

# **Modellering og styring af mobile robotter**

Dina Friesel

Kongens Lyngby 2007  
IMM-PHD-2007-70

Technical University of Denmark  
Informatics and Mathematical Modelling  
Building 321, DK-2800 Kongens Lyngby, Denmark  
Phone +45 45253351, Fax +45 45882673  
[reception@imm.dtu.dk](mailto:reception@imm.dtu.dk)  
[www.imm.dtu.dk](http://www.imm.dtu.dk)

IMM-PHD: ISSN 0909-3192

# Summary

---

This thesis describes the development of mathematical models and controllers for mobile robots. Several mathematical models are made, all of them in continuous time. For each model a LQR-controller has been designed. The models with controllers have been tested on several examples and the simulation results have been compared. An analysis of the simulation results is done.



# Resumé

---

Denne rapport er dokumentation for udviklingen af matematiske modeller og reguleringsmetoder for den mobile robot. Der opstilles forskellige matematiske modeller, der alle er i kontinuert tid. Til hver af de opstillede modeller bruges LINEAR QUADRATIC REGULATOR, som også er i kontinuert tid. Reguleringerne afprøves på forskellige eksempler og metoderne sammenlignes. Rapporten afsluttes med en konklusion, hvor resultaterne diskuteres.



# Indhold

---

Summary	i
Resumé	iii
Symbolliste	vii
Figurer	ix
Tabeller	1
<b>1 Indledning</b>	<b>1</b>
<b>2 Matematiske modeller</b>	<b>3</b>
2.1 Det kartesiske koordinatsystem . . . . .	4
2.1.1 Model 1 . . . . .	6
2.1.2 Model 2 . . . . .	7
2.1.3 Model 3 . . . . .	8

2.1.4	Det lokale koordinatsystem . . . . .	11
2.2	Det polære koordinatsystem . . . . .	12
<b>3</b>	<b>Simuleringsplatform</b>	<b>15</b>
<b>4</b>	<b>Regulering og styring</b>	<b>19</b>
4.1	Styring til x-aksen uden hastighedsregulering . . . . .	21
4.2	Styring til x-aksen med hastighedsregulering . . . . .	30
4.3	Waypointstyring . . . . .	35
4.3.1	Waypointstyring uden hastighedsregulering . . . . .	36
4.3.2	Waypointstyring med hastighedsregulering . . . . .	38
4.4	Liniestyling . . . . .	40
4.5	Metodesammenligning . . . . .	43
<b>5</b>	<b>Konklusion</b>	<b>55</b>
	<b>Litteratur</b>	<b>57</b>
<b>A</b>	<b>Grafer</b>	<b>59</b>
A.1	Styring til x-aksen uden hastighedsregulering . . . . .	59
A.1.1	Parametertuning for $y > 0$ . . . . .	59
A.1.2	Parametertuning for $y < 0$ . . . . .	66
A.2	Styring til x-aksen med hastighedsregulering . . . . .	73
A.2.1	variation af $q_1$ . . . . .	74
A.2.2	variation af $q_2$ . . . . .	76



---

A.2.3	variation af r1	80
A.2.4	variation af r2	82
A.2.5	2. variation af q1	85
A.2.6	2. variation af q2	89
A.2.7	2. variation af r1	93
<b>B</b>	<b>Matlab-kode</b>	<b>97</b>
B.1	YTHcontrol.m	97
B.2	hastighedkontrol.m	98
B.3	THwaypointkontrol.m	99
B.4	liniestyringkontrol.m	100
B.4.1	koordinatskift.m	100
B.5	liniestyringkontrol2.m	101
B.6	Diagram over MATLAB-filerne	102



# Symbolliste

---

## Robotten

$v$	robottens fremadrettede hastighed; [ $m/s$ ]
$\omega$	robottens omdrejningshastighed; [ $rad/s$ ]
$v_l$	hastigheden på robottens venstre hjul; [ $m/s$ ]
$v_r$	hastigheden på robottens højre hjul; [ $m/s$ ]
$b$	afstanden mellem robottens hjul; [ $m$ ]
$\theta$	vinklen robotten danner med x-aksen; [ $rad$ ]

## Det kartesiske koordinatsystem

$(x, y)$	robottens position i det kartesiske koordinatsystem; [ $m$ ]
$\underline{u}$	inputvektor
$\underline{p}$	outputvektor
$\underline{q}$	tilstandsvektor
$\underline{\underline{A}}$	systemmatrix
$\underline{\underline{B}}$	inputmatrix

**Det lokale koordinatsystem**

$\underline{x}_g, \underline{y}_g$	akserne i det globale koordinatsystem
$\underline{x}_l, \underline{y}_l$	akserne i det lokale koordinatsystem
${}^lM_g$	basisskifte matricen for transformation fra globalt til lokalt
$x_{ref}$	lokale koordinatsystems nulpunkt som global x-koordinat; [m]
$x_0$	robottens position som global x-koordinat; [m]
$x_{0l}$	robottens position som lokal x-koordinat; [m]
$y_{ref}$	lokale koordinatsystems nulpunkt som global y-koordinat; [m]
$y_0$	robottens position som global y-koordinat; [m]
$y_{0l}$	robottens position som lokal y-koordinat; [m]
$\theta_{ref}$	lokale koordinatsystems vinkel i forhold til det globale koordinatsystem; [rad]
$\theta_0$	robottens vinkel i forhold til det globale koordinatsystem; [rad]
$\theta_{0l}$	robottens vinkel i forhold til det lokale koordinatsystem; [rad]

**Det polære koordinatsystem**

$\rho$	afstanden fra robotten til referencepunktet; [m]
$\alpha$	vinkelforskellen mellem robottens vinkel og vinklen for liniestykket mellem robotten og referencepunktet; [rad]
$\beta$	vinklen på den linie, robotten skal køre; [rad]

# Figurer

---

2.1	<i>Robotten i det kartesiske koordinatsystem</i> . . . . .	4
2.2	<i>Bodeplot for model 1: <math>G_1(s) = \frac{\theta(s)}{\omega(s)}</math></i> . . . . .	6
2.3	<i>Bodeplot for model 2: <math>G_{21}(s) = \frac{y(s)}{\omega(s)}</math>; <math>G_{22}(s) = \frac{\theta(s)}{\omega(s)}</math></i> . . . . .	7
2.4	<i>Bodeplot for model 3: <math>Gx_{31}(s) = \frac{x(s)}{v(s)}</math>; <math>Gx_{32}(s) = \frac{x(s)}{\omega(s)}</math></i> . . . . .	8
2.5	<i>Bodeplot for model 3: <math>Gy_{31}(s) = \frac{y(s)}{v(s)}</math>; <math>Gy_{32}(s) = \frac{y(s)}{\omega(s)}</math></i> . . . . .	9
2.6	<i>Bodeplot for model 3: <math>G\theta_{31}(s) = \frac{\theta(s)}{v(s)}</math>; <math>G\theta_{32}(s) = \frac{\theta(s)}{\omega(s)}</math></i> . . . . .	10
2.7	<i>Lokalt koordinatsystem i det globale koordinatsystem</i> . . . . .	11
2.8	<i>Robotten i det polære koordinatsystem</i> . . . . .	13
3.1	<i>Simulink model for robotens bane</i> . . . . .	16
3.2	<i>Den teoretiske bane og den simulerede bane med start i: <math>(x, y) = (0, 0)</math>; <math>\theta = 0</math>; <math>v_l = 0.3 \text{ m/s}</math>; <math>v_r = 0.5 \text{ m/s}</math></i> . . . . .	17
3.3	<i>Robotens bane, hvor <math>v_l</math> reduceres og efterfølgende reduceres <math>v_r</math></i> . . .	18

4.1	<i>Simulink model for system med tilbagekobling</i> . . . . .	20
4.2	<i>Robottens bane for variation af <math>q</math> og <math>r = 0.8</math></i> . . . . .	22
4.3	<i><math>y</math> som funktion af tiden for variation af <math>q</math> og <math>r = 0.8</math></i> . . . . .	23
4.4	<i>Robottens bane for variation af <math>r</math> og <math>q=0.6</math></i> . . . . .	23
4.5	<i><math>y</math>-værdien i forhold til tiden for variation af <math>r</math> og <math>q=0.6</math></i> . . . . .	24
4.6	<i>Robottens bane for forskellige startvinkler mellem <math>]-\pi, \pi]</math></i> . . . . .	25
4.7	<i>Robottens bane for forskellige startværdier for <math>y_0, x_0 = 2</math> og <math>\theta_0 = \pi</math></i> . . . . .	25
4.8	<i>Robottens bane for variation af <math>r_v</math></i> . . . . .	26
4.9	<i>Hastigheden på venstre hjul som funktion af tiden for variation af <math>r_v</math></i> . . . . .	27
4.10	<i>Hastigheden på højre hjul som funktion af tiden for variation af <math>r_v</math></i> . . . . .	27
4.11	<i>Robottens bane for start på <math>x</math>-aksen med vinklen <math>\theta_{ref} = \pi/2</math> for reference vinkel på både <math>0</math> og <math>2\pi</math></i> . . . . .	28
4.12	<i><math>y</math>-værdien som funktion af tiden for start på <math>x</math>-aksen med vinklen <math>\pi/2</math> for reference vinkel på både <math>0</math> og <math>2\pi</math></i> . . . . .	29
4.13	<i>Robottens bane for startposition i <math>(2,7)</math> med vinklen <math>\pi/2</math>, hvor der først kun styres på vinklen og herefter (<math>y &lt; 3.5</math>) på <math>y</math> og vinklen</i> . . . . .	29
4.14	<i>Variation af <math>q_1, q_2 = 0.6, r_1=0.6</math> og <math>r_2=0.8</math></i> . . . . .	31
4.15	<i>Robottens bane for variation af <math>q_2, q_1 = 0.2, r_1=0.6</math> og <math>r_2=0.8</math></i> . . . . .	32
4.16	<i>Variation af <math>r_1, q_1 = 0.2, q_2=1</math> og <math>r_2=0.8</math></i> . . . . .	33
4.17	<i>variation af <math>r_2, q_1=0.2, q_2=1</math> og <math>r_1=0.6</math></i> . . . . .	34
4.18	<i>Robottens bane for startpunktet <math>(-7,3)</math> med vinklen <math>\pi/4</math></i> . . . . .	35
4.19	<i>Vinkelstyring for 3 punkter</i> . . . . .	37
4.20	<i>Vinkelstyring for 3 punkter</i> . . . . .	38

4.21	<i>Simulink-model for det polære koordinatsystem</i>	39
4.22	<i>Waypointstyring: Startpunkt i (2,2) med vinklen <math>\theta_0 = 0</math>, slutpunkt i (5,7) med vinklen <math>\theta = 0</math></i>	39
4.23	<i>Simulink model, hvor der er indsat koordinatskift, så koordinaterne transformeres fra globalt til lokalt koordinatsystem</i>	41
4.24	<i>Robottens bane for startpunktet (2,3) med vinklen <math>\pi/4</math> og referencepunktet (10,9) med vinklen <math>\pi/8</math></i>	42
4.25	<i>Robottens bane med startpunktet (2,3) med vinklen <math>\pi/4</math> for liniestyring til y-aksen</i>	42
4.26	<i>Robottens bane for kørsel med liniestyring til tre referencelinier</i>	43
4.27	<i>Robottens bane for kørsel gennem en døråbning 1</i>	45
4.28	<i>Robottens bane for kørsel gennem en døråbning 2</i>	46
4.29	<i>Robottens bane for kørsel gennem en døråbning 3</i>	47
4.30	<i>Robottens bane for kørsel gennem en døråbning 4</i>	48
4.31	<i>Gennem gang med 4 referencepunkter</i>	50
4.32	<i>Gennem gang med 4 referencepunkter</i>	51
4.33	<i>Gennem gang med 4 referencepunkter</i>	52
A.1	<i>x som funktion af tiden for variation af q og r = 0.8</i>	60
A.2	<i><math>\omega</math> som funktion af tiden for variation af q og r = 0.8</i>	60
A.3	<i><math>v_l</math> som funktion af tiden for variation af q og r = 0.8</i>	61
A.4	<i><math>v_r</math> som funktion af tiden for variation af q og r = 0.8</i>	61
A.5	<i>x som funktion af tiden for variation af r og q = 0.6</i>	62
A.6	<i><math>\omega</math> som funktion af tiden for variation af r og q = 0.6</i>	62
A.7	<i><math>v_l</math> som funktion af tiden for variation af r og q = 0.6</i>	63

A.8	$v_r$ som funktion af tiden for variation af $r$ og $q = 0.6$ . . . . .	63
A.9	$x$ som funktion af tiden for variation af $r_v$ . . . . .	64
A.10	$y$ som funktion af tiden for variation af $r_v$ . . . . .	64
A.11	$\theta$ som funktion af tiden for variation af $r_v$ . . . . .	65
A.12	$\omega$ som funktion af tiden for variation af $r_v$ . . . . .	65
A.13	Robottens bane for variation af $q$ og $r = 0.8$ . . . . .	66
A.14	$x$ som funktion af tiden for variation af $q$ og $r = 0.8$ . . . . .	66
A.15	$y$ som funktion af tiden for variation af $q$ og $r = 0.8$ . . . . .	67
A.16	$\omega$ som funktion af tiden for variation af $q$ og $r = 0.8$ . . . . .	67
A.17	$v_t$ som funktion af tiden for variation af $q$ og $r = 0.8$ . . . . .	68
A.18	$v_r$ som funktion af tiden for variation af $q$ og $r = 0.8$ . . . . .	68
A.19	Robottens bane for variation af $r$ og $q = 0.6$ . . . . .	69
A.20	$x$ som funktion af tiden for variation af $r$ og $q = 0.6$ . . . . .	69
A.21	$y$ som funktion af tiden for variation af $r$ og $q = 0.6$ . . . . .	70
A.22	$\omega$ som funktion af tiden for variation af $r$ og $q = 0.6$ . . . . .	70
A.23	$v_t$ som funktion af tiden for variation af $r$ og $q = 0.6$ . . . . .	71
A.24	$v_r$ som funktion af tiden for variation af $r$ og $q = 0.6$ . . . . .	71
A.25	Robottens bane for start på $x$ -aksen med vinklen $\theta_0 = -\pi/2$ . . . . .	72
A.26	$y$ som funktion af tiden for start på $x$ -aksen med vinklen $\theta_0 = -\pi/2$ . . . . .	72
A.27	$\theta$ som funktion af tiden for variation af $q_1, q_2 = 0.6, r_1=0.6$ og $r_2=0.8$ . . . . .	74
A.28	$\omega$ som funktion af tiden for variation af $q_1, q_2 = 0.6, r_1=0.6$ og $r_2=0.8$ . . . . .	74



A.29 $v_l$ som funktion af tiden for variation af $q_1$ , $q_2 = 0.6$ , $r_1=0.6$ og $r_2=0.8$ . . . . .	75
A.30 $v_r$ som funktion af tiden for variation af $q_1$ , $q_2 = 0.6$ , $r_1=0.6$ og $r_2=0.8$ . . . . .	75
A.31 $x$ som funktion af tiden for variation af $q_2$ , $q_1 = 0.2$ , $r_1=0.6$ og $r_2=0.8$ . . . . .	76
A.32 $y$ som funktion af tiden for variation af $q_2$ , $q_1 = 0.2$ , $r_1=0.6$ og $r_2=0.8$ . . . . .	76
A.33 $\theta$ som funktion af tiden for variation af $q_2$ , $q_1 = 0.2$ , $r_1=0.6$ og $r_2=0.8$ . . . . .	77
A.34 $v$ som funktion af tiden for variation af $q_2$ , $q_1 = 0.2$ , $r_1=0.6$ og $r_2=0.8$ . . . . .	77
A.35 $\omega$ som funktion af tiden for variation af $q_2$ , $q_1 = 0.2$ , $r_1=0.6$ og $r_2=0.8$ . . . . .	78
A.36 $v_l$ som funktion af tiden for variation af $q_2$ , $q_1 = 0.2$ , $r_1=0.6$ og $r_2=0.8$ . . . . .	78
A.37 $v_r$ som funktion af tiden for variation af $q_2$ , $q_1 = 0.2$ , $r_1=0.6$ og $r_2=0.8$ . . . . .	79
A.38 $\theta$ som funktion af tiden for variation af $r_1$ , $q_1 = 0.2$ , $q_2=1$ og $r_2=0.8$ . . . . .	80
A.39 $\omega$ som funktion af tiden for variation af $r_1$ , $q_1 = 0.2$ , $q_2=1$ og $r_2=0.8$ . . . . .	80
A.40 $v_l$ som funktion af tiden for variation af $r_1$ , $q_1 = 0.2$ , $q_2=1$ og $r_2=0.8$ . . . . .	81
A.41 $v_r$ som funktion af tiden for variation af $r_1$ , $q_1 = 0.2$ , $q_2=1$ og $r_2=0.8$ . . . . .	81
A.42 $x$ som funktion af tiden for variation af $r_2$ , $q_1=0.2$ , $q_2=1$ og $r_1=0.6$	82
A.43 $\theta$ som funktion af tiden for variation af $r_2$ , $q_1=0.2$ , $q_2=1$ og $r_1=0.6$	82
A.44 $v$ som funktion af tiden for variation af $r_2$ , $q_1=0.2$ , $q_2=1$ og $r_1=0.6$	83

A.45 $\omega$ som funktion af tiden for variation af $r_2$ , $q_1=0.2$ , $q_2=1$ og $r_1=0.6$	83
A.46 $v_i$ som funktion af tiden for variation af $r_2$ , $q_1=0.2$ , $q_2=1$ og $r_1=0.6$	84
A.47 $v_r$ som funktion af tiden for variation af $r_2$ , $q_1=0.2$ , $q_2=1$ og $r_1=0.6$	84
A.48 Robotten bane for 2. variation af $q_1$ , $q_2=1$ , $r_1=0.6$ og $r_2=0.3$	85
A.49 $x$ som funktion af tiden for 2. variation af $q_1$ , $q_2=1$ , $r_1=0.6$ og $r_2=0.3$	85
A.50 $y$ som funktion af tiden for 2. variation af $q_1$ , $q_2=1$ , $r_1=0.6$ og $r_2=0.3$	86
A.51 $\theta$ som funktion af tiden for 2. variation af $q_1$ , $q_2=1$ , $r_1=0.6$ og $r_2=0.3$	86
A.52 $v$ som funktion af tiden for 2. variation af $q_1$ , $q_2=1$ , $r_1=0.6$ og $r_2=0.3$	87
A.53 $\omega$ som funktion af tiden for 2. variation af $q_1$ , $q_2=1$ , $r_1=0.6$ og $r_2=0.3$	87
A.54 $v_i$ som funktion af tiden for 2. variation af $q_1$ , $q_2=1$ , $r_1=0.6$ og $r_2=0.3$	88
A.55 $v_r$ som funktion af tiden for 2. variation af $q_1$ , $q_2=1$ , $r_1=0.6$ og $r_2=0.3$	88
A.56 Robotten bane for 2. variation af $q_2$ , $q_1=0.2$ , $r_1=0.6$ og $r_2=0.3$	89
A.57 $x$ som funktion af tiden for 2. variation af $q_2$ , $q_1=0.2$ , $r_1=0.6$ og $r_2=0.3$	89
A.58 $y$ som funktion af tiden for 2. variation af $q_2$ , $q_1=0.2$ , $r_1=0.6$ og $r_2=0.3$	90
A.59 $\theta$ som funktion af tiden for 2. variation af $q_2$ , $q_1=0.2$ , $r_1=0.6$ og $r_2=0.3$	90
A.60 $v$ som funktion af tiden for 2. variation af $q_2$ , $q_1=0.2$ , $r_1=0.6$ og $r_2=0.3$	91

A.61 $\omega$ som funktion af tiden for 2. variation af $q_2$ , $q_1=0.2$ , $r_1=0.6$ og $r_2=0.3$ . . . . .	91
A.62 $v_l$ som funktion af tiden for 2. variation af $q_2$ , $q_1=0.2$ , $r_1=0.6$ og $r_2=0.3$ . . . . .	92
A.63 $v_r$ som funktion af tiden for 2. variation af $q_2$ , $q_1=0.2$ , $r_1=0.6$ og $r_2=0.3$ . . . . .	92
A.64 Robotten bane for 2. variation af $r_1$ , $q_1=0.2$ , $q_2=1$ og $r_1=0.6$ . . .	93
A.65 $x$ som funktion af tiden for 2. variation af $q_1$ , $q_2=1$ , $r_1=0.6$ og $r_2=0.3$ . . . . .	93
A.66 $y$ som funktion af tiden for 2. variation af $r_1$ , $q_1=0.2$ , $q_2=1$ og $r_1=0.6$ . . . . .	94
A.67 $\theta$ som funktion af tiden for 2. variation af $r_1$ , $q_1=0.2$ , $q_2=1$ og $r_1=0.6$ . . . . .	94
A.68 $v$ som funktion af tiden for 2. variation af $r_1$ , $q_1=0.2$ , $q_2=1$ og $r_1=0.6$ . . . . .	95
A.69 $\omega$ som funktion af tiden for 2. variation af $r_1$ , $q_1=0.2$ , $q_2=1$ og $r_1=0.6$ . . . . .	95
A.70 $v_l$ som funktion af tiden for 2. variation af $r_1$ , $q_1=0.2$ , $q_2=1$ og $r_1=0.6$ . . . . .	96
A.71 $v_r$ som funktion af tiden for 2. variation af $r_1$ , $q_1=0.2$ , $q_2=1$ og $r_1=0.6$ . . . . .	96
B.1 Diagram over, hvordan de forskellige vedlagte MATLAB-filer bruger hinanden . . . . .	102



# Tabeller

---

4.1	<i>Fordele (+) og ulemper (÷) ved waypointstyring og liniestyring for de opstillede eksempler . . . . .</i>	53
-----	---	----



# Indledning

---

Anvendelsen af robotter er stadig mere udbredt. Overalt i industrien benyttes robotter, til at afløse den menneskelige arbejdskraft. Årsagen til dette kan være farligt arbejdsmiljø, trivielt arbejde, reducere af lønningsomkostninger eller lignende. Samtidig er kravene til robotterne blevet større og større. De skal kunne langt mere og være mere præcise. Dette stiller større krav til regulering og styring af robotterne. Derfor er det meget vigtigt, at man har en god matematisk model af robotten, før der beregnes en dertil egnet regulator.

Dette er et bachelorprojekt, skrevet på 6. semester, på civilbachelorlinjen Matematik og Teknologi. Projektet er skrevet hos Institut for Informatik og Matematisk Modellering (IMM) på DTU.

Dette projekt går ud på at lave en matematisk model af den mobile robots bane og efterfølgende at afprøve en række forskellige reguleringsalgoritmer til banestyring. Både de matematiske modeller og reguleringsalgoritmer efterprøves kun ved simulering, ikke i praksis. Til simulering bruges MATLAB og SIMULINK. De brugte programmer findes på den vedlagte CD og diagram ses i bilag B.6.

Der opstilles forskellige matematiske modeller. Jeg begynder med en simpel model og herefter udvides den gradvist til mere detaljerede modeller, disse er beskrevet i kartesiske og polære koordinater. Alle modellerne er opstillet i kontinuert tid, og alle reguleringsalgoritmer er også i kontinuert tid. De forskellige

metoder bliver analyseret i både tids- og i frekvensdomaine.

Modellerne er udgangspunktet til at afprøve forskellige reguleringsalgoritmer til styring af den mobile robot. Formålet med det er at finde den bedst egnede reguleringsmetode for at styre robotten ad en bestemt bane. Jeg kigger på hvor nøjagtigt robotten kører og hvor lang tid det tager robotten at gennemkøre banen.

Jeg antager i dette projekt, at robotten er ideel. Det betyder, at de dynamiske forhold for selve robotten ikke beskrives. Den mobile robot er drevet af to uafhængige motorer, der driver de to forreste hjul på robotten (robottens styrhjul). Det tredje hjul, baggest på robotten, har ingen betydning for robottens bevægelser og indgår ikke i den opstillede model. Robottens position er bestemt som midtpunktet mellem robottens styrehjul.

Rapporten afsluttes med en sammenligning af styringmetoderne og en konklusion, hvor resultaterne diskuteres.

MATLAB filerne der beregner reguleringsalgoritmerne findes i bilag [B](#). SIMULINK-modellerne for hver metode, er indsat som figur i metode-afsnittet. Alle grafer der er væsentlige for forståelsen er placeret i rapporten, og alle andre grafer fra simuleringerne er placeret i bilag [A](#).

Jeg takker min vejleder Niels Kjølstad Poulsen og Ph.D. studerende Sven Creutz Thomsen for deres hjælp og vejledning i hele projektforløbet.



## KAPITEL 2

# Matematiske modeller

---

I dette kapitel beskrives de grundlæggende matematiske modeller for den mobile robots bane. Selve robotten antages at være ideel.

Principskitse af den mobile robot ses på figur 2.1 side 4. De to hjul foran på robotten (styrehjul) drives ved hjælp af motorer. Hjulet bagpå, det såkaldte ”kastorhjul”, kan ikke styres. De to styrehjul har hastighederne  $v_r$  og  $v_l$ . Robotens fremadrettede hastighed  $v$  og rotationshastighed  $\omega$  er hhv.:

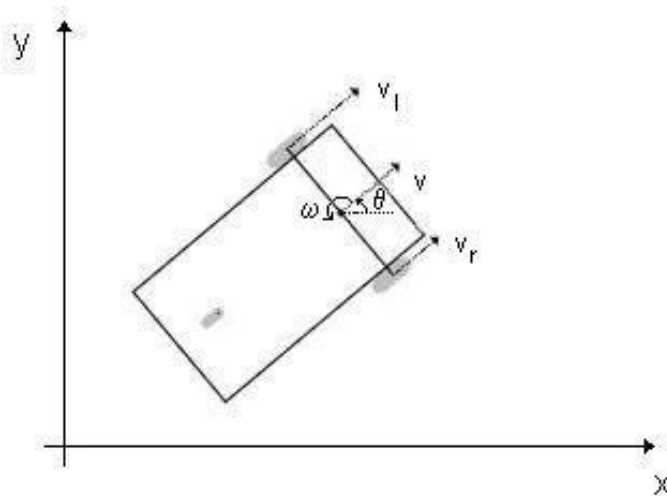
$$v = \frac{v_r + v_l}{2} \quad (2.1)$$

$$\omega = \frac{v_r - v_l}{b} \quad (2.2)$$

$b$  er afstanden i meter mellem de to styrehjul.

Fremover er hastighederne  $v$ ,  $v_r$  og  $v_l$  givet i  $m/s$  og rotationshastigheden  $\omega$  i  $rad/s$ .

Modelleringen af den mobile robot vil ske både i det kartesiske koordinatssystem (som ses på figur 2.1 side 4) og i det polære koordinatssystem (se afnit 2.2 side 12 og figur 2.8 side 13). Der udføres analyse i frekvensdomæne og tidsdomæne.



