

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RENAN DOS SANTOS FAGUNDES

ROBÔ PÊNULO INVERTIDO

Porto Alegre

2015

RENAN DOS SANTOS FAGUNDES

ROBÔ PÊNULO INVERTIDO

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para Graduação em Engenharia Elétrica.

ORIENTADOR: Prof. Dr. Valner João Brusamarelo

Porto Alegre

2015

RENAN DOS SANTOS FAGUNDES

ROBÔ PÊNULO INVERTIDO

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de “Projeto de Diplomação”, do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Dr. Valner João Brusamarelo, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Aprovado em: ____/____/____

Banca Examinadora:

Prof. Dr. Valner João Brusamarelo, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Renato Ventura Bayan Henriques, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Jeferson Vieira Flores, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

AGRADECIMENTOS

Gostaria de agradecer a minha família, pelo apoio e compreensão nas horas em que estive distante. A Universidade Federal do Rio Grande do Sul pela qualidade de ensino. Agradeço a cada trabalho difícil, agradeço à cada prova complicada, pois foram esses obstáculos que me fizeram o profissional que sou hoje.

Agradeço a todos os amigos que fiz nesses vários anos de faculdade, pois todos me ajudaram de alguma forma.

Agradeço à minha namorada Carolina Ávila Fochessato pelo apoio nos momentos difíceis.

Agradeço à minha família, meu pai, minha mãe que foram dois grandes exemplos na minha vida. Agradeço aos meus irmãos Igor, Tais e Hortênsia, vocês são os melhores companheiros que eu poderia ter na vida!

Agradeço à Deus pela Vida!

Muito Obrigado!

Trabalhe como se tudo dependesse de você, reze como se
tudo dependesse de Deus.

Autor desconhecido

RESUMO

Este trabalho visa a construção de um robô pêndulo invertido e análise de alguns métodos existentes de fusão de informação para cálculo do ângulo deste robô, combinados com as metodologias de controle de estabilidade do pêndulo invertido. A análise consiste em estudar como os algoritmos se comportam com os métodos de controle da planta e medir o ângulo do pêndulo com confiabilidade, usando os métodos de fusão de sensores estudados, Filtro Kalman e Filtro Complementar. As metodologias de sistema de controle mais utilizadas para este fim são o PID e o LQR. Neste trabalho são implementadas as duas metodologias citadas, e comparados os seus resultados.

Palavras-chave: Robô Pêndulo Invertido. Filtro de Kalman. Filtro Complementar. Controlador PID. Controlador LQR.

ABSTRACT

This work aims to construction of an inverted pendulum robot and analysis of some existing methods of information fusion to calculate the angle of this robot, combined with the methodologies of the inverted pendulum stability control . This analysis intends to study how the algorithms behave with the plant control methods and measure the pendulum angle reliability, using the studied methods of fusion sensors, Kalman Filter and complementary Filter. The control system methodologies most used for this purpose are the PID and LQR . In this work the two cited methodologies are implemented , and compared their results.

Keywords: Inverted Pendulum Robot. Kalman Filter. Complementary Filter. PID Controler. LQR Controler.

LISTA DE ILUSTRAÇÕES

Figura 1 - Exoesqueleto, trabalho do grupo do Neurocientista Miguel Nicolelis.	12
Figura 2: Segway.....	13
Figura 3: Diagrama esquemático elétrico e mecânico do motor de corrente contínua.....	15
Figura 4: Diagrama de corpo livre da roda.....	18
Figura 4: Diagrama de corpo livre da roda.....	20
Figura 6: Modelo do sistema do espaço de estados.....	26
Figura 7: Estrutura de equações do Filtro Kalman.....	28
Figura 8: Estrutura no espaço de estados do filtro Kalman.....	32
Figura 9: Estimação do ângulo de inclinação usando acelerômetros e giroscópio	35
Figura 10: Sistema de controle digital.....	36
Figura 9: Parcelas do controlador PID	38
Figura 10: Estrutura de controle LQR.....	39
Figura 13 - Partes de do robô pêndulo.....	41
Figura 11 – Robô pêndulo montado.	43
Figura 15 – diagrama de alimentação.....	45
Figura 16 – Kit de desenvolvimento Launchpad da Texas Instruments.....	45
Figura 17 – Diagrama de blocos do projeto.	46
Figura 18 – Comparativo de performance entre Cortex M3 e M4+FPU.	47
Figura 19 – Disposição geométrica dos sensores inerciais.	48
Figura 20 – (A)Eixos do Sensor (B) Placa com sensor montado	48
Figura 21 – Funcionamento do Encoder.....	50
Figura 22 – Encoder utilizado montado no motor.....	51
Figura 23 – diagrama de blocos do filtro complementar.....	52
Figura 24 – diagrama de entradas e saídas do filtro Kalman.	54
Figura 25 – Estrutura do controlador LQR.....	57
Figura 26 – Diagrama de blocos do robô controlado pelo LQR.	59
Figura 27 – Ensaio Ziegler e Nichols.	59
Figura 28 – Resposta do sistema em laço aberto ao impulso.....	62
Figura 29 – Resposta ao impulso do sistema com controlador PID.....	63
Figura 30 – Resposta ao impulso do sistema com controlador LQR.	64
Figura 31 – Robô na posição inicial do teste de resposta ao degrau	65
Figura 32 – Robô na posição inicial do teste de resposta ao degrau	65
Figura 33 – Fluxograma do algoritmo do robô.....	67
Figura 34 – Resposta ao degrau do sistema formado por PID e Filtro Complementar.....	68

Figura 35 – Resposta ao degrau do sistema formado por PID e Filtro Kalman	69
Figura 36 – Resposta ao degrau do sistema formado por LQR e Filtro Complementar	70
Figura 37 – Resposta ao degrau do sistema formado por PID e Filtro Kalman	70
Figura 38 – Resposta ao degrau do sistema formado por PID e Filtro Kalman	72
Figura 39 – Sinais de ângulo estimado pelos métodos Filtro Kalman e filtro complementar..	73
Figura 40 – Sinais de ângulo estimado pelos métodos Filtro Kalman e filtro complementar..	74

LISTA DE TABELAS

Tabela 1 – Dados de ensaio do motor.....	42
Tabela 2 – Dados de ensaio do motor.....	44
Tabela 3: sensibilidades do giroscópio.....	49
Tabela 4: Erros do giroscópio.....	49
Tabela 5: sensibilidade do acelerômetro	49
Tabela 6: erros do acelerômetro	50
Tabela 7: Parâmetros de sintonia de PID Ziegler e Nichols.....	60
Tabela 8: Resumo das estatísticas dos sinais de controle.....	72
Tabela 9: Resumo das variâncias dos sinais de ângulo e do Filtro Kalman e Filtro Complementar	74
Tabela 10: Resumo das variâncias dos sinais de ângulo e velocidade angular	75

LISTA DE SIGLAS

AC	Corrente Alternada
LQR	Linear quadratic Regulator (Regulador Linear Quadrático)
LQG	Linear Quadratic Gaussian Control
PID	Proporcional Integral Derivativo
DC	Direct Current
IEM	Interferência Eletro Magnética

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVOS	13
2	REFERENCIAL TÉORICO	15
2.1	MODELO DO PÊNDBULO INVERTIDO	15
2.1.1	MODELO DO MOTOR DC.....	15
2.1.2	MODELO DA ESTRUTURA DO PÊNDBULO INVERTIDO	17
2.2	TEORIA DE ESTIMAÇÃO.....	23
2.3	FILTRO KALMAN.....	24
2.3.1	SISTEMA E MODELO DE OBSERVAÇÃO (ESTIMAÇÃO DO	
PROCESSO)		25
2.3.2	TEORIA E ALGORITMO	26
2.3.3	ORIGENS COMPUTACIONAIS DO FILTRO DE KALMAN.....	28
2.3.4	EQUAÇÕES DA ETAPA DE PREDIÇÃO	30
2.3.5	EQUAÇÕES DA ETAPA DE ATUALIZAÇÃO	30
2.3.6	INTERPRETAÇÃO DO FILTRO KALMAN	31
2.3.7	SINTONIA DO FILTRO	33
2.4	FILTRO COMPLEMENTAR	34
2.5	CONTROLE PID	36
2.5.1	O CONTROLADOR PID	36
2.5.2	O CONTROLADOR PID DIGITAL.....	37
2.5	O CONTROLADOR LQR	38
3	METODOLOGIA.....	41
3.1	MODELAGEM DO ROBÔ	41
3.1.1	MODELAGEM DO MOTOR	41
3.1.2	MODELAGEM DA ESTRUTURA	43
3.1.3	SISTEMA ELÉTRICO	45
3.1.4	MICROCONTROLADOR.....	45
3.1.5	SENSORES	47
3.1.6	ENCODERES	50
3.2	IMPLEMENTAÇÃO DE ALGORITMOS DE FUSÃO DE INFORMAÇÃO ...	51
3.2.1	FILTRO COMPLEMENTAR.....	52
3.2.2	FILTRO KALMAN.....	53
3.3	IMPLEMENTAÇÃO DE ALGORITMOS DE CONTROLE	56
3.3.1	LQR – LINEAR QUADRATIC REGULATOR	57
3.3.2	CONTROLADOR PID	59
4	RESULTADOS.....	62
4.1	SIMULAÇÕES.....	62
4.2	IMPLEMENTAÇÃO DO ROBÔ PÊNDBULO INVERTIDO	67
4.2.1	CONTROLE PID COM FILTRO COMPLEMENTAR.....	67
4.2.2	CONTROLE PID COM FILTRO KALMAN	69
4.2.3	CONTROLE LQR COM FILTRO COMPLEMENTAR	69

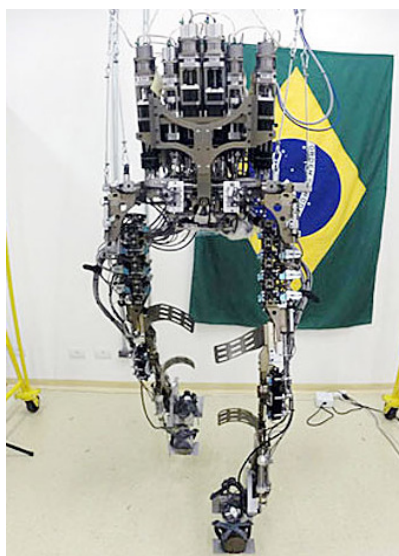
4.2.4	CONTROLE LQR COM FILTRO KALMAN	70
4.3	ANÁLISE DOS CONTROLADORES	71
4.4	ANÁLISE DOS MÉTODOS DE FUSÃO DE SENSORES	73
4.4.1	ANÁLISE DA FUSÃO DE INFORMAÇÃO DE ÂNGULO ENTRE O FILTRO KALMAN E O FILTRO COMPLEMENTAR	73
4.4.2	ANÁLISE DA FUSÃO DE INFORMAÇÃO DE VELOCIDADE ANGULAR ENTRE O FILTRO KALMAN E O FILTRO COMPLEMENTAR	74
5	CONCLUSÕES.....	76
	REFERÊNCIAS	77
	APÊNDICE - TABELA DE CUSTOS DO PROJETO	80

1 INTRODUÇÃO

Pêndulo invertido é uma estrutura onde o ponto de articulação fica abaixo do centro de gravidade. Este dispositivo além de ser um problema clássico na dinâmica, é usado em muitas aplicações de robótica. Assim, os aspectos científicos referentes à instrumentação e controle de estabilidade são de fundamental importância em diversas aplicações, pois o sistema é não linear e instável.

O pêndulo invertido é também o modelo de muitos dispositivos. Pode-se citar como exemplo desde o corpo humano, onde através da fusão das informações dos sentidos da visão, audição e propriocepção, o cérebro calcula o ângulo que o corpo faz com o solo e ativa os músculos para controlar o equilíbrio, balanço e a trajetória do corpo. O estudo do pêndulo invertido está diretamente relacionado com mais recentes avanços na área de exoesqueletos e próteses para amputados, paraplégicos e tetraplégicos. Trabalhos como o do Brasileiro Miguel Nicolelis usam tais conhecimentos.

Figura 1 - Exoesqueleto, trabalho do grupo do Neurocientista Miguel Nicolelis.



Além do corpo, o foguete também é um exemplo de pêndulo invertido pois sua parte ativa, que controla o ângulo formado com o plano da terra, está abaixo do seu centro de gravidade. O Segway PT (Segway Personal Transporting), figura (2), é outro ótimo exemplo de um objeto útil para as pessoas, que só existe graças a algoritmos de fusão de informação e

sistemas de controle aplicados ao problema de equilíbrio do pêndulo invertido. Estes algoritmos são capazes de fazê-lo ficar de pé e ainda seguir os controles de uma pessoa sendo carregada por ele.

Figura 2: Segway



Fonte: www.segway.com

1.1 OBJETIVOS

Neste trabalho serão analisados alguns métodos existentes de fusão de informação para cálculo do ângulo que o pêndulo forma com a vertical em conjunto com as metodologias de controle de estabilidade do pêndulo invertido. Esta análise visa estudar como os algoritmos se comportam com os métodos de controle da planta. Existem muitas soluções para este problema, algumas são estudadas pela área do conhecimento de fusão de informação, que visa retirar uma informação o mais confiável possível de um conjunto de dados com características diferentes. As técnicas que serão exploradas neste trabalho são Filtro Kalman e Filtros complementares, ambos com vantagens e desvantagens, e desempenho diferentes.

As metodologias de sistema de controle mais utilizadas para este fim são o PID (Proporcional Integral Derivativo) e o LQR (*Linear Quadratic Regulator*). Neste trabalho serão implementadas as duas para comparar os resultados.

Assim, os objetivos deste trabalho são:

- Medir o ângulo do pêndulo com a maior exatidão possível, dados os métodos de fusão de sensores estudados, filtro Kalman e filtro complementar, e comparar os seus resultados.
- Implementar o robô pêndulo, e manter o equilíbrio do protótipo utilizando os algoritmos de controle, PID e LQR, e comparar os seus resultados.

Neste documento no capítulo 2 serão abordados os aspectos teóricos do modelo do pêndulo invertido, dos algoritmos de fusão de sensores e sistemas de controle. No Capítulo 3 são descritos a implementação do robô e dos algoritmos. No capítulo 4 estão os resultados obtidos. E no capítulo 5 encontra-se a conclusão deste projeto.

2 REFERENCIAL TEÓRICO

2.1 MODELO DO PÊNULO INVERTIDO

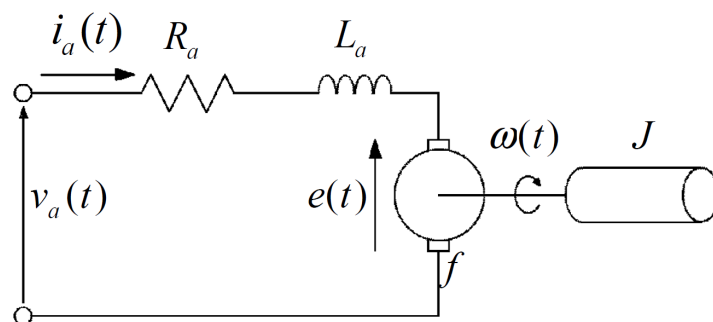
Para projetar um sistema de controle eficiente e um sistema de fusão de informações é necessário desenvolver o modelo matemático do robô que descreva seu sistema dinâmico. Esta modelagem pode ser feita tanto em função de transferência como no espaço de estados, neste trabalho foi escolhido a segunda. Para facilitar a desenvolvimento da modelagem dividiremos este capítulo em duas partes, primeira descrevendo o modelo do motor DC usado no projeto e a segunda descrevendo o modelo dinâmico do robô pendulo invertido propriamente dito.

2.1.1 MODELO DO MOTOR DC

A estrutura base do robô é formada pelas rodas e o corpo do robô pêndulo, como as rodas são acionadas por motores, o modelo matemático dos motores é fundamental para a modelagem da estrutura completa do robô. Este modelo descrevera no espaço de estados o motor DC e será usado na modelagem do robô pendulo invertido para fornecer uma relação entre a entrada de tensão dos motores e o controle de torque necessário para equilibrar robô. Esta modelagem foi feita baseada em [11].

A figura 3 mostra o diagrama esquemático do motor DC. Quando uma tensão é aplicada nos terminais do motor, a corrente $I_a(t)$ é gerada na armadura do motor.

Figura 3: Diagrama esquemático elétrico e mecânico do motor de corrente continua



Fonte: Baseado em [24]

Onde:

R_a e L_a = Resistência e Indutância da armadura

$I_a(t)$ = corrente na armadura

$V_a(t)$ = tensão aplicada na armadura do motor

$e(t)$ = força contra eletromotriz induzida no motor

$\omega(t)$ = velocidade angular do rotor

f = coeficiente de atrito viscoso

j = momento de inércia da carga acoplada ao motor

O motor produz um torque τ_m proporcional a esta corrente, esta relação é expressa conforme a eq (1), onde K_m é uma constante que depende de aspectos construtivos deste.

$$\tau_m = k_m i_a(t) \quad \text{Eq.(1)}$$

O resistor e o indutor em série com a fonte de tensão $e(t)$ são usados para modelar o circuito elétrico do motor. A força contra eletromotriz $e(t)$ é produzida porque as espiras do rotor estão em um campo magnético variável. A tensão gerada pode ser aproximada como uma relação linear com a velocidade do eixo, descrita na equação eq.(2), onde ω é a velocidade angular do eixo do motor.

$$e(t) = k_e \omega(t) \quad \text{Eq.(2)}$$

Assim, pode-se expressar o modelo elétrico do motor exposto na figura (3.1) através da lei das tensões de Kirchoff pela equação eq.(3) abaixo:

$$\sum \text{Tensões} = V_a(t) - R_a i_a(t) - L_a \frac{di(t)}{dt} - e(t) = 0 \quad \text{Eq.(3)}$$

Em seguida será analisado o modelo mecânico do motor, assim usando a equação do movimento rotacional aplicado ao motor é expresso na eq (4). Onde $f\omega$ representa o atrito viscoso e t_c representa o torque externo de perturbação e $j\dot{\omega}$ representa o torque resultante.

$$\sum \text{Torques} = \tau_m - t_c - f\omega(t) = j\dot{\omega} \quad \text{Eq.(4)}$$

Substituído as equações eq.(1) e eq.(2) nas equações do modelo elétrico e mecânico, aplicando a simplificação de que para modelos reais $L_a \approx 0$ e $f \approx 0$, assim a indutância do motor L_a e o atrito viscoso $f\omega$ serão desprezados nessa análise e rearranjando os termos, as equações eq.(5) e eq.(6) podem ser reexpressas como segue:

$$i_a(t) = -\frac{k_e}{R_a} \omega(t) + \frac{V_a}{R_a} \quad \text{Eq.(5)}$$

$$\frac{d\omega(t)}{dt} = \frac{k_m}{J} i_a(t) - \frac{\tau_c}{J} \quad \text{Eq.(6)}$$

Substituído eq.(5) na eq.(6):

$$\frac{d\omega(t)}{dt} = \frac{-k_m k_e}{JR_a} \omega(t) + \frac{k_m}{JR_a} V_a - \frac{\tau_c}{J} \quad \text{Eq.(7)}$$

A dinâmica do motor pode ser representada no espaço de estados, este é um sistema de equações diferenciais de primeira ordem, com parâmetros ângulo θ , e velocidade angular ω , como segue a equação eq.(8):

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-k_m k_e}{JR_a} \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_m}{JR_a} & -\frac{1}{J} \end{bmatrix} \begin{bmatrix} V_a \\ \tau_c \end{bmatrix} \quad \text{Eq.(8)}$$

$$y = [1 \quad 0] \begin{bmatrix} \theta \\ \omega \end{bmatrix} + [0 \quad 0] \begin{bmatrix} V_a \\ \tau_c \end{bmatrix} \quad \text{Eq.(9)}$$

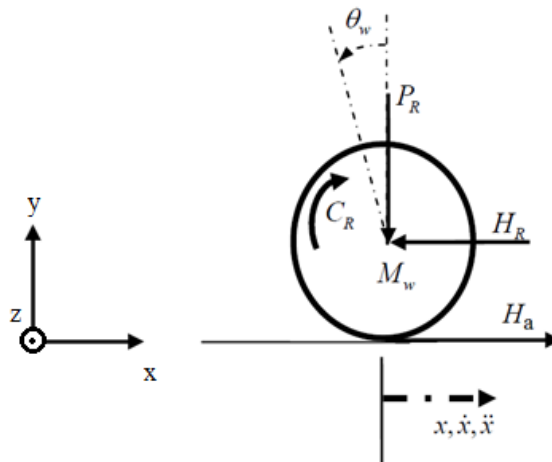
Também é interessante expressar o torque do motor em função dos parâmetros do motor e da tensão aplicada. Assim substituindo a eq.(5) na eq.(1)

$$\tau_m = \frac{k_m}{R_a} V_a - \frac{k_e k_m}{R_a} \omega(t) \quad \text{Eq.(10)}$$

2.1.2 MODELO DA ESTRUTURA DO PÊNDELO INVERTIDO

A estrutura do robô auto equilibrado pode ser dividida em duas partes, sendo a roda e o corpo do robô pêndulo. As rodas são coaxiais acionadas por dois motores independentes, o direito e o esquerdo. O robô é simétrico com relação ao lado direito e esquerdo, assim modelando um dos lados, para obter o modelo final deve-se espelhar este. Esta modelagem foi feita baseada em [11].

Figura 4: Diagrama de corpo livre da roda



Onde:

H_j = força de atrito

\ddot{x} = aceleração no eixo x

M_w é a massa da roda,

H_r é a força de reação entre o chassi e roda,

H_a é a força de atrito.

