

Peter Eliasson  
Lars Forssell  
Johan Hamberg  
Maria Sjöblom  
Sven-Lennart Wirkander

## Konstruktionsmetoder för robotstyrautomater



## Metodrapport

Peter Eliasson  
Lars Forssell  
Johan Hamberg  
Maria Sjöblom  
Sven-Lennart Wirkander

# Konstruktionsmetoder för robotstyrautomater

Utgivare Totalförsvarets forskningsinstitut Avdelningen för Systemteknik 172 90 STOCKHOLM	Rapportnummer, ISRN FOI-R--0326--SE	Klassificering Metodrapport
	Forskningsområde Bekämpning	
	Månad, år December 2001	Projektnummer E6004
	Verksamhetsgren Uppdragsfinansierad verksamhet	
	Delområde VVS med styrda vapen	
Författare/redaktör Peter Eliasson Lars Forssell Johan Hamberg Maria Sjöblom Sven-Lennart Wirkander	Projektledare Johan Hamberg	
	Godkänd av Monica Dahlén	
	Tekniskt och/eller vetenskapligt ansvarig Martin Hagström	
Rapportens titel Konstruktionsmetoder för robotstyrautomater		
Sammanfattning <p>I denna rapport görs en jämförelse av olika metoder för konstruktion av styrautomater för robotar samt för modellering av robotar. Kapitel 1 är en översikt av och jämförelse mellan äldre och modernare konstruktions- och modelleringsmetoder. Kapitel 2 redogör för den teoretiska bakgrunden till robusthetsanalys och genomför en sådan robusthetsanalys av en enkel robotmodell med och utan regulator. Resultaten av <math>\mu</math>-analys jämförs med en explicit teoretisk analys av stabiliteten och överensstämmelsen är god. I kapitel 3 redovisas en ekvationsbaserad modell av en SoFRam-robot och i kapitel 4 skisseras en traditionell parameterstyrd styrautomat till denna. Kapitlen 5 och 6 beskriver hur konstruktion av olinjära styrautomater i vissa fall delvis kan automatiseras. Kapitel 5 beskriver hur en ekvationsbaserad modell implementerad i det symboliska språket Maple kan transformeras till en form användbar i simuleringsmiljön Simulink, medan kapitel 6 visar hur regulatorer till system erhållna på detta sätt kan erhållas med hjälp av Matlab toolboxen Nonlinear Synthesis Toolbox. Dessa kapitel redovisar nuläget inom en pågående verksamhet.</p>		
Nyckelord		
Övriga bibliografiska uppgifter	Språk Svenska	
ISSN 1650-1942	Antal sidor 49	
Distribution Enligt missiv	Pris Enligt prislista Sekretess Öppen	

Issuing organization Swedish Defence Research Agency System Technology Division SE-172 90 STOCKHOLM Sweden	Report number, ISRN	Report type
	FOI-R--0326--SE	Methodology report
	Research area code	
	Combat	
	Month year	Project no.
	December 2001	E6004
	Customers code	
	Contracted Research	
	Sub area code	
	Weapons and Protection	
Author/s (editor/s) Peter Eliasson Lars Forssell Johan Hamberg Maria Sjöblom Sven-Lennart Wirkander	Project manager	
	Johan Hamberg	
	Approved by	
	Monica Dahlén	
	Scientifically and technically responsible	
	Martin Hagström	
Report title Synthesis methods for missile autopilots		
Abstract <p>In this report different missile control design methods and modelling methodologies are compared. In chapter 1 traditional and modern design and modelling methodologies are reviewed and compared. In chapter 2 the theory of robustness analysis is covered and such an analysis is applied to a simple missile model. The results of an explicit robust stability analysis and a <math>\mu</math> analysis are compared and they are seen to agree. In chapter 3 an equation based SoFRam missile model is given and chapter 4 provides a sketch of a traditional gain scheduling controller for this model. Chapters 5 and 6 describe how the design of nonlinear controllers can be partly automatized. Chapter 5 describes how an equation based model implemented in the symbolic language Maple can be transformed to a form that can be used by the simulation environment Simulink. Chapter 6 shows how controllers to models obtained this way can be produced by the Nonlinear Synthesis Toolbox. These two chapters describe ongoing work</p>		
Keywords		
Further bibliographic information	Language	
	Swedish	
ISSN	Pages	
1650-1942	49	
Distribution	Price Acc. to pricelist	
By sendlist	Security classification Unclassified	



## Innehåll

<b>1</b>	<b>Metoder för modellering och styrning</b>	<b>1</b>
1.1	Inledning . . . . .	1
1.2	Metoder för modellering . . . . .	1
1.3	Icke objektsorienterade modeller . . . . .	2
1.4	Objektsorienterad proceduriell modellering . . . . .	2
1.5	Objektsorienterad ekvationsbaserad modellering . . . . .	2
1.6	Ekvationsbaserad modellering i denna rapport och i framtiden . . . . .	2
1.7	Linjära och pseudolinjära metoder för styrning . . . . .	3
1.8	Parameterstyrning . . . . .	3
1.9	Adaptiv reglering . . . . .	3
1.10	Robust reglering . . . . .	4
1.11	Robust parameterstyrning . . . . .	4
1.12	Olinjära metoder för styrning . . . . .	4
1.13	Exakt linjärisering . . . . .	5
1.14	Backstepping . . . . .	5
1.15	Passivitet . . . . .	5
1.16	Metoder för robotstyrning . . . . .	6
<b>2</b>	<b>Robust stabilitetsanalys</b>	<b>7</b>
2.1	Omformulering av system med parametriserbara osäkerheter till LFT-form . . . . .	7
2.2	Begreppet $\mu$ och dess samband med minsta destabiliserande störning . . . . .	8
2.3	Användning av Nyquistkriteriet . . . . .	9
2.4	Beskrivning av enkel missil . . . . .	11
2.4.1	Omskrivning av modellen till standardform . . . . .	12
2.4.2	Transformerering till LFT:er . . . . .	12
2.5	Stabilitetsanalys av ostyrd enkel missil . . . . .	14
2.5.1	Teoretisk analys . . . . .	14
2.5.2	$\mu$ -analys . . . . .	15
2.6	Konstruktion av styrautomat för enkel missil . . . . .	17
2.7	Stabilitetsanalys av styrd enkel missil . . . . .	18
<b>3</b>	<b>Modellering av en Ramrobot</b>	<b>21</b>
3.1	Översikt . . . . .	21
3.1.1	Boostfas . . . . .	21
3.1.2	Coastfas . . . . .	21
3.1.3	Sustainerfas . . . . .	21
3.2	Missilmodell . . . . .	21
3.2.1	Omgivning . . . . .	21
3.2.2	Aerodynamik . . . . .	22
3.2.3	Motor1 (Krutdriven booster) . . . . .	24
3.2.4	Motor2 (SOFRammotor) . . . . .	25
3.2.5	Body . . . . .	26
3.2.6	Kinematik . . . . .	28
3.2.7	Sensorer . . . . .	29
3.2.8	Servo . . . . .	29

3.2.9	Inmätning . . . . .	30
3.3	Förenklingar . . . . .	30
3.3.1	Omgivning . . . . .	30
3.3.2	Aerodynamik . . . . .	30
3.3.3	Motor1 (Krutmotor) . . . . .	31
3.3.4	Motor2 (SOFRammmotor) . . . . .	31
3.3.5	Body . . . . .	31
3.3.6	Kinematik . . . . .	31
3.3.7	Sensorer . . . . .	32
3.3.8	Servo . . . . .	32
3.4	Indata till modellen . . . . .	32
3.4.1	Omgivning . . . . .	32
3.4.2	Aerodynamik . . . . .	32
3.4.3	Motor1 (Krutmotor) . . . . .	33
3.4.4	Motor2 (SOFRammmotor) . . . . .	35
3.4.5	Body . . . . .	36
3.4.6	Dynamik . . . . .	36
3.4.7	Kinematik . . . . .	37
3.4.8	Sensorer . . . . .	37
3.4.9	Servo . . . . .	37
<b>4</b>	<b>Loop shaping</b>	<b>39</b>
4.1	Linjäriserad rolldynamik . . . . .	40
4.2	Linjäriserad tippdynamik . . . . .	41
<b>5</b>	<b>Beräkningsprocess av robotmodell</b>	<b>43</b>
5.1	Implementation av robotmodell . . . . .	43
5.2	Ekvationsgenerering av C-kod . . . . .	43
5.3	Generering av S-funktion . . . . .	44
5.4	Numerisk modellanalys och resultat . . . . .	44
5.5	MATLAB . . . . .	44
5.6	Simulink . . . . .	44
5.7	Nonlinear Synthesis Tools . . . . .	44
<b>6</b>	<b>Utvärdering av Nonlinear Synthesis Tools</b>	<b>45</b>
6.1	Programvaran . . . . .	45
6.2	Testfall . . . . .	46
6.3	Resultat . . . . .	48



# 1. Metoder för modellering och styrning

## 1.1 Inledning

Allt högre prestanda- och säkerhetskrav ställs på moderna farkost- och robotsystem. Inte minst för styrningens del innebär detta ett radikalt nytänkande. Styrprinciper och konstruktionsmetoder som tidigare betraktats som "akademiska" prövas nu av industrin för såväl flygplan som robotar. Detta gäller exempelvis för "robusta linjära och pseudolinjära metoder" som  $\mathcal{H}^\infty$ ,  $\mu$ -analys/syntes och robust parameterstyrning och för genuint olinjära metoder som exakt linjärisering, backstepping, passivitetsbaserade och andra strukturutnyttjande metoder (jfr [5]).

Framtida robotforskning kommer att behöva utnyttja allt större delar av den *olinjära systemteorin* och allt *noggrannare beskrivning/modellering* av fysiken. Vidare erfordras mer systematik och högnivåhantering vid modellering och simulering.

Ett huvudsyfte med projektet Optimering av Robotprestanda är att bygga upp erforderlig kompetens för FOIs och försvarsmaktens kommande behov av modellering, simulering och värdering av robotsystem. Verksamheten är till sin natur huvudsakligen teoretisk/matematisk.

I rapporten beskrivs delar av denna utveckling.

## 1.2 Metoder för modellering

De ekvationer som beskriver en robots rörelse kan anses bestå av fyra grupper:

- generella kinematiska samband
- generella balansekvationer för massa, rörelsemängd och rörelsemängdsmoment
- specifika uttryck för tillförseltermerna i balansekvationerna, nämligen relativa mass-strömmen (motormodell) samt uttryck för yttre krafter och moment på roboten (aerodynamiska krafter och moment vid olika roderkonfigurationer, tyngdkraft)
- inre dynamik i roboten (dynamik för roderservon, målsökare, gyron m.m.)

Ett traditionellt och tämligen primitivt sätt att modellera farkostdynamiken är att uttrycka samtliga dessa samband i ett proceduriellt datorprogram, d.v.s. att blanda ekvationssambanden från modelleringen med specifika algoritmer för beräkningar och simuleringar. Resultatet blir därvid ofta svåröverskådligt och svårt att underhålla och programmeraren måste själv tänka ut i vilken ordning olika storheter skall beräknas.

För att delvis komma tillrätta med sådana problem har sedan en tid olika former av objektsorienterad modellering/programmering tillämpats; olika delar av systemet har kapslats in i objekt med väldefinierade gränssnitt mot övriga delar av modellen/programmet. En svårighet med detta förfarande är emellertid att de fysikaliskt naturliga objekten sällan eller aldrig överensstämmer med de proceduriellt naturliga objekten.

Ett fysikaliskt naturligt objekt, såsom en delkropp eller ett mekaniskt eller elektriskt kopplingselement ger *samband* mellan sina gränssnittsvariabler, snarare än en *procedur* för att beräkna vissa gränssnittsvariabler ur andra (jmf [17]). Fysikaliskt

naturlig objektsorientering leder snarare till en ekvationsbaserad än till en tilldelningsbaserad modellering (se [3]).

### 1.3 Icke objektsorienterade modeller

Åtskilliga äldre modeller för robotar finns på denna form inom FOI, FM och annorstädes. På sikt bör dessa avskaffas till förmån för modernare modellarkitekturer. Detta innebär inte att själva formatet helt och hållet avskaffas, men modeller i detta format bör i framtiden komma ut som biprodukter av en rationellare modelleringsmetod.

### 1.4 Objektsorienterad proceduriell modellering

Som exempel på objektsorienterad proceduriell modellering kan nämnas modeller utförda i Simulink.

Simulink är blockschemabaserat och har ett blockschemabaserat grafiskt användar-gränssnitt. Ett block kan definieras med hjälp av det grafiska användar-gränssnittet eller genom programmering av en s.k. S-funktion. Endast undantagsvis motsvarar de olika blocken fysiska enheter i det system man modellerar. Att byta ut någon enhet, t.ex. motorn i en robot, innebär i allmänhet väsentliga förändringar i blockschemat.

Styrautomater, åtminstone traditionella sådana, har dock normalt en naturlig blockschemastruktur, och för dessa är blockschemabaserad modellering både naturlig och enkel. Simulink- och Matlabformaten är viktiga, kanske framför allt genom att det finns så mycket reglerhjälpmedel utvecklade för detta format.

### 1.5 Objektsorienterad ekvationsbaserad modellering

Det modernaste och på många sätt bästa sättet att modellera komplexa system är att respektera systemets egna inneboende hierarkiska struktur. Detta kräver att modelleringen är ekvationsbaserad. Man kan därvid låta datorn utföra mer av rutinarbetet, nämligen att förenkla uttryck, eliminera variabler o.s.v, vilket bl.a. eliminerar en mänsklig felkälla i detta skede. Ekvationsbaserad modellering har möjliggjorts bl.a. av tillkomsten av datoralgebrasystem (Mathematica, Maple m.fl.) och rutiner för numerisk lösning av differentialalgebraiska ekvationer.

### 1.6 Ekvationsbaserad modellering i denna rapport och i framtiden

I denna rapport beskrivs en robot av Meteorotyp med ekvationsbaserad modellering i kapitel 3. I kapitel 6 påvisas en möjlig användning av ekvationsbaserad modellering, nämligen automatisk generering av proceduriella modeller.

I kapitel 5 beskrivs hur denna ekvationsbaserade robotmodell (halv-)automatiskt transformerats till en S-funktion, d.v.s. en funktion som kan anropas av Simulink. Ekvationerna har skrivits in i datoralgebraprogrammet Maple och där transformerats till ett dynamiskt system på standardform,  $\dot{x} = f(x, u)$ , vars högerled  $f(x, u)$  sedan exporterats (via ett mellansteg som C-program) i form av en S-funktion.

Denna S-funktion kan sedan exempelvis användas för automatisk generering av regulatorer med Matlab-toolboxen Nonlinear Synthesis Toolbox. I rapporten beskrivs denna toolbox och en olinjär regulator som konstruerats för ett enklare modellsystem (Rösslers system), vilket också erhållits från Maple via ovan beskrivna metod. Meningen är förstas att hela detta förfarande skall kunna tillämpas på robotmodellen själv i en snar framtid.

Under 2002 kommer även andra, kommersiella, metoder för ekvationsbaserad modellering att utprovas vid sidan av den ovan beskrivna "hemgjorda" kodkonverteringen.

## 1.7 Linjära och pseudolinjära metoder för styrning

Med pseudolinjära metoder avses metoder som återför reglering av olinjära system på teorin för reglering av linjära system. (S.k. exakt linjärisering [7], som ju återför olinjära styrproblem på linjära system via en genuint olinjär transformation och därvid använder teori för olinjära system betraktas därför inte som en pseudolinjär metod.)

Linjära och pseudolinjära metoder behandlas i denna rapportens kapitel 2 och 4.

En beprövad metod för konstruktion av styrautomater för robotar är klassisk parameterstyrning.

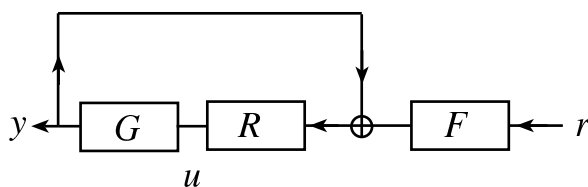
## 1.8 Parameterstyrning

Parameterstyrning, eller gain scheduling, innebär att man konstruerar en parameterberoende linjär regulator till ett parameterberoende linjärt dynamiskt system.

Betrakta som ett enkelt exempel följande överföringsfunktion

$$G(s) = \frac{1+s}{1+\alpha s}$$

där  $\alpha > 0$  är en konstant parameter. Återkoppling med den  $\alpha$ -beroende regulatorn  $R(s) = \frac{1+\alpha s}{3+3s}$  och det konstanta förfiltret  $F(s) = 2$  i följande regulatorstruktur ger en parameterberoende följning,  $y = r$ .



Figur 1.1: Det slutna systemet

I praktiken brukar man inte som i detta exempel helt eliminera parameterberoendet, utan bara minska det.

På detta sätt erhålles en familj av regulatorer som fungerar bra för var sitt *konstanta* värde på parametern, i detta fall  $\alpha$ . Man kan nu fråga sig om denna parameterberoende regulator fungerar även om värdet av  $\alpha$  inte är konstant, utan ändras sig med tiden.

Klassisk parameterstyrning bygger på att så är fallet i många praktiska problem. Regulatorn konstrueras som för konstanta  $\alpha$  och det slutna systemet simuleras för ett tillräckligt antal situationer med varierande  $\alpha$  för att man skall kunna lita på konstruktionen. Denna metod är tidsödande och matematiskt osträngt, men har använts med stor framgång under lång tid.

I kapitel 4 diskuteras denna klassiska metod tillsammans med s.k. loop shaping för styrautomatkonstruktion för en robot av Meteortyp.

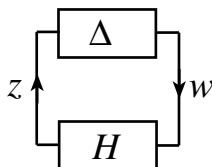
## 1.9 Adaptiv reglering

Om parametern är konstant men okänd, är det en naturlig strategi att först försöka skatta parametern med hjälp av in- och utsignalerna och sedan använda skattningen som parametervärde för regulatorn. Detta förfarande kan itereras, för att successivt förbättra skattningen och därmed regleringen.