Figur 2.1: Robotten i det kartesiske koordinatsystem

## 2.1 Det kartesiske koordinatsystem

Det antages, at robotten er ideel, dvs. begge motorer er ideelle, og dermed kan robottens bane beskrives ved hjælp af følgende tre differentialligninger:

$$\dot{x} = v \cdot \cos(\theta) \quad (2.3)$$

$$\dot{y} = v \cdot \sin(\theta) \quad (2.4)$$

$$\dot{\theta} = \omega \quad (2.5)$$

hvor:

$(x, y)$  er robottens position

$\theta$  er den vinkel robotten danner med x-aksen

$x$  og  $y$  givet i meter og  $\theta$  i  $rad$ .

Differentialligningerne (2.3), (2.4) og (2.5), der beskriver robottens bane, skrives i form af tilstandsligninger:

$$\underline{\dot{q}} = \underline{A} \cdot \underline{q} + \underline{B} \cdot \underline{u} \quad (2.6)$$

$$\underline{p} = \underline{C} \cdot \underline{q} + \underline{D} \cdot \underline{u} \quad (2.7)$$

hvor  $\underline{q}$  er tilstandsvektoren,  $\underline{u}$  er inputvektor og  $\underline{p}$  er outputvektor.

I de modeller, der bliver opstillet for robotens bane, er outputvektor  $\underline{p}$  identisk med tilstandsvektoren  $\underline{q}$ , dvs.  $\underline{C} = \underline{I}$  og  $\underline{D} = \underline{0}$

Ligningerne (2.3), (2.4) og (2.5) beskriver et ulineært system, der kan skrives på følgende måde:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.8)$$

Tilstandsligningerne (2.8 side 5) lineariseres, fordi opgaven går ud på at lave en lineær regulator. Lineariseringen ses nedenfor.

$$\begin{aligned} \dot{x} &\simeq x_0 + \frac{\partial}{\partial v} \Big|_{\theta=\theta_0}^{v=v_0} (v \cdot \cos(\theta)) (v - v_0) + \frac{\partial}{\partial \theta} \Big|_{\theta=\theta_0}^{v=v_0} (v \cdot \cos(\theta)) (\theta - \theta_0) \\ &= v_0 \cdot \cos(\theta_0) + \cos(\theta_0) \cdot (v - v_0) - v_0 \cdot \sin(\theta_0) \cdot (\theta - \theta_0) \end{aligned}$$

$$\begin{aligned} \dot{y} &\simeq y_0 + \frac{\partial}{\partial v} \Big|_{\theta=\theta_0}^{v=v_0} (v \cdot \sin(\theta)) (v - v_0) + \frac{\partial}{\partial \theta} \Big|_{\theta=\theta_0}^{v=v_0} (v \cdot \sin(\theta)) (\theta - \theta_0) \\ &= v_0 \cdot \sin(\theta_0) + \sin(\theta_0) \cdot (v - v_0) + v_0 \cdot \cos(\theta_0) \cdot (\theta - \theta_0) \end{aligned}$$

$$\dot{\theta} = \omega$$

Der vælges  $\theta_0 = 0$ , da dette giver en meget simpel model. Senere undersøges det, i hvor stort interval af  $\theta$ , omkring 0, denne model virker.

$$\begin{aligned} \dot{x} &\simeq v_0 \cdot \cos(0) + \cos(0) \cdot (v - v_0) - v_0 \cdot \sin(0) \cdot (\theta - 0) \\ &= v_0 + (v - v_0) = v \end{aligned}$$

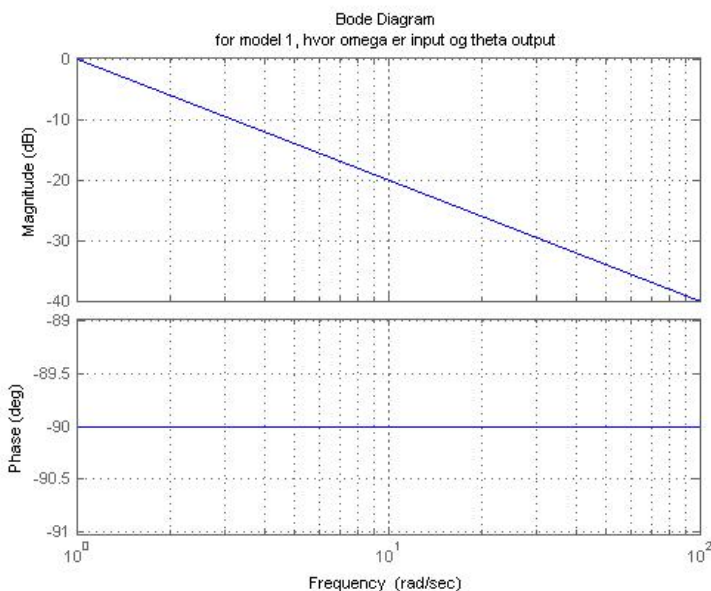
$$\begin{aligned} \dot{y} &= v_0 \cdot \sin(0) + \sin(0) \cdot (v - v_0) + v_0 \cdot \cos(0) \cdot (\theta - 0) \\ &= v_0 \cdot \theta \end{aligned}$$

$$\dot{\theta} = \omega$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & v_0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.9)$$

De lineariserede tilstandsligninger (ligning (2.9)) er udgangspunktet for opstilling af tre forskellige modeller for regulering af robotens bane. Disse tre modeller beskrives og analyseres i de følgende afsnit (se afsnit 2.1.1 side 6, 2.1.2 side 7 og 2.1.3 side 8).

### 2.1.1 Model 1



Figur 2.2: Bodeplot for model 1:  $G_1(s) = \frac{\theta(s)}{\omega(s)}$

Model 1 er den simpleste model. Denne model bruges til vinkelstyring, dvs. der styres på vinklen  $\theta$  ved hjælp af omdrejningshastigheden  $\omega$ . Model 1 skal bruges i tilfælde, hvor det kun er vigtigt at få drejet robotten i en bestemt retning. Derfor bliver tilstandsligningerne for model 1 reduceret til formen:

$$\begin{bmatrix} \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \begin{bmatrix} \theta \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} \omega \end{bmatrix} \quad (2.10)$$

Bodeplottet (frekvenskarakteristik) for denne model er vist på figur 2.2 side 6.

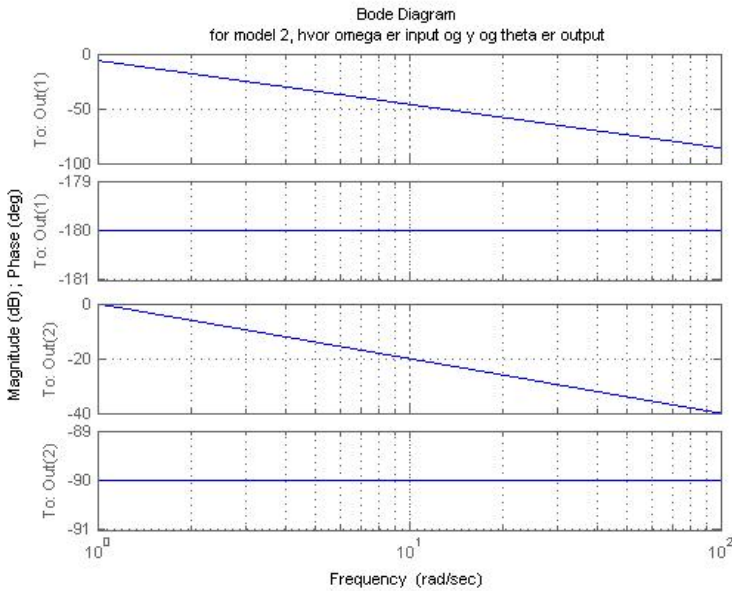
Udfra bodeplottet på figur 2.2 ses, at modellen er en enkelt integrator. Dette stemmer overens med at  $\dot{\theta} = \omega$ .

### 2.1.2 Model 2

Model 2 skal bruges til at styre robotten med konstant hastighed ( $v_0 = 0.5m/s$ ). Her reguleres der på både  $y$  og  $\theta$  ved hjælp af  $\omega$ . Tilstandsligningerne for model 2 er:

$$\begin{bmatrix} \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & v_0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [\omega] \quad (2.11)$$

Bodeplot for denne model er vist på figur 2.3 side 7.



Figur 2.3: Bodeplot for model 2:  $G_{21}(s) = \frac{y(s)}{\omega(s)}$ ;  $G_{22}(s) = \frac{\theta(s)}{\omega(s)}$

Model 2 har to output  $y$  og  $\theta$ , outputtet  $\theta$  er identisk med output fra model 1 (se afnit 2.1.1 side 6). Dette kan også ses på de to nederste grafer i bodeplottet figur 2.3 side 7, som er identiske med figur 2.2 side 6. De to øverste grafer på

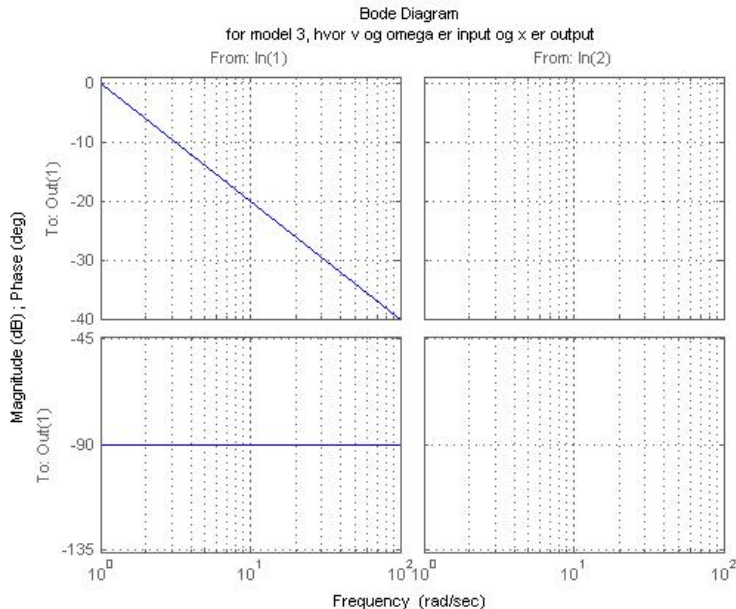
figur 2.3 side 7 viser frekvenskarakteristikker for  $G(s) = y(s)/\omega(s)$ , som er en dobbelt integrator. Dette er fordi  $y(t) = \int \int v_0 \theta(t) dt = \int \int v_0 \omega(t) dt$

### 2.1.3 Model 3

Model 3 skal bruges i forbindelse med hastighedsregulering. Det er den samme model som model 2 udvidet med hastighedsregulering, så der styres på  $x$ ,  $y$  og  $\theta$  ved hjælp af  $v$  og  $\omega$ . Tilstandsligningerne for model 3 ses nedenfor.

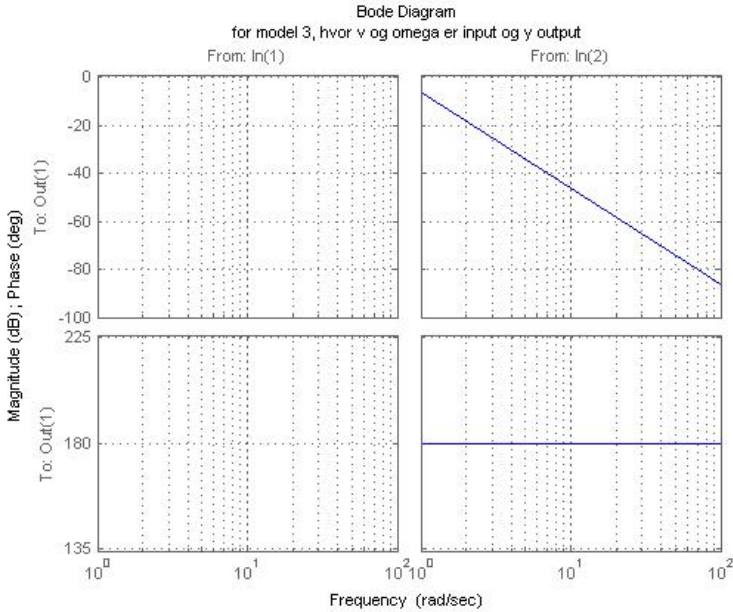
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & v_0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.12)$$

Starthastigheden er sat til  $v_0 = 0.5 \text{ m/s}$ . Bodeplottene for de tre output er vist på tre separate figurer



Figur 2.4: Bodeplot for model 3:  $Gx_{31}(s) = \frac{x(s)}{v(s)}$ ;  $Gx_{32}(s) = \frac{x(s)}{\omega(s)}$

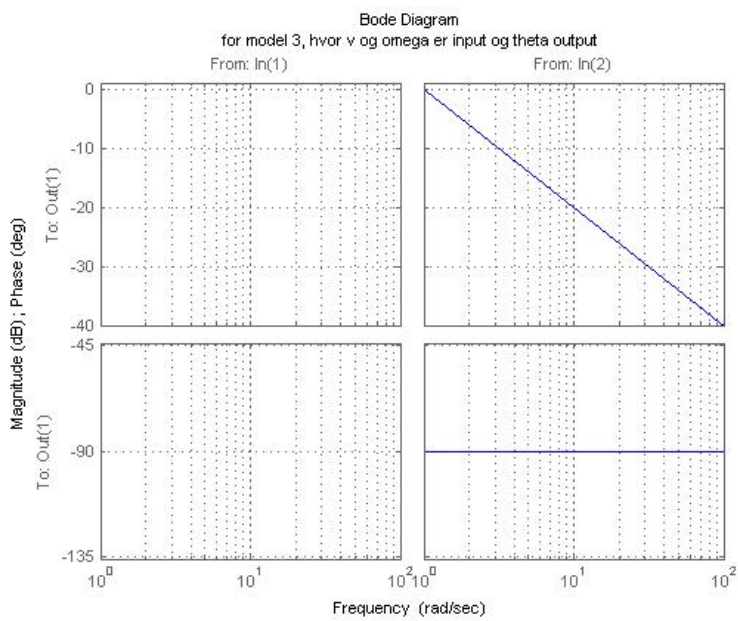
Bodeplottet for output  $x$  ses på figur 2.4 side 8. Forholdet mellem det første input  $v$  og output  $x$  er en integrator. Der er ingen afhængighed mellem det andet input  $\omega$  og output  $x$ .



Figur 2.5: Bodeplot for model 3:  $Gy_{31}(s) = \frac{y(s)}{v(s)}$ ;  $Gy_{32}(s) = \frac{y(s)}{\omega(s)}$

Output  $y$  (se figur 2.5 side 9) er ikke afhængigt af første input  $v$ . For det andet input  $\omega$  ses igen en dobbelt integrator som i model 2 (se afsnit 2.1.2 side 7).

På figur 2.6 side 10 ses bodeplottet for output  $\theta$ . Der er ingen afhængighed mellem det første input  $v$  og output  $\theta$ . Forholdet mellem andet input  $\omega$  og output  $\theta$  er en integrator som i både model 1 og model 2 (se afsnit 2.1.1 side 6 og afsnit 2.1.2 side 7).

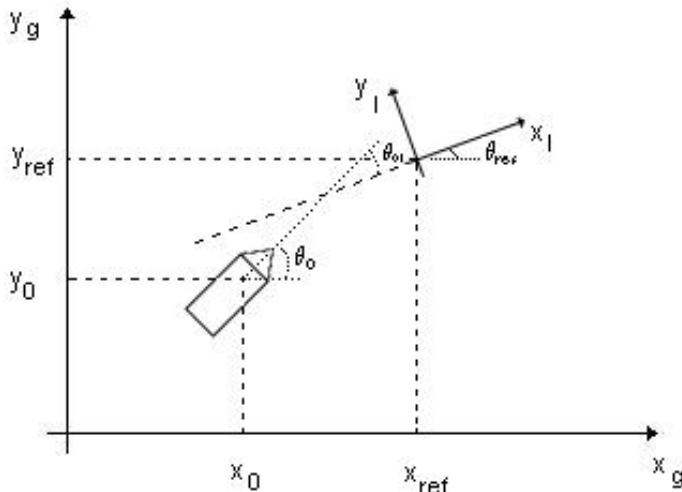


Figur 2.6: Bodeplot for model 3:  $G\theta_{31}(s) = \frac{\theta(s)}{v(s)}$ ;  $G\theta_{32}(s) = \frac{\theta(s)}{\omega(s)}$



### 2.1.4 Det lokale koordinatsystem

De opstillede modeller skal bruges til at få robotten til at køre langs x-aksen. Robotten skal også kunne følge en villkårlig linie i  $x,y$ -planet. Derfor indføres et lokalt koordinatsystem i det globale koordinatsystem. Det globale koordinatsystem ses på figur 2.1 side 4. Det lokale koordinatsystem tager udgangspunkt i det referencepunkt, som robotten skal køre til og den referencevinkel, som robotten skal følge. x-aksen i det lokale koordinatsystem har retningen som robotten skal køre, og y-aksen står vinkelret på x-aksen. Nulpunktet for det lokale koordinatsystem bliver placeret i det referencepunkt, som robotten skal finde hen til. Alt dette illustreres på figur 2.7 side 11.



Figur 2.7: Lokalt koordinatsystem i det globale koordinatsystem

Idet der laves et nyt lokalt koordinatsystem, som vist på figur 2.7 side 11, kan robotten altid styres mod "x-aksen" (i det lokale koordinatsystem) uanset hvor robotten skal styres hen (i det globale koordinatsystem).

For at finde robottens position i det lokale koordinatsystem, skal der foretages basisskifte på robottens globale koordinater (se [LA 1999]). Det oprindelige (globale) koordinatsystem har akserne  $x_g$  og  $y_g$  og det nye (lokale) koordinatsystem har akserne  $x_l$  og  $y_l$ . For at transformere fra globalt til lokalt koordinatsystem, skal basisskifte matricen  ${}^lM_g$  bruges. Denne findes ved at:

$$\underline{{}^lM_g} = \left[ \underline{{}^gM_l} \right]^{-1} \quad (2.13)$$

hvor  $\underline{\underline{g}}M_l$  består af vektorerne, som beskriver det lokale koordinatsystems akser set udfra det globale koordinatsystem.

$$\begin{aligned}\underline{\underline{l}}M_g &= \begin{bmatrix} \cos(\theta_{ref}) & -\sin(\theta_{ref}) \\ \sin(\theta_{ref}) & \cos(\theta_{ref}) \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \cos(\theta_{ref}) & \sin(\theta_{ref}) \\ -\sin(\theta_{ref}) & \cos(\theta_{ref}) \end{bmatrix}\end{aligned}\quad (2.14)$$

Robottens position og retning i forhold til det lokale koordinatsystem findes ved hjælp af formlerne:

$$\begin{bmatrix} x_{0l} \\ y_{0l} \end{bmatrix} = \underline{\underline{l}}M_g \begin{bmatrix} x_0 - x_{ref} \\ y_0 - y_{ref} \end{bmatrix}\quad (2.15)$$

$$\theta_{0l} = \theta_0 - \theta_{ref}\quad (2.16)$$

hvor alle variable med indeks "l" er set fra det lokale koordinatsystem, og alle andre variable er set fra det globale koordinatsystem. Altså er  $x_{0l}$ ,  $y_{0l}$  og  $\theta_{0l}$  de nye lokale koordinater.

## 2.2 Det polære koordinatsystem

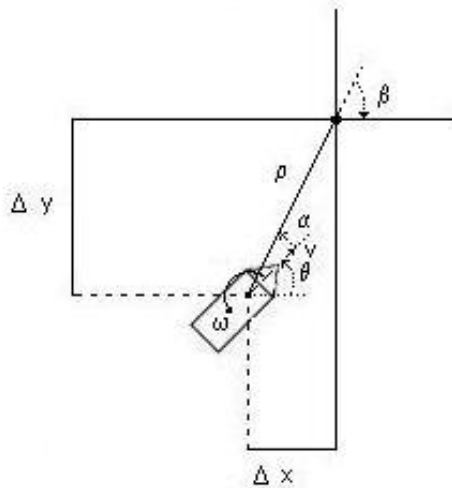
En anden mulighed er at beskrive robottens bane ved hjælp af de polære koordinater [Robots 2004]. Dette resulterer i en lignende effekt som ved indførelse af det lokale koordinatsystem med hensyn til regulering af robottens bevægelser

På figur 2.8 side 13 ses det polære koordinatsystem, hvor:

$\rho$  er afstanden fra robotten til referencepunktet

$\alpha$  er vinkelforskellen mellem robottens vinkel og vinklen for liniestykket mellem robotten og referencepunktet

$\beta$  vinklen på den linie, robotten skal køre



Figur 2.8: Robotten i det polære koordinatsystem

Disse værdier kan bestemmes ud fra koordinaterne i det kartesiske koordinatsystem (se afsnit 2.1 side 4) ved hjælp af formlerne:

$$\rho = \sqrt{\Delta x^2 + \Delta y^2} \quad (2.17)$$

$$\alpha = -\theta + \arctan\left(\frac{\Delta y}{\Delta x}\right) \quad (2.18)$$

$$\beta = -\theta_{ref} \quad (2.19)$$

Ligningerne (2.17), (2.18) og (2.19) kan skrives i form af tilstandsligninger:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos(\alpha) & 0 \\ \frac{\sin(\alpha)}{\rho} & -1 \\ -\frac{\sin(\alpha)}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.20)$$

For  $v = -v$  (ændring af retning) fås:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & 0 \\ -\frac{\sin(\alpha)}{\rho} & 1 \\ \frac{\sin(\alpha)}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.21)$$

Modellen gælder ikke for  $\rho = 0$ , da den indeholder  $\frac{\sin(\alpha)}{\rho}$ . For at løse dette

problem beskrives kontrolvariablerne (lineær beskrivelse) som:

$$v = k_\rho \rho \quad (2.22)$$

$$\omega = k_\alpha \alpha + k_\beta \beta \quad (2.23)$$

Dette giver:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho \rho \cos(\alpha) \\ k_\rho \sin(\alpha) - k_\alpha \alpha - k_\beta \beta \\ -k_\rho \sin(\alpha) \end{bmatrix} \quad (2.24)$$

Dette system lineariseres nu omkring nulpunktet  $(\rho, \alpha, \beta) = (0, 0, 0)$ , dvs.  $\cos(x) = 1$  og  $\sin(x) = x$  for små værdier af  $x$ .

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho & 0 & 0 \\ 0 & -(k_\alpha - k_\rho) & -k_\beta \\ 0 & -k_\rho & 0 \end{bmatrix} \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix} \quad (2.25)$$

hvor  $k = (k_\rho, k_\alpha, k_\beta) = (3, 8, -1.5)$ .

Disse værdier for  $k$  er valgt ud fra stabilitets analyse se [\[Robots 2004\]](#).

## KAPITEL 3

# Simuleringsplatform

---

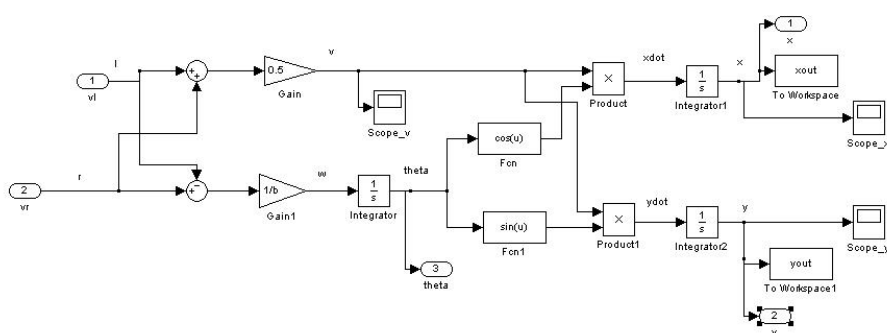
I dette kapitel beskrives en SIMULINK-model af den matematiske model fra kapitel 2. Modellen testes på to eksempler for at afprøve dens gyldighed.

En SIMULINK-model for robotens bane opbygges ud fra de tre differentialligninger (2.3), (2.4) og (2.5) side 4. Denne model har hastighederne på robotens venstre- og højrehjul ( $v_l, v_r$ ) som input og positionen ( $x, y$ ) som output (se figur 3.1 side 16).

I det første eksempel starter roboten i punktet (0,0) med vinkel  $\theta = 0$  og afstanden mellem venstre- og højrehjul er givet ved  $b = 5 \cdot 10^{-2} m$ , altså 5 cm. Venstre- og højrehjul gives hastighederne  $v_l = 0.3 m/s$  og  $v_r = 0.5 m/s$ . Teoretisk set burde roboten køre i en cirkel, denne cirkels radius bestemmes ud fra ligningerne (2.3) side 4 og (2.4) side 4.

$$\begin{aligned}x(t) &= \int v \cdot \cos(\theta) dt \\y(t) &= \int v \cdot \sin(\theta) dt\end{aligned}$$

hvor  $\theta = \omega t$  (ligning (2.5) side 4), fordi  $\omega$  er konstant.  $x$ - og  $y$ -koordinaterne for



Figur 3.1: Simulink model for robotens bane

banen kan nu findes som:

$$x(t) = \int v \cdot \cos(\omega t) dt = \frac{v}{\omega} \sin(\omega t)$$

$$y(t) = \int v \cdot \sin(\omega t) dt = -\frac{v}{\omega} \cos(\omega t)$$

og cirkelns radius er givet ved  $\sqrt{x^2 + y^2}$ , hvor

$$\begin{aligned} r^2 &= x(t)^2 + y(t)^2 \\ &= \left(\frac{v}{\omega} \sin(\omega t)\right)^2 + \left(-\frac{v}{\omega} \cos(\omega t)\right)^2 \\ &= \left(\frac{v}{\omega}\right)^2 (\sin^2(\omega t) + \cos^2(\omega t)) \\ &= \left(\frac{v}{\omega}\right)^2 \end{aligned}$$

Cirkelns radius er derfor givet ved:

$$r = \left| \frac{v}{\omega} \right| \quad (3.1)$$

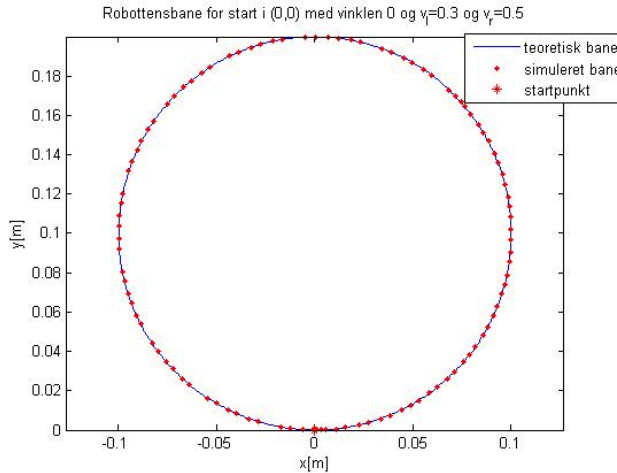
I dette eksempel er:

$$v = \frac{v_r + v_l}{2} = \frac{0.5 \text{ m/s} + 0.3 \text{ m/s}}{2} = 0.4 \text{ m/s}$$

$$\omega = \frac{v_r - v_l}{b} = \frac{0.5 \text{ m/s} - 0.3 \text{ m/s}}{0.05 \text{ m}} = 4 \text{ rad/s}$$

Derfor skal robotten køre på en cirkel med radius  $0.1 \text{ m}$  og med centrum i  $(-0.1 \cdot \sin(\theta), 0.1 \cdot \cos(\theta))$ , altså i  $(0, 0.1)$ .

