

TRABALHO DE GRADUAÇÃO

**SIMULAÇÃO E CONTROLE DE UM PÊNDULO
DUPLO INVERTIDO**

Carlos Wellington Passos Gonçalves

Brasília, dezembro de 2006

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

**SIMULAÇÃO E CONTROLE DE UM PÊNULO
DUPLO INVERTIDO**

Carlos Wellington Passos Gonçalves

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Mecatrônica

Banca Examinadora

Prof. Geovany Araújo Borges, UnB/ENE (Ori-
entador)

Profa. Flavia Maria Guerra de Sousa Aranha
Oliveira, UnB/ENE

Prof. João Yoshiyuki Ishihara, UnB/ENE

Dedicatória

Dedico este trabalho aos meus pais, Suzana e Carlos, e a minha irmã Liudimila.

Ao meu tio Gonçalo.

A todos meus familiares e amigos de Aracaju. Minha família aracajuana.

A todos meus amigos de Brasília. Minha família brasiliense.

Carlos Wellington Passos Gonçalves

Agradecimentos

Gostaria de agradecer principalmente a Deus por tudo que consegui para chegar até aqui.

Aos meus pais, que fizeram de tudo para que conseguisse buscar uma oportunidade de estudo melhor em Brasília.

A minha irmã Liudimila e meus primos, pelos anos inesquecíveis em Aracaju.

Ao meu tio Gonçalo, por me convencer a estudar para a última etapa do PAS.

A todos os meus familiares e amigos de Aracaju. Minha família aracajuana.

Ao professor Geovany Araújo Borges por acreditar que poderia fazer um bom trabalho de graduação.

A todos integrantes do LARA, companheiros e testemunhas do progresso desse trabalho.

A todos meus grandes amigos de Brasília. Minha família brasiliense.

Carlos Wellington Passos Gonçalves

RESUMO

Este trabalho consiste na simulação e controle de um pêndulo duplo invertido. Ele tem como objetivo fornecer uma ferramenta que auxilie na implementação física deste sistema, bem como fornecer informações sobre os tópicos aqui estudados. A não-linearidade intrínseca do sistema e seu controle sub-atuado aumentam ainda mais o desafio e a motivação para este trabalho. Por possuir estas características, o mesmo experimento pode ser utilizado em várias linhas de pesquisa, podendo mesmo auxiliar na composição de “laboratórios abertos”. Neste trabalho foram exploradas técnicas de controle e estimação de estados não comumente abordadas durante os cursos de graduação de Controle e Automação, tais como controle no espaço de estados, estimação no espaço de estados, controle ótimo e filtro de Kalman. Resultados satisfatórios foram verificados por meio de simulações, em que observou-se a minimização dos erros de controle e de estimação, das técnicas de controle ótimo e do Filtro de Kalman, respectivamente. Contudo, mesmo estas são prejudicadas por necessitarem de um modelo linearizado do pêndulo duplo invertido. Esta limitação explica porque existem certas configurações físicas e posições de equilíbrio em que não podemos controlar o sistema.

ABSTRACT

This work presents the modeling and simulation-based control of an inverted double pendulum. This tool will be used in the design of a the physical implementation, and in supplying informations on the topics studied here. The intrinsic non-linearity of the system and its under-actuated control increase even more the challenge and the motivation of this work. Due to these characteristics, the same experiment can be used in many research topics, and even be used in the composition of “open laboratories”. In this work there were applied techniques of control and state estimation which are not usually studied during the undergraduated course on Control and Automation. Namely, control in state-space, state estimation in state-space, optimal control and the Kalman Filter. Results obtained in simulations, in which it was verified succesful minimization of control and estimation errors, related to optimal control and Kalman Filter techniques, respectively. However, even these techniques are harmed for needing a linearized model of the inverted double pendulum. This limitation explains why there are certain physical configurations and positions of balance that cannot be controlled.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO E MOTIVAÇÃO	1
1.2	DEFINIÇÃO DOS PROBLEMAS PROPOSTOS	2
1.3	OBJETIVOS DO PROJETO	3
1.4	APRESENTAÇÃO DO MANUSCRITO	3
2	FUNDAMENTAÇÃO TEÓRICA	5
2.1	LINEARIZAÇÃO LOCAL DE SISTEMAS NÃO LINEARES	5
2.2	ESTIMADORES DE ESTADOS	6
2.2.1	ESTIMADORES DE ESTADOS	7
2.2.2	FILTRO DE KALMAN	8
2.3	CONTROLE NO ESPAÇO DE ESTADOS	12
2.3.1	ADICIONAR ENTRADA REFERÊNCIA	13
2.4	CONTROLE ÓTIMO	14
3	METODOLOGIAS E RESULTADOS	17
3.1	INTRODUÇÃO	17
3.2	MODELAGEM DO SISTEMA	17
3.3	APROXIMAÇÃO NUMÉRICA DO MODELO	21
3.4	ANÁLISE DO COMPORTAMENTO CAÓTICO DO PÊNDULO DUPLO INVERTIDO	24
3.5	LINEARIZAÇÃO DO SISTEMA	25
3.6	CONTROLADORES E ESTIMADORES DE ESTADO	28
3.6.1	CONTROLADOR NO ESPAÇO DE ESTADOS	28
3.6.2	CONTROLE ÓTIMO	29
3.6.3	ESTIMADOR CORRENTE	32
3.6.4	FILTRO DE KALMAN ESTENDIDO	33
3.7	ANÁLISE DE CONTROLABILIDADE E OBSERVABILIDADE	35
3.8	O SIMULADOR	40
4	CONCLUSÕES	44
	REFERÊNCIAS BIBLIOGRÁFICAS	45
	ANEXOS	46
I	DESCRIÇÃO DO CONTEÚDO DO CD	47

LISTA DE FIGURAS

1.1	Esquema de funcionamento do experimento “ <i>Ball&Beam</i> ” [1].	1
1.2	Esquema de funcionamento do pêndulo invertido [1].	2
1.3	Pêndulo duplo ideal [2].	2
1.4	Característica caótica do movimento do pêndulo duplo ideal [2]. Estes gráficos foram obtidos a partir de simulações matemáticas de pêndulos compostos por barras de mesma massa e comprimentos de 0.5 m [2].	3
1.5	Pêndulo duplo invertido na configuração “ <i>arm-driven</i> ” segundo [3].	3
2.1	Diagrama do algoritmo para o filtro de Kalman Discreto.	10
2.2	Diagrama do algoritmo para o filtro de Kalman Estendido.	12
2.3	Esquema da topologia de um sistema de controle com entrada referência e estimador de estados.	14
3.1	Os gráficos (a) e (b) correspondem respectivamente aos diagramas de corpo livre do $link_1$ e $link_2$.	17
3.2	Vetores unitários \hat{n}_1 e \hat{t}_1 para $link_1$.	19
3.3	Vetores unitários \hat{n}_2 e \hat{t}_2 para $link_2$.	19
3.4	Pêndulo duplo invertido obtido para simulação no MATLAB.	23
3.5	Os gráficos (a), (b), (c) e (d) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 e $\dot{\theta}_2$ obtidos durante simulação dos métodos de Euler e Runge-Kutta de quarta ordem.	24
3.6	Os gráficos (a), (b), (c) e (d) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 e $\dot{\theta}_2$ obtidos durante simulação dos métodos de Euler e Runge-Kutta de quarta ordem com amortecimento nas juntas.	25
3.7	Os gráficos (a), (b), (c) e (d) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 e $\dot{\theta}_2$ obtidos durante simulação do comportamento caótico do pêndulo duplo invertido ideal.	26
3.8	Os gráficos (a), (b), (c) e (d) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 e $\dot{\theta}_2$ obtidos durante simulação do comportamento caótico do pêndulo duplo invertido ideal com amortecimento nas juntas.	27
3.9	Os gráficos (a), (b), (c), (d) e (f) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 , $\dot{\theta}_2$ e u_k obtidos durante simulação do controle do pêndulo duplo invertido utilizando os coeficientes da Tabela (3.2).	30
3.10	Os gráficos (a), (b), (c), (d) e (f) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 , $\dot{\theta}_2$ e u_k obtidos durante simulação do controle do pêndulo duplo invertido utilizando os coeficientes da Tabela (3.2) e controle ótimo com os coeficientes $\rho = 1$ e $Q_2 = 10$.	31
3.11	Os gráficos (a), (b), (c), (d) e (f) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 , $\dot{\theta}_2$ e u_k obtidos durante simulação do controle do pêndulo duplo invertido utilizando os coeficientes da Tabela (3.2), controle ótimo com os coeficientes $\rho = 1$ e $Q_2 = 10$ e Estimador Corrente.	33
3.12	Os gráficos (a), (b), (c), (d) e (f) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 , $\dot{\theta}_2$ e u_k obtidos durante simulação do controle do pêndulo duplo invertido utilizando os coeficientes da Tabela (3.2), controle ótimo com os coeficientes $\rho = 1$ e $Q_2 = 10$ e Filtro de Kalman Estendido.	36
3.13	Os gráficos (a) e (b) correspondem respectivamente ao valor singular mínimo da matriz de controlabilidade CO e observabilidade OB a partir dos valores da Tabela (3.3).	37
3.14	Os gráficos (a) e (b) correspondem respectivamente ao valor singular mínimo da matriz de controlabilidade CO e observabilidade OB a partir dos valores da Tabela (3.4).	38
3.15	Os gráficos (a) e (b) correspondem respectivamente ao valor singular mínimo da matriz de controlabilidade CO e observabilidade OB a partir dos valores da Tabela (3.5).	39

3.16 Interface para o simulador.	41
3.17 Interface para o simulador utilizada para estudo de possível configuração mecânica.....	42
3.18 Os gráficos (a), (b), (c), (d) e (f) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 , $\dot{\theta}_2$ e u_k para a simulação cuja interface está representada na Figura (3.17).	43

LISTA DE TABELAS

3.1	Configuração para simulação dos métodos de Euler e Runge-Kutta de quarta ordem.	23
3.2	Configuração para simulação do controlador de espaço estados utilizando sensores para todas as variáveis de estado.	29
3.3	Configuração do pêndulo duplo invertido para análise da controlabilidade e observabilidade a partir da variação da posição de equilíbrio.	37
3.4	Configuração do pêndulo duplo invertido para análise da controlabilidade e observabilidade a partir da variação do comprimento das barras.	38
3.5	Configuração do pêndulo duplo invertido para análise da controlabilidade e observabilidade a partir da variação da massa das barras.	38
3.6	Configuração para simulação do controlador ótimo utilizando o Filtro de Kalman Estendido para análise do limite de controle do pêndulo duplo invertido.	39

LISTA DE SIMBOLOS

Símbolos Latinos

x	Vetor de estados do sistema	
y	Vetor de saídas do sistema	
z	Estimação para o vetor de saídas do sistema	
u	Vetor de sinais de controle do sistema	
u	Sinal de controle do sistema	
Φ	Matriz do sistema discretizado	
Γ	Matriz de entrada discretizada do sistema	
H	Matriz de saída do sistema	
L_p	Matriz de estimação preditiva	
L_c	Matriz de estimação corrente	
M	Matriz de ponderação da inovação do Filtro de Kalman	
K	Matriz de controle do sistema	
Q_1	Matriz ponderação de x para o controle ótimo	
Q_2	Matriz ponderação de u para o controle ótimo	
v	Ruído de medição	
w	Ruído de processo	
g	Aceleração gravitacional	$9,81 \text{ m/s}^2$

Símbolos Gregos

θ	ângulo	<i>radianos</i>
ω	velocidade angular	<i>radianos/s</i>
α	aceleração angular	<i>radianos/s²</i>

Subscritos

k	k-ésima iteração ou amostra
$otimo$	valor ótimo
aux	auxiliar

Sobrescritos

$\hat{}$	estimação
$-$	valor <i>a priori</i>
T	valor transposto

Siglas

ZOH	Zero Order Holder
LQR	Linear Systems Quadratic cost Regulator
SVD	Singular Valeu Decomposion

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO E MOTIVAÇÃO

A maneira que atualmente os cursos de Controle e Automação são ministrados ainda baseia-se primordialmente na resolução de problemas clássicos nas salas de aulas. Ou seja, apenas são resolvidos problemas já propostos com metodologias específicas para que o conhecimento sobre tais técnicas seja aplicado e assim mais facilmente fixado pelos alunos.

Contudo, ao serem utilizados modelos já prontos para aplicar técnicas de controle, as importantes etapas de modelamento e identificação do sistema a ser controlado são ignoradas. Para contemplar este aspecto, estão sendo utilizados cada vez mais experimentos em laboratório para que sejam estudados todos os aspectos de um projeto de controle, seja este utilizando técnicas clássicas ou modernas.

Cada vez mais está sendo utilizado o conceito de “laboratórios abertos”, nos quais os estudantes podem estudar os processos a serem controlados e buscar soluções próprias para os mesmos. São então analisadas questões como confiabilidade, erros estacionários, estabilidade, entre outras. Dessa forma, em um único espaço físico, é possível utilizar diversos experimentos para diferentes tipos de controle. Dentre estes podemos citar os controles: analógico, digital, não-linear, ótimo, robusto e adaptativo [1].

A implementação de ferramentas em softwares como MATLAB nos laboratórios é muito mais barata e até mesmo mais confiável do que experimentos mecânicos, contudo sistemas físicos possuem imperfeições que não são muitas vezes modeladas e não-linearidades que tornam ainda mais interessante e desafiadora a implementação do controle [1].

Com essas idéias em vista, estes experimentos podem ser cópias em escala de sistemas reais, como o caso do controle de um aeromodelo de helicóptero [1], ou podem ser experimentos que a primeira vista não possuem ligação direta com sistemas reais, como os experimentos do “*Ball&Beam*” e do pêndulo invertido [1].

O “*Ball&Beam*” é um experimento no qual é necessário equilibrar uma bola em cima de uma rampa de modo que a primeira fique no centro da última, explorando a característica não-linear do sistema. A Figura (1.1) mostra seu esquema de funcionamento [1].

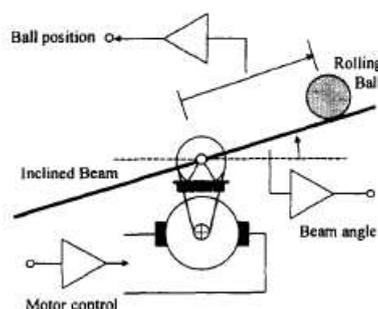


Figura 1.1: Esquema de funcionamento do experimento “*Ball&Beam*” [1].

O pêndulo invertido mostrado na Figura (1.2) também é um sistema altamente não linear. O objetivo de controle é que, através do movimento da sua base, seja possível equilibrá-lo na vertical e ainda deslocá-lo de um lado para outro.

Um experimento também muito interessante é o equilíbrio de um pêndulo duplo invertido. Além do movimento rotacional da barra inferior, que por sua vez é não-linear, também existe o movimento

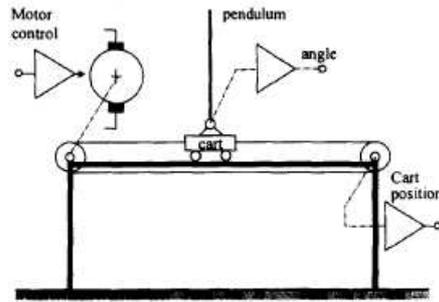


Figura 1.2: Esquema de funcionamento do pêndulo invertido [1].

translacional e rotacional da barra superior, que é ainda mais não-linear. Segundo [2, 4], o pêndulo duplo ideal, ou seja, que não possui qualquer amortecimento produzido nas juntas ou qualquer ação de controle, é um dos sistemas mecânicos que melhor descreve o caos. A Figura (1.3) mostra seu esquema mecânico.

A característica caótica do movimento é percebida quando o sistema é liberado para o movimento a partir de uma posição inicial θ_1 e θ_2 e percebemos que seu movimento é totalmente diferente de outro ensaio em que a posição inicial corresponde a θ_1 e $\theta_2 + 0.01$. Este fato é elucidado na Figura (1.4)

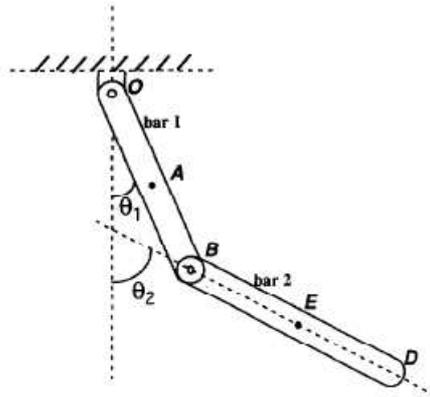


Figura 1.3: Pêndulo duplo ideal [2].

A configuração escolhida para o pêndulo duplo invertido que será estudado neste trabalho é mostrada na Figura(1.5). Com esta configuração subatuada podemos obter um experimento que pode ser utilizado no estudo de várias técnicas de controle, sejam estas clássicas ou não.

1.2 DEFINIÇÃO DOS PROBLEMAS PROPOSTOS

Primeiramente deve ser modelado o comportamento dinâmico do pêndulo duplo invertido, encontrando um sistema de equações diferenciais que determinam todas as suas variáveis de estado. A partir daí, deve ser obtida uma solução numérica para este sistema de forma que simulações já possam ser feitas no software MATLAB.

O estudo que será aqui feito sobre o controle e estimação de estados para o pêndulo duplo invertido necessita que o caracterizemos linearmente no espaço de estados. Deste modo é preciso fazer uma linearização local da resposta já encontrada para o sistema. Obtendo o modelo linear do sistema, podemos analisar a implementação do controle no espaço de estados, do controle ótimo, da estimação no espaço de estado e da estimação com Filtro de Kalman.

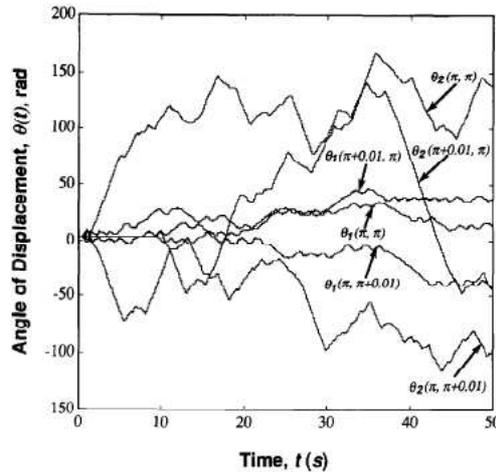


Figura 1.4: Característica caótica do movimento do pêndulo duplo ideal [2]. Estes gráficos foram obtidos a partir de simulações matemáticas de pêndulos compostos por barras de mesma massa e comprimentos de 0.5 m [2].

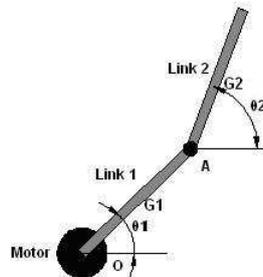


Figura 1.5: Pêndulo duplo invertido na configuração “arm-driven” segundo [3].

Com os algoritmos de controle e de estimação já implementados, será desenvolvido um simulador que dê a liberdade ao usuário de dimensionar o seu pêndulo e aplicar a ele as técnicas de controle e estimação já mencionadas, combinando-as da maneira que for conveniente.

1.3 OBJETIVOS DO PROJETO

Produzir um documento que possa ser futuramente utilizado na introdução aos estudos das técnicas de controle e estimação aqui utilizadas, já que as mesmas não são geralmente tópicos do estudo de graduação dos cursos de Controle e Automação. Contudo, principalmente, este trabalho deve propor um simulador confiável para que uma possível implementação física do pêndulo duplo invertido seja planejada com maior eficiência.

1.4 APRESENTAÇÃO DO MANUSCRITO

No capítulo 2 está disponível a fundamentação teórica utilizada para cada um dos tópicos mais pertinentes deste trabalho. Em seguida, o capítulo 3 descreve a metodologia empregada no desenvolvimento de cada etapa do projeto, juntamente com análise dos resultados obtidos e no caso de simulações, as suas

devidas interpretações. As conclusões que podem ser retiradas do projeto estão no capítulo 4.

Nos anexos estão o CD que acompanha este trabalho e as devidas informações sobre o mesmo.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 LINEARIZAÇÃO LOCAL DE SISTEMAS NÃO LINEARES

Muitas vezes os sistemas dinâmicos que necessitam de controle não são lineares, deste modo não é possível utilizar diretamente as técnicas de controle linear, necessitando então de uma aproximação linear do sistema.

