado um ESP 32 o qual recebe os dados do arduino via comunicação UART e transmite para a nuvem via protocolo MQTT, onde os dados foram coletados com o auxílio de um código implementado em python, e para dados de calibração foi utilizado a comunicação serial via cabo. A modificação dos parâmetros dos controladores foi feita via página WEB por uma softAP gerada pelo ESP.

Resultados e discussão

Controlador por modos deslizantes (SMC)

Os ganhos do controlador SMC para *pitch* e *roll* são: $K = 138,38$; $k_1 = 2,15$; $\mu = 27$, para *yaw* são: $k = 120,66$; $k_1 = 2,3$; $\mu = 16$.

Na Figura 2 observa-se o comportamento do controlador SMC no caso sem falha e na Fig. 3 o sinal de controle correspondente. A dificuldade do controlador em atingir ângulo de rolagem -5° deve-se à perturbação causada pelo cabo de dados pendurado na lateral. Verifica-se que os controladores de *pitch* e *yaw* são eficientes.

Figura 2 – SMC sem falha.

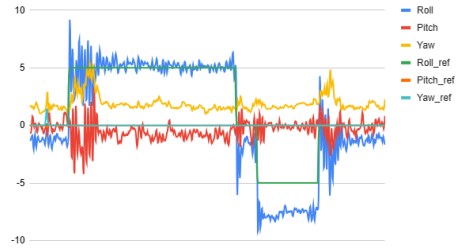


Figura 3 – Sinal de controle PWM (caso: sem falha)

Nas Figuras 4 e 5 verifica-se a impossibilidade do controlador em manter a estabilidade do drone para uma falha do motor direito dianteiro (RF) com PWM máximo de 1300. Nas Figuras 6 e 7 relaxou-se à falha, permitindo sinal PWM no motor RF de até 1325, verifica-se que o controlador tenta eventualmente fazer o drone seguir a referência de atitude. Finalmente na Figura 8 com falha e até PWM=1350 no motor, o drone recupera a capacidade de estabilizar os ângulos do drone.

Figura 4 – SMC com falha de motor PWM de 1300.

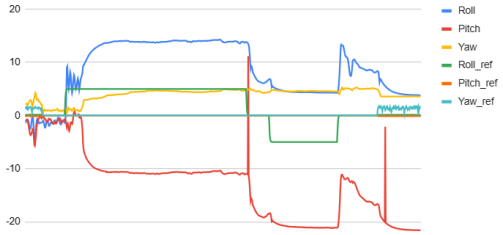


Figura 5 – Sinal de controle (caso: falha PWM=1300).

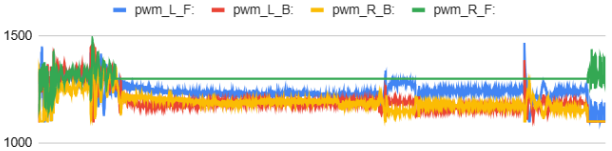


Figura 6 – SMC com falha de motor PWM de 1325.

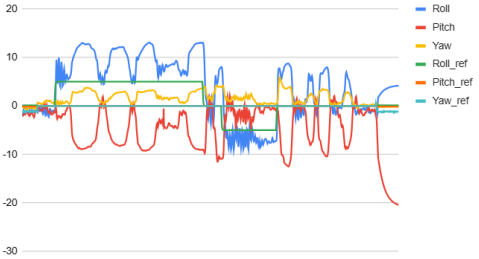
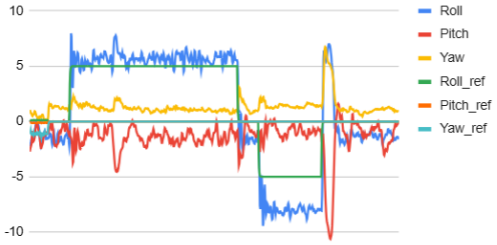


Figura 7 – Sinal de controle (caso: falha PWM=1325).

Figura 8 – SMC com falha de motor de PWM de 1350.



Controlador Proporcional, Integral, Derivativo (PID)

Os ganhos do controlador PID para *pitch* e *roll* são: $k_p = 6,056$; $k_i = 0,393$; $k_d = 5,417$, para *yaw* são: $k_p = 5,597$; $k_i = 0,087$; $k_d = 2,234$.