SIMULINK-modellen køres for dette eksempel og den simulerede bane er plottet sammen med den teoretiske bane på figur 3.2 side 17.



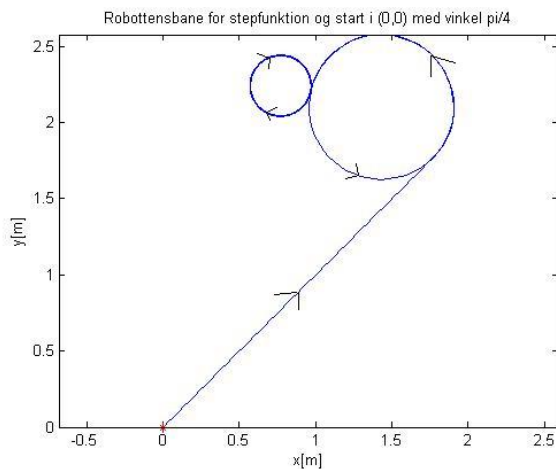
Figur 3.2: Den teoretiske bane og den simulerede bane med start  $i: (x, y) = (0, 0)$ ;  $\theta = 0$ ;  $v_l = 0.3 \text{ m/s}$ ;  $v_r = 0.5 \text{ m/s}$

Det ses på figur 3.2 side 17, at den simulerede bane og den teoretiske bane er ens. Det viser, at SIMULINK-modellen er rigtig.

Det andet eksempel går ud på at undersøge SIMULINK-modellen for forskellige påvirkninger. I stedet for konstanter for  $v_l$  og  $v_r$  indsættes nu STEP-funktioner, så robotten kan køre i forskellige retninger. Robotten starter i punktet  $(0, 0)$  med vinklen  $\theta = \pi/4$  og med starthastighed  $0.5 \text{ m/s}$  på begge hjul. Der er indsat et step til tiden 5 sekunder, hvor hastigheden på venstrehjul reduceres til  $0.45 \text{ m/s}$  og et andet step til tiden 15 sekunder, hvor hastigheden på højrehjul reduceres til  $0.35 \text{ m/s}$ .

Robotten burde de første 5 sekunder køre ad en ret linie, fordi hastighederne på begge hjul er ens. Denne linie danner en vinkel på  $\pi/4$  med x-aksen, da startvinklen er  $\theta = \pi/4$ . Længden på denne linie er bestemt af den hastighed robotten kører med. Da robotten kører  $0.5 \text{ m/s}$  i 5 sekunder, burde den køre  $2.5 \text{ m}$ , dvs. robotten burde begynde at dreje i punktet  $\left(\frac{2.5}{\sqrt{2}}, \frac{2.5}{\sqrt{2}}\right) = (1.7678, 1.7678)$ .  $v_l$  der reduceres på det tidspunkt (5 sec.) og robotten skal køre i en cirkel mod uret. Denne cirkel har radius  $0.475 \text{ m}$  (givet ved ligning (3.1) side 16). Til tiden 15 sekunder reduceres  $v_r$ , så denne er  $0.1 \text{ m/s}$  lavere end  $v_l$ , derfor skal robotten køre med uret i en cirkel med radius  $0.2 \text{ m}$ .

For at se, om simuleringen af robotens bane stemmer overens med den teoretisk bane, plottes den simulerede bane, se figur 3.3 side 18.



Figur 3.3: Robotens bane, hvor  $v_l$  reduceres og efterfølgende reduceres  $v_r$ .

På figur 3.3 side 18 ses det, at robotens simulerede bane stemmer overens med den bane roboten teoretisk burde køre i dette eksempel.

Begge eksempler viser, at SIMULINK-modellen (på figur 3.1 side 16) virker korrekt. Derfor bruges modellen i afsnit 4 side 19 for at få roboten til at køre forskellige baner ved hjælp af regulering og styring.



# Regulering og styring

---

I dette kapitel beskrives forskellige metoder til styring og regulering af robotens bane. De forskellige regulerings- og styringsmetoder indføres i SIMULINK-modellen og afprøves på eksempler. Til sidst sammenlignes metoderne.

SIMULINK-modellen fra figur 3.1 side 16 udvides med en reguleringsalgoritme. Det betyder, at der indføres tilbagekobling på systemet. Robotten skal følge en bestemt bane. Denne SIMULINK-model ses på figur 4.1 side 20.

På figur 4.1 side 20 ses to delsystemer, i SIMULINK kaldes det for "subsystems".

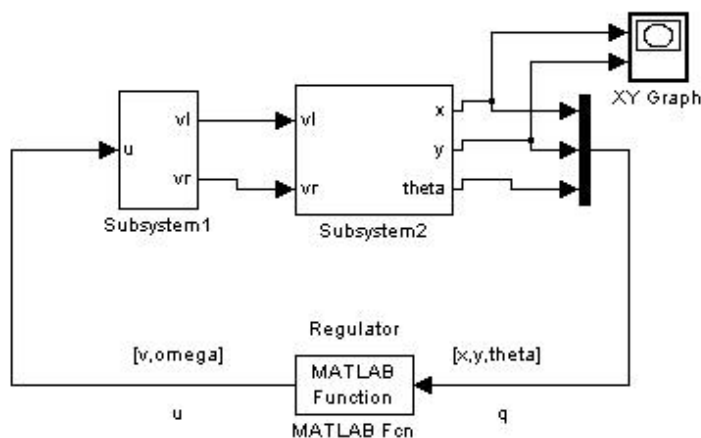
*Subsystem1* beregner  $v_l$  og  $v_r$  ved hjælp af formel (2.1) og (2.2) side 3 og:

$$u = [v, \omega]^T \text{ er inputvektor}$$

*Subsystem2* er den model, som er vist på figur 3.1 side 16. I den beregnes  $x$ ,  $y$  og  $\theta$  ud fra  $v_l$  og  $v_r$ .

Regulatoren er en MATLAB-funktion, der beregner reguleringsalgoritmen.

Der findes forskellige metoder til at beregne reguleringsalgoritmen til tilbagekoblingen. I dette projekt er reguleringsalgoritmen den såkaldte LQR-metode, som betyder LINEAR QUADRATIC REGULATOR. LQR-metoden sikrer, at systemet er



Figur 4.1: Simulink model for system med tilbagekobling

stabil, samtidig med at forskellen mellem den ønskede bane og den aktuelle bane minimeres [LQR 1995]. I metoden minimeres følgende kostfunktion [Control 2006]:

$$J = \int_0^{\infty} (q^T Q q + u^T R u) dt$$

Valg af  $Q$  og  $R$  beskrives for de forskellige modeller.

Når  $Q$  og  $R$  er valgt benyttes MATLAB-funktionen  $lqr$

$$L = lqr(A, B, Q, R)$$

Så kan inputtet  $u$  bestemmes som:

$$u = -L*(q-q_{ref})$$

hvor  $q_{ref}$  er referencen robotten skal styres hen til.

Robotten skal kunne køre ad forskellige baner. Derfor er det første at få robotten til at køre til x-aksen og herefter følge x-aksen. Det andet er så at flytte x-aksen (se afsnit 2.1.4 side 11), så robotten kan køre ad en ønsket bane.

## 4.1 Styring til x-aksen uden hastighedsregulering

I dette afsnit opstilles den simple model (uden hastighedsstyring) for at styre mod x-aksen. Denne model vil senere blive udvidet til også at regulere på hastigheden.

Robotten skal styres mod x-aksen og så følge den. Der bruges model 2 (se afsnit 2.1.2 side 7). Her reguleres kun på  $\theta$  og  $y$ .

$$\begin{bmatrix} \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & v_0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [\omega] \quad (4.1)$$

Robottens fremadrettede hastighed  $v$  holdes konstant  $0.5 \text{ m/s}$ .

Da systemet er lineariseret i  $\theta_0 = 0$ , undersøges det i hvilket vinkelområde denne linearisering fungerer. Hvis robotten er udenfor dette område, vil den blive styret derind ved hjælp af vinkelstyring (model 1, se afsnit 2.1.1 side 6).

MATLAB funktionen for denne styring findes i bilag B.1 side 97 som `YTHkontrol.m`.

Robotten starter i punktet  $(2, 3)$  og med  $\theta = 0$ , og der testes hvilke værdier af  $Q$  og  $R$ , der giver den bedste bane.

For

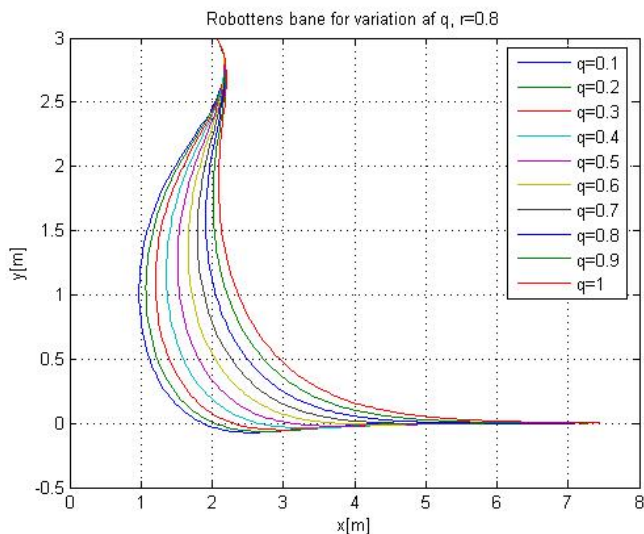
$$Q = \begin{bmatrix} 1 & 0 \\ 0 & q^2 \end{bmatrix} \quad (4.2)$$

$$R = [r^2] \quad (4.3)$$

laves der parametertuning for både  $q$  og  $r$ . Først fastholdes  $r = 0.8$  og der afprøves følgende  $q$ -værdier:

$$[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1] \quad (4.4)$$

$y(x)$  og  $y(t)$  plottes (se figur 4.2 side 22 og 4.3 side 23). Plot af  $x(t)$ ,  $\theta(t)$ ,  $\omega(t)$ ,  $v_l(t)$  og  $v_r(t)$  ses i bilag A.1 side 59. Det er i dette tilfælde vigtigt at komme hurtigt ned mod x-aksen ( $y = 0$ ). Derfor vælges den  $q$ -værdi, som får robotten til at komme hurtigst mod x-aksen og som følger x-aksen med mindst mulig afvigelse.



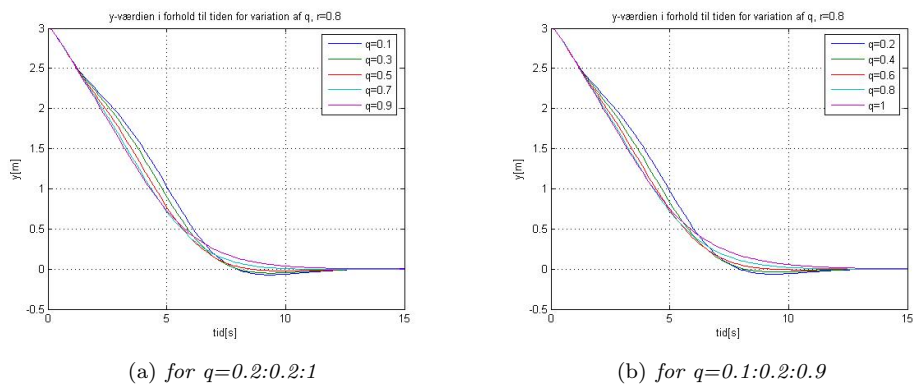
Figur 4.2: Robottens bane for variation af  $q$  og  $r = 0.8$

På figur 4.2 side 22 og 4.3 side 23 ses det, at det vil være mest fordelagtigt at benytte værdien  $q = 0.6$ , da denne værdi får robotten hurtigst ned til  $y = 0$  (se figur 4.3 side 23) og giver ikke oversving (robotten kører ikke under x-aksen).

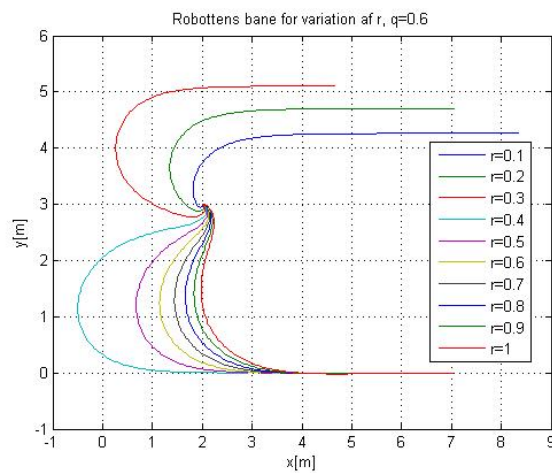
Nu fastholdes  $q = 0.6$  og  $r$  varieres for værdierne i ligning (4.4) side 21. Plot for robottens bane  $y(x)$  og  $y(t)$  ses på figur 4.4 side 23 og figur 4.5 side 24. Plot af  $x(t)$ ,  $\theta(t)$ ,  $\omega(t)$ ,  $v_l(t)$  og  $v_r(t)$  ses i bilag A.1 side 59.

Ud fra figur 4.4 side 23 ses det, at robotten ikke kan styre til x-aksen for  $r < 0.3$ . Dette er fordi vægtningen af omdrejningshastigheden  $\omega$  er for lille og reguleringen for svag. Dette gør, at robotten bevæger sig længere væk fra x-aksen, og når robotten er for langt væk fra x-aksen, fungerer denne model for styringen ikke i hele vinkelområdet (dette beskrives senere, se figur 4.7 side 25).

0.8, som først valgt, er en god værdi for  $r$  (se figur 4.4 side 23 og figur 4.5 side 24). Dette er fordi der næsten ingen forskel er på, hvor hurtigt robotten når x-aksen for  $r = 0.8$ ,  $r = 0.9$  og  $r = 1$ , men for  $r = 0.8$  er x-værdien mindre for  $y = 0$ .



Figur 4.3:  $y$  som funktion af tiden for variation af  $q$  og  $r = 0.8$

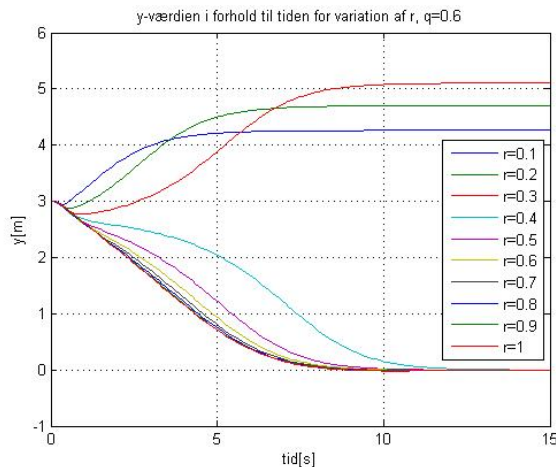


Figur 4.4: Robottens bane for variation af  $r$  og  $q=0.6$

For de valgte værdier  $q = 0.6$  og  $r = 0.8$  er  $L$ :

$L =$

1.2500      1.3463



Figur 4.5:  $y$ -værdien i forhold til tiden for variation af  $r$  og  $q=0.6$

Der undersøges nu i hvor stort et vinkelområde, omkring  $\theta = 0$  denne regulering fungerer. Dette gøres ved at indsætte forskellige startvinkler i området  $]-\pi; \pi]$  i simuleringen.

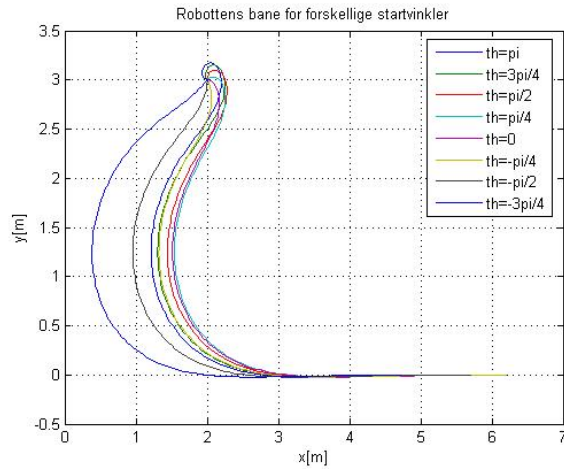
Robotten starter i punktet  $(2, 3)$  og der gives følgende startvinkler:

$$\theta_0 = [-3\pi/4, -\pi/2, -\pi/4, 0, \pi/4, \pi/2, 3\pi/4, \pi]$$

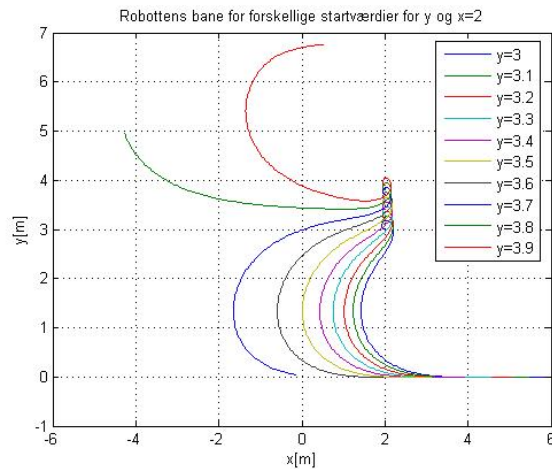
Simuleringen viser, at reguleringen virker for disse startvinkler, når robotten er placeret  $3\text{ m}$  fra  $x$ -aksen (se figur 4.6 side 25).

Denne regulering virker ikke, når robotten har for stor afstand til  $x$ -aksen. Det er derfor nødvendigt at styre robotten tættere på  $x$ -aksen (med vinkelstyring, se afsnit 2.1.1 side 6), hvis denne er for langt væk. Det undersøges derfor, hvor langt fra  $x$ -aksen denne regulering virker. Robotten starter derfor med vinklen  $\theta = \pi$ , hvor startpunktet rykkes længere og længere væk fra  $x$ -aksen. Robottens baner for disse simuleringer ses på figur 4.7 side 25.

På figur 4.7 side 25 ses det, at reguleringen ikke fungerer for  $y = 3.8$ . Jeg har derfor valgt at lægge et "y-bånd" omkring  $x$ -aksen for  $y = \pm 3.5$ . Dette betyder, at når robotten er længere væk fra  $x$ -aksen end  $3.5\text{ m}$  styres den vinkelret ( $\theta = -\pi/2$ ) ind mod  $x$ -aksen ved hjælp af model 1 afsnit 2.1.1 side 6. Så snart robotten kommer indenfor  $3.5$  meters afstand til  $x$ -aksen, skiftes fra vinkelstyring til regulering af  $y$  og  $\theta$ , som beskrevet ovenfor.



Figur 4.6: Robottens bane for forskellige startvinkler mellem  $]-\pi, \pi]$



Figur 4.7: Robottens bane for forskellige startværdier for  $y_0$ ,  $x_0 = 2$  og  $\theta_0 = \pi$

Der laves parametertuning for vinkelstyringen (model 1).

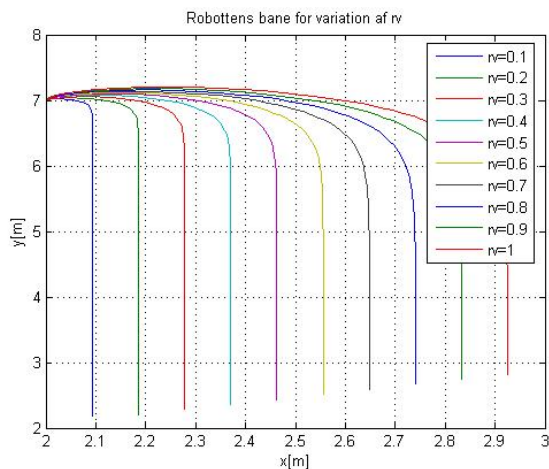
$$\begin{bmatrix} \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \begin{bmatrix} \theta \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} \omega \end{bmatrix} \quad (4.5)$$

Her er:

$$Q = [1]$$

$$R = [r_v^2]$$

hvor  $r_v$  varieres for værdierne i ligning (4.4). Der startes i punktet (2, 7) med vinklen  $\pi/2$  og der styres kun på vinklen i 10 sekunder.  $y(x)$  og hastighederne på hjulene ( $v_l(t)$  og  $v_r(t)$ ) i forhold til tiden plottes (se figur 4.8 side 26, 4.9 side 27 og 4.10 side 27). Plot af  $x(t)$ ,  $y(t)$ ,  $\theta(t)$  og  $\omega(t)$  ses i bilag A.1 side 59. I praksis er det vigtigt, at forskellen mellem  $v_l$  og  $v_r$  ikke er for stor, og at de begge to er positive, ellers kan det ikke realiseres.



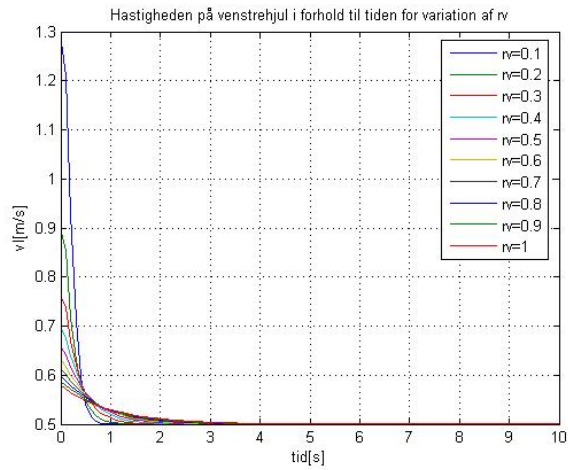
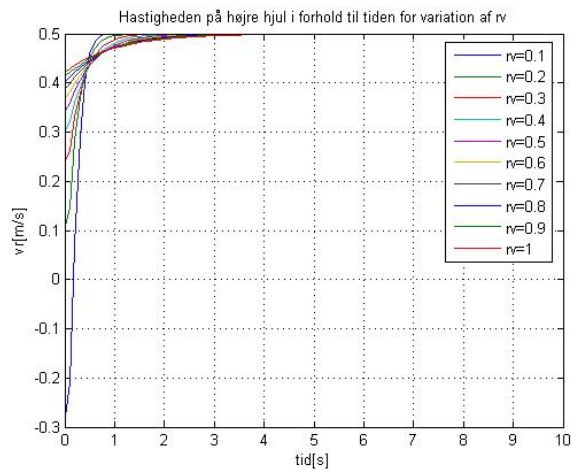
Figur 4.8: Robottens bane for variation af  $r_v$

På figur 4.8 side 26 ses det, at robotten drejer skarpere jo lavere  $r_v$  er, og derfor bliver forskellen mellem hjulenes hastigheder (se figur 4.9 side 27 og figur 4.10 side 27) større jo lavere  $r_v$  er. Det er derfor nødvendigt at vælge en værdi, så hjulenes hastigheder ikke har forskellige fortegn, og forskellen mellem dem ikke er urealistisk stor. Jeg vælger derfor at bruge  $r_v = 0.8$ , da forskellen mellem  $v_l$  og  $v_r$  her ikke bliver større end  $0.2 \text{ m/s}$ , og der sker ikke stor ændring i hastigheden på kort tid. For  $r_v = 0.8$  bliver:

L =

1.2500

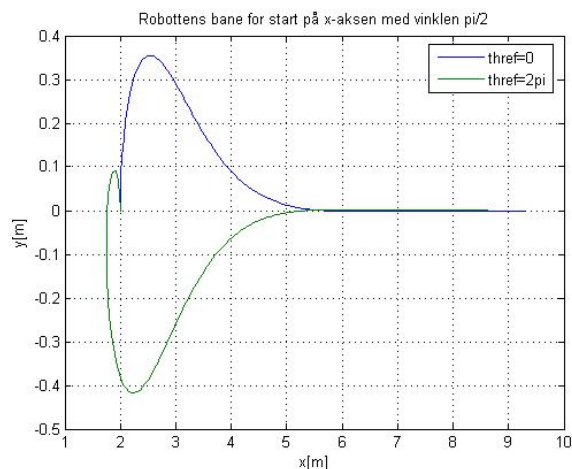


Figur 4.9: Hastigheden på venstre hjul som funktion af tiden for variation af  $r_v$ Figur 4.10: Hastigheden på højre hjul som funktion af tiden for variation af  $r_v$

Det samme gøres også for  $y < 0$ , men resultaterne viser sig at være de samme som for  $y > 0$  (bare spejlet i x-aksen) se bilag A.1.2 side 66.

Når robotten starter på x-aksen (dvs.  $y = 0$ ) med vinklen enten  $\pi/2$  eller  $-\pi/2$  undersøges det, hvilken vej robotten skal dreje for at komme hurtigst ind til x-aksen igen (med vinklen  $\theta = 0$ ).

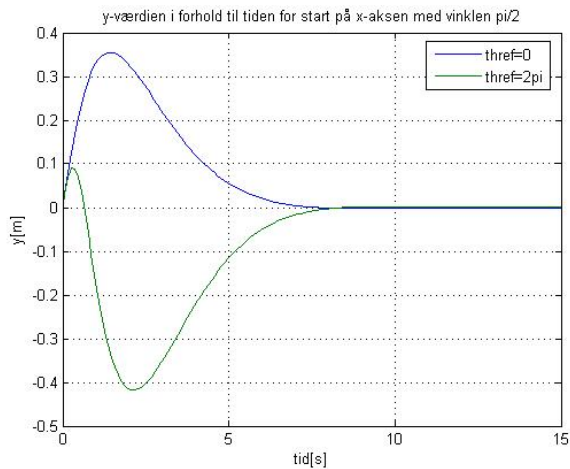
Robotten startes i  $(2, 0)$  med vinklen  $\pi/2$ . Der styres først mod  $\theta = 0$  (drejning med uret) og herefter mod  $\theta = 2\pi$  (dejning mod uret). Robottens bane og  $y$ -værdien som funktion af tiden plottes, se figur 4.11 side 28 og figur 4.12 side 29



Figur 4.11: Robottens bane for start på x-aksen med vinklen  $\theta_{ref} = \pi/2$  for reference vinkel på både 0 og  $2\pi$

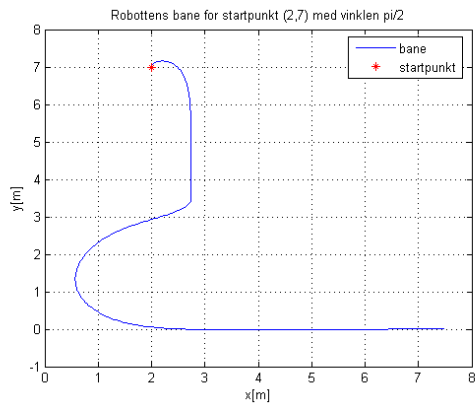
På figur 4.11 side 28 og figur 4.12 side 29 kan det være svært at se, hvilken af banerne først når tilbage til  $y = 0$ . Derfor ses på de numeriske data for banerne. For banen med  $\theta_{ref} = 0$  når robotten tilbage til x-aksen til tiden 7.6627 s, og for banen med  $\theta_{ref} = 2\pi$  er det først til tiden 8.3627 s. Det samme gøres for startvinklen  $\theta = -\pi/2$  og her er resultatet også at robotten kommer hurtigst tilbage til x-aksen med referencen  $\theta = 0$ , se figur i bilag A.1.2. Derfor vil  $\theta = 0$  være referencen for styringen i hele vinkelintervallet  $]-\pi; \pi]$  for alle startpositioner indenfor  $y$ -båndet.