Principen [1] för *adaptiv reglering* är att uppdateringen av skattningen och regleringen sker kontinuerligt och parallellt. Estimatet,  $\hat{\alpha}$ , blir därvid en inre tillståndsvariabel i den adaptiva regulatorn. Stabilitets- och prestandafrågor avgörs normalt med något slags Lyapunovöverläggning. Adaptiva metoder har ej undersökts i föreliggande studie.

### 1.10 Robust reglering

Genom att framställa parameterberoendet med hjälp av en s.k. bruten linjär transformation (linear fractional transformation, LFT), se kapitel 2, kan stabilitetsfrågor (och faktiskt även prestandafrågor) återföras på stabiliteten hos ett slutet system som i 1.1 jmf [20]



Figur 1.2: System med osäkerheter

Här representerar  $H$  ett dynamiskt system och  $\Delta$ , ”osäkerheterna”, ett annat. Vi särskiljer några viktiga fall:

- $H$  och  $\Delta$  är båda linjära och tidsinvarianta
- $\Delta$  är tidsvariabelt statistiskt, d.v.s  $w(t) = \Delta(t)z(t)$ .

I det förra fallet kan stabiliteten avgöras med  $\mu$ -analys. Detta avhandlas i kapitel 2.

I det senare fallet kan lågförstärkningssatsen återopas. Om  $\|\cdot\|_{ind L^2}$  betecknar den  $L^2$ -inducerade operatornormen, så gäller att villkoret

$$\|H\|_{ind L^2} \|\Delta\|_{ind L^2} < 1$$

medför att systemet är stabilt.

Vidare gäller att  $\|\Delta\|_{ind L^2} = \sup_t (\bar{\sigma}(\Delta(t)))$  och att  $\|H\|_{ind L^2} = \|H\|_{\mathcal{H}^\infty}$ . Denna observation tillåter oss att använda  $\mathcal{H}^\infty$ -metoder för olinjära system för vilka olineariteterna kan beskrivas via  $\Delta$ -blocket ovan.

### 1.11 Robust parameterstyrning

Robust parameterstyrning är resultatet av att använda metoderna för robust reglering ( $\mu$ -analys,  $\mathcal{H}^\infty$ -normer o.s.v.) för analys av det slutna system som erhålles vid parameterstyrning. Även syntesmetoder finns som bygger på robust parameterstyrning, se [6].

### 1.12 Olinjära metoder för styrning

Det finns ett flertal metoder för att reglera olinjära system.

Vissa metoder är *ad hoc*-metoder i den meningen att det inte finns några teoretiska resultat som garanterar stabilitet eller prestanda. Ett exempel på en sådan metod är konstruktion med tillståndsberoende Riccati-ekvation (jmf [15]). Den tillståndsberoende Riccati-ekvationen kan anses vara en approximativ form av Hamilton-Jacobi-Bellman-ekvationen för ett optimalstyrningsproblem med kvadratisk integralkriterium.

Andra metoder är strukturspecifika: om systemet har vissa stuktoregenskaper (t.ex. triangulariserbarhet eller passivitet) kan metoden användas och har vissa garanterade egenskaper. Exempel på strukturspecifika metoder är exakt linjärisering [7], backstepping [18], passivitetsbaserad konstruktion [16] och Lagrangesk styrning [4].

### 1.13 Exakt linjärisering

Exakt linjärisering innebär att systemet undergår en återkopplingstransformation så att det slutna systemet blir ett linjärt system. Teorin är utvecklad för olinjära system som är affina i insignalvektorn  $u$

$$\dot{x} = f(x) + g(x)u$$

Teorins huvudresultat ger nödvändiga och tillräckliga villkor på vektorfälten  $f$  och  $g$  för att en sådan transformation skall kunna göras, samt explicita formler för den erforderliga återkopplingstransformationen. Villkoren är av differentialgeometrisk natur och är avgörbara med explicita beräkningar (se [7]). Villkoren är också nödvändiga och tillräckliga för att systemet skall kunna bringas på triangulär form med ett olinjärt koordinatbyte i tillståndsrummet [8].

Iden bakom exakt linjärisering är mycket enkel och framgår av följande urartade exempel. Antag att det bara finns en styrsignal  $u$  och tillståndsvariabel  $x$ , d.v.s. att  $x$  ovan är skalärt. Villkoren för exakt linjäriserbarhet kollapsar då till villkoret att  $g(0) \neq 0$  och den linjäriserande regulatorn ges av  $u = \frac{v-f(x)}{g(x)}$  och resulterar i det nya linjära systemet  $\dot{x} = v$ .

Redan detta urartade exempel visar en svaghet hos exakt linjärisering. Om syftet är att stabilisera origo och  $f(x)$  är en stabiliserande olinjär term, så innebär cancellationsrermen  $-\frac{f(x)}{g(x)}$  en destabiliserande term. Detta leder till orobusthet, jfr [18].

### 1.14 Backstepping

Backstepping ([18], [5]) är en metod som är tillämpbar på samma klass av system som metoden med exakt linjärisering. Med en koordinatstransformation i tillståndsrummet kan systemet fås på formen

$$\begin{aligned}\dot{x}_1 &= f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 &= f_2(x_1, x_2) + g_2(x_1, x_2)x_3 \\ &\dots \\ \dot{x}_{n-1} &= f_{n-1}(x_1, \dots, x_{n-1}) + g_{n-1}(x_1, \dots, x_{n-1})x_n \\ \dot{x}_n &= f_n(x_1, \dots, x_n) + g_n(x_1, \dots, x_n)u\end{aligned}$$

Metoden går nu ut på att rekursivt konstruera regulatorer för trunkerade delsystem. I den första ekvationen,  $\dot{x}_1 = f_1(x_1) + g_1(x_1)x_2$ , kan tillståndsvariabeln  $x_2$  temporärt betraktas som en ("virtuell") styrsignal. En styrlag  $x_2 = h_1(x_1)$  konstrueras så att den valda funktionen  $V_1(x_1)$  blir en Lyapunovfunktion till detta (tänkta) delsystem. I nästa steg konstrueras en styrlag så att  $x_3 = h_1(x_1, x_2)$  så att funktionen  $V_2(x_1, x_2) = V_1(x_1) + v_2(x_2 - h_1(x_1))$  blir en Lyapunovfunktion, o.s.v. I det sista steget blir den virtuella styrvariabeln lika med den faktiska styrvariabeln och vi erhåller en stabiliserande regulator tillsammans med en Lyapunovfunktion.

Det är vanligt att välja  $V_1(x_1) = \frac{1}{2}x_1^2$  och  $v_2 = v_2 = \dots = v_n = V_1$ .

Metoden bygger som synes på ett flertal val, och särskilt bra robusthetsegenskaper garanteras om man väljer samtliga virtuella styrlagar enligt någon optimalstyrningsprincip. Med inversoptimal konstruktion ([18], [5]) kan man kringgå problemet att lösa optimeringsproblem. Optimeringskriteriet blir istället framräknat från den valda styrlagen.

### 1.15 Passivitet

Ett system

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}$$

(med  $u$  och  $y$  båda  $m$ -dimensionella) är passivt om det finns en Lyapunovfunktion  $S(x)$ , här kallad lagringsfunktion, storage function, sådan att  $u^T y \geq \dot{S}$  identiskt. Passiva system har många användbara egenskaper. Parallellkoppling av två passiva system ger passiva system. (I båda fallen har det sammansatta systemet en lagringsfunktion som är summan av delsystemens lagringsfunktioner). Ett observerbart passivt system stabiliseras uppenbarligen asymptotiskt av styrlagen  $u = -y$ .

Passivitet förekommer ofta i fysikaliska system, där lagringsfunktionen kan motsvara t.ex. energi, entalpi eller fri energi, beroende på situationen. Passivitetsöverläggningar ligger också bakom mer specifika strukturutnyttjande styrmetoder som exempelvis Lagrangesk styrning [4]. Det skall också nämnas att passivitet spelar en avgörande roll för backstepping med inversoptimalitet [18].

### 1.16 Metoder för robotstyrning

Samtliga ovannämnda styrmetoder har potentiella tillämpningar på såväl flygplan som robotar. Adaptiva metoder kan vara ett alternativ till robust parameterstyrning i vissa fall. Exakt linjärisering och dess mer robusta motsvarighet, backstepping, har förmodligen störst tillämpningspotential för mindre delsystem av farkosten. De strukturella villkoren för metoden innebär inskränkningar som växer med tillståndsrummets dimension. Pseudolinjära metoder blir ofta konservativa, ett problem som snabbt växer med osäkerheternas storlek. Det är troligt att de lämpar sig bäst som en övergripande robust överrock utanpå lokala olinjära regulatorer.

## 2. Robust stabilitetsanalys

### 2.1 Omformulering av system med parametriserbara osäkerheter till LFT-form

Betrakta det linjära dynamiska systemet

$$\dot{x} = A(p)x + B(p)u, \quad (2.1)$$

$$y = C(p)x + D(p)u, \quad (2.2)$$

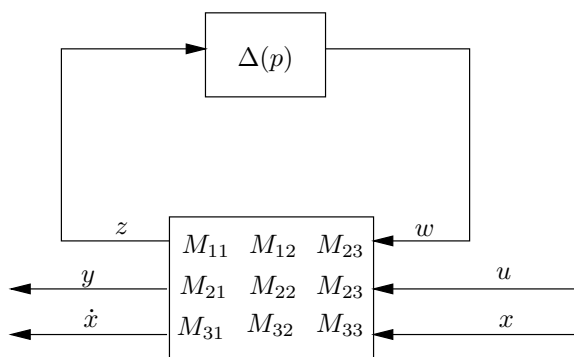
där

$x \in R^n$  är tillståndsvektorn,  $u \in R^m$  är insignalvektorn,  $y \in R^r$  är utsignalvektorn och  $p \in R^q$  är en vektor av okända parametrar.

$A(p)$ ,  $B(p)$ ,  $C(p)$  och  $D(p)$  är matriser av passande dimensioner, vars element alltså får bero av de okända parametrarna.

Om matriselementen är rationella funktioner av  $p$ 's element, är det möjligt att separera  $p$  från det övriga systemet enligt följande blockschema:

(se [14])



Figur 2.1: LFT-representation av linjärt dynamiskt system med osäkerhetsparametrar. Externa tillstånd.

Här är  $\Delta(p)$  en diagonalmatris innehållande elementen  $p_1, p_2, \dots, p_q$  i  $p$ -vektorn, vart och ett eventuellt upprepat ett antal gånger så att  $\Delta(p) \in R^{s \times s}$ , där  $s \geq q$ .

Detta sätt att representera system (2.1),(2.2) kallas Linear Fractional Transformation (LFT).

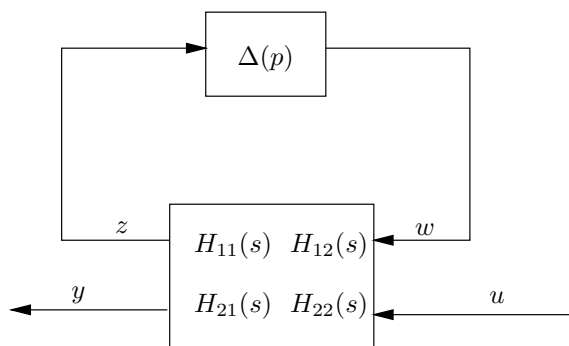
Matriserna  $M_{ij}$ ,  $i, j = 1, 2, 3$ , har lämpliga dimensioner, så att t.ex.

$$\dot{x} = M_{31}w + M_{32}u + M_{33}x. \quad (2.3)$$

Om sambandet

$$\dot{x} = sx \quad (2.4)$$

utnyttjas, kan  $x$  och  $\dot{x}$  elimineras i utbyte mot att  $s$  införes i (de nya) blockmatriserna  $H_{ij}(s)$ ,  $i, j = 1, 2$  och vi får följande LFT:



Figur 2.2: *LFT-representation av linjärt dynamiskt system med osäkerhetsparametrar. Interna tillstånd.*

Efter utnyttjande av sambandet

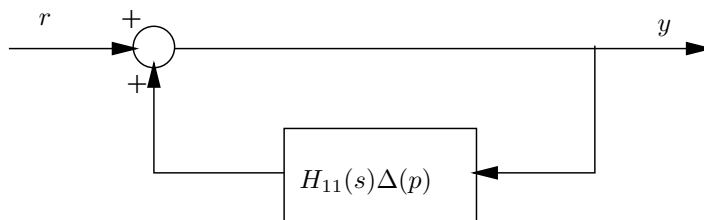
$$w = \Delta(p) z \quad (2.5)$$

kan man enkelt visa att

$$y = \left[ H_{21} \Delta (I - H_{11} \Delta)^{-1} H_{12} + H_{22} \right] u, \quad (2.6)$$

där argumenten  $s$  och  $p$  för enkelhets skull utelämnats.

Faktorn  $(I - H_{11} \Delta)^{-1}$  kan betraktas som en överföringsfunktion för följande system:



Figur 2.3: *Blockdiagram för överföringsfunktionen  $(I - H_{11} \Delta)^{-1}$*

och stabilitet hos detta system är därför ekvivalent med stabilitet hos systemet (2.1) (2.2) I det följande härleds en metod för analys av just detta.

## 2.2 Begreppet $\mu$ och dess samband med minsta destabiliserande störning

Låt  $\mathcal{D}$  vara en tillåten mängd av "störningar" som består av  $m$ -dimensionella kvadratiska matriser, alltså  $\mathcal{D} \subseteq C^{m \times m}$ .

Låt en matris  $M \in C^{m \times m}$  vara given. Definiera en mängd  $S(M)$ , bestående av sigma-normer för sådana störningar  $\Delta \in \mathcal{D}$  som gör matrisen  $I - M \cdot \Delta$  singular, alltså

$$S(M) \hat{=} \{ \bar{\sigma}(\Delta) : \det(I - M \cdot \Delta) = 0, \Delta \in \mathcal{D} \}. \quad (2.7)$$

Först visar vi att normen,  $\bar{\sigma}(\Delta)$ , för en störning  $\Delta$  måste vara  $\geq \frac{1}{\bar{\sigma}(M)}$  för att kunna tillhöra  $S(M)$



Antag därför, att  $\bar{\sigma}(\Delta) < \frac{1}{\bar{\sigma}(M)}$  och  $\det(I - M \cdot \Delta) = 0$ . Då finns ett  $x \in R^n$  så att  $x \neq 0$  och  $(I - M \cdot \Delta)x = 0$ , dvs  $(M \cdot \Delta)x = 1 \cdot x$ . Detta betyder att  $M \cdot \Delta$  har ett egenvärde  $= 1$ , varur följer att spektralradien,  $\rho(M \cdot \Delta)$ , för  $M \cdot \Delta$  är  $\geq 1$ . Men spektralradien för en matris är högst lika med normen för matrisen, oberoende av vilken norm som används ( $|\lambda| \|x\| = \|\lambda x\| = \|Ax\| \leq \|A\| \|x\| \implies |\lambda| \leq \|A\|$ , där  $\lambda$  är ett egenvärde till  $A$  svarande mot egenvektorn  $x$ ). Med utnyttjande av normegenskapen hos det maximala singulara värdet skulle vi då få  $\rho(M \cdot \Delta) \leq \bar{\sigma}(M \cdot \Delta) \leq \bar{\sigma}(M) \bar{\sigma}(\Delta) < 1$  enligt begynnelseantagandet, vilket strider mot vårt just erhållna resultat att  $M \cdot \Delta$  har ett egenvärde  $= 1$ .

$M$ -matrisens  $\mu$ -värde,  $\mu_{\mathcal{D}}(M)$ , definieras nu i termer av den "minsta" störning som gör  $I - M \cdot \Delta$  singular, enligt följande:

$$\frac{1}{\mu_{\mathcal{D}}(M)} \hat{=} \min S(M) \quad (2.8)$$

Enligt ovan är storleken (dvs normen) av denna minsta destabiliserande störning  $\geq \frac{1}{\bar{\sigma}(M)}$ . Alltså gäller  $\frac{1}{\mu_{\mathcal{D}}(M)} \geq \frac{1}{\bar{\sigma}(M)}$ , eller  $\mu_{\mathcal{D}}(M) \leq \bar{\sigma}(M)$ .

I det fall då alla störningar i  $C^{m \times m}$  är tillåtna, dvs  $\mathcal{D} = C^{m \times m}$ , kan vi visa att likhet gäller. Om vi nämligen antar att matrisen  $M$  singularvärdessuppdelas enligt  $M = U \Sigma V^H$ , med  $\Sigma$  diagonal och  $U U^H = V V^H = I$ , och konstruerar störningen  $\Delta_0 = \frac{1}{\bar{\sigma}(M)} V U^H \in C^{m \times m} = \mathcal{D}$ , så gäller dels att  $\bar{\sigma}(\Delta_0) \in S(M)$ , eftersom  $\det(I - M \cdot \Delta_0) = \det\left(I - U \Sigma V^H \frac{1}{\bar{\sigma}(M)} V U^H\right) =$

$\det U \cdot \det\left(I - \frac{1}{\bar{\sigma}(M)} \Sigma\right) \cdot \det U^H = 0$ , eftersom  $\bar{\sigma}(M)$  är ett av diagonalmatrisen  $\Sigma$ :s element, dels att  $\bar{\sigma}(\Delta_0) = \bar{\sigma}\left(\frac{1}{\bar{\sigma}(M)} V U^H\right) = \frac{1}{\bar{\sigma}(M)} \bar{\sigma}(V \cdot I \cdot U^H) = \frac{1}{\bar{\sigma}(M)} \cdot 1 = \frac{1}{\bar{\sigma}(M)}$ , varur följer att  $\min S(M) \leq \frac{1}{\bar{\sigma}(M)}$ . Olikhet gäller således åt båda hållen, varur likhet följer.