Analisando um sistema com n estados, m sinais de controle e p saídas, em que os estados são representados por $x_i, i = 1 \dots n$, os sinais de controle por $u_i, i = 1 \dots m$, e as saídas por $y_i, i = 1 \dots p$. Se cada estado e cada saída forem representados por equações não-lineares, pode-se representar esse sistema com o conjunto de equações:

$$\dot{x}_1 = f_1(x_1, \dots, x_n, u_1, \dots, u_m, t), \quad (2.1)$$

$$\dot{x}_2 = f_2(x_1, \dots, x_n, u_1, \dots, u_m, t), \quad (2.2)$$

$$\vdots \quad (2.3)$$

$$\dot{x}_n = f_n(x_1, \dots, x_n, u_1, \dots, u_m, t), \quad (2.4)$$

$$y_1 = h_1(x_1, \dots, x_n, u_1, \dots, u_m, t), \quad (2.5)$$

$$y_2 = h_2(x_1, \dots, x_n, u_1, \dots, u_m, t), \quad (2.6)$$

$$\vdots \quad (2.7)$$

$$y_p = h_p(x_1, \dots, x_n, u_1, \dots, u_m, t). \quad (2.8)$$

Representando estas equações na forma matricial, tem-se que:

$$\dot{x} = \mathbf{f}(x, u, t), \quad (2.9)$$

$$y = \mathbf{h}(x, u, t). \quad (2.10)$$

Admitindo que as funções \mathbf{f} e \mathbf{h} são estacionárias, ou seja, dado um tempo inicial t_0 :

$$\mathbf{f}(x, u, t) = \mathbf{f}(x, u, t_0), \quad (2.11)$$

$$\mathbf{h}(x, u, t) = \mathbf{h}(x, u, t_0). \quad (2.12)$$

É obtida então uma nova simplificação do modelo mostrada abaixo [5].

$$\dot{x} = \mathbf{f}(x, u), \quad (2.13)$$

$$y = \mathbf{h}(x, u). \quad (2.14)$$

Admitindo que x e u são próximos de se seus valores iniciais x_0 e u_0 , é encontrado que:

$$\mathbf{f}(x_0, u_0) = 0. \quad (2.15)$$

Ao aproximar x e u pelas equações (2.16) e (2.17), são obtidas respectivamente.

$$x = x_0 + \delta x, \quad (2.16)$$

$$u = u_0 + \delta u. \quad (2.17)$$

Considerando δx e δu muito pequenos, pode-se desprezar termos de ordem superior dos mesmos. Com isso, realizando a expansão em séries de Taylor das equações (2.13) e (2.14), são obtidas as equações [5]:

$$\frac{d(\mathbf{x}_0 + \delta \mathbf{x})}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (2.18)$$

$$\dot{\delta \mathbf{x}} \approx \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) + \mathbf{f}_x(\mathbf{x}_0, \mathbf{u}_0)\delta \mathbf{x} + \mathbf{f}_u(\mathbf{x}_0, \mathbf{u}_0)\delta \mathbf{u}, \text{ como } \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) = 0, \quad (2.19)$$

$$\dot{\delta \mathbf{x}} \approx \mathbf{f}_x(\mathbf{x}_0, \mathbf{u}_0)\delta \mathbf{x} + \mathbf{f}_u(\mathbf{x}_0, \mathbf{u}_0)\delta \mathbf{u}, \quad (2.20)$$

$$(\mathbf{y}_0 + \delta \mathbf{y}) = \mathbf{h}(\mathbf{x}, \mathbf{u}), \quad (2.21)$$

$$(\mathbf{y}_0 + \delta \mathbf{y}) \approx \mathbf{h}(\mathbf{x}_0, \mathbf{u}_0) + \mathbf{h}_x(\mathbf{x}_0, \mathbf{u}_0)\delta \mathbf{x} + \mathbf{h}_u(\mathbf{x}_0, \mathbf{u}_0)\delta \mathbf{u}, \quad (2.22)$$

$$\text{deste modo } \mathbf{y}_0 = \mathbf{h}(\mathbf{x}_0, \mathbf{u}_0), \quad (2.23)$$

$$\delta \mathbf{y} \approx \mathbf{h}_x(\mathbf{x}_0, \mathbf{u}_0)\delta \mathbf{x} + \mathbf{h}_u(\mathbf{x}_0, \mathbf{u}_0)\delta \mathbf{u}. \quad (2.24)$$

A representação \mathbf{f}_x indica uma matriz onde cada linha i é representada pelo gradiente de $f_i(\mathbf{x}, \mathbf{u})$ em relação ao vetor \mathbf{x} . Como mostrado abaixo.

$$\mathbf{f}_x = \begin{bmatrix} \frac{df_1}{dx} \\ \frac{df_2}{dx} \\ \vdots \\ \frac{df_n}{dx} \end{bmatrix}. \quad (2.25)$$

A mesma regra segue para \mathbf{f}_u , \mathbf{h}_x e \mathbf{h}_u .

O modelo foi então linearizado, contudo este será agora analisado em função de δx , δu e δy . Com isto em mente, são feitas alterações nas equações (2.20) e (2.24) de modo a facilitar a sua utilização, voltando então a trabalhar com \mathbf{x} , \mathbf{u} e \mathbf{y} , obtendo [5]:

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u}, \quad (2.26)$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{J}\mathbf{u}, \quad (2.27)$$

$$\text{onde } \mathbf{F} = \mathbf{f}_x(\mathbf{x}_0, \mathbf{u}_0) \quad , \quad \mathbf{G} = \mathbf{f}_u(\mathbf{x}_0, \mathbf{u}_0), \quad (2.28)$$

$$\mathbf{H} = \mathbf{h}_x(\mathbf{x}_0, \mathbf{u}_0) \quad , \quad \mathbf{J} = \mathbf{h}_u(\mathbf{x}_0, \mathbf{u}_0). \quad (2.29)$$

2.2 ESTIMADORES DE ESTADOS

Muitas vezes não há acesso a todos os estados na saída do sistema, ou seja, não existem sensores para cada um destes. Desta forma, é preciso estimar os estados a partir do modelo do sistema e das medições que são obtidas na saída.

Tendo então um modelo em espaço de estados:

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u}, \quad (2.30)$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{J}\mathbf{u}. \quad (2.31)$$

Assumindo que será feito o controle digital do sistema com um tempo de amostragem de T segundos, e que o retentor é de ordem zero (ZOH, zero order holder), pode-se obter as equações abaixo [5]:

$$\mathbf{x}_{k+1} = \mathbf{\Phi}\mathbf{x}_k + \mathbf{\Gamma}u_k, \quad (2.32)$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{J}u_k, \quad (2.33)$$

$$\text{onde } \mathbf{\Phi} = e^{\mathbf{F}T}, \quad (2.34)$$

$$\mathbf{\Gamma} = \int_0^T e^{\mathbf{F}\mu} d\mu \mathbf{G}. \quad (2.35)$$

O índice x_k é a “k-ésima” amostra do estado x , u_k é a “k-ésima” amostra do sinal de controle u , e y_k é a “k-ésima” amostra de y . No decorrer deste trabalho será utilizada uma notação ainda mais simplificada por admitirmos que a saída y apenas depende dos estados, logo $J = 0$.

Serão estudadas duas abordagens para realizar a estimação dos estados, a Estimação no Espaço de Estados e o Filtro de Kalman.

2.2.1 Estimadores de Estados

A idéia principal da estimação em espaço de estados é obter uma estimação auxiliar a partir somente do modelo do sistema, e em seguida corrigi-la a partir de uma realimentação com as saídas do sistema [5]. Serão analisados aqui dois tipos de estimadores em espaço de estado: o preditivo e o corrente.

O Estimador Preditivo estima um estado \hat{x}_k a partir do seu estado anterior \hat{x}_{k-1} e de uma medição anterior y_{k-1} , enquanto que o Estimador Corrente estima um estado \hat{x}_k a partir do seu estado anterior \hat{x}_{k-1} e de uma medição corrente y_k . Dessa forma, o Estimador Corrente é mais preciso que o Estimador Preditivo.

2.2.1.1 Estimador preditivo

O Estimador Preditivo terá a seguinte formulação [5]:

$$\hat{x}_{k+1} = \Phi \hat{x}_k + \Gamma u_k + L_p (y_k - H \hat{x}_k). \quad (2.36)$$

Para calcular a matriz de estimação L_p deve-se ter em mente que esta deve controlar a dinâmica do erro de estimação e_k . Este erro pode ser calculado subtraindo a equação (2.36) da equação (2.32), obtendo:

$$e_k = x_k - \hat{x}_k, \quad (2.37)$$

$$= (\Phi - L_p H) e_{k-1}. \quad (2.38)$$

Então, é necessário encontrar uma matriz L_p que faça com que o erro convirja para zero a partir de qualquer e_0 . Deve-se escolher os n pólos $\beta_i, i = 1 \dots n$ no plano z para o estimador e igualar as equações (2.39) e (2.40).

$$|zI - \Phi + L_p H| = 0, \quad (2.39)$$

$$(z - \beta_1)(z - \beta_2) \cdots (z - \beta_n) = 0. \quad (2.40)$$

Igualando cada coeficiente $z^i, i = 0 \dots n$ da equação (2.39) aos da equação (2.40) será encontrado o valor de L_p . A escolha dos pólos deve acontecer de forma que estes sejam mais rápidos que os pólos do sistema controlado. Assim os pólos do estimador não irão influenciar na dinâmica do controlador [5].

Como os pólos do estimador não aparecem no sistema, ou seja, apenas produzem grandes valores nas equações feitas no equipamento que faz o controle digital, estes podem ser suficientemente rápidos e não provocarão um carregamento no atuador do sistema. Contudo, pólos muito rápidos nos estimadores deixam o sistema mais suscetível aos erros de medição que venham a ocorrer [5].

2.2.1.2 Estimador Corrente

O Estimador Preditivo é utilizado quando é necessário dar uma “folga” maior para o cálculo computacional da estimação, já que este utiliza y_{k-1} . Contudo, muitas vezes o tempo computacional necessário para a estimação é muito menor que o tempo de amostragem, logo pode-se utilizar o valor corrente da saída, utilizando então o Estimador Corrente [5].

Desta forma, o Estimador Corrente é modelado pelas equações:

$$\hat{\mathbf{x}}_k = \mathbf{x}_{aux} + \mathbf{L}_c(\mathbf{y}_k - \mathbf{H}\mathbf{x}_{aux}), \quad (2.41)$$

$$\mathbf{x}_{aux} = \Phi\hat{\mathbf{x}}_{k-1} + \Gamma u_{k-1}. \quad (2.42)$$

Substituindo a equação (2.42) na equação (2.41), é obtido:

$$\hat{\mathbf{x}}_k = (\Phi - \mathbf{L}_c\mathbf{H}\Phi)\hat{\mathbf{x}}_{k-1} + (\Gamma - \mathbf{L}_c\mathbf{H}\Gamma)u_{k-1} + \mathbf{L}_c\mathbf{y}_k. \quad (2.43)$$

Como a matriz \mathbf{L}_p , a matriz \mathbf{L}_c também será responsável por posicionar os pólos do erro de estimação, de modo que este convirja para zero partindo de qualquer e_0 . A expressão do erro e_k ficará então:

$$\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k, \quad (2.44)$$

$$= (\Phi - \mathbf{L}_c\mathbf{H}\Phi)\mathbf{e}_{k-1}. \quad (2.45)$$

É obtida então a equação característica do Estimador Corrente:

$$|z\mathbf{I} - \Phi + \mathbf{L}_c\mathbf{H}\Phi| = 0. \quad (2.46)$$

Deve-se utilizar os mesmos conceitos de escolha dos pólos do Estimador Preditivo para o Estimador Corrente. Logo, para um sistema de ordem n :

$$|z\mathbf{I} - \Phi + \mathbf{L}_c\mathbf{H}\Phi| = (z - \beta_1)(z - \beta_2) \cdots (z - \beta_n), \quad (2.47)$$

Ambas matrizes \mathbf{L}_p e \mathbf{L}_c são determinadas apenas uma única vez e devem ser rápidas o suficiente para que a estimação não atrapalhe o controle do sistema. Contudo, quanto mais rápidos os pólos de estimação, maior a sensibilidade do sistema a erros de medição. O Filtro de Kalman resolve este problema, já que avalia justamente as incertezas sobre o modelo e a saída do sistema.

2.2.2 Filtro de Kalman

O Filtro de Kalman foi proposto por Rudolph E. Kalman em 1960, e tinha como objetivo produzir um estimador de estados utilizando-se de uma “predição”, a partir de estados anteriores, e de uma “correção”, a partir da realimentação da saída do sistema. Com isso foi obtido um algoritmo recursivo que solucionou problemas de filtragem de dados discretos em sistemas lineares [6].

Com a utilização de computadores, que tornam a implementação do algoritmo prática, o Filtro de Kalman é amplamente utilizado em sistemas de navegação autônoma ou assistida. Sua principal característica é encontrar uma solução ótima que minimiza a covariância do erro de estimação. Em outras palavras, encontra uma solução que minimiza as incertezas a cerca de cada estimação [6].

2.2.2.1 Filtro de Kalman Discreto

O Filtro de Kalman assume que é possível estimar o estado \mathbf{x} de um processo controlado discreto no tempo a partir de uma equação estocástica com a seguinte forma [6]:

$$\mathbf{x}_k = \Phi\mathbf{x}_{k-1} + \Gamma u_{k-1} + \mathbf{w}_{k-1}. \quad (2.48)$$

E sua medição z é definida por:

$$z_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k. \quad (2.49)$$

Os índices w_k e v_k representam os ruídos presentes no processo e na etapa de medição, respectivamente. Estes ruídos serão considerados independentes entre si e terão a forma de “ruídos brancos”, ou seja, não há correlação entre os valores da mesma distribuição. Os ruídos serão aproximados por distribuições normais com média nula e variância não-nula. Esta é uma das melhores maneiras de modelar tais ruídos, já que pelo teorema central do limite a soma de variáveis aleatórias com qualquer distribuição tende a uma distribuição normal [6].

Desta forma, obtêm-se as distribuições de w e v :

$$p(\mathbf{w}) \sim N(0, \mathbf{Q}), \quad (2.50)$$

$$p(\mathbf{v}) \sim N(0, \mathbf{R}). \quad (2.51)$$

As matrizes \mathbf{Q} e \mathbf{R} são respectivamente as matrizes de covariância dos ruídos de processo e de medição.

A estimação do estado \mathbf{x}_k feita na etapa de “predição”, *a priori*, será representada por $\hat{\mathbf{x}}_k^-$. Já na etapa de “correção”, *a posteriori*, será obtida a estimação $\hat{\mathbf{x}}_k$. Com isso, é obtida a equação do erro *a priori*:

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-. \quad (2.52)$$

E a equação do erro *a posteriori*:

$$\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k. \quad (2.53)$$

Podem então ser definidas as matrizes \mathbf{P}_k^- e \mathbf{P}_k como sendo respectivamente as matrizes de covariância do erro *a priori* e *a posteriori* [6]:

$$\mathbf{P}_k^- = E(\mathbf{e}_k^- \mathbf{e}_k^{-T}), \quad (2.54)$$

$$\mathbf{P}_k = E(\mathbf{e}_k \mathbf{e}_k^T). \quad (2.55)$$

Onde e^T indica a transposta da matriz e .

Pela equação (2.48), pode-se obter [6]:

$$E(\mathbf{x}_k) = \hat{\mathbf{x}}_k, \quad (2.56)$$

$$E((\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T) = E(\mathbf{e}_k \mathbf{e}_k^T), \quad (2.57)$$

$$= \mathbf{P}_k. \quad (2.58)$$

Com isso, pode-se alterar a equação (2.48) para encontrar $\hat{\mathbf{x}}_k^-$ da seguinte forma.

$$\hat{\mathbf{x}}_k^- = \Phi \hat{\mathbf{x}}_{k-1} + \Gamma u_{k-1}. \quad (2.59)$$

E com essa fórmula, podemos encontrar:

$$E(\mathbf{e}_k^- \mathbf{e}_k^{-T}) = \Phi \mathbf{P}_{k-1} \Phi^T + \mathbf{Q}, \quad (2.60)$$

$$= \mathbf{P}_k^-. \quad (2.61)$$

Assim, foram encontrados todos os coeficientes *a priori* do sistema. É preciso estabelecer uma relação entre a estimação *a posteriori* e os demais coeficientes. Kalman propôs a seguinte equação [6]:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{M}_k (\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_k^-). \quad (2.62)$$

Na equação (2.62) a matriz \mathbf{M}_k pondera a inovação do sistema, diferença entre a medição \mathbf{z}_k e a estimativa da medição $\mathbf{H} \hat{\mathbf{x}}_k^-$. Podemos analisar que quanto melhor a estimação, ou seja, \mathbf{P}_k^- é cada vez mais próximo de zero, \mathbf{M}_k tenderá a zero [6].

O valor de M_k deve minimizar a covariância do erro *a posteriori*, e_k . Substituindo então a equação (2.62) na equação (2.53), calculando a sua covariância, obtendo a derivada do traço desta matriz com relação a M_k e igualando a zero, teremos o valor de M_k mostrado abaixo [6].

$$M_k = P_k^- H^T (H P_k^- H^T + R)^{-1}. \quad (2.63)$$

Como mencionado anteriormente, tem-se que:

$$\lim_{P_k^- \rightarrow 0} M_k = 0. \quad (2.64)$$

Uma outra análise pode ser feita quando a incerteza sobre a medição, matriz R , diminui. Desta forma, a estimação z_k deve ser mais utilizada no cálculo do estado \hat{x}_k . Logo o valor de M_k deve permitir que isto aconteça e este raciocínio é comprovado quando é calculado o limite abaixo.

$$\lim_{R \rightarrow 0} M_k = H^{-1}. \quad (2.65)$$

Neste caso M_k tem seu valor máximo, deixando

$$\hat{x}_k = H^{-1} z_k. \quad (2.66)$$

A justificativa para a equação (2.62) está no fato da mesma refletir a média da distribuição dos estados do sistema, sendo esta condicionada a partir de todas as medições z_k (regra de Bayes). Para este trabalho, basta a idéia de que ainda são mantidos os dois momentos da distribuição (equações (2.56) e (2.58)) [6].

Desta forma, o último elemento *a posteriori* a ser calculado, P_k , será [6]:

$$P_k = (I - M_k H) P_k^-. \quad (2.67)$$

Com esta equação é concluída a etapa de “correção” do algoritmo. Este pode então ser representado pelo diagrama da Figura (2.1).

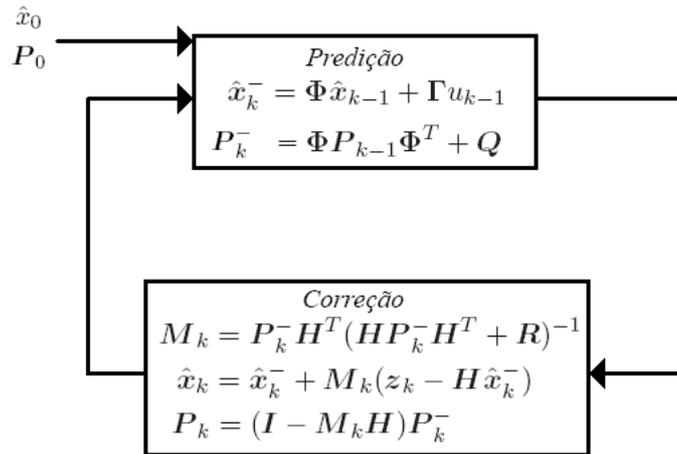


Figura 2.1: Diagrama do algoritmo para o filtro de Kalman Discreto.

A partir de uma estimação inicial \hat{x}_0 dos estados, e uma covariância inicial dos mesmos, P_0 , cada nova iteração do algoritmo nos levará a uma estimação ótima dos estados, ou seja, uma estimação na qual é minimizado seu erro.

2.2.2.2 Filtro de Kalman Estendido

Para sistemas nos quais não é possível representar os estados a serem controlados por uma equação estocástica linear, deve-se utilizar o filtro de Kalman Estendido.

Tendo então:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, u_{k-1}, \mathbf{w}_{k-1}), \quad (2.68)$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k). \quad (2.69)$$

Onde $f(\mathbf{x}_{k-1}, u_{k-1}, \mathbf{w}_{k-1})$ e $h(\mathbf{x}_k, \mathbf{v}_k)$ são não-lineares e \mathbf{w}_k e \mathbf{v}_k são respectivamente os ruídos presentes no processo e na etapa de medição. A distribuição de \mathbf{w}_k está representada na equação (2.50) e a distribuição de \mathbf{v}_k na equação (2.51). Assim, considerando as mesmas representações dos estados e covariâncias *a priori* e *a posteriori*, pode-se linearizar as equações (2.68) e (2.69) e obter [6]:

$$\mathbf{x}_k \approx \hat{\mathbf{x}}_k^- + \Phi_k(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^-) + \mathbf{W}_k \mathbf{w}_{k-1}, \quad (2.70)$$

$$\mathbf{z}_k \approx \hat{\mathbf{z}}_k^- + \mathbf{H}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \mathbf{V}_k \mathbf{v}_{k-1}. \quad (2.71)$$

Onde:

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}^-, u_{k-1}, 0), \quad (2.72)$$

$$\hat{\mathbf{z}}_k^- = h(\hat{\mathbf{x}}_k^-, 0), \quad (2.73)$$

$$\Phi_{k[i,j]} = \frac{\partial f^{[i]}}{\partial \mathbf{x}^{[j]}}(\hat{\mathbf{x}}_{k-1}^-, u_{k-1}, 0), \quad (2.74)$$

$$\mathbf{W}_{k[i,j]} = \frac{\partial f^{[i]}}{\partial \mathbf{w}^{[j]}}(\hat{\mathbf{x}}_{k-1}^-, u_{k-1}, 0), \quad (2.75)$$

$$\mathbf{H}_{k[i,j]} = \frac{\partial h^{[i]}}{\partial \mathbf{x}^{[j]}}(\hat{\mathbf{x}}_{k-1}^-, 0), \quad (2.76)$$

$$\mathbf{V}_{k[i,j]} = \frac{\partial h^{[i]}}{\partial \mathbf{v}^{[j]}}(\hat{\mathbf{x}}_{k-1}^-, 0). \quad (2.77)$$

Desta forma, além de uma estimativa *a priori* dos estados, $\hat{\mathbf{x}}_k^-$, também existe uma estimativa *a priori* da medição, aqui chamada de $\hat{\mathbf{z}}_k^-$. São então obtidos os erros referentes aos estados e à medição:

$$\mathbf{e}_{xk} = \mathbf{x}_k - \hat{\mathbf{x}}_k^-, \quad (2.78)$$

$$\approx \Phi(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^-) + \epsilon_k \text{ onde } p(\epsilon_k) \sim N(0, \mathbf{W}_k \mathbf{Q} \mathbf{W}_k^T), \quad (2.79)$$

$$\mathbf{e}_{zk} = \mathbf{z}_k - \hat{\mathbf{z}}_k^-, \quad (2.80)$$

$$\approx \mathbf{H}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \mu_k, \quad (2.81)$$

$$\approx \mathbf{H} \mathbf{e}_{xk} + \mu_k \text{ onde } p(\mu_k) \sim N(0, \mathbf{V}_k \mathbf{Q} \mathbf{V}_k^T). \quad (2.82)$$

O valor representado por \mathbf{e}_{zk} é justamente a inovação do modelo de Kalman, já que representa a diferença da medição atual, \mathbf{z}_k , e seu valor esperado, $\hat{\mathbf{z}}_k^-$, obtido por $h(\hat{\mathbf{x}}_k^-, 0)$.