Nu kan robotten finde ind til x-aksen med en vilkårlig startposition og startvinkel og herefter følge x-aksen i den positive retning. Et eksempel på dette ses på figur 4.13 side 29, hvor der først foretages vinkelstyring (model 1 afsnit 2.1.1 side 6)



Figur 4.12:  $y$ -værdien som funktion af tiden for start på  $x$ -aksen med vinklen  $\pi/2$  for reference vinkel på både  $0$  og  $2\pi$

og herefter styres der på både  $y$  og  $\theta$  (model 2 afsnit 2.1.2 side 7). Robotten starter i  $(2, 7)$  med vinklen  $\pi/2$ .



Figur 4.13: Robottens bane for startposition i  $(2, 7)$  med vinklen  $\pi/2$ , hvor der først kun styres på vinklen og herefter ( $y < 3.5$ ) på  $y$  og vinklen

På figur 4.13 side 29 ses det, at reguleringen fungerer korrekt og skifter fra vinkelstyring til at styre på både  $y$  og  $\theta$  for  $y < 3.5$ .

De valgte værdier for  $Q$  og  $R$  får robotten til at følge den ønskede bane, samtidig med, at de er valgt ud fra mere realistiske hastighedsændringer i forhold til tiden, samtidig med at robotten kører efter hensigten.

## 4.2 Styring til x-aksen med hastighedsregulering

Der indføres nu yderligere hastighedsregulering til styringen mod x-aksen. Det betyder, at der reguleres på både  $v$  og  $\omega$ . Til dette bruges model 3 afsnit 2.1.3 side 8.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & v_0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (4.6)$$

Til reguleringen bruges igen LQR-metoden.  $Q$  og  $R$  har nu andre dimensioner end før og dermed flere parametre, der skal tunes.

$$Q = \begin{bmatrix} q_1^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & q_2^2 \end{bmatrix} \quad (4.7)$$

$$R = r_1^2 \begin{bmatrix} 1 & 0 \\ 0 & r_2^2 \end{bmatrix} \quad (4.8)$$

MATLAB koden for dette afsnit findes i bilag B.2 side 98.

Der indføres et referencepunkt for x-koordinaten idet robotten ellers kører mod  $x = 0$ . Dette vil nemlig give problemer, hvis robotten startes i 1. kvadrant i koordinatsystemet og styrer mod:  $x = 0$ ,  $y = 0$  og  $\theta = 0$ . Robotten skal derfor køre til  $(10, 0)$  med vinklen  $\theta = 0$ . Der startes igen i punktet  $(2, 3)$  med vinklen  $\theta_0 = 0$ .

Der laves nu parametertuning på parametrene:  $q_1$ ,  $q_2$ ,  $r_1$  og  $r_2$ . Det er igen vigtigt at få robotten hurtigt ned mod x-aksen, altså  $y = 0$  og herefter få den til at følge x-aksen med mindst mulig afvigelse.

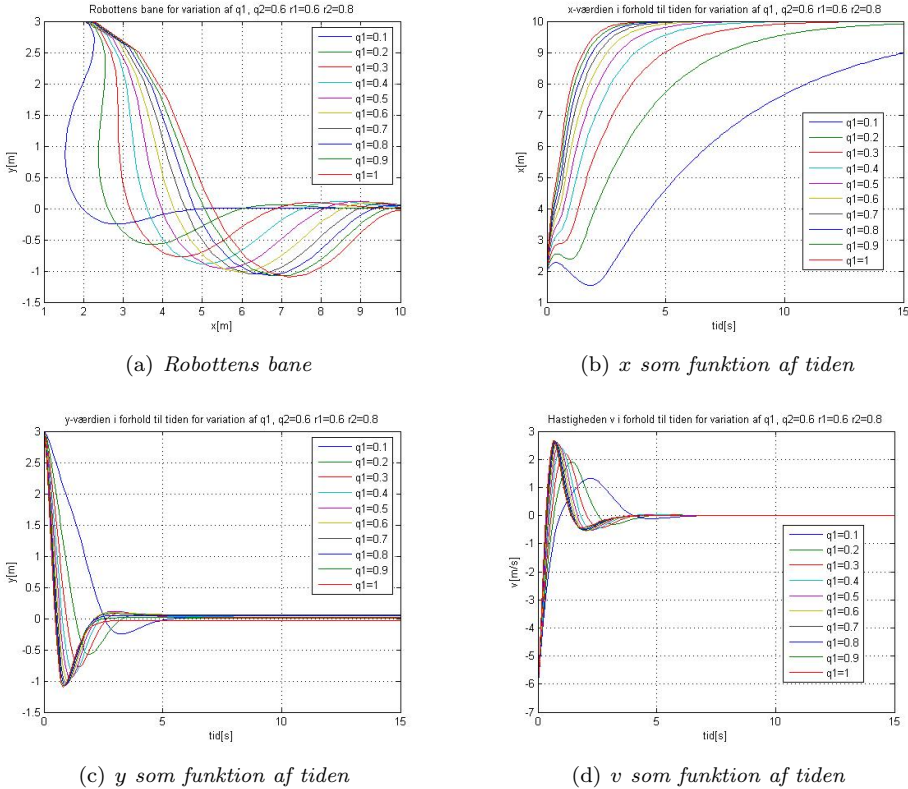
Først varieres  $q_1$ , og de tre andre parametre fastholdes:

$$q_2 = 0.6 \quad r_1 = 0.6 \quad r_2 = 0.8$$

$q_1$  gives værdierne værdierne:

$$[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1] \quad (4.9)$$

For at bestemme, hvilken værdi for  $q_1$  der skal bruges, plottes robotens bane  $y(x)$  og værdierne for  $x(t)$ ,  $y(t)$ ,  $\theta(t)$ ,  $v(t)$ ,  $\omega(t)$ ,  $v_l(t)$  og  $v_r(t)$ . Plottene for  $y(x)$ , og  $x(t)$ ,  $y(t)$  og  $v(t)$  ses på figur 4.14 side 31 og de restende plot ses i bilag A.2.1 side 74.



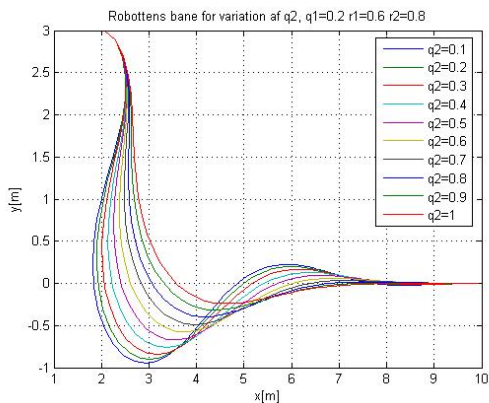
Figur 4.14: Variation af  $q_1$ ,  $q_2 = 0.6$ ,  $r_1=0.6$  og  $r_2=0.8$

På figur 4.14(a) side 31 ses det, at robotten følger  $x$ -aksen "bedst" (med lille afvigelse) for de lave værdier af  $q_1$ . Det ses derimod på figur 4.14(b) og 4.14(c) side 31, at  $q_1 = 0.1$  ikke er en passende værdi. Dette skyldes, at robotten ikke når punktet  $x = 10$  på de 15 sekunder simuleringen kører (figur 4.14(b)), og at robotten bruger lang tid på at nå  $y = 0$  (figur 4.14(c)). Det ses også (på figur 4.14(d) side 31), at hastigheden bliver alt for stor for de høje værdier af  $q_1$ . Det er urealistisk at gennemføre i praksis. Derfor har jeg valgt at fortsætte med værdien  $q_1 = 0.2$ .

Nu varieres  $q_2$  med samme værdier i ligning (4.9), og de andre fastholdes:

$$q_1 = 0.2 \quad r_1 = 0.6 \quad r_2 = 0.8$$

Igen plottes robotens bane  $y(x)$  og værdierne for  $x(t)$ ,  $y(t)$ ,  $\theta(t)$ ,  $v(t)$ ,  $\omega(t)$ ,  $v_l(t)$  og  $v_r(t)$ . Plottene for  $y(x)$  ses på figur 4.15 side 32, og de restende plot ses i bilag A.2.2 side 76.



Figur 4.15: Robotens bane for variation af  $q_2$ ,  $q_1 = 0.2$ ,  $r_1=0.6$  og  $r_2=0.8$

Robotens bane for variation af  $q_2$  ses på figur 4.15 side 32. Her har jeg valgt værdien  $q_2 = 1$ , da de andre værdier giver for stor afvigelse fra x-aksen.

Nu varieres  $r_1$  med samme værdier som  $q_1$  før, og de andre fastholdes:

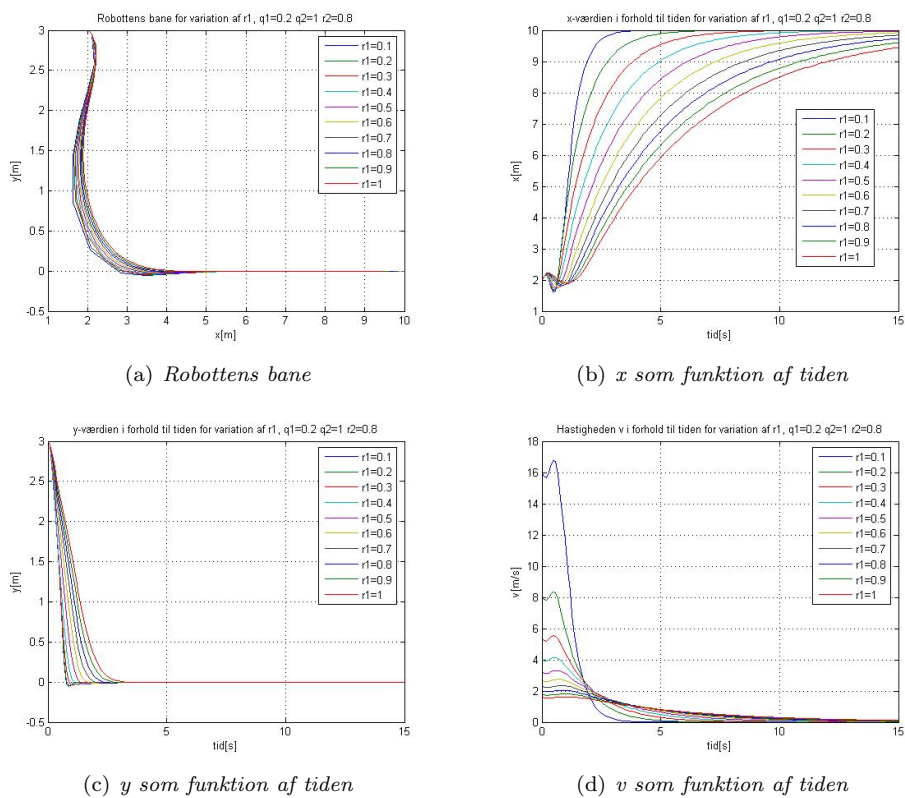
$$q_1 = 0.2 \quad q_2 = 1 \quad r_2 = 0.8$$

Robotens bane  $y(x)$  og værdierne for  $x(t)$ ,  $y(t)$ ,  $\theta(t)$ ,  $v(t)$ ,  $\omega(t)$ ,  $v_l(t)$  og  $v_r(t)$  plottes. Plottene for  $y(x)$ , og  $x(t)$ ,  $y(t)$  og  $v(t)$  ses på figur 4.16 side 33, og de restende plot er i bilag A.2.3 side 80.

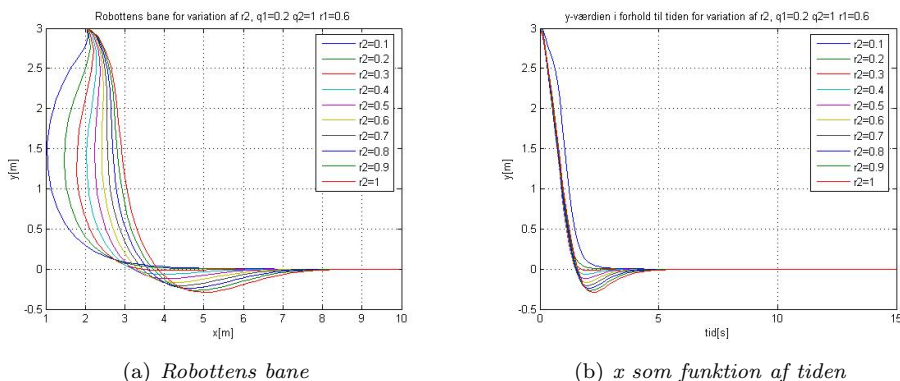
På figur 4.16(d) side 33 ses det, at roboten kører meget hurtigt for lave værdier af  $r_1$ . Robotens bane har en stor afvigelse (se figur 4.16(a) side 33) fra x-aksen for de lave værdier af  $r_1$ . For høje værdier af  $r_1$  er roboten længe om at nå  $y = 0$  (se figur 4.16(c) side 33) og når ikke  $x = 10$  (se figur 4.16(b) side 33) på de 15 sekunder simuleringen kører. Jeg har derfor valgt at bruge  $r_1 = 0.6$ .

Nu varieres  $r_2$  med værdierne fra ligning (4.9) og de andre fastholdes:

$$q_1 = 0.2 \quad q_2 = 1 \quad r_1 = 0.6$$

Figur 4.16: Variation af  $r_1$ ,  $q_1 = 0.2$ ,  $q_2=1$  og  $r_2=0.8$

Robottens bane  $y(x)$  og værdierne for  $x(t)$ ,  $y(t)$ ,  $\theta(t)$ ,  $v(t)$ ,  $\omega(t)$ ,  $v_l(t)$  og  $v_r(t)$  plottes. Plottene for  $y(x)$  og  $y(t)$  ses på figur 4.17 side 34 og de restende plot er i bilag A.2.4 side 82.



Figur 4.17: variation af  $r_2$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$

Det ses, at  $r_2 = 0.3$  er den værdi, hvor afvigelsen fra robottens bane til x-aksen er mindst (se figur 4.17(a) side 34) og hvor robotten hurtigst får værdien  $y = 0$  uden at komme for langt under 0 (se figur 4.17(b) side 34).

Der laves nu parametertuning igen med de nye parametre for at se om det kan gøres bedre, graferne ses i bilag A.2.5 side 85, A.2.6 side 89, A.2.7 side 93.

Det viser sig, at de fundne parametre fra første parametertuning:

$$q_1 = 0.2 \quad q_2 = 1 \quad r_1 = 0.6 \quad r_2 = 0.3 \quad (4.10)$$

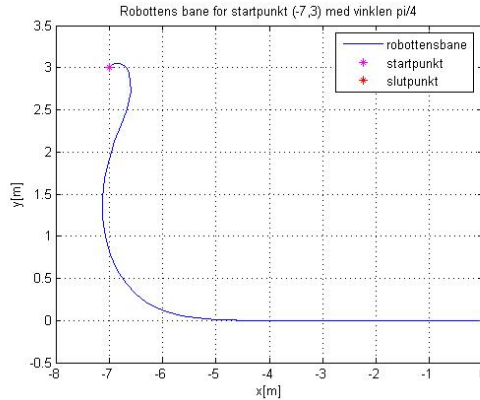
er de bedste værdier til denne styring af robotten. Derfor vil jeg bruge disse parametre fremover, og L-værdien bliver:

L =

$$\begin{array}{ccc} 0.3333 & 0 & 0 \\ 0 & 5.5556 & 6.0349 \end{array}$$

Jeg afprøver nu styringen for de valgte parametre ved at vælge et andet startpunkt og -vinkel. Robotten starter i punktet  $(-7, 3)$  med startvinklen  $\theta_0 = \pi/4$ , og banen for simuleringen plottes (se figur 4.18 side 35).





Figur 4.18: Robotens bane for startpunktet  $(-7,3)$  med vinklen  $\pi/4$

På figur 4.18 side 35 ses robotens bane efter parametertuning for startpunktet  $(-7,3)$  med vinklen  $\theta = \pi/4$ . Robotten kører ind mod x-aksen og følger herefter akseren til punktet  $(0,0)$  med vinklen  $\theta = 0$ .

Denne metode (styring til x-aksen med hastighedsregulering) gør, at robotten kører hurtigere jo længere væk fra referencen den befinder sig. Dette bevirker, at robotten kan køre til et bestemt punkt på x-aksen og sætte farten ned, jo tættere på punktet den kommer.

## 4.3 Waypointstyring

Der indføres nu waypointstyring. Det betyder, at der skal køres efter et punkt (waypoint) i koordinatsystemet i stedet for ad x-aksen som før. Der er forskellige måder dette kan gøres på. Den ene måde er vinkelstyring, så den direkte vinkel mellem robotten og punktet bruges som referencevinkel. Ved vinkelstyring kører robotten med konstant hastighed. En anden måde er at styre efter punktets koordinater. Derfor indføres hastighedsstyring. Man kan også indføre det polære koordinatsystem, så der styres på afstanden til punktet og vinkler (som beskrevet i afsnit 2.2 side 12). I denne model styres der også på robotens hastighed.

### 4.3.1 Waypointstyring uden hastighedsregulering

For at robotten kan finde hen til en waypoint med konstant hastighed er det kun nødvendigt at styre på den vinkel, linien mellem robotten og punktet danner med x-aksen. Denne vinkel  $\theta_{ref}$  findes som:

$$\theta_{ref} = \arctan\left(\frac{\Delta y}{\Delta x}\right) \quad (4.11)$$

hvor  $\Delta y$  og  $\Delta x$  ses på figur 2.8 side 13. I simuleringen benyttes MATLAB funktionen `atan2`. For at få robotten til at bevæge sig i retning af punktet benyttes igen LQR regulering, og  $\omega$  findes som:

$$\omega = -L \cdot (\theta - \theta_{ref})$$

hvor  $\theta$  er vinklen, robotten danner med x-aksen.

Det undersøges, hvordan robotten kører til tre forskellige punkter (waypoints) i en bestemt rækkefølge fra et valgt startpunkt og startvinkel. Hertil benyttes SIMULINK modellen, som ses på figur 4.1 side 20. De tre waypoints og vinklen gemmes i MATLAB som globale variable, så der i MATLAB funktionen `THwaypointkontrol.m` (se bilag B.3) kan skiftes, hvilken reference robotten skal køre efter. For hver iteration i SIMULINK beregnes på ny vinklen mellem robotten og punktet ved hjælp af MATLAB funktionen `atan2`. Herefter benyttes vinkelstyringen som beskrevet i model 1 (se afsnit 2.1.1 side 6) til at få robotten til at køre hen til referencepunktet. Der sættes en tolerance for, hvor tæt robotten skal komme på punktet, før der skiftes til nyt waypoint. Denne tolerance har jeg sat til 5 cm.

Et eksempel på waypointstyring uden hastighedsregulering ses på figur 4.19 side 37, hvor robotten starter i punktet (2, 3) med vinklen  $\pi/4$ . Waypointene er valgt til:

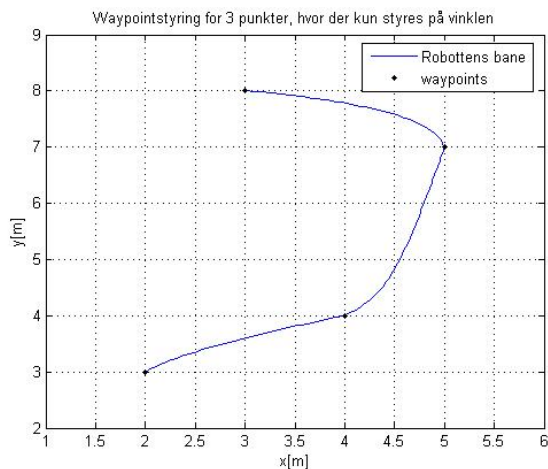
$$(4, 4) \quad (5, 7) \quad (3, 8)$$

Den første referencevinkel bliver derfor:

```
atan2(2,1) =
```

```
0.4636
```

På figur 4.19 side 37 ses det, at robotten kommer hen til hvert af de tre waypoints og stopper, når det sidste waypoint er nået. Når robotten er inden for en



Figur 4.19: *Vinkelstyring for 3 punkter*

radius af 5 cm fra referencepunktet, bliver robotten styret hen mod det næste referencepunkt.

Et andet eksempel ses på figur 4.20 side 38, hvor robotten starter i punktet (2, 3) med vinklen  $\pi/4$ . Waypointene er valgt til:

$$(4, 7) \quad (7, 2) \quad (8, 3)$$

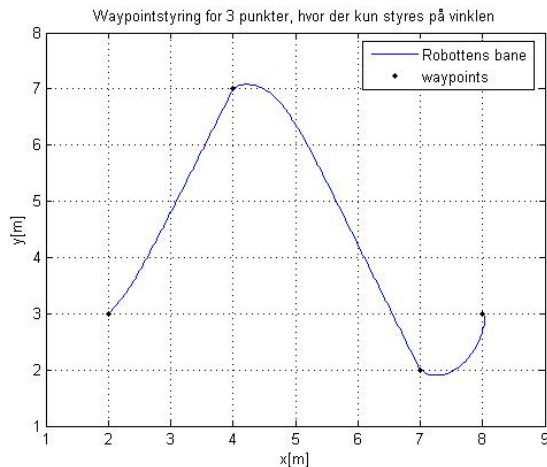
Den første referencevinkel bliver derfor:

$$\text{atan2}(4, 2) =$$

$$1.1071$$

I dette eksempel (se figur 4.20 side 38) kommer robotten også uden problemer hen til alle tre waypoints og stopper ved det sidste, selvom der i dette tilfælde skal drejes meget skarpere end i det andet eksempel (se figur 4.19 side 37).

Denne metode gør det muligt at få robotten til at køre til flere efterfølgende punkter. Robotten kører kun efter referencepunkterne og ikke efter en bestemt bane. Simuleringen viser, at robotten når alle punkter uanset, hvordan de ligger i forhold til hinanden.



Figur 4.20: *Vinkelstyring for 3 punkter*

### 4.3.2 Waypointstyring med hastighedsregulering

Der tilføjes nu hastighedsregulering til waypointstyringen. Dette gør, at man nu både kan styre hen mod punktet og samtidig mod en valgt vinkel (som ikke behøver at være den samme som den direkte vinkel mellem robot og punkt). Kan derfor man komme hen til et waypoint med en slutvinkel. Dette kan gøres både i det kartesiske og polære koordinatsystem.

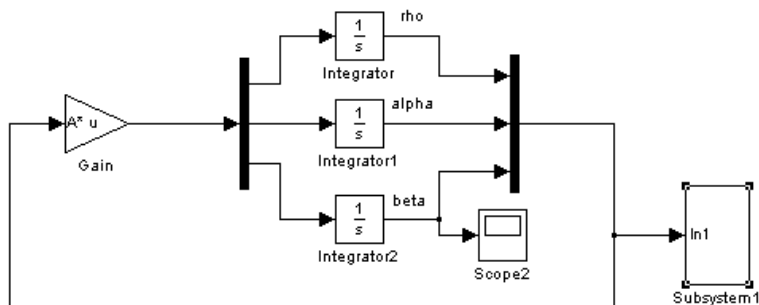
I det kartesiske koordinatsystem kan dette gøres ved at bruge reguleringen beskrevet i afsnit 4.2 side 30, hvor der bare lægges mindre vægt på styringen af vinklen ( $q_2$  skal have en mindre værdi). Her kan der også indføres et lokalt koordinatsystem, som gør, at der ikke skal ændres på vægten af  $x$  ( $q_1$  skal ikke ændres). I stedet for at ændre på parametrene bruger man det polære koordinatsystem, som beskrevet i afsnit 2.2 side 12.

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho & 0 & 0 \\ 0 & -(k_\alpha - k_\rho) & -k_\beta \\ 0 & -k_\rho & 0 \end{bmatrix} \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix} \quad (4.12)$$

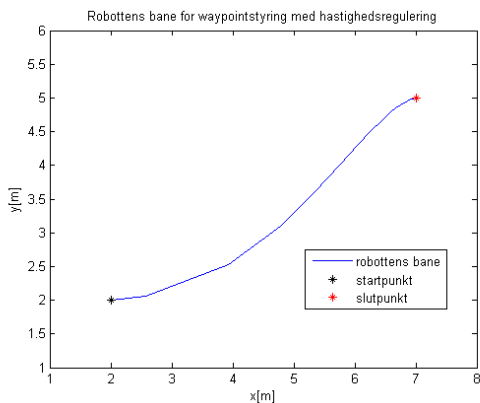
hvor  $k = (k_\rho, k_\alpha, k_\beta) = (3, 8, -1.5)$ .

SIMULINK-modellen for det polære koordinatsystem ses på figur 4.21 side 39. Denne model benyttes til at få robotten til at køre mod et waypoint og ende

med en bestemt slutvinkel. Et eksempel på dette ses på figur 4.22 side 39, hvor robotten starter i punktet  $(2, 2)$  med vinklen  $\theta_0 = 0$  og kører til waypoint  $(5, 7)$  med slutvinklen  $\theta = 0$ .



Figur 4.21: Simulink-model for det polære koordinatsystem



Figur 4.22: Waypointstyring: Startpunkt i  $(2, 2)$  med vinklen  $\theta_0 = 0$ , slutpunkt i  $(5, 7)$  med vinklen  $\theta = 0$

På figur 4.22 side 39 ses det, at robotten kører hen til punktet og først, når den kommer tæt på punktet begynder den at dreje. Inden den når punktet har den slutvinklen 0.

I denne metode benyttes parametre som beskrevet i [Robots 2004] og banen er jævn.

## 4.4 Liniestyring

Liniestyring er waypointstyring, hvor der tilføjes en linie, som robotten skal køre ad til punktet. Det er det samme som at køre ind ad x-aksen og følge denne til et bestemt punkt, men nu skal det gøres for et vilkårligt punkt og en vilkårlig linie i koordinatsystemet. Det er derfor vigtigt både at nå punktet med en bestemt vinkel (som ved waypointstyring se afsnit 4.3.2 side 38), og at vinklen opnås før punktet.

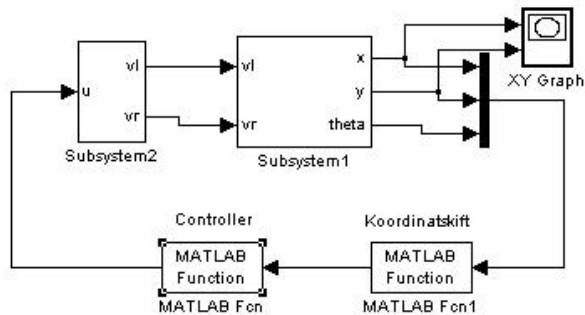
Hvis robotten skulle følge en anden linie end x-aksen, ville det være nødvendigt at ændre på parametrene i LQR hver gang der skal køres efter en bestemt linie. Der indføres nu det lokale koordinatsystem (se afsnit 2.1.4 side 11), så de samme parametre som ved styring til x-aksen kan bruges.

