

**UNIVERSIDADE DE CAXIAS DO SUL
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO INDUSTRIAL**

MATHEOS HAHN BIRCK

CONTROLE DE NAVEGAÇÃO DE DRONE ATRAVÉS DE ESPAÇO DE ESTADOS

CAXIAS DO SUL

2016

MATHEOS HAHN BIRCK

CONTROLE DE NAVEGAÇÃO DE DRONE ATRAVÉS DE ESPAÇO DE ESTADOS

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia de Controle e Automação de Caxias do Sul, como requisito parcial à obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientador Prof. Dr. Julio Cesar Ceballos Aya

CAXIAS DO SUL

2013

MATHEOS HAHN BIRCK

CONTROLE DE NAVEGAÇÃO DE DRONE ATRAVÉS DE ESPAÇO DE ESTADOS

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia de Controle e Automação da Universidade de Caxias do Sul, como requisito parcial à obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Aprovado em

Banca Examinadora

Prof. Dr. Julio Cesar Ceballos Aya
Universidade de Caxias do Sul – UCS

Prof. Dr. Guilherme Holsbach Costa
Universidade de Caxias do Sul – UCS

Prof. Ma.. Patricia Giacomelli
Universidade de Caxias do Sul – UCS

Dedico este trabalho a minha mãe, a minha namorada, a familiares e amigos que sempre me apoiaram e incentivaram a não desistir de meus sonhos e ideais.

AGRADECIMENTOS

Agradeço ao comitê de professores do curso da Engenharia Elétrica, Engenharia Mecânica e da Engenharia de Materiais e, em especial aos professores do curso de Engenharia de Controle e Automação que sempre estiveram dispostos e disponíveis a ajudar quando necessário. Agradeço também a minha mãe que sempre me auxiliou tanto psicologicamente quanto financeiramente, e que sempre me apoiou em minhas decisões e atos. À minha namorada pela compreensão dos muitos finais de semana usados para estudos ou quando eu me sentia exausto pela árdua rotina semanal. Aos meus amigos também agradeço pela compreensão sobre a falta de tempo para com eles, e pelos momentos em que me fizeram desviar o pensamento de toda a confusão e euforia nos finais dos semestres com conversas e churrascos no curto tempo disponível. Agradeço ao pessoal do Instituto de Materiais Cerâmicos da UCS, que sempre foi atencioso e preocupado com meu bem-estar e meu desenvolvimento pessoal. E agradeço também a todos que de algum modo contribuíram em minha formação acadêmica, ascensão profissional ou que em meu período de intercâmbio me mantiveram forte e fiel a meus princípios.

*“A mente que se abre a uma nova ideia jamais
voltará ao tamanho original. – Albert
Einstein”*

RESUMO

O presente trabalho de conclusão de curso aborda o tema veículos aéreos não tripulados, particularizando o quadricóptero(drone) como alvo de estudo e visa implementar um controle de trajetória e de estabilização em espaço de estados utilizando de equações obtidas a partir dos princípios de Newton-Euler. O objetivo do trabalho consiste em definir um controle utilizando o modelo matemático simplificado de um quadricóptero afim de obter a estabilização do quadricóptero. Para tal fim, uma função custo visando a otimização do consumo de energia foi implementada. O controle de estabilização, assim como as trajetórias pré definidas, foi realizado através de simulações, nas quais ambas são relativas ao modelo linearizado. Os resultados apresentados demonstram que a técnica aplicada é promissora e funcional, uma vez que foi obtida a estabilidade em regime permanente.

Palavra-chaves: Quadricoptero, espaço de estados, drone, modelo.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1- Representação de giro..... | 11 |
| Figura 2 - Movimento roll..... | 12 |
| Figura 3 - Movimento yaw..... | 13 |
| Figura 4 - parado em equilíbrio (pairando)..... | 13 |
| Figura 5 - Movimento de descida..... | 14 |
| Figura 6 - Comandos roll, pitch e yaw..... | 14 |
| Figura 7 - Representação cinemática do quadricóptero..... | 16 |
| Figura 8 - Sistemas de coordenadas inerciais..... | 17 |
| Figura 9 - Representação em espaço de estados..... | 22 |
| Figura 10 - Sistema realimentado com ganho K..... | 23 |
| Figura 11 - Sistema de controle retroação do observador de ordem completa..... | 25 |
| Figura 12 - Sistema em malha aberta..... | 28 |
| Figura 13 - Sistema com realimentação de K..... | 30 |
| Figura 14- Sistema com observador de ordem completa..... | 31 |
| Figura 15 - Resposta do sistema em malha aberta..... | 32 |
| Figura 16 - K do sistema ao utilizar lqr..... | 32 |
| Figura 17 - Polos do sistema em malha fechada com ganho k..... | 33 |
| Figura 18 - Zoom do mapa dos polos da Figura 17..... | 33 |
| Figura 19 - Polos do sistema em malha fecha com ganho K..... | 34 |
| Figura 20 - Análise do controle com ganho K..... | 34 |
| Figura 21 - K do observador de ordem completa..... | 35 |
| Figura 22- Saída do sistema com observador de ordem plena..... | 35 |
| Figura 23 - Trajetória circular e subida..... | 36 |
| Figura 24 - Trajetória Elipsoidal e subida..... | 37 |
| Figura 25 - deslocamento vertical e em equilíbrio..... | 38 |
| Figura 26 - Erro relativo a trajetória 1 (círculo e subida)..... | 38 |
| Figura 27 - Erro relativo a trajetória 2 (elipse e subida)..... | 39 |
| Figura 28 - - Erro relativo a trajetória 3 (pairar)..... | 40 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------|--|
| ABNT | Associação Brasileira de Normas Técnicas |
| LQR | Linear-Quadrático Regulador |
| MIMO | Multiple Input Multiple Output |
| TCC | Trabalho de Conclusão do Curso |
| SISO | Single Input Single Output |
| UCS | Universidade de Caxias do Sul |

SUMÁRIO

| | | |
|--------------|---|-----------|
| 1 | INTRODUÇÃO | 10 |
| 1.1 | JUSTIFICATIVA DO TABALHO | 10 |
| 1.2 | OBJETIVOS | 10 |
| 1.2.1 | Objetivos geral:..... | 10 |
| 1.2.2 | Objetivos específicos:..... | 10 |
| 2 | EMBASAMENTO TEÓRICO | 11 |
| 2.1 | GRAUS DE LIBERDADE..... | 11 |
| 2.2 | MODELO MATEMÁTICO | 15 |
| 2.3 | CONTROLE ATRAVÉS DO ESPAÇO DE ESTADOS | 22 |
| 2.4 | MATRIZ DE GANHO K POR ALOCAÇÃO DE PÓLOS..... | 23 |
| 2.5 | OBSERVADOR DE ORDEM PLENA..... | 24 |
| 2.6 | CONTROLE ÓTIMO – <i>LINEAR QUADRATIC REGULATION</i> | 25 |
| 2.6.1 | Regra de Bryson..... | 26 |
| 3 | IMPLEMENTAÇÃO | 28 |
| 4 | RESULTADOS | 32 |
| 5 | CONSIDERAÇÕES FINAIS..... | 41 |
| | REFERÊNCIAS | 42 |
| | ANEXO A – Programa matlab | 43 |

1 INTRODUÇÃO

Um quadricóptero é um helicóptero de quatro motores onde os dois motores giram em um sentido e os outros dois no sentido oposto, sendo que são dispostos de modo intercalado no formato de um “X” onde os quatro ângulos internos do “X” são de 90° e os motores são posicionados nas extremidades do mesmo. O centro de massa do quadricóptero é localizado no centro do veículo e as forças geradas pelas hélices são perpendiculares ao plano dos motores. A principal vantagem de um quadricóptero em relação ao helicóptero é a simetria na construção mecânica, onde devido à disposição dos motores é possível obter uma melhor estabilização.

O intuito deste Trabalho de Conclusão de Curso é projetar um controle para modelo matemático linearizado de um drone, utilizando o método de espaço de estados.

1.1 JUSTIFICATIVA DO TABALHO

O presente trabalho possui basicamente dois motivos para sua realização. Sendo o primeiro desenvolver melhor a parte de controle e de programação e o segundo, a oportunidade de trabalhar com um sistema não-linear, que não foi trabalhada durante o curso. Também é fonte de inspiração saber que o controle aplicado pode ser útil em outras aplicações similares.

1.2 OBJETIVOS

1.2.1 Objetivos geral:

Controlar a trajetória de um quadricóptero e observar os resultados através de simulações.

1.2.2 Objetivos específicos:

A partir do objetivo geral, foram definidos alguns objetivos específicos, como:

- a) Rever o método de controle em espaço de estados;
- b) Verificar a observabilidade e a controlabilidade;
- c) Projetar o controlador;

2 EMBASAMENTO TEÓRICO

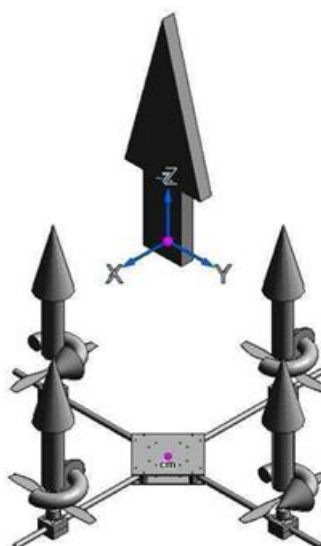
Nesta seção serão descritos sucintamente os conceitos conhecidos que são de vital importância na realização do projeto.

2.1 GRAUS DE LIBERDADE

Os quadricópteros são caracterizados por possuírem seis graus de liberdade. Três dos graus são bem conhecidos, são os deslocamentos lineares sobre os eixos x , y e z . Os outros três graus de liberdade são: *roll*, *pitch* e *yaw*, traduzidos para o português como rolagem, arfagem e guinada, respectivamente. Para entender melhor as definições, A. Sreekumar, P. Hithsan, M.K. Anand (2011) argumenta que quadricóptero utiliza quatro motores e considera que o mesmo deve ser um corpo rígido. As figuras a seguir representarão os movimentos básicos de um quadricóptero, nos quais o impulso é proporcional à espessura das setas.

Para se manter em equilíbrio em relação ao eixo central dois dos motores giram no sentido horário, enquanto os outros dois giram no sentido anti-horário. Para manter o quadricóptero a uma altura fixa as hélices devem gerar uma força de empuxo igual à força peso do quadricóptero. A representação é mostrada na Figura 1.

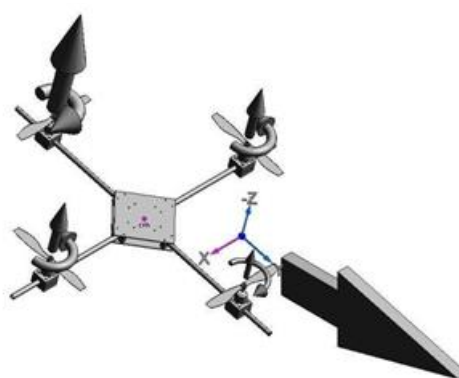
Figura 1- Representação de giro



Fonte: 1 - A. Sreekumar, P. Hithsan, M.K. Anand (2011)

A rolagem (*roll*) é obtida aumentando (ou diminuindo) a velocidade de um motor e ao mesmo tempo diminuindo (ou aumentando) a velocidade do motor oposto. Os outros dois motores continuam com suas velocidades constantes para manter a altura que estava, ou e aumentam/diminuem a velocidade (sincronizados a uma velocidade de mesmo valor), para elevar ou baixar, respectivamente, o quadricóptero. Isso causa um torque em relação ao eixo central, fazendo com que o quadricóptero se incline na direção da hélice que está com menor velocidade, como mostrado na Figura 2.

Figura 2 - Movimento roll

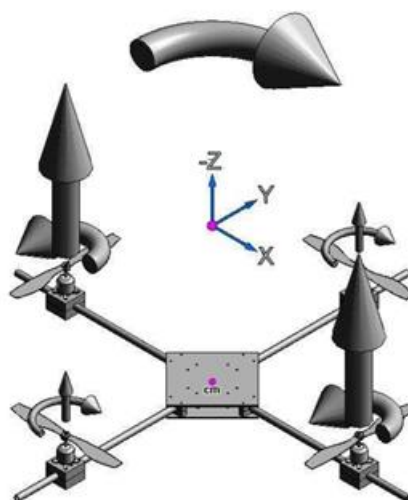


Fonte: 2 - A. Sreekumar, P. Hithsan, M.K. Anand (2011)

O movimento de arfagem (*pitch*) assim como o de rolagem, é um movimento na horizontal. A diferença entre eles é que o movimento de arfagem é definido como movimento para frente e para trás e o de rolamento como movimento para os lados (direita e esquerda).