Usando a lei de Newton aplicada ao movimento linear e rotatório resulta:

$$\sum F_x = M\ddot{x} \therefore M_w\ddot{x} = H_a - H_r \quad \text{Eq.(11)}$$

$$\sum M_o = I\alpha \therefore J_w\ddot{\theta}_w = C_r - H_a r \quad \text{Eq.(12)}$$

Onde:

M_o = forças ao redor do centro da roda

J_w = Momento de inércia da roda

α = aceleração angular do eixo (rad/s^2)

$\ddot{\theta}_w$ é a aceleração angular da roda

C_r = torque do motor

r = raio da roda

Isolando a força de atrito e substituindo em eq.(11)

$$M_w \ddot{x} = \frac{C_r}{r} - \frac{J_w \ddot{\theta}_w}{r} - H_r \quad \text{Eq.(13)}$$

e considerando o torque da roda igual ao torque do motor ($C_r = \tau_m$), conforme a equação eq (10) e o momento inercial da roda como o momento da carga acoplada ao motor:

$$M_w \ddot{x} = \frac{k_m}{r R_a} V_a - \frac{k_e k_m}{r R_a} \dot{\theta}_w - \frac{J_w}{r} \ddot{\theta}_w - H_r \quad \text{Eq.(14)}$$

O movimento linear no eixo x é mais interessante, pois este será usado logo em seguida na dedução do chassi do robô pêndulo. Como o movimento linear age no centro da roda, a rotação angular pode ser transformada em movimento linear pela mudança de variáveis:

$$\ddot{\theta}_w r = \ddot{x} \Rightarrow \ddot{\theta}_w = \frac{\ddot{x}}{r} \quad \text{e} \quad \dot{\theta}_w r = \dot{x} \Rightarrow \dot{\theta}_w = \frac{\dot{x}}{r}$$

Assim a equação eq.(14) pode ser reescrita da seguinte forma:

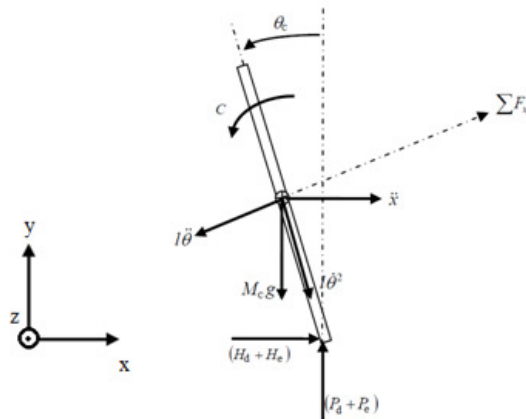
$$M_w \ddot{x} = \frac{k_m}{r R_a} V_a - \frac{k_e k_m}{r^2 R_a} \dot{x} - \frac{J_w}{r^2} \ddot{x} - H_r \quad \text{Eq.(15)}$$

Lembrando que este é o modelo para uma roda, sendo o modelo simétrico, o modelo da outra roda é análogo. Sendo assim o modelo para o robô inteiro é:

$$2 \left(M_w + \frac{J_w}{r^2} \right) \ddot{x} = \frac{2k_m}{r R_a} V_a - \frac{2k_e k_m}{r^2 R_a} \dot{x} - (H_{r_{direita}} + H_{r_{esquerda}}) \quad \text{Eq.(16)}$$

A seguir são descritas as equações do chassi do pêndulo. A figura 5 mostra o diagrama de corpo livre do chassi.

Figura 5: Diagrama de corpo livre da roda



Onde:

C = torque aplicado pelo motor na roda

J_p = momento de inércia do chassi

M_c = massa do chassi

θ_c = ângulo de rotação do chassi

g = aceleração da gravidade

H_d e H_e = força de reação entre as rodas e o chassi

l = distância do eixo de rotação até o centro de massa

Novamente, usando a lei de movimento de Newton, a soma das forças na direção horizontal é dada pela equação:

$$\sum F_x = M_c \ddot{x} \therefore (H_d + H_e) - M_c l \ddot{\theta}_c \cos \theta_c + M_c l \dot{\theta}_c^2 \sin \theta_c = M_c \ddot{x} \quad \text{Eq.(17)}$$

Assim:

$$(H_d + H_e) = M_c \ddot{x} + M_c l \ddot{\theta}_c \cos \theta_c - M_c l \dot{\theta}_c^2 \sin \theta_c \quad \text{Eq.(18)}$$

A soma das forças perpendiculares ao pêndulo

$$\sum F_{xp} = M_c \ddot{x} \cos \theta_p \quad \text{Eq.(19)}$$

$$(H_d + H_e) \cos \theta_c + (P_d + P_e) \sin \theta_c - M_c g \sin \theta_c - M_c l \ddot{\theta}_c = M_c \ddot{x} \cos \theta_c$$

A soma dos momentos ao redor do centro de massa do pêndulo

$$\sum M_o = J_c \alpha \quad \text{Eq.(20)}$$

$$-(H_d + H_e) l \cos \theta_c - (P_d + P_e) l \sin \theta_c - (C_d + C_e) = J_c \ddot{\theta}_c$$

O torque aplicado pelos motores no robô é descrito pela equação eq.(10). Aplicando a mudança de variável $C_d + C_e = 2\tau_m$ e rearranjando os termos, obtém-se:

$$[-(H_d + H_e) l \cos \theta_c - (P_d + P_e) l \sin \theta_c] = J_c \ddot{\theta}_c + \frac{2k_m}{R_a} V_a - \frac{2k_e k_m \dot{x}}{R_a r} \quad \text{Eq.(21)}$$

Neste ponto é interessante unir as equações de momento e de força, para isso multiplica-se a eq.(18) por $-l$, que resulta:

$$-(H_d + H_e) l \cos \theta_c - (P_d + P_e) l \sin \theta_c + M_c g l \sin \theta_c + M_c l^2 \ddot{\theta}_c = -M_c \ddot{x} l \cos \theta_c \quad \text{Eq.(22)}$$

e substituídos a equação eq.(20) na equação eq.(21)

$$J_c \ddot{\theta}_c + \frac{2k_m}{R_a} V_a - \frac{2k_e k_m \dot{x}}{R_a r} + M_c g l \sin \theta_c + M_p l^2 \ddot{\theta}_c = -M_c \ddot{x} l \cos \theta_c \quad \text{Eq.(23)}$$

E para eliminar a expressão $(H_d + H_e)$ da dinâmica dos motores, a eq.(17) é substituída na equação eq.(16),

$$2 \left(M_w + \frac{J_w}{r^2} \right) \ddot{x} = \frac{2k_m}{r R_a} V_a - \frac{2k_e k_m \dot{x}}{r^2 R_a} - M_c \ddot{x} - M_c l \ddot{\theta}_c \cos \theta_c + M_c l \dot{\theta}_c^2 \sin \theta_c \quad \text{Eq.(24)}$$

Ambas as equações acima são não lineares, porém podem ser linearizadas para valores pequenos de ângulo próximos a $\theta_p = \pi + \phi$, onde ϕ representa um ângulo pequeno a partir da direção vertical.

Assim, aplicando as seguintes relações de transformação de linearização:

$$\cos\theta_c = 1, \sin\theta_c = -\phi, e \left(\frac{d\theta_c}{dt} \right)^2 = 0, \text{ ou seja, } \ddot{\theta}_c = 0. \quad \text{Eq.(25)}$$

Rearranjando as eq. (24) e eq.(25) e aplicando a linearização, obtém-se as equações que descrevem o movimento do robô pêndulo.

$$(M_c l^2 + J_c)\ddot{\phi}_c + \frac{2k_m}{R_a} V_a - \frac{2k_e k_m}{r R_a} \dot{x} - M_c g l \phi = M_c l \ddot{x} \quad \text{Eq.(26)}$$

$$\frac{2k_m}{r R_a} V_a = \left(2M_w + \frac{2J_c}{r^2} + M_c \right) \ddot{x} + \frac{2k_e k_m}{r^2 R_a} \dot{x} - M_c l \ddot{\phi} \quad \text{Eq.(27)}$$

As variáveis de interesse para montar o modelo no espaço de estados são $\ddot{\phi}$ e \ddot{x} . Isolando essas variáveis das equações eq.(26) e eq.(27), pode-se montar as matrizes do modelo no espaço de estados, como segue:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2k_m K_e (M_c l r - J_c - M_c l^2)}{R_a r^2 \alpha} & \frac{M_c^2 g l^2}{\alpha} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2k_e k_m (r \beta - M_c l)}{R_a r^2 \alpha} & \frac{M_p g l}{M_w l^2 + J_c} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2k_m (I_c + M_c l^2 - M_c l r)}{R_a r \alpha} \\ 0 \\ \frac{2k_m (M_c l - r \beta)}{R_a r \alpha} \end{bmatrix} V_a \quad \text{Eq.(28)}$$

Onde:

$$\alpha = \left[I_c \beta + 2M_c l^2 \left(M_w + \frac{J_w}{r^2} \right) \right]$$

e

$$\beta = \left(2M_w + \frac{2J_w}{r^2} + M_c \right)$$

O modelo apresentado assume que as rodas estão sempre em contato com o solo e que não escorregam.

2.2 TEORIA DE ESTIMAÇÃO

Estimação é um processo estatístico de extração de informações sobre o valor de parâmetros a partir da medição de uma determinada grandeza. Em geral, os dados são amostras de uma medida e os parâmetros são características do sistema que se deseja estudar, por exemplo, medir a aceleração de um móvel para extrair, a partir das medidas, a velocidade do mesmo. Os valores medidos guardam uma relação com os parâmetros, isto é, a medição de aceleração de um móvel guarda uma relação com as variáveis posição e velocidade do mesmo.

Em geral os dados de medidas são corrompidos por imperfeições (ruídos) de informação. Em um contexto de engenharia, usualmente há interesse na utilização de sinais elétricos na saída de sensores, estes, por sua vez, apresentam imperfeições como: ruído elétrico, IEM (Interferencia Eletro Magnetica), Erro de Amostragem, erro de calibração do sensor, etc. [6]. Por estarem corrompidos por ruídos, os sinais na saída de sensores serão aqui considerados como variáveis aleatórias, estando constituídos por duas parcelas, uma deterministicamente relacionada a seus respectivos parâmetros e uma aleatória.

Um dos primeiros estímulos para o desenvolvimento de teoria de estimação foram os estudos astronômicos, no século XV, no qual o movimento dos planetas e cometas utilizava dados de medidas de telescópios rudimentares. O movimento desses corpos pode ser caracterizado por 6 parâmetros e o problema na época era estimar estes parâmetros a partir das medidas ruidosas. Para resolver este problema, Karl Friedrich Gauss propôs o método dos mínimos quadrados no seu trabalho “Theoria Motus Corpum Coelestium” [10], no qual descreve a base para o estudo de estimação. Gauss sugere que o valor mais apropriado para um parâmetro desconhecido é o valor mais provável. O valor da diferença entre o valor medido e valor computado é chamado de residual. Assim o método de mínimos quadrados se preocupa em determinar o valor mais provável dos parâmetros. Estes são os valores que minimizam a soma dos quadrados dos residuais [10]

Por exemplo, supondo que m medidas de uma variável amostrada nos instantes (t_1, t_2, \dots, t_n) são indicadas a cada instante como y_k , considerando que os parâmetros x são determinados a cada instante a partir dos dados amostrados e são relacionados pela seguinte equação (30):

$$z_k = H_k x_k + v_k \quad \text{Eq.(29)}$$

Onde v_k representa o erro de medição que ocorre a cada instante de tempo. Como pode ser visto na equação os parâmetros e os dados de observação são considerados linearmente relacionados pelo fator H_k que é a matriz de transição de estados. Tomando a estimativa de x baseada nas n amostras (z_1, z_2, \dots, z_n) como sendo \hat{x}_n . Então o valor residual associado com k ésima medida é dado pela equação (31)

$$r_k \triangleq z_k - H_k \hat{x}_n \quad k = 0, 1, \dots, n \quad \text{Eq.(30)}$$

O método dos mínimos quadrados consiste em determinar os valores mais prováveis de x , isto é, determinar \hat{x}_n . Este valor mais provável é tal que minimiza a soma dos quadrados dos resíduos. Assim, escolhendo x , tal que a função L_n eq. (32) seja minimizada.

$$L_n = \frac{1}{2} \sum_{k=0}^n [z_k - H_k X]^T W_k [z_k - H_k X] \quad \text{Eq.(31)}$$

Os elementos da matriz W_k são selecionados para indicar o grau de confiança de cada medida. Como será explicado melhor no filtro de Kalman W_k está relacionado com o inverso da matriz de covariância do ruído de medida.

2.3 FILTRO KALMAN

Em 1960, R. E. Kalman publicou seu famoso artigo [3] descrevendo uma solução recursiva para o problema de filtragem ótima aplicada a sistemas lineares no espaço de estados [2]. Originalmente desenvolvido para utilização na navegação espacial, o filtro de Kalman se revelou útil para muitas aplicações e é principalmente utilizado para estimar os estados do sistema que apenas podem ser observados indiretamente.

O filtro de Kalman é um método de estimação estocástica que combina modelos determinísticos e estocásticos de modo a obter estimativas ótimas de variáveis de estado de sistemas lineares [8] capaz de estimar variáveis de estado de um sistema linear a partir de medições incompletas ou ruidosas. Trata-se de um conjunto de equações matemáticas que fornece uma solução eficiente computacional (recursivo) do método dos mínimos quadrados. O filtro é muito poderoso em vários aspectos: ele suporta estimativas do passado, presente e estados ainda futuros, e pode fazê-lo mesmo quando a natureza precisa do sistema modelado é desconhecida. [2]

Esta ferramenta pode estimar as variáveis de uma ampla gama de processos. Em termos matemáticos, pode-se dizer que um filtro de Kalman estima os estados de um sistema linear. Este conjunto de equações, não só funciona bem na prática, mas é teoricamente atraente, porque pode-se demonstrar que de todos os possíveis filtros, ele é o que minimiza a variância do erro de estimativa. Filtros de Kalman são frequentemente implementados em sistemas de controle embarcados porque, para controlar um processo, é necessário primeiro uma estimativa precisa das variáveis deste processo [1].

2.3.1 SISTEMA E MODELO DE OBSERVAÇÃO (ESTIMAÇÃO DO PROCESSO)

O problema que o filtro Kalman procura resolver é a estimação de um conjunto de parâmetros que variam no tempo. Sendo assim, se aplica para estimar o estado de “ $x[n]$ ” a partir das medidas « $y[n]$ » de um sistema de tempo discreto que é descrito pelo sistema linear de equações estocásticas a seguir, também chamadas de equações de transição de estado:

Equação de estado:

$$x_{k+1} = Ax_k + Bu_k + w_k \quad \text{Eq.(32)}$$

Equação de saída:

$$y_k = Cx_k + z_k \quad \text{Eq.(33)}$$

Onde:

A é a matriz de transição de estado, é a matriz que relaciona um estado com o proximo

B é a matriz de transição das entradas de controle

C é a matriz de transição de saída

x_k é o vetor de estados do sistema;

u_k é o vetor de entradas determinísticas;

w_k é o vetor de variáveis aleatórias, isto é, perturbações nos estados do sistema, é chamado de ruído do processo, neste modelo serão considerados ruídos branco gaussiano;

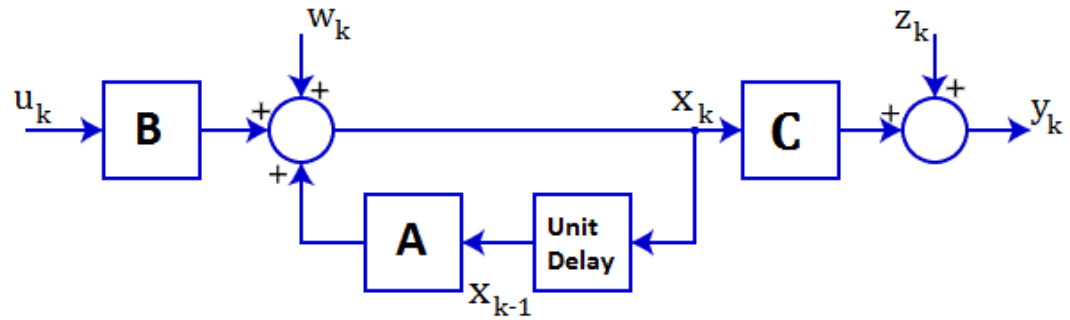
z_k é o vetor de variáveis aleatórias que perturbam a saída do sistema. Como a medida deste é feita na saída, este sinal é conhecido como ruído de medição, ou ruído de saída;

y_k é o vetor de saída, também chamado de vetor de medida;

k é o índice do tempo

Este modelo observado descrito pelas equações acima, pode ser representado pelo diagrama em blocos na figura (6).

Figura 6: Modelo do sistema do espaço de estados



Fonte: Baseado em [9]

A matriz A [$n \times n$] na equação de estado relaciona o estado no instante k com o estado no instante $k-1$, na ausência de entrada ou ruído de processo. A matriz B [$n \times 1$] relaciona a entrada de controle u para o estado x . E a matriz C [$m \times n$] na equação de saída relaciona o estado x com a saída y , sendo n o número de estados e m o número de saídas. [2] Para este trabalho, serão considerados apenas sistemas invariantes no tempo, assim as matrizes A , B e C são constantes, logo são as mesmas para todo k .

O vetor x contém todas as informações sobre o estado atual do sistema, mas não se pode medir x diretamente. Em vez disso, mede-se y , que é uma função de x que está corrompida pelo ruído z . Pode-se usar y para tentar obter uma estimativa de x , mas não podemos necessariamente obter a informação a partir de y pelo seu valor nominal, porque ele está corrompido por ruído. [1]

2.3.2 TEORIA E ALGORITMO

Com base no modelo do sistema dinâmico e nas medidas disponíveis de u e de y pretende-se estimar o estado x do sistema. Para obter o melhor estimador, este deve satisfazer a dois critérios:

1. Que o valor médio da estimativa do estado seja igual ao valor médio do verdadeiro estado. Ou seja, a estimativa não deve parecer tendenciosa. Matematicamente, diz-se que o valor esperado da estimativa deve ser igual ao valor esperado do estado. Para este objetivo, pode-se dizer que o Filtro Kalman é um estimador “não tendencioso” ou “unbiased”, isto é, $E[\hat{x}_k] = E[x_k]$. [1]
2. Deseja-se uma estimativa de estado que varia tão pouco quanto possível do valor do estado verdadeiro. Isto é, além da média da estimativa de estado ser igual à média do estado, também deseja-se um estimador com a menor variação possível

da estimativa de estado. Matematicamente, diz-se que deseja-se encontrar o estimador com a menor variância possível.

O filtro de Kalman se vale do estimador não tendencioso de variância mínima (*minimum variance unbiased estimator - MVUE*). Este estimador minimiza o erro médio quadrático, assim o valor estimado é aquele que minimiza a variância do erro:

$$\hat{x} = \operatorname{argmin}_{\hat{x}} E[\|x_k - \hat{x}_k\|^2]$$

O termo $E[\|x_k - \hat{x}_k\|^2]$, então chamado de *variância do erro*, é fortemente relacionado com a matriz de covariância do erro $E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$. Especificamente a variância do erro de um estimador é igual ao traço da matriz de covariância do erro.

$$E[\|x_k - \hat{x}_k\|^2] = \operatorname{trace}E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$$

O algoritmo de Kalman é um filtro linear de variância de erro mínima. O filtro de Kalman é o estimador que satisfaz os dois critérios estabelecidos anteriormente. Mas sua solução não se aplica a menos que sejam satisfeitas certas suposições sobre o ruído modelado nas equações de estado e saída. Lembrando o modelo de sistema, w é o ruído do processo e z é o ruído de medição, e assume-se que o valor médio de ambos é zero. Assume-se, ainda, que não existe correlação entre w e z , ou seja, a qualquer momento k , w_k , e z_k são variáveis aleatórias independentes. Também são consideradas ruídos brancos, ou seja, possuem espectro largo, e com distribuições normais de probabilidade dados por:

$$p(w_k) \sim N[0, Q_k]$$

$$p(z_k) \sim N[0, R_k]$$

Onde Q_k e R_k são as covariâncias de w_k e de z_k respectivamente. As matrizes de covariância dos ruídos Q_k e R_k são definidas como:

$$E[w_k w_l^T] = Q_k \delta(k - l) = \begin{cases} Q_k & k = l \\ 0 & \text{caso contrario} \end{cases}$$

$$E[z_k z_l^T] = R_k \delta(k - l) = \begin{cases} R_k & k = l \\ 0 & \text{caso contrario} \end{cases}$$

$$E[w_k z_l^T] = 0 \text{ para todo } k, l$$

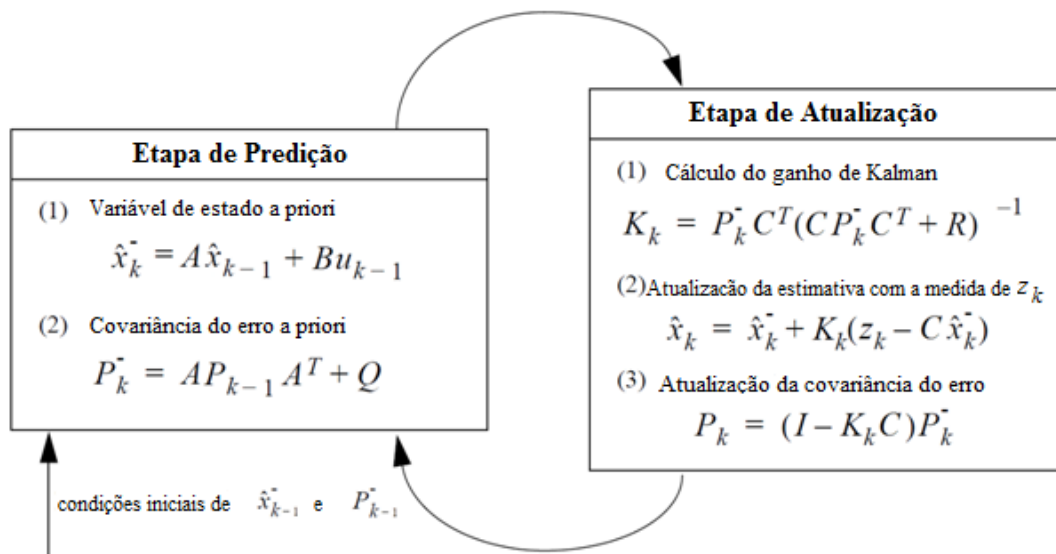
Onde w_k^T e z_k^T indicam a matriz transposta de w_k e z_k . Serão consideradas as duas variáveis aleatórias e independentes.