I SIMULINK-modellen (se figur 4.23 side 41) indsættes der en ekstra funktion foran regulatoren. Funktionen transformerer robottens koordinater fra globale til lokale koordinatsystem ved hjælp af basisskifte matricen (bilag B.4.1 side 100 `koordinatskift.m`). Inputtet til regulatoren er de lokale koordinater, og robottens styres mod nulpunktet  $(x_l, y_l, \theta_l) = (0, 0, 0)$  (se bilag B.4 side 100 `liniestyringkontrol.m`). Dette gør, at der kan bruges de samme parametre i LQR-reguleringen, som blev fundet ved parametertuning for styring til x-aksen (se afsnit 4.2 side 30). Dermed kan samme MATLAB algoritme som før benyttes (se bilag B.2 side 98 `hastighedkontrol.m`).

Der er vist et eksempel, hvor robotten skal køre til punktet  $(10, 9)$  ad linien med hældningen  $\pi/8$  fra startpunktet  $(2, 3)$  med vinklen  $\pi/4$ . Robottens bane og x-aksen i det lokale koordinatsystem (referencelinien) plottes sammen, hvoraf der ses, hvor godt robotten følger linien. Plottet ses på figur 4.24 side 42.

På figur 4.24 side 42 ses det, at robotten finder ind til den lokale x-akse (referencelinien) og følger den til det lokale nulpunkt (referencepunktet). Der er indført et lokalt koordinatsystem, og det er nu også muligt at få robotten til at følge den globale y-akse med de samme parametre som til x-aksen. Dette ses i et eksempel på figur 4.25 side 42. Hvis det lokale koordinatsystem ikke var indført, skulle  $Q$  og  $R$  ændres. I dette eksempel skulle vægten ligge på at få  $x$  mod 0 og  $\theta$  mod  $\pi/2$ , men dette er ikke nødvendigt med det lokale koordinatsystem.

I denne metode kan der også (som ved waypointstyring) køres efter tre forskellige punkter ad tre forskellige linier. Der indføres igen globale variable, så MATLAB funktionen (se bilag B.5 side 101 `hastighedkontrol2.m`) kan skifte mellem referencepunkterne.

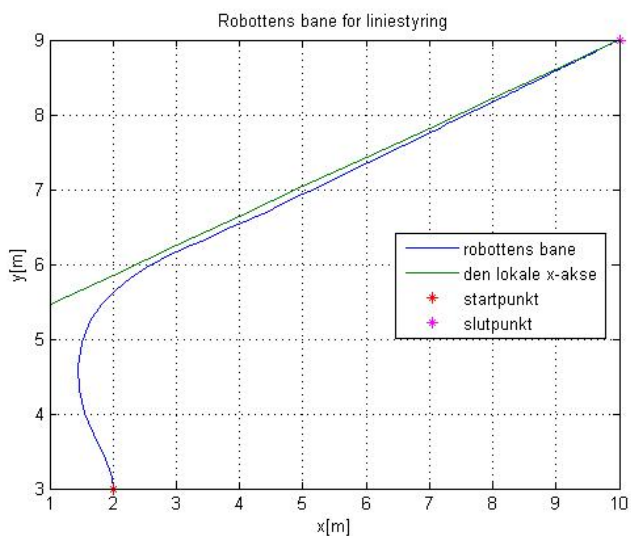


Figur 4.23: Simulink model, hvor der er indsat koordinatskift, så koordinaterne transformeres fra globalt til lokalt koordinatsystem

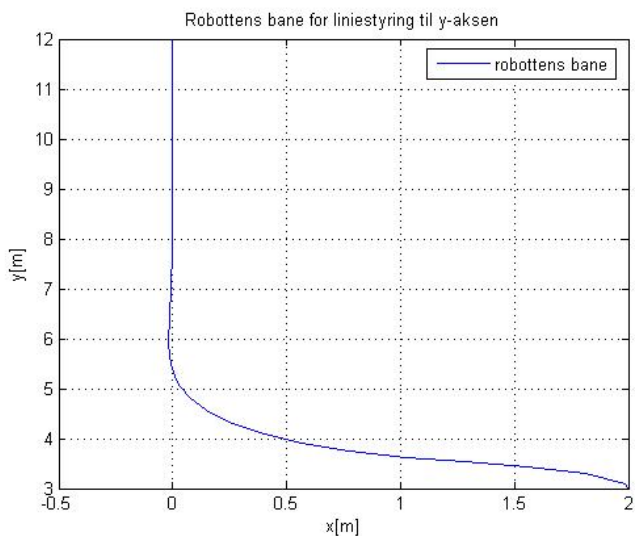
Jeg har valgt punkterne (4,6), (8,6) og (5,4) med referencevinklerne på hhv.  $\pi/4$ ,  $-\pi/6$  og  $-7\pi/8$ . Plot for robotens bane og de tre referencelinier ses på figur 4.26 side 43

På figur 4.26 side 43 ses robotens bane og referencelinier. Robotten kører som den skal. Den følger linierne hen til punkterne, og når den er inden for en afstand af 5 cm fra punkterne, skifter den til ny reference og stopper ved sidste punkt.

I denne metode kan robotten køre ad en referencelinie til et referencepunkt. Ved at få robotten til at køre efter flere efterfølgende referencelinier og -punkter, er det nu muligt at få den til at køre ad en bestemt bane og ikke kun efter punkter. Metoden er baseret på indførelse af det lokale koordinatsystem.

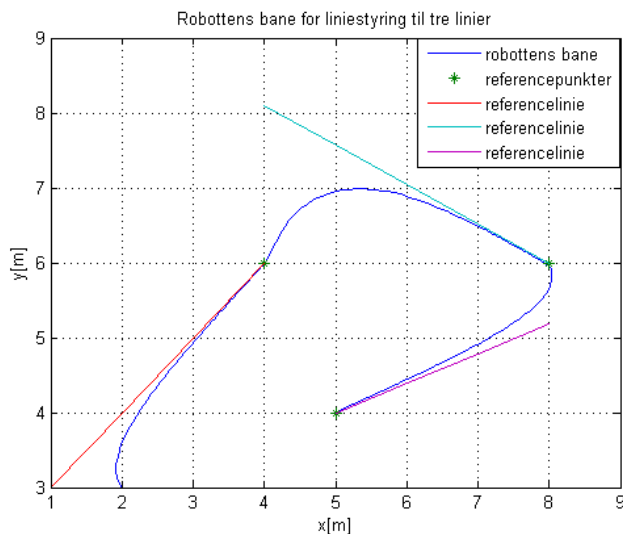


Figur 4.24: Robottens bane for startpunktet  $(2,3)$  med vinklen  $\pi/4$  og referencpunktet  $(10,9)$  med vinklen  $\pi/8$



Figur 4.25: Robottens bane med startpunktet  $(2,3)$  med vinklen  $\pi/4$  for liniestyring til y-aksen





Figur 4.26: Robottens bane for kørsel med liniestyling til tre referencelinier

## 4.5 Metodesammenligning

I dette afsnit sammenlignes de forskellige metoder, jeg har arbejdet med: waypointstyring og liniestyling.

Jeg ser på to eksempler:

- Kørsel gennem en døråbning
- Kørsel i en gang

Gennem disse to eksempler vil jeg se på, hvilke fordele og ulemper de to metoder har.

Til sammenligning benyttes waypointstyring uden hastighedsregulering. Dette skyldes, at der køres efter flere efterfølgende referencepunkter, og der skal skiftes til ny reference i en valgt afstand mellem robotten og referencepunktet. I dette tilfælde vil waypointstyring med hastighedsregulering ikke opnå den slutvinkel som ville være fordel for denne styring, og derfor benyttes denne metode ikke

til sammenligningen. De to metoder, der sammenlignes, er waypointstyringen uden hastighedsregulering og liniestyling (med hastighedsregulering).

## Kørsel gennem døråbning

Robotten kommer kørende langs en væg og drejer ind gennem en døråbning, hvorefter den skal dreje og køre ned langs væggen på den anden side af døråbningen. Døren er 1 m bred, og der placeres tre referencepunkter/linier på banen, et på hver side af døråbningen og et for enden af væggen. Afstanden mellem robotten og væggen, når den kører langs med væggen, ændres. Jeg vil se på, hvilken forskel der er på robotens bane, når denne afstand ændres. Jeg afprøver også forskellige tolerancer for, hvornår der skiftes mellem de forskellige referencepunkter, altså hvor tæt på referencepunktet robotten skal være, før den kører efter en ny reference, og ser på hvordan robotens bane ændres. Jeg sammenligner både den bane robotten kører og hvor lang tid det tager robotten at gennemføre banen med de to metoder. Dette gøres for fire forskellige baner.

### Døråbning 1

Robotten kører 2 m fra væggen og skifter reference 0.5 m fra referencepunktet. Referencepunkterne er:

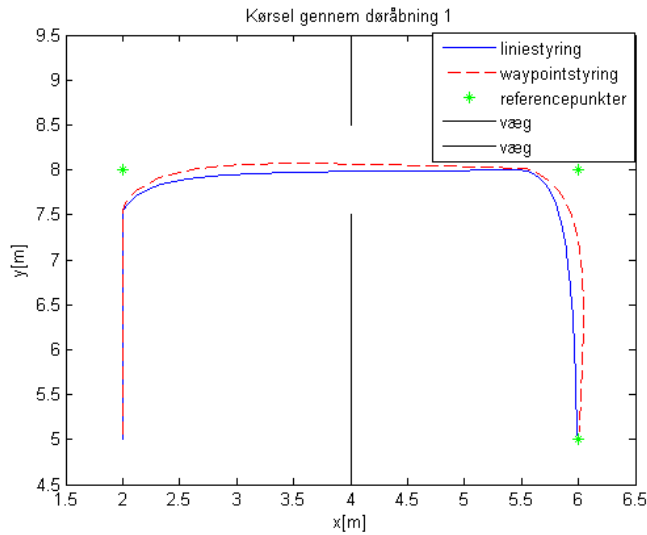
$$(2, 8) \quad (6, 8) \quad (6, 5)$$

og referencevinklerne til liniestylingen er:

$$\pi/2; \quad 0; \quad -\pi/2$$

Robotten starter i punktet (2, 5) med vinklen  $\pi/2$ . Robotten kører efter første referencepunkt, og når den er inden for en afstand af 0.5 m fra referencepunktet, fortsætter den efter det næste referencepunkt. Sådan fortsættes der til sidste referencepunkt og robotten stopper, når den er inden for en afstand af 5 cm fra det sidste referencepunkt. Dette gøres med både waypointstyring og liniestyling. Robotens bane, væggen og referencepunkterne plottes for både waypointstyring og liniestyling (se figur 4.27 side 45).

På figur 4.27 side 45 ses det, at robotten kommer gennem døråbningen uden at ramme dørkarmen for både waypointstyring og liniestyling. Liniestylingen drejer skarpere end waypointstyringen, dvs. waypointstyringen bruger mere plads til at dreje på. Det tager robotten 24.0968 sekunder at gennemføre banen med liniestyling og 18.7409 sekunder med waypointstyringen. Grunden til, at waypointstyringen er hurtigst, kan være, at den kører med en konstant hastighed



Figur 4.27: *Robottens bane for kørsel gennem en døråbning 1*

på  $0.5 \text{ m/s}$ . Liniestyringen regulerer på hastigheden og derfor kører robotten langsommere jo tættere den kommer på referencelinien.

## Døråbning 2

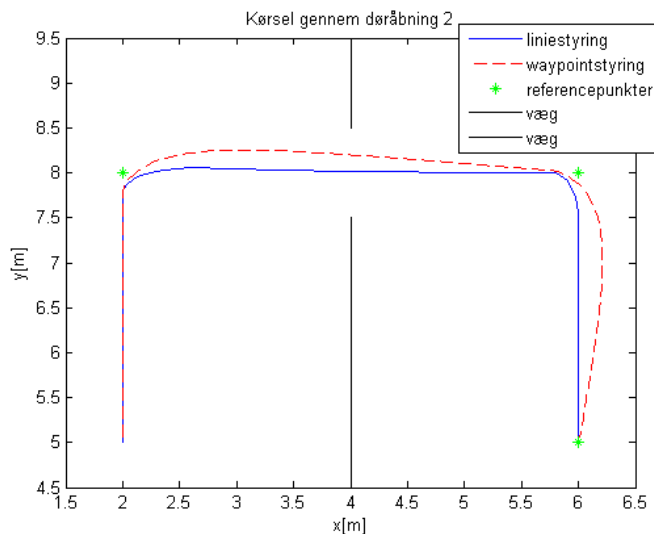
Robotten kører  $2 \text{ m}$  fra væggen og skifter reference  $0.25 \text{ m}$  fra referencepunktet. Referencepunkterne er:

$$(2, 8) \quad (6, 8) \quad (6, 5)$$

og referencevinklerne til liniestyringen er:

$$\pi/2; \quad 0; \quad -\pi/2$$

Robotten starter i punktet  $(2, 5)$  med vinklen  $\pi/2$ . Robotten kører efter første referencepunkt, og når den er inden for en afstand af  $0.25 \text{ m}$  fra referencepunktet, fortsætter den efter det næste referencepunkt. Sådan fortsættes der til sidste referencepunkt, og robotten stopper, når den er inden for en afstand af  $5 \text{ cm}$  fra det sidste referencepunkt. Dette gøres med både waypointstyring og liniestyring. Robottens bane, væggen og referencepunkterne plottes for både waypointstyring og liniestyring (se figur 4.28 side 46).



Figur 4.28: Robotens bane for kørsel gennem en døråbning 2

På figur 4.28 side 46 ses det, at robotten kører igennem døråbningen uden at ramme dørkarmen. Det ses (som ved døråbning 1) at waypointstyringen bruger mere plads for at dreje. Liniestyringen kører igennem midt i døråbningen, hvorimod waypointstyring kører tættere på den ene side af døråbningen fordi den først begynder at dreje 25 cm fra referencepunktet. Ligesom i døråbning 1 er kommer robotten hurtigere gennem banen med waypointstyring. Det tager robotten 19.8000 sekunder med waypointstyring og 28.1865 sekunder med liniestyring at gennemføre banen. Forskellen mellem tiderne er her meget større end for døråbning 1. Robotten er næsten 50 % langsommere med liniestyring end med waypointstyring, hvor den kun er ca. 30 % langsommere i døråbning 1.

### Døråbning 3

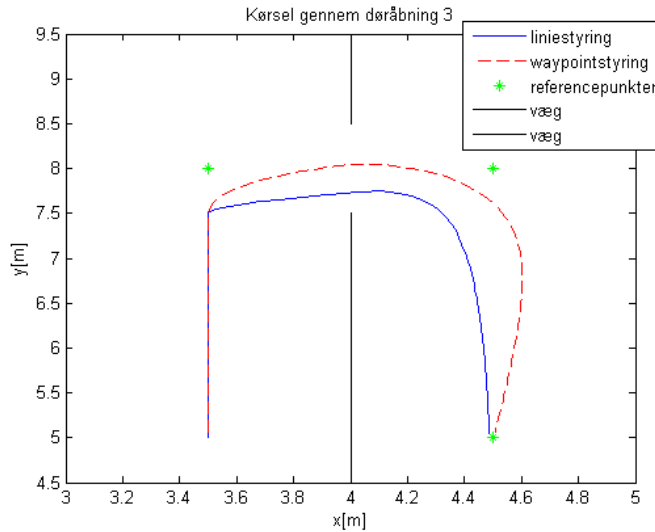
Robotten kører 0.5 m fra væggen og skifter reference 0.5 m fra referencepunktet. Referencepunkterne er:

$$(3.5, 8) \quad (4.5, 8) \quad (4.5, 5)$$

og referencevinklerne til liniestyringen er:

$$\pi/2; \quad 0; \quad -\pi/2$$

Robotten starter i punktet  $(3.5, 5)$  med vinklen  $\pi/2$ . Robotten kører efter første referencepunkt, og når den er inden for en afstand af  $0.5\text{ m}$  fra referencepunktet, fortsættes efter det næste referencepunkt. Sådan fortsættes der til sidste referencepunkt, og robotten stopper, når den er inden for en afstand af  $5\text{ cm}$  fra det sidste referencepunkt. Dette gøres med både waypointstyring og liniestyling. Robottens bane, væggen og referencepunkterne plottes for både waypointstyring og liniestyling (se figur 4.29 side 47).



Figur 4.29: Robottens bane for kørsel gennem en døråbning 3

På figur 4.29 side 47 ses det, at robotten med liniestyling ikke kommer hen til anden referencelinie før den kører efter tredje referencelinie. Robotten kommer i dette tilfælde igennem midten af døråbningen med waypointstyringen. Dette er igen fordi waypointstyringen bruger mere plads til at dreje på, og det er i dette tilfælde en fordel. Det tager robotten  $20.2927$  sekunder med liniestyling og  $13.1247$  sekunder med waypointstyring at gennemføre banen. Liniestylingen igen ca.  $50\%$  langsommere end waypointstyringen.

#### Døråbning 4

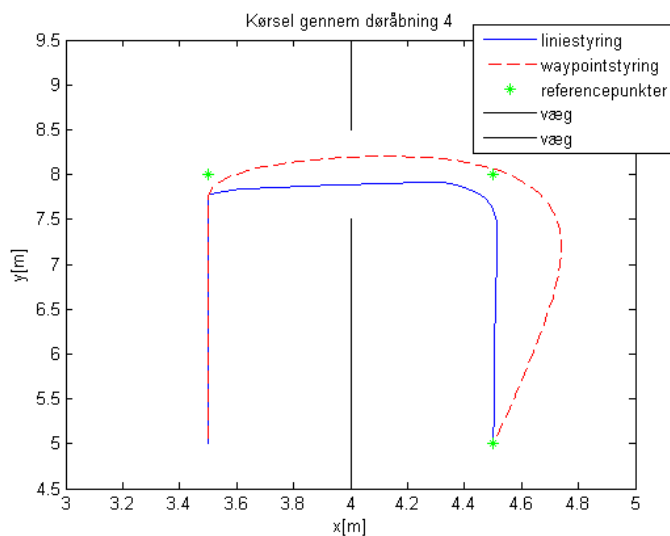
Robotten kører  $0.5\text{ m}$  fra væggen og skifter reference  $0.25\text{ m}$  fra referencepunktet. Referencepunkterne er:

$$(3.5, 8) \quad (4.5, 8) \quad (4.5, 5)$$

og referencevinklerne til liniestylingen er:

$$\pi/2; \quad 0; \quad -\pi/2$$

Robotten starter i punktet  $(3.5, 5)$  med vinklen  $\pi/2$ . Robotten kører efter første referencepunkt, og når den er inden for en afstand af  $0.25 \text{ m}$  fra referencepunktet, fortsættes efter det næste referencepunkt. Sådan fortsættes der til sidste referencepunkt, og robotten stopper, når den er inden for en afstand af  $5 \text{ cm}$  fra det sidste referencepunkt. Dette gøres med både waypointstyring og liniestyling. Robottens bane, væggen og referencepunkterne plottes for både waypointstyring og liniestyling (se figur 4.30 side 48).



Figur 4.30: Robottens bane for kørsel gennem en døråbning 4

På figur 4.30 side 48 ses det, at robotten med liniestyling kører gennem i midten af døråbningen og følger den anden referencelinie. Robotten bruger igen mere plads til at dreje med waypointstyringen og kommer derfor ikke igennem midten af døråbningen fordi den først begynder at dreje  $25 \text{ cm}$  fra referencepunktet. Den har også et oversving, når der skiftes fra det andet referencepunkt til tredje referencepunkt. Det tager robotten  $24.2068$  sekunder med liniestyling og  $14.1000$  sekunder med waypointstyring at gennemføre banen. Robotten bruger over  $70 \%$  længere tid at gennemføre banen med liniestyling end med waypointstyringen.

Simuleringerne for døråbning 1, 2, 3, og 4 sammenlignes i tabel 4.1 på side 53.

## Kørsel i en gang

Robotten skal følge en kvadratisk bane med vægge på begge sider, denne bane kaldes en gang (se figur 4.31 side 50). På gangen er der fire referencepunkter/linier, som robotten skal køre efter. Jeg vil sammenligne, hvordan robotten kører ved forskellige bredder af gangen og hvor tæt på et referencepunkt, robotten skal være, før der skiftes til næste referencepunkt. Jeg sammenligner både den bane, robotten kører og hvor lang tid, det tager robotten at gennemføre banen med de to metoder. Dette gøres for tre forskellige baner.

### Gang 1

Gangen er 1 m bred og der køres efter nyt referencepunkt 0.5 m inden punktet er nået. Referencepunkterne er:

$$(2, 5) \quad (8, 5) \quad (8, 2) \quad (2, 2)$$

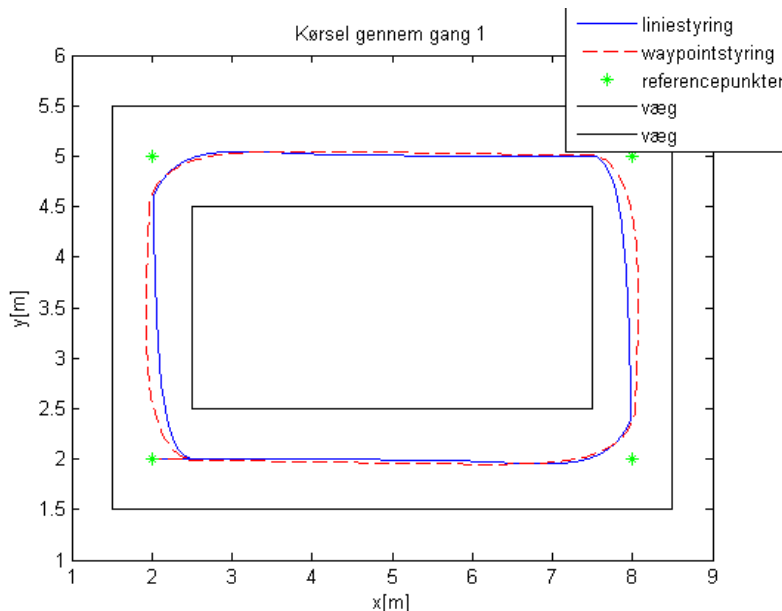
og referencevinklerne til liniestylingen er:

$$\pi/2; \quad 0; \quad -\pi/2; \quad -\pi$$

Robotten starter i punktet (2.5, 2) med vinklen 0. Robotten kører efter første referencepunkt, og når den er inden for en afstand af 0.5 m fra referencepunktet fortsætter den efter det næste referencepunkt. Sådan fortsættes der til sidste referencepunkt, og robotten stopper, når den er inden for en afstand af 5 cm fra det sidste referencepunkt. Dette gøres med både waypointstyring og liniestyling. Robottens bane, væggen og referencepunkterne plottes for både waypointstyring og liniestyling (se figur 4.31 side 50).

På figur 4.31 side 50 ses robottens bane for waypointstyring og liniestyling i gang 1. Der er ikke stor forskel på banerne for waypointstyring og liniestyling, men waypointstyringen kører lidt tættere på referencelinien for liniestylingen. Gangen er 1 m bred og derfor er det ikke noget problem for robotten at komme rundt uden at støde ind i væggene for hverken waypointstyring eller liniestyling.

Robotten bruger 32.9233 sekunder for at gennemføre banen med liniestyling og 34.8803 sekunder med waypointstyring. Robotten med liniestylingen er her ca. 5 % hurtigere end den med waypointstyringen til at gennemføre banen. Denne forskel på tiderne og banerne, robotten kører, er ikke signifikant. Derfor vil jeg lave samme undersøgelser i en smallere gang, så jeg kan se, med hvilken af metoderne robotten følger banen bedst.



Figur 4.31: Gennem gang med 4 referencepunkter

## Gang 2

Gangen er  $0.5\text{ m}$  bred og der køres efter nyt referencepunkt  $0.5\text{ m}$  inden punktet er nået. Referencepunkterne er:

$$(2, 5) \quad (8, 5) \quad (8, 2) \quad (2, 2)$$

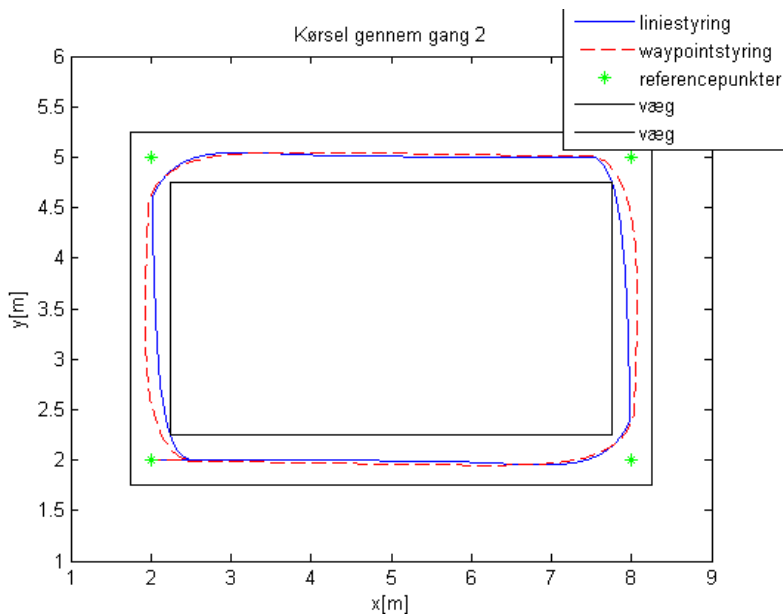
og referencevinklerne til liniestyningen er:

$$\pi/2; \quad 0; \quad -\pi/2; \quad -\pi$$

Robotten starter i punktet  $(2.5, 2)$  med vinklen  $0$ . Robotten kører efter første referencepunkt, og når den er inden for en afstand af  $0.5\text{ m}$  fra referencepunktet, fortsætter den efter det næste referencepunkt. Sådan fortsættes der til sidste referencepunkt, og robotten stopper, når den er inden for en afstand af  $5\text{ cm}$  fra det sidste referencepunkt. Dette er samme bane som ved gang 1, men her er gangen kun  $0.5\text{ m}$  bred. Robotten kører banen med både waypointstyring og liniestyling. Robottens bane, væggen og referencepunkterne plottes for både waypointstyring og liniestyling (se figur 4.32 side 51).

På figur 4.32 side 51 ses det, at robotten med liniestyling har svært ved at





Figur 4.32: *Gennem gang med 4 referencepunkter*

køre gennem gangen uden at støde ind i væggene, hvorimod den med waypointstyringen ingen problemer har med at komme igennem. Det tager igen robotten 32.9233 sekunder at komme igennem banen med liniestyring og 34.8803 sekunder med waypointstyringen. Robotten kører hurtigere med liniestyring, men banen er bedre med waypointstyring.