O movimento de guinada (*yaw*) é dado através do aumento (ou diminuição) de velocidade de dois dos motores que giram para o mesmo lado enquanto os outros dois, ao mesmo tempo, diminuem (ou aumentam) a velocidade. Isso leva a um torque em volta do eixo z e faz com que rotacione em relação a este eixo. O sentido de giro será o mesmo dos motores com menor velocidade, como é mostrado na Figura 3.

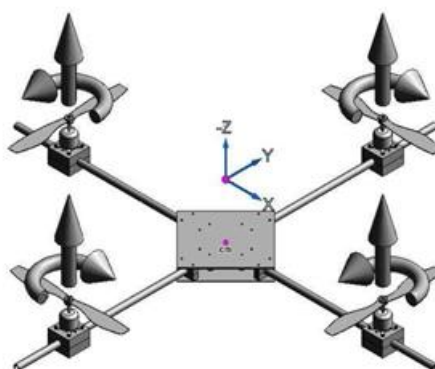
Figura 3 - Movimento yaw



Fonte: 3 - A. Sreekumar, P. Hithsan, M.K. Anand (2011)

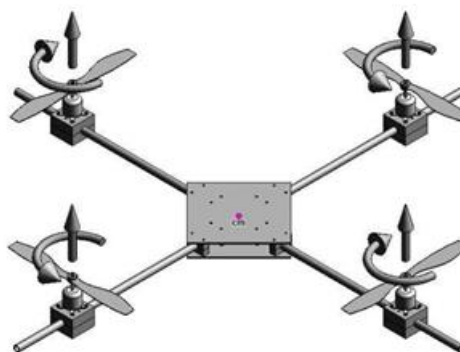
O movimento de aceleração (subida e descida) é realizado através do aumento (ou diminuição) da velocidade de todos os motores juntos e sincronizados. Para subir deve-se aumentar a velocidade dos motores, para descer deve-se reduzir e para pairar a velocidade dos motores devem gerar uma força igual à força peso do quadricóptero. A representação dos movimentos de flutuação e de descida são demonstrados na Figura 4 e Figura 5 respectivamente.

Figura 4 - parado em equilíbrio (pairando)



Fonte: 4 - A. Sreekumar, P. Hithsan, M.K. Anand (2011)

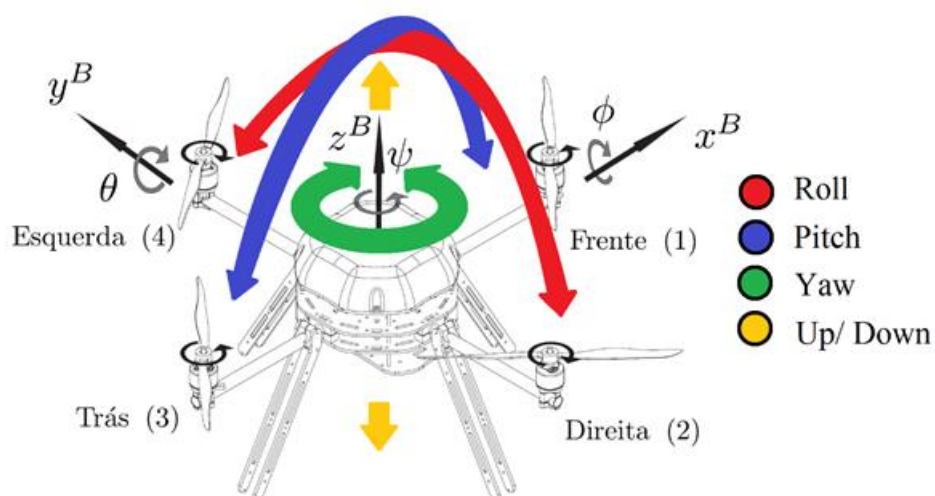
Figura 5 - Movimento de descida



Fonte: 5 - A. Sreekumar, P. Hithsan, M.K. Anand (2011)

Nas figuras anteriores é representada a diferença de torque que os motores devem fornecer a fim de executar os movimentos descritos individualmente. Uma visão mais clara do que é cada movimento é apresentada na Figura 6.

Figura 6 - Comandos roll, pitch e yaw



Fonte: 6 - Henrique Ribeiro Delgado da Silva (2012), modificado.

Como é possível observar na Figura 6 *roll* é o movimento rotacional em torno do eixo x , *pitch* o movimento de rotação sobre o eixo y e *yaw* é a rotação dada em torno de z .

2.2 MODELO MATEMÁTICO

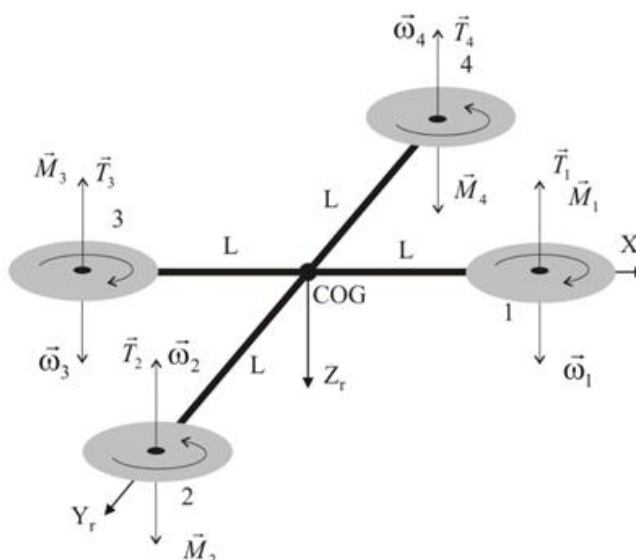
Conforme Andrew Gibiansky (2014) o quadricóptero possui seis graus de liberdade (três de translação e três de rotação) e apenas quatro entradas, sendo estas independentes (motores). Sabendo disso é possível deduzir que a sua dinâmica se trata de um problema não-linear, especialmente devido aos efeitos aerodinâmicos. Ainda, conforme Andrew Gibiansky (2014), outro problema de alto impacto é o fato de, diferentemente de veículos terrestres, os veículos aéreos praticamente não são afetados por forças de atrito, que poderiam suavizar os seus movimentos. Assim o próprio quadricóptero deve propiciar o amortecimento necessário a fim de parar o movimento e permanecer estável.

A modelagem foi feita a partir das equações de Newton, e para simplificar o modelo matemático algumas características podem ser consideradas de acordo com Silva(2012). São elas:

- a) A origem do referencial do veículo é coincidente com o seu centro de massa;
- b) O quadricóptero é simétrico ao longo dos eixos horizontais X_r e Y_r ;
- c) O quadricóptero é um corpo rígido;
- d) A inclinação magnética, a vibração dos motores, a turbulência causada pela movimentação de ar junto ao solo e as não linearidades da bateria são desprezadas;
- e) Não existe escorregamento entre as hélices e o rotor do motor;
- f) Os quatro motores são iguais;

De acordo com AiMT (2010) a representação da cinemática do quadricóptero é dada de acordo com a Figura 7, onde também é representada a origem do centro de coordenadas X, Y e Z pertencentes ao sistema de coordenadas referente ao quadricóptero, sendo que a origem é posicionada no seu centro de gravidade. Também está representada a direção das velocidades angulares ω_i , a direção positiva do torque M_i e a direção positiva da propulsão T_i .

Figura 7 - Representação cinemática do quadricóptero



Fonte: 7- AiMT(2010)

Das equações de Newton define-se que um corpo estará em equilíbrio estático se (1) e (2) forem verdadeiras. Onde significa que o somatório das forças e dos momentos existentes, respectivamente, devem ser zero. E, $m\vec{V}$ e \vec{H} representam o somatório dos momentos e dos impulsos, respectivamente, em relação ao referencial inercial.

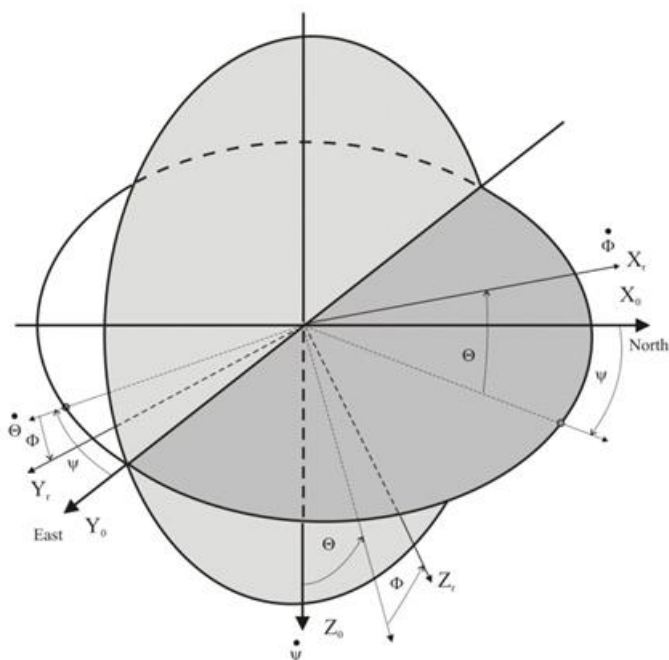
$$\sum \vec{F} = \frac{d(m\vec{V})}{dt} = 0 \quad (1)$$

$$\sum \vec{M} = \frac{d(\vec{H})}{dt} = 0 \quad (2)$$

Conforme Silva(2012), quando se lida com veículos aéreos é necessário descrever, não somente sua posição, mas também sua altitude. Sendo assim, AiMT (2010) é representada um referencial inercial como mostrado na Figura 8, que representa o sistema de coordenadas fixo a terra. As coordenadas de posição do quadricóptero serão em relação a este referencial.

As coordenadas de posição do quadricóptero serão dadas em relação a este referencial, utilizando a matriz de rotação (3) calculada. Esta é a matriz que fornece a relação entre o referencial inercial e o referencial do quadricóptero.

Figura 8 - Sistemas de coordenadas inerciais



Fonte: 8 - AiMT(2010)

$$R_r = R(\Psi)R(\Theta)R(\Phi) = \begin{bmatrix} \cos \Psi \cos \Theta & \cos \Psi \sin \Theta \sin \Phi - \sin \Psi \cos \Phi & \cos \Psi \sin \Theta \cos \Phi + \sin \Psi \sin \Phi \\ \sin \Psi \cos \Theta & \sin \Psi \sin \Theta \sin \Phi + \cos \Psi \cos \Phi & \sin \Psi \sin \Theta \cos \Phi - \cos \Psi \sin \Phi \\ -\sin \Theta & \cos \Theta \sin \Phi & \cos \Theta \cos \Phi \end{bmatrix} \quad (3),$$

onde Φ representa o ângulo de rolagem (*roll*), Θ o ângulo de arfagem (*pitch*) e Ψ o ângulo de guinada (*yaw*).