2.3.3 ORIGENS COMPUTACIONAIS DO FILTRO DE KALMAN

O Filtro Kalman estima o estado do processo utilizando uma arquitetura de controle realimentado como mostrado na figura (6). O filtro prediz o valor do estado e mede a saída (ruidosa) do sistema. Assim pode-se dividir as equações do filtro em duas: equações de predição e equações de atualização.

Para ilustrar o algoritmo de Kalman, será utilizada a figura (7), cujas condições iniciais são: $\hat{x}_{k-1} = \hat{x}_0$ e $\hat{P}_{k-1} = \hat{P}_0$. A fase de predição utiliza a estimativa do estado da iteração anterior para estimar o estado na iteração atual. Esta estimativa do estado atual com base em estados passados também é conhecida como a estimativa de estado *a priori*, porque embora seja uma estimativa do estado na iteração atual, não inclui informação das medidas da iteração atual, ou seja, nesta fase os valores de estados são “estimados” com base na dinâmica do sistema modelado e nos valores de estados anteriores.

Figura 7: Estrutura de equações do Filtro Kalman



Fonte: imagem baseada em [2].

Na fase de atualização ou correção, o estado a priori é combinado com informação de saída atual para refinar a estimativa do estado. Esta fase gera a estimativa do estado a posteriori, pois utiliza também a informação da observação da saída do sistema, ou seja, o valor do estado é “corrigido” com base na informação da observação da saída do sistema e da ponderação da variável ganho de Kalman, como será visto em seguida.

Inicialmente serão definidas algumas notações usadas no ponto de vista de estados. Define-se

$$\hat{x}_{k-1}$$

como sendo o estado do instante anterior estimado. O acento circunflexo na variável x significa estimativa, uma vez que não se pode observar o valor verdadeiro de x , apenas fazer-se uma estimativa.

Define-se também

$$\hat{x}_k^-$$

como sendo o estado a priori, significa a estimativa da matriz de estados no momento corrente k baseado nos estados prévios do sistema e nas estimativas dos estados anteriores à este. O sinal “-” sobre-escrito significa “a priori”, o que significa que a estimativa é feita a partir das estimativas anteriores do sistema.

Define-se também a variável

$$\hat{x}_k$$

como sendo o estado a posteriori, que é estimado no instante k com base nas observações de y_k e observações anteriores a esta. Estas notações se aplicam também às outras matrizes que serão expostas a seguir.

O vetor de estados iniciais, \hat{x}_0 é um vetor aleatório descorrelacionado com os ruídos w e z . A média e a matriz de covariância iniciais da variável estimada são:

$$\hat{x}_0 = E[x_0]$$

$$\hat{P}_0 = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$$

Pode-se definir também a estimativa de erro a priori e a posteriori como sendo:

$$e_k^- \equiv x_k - \hat{x}_k^- \quad \text{Eq.(34)}$$

$$e_k \equiv x_k - \hat{x}_k \quad \text{Eq.(35)}$$

E assim pode-se definir a matriz de covariância do erro a priori e a posteriori.

$$P_k^- = E[e_k^- e_k^{-T}] \quad \text{Eq.(36)}$$

$$P_k = E[e_k e_k^T] \quad \text{Eq.(37)}$$

2.3.4 EQUAÇÕES DA ETAPA DE PREDIÇÃO

Inicialmente a dedução das equações do filtro de Kalman da fase de Predição, neste passo a solução para minimização do valor médio quadrático do erro a posteriori é dado por

$$E[\|x_k - \hat{x}_k\|^2] = E[(x_k - \hat{x}_k)^T (x_k - \hat{x}_k)] \quad \text{Eq.(38)}$$

é quando a estimativa do estado a posteriori é o valor esperado do estado. Assim o estado predito é dado pela equação abaixo, onde usa-se o fato do ruído do processo ter média zero e que o valor de entrada u_{k-1} é conhecido.

$$\begin{aligned} \hat{x}_k &= E[x_{k+1}] \\ &= E[Ax_{k-1} + Bu_{k-1} + w_{k-1}] \\ &= A E[x_{k-1}] + B E[u_{k-1}] + E[w_{k-1}] \\ &= A \hat{x}_{k-1} + B u_{k-1} \end{aligned} \quad \text{Eq.(39)}$$

Para realizar a estimativa da matriz de covariância a priori P_k^- usa-se o critério do mínimo erro médio quadrático da estimativa do estado a posteriori, considerando que \hat{x}_k e w_k são descorrelacionados,

$$\begin{aligned} P_k^- &= E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T] \\ &= A E[(x_k - \hat{x}_{k-1})(x_k - \hat{x}_{k-1})^T] A^T + E[w_k w_k^T] \\ &= A P_k A^T + Q_k \end{aligned} \quad \text{Eq.(40)}$$

2.3.5 EQUAÇÕES DA ETAPA DE ATUALIZAÇÃO

Na fase de atualização, procura-se encontrar uma equação que estime o estado a posteriori \hat{x}_k como uma combinação linear da estimação do estado a priori \hat{x}_k^- e uma diferença ponderada entre a observação atual y_k e a saída predita estimada por $C\hat{x}_k^-$ (lembrando que $y_k = Cx_k + z_k$), como mostra a equação a seguir:

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - C\hat{x}_k^-) \quad \text{Eq.(41)}$$

A diferença $(y_k - C\hat{x}_k^-)$ na equação eq.(41) é chamada de resíduo. O resíduo mede a discrepância entre a saída medida y_k e a saída predita $C\hat{x}_k^-$. Isto significa que ele dá uma estimativa do ruído de medida z_k . A matriz K_k [n x m], também chamada de ganho K, é calculada para minimizar o valor da matriz de covariância do erro a posteriori P_k . Esta

minimização é alcançada fazendo as substituições da equação (41) na equação (35) e após na equação (37), realizando as operações indicadas, tem-se que:

$$P_k = (I - K_k C) P_k^- (I - K_k C)^T + K_k R_k K_k^T \quad \text{Eq.(42)}$$

Tomando a derivada do traço da matriz com relação à K e igualando a zero, e então resolvendo a equação para K (para mais detalhes [2][6][7]), tem-se que:

$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R}$$

$$K_k = P_k^- C^T (C P_k^- C^T + R)^{-1} \quad \text{Eq.(43)}$$

Este valor de K é tal que minimiza o erro médio quadrático da estimação assim chamada de Ganho Kalman. A matriz de covariância associada à estimativa ótima pode ser simplificada substituindo a equação (43) na equação (42). [7] como:

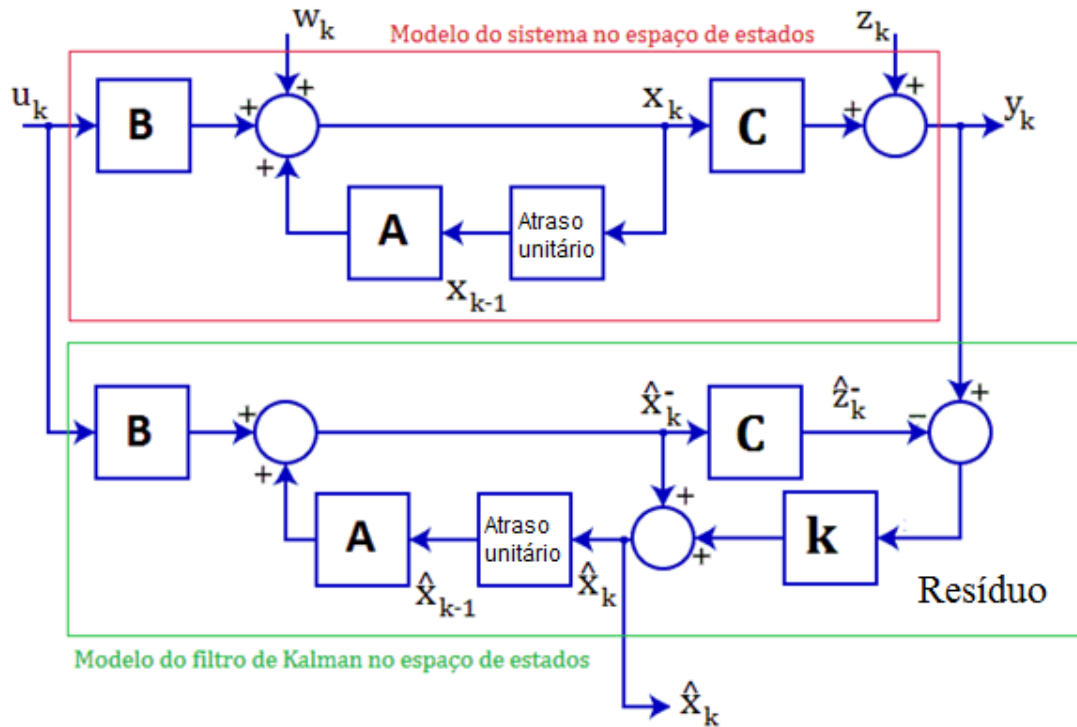
$$P_k = (I - K_k C) P_k^- \quad \text{Eq.(44)}$$

Porém, a expressão eq.(43) é válida para qualquer ganho K , ótimo ou sub-ótimo, enquanto a eq.(44) é válida apenas para o ganho ótimo.

2.3.6 INTERPRETAÇÃO DO FILTRO KALMAN

A figura (8) resume as etapas do algoritmo na forma de diagrama de blocos.

Figura 8: Estrutura no espaço de estados do filtro Kalman



Fonte: Baseado em [9]

O valor marcado no diagrama com “Resíduo” define a diferença entre a medida de Y_k e a medida predita de $C\hat{x}_k^-$, fazendo uso dessa informação disponível a cada instante k . O Residual é uma importante medida de quão bem a estimação está sendo realizada.

$$Residuo = y_k - C\hat{x}_k^-$$

A partir da equação (43) do ganho Kalman, pode-se observar que quanto menor for a covariância do ruído de saída R_k , o ganho K torna-se mais próximo de C_k^{-1} , ou seja, o valor do ganho K_k será tanto maior quanto menor for o ruído na medição R_k .

$$\lim_{R_k \rightarrow 0} K_k = C_k^{-1}$$

Analisando a equação do ganho K , eq.(43) percebe-se que quanto maior o ruído de saída Z_k , isto significa, maior o R e conseqüentemente K será menor, e a saída do filtro irá ponderar com menor peso a medida y e no cálculo do próximo \hat{x}^- . Por outro lado, se o erro de saída for pequeno, R será pequeno e K será maior, e o filtro dará maior credibilidade à medida y e no cálculo de \hat{x}^- . E ainda, quanto mais a covariância do erro a priori se aproxima de zero, o ganho K pondera o resíduo mais levemente.

$$\lim_{P_k^- \rightarrow 0} K_k = 0$$

Outro jeito de pensar sobre a ponderação de K é quanto mais a medida da covariância do erro R_k se aproxima de zero, mais confiável será a medida atual Y_k , enquanto que a medida predita $C\hat{x}_k^-$ será menos confiável. Por outro lado, quanto mais a covariância do erro do estado a priori P_k^- se aproxima de zero, pior será a estimativa, ou seja, menos confiável será o valor da medida atual Y_k , enquanto a medida predita $C\hat{x}_k^-$ será mais confiável.

2.3.7 SINTONIA DO FILTRO

O desempenho do filtro pode variar muito se os parâmetros não estão ajustados corretamente, portanto, serão expostas algumas características dos mesmos. Os parâmetros ajustáveis são:

1. A matriz de covariância inicial.
2. A estimativa do vetor de estado inicial.
3. A matriz Q e seu correspondente vetor w ruído aleatório.
4. A matriz R e a seu correspondente ruído aleatório vetor v .

A matriz de covariância inicial é a variação esperada do erro da estimativa do estado para o parâmetro correspondente e é representado pelos elementos da diagonal desta matriz. Para o filtro funcionar corretamente, os valores da matriz covariância precisam ser grande o suficiente, de tal modo que a diferença entre o estado inicial e a estimativa de estado inicial estejam na faixa que é admissível de acordo com a matriz de covariância. Se a estimativa de estado inicial é bastante precisa, então a matriz covariância apenas precisa permitir um pequeno erro. A estimativa do Estado vai demorar mais tempo para convergir como a matriz de covariância fica maior. A matriz P_0 diferente renderá uma magnitude característica transitória diferente, mas a sua duração será a mesma e as condições de estado estacionário não são afetadas.

Uma estimativa inicial do sistema deve estar disponível para o algoritmo do filtro funcionar. Não existe um requisito geral para a estimativa inicial ser precisa, desde que os valores iniciais da matriz Q sejam suficientemente grandes para o filtro funcionar corretamente.

A matriz Q representa a covariância do vetor erro do sistema w . Assume-se que os elementos de W são não correlacionados; portanto a matriz Q é uma matriz diagonal. Aumentar Q indica ruídos elevados no sistema dinâmico ou aumento da incerteza no modelo. Como resultado, o ganho K do filtro geralmente aumenta e as medições são ponderadas mais intensamente, isso é razoável, já que o aumento de Q indica que deve-se colocar menos confiança na saída do próprio modelo de dinâmica do filtro.

A matriz R representa a covariância da matriz de erros de medição v . Esta matriz indica o quão grande se espera que seja o erro de medição. Aumentar R indica que a medição está sujeita a um forte ruído e assim devem ser ponderados mais levemente pelo filtro. Este valor está ligado à variância do sensor.

2.4 FILTRO COMPLEMENTAR

Outra maneira de obter uma informação confiável de ângulo de inclinação é o filtro complementar. Este método leva em consideração as características dos dois tipos de sensores envolvidos: acelerômetros e giroscópio. A maioria dos sensores inerciais medem taxas, isto é, variações como velocidade ou aceleração, podendo ser linear ou angular. Usualmente as informações de variação são válidas sobre longo período de tempo. Elas devem ser integradas para obter o deslocamento, porém pequenos erros estáticos na medida causam crescimento ilimitado na grandeza integrada. A seguir, são explicadas as características dos sensores envolvidos no sistema do pêndulo.

O problema com os acelerômetros:

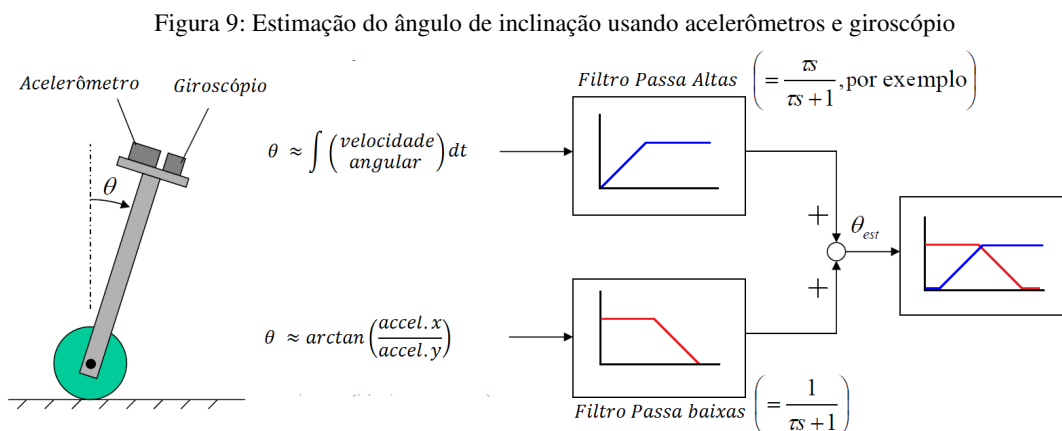
O ângulo é calculado pela função arco tangente do valor da razão dos sinais dos acelerômetros. Como estes medem todas as acelerações que estão agindo no objeto, eles vão medir muito mais que simplesmente a aceleração referente à gravidade. Até mesmo as pequenas forças agindo no pêndulo irão perturbar o cálculo de ângulo. No caso do pêndulo, este tem a tendência a oscilar em torno da posição de equilíbrio, logo a medida de ângulo estará comprometida pelas acelerações causadas por esta oscilação. Assim, o sinal de ângulo apresenta um ruído aditivo, que pode-se perceber que é de alta frequência, ou pelo menos possui frequência acima da que se espera.

O problema com os giroscópios:

Sabendo que o giroscópio mede velocidade angular, o valor do ângulo é calculado pela integração dos valores de giroscópio ao longo do tempo. A medição através desta técnica tem a tendência de flutuar, e não retornar a zero quando o sistema volta à sua posição original. Isto se deve às imperfeições do sensor que causa um erro na medida e, como as amostras são somadas, o erro se acumula, causando *drift*, isto é, a medida de ângulo diverge com o tempo. Porém a medida no instante inicial é confiável. Assim, pode-se concluir que o ruído da medida de ângulo causado pelo processo de integração possui frequência baixa.

O filtro complementar é concebido sem qualquer referência ao filtro Kalman, embora esteja relacionado a ele. O projeto do filtro complementar não considera qualquer descrição estatística do ruído que corrompe os sinais, e o seu filtro é obtido por uma simples análise no domínio da frequência.

A estrutura de filtro complementar é semelhante ao resultado final do filtro de Kalman porém sem o ganho Kalman, que determina estatisticamente qual dos sinais é mais confiável, e no filtro complementar, este valor está relacionado com a frequência de corte dos filtros. A ideia básica da estimação do ângulo de inclinação usando acelerômetros e giroscópio está exposta na figura (9).



Fonte: Baseado [18]

2.5 CONTROLE PID

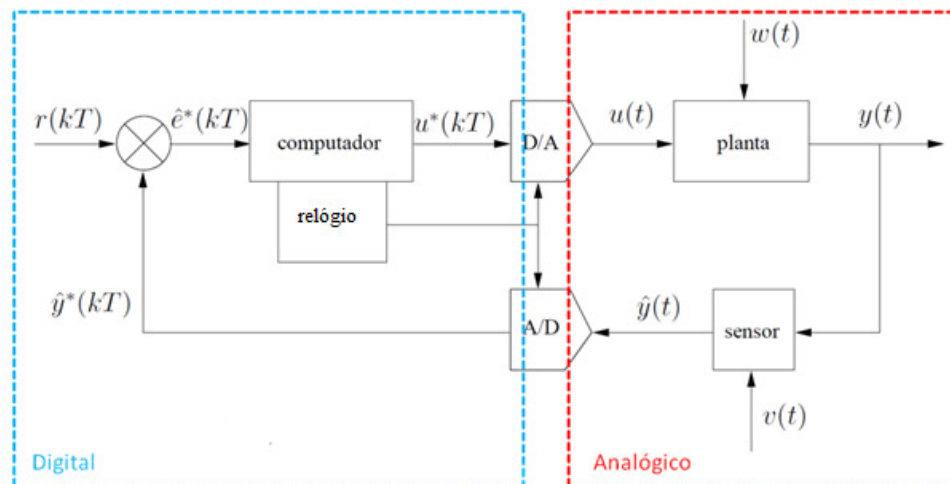
Os processos industriais são controlados desde o século 18, quando em 1788, James Watt desenvolveu seu famoso aparato para controlar a velocidade da máquina a vapor. Porém o primeiro instrumento de controle proporcional, integral e derivativo foi desenvolvido nos

anos 40 pela Taylor Company. Apesar de antiga, esta tecnologia não mudou muito e preserva os princípios da máquina de Watt. Os principais elementos de um sistema de controle são a planta, que é o processo ou o dispositivo que se deseja controlar; o sensor, o qual permite a leitura da variável que se deseja controlar, e permite a realimentação. Ainda o valor de referência, que é o valor que se deseja que a planta assuma, além de perturbações e, o mais importante, o controlador, que tem a função de conectar os elementos físicos da planta com os valores de referência, ou seja, ele atua na planta afim de que esta responda da maneira que se deseja. [19]

2.5.1 O CONTROLADOR PID

A figura (10) mostra a topologia típica de um sistema de controle digital. Mesmo sendo digital, este sistema é formado por uma parte analógica. O valor T é o intervalo de amostragem, ou seja, a frequência de amostragem é $f_s = 1/T$. A principal função do controlador é tornar $\hat{e}^*(kT) = 0$ ou o mais próximo de zero possível e para isso, o valor de referência $r(kT)$ deve ser igual a $\hat{y}^*(kT)$ ou tornar-se o mais próximo disso.

Figura 10: Sistema de controle digital



Fonte: Baseado em [7]

É interessante notar as unidades de medida que cada sinal representa: **Sinal de realimentação** $\hat{y}(t)$: este é o próprio sinal de saída, medido pelo sensor e digitalizado, porém continua representando a mesma grandeza. No caso do robô pêndulo, a **variável de processo** é o ângulo que o robô faz com a vertical, também chamado de inclinação θ . **Referência**: representa o valor desejado da variável do processo, $y(t)$, ou seja, é interpretado como sendo a mesma grandeza. Neste caso representa θ . A **variável de controle**, ou saída do controlador é

aplicada à planta com a intenção de fazê-la atingir um determinado comportamento. Neste caso é aplicada uma tensão e a planta transforma em ângulo. Assim, o sinal de controle representa V/θ . O sinal de **erro** é resultado da diferença entre o sinal de referência e de saída, ambos representam a grandeza θ .

A variável de processo é o grande objetivo final do controle, e as características deste sinal serão as métricas de avaliação do sistema controlado. Esta característica principal usada neste trabalho será o erro ($\hat{e}^*(KT)$), pois o pêndulo invertido tem a tendência a oscilar em torno da vertical ao longo do tempo em um equilíbrio instável.