I det allmänna fallet då inte nödvändigtvis alla störningar i  $C^{m \times m}$  är tillåtna, utan vi bara vet att  $\mathcal{D} \subseteq C^{m \times m}$ , blir förstas inversen av  $\mu$ -värdet större eller lika stor, eftersom minimeringen då sker över en delmängd av  $C^{m \times m}$ . Detta kan skrivas  $1/\mu_{C^{m \times m}}(M) \leq 1/\mu_{\mathcal{D}}(M)$  eller

$$\mu_{\mathcal{D}}(M) \leq \mu_{C^{m \times m}}(M) \quad (2.9)$$

### 2.3 Användning av Nyquistkriteriet

Antag att det nominella systemet, dvs systemet i figur 2.2 med  $\Delta(p) = 0$ , är stabilt.  $H_{11}(s)$  är då en stabil överföringsfunktion, vilket i sin tur innebär att den totala krets förstärkningen  $H_{11}(s) \cdot \Delta(p)$  i det återkopplade systemet enligt figur 2.3 också är en stabil överföringsfunktion, eftersom  $\Delta(p)$  ju är en konstant matris. Vår uppgift är nu, som tidigare påpekats, att analysera hur stabiliteten hos detta återkopplade system påverkas av  $\Delta(p)$ .

Låt därför

$$\begin{bmatrix} A_{ol} & B_{ol} \\ C_{ol} & D_{ol} \end{bmatrix} \quad (2.10)$$

vara en realisation av det öppna systemet  $H_{11}(s) \cdot \Delta$ , dvs

$$H_{11}(s) \cdot \Delta = C_{ol}(sI - A_{ol})^{-1} B_{ol} + D_{ol} \quad (2.11)$$

(observera att matriserna  $A_{ol}, B_{ol}, C_{ol}$  och  $D_{ol}$ , liksom  $\Delta$ , är beroende av parame-tervektorn  $p$ ).

Då ligger rötterna till detta öppna systems karakteristiska polynom

$$\Phi_{ol}(s) = \det(sI - A_{ol}) \quad (2.12)$$

i det komplexa vänstra halvplanet (VHP).

Tillståndsekvationerna för det återkopplade systemet i figur 2.3 blir nu

$$\dot{x} = A_{ol}x + B_{ol}y, \quad (2.13)$$

$$y - r = C_{ol}x + D_{ol}y. \quad (2.14)$$

Från ekv. (2.14) får man då

$$y = (I - D_{ol})^{-1} (C_{ol}x + r), \quad (2.15)$$

vilket ger följande standardform:

$$\dot{x} = \left[ A_{ol} + B_{ol} (I - D_{ol})^{-1} C_{ol} \right] x + B_{ol} (I - D_{ol})^{-1} r, \quad (2.16)$$

$$y = (I - D_{ol})^{-1} C_{ol}x + (I - D_{ol})^{-1} r, \quad (2.17)$$

varur det karakteristiska polynomet för det återkopplade systemet kan identifieras som

$$\Phi_{cl}(s) = \det \left( sI - A_{ol} - B_{ol} (I - D_{ol})^{-1} C_{ol} \right). \quad (2.18)$$

Rötterna till detta polynom är alltså desamma som polerna till det återkopplade systemets överföringsfunktion  $(I - H_{11}(s) \cdot \Delta)^{-1}$ .

Vi får nu användning för följande matrisidentitet, vilken lätt kan verifieras med direkt multiplikation :

$$= \begin{bmatrix} I & A_{12}A_{22}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{11} - A_{12}A_{22}^{-1}A_{21} & 0 \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} I & 0 \\ A_{22}^{-1}A_{21} & I \end{bmatrix} \\ = \begin{bmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{bmatrix} \begin{bmatrix} I & A_{11}^{-1}A_{12} \\ 0 & I \end{bmatrix}. \quad (2.19)$$

Sätt in

$$\begin{aligned} A_{11} &= I - D_{ol}, & A_{12} &= C_{ol}, \\ A_{21} &= B_{ol}, & A_{22} &= sI - A_{ol} \end{aligned} \quad (2.20)$$

och tag determinanten av båda led, så får man, eftersom determinanterna av samtliga yttrematriser = 1,

$$\begin{aligned} &\det \left( I - D_{ol} - C_{ol} (sI - A_{ol})^{-1} B_{ol} \right) \det (sI - A_{ol}) \\ &= \det (I - D_{ol}) \det \left( sI - A_{ol} - B_{ol} (I - D_{ol})^{-1} C_{ol} \right), \end{aligned} \quad (2.21)$$

eller, med hjälp av ekv. (2.11), (2.12) och (2.18).

$$\det (I - H_{11}(s) \cdot \Delta) = c \frac{\Phi_{cl}(s)}{\Phi_{ol}(s)}, \quad (2.22)$$

där  $c = \det (I - D_{ol})$ , och alltså konstant.

Som tidigare påpekats, gäller förutsättningen att det öppna systemet är stabilt, vilket innebär att  $\Phi_{ol}(s)$  inte har rötter i högra halvplanet (HHP). Vi söker ett kriterium för att det slutna systemet också är stabilt, dvs för att  $\Phi_{cl}(s)$  inte heller har

några rötter i HHP. Här kommer den s k argumentprincipen för komplexa rationella funktioner till användning.

Argumentprincipen säger, att när den oberoende variabeln  $s$  för en rationell komplex funktion genomlöper en sluten kurva i det komplexa planet, så omkretsar motsvarande bildkurva origo ett antal gånger, som är lika med skillnaden i antal täljare och nämnarnollställen som ligger innanför den slutna kurvan ifråga.

Om vi alltså låter den slutna kurvan utgöras av imaginäraxeln tillsammans med en halvcirkel med "stor" radie i HHP (en s k Nyquist-kontur), kommer funktionen  $\det(I - H_{11}(s) \cdot \Delta)$  att omkretsa origo lika många gånger som det finns rötter till  $\Phi_{cl}(s)$  innanför denna kurva, och om man tänker sig att halvcirkelns radie är oändligt stor, kan man genom att observera bildkurvans beteende på detta sätt avgöra hur många rötter detta polynom har i HHP (obs återigen att  $\Phi_{ol}(s)$  saknar rötter i detta område).

Det avgörande blir hur funktionen  $\det(I - H_{11}(i\omega) \cdot \Delta)$  uppför sig då  $\omega$  genomlöper de reella talen från  $-\infty$  till  $+\infty$ , eftersom halvcirkeln med mycket stor radie i praktiken avbildas på en enda punkt.

Vi vill alltså studera vilka störningar  $\Delta$  som krävs för att denna funktion skall omkretsa origo. Om  $\Delta = 0$ , så är  $\det(I - H_{11}(i\omega) \cdot \Delta) = 1, \forall \omega$ , dvs hela Nyquist-konturen avbildas på punkten 1, och omkretsar därför inte origo, vilket ju stämmer med förutsättningen att det ostörda systemet är stabilt. Det är också uppenbart, att kurvan kommer att hålla sig i närheten av punkten 1 så länge som  $\Delta$  är tillräckligt litet. Av resonemanget ovan framgår i själva verket, att kurvan inte kan nå fram till origo om  $\bar{\sigma}(\Delta) < \frac{1}{\bar{\sigma}(H_{11}(i\omega))}, \forall \omega$ .

När  $\Delta$  ökas till sin storlek, finns möjligheten att bildkurvan träffar origo, dvs  $\exists \Delta \in \mathcal{D}: \exists \omega \in R: \det(I - H_{11}(i\omega) \cdot \Delta) = 0$ . Sökandet efter den minsta tillåtna störning som kan orsaka instabilitet görs med stegning genom "alla" frekvenser  $\omega$ , och bestämning för varje  $\omega$  av det minsta  $\Delta \in \mathcal{D}$  som gör matrisen  $I - H_{11}(i\omega) \cdot \Delta$  singular. Enligt definitionen av  $\mu_{\mathcal{D}}$  ovan är detta ingenting annat än inversen av  $\mu_{\mathcal{D}}(H_{11}(i\omega))$ , eftersom tillämpning av definitionerna i ekv. (2.7) och (2.8) ger

$$\frac{1}{\mu_{\mathcal{D}}(H_{11}(i\omega))} = \min \{ \bar{\sigma}(\Delta) : \det(I - H_{11}(i\omega) \cdot \Delta) = 0, \Delta \in \mathcal{D} \} \quad (2.23)$$

Om vi därför ritar upp denna storhet som funktion av frekvensen  $\omega$ , så kan vi bestämma vilken frekvens som kräver den minsta störningen för att uppfylla singularitetsvillkoret. Då vet vi att systemet kommer att behålla sin stabilitet för alla störningar som är mindre, eftersom  $\det(I - H_{11}(i\omega) \cdot \Delta) \neq 0$  för alla sådana  $\Delta$  samt alla  $\omega$ .

## 2.4 Beskrivning av enkel missil

En HVM-missil vid givet Mach-tal antas kunna beskrivas med följande linjära 2-tillståndsmodell:

$$\dot{\alpha} = k_1(-C_T + C_{N\alpha})\alpha + q + k_1 C_{N\delta} \delta \quad (2.24)$$

$$\dot{q} = k_2 C_{m\alpha} \alpha + k_2 \frac{d}{2v} q + k_2 C_{m\delta} \delta, \quad (2.25)$$

$$\eta = v(\dot{\alpha} - q) \quad (2.26)$$

där  $k_1, k_2, d$ , och  $v$  är konstanter vilka bestäms av missilens egenskaper, inklusive dess hastighet.

$\alpha$  är anfallsvinkeln,  $q$  missilens vinkelhastighet, styrsignalen  $\delta$  är rodervinkeln och utsignalen utgörs av  $\eta$  är tvärsaccelerationen.

Modellosäkerheten ligger i de aerodynamiska derivatorna  $C_T, C_{N\alpha}, C_{m\alpha}, C_{N\delta}$ , and  $C_{m\delta}$ , vilka skrivs som

$$C_T = C_{T0} + p_T f_T C_{T0} \quad (2.27)$$

$$C_{N\alpha} = C_{N\alpha 0} + p_{N\alpha} f_{N\alpha} C \quad (2.28)$$

$$C_{m\alpha} = C_{m\alpha 0} + p_{m\alpha} f_{m\alpha} C_{m\alpha 0} \quad (2.29)$$

$$C_{N\delta} = C_{N\delta 0} + p_{N\delta} f_{N\delta} C_{N\delta 0} \quad (2.30)$$

$$C_{m\delta} = C_{m\delta 0} + p_{m\delta} f_{m\delta} C_{m\delta 0} \quad (2.31)$$

med  $C_{T0}, C_{N\alpha 0}, C_{m\alpha 0}, C_{N\delta 0}$ , och  $C_{m\delta 0}$  som nominella värden för respektive aerodynamiska derivator,  $f_T, f_{N\alpha}, f_{m\alpha}, f_{N\delta}$ , och  $f_{m\delta}$  som maximala relativa fel och  $p_T, p_{N\alpha}, p_{m\alpha}, p_{N\delta}$ , and  $p_{m\delta}$  som osäkerhetsparametrar, vilka samtliga tillhör intervallet  $[-1, 1]$ .

**2.4.1 Omskrivning av modellen till standardform** Med dessa osäkerheter insatta, och med  $x_1 \hat{=} \alpha$ ,  $x_2 \hat{=} q$ ,  $u \hat{=} \delta$ , och  $y \hat{=} \eta$ , kan modellen skrivas

$$\dot{x}_1 = a_{11}(p_1, p_2)x_1 + x_2 + b_1(p_4)u \quad (2.32)$$

$$\dot{x}_2 = a_{21}(p_3)x_1 + a_{22}x_2 + b_2(p_5)u, \quad (2.33)$$

$$y = v(\dot{x}_1 - x_2) \quad (2.34)$$

där  $a_{11}, a_{21}, b_1$ , och  $b_2$  är linjära funktioner av sina argument, enligt

$$a_{11}(p_1, p_2) = a_{11}^0 + a_{11}^1 p_1 + a_{11}^2 p_2 \quad (2.35)$$

$$a_{21}(p_3) = a_{21}^0 + a_{21}^1 p_3 \quad (2.36)$$

$$b_1(p_4) = b_1^0 + b_1^1 p_4 \quad (2.37)$$

$$b_2(p_5) = b_2^0 + b_2^1 p_5 \quad (2.38)$$

De linjära koefficienterna beräknas här uttryckta i de ursprungligt givna systemkonstanterna som

$$a_{11}^0 = k_1(C_{N\alpha 0} - C_{T0}), \quad a_{11}^1 = k_1 f_{N\alpha} C_{N\alpha 0}, \quad a_{11}^2 = -k_1 f_T C_{T0} \quad (2.39)$$

$$a_{21}^0 = k_2 C_{m\alpha 0}, \quad a_{21}^1 = k_2 f_{m\alpha} C_{m\alpha 0} \quad (2.40)$$

$$a_{22} = k_2 d / 2v \quad (2.41)$$

$$b_1^0 = k_1 C_{N\delta 0}, \quad b_1^1 = k_1 f_{N\delta} C_{N\delta 0} \quad (2.42)$$

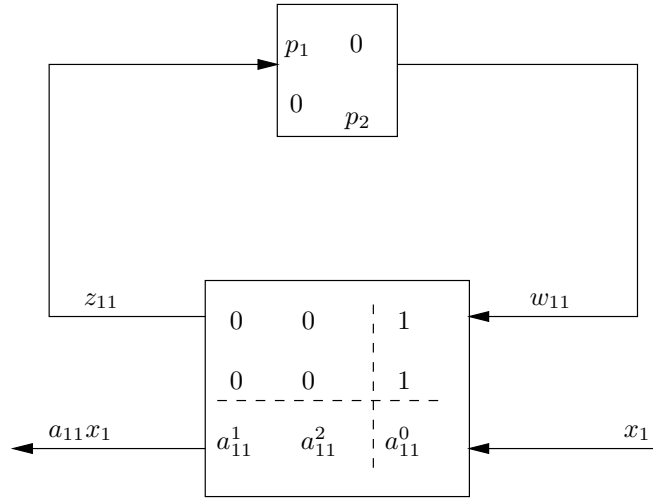
$$b_2^0 = k_2 C_{m\delta 0}, \quad b_2^1 = k_2 f_{m\delta} C_{m\delta 0} \quad (2.43)$$

Vi har också ändrat beteckningar för osäkerheterna enligt  $p_T = p_1, p_{N\alpha} = p_2, p_{m\alpha} = p_3, p_{N\delta} = p_4, p_{m\delta} = p_5$ .

**2.4.2 Transformerings till LFT:er** Det linjära uttrycket för  $a_{11}(p_1, p_2)$  ovan kan betraktas som en in-ut-relation på följande LFT-form:

och  $a_{21}(p_3), b_1(p_4)$ , and  $b_2(p_5)$  har alla den enklare strukturen enligt figur 2.5, där asteriskerna betecknar olika värden för olika koefficienter.

Inspektion av ekvationerna (2.32), (2.33) och (2.34) visar att hela systemet nu kan representeras av blockdiagrammet i figur 2.6.

Figur 2.4: LFT-representation för  $a_{11}(p_1, p_2)$ 

Detta diagram kan, i sin tur skrivas som ekvationerna

$$\bar{z}_{11} = O_{2 \times 2} \bar{w}_{11} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} x_1 \quad (2.44)$$

$$z_{21} = 0 \cdot w_{21} + 1 \cdot x_1 \quad (2.45)$$

$$z_1 = 0 \cdot w_1 + 1 \cdot u \quad (2.46)$$

$$z_2 = 0 \cdot w_2 + 1 \cdot u \quad (2.47)$$

$$\dot{x}_1 = [a_{11}^1, a_{11}^2] \cdot \bar{w}_{11} + b_1^1 w_1 + a_{11}^0 x_1 + x_2 + b_1^0 u \quad (2.48)$$

$$\dot{x}_2 = a_{21}^1 w_{21} + b_2^1 w_2 + a_{21}^0 x_1 + a_{22} x_2 + b_2^0 u \quad (2.49)$$

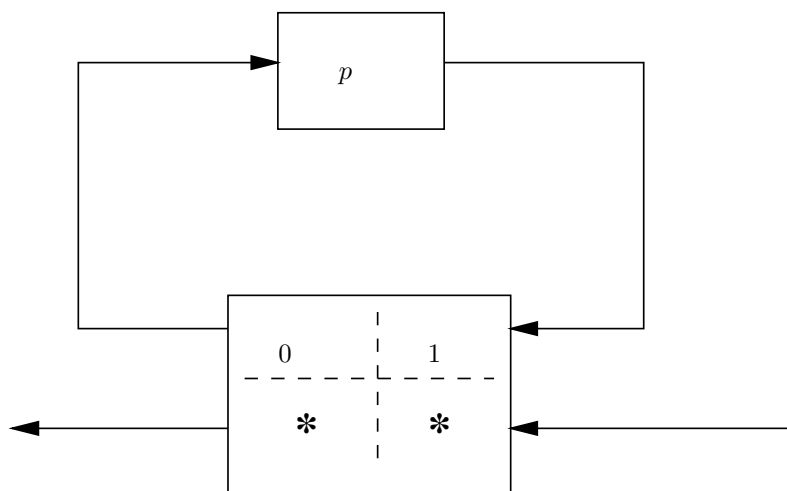
$$y = [va_{11}^1, va_{11}^2] \cdot \bar{w}_{11} + vb_1^1 w_1 + va_{11}^0 x_1 + vb_1^0 u \quad (2.50)$$

eller, i vektorform:

$$\begin{bmatrix} \bar{z}_{11} \\ z_{21} \\ z_1 \\ z_2 \\ \dot{x}_1 \\ \dot{x}_2 \\ y \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ a_{11}^1 & a_{11}^2 & 0 & b_1^1 & 0 & a_{11}^0 & 1 & b_1^0 \\ 0 & 0 & a_{21}^1 & 0 & b_2^1 & a_{21}^0 & a_{22} & b_2^0 \\ va_{11}^1 & va_{11}^2 & 0 & vb_1^1 & 0 & va_{11}^0 & 0 & vb_1^0 \end{bmatrix} \begin{bmatrix} \bar{w}_{11} \\ w_{21} \\ w_1 \\ w_2 \\ x_1 \\ x_2 \\ u \end{bmatrix} \quad (2.51)$$

vilket kan skrivas som

$$\bar{z} = M\bar{w} \quad (2.52)$$

Figur 2.5: LFT-representation för  $a_{21}(p)$ ,  $b_1(p)$  och  $b_2(p)$ 

med vektorerna och matrisen partitionerade enligt

$$\bar{z} = \begin{bmatrix} \bar{z}_{11} \\ z_{21} \\ z_1 \\ z_2 \end{bmatrix}, \bar{w} = \begin{bmatrix} \bar{w}_{11} \\ w_{21} \\ w_1 \\ w_2 \end{bmatrix} \quad (2.53)$$

och

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \quad (2.54)$$

där tydligen

$$M_{11} = O_{5 \times 5}, M_{12} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, M_{13} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad (2.55)$$

$$M_{21} = \begin{bmatrix} a_{11}^1 & a_{11}^2 & 0 & b_1^1 & 0 \\ 0 & 0 & a_{21}^1 & 0 & b_2^1 \end{bmatrix}, M_{22} = \begin{bmatrix} a_{11}^0 & 1 \\ a_{21}^0 & a_{22} \end{bmatrix}, M_{23} = \begin{bmatrix} b_1^0 \\ b_2^0 \end{bmatrix} \quad (2.56)$$

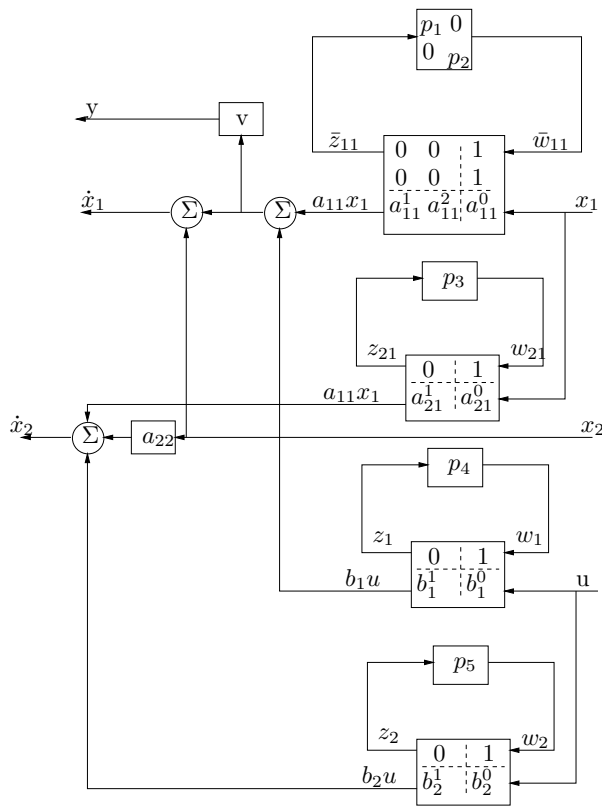
$$M_{31} = [va_{11}^1 \quad va_{11}^2 \quad 0 \quad vb_1^1 \quad 0], M_{32} = [va_{11}^0 \quad 0], M_{33} = [vb_1^0] \quad (2.57)$$

Med dessa symboler kan missilen representeras som i figur 2.7.