### Gang 3

Gangen er 0.5 m bred og der køres efter nyt referencepunkt 0.25 m inden punktet er nået. Referencepunkterne er:

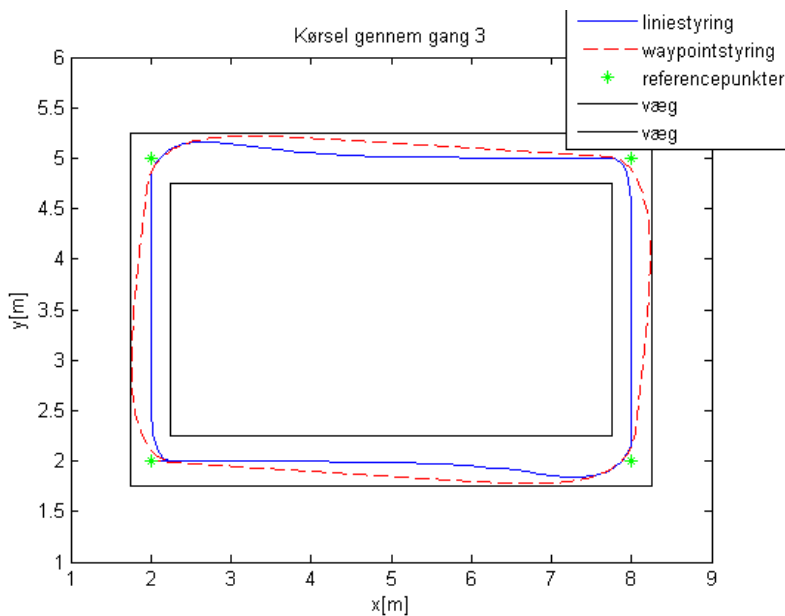
$$(2, 5) \quad (8, 5) \quad (8, 2) \quad (2, 2)$$

og referencevinklerne til liniestyringen er:

$$\pi/2; \quad 0; \quad -\pi/2; \quad -\pi$$

Robotten starter i punktet (2.25, 2) med vinklen 0. Robotten kører efter første referencepunkt, og når den er inden for en afstand af 0.25 m fra referencepunktet,

fortsætter den efter det næste referencepunkt. Sådan fortsættes der til sidste referencepunkt, og robotten stopper, når den er inden for en afstand af 5 cm fra det sidste referencepunkt. Dette gøres med både waypointstyring og liniestyling. Robottens bane, væggen og referencepunkterne plottes for både waypointstyring og liniestyling (se figur 4.33 side 52).



Figur 4.33: Gennem gang med 4 referencepunkter

På figur 4.33 side 52 ses det, at robotten med waypointstyring har svært ved at køre gennem gangen uden at støde ind i væggene, hvorimod den med liniestylingen ingen problemer har med at komme igennem. Det tager robotten 39.1862 sekunder at komme igennem banen med liniestyling og 36.2000 sekunder med waypointstyringen. Robotten kører ca. 8 % langsommere med liniestyling end med waypointstyring, men banen er bedre med liniestyling.

I tabel 4.1 side 53 har jeg lavet et skema, der viser fordele og ulemper ved de to metoder for de forskellige baner.

## Fordele og ulemper

	Waypointstyring		Liniestyring	
	tid	bane	tid	bane
døråbning 1	+	+	÷ (30%)	+
døråbning 2	+	÷	÷ (50%)	+
døråbning 3	+	+	÷ (50%)	÷
døråbning 4	+	÷	÷ (70%)	+
gang 1	÷ (5%)	+	+	+
gang 2	÷ (5%)	+	+	÷
gang 3	+	÷	÷ (8%)	+

Tabel 4.1: Fordele (+) og ulemper (÷) ved waypointstyring og liniestyring for de opstillede eksempler

Det ses på tabel 4.1 side 53, at waypointstyringen i de fleste tilfælde er hurtigst til at gennemkøre banen, mens liniestyringen er mere nøjagtig. Liniestyringen bliver meget langsommere jo mindre tolerancen bliver (altså jo tættere på referencepunktet robotens skal være før der skiftes til nyt referencepunkt). Grunden til dette er, at liniestyring indeholder hastighedsregulering og dette medfører at roboten kører langsommere jo tættere den kommer på referencen.

Valg af metode afhænger af, hvilken bane man skal køre på og hvilke krav der i øvrigt stilles til gennemkørsel af banen (gennemkørselstid, banens udformning osv.).



# Konklusion

---

I dette projekt har jeg opstillet matematiske modeller for mobile robotter. Forskellige former for styring er blevet implementeret, for at få robotten til at køre ad forskellige baner.

Robottens bane kan beskrives ved hjælp af tre ulineære differentiallyigninger i det kartesiske koordinatsystem. Der blev besluttet, at reguleringen af robotten skal udføres ved hjælp af lineære, kontinuerte regulatorer. Derfor lineariseres de ulineære differentiallyigninger, og der opstilles tre modeller i kartesiske koordinater og et i polære koordinater. Alle modeller er beskrevet ved hjælp af tilstandsligninger og afprøvet på simple eksempler. Jeg kan derfor konkludere, at alle modeller virker hensigtsmæssigt.

Disse modeller bliver brugt til at lave forskellige reguleringsalgoritmer til at styre robotten med. Robotten styres ved hjælp af LQR (LINEAR QUADRATIC REGULATOR), der har den fordel at den sikrer at det lukkede-sløjfesystem er stabilt. LQR er i princippet en forstærkningsfaktor i tilbagekoblingssløjfen for tilstandende i modellen, dvs. matrixen  $L$ .  $L$  findes ud fra elementer  $Q$  og  $R$  i den såkaldte kostfunktion  $J$ . Jeg har beregnet og afprøvet LQR for følgende styringer:

- vinkelstyring
- styring på vinklen og  $y$ -koordinaten

- hastighedsstyring

Disse tre styringer er blevet udvidet til waypointstyring og liniestyring. Begge reguleringsmetoder blev afprøvet ved hjælp af simuleringer på syv eksempler. Fordele og ulemper ved reguleringsmetoderne ses i tabel 4.1 side 53. Udfra dette kan jeg konkludere, at waypointstyringen generelt er hurtigere end liniestyringen, men liniestyringen er mere nøjagtig. I et af tilfældene er liniestyringen 70 % langsommere end waypointstyringen. Det endelige valg af reguleringsmetode afhænger af opgaven, og de krav der bliver stillet til robotten.

Forslag til videreudvikling af projektet:

- opstille både modellerne og beregne regulatoren i diskret tid
- indføre navigation af robotten ved hjælp af Kalman filteret
- udvide modellen til at indeholde robotens dynamik

# Litteratur

---

- [Robots 2004] Siegwart, Roland og Nourbakhsh, Illah R. (2004): “*Introduction to Autonomous Mobile Robots: Chapter 3.6 - Motion Control (Kinematic Control)*”, 1. udgave, Massachusetts Institute of Technology, USA.
- [LA 1999] Eising, Jens (1999): “*Lineær Algebra: Kapitel 5.5 - Basissikfte*”, 1. udgave, Institut for Matematik, Danmarks Tekniske Universitet.
- [Control 2006] Franklin, Gene F., Powell, J. David og Emami-Naeini, Abbas (2006): “*Feedback control of Dynamic Systems: Chapter 7.6.2 - Symmetric Root Locus (SRL)*”, 5. udgave, Prentice Hall (Person Education), New Jersey USA.
- [Bak 2000] Bak, Martin (2000): “*Control of Systems with Constraints: Chapter 4.1 - The Mobile Robot + Chapter 3.2.2 - Specification of the Trajectory*”, nov. 2004, Institut for Informatik og Matematisk Modellering, Danmarks Tekniske Universitet.
- [Jannerup 2004] Jannerup, Ole og Sørensen, Paul Haase (2004): “*Reguleringsteknik*”, 3. udgave, Automation, Ørsted, Danmarks Tekniske Universitet.
- [LQR 1995] Dorato, Peter; Abdallah, Chaouki og Cerone, Vito (1995): “*Linear-Quadratic Control: An Introduktion*”, 1. udgave, Prentice-Hall, New Jersey USA.





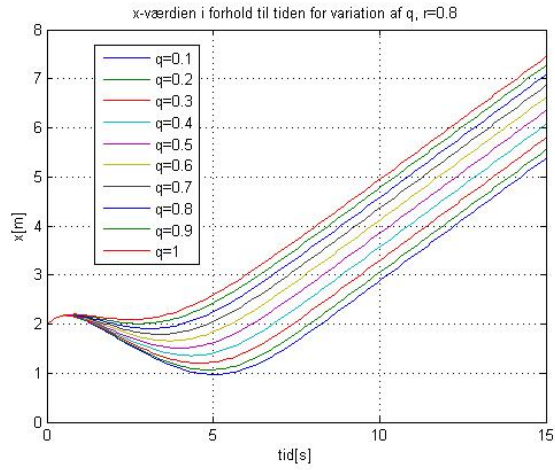
## BILAG A

# Grafer

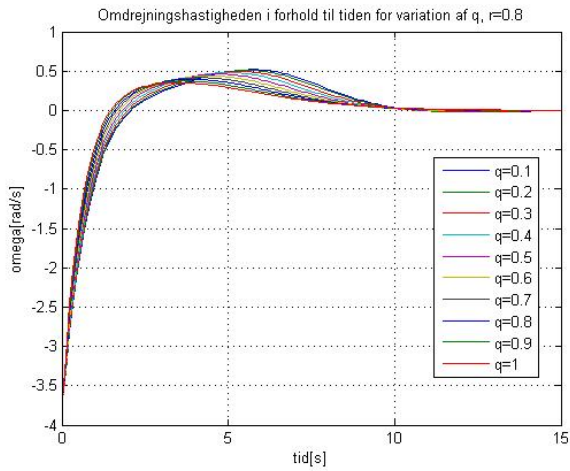
---

### A.1 Styring til x-aksen uden hastighedsregulering

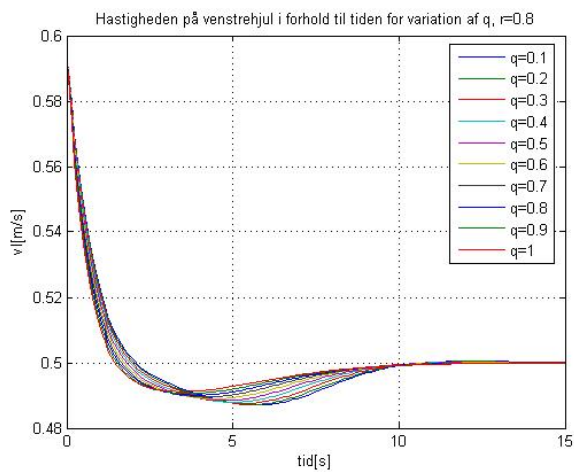
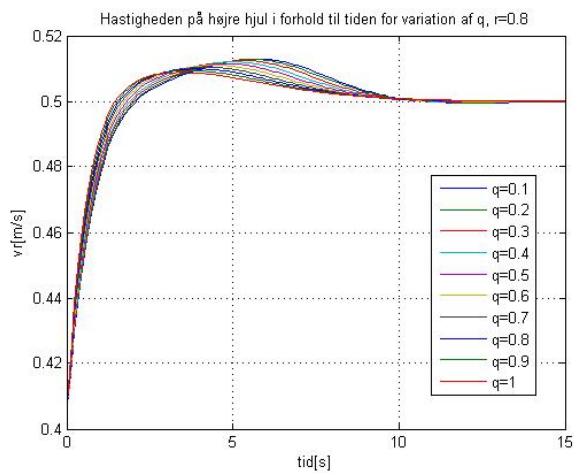
#### A.1.1 Parametertuning for $y > 0$

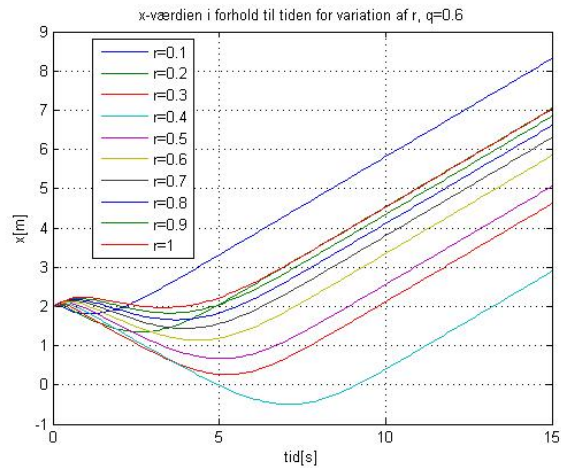


Figur A.1:  $x$  som funktion af tiden for variation af  $q$  og  $r = 0.8$

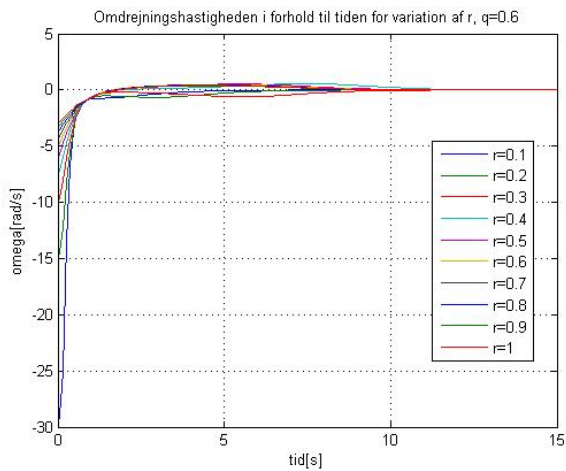


Figur A.2:  $\omega$  som funktion af tiden for variation af  $q$  og  $r = 0.8$

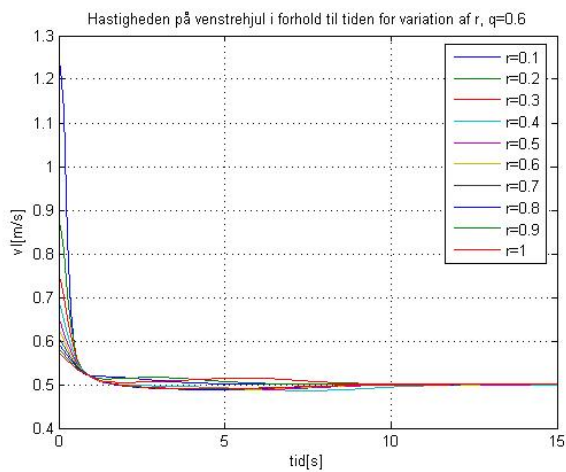
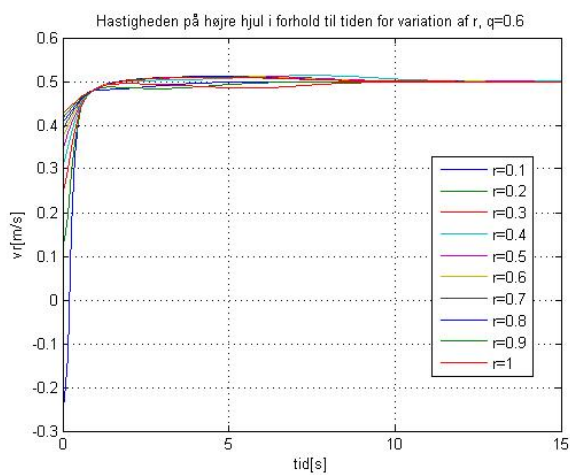
Figur A.3:  $v_l$  som funktion af tiden for variation af  $q$  og  $r = 0.8$ Figur A.4:  $v_r$  som funktion af tiden for variation af  $q$  og  $r = 0.8$

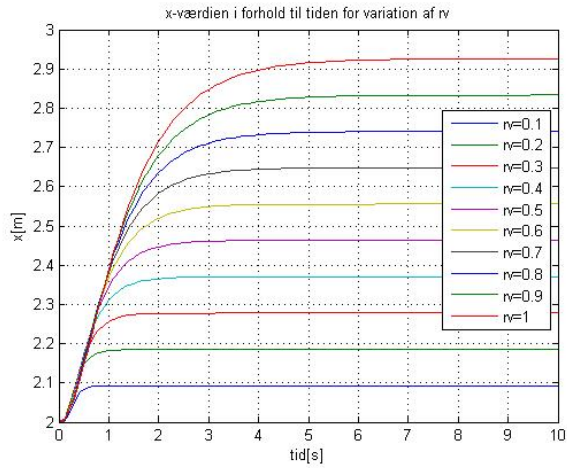


Figur A.5:  $x$  som funktion af tiden for variation af  $r$  og  $q = 0.6$

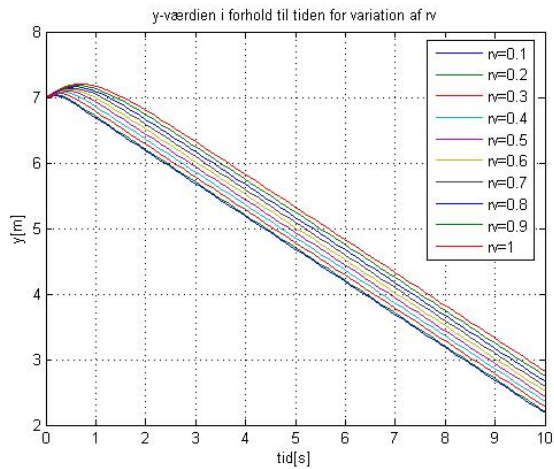


Figur A.6:  $\omega$  som funktion af tiden for variation af  $r$  og  $q = 0.6$

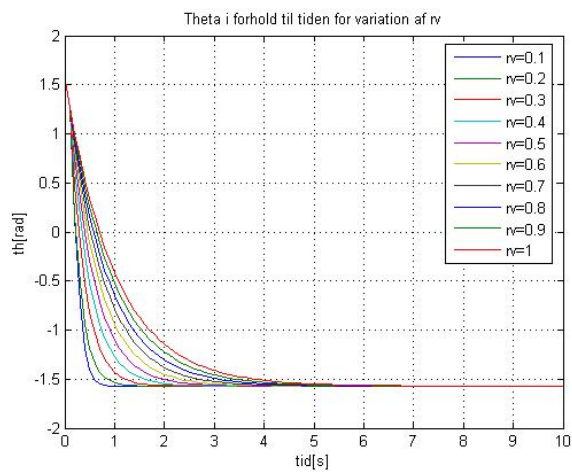
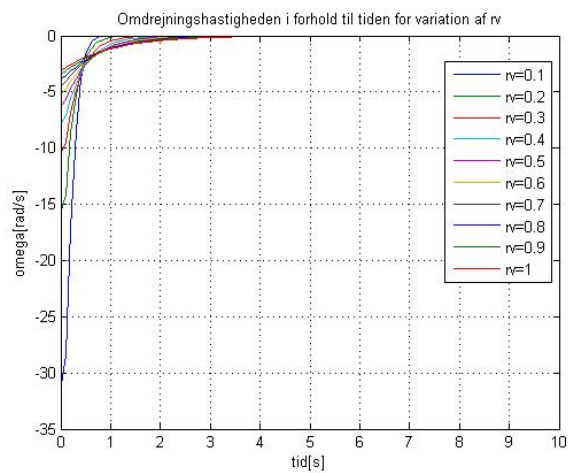
Figur A.7:  $v_l$  som funktion af tiden for variation af  $r$  og  $q = 0.6$ Figur A.8:  $v_r$  som funktion af tiden for variation af  $r$  og  $q = 0.6$



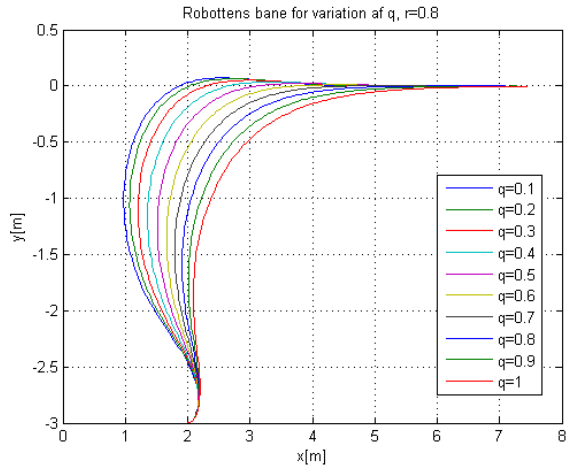
Figur A.9:  $x$  som funktion af tiden for variation af  $r_v$



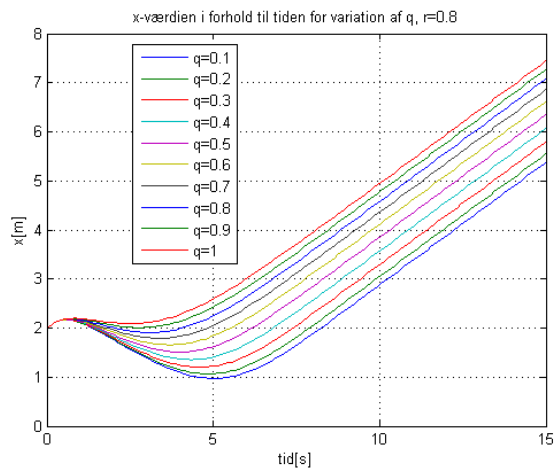
Figur A.10:  $y$  som funktion af tiden for variation af  $r_v$

Figur A.11:  $\theta$  som funktion af tiden for variation af  $r_v$ Figur A.12:  $\omega$  som funktion af tiden for variation af  $r_v$

### A.1.2 Parametertuning for $y < 0$

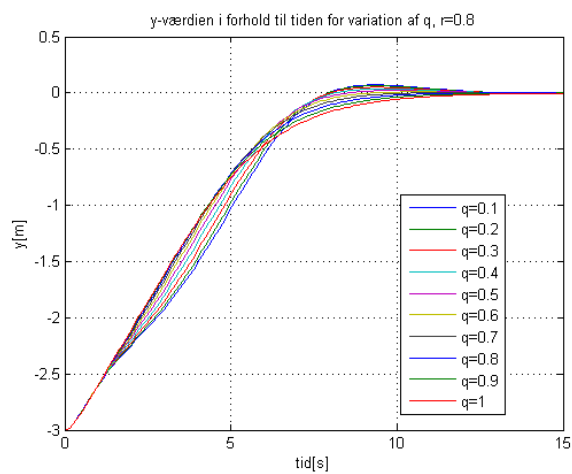
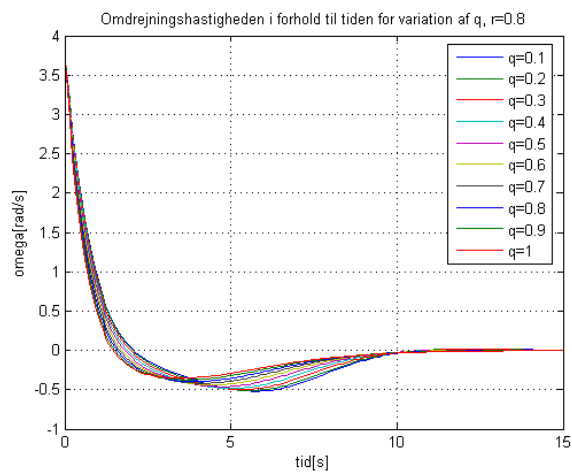


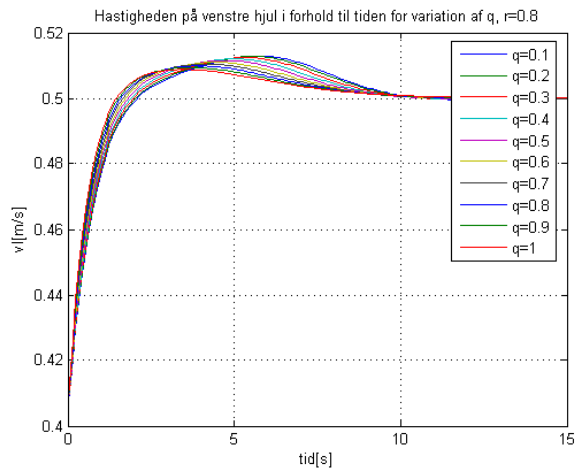
Figur A.13: Robottens bane for variation af  $q$  og  $r = 0.8$



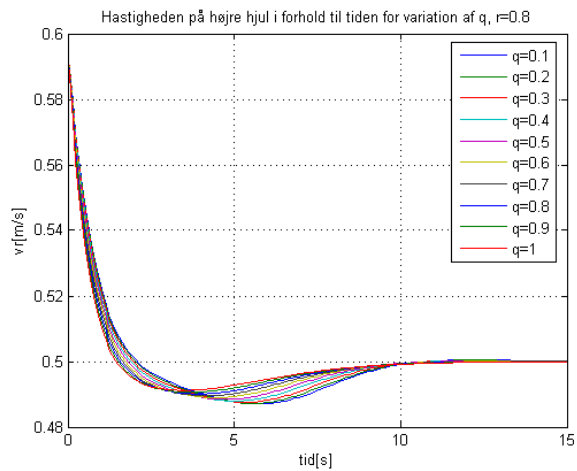
Figur A.14:  $x$  som funktion af tiden for variation af  $q$  og  $r = 0.8$



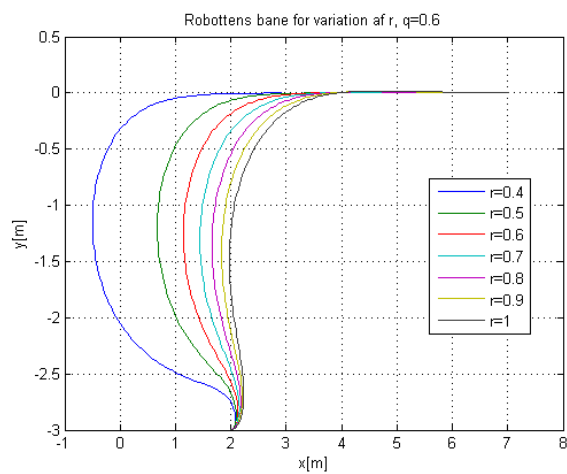
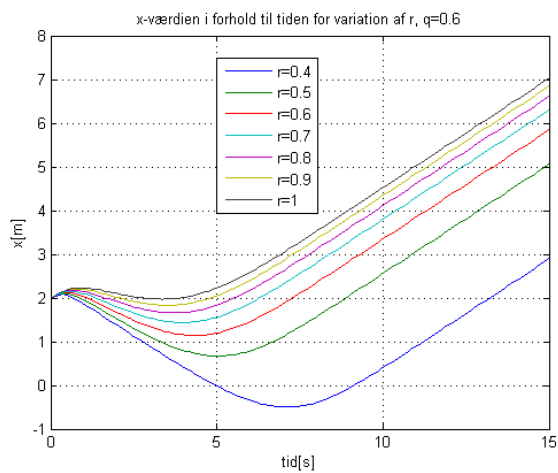
Figur A.15:  $y$  som funktion af tiden for variation af  $q$  og  $r = 0.8$ Figur A.16:  $\omega$  som funktion af tiden for variation af  $q$  og  $r = 0.8$