A matrix de covariância de \mathbf{e}_{xk} é \mathbf{P}_k^- e terá o valor [6]:

$$\mathbf{P}_k^- = \Phi_k \mathbf{P}_{k-1} \Phi_k^T + \mathbf{W}_k \mathbf{Q} \mathbf{W}_k^T. \quad (2.83)$$

Pode-se obter uma relação entre $\hat{\mathbf{x}}_k$ e $\hat{\mathbf{x}}_k^-$ similar a equação (2.62) que minimize a matriz de covariância do erro \mathbf{e}_{xk} . Pode-se então ponderar a participação da inovação do sistema no cálculo de $\hat{\mathbf{x}}_k$. Deste modo, é obtida a equação abaixo [6]:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{M}_k(\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-, 0)). \quad (2.84)$$

A equação (2.84) foi escolhida pelo mesmo motivo da equação (2.62), ou seja, representar a média da distribuição condicionada a z_k de x_k . Com isso, os mesmos passos utilizados para encontrarmos as equações (2.63) e (2.67) foram utilizados para encontrarmos as equações (2.85) e (2.86) [6].

$$M_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R V_k^T)^{-1}, \quad (2.85)$$

$$P_k = (I - M_k H_k) P_k^-. \quad (2.86)$$

Com isso, já foram obtidas todas as equações necessárias para as etapas de “predição” e “correção”, tendo o algoritmo representado no diagrama da Figura (2.2). Este é muito semelhante ao da Figura(2.1), contudo é necessário calcular as matrizes Φ_k , H_k , W_k e V_k a cada iteração.

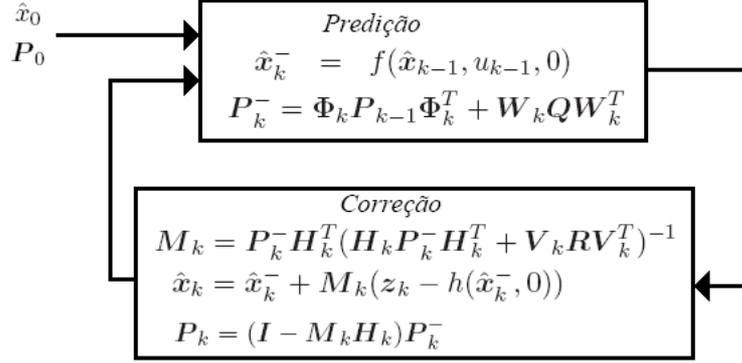


Figura 2.2: Diagrama do algoritmo para o filtro de Kalman Estendido.

2.3 CONTROLE NO ESPAÇO DE ESTADOS

O controle proporcional no espaço de estados é relativamente simples. Tendo um sistema representado da mesma maneira encontrada nas equações (2.32) e (2.33):

$$x_{k+1} = \Phi x_k + \Gamma u_k, \quad (2.87)$$

$$y_k = H x_k + J u_k. \quad (2.88)$$

Inicialmente será admitido que todas as variáveis de estado estão disponíveis na saída, depois serão adicionados os sinais dos estimadores e comprovado, pelo Princípio da Separação, que os pólos do controlador permanecerão os mesmos [5].

Deste modo, admitindo que ainda não é analisada uma entrada referência, a lei de controle do sistema será:

$$u_k = -K x_k. \quad (2.89)$$

Substituindo na equação (2.32), é encontrada a equação do estado x_k :

$$x_{k+1} = \Phi x_k - \Gamma K x_k, \quad (2.90)$$

$$= (\Phi - \Gamma K) x_k. \quad (2.91)$$

Para um sistema de ordem n , é necessário escolher os n pólos do sistema controlado $\beta_i, i = 1 \dots n$. Será então encontrado um valor para K de modo que os coeficientes de $z^i, i = 0 \dots n$ sejam os mesmos

nas equações (2.92) e (2.93).

$$|z\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{K}| = 0, \quad (2.92)$$

$$(z - \beta_1)(z - \beta_2) \cdots (z - \beta_n) = 0. \quad (2.93)$$

Contudo, quando é necessário o uso de estimadores, a realimentação será sobre os estados estimados. Modificando a lei de controle [5]:

$$u_k = -\mathbf{K}\hat{\mathbf{x}}_k. \quad (2.94)$$

Em que \mathbf{K} é o ganho de realimentação do sistemas em que todos os estados eram realimentados e mais uma vez não é considerada ainda uma entrada referência para o sistema. Deste modo, tomando o Estimador Preditivo como exemplo, tem-se:

$$\mathbf{x}_{k+1} = \mathbf{\Phi}\mathbf{x}_k - \mathbf{\Gamma}\mathbf{K}\hat{\mathbf{x}}_k, \quad (2.95)$$

$$= \mathbf{\Phi}\mathbf{x}_k - \mathbf{\Gamma}\mathbf{K}(\mathbf{x}_k - \mathbf{e}_k). \quad (2.96)$$

Onde \mathbf{e}_k é nosso erro de estimação. Utilizando a equação (2.38) são obtidas duas equações combinadas que representam todo o sistema [5].

$$\begin{bmatrix} \mathbf{e}_{k+1} \\ \mathbf{x}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} - \mathbf{L}_p\mathbf{H} & 0 \\ \mathbf{\Gamma}\mathbf{K} & \mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{e}_k \\ \mathbf{x}_k \end{bmatrix}. \quad (2.97)$$

Assim, a equação característica do sistema é dada por [5]:

$$\begin{vmatrix} z\mathbf{I} - \mathbf{\Phi} + \mathbf{L}_p\mathbf{H} & 0 \\ \mathbf{\Gamma}\mathbf{K} & z\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{K} \end{vmatrix} = 0, \quad (2.98)$$

$$|z\mathbf{I} - \mathbf{\Phi} + \mathbf{L}_p\mathbf{H}||z\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{K}| = \alpha_c(z)\alpha_e(z) = 0. \quad (2.99)$$

As raízes do polinômio $\alpha_c(z)$ são os pólos do sistema controlado (a partir da realimentação de todos os estados), e $\alpha_e(z)$ é o polinômio no qual as raízes são os pólos do estimador [5].

É possível então substituir nas equações (2.36) e (2.43) as devidas realimentações dos estados e obter os valores de $\hat{\mathbf{x}}_{k+1}$ respectivamente para o Estimador Preditivo e o Estimador Corrente. Substituindo a equação (2.59) em (2.62) e substituindo u_{k-1} temos o valor de $\hat{\mathbf{x}}_{k+1}$ para o Filtro de Kalman. Estas estimações estão mostradas abaixo [5]:

$$\text{Estimador Preditivo} \quad \hat{\mathbf{x}}_{k+1} = (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K} - \mathbf{L}_p\mathbf{H})\hat{\mathbf{x}}_k + \mathbf{L}_p\mathbf{y}_k, \quad (2.100)$$

$$\text{Estimador Corrente} \quad \hat{\mathbf{x}}_{k+1} = (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K} - \mathbf{L}_c\mathbf{H}\mathbf{\Phi} + \mathbf{L}_c\mathbf{H}\mathbf{\Gamma}\mathbf{K})\hat{\mathbf{x}}_k + \mathbf{L}_c\mathbf{y}_{k+1}, \quad (2.101)$$

$$\text{Filtro de Kalman} \quad \hat{\mathbf{x}}_{k+1} = (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K} - \mathbf{M}_k\mathbf{H}\mathbf{\Phi} + \mathbf{M}_k\mathbf{H}\mathbf{\Gamma}\mathbf{K})\hat{\mathbf{x}}_k + \mathbf{M}_k\mathbf{z}_{k+1}. \quad (2.102)$$

Pode ser observado que a fórmula para o estimador com Filtro de Kalman Discreto é bem similar a entrada para o Estimador Corrente, contudo vale lembrar que a matriz de ponderação da inovação \mathbf{M}_k deve ser recalculada para cada nova amostra k .

2.3.1 Adicionar entrada referência

Até agora apenas sistemas em que não há entrada referência foram estudados. Porém, muitas vezes é necessário que o sistema atue de forma a seguir uma referência, ou até seguir uma trajetória. A alteração da lei de controle para sistemas nos quais é necessária uma entrada referência não é difícil. Tomando inicialmente um sistema em que não é necessária a estimação de estados, tem-se:

$$\mathbf{x}_{k+1} = \mathbf{\Phi}\mathbf{x}_k + \mathbf{\Gamma}u_k, \quad (2.103)$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k. \quad (2.104)$$

O sinal de controle então será [5]:

$$u_k = -\mathbf{K}(\mathbf{x}_k - \mathbf{x}_r), \quad (2.105)$$

$$\text{onde } \mathbf{x}_r = \mathbf{N}_x \mathbf{r} \quad (2.106)$$

$$\text{e } \mathbf{y}_r = \mathbf{H} \mathbf{x}_r, \quad (2.107)$$

$$= \mathbf{H} \mathbf{N}_x \mathbf{r}. \quad (2.108)$$

É preciso encontrar a matriz \mathbf{N}_x que transforme a referência \mathbf{r} em um estado referência \mathbf{x}_r . Em seguida é necessário estipular \mathbf{r} de modo ao alcançarmos uma saída \mathbf{y}_r desejável.

Contudo, em muitos sistemas, o valor do sinal de controle em regime estacionário, u_{ss} , não será nulo. Deste modo, a equação (2.105) não estaria correta, já que o vetor de estado em regime estacionário \mathbf{x}_{ss} deveria ser \mathbf{x}_r , logo u_{ss} seria igual a zero.

Para resolver este problema, basta que adicionar na equação de u_k um termo que dependa diretamente da referência \mathbf{r} , tendo então [5]:

$$u_k = -\mathbf{K}(\mathbf{x}_k - \mathbf{x}_r) + u_{ss}, \text{ onde } u_{ss} = \mathbf{N}_u \mathbf{r}. \quad (2.109)$$

Se há um sistema em que é necessário o uso de estimadores, apenas é necessário alterar a equação (2.109) substituindo \mathbf{x}_k para a estimação utilizada.

O diagrama da Figura (2.3) mostra como fica a topologia do sistema de controle com entrada referência, controlador e estimador de estados.

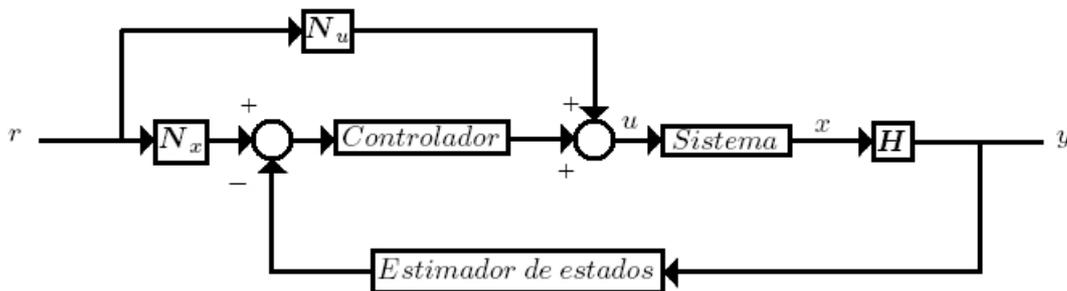


Figura 2.3: Esquema da topologia de um sistema de controle com entrada referência e estimador de estados.

2.4 CONTROLE ÓTIMO

O controle ótimo é caracterizado por encontrar uma solução ótima para a matriz \mathbf{K} que é encontrada na equação (2.89). Esta solução é ótima em relação a atenuação da variação do sinal de saída ou do vetor de estados. Desta forma o problema de escolha dos possíveis pólos do sistema controlado não está nas mãos do projetista. Isto ocorre porque o controle ótimo é baseado na procura de uma matriz \mathbf{K} que minimize uma função de custo [5]. Esta função pode ser representada como:

$$J = \frac{1}{2} \sum_{k=0}^N (\mathbf{x}_k^T \mathbf{Q}_1 \mathbf{x}_k + \mathbf{u}_k^T \mathbf{Q}_2 \mathbf{u}_k). \quad (2.110)$$

As matrizes Q_1 e Q_2 são diagonais e com elementos positivos ou nulos. Estas matrizes são responsáveis por ponderar respectivamente o quadrado dos termos de x_k e u_k . Com o aumento o valor da matriz Q_1 , está sendo forçada uma minimização maior da variação dos estados, já que os estados irão compor uma parte mais significativa no cálculo de J . A mesma idéia é válida para u_k quando é elevado o valor de Q_2 [5].

Deste modo, o projetista só precisa estabelecer o quando o controle estará preocupado em minimizar os estados e o sinal de controle. Os pólos encontrados para o sistema serão aqueles que resolverão o problema de forma ótima.

Uma das maneiras encontradas para solucionar esta equação de custo é a utilização de multiplicadores de Lagrange [5]. Tendo um sistema:

$$x_{k+1} = \Phi x_k + \Gamma u_k, \quad (2.111)$$

$$-x_{k+1} + \Phi x_k + \Gamma u_k = 0. \quad (2.112)$$

Como a equação (2.112) é nula, podemos multiplicá-la por um multiplicador de Lagrange λ_{k+1} e adicionar na equação (2.110) e obter [5]:

$$J' = \sum_{k=0}^N \left(\frac{1}{2} x_k^T Q_1 x_k + \frac{1}{2} u_k^T Q_2 u_k + \lambda_{k+1} (-x_{k+1} + \Phi x_k + \Gamma u_k) \right). \quad (2.113)$$

Deve-se então encontrar os valores de x_k , u_k e λ_{k+1} que minimizam a equação (2.113). Deste modo, tem-se:

$$\frac{\partial J'}{\partial u_k} = u_k^T Q_2 + \lambda_{k+1}^T \Gamma = 0, \quad (2.114)$$

$$\frac{\partial J'}{\partial \lambda_{k+1}} = -x_{k+1} + \Phi x_k + \Gamma u_k = 0, \quad (2.115)$$

$$\frac{\partial J'}{\partial x_k} = x_k^T Q_1 - \lambda_k^T + \lambda_{k+1}^T \Phi = 0. \quad (2.116)$$

É necessário então encontrar uma solução para as três equações abaixo:

$$x_{k+1} = \Phi x_k + \Gamma u_k, \quad (2.117)$$

$$u_k = -Q_2^{-1} \Gamma^T \lambda_{k+1}, \quad (2.118)$$

$$\lambda_{k+1} = \Phi^{-T} \lambda_k - \Phi^{-T} Q_1 x_k. \quad (2.119)$$

Desta maneira, apenas é necessário conhecer os valores iniciais x_0 , u_0 e λ_0 para resolver o problema, contudo não é conhecido o valor de λ_0 .

Analisando a equação (2.117), pode-se ver que u_N não influencia x_N , logo seu valor deve ser nulo de forma a minimizarmos a equação (2.113). Pela equação (2.118), temos então que $\lambda_{N+1} = 0$. Logo, pela equação (2.119):

$$\lambda_N = Q_1 x_N. \quad (2.120)$$

Contudo, ainda assim foi obtido o valor final λ_N , enquanto são conhecidos apenas o valores iniciais dos estados. Um método chamado *sweep method* desenvolvido por Bryson e Ho (1975) resolve este problema [5]. Ele assume que:

$$\lambda_k = S_k x_k. \quad (2.121)$$

Substituindo na equação (2.118) a equação (2.121), tem-se:

$$u_k = -(Q_2 + \Gamma^T S_{k+1} \Gamma)^{-1} \Gamma^T S_{k+1} \Phi x_k, \quad (2.122)$$

$$= -R^{-1} \Gamma^T S_{k+1} \Phi x_k, \quad (2.123)$$

$$\text{onde } R = Q_2 + \Gamma^T S_{k+1} \Gamma. \quad (2.124)$$

Substituindo agora na equação (2.119) e utilizando a equação (2.123), é obtido:

$$\mathbf{S}_k \mathbf{x}_k = \Phi^T \mathbf{S}_{k+1} \mathbf{x}_{k+1} + \mathbf{Q}_1 \mathbf{x}_k, \quad (2.125)$$

$$= \Phi^T \mathbf{S}_{k+1} (\Phi \mathbf{x}_k + \Gamma \mathbf{u}_k) + \mathbf{Q}_1 \mathbf{x}_k, \quad (2.126)$$

$$= \Phi^T \mathbf{S}_{k+1} (\Phi \mathbf{x}_k - \Gamma \mathbf{R}^{-1} \Gamma^T \mathbf{S}_{k+1} \Phi \mathbf{x}_k) + \mathbf{Q}_1 \mathbf{x}_k, \quad (2.127)$$

$$\text{agrupando: } (\mathbf{S}_k - \Phi^T \mathbf{S}_{k+1} \Phi + \Phi^T \mathbf{S}_{k+1} \Gamma \mathbf{R}^{-1} \Gamma^T \mathbf{S}_{k+1} \Phi - \mathbf{Q}_1) \mathbf{x}_k = 0. \quad (2.128)$$

Como esta equação deve ser nula para todo \mathbf{x}_k , é encontrado:

$$\mathbf{S}_k = \Phi^T (\mathbf{S}_{k+1} - \mathbf{S}_{k+1} \Gamma \mathbf{R}^{-1} \Gamma^T \mathbf{S}_{k+1}) \Phi + \mathbf{Q}_1, \quad (2.129)$$

$$= \Phi^T \mathbf{M}_{k+1} \Phi + \mathbf{Q}_1, \quad (2.130)$$

$$\text{onde } \mathbf{M}_{k+1} = (\mathbf{S}_{k+1} - \mathbf{S}_{k+1} \Gamma \mathbf{R}^{-1} \Gamma^T \mathbf{S}_{k+1}). \quad (2.131)$$

A equação (2.130) também é chamada de *equação de Riccati* e como esta não é linear, sua solução é difícil [5]. Contudo é possível encontrar os valores de \mathbf{S}_k através da recursividade da *equação de Riccati*. Para isso, é substituída a equação (2.121) na equação (2.120):

$$\mathbf{S}_N = \mathbf{Q}_1. \quad (2.132)$$

Como $\lambda_{N+1} = 0$, através da equação (2.121) pode-se perceber que $\mathbf{S}_{N+1} = 0$. Agora, apesar das iterações serem realizadas do ponto final para o inicial, podemos calcular todos termos \mathbf{S}_k e com estes substituir na equação (2.123) obtendo o sinal de controle. Modificando a mesma, tem-se:

$$\mathbf{u}_k = -\mathbf{R}^{-1} \Gamma^T \mathbf{S}_{k+1} \Phi \mathbf{x}_k, \quad (2.133)$$

$$\mathbf{u}_k = -\mathbf{K}_k \mathbf{x}_k, \quad (2.134)$$

$$\text{portanto } \mathbf{K}_k = \mathbf{R}^{-1} \Gamma^T \mathbf{S}_{k+1} \Phi. \quad (2.135)$$

Desta forma, é encontrado um ganho ótimo variante no tempo, e como seu cálculo não envolve qualquer valor inicial \mathbf{x}_0 , pode-se utilizá-lo para diversas condições iniciais de um mesmo sistema. Contudo, é preciso estipular um número N de iterações que levariam o sistema ao regime permanente [5].

Uma boa sugestão para escolha da matriz de ponderação \mathbf{Q}_1 é a seguinte [5]:

$$\mathbf{Q}_1 = \rho \mathbf{H}^T \mathbf{H}. \quad (2.136)$$

Dessa maneira, é encontrada uma matriz diagonal onde o multiplicador ρ pondera, da mesma maneira, todos os estados que compõe a saída.

Foi possível encontrar \mathbf{K}_k para cada iteração k até $k = N$. Entretanto, muitos sistemas devem funcionar por muito tempo, sistemas que controlam satélites, por exemplo, funcionam por anos. Deste modo, deve ser possível utilizar um método de controle ótimo que encontre um valor de \mathbf{K} que seja ótimo por tempo indeterminado. Este valor é chamado de \mathbf{K}_∞ pois seria o valor do qual \mathbf{K} se aproxima quando o tempo vai para infinito (regime permanente). A solução deste problema é chamada de *LQR Steady-State Optimal Control*, onde LQR está para *Linear systems Quadratic cost Regulator* [5]. Contudo para calcular \mathbf{K}_∞ deve-se encontrar \mathbf{S}_∞ através da equação:

$$\mathbf{S}_\infty = \Phi^T (\mathbf{S}_\infty - \mathbf{S}_\infty \Gamma \mathbf{R}^{-1} \Gamma^T \mathbf{S}_\infty) \Phi + \mathbf{Q}_1. \quad (2.137)$$

Esta por sua vez é chamada de *Equação algébrica de Riccati*. Resolvendo-a, é obtido o valor de \mathbf{S}_∞ , podendo então substituí-lo na equação (2.135) e obter o valor de \mathbf{K}_∞ . Contudo, muitas vezes não é possível encontrar solução para esta equação, sendo necessário recorrer a uma solução numérica para a mesma [5]. O MATLAB possui uma função *dlqr.m* que encontra o ganho ótimo para o regime estacionário.