## 2.5 Stabilitetsanalys av ostyrd enkel missil

**2.5.1 Teoretisk analys** Vi vill nu prova den tidigare härledda  $\mu$ -metoden genom att tillämpa den på den ovan beskrivna enkla missilmodellen (ekv. (2.32), (2.33)). Detta systems stabilitet bestäms helt och hållet av matrisen

$$A_m = \begin{bmatrix} a_{11}(p_1, p_2) & 1 \\ a_{21}(p_3) & a_{22} \end{bmatrix} \quad (2.58)$$



Figur 2.6: LFT-representation för tillståndsekvationerna (2.32) - (2.34)

med karakteristiska polynomet

$$\Phi_m(s) = s^2 - (a_{11}(p_1, p_2) + a_{22})s + a_{11}(p_1, p_2)a_{22} - a_{21}(p_3). \quad (2.59)$$

Med aktuella numeriska värden på de i ekv. (2.35) och (2.36) ingående konstanterna (se ekvation (2.39) (2.40) och (2.41), samt  $p_1$  och  $p_2$  begränsade till intervallet  $[-1, 1]$ ), blir de karakteristiska rötterna komplexkonjugerade med realdelen  $\frac{1}{2}(a_{11}(p_1, p_2) + a_{22})$ .

Detta betyder att stabiliteten hos systemet enbart beror av parametrarna  $p_1$  och  $p_2$  men är oberoende av  $p_3$  (samt förstås  $p_4$  och  $p_5$ ).

Det område i  $p_1 - p_2$ -planet som ger stabilt oreglerat system ges således av olikheten

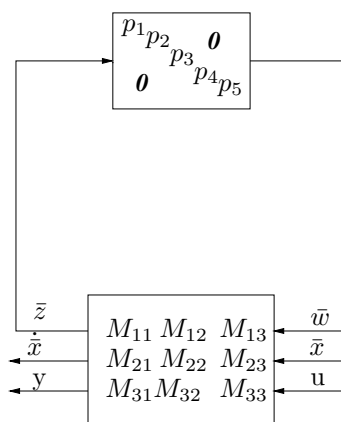
$$a_{11}(p_1, p_2) + a_{22} < 0 \quad (2.60)$$

eller, med insättning av det linjära uttrycket för  $a_{11}(p_1, p_2)$  enligt ekvation (2.35),

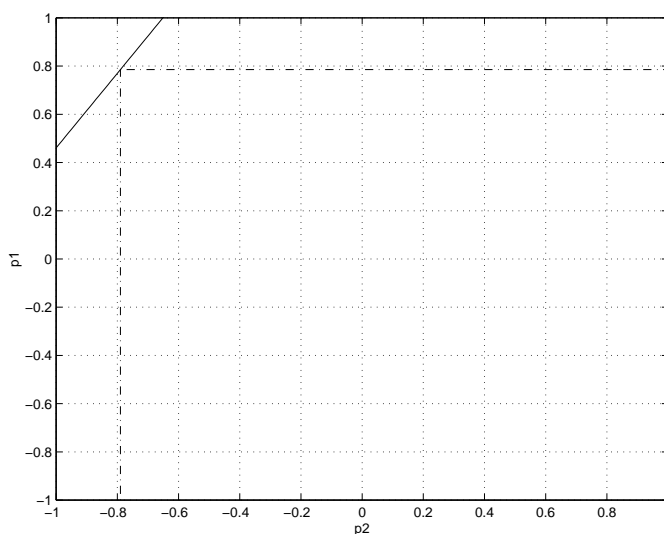
$$p_1 < -\frac{a_{11}^2 p_2 + a_{11}^0 + a_{22}}{a_{11}^1}. \quad (2.61)$$

Detta är det område som ligger under den heldragna linjen i figur 2.8.

**2.5.2  $\mu$ -analys** För att testa metoden med  $\mu$ -analys genom att jämföra dess resultat med det teoretiska ovan, använder vi oss nu av vetskapen att stabiliteten enbart



Figur 2.7: LFT för ekvation (2.52)



Figur 2.8: Stabilitetsområde för ostyrd missil

avgörs av värdena hos  $p_1$  och  $p_2$ . Vi kan då sätta övriga parametrar = t ex 0 och göra om systemet till en LFT av samma form som i figur 2.2 med

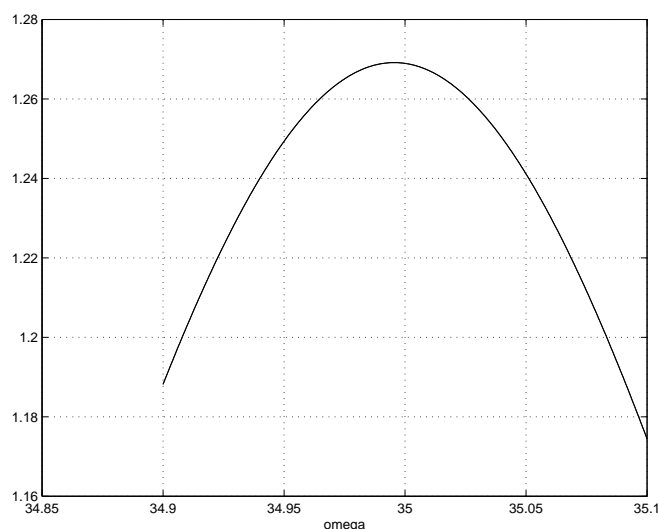
$$\Delta(p) = \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix}. \quad (2.62)$$

Överföringsfunktionen  $H_{11}(s)$  blir således av dimensionen  $2 \times 2$  och bestäms av MATLAB-programmen `approxLPV` och `rat2lft`, vilka beskrivs i [14]. Därefter anropas programmet `mu`, som ingår i  $\mu$ -Analysis and Synthesis Toolbox och bestämmer  $\mu$  för ett ändligt antal specificerade frekvenser. Med reellt  $\Delta(p)$  är  $\mu \neq 0$  endast för ett fåtal frekvenser, eftersom ett nödvändigt krav för singularitet hos  $I - H_{11}(i\omega) \cdot \Delta$  är att åtminstone något element i  $H_{11}(i\omega)$  är rent reellt, något som oftast inte blir fallet om de diskreta frekvenserna väljs utan stor insikt om systemets egenskaper.

Det enklaste sättet att kringgå detta problem, åtminstone med de programhjälpmedel som står till vårt förfogande, är att tillåta  $\Delta(p)$  att vara komplex, dvs vi låter  $\mathcal{D}$  vara  $C^{m \times m}$  istället för  $R^{m \times m}$ . Priset för detta är att metoden blir konservativ, dvs störningar som egentligen inte destabiliserar systemet blir otillåtna.



I figur 2.9 visas  $\mu_{\mathcal{D}}(H_{11}(i\omega))$  i närheten av sitt maximum. Vi kan avläsa  $\max_{\omega} \mu_{\mathcal{D}}(H_{11}(i\omega)) = 1.2692$ . Om vi alltså bara tillåter (reella) störningar som inte har större norm än störningen  $\Delta_{\max}$  där  $\bar{\sigma}(\Delta_{\max}) = \max\{|p_1^{\max}|, |p_2^{\max}|\} = \frac{1}{\max_{\omega} \mu_{\mathcal{D}}(H_{11}(i\omega))} = 0.7879$ , så är stabilitet garanterad. Detta svarar mot det område i figur 2.8 som ligger under den horisontella och till höger om den vertikala streckade linjen. Detta är ju en delmängd av hela stabilitetsområdet, och vi ser alltså att metoden i detta fall ger ett korrekt, om än konservativt, resultat.



Figur 2.9:  $\mu$  som funktion av frekvensen med oreglerad missil

## 2.6 Konstruktion av styrautomat för enkel missil

Stabilitetsanalysen ovan av den enkla missilen ger oss ett område i parameterplanet med garanterad stabilitet. Emellertid är systemet, trots dess rent formella stabilitet, helt oacceptabelt vad gäller dess dämpningsegenskaper. Det nominella systemet ( $p_1 = p_2 = 0$ ) har en tidskonstant på 4 s (egenfrekvens 35 rad/s) och måste därför regleras.

Regulatorn måste konstrueras utifrån vad som är känt om systemet. Osäkerhetsparametrarna är naturligtvis okända, varför styrautomaten får baseras på det nominella systemet. Resultatet blir således ett system som har önskade egenskaper under förutsättning att  $p_1 = p_2 = \dots = p_5 = 0$ .

Det enkla missilsystemet tänks alltså beskrivet som (jfr ekvation (2.32) - (2.34):

$$\begin{aligned} \dot{x} &= A_m x + B_m u \\ y &= C_m x + D_m u. \end{aligned} \quad (2.63)$$

Med tillståndsåterkopplingen

$$u = -Kx + Fr, \quad (2.64)$$

där  $K$  och  $F$  är konstanter och  $r$  är en referenssignal (dvs tvärsaccelerationens börvärde), erhålls systemet

$$\begin{aligned} \dot{x} &= (A_m - B_m K) x + B_m Fr \\ y &= (C_m - D_m K) x + D_m Fr. \end{aligned} \quad (2.65)$$

Stabiliteten hos detta återkopplade system bestäms alltså av egenvärdena hos  $A_m - B_m K$ . Dessa kan genom lämpligt val av vektorn  $K$  placeras fritt (under den självklara begränsningen att komplexa egenvärden uppträder i konjugerade par).

Lämpliga egenvärden är de som svarar mot en relativ dämpning  $\zeta = 0.7$  och egenfrekvensen  $\omega_0 = 25 \text{ rad/s}$ , vilket ger en tidskonstant på  $\frac{1}{\zeta \omega_0} \approx 0.06 \text{ s}$ .

För att det statiska felet skall elimineras, dvs tvärsaccelerationen ska svänga in mot det önskade värdet, skalas börvärdet  $r$  med faktorn  $F$ , vilken väljs så att utsignalen  $y_\infty$  efter insvängning vid pålagt konstant börvärde  $r_0$  blir lika med detta börvärde. Efter insvängning är tillståndet

$$x_\infty = -(A_m - B_m K)^{-1} B_m F r_0, \quad (2.66)$$

vilket ger

$$y_\infty = \left[ -(C_m - D_m K) (A_m - B_m K)^{-1} B_m + D_m \right] F r_0, \quad (2.67)$$

varför valet av  $F$  blir

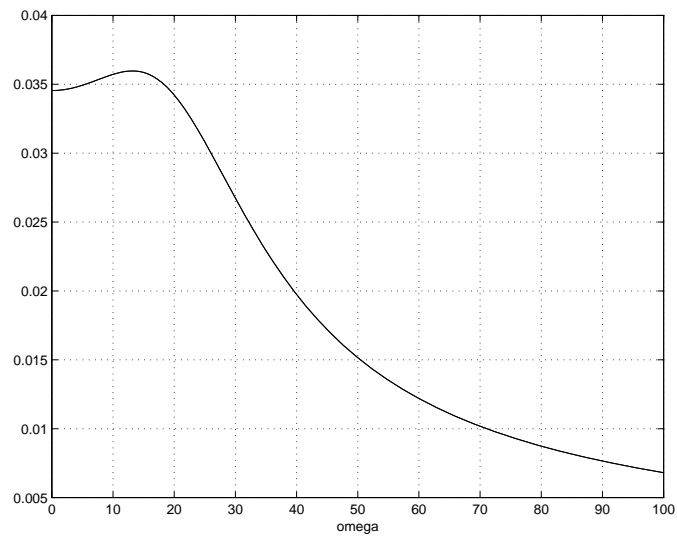
$$F = \left[ -(C_m - D_m K) (A_m - B_m K)^{-1} B_m + D_m \right]^{-1} \quad (2.68)$$

## 2.7 Stabilitetsanalys av styrd enkel missil

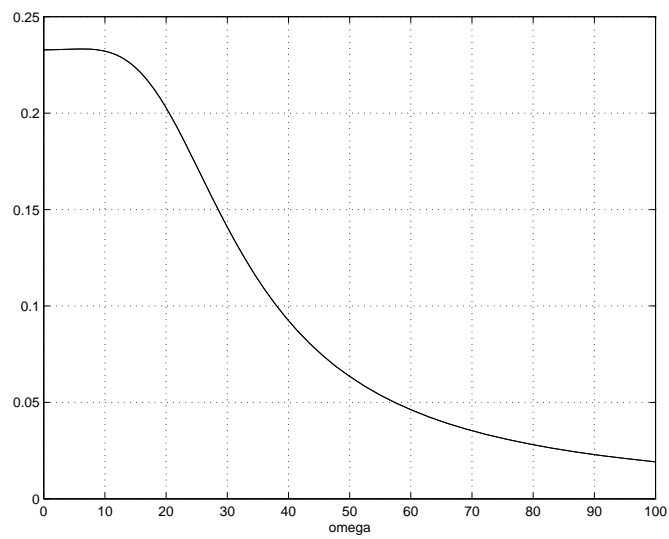
Det faktum att osäkerhetsparametrarnas värden i verkligheten avviker från 0 medför naturligtvis avvikelser hos det reglerade systemet från det önskade beteendet. Stabiliteten hos detta system beror på samtliga 5 osäkerhetsparametrar, till skillnad från det oreglerade systemets stabilitet, som ju påverkas endast av  $p_1$  och  $p_2$ . Av beräknings-tekniska skäl nöjer vi oss emellertid i denna rapport med att analysera stabiliteten hos det reglerade systemet dels i det fall då endast  $p_1$  och  $p_2$  får avvika från 0, dels i det fall då även  $p_3$  släpps fri.

I figur 2.10 visas  $\mu$  som funktion av frekvensen i det förra fallet, dvs då endast  $p_1$  och  $p_2$  är fria. Vi ser att kurvan har maximum = 0.036. Av detta drar vi slutsatsen att systemet är stabilt om  $|p_1|$  och  $|p_2|$  båda understiger  $\frac{1}{0.036} \approx 27$ .

I figur 2.11 ses motsvarande kurva då  $p_1, p_2$  och  $p_3$  är fria. Denna kurva har sitt största värde = 0.23, vilket innebär att stabilitet råder om beloppen av alla tre parametrarna är mindre än 4.2.



Figur 2.10:  $\mu$  som funktion av frekvensen för reglerad missil med två osäkerhetsparametrar

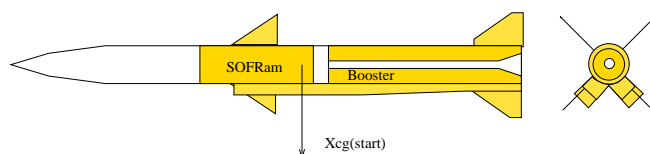


Figur 2.11:  $\mu$  som funktion av frekvensen för reglerad missil med tre osäkerhetsparametrar



## 3. Modellering av en Ramrobot

### 3.1 Översikt



Figur 3.1: Principskiss över missilen från sidan och bakifrån

Skjutförloppet är tänkt att inledas med en boostfas och sedan, efter en kort coastfas, övergå till en sustainfas.

**3.1.1 Boostfas** Under boostfasen brinner motor1 dvs den krutdrivna boostern. Boosterns uppgift är att ge missilen en tillräckligt hög hastighet för att rammotorn skall fungera d v s i ett intervallet mellan ca mach 2 och mach 4.

Boostfasen varar i 3s. Under de första 0.1s är missilen ostyrd och rodren är låsta. Sedan kommer en 0.4s lång stabiliserings fas. De återstående 2.5s används för att göra en lämplig upptagning. Upptagningens storlek beror på hur långt bort målet befinner sig och vilken hastighet man vill att missilen skall ha vid boostfasens slut.

**3.1.2 Coastfas** Under denna fas på 0.5s skall anfallsvinkeln och snedanblåsningvinkeln styras ned till noll d v s  $\alpha = \beta = 0$ . Detta gör man för att säkerställa att rammotorn kan starta.