Segundo Blacklock (1991) a equação de somatório de forças, pode ser reescrita conforme o conjunto de equações (4):

$$\begin{aligned} \sum F_{xr} &= m(v_{xr} + v_{zr}\omega_{yr} - v_{yr}\omega_{zr}) \\ \sum F_{yr} &= m(v_{yr} + v_{xr}\omega_{zr} - v_{zr}\omega_{xr}) \\ \sum F_{zr} &= m(v_{zr} + v_{yr}\omega_{xr} - v_{xr}\omega_{yr}) \end{aligned} \quad (4)$$

Sendo que v representa a velocidade linear e ω a velocidade angular no respectivo eixo. O lado esquerdo de (4), representam a soma de todos os componentes de forças que atuam

no quadricóptero. E a equação de somatório de momentos pode ser escrita de acordo com o conjunto de equações (5), na qual ω representa a velocidade angular no respectivo eixo e J o momento de inércia e, devido a simetria, $J_y = J_x$:

$$\begin{aligned}\sum M_{xr} &= J_x \dot{\omega}_{xr} + \omega_{yr} \omega_{zr} (J_z - J_y) \\ \sum M_{yr} &= J_y \dot{\omega}_{yr} + \omega_{zr} \omega_{xr} (J_x - J_z) \\ \sum M_{zr} &= J_z \dot{\omega}_{zr} + \omega_{xr} \omega_{yr} (J_y - J_x)\end{aligned}\tag{5}$$

Para expressar o movimento de um quadricóptero no referencial inercial $(x, y, z)_o$, visto na Figura 8, é utilizada a trigonometria realizando a projeção da velocidade angular em relação aos ângulos *roll*, *pitch* e *yaw*. Resultando no conjunto de equações (6):

$$\begin{aligned}\dot{\Theta} &= \omega_{yr} \cos \Phi - \omega_{zr} \sin \Phi \\ \dot{\Phi} &= \omega_{zr} + \omega_{yr} \sin \Phi \tan \Theta + \omega_{zr} \cos \Phi \tan \Theta \\ \dot{\Psi} &= \omega_{yr} \left(\frac{\sin \Phi}{\cos \Theta} \right) + \omega_{zr} \left(\frac{\cos \Phi}{\cos \Theta} \right)\end{aligned}\tag{6}$$

Inicialmente foram consideradas as forças que atuam sobre o quadricóptero supondo que o mesmo encontra-se pairando. Onde o conjunto de equações (7), T_i representa o impulso dos motores, g a gravidade e m a massa do quadricóptero.

$$\begin{aligned}\sum F_{xr} &= -mg \sin \Theta \\ \sum F_{yr} &= mg \cos \Theta \sin \Phi \\ \sum F_{zr} &= -T_1 - T_2 - T_3 - T_4 + mg \cos \Theta \cos \Phi\end{aligned}\tag{7}$$

O principal motivo para a existência de momentos no quadricóptero se deve aos motores. A seguir é representado o cálculo do somatório de momentos, onde L é o comprimento

do braço do quadricóptero, como mostrado na Figura 7, ω_i é a velocidade angular dos motores e J_{mp} o momento de inércia do motor.

O conjunto de equações (8) será bem representada no referencial quadricóptero desde que a distribuição de massa no corpo seja conhecida e que a massa do quadricóptero é simetricamente distribuída em relação aos planos $(x, y)_r$, $(x, z)_r$ e $(y, z)_r$ observados na Figura 7. Assim, os momentos de inércia podem ser desprezados, pois a distribuição de forma simétrica fará com que os seus valores se anulem, e a equação pode ser descrita como foi apresentado em (5).

$$\begin{aligned}\sum M_{xr} &= -LT_2 + LT_4 + J_{mp}\omega_{yr}(\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ \sum M_{yr} &= LT_1 - LT_3 + J_{mp}\omega_{xr}(-\omega_1 + \omega_2 - \omega_3 + \omega_4)\end{aligned}\quad (8)$$

$$\sum M_{zr} = k_{MT}(-T_1 + T_2 - T_3 + T_4)$$

Os momentos gerados pelas hélices são considerados proporcionais ao impulso que as mesmas fornecem. Assim, k_{MT} representa a razão entre o momento e impulso das hélices utilizadas. Portanto assume-se que as velocidades angulares das hélices são proporcionais ao impulso das mesmas. Logo tem-se (9).

$$\omega_i = k_{\omega T} T_i \quad (9)$$

De acordo com AiMT (2010) as equações apresentadas até o momento após equacionadas podem ser escritas como variáveis de estado. Após o equacionamento resolvido as equações são dadas de acordo com o conjunto de equações (10) :

$$\begin{aligned}\dot{x} &= \cos \Psi \cos \Theta v_{xr} + (\cos \Psi \sin \Theta \sin \Phi - \sin \Psi \cos \Phi)v_{yr} + (\cos \Psi \sin \Theta \cos \Phi \\ &\quad + \sin \Psi \sin \Phi)v_{zr}\end{aligned}$$

$$\begin{aligned}\dot{y} &= \sin \Psi \cos \Theta v_{xr} + (\sin \Psi \sin \Theta \sin \Phi + \cos \Phi \cos \Psi)v_{yr} + (\sin \Psi \sin \Theta \cos \Phi \\ &\quad - \cos \Psi \sin \Phi)v_{zr}\end{aligned}$$

$$\dot{z} = -\sin \Theta v_{xr} + \cos \Theta \sin \Phi v_{yr} + \cos \Theta \cos \Phi v_{zr}$$

$$\dot{\Theta} = \omega_{yr} \cos \Phi - \omega_{zr} \cos \Phi$$

$$\dot{\Phi} = \omega_{zr} + \omega_{yr} \sin \Phi \tan \Theta + \omega_{zr} \cos \Phi \tan \Theta$$

$$\dot{\Psi} = \omega_{yr} \left(\frac{\sin \Phi}{\cos \Theta} \right) + \omega_{zr} \left(\frac{\cos \Phi}{\cos \Theta} \right)$$

$$\dot{v}_{xr} = -v_{zr} \omega_{yr} + v_{yr} \omega_{zr} - g \sin \Theta \quad (10)$$

$$\dot{v}_{yr} = -v_{xr} \omega_{zr} + v_{zr} \omega_{xr} + g \cos \Theta \sin \Phi$$

$$\dot{v}_{zr} = -v_{yr} \omega_{xr} + v_{xr} \omega_{yr} + g \cos \Theta \cos \Phi - \frac{T}{m}$$

$$\dot{\omega}_{xr} = \frac{1}{J_x} \left[-\omega_{yr} \omega_{zr} (J_z - J_y) + M_x - \frac{k_{\omega T}}{k_{MT}} J_{mp} \omega_{yr} M_z \right]$$

$$\dot{\omega}_{yr} = \frac{1}{J_y} \left[-\omega_{zr} \omega_{xr} (J_x - J_z) + M_y - \frac{k_{\omega T}}{k_{MT}} J_{mp} \omega_{xr} M_z \right]$$

$$\dot{\omega}_{zr} = \frac{M_z}{J_z}$$

Sendo que (10) é um sistema não-linear, no qual $[x \ y \ z]^T$ representam a posição do quadricóptero no sistema de coordenadas inercial.

Para linearização das equações de estado de (10), segundo Franklin, G.F, Powell, J.D. e EEmami-Naeini, A. é possível assumir ângulos de valor muito pequenos, para que possa ser feito as considerações (11) e (12).

$$\sin \Theta = \sin \Phi = \sin \Psi = 0 \quad (11)$$

$$\cos \Theta = \cos \Phi = \cos \Psi = 1 \quad (12)$$

Observando a relação $\dot{X} = AX + Bu$, o vetor de estados X e o vetor de entradas u podem ser representados, respectivamente, por (13) e (14):

$$X = [x \quad y \quad z \quad \theta \quad \Phi \quad \Psi \quad v_{xr} \quad v_{yr} \quad v_{zr} \quad \omega_{xr} \quad \omega_{yr} \quad \omega_{zr}]^T \quad (13)$$

$$u = [\Delta T \quad \Delta M_x \quad \Delta M_y \quad \Delta M_z] \quad (14)$$

Considerando (13) e (14) as matrizes A e B serão dadas, respectivamente, por (15) e (16), onde a matriz A é dada conforme a ordem do vetor de estado X .

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (15)$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{J_x} & 0 & 0 \\ 0 & 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & 0 & \frac{1}{J_z} \end{bmatrix} \quad (16)$$

2.3 CONTROLE ATRAVÉS DO ESPAÇO DE ESTADOS

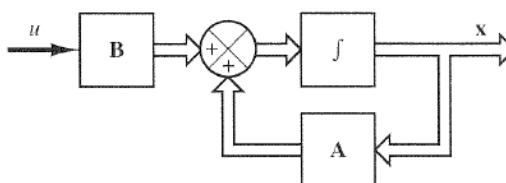
Atualmente é considerado que um sistema pode ter múltiplas entradas e múltiplas saídas, sendo que elas podem estar relacionadas entre si. Esses sistemas são mais complexos que sistemas habituais onde um controle é realizado baseado na relação de uma entrada e uma saída, ou função de transferência. Quando se trata de sistemas mais elaborados é necessário reduzir a complexidade das expressões matemáticas, e também utilizar de computadores para o processamento de equações de ordem mais elevada. Tendo em vista essas considerações, uma abordagem no espaço de estados é a mais indicada.

Este tipo de controle é baseado em um sistema de equações em termos de n equações diferenciais, que podem ser combinadas em uma equação diferencial vetorial-matricial. O intuito de usar a notação matricial é de simplificar a representação matemática do sistema de equações. Para sistemas do tipo SISO pode não ser notável a diferença entre trabalhar com espaço de estados ou função de transferência. Mas quando se trata de sistemas MIMO, a análise de sistemas mais elaborados, pode ser comparada as etapas de um processo de equações diferenciais simples. Conclui-se que ao trabalhar em espaço de estados a diferença de complexidade entre sistemas tipo SISO e tipo MIMO é reduzida.

Para trabalhar em espaço de estados é necessário representar o sistema na forma matricial. Por exemplo quando se tem o sistema em forma de função de transferência, segundo Ogata (2010) é possível, através das matrizes canônicas transformar as funções de transferência para espaço de estados e assim realizar operações na forma matricial. Mas o caminho para encontrar as matrizes, não necessariamente tem que ser como o descrito, existem outras maneiras. No contexto desse trabalho, a forma matricial como mostrado no tópico anterior, é proveniente das equações de estado encontradas.

Na Figura 9 é apresentado a representação do sistema em espaço de estados.

Figura 9 - Representação em espaço de estados



Fonte: 9 - Ogata (2010)

Outro t3pico importante quando o assunto 3 controle de um sistema, se refere a controlabilidade do mesmo. Segundo Ogata(2010) o conceito de controlabilidade para o espa3o de estados 3 dado atrav3s de (17), e 3 importante pois caso o sistema seja totalmente control3vel, 3 dito que pode-se alocar os polos arbitrariamente.

$$Cx = [B \quad : \quad A.B \quad : \dots : \quad A^{n-1}.B] \quad (17)$$

Para que o sistema seja considerado totalmente control3vel, o posto da matriz de controlabilidade Cx deve ser igual a dimens3o da matriz A .

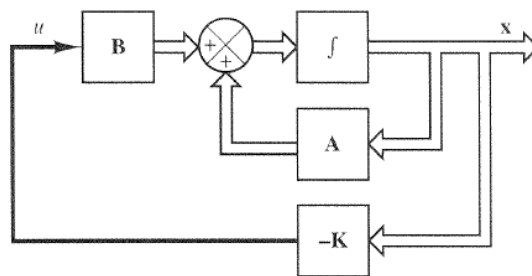
T3o importante quanto a controlabilidade, 3 a observabilidade, que permite estimar estados aos quais n3o se tem acesso. Para que um sistema seja observ3vel o posto da matriz Ox deve ser igual a ordem n da Matriz A , onde Ox 3 dado por (18).

$$Ox = \begin{bmatrix} C \\ C.A \\ \vdots \\ C.A^{n-1} \end{bmatrix} \quad (18)$$

2.4 MATRIZ DE GANHO K POR ALOCA33O DE P3LOS

Segundo Ogata (2010) ao inserir uma matriz de realimenta33o conforme Figura 10 3 poss3vel obter o controle sobre o sistema, caso ele seja dito control3vel. 3 poss3vel notar que as caracter3sticas, tanto transit3rias quanto de estabilidade, ser3o determinadas por $A-BK$. Logo uma escolha adequada de ganho K pode deixar o sistema assint3ticamente est3vel.

Figura 10 - Sistema realimentado com ganho K



Fonte: 10 - Ogata (2010)

A determinação da matriz K calculada manualmente é possível, mas não é indicado pois seria muito fácil de cometer um erro e, caso algum parâmetro mudasse, seria muito exaustivo de refazer. Assim, existem algumas ferramentas que permitem o seu cálculo. No matlab existem comandos específicos para quando se está trabalhando em função de transferência, bem como para quando se está trabalhando em espaço de estados.

Dentre as funções disponíveis para encontrar K , com o sistema representado em espaço de estados, estão os comandos *place* e *lqr*. Para utilizar o comando *place* são necessárias as matrizes A e B , e também ter a localização dos pólos desejados (utilizando do conceito e restrição visto no tópico anterior). No comando *lqr* as matrizes A e B também são necessárias, e outras duas matrizes de ponderação Q e R que serão vistas mais detalhadamente no tópico 2.6.

2.5 OBSERVADOR DE ORDEM PLENA

Em um sistema, os observadores podem ser usados para substituir sensores no sistema de controle. Em termos práticos podem ser considerados algoritmos que combinam sinais percebidos com outros dados e informações do sistema de controle para produzir sinais observados.