Muitos dos softwares que solucionam a *Equação algébrica de Riccati* utilizam de um método chamado de *Decomposição em autovalores* [5]. Este método contudo não será abordado neste trabalho, apenas fica como sugestão a quem procurar entender mais sobre o assunto.

3 METODOLOGIAS E RESULTADOS

3.1 INTRODUÇÃO

Para o desenvolvimento desse trabalho, antes de tudo é preciso conhecer o modelo dinâmico do sistema ilustrado na Figura (1.5). A partir da modelagem dinâmica, pode-se cogitar o quanto não-linear o sistema realmente é, e a partir disso, pode-se escolher qual algoritmo numérico é capaz de representar a dinâmica real do pêndulo durante o seu movimento. Isto é necessário já que o pêndulo será representado a partir de um sistema de equações diferenciais.

Possuindo então uma simulação confiável, serão utilizados os tópicos já explicitados no capítulo 2 para desenvolver finalmente o controle do sistema.

Finalmente será desenvolvida uma plataforma de simulação que permite, antes de uma implementação prática do pêndulo duplo invertido como aqui estudado, testar as configurações físicas possíveis (comprimento das barras, massa das barras, torque máximo necessário ao motor, saturação aceitável do torque do motor, entre outras), bem como técnicas de controle.

3.2 MODELAGEM DO SISTEMA

A modelagem do sistema foi feita a partir de diagramas de corpo livre. A partir do sistema já representado na Figura (1.5), obtemos o diagrama de corpo livre para a barra inferior do pêndulo, que iremos chamar a partir de agora de $link_1$, e o diagrama de corpo livre para a barra superior, $link_2$.

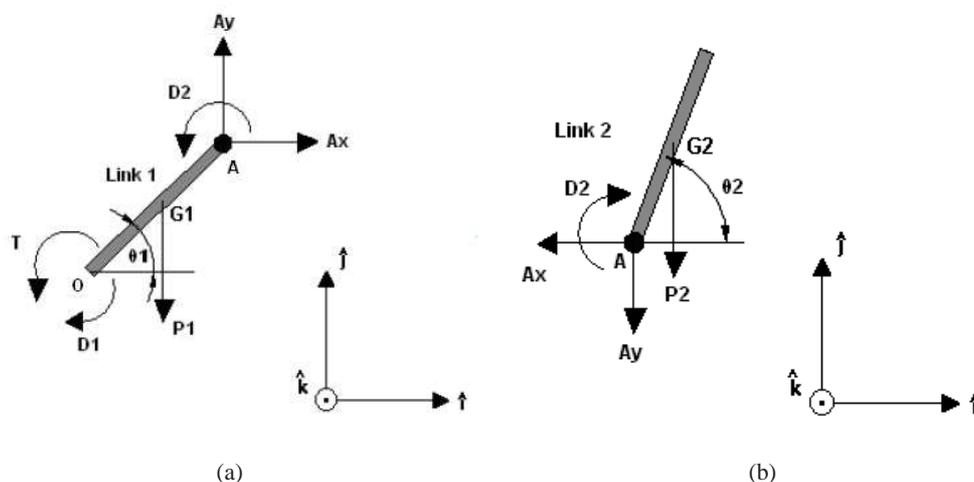


Figura 3.1: Os gráficos (a) e (b) correspondem respectivamente aos diagramas de corpo livre do $link_1$ e $link_2$.

O ponto **O** representa o eixo do motor a qual será conectada o $link_1$, o ponto **A** representa a junta que une as duas barras do pêndulo duplo. Os pontos **G1** e **G2** indicam os centros de massa para o $link_1$ e $link_2$, respectivamente. Será considerado que as barras possuem os centros de massa em seu centro geométrico.

Para o modelamento matemático, os ângulos de cada uma das barras em relação à horizontal serão representados por θ_1 e θ_2 , como mostrado nas Figuras (3.1(a)) e (3.1(b)). A velocidade angular, a aceleração angular, o comprimento e a massa de cada $link_i, i = 1 \dots 2$ serão representadas respectivamente por ω_i ,

α_i, l_i e m_i .

Serão considerados ainda o amortecimento provocado em cada junta pelos torques D_1 e D_2 , que são [7]:

$$D_1 = d_1\omega_1, \quad (3.1)$$

$$D_2 = d_2(\omega_2 - \omega_1). \quad (3.2)$$

Os coeficientes de amortecimento das juntas do ponto **O** e **A** são respectivamente d_1 e d_2 . Quando ambos coeficientes de amortecimento são nulos, estará sendo representado o comportamento do pêndulo duplo invertido ideal, no qual não há perda de energia.

Respeitando o sentido dos vetores unitários aqui utilizados (\hat{i} , \hat{j} e \hat{k}) e representando a aceleração gravitacional por g , pode-se obter as equações de movimento do sistema:

Diagrama de corpo livre do $link_1$.

$$\sum M_O : -A_x l_1 \sin(\theta_1) + A_y l_1 \cos(\theta_1) + T - d_1\omega_1 + d_2(\omega_2 - \omega_1) - \frac{m_1 g l_1}{2} \cos(\theta_1) = I_{O1} \alpha_1. \quad (3.3)$$

Diagrama de corpo livre do $link_2$.

$$\sum F_x : -A_x = a_{G2x} m_2, \quad (3.4)$$

$$\sum F_y : -A_y - m_2 g = a_{G2y} m_2, \quad (3.5)$$

$$\sum M_{G2} : \frac{-A_x l_2}{2} \sin(\theta_2) + \frac{A_y l_2}{2} \cos(\theta_2) - d_2(\omega_2 - \omega_1) = I_{G2} \alpha_2. \quad (3.6)$$

A equação (3.3) indica o somatório dos momentos aplicados ao $link_1$ com relação ao ponto **O** na direção do vetor unitário \hat{k} . O momento de inércia do $link_1$ em relação ao ponto **O** é dado por I_{O1} e o torque aplicado pelo motor é T .

A equação (3.4) indica o somatório de forças aplicadas no $link_2$ na direção do vetor unitário de \hat{i} . A equação (3.5) representa também o somatório de forças no $link_2$, agora na direção do vetor unitário \hat{j} . E finalmente, a equação (3.6) representa o somatório dos momentos aplicados com relação ao centro de massa do $link_2$ na direção do vetor unitário \hat{k} .

O momento de inércia do $link_2$ em relação ao seu centro de massa **G2** é I_{G2} . Também estão representadas as acelerações a_{G2x} e a_{G2y} do centro de massa do $link_2$ com direções \hat{i} e \hat{j} , respectivamente.

Contudo, apenas estas equações não são suficientes para resolução do sistema. Deste modo, é necessário encontrar outras representações de a_{G2x} e a_{G2y} para então encontrar os valores de A_x e A_y (forças presentes no ponto **A**). Para isso será utilizado o conceito de aceleração relativa.

Primeiramente, é preciso encontrar a aceleração linear do ponto **A**, a_A . Tem-se que:

$$a_A = a_O + a_{A/O}. \quad (3.7)$$

A aceleração do ponto **O**, a_O , já sabemos que é nula, pois este não se movimenta. A aceleração relativa entre o ponto **A** e o ponto **O** representa a aceleração de **A** para um observador em **O** [8]. Para calculá-la, é necessário utilizar os vetores unitários \hat{n} e \hat{t} , que representam respectivamente a direção normal de tangente do movimento de um ponto no espaço [8]. A Figura (3.2) representa os vetores \hat{n}_1 e \hat{t}_1 que ajudarão na determinação de a_A .

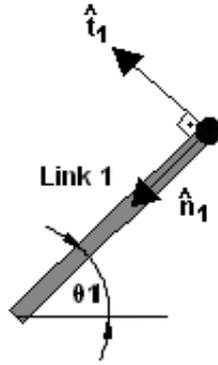


Figura 3.2: Vetores unitários \hat{n}_1 e \hat{t}_1 para $link_1$.

O vetor \hat{n}_1 representa a porção da aceleração centrípeta do ponto **A**, enquanto que \hat{t}_1 a tangencial. Temos então que:

$$a_A = a_O + a_{A/O}, \quad (3.8)$$

$$a_A = l_1 \omega_1^2 \hat{n}_1 + l_1 \alpha_1 \hat{t}_1 \quad , \text{ decompondo nos vetores } \hat{i} \text{ e } \hat{j}, \quad (3.9)$$

$$a_A = l_1 (-\omega_1^2 \cos(\theta_1) - \alpha_1 \sin(\theta_1)) \hat{i} + l_1 (-\omega_1^2 \sin(\theta_1) + \alpha_1 \cos(\theta_1)) \hat{j}. \quad (3.10)$$

Com o valor de a_A , é possível agora encontrar a_{G2} :

$$a_{G2} = a_A + a_{G2/A}. \quad (3.11)$$

A Figura (3.3) indica os vetores unitários \hat{n}_2 e \hat{t}_2 que serão utilizados para definir $a_{G2/A}$.

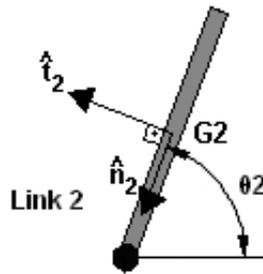


Figura 3.3: Vetores unitários \hat{n}_2 e \hat{t}_2 para $link_2$.

Então:

$$a_{G2/A} = \frac{l_2}{2} (-\omega_2^2 \cos(\theta_2) - \alpha_2 \sin(\theta_2)) \hat{i} + \frac{l_2}{2} (-\omega_2^2 \sin(\theta_2) + \alpha_2 \cos(\theta_2)) \hat{j}, \quad (3.12)$$

$$\text{portanto } a_{G2} = (-l_1 \omega_1^2 \cos(\theta_1) - l_1 \alpha_1 \sin(\theta_1) - \frac{l_2 \omega_2^2}{2} \cos(\theta_2) - \frac{l_2 \alpha_2}{2} \sin(\theta_2)) \hat{i} + (-l_1 \omega_1^2 \sin(\theta_1) + l_1 \alpha_1 \cos(\theta_1) - \frac{l_2 \omega_2^2}{2} \sin(\theta_2) + \frac{l_2 \alpha_2}{2} \cos(\theta_2)) \hat{j}. \quad (3.13)$$

Agora são conhecidos os valores de a_{G2x} e a_{G2y} :

$$a_{G2x} = (-l_1\omega_1^2 \cos(\theta_1) - l_1\alpha_1 \sin(\theta_1) - \frac{l_2\omega_2^2}{2} \cos(\theta_2) - \frac{l_2\alpha_2}{2} \sin(\theta_2)), \quad (3.14)$$

$$a_{G2y} = (-l_1\omega_1^2 \sin(\theta_1) + l_1\alpha_1 \cos(\theta_1) - \frac{l_2\omega_2^2}{2} \sin(\theta_2) + \frac{l_2\alpha_2}{2} \cos(\theta_2)). \quad (3.15)$$

Pode-se então substituir respectivamente os valores de a_{G2x} e a_{G2y} nas equações (3.4) e (3.5). Obtendo:

$$A_x = m_2(l_1\omega_1^2 \cos(\theta_1) + l_1\alpha_1 \sin(\theta_1) + \frac{l_2\omega_2^2}{2} \cos(\theta_2) + \frac{l_2\alpha_2}{2} \sin(\theta_2)), \quad (3.16)$$

$$A_y = m_2(l_1\omega_1^2 \sin(\theta_1) - l_1\alpha_1 \cos(\theta_1) + \frac{l_2\omega_2^2}{2} \sin(\theta_2) - \frac{l_2\alpha_2}{2} \cos(\theta_2) - g). \quad (3.17)$$

Substituindo os valores de A_x e A_y nas equações (3.3) e (3.6), e utilizando algumas identidades trigonométricas:

$$\begin{aligned} I_{O1}\alpha_1 &= T - d_1\omega_1 + d_2(\omega_2 - \omega_1) - \frac{m_1l_1g}{2} \cos(\theta_1) - m_2l_1g \cos(\theta_1) \\ &+ m_2(-l_1^2\alpha_1 + \frac{l_1l_2}{2}(\omega_2^2 \sin(\theta_2 - \theta_1) - \alpha_2 \cos(\theta_2 - \theta_1))), \end{aligned} \quad (3.18)$$

$$\begin{aligned} I_{G2}\alpha_2 &= -d_2(\omega_2 - \omega_1) - \frac{m_2l_2g}{2} \cos(\theta_2) \\ &+ m_2(-\frac{l_2^2\alpha_2}{4} + \frac{l_1l_2}{2}(-\omega_1^2 \sin(\theta_2 - \theta_1) - \alpha_1 \cos(\theta_2 - \theta_1))). \end{aligned} \quad (3.19)$$

Resolvendo um sistema com estas duas equações, obtêm-se:

$$\alpha_1 = \frac{(m_2l_2^2 + 4I_{G2})N_1 - (2m_2l_1l_2 \cos(\theta_2 - \theta_1))N_2}{D}, \quad (3.20)$$

$$\alpha_2 = \frac{-(2m_2l_1l_2 \cos(\theta_2 - \theta_1))N_1 + 4(m_2l_1^2 + I_{O1})N_2}{D}, \quad (3.21)$$

$$\begin{aligned} \text{onde } N_1 &= -T + d_1\omega_1 - d_2(\omega_2 - \omega_1) + \frac{1}{2}m_1l_1g \cos(\theta_1) \\ &+ m_2l_1 \cos(\theta_1) - \frac{1}{2}m_2l_1l_2\omega_2^2 \sin(\theta_2 - \theta_1), \end{aligned} \quad (3.22)$$

$$N_2 = d_2(\omega_2 - \omega_1) + \frac{1}{2}m_2l_2g \cos(\theta_2) + \frac{1}{2}m_2l_1l_2\omega_1^2 \sin(\theta_2 - \theta_1), \quad (3.23)$$

$$D = -m_2^2l_1^2l_2^2 - 4m_2l_1^2I_{G2} - I_{O1}m_2l_2^2 - 4I_{O1}I_{G2} + m_2^2l_1^2l_2^2 \cos^2(\theta_2 - \theta_1). \quad (3.24)$$

O momento de inércia de cada uma das barras do pêndulo depende diretamente de seu formato. Daqui em diante, estas barras serão analisadas como “barras esguias”, ou seja, não possuem espessura. Deste modo, pode-se substituir em ambas equações (3.20) e (3.21) os valores dos momentos de inércia $I_{O1} = m_1l_1^2/3$ e $I_{G2} = m_2l_2^2/12$ [8]. Deste modo:

$$\begin{aligned} \alpha_1 &= \frac{-12N_1}{l_1^2(12m_2 + 4m_1 - 9m_2 \cos^2(\theta_2 - \theta_1))} \\ &+ \frac{18 \cos(\theta_2 - \theta_1)N_2}{l_1l_2(12m_2 + 4m_1 - 9m_2 \cos^2(\theta_2 - \theta_1))}, \end{aligned} \quad (3.25)$$

$$\begin{aligned} \alpha_2 &= \frac{18 \cos(\theta_2 - \theta_1)N_1}{l_1l_2(12m_2 + 4m_1 - 9m_2 \cos^2(\theta_2 - \theta_1))} \\ &- \frac{12(3m_2 + m_1)N_2}{m_2l_2^2(12m_2 + 4m_1 - 9m_2 \cos^2(\theta_2 - \theta_1))}. \end{aligned} \quad (3.26)$$

Como $\omega_i = \dot{\theta}_i$, $\alpha_i = \dot{\omega}_i = \ddot{\theta}_i$ e $i = 1 \dots 2$, o sistema pode ser representado através das equações diferenciais:

$$\mathbf{f}(\boldsymbol{\theta}, T) = \dot{\boldsymbol{\theta}} \quad (3.27)$$

$$f_1(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T) = \dot{\theta}_1, \quad (3.28)$$

$$\begin{aligned} f_2(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T) &= \ddot{\theta}_1, \\ &= \frac{-12N_3}{l_1^2(12m_2 + 4m_1 - 9m_2 \cos^2(\theta_2 - \theta_1))} \\ &+ \frac{18 \cos(\theta_2 - \theta_1)N_4}{l_1 l_2 (12m_2 + 4m_1 - 9m_2 \cos^2(\theta_2 - \theta_1))}, \end{aligned} \quad (3.29)$$

$$f_3(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T) = \dot{\theta}_2, \quad (3.30)$$

$$\begin{aligned} f_4(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T) &= \ddot{\theta}_2, \\ &= \frac{18 \cos(\theta_2 - \theta_1)N_3}{l_1 l_2 (12m_2 + 4m_1 - 9m_2 \cos^2(\theta_2 - \theta_1))} \\ &- \frac{12(3m_2 + m_1)N_4}{m_2 l_2^2 (12m_2 + 4m_1 - 9m_2 \cos^2(\theta_2 - \theta_1))}, \end{aligned} \quad (3.31)$$

$$\begin{aligned} \text{onde } N_3 &= -T + d_1 \dot{\theta}_1 - d_2 (\dot{\theta}_2 - \dot{\theta}_1) + \frac{1}{2} m_1 l_1 g \cos(\theta_1) \\ &+ m_2 l_1 \cos(\theta_1) - \frac{1}{2} m_2 l_1 l_2 \dot{\theta}_2^2 \sin(\theta_2 - \theta_1), \end{aligned} \quad (3.32)$$

$$N_4 = d_2 (\dot{\theta}_2 - \dot{\theta}_1) + \frac{1}{2} m_2 l_2 g \cos(\theta_2) + \frac{1}{2} m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_2 - \theta_1). \quad (3.33)$$

Pode-se observar que as variáveis de estado do sistema são exatamente θ_1 , $\dot{\theta}_1$, θ_2 e $\dot{\theta}_2$, já que é possível modelar todo o sistema a partir das mesmas. É necessário agora determinar um algoritmo numérico para resolver este sistema de equações diferenciais e então simular o pêndulo ainda sem controle.

A fim de serem utilizados estimadores de estados, nem todas as variáveis de estado serão medidas através de sensores. Por isso, deve-se definir um vetor saída do sistema \mathbf{y} que deverá ter as duas variáveis que desejamos controlar : θ_1 e θ_2 .

Deste modo:

$$\mathbf{y} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \mathbf{H} \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix}, \quad (3.34)$$

$$\text{onde } \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (3.35)$$

3.3 APROXIMAÇÃO NUMÉRICA DO MODELO

Com o modelo em mãos, é preciso analisar qual algoritmo numérico pode dar uma resposta confiável ao sistema de equações diferenciais.

3.3.0.1 Método de Euler

O Método de Euler utilizará a informação sobre a derivada da função, calculada no tempo t , para aproximar o valor da função no tempo $t + h$ [9]. Dessa forma, é necessário representar as derivadas das

variáveis de estado do nosso sistema. Tem-se:

$$\frac{d\theta_1}{dt} = f_1(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T), \quad (3.36)$$

$$\frac{d^2\theta_1}{dt^2} = f_2(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T), \quad (3.37)$$

$$\frac{d\theta_2}{dt} = f_3(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T), \quad (3.38)$$

$$\frac{d^2\theta_2}{dt^2} = f_4(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T). \quad (3.39)$$

Tendo o vetores \mathbf{x} e $\mathbf{f}(\mathbf{x}, T)$:

$$\mathbf{x} = \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix}, \quad \mathbf{f}(\mathbf{x}, T) = \begin{bmatrix} f_1(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T) \\ f_2(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T) \\ f_3(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T) \\ f_4(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, T) \end{bmatrix}. \quad (3.40)$$

A função $\mathbf{f}(\mathbf{x}, T)$ não é variável no tempo, logo, desde já, é possível discretizar a progressão dos estados \mathbf{x} . Deste modo, um estado \mathbf{x}_k ocorre no tempo t , enquanto que o estado \mathbf{x}_{k+1} ocorre em $t + Ts$. O intervalo Ts é o tempo de amostragem do sistema.

O torque aplicado pelo motor no $link_1$ também pode ser modelado como T_k , contudo, ainda não será aplicado torque ao sistema, já que os valores aplicados devem ser obtidos através dos métodos de controle.

A partir daí, é obtido por meio do método de Euler a primeira solução numérica para o sistema de equações diferenciais [9].

$$\mathbf{x}_{k+1} = \mathbf{x}_k + Ts\mathbf{f}(\mathbf{x}_k, T_k). \quad (3.41)$$

3.3.0.2 Método de Runge-Kutta de quarta ordem

O método de Euler é aplicado sem grandes perdas quando o sistema apresenta uma variação não muito acentuada, ou seja, seu movimento é mais suave. Contudo, o pêndulo duplo invertido ideal é um sistema extremamente caótico [2], portanto, é necessário analisar outras possibilidades de solução numérica do sistema de equações diferenciais.

Uma solução numérica utilizando derivadas de ordem superior produzirá uma resposta mais precisa, contudo seriam necessárias as fórmulas dessas derivadas. Contudo, como ainda não foi obtida a resposta para o sistema, seus valores não são conhecidos.

O método de Runge-Kutta de quarta ordem utiliza-se da derivada primeira da função para aproximar as derivadas de ordem 2, 3 e 4 [10]. Isso é feito por meio do cálculo da derivada da função para o ponto inicial \mathbf{x}_k , o ponto final \mathbf{x}_{k+1} e mais dois pontos entre eles.