2.5.2 O CONTROLADOR PID DIGITAL

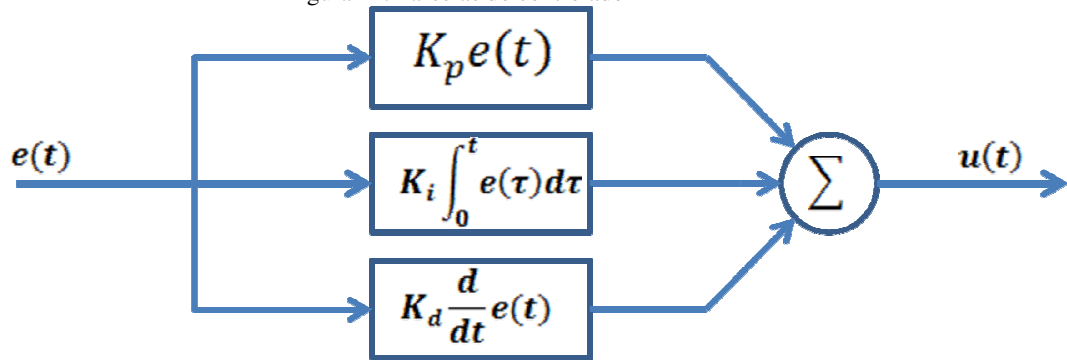
O controle PID foi pensado inicialmente como um dispositivo analógico, e possui a seguinte expressão matemática:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad \text{Eq.(45)}$$

O sinal de controle do controlador PID é formado pela soma da saída de três blocos em paralelo, os quais podem ser equacionados separadamente. As parcelas Proporcional, Integral e Derivativa têm com entrada o sinal de erro, ou seja, o valor de referência menos o valor da realimentação dado pelo sensor.

Para ser implementado de forma digital, a equação diferencial (45) deve ser transformada em uma equação a diferenças. Existem vários métodos de aproximação, como o *forward difference*, *backward difference*, entre outros. Neste caso, será utilizada a aproximação de Tustin [25], também conhecido como método bilinear ou método dos trapézios. Assim a eq.(45) se transforma na equação eq.(46). Esta equação é possível de ser implementada em um microcontrolador ou FPGA

Figura 11: Parcelas do controlador PID



Fonte: Própria

$$u[n] = K_p e[n] + K_i \left(u[n-1] + \frac{T}{2} (e[n] + e[n-1]) \right) + K_d \left(\frac{2u[n] - u[n-1]}{T} - e[n-1] \right) \quad \text{Eq.(46)}$$

$$n = kT, \forall k = 0, 1, 2 \dots$$

2.6 O CONTROLADOR LQR

O controle LQR é uma técnica de controle moderna no espaço de estados para projeto de reguladores dinâmicos ótimos [Richi Chi ooi]. Esta técnica faz parte de uma área de sistemas de controle chama “controle ótimo. O regulador proporciona uma lei de controle ótima para um sistema linear com índice de desempenho quadrático.[17]

Dado o sistema no espaço de estados conforme conjunto de equações:

Equação de estado:

$$\dot{x}(t) = Ax(t) + Bu(t) + w \quad \text{Eq.(47)}$$

Equação de saída:

$$y(t) = Cx(t) + z \quad \text{Eq.(48)}$$

Onde:

x é o vetor de estados do sistema;

u é o vetor de entradas determinísticas;

w é o vetor de variáveis aleatórias que perturbam o estado do sistema, é chamado de ruído do processo, são as incertezas do sistema ou ruídos que agem sobre o sistema;

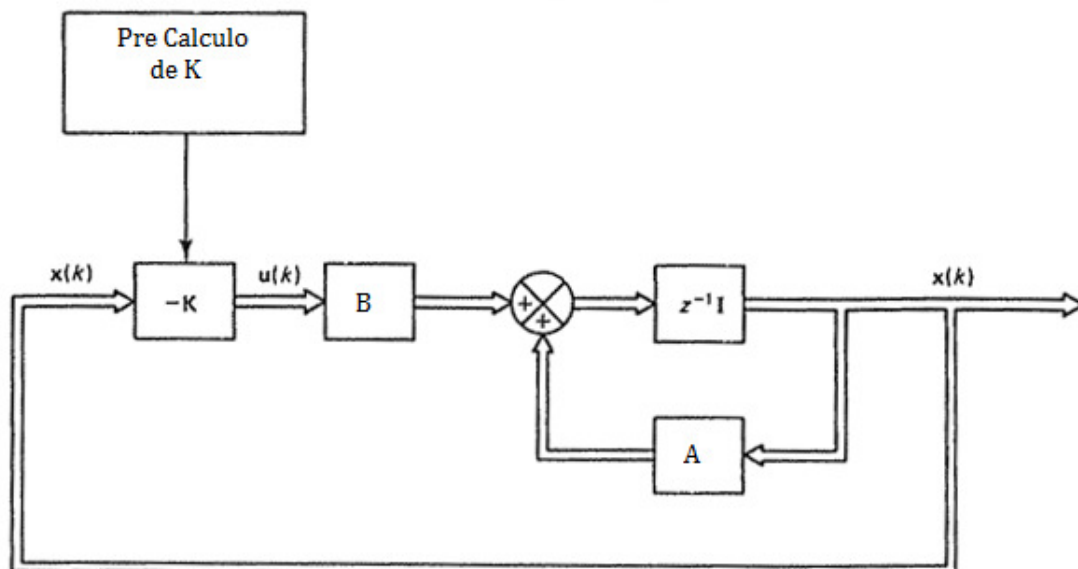
z é o vetor de variáveis aleatórias que perturbam a saída do sistema. Como a medida deste é feita na saída, este sinal é conhecido como ruído de medição, ou ruído de saída;

y é o vetor de saída, também chamado de vetor de medida;

Este modelo é semelhante ao descrito para o filtro de Kalman, porém está descrito para tempo contínuo. Para sua implementação será necessário um processo de discretização da planta que será descrito no capítulo 4.

O princípio de funcionamento do controlador está mostrado na figura 10. Onde mostra a planta modelada pelas matrizes A, B e C esta representada pelo sistema em blocos e o controlador pelo bloco $-K(k)$.

Figura 12: Estrutura de controle LQR



Fonte: OGATA, Katsuhiko, Discrete-time Control Systems

Três suposições são feitas para que o controlador LQR seja implementado:

- 1) Todas as variáveis de estado $x(t)$ estão disponíveis para realimentação, isto é, podem ser medidas por sensores.
- 2) O sistema é estabilizável, isto significa que todos os modos instáveis são controláveis.

3) O sistema é detectável, possuindo todos os modos instáveis observáveis.

Para verificar que o sistema é controlável e observável é possível utilizar as funções $obsv(A,C)$ e $ctrb(A,B)$ no Matlab.

A lei de controle LQR baseia-se na minimização da equação de custos:

$$J = \int_0^{\infty} x^T(t) Qx(t) + u^T(t)Ru(t)dt \quad \text{Eq.(49)}$$

Onde Q e R são parâmetros de ponderação que penalizam os estados e a variável de controle respectivamente. Assim, estas matrizes são os parâmetros de sintonia do controlador. A matriz Q é uma matriz diagonal principal, onde cada componente representa a penalização de cada um estado do sistema. R é uma matriz quadrada de tamanho igual ao numero de saídas do controlador, logo normalmente é um escalar, e penaliza a saída do controlador.

Em função do índice quadrático de desempenho da função de custo, o sistema tem uma solução matemática que leva à seguinte “lei de controle ótima”

$$u(t) = -Kx(t) \quad \text{Eq.(50)}$$

Onde $u(t)$ é o sinal gerado pelo controlador e K é o o vetor de ganhos calculados pela equação:

$$K = R^{-1}B^T S \quad \text{Eq.(51)}$$

Onde S pode ser encontrado pela solução da equação algébrica de Riccati.

$$SA + A^T S - SBR^{-1}B^T S = 0 \quad \text{Eq.(52)}$$

O vetor K pondera a realimentação de cada variável de estado que realimenta o sistema. As matrizes Q e R fazem a ajuste relativo à importância de cada variável de estado na função de custos (J) eq.49.

3 METODOLOGIA

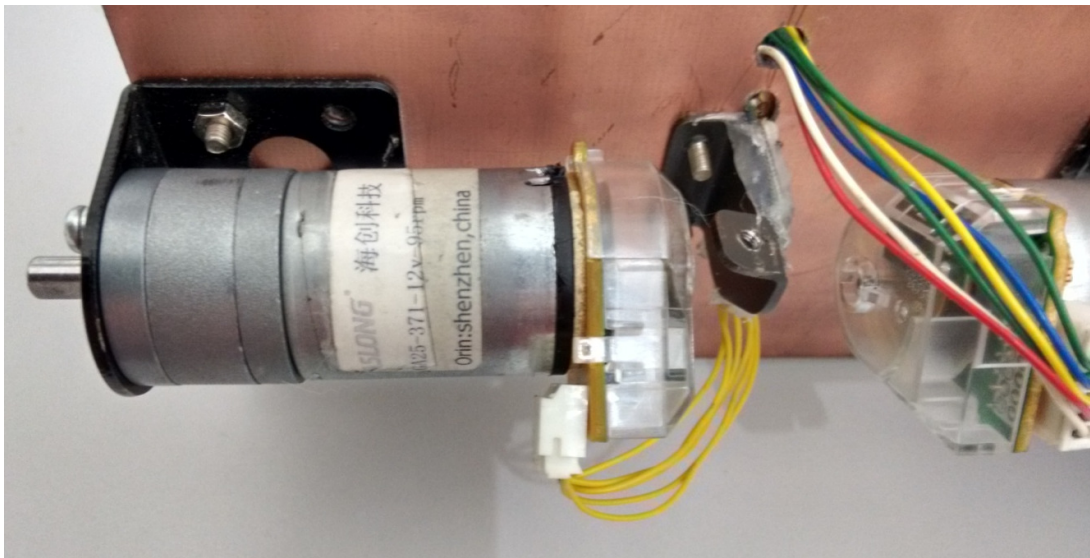
Neste capítulo, será feito um estudo detalhando os principais componentes do robô, bem como seus aspectos construtivos, a modelagem matemática e os algoritmos aplicados no desenvolvimento neste trabalho. Ainda serão apresentados os modelos elétricos e mecânico do dispositivo, a descrição dos componentes utilizados e o estudo dos algoritmos de fusão de informação e sistema de controle em malha fechada utilizado neste projeto.

3.1 MODELAGEM DO ROBÔ

3.1.1 MODELAGEM DO MOTOR

A escolha do motor é crítica para controlar o robô. Para que permaneça na posição vertical, o motor deve ter torque e velocidade de rotação adequada ao sistema. Caso o torque seja baixo, o motor não se sustentará. E caso a velocidade seja muito baixa este apresentará comportamento errático, oscilando, mas sem conseguir estabilizar na posição vertical. Na figura (13) é apresentada a imagem do motor fixo na estrutura.

Figura 13 - Partes de do robô pêndulo.



Fonte: Própria

O motor escolhido para o projeto foi o JGA25-371-12V-431RPM, pois possui encoder e conjunto de engrenagens para reduzir a velocidade na razão de 1:9,28. A folha de dados do

motor apresenta os dados de ensaios mostrados na tabela (1) abaixo. Segundo o folha de dados, todos os ensaios foram realizados com a tensão de operação do motor de 12V.

Tabela 1 – Dados de ensaio do motor.

	Ensaio Sem Carga	Ensaio Com Carga	Ensaio à Rotor Bloqueado
Corrente (mA)	46	300	1000
Rotação (RPM)/(rad/s)	463 / 48.485	370 / 38.746	0
Torque (Kgcm)(Nm)	-	0,23 / 0,0225	1,1 / 0,108

Fonte: folha de dados do Motor.

O valor de resistência interna do motor, pode ser obtido de duas formas: pelo ensaio à rotor bloqueado, ou pela simples medição com o ohmímetro. Do ensaio com rotor bloqueado realizado pelo fabricante consta o valor de 12Ω e com o ohmímetro foi medido 10.8Ω em um motor e 11.0Ω no outro. Optou-se por utilizar o valor médio das resistências medidas de 10.9Ω , visto que se trata do componente utilizado no projeto, e a folha de dados foi feita com ensaio de amostra de vários motores, e seus dados podem variar de um lote para outro.

Os parâmetros de coeficiente elétrico K_e e coeficiente mecânico K_m foram calculados a partir dos dados do catálogo do motor. O K_e calculado como segue a equação

$$V_a = R \cdot i + K_e \cdot \omega$$

Onde:

V_a é a tensão aplicada no motor

R é a resistência do motor, $10,9\Omega$, neste caso

i corrente que circula no motor

K_e é o coeficiente elétrico, como visto no capítulo anterior

ω é a velocidade angular do motor.

A partir dos dados do ensaio do motor com carga, obtem-se o valor de $K_e = 1.2339[\text{Vs/rad}]$. O K_m é calculado na equação:

$$\tau_m = K_m \cdot i$$

Onde:

τ_m é o torque que o motor imprime

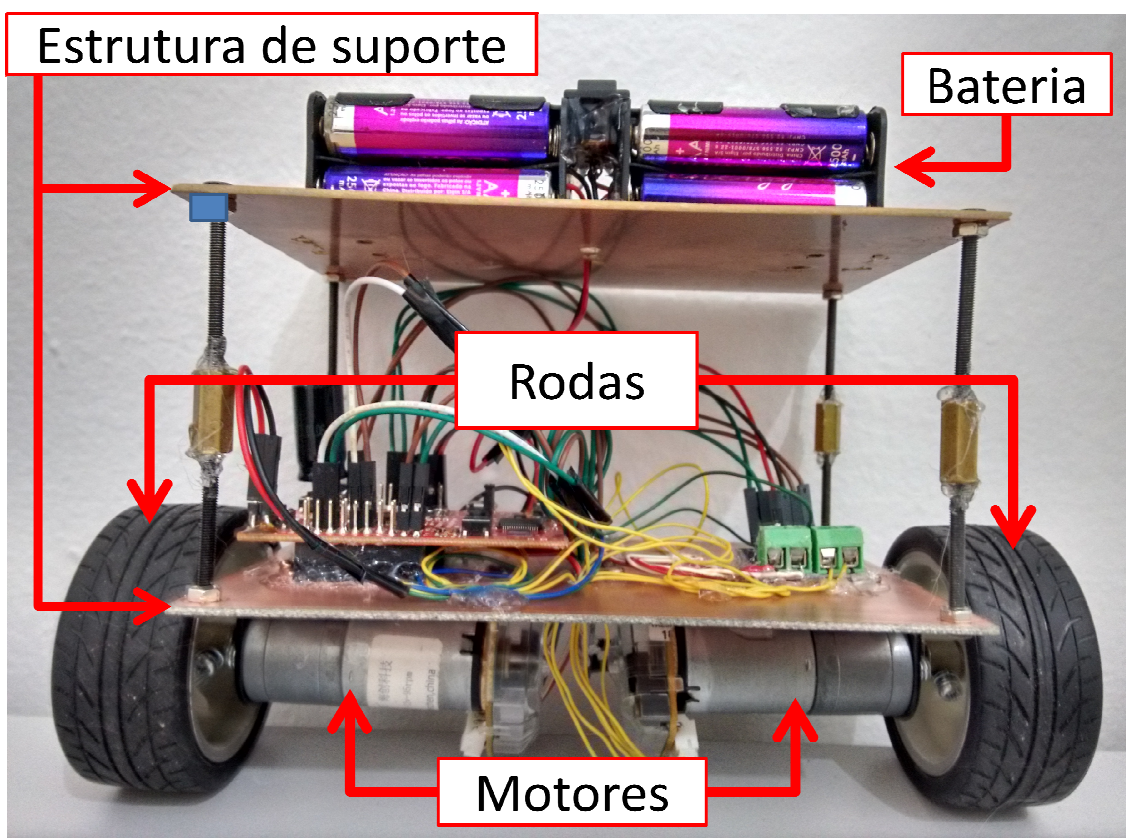
K_m é o coeficiente mecânico;

Utilizando-se os dados do ensaio com rotor bloqueado para o cálculo, obtém-se $k_m = 0.114$ [Nm/A]

3.1.2 MODELAGEM DA ESTRUTURA

A estrutura mecânica do robô pêndulo invertido desenvolvido neste trabalho é constituída de quatro partes principais, como mostrado na figura (10) abaixo, as quais são: rodas, motores, bateria e estrutura de suporte.

Figura 14 – Robô pêndulo montado.



Fonte: Própria

A estrutura do robô foi pensada para ser simples, constituída de duas placas de fibra de vidro de tamanho 100x150mm, separadas de 75 mm. Estas placas são as mesmas que são comumente utilizadas para confecção de circuitos impressos.

Na parte inferior, foi fixada a placa do microcontrolador Tiva TM4C123G, a placa de acionamento dos motores e, embaixo, os motores. Na placa superior foi colocado o conjunto de baterias.

A modelagem matemática do robô inspirado no trabalho de [11] descrita no capítulo de referencial teórico, requer que se conheçam diversas medidas de aspectos construtivos do robô, tais como peso e dimensões de cada uma das suas partes. Para tanto, cada peça foi pesada e medida, para o levantamento dos parâmetros e cálculo dos momentos de inércia.

Cada parte foi modelada como uma peça simples para o cálculo do momento de inércia. Ao final, o momento de inércia do robô pêndulo invertido é o somatório dos momentos de inércia de suas partes.

O motor e as rodas foram modelados como cilindro, que gira no seu próprio eixo. A bateria foi modelada como paralelepípedo e os suportes foram modelados como placas delgadas. No caso das pilhas e do suporte, estes giram ao redor de um eixo que passa fora do seu centro, e para calcular o momento de inércia foi utilizado o teorema de Steiner. Esse teorema consiste em uma correção no momento de inércia de um corpo quando o eixo de rotação é paralelo e deslocado ao eixo central da peça.

Assim para o robô construído neste projeto, o cálculo dos parâmetros resultou na seguinte tabela:

Tabela 2 – Dados de ensaio do motor.

Parâmetro	Valor	Unidade
Momento de inércia da Roda	1.90125e-04	Kg. m ²
Momento de inércia do pêndulo	4.4317e-3	Kg. m ²
Massa do pêndulo	0.655	Kg
Massa da Roda	0.090	Kg
Raio da roda	0.065	m
Distancia até o centro de massa	0.06064	m

Fonte: Própria

A partir destes valores foram calculados os coeficientes das matrizes A e B que constituem o modelo do pêndulo invertido.

Equações do modelo

$$\dot{x} = A \cdot x + B \cdot u$$

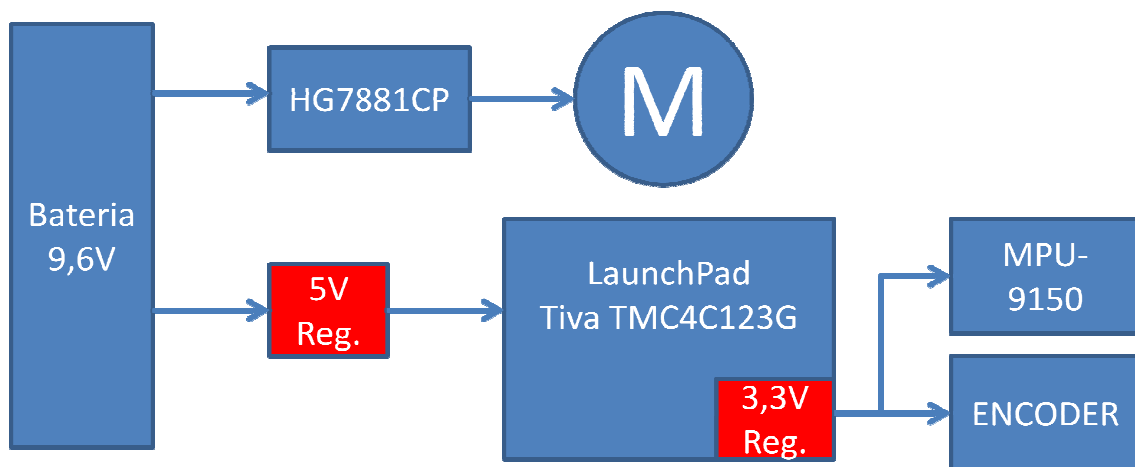
$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.6565 & 3.3331 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 3.2326 & 77.6204 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1894 \\ 0 \\ -0.9326 \end{bmatrix} \cdot Va$$

3.1.3 SISTEMA ELÉTRICO

O robô é alimentado a partir de um conjunto de oito baterias recarregáveis de tensão nominal de 1.2 V cada, num total de cerca de 9,6V. A bateria alimenta diretamente a etapa de potência dos motores e a alimenta a placa do microcontrolador através de um regulador de 5V, 7805. O microcontrolador funciona com tensão de 3.3V e o kit de desenvolvimento Tiva launchpad possui embutido um regulador para alimentar o microcontrolador.

Os sensores MPU-9150 e os encoders são alimentados pelo regulador 3,3V presente na placa de desenvolvimento, como mostra a figura(15).

Figura 15 – diagrama de alimentação.

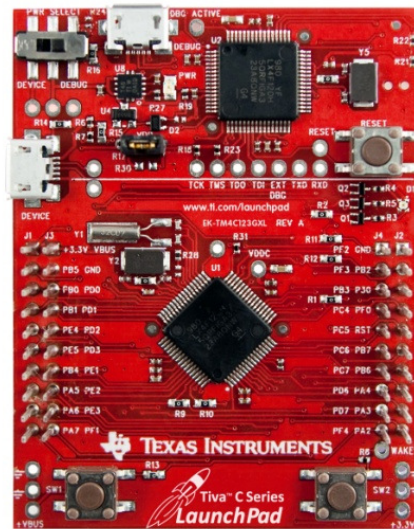


Fonte: .Própria

3.1.4 MICROCONTROLADOR:

Neste projeto foi utilizado um kit de desenvolvimento da Texas Instrument Launchpad, com microcontrolador Tiva TM4C123G. Este microcontrolador conta com canal de comunicação I2C utilizado na comunicação com o sensor; 16 canais PWM, utilizando quatro, sendo dois para cada motor, para que seja possível fazer a reversão de rotação do motor; e três canais de QEI (Quadrature Encoder Interface) utilizando 2 na leitura do encoder. A figura(16) mostra a placa Tiva Launchpad utilizada no projeto.