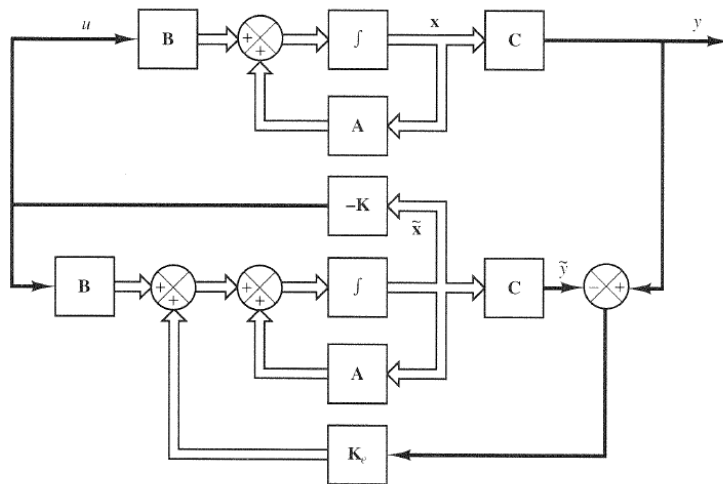
Conforme Ellis (2002) existem pelo menos quatro desvantagens causados por sensores, são eles:

- a) Sensores são caros, o que pode aumentar substancialmente o preço de um sistema de controle;
- b) Suas ligações associadas reduzem a confiabilidade do sistema de controle;
- c) Alguns sinais são impossíveis de medir, pois as variáveis a serem medidos podem não ser acessíveis por diversas razões, como ambientes insalubres e de restrita movimentação entre o controlador e o sensor, etc.;
- d) Sensores usualmente induzem erros tais como ruídos estocásticos, erros cíclicos e flexibilidade limitada.

Assim, pode-se concluir que, um observador de estados é utilizado para estimar os estados aos quais não é possível mensurar. Existem diversos tipos de observadores, e conseqüentemente diferentes formas de obtê-los. No presente trabalho será utilizado um observador de ordem plena demonstrado na Figura 11, que representa o efeito da adição do observador em malha fechada. Em algumas situações, alguns estados são mensuráveis enquanto

outros são inacessíveis, nesses casos é possível usar um estimador de estados de ordem reduzida que retorna somente a estimação para os estados que não são medidos. Para o presente sistema um sistema de ordem mínima também pode, e será usado, se necessário.

Figura 11 - Sistema de controle retroação do observador de ordem completa



Fonte: 11 - Ogata (2010)

Assim, a dinâmica do sistema é representada por (19), onde K_e pode ser obtido pelos mesmos métodos possíveis para K e deve possuir os autovalores ao menos cinco vezes mais rápidos do que o sistema.

$$\begin{bmatrix} \dot{X} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - K_e C \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} \quad (19)$$

2.6 CONTROLE ÓTIMO – LINEAR QUADRATIC REGULATION

Utilizando o comando *place* os pólos podem ser alocados arbitrariamente, mas com os resultados não foram satisfatórios foi necessário aprimorar o método da obtenção de K . a partir de então, o comando utilizado para encontrar a estimar os pólos foi a função *lqr*.

Segundo Anderson e Moore (1990), para definir um controle ótimo é necessário encontrar o melhor desempenho possível com relação a algum determinado critério ideal. O controlador LQR implementado aqui utiliza uma função custo contendo fatores de ponderação fornecido pelo desenvolvedor do controlador. Outro motivo para a utilização de deste controlador é o fato de ele ser aplicável aos sistemas tipo MIMO.

O objetivo da função custo apresentada é diminuir a energia gasta no sistema. Para tempo contínuo a função custo (20), segundo Franklin , Workman e Powell (1997), é descrita por:

$$J_{lqr} = \int_0^{\infty} (x^T Q x + u^T R u) dt, \quad (20)$$

na qual Q ($n \times n$, onde n corresponde ao número de estados para controlar) e R ($m \times m$, onde m corresponde ao número de entradas do processo) são ambas matrizes de ponderação definida positiva e simétrica. E, $x^T Q x$ representa o custo dos estados com peso Q e $u^T R u$ o controle de custo com peso R . Para tempo discreto se assume (21), que minimiza a função de custo.

$$u^K = -Kx, \quad (21)$$

sendo u^K é o vetor de controle, e a matriz de ganho LQR K é dada por (22).

$$K = R^{-1} B^T P \quad (22)$$

E P é derivado da equação algébrica de Riccati (23).

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (23)$$

Esta técnica de projeto LQR apresentada assume que a ordem do controlador irá ser igual à ordem do sistema, assumindo ter acesso completo a todos os estados. A matriz Q terá dimensão 12×12 , caso o acesso seja reduzido a uma quantidade menor de estados, a matriz será de dimensão relativa aos números acessíveis de estados.

2.6.1 Regra de Bryson

A primeira escolha para matrizes Q e R é encontrada pela regra do Bryson, que afirma essas matrizes como sendo diagonal. A representação pode ser vista em (24), (25), (26) e (27):

$$Q = \frac{1}{x_{i,max}^2} \quad (24)$$

$$R = \frac{1}{u_{i,max}^2} \quad (25)$$

$$Q = \begin{bmatrix} \frac{1}{x_{1,max}^2} & 0 & \dots & 0 \\ 0 & \frac{1}{x_{2,max}^2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{x_{n,max}^2} \end{bmatrix} \quad (26)$$

$$R = \rho \begin{bmatrix} \frac{1}{u_{1,max}^2} & 0 & \dots & 0 \\ 0 & \frac{1}{u_{2,max}^2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{u_{m,max}^2} \end{bmatrix} \quad (27)$$

Onde, segundo Naeini, Franklin e Powell (2002), $x_{i,max}$ e $u_{i,max}$ representam a maior resposta desejada ou entrada de controle para referente ao estado. E ρ é utilizado como a última ponderação relativa entre o controle de estado e a entrada.

Assim, basicamente, a regra de Bryson consiste em ponderar as variáveis que aparecem na função de custo de modo a que o valor máximo aceitável para cada termo seja inserido.

3 IMPLEMENTAÇÃO

Para simulações foram considerados os parâmetros: massa=1kg, gravidade=9.8 m/s² e valores referentes ao motor segundo Silva(2012) sendo $J_x=8.1e-3$ kgm², $J_y=8.1e-3$ kgm² e $J_z=14.2e-3$ kgm².

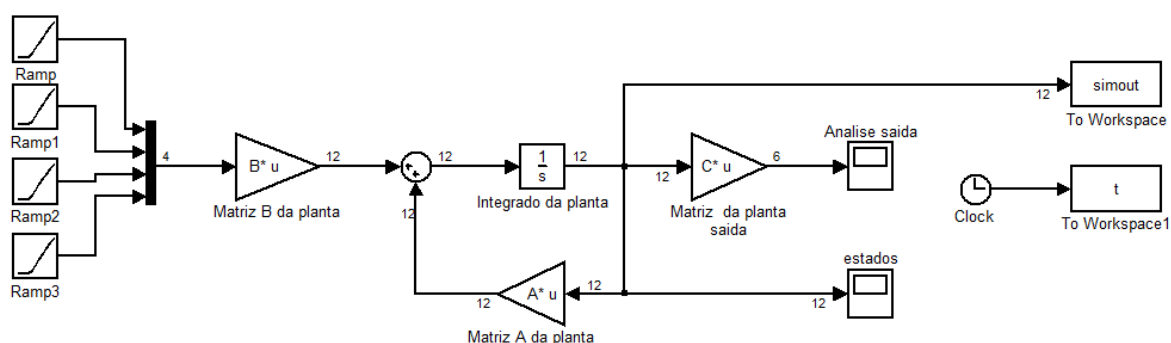
Para realizar os cálculos e simulações foi utilizado o software Matlab. As matrizes (3) e (4) foram escritas no software. O primeiro passo foi verificar a controlabilidade e a observabilidade do sistema utilizando os comandos *rank*, $Cx=ctrb(A,B)$ e $Ox=obsv(A,C)$, respectivamente. Sendo Cx a matriz de controlabilidade e Ox a matriz de observabilidade. O resultado do comando *rank* da matriz de controlabilidade e da matriz de observabilidade deve ser igual a dimensão da matriz A para atender os requisitos.

Utilizando da relação $Y = CX + Du$ e de acordo com (1) a matriz C foi definida para repassar a saída os estados desejados. Logo a matriz C foi definida de como (28):

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (28)$$

Assim o sistema pode ser representado de acordo com a Figura 12.

Figura 12 - Sistema em malha aberta



Fonte: 12 - Autor

A fim de implementar o controle era necessário verificar onde estavam os pólos do sistema. Através do comando $polos=eig(A)$ os polos foram obtidos e foi verificado, como

mostrado nos resultados na **Erro! Fonte de referência não encontrada.**, que todos estavam localizados em 0.

Uma vez que o sistema não é totalmente estável, como mostrado nos resultados que serão exibidos no tópico seguinte, onde somente alguns estados convergem, uma matriz de realimentação deveria ser encontrada a fim de estabilizar o sistema. Tendo em vista que o sistema se mostrou como totalmente controlável e que os pólos poderiam ser alocados arbitrariamente, a função $K1=place(A, B, J)$ foi utilizada, onde J é um vetor de mesma dimensão do vetor dos pólos originais, no qual os pólos são alocados de acordo com o projetista. Com pouca prática em sistemas MIMO e de ordem elevada, os pólos que foram alocados não estavam gerando respostas satisfatórias. Assim foi necessário buscar outra alternativa para encontrar a matriz de realimentação.

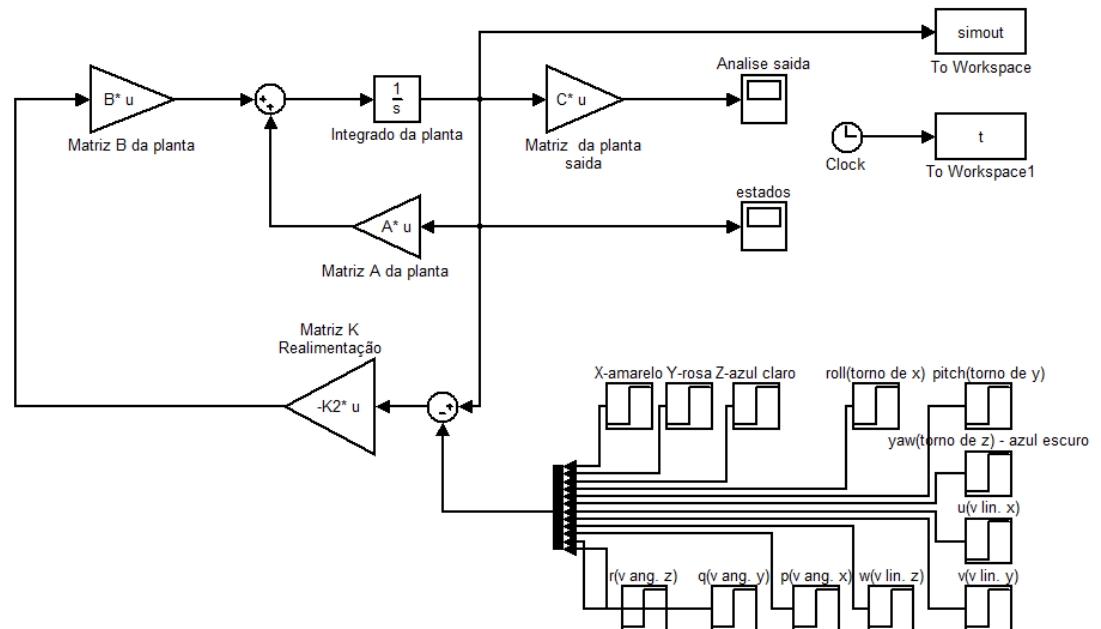
Com base nos artigos e teses de mestrado encontrados de Domingues (2009) e Silva (2012) o comando $[K2, S, E]=LQR(A, B, Q, R)$ foi implementado. Onde $K2$ é a matriz de realimentação, S é a solução associado a equação algébrica de Riccati e E são os pólos em malha fechada do sistema realimentado ($E = EIG(A-B*K)$). Os valores de Q e R são dados de acordo com 2.6.1 e após diversas tentativas de combinações foram definidos como (29) e (30), respectivamente:

$$Q = \begin{bmatrix} \frac{1}{0.4^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{0.4^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{0.4^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{0.004^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{0.004^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{0.004^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{0.1^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{0.1^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{0.1^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{0.001^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{0.001^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{0.001^2} & 0 \end{bmatrix} \quad (29)$$

$$R = 3 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (30)$$

Então o diagrama de blocos foi feito conforme Figura 13.