**3.1.3 Sustainerfas** För att SOFRammotorn inte skall slockna måste fart, höjd, anfalls- och snedanblåsningvinklar hållas inom vissa begränsningar (se senare kapitel). Om motorn slocknat så kan den inte tändas igen.

Under de första 0.5s skall  $\alpha$  och  $\beta$  fortsätta att hållas till noll.

Hur länge rammotorn brinner beror på hur fort missilen flyger, på vilken höjd och med vilken anfallsvinkel.

### 3.2 Missilmodell

**3.2.1 Omgivning** I detta avsnitt beskrivs yttre förhållanden såsom ljudhastighets och lufttäthets variation med höjd och marktemperatur. Här beräknas även g-kraftens komponenter i kroppsfasta systemet (body).

$T_{mark}$	: marktemperatur	[ °C]
$T_0$	: -273.14	[ °C]
$p_{mark}$	: 1000.0	[ hPa]
$k_p$	: $-1.5688 \cdot 10^{-4}$	[ /m]
$R$	: 287.04	[ kJ/kgK]
$g$	: 9.81	[ m/s <sup>2</sup> ]
$T_{g2b}$	: Transformationsmatris	[-]

Höjd tropopaus [m]:

$$H_{trop} = 9600 + 80T_{mark} \quad (3.1)$$

Temperaturgradient [ $^{\circ}C/m$ ]:

$$T_{grad} = -0.0061 + 18.1 \cdot 10^{-6}T_{mark} \quad (3.2)$$

Temperatur [ $^{\circ}C$ ]:

$$T = \begin{cases} T_{mark} + T_{grad} * hojd & om\ hojd \leq H_{trop} \\ T_{mark} + T_{grad} * H_{trop} & annars \end{cases} \quad (3.3)$$

Ljudhastighet [m/s]:

$$v_{ljud} = 20.046\sqrt{T - T_0} \quad (3.4)$$

Omgivningstryck [ $N/m^2$ ]:

$$p = \begin{cases} p_{mark} \left( \frac{T_{mark} - T_0}{T - T_0} \right)^{-5.7} & om\ hojd \leq H_{trop} \\ p_{mark} \left( \frac{T_{mark} - T_0}{T - T_0} \right)^{-5.7} \cdot e^{k_p(hojd - H_{trop})} & annars \end{cases} \quad (3.5)$$

Lufttäthet [ $kg/m^3$ ]:

$$\rho = \frac{100 \cdot p}{R(T - T_0)} \quad (3.6)$$

gravitation i body [ $m/s^2$ ]:

$$g_b = T_{g2b} \begin{pmatrix} g \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{pmatrix} g \quad (3.7)$$

**3.2.2 Aerodynamik** Huvudsyftet med detta avsnitt är att räkna ut de aerodynamiska krafterna och momenten, dvs de krafter och moment som uppstår p g a att missilen rör sig med en viss hastighet i ett lufthav. Utöver detta beräknas även de vinklar, anfallsvinkel och snedanblåsningvinkel, med vilka vinden infaller mot missilkroppen.

Eventuell styrning via aerodynamiska roder fogas in i detta avsnitt via rodervinkelutslag,  $\delta$ .

$d$	: referenslangd	[ m ]
$X_{ref}$	: aerodynamisk referenspunkt	[ m ]
$X_{gc}$	: masscentrum lage	[ m ]
$C_T$	$= C_T(Mach)$	[ - ]
$C_C$	$= C_C(Mach, \alpha, \beta)$	[ - ]
$C_N$	$= C_N(Mach, \alpha, \beta)$	[ - ]
$C_{T_{end}}$	$= C_{T_{end}}(\dot{m}_{m2})$	[ - ]
$C_{T_{\delta_r}}$	$= C_{T_{\delta_r}}(Mach, \alpha, \beta)$	[ - ]
$C_{T_{\delta_e}}$	$= C_{T_{\delta_e}}(Mach)$	[ - ]
$C_{T_{\delta_a}}$	$= C_{T_{\delta_a}}(Mach)$	[ - ]
$C_{C_{\delta}}$	$= C_{C_{\delta}}(Mach, \alpha, \beta)$	[ - ]
$C_{N_{\delta}}$	$= C_{N_{\delta}}(Mach, \alpha, \beta)$	[ - ]
$C_l$	$= C_l(Mach)$	[ - ]
$C_m$	$= C_m(Mach, \alpha, \beta)$	[ - ]
$C_n$	$= C_n(Mach, \alpha, \beta)$	[ - ]
$C_{l_{\delta}}$	$= C_{l_{\delta}}(Mach, \alpha, \beta)$	[ - ]
$C_{m_{\delta}}$	$= C_{m_{\delta}}(Mach, \alpha, \beta)$	[ - ]
$C_{n_{\delta}}$	$= C_{n_{\delta}}(Mach, \alpha, \beta)$	[ - ]
$C_{l_p}$	$= C_{l_p}(Mach)$	[ - ]
$C_{m_q}$	$= C_{m_q}(Mach)$	[ - ]
$C_{n_r}$	$= C_{n_r}(Mach)$	[ - ]
$\delta_r$	: sidroder {rudder}	[ rad ]
$\delta_e$	: höjdroder {elevator}	[ rad ]
$\delta_a$	: rollroder {aileron}	[ rad ]

Totalhastighet [m/s]:

$$v_{tot} = \sqrt{v_x^2 + v_y^2 + v_z^2} \quad (3.8)$$

Machtal [-]:

$$Mach = \frac{v_{tot}}{v_{ljud}} \quad (3.9)$$

Anfallsvinkel [grader]:

$$\alpha = \arctan \frac{v_z}{v_x} : [-90, 90] \quad (3.10)$$

Snedanblåsningsvinkel <sup>1</sup> [grader]:

$$\beta = \arcsin \frac{v_y}{v_{tot}} : [-90, 90] \quad (3.11)$$

Rodervinklar (x-konfig) [rad]:

$$\begin{aligned} \delta_r &= \frac{-\delta_1 - \delta_2 + \delta_3 + \delta_4}{4} \\ \delta_e &= \frac{\delta_1 - \delta_2 - \delta_3 + \delta_4}{4} \\ \delta_a &= \frac{\delta_1 + \delta_2 + \delta_3 + \delta_4}{4} \end{aligned} \quad (3.12)$$

<sup>1</sup>Amn: För symmetriska missiler används ofta istället en vinkel som definieras på följande sätt:  
 $\arctan \frac{v_y}{v_x}$

Dynamiskt tryck [N/m<sup>2</sup>]:

$$q = \frac{\rho v_{tot}^2}{2} \quad (3.13)$$

Referensarea [m<sup>2</sup>]:

$$S_{ref} = \frac{\pi d^2}{4} \quad (3.14)$$

Kraft ekv [N]:

$$F_A = \begin{cases} -qS_{ref}(C_T + C_{T_{end}} + C_{T_{\delta_r}} \delta_r + C_{T_{\delta_e}} \delta_e + C_{T_{\delta_a}} \delta_a) \\ -qS_{ref}(C_C + C_{C_\delta} \delta_r) \\ -qS_{ref}(C_N + C_{N_\delta} \delta_e) \end{cases} \quad (3.15)$$

Moment ekv [Nm]:

$$M_A = \begin{cases} qS_{ref}d(C_l + C_{l_\delta} \delta_a + \frac{d}{2v_{tot}} C_{l_p} \omega_x) \\ qS_{ref}d(C_m + C_{m_\delta} \delta_e + \frac{d}{2v_{tot}} C_{m_q} \omega_y - \frac{X_{ref} - X_{cgx}}{d} (C_N + C_{N_\delta} \delta_e)) \\ qS_{ref}d(C_n + C_{n_\delta} \delta_r + \frac{d}{2v_{tot}} C_{n_r} \omega_z - \frac{X_{ref} - X_{cgx}}{d} (C_C + C_{C_\delta} \delta_r)) \end{cases} \quad (3.16)$$

**3.2.3 Motor1 (Krutdriven booster)** I detta avsnitt beskrivs en krutmotors inre liv. Som drivkraft läses en normerad kurva in. Den skalas sedan med specifikkimpuls, krutmassa och brinntid till en kraft. Observera att denna beskrivning bara gäller en motor med en krutcylander som brinner symmetriskt inifrån och ut mot skalet.

Eventuell dysstyrning av boostern fogas in i detta avsnitt via dyskrafter,  $\delta_{dys1}$ .

$m_k$	: krutets massa	[kg]
$I_{sp}$	: specifikkimpuls för krutet	[Ns/kg]
$I_{tot}$	: totalimpuls för krutet	[Ns]
$t_b$	: brinntid för krutet	[s]
$P'_D(t)$	: normerad drivkraft	[N]
$\rho_k$	: krutets densitet	[kg/m <sup>3</sup> ]
$L$	: krutcylinderns längd	[m]
$r$	: krutcylinderns ytterradi	[m]
$x_{dys1}$	: dysans läge	[m]
$r_{dys1}$	: dysans radi	[m]
$V_{dys1}$	: dysans hastighet	[m/s]
$\delta_{dys1}$	: dysstyrning	[N]

Drivkraft [N]:

$$P_D = P'_D(t) \frac{I_{sp} m_{k0}}{t_b} \quad (3.17)$$

$$\int_{t_0}^{t_b} P'_D(t) dt = 1.0$$

Massflöde [kg/s] och massa [kg]:

$$\dot{m}_1 = -\frac{P_D}{I_{sp}} = -\frac{P_D m_{k0}}{I_{tot}} \quad (3.18)$$

$$m_{m1}(t) = m_{m1}(t-1) + \dot{m}_1 dt \quad (3.19)$$

$$m_k(t) = m_{m1}(t) - m_{m10} \quad (3.20)$$



Masscentrum [m] och masscentrumhastighet [m/s]:

$$X_{cg_{m1}} = \frac{X_{cg_{m10}}m_{m10} + X_{cg_k}m_k}{m_{m1}} \quad (3.21)$$

$$\dot{X}_{cg_{m1}} = \dot{m}_1 \frac{X_{cg_k} - X_{cg_{m1}}}{m_{m1}} + m_k \frac{\dot{X}_{cg_k}}{m_{m1}} \quad (3.22)$$

Tröghetsmoment och dess derivata m a p  $x=0$  [ $\text{kgm}^2$ ][ $\text{kgm}^2/\text{s}$ ]  
Krutcylinder som brinner inifrån och ut mot skalet (nollskilda element):

$$\begin{aligned} I(1,1)_{m1} &= I(1,1)_{m10} + m_k r^2 - \frac{m_k^2}{2\rho_k L \pi} \\ I(2,2)_{m1} &= I(2,2)_{m10} + \frac{m_k r^2}{2} - \frac{m_k^2}{4\rho_k L \pi} + \frac{m_k L^2}{12} + m_{m1} X_{cg_{m1,x}}^2 \\ I(3,3)_{m1} &= I(2,2)_{m1} \end{aligned} \quad (3.23)$$

$$\begin{aligned} \dot{I}(1,1)_{m1} &= \dot{m}_1 r^2 - \dot{m}_1 \frac{m_k}{\rho_k L \pi} \\ \dot{I}(2,2)_{m1} &= \frac{\dot{m}_1 r^2}{2} - \dot{m}_1 \frac{m_k}{2\rho_k L \pi} + \dot{m}_1 \frac{L^2}{12} + \dot{m}_1 X_{cg_{m1,x}}^2 + m_k X_{cg_{m1,x}} \dot{X}_{cg_{m1,x}} \\ \dot{I}(3,3)_{m1} &= \dot{I}_{m1}(2,2) \end{aligned} \quad (3.24)$$

Framdrivande kraft [N]:

$$F_{m1} = \begin{pmatrix} \sqrt{P_D^2 - \delta_{dys1_y}^2 - \delta_{dys1_z}^2} \\ \delta_{dys1_y} \\ \delta_{dys1_z} \end{pmatrix} \quad (3.25)$$

Den framdrivande gasens hastighet [m/s]:

$$V_{gas1} = -F_{m1}/\dot{m}_1 \quad (3.26)$$

**3.2.4 Motor2 (SOFRammotor)** Detta är en rammotor med fast bränsle (Solid-Fuel). Observera att denna beskrivning endas gäller en SOFRammotor där bränslet brinner bakifrån och framåt. Både drivkraften och massflödet läses från tabeller, (eller funktioner som representerar tabelldata).

Eventuell dysstyrning av SOFRammotorn fogas in i detta avsnitt via dyskrafter,  $\delta_{dys2}$ .

$th$	: gaspådrag	[–]
$\phi(th) : [0.4..1.0]$	: förbränningsgrad	[–]
$\gamma(v_{tot}, \beta, \alpha) : [0/1]$	: livhållningsparameter	[–]
$\rho_b$	: bränslets densitet	[ $\text{kg}/\text{m}^3$ ]
$X_{m2}$	: motorns läge (framkant)	[m]
$L$	: krutcylinderns längd	[m]
$r$	: krutcylinderns ytterradie	[m]
$x_{cg_{m20}}$	: tom motors masscentrum	[m]
$m_{m20}$	: tom motors massa	[kg]
$x_{cg_b}$	: bränslets masscentrum	[m]
$\dot{x}_{cg_b}$	: bränslets masscentrumhastighet	[m]
$m_{b0}$	: bränslets totala massa	[kg]
$x_{dys2}$	: dysans läge	[m]
$r_{dys2}$	: dysans radie	[m]
$V_{dys2}$	: dysans hastighet	[m/s]
$\delta_{dys2}$	: dysstyrning	[N]

Drivkraft [N]:

$$P_D = P_D(Mach, hojd, \alpha, \phi, \beta, \gamma) \quad (3.27)$$

Massflöde [kg/s]:

$$\dot{m}_2 = \frac{\dot{m}_2(Mach, hojd, \alpha)}{FuelAirRatio} \quad (3.28)$$

Massa [kg]:

$$m_{m2}(t) = m_{m2}(t-1) + \dot{m}_2 dt \quad (3.29)$$

$$m_b(t) = m_{m2}(t) - m_{m20} \quad (3.30)$$

Masscentrum [m] och masscentrumhastighet [m/s]:

$$X_{cg_b} = X_{m2} + \frac{m_b(t) L}{m_{b0} 2} \quad (3.31)$$

$$\dot{X}_{cg_b} = \frac{\dot{m}_2 L}{m_{b0} 2} \quad (3.32)$$

$$X_{cg_{m2}} = \frac{X_{cg_{m20}} m_{m20} + X_{cg_b} m_b}{m_{m2}} \quad (3.33)$$

$$\dot{X}_{cg_{m2}} = \dot{m}_2 \frac{X_{cg_b} - X_{cg_{m2}}}{m_{m2}} + m_b \frac{\dot{X}_{cg_b}}{m_{m2}} \quad (3.34)$$

Tröghetsmoment och dess derivata m a p x=0 [kgm<sup>2</sup>] [kgm<sup>2</sup>/s]

$$\begin{aligned} I(1,1)_{m2} &= I(1,1)_{m20} + m_b \frac{r^2}{2} \\ I(2,2)_{m2} &= I(2,2)_{m20} + \frac{m_b r^2}{4} + \frac{m_b^3}{12\rho_b^2 L \pi^2 r^4} + m_{m2} X_{cg_{m2}.x}^2 \\ I(3,3)_{m2} &= I(2,2)_{m2} \end{aligned} \quad (3.35)$$

$$\begin{aligned} \dot{I}(1,1)_{m2} &= \dot{m}_2 \frac{r^2}{2} \\ \dot{I}(2,2)_{m2} &= \frac{\dot{m}_2 r^2}{4} + \dot{m}_2 \frac{m_b^2}{4\rho_b^2 L \pi^2 r^4} + \dot{m}_2 X_{cg_{m2}.x}^2 + m_2 X_{cg_{m2}.x} \dot{X}_{cg_{m2}.x} \\ \dot{I}(3,3)_{m2} &= \dot{I}_{m2}(2,2) \end{aligned} \quad (3.36)$$

Framdrivande kraft [N]:

$$F_{m2} = \begin{pmatrix} \sqrt{P_D^2 - \delta_{dys2_y}^2 - \delta_{dys2_z}^2} \\ \delta_{dys2_y} \\ \delta_{dys2_z} \end{pmatrix} \quad (3.37)$$

Den framdrivande gasens hastighet [m/s]:

$$V_{gas2} = -F_{m2}/\dot{m}_2 \quad (3.38)$$

**3.2.5 Body** Huvudsyftet med detta avsnitt är att beräkna missilens acceleration och vinkelacceleration utgående från totala krafter och moment. För att uppnå detta kopplas de olika motorerna samman med övriga kroppen. Här har vi två motorer, principiellt skulle det kunna vara fler, eller färre. Utöver motordata behöver man de aerodynamiska krafterna och momenten.

$$\begin{aligned} I_{B0} &: \text{kroppens tröghetsmoment} & [kgm^2] \\ x_{cg_B} &: \text{kroppens masscentrum} & [m] \\ m_B &: \text{kroppens massa} & [kg] \end{aligned}$$

Total massa [kg] och massflöde [kg/s]:

$$m = m_B + m_{m1} + m_{m2} + \varepsilon_m \quad (3.39)$$

$$\dot{m} = \dot{m}_{m1} + \dot{m}_{m2} \quad (3.40)$$

Totalt masscentrum [m]:

$$X_{cg} = \frac{X_{cgB}m_B + X_{cgm1}m_{m1} + X_{cgm2}m_{m2}}{m} + \varepsilon_{cg} \quad (3.41)$$

Total masscentrumhastighet [m/s]:

$$\dot{X}_{cg} = \frac{X_{cgm1}\dot{m}_{m1} + X_{cgm2}\dot{m}_{m2} - X_{cg}\dot{m}}{\frac{m}{m}} + \frac{\dot{X}_{cgm1}m_{m1} + \dot{X}_{cgm2}m_{m2}}{m} + \dot{\varepsilon}_{cg} \quad (3.42)$$