Aplicando o algoritmo ao nosso sistema é obtida a segunda solução numérica para o sistema de equações diferenciais [9]:

$$\mathbf{k}_1 = \mathbf{f}(\mathbf{x}_k, T_k), \quad (3.42)$$

$$\mathbf{k}_2 = \mathbf{f}\left(\mathbf{x}_k + \frac{\mathbf{k}_1}{2}, T_k\right), \quad (3.43)$$

$$\mathbf{k}_3 = \mathbf{f}\left(\mathbf{x}_k + \frac{\mathbf{k}_2}{2}, T_k\right), \quad (3.44)$$

$$\mathbf{k}_4 = \mathbf{f}(\mathbf{x}_k + \mathbf{k}_3, T_k), \quad (3.45)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4). \quad (3.46)$$

A maior precisão da resposta não pode ser garantida ao utilizar-mos o método de Runge-Kutta, contudo, na maioria dos casos ela é superior ao método de Euler [9].

Para analisar a diferença das duas respostas foi desenvolvido uma simulação em MATLAB. A simulação foi feita baseando-se numa configuração de pêndulo duplo invertido que poderia ser implementada fisicamente. Esta configuração e os valores iniciais dos estados estão representados na Tabela (3.1). Será utilizada uma aceleração gravitacional $g = 9,81m/s^2$ para esta e todas as demais simulações.

Tabela 3.1: Configuração para simulação dos métodos de Euler e Runge-Kutta de quarta ordem.

m_1 (kg)	0,3
m_2 (kg)	0,5
l_1 (m)	0,3
l_2 (m)	0,3
d_1 (N.m.s)	0
d_2 (N.m.s)	0
T_s (s)	0,001
$\theta_{1inicial}$ (rad)	$\frac{\pi}{4}$
$\dot{\theta}_{1inicial}$ (rad/s)	0
$\theta_{2inicial}$ (rad)	$\frac{\pi}{2}$
$\dot{\theta}_{2inicial}$ (rad/s)	0

A Figura (3.4) mostra como é representado o pêndulo duplo invertido através da simulação no MATLAB nas configurações da Tabela (3.1).

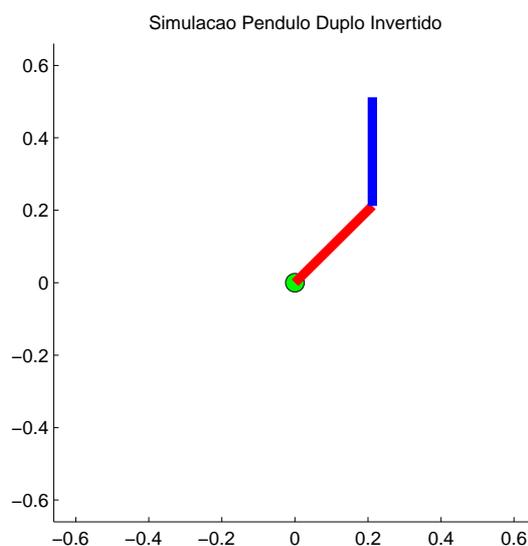


Figura 3.4: Pêndulo duplo invertido obtido para simulação no MATLAB.

Nas Figuras (3.5(a)), (3.5(b)) (3.5(c)) e (3.5(d)) estão respectivamente os valores encontrados durante a simulação para θ_1 , $\dot{\theta}_1$, θ_2 e $\dot{\theta}_2$.

Em azul estão os valores obtidos com o Método de Euler, e em vermelho os obtidos com o Método de Runge-Kutta de quarta ordem. Em verde é representada a resposta do sistema obtida com o Método de Runge-Kutta, porém com $T_s = 0,0001s$.

Pode-se ver que as curvas verde e vermelha seguem juntas até aproximadamente 6 segundos, enquanto que o Método de Euler em menos de um segundo já havia se distanciado e acaba completamente diferente

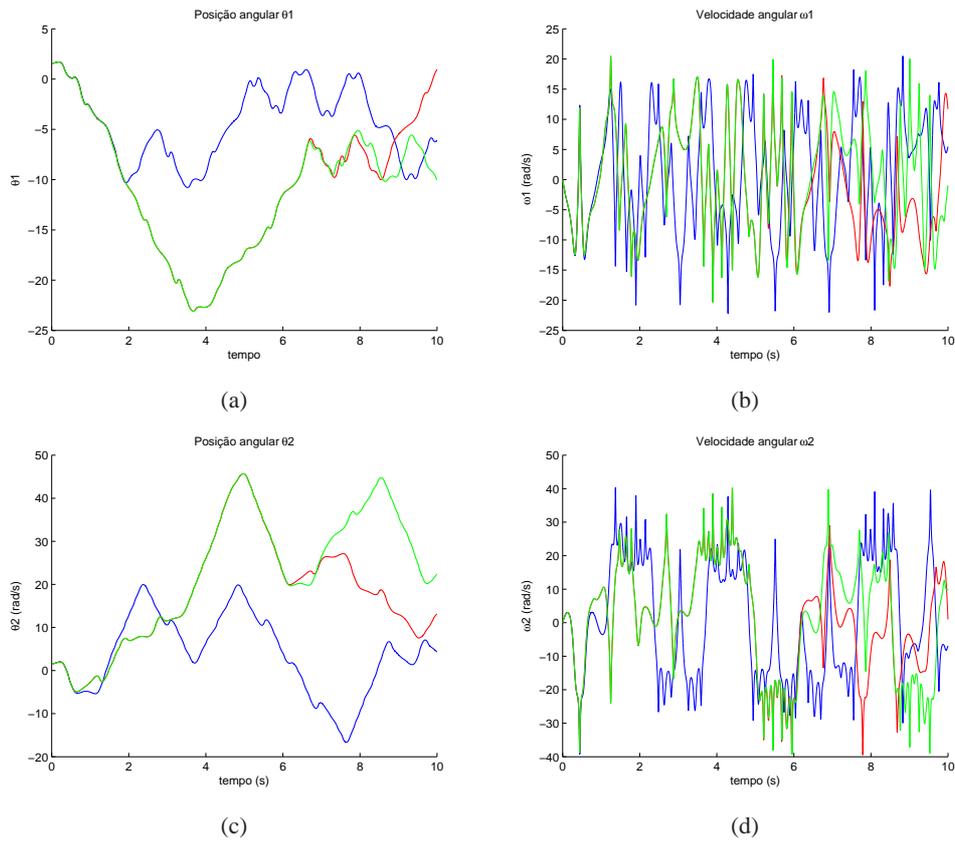


Figura 3.5: Os gráficos (a), (b), (c) e (d) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 e $\dot{\theta}_2$ obtidos durante simulação dos métodos de Euler e Runge-Kutta de quarta ordem.

das duas curvas. Contudo, ao ser adicionado o amortecimento nas duas juntas ($d_1 = 0,01$, $d_2 = 0,01$) é obtido nas curvas (3.6(a)), (3.6(b)), (3.6(c)) e (3.6(d)) um comportamento diferente.

Este resultado indica que, ao ser reduzida a dinâmica do sistema, o Método de Euler permanece mais tempo acompanhando a curva mais precisa do Método de Runge-Kutta de quarta ordem, e quando diverge desta, apresenta uma pequena diferença na fase e magnitude. As duas curvas de Runge-Kutta agora permanecem juntas até o fim da simulação, logo o menor tempo de amostragem para a curva verde não melhora sua aproximação.

A partir dos resultados obtidos, há segurança na escolha do Método de Runge-Kutta para resolver o sistema de equações diferenciais. O tempo de amostragem $T_s = 0,001s$ também é suficiente para a aplicação, já que, como o sistema será controlado, este deve ter um movimento muito mais suave que o mostrado nas Figuras (3.5(a)), (3.5(b)) (3.5(c)) e (3.5(d)).

3.4 ANÁLISE DO COMPORTAMENTO CAÓTICO DO PÊNDBULO DUPLO INVERTIDO

Já decidido qual método será utilizado para encontrar a resposta numérica do sistema de equações lineares representado por $\mathbf{f}(\mathbf{x}, T)$, é possível comprovar a característica caótica do movimento do pêndulo duplo invertido ideal.

A partir das condições da Tabela (3.1) foram obtidos para cada uma das quatro variáveis de estado dois gráficos. Os gráficos em azul representam exatamente os resultados obtidos com o Método de Runge-

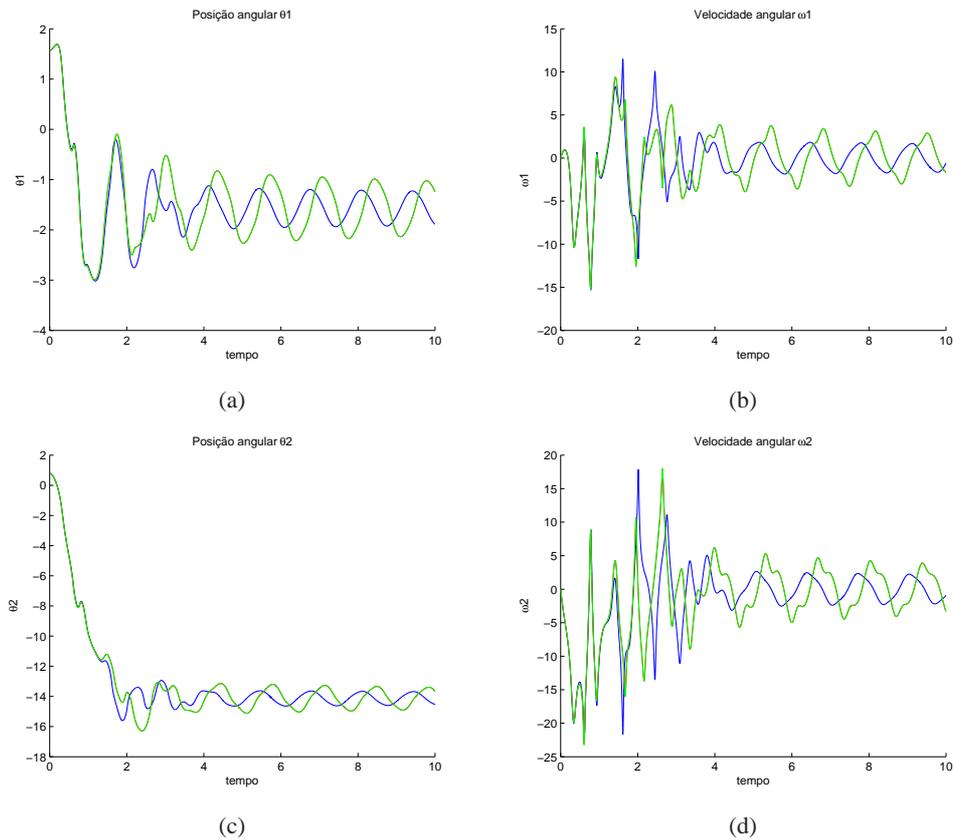


Figura 3.6: Os gráficos (a), (b), (c) e (d) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 e $\dot{\theta}_2$ obtidos durante simulação dos métodos de Euler e Runge-Kutta de quarta ordem com amortecimento nas juntas.

Kutta de quarta ordem para as condições da Tabela (3.1), já o gráfico em vermelho também utiliza o mesmo método, contudo, a única diferença das condições de simulação está no valor inicial de θ_2 que será $\pi/2 + 0,01$ radianos. Apenas esta mínima diferença já provoca um comportamento totalmente diferente no pêndulo. Isto está elucidado nas Figuras (3.7(a)), (3.7(b)), (3.7(c)) e (3.7(d)).

Se for adicionado agora o amortecimento nas duas juntas ($d_1 = 0,01$, $d_2 = 0,01$), serão obtidos os gráficos (3.8(a)), (3.8(b)), (3.8(c)) e (3.8(d)). O movimento do pêndulo fica muito mais suave e desse modo perde sua característica caótica.

3.5 LINEARIZAÇÃO DO SISTEMA

A fim de utilizar as técnicas de controle e de estimação de estados dissertadas no capítulo 2, é necessário expressar linearmente o sistema no espaço de estados. É necessário então recorrer a uma linearização local do modelo.

A linearização prejudicará o controle e a estimação, já que estes só serão feitos corretamente se as variáveis de estados estiverem em torno do ponto de operação em que foi feita a linearização.

A partir daqui T , torque aplicado pelo motor conectado ao $link_1$, será substituído pelo sinal de controle

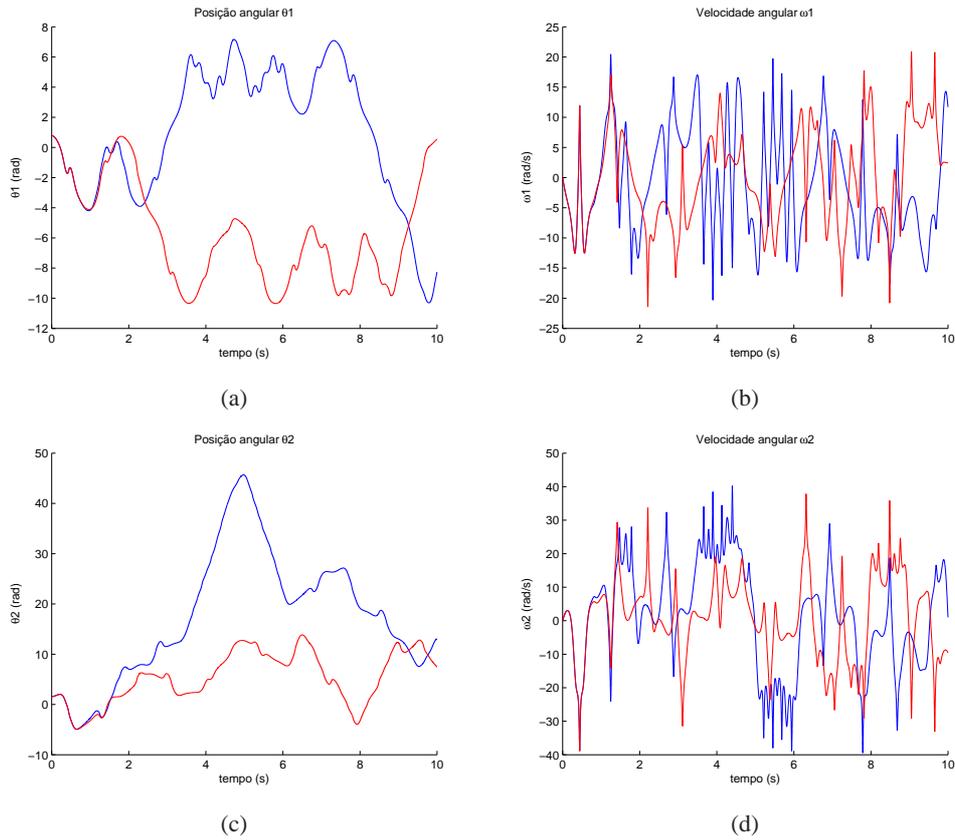


Figura 3.7: Os gráficos (a), (b), (c) e (d) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 e $\dot{\theta}_2$ obtidos durante simulação do comportamento caótico do pêndulo duplo invertido ideal.

u . Então, para um ponto de equilíbrio $\mathbf{f}(\mathbf{x}_0, u_0) = 0$, tem-se:

$$\delta \dot{\mathbf{x}} = \mathbf{F} \delta \mathbf{x} + \mathbf{G} \delta u, \quad (3.47)$$

$$\mathbf{y} = \mathbf{H}(\delta \mathbf{x} + \mathbf{x}_0), \quad (3.48)$$

$$\text{onde } \mathbf{F} = \mathbf{f}_x(\mathbf{x}_0, u_0), \quad (3.49)$$

$$\mathbf{G} = \mathbf{f}_u(\mathbf{x}_0, u_0), \quad (3.50)$$

$$\mathbf{y}_0 = \mathbf{H} \mathbf{x}_0. \quad (3.51)$$

Analisando a equação $\mathbf{f}(\mathbf{x}_0, u_0) = 0$, percebe-se que:

$$\mathbf{f}(\mathbf{x}_0, u_0) = \begin{bmatrix} \dot{\theta}_{1inicial} \\ f_2(\mathbf{x}_0, u_0) \\ \dot{\theta}_{2inicial} \\ f_4(\mathbf{x}_0, u_0) \end{bmatrix} = 0, \quad (3.52)$$

$$\text{portanto } \dot{\theta}_{1inicial} = 0, \quad \dot{\theta}_{2inicial} = 0. \quad (3.53)$$

Deste modo, não é possível obter um ponto de equilíbrio para a linearização se a velocidade angular inicial de ambas as barras do pêndulo não for zero. Assim, u_0 deve ser responsável por anular as acelerações angulares iniciais $\ddot{\theta}_{1inicial}$ e $\ddot{\theta}_{2inicial}$. Isso só será possível se $\theta_{2inicial} = \frac{\pi}{2}$, pois se o $link_1$ permanecer parado o $link_2$ não irá se mover já que não haverá torque resultante sobre ele.

Substituindo então os valores necessários em \mathbf{x}_0 e admitindo um ângulo inicial para o $link_1$, pode-se, com ambas as equações $f_2(\mathbf{x}_0, u_0) = 0$ e $f_4(\mathbf{x}_0, u_0) = 0$, chegar ao mesmo valor de u_0 .

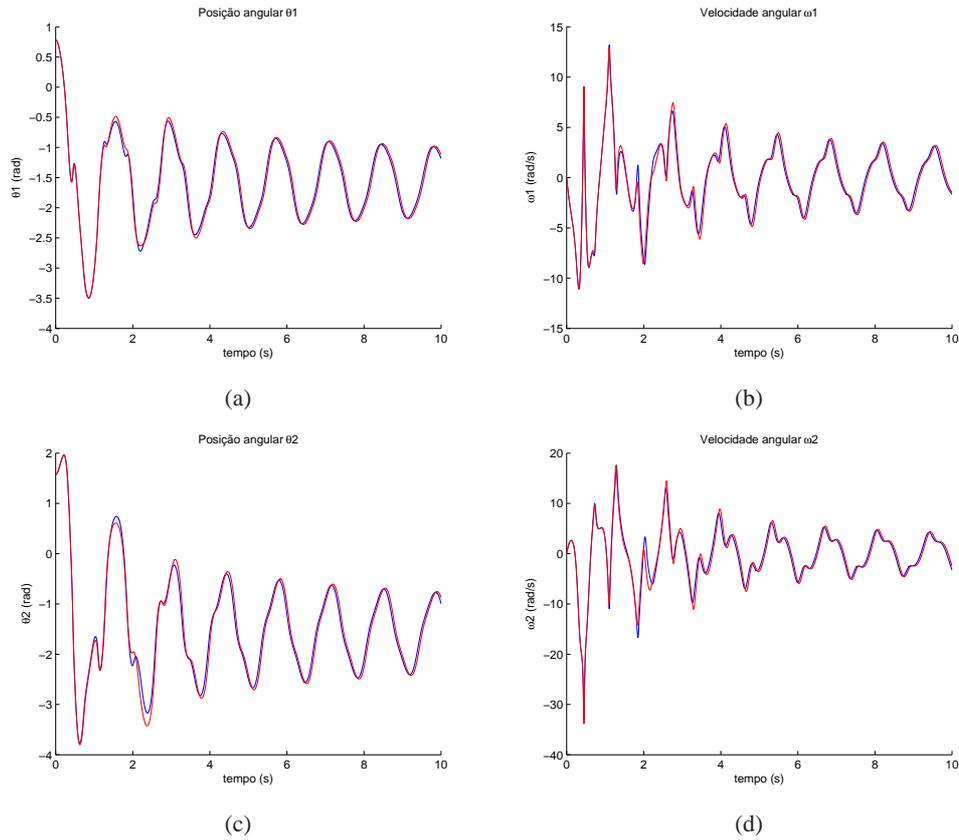


Figura 3.8: Os gráficos (a), (b), (c) e (d) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 e $\dot{\theta}_2$ obtidos durante simulação do comportamento caótico do pêndulo duplo invertido ideal com amortecimento nas juntas.

O cálculo dos gradientes $\mathbf{f}_x(\mathbf{x}_0, u_0)$ e $\mathbf{f}_u(\mathbf{x}_0, u_0)$ foi obtido com o MATLAB por meio da sua *Symbolic Toolbox* utilizando a função *jacobian.m*. Como estas equações são muito complexas e não acrescentariam muito a este trabalho, elas são unicamente encontradas no CD que acompanha o mesmo.

O importante é que agora já existe uma aproximação linear em espaço de estados do sistema não-linear. É necessário agora discretizar o sistema considerando o retentor de ordem zero.

Utilizando um tempo de amostragem $T_s = 0,001s$ e as equações (2.34) e (2.35), tem-se:

$$\delta \mathbf{x}_{k+1} = \mathbf{\Phi} \delta \mathbf{x}_k + \mathbf{\Gamma} \delta u_k, \quad (3.54)$$

$$\mathbf{y}_k = \mathbf{H}(\delta \mathbf{x}_k + \mathbf{x}_0). \quad (3.55)$$

Utilizando os coeficientes contidos na Tabela (3.1) seriam obtidas as seguintes matrizes $\mathbf{\Phi}$ e $\mathbf{\Gamma}$.

$$\mathbf{\Phi} = \begin{bmatrix} 1,0000 & 0,0010 & 0,0000 & 0,0000 \\ 0,0364 & 1,0000 & -0,0210 & 0,0000 \\ 0,0000 & 0,0000 & 1,0000 & 0,0010 \\ 0,0386 & 0,0000 & 0,0713 & 1,0000 \end{bmatrix}, \quad (3.56)$$

$$\mathbf{\Gamma} = \begin{bmatrix} 0,0000 \\ 0,0269 \\ 0,0000 \\ 0,0286 \end{bmatrix}. \quad (3.57)$$

O valor de u_0 para o equilíbrio seria de $1,3527 Nm$.