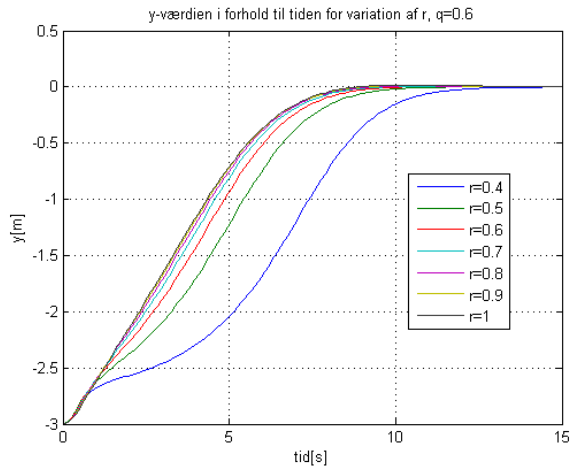


Figur A.17:  $v_l$  som funktion af tiden for variation af  $q$  og  $r = 0.8$

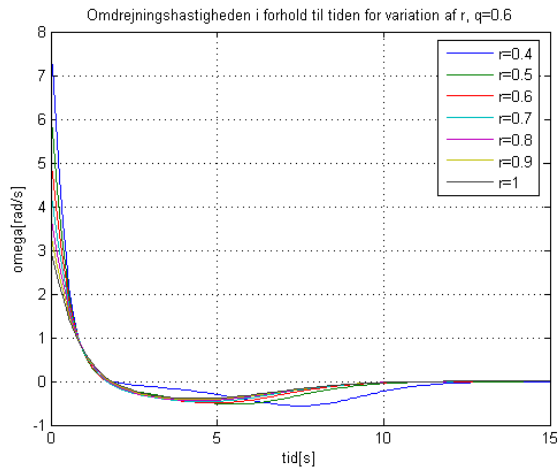


Figur A.18:  $v_r$  som funktion af tiden for variation af  $q$  og  $r = 0.8$

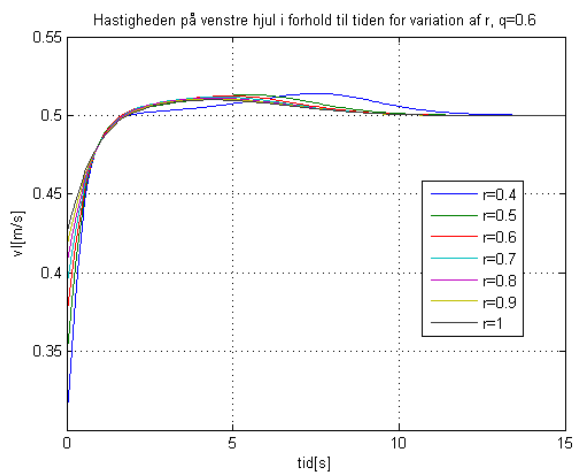
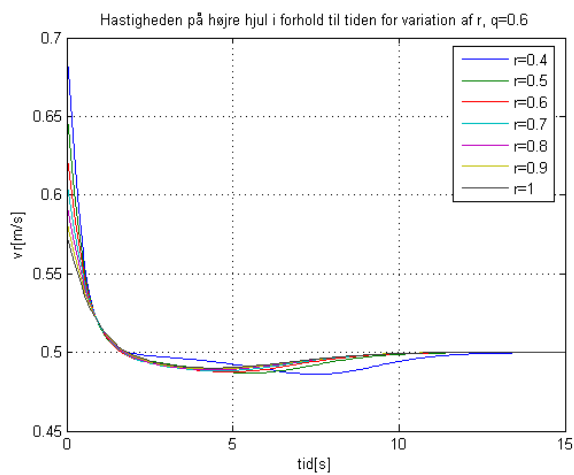
Figur A.19: Robottens bane for variation af  $r$  og  $q = 0.6$ Figur A.20:  $x$  som funktion af tiden for variation af  $r$  og  $q = 0.6$

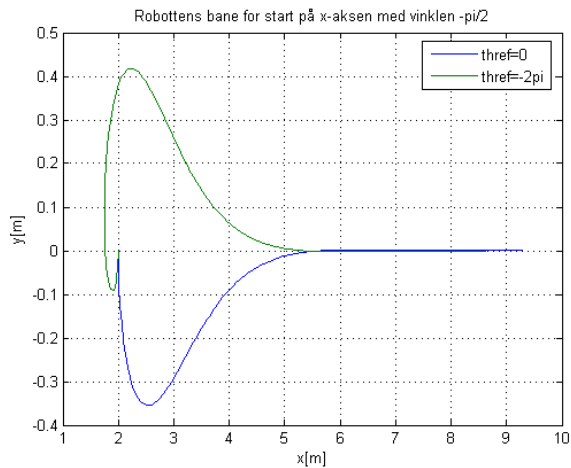


Figur A.21:  $y$  som funktion af tiden for variation af  $r$  og  $q = 0.6$

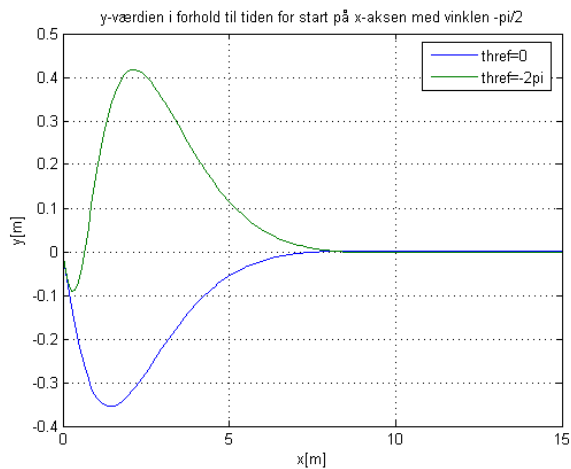


Figur A.22:  $\omega$  som funktion af tiden for variation af  $r$  og  $q = 0.6$

Figur A.23:  $v_l$  som funktion af tiden for variation af  $r$  og  $q = 0.6$ Figur A.24:  $v_r$  som funktion af tiden for variation af  $r$  og  $q = 0.6$



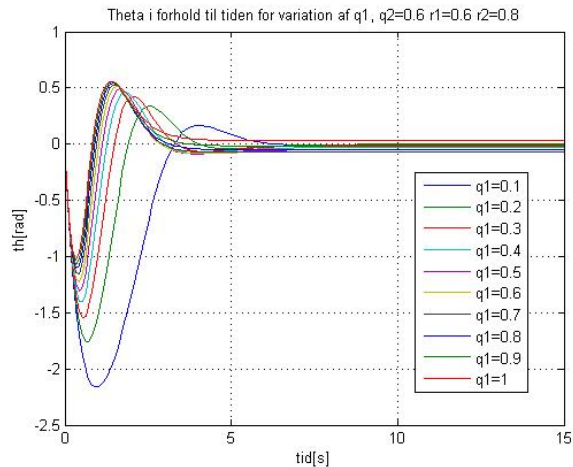
Figur A.25: Robottens bane for start på  $x$ -aksen med vinklen  $\theta_0 = -\pi/2$



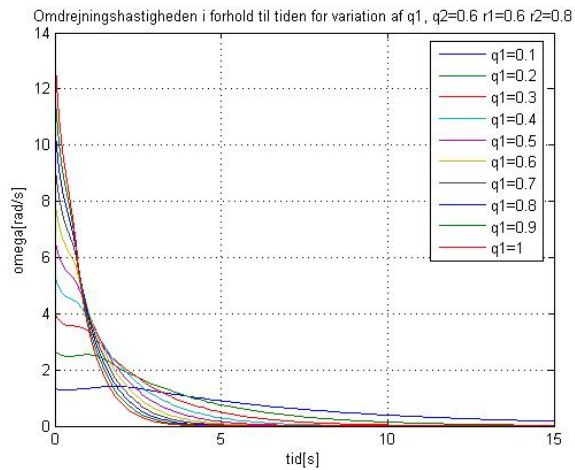
Figur A.26:  $y$  som funktion af tiden for start på  $x$ -aksen med vinklen  $\theta_0 = -\pi/2$

## A.2 Styring til x-aksen med hastighedsregulering

### A.2.1 variation af $q_1$

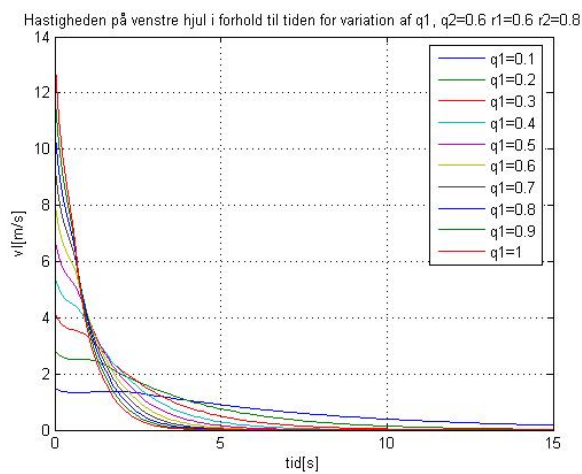


Figur A.27:  $\theta$  som funktion af tiden for variation af  $q_1$ ,  $q_2 = 0.6$ ,  $r_1=0.6$  og  $r_2=0.8$

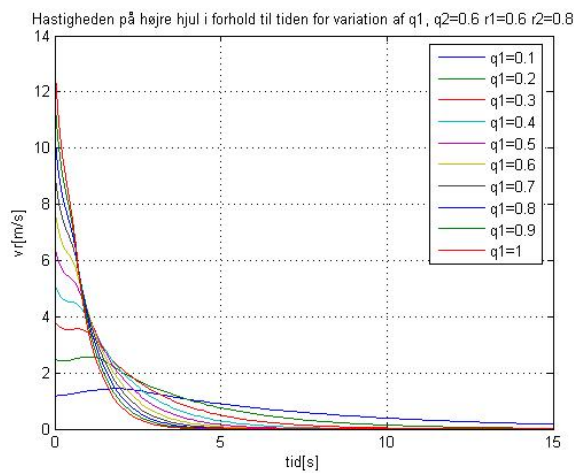


Figur A.28:  $\omega$  som funktion af tiden for variation af  $q_1$ ,  $q_2 = 0.6$ ,  $r_1=0.6$  og  $r_2=0.8$

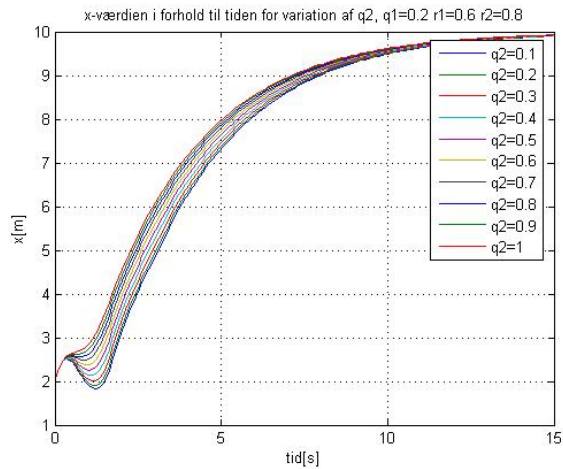




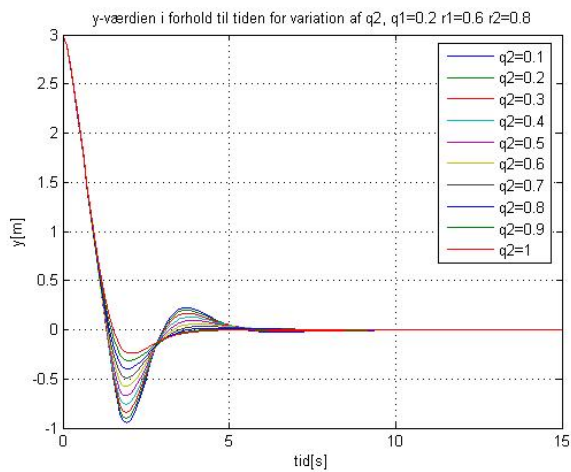
Figur A.29:  $v_l$  som funktion af tiden for variation af  $q_1$ ,  $q_2 = 0.6$ ,  $r_1=0.6$  og  $r_2=0.8$



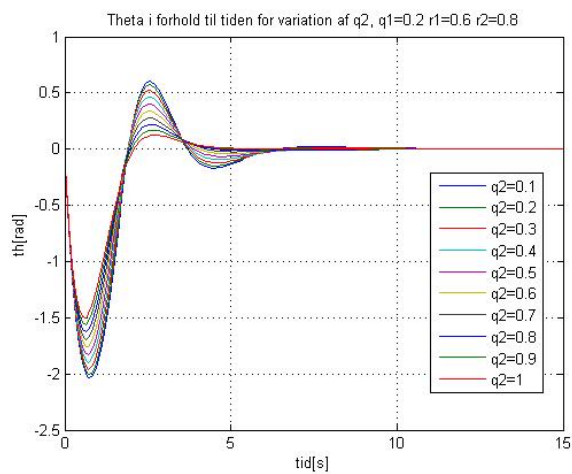
Figur A.30:  $v_r$  som funktion af tiden for variation af  $q_1$ ,  $q_2 = 0.6$ ,  $r_1=0.6$  og  $r_2=0.8$

A.2.2 variation af  $q_2$ 

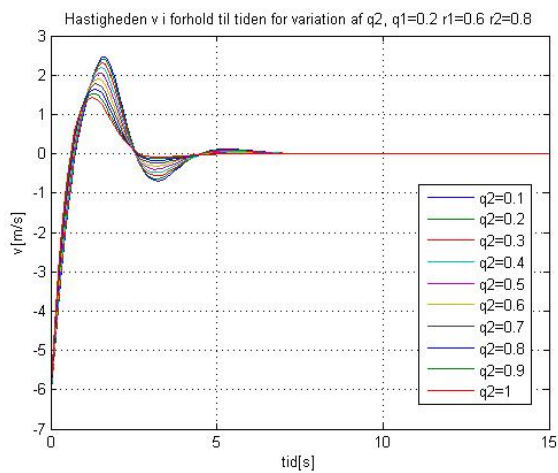
Figur A.31:  $x$  som funktion af tiden for variation af  $q_2$ ,  $q_1 = 0.2$ ,  $r_1 = 0.6$  og  $r_2 = 0.8$



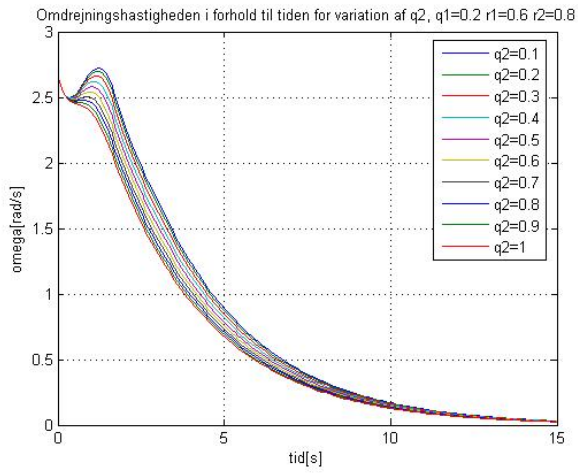
Figur A.32:  $y$  som funktion af tiden for variation af  $q_2$ ,  $q_1 = 0.2$ ,  $r_1 = 0.6$  og  $r_2 = 0.8$



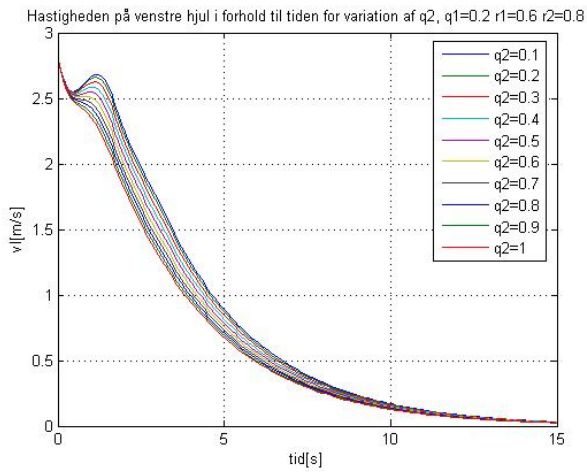
Figur A.33:  $\theta$  som funktion af tiden for variation af  $q_2$ ,  $q_1 = 0.2$ ,  $r_1=0.6$  og  $r_2=0.8$



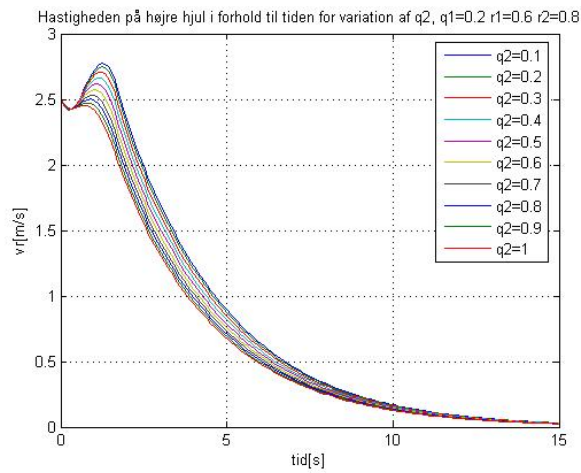
Figur A.34:  $v$  som funktion af tiden for variation af  $q_2$ ,  $q_1 = 0.2$ ,  $r_1=0.6$  og  $r_2=0.8$



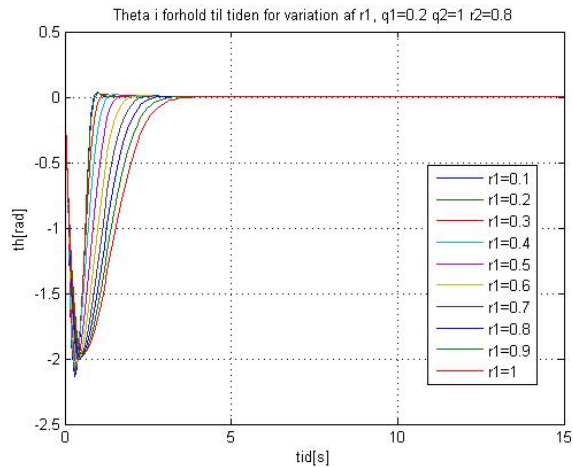
Figur A.35:  $\omega$  som funktion af tiden for variation af  $q_2$ ,  $q_1 = 0.2$ ,  $r_1=0.6$  og  $r_2=0.8$



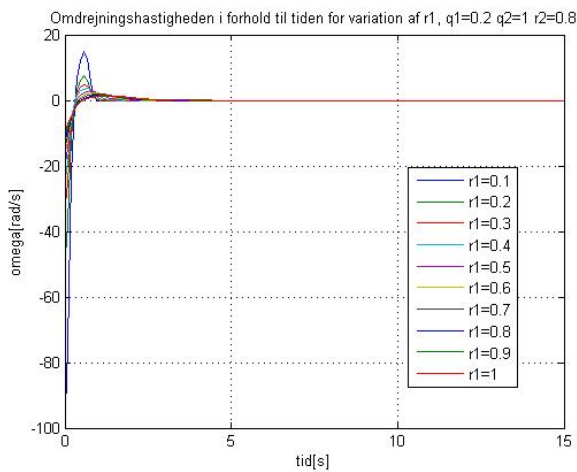
Figur A.36:  $v_l$  som funktion af tiden for variation af  $q_2$ ,  $q_1 = 0.2$ ,  $r_1=0.6$  og  $r_2=0.8$



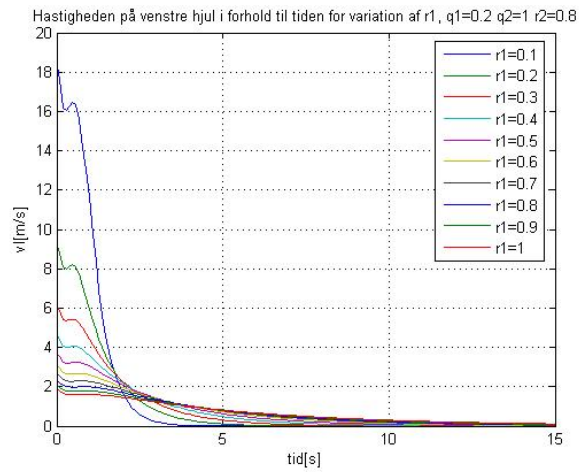
Figur A.37:  $v_r$  som funktion af tiden for variation af  $q_2$ ,  $q_1 = 0.2$ ,  $r_1=0.6$  og  $r_2=0.8$

A.2.3 variation af  $r_1$ 

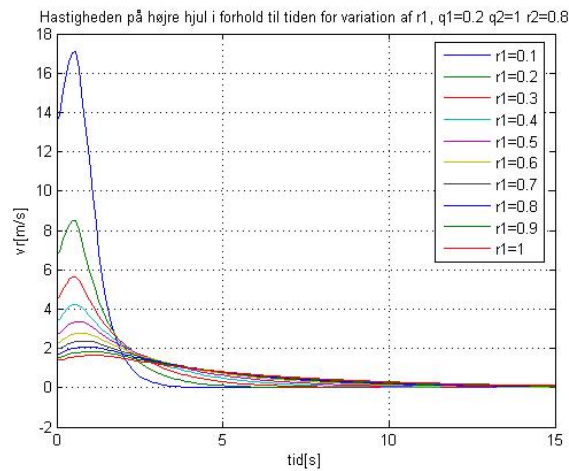
Figur A.38:  $\theta$  som funktion af tiden for variation af  $r_1$ ,  $q_1 = 0.2$ ,  $q_2=1$  og  $r_2=0.8$



Figur A.39:  $\omega$  som funktion af tiden for variation af  $r_1$ ,  $q_1 = 0.2$ ,  $q_2=1$  og  $r_2=0.8$

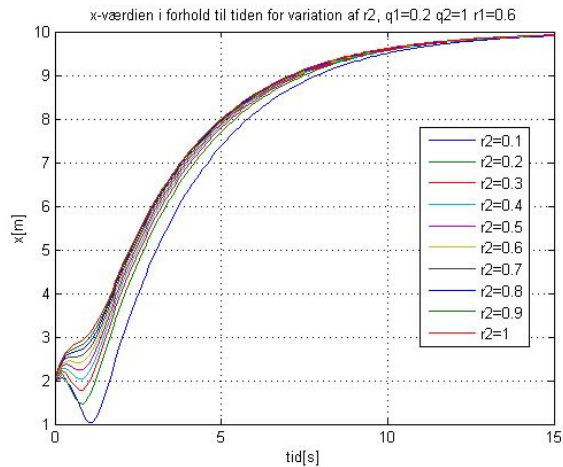


Figur A.40:  $v_l$  som funktion af tiden for variation af  $r_1$ ,  $q_1 = 0.2$ ,  $q_2=1$  og  $r_2=0.8$

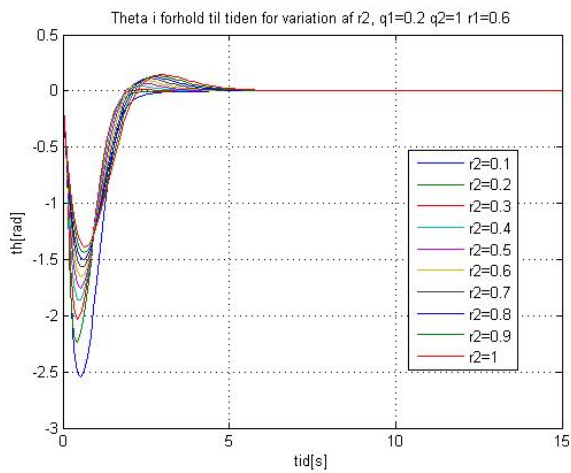


Figur A.41:  $v_r$  som funktion af tiden for variation af  $r_1$ ,  $q_1 = 0.2$ ,  $q_2=1$  og  $r_2=0.8$

### A.2.4 variation af $r_2$

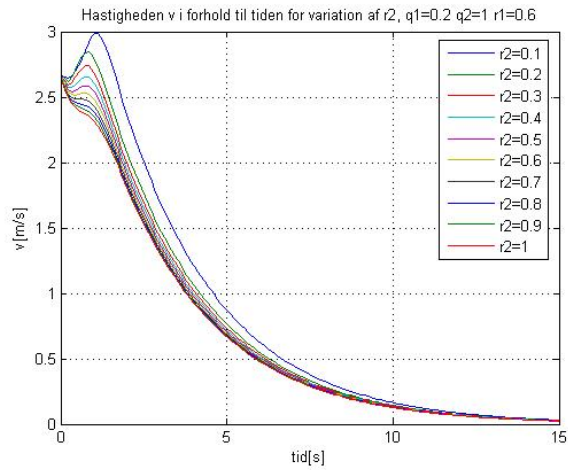


Figur A.42:  $x$  som funktion af tiden for variation af  $r_2$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$

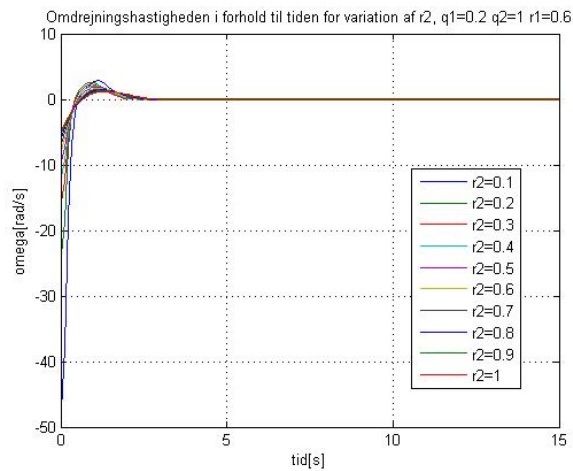


Figur A.43:  $\theta$  som funktion af tiden for variation af  $r_2$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$

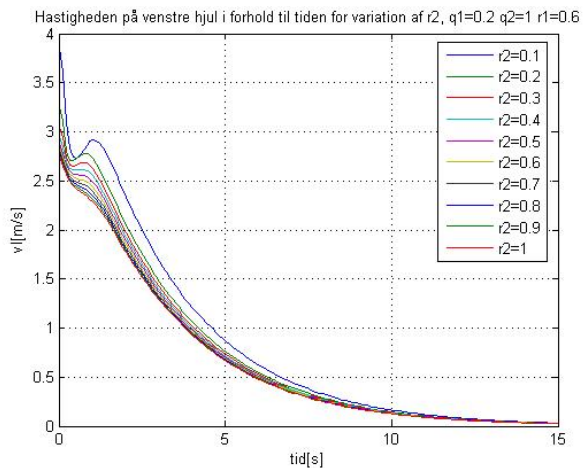




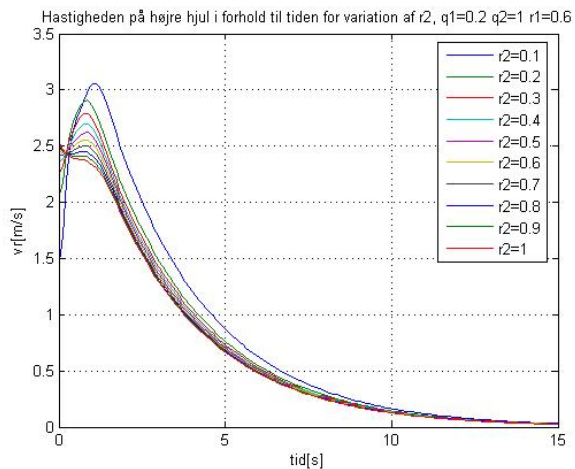
Figur A.44:  $v$  som funktion af tiden for variation af  $r_2$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$