Figura 13 - Sistema com realimentação de K



Fonte: 13 - Autor

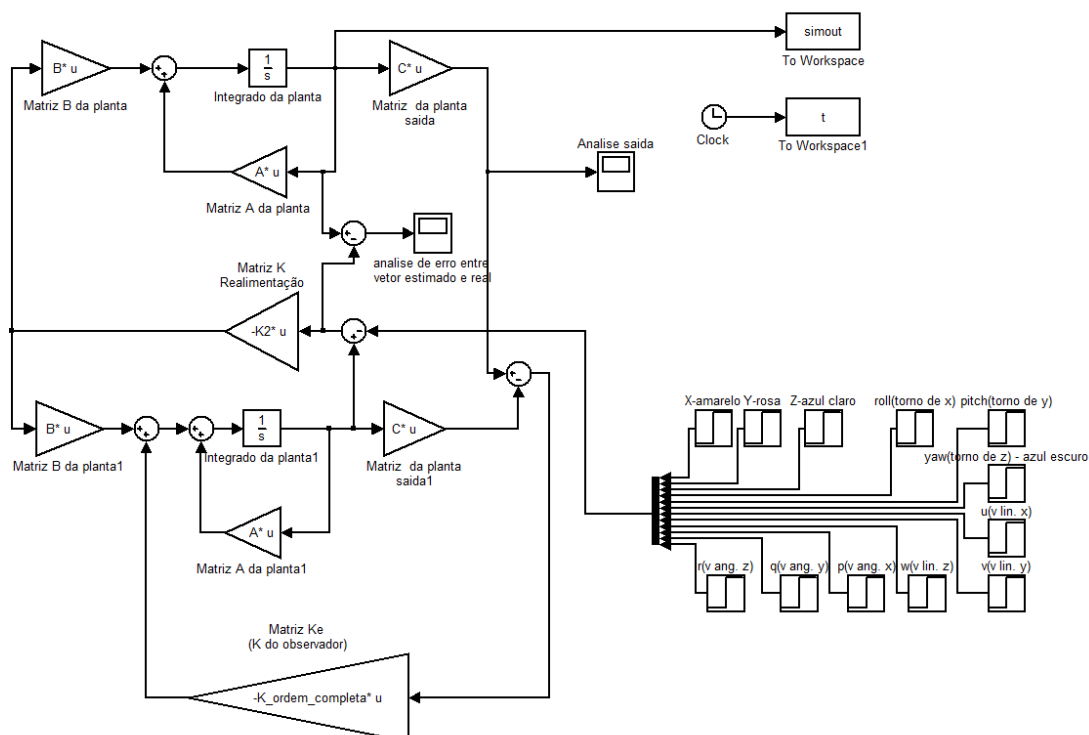
Com isto o sistema se tornou estável como será visto no tópico 4. Assim o próximo passo foi a implementação de um observador a fim de estimar os estados. Foi projetado um observador de ordem plena visto na Figura 11, ou seja, os doze estados serão estimados. Uma vez que Ogata(2010) define que os mesmos métodos podem ser implementados para escolha do ganho K_e do observador, com a condição de ter os pólos ao menos 5 vezes mais rápidos do que o sistema, o polo mais lento foi selecionado com a função $\min(E)$. Onde E é proveniente da função LGR do matlab. A função seleciona o menor valor dentro do vetor E, neste caso o valor mais próximo de zero.

Para utilizar a função `place`, sabendo que a mesma não permite que nenhum dos polos deve ter uma multiplicidade maior do que o número de entradas, foi criado um vetor de doze polos derivado do menor valor do vetor E. para isto, alguns polos foram divididos por valores entre um e dois. Três foram divididos por 1.2, três 1.5 e três por 1.8, sendo que três foram mantidos no valor original (menor valor do vetor E). Este vetor foi então multiplicado por cinco para atender a regra do controlador.

Então o comando $K_ordem_completa=place(A',C',Poc)'$ foi aplicado, onde $K_ordem_completa$ é matriz de realimentação do observador e Poc o vetor de polos ao menos 5 vezes mais rápidos do que o polo mais lento do sistema original.

O diagrama de blocos para o sistema com um observador de ordem completa fica de acordo com a Figura 14.

Figura 14- Sistema com observador de ordem completa

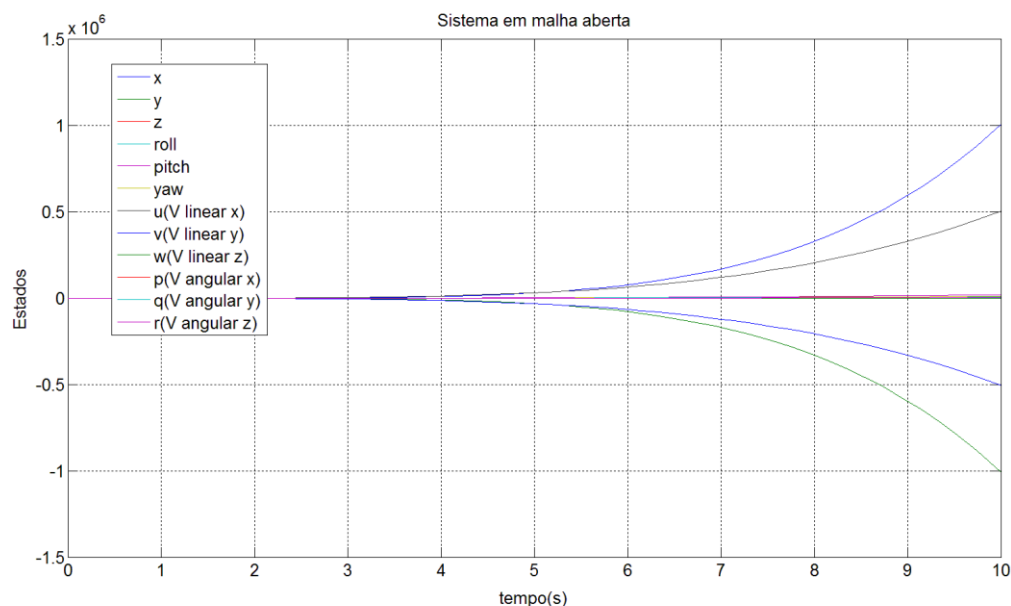


Fonte: 14 - Autor

4 RESULTADOS

Na Figura 15 pode ser visto do sistema em malha aberta, onde todos os pólos estavam localizados na origem.

Figura 15 - Resposta do sistema em malha aberta



Fonte: 15 - Autor

É possível verificar na Figura 15 que o sistema não é totalmente estável, apesar de alguns valores convergirem, os valores de x, y e suas respectivas velocidades lineares não convergem. O fato de todos os pólos do sistema serem em zero afeta esta estabilidade, pois estão o limiar da estabilidade. Ogata (2010) diz que para o sistema ser estável os pólos devem estar no plano negativo.

Após utilizar o método LQR, os valores de K do sistema e os polos podem ser visualizados na Figura 16 e Figura 17, respectivamente.

Figura 16 - K do sistema ao utilizar lqr

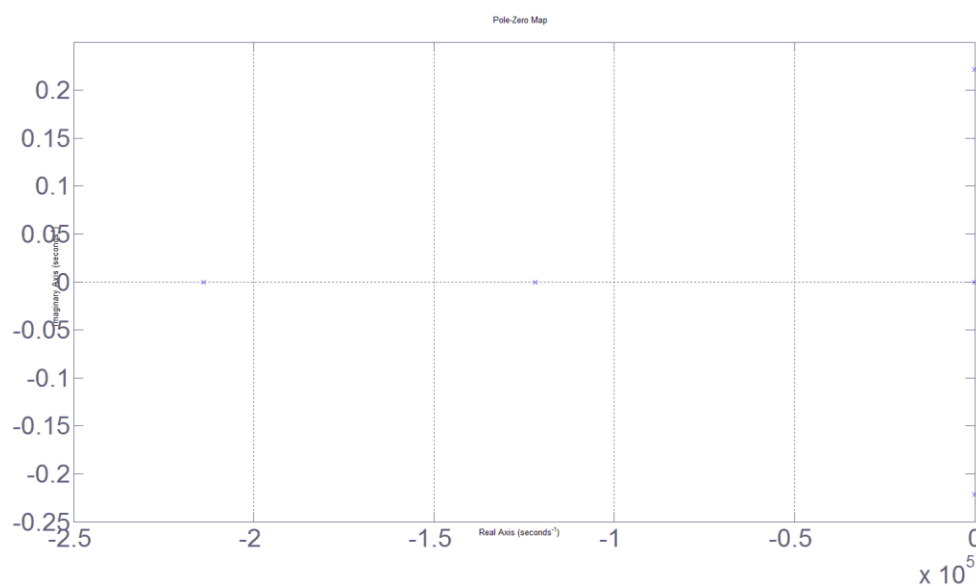
$K_2 =$

$1.0e+003 *$

| | | | | | | | | | | | |
|--------|---------|---------|---------|--------|--------|--------|---------|---------|---------|--------|--------|
| 0 | 0.0000 | -0.0043 | -0.0000 | 0 | 0.0000 | 0 | 0.0000 | -0.0176 | -0.0000 | 0 | 0.0000 |
| 0 | -0.0043 | 0.0000 | 1.1998 | 0 | 0.0000 | 0 | -0.0369 | 0.0000 | 1.7321 | 0 | 0.0000 |
| 0.0043 | 0 | 0 | 0 | 1.1998 | 0 | 0.0369 | 0 | 0 | 0 | 1.7321 | 0 |
| 0 | -0.0000 | -0.0000 | 0.0000 | 0 | 0.4330 | 0 | -0.0000 | -0.0000 | 0.0000 | 0 | 1.7321 |

Fonte: 16 - Autor

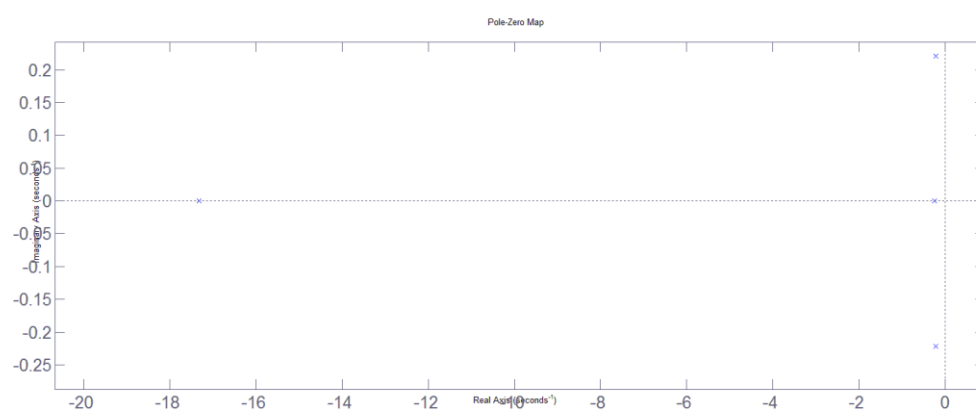
Figura 17 - Polos do sistema em malha fechada com ganho k



Fonte: 17 - Autor

Nota-se na Figura 17 é possível notar que todos são negativos. E, através, da Figura 18 e da Figura 19 é possível no que a maioria dos pólos estão próximo ao zero (pólos dominantes).

Figura 18 - Zoom do mapa dos polos da Figura 17



Fonte: 18 - Autor

Figura 19 - Polos do sistema em malha fecha com ganho K

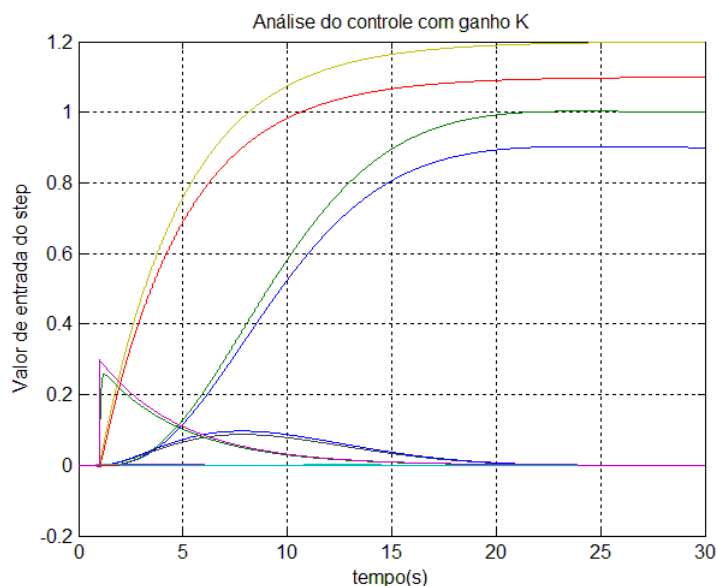
```

E =
    1.0e+005 *
    -2.138334330330486
    -0.000002213594362 + 0.000002213594362i
    -0.000002213594362 - 0.000002213594362i
    -0.000002500000000
    -2.138334330330486
    -0.000173187033860
    -1.219754089834675
    -0.000002213594362 + 0.000002213594362i
    -0.000002213594362 - 0.000002213594362i
    -0.000002500260512
    -0.000002500000001
    -0.000002500000000
  
```

Fonte: 19 - Autor

Com intuito de verificar o controle projetado, a função *step* foi utilizado e aos estados *x*, *y*, *z* e *yaw* foram atribuídos os valores respectivos de 0.9, 1, 1.1 e 1.2. Para os demais valores foi atribuído zero, e com isso a Figura 20 foi gerada.