Totalt tröghetsmoment och dess derivata [kgm<sup>2</sup>] [kgm<sup>2</sup>/s]:

$$I_0 = I_{B0} + I_{m1} + I_{m2} \quad (3.43)$$

$$I_{cg} = I_0 + m \begin{pmatrix} -(X_{cg_y}^2 + X_{cg_z}^2) & X_{cg_x}X_{cg_y} & X_{cg_x}X_{cg_z} \\ X_{cg_x}X_{cg_y} & -(X_{cg_x}^2 + X_{cg_z}^2) & X_{cg_y}X_{cg_z} \\ X_{cg_x}X_{cg_z} & X_{cg_y}X_{cg_z} & -(X_{cg_x}^2 + X_{cg_y}^2) \end{pmatrix} + \varepsilon_I \quad (3.44)$$

$$\begin{aligned} \dot{I}_{cg} &= \dot{I}_{m1} + \dot{I}_{m2} \\ &+ m \begin{pmatrix} -2(X_{cg_y}\dot{X}_{cg_y} + X_{cg_z}\dot{X}_{cg_z}) & X_{cg_x}\dot{X}_{cg_y} + X_{cg_y}\dot{X}_{cg_x} & X_{cg_x}\dot{X}_{cg_z} + X_{cg_z}\dot{X}_{cg_x} \\ X_{cg_x}\dot{X}_{cg_y} + X_{cg_y}\dot{X}_{cg_x} & -2(X_{cg_x}\dot{X}_{cg_x} + X_{cg_z}\dot{X}_{cg_z}) & X_{cg_y}\dot{X}_{cg_z} + X_{cg_z}\dot{X}_{cg_y} \\ X_{cg_x}\dot{X}_{cg_z} + X_{cg_z}\dot{X}_{cg_x} & X_{cg_y}\dot{X}_{cg_z} + X_{cg_z}\dot{X}_{cg_y} & -2(X_{cg_x}\dot{X}_{cg_x} + X_{cg_y}\dot{X}_{cg_y}) \end{pmatrix} \\ &+ \dot{m} \begin{pmatrix} -(X_{cg_y}^2 + X_{cg_z}^2) & X_{cg_x}X_{cg_y} & X_{cg_x}X_{cg_z} \\ X_{cg_x}X_{cg_y} & -(X_{cg_x}^2 + X_{cg_z}^2) & X_{cg_y}X_{cg_z} \\ X_{cg_x}X_{cg_z} & X_{cg_y}X_{cg_z} & -(X_{cg_x}^2 + X_{cg_y}^2) \end{pmatrix} \end{aligned} \quad (3.45)$$

Avstånd mellan masscentrum och dysa [m]:

$$\Delta_1 = x_{dys1} - x_{cg}$$

$$\Delta_2 = x_{dys2} - x_{cg}$$

Dysmoment [Nm]:

$$X_1 = \begin{pmatrix} \frac{r_{dys1}^2}{2} + \Delta_{1y}^2 + \Delta_{1z}^2 & -\Delta_{1x}\Delta_{1y} & -\Delta_{1x}\Delta_{1z} \\ -\Delta_{1x}\Delta_{1y} & \frac{r_{dys1}^2}{4} + \Delta_{1x}^2 + \Delta_{1z}^2 & -\Delta_{1y}\Delta_{1z} \\ -\Delta_{1x}\Delta_{1z} & -\Delta_{1y}\Delta_{1z} & \frac{r_{dys1}^2}{4} + \Delta_{1x}^2 + \Delta_{1y}^2 \end{pmatrix}$$

$X_2$  p s s som ovan.

$$M_{dys} = \begin{cases} -\dot{m}_1(X_1\omega + \Delta_1 \times V_{gas1}) - \dot{m}_2(X_2\omega + \Delta_2 \times V_{gas2}) & (a) \\ -\dot{m}_1X_1\omega + \Delta_1 \times F_{m1} - \dot{m}_2X_2\omega + \Delta_2 \times F_{m2} & (b) \end{cases} \quad (3.46)$$

Vinkelacceleration [rad/s<sup>2</sup>]:

$$\dot{\omega} = I_{cg}^{-1}(M_A - M_{dys} - \dot{I}_{cg}\omega - \omega \times (I_{cg}\omega)) \quad (3.47)$$

Acceleration i kroppsfast system [m/s<sup>2</sup>]:

$$a_b = \begin{cases} \frac{F_A - \dot{m}(V_{gas1} + V_{gas2} + 2\omega \times \Delta + V_{dys} - 2\dot{x}_{cg}) - \dot{m}\Delta}{m} + g_b & (a) \\ \frac{F_A + F_{m1} + F_{m2}}{m} + g_b & (b) \end{cases} \quad (3.48)$$

Acceleration i fix referensram [m/s<sup>2</sup>]:

(skall användas för att integrera fram hastigheten)

$$a_{fix} = a_b - \omega \times v \quad (3.49)$$

**3.2.6 Kinematik** Huvudsyftet med detta avsnitt är att räkna ut hastighet och läge i jordfast system (ground). Här räknar man ut hastighet i kroppsfast system och vinkelhastighet.

Vinkelhastighetsvektorn  $\omega$ 's komponenter kan betecknas  $(\omega_x, \omega_y, \omega_z)$  eller (p,q,r).

Hastighet i kroppsfast system [m/s]:

$$v_b = v_0 + \int a_{fix} dt \quad (3.50)$$

Vinkelhastighet [rad/s]:

$$\omega = \omega_0 + \int \dot{\omega} dt \quad (3.51)$$

För att transformera från kroppsfast till jordfast system skapas en transformationsmatris endera via quaternioner eller eulervinklar. Ekvationerna för båda förfaringssätten samt omvandling mellan dem finns beskrivna nedan.

Quaternioner [-]:

$$\epsilon = |q| - 1.0 \approx 0 \quad (3.52)$$

$$\dot{q} = \frac{1}{2} \begin{pmatrix} -\epsilon & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & -\epsilon & \omega_z & -\omega_y \\ \omega_y & -\omega_z & -\epsilon & \omega_x \\ \omega_z & \omega_y & -\omega_x & -\epsilon \end{pmatrix} q \quad (3.53)$$

$$q = q_0 + \int \dot{q} dt \quad (3.54)$$

Quaternioner till euler [rad]:

$$\begin{aligned} \phi &= \arctan \frac{2(q_3 q_4 + q_1 q_2)}{q_1^2 - q_2^2 - q_3^2 + q_4^2} \\ \theta &= -\arcsin(2(q_2 q_4 - q_1 q_3)) \\ \psi &= \arctan \frac{2(q_2 q_3 + q_1 q_4)}{q_1^2 + q_2^2 - q_3^2 - q_4^2} \end{aligned} \quad (3.55)$$

Euler till quaternioner [-]:

$$q = \begin{pmatrix} \cos \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \sin \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \sin \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} + \cos \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} \\ \sin \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} + \cos \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \end{pmatrix} \quad (3.56)$$

Eulervinklar och dessas derivator [rad] [rad/s]:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (3.57)$$

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} \phi_0 \\ \theta_0 \\ \psi_0 \end{pmatrix} + \int \begin{pmatrix} \dot{\phi} dt \\ \dot{\theta} dt \\ \dot{\psi} dt \end{pmatrix} \quad (3.58)$$

Koordinattransformationer [-]:

$$T_{b2g} = \begin{pmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{pmatrix} \quad (3.59)$$

$$T_{b2g} = \left\{ \begin{array}{l} s = \sin \\ c = \cos \end{array} \right\} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \quad (3.60)$$

Hastighet i jordfast system [m/s]:

$$v_g = T_{b2g}v_b \quad (3.61)$$

Lägeskoordinater i jordfast system [m]:

$$x = x_0 + \int v_g dt \quad (3.62)$$

**3.2.7 Sensorer** I detta avsnitt beskrivs ekvationerna för en accelerometer och ett gyro inklusive några vanliga störningar. Observera att ekvationerna bara gäller en sensor och därför måste upprepa om man t ex vill använda flera accelerometrar. En "ren" x-accelerometer har  $dir_{acc} = [1, 0, 0]$

$dir_{acc}$	: accelerometers mätning	[-]
$dir_{gyro}$	: gyrots mätning	[-]
$r_{acc}$	: accelerometers position	[m]

Accelerometer [m/s<sup>2</sup>]:

$$Acc_{acc} = \begin{pmatrix} (Acc_x - g_x)dir_{acc_x} \\ (Acc_y - g_y)dir_{acc_y} \\ (Acc_z - g_z)dir_{acc_z} \end{pmatrix} - (\dot{\omega} \times r_{acc} + \omega \times (\omega \times r_{acc})) - (2(\omega \times \dot{x}_{cg}) + \ddot{x}_{cg}) + Bias_{acc} + Noise_{acc} \quad (3.63)$$

Gyro [rad/s]:

$$\omega_{gyro} = \begin{pmatrix} \omega_x dir_{gyro_x} \\ \omega_y dir_{gyro_y} \\ \omega_z dir_{gyro_z} \end{pmatrix} + Bias_{gyro} + Noise_{gyro} \quad (3.64)$$

**3.2.8 Servo** I detta avsnitt beskrivs servot som en överföringsfunktion. Detta är en mycket vanlig förenkling som fungerar tillräckligt bra i de flesta applikationer. Vid behov kan modellen byggas ut till att innehålla olika begränsningar, glapp och störningar.

Kommenderade rodervinklar (x-konfig) [rad]:

$$\begin{aligned} \delta_{1k} &= \delta_a - \delta_r + \delta_e \\ \delta_{2k} &= \delta_a - \delta_r - \delta_e \\ \delta_{3k} &= \delta_a + \delta_r - \delta_e \\ \delta_{4k} &= \delta_a + \delta_r + \delta_e \end{aligned} \quad (3.65)$$

Utstyrd rodervinkel hos servot [rad]:

$$\delta \approx \frac{1}{1 + \frac{2\zeta_0}{\omega_0}s + \frac{1}{\omega_0^2}s^2} \delta_k \approx \frac{1}{1 + \frac{1}{\omega_0}s} \delta_k \approx \delta_k \quad (3.66)$$

**3.2.9 Inmätning** Även om det mest naturliga här är någon form av radarmålsökare så finns det många olika metoder att mäta in en missil t ex:

Målsökare:  
 Aktivradar  
 Passivradar  
 IR

Markinmätning:  
 Markradar  
 Laserinmätning  
 Bild  
 ...

I det följande antas ideala inmätare.

### 3.3 Förenklingar

Följande förenklingar görs i Mapleimplementationen beskriven i kapitel 5.

Roderservot utgår. Transformation mellan  $(\delta_1, \delta_2, \delta_3, \delta_4)$  och  $(\delta_r, \delta_e, \delta_a)$  utesluts.  
 Ingen dysstyrning.  
 Totala masscentrum ligger på x-axeln.  
 Dysornas angerppspunkt ligger på x-axeln

**3.3.1 Omgivning** Enligt kap ( 3.2.1).

**3.3.2 Aerodynamik** Ekvation ( 3.12) bortfaller.

Anmärkning 1:

För en icke symmetrisk robot som använder "bank to turn" gäller idealt att  $\beta=0$  därför sätts:

$$\begin{aligned} C_C &= 0 \\ C_{C\delta} &= 0 \\ C_n &= 0 \\ C_{n\delta} &= 0 \\ C_{n_r} &= 0 \end{aligned}$$

Anmärkning 2: Normalt definieras snedanblåsningsvinkeln  $\beta$  enl ekv 3.11 men för symmetriska missiler används ofta följande definition:  $\beta' = \arctan \frac{v_y}{v_x}$

Anmärkning 3:

Om vi har en symmetrisk robot och  $\beta'$  definieras enl ovan gäller följande:

$$\begin{aligned} C_N &= C_N(Mach, \alpha, \beta') = C_C(Mach, \beta', \alpha) \\ C_{N\delta} &= C_{N\delta}(Mach) = C_{C\delta}(Mach) \\ C_n &= C_n(Mach, \alpha, \beta') = - C_m(Mach, \beta', \alpha) \\ C_{n\delta} &= C_{n\delta}(Mach) = - C_{m\delta}(Mach) \\ C_{n_r} &= C_{n_r}(Mach) = C_{m_p}(Mach) \end{aligned}$$

**3.3.3 Motor1 (Krutmotor)** Beräkna  $\dot{m}$  med hjälp av en normerad tabell som anger drivkraftkurvas form. Denna tabell skalas till rätt nivå m h a specifikimpuls, krutets massa och brinntiden.

Masscentrum för krutet sammanfaller med den för "motorskalet" under hela brinntiden dvs  $X_{cg_k} = X_{cg_{m0}}$  och  $\dot{X}_{cg_k} = 0$ . Detta medför att:

$$\begin{aligned} \text{ekvation (3.21)} & \Rightarrow X_{cg_m} = X_{cg_{m0}} \\ \text{ekvation (3.22)} & \Rightarrow \dot{X}_{cg_m} = 0 \\ \text{ekvation (3.23)} & \Rightarrow I = I_{m1} + k_1 m_k + k_2 m_k^2 \approx I_{m1} + k_1 m_k \\ \text{ekvation (3.24)} & \Rightarrow \dot{I} = k_1 \dot{m} + 2k_2 m_k \dot{m} \approx k_1 \dot{m} \end{aligned}$$

I denna missil används roderstyrning och inte dysstyrning, dvs  $\delta_{dys_y} = \delta_{dys_z} = 0$  detta medför att:

$$\text{ekvation (3.25)} \Rightarrow F = \begin{pmatrix} P_D \\ 0 \\ 0 \end{pmatrix}$$

**3.3.4 Motor2 (SOFRammotor)** Här är  $\dot{m}$  tabellerad.

**3.3.5 Body** Masscentrum ligger på robotens x-axel dvs  $X_{cg_y} = X_{cg_z} = 0$  detta ger:

$$\text{ekvation (3.44)} \Rightarrow I_{cg} = I_0 + m \begin{pmatrix} 0 & 0 & 0 \\ 0 & -X_{cg_x}^2 & 0 \\ 0 & 0 & -X_{cg_x}^2 \end{pmatrix} + \varepsilon_I$$

ekvation (3.45)  $\Rightarrow$

$$\begin{aligned} \dot{I}_{cg} = \dot{I}_m & + m \begin{pmatrix} 0 & 0 & 0 \\ 0 & -2X_{cg_x} \dot{X}_{cg_x} & 0 \\ 0 & 0 & -2X_{cg_x} \dot{X}_{cg_x} \end{pmatrix} + \\ & \dot{m} \begin{pmatrix} 0 & 0 & 0 \\ 0 & -X_{cg_x}^2 & 0 \\ 0 & 0 & -X_{cg_x}^2 \end{pmatrix} \end{aligned}$$

Dysornas angreppspunkt ligger på x-axeln dvs  $\Delta_y = \Delta_z = 0$ :

$$\text{ekvation (3.2.5)} \Rightarrow X = \begin{pmatrix} \frac{r_{dys}^2}{2} & 0 & 0 \\ 0 & \frac{r_{dys}^2}{4} + \Delta_x^2 & 0 \\ 0 & 0 & \frac{r_{dys}^2}{4} + \Delta_x^2 \end{pmatrix}$$

Ekvation (3.46) a och b är ekvivalenta varför man väljer den man känner sig mest konfortabel med.

Ekvation (3.48) b är något förenklad jämfört med a men i de flesta tillämpningar är b tillräckligt noggrann. I Mapleimplementationen beskriven i kapitel 5 används b..

**3.3.6 Kinematik** Man väljer ofta mellan att använda quaternioner eller euler-vinklar för att beräkna transformationsmatrisen body2ground. Quaternionerna ger ett tillstånd extra men saknar singulära punkter varför dessa bör användas.

Här använder vi alltså ekvation (3.52), (3.53), (3.54), (3.55), (3.59) samt (3.61) och (3.62)

Anmärkning 1. Ekvation (3.55) dvs beräkning av eulervinklar behöver inte göras, men man vill oftast ha en uppfattning om dem, varför de ändå bör beräknas.

Anmärkning 2. Eftersom man ändå räknar ut eulervinklarna kan man välja ekvation (3.60) i stället för (3.59) att räkna ut transformationsmatrisen.