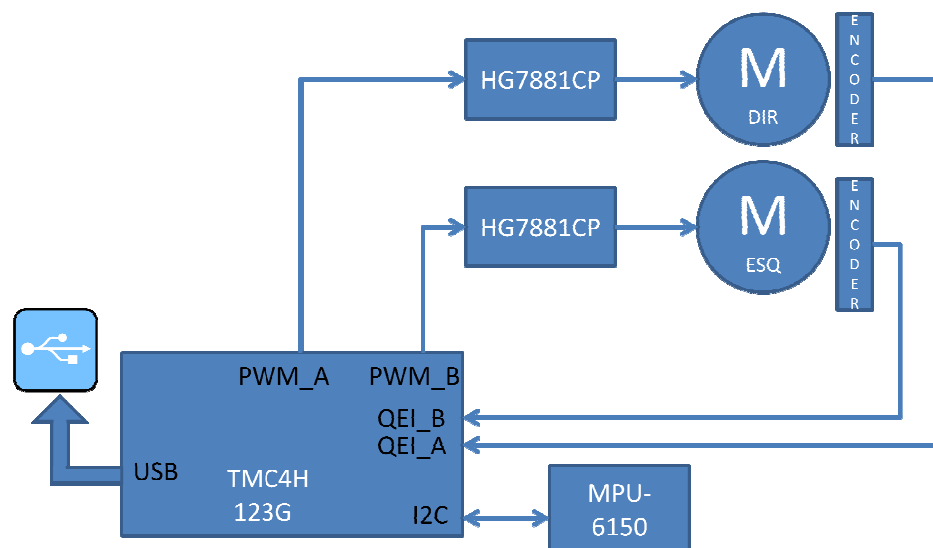
Figura 16 – Kit de desenvolvimento Launchpad da Texas Instruments.



Fonte: Texas Instruments

A figura (17) mostra o diagrama de blocos dos periféricos do robô pêndulo invertido. Nele consta o CI MPU9150, que contém o giroscópio e acelerômetros e envia os dados via I2C; o CI HG7881CP que contém a etapa de potência em ponte H para acionamento de cada um dos motores e estão conectados em canais PWM do microcontrolador; os encoders, cujos sinais são decodificados pelo periférico QEI embutido no processador e a interface USB utilizada para debug do projeto. Todos estes componentes são conectados diretamente ao microcontrolador na placa de desenvolvimento Tiva LaunchPad.

Figura 17 – Diagrama de blocos do projeto.



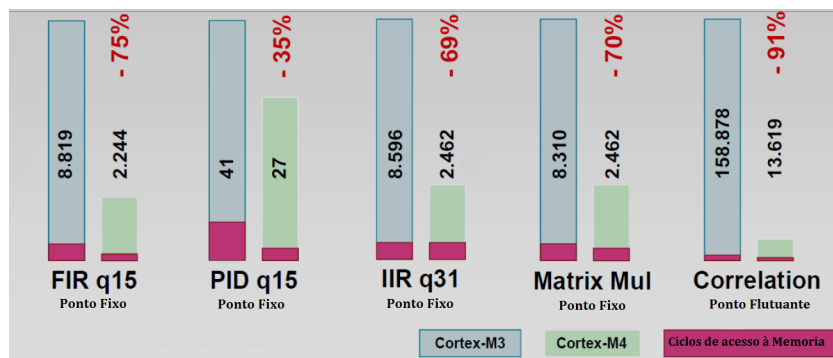
Fonte: .Própria

Além dos aspectos de interface com outros periféricos do sistema, este processador conta com Núcleo ARM Cortex M4 de 32 bits, com frequência de operação de até 80MHz com capacidade de até 100 MIPS (*mega instructions per second*), o que o torna capaz de processar todos os sinais em tempo de uma amostra de 0,002segundo, dado que a frequência de amostragem foi de 500Hz. Para tanto foi utilizado um *timer* do microcontrolador para garantir a frequência de amostragem.

A linha de núcleos Cortex® M4F® da ARM® possui como principal avanço o processador anexo de ponto flutuante (*Floating Point Unit – FPU*) capaz de realizar multiplicação de números em ponto flutuante em 1 ciclo de máquina e divisões em 14 ciclos de máquina. No projeto esta unidade é utilizada para o cálculo do arco tangente dos acelerômetros.

Este hardware possui desempenho muito superior em comparação com o modelo antecessor, como mostra a figura (14) a seguir. Neste gráfico estão expostos os tempos médios de execução (medidos em ciclos de máquina) das principais rotinas mais dispendiosas de tempo de processamento, tais como PID, filtros e multiplicação de matrizes.

Figura 18 – Comparativo de performance entre Cortex M3 e M4+FPU.



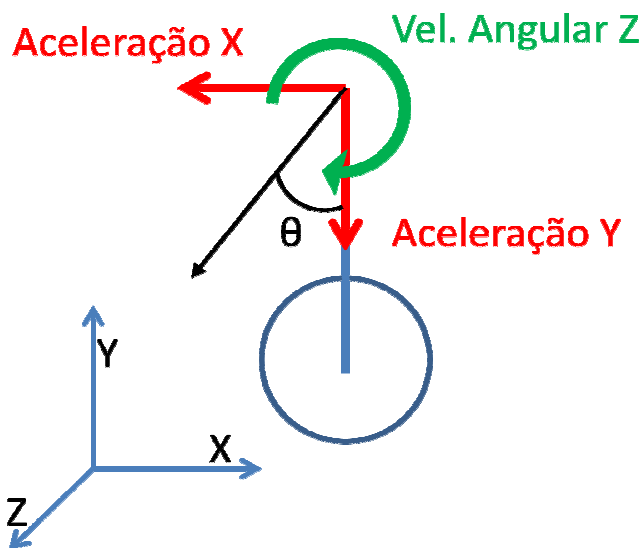
Fonte: .Texas Instruments

Este microcontrolador foi escolhido por possuir todos os hardwares necessários ao projeto ainda apresentar baixo custo. A placa de desenvolvimento com gravador embutido, botões e LED's de debug custa cerca de U\$13,00.

3.1.5 SENSORES

Três sensores foram utilizados para o cálculo do ângulo teta com relação a vertical, sendo dois acelerômetros e um giroscópio. Estes foram montados segundo a geometria mostrada na figura (15) abaixo.

Figura 19 – Disposição geométrica dos sensores inerciais.



Fonte: Própria

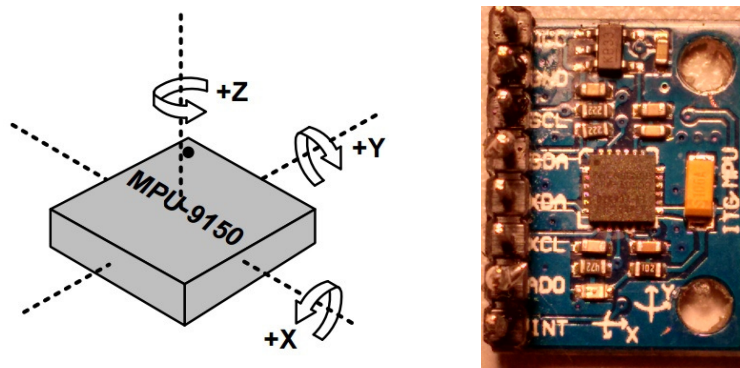
A montagem foi feita de maneira que o ângulo θ , seja calculado pelo arco tangente dos sinais dos acelerômetros e o mesmo ângulo seja calculado pela integral do sinal do giroscópio. Pois como foi descrito no capítulo anterior, a informação de ambos os sensores possui erros que a fusão de informação pode compensar.

O sensor foi posicionado na parte inferior do robô, próximo à linha de eixo dos motores, para que as acelerações decorrentes da oscilação do robô influenciem o mínimo possível as medidas. Esta configuração de posição do sensor não afeta o giroscópio, pois a velocidade angular é a mesma não importando o raio onde se meça, nem o ângulo calculado, pois o ângulo é o mesmo, independente da distância até o eixo do motor.

O sensor utilizado foi o MPU-9150, que apresenta acelerômetro de três eixos, e giroscópio de três eixos. Ele foi escolhido por ser um componente de baixo custo, fácil acesso no mercado e já ter os sensores alinhados internamente, ou seja, como o componente é monolítico, apenas precisa estar alinhado com a estrutura.

A figura (16) A mostra os eixos do sensor, e a figura (16) B mostra o sensor montado na placa. Esta é uma placa separada da placa do processador, e serve para ser fixada na parte que se deseja medir os movimentos.

Figura 20 – (A)Eixos do Sensor (B) Placa com sensor montado



Fonte: .Datasheet Inversense MPU-9150

Este sensor possui sensibilidade ajustável, como mostra a tabela (3). Isto se dá através da configuração do fundo de escala. O MPU-9150 possui filtro e amplificador interno para tratamento dos sinais dos sensores giroscópio e acelerômetro. Possui ainda conversor A/D interno de 16 bits para cada canal de medição

Tabela 3: sensibilidades do giroscópio

Fundo de escala do giroscópio	Fator de escala
$\pm 250^\circ/\text{s}$	131 LSB/ $(^\circ/\text{s})$
$\pm 500^\circ/\text{s}$	65,5 LSB/ $(^\circ/\text{s})$
$\pm 1000^\circ/\text{s}$	32,8 LSB/ $(^\circ/\text{s})$
$\pm 2000^\circ/\text{s}$	16,4 LSB/ $(^\circ/\text{s})$

Fonte: Datasheet Inversense MPU-9150

O fundo de escala escolhido foi o de $\pm 250^\circ/\text{s}$ por ser sensível e não saturar em nenhum ensaio. A folha de dados traz também os valores erro como mostra a tabela (4):

Tabela 4: Erros do giroscópio

Erro	Valor
Tolerância	$\pm 3\%$
Sensibilidade a temperatura para operação na faixa entre 25° e 85°	$\pm 0.04\%$
Não linearidade	$\pm 0.2\%$.

Fonte: Folha de dados Inversense MPU-9150

Todos os valores percentis são em função do fundo de escala escolhido. Assim chegamos a um erro total máximo de 6.48% do fundo de escala. Isto significa uma variação do de $16,2^\circ/\text{s}$. A folha de dados informa também o valor máximo de “Zero Rate Output” (ZRO) que é de $\pm 20^\circ/\text{s}$. Assim podemos concluir que o valor máximo de erro do sensor será $56,2^\circ/\text{s}$.

Tabela 5: sensibilidade do acelerômetro

Fundo de escala do acelerômetro	Fator de escala
---------------------------------	-----------------

±2g	16384 LSB/g
±4g	8192 LSB/g
±8g	4096 LSB/g
±16g	2048 LSB/g

Fonte: Folha de dados Inversense MPU-9150

Neste caso também o valor escolhido de fundo de escala foi o menor, ±2g. Isto torna a medida mais sensível. A tabela (6) traz os valores de erro dos acelerômetros.

Tabela 6: erros do acelerômetro

Erro	Valor
Calibração inicial	±3%
Sensibilidade a temperatura para operação na faixa entre 25° e 85°	±0.04%
Não linearidade	±0.5%.

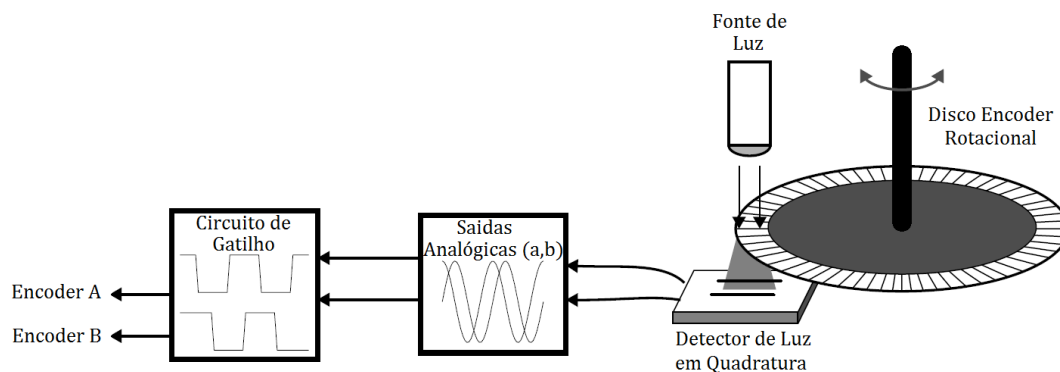
O somatório de erros mostra um total máximo de 7.04% do fundo de escala, isto é, erro total de 0.0502g. A folha de dados traz também o valor de ruído total de 4mg, assim será considerado o erro total do acelerômetro de 0.0542g.

3.1.6 ENCODER

Um encoder em quadratura converte deslocamento linear ou circular em dois sinais de onda quadrada, defasados 90°. Monitorando o número de pulsos é possível saber o deslocamento angular e pela derivada é possível calcular a velocidade angular. E analisando a fase dos sinais é possível saber a direção de rotação.

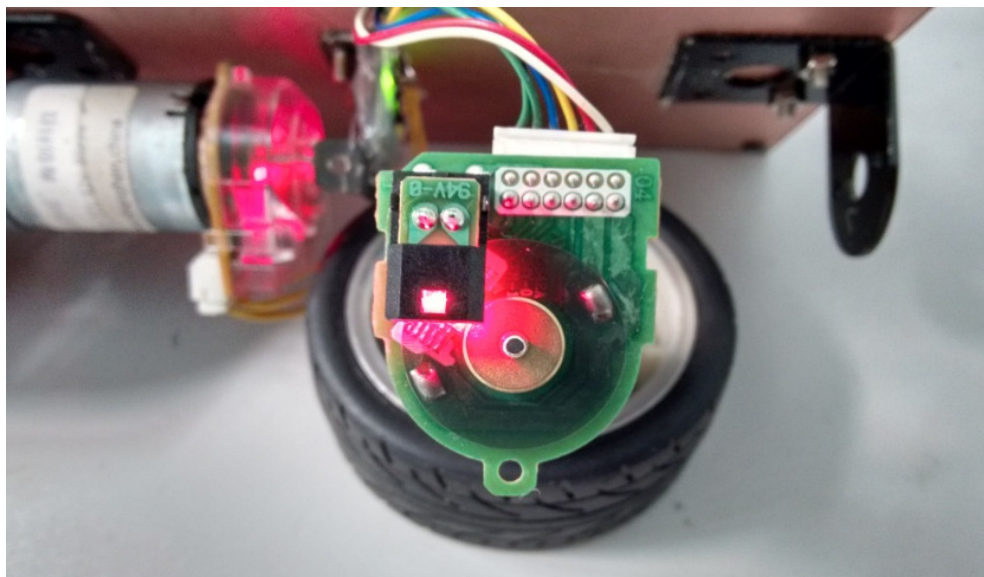
O funcionamento do encoder é mostrado na figura (21), onde o disco gira com o eixo do motor. As marcas na borda do disco criam sinais de luz e sombra no detector de luz, que os transforma em sinais elétricos. Todo este sistema está embutido no motor, como mostra a figura (22).

Figura 21 – Funcionamento do Encoder.



Fonte: .Kalman Quadrature [26]

Figura 22 – Encoder utilizado montado no motor.



Fonte: .própria

A folha de dados do motor mostra que o disco dentado possui 334 ranhuras, e como o sistema lê em quadratura, ou seja, existem dois feixes de luz, que geram duas ondas quadradas defasadas de 90° . Para cada ranhura são geradas 4 bordas, sendo duas de subida e duas de descida. Assim, para cada rotação do motor o canal de medição QEI conta $334 \times 4 = 1336$ bordas. A caixa de redução acoplada ao motor possui relação de $1 : 9,28$ assim, para cada revolução do eixo da roda, são contados 12398 bordas.

3.2 IMPLEMENTAÇÃO DE ALGORITMOS DE FUSÃO

Os métodos de fusão de informação implementados neste trabalho foram utilizados para estimar a posição angular θ e a sua derivada, ou seja, a velocidade angular, em um sistema unidimensional. Pressupõe-se que o pêndulo oscila em apenas uma direção. Nesta

configuração, o sistema de fusão de informação utiliza os dados dos acelerômetros e do giroscópio para estimar o ângulo e velocidade angular do pêndulo.

Nas próximas seções serão descritas os métodos de Filtro de Kalman e Filtro complementar e suas implementações no projeto. .

3.2.1 FILTRO COMPLEMENTAR

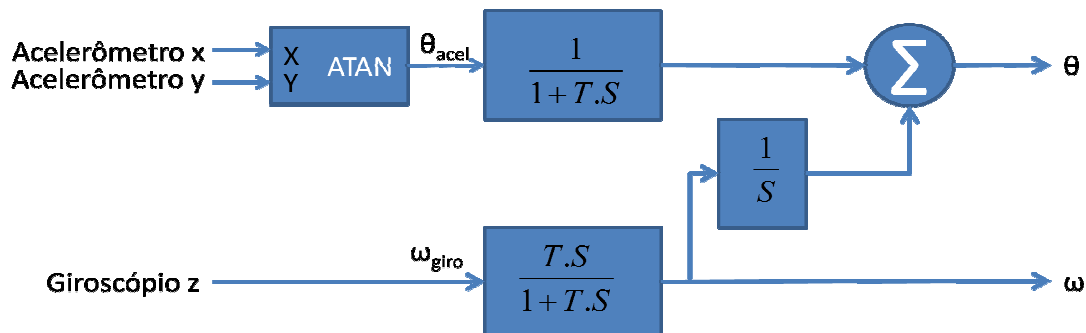
A seguir, será analisado a implementação do filtro complementar. Este filtro é composto pela soma de dois filtros um passa altas e um passa baixas, como já explicado no capítulo anterior. A equação a seguir descreve o filtro implementado:

$$\hat{\theta}_k = \frac{1}{1 + T.S} \theta_{acel} + \frac{T.S}{1 + T.S} \cdot \frac{1}{S} \omega_{giro}$$

Esta equação descreve um filtro passa baixas aplicado ao sinal de θ_{acel} , isto é, ao valor de ângulo proveniente do cálculo do arco tangente dos acelerômetros, e um filtro passa baixas à integral do sinal do giroscópio. Ambos os filtros são de primeira ordem. É possível implementar filtros de ordem mais elevada, mas fogem ao escopo deste trabalho.

O sinal da velocidade angular pode ser tomado da saída do filtro passa altas, antes do integrador. Como trata-se de sistemas lineares a troca de ordem não altera o resultado final. Assim, o diagrama de blocos do filtro complementar pode ser tal como a figura (23).

Figura 23 – diagrama de blocos do filtro complementar.



Fonte: .própria

O valor de T determina o valor de frequência digital de corte [23]. Ambos os filtros devem ser projetados de tal forma que a frequência de corte do filtro passa baixas seja a mesma do filtro passa altas, o que fará com que a soma dos dois filtros será unitária. Isto garante que o sistema responda de forma plana para todas as frequências, não causando amplificação nem atenuação em determinadas faixas de frequência digital. [15]

Para serem implementados no microcontrolador, os filtros foram transformados para o domínio tempo na forma de séries. Esta transformação gerou filtros do tipo de resposta infinita ao impulso, como pode ser visto nas equações a seguir. Isto significa que a saída depende da entrada do filtro e das saídas anteriores, esta é uma das principais características dos filtros do tipo resposta infinita ao impulso.

A transformação resultou na seguinte equação para a estimação do ângulo θ :

$$\hat{\theta}_k = \alpha \cdot (\hat{\theta}_{k-1} + G_k \cdot dt) + (1 - \alpha) \cdot \theta_k$$

Onde

α é o parâmetro que dá a frequência de corte dos filtros.

$\hat{\theta}_k$ é o ângulo estimado pelo filtro, ou seja, é a saída do filtro

G_k é o valor do giroscópio

θ_k é o valor de ângulo calculado a partir dos acelerômetros

E na seguinte equação para a estimação da velocidade angular:

$$\hat{\omega}_k = \alpha \cdot G_k - (1 - \alpha) \cdot \hat{\omega}_{k-1}$$

Onde $\hat{\omega}_k$ é a estimativa da velocidade angular sem a *drift*.

Como trata-se de um filtro com resposta infinita ao impulso, seus parâmetros devem obrigatoriamente ser menores que a unidade, caso contrário, pode instabilizar o filtro, fazendo-o divergir. Foram testados vários valores de α empiricamente, o valor que apresentou resultado mais satisfatório foi $\alpha = 0,98$. [23]

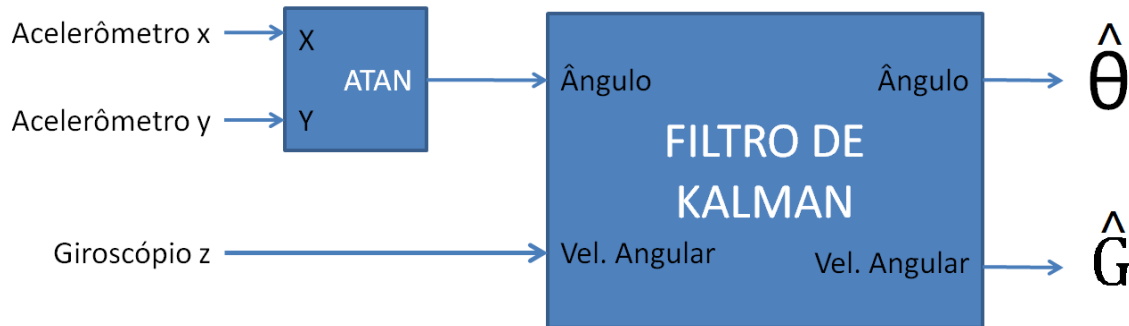
3.2.2 FILTRO DE KALMAN:

Como visto no capítulo anterior, o filtro Kalman necessita do modelo matemático do sistema do qual estimará as variáveis de estado. A estrutura do sistema modelado para o cálculo do ângulo utilizado neste projeto foi simplificada para levar em consideração apenas a geometria e as características dos sensores de aceleração e de giro. O modelo utilizado é mostrado na figura (24).

Como se deseja calcular o mesmo ângulo θ proveniente de duas fontes diferentes, ou seja, deseja-se estimar o ângulo θ proveniente dos acelerômetros sem o ruído causado pela oscilação do móvel, e o mesmo ângulo θ porém sem a *drift* causada pela integração numérica, os estados escolhidos são o próprio ângulo θ e a *drift* de giro.

Ambos os sinais de ângulo e giro oscilam de acordo com forças externas, mas que medem velocidade angular. Sendo assim, podemos considerar como entrada deste sistema o sinal de velocidade angular medido pelo giroscópio.