Figura 20 - Análise do controle com ganho K



Fonte: 20 - Autor

Como é possível visualizar na Figura 20 os valores convergiram para a referência aplicada, e os demais valores convergiram para zero. Após implementar observador de ordem completa, o K do observador é mostrado na Figura 21.

Figura 21 - K do observador de ordem completa

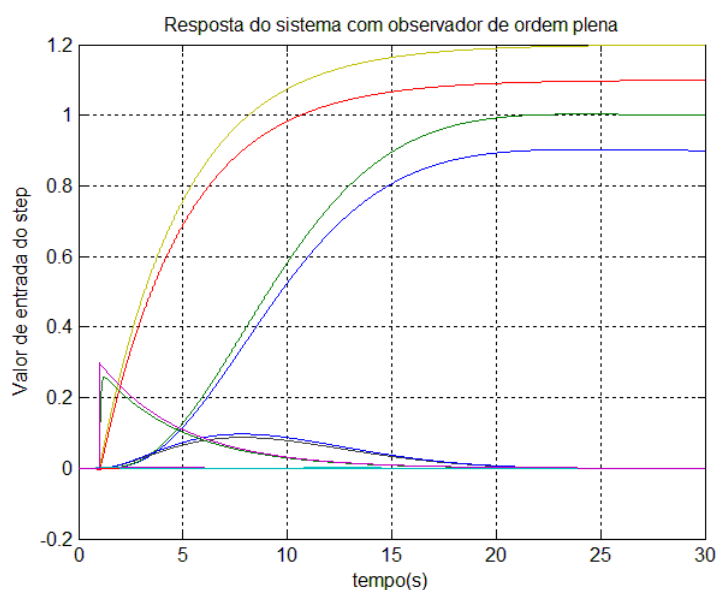
```

K_orden_completa =
    1.6352    0   -0.0746    0    0    0
    0    1.4583    0    0    0    0
   -0.0746    0    1.4898    0    0    0
    0    0    0    1.6667    0    0
    0    0    0    0    0.8750    0
    0    0    0    0    0    0.8750
    0.4815    0   -0.0933    0    9.8000    0
    0    0.2604    0   -9.8000    0    0
   -0.0933    0    0.2998    0    0    0
    0    0    0    0.5208    0    0
    0    0    0    0    0.1562    0
    0    0    0    0    0    0.1562
  
```

Fonte: 21 - Autor

Na Figura 22 é demonstrada os estados ao aplicar o observador, onde é possível observar que os mesmos são muito semelhantes aos estados sem o observador. O que mostra que o observador está estimando corretamente os estados.

Figura 22- Saída do sistema com observador de ordem plena

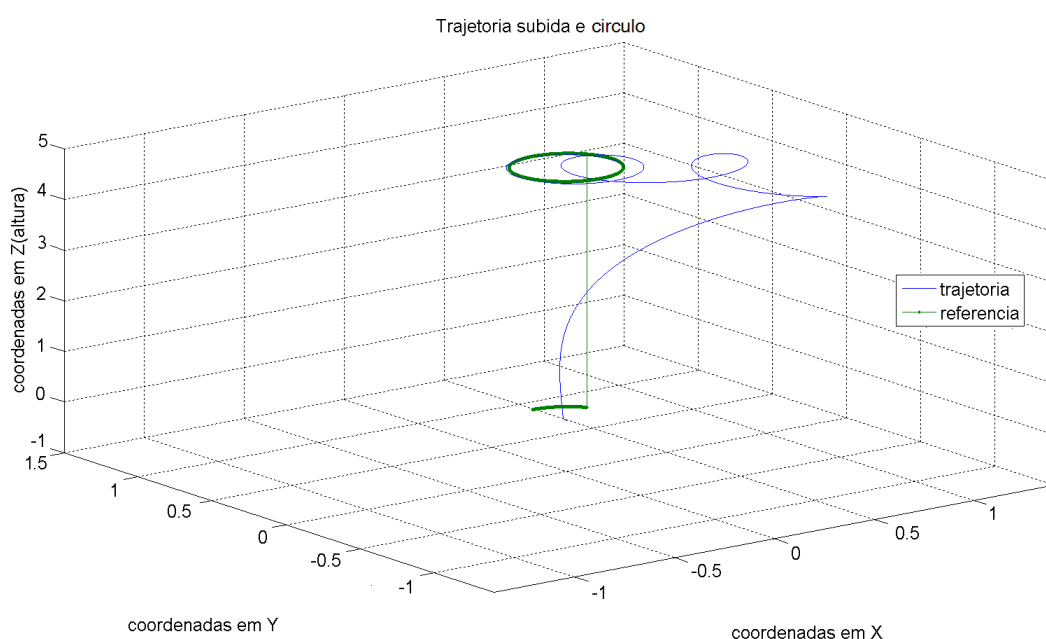


Fonte: 22 - Autor

Após as simulações para verificação de estabilidade, algumas trajetórias foram implementadas a fim de verificar o comportamento em um espaço tridimensional. Logo, foram feitas três simulações em plano 3D.

A primeira é uma trajetória onde o drone deveria subir rapidamente a 5 metros de altura e ficar numa trajetória de modo a realizar um círculo. O resultado obtido é mostrado na Figura 23, onde é possível verificar que apesar da instabilidade inicial, o objetivo da trajetória foi alcançado.

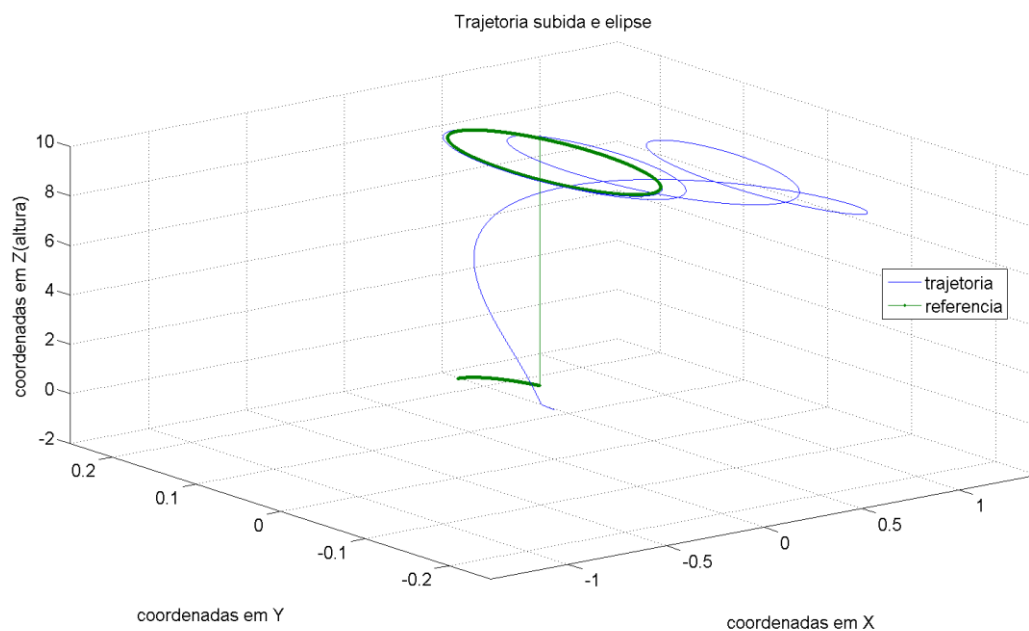
Figura 23 - Trajetória circular e subida



Fonte: 23 - Autor

A segunda simulação foi semelhante ao primeiro, mas para verificar o funcionamento da trajetória, a altura foi modificada para 10 metros e a trajetória passou a ser elíptica. A representação segue na Figura 24.

Figura 24 - Trajetória Elipsoidal e subida

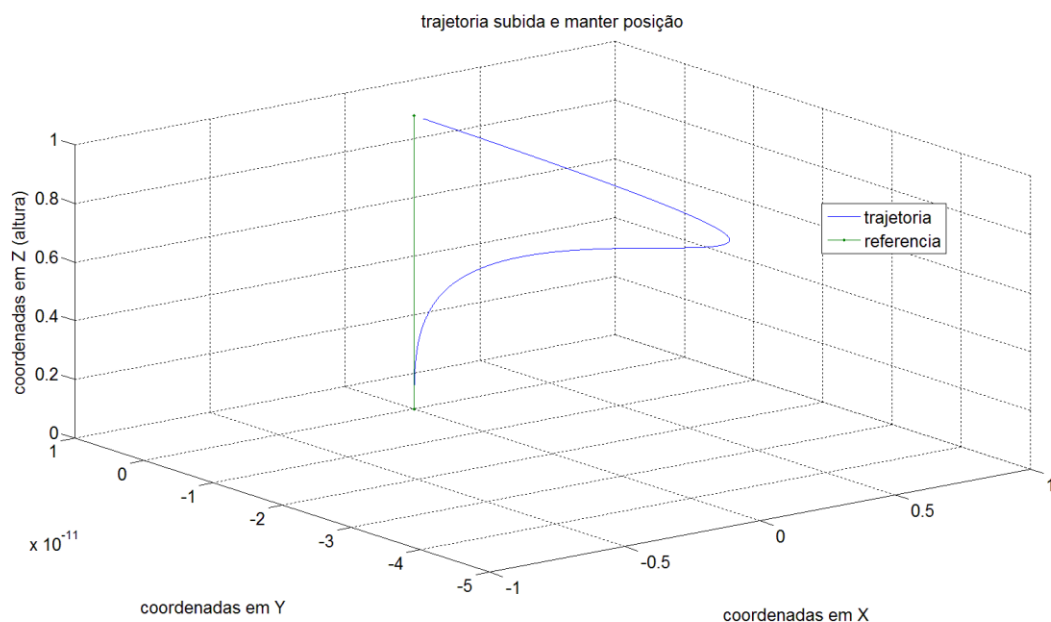


Fonte: 24 - Autor

O resultado mostrado na Figura 24 mostra que ao alterar o percurso, o quadricóptero seguirá a referência. Apesar de novamente apresentar um distúrbio no período inicial, a trajetória estabilizou conforme o desejado, como pode ser visualizado na forma elíptica e na altura final.

Na terceira trajetória o objetivo era somente subir e permanecer na posição, e é representado na Figura 25.

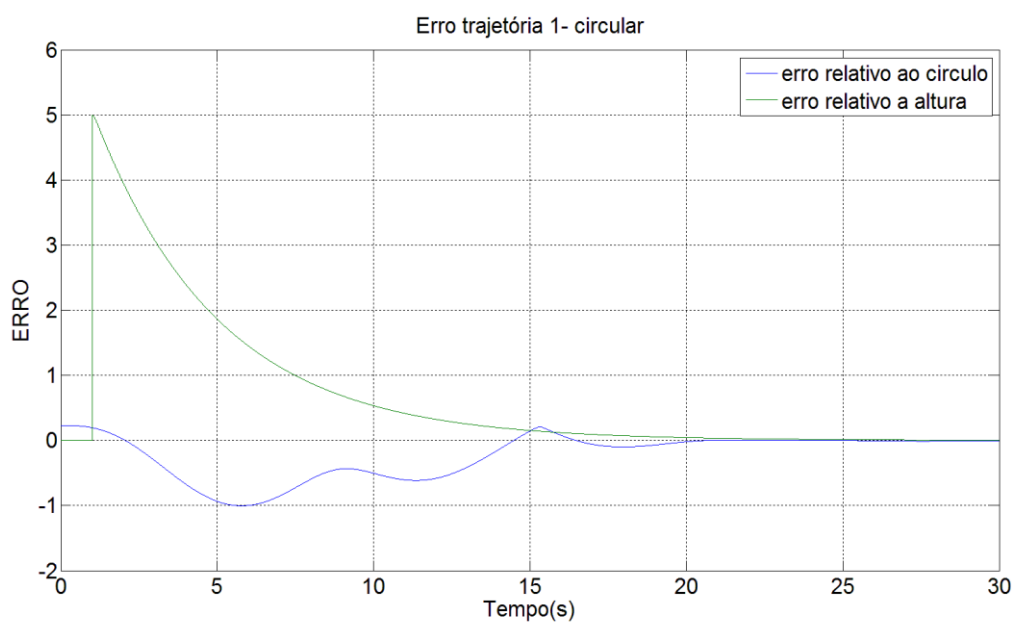
Figura 25 - deslocamento vertical e em equilíbrio



Fonte: 25 - Autor

É possível notar novamente, na Figura 25, a distorção na parte inicial. No entanto é necessário ressaltar que a escala no eixo y do gráfico possui valores que podem ser considerados desprezíveis se compararmos a escala a metros ou mesmo centímetros. Também é possível reparar que convergiu para o valor esperado.