**3.3.7 Sensorer** Använd ideala sensorer fast med brus och eventuell bias dvs Accelerometrar:

$$Acc_{acc_i} = Acc_i + Bias_{acc_i} + Noise_{acc_i}$$

Gyron:

$$\omega_{gyro_i} = \omega_i + Bias_{gyro_i} + Noise_{gyro_i}$$

**3.3.8 Servo** Ekvation (3.65) bortfaller.

Idealt servo används, dvs ekvation (3.66) blir  $\delta = \delta_k$

### 3.4 Indata till modellen

**3.4.1 Omgivning** Initieringsdata:

Start höjd	: bärande flygplans höjd = -z !	[m]
g	: 9.81	[m/s <sup>2</sup> ]
marktemp	: 15	[grader]

**3.4.2 Aerodynamik** Dessa data används till aerodynamiken:

$d$	: 0.203	[m]
$x_{ref}$	: 2.0	[m]
$C_{T_{\delta_r}}$	: 0.0	[-]
$C_{T_{\delta_e}}$	: 0.0	[-]
$C_{T_{\delta_a}}$	: 0.0	[-]
$C_C$	: 0.0	[-]
$C_{C_{\delta}}$	: 0.0	[-]
$C_n$	: 0.0	[-]
$C_{n_{\delta}}$	: 0.0	[-]
$C_{n_r}$	: 0.0	[-]
$C_{N_{\delta}}$	: 6	[-] (home made)
$C_{m_{\delta}}$	: -50	[-] (home made)
$C_{m_q}$	: -1000	[-] (home made)
$C_l$	: 0	[-] (home made)
$C_{l_{\delta}}$	: 6	[-] (home made)
$C_{l_p}$	: -20	[-] (home made)

$C_T \geq 1.2$  approximeras med följande uttryck:

$$C_T = 1.642 + 2.0 \cdot 10^{-6} Hojd - 0.808 Mach + 0.179 Mach^2 + 0.0143 Mach^3 \quad (3.67)$$

$C_N$  approximeras med följande uttryck:

$$C_N = 0.082 - 0.13 Mach + 0.019 Mach^2 + \alpha(0.65 - 0.15 Mach + 0.02 Mach^2) \quad (3.68)$$

$C_m$  approximeras med följande uttryck:

$$C_m = 0.11 + 0.06 Mach - 0.009 Mach^2 + \alpha(-3.52 + 1.89 Mach - 0.3 Mach^2) \quad (3.69)$$



Mach	höjd = 0	höjd = 11 km	home made
0.00	0.360	0.38	x
0.50	0.360	0.38	x
0.95	0.460	0.48	x
1.00	0.820	0.84	x
1.20	0.820	0.84	x
1.93	0.634	0.662	
2.15	0.572	0.595	
2.51	0.496	0.519	
3.06	0.409	0.432	
3.66	0.346	0.363	

Tabell 3.1:  $C_T$  [-]

Mach	$\alpha = 0$	$\alpha = 10$	home made
0.00	-0.0700	4.7000	x
1.50	-0.0700	4.7000	x
1.93	-0.0841	4.2712	
2.15	-0.1028	4.0914	
2.51	-0.1187	3.9269	
3.06	-0.1352	3.7751	
3.66	-0.1278	3.6028	

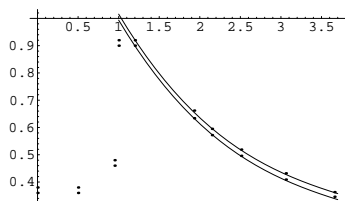
Tabell 3.2:  $C_N$  [-]

Mach	$\alpha = 0$	$\alpha = 10$	home made
0.00	0.18	-13.7	x
1.50	0.18	-13.7	x
1.93	0.18	-9.16	x
2.15	0.20	-7.75	x
2.51	0.20	-6.46	x
3.06	0.20	-5.49	x
3.66	0.20	-5.42	x

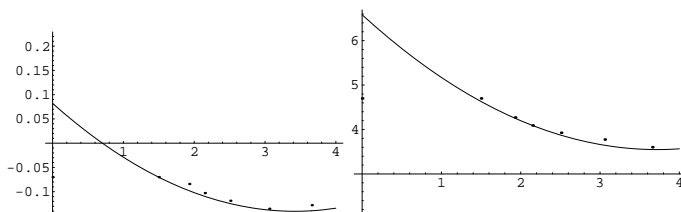
Tabell 3.3:  $C_m$  [-]

**3.4.3 Motor1 (Krutmotor)** Motorns massa skall initieras till skalets massa + bränslets massa d v s:

$$m_{m1}(0) = m_{m10} + m_{k0}$$



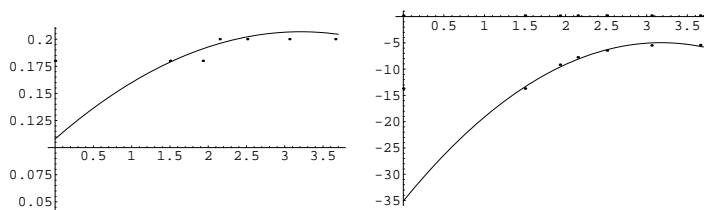
Figur 3.2:  $C_T(\text{Mach}, \text{höjd})$ , data enligt tabell 3.1 samt ekvation (3.67)



(a)  $C_N(\text{Mach}, \alpha=0)$ , data enligt tabell 3.2 samt ekvation (3.68)

(b)  $C_N(\text{Mach}, \alpha=10\text{grader})$ , data enligt tabell 3.2 samt ekvation (3.68)

Figur 3.3:  $C_N$  för två fall



(a)  $C_m(\text{Mach}, \alpha=0)$ , data enligt tabell 3.3 samt ekvation (3.69)

(b)  $C_m(\text{Mach}, \alpha=10\text{grader})$ , data enligt tabell 3.3 samt ekvation (3.69)

Figur 3.4:  $C_m$

Följande data gäller för boostern:

$m_{k0}$	: 53.0	[kg]	$m_{m10}$	: 9.5	[kg]
$I_{sp}$	: 2130.0	[Ns/kg]			
$t_0$	: 0.0	[s]	$t_b$	: 3.0	[s]
$P'_D(t)$	: $\begin{cases} 1 & t : [t_0..t_0 + t_b] \\ 0 & \text{annars} \end{cases}$	[–]			
$I(1,1)_{m10}$	: 0.07	[kgm <sup>2</sup> ]	$I(2,2)_{m20}$	: 180.0	[kgm <sup>2</sup> ]
$\rho_k$	: 1500.0	[kg/m <sup>3</sup> ]			
$L$	: 1.2	[m]			
$r$	: 0.09	[m]			
$x_{cg_{m10}}$	: (2.8, 0, 0)	[m]	$x_{cg_k}$	: (2.8, 0, 0)	[m]
$x_{dys1}$	: (3.5, 0, 0)	[m]	$r_{dys1}$	: 0.09	[m]

### 3.4.4 Motor2 (SOFRammotor) Följande förhållanden måste vara uppfyllda för

$$\text{att rammotorn skall fungera } \gamma = 1 \text{ endast om: } \begin{cases} Mach & : [2.12..4.00] & [-] \\ hojd & : [0..11000] & [m] \\ t & : [t_0 + t_b + 0.5..t(m_b = 0)] & [s] \\ \alpha & : [0..10] & [grader] \\ \beta & : [-1..1] & [grader] \end{cases}$$

Om motorn slocknat, kan den inte tändas igen, dvs  $\gamma$  kan inte slå om till 1 igen om den varit 1 någon gång tidigare.

Motorns massa skall initieras till skalets massa + bränslets massa d v s:

$$m_{m2}(0) = m_{m20} + m_{b0}$$

När motorn brunnit ut ger den ett tilläggsmotstånd:

$$C_{T_{end}} : -0.15 \quad [-]$$

Följande data gäller för motorn:

Mach	hojd = 0km				hojd = 11km			
	$\phi = 0.4$		$\phi = 1.0$		$\phi = 0.4$		$\phi = 1.0$	
	$\alpha = 0$	$\alpha = 10$	$\alpha = 0$	$\alpha = 10$	$\alpha = 0$	$\alpha = 10$	$\alpha = 0$	$\alpha = 10$
1.93	0.6829	1.1275	1.4655	2.0623	0.9109	1.3999	1.8396	2.5094
2.15	0.5726	0.9785	1.2627	1.8067	0.7742	1.2201	1.5956	2.2061
2.51	0.4237	0.7879	0.9935	1.4872	0.5913	0.9934	1.2739	1.8311
3.06	0.2587	0.5851	0.6956	1.1523	0.3892	0.7540	0.9189	1.4412
3.66	0.1390	0.4355	0.4711	0.9006	0.2381	0.5736	0.6509	1.1506

Tabell 3.4:  $P_D$  [N]

Mach	höjd = 0km		höjd = 11km	
	$\alpha = 0$	$\alpha = 10$	$\alpha = 0$	$\alpha = 10$
1.93	6.5861	7.8586	1.6960	2.0233
2.15	7.3186	8.7712	1.8844	2.2584
2.51	8.4752	10.3786	2.1822	2.6732
3.06	10.1915	13.1819	2.6241	3.3939
3.66	12.0142	16.7056	3.0934	4.3010

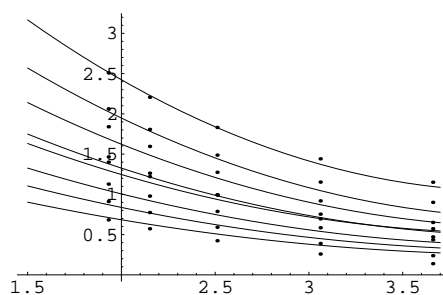
Tabell 3.5:  $\dot{m}$  [kg/s]

$P_D$  kan hjälpligt approximeras med följande uttryck:

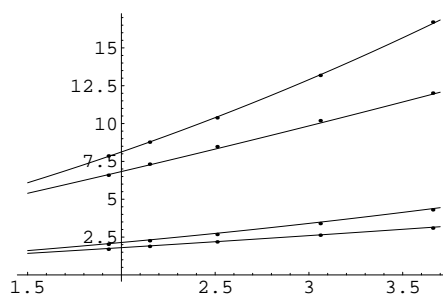
$$P_D = (8.1 + 0.38\alpha)(1.34 + 0.00003Hojd)(0.0525 - 0.0215Mach + 0.00255Mach^2)(1.2 + 5\phi) \quad (3.70)$$

$\dot{m}$  approximeras med följande uttryck:

$$\dot{m} = (1.9 - 0.01\alpha + 0.023\alpha Mach)(1.27 - 0.000085Hojd)(0.57 + 1.05Mach - 0.04Mach^2) \quad (3.71)$$



Figur 3.5:  $P_D(\text{Mach}, \text{höjd}, \phi, \alpha)$ , data enligt tabell 3.4 samt ekvation (3.70)



Figur 3.6:  $\dot{m}(\text{Mach}, \text{höjd}, \alpha)$ , data enligt tabell 3.5 samt ekvation (3.71)

$FuelAirRatio$	: 14.21	[—]
$m_{b0}$	: 17.5	[kg]
$m_{m20}$	: 10.0	[kg]
$X_{m2}$	: 1.4	[m]
$L$	: 0.700	[m]
$x_{cg_{m20}}$	: (1.6, 0, 0)	[m]
$x_{cg_k}$	: (1.6, 0, 0)	[m]
$x_{dys2}$	: (3.5, 0, 0)	[m]
$r_{dys2}$	: 0.09*	[m]
$I(1, 1)_{m20}$	: 0.07	[kgm <sup>2</sup> ]
$I(2, 2)_{m20}$	: 180.0	[kgm <sup>2</sup> ]
$r$	: 0.09	[m]
$\rho_b$	: 1500.0	[kg/m <sup>3</sup> ]

### 3.4.5 Body Följande data gäller kroppen (motorerna borträknade):

$m_B$	: 123.5	[kg]
$x_{cg_B}$	: (1.9, 0, 0)	[m]
$I(1, 1)_{B0}$	: 0.86	[kgm <sup>2</sup> ]
$I(2, 2)_{B0}$	: 187.0	[kgm <sup>2</sup> ]
$v_{dys}$	: 0	[m/s]

### 3.4.6 Dynamik Initieringsdata:

$v_0$  : bärande flygplans fart

$\omega_0$  : bärande flygplans vinkelhastighet

**3.4.7 Kinematik** Initieringsdata: $q_0$  : sätt in bärande flygplans eulervinklar i ekvation (3.56) $x_0$  : bärande flygplans position**3.4.8 Sensorer** Inga indata**3.4.9 Servo** Inga indata



## 4. Loop shaping

Loop shaping är en vanlig metod som används frekvent inom industrin för att designa styrsystem till missiler. (Det kan liknas vid ett bandpassfilter där man släpper igenom signaler inom det nyttiga frekvensområdet och filtrerar bort så mycket som möjligt av den signal som ligger utanför då detta kan betraktas som brus/störningar.)

Utgående från momentekvationerna kan man räkna ut överföringsfunktionen från rodervinkel till vinkelhastighet. Parametrarna i överföringsfunktionen tabelleras och används dels för att modellera en linjäriserad modell över missilen och dels för att "förkorta bort" de starkt varierande delarna så att man sedan kan klara sig med ett enklare styrsystem.

Vanligen tabelleras parametrarna som funktion av tid, höjd och fart, eller motsvarande. Eftersom aerodynamiken ofta är starkt beroende av  $\alpha$  och  $\beta$  så borde de egentligen ingå, men dessa vinklar är både svåra att mäta och osäkra att estimeras så tyvärr är de för osäkra för att användas som styrvariabler.

Utgående från missilens överföringsfunktion bestämmer man hur man skall forma styrfiltren för att totala systemets överföringsfunktion skall bli "så bra som möjligt". Här tittar man på Bode- och/eller Nicholsdiagram och bedömer fasmarginal, amplitudmarginal (tillräckligt stabil), undertryckning av amplitudtoppar i slutna looperna inom önskat frekvensområde (signaler med olika frekvens ger lika stort bidrag till totalsignalen) utan att få för låg förstärkning (dålig manövrerbarhet). Styrfiltrets frekvensspann skall läggas så att så mycket som möjligt av den nyttiga signalen släpps igenom samtidigt som så mycket som möjligt av bruset filtreras bort. Dessutom kan man behöva räkna brusintegraler <sup>1</sup> i kritiska punkter i looperna för att t ex inte ett servo skall överstyras.

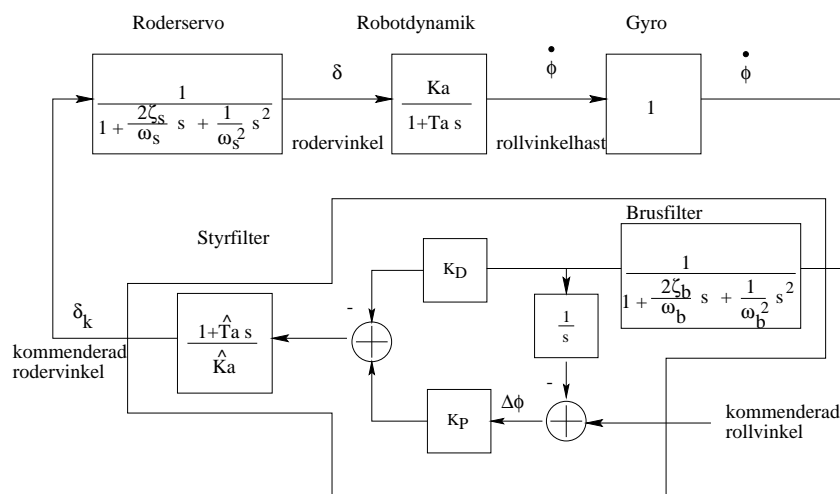
---

<sup>1</sup>skicka in vitt brus i systemet och räkna ut integralen av bruset inom aktuellt frekvensområde

#### 4.1 Linjäriserad rollodynamik

Utgående från momentekvationerna kan man räkna ut överföringsfunktionen från rodervinkel till rollvinkelhastighet, denna överföringsfunktion är för rollsystemet relativt enkel (första ordningens filter). Efter att man som del i styrsystemet har "förkortat bort" de starkt varierande delarna så kan man klara sig med ett enklare tidskonstant styrsystem.

I detta styrsystem styrs rollvinkeln till önskad referens, kommenderad rollvinkel, utan krav på rollvinkelhastigheten. I praktiken måste vinkelhastigheten begränsas. Styrsystemet dimensioneras så att missilen intar önskad rollvinkel så fort som möjligt utan att vinkelhastigheten överstiger specificerat värde, (Gyrot mätas vid någon hastighet), och utan att systemet blir instabilt (svängigt). Prestanda för systemet kan förbättras väsentligt genom att begränsa  $\Delta\phi$ , se Figure 4.1



Figur 4.1: Blockdiagram över rollsystemet inklusive styrfilter

Överföringsfunktion från kommenderat roder till rollvinkelhastighet:

$$\dot{\phi} = \frac{K_a}{1 + T_a s} \delta \quad (4.1)$$

Parametrarna i överföringsfunktionerna ovan är variabla och räknas ut på följande sätt:

$$K_a = -\frac{2VC_{l\delta}}{dC_{lp}} \quad (4.2)$$

$$T_a = -\frac{2VI_x}{qSd^2C_{lp}} \quad (4.3)$$

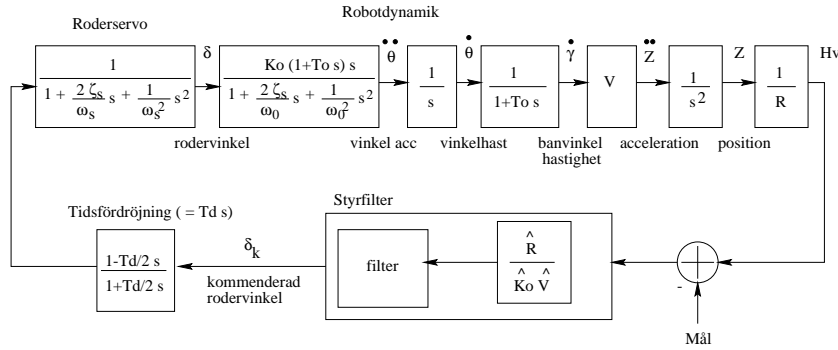


## 4.2 Linjäriserad tippdynamik

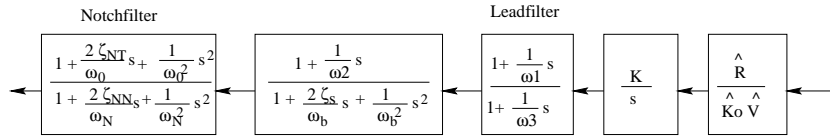
Utgående från momentekvationerna kan man räkna ut överföringsfunktionen från roder vinkel till tippvinkelhastighet.

I detta exempel är det höjdvinkeln som är mätsignalen. Här förkortas egenförstärkningen,  $K_0$ , samt hastighet och avstånd bort. Trots detta måste styrsystemfiltrena vara variabla, på samma sätt som parametrarna i dynamiken. Dvs alla  $K$ ,  $\omega$  och  $\zeta$  i Figur 4.3 varierar med tid, höjd och fart.

För att dämpa egensvängningarna kan man göra en gyroåterkoppling (via  $\dot{\theta}$ ) eller använda ett notchfilter <sup>2</sup>



Figur 4.2: Blockdiagram över linjäriserad tippdynamik



Figur 4.3: Blockdiagram över styrsystemet i tipp.  $\omega_1 \approx \omega_2 \ll \omega_3 \ll \omega_b$ ,  $\omega_N \approx \omega_0$ ,  $\zeta_{NT} > \zeta_0$ ,  $\zeta_{NN} \approx 10\zeta_{NT}$

Överföringsfunktion från roder till tippvinkelhastighet:

$$\dot{\theta} = \frac{K_0(1 + \tau_0 s)}{1 + \frac{2\zeta_0 s}{\omega_0} + \frac{s^2}{\omega_0^2}} \delta \quad (4.4)$$

Överföringsfunktion från tippvinkelhastighet till normalacceleration:

$$\ddot{z} = \frac{V}{1 + \tau_0 s} \dot{\theta} \quad (4.5)$$

Parametrarna i överföringsfunktionerna ovan är variabla och räknas ut på följande sätt:

Egenfrekvens:

$$\omega_0 = \sqrt{-\frac{qSd}{I_y} \left( C_{m\alpha} - \frac{\Delta x}{d} C_{N\alpha} \right) - \left( \frac{qSd^2}{2VI_y} C_{mq} - \frac{\dot{I}_y}{I_y} \right) \left( \frac{qS}{mV} C_{N\alpha} + \frac{P_D - qSC_T}{mV} \right)} \quad (4.6)$$

<sup>2</sup>Filter som bibehåller egenfrekvensen men ökar relativa dämpningen och därmed undertrycker resonansstoppen vid egenfrekvensen  $\omega_0$ .