Figura 24 – diagrama de entradas e saídas do filtro Kalman.



Fonte: .própria

O modelo criado para descrever a estrutura formada pelos sensores apresenta como saídas mensuráveis os sinais dos acelerômetros e de giro, como é mostrado na figura (24). Assim pode-se deduzir o modelo do sistema pelas equações no espaço de estado abaixo.

$$\begin{aligned}\theta_k &= \theta_{k-1} + (G_k - G_{d_{k-1}}) \cdot dt \\ &= \theta_{k-1} + G_k \cdot dt - G_{d_{k-1}} \cdot dt\end{aligned}$$

e

$$G_{d_k} = G_{d_{k-1}}$$

Onde:

θ representa o ângulo com a vertical,

G_d é o *drift* do giro, ou seja, o erro do giroscópio, e

G_k é o sinal de velocidade angular, medido pelo giroscópio, o qual será a entrada deste sistema.

Assim, podemos reescrever as equações no formato de matrizes do espaço de estados no domínio digital, como segue:

$$\begin{aligned}X_k &= A \cdot X_{k-1} + B \cdot u_{k-1} \\ \begin{bmatrix} \theta \\ G_d \end{bmatrix}_k &= \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \theta \\ G_d \end{bmatrix}_{k-1} + \begin{bmatrix} dt \\ 0 \end{bmatrix} \cdot G_k\end{aligned}$$

Importante notar que para este modelo é considerada constante. A folha de dados do sensor diz que o valor de giro pode derivar dentro do intervalo de $-20^\circ/s$ até $+20^\circ/s$ sobre o

valor de velocidade angular. Porém, como não se sabe o valor exato, este será estimado dinamicamente.

Os valores que interessam para este projeto são o ângulo e a velocidade angular, portanto estes serão os valores de saída do filtro. Assim as equações no espaço de estados da saída é:

Equação de saída:

$$\begin{aligned}\theta_k &= \theta_k \\ \dot{\theta}_k &= G_k - \text{Deriva Giro}\end{aligned}$$

Na forma de matriz:

$$\begin{aligned}Y_k &= C \cdot X_{k-1} + D \cdot u_{k-1} \\ Y_k = \begin{bmatrix} \theta \\ G \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} \theta \\ G_d \end{bmatrix}_{k-1} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot G_{k-1}\end{aligned}$$

O algoritmo de filtro Kalman, como descrito no capítulo anterior, está dividido em duas etapas. Na primeira etapa, de predição, são calculados os valores dos estados a priori, e a matriz de covariância, como mostram as equações eq.(13) e eq.(12).

As condições iniciais utilizadas nesta etapa são os valores iniciais para os estados,

$$\begin{bmatrix} \theta \\ G_d \end{bmatrix}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

E a matriz de covariância a posteriori inicial utilizada é uma matriz identidade

$$P_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

E a matriz de covariância dos ruídos Q é um parâmetro do sistema, e é montada como segue:

$$Q = \begin{bmatrix} Q_{acel} & 0 \\ 0 & Q_{giro} \end{bmatrix}$$

Os valores de Q_{acel} e Q_{giro} são baseados nos dados levantados do datasheet do sensor e são constantes no processo.

Assim temos que:

$$\begin{bmatrix} \hat{\theta}^- \\ \hat{G}_d^- \end{bmatrix}_k = \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \hat{\theta} \\ \hat{G}_d \end{bmatrix}_{k-1} + \begin{bmatrix} dt \\ 0 \end{bmatrix} \cdot G_k$$

e

$$P_k^- = \begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix} \cdot [P_{k-1}] \cdot \begin{bmatrix} 1 & 0 \\ -dt & 1 \end{bmatrix} + \begin{bmatrix} Q_{acel} & 0 \\ 0 & Q_{giro} \end{bmatrix}$$

A segunda etapa do algoritmo, de atualização, calcula inicialmente o valor do ganho K_k , descrito na eq.(15), e atualização de \hat{x}_k , como descrito na eq.(13) e por fim atualiza o novo valor da matriz de covariância do P_k como visto na eq.(16).

Para o cálculo de K_k , o valor de R representa a variância do ruído de medida, como visto no capítulo anterior. Assim o valor utilizado representa o valor de ruído de quantização inserido por um conversor A/D de 16bits que é o caso do CI MPU-9150 utilizado. Utilizando a equação $SNR = 6,02 \cdot N + 1,67 [dB]$ para calcular o ruído de quantização de um conversor A/D de 16 bits, para um sinal que excursiona de 0V a 3V, obtem-se uma amplitude de ruído de $3,74E-5$.

A atualização do valor de \hat{x}_k , é feita da seguinte forma:

$$\begin{bmatrix} \hat{\theta} \\ \hat{G}_d \end{bmatrix}_k = \begin{bmatrix} \hat{\theta}^- \\ \hat{G}_d^- \end{bmatrix}_k + [K_k] \cdot \left(\begin{bmatrix} \hat{\theta} \\ \hat{G} \end{bmatrix}_k - [P_k^-] \cdot \begin{bmatrix} \theta \\ G_d \end{bmatrix}_{k-1} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot G_k \right)$$

O valor entre parênteses é chamado de resíduo e sua análise fornece informação sobre a convergência do método de estimação.

Por fim é atualizado o valor da matriz de covariância do erro a posteriori pela eq.(16)

$$P_k = (I - K_k C) P_k^-$$

Todo este processo deve ser executado a cada nova amostra dos sensores. Foram realizados experimentos com frequência amostral de 1KHz, 500Hz, 200Hz e 100Hz e os experimentos não mostraram grande diferença de desempenho entre si. Então foi utilizado o valor de 500Hz.

3.3 ALGORITMOS DE CONTROLE

De posse das medidas dos sensores e das variáveis que precisam ser controladas na planta do robô, parte-se para a implementação dos controladores. Na literatura existem dois métodos mais utilizados, o Controlador PID e o LQR. Ambos foram estudados no capítulo 2 e serão implementados. Neste capítulo serão expostos os detalhes da implementação desses controladores e suas características.

3.3.1 LQR – LINEAR QUADRATIC REGULATOR

A implementação do controle LQR inicia-se na modelagem do robô pêndulo invertido, como descrito no capítulo 2. As matrizes A e B, descritas no referencial teórico, e as matrizes Q e R, descritas no filtro Kalman, são a base para a implementação do controlador.

Porém as equações que descrevem o funcionamento do sistema no espaço de estados estão no domínio contínuo, e devem ser discretizadas para que seja implementado o algoritmo numérico de controle no microcontrolador. Existem vários métodos de discretização, e para este trabalho, foi utilizado o método “Zero Order Holder” (ZOH).

Este método

$$\dot{x}(t) = A.x(t) + B.u(t) + w(t)$$

$$y(t) = Cx(t) + Du(t) + v(t)$$

Com as covariâncias igual ao assumido na dedução do Filtro Kalman

$$w(t) \sim N(0, Q)$$

$$v(t) \sim N(0, R)$$

A discretização ZOH fornece as seguintes saídas:

$$A_d = e^{AT} = \mathcal{L}^{-1}\{(sI - A)^{-1}\}_{t=T}$$

$$B_d = \left(\int_{\tau=0}^T e^{A\tau} d\tau \right) \cdot B$$

$$C_d = C$$

$$D_d = D$$

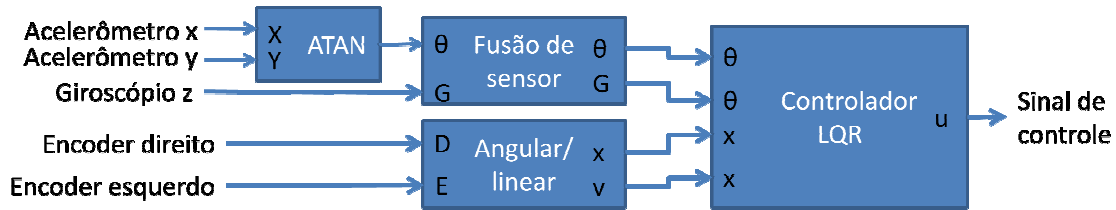
$$Q_d = \int_{\tau=0}^T e^{A\tau} Q e^{A\tau} d\tau$$

$$R_d = \frac{1}{T} R$$

Onde T é o período de amostragem e A^T é a matriz transposta de A . Para realizar esta transformação foi utilizado o Matlab, pois possui funções específicas para realizar estas transformações.

O controlador LQR do robô pêndulo invertido se baseia na regulação das quatro variáveis de estados descritas no capítulo sobre o modelo do pêndulo invertido. A estrutura do controlador do robô pêndulo invertido é mostrada na figura (25). O controlador é alimentado pelos valores de estado; o ângulo θ e a velocidade angular são fornecidos pelo algoritmo de fusão de informação, e o valor da posição e velocidade da roda vem dos encoders.

Figura 25 – Estrutura do controlador LQR.



Fonte: .própria

Os encoders medem o deslocamento angular da roda, a partir deste valor calcula-se o deslocamento linear e velocidade linear do robô. Como o robô foi projetado inicialmente para se deslocar em uma única direção, ambos os encoders medem o mesmo valor, porém para compensar possíveis deslizos o deslocamento angular considerado é a média dos dois valores lidos dos encoders.

O deslocamento em x é calculado pela equação:

$$X_k[m] = \frac{\text{Valor do encoder}}{\text{Encoder Max}} \cdot 2 \cdot \pi \cdot r$$

Onde:

Encoder Max é o valor lido para um giro completo da roda e

r é o raio da roda do robô.

A velocidade é calculada pela derivada numérica do deslocamento. Assim, é calculada pela seguinte equação:

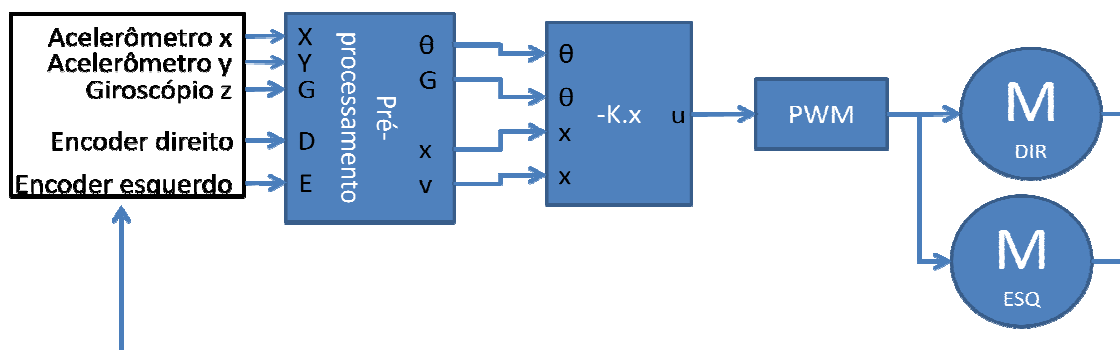
$$V[m/s] = \frac{(X_k - X_{k-1})}{dt}$$

Onde dt é o inverso da frequência de amostragem.

Como visto no capítulo anterior, o valor do sinal de controle é função das variáveis de estados. E o cálculo dos parâmetros K do controle LQR foi feito utilizando a função LQR do Matlab, a partir das matrizes A_d , B_d , Q_d e R_d . Assim a função $K = \text{lqrd}(A_d, B_d, Q_d, R_d, FS)$, onde FS é a frequência de amostragem, resultou nos seguintes valores de $K = [-31,6228; -28,8728; -274,2128; -29,1567]$.

Assim, $u = -K \cdot \hat{x}$ é o valor de tensão que deve ser aplicado ao motor. No projeto este valor é o que controla o valor de PWM aplicado ao motor. Como mostra a figura(26)

Figura 26 – Diagrama de blocos do robô controlado pelo LQR.



Fonte: .própria

3.3.2 CONTROLADOR PID

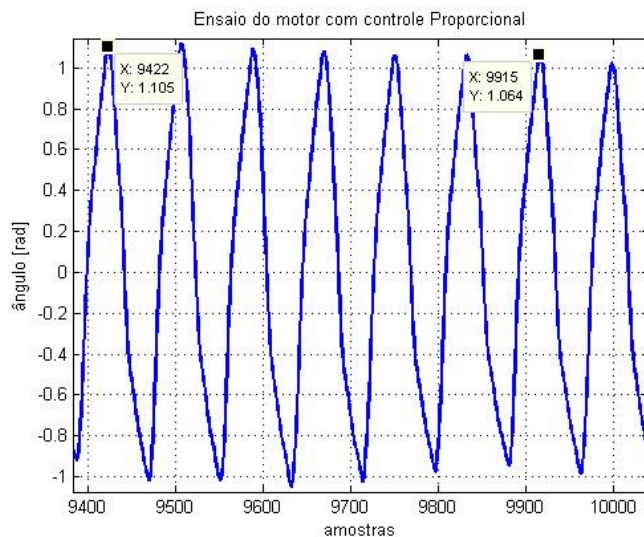
O controlador PID implementado neste projeto recebe como informação o ângulo. Este controle possui como variável controlada apenas o ângulo que o pêndulo faz com a vertical, pois o controlador PID é um controle de apenas uma variável, sendo assim, matematicamente bem mais simples que o LQR que necessita uma complexa modelagem matemática para ser implementado.

A sintonia do controlador, ou seja, o cálculo dos parâmetros K_p , K_i e K_d utilizado neste projeto é relativamente simples. O método de Ziegler e Nichols é um método muito utilizado na indústria, e consiste em aproximar o comportamento da planta ao comportamento de primeira ordem, ou seja, planta com um polo. Assim, ele estima o valor do ganho do sistema e do valor do menor polo.

Neste trabalho foi utilizado o método de Ziegler-Nichols para sintonia de controlador PID, como realizado em [12]. Este procedimento requer que seja feito um ensaio apenas com um controlador proporcional, elevando o ganho até que a variável controlada oscile. Assim o ganho no qual a variável controlada oscila é o ganho K_u , e o período de oscilação é a variável P_u .

O ensaio realizado resultou na figura (27), onde estão marcados os pontos de 6 ciclos de oscilação. O valor de P_u considerado é a média dos 6 ciclos, assim $P_u = 82,16$ amostras. E o valor de K_u utilizado no código foi de 200.

Figura 27 – Ensaio Ziegler e Nichols.



Fonte: .própria

De posse desses dados, utiliza-se as fórmulas de Ziegler-Nichols de sintonia de controlador, da tabela (7).

Tabela 7: Parâmetros de sintonia de PID Ziegler e Nichols

Controlador	K_P	T_I	T_D
PID	$0.65K_u$	$0.5P_u$	$0.12P_u$

Fonte: [12]

A tabela 7 fornece as informações dos parâmetros T_i e T_d . Os referentes ganhos integral e diferencial dependem também do período entre as amostras do sinal, como mostrado nas equações abaixo. A frequência de amostragem utilizada neste ensaio foi de 500Hz, logo, $T_S = 0,002$.

$$T_I = P_u * 0,5 = 81,46 \text{ amostras} * T_S = 0,16292s$$

$$T_D = P_u * 0,12 = 0,16292 * 0,12 = 0,0195504s$$

Assim, pode-se calcular:

$$K_P = K_u * 0,65 = 130$$

$$K_i = \frac{K_P \cdot T_S}{T_I} = \frac{130 * 0,002}{0,1629s} = 1,60$$

e

$$K_d = \frac{K_p \cdot T_D}{T_S} = \frac{130 * 0,0195504}{0,002} = 1270,78$$

Assim chegamos aos parâmetros:

$$K_p = 130$$

$$K_i = 1,60$$

$$K_D = 1270,78$$

Com estes dados foi realizado os ensaios do controlador PID, como serão vistos no capítulo a seguir de resultados.

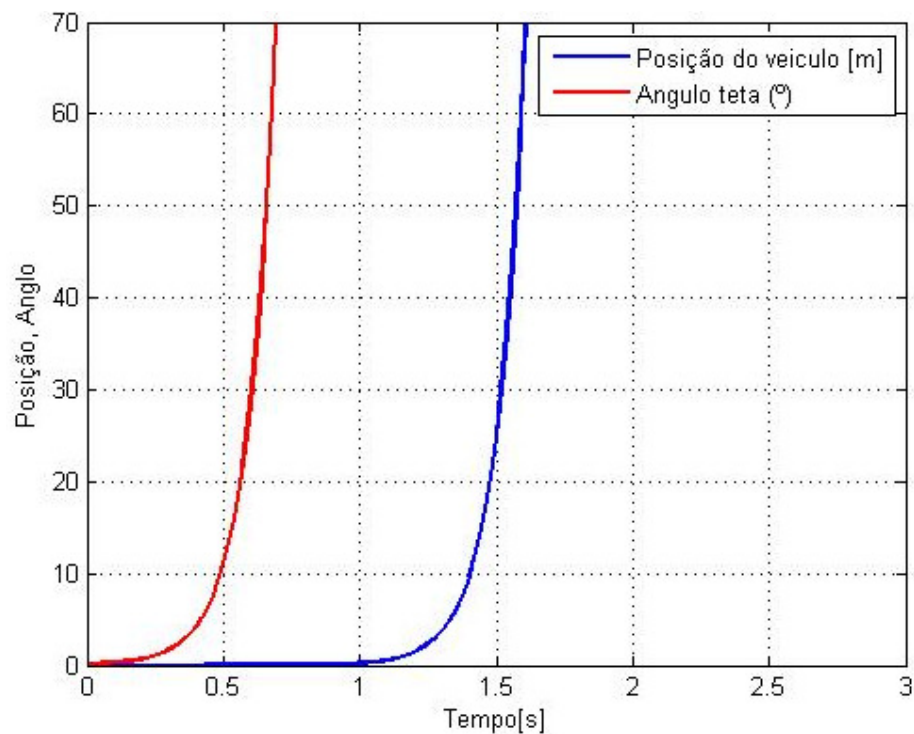
4 RESULTADOS

4.1 SIMULAÇÕES

Após a modelagem matemática e a identificação dos parâmetros do motor e da estrutura mecânica do robô, foi realizada a simulação do comportamento do robô em malha aberta. Também é importante de verificar a funcionalidade dos métodos de controle utilizados neste projeto.

Na figura 28 consta o gráfico da resposta ao impulso na planta em laço aberto

Figura 28 – Resposta do sistema em laço aberto ao impulso.



Fonte: Própria

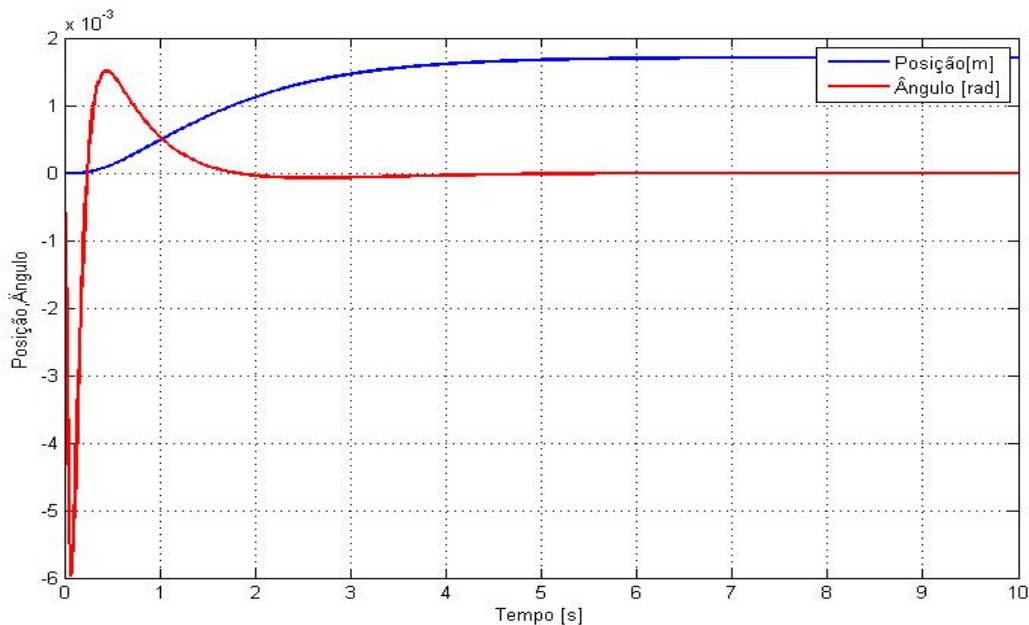
Os autovalores da matriz A do sistema formam os polos do sistema. Os valores de polos positivos indicam instabilidade do sistema. Os autovalores do robô (calculados no Matlab) são: **0**; **-0.00890229**; **-13.12038**; **+13.11821**, o que deixa claro a instabilidade da planta.

As simulações dos controladores visam estudar o comportamento da planta de forma simplificada, em condições ideais. Fornecendo assim dados para implementação prática do

robô. O controlador PID foi sintonizado com os parâmetros gerados pelo método de Ziegler e Nichols.

A figura 29 mostra a resposta ao impulso do robô. O controlador PID controla o ângulo mas não a posição do robô, portanto este desloca-se e não retorna à posição inicial.

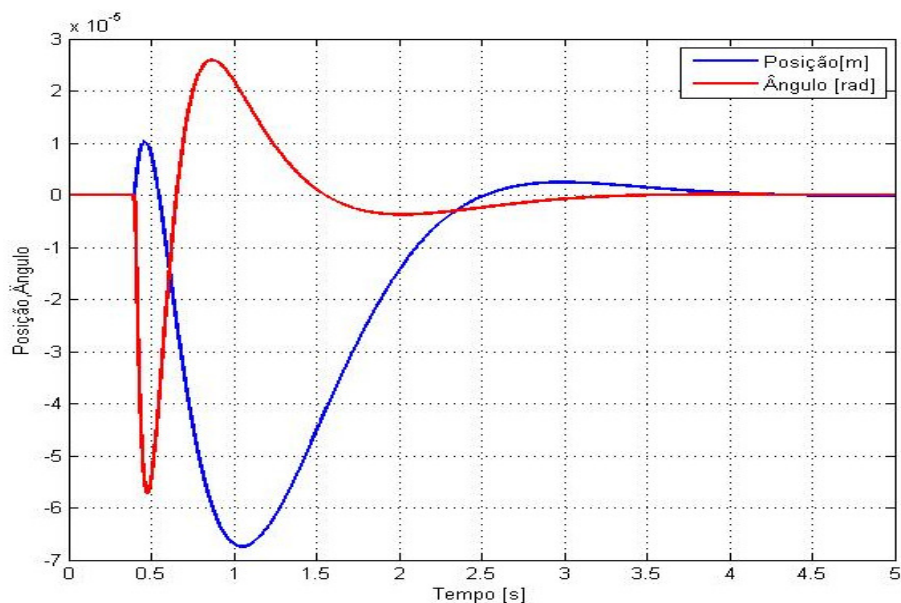
Figura 29 – Resposta ao impulso do sistema com controlador PID.