3.6 CONTROLADORES E ESTIMADORES DE ESTADO

Já possuindo o modelo em espaço de estados do sistema linearizado, primeiramente serão desenvolvidos os controladores. Será possível então analisar se o sistema pode ser controlado se tivéssemos na saída todos os estados do sistema.

Ao adicionarmos os estimadores, podemos então comparar sua eficiência com os resultados obtidos apenas com os controladores.

3.6.1 Controlador no espaço de estados

Da mesma forma que foi trabalhado na fundamentação teórica, primeiramente o sistema será controlado permanecendo a entrada referência sempre nula, em seguida adicionaremos um entrada referência não-nula.

A lei de controle será:

$$\delta u_k = -\mathbf{K} \delta \mathbf{x}_k. \quad (3.58)$$

A escolha dos pólos para o sistema contínuo é uma etapa complicada deste método de controle [5]. Os pólos do sistema contínuo devem ser escolhidos de forma que o sistema não saia do seu ponto de aplicação, ou seja, saia da zona em que a linearização é confiável, já que $\delta \mathbf{x}$ e δu devem ter valores reduzidos. Então estes pólos não devem ser nem muito rápidos, nem muito lentos. Abaixo estão os pólos $p_i, i = 1 \dots 4$ encontrados a partir de várias tentativas, bem como os respectivos pólos no domínio $z, \beta_i, i = 1 \dots 4$:

$$p_1 = -12, p_2 = -10, p_3 = -5, p_4 = -3, \quad (3.59)$$

$$\beta_1 = e^{p_1 T_s} = 0,9881, \beta_2 = e^{p_2 T_s} = 0,9900, \quad (3.60)$$

$$\beta_3 = e^{p_3 T_s} = 0,9950, \beta_4 = e^{p_4 T_s} = 0,9970. \quad (3.61)$$

O ganho de realimentação \mathbf{K} deve então resolver a equação (3.62).

$$|z\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{K}| = (z - \beta_1)(z - \beta_2)(z - \beta_3)(z - \beta_4) = 0. \quad (3.62)$$

Foi utilizada a função *place.m* do MATLAB para resolver a equação (3.62). Uma das limitações dessa função é que não pode ser encontrado \mathbf{K} quando temos pólos repetidos, fator observado na escolha dos pólos do controlador.

O simulador então utiliza-se do Método de Runge-Kutta de quarta ordem para representar o modelo dinâmico e os valores obtidos das variáveis de estado (valores medidos pelos supostos sensores) são subtraídos dos seus valores iniciais para então serem utilizados na lei de controle. O valor δu_k é então somado a u_0 para poder ser utilizado no método de Runge-Kutta.

Para adicionar a entrada referência, precisamos antes de tudo aplicá-la ao ponto de operação em que o sistema foi linearizado. Deste modo:

$$\delta \mathbf{r} = \mathbf{r} - \mathbf{x}_0, \quad (3.63)$$

$$\delta \mathbf{x}_r = \delta \mathbf{r}. \quad (3.64)$$

Alterando a lei de controle, tem-se:

$$\delta u_k = -\mathbf{K}(\delta \mathbf{x}_k - \delta \mathbf{x}_r), \quad (3.65)$$

$$\text{contudo } u_k = \delta u_k + u_0 + \delta u_r, \quad (3.66)$$

$$\text{onde } \delta u_r = u_{ss} - u_0, \quad (3.67)$$

$$\text{e } f_2(\mathbf{x}_r, u_{ss}) = 0, \quad (3.68)$$

$$\text{deste modo } u_k = \delta u_k + u_{ss}. \quad (3.69)$$

Com a adição do termo u_{ss} ao sinal δu_k , é obtida uma ligação direta com o sinal de controle a partir da referência, concluindo então a determinação do controlador.

As Figuras (3.9(a)), (3.9(b)), (3.9(c)) e (3.9(d)) mostram os resultados obtidos em cada variável de estado ao realizarmos o controle do pêndulo de modo que este saísse de uma posição inicial x_0 para uma posição final x_{final} , passando por posições intermediárias equidistantes 5° umas das outras. Nestas posições são recalculadas as matrizes Φ e Γ devido a novas linearizações do modelo.

Contudo, há desvios nos valores iniciais de θ_1 e θ_2 , chamados respectivamente de $desvio_1$ e $desvio_2$, bem como ruídos na medição das variáveis de estado. Os ruídos são modelados como ruídos brancos de média zero e variância de $1/9^\circ$. A Tabela (3.2) mostra os valores de todos os coeficientes utilizados na obtenção destes gráficos.

Tabela 3.2: Configuração para simulação do controlador de espaço estados utilizando sensores para todas as variáveis de estado.

m_1 (kg)	0,3
m_2 (kg)	0,5
l_1 (m)	0,3
l_2 (m)	0,3
d_1 (N.m.s)	0,01
d_2 (N.m.s)	0,01
Ts (s)	0,001
$\theta_{1inicial}$ (rad)	$\frac{\pi}{4}$
$\dot{\theta}_{1inicial}$ (rad/s)	0
$\theta_{2inicial}$ (rad)	$\frac{\pi}{2}$
$\dot{\theta}_{2inicial}$ (rad/s)	0
θ_{1final} (rad)	$3\frac{\pi}{4}$
$\dot{\theta}_{1final}$ (rad/s)	0
θ_{2final} (rad)	$\frac{\pi}{2}$
$\dot{\theta}_{2final}$ (rad/s)	0
$desvio_1$ ($^\circ$)	-10
$desvio_2$ ($^\circ$)	10
ruído de medição	SIM

Analisando a Figura (3.9(e)), pode-se perceber que o atuador é mais exigido na etapa de correção dos desvios iniciais $desvio_1$ e $desvio_2$. Depois de atingido o equilíbrio para a primeira referência intermediária em $\theta_{1inicial} + 5^\circ$, o sinal de controle varia de forma suave, contudo, os erros de medição são propagados para o mesmo.

3.6.2 Controle Ótimo

O sistema do pêndulo duplo invertido deve ser constantemente controlado, sendo assim, não há um número finito N de amostras no qual o sistema chega ao seu regime permanente. Deste modo, deve-se utilizar o controle LQR *Steady-State Optimal Control*, solucionando para no nosso sistema a *Equação algébrica de Riccati*, equação (2.137).

Para resolver a *Equação algébrica de Riccati* foi utilizada a função *dlqr.m* do pacote de controle do MATLAB. Esta função tem como argumentos as matrizes do sistema, Φ e Γ , e as matrizes de ponderação Q_1 e Q_2 .

A matriz de ponderação Q_2 é quadrada e de ordem 1, isso porque só há um único sinal de controle u_k .

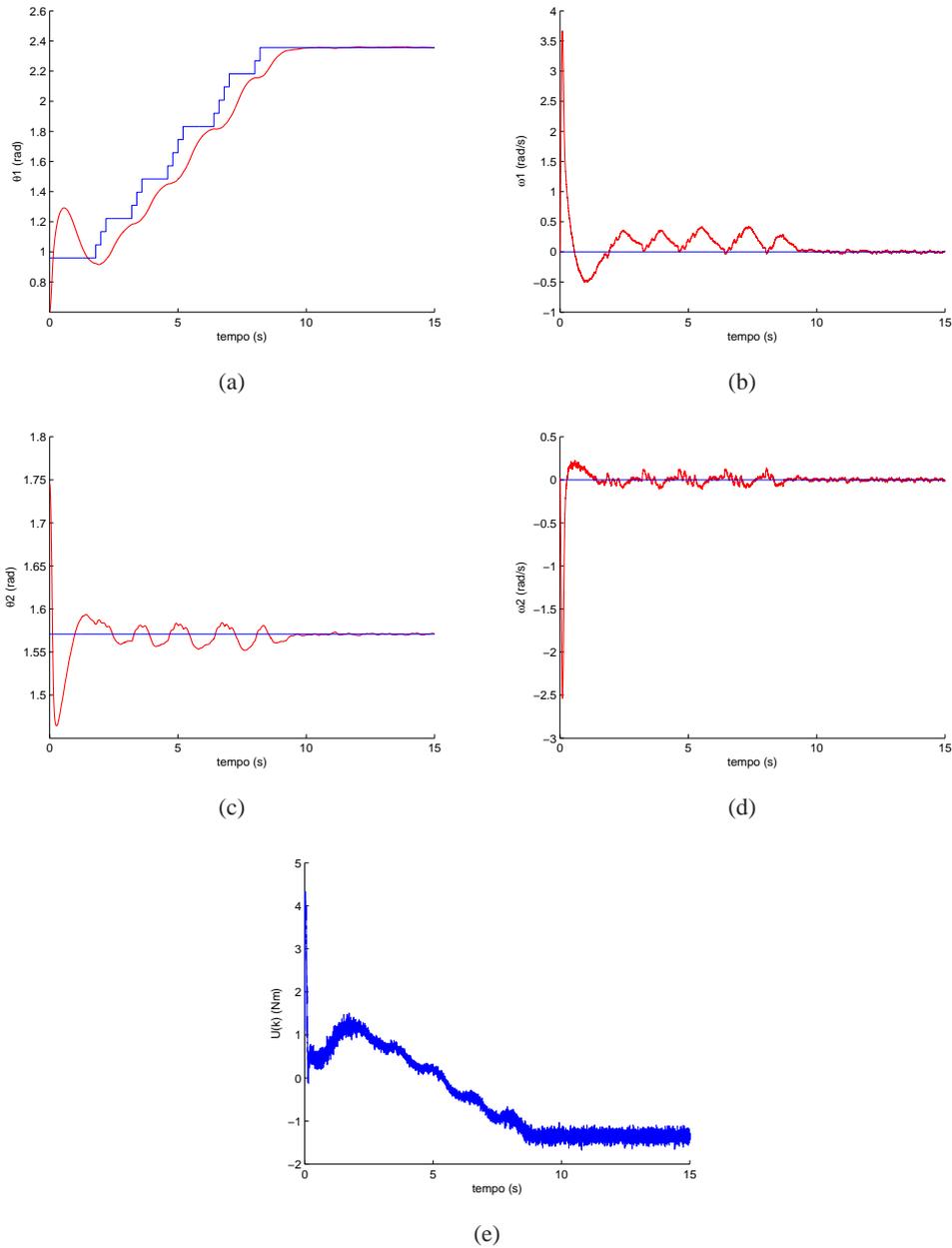


Figura 3.9: Os gráficos (a), (b), (c), (d) e (f) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 , $\dot{\theta}_2$ e u_k obtidos durante simulação do controle do pêndulo duplo invertido utilizando os coeficientes da Tabela (3.2).

Já a matriz Q_1 foi determinada seguindo a sugestão da equação (2.136). Foi obtido então:

$$Q_1 = \rho H^T H, \quad (3.70)$$

$$= \rho \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.71)$$

Quanto maior o valor de ρ , os pólos ótimos atuarão de forma a minimizar a variação das variáveis de estado. Quanto maior Q_2 , os pólos atuaram de forma a minimizar o sinal de controle. Assim, para evitar que um sinal de controle seja grande o suficiente para retirar o sistema da zona de trabalho, Q_2 deve ser maior que ρ .

Pode ser encontrado um vetor \mathbf{K}_{otimo} a partir da solução da *Equação algébrica de Riccati* para uma simulação de pêndulo duplo com as características da Tabela (3.2), e com os coeficientes $\rho = 1$ e $\mathbf{Q}_2 = 10$. A nova lei de controle deve então respeitar a equação (3.72).

$$\delta u_k = -\mathbf{K}_{otimo}(\delta \mathbf{x}_k - \delta \mathbf{x}_r). \quad (3.72)$$

Ao simular agora o sistema com a lei de controle da equação (3.72), são obtidos os gráficos das Figuras (3.10(a)), (3.10(b)), (3.10(c)), (3.10(d)) e (3.10(e)).

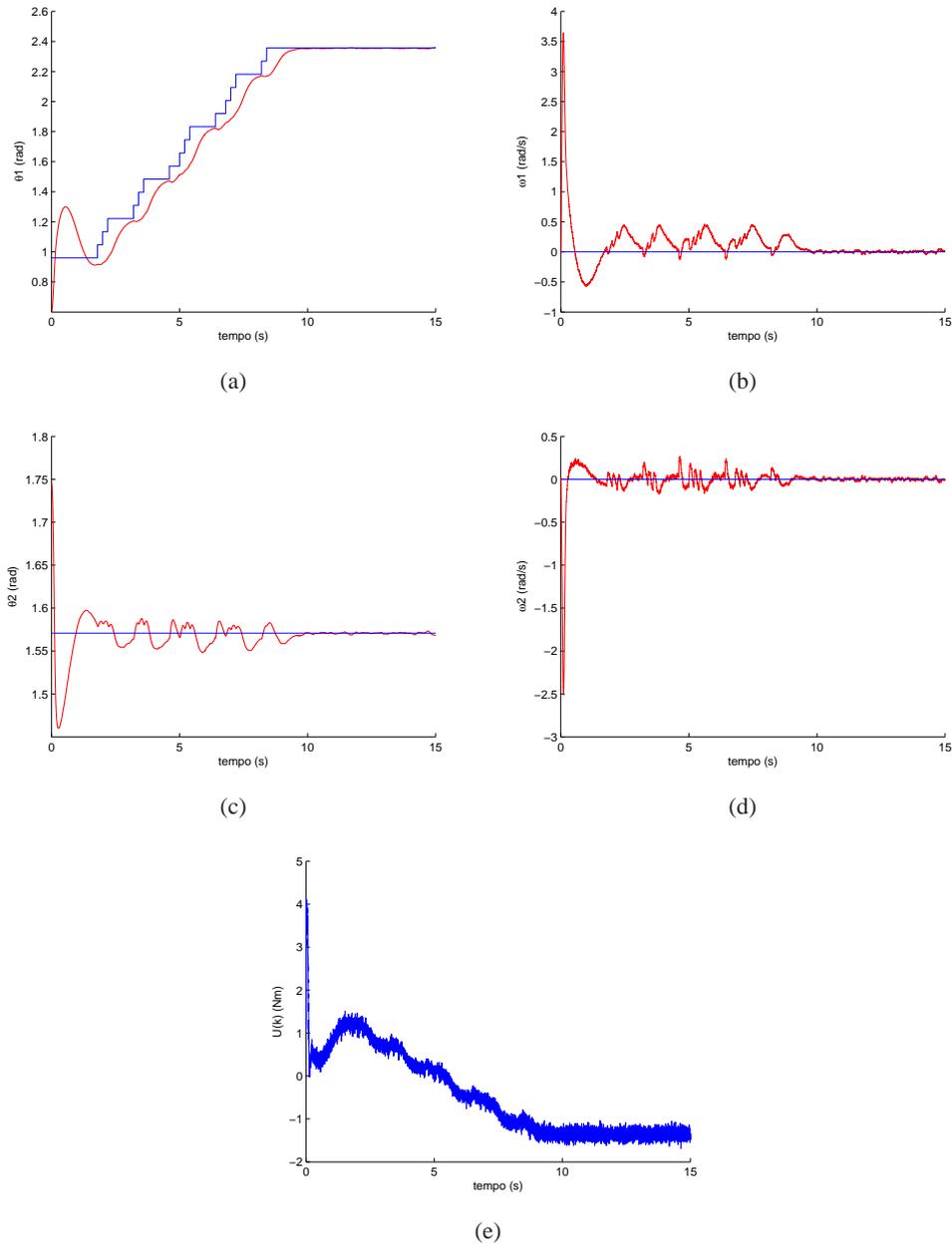


Figura 3.10: Os gráficos (a), (b), (c), (d) e (f) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 , $\dot{\theta}_2$ e u_k obtidos durante simulação do controle do pêndulo duplo invertido utilizando os coeficientes da Tabela (3.2) e controle ótimo com os coeficientes $\rho = 1$ e $\mathbf{Q}_2 = 10$.

Pode então ser feita uma comparação entre os pólos ótimos obtidos, com os pólos escolhidos anterior-

mente para realizar o controle no espaço de estados. Os pólos ótimos $p_{i_{otimo}}, i = 1 \dots 4$ são:

$$p_{1_{otimo}} = -10,8732 \quad , \quad p_{2_{otimo}} = -9,2417, \quad (3.73)$$

$$p_{3_{otimo}} = -4,6027 + 0,4641j \quad , \quad p_{4_{otimo}} = -4,6027 + 0,4641j. \quad (3.74)$$

Pode-se ver que os pólos ótimos dominantes são mais rápidos que os pólos dominantes escolhidos previamente. Isto também pode ser visto comparando os gráficos (3.10(a)) e (3.9(a)) e percebendo que o primeiro chega mais rápido à referência.

3.6.3 Estimador Corrente

Como o nosso tempo de amostragem T_s é relativamente alto com relação ao tempo de processamento computacional, o Estimador Corrente pode ser utilizado sem nos preocupações com atrasos embutidos no sistema.

Como visto na seção 2.3, não há prejuízo em utilizar os mesmos vetores \mathbf{K} e \mathbf{K}_{otimo} juntamente com os estimadores de estados. Por isso, é possível modificar a equação (2.101) de forma a utilizar o controle ótimo e a entrada referência. Tem-se então:

$$\begin{aligned} \widehat{\delta \mathbf{x}}_{k+1} &= (\Phi - \Gamma \mathbf{K}_{otimo} - \mathbf{L}_c \mathbf{H} \Phi + \mathbf{L}_c \mathbf{H} \Gamma \mathbf{K}_{otimo}) \widehat{\delta \mathbf{x}}_k \\ &+ (\Gamma \mathbf{K}_{otimo} - \mathbf{L}_c \mathbf{H} \Gamma \mathbf{K}_{otimo}) \delta \mathbf{x}_r + \mathbf{L}_c \delta \mathbf{y}_{k+1}, \end{aligned} \quad (3.75)$$

$$\delta u_k = -\mathbf{K}_{otimo} (\widehat{\delta \mathbf{x}}_k - \delta \mathbf{x}_r). \quad (3.76)$$

Mas é preciso escolher os pólos do Estimador Corrente para resolver sua equação característica. Como os pólos do estimador devem ser muito mais rápidos que os pólos do controlador, pode-se utilizar os seguintes pólos $p_{e_i}, i = 1 \dots 4$ no domínio s para então encontrar os pólos $\beta_{e_i}, i = 1 \dots 4$ no domínio z .

$$p_{e_1} = -50 \quad , p_{e_2} = -56 \quad , p_{e_3} = -57 \quad , p_{e_4} = -58, \quad (3.77)$$

$$\beta_{e_1} = e^{p_{e_1} T_s} = 0,9512 \quad , \quad \beta_{e_2} = e^{p_{e_2} T_s} = 0,9455, \quad (3.78)$$

$$\beta_{e_3} = e^{p_{e_3} T_s} = 0,9446 \quad , \quad \beta_{e_4} = e^{p_{e_4} T_s} = 0,9436. \quad (3.79)$$

A resolução da equação característica do Estimador Corrente (3.80) pode ser feita com o uso da função `place.m` do MATLAB, como foi feito para encontrar \mathbf{K} .

$$|z\mathbf{I} - \Phi + \mathbf{L}_c \mathbf{H} \Phi| = (z - \beta_{e_1})(z - \beta_{e_2})(z - \beta_{e_3})(z - \beta_{e_4}). \quad (3.80)$$

Durante o controle do sistema, serão realizadas novas linearizações do mesmo, logo a matriz \mathbf{L}_c também terá que ser recalculada para cada nova linearização.

A simulação a partir dos coeficientes da Tabela (3.2), com aplicação do controle ótimo utilizando os coeficientes $\rho = 1$ e $\mathbf{Q}_2 = 10$, e utilizando o Estimador Corrente pode ser vista com as Figuras (3.11(a)), (3.11(b)), (3.11(c)), (3.11(d)) e (3.11(e)). Em vermelho estão os valores reais das variáveis de estado, enquanto que a suas estimações estão em azul. A referência está representada na cor preta.

Pode ser visto que o sistema leva muito mais tempo para chegar na sua posição final, e mesmo assim há um erro estacionário em θ_1 de 0.1 radianos ($5,7296^\circ$). Este erro ocorre devido aos erros estacionários de estimação nas velocidades angulares de ambos $link_1$ e $link_2$.

Pode-se perceber também que para cada mudança de referência a estimação dos estados é comprometida. Ocorrem sobressaltos de todas estimações exatamente quando mudamos a referência. Neste mesmo momento são alterados o ponto de aplicação da linearização, e todas as matrizes dependentes de \mathbf{x}_0 e u_0 são recalculadas.

Mas, mesmo assim, o sistema pode ser controlado, e os estados estimados. É esperado que sejam encontrados resultados melhores utilizando o Filtro de Kalman, já que este busca uma estimação ótima para cada novo $\hat{\mathbf{x}}_k$.