Na Figura 9 ilustra-se a resposta do quadricóptero de 3 gdl no caso sem falha mostrando a capacidade do controlador PID em fazer o rastreamento das trajetórias mesmo com certo *overshoot*. Na figura 10 percebe-se que com falha do motor RF com saturação PWM=1300 (Fig. 11), o controlador tenta rastrear a referência embora com muita dificuldade, o mesmo é observado com falha PWM=1325. Já no caso de falha PWM=1350 (Fig. 12) o controlador PID recupera a capacidade de lidar com a perturbação causada pela falha e consegue rastrear satisfatoriamente a trajetória desejada.

Figura 9 – SMC sem falha.

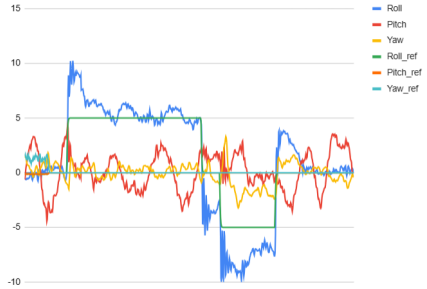


Figura 10 – SMC com falha de motor, PWM de 1300.

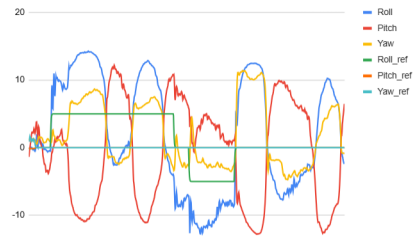


Figura 11 – SMC com falha de motorde PWM de 1325.

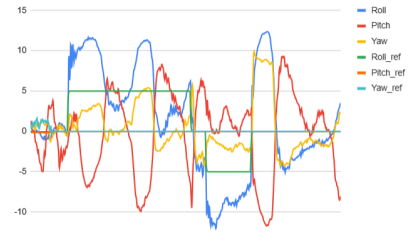
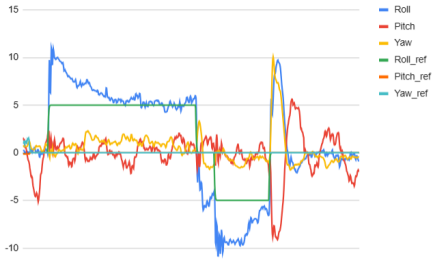


Figura 12 – SMC com falha de motor PWM de 1350.



Conclusões

No presente trabalho foram programadas as leis de controle PID e SMC em uma placa de desenvolvimento Arduino para o controle de um drone quadricóptero de 3 gdl. Códigos e rotinas de programação para a leitura dos sensores, implementação das leis de controle, injeção da falha e armazenamento dos resultados foram desenvolvidos. Testes experimentais demonstraram a eficiência de ambos os controladores para realizar o rastreamento dos sinais de referência dos ângulos de atitude. Em ambos os casos, demonstrou-se que pelo menos uma redução de até 200 no sinal PWM de um dos motores é contornado pelas leis de controle tolerantes a falhas passivas. Percebeu-se uma ligeira superioridade do controlador PID em relação ao SMC (notado no caso de falha PWM=1300). Testes futuros serão realizados para sintonizar de uma melhor maneira os ganhos de ambos os controladores.

Agradecimentos

O presente trabalho foi realizado com apoio da Fundação de Amparo à Pesquisa de Minas Gerais (**FAPEMIG**) por meio do Programa PIBIC da UNIFEI e do projeto APQ-01656-2021 sob a coordenação do Prof. Yohan Díaz. Agradecemos também à DPI da UNIFEI e ao Grupo de Pesquisa VisCap pelo acesso às instalações.

Referências

DIAZ-MENDEZ, Yohan *et al*. Conditional integrator sliding mode control of an unmanned quadrotor helicopter. **Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering**, v. 236, n. 3, p. 458-472, 2022.

OGATA, Katsuhiko. **Modern control engineering fifth edition**. 2010.

SLOTINE, Jean-Jacques E. et al. **Applied nonlinear control**. Englewood Cliffs, NJ: Prentice hall, 1991.