Förstärkning:

$$K_0 = \frac{qSd}{I_y mV} \frac{C_{m\delta}(qSC_{N\alpha} + P_D - qSC_T) - qSC_{N\delta}(C_{m\alpha} - \frac{\Delta x}{d} C_{N\alpha})}{\omega_0^2} \quad (4.7)$$

Relativ dämpning

$$\zeta_0 = \frac{-\frac{qSd^2}{2I_y V} C_{mq} + \frac{i_y}{I_y} + \frac{qSC_{N\alpha} + P_D - qSC_T}{mV}}{2\omega_0} \quad (4.8)$$

Svängtidskonstant:

$$\tau_0 = -\frac{qSd}{I_y} \frac{C_{m\delta}}{K_0 \omega_0^2} \quad (4.9)$$

$C_{N\alpha}$  och  $C_{m\alpha}$  avser  $C_N$  och  $C_m$ 's derivata med avseende på  $\alpha$  d v s

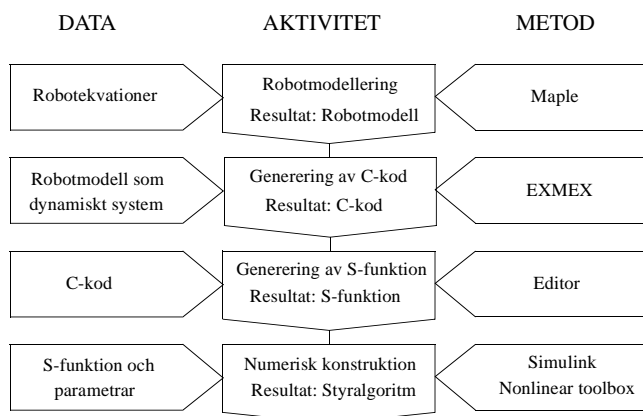
$$C_{N\alpha} = \frac{C_N(\alpha) - C_N(\alpha - \Delta\alpha)}{\Delta\alpha} \quad (4.10)$$

och

$$C_{m\alpha} = \frac{C_m(\alpha) - C_m(\alpha - \Delta\alpha)}{\Delta\alpha} \quad (4.11)$$

## 5. Beräkningsprocess av robotmodell

Process för implementation och analys av robotmodellen kan betraktas i Fig. 5.1 och kommer att utförligare beskrivas i följande avsnitt.



Figur 5.1: Beräkningsprocess av robotmodell

### 5.1 Implementation av robotmodell

Ekvationerna för robotmodellen implementeras i ett symboliskt beräkningsprogram. I detta projekt har beräkningsprogrammet Maple använts för att hyfsa modellen och transformera den till ett dynamiskt system på standardform, se [11]. Maple är ett symboliskt eller algebraiskt beräkningssystem som används för att formulera, lösa och undersöka matematiska modeller samt symboliskt manipulera dessa. Genom att använda Maple har robotmodellen blivit mer generell och lättöverskådlig samt enklare att implementera, manipulera, ändra och utvidga än om modellen direkt kodats till exempelvis C- eller Fortrankod.

Specifika indata för robotmodellen anges som parametrar och aerodata för modellen är tabellerade mätvärden, från vilka approximativa polynom bestäms, se kapitel 3.4.2.

De för modellen beräknade tillståndsvariablerna,  $X$ , presenteras i Ekv. (5.1).

$$X = ( x \quad y \quad z \quad q_0 \quad q_1 \quad q_2 \quad q_3 \quad \omega_x \quad \omega_y \quad \omega_z )^T \quad (5.1)$$

### 5.2 Ekvationsgenerering av C-kod

I beräkningsprocessen för robotmodellen används exmex [9], för att generera C-kod för robotmodellens ekvationer vilka är implementerade i Maple.

Programmet exmex är en kodgenereringsprocedur som är implementerad som ett Maplepaket för att kunna generera effektiv numerisk C-kod för system av ordinära differentialekvationer. Den kod som genereras av exmex är konstruerad för att länkas

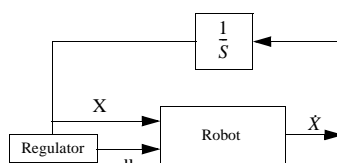
som externa funktioner till MATLAB och anropas på samma sätt som andra funktioner i MATLAB-miljön. Funktioner för anrop från Simulink kan också skrivas av exmex.

### 5.3 Generering av S-funktion

Utifrån den genererade C-koden för robotmodellen konstrueras en S-funktion, vilken skrivs i C-kod för det dynamiska systemet i syfte att konstruera egna funktioner i Simulink. S-funktionen används vid konstruktion av blockschema i Simulink och gör det möjligt att använda Simulink som analysverktyg.

### 5.4 Numerisk modellanalys och resultat

Ett blockschema för modellen konstrueras i Simulink tillsammans med ett icke-linjärt syntesverktyg (Nonlinear Synthesis Tools) vilken möjliggör en relativt snabb konstruktion av icke-linjära styrprogram. Blockschemat i Simulink ser principiellt ut enligt Fig. 5.2. Konstruerad styrprogram implementeras i robotmodellen och testas.



Figur 5.2: Blockschema i Simulink för numerisk konstruktion av styrprogram

### 5.5 MATLAB

MATLAB (Matrix Laboratory) är ett interaktivt program för numerisk beräkning och datavisualisering, se [13]. MATLAB är byggt på sofistikerad matrismanipulation för att analysera linjära ekvationssystem.

### 5.6 Simulink

Simulink är en utvidgning av MATLAB och används för att modellera, simulera och analysera fysikaliska och matematiska system och använder blockdiagram för modellkonstruktion samt innehåller bibliotek med linjära och icke-linjära komponenter, se [12]. modellsimulering sker numerskt och kan postprocessas.

### 5.7 Nonlinear Synthesis Tools

Nonlinear Synthesis Tools består av MATLAB-funktioner som möjliggör snabb konstruktion av icke-linjära styrprogram, se [15]. Denna användes tillsammans med Simulink för att konstruera styrprogram till robotmodellen samt för att utvärdera de olika icke-linjära metoderna i toolboxen.

## 6. Utvärdering av Nonlinear Synthesis Tools

Inom ramen för projektet Optimering av robotprestanda har ett programpaket avsett för design av icke linjära regulatorer utvärderats.

### 6.1 Programvaran

Nonlinear Synthesis Tools (NST) är ett program utvecklat för design av icke linjära regulatorer för dynamiska system. NST är ett tillägg (en s.k. toolbox) utvecklat av företaget Optimal Synthesis Inc ([www.optisyn.com](http://www.optisyn.com)) till simuleringsmiljön MATLAB/Simulink ([www.mathworks.com](http://www.mathworks.com)). Programpaketet är beskrivet i detalj i [15]. NST bygger på principen att den designade regulatorn numeriskt utvärdera det givna systemet och utifrån detta beräkna en lämplig styrsignal. Resultatet av en syntes med hjälp av NST är alltså inte en explicit styrlag på sluten form, utan en numerisk algoritm som beror på den valda design metodiken. NST tillhandahåller fem olika designmetodiker som sinsemellan bygger på olika reglerprinciper:

- Quickest Descent Method
- Predictive Control
- State-Dependent Riccati Equation Method
- Recursive Backstepping Method
- Feedback Linearization Method

De olika designmetodikerna är beskrivna mera i detalj i referens [19], [10], [2], [18] och [7].

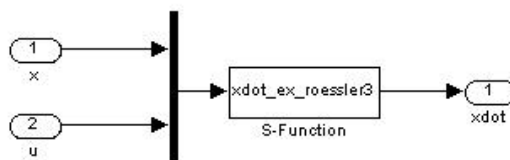
Ett givet dynamiskt system som skall regleras måste vara modellerat matematiskt på en given form för att NST skall kunna användas:

$$\dot{x} = f(t, x) + g(t, x) u \quad (6.1)$$

Som synes måste systemet vara linjärt i insignalen  $u$ , dvs. systemet skall vara affint. Den matematiska modellen av systemet implementeras sedan som ett blockdiagram i Simulink, alternativt som en C- eller FORTRAN mex-funktion. I Figur 6.1 ses implementationen av ett typiskt system.

Givet att systemet är modellerat på den form som NST föreskriver så kommer designprocessen att bestå av följande steg:

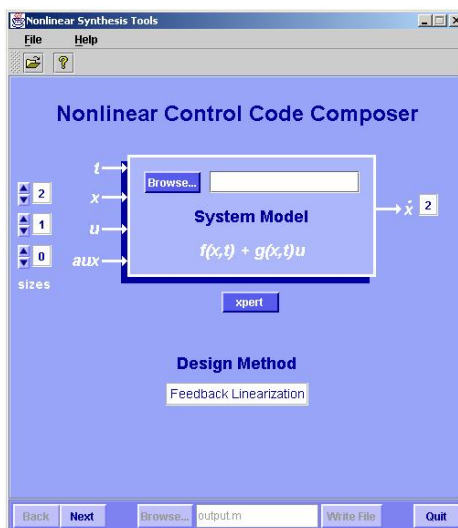
- Val av designmetod. Beroende på det specifika problem som är för handen så kommer de (fem) olika designmetoderna att resultera i olika bra regulatorer. Det som avgör valet av designmetod är i huvudsak problemets struktur, dvs. utseendet på differentialekvationerna.
- Använd det grafiska gränssnittet för att skapa en MATLAB m-fil som implementerar den valda designmetodiken i modellen. Här specificeras vilket system som skall regleras samt antalet in- och utsignaler. Gränssnittet visas i Figur 6.2, där metoden Feedback Linearization vald, samt att det ospecificerade systemet



Figur 6.1: Simulinkmodell av Rösslers system.

har två tillstånd och en insignal. I gränssnittet måste man även sätta olika designparametrar, vilka beror på den valda designmetoden.

- Den resulterande m-filen, här `output.m`, implementeras i en Simulinkmodell där den gjorda designen kan utvärderas. Ett exempel på en sådan modell kan ses i Figur 6.3.

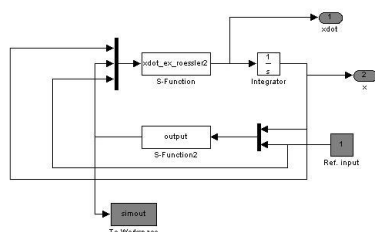


Figur 6.2: Gränssnitt till NST.

## 6.2 Testfall

NST innehåller ett antal olika demonstrationssystem som på ett mycket föredömligt sätt lotsar en ny användare igenom programmet. De olika designfallen belyser både olika designmetodiker samt principiellt olika typer av system. För att till fullo kunna utvärdera programmet skapades dock ett alldeles eget dynamiskt system, för att kunna testa hela kedjan ifrån implementering, via design till utvärdering av den gjorda designen.

I syfte att få ett enkelt, men samtidigt ett utmanande dynamiskt system, valdes ett Rössler-system. Systemet beskrivs av tre ordinära ickelinjära differentialekvationer,



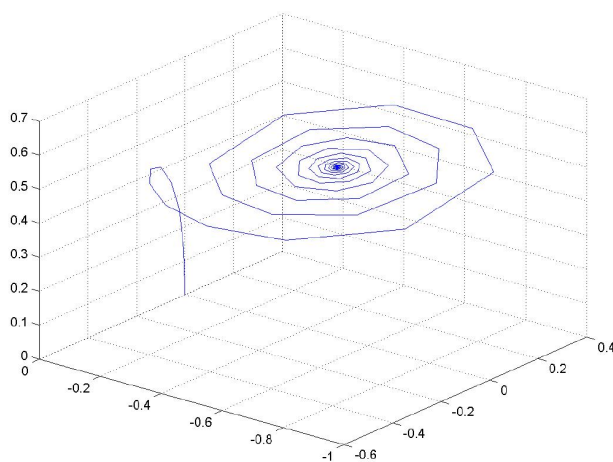
Figur 6.3: Modell av Rösslers system med återkoppling.

se ekvation (6.2).

$$\begin{cases} \dot{x} = -y - z \\ \dot{y} = x + ay \\ \dot{z} = b + (x - c)z \end{cases} \quad (6.2)$$

Om parametrarna  $a$ ,  $b$  och  $c$  väljs på ett lämpligt sätt, (6.3), så kommer systemet att beskriva oscillationer kring en jämviktpunkt.

$$\begin{cases} a = 0 \\ b = 2 \\ c = 4 \end{cases} \quad (6.3)$$



Figur 6.4: Simulering av det öppna systemet.

Systemet har dock ingen insignal, och kommer därmed inte att vara styrbart. Systemet har därför augmentersats med en insignal  $u$ , samt ytterligare en tillståndsvariabel  $e$  där referenssignalen introduceras. Det resulterande systemet, med in- respektive

referenssignal är givet i ekvation (6.4). Trajektorian för systemet kan ses i figur 4.

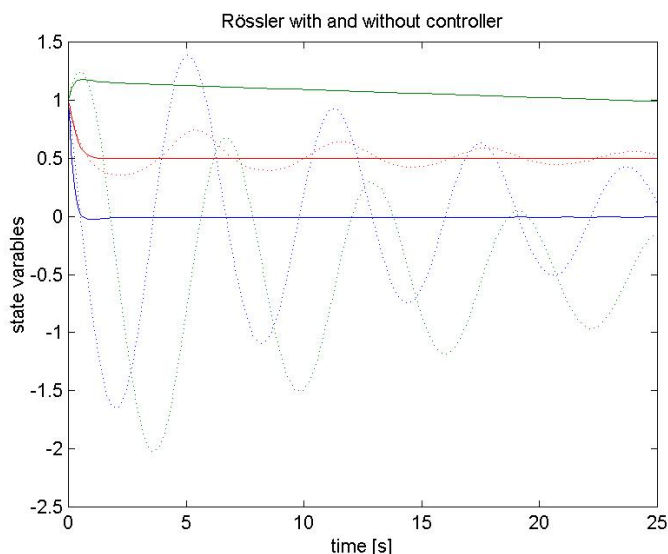
$$\begin{cases} \dot{x} = -y - z + u \\ \dot{y} = x \\ \dot{z} = 2 + (x - 4)z \\ e = x_c - x \end{cases} \quad (6.4)$$

Målet för designen valdes till att med hjälp av designmetoden Recursive Backstepping konstruera en regulator som driver tillståndet till jämviktspunkten. Resultatet kan ses i Figur 6.5, där man kan notera att regulatordesignen lyckades för de första två tillståndsvariablerna  $x$  och  $y$  men fallerade för det tredje,  $z$ . Orsaken till detta är att den valda metodiken kräver att systemet är på s.k. triangulär form, vilket inte för det aktuella systemet är fallet. Något ytterligare försök med någon annan, bättre lämpad, designmetodik har inte gjorts.

### 6.3 Resultat

Designprocessen är tämligen automatiserad och rättfram och det grafiska gränssnittet ger ett bra stöd. Dokumentationen samt de medföljande exemplen ger också en utmärkt introduktion till programmet och designprocessen. Dock krävs, som alltid, en relativt god insikt i de olika för- och nackdelar som de olika designmetoderna har för att kunna göra en lyckad design för ett specifikt system.

Några slutsatser om hur användbart programmet är för reglerdesign för mera komplexa system kan inte dras baserat på denna studie. Dock verkar programvaran definitivt vara lämpligt för konceptstudier.



Figur 6.5: Rösslers system med och utan återkoppling.



## Litteraturförteckning

- [1] M. Bodson and S. Sastry. *Adaptive Control, Stability Convergence and Robustness*. Prencice Hall, 1989.
- [2] J. Cloutier. *State-Dependent Riccati Equation Techniques: An Overview*. ACC, Albuquerque, 1997.
- [3] H. Elmquist and S. E. Mattsson. Introduction to the physical modeling language modelica. In *Proc of the 9th European Simulation Society*, 1997.
- [4] J. Hamberg. General matching conditions in the theory of controlled lagrangians. In *Proc. of IEEE CDC*, volume 38, pages 2519–2523, 1999.
- [5] O. Härkegård. Flight control design using backstepping. Technical Report LiU-TEK-LIC-2001:12, LiU, 2001.
- [6] A. Helmersson. *Methods for Robust Gain Scheduling*. PhD thesis, LiU, 1995.
- [7] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, 1989.
- [8] M. Krstic, I. Kanellakopoulos, and P. Kokotovic. *Nonlinear and Adaptive Control Design*. John Wiley & Sons, 1995.
- [9] A. Lennartsson. *Efficient Multibody Dynamics*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 1999.
- [10] W. Levine. *The Control Handbook*. CRC Press, 1996.
- [11] Maplesoft, Inc. <http://www.maplesoft.com>.
- [12] The Mathworks, Inc. <http://www.mathworks.com>.
- [13] The MathWorks, Inc., Natick, Massachusetts. *MATLAB External Interface Guide*.
- [14] R. Molin. A tool for LPV and LFT modelling. Technical Report FOA-R-99-01295-517-SE, Försvarets Forskningsanstalt, FOA, FOI, SE-17290 Stockholm, 1999.
- [15] Optimal Synthesis, Inc., Palo Alto, CA. *Nonlinear Synthesis Tools*, 2000. [www.optisyn.com](http://www.optisyn.com).
- [16] R. Ortega, A. Loria, P. J. Nicklasson, and H. Sira-Ramirez. *Passivity-based Control of Euler-Lagrange Systems*. Springer, 1998.
- [17] J. W. Polderman and J. C. Willems. *Introduction Mathematical Systems Theory A Behavioural Approach*. Springer, 1998.
- [18] R. Sepulchre, M. Jankovic, and P. Kokotovic. *Constructive Nonlinear Control*. Springer-Verlag, 1997.
- [19] T. Vincent and W. Gantham. *Nonlinear and optimal Control Systems*. Wiley & Sons, 1997.
- [20] K. Zhou, J. C. Doyle, and K. Glover. *Robust and optimal control*. Prentice Hall, 1996.