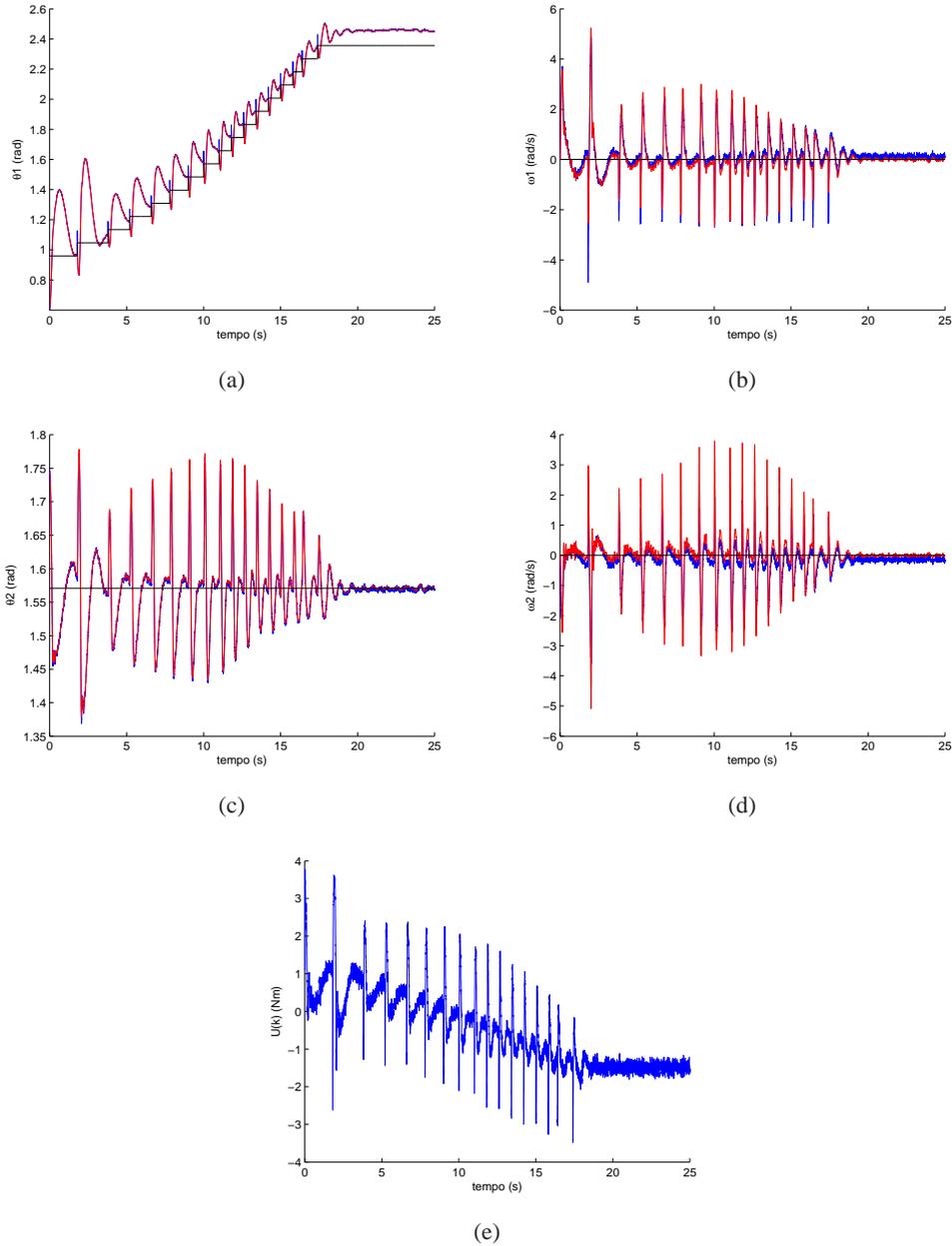


Figura 3.11: Os gráficos (a), (b), (c), (d) e (f) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 , $\dot{\theta}_2$ e u_k obtidos durante simulação do controle do pêndulo duplo invertido utilizando os coeficientes da Tabela (3.2), controle ótimo com os coeficientes $\rho = 1$ e $Q_2 = 10$ e Estimador Corrente..

3.6.4 Filtro de Kalman Estendido

Para aplicar o Filtro de Kalman Estendido, é preciso utilizar uma solução numérica para o sistema de equações não lineares que modela o pêndulo duplo invertido. Um bom método aqui já estudado é o Método de Euler. Deste modo:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + Ts\mathbf{f}(\mathbf{x}_{k-1}, u_{k-1}) + w_{k-1} = f(\mathbf{x}_{k-1}, u_{k-1}, w_{k-1}), \quad (3.81)$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + r_{k-1} = h(\mathbf{x}_{k-1}, u_{k-1}, w_{k-1}), \quad (3.82)$$

$$\text{onde } p(w) \sim N(0, \mathbf{Q}), \quad (3.83)$$

$$p(v) \sim N(0, \mathbf{R}). \quad (3.84)$$

Pode-se então encontrar a estimação *a priori* a partir da equação (3.85).

$$\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_{k-1} + Ts\mathbf{f}(\hat{\mathbf{x}}_{k-1}, u_{k-1}). \quad (3.85)$$

As matrizes $\Phi_{k[i,j]}$, $W_{k[i,j]}$, $H_{k[i,j]}$ e $V_{k[i,j]}$ podem então serem obtidas:

$$\Phi_{k[i,j]} = \frac{\partial f^{[i]}}{\partial \mathbf{x}^{[j]}}(\hat{\mathbf{x}}_{k-1}, u_{k-1}, 0) = \mathbf{I} + Ts\mathbf{f}_x(\hat{\mathbf{x}}_{k-1}, u_{k-1}), \quad (3.86)$$

$$W_{k[i,j]} = \frac{\partial f^{[i]}}{\partial w^{[j]}}(\hat{\mathbf{x}}_{k-1}, u_{k-1}, 0) = 1, \quad (3.87)$$

$$H_{k[i,j]} = \frac{\partial h^{[i]}}{\partial \mathbf{x}^{[j]}}(\hat{\mathbf{x}}_{k-1}, 0) = \mathbf{H}, \quad (3.88)$$

$$V_{k[i,j]} = \frac{\partial h^{[i]}}{\partial v^{[j]}}(\hat{\mathbf{x}}_{k-1}, 0) = 1. \quad (3.89)$$

Substituindo o valor destas matrizes nas equações (2.72), (2.83), (2.85), (2.84) e (2.86), tem-se o algoritmo do Filtro de Kalman Estendido para o pêndulo duplo invertido.

Predição

$$\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_{k-1} + Ts\mathbf{f}(\hat{\mathbf{x}}_{k-1}, u_{k-1}), \quad (3.90)$$

$$\mathbf{P}_k^- = \Phi_k \mathbf{P}_{k-1} \Phi_k^T + \mathbf{Q}. \quad (3.91)$$

Correção

$$\mathbf{M}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R})^{-1}, \quad (3.92)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{M}_k (\mathbf{y}_k - \mathbf{H} \hat{\mathbf{x}}_{k-1}^-), \quad (3.93)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{M}_k \mathbf{H}_k) \mathbf{P}_k^-. \quad (3.94)$$

Para dar início ao algoritmo, é necessário uma estimação inicial $\hat{\mathbf{x}}_0$, uma matriz de covariância do erro de estimação inicial \mathbf{P}_0 , uma matriz de covariância do erro do processo \mathbf{Q} e uma matriz de covariância do erro de medição \mathbf{R} .

Tem-se que $\hat{\mathbf{x}}_0 = \mathbf{x}_0$. Para as matrizes \mathbf{P}_0 e \mathbf{Q} pode-se admitir que ambas possuem o formato:

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}, \quad (3.95)$$

$$\mathbf{Q} = \sigma_Q^2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}, \quad (3.96)$$

$$\text{onde } \sigma_Q^2 = 1. \quad (3.97)$$

Para a matriz \mathbf{Q} , está sendo assumido que a incerteza de modelamento dos estados que realmente não são medidos por sensores, $\dot{\theta}_1$ e $\dot{\theta}_2$, são menores que a variância dos estados medidos, θ_1 e θ_2 . Como os

estados $\dot{\theta}_1$ e $\dot{\theta}_2$ apenas podem ser estimados, esta matriz Q ajudará o algoritmo a chegar mais rapidamente a uma matriz P_k mínima, já que há confiança no modelo dimensionado.

Para a matriz P_0 há uma explicação similar. Está sendo considerado que a incerteza sobre a velocidade inicial das barras é menor que a incerteza inicial da posição das mesmas. Logo está sendo admitido que o erro de estimação de x_0 é reduzido, e principalmente, o erro dos estados que não podemos medir é inferior aqueles que têm sensores.

O valor admitido para a matriz R será:

$$R = \begin{bmatrix} 0,0019 & 0 \\ 0 & 0,0019 \end{bmatrix}. \quad (3.98)$$

Anteriormente, já foi elucidado que os sensores têm um ruído de distribuição normal com média nula e variância de $1/9^\circ = 0,0019 \text{ rad}$. Logo a matriz de covariância do erro de medição R deve ter os mesmos valores. Considerando R diagonal, está sendo afirmado que os erros de cada sensor não se correlacionam, fato que não traz nenhum prejuízo para a nossa aplicação.

As Figuras (3.12(a)), (3.12(b)), (3.12(c)), (3.12(d)) e (3.12(e)) mostram os resultados obtidos na simulação do pêndulo duplo invertido descrito na Tabela (3.2), utilizando o controle ótimo com os coeficientes $\rho = 1$ e $Q_2 = 10$, e estimação por Filtro de Kalman Estendido. Nos gráficos onde foram representadas as variáveis de estado existem 3 curvas. Em vermelho está o real valor das variáveis de estado durante a simulação, em preto a estimação obtida graças ao Filtro de Kalman Estendido, e na cor azul a referência.

Como já era esperado, a estimação utilizando o Filtro de Kalman Estendido é muito superior a obtida com o Estimador Corrente. Não são observados saltos na estimação durante a mudança do ponto de operação da linearização, como visto anteriormente na utilização do Estimador Corrente. O erro de estimação estacionário é bastante reduzido, e principalmente, a partir destes resultados pode-se utilizar apenas sensores de posição angular para ambas barras, diminuindo o custo do projeto.

3.7 ANÁLISE DE CONTROLABILIDADE E OBSERVABILIDADE

Já foi visto que a configuração utilizada na Tabela (3.2) produz um sistema controlável, todos os estados podem ser alcançados pelo sinal de controle, e observável, todos os estados podem ser estimados pela uma leitura das saídas do sistema [5].

Para identificar outras configurações de pêndulo duplo invertido, estas devem ainda manter as propriedades de controlabilidade e observabilidade. Deve-se primeiramente obter as matrizes de controlabilidade CO e observabilidade OB [7].

$$CO = [\Gamma \quad \Phi\Gamma \quad \Phi^2\Gamma \quad \Phi^3\Gamma], \quad (3.99)$$

$$OB = \begin{bmatrix} H \\ H\Phi \\ H\Phi^2 \\ H\Phi^3 \end{bmatrix}. \quad (3.100)$$

Assim, se ambas tiverem posto 4, o sistema será controlável e observável. Mas não temos nenhuma informação quantitativa sobre estas propriedades.

Uma das ferramentas que pode ser utilizada para quantificar o quão perto de um posto inferior a 4 as matrizes CO e OB estão é a decomposição em valores singulares, SVD (Singular Value Decomposition) [9].

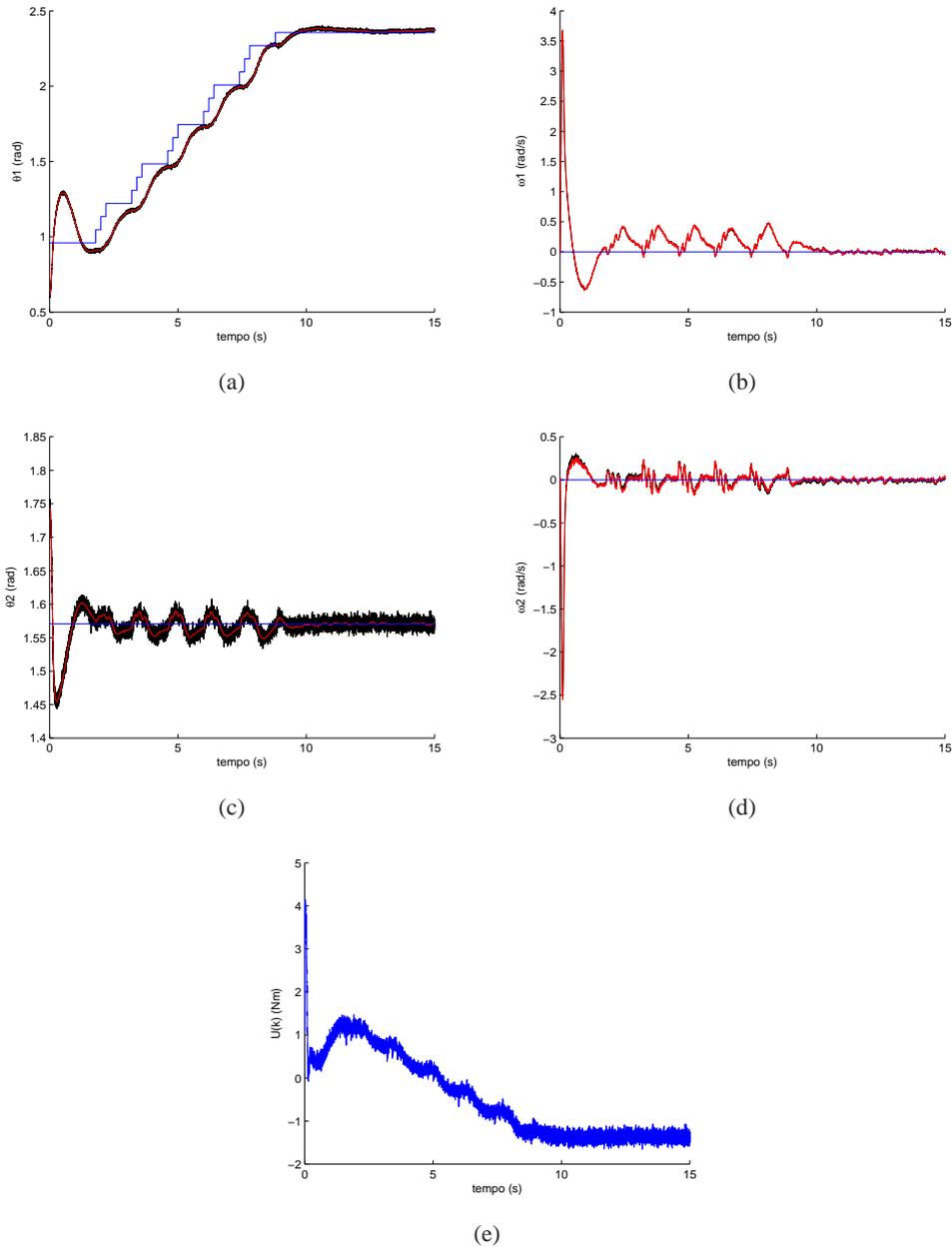


Figura 3.12: Os gráficos (a), (b), (c), (d) e (f) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 , $\dot{\theta}_2$ e u_k obtidos durante simulação do controle do pêndulo duplo invertido utilizando os coeficientes da Tabela (3.2), controle ótimo com os coeficientes $\rho = 1$ e $Q_2 = 10$ e Filtro de Kalman Estendido.

A decomposição em valores singulares mostra que tendo uma matriz $A_{[m,n]}$:

$$\mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^T, \quad (3.101)$$

$$\mathbf{W} = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix}. \quad (3.102)$$

A matriz \mathbf{U} é $m \times n$ e \mathbf{V} é quadrada de ordem n . Ambas matrizes são ortogonais, e os valores da diagonal de \mathbf{W} são os valores singulares $\sigma_i, i = 1 \dots n$ da matriz \mathbf{A} [9].

O menor valor singular de A indica a distância do posto de A para um posto não máximo. Logo se algum dos valores singulares for nulo, a matriz A não terá posto máximo [11].

Para diferentes configurações pode-se decompor as matrizes CO e OB obtidas em seus valores singulares e obter um gráfico a partir de seu valor singular mínimo. Serão analisadas as relações existentes destas duas propriedades com as posições de equilíbrio θ_1 e θ_2 , as massas m_1 e m_2 e os comprimentos l_1 e l_2 .

Nas Figuras (3.13(a)) e (3.13(b)) está representada a variação do valor singular mínimo com relação a posição de equilíbrio θ_1 e θ_2 para as matrizes de controlabilidade e observabilidade, respectivamente. Na Tabela (3.3) estão representados como variam os coeficientes da configuração do pêndulo para obter estes dados.

Tabela 3.3: Configuração do pêndulo duplo invertido para análise da controlabilidade e observabilidade a partir da variação da posição de equilíbrio.

m_1 (kg)	0,3
m_2 (kg)	0,5
l_1 (m)	0,3
l_2 (m)	0,3
d_1 (N.m.s)	0,01
d_2 (N.m.s)	0,01
Ts (s)	0,001
$\theta_{1\text{inicial}}$ (rad)	$\frac{\pi}{4} \dots 3\frac{\pi}{4}$
$\dot{\theta}_{1\text{inicial}}$ (rad/s)	0
$\theta_{2\text{inicial}}$ (rad)	$\frac{\pi}{4} \dots 3\frac{\pi}{4}$
$\dot{\theta}_{2\text{inicial}}$ (rad/s)	0

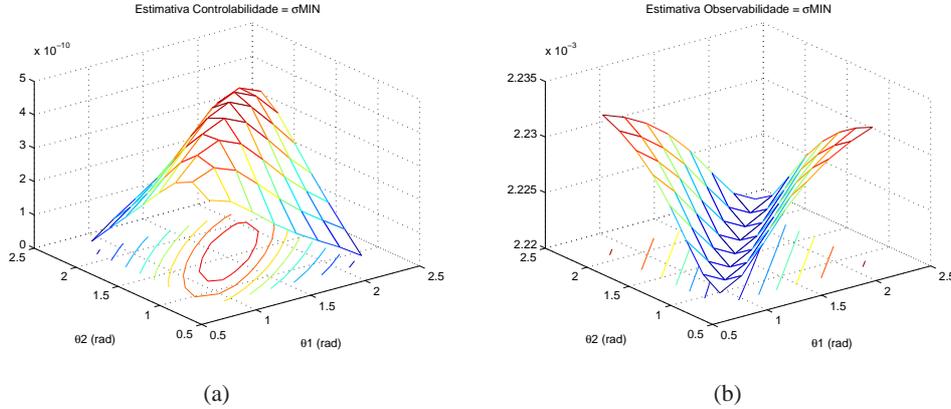


Figura 3.13: Os gráficos (a) e (b) correspondem respectivamente ao valor singular mínimo da matriz de controlabilidade CO e observabilidade OB a partir dos valores da Tabela (3.3).

Os valores obtidos em ambos gráficos são muito pequenos, mas mostram uma característica interessante para o sistema. Os pontos onde o sistema é mais controlável são os que este é menos observável, e vice-versa. Isso pode ser explicado pelo fato que quando mais alto o centro de massa das duas barras estiver, maior será a energia armazenada, maior a dinâmica do sistema, e portanto menor a observabilidade. Contudo, quando $\theta_1 = \pi/2$, uma pequena variação no sinal de controle provoca uma grande variação na base do $link_2$, potencializando o controle.

Os gráficos (3.14(a)) e (3.14(b)) foram obtidos através das configurações possíveis do pêndulo a partir da Tabela (3.4). Estes mostram que as propriedades de controlabilidade e observabilidade variam pouco com relação a variação dos comprimentos das barras.

Tabela 3.4: Configuração do pêndulo duplo invertido para análise da controlabilidade e observabilidade a partir da variação do comprimento das barras.

m_1 (kg)	0,3
m_2 (kg)	0,3
l_1 (m)	0,05 ... 0,6
l_2 (m)	0,05 ... 0,6
d_1 (N.m.s)	0,01
d_2 (N.m.s)	0,01
Ts (s)	0,001
$\theta_{1\text{inicial}}$ (rad)	$\frac{\pi}{4}$
$\dot{\theta}_{1\text{inicial}}$ (rad/s)	0
$\theta_{2\text{inicial}}$ (rad)	$\frac{\pi}{2}$
$\dot{\theta}_{2\text{inicial}}$ (rad/s)	0

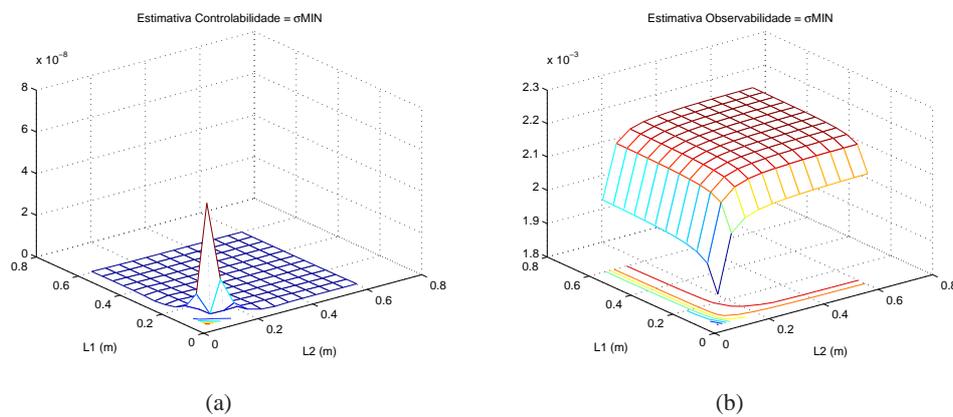


Figura 3.14: Os gráficos (a) e (b) correspondem respectivamente ao valor singular mínimo da matriz de controlabilidade CO e observabilidade OB a partir dos valores da Tabela (3.4).

Os gráficos (3.15(a)) e (3.15(b)) representam respectivamente a variação do valor singular mínimo das matrizes de controlabilidade e observabilidade para uma variação da massa de ambas as barras. Os valores utilizados para obter estes gráficos estão relacionados na Tabela (3.5).