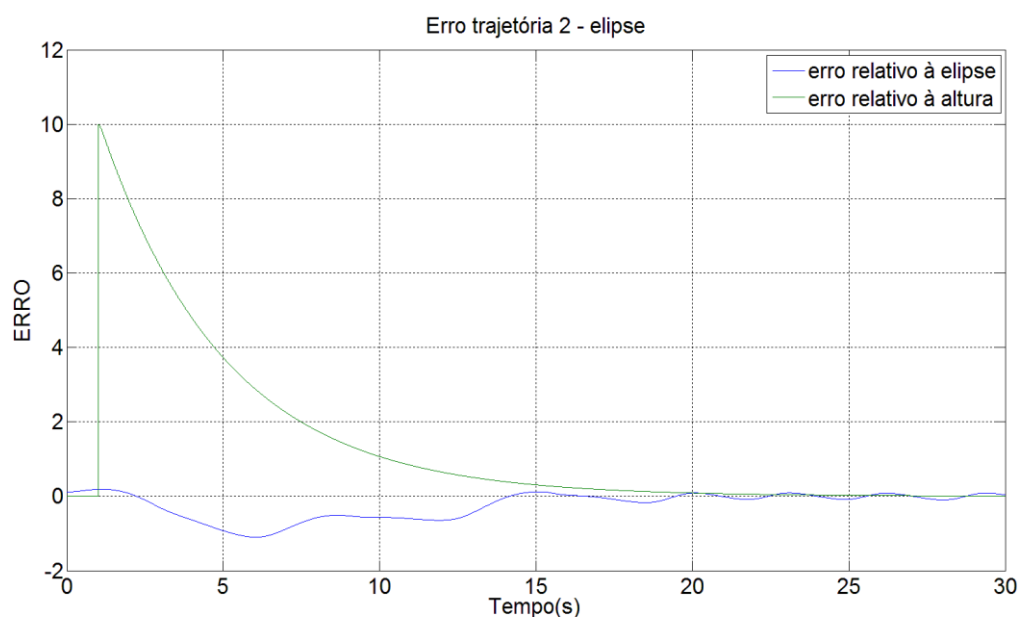
Figura 26 - Erro relativo a trajetória 1 (círculo e subida)



Fonte: 26 - Autor

Na Figura 26 é apresentado o gráfico do erro da trajetória 1(circular), em que é possível observar que tanto a altura quanto as coordenadas x e y, que correspondem ao círculo, convergem para o valor esperado. Aos quinze segundos o sistema apresenta uma resposta próxima ao esperado e, aos vinte segundos estabilizando nos valores finais desejados, onde o erro é tende a zero.

Figura 27 - Erro relativo a trajetória 2 (elipse e subida)



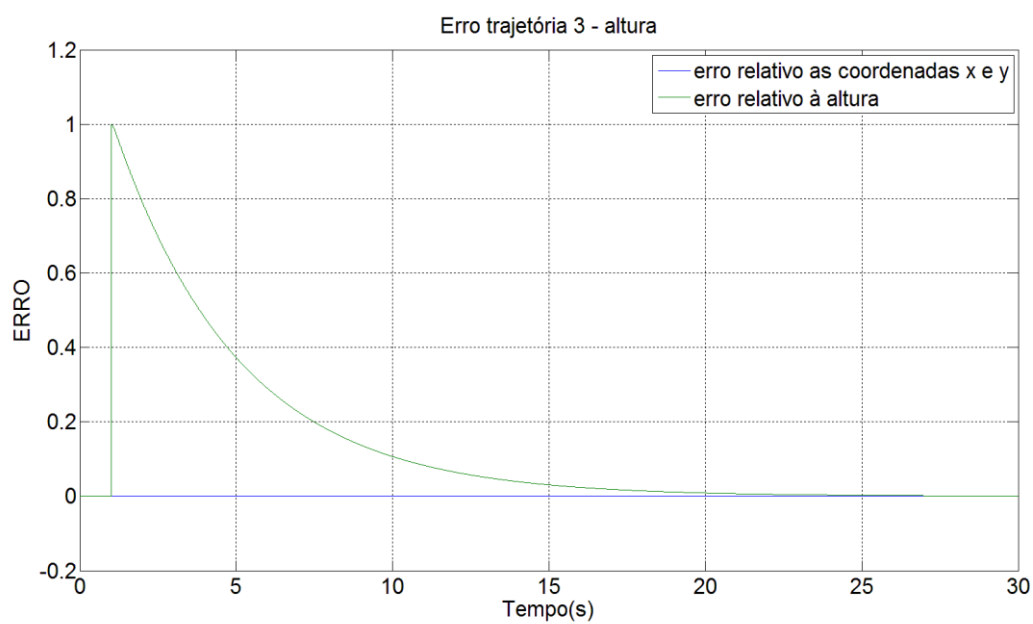
Fonte: 27 - Autor

Semelhante ao comportamento do erro da trajetória 1, o erro da trajetória 2(elipse) apresentado na Figura 27 também apresenta uma estabilização após quinze segundos, tendendo definitivamente a zero após vinte segundos.

Para a trajetória 3, focada somente no deslocamento vertical, a Figura 28 apresenta um erro inicial na altura, em que necessita de vinte segundos para a estabilização. Após este tempo todos os erros tendem a zero, caracterizando o ato de pairar no ar mantendo a posição de altura e de coordenadas x e y.

Também, com o gráfico demonstrado na Figura 28 é possível entender melhor a trajetória apresentada na Figura 25.

Figura 28 - - Erro relativo a trajetória 3 (pairar)



Fonte: 28 - Autor

5 CONSIDERAÇÕES FINAIS

Durante o desenvolvimento do trabalho mostrou-se a importância dos modelos matemáticos para poder obter o controle de uma planta, principalmente quando se refere a um modelo complexo como MIMO, e que contenha todos os graus de liberdade possíveis conhecidos.

Ao comparar a saída do sistema com o ganho K na Figura 20 e a saída do sistema com observador na Figura 22, é possível concluir que o observador está estimando os estados de forma muito próxima ao valor real.

Apesar do tempo de estabilização ainda ser elevado como mostrado na Figura 20 e nas três trajetórias realizadas (Figura 23, Figura 24 e Figura 25), de forma geral, pode-se afirmar que a metodologia proposta é promissora para a realização do controle.

Com os gráficos de erros apresentados na Figura 26 e na Figura 27 é possível verificar que a trajetória simulada está próxima da trajetória de referência, sendo que o maior erro inicial é na altura. Este erro é obtido devido a entrada colocada na simulação, na qual foi utilizado o comando *step*, que efetua a transição imediata entre os estados, porém fisicamente essa transição é inviável. Assim é possível concluir que a metodologia abordada funciona.

E, com o gráfico de erro apresentado na Figura 28 é possível comprovar que o quadricóptero realizaria a trajetória de subida mantendo acurácia nos valores das coordenadas x e y . O valor relativo à altura alcança e mantém a estabilidade após vinte segundos.

REFERÊNCIAS

Advances in Military Technology - Department of Control and Instrumentation, Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno, Czech Republic, “Modelling and Control of a Quadcopter”, December 2010

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6022**: Informação e documentação: artigo em publicação periódica científica impressa – apresentação. Rio de Janeiro, 2003.

B. D. O. Anderson and J. B. Moore. Optimal Control: Linear Quadratic Methods. Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

BLAKELOCK, JH. Automatic Control of Aircraft and Missiles. New York; Wiley, 1991. 672 p. ISBN 978-0-471-50651-5.

Controlo de Formações de Veículos Aéreos não Tripulados, Henrique Ribeiro Delgado da Silva, 2012.

Ellis, George (2002). Observers in Control System - A Practical Guide. Academic PRESS, Elsevier Science (USA).

JOHN J. CRAIG. Robótica, 3ª edição, 2013.

G. F. Franklin, J. D. Powell, and A. Emami-Naeini. Feedback Control of Dynamic Systems. Prentice Hall, Upper Saddle River, NJ, 4th edition, 2002.

G.F. Franklin, M.L. Workman, and D. Powell. Digital control of dynamic systems. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1997.

OGATA, Katsuhiko. Engenharia de Controle Moderno. 5. ed. São Paulo: Pearson Hall, 2010.

Quadrotor prototype, Jorge Miguel Brito Domingues, 2009.

ANEXO A – Programa matlab

```

tic()
clc
close all
clear all

%%
%% ===== %
% ===== Controle de um quadricóptero ===== %
% ===== por variáveis de estado ===== %
% ===== (alocação de pólos) ===== %
% ===== %

%% Autor: Matheos Hahn Birck
% Disciplina: Trabalho de conclusão II
% Orientador: Julio Ceballos Aya

% Equação em variáveis de estado:
%  $\dot{x}_{\text{ponto}}(t) = Ax(t) + Bu(t)$ 
%  $\dot{y}(t) = Cx(t) + Du(t)$ 

% onde: A,B,C e D são matrizes que representam o modelo matemático do
% quadricóptero

% x(t): vetor de estados (1xn)
% u(t): vetor de sinais de entrada(1xp)
% y(t): vetor de sinais de saída (1xq)
% A: matriz de dinâmica (nxn)
% B: matriz de entrada(nxp)
% C: matriz de saída(qxn)
% D: matriz do sistema(qxp), em muitos casos nula

%% meu vetor x será:
% X= [ x
%      y
%      z
%      roll
%      pitch
%      yaw
%      u(V linear x)
%      v(V linear y)
%      w(V linear z)
%      p(V angular x)
%      q(V angular y)
%      r(V angular z)]

%% Parametros motor
% valores usados do artigo (http://www.usp.br/ldsv/wp-content/uploads/2014/10/Relat%C3%B3rio\_FINAL\_IC\_AEP\_HENRIQUE\_KOKRON\_RODRIGUES.pdf)
% Controle de Formações de Veículos Aéreos não Tripulados ( Henrique
% Ribeiro Delgado da Silva)
m=1 ;% kg -->do artigo original: 1
Jx=8.1e-3 ; %kgm2 -->do artigo original: 8.1e-3
Jy=8.1e-3 ; %kgm2 -->do artigo original: 8.1e-3
Jz=14.2e-3 ; %kgm2 -->do artigo original: 14.2e-3
g=9.8 ;%m/s2

%% Matrizes originais
% Matriz A
A=zeros(12);
A(1,7)=1;
A(2,8)=1;
A(3,9)=1;
A(4,10)=1;
A(5,11)=1;