Fonte: Própria

A figura 30 apresenta a simulação da resposta ao impulso do robô com o controlador LQR. Este impulso simula uma perturbação no robô. Assim é possível perceber que o modelo simulado oscila muito pouco antes de chegar novamente à posição de referência.

Figura 30 – Resposta ao impulso do sistema com controlador LQR.



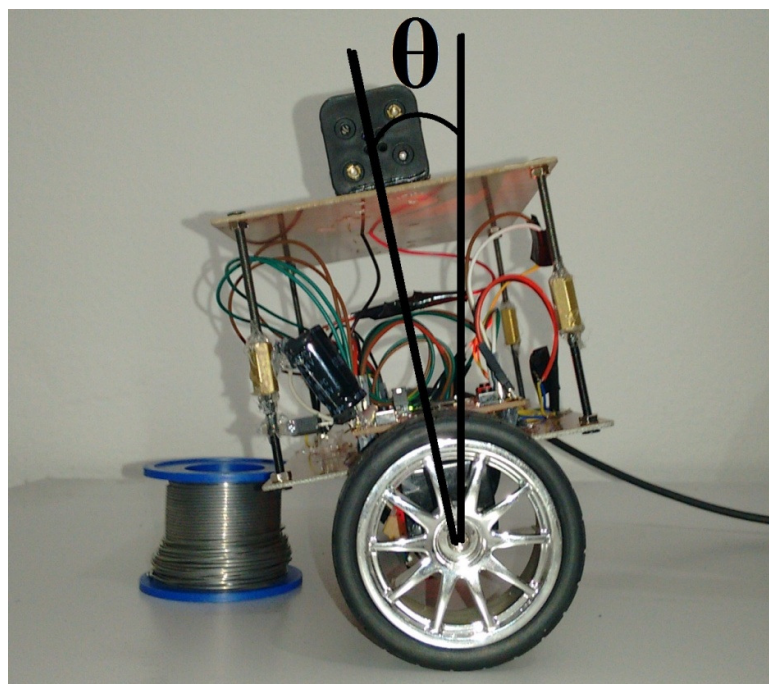
Fonte: .Própria

4.2 IMPLEMENTAÇÃO DO ROBÔ PÊNULO INVERTIDO

Foram desenvolvidos dois tipos de teste para avaliar o funcionamento dos métodos de controle estudados neste projeto. O primeiro teste consiste em fazer o sistema responder a um degrau, ou seja, o robô parte de uma posição inclinada de 0,19 radianos, cerca de 11° , devendo equilibrar-se em zero radianos, como mostrado na figura 31. Este teste foi aplicado para cada uma das quatro combinações de controlador e método de fusão de sensores. O segundo teste consiste em uma análise estatística do sinal do controlador e sinais dos sensores. Por fim, é mostrada a análise de tempo de execução de cada algoritmo.

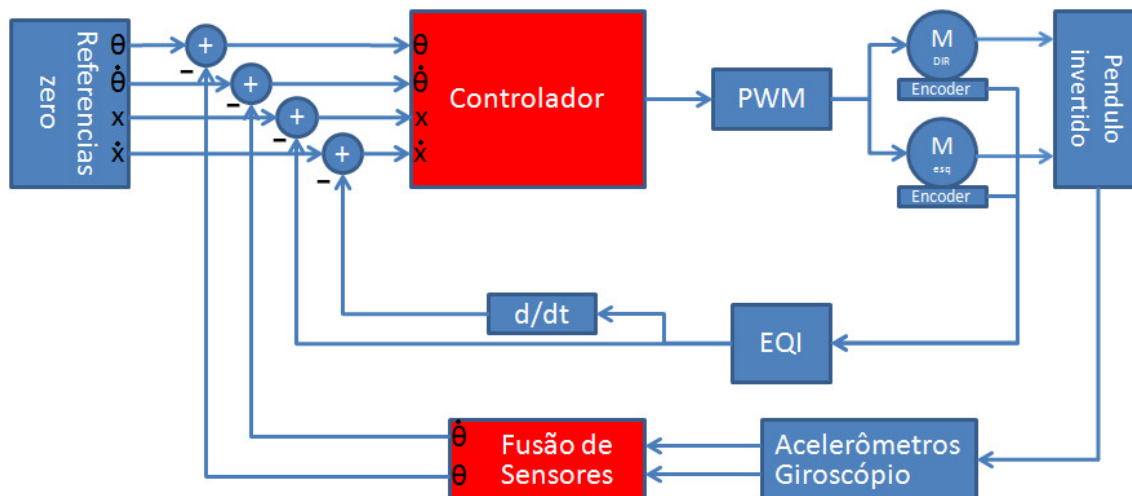
A arquitetura do sistema do robô utilizado para realizar os testes pode ser vista na figura 31. Em cada ensaio uma combinação diferente de controlador e do método de fusão de informação foi configurado.

Figura 31 – Robô na posição inicial do teste de resposta ao degrau



Fonte: .Própria

Figura 32 – Robô na posição inicial do teste de resposta ao degrau



Fonte: .Própria

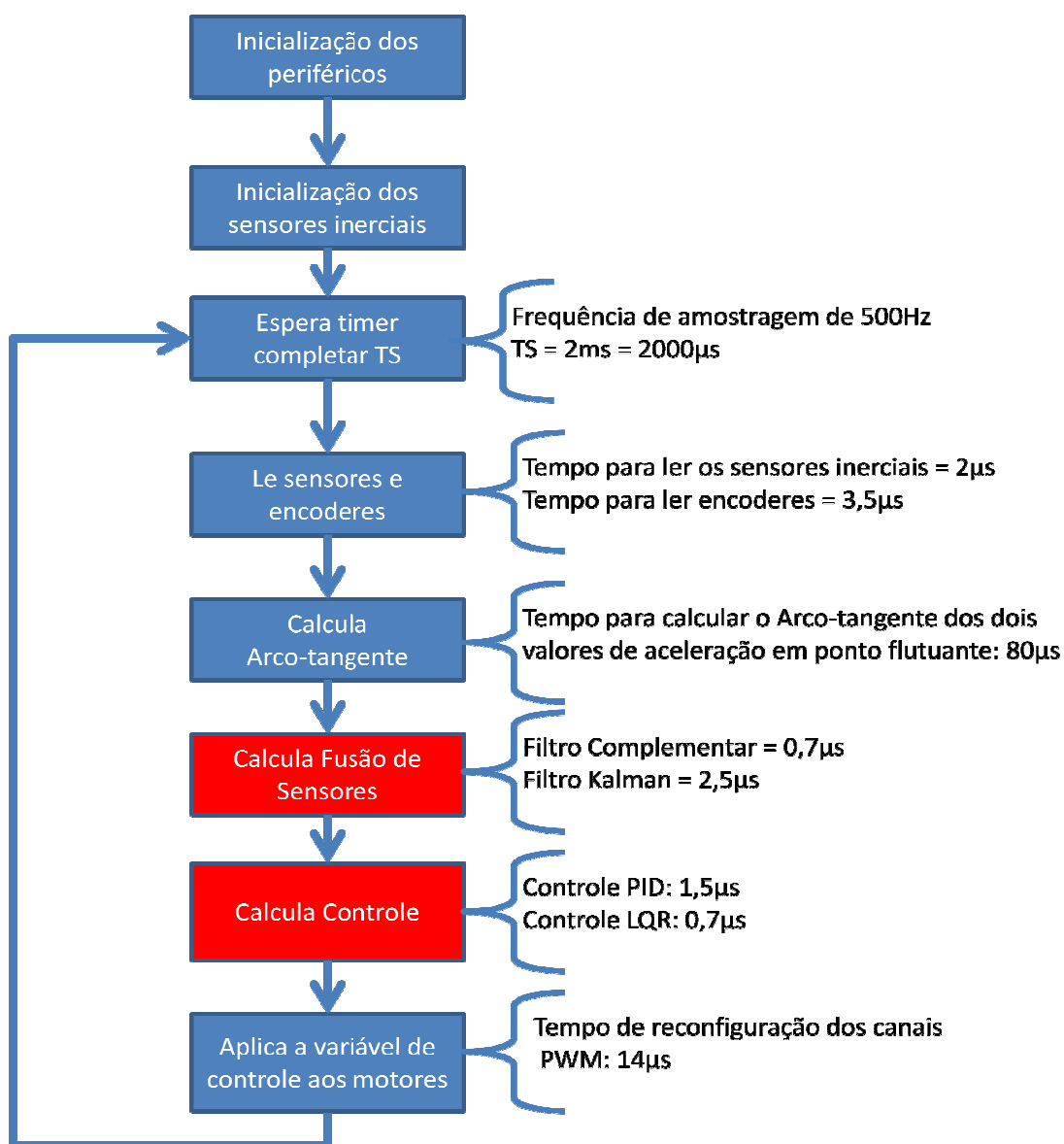
O ângulo inicial foi medido por ambos os sensores (acelerômetro e giroscópio) e posteriormente processados pelos sistemas de fusão de informação. Foram realizadas 5120 medições, o que significa cerca de 5 segundos de medições, e para tal configuração ambos os sistemas de fusão de informação forneceram como valor médio 0,19 radianos. Optou-se por confiar nesta informação, dado que o sistema estava em repouso, logo a informação já estava estabilizada.

A estrutura montada para os teste pode ser vista na figura 32, onde consta a estrutura de um sistema de controle realimentado. Os sensores inerciais como já abordado nos capítulos anteriores são os acelerômetros e o giroscópio. Estes passam pela etapa de fusão de informação, para obter-se uma estimativa de ângulo mais confiável.

Os subsistemas testados são: a Fusão de sensores e o Controlador, mostrados em vermelho na figura 32. O controlador LQR utiliza como variáveis controladas todas as variáveis de estados, isto é x, \dot{x}, θ e $\dot{\theta}$. Já o controlador PID apenas considera a informação de ângulo θ .

Todo o ciclo de código implementado no processador pode ser visto na figura 33, onde consta o fluxograma do projeto. As etapas de fusão de sensores e controle estão em vermelho. Também estão marcados na figura os tempos gastos em cada operação realizada pelo processador. Pode-se notar que as operações de cálculo de arco-tangente e reconfiguração de canais PWM tomam praticamente todo o tempo do processo, e as diferença de tempo de processamento entre os algoritmos testados é pequena frente ao tempo gasto por essas funções.

Figura 33 – Fluxograma do algoritmo do robô



Fonte: .Própria

4.2.1 CONTROLE PID COM FILTRO COMPLEMENTAR

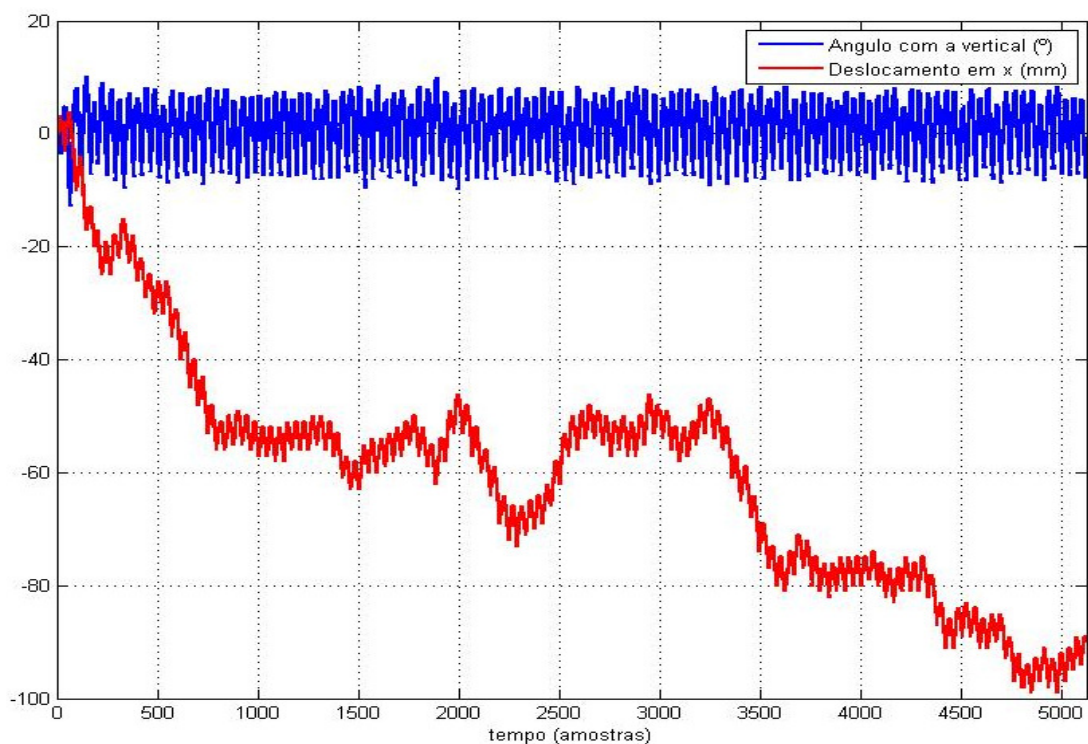
Neste ensaio foi configurado o filtro complementar como método de fusão de sensores e o controlador PID, cuja única variável controlada é o ângulo θ . A figura 33 mostra o gráfico da resposta ao salto do ângulo θ do robô e a posição x medida pelo encoder. Esta informação diz o quando o robô se deslocou linearmente enquanto busca a posicionar o ângulo θ igual a zero.

Como pode-se ver na figura 34, como o controlador PID não controla a variável “x” esta desloca-se. Porém, embora se desloque no eixo horizontal e apresente elevada oscilação, o sistema nessa configuração conseguiu se sustentar pelos 5 segundos de teste.

A variância do sinal de ângulo foi de 25,011. Esta métrica será um parâmetro comparativo entre os todos os testes realizados. Servindo para determinar qual combinação apresenta menos oscilação final.

Esta métrica foi escolhida por estar relacionada com a potência AC do sinal, ou seja, potência do sinal oscilante. O RMS está relacionado com a potência total do sinal, logo a potência DC mais a potência AC. Porém, neste sistema deseja-se o sinal de ângulo com média igual a zero, pois uma tendência no ângulo causará um deslocamento no eixo x, devido à tentativa do controlador de corrigir esta tendência. Assim a medida de potência apenas da parte AC é mais adequada, pois evita a medida do deslocamento no eixo x, e correções do controlador. Em outras palavras esta medida pondera apenas a oscilação do ângulo, e quanto menos a oscilação final melhor será considerada a combinação.

Figura 34 – Resposta ao degrau do sistema formado por PID e Filtro Complementar

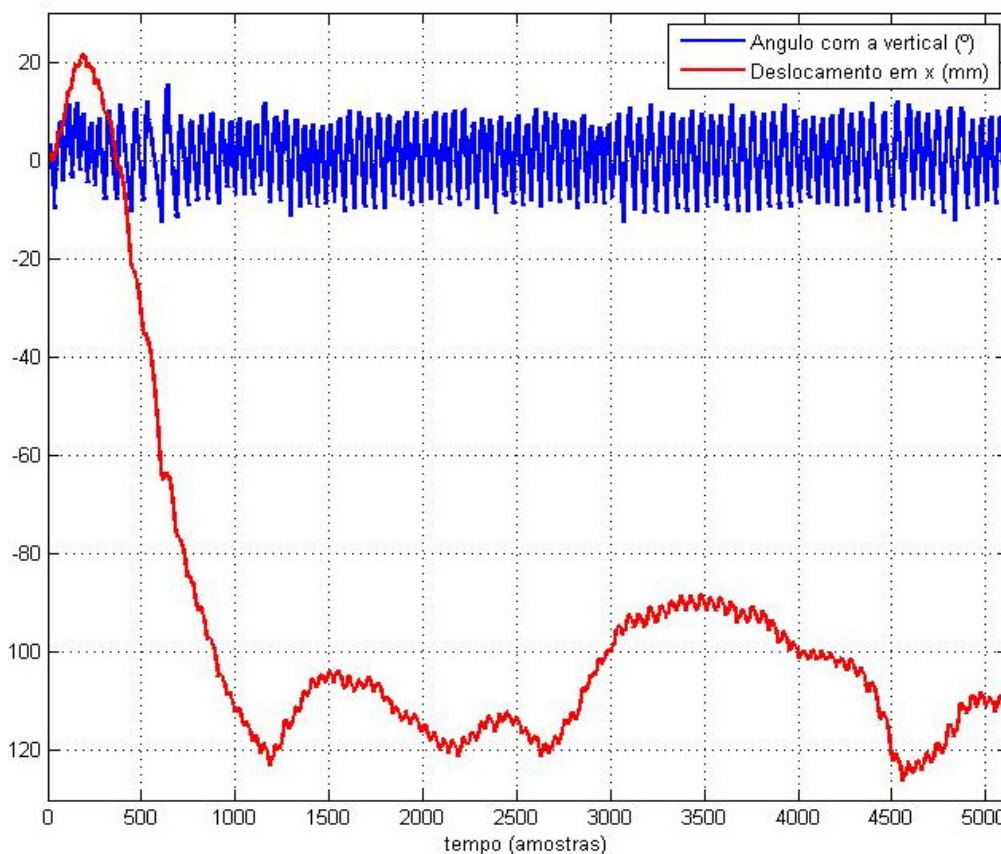


Fonte: .Própria

4.2.2 CONTROLE PID COM FILTRO KALMAN

No teste onde foi utilizado o controlador PID, porém com o filtro Kalman, percebe-se uma pequena redução na oscilação do ângulo e até mesmo do sinal de deslocamento. A variância do sinal de ângulo é: 32,222, mostrando um aumento em relação ao teste anterior. A figura 35 mostra a resposta ao salto desta combinação. Isto será explicado no teste seguinte.

Figura 35 – Resposta ao degrau do sistema formado por PID e Filtro Kalman

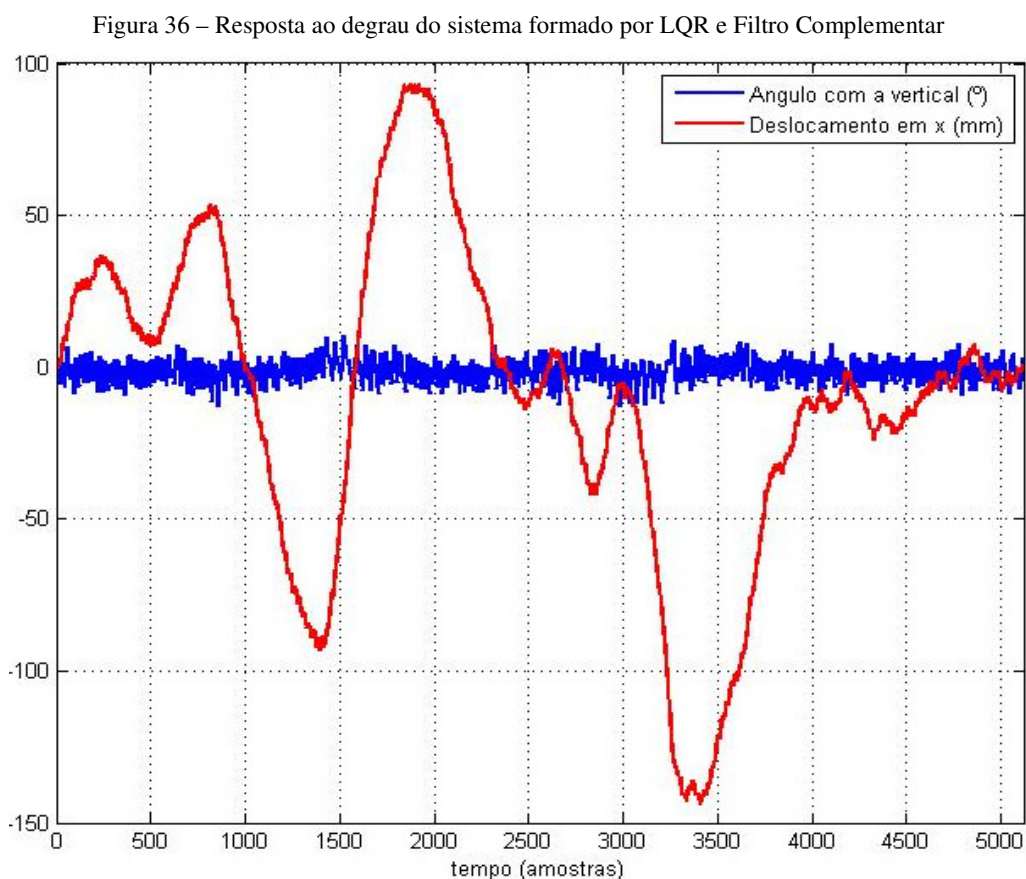


Fonte: .Própria

4.2.3 CONTROLE LQR COM FILTRO COMPLEMENTAR

O ensaio do sistema de controle LQR e fusão de sensores com filtro complementar apresentou uma sensível redução na oscilação do ângulo, como mostra a figura 36, onde pode-se notar que o sistema de controle tem uma grande influência na oscilação final do sistema.