Tabela 3.5: Configuração do pêndulo duplo invertido para análise da controlabilidade e observabilidade a partir da variação da massa das barras.

m_1 (kg)	0,1 ... 2
m_2 (kg)	0,1 ... 2
l_1 (m)	0,3
l_2 (m)	0,3
d_1 (N.m.s)	0,01
d_2 (N.m.s)	0,01
Ts (s)	0,001
$\theta_{1\text{inicial}}$ (rad)	$\frac{\pi}{4}$
$\dot{\theta}_{1\text{inicial}}$ (rad/s)	0
$\theta_{2\text{inicial}}$ (rad)	$\frac{\pi}{2}$
$\dot{\theta}_{2\text{inicial}}$ (rad/s)	0

A partir dos gráficos (3.15(a)) e (3.15(b)), pode-se observar que o sistema torna-se mais controlável

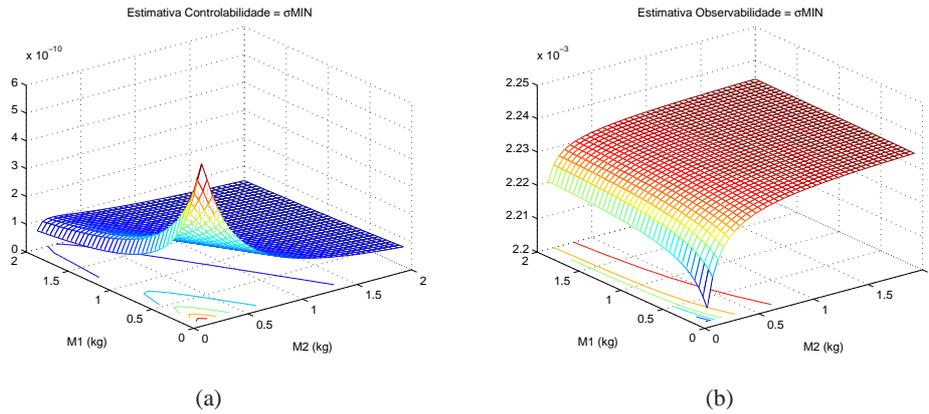


Figura 3.15: Os gráficos (a) e (b) correspondem respectivamente ao valor singular mínimo da matriz de controlabilidade CO e observabilidade OB a partir dos valores da Tabela (3.5).

quando as barras possuem massas reduzidas, enquanto que este fica mais observável quando maiores as massas. A observabilidade também é beneficiada quando a massa do $link_2$ é superior a do $link_1$.

Estas informações podem auxiliar na montagem mecânica de um pêndulo duplo invertido na configuração *arm-driven*, contudo não explicam totalmente o comportamento do mesmo durante uma simulação. Isto ocorre porque toda a análise sobre a controlabilidade e observabilidade é limitada pela linearização do sistema.

Pode ser notada esta característica ao tentar equilibrar o pêndulo com o ângulo $\theta_1 = 0$. De acordo com gráficos (3.13(a)) e (3.13(b)) este sistema poderia ser controlado, contudo a simulação mostra que este controle é impossível. Nesta posição, o sinal de controle necessita de uma grande variação para produzir um deslocamento da base do $link_2$, e quando este o faz, o sistema já saiu do seu ponto de operação.

Através de simulações utilizando a configuração da Tabela (3.6), pode ser observado que o sistema deixa de ser controlável quando θ_1 é inferior a aproximadamente 30° .

Tabela 3.6: Configuração para simulação do controlador ótimo utilizando o Filtro de Kalman Estendido para análise do limite de controle do pêndulo duplo invertido.

m_1 (kg)	0,3
m_2 (kg)	0,5
l_1 (m)	0,3
l_2 (m)	0,3
d_1 (N.m.s)	0,01
d_2 (N.m.s)	0,01
T_s (s)	0,001
$\theta_{1\text{inicial}}$ (rad)	$\frac{\pi}{2}$
$\dot{\theta}_{1\text{inicial}}$ (rad/s)	0
$\theta_{2\text{inicial}}$ (rad)	$\frac{\pi}{2}$
$\dot{\theta}_{2\text{inicial}}$ (rad/s)	0
$\theta_{1\text{final}}$ (rad)	0
$\dot{\theta}_{1\text{final}}$ (rad/s)	0
$\theta_{2\text{final}}$ (rad)	0
$\dot{\theta}_{2\text{final}}$ (rad/s)	0
$desvio_1$ ($^\circ$)	-10
$decivo_2$ ($^\circ$)	10
ruído de medição	SIM

3.8 O SIMULADOR

O simulador produzido tem como funcionalidades possibilitar ao usuário criar seu pêndulo duplo invertido e simulá-lo de diversas maneiras.

Dimensionamento físico:

- O usuário pode escolher a massa e o comprimento de cada barra do pêndulo.
- Os coeficiente de amortecimento d_1 e d_2 também podem ser modificados.
- O sinal de controle, bem como a velocidade angular do $link_1$ podem ser saturados, podendo assim simular o comportamento de diferentes atuadores para o mesmo sistema.

Controle:

- Pode ser utilizado o controle ótimo, ou o controle no espaço de estados.
- Ao escolher a utilização do controle ótimo, o usuário pode modificar os coeficientes ρ e Q_2 de forma a ponderar a função de custo (2.110).
- Ao não utilizar o controle ótimo, o sistema será controlado de modo a apresentar os pólos utilizados na equação (3.62)

Estimadores de estados:

- Podem ser utilizados o Estimador Corrente ou o Filtro de Kalman Estendido.
- O Estimador Corrente terá os pólos utilizados na equação (3.80), enquanto que o Filtro de Kalman Estendido terá as suas matrizes P_0 , Q , e R com os valores representados nas identidades (3.95), (3.96) e (3.98).
- O usuário pode ainda escolher utilizar todos os estados realimentados, ou seja, cada variável de estado seria medida por um sensor.

A simulação:

- O usuário pode escolher o ponto de partida e de chegada do pêndulo.
- Em toda simulação as velocidades angulares iniciais de cada barra serão consideradas nulas.
- O movimento do pêndulo para a simulação a que ele foi configurado sempre é mostrado na janela do simulador.
- Podem ser determinados os desvios iniciais nos ângulos θ_1 e θ_2 .
- O tempo final de simulação também é determinado pelo usuário.
- Em todas as simulações haverá ruído de medição nos sensores. Este ruído tem distribuição normal de média nula e variância de $1/9^\circ$.
- O usuário pode escolher se quer simular o sistema como um regulador, onde o ponto de equilíbrio é alterado levando o pêndulo a posição final e a entrada referência é constante e nula (reguladores consecutivos), ou se quer usar um controlador que segue uma referência (controladores consecutivos).
- Para que a simulação tenha início, é preciso apenas que seja pressionado o botão PLAY do simulador.

- Ao fim da simulação, ao ser pressionado o botão REFRESH, o pêndulo voltará para a sua posição inicial.

Gráficos:

- Se escolhida a opção de geração de gráficos, serão apresentados os gráficos relativos a cada variável de estado e também ao sinal de controle.
- Para os gráficos das variáveis de estado teremos em vermelho o valor real da variável durante a simulação, em azul o valor estimado por meio do Estimador Corrente, em preto o valor estimado pelo Filtro de Kalman Estendido, e em *cian* a referência.
- Se gerados os gráficos, o botão REFRESH quando apertado apagará os mesmos.
- Se ao fim de uma simulação na qual foram gerados gráficos for mais uma vez pressionado o botão PLAY, este iniciará uma nova simulação e apagará os gráficos anteriores.

A Figura (3.16) mostra a interface construída para o simulador.

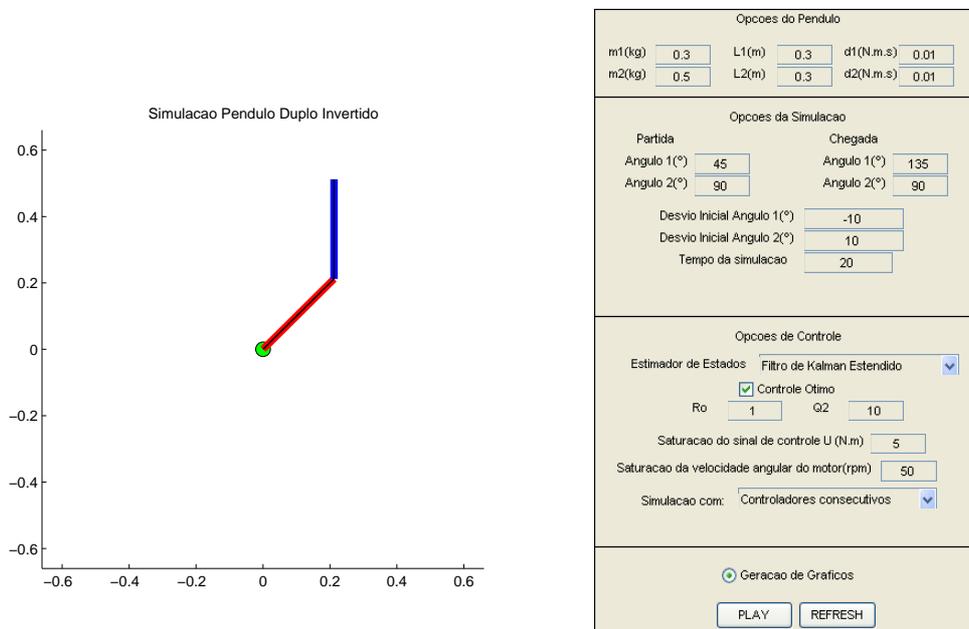


Figura 3.16: Interface para o simulador.

Um exemplo dos resultados encontrados utilizando o simulador está mostrado nas Figuras (3.18(a)), (3.18(b)), (3.18(c)), (3.18(d)) e (3.18(e)). Neste caso, os valores dos coeficientes que caracterizam a simulação e as características físicas do pêndulo estão relacionados na interface de simulação da Figura (3.17).

Pode-se ver que a simulação feita a partir da interface da Figura (3.17) é possível de ser implementada fisicamente. A saturação no motor apenas prejudicou a fase de correção dos desvios iniciais em ambas barras.

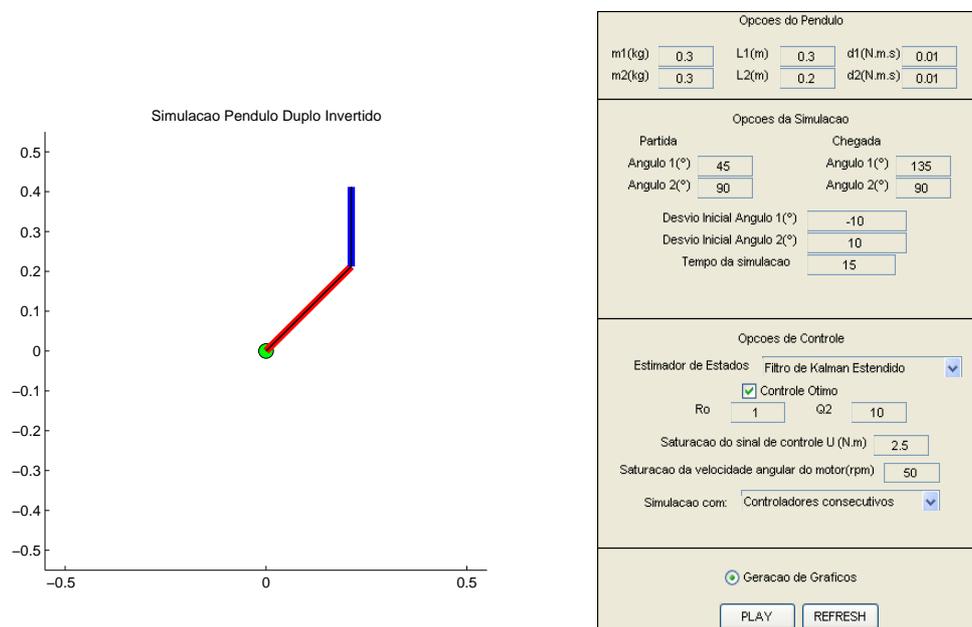
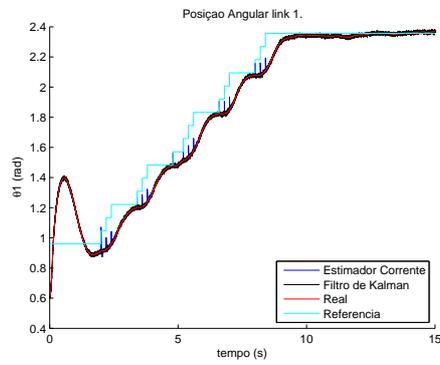
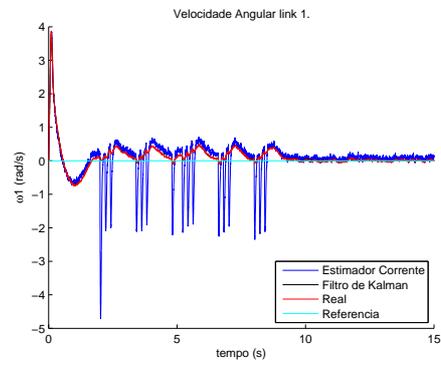


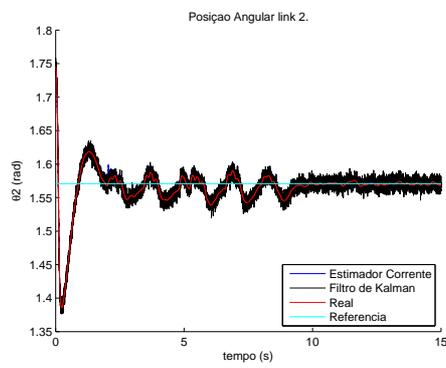
Figura 3.17: Interface para o simulador utilizada para estudo de possível configuração mecânica.



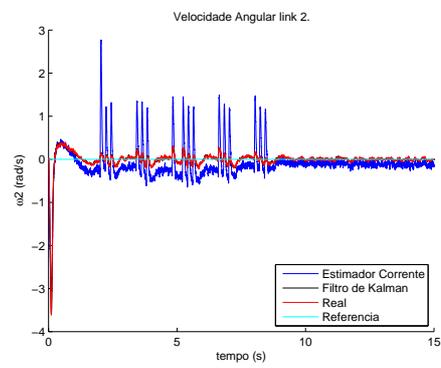
(a)



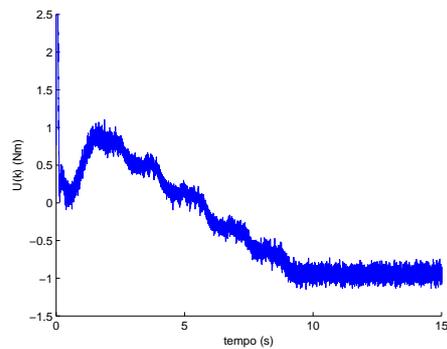
(b)



(c)



(d)



(e)

Figura 3.18: Os gráficos (a), (b), (c), (d) e (f) correspondem respectivamente aos gráficos de θ_1 , $\dot{\theta}_1$, θ_2 , $\dot{\theta}_2$ e u_k para a simulação cuja interface está representada na Figura (3.17).

4 CONCLUSÕES

Neste trabalho, foi desenvolvido um estudo sobre a implementação de uma futura bancada experimental para o estudo de diversas técnicas de controle. O sistema escolhido então foi o pêndulo duplo invertido na sua configuração *arm-driven*.

O pêndulo duplo invertido possui duas saídas a serem controladas, contudo é sub-atuado e ainda apresenta um comportamento completamente não linear. Logo não é possível aplicar as mesmas técnicas de controle clássicas. Deste modo, é um experimento muito interessante para que técnicas de controle como o controle no espaço de estados, controle ótimo e outras ainda mais modernas sejam implementadas.

Por utilizar apenas dois sensores de posição angular, devem ser implementados estimadores de estados. Estes podem ser obtidos através do espaço dos estados, ou até ser encontrados por meio de métodos estocásticos como o Filtro de Kalman. Estes estimadores serão responsáveis por obter estimativas sobre as variáveis de estado que não estão diretamente nas saídas do sistema.

Após feito seu modelamento matemático e obtido o sistema de equações diferenciais que caracteriza o pêndulo duplo invertido, foram sugeridos dois métodos numéricos para resolvê-lo: o Método de Euler e o Método de Runge-Kutta de quarta ordem. O sistema apresenta uma dinâmica muito elevada quando não consideramos o amortecimento provocado nas juntas do mesmo, e neste caso limite o Método de Euler não obtém a precisão desejada, logo foi utilizado o método de Runge-Kutta de quarta ordem. Com este método foram obtidos dados muito coerentes, o que permitiu utilizar um tempo de amostragem relativamente alto, 0,001s.

Com o modelo em mãos foi sugerida uma configuração física para o pêndulo duplo invertido (comprimento e massa das barras, e coeficientes de amortecimento), e foi possível o estudo das técnicas de controle e de estimação já mencionadas.

Como era esperado, aquelas em que são encontradas soluções ótimas para o problema de controle (controle ótimo) e para o problema de estimação de estados (Filtro de Kalman) prevaleceram aos métodos em que o projetista precisa intuitivamente alocar os pólos do controlador ou do estimador. Contudo, a linearização do sistema para que fosse possível a utilização destes métodos cria zonas muito reduzidas ao redor do ponto de aplicação da linearização em que o modelo linear se aproxima do não-linear. Deste modo, apesar de uma análise de controlabilidade e observabilidade indicar que o pêndulo duplo invertido é controlável para todas as posições angulares, foi visto nas simulações que isto não é possível.

Foi desenvolvido um simulador com uma interface amigável para que os usuários possam simular as diversas combinações possíveis a partir das técnicas estudadas, obtendo gráficos e modificando as características físicas do pêndulo da maneira desejada.

Finalmente, este trabalho busca fornecer um apanhado de técnicas de controle e estimação de estado que não são comumente vistas nos cursos de Controle e Automação e podem ser úteis na aplicação do pêndulo duplo invertido, como em outros sistemas nos quais as técnicas clássicas não são suficientes. Principalmente, este trabalho visou fornecer um simulador confiável para auxiliar no planejamento de futuras implementações físicas do pêndulo duplo invertido.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] HORÁČEK, P. Laboratory experiments for control theory courses: A survey. *Annual Reviews in Control*, v. 24, p. 151–162, 2000.
- [2] ZHOU, Z.; WHITEMAN, C. Motions of a double pendulum. *Nonlinear Analysis, Theory, Methods & Applications*, v. 26, p. 1177–1191, 1996.
- [3] AWTAR, S. et al. Inverted pendulum systems: rotary and arm-driven - a mechatronic system design case study. *Mechatronics*, v. 12, p. 357–370, 2002.
- [4] STACHOWIAK, T.; OKADA, T. A numerical analysis of chaos in the double pendulum. *Chaos, Solutions and Fractals*, v. 29, p. 417–422, 2006.
- [5] FRANKLIN, G. F.; POWELL, J. D.; WORKMAN, M. *Digital Control of Dynamic Systems*. [S.l.]: Addison Wesley Longman, Inc., 1998.
- [6] WELCH, G.; BISHOP, G. *An Introduction to the Kalman Filter*. Agosto 2001. SIGGRAPH 2001.
- [7] NISE, N. S. *Engenharia de Sistemas de Controle*. 3. ed. [S.l.]: LTC, 2002.
- [8] MERIAN, L. G.; KRAIGE, L. G. *Mecânica, Dinâmica*. 4. ed. [S.l.]: LTC, 1999.
- [9] PRESS, W. H. et al. *Numerical Recipes in C*. [S.l.]: Cambridge University Press, 1992.
- [10] RUGGIERO, M. A. G.; LOPES, V. L. da R. *Cálculo Numérico*. [S.l.]: McGraw-Hill Ltda, 1988.
- [11] ABIDO, M. A. Analysis and assessment of statcom-based damping stabilizers for power system stability enhancement. *Electric Power Systems Research*, v. 73, p. 177–185, 2005.

I. DESCRIÇÃO DO CONTEÚDO DO CD

No CD que encontra-se anexado a este relatório estão os arquivos desenvolvidas no MAPLE e MATLAB que auxiliaram na produção do modelo dinâmico, na solução dos problemas numéricos, confecção dos gráficos, e desenvolvimento do simulador

MAPLE

- ModelamentoPendulum.mws: arquivo desenvolvido para resolução do sistema obtido com as equações (3.18) e (3.19).

MATLAB

- Simulacao_Euler_RK.m: arquivo desenvolvido para obtenção dos gráficos referentes a análise dos métodos de Euler e Runge-Kutta de quarta ordem.
- Simulacao_CAOS.m: arquivo desenvolvido para obtenção dos gráficos referentes a análise do movimento caótico do pêndulo duplo invertido.
- Contr_Obser.m: arquivo desenvolvido para obtenção dos gráficos referentes a análise de controlabilidade e observabilidade do pêndulo duplo invertido.
- SIMULADOR
 - Gera_Ganhos_Controlador.m: arquivo desenvolvido para gerar os ganhos K e K_{otimo} .
 - Gera_Matrizes.m: arquivo desenvolvido para gerar as matrizes que identificam o espaço de estados para cada linearização local.
 - Gradiente_Sist.m: arquivo que contem o gradiente do sistema modelado.
 - PD_DiLinear.m: arquivo que contem as equações referentes ao Estimador Corrente.
 - PD_Dinamica.m: arquivo que contem as equações diferenciais que caracterizam o pêndulo duplo invertido.
 - PENDULUM.m: arquivo que simula o controle e gera os gráficos para o simulador utilizando reguladores consecutivos.
 - PENDULUM2.m: arquivo que simula o controle e gera os gráficos para o simulador utilizando controladores consecutivos.
 - Uinicial.m: arquivo desenvolvido para encontrar o valor de u_0 para cada linearização local.
 - Pendulo_GUI.m: arquivo que gera a interface gráfica do simulador.