```

```

A(6,12)=1;
A(7,5)= g;
A(8,4)=-g;

% Matriz B
B=zeros(12,4);
B(9,1)= -(1/m);
B(10,2)= 1/Jx;
B(11,3)= 1/Jy;
B(12,4)= 1/Jz;

% Matriz C
C=[eye(6) zeros(6)]; % nos estados que quero na saída é colocado 1
% C=[zeros(6) zeros(6)];
% C(1,1)= 1; % x
% C(2,2)= 1; % y
% C(3,3)= 1; % z
% C(4,4)= 1; % roll (x)
% C(5,5)= 1; % pitch (y)
% C(6,6)= 1; % yaw (z)

% Matriz D
D=zeros(12,4);
%iu=4; % não necessário neste caso. É o número de entradas do sistema. Só é
%usado quando se trabalha com a função de transferência.

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Observabilidade %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Passo 1:verificar se é observável. Matriz de observabilidade:
Ox=obsv(A,C);
% Determinante_Ox =det(Ox);
ordem_sis=size(A); % a matriz A sempre será quadrada, a ordem é dada por qualquer
um dos termos.
n=rank(Ox);% posto da matriz Cx

if ordem_sis(1)==n
    fprintf('É observável, posto da matriz de observabilidade é igual a ordem do
sistema\n\n')
else
    fprintf('Não é totalmente observável, pois o posto da matriz de observabilidade
é diferente da ordem do sistema.\n\n')
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Controlabilidade %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Passo 2:verificar se é controlável. Matriz de controlabilidade:
Cx= ctrb(A,B);
% Determinante_Cx =det(Cx);
ordem_sis=size(A); % a matriz A sempre será quadrada, a ordem é dada por qualquer
um dos termos.
n2=rank(Cx);% posto da matriz Cx

if ordem_sis(1)==n2
    fprintf('É controlável, posto da matriz de controlabilidade é igual a ordem do
sistema\n\n')
else
    fprintf('Não é totalmente controlável, pois o posto da matriz de
controlabilidade é diferente da ordem do sistema.\n\n')
end

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Localização dos polos%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%localização dos pólos do sistema original raízes do denominador da função,
%usar roots(coeficientes da equação característica)
%ou função "eig". Os auto valores são os polos do sistema.
%Raízes do det(sI-A)= AUTOVALORES de A = polos
polos=eig(A)% os pólos são encontrados para saber onde estão, e a partir deles
estimar pólos novos n

```

```

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Polinomio caracteristico %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%pol_caract= poly(A); % só é utilizado quando trabalhado com função de
transferencia

% eu verifiquei... usandoa função [num,den] = ss2tf(A, B, C1, D, iu) o cálculo de
"num"
% deu o mesmo de poly. Vale lembrar que a matriz C é de dimensão diferente neste
caso para
% acerto de calculos entre matrizes(é preenchida com zeros).

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Formas canônica %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% A forma canônica controlável ou observável, neste caso não é necessária pois eu
já estou
% com o sistema em variáveis de estado. É só desejável se eu tiver função de
transferencia
% e quiser passar pra espaço de estados.

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Matriz ganho K %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% São tres métodos:
% no primeiro a partir dos novos polos encontrados em, determinar novos
% polos arbitrarios e usar a função "place" ou "arcker".
%
% no segundo usar a matriz de transformação T.
% K= [ak(n)- a(n) |ak(n-1)- a(n-1) |... |ak(2)- a(2) |ak(1)- a(1)] T^-1 onde a(n)
% são referentes ao polinomio caracteristicos e a matriz T=MW . Matriz M é
% a matriz de controlabilidade e a matriz W é dada por:
%
%      | a(n-1) a(n-2) ... a(1) 1 |
%      | a(n-2) a(n-3) ... 1 0 |
% W = |   :           :           :   : |
%      | a(1)      1     ... 0 0 |
%      | 1         0     ... 0 0 |
%
% no terceiro, comando "lqr", calcula o ganho ótimo da matriz K a solução
% de S associada a equação de Ricatti e a matriz E são os polos em malha
% fechada--> E = EIG(A-B*K).
% [K,S,E] = LQR(SYS,Q,R,N)

%% % ----- primeiro metodo -----
% A verdade é que não funciona para qualquer pólos eu tentei pólos
% arbitrários e muitas vezes deu erro usando o comando rand para gerar
% polos, a função provou funcionar. No entanto para estimar meus pólos por
% conta própria é necessario uma certa experiencia ou alguma dica de valor
% inicial, ainda mais para um sistema de ordem tão grande. Foram tentados
% combinações, mas como era de se esperar os resultados não estavam
% satisfatórios, logo este método não seria o mais viável. Abaixo segue um
% exemplo do comando.
%
% polos novos/desejados
%J=rand(1,12);
%[K1,PREC] = place(A,B,J) % para sistemas de multiplas entradas é necessario
% utilizar o comando place, pois o comando acker somente para SISO.
%polos_em_mf= eig(A-B*K1) % para verificar se a função funcionou

%% % ----- segundo metodo -----
% W=[ pol_caract(1,11) pol_caract(1,10) pol_caract(1,9) pol_caract(1,8)
pol_caract(1,7) pol_caract(1,6) pol_caract(1,5) pol_caract(1,4) pol_caract(1,3)
pol_caract(1,2) pol_caract(1,1) 1;
%      pol_caract(1,10) pol_caract(1,9) pol_caract(1,8) pol_caract(1,7)
pol_caract(1,6) pol_caract(1,5) pol_caract(1,4) pol_caract(1,3) pol_caract(1,2)
pol_caract(1,1) 1 0;
%      pol_caract(1,9) pol_caract(1,8) pol_caract(1,7) pol_caract(1,6)
pol_caract(1,5) pol_caract(1,4) pol_caract(1,3) pol_caract(1,2) pol_caract(1,1) 1 0
0;
%      pol_caract(1,8) pol_caract(1,7) pol_caract(1,6) pol_caract(1,5)
pol_caract(1,4) pol_caract(1,3) pol_caract(1,2) pol_caract(1,1) 1 0 0 0;

```

```

%      pol_caract(1,7) pol_caract(1,6) pol_caract(1,5) pol_caract(1,4)
pol_caract(1,3) pol_caract(1,2) pol_caract(1,1) 1 0 0 0 0;
%      pol_caract(1,6) pol_caract(1,5) pol_caract(1,4) pol_caract(1,3)
pol_caract(1,2) pol_caract(1,1) 1 0 0 0 0 0;
%      pol_caract(1,5) pol_caract(1,4) pol_caract(1,3) pol_caract(1,2)
pol_caract(1,1) 1 0 0 0 0 0 0;
%      pol_caract(1,4) pol_caract(1,3) pol_caract(1,2) pol_caract(1,1) 1 0 0 0 0 0 0
0;
%      pol_caract(1,3) pol_caract(1,2) pol_caract(1,1) 1 0 0 0 0 0 0 0 0;
%      pol_caract(1,2) pol_caract(1,1) 1 0 0 0 0 0 0 0 0 0;
%      pol_caract(1,1) 1 0 0 0 0 0 0 0 0 0 0;
%      1 0 0 0 0 0 0 0 0 0 0 0]
%
% T=Cx*W
% % a formula diz assim T=Cx*w, mas as dimensoes não fecham, para fechar ...
% % Cx deveria ser transposto, e depois calcular os valores de K, por essas
% % incertezas nos metodos 1 e 2, eu usei o 3.
% % T=Cx'*W
% %K= [ak(n)- a(n) |ak(n-1)- a(n-1)|...|ak(2)- a(2) |ak(1)- a(1)] T^-1

%% ----- terceiro metodo -----
% Matrizes Q e R definidas a seguir para encontrar a matriz K. E também
% os polos (E) melhores definidos para o sistema.

% Os valores de Q e R são definidos de acordo com a regra de bryson
% http://www.uz.zgora.pl/~wpaszke/materialy/kss/lqrnotes.pdf
% https://www.coursehero.com/file/p129oj7/Brysons-rule-A-first-choice-for-the-
% http://robotics.itee.uq.edu.au/~metr4202/2014/lectures/Hespanha-Linear%20Systems-
% Part%20IV.pdf

% os valores de Q e R São definidos de acordo com o valor máximo aceitado
% para cada item. Valores estes definidos entre Q(i,i)=1/((valor maximo do estado
corresponde)^2).
% Neste caso Q(1,1) representa a posição x.
Q=eye(12);
a=sqrt(1/12);
aux_a=a^2;
%aux_a=1;
Q(1,1)=aux_a/(0.4^2);      % x
Q(2,2)=aux_a/(0.4^2);      % y
Q(3,3)=aux_a/(0.4^2);      % z
Q(4,4)=aux_a/(0.004^2);    % roll
Q(5,5)=aux_a/(0.004^2);    % pitch
Q(6,6)=aux_a/(0.004^2);    % yaw
Q(7,7)=aux_a/(0.1^2);      % u - V linear x
Q(8,8)=aux_a/(0.1^2);      % v - V linear y
Q(9,9)=aux_a/(0.1^2);      % w - V linear z
Q(10,10)=aux_a/(0.001^2);  % p - V angular x
Q(11,11)=aux_a/(0.001^2);  % q - V angular y
Q(12,12)=aux_a/(0.001^2);  % r - V angular z

R=eye(4);
u_motor=3;
b=sqrt(1/4);
aux_b=b^2;
%aux_b=1;
R(1,1)=aux_b/(u_motor^2);
R(2,2)=aux_b/(u_motor^2);
R(3,3)=aux_b/(u_motor^2);
R(4,4)=aux_b/(u_motor^2);

% comando LQR... volta matriz K, solução S de ricatti e polos E em malha fechada
[K2,S,E]=LQR(A,B,Q,R) % K= optimal gain matrix K, S= the solution S of the
associated algebraic Riccati equation e a matriz E the closed-loop eigenvalues E =
EIG(A-B*K).

```

```

fprintf('K2= matriz ganho K, S= Solução de S associada a equação de Ricatti e a
matriz E são os polos em malha fechada--> E = EIG(A-B*K).\n\n')

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Observador - matriz L(ordem plena) %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% no observador de ordem PLENA são estimados todos os estados

% Segundo Ogata (quinta edição , pagina 687), para projetar o K do observador é
% possível estimar um Ke multiplicando meus polos dominantes(perto de zero) por 5,
% deixando assim o sistema do observador/estimador 5 vezes mais rápido.
polo_mais_proximo=min(E)
aux=[polo_mais_proximo/1.2 polo_mais_proximo/1.2 polo_mais_proximo/1.2
polo_mais_proximo/1.5 polo_mais_proximo/1.5 polo_mais_proximo/1.5
polo_mais_proximo/1.8 polo_mais_proximo/1.8 polo_mais_proximo/1.8 polo_mais_proximo
polo_mais_proximo polo_mais_proximo];
Poc=aux*5;

% Para o Observador eu preciso observar os estados que não consigo medir.
% segundo ogata(quinta edicao, problema A.10.10 para o problema dual a
% matriz Ke(L) é o transposto conjugado de K.
% L = place(A',Co',Po)';% fonte: http://html.rincondelvago.com/observadores-de-
estado.html
% tamanho_L= size(L);
K_orden_completa=place(A',C',Poc)';

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Observador - matriz L(ordem minima) %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% no observador de ordem MINIMA são estimados os estados aos quais eu nao
% tenho como mensurar.

% para isso deve ser determinadas novas matrizes A, B, C e D para o
% observador. As matrizes Aaa, Aab, Aba, Abb, Ba e Bb são formados dos
% fragmentos das originais( ogata 5th ed, pg 696.)

Aaa= A(1:6,1:6); % A(linha 1 até linha 6 ,coluna 1 até coluna 6)
Aab= A(1:6,7:12); % A(linha 1 até linha 6 ,coluna 7 até coluna 12)
Aba= A(7:12,1:6); % A(linha 7 até linha 12 ,coluna 1 até coluna 6)
Abb= A(7:12,7:12); % A(linha 7 até linha 12 ,coluna 7 até coluna 12)
Ba= B(1:6,1:4); % A(linha 1 até linha 6 ,coluna 1 até coluna 4)
Bb= B(7:12,1:4); % A(linha 7 até linha 12 ,coluna 1 até coluna 4)

% Segundo Ogata (quinta edição , pagina 687),como no observador de ordem
% plena, para projetar o K do observador é possível estimar um Ke
% multiplicando meus polos dominantes(perto de zero)por 5.

% função para encontra o polo mais proximo de zero da matriz "E"
% proveniente dos polos em malha fechada do comando LQR
E_linha=[polo_mais_proximo polo_mais_proximo polo_mais_proximo polo_mais_proximo
polo_mais_proximo polo_mais_proximo];
%E_linha=[AZUL(amplitude) VERDE(amplitude) VERMELHO VERDE(longitude)
AZUL(longitude) AMARELO];
Po=E_linha*5;

% % segundo ogat(quinta edicao, problema A.10.10 para o problema dual a
% % matriz Ke(L) é o transposto conjugado de K.
Ke = place(Abb',Aab',Po)'% fonte: http://html.rincondelvago.com/observadores-de-
estado.html

% Matrizes de ganho
A_chapeu = Abb - (Ke*Aab);
%B_chapeu = Aba - (Ke*Aaa);%no diagrama ta assim
B_chapeu =(A_chapeu*Ke)+ Aba - (Ke*Aaa); % conforme formula é assim
F_chapeu = Bb - (Ke*Ba);
C_chapeu=[zeros(6);eye(6)];
D_chapeu= [eye(6) ; Ke];
%%
toc()

```