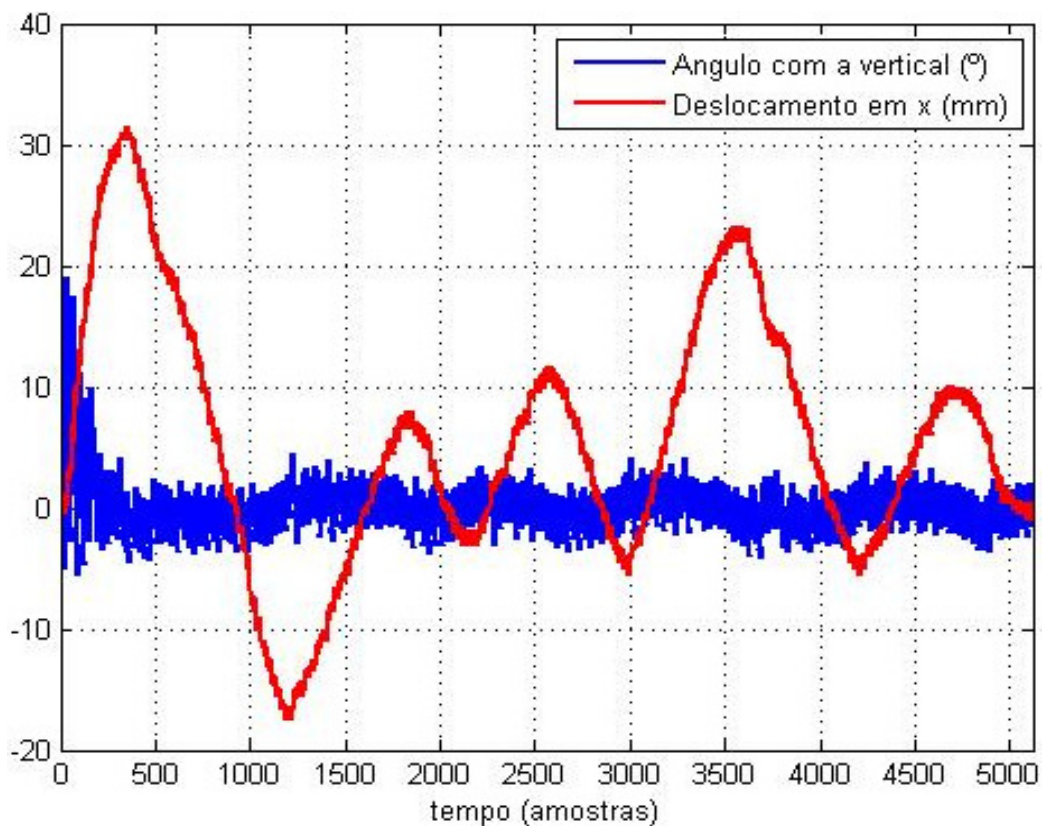
O controlador LQR melhorou bastante o sistema em comparação com a combinação entre PID e Filtro complementar, pois além de diminuir a oscilação final também faz com que o sistema volte à posição de origem, como pode ser visto na figura, no gráfico em vermelho. Pode-se ainda ver que o robô se deslocou no eixo x para permitir o equilíbrio, porém, tem a tendência de voltar à posição inicial. A variância do sinal de ângulo deste ensaio é 12,026, praticamente um terço da combinação PID e filtro complementar.



4.2.4 CONTROLE LQR COM FILTRO KALMAN

Ao final foi testado a combinação de controlador LQR e fusão de sensores com o filtro Kalman. Este caso apresentou os melhores resultados, como pode ser visto na figura 37. Além de apresentar a menor oscilação de ângulo de todas combinações, com variância de 4,01, o robô foi capaz de se deslocar da menor distância do ponto de partida.

Figura 37 – Resposta ao degrau do sistema formado por PID e Filtro Kalman



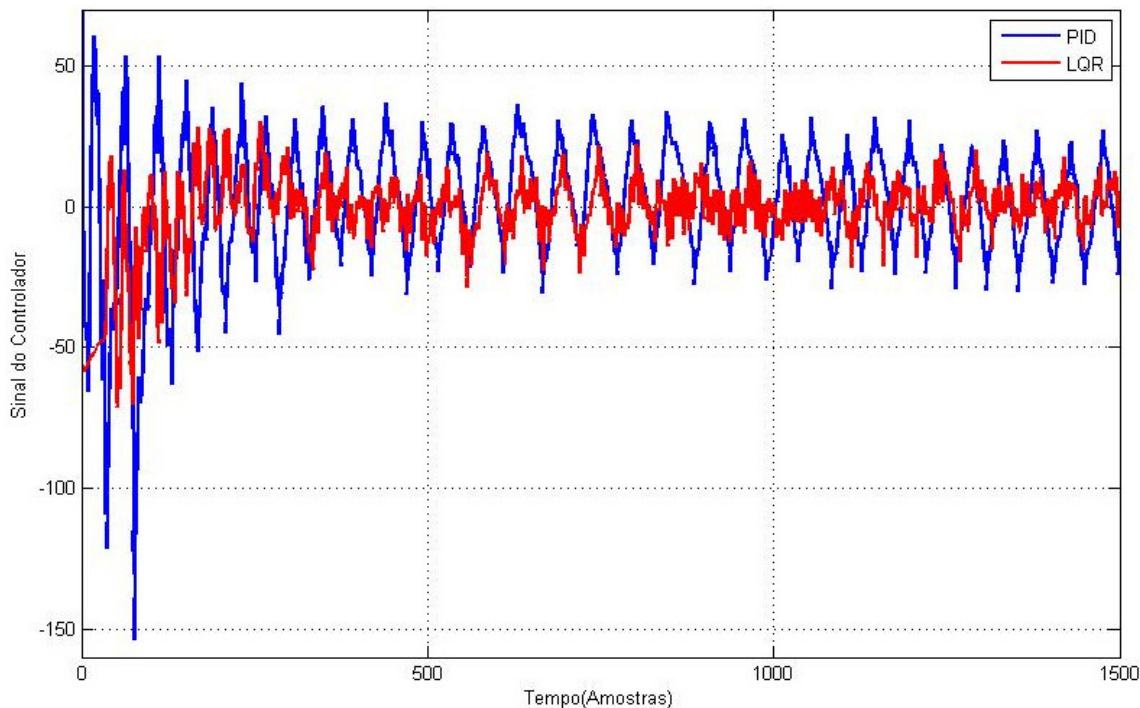
Fonte: .Própria

4.3 ANÁLISE DOS CONTROLADORES

Na segunda parte dos testes foi analisado o comportamento estatístico dos controladores e os métodos de fusão de informação, separadamente.

A figura 32 mostra os sinais de controle do robô pêndulo. Como consta no capítulo 2, o controle LQR busca controlar a planta com custo mínimo, e conforme pode ser visto na figura 38, o sinal de controle LQR é sempre menor que o PID, além de conseguir equilibrar a planta com muito menos oscilação, conforme visto nos testes de resposta ao degrau.

Figura 38 – Resposta ao degrau do sistema formado por PID e Filtro Kalman



Fonte: .Própria

É interessante notar que os dois sinais vieram de ensaio diferentes. O sinal em azul veio de um ensaio realizado com a planta sendo controlado pelo controlador PID e o vermelho veio de um ensaio controlado pelo controlador LQR. Ambos os sinais de controle seguem o mesmo padrão porque o sistema é consideravelmente repetitivo.

Na tabela 8 consta o resumo das estatísticas de sinais de controle. Nela pode-se notar que o controlador LQR apresenta menos variância, ou seja, a energia do sinal de controle do controlador LQR é menor que a do PID. Isto mostra que o controlador LQR é muito mais eficiente para este projeto, uma vez que gera um sinal muito menor para controlar o robô, fazendo com que este oscile muito menos. O controlador PID além de gastar muito mais energia de controle, causa maior oscilação da planta, como já visto nos ensaios anteriores.

Tabela 8: Resumo das estatísticas dos sinais de controle

	PID	LQR
Media	0,1939	-0,6937
Variância	299,5785	99,8715

Fonte: Própria

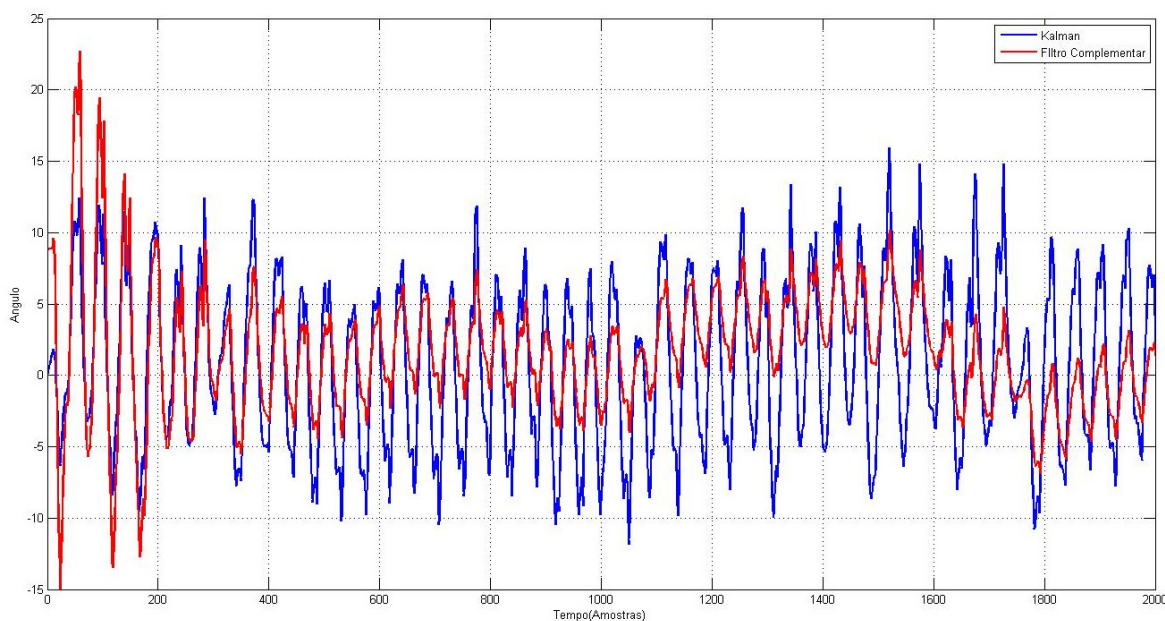
4.4 ANÁLISE DOS MÉTODOS DE FUSÃO DE SENSORES

Esta etapa de testes visa a análise dos métodos de fusão de sensores utilizados neste projeto (filtro Kalman e filtro complementar) na medição do ângulo e velocidade angular. Após, será feita uma análise das estatísticas dos sinais de ângulo e velocidade angular medidos pelos métodos.

4.4.1 ANÁLISE DA FUSÃO DE INFORMAÇÃO DE ÂNGULO ENTRE FILTRO KALMAN E FILTRO COMPLEMENTAR

A figura 39 mostra os sinais de ângulo fornecidos pelos algoritmos de fusão de sensores. Pode-se perceber que o sinal do filtro Kalman tem maior amplitude que o sinal de ângulo fornecido pelo filtro complementar. Isto se deve à frequência de corte do filtro passa baixas do filtro complementar. Por ter uma banda de frequência limitada, o filtro passa baixas não consegue seguir variações bruscas de ângulo. Conseqüentemente, a oscilação medida pelo filtro Kalman é maior, pois o mesmo não se baseia em limitações de frequência, mas sim em dados estatísticos dos sinais de ângulo e velocidade angular, bem como nas características do sistema.

Figura 39 – Sinais de ângulo estimado pelos métodos Filtro Kalman e filtro complementar



Fonte: .Própria

A tabela 9 traz os valores de variância dos sinais medidos pelos dois métodos. Como visto na figura 38 a energia do sinal do Kalman é maior, representando que este método mede uma banda de frequência maior que o filtro complementar.

Tabela 9: Resumo das variâncias dos sinais de ângulo e do Filtro Kalman e Filtro Complementar

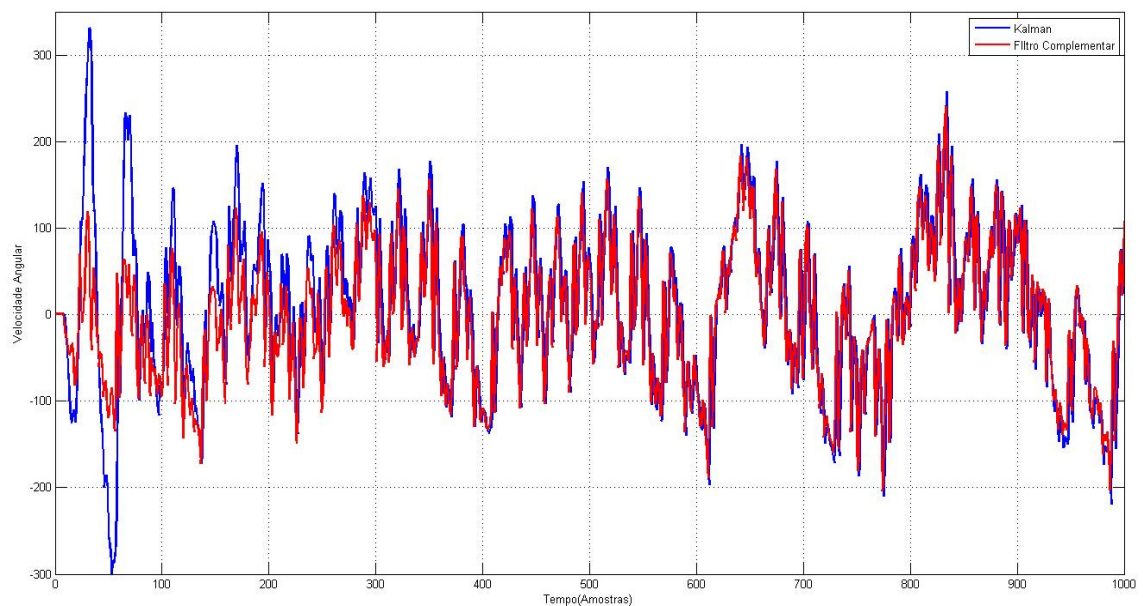
	Kalman	Filtro Complementar
Variância do ângulo	1,17497e-2	6,6637e-3

Fonte: Própria

4.4.2 ANÁLISE DA FUSÃO DE INFORMAÇÃO DE VELOCIDADE ANGULAR ENTRE FILTRO KALMAN E FILTRO COMPLEMENTAR

Nesta etapa pode-se verificar que os sinais de velocidade angular medidos pelo filtro Kalman e filtro complementar são bastante parecidos, como pode ser visto na figura 40. O sinal de velocidade angular passa por um filtro passa altas no filtro complementar. Ele raramente, durante o processo de controle de equilíbrio do robô, a velocidade angular apresenta sinais de baixas frequências. Porém, no momento de partida o filtro Kalman apresenta uma maior amplitude, isto se deve ao tempo de estabilização do algoritmo.

Figura 40 – Sinais de ângulo estimado pelos métodos Filtro Kalman e filtro complementar



Fonte: .Própria

Como pode se visto na tabela10 os valores de variância da velocidade angular calculadas pelos métodos não difere muito, visto que a banda de frequência cortada pelo filtro complementar da estimação da velocidade angular contem apenas a tendência de giro.

Tabela 10: Resumo das variâncias dos sinais de ângulo e velocidade angular

	Kalman	Filtro Complementar
Variância da velocidade angular	1,9135	1,5815

Fonte: Própria

5 CONCLUSÃO

Neste trabalho foi construído um robô móvel do tipo pêndulo invertido auto equilibrado sobre duas rodas. Foram realizados estudo e implementação de algoritmos de fusão de sensores aplicados aos sensores inerciais para estimar valores confiáveis de ângulo e velocidade angular. Também foram estudados e implementados sistemas de controle utilizados para manter o equilíbrio do robô na posição vertical.

O estudo de pêndulo invertido é importante pois aplica-se a vários objetos tais como o transportador pessoal *Segway*, foguetes, ou até mesmo estruturas de exoesqueleto que auxiliam pessoas com deficiência locomotora a caminhar. Além disso, o estudo de sistemas não linear e intrinsecamente instável fornece base para teste de algoritmos de controle cada vez mais sofisticados. Robôs auto equilibrados em duas rodas tem se tornado cada vez mais interessantes para a indústria, uma vez que permitem grande flexibilidade e eficiência no transporte de pessoas e/ou cargas. Assim o estudo para tornar o controle desta planta cada vez mais robusto e eficiente é uma tendência da indústria.

Para este projeto, o robô foi construído com dois motores DC com encoders acoplados, estrutura feita com placas de fibra de vidro e alimentado por oito pilhas recarregáveis. Foram utilizados acelerômetros e giroscópio como sensores para o equilíbrio do robô. E para processar os dados dos sensores e sistema de controle foi utilizado um processador da Texas Instruments da família Tiva, com núcleo ARM Cortex M4F com co-processador ponto flutuante.

Neste trabalho, a estimação dos valores de ângulo e velocidade angular foi realizada de duas formas pelos algoritmos de fusão de sensor: filtro complementar e filtro Kalman, para comparar seus resultados. Além disso, foram implementados os métodos de controle PID e LQR para controlar o robô e fazê-lo equilibrar-se na posição vertical. Os ensaios realizados nos algoritmos testados mostraram resultados muito bons no método de filtro Kalman e do controlador LQR. A combinação destes métodos produziu os melhores resultados de equilíbrio do robô.

Como melhoria para o projeto pode-se considerar a implementação do Controlador LGR (Regulador Linear Gaussiano), que consiste em um controlador LQR e um filtro Kalman que considere o modelo da planta usada. Para isto, faz-se necessário um processador mais rápido, pois requer que as matrizes do filtro Kalman sejam de tamanho 4×4 , o que torna o algoritmo mais complexo e com esforço computacional consideravelmente maior.

REFERÊNCIAS

- [1] D. Simon, “**Kalman Filtering.**” *Embedded Systems Programming*, 2001. Retrieved
- [2] Greg Welch, Gary Bishop, **An Introduction to the Kalman Filter**, September 17, 1997
- [3] R. E. KALMAN, **A New Approach to Linear Filtering and Prediction Problems**, Transactions of the ASME–Journal of Basic Engineering, 1960
- [4] Benoît CLEMENT, **Eléments de base pour comprendre le Filtre de Kalman**, Notes de Cours ENSTA – Bretagne, octobre 2010
- [5] Kelly, Alonzo (1994). "A **3D state space formulation of a navigation Kalman filter for autonomous vehicles**". DTIC Document: 13.
- [6] Reid, Ian; Term, Hilary. "**Estimation II**". www.robots.ox.ac.uk. Oxford University. Retrieved 6 August 2014.
- [7] Walter Fetter Lages, **Filtro Kalman**, Apostila Curso de Controle Digital, 7 de Junho de 2009 (<http://www.ece.ufrgs.br/~fetter/eng04037/kalman.pdf>)
- [8] Miguel Velhote Correia. **Análise de Movimento em Sequências de Imagens**. Master’s thesis, Faculdade de Engenharia da Universidade do Porto, February 1995.
- [9] Erik Cheever’s Home Page, professor at Department of Engineering of Swarthmore College, **An Introduction to escalar Kalman Filter**, www.swarthmore.edu/NatSci/echeeve1/Ref/Kalman/ScalarKalman.html acessado em 05/04/2015
- [10] H. W. Sorenson, Junho de 1970, Least-squares estimation from Gauss to Kalman
- [11] R.C Ooi, “Balancing a Two-Wheeled Autonomous Robot,” 2003. Retrieved January 18, 2009 from the World Wide Web:<http://robotics.ee.uwa.edu.au/theses/2003-Balance-Ooi.pdf>>.
- [12] José Luis Corona Miranda, “Application of Kalman Filtering and PID Control for Direct Inverted Pendulum Control”, 2009, California State University, Chico”.

[13] Texas instruments Training Technical Organization, **Getting Started ith TM4C123G LaunchPad Workshop**”, Revision 1.22 Novembro de 2013.

[14] Kristian Lauszus, “**A practical approach to Kalman filter and how to implement it**” setembro, 2012, TKJ Electronics

[15] Walter T. HIGGINS, JR, **A Comparison os complementary and Kalman Filter**”, IEEE Tranactions on Aerospace and Electronic System,Vol AES-11,NO.3 maio 1975.

[16] Jianhai Han, Xiangpan Li, and Qi Qin, **Design of Two-Wheeled Self-Balancing Robot Based on Sensor Fusion Algorithm**, Journal ref: International Journal of Automation Technology, Vol.8, No.2 pp. 216-221, 2014

[17] Ragnar Eide, Per Magne Egelid, Alexander Stamsø, Hamid Reza Karimi, **LQG Control Design for Balancing an Inverted Pendulum Mobile Robot.**, Intelligent Control and Automation, 2011, 2, 160-166.

[18] Sanghyuk Park, Jonathan How, **Examples of Estimation Filters from Recent Aircraft Projects at MIT**, November 2004.

[19] Microchip AN937 disponível em < <http://ww1.microchip.com/downloads/en/AppNotes/00937a.pdf>>

[20] Katsuhiko OGATA, **Discrete-Time Control Systems**, second edition, Prentice_Hall international, Inc.

[21] – Venema, Steven C. **A Kalman Filter Calibration Method For Analog Quadrature Position Encoders**. 1994. A thesis of Master of Science in Eltrical Engineering, University of Washington.

[22] Shane Colton, **The Balance Filter , A Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing Platform**, Junho 2007

[23] OlliW's Bastelseiten, IMU Data Fusing: Complementary, Kalman, and Mahony Filter, disponível em < <http://www.olliw.eu/2013/imu-data-fusing/>>

[24] BAZANELLA, Alexandre Sanfelice ; GOMES DA SILVA JR, João Manoel . **Sistemas de Controle: Princípios e Métodos de Projeto**. Porto Alegre: Editora da UFRGS, 2006

[25] Walter Fetter Lages, **Modelagem de Sistemas Discretos**, Apostila Curso de Controle Digital, 30 de agosto de 2011 (<http://www.ece.ufrgs.br/~fetter/eng04037/model.pdf>)

APÊNDICE -

Tabela de custos do Projeto

Componente	Quantidade	Valor unitário(R\$)
Kit LaunchPad	1	36,50
Motore ga25-371-12v-431rpm com encoder	2	33,50
MPU-9150	1	36,08
Rodas	2	22,95
Pilhas recarregáveis	8	11,00
Placa de hc788G	1	12,76
Cabos	50	0,1
Placa de fibra de vidro	2	5,76
Total		303,26