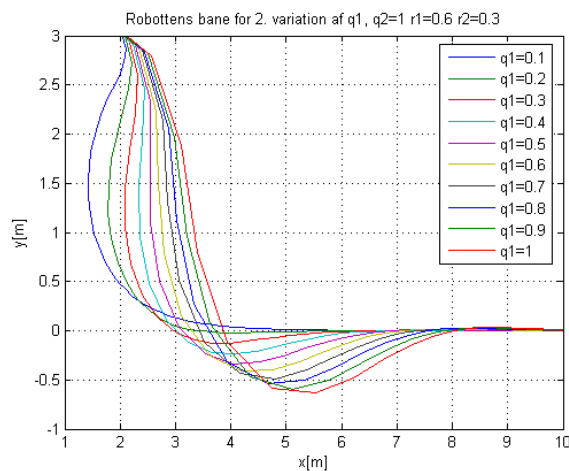
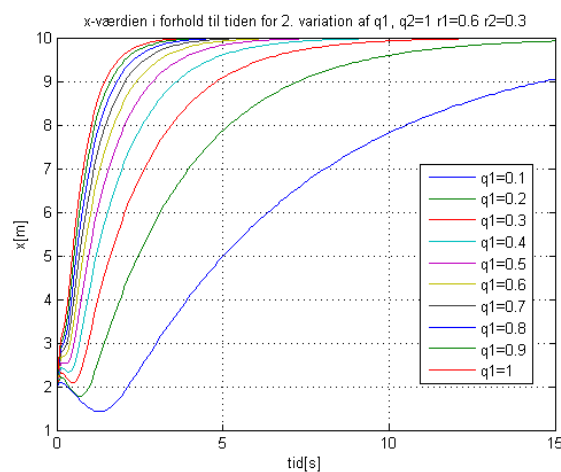
Figur A.45:  $\omega$  som funktion af tiden for variation af  $r_2$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$

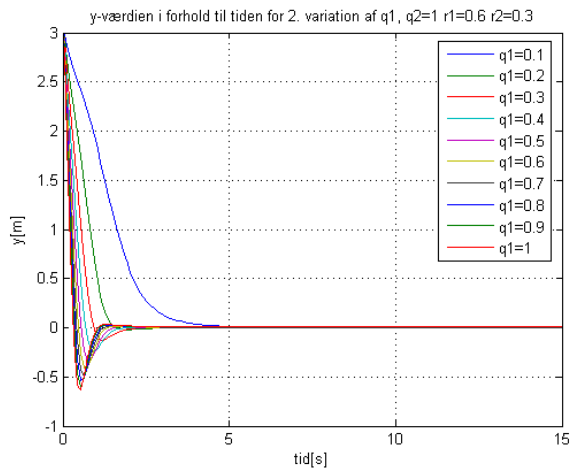


Figur A.46:  $v_l$  som funktion af tiden for variation af  $r_2$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$

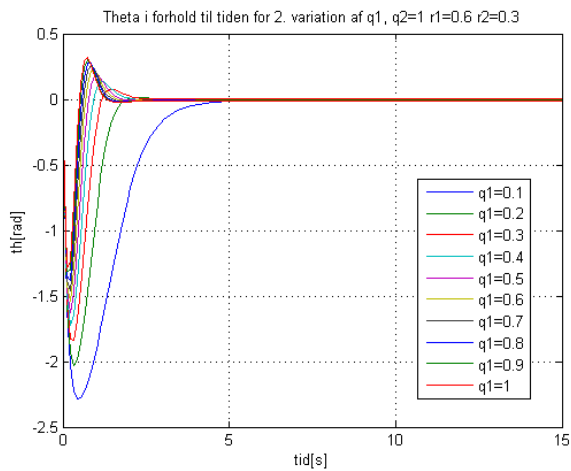


Figur A.47:  $v_r$  som funktion af tiden for variation af  $r_2$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$

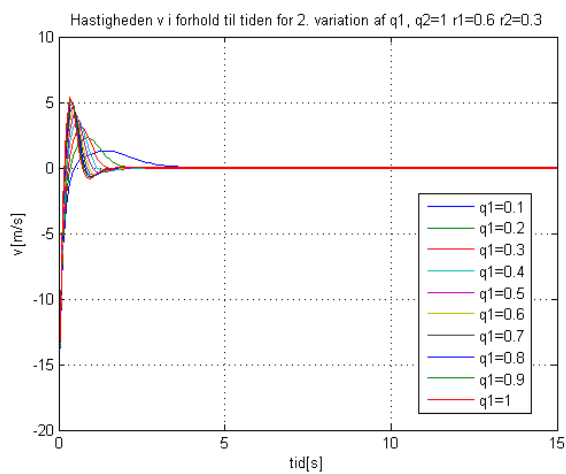
A.2.5 2. variation af  $q_1$ Figur A.48: Robotten bane for 2. variation af  $q_1$ ,  $q_2=1$ ,  $r_1=0.6$  og  $r_2=0.3$ Figur A.49:  $x$  som funktion af tiden for 2. variation af  $q_1$ ,  $q_2=1$ ,  $r_1=0.6$  og  $r_2=0.3$



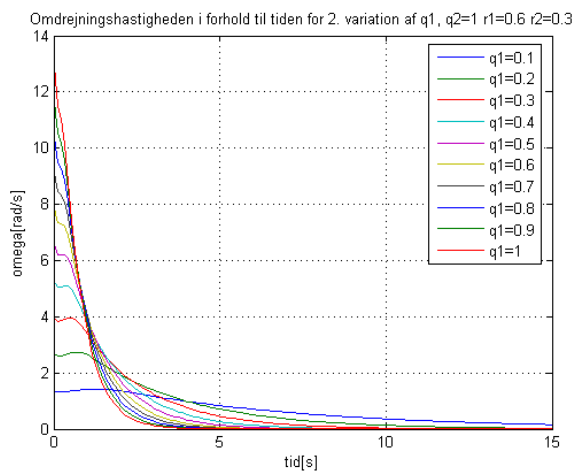
Figur A.50:  $y$  som funktion af tiden for 2. variation af  $q_1$ ,  $q_2=1$ ,  $r_1=0.6$  og  $r_2=0.3$



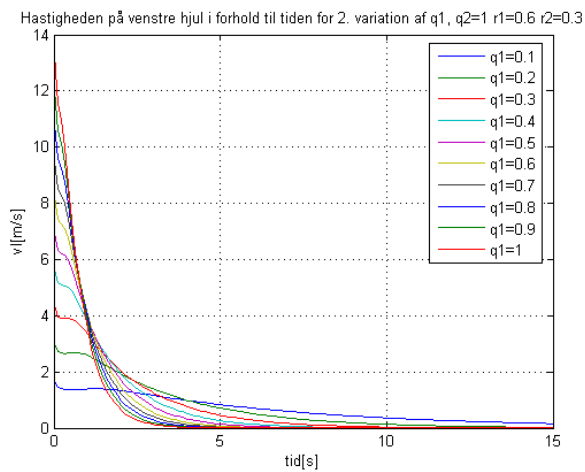
Figur A.51:  $\theta$  som funktion af tiden for 2. variation af  $q_1$ ,  $q_2=1$ ,  $r_1=0.6$  og  $r_2=0.3$



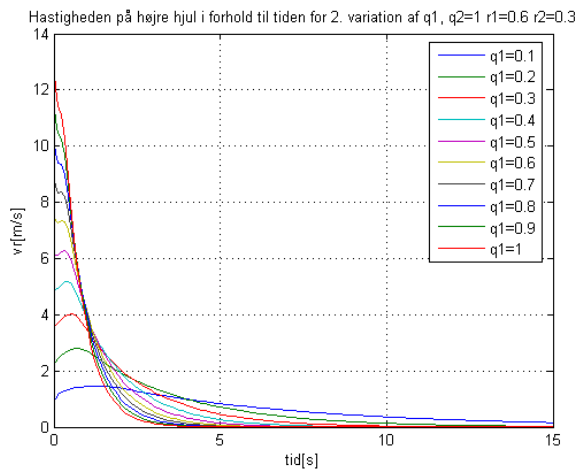
Figur A.52:  $v$  som funktion af tiden for 2. variation af  $q_1$ ,  $q_2=1$ ,  $r_1=0.6$  og  $r_2=0.3$



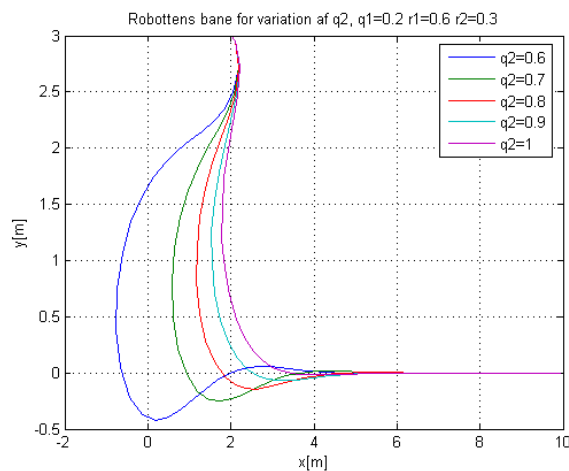
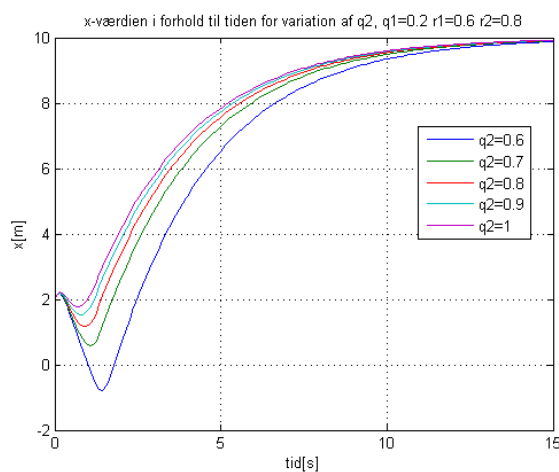
Figur A.53:  $\omega$  som funktion af tiden for 2. variation af  $q_1$ ,  $q_2=1$ ,  $r_1=0.6$  og  $r_2=0.3$

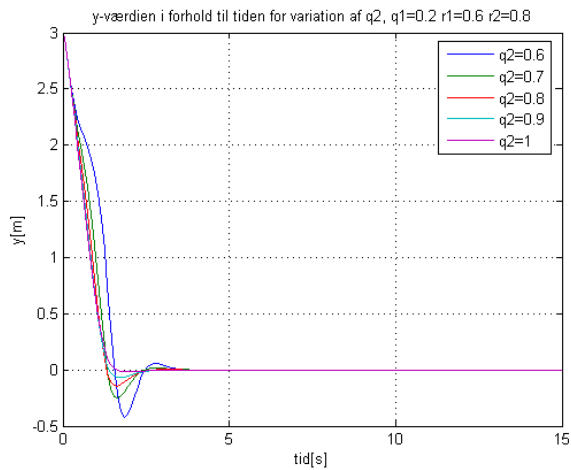


Figur A.54:  $v_l$  som funktion af tiden for 2. variation af  $q_1$ ,  $q_2=1$ ,  $r_1=0.6$  og  $r_2=0.3$

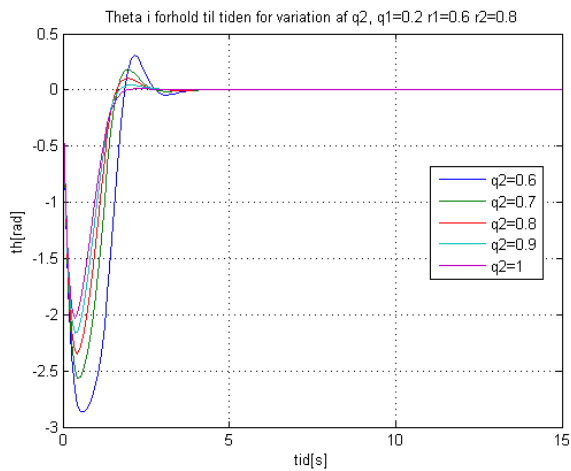


Figur A.55:  $v_r$  som funktion af tiden for 2. variation af  $q_1$ ,  $q_2=1$ ,  $r_1=0.6$  og  $r_2=0.3$

A.2.6 2. variation af  $q_2$ Figur A.56: Robotten bane for 2. variation af  $q_2$ ,  $q_1=0.2$ ,  $r_1=0.6$  og  $r_2=0.3$ Figur A.57:  $x$  som funktion af tiden for 2. variation af  $q_2$ ,  $q_1=0.2$ ,  $r_1=0.6$  og  $r_2=0.3$

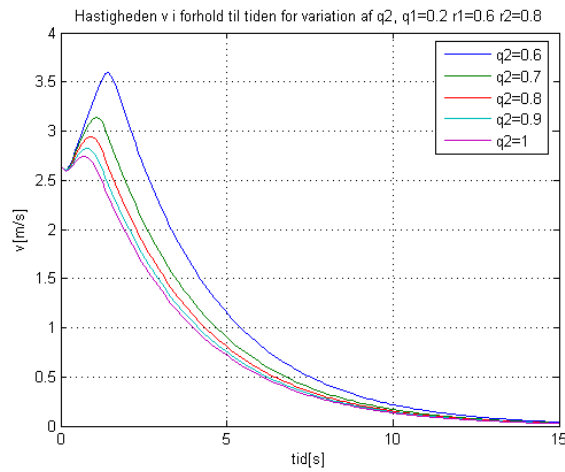


Figur A.58:  $y$  som funktion af tiden for 2. variation af  $q_2$ ,  $q_1=0.2$ ,  $r_1=0.6$  og  $r_2=0.3$

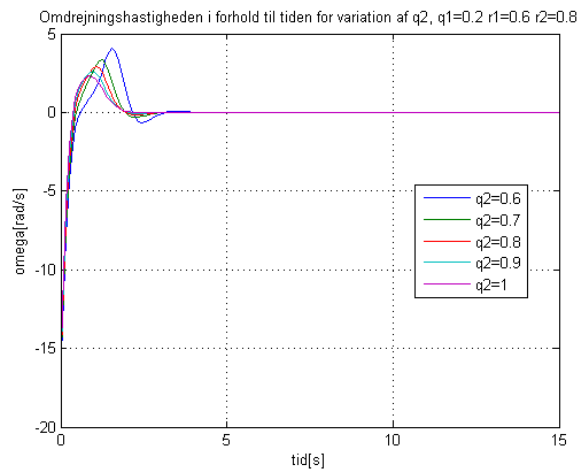


Figur A.59:  $\theta$  som funktion af tiden for 2. variation af  $q_2$ ,  $q_1=0.2$ ,  $r_1=0.6$  og  $r_2=0.3$

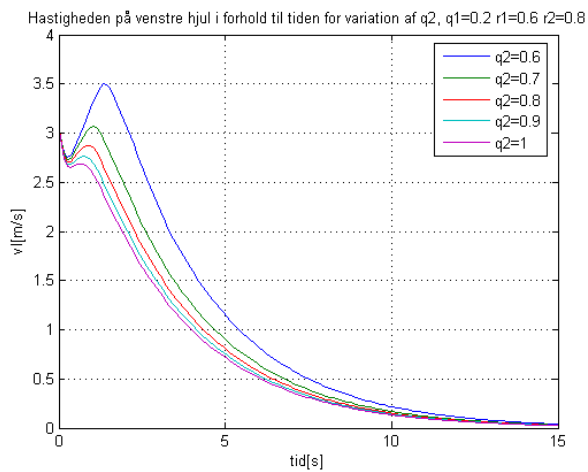




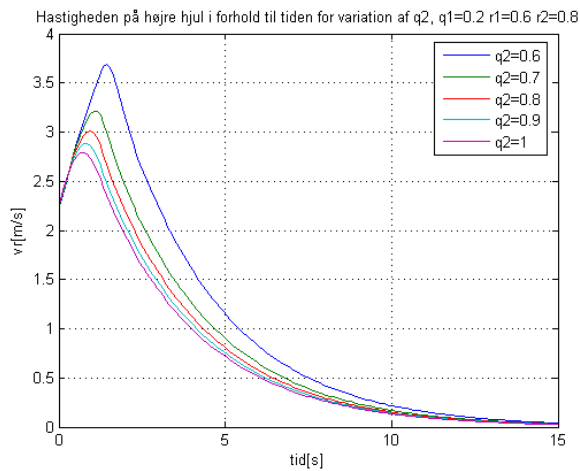
Figur A.60:  $v$  som funktion af tiden for 2. variation af  $q_2$ ,  $q_1=0.2$ ,  $r_1=0.6$  og  $r_2=0.3$



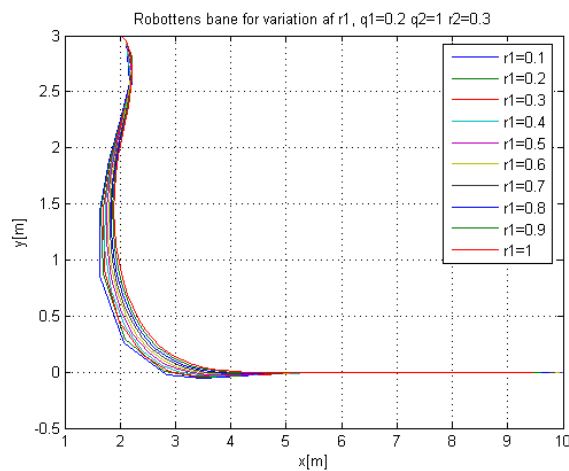
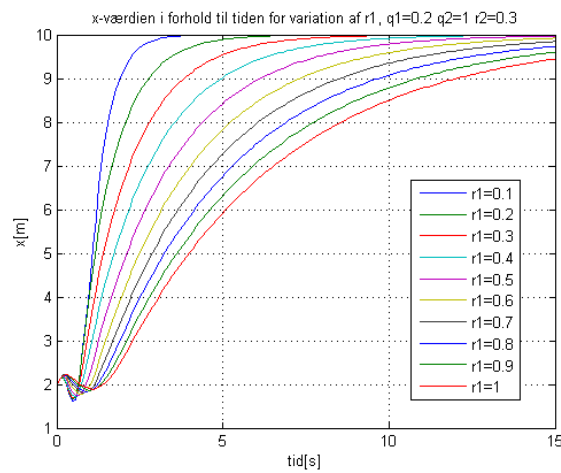
Figur A.61:  $\omega$  som funktion af tiden for 2. variation af  $q_2$ ,  $q_1=0.2$ ,  $r_1=0.6$  og  $r_2=0.3$

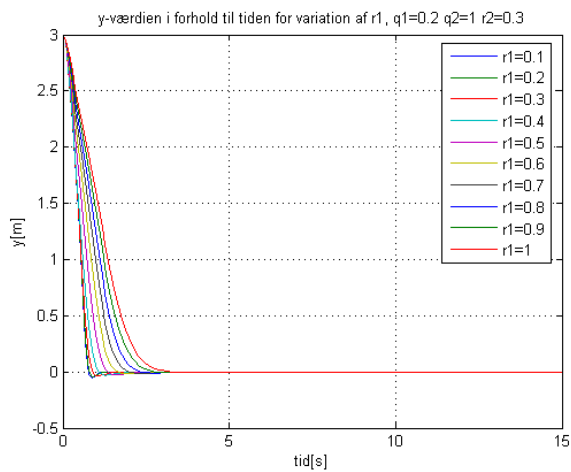


Figur A.62:  $v_l$  som funktion af tiden for 2. variation af  $q_2$ ,  $q_1=0.2$ ,  $r_1=0.6$  og  $r_2=0.3$

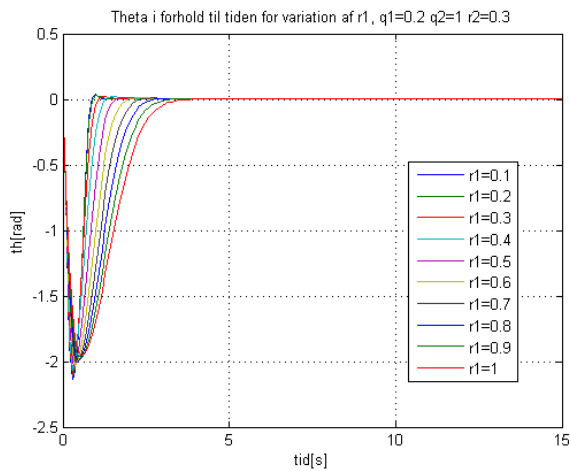


Figur A.63:  $v_r$  som funktion af tiden for 2. variation af  $q_2$ ,  $q_1=0.2$ ,  $r_1=0.6$  og  $r_2=0.3$

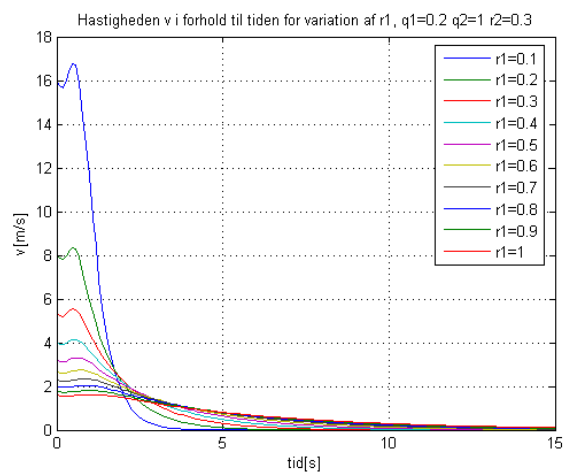
A.2.7 2. variation af  $r_1$ Figur A.64: Robotten bane for 2. variation af  $r_1$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$ Figur A.65:  $x$  som funktion af tiden for 2. variation af  $q_1$ ,  $q_2=1$ ,  $r_1=0.6$  og  $r_2=0.3$



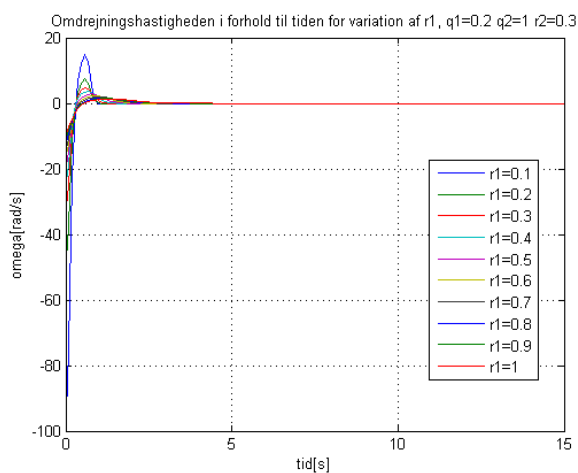
Figur A.66:  $y$  som funktion af tiden for 2. variation af  $r_1$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$



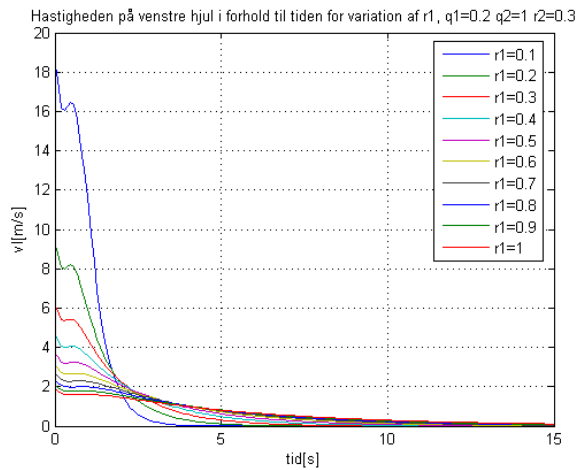
Figur A.67:  $\theta$  som funktion af tiden for 2. variation af  $r_1$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$



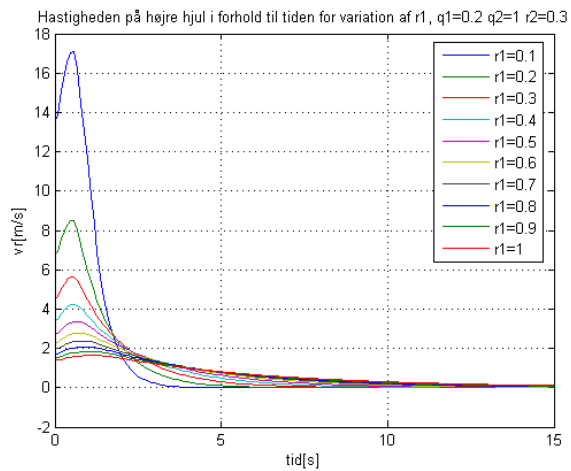
Figur A.68:  $v$  som funktion af tiden for 2. variation af  $r_1$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$



Figur A.69:  $\omega$  som funktion af tiden for 2. variation af  $r_1$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$



Figur A.70:  $v_l$  som funktion af tiden for 2. variation af  $r_1$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$



Figur A.71:  $v_r$  som funktion af tiden for 2. variation af  $r_1$ ,  $q_1=0.2$ ,  $q_2=1$  og  $r_1=0.6$

## BILAG B

# Matlab-kode

---

## B.1 YTHcontrol.m

```
1 % Bachelorprojekt for Dina Friesel (s042196)
2 % Funktion som får q=[y, theta] og bestemmer omega ved
3 % formlen omega=-L*q
4
5 function u = YTHcontrol(x, v0)
6
7 q = 0.6;
8 r = 0.8;
9 rv=0.8;
10
11 % hvis robotten er over y-båndet (y>3) skal der kun styres
12 % på vinklen mod -pi/2
13 if x(1)>3.5
14     A = 0;
15     B = 1;
16     Q = 1;
17     R = rv^2;
18     L = lqr(A,B,Q,R);
19     u=-L*(x(2)-(-pi/2));
20
21 % hvis robotten er under y-båndet (y<-3) skal der kun styres
22 % på vinklen mod pi/2
23 else if x(1)<-3.5
24     A = 0;
25     B = 1;
```

```

26     Q = 1;
27     R = rv^2;
28     L = lqr(A,B,Q,R);
29     u=-L*(x(2)-pi/2);
30
31     % hvor robotten er indenfor y-båndet styres der på både y og
32     % theta mod 0
33     else
34         A = [ 0 v0; 0 0 ];
35         B = [0; 1];
36         Q = diag([1 q^2]);
37         R = r^2 * diag(1);
38         L = lqr(A,B,Q,R);
39         u=-L*x;
40     end
41 end

```

## B.2 hastighedskontrol.m

```

1  % Bachelorprojekt for Dina Friesel (s042196)
2  % Funktion som tager q=[x,y,theta] ind og generere u=[v,omega]
3  % ved hjælp af u=-L*q
4
5  function u = hastighedskontrol(x, v0)
6
7  q1=0.2;
8  q2=1;
9  r1=0.6;
10 r2=0.3;
11 rv=0.8;
12
13 % hvis robotten er over y-båndet (y>3) skal der kun styres
14 % på vinklen mod -pi/2
15 if x(2)>3.5
16     A = 0;
17     B = 1;
18     Q = 1;
19     R = rv^2;
20     L = lqr(A,B,Q,R);
21     u(2)=-L*(x(3)-(-pi/2));
22     u(1)=v0;
23
24 % hvis robotten er under y-båndet (y<-3) skal der kun styres
25 % på vinklen mod pi/2
26 else if x(2)<-3.5
27     A = 0;
28     B = 1;
29     Q = 1;
30     R = rv^2;
31     L = lqr(A,B,Q,R);
32     u(2)=-L*(x(3)-pi/2);
33     u(1)=v0;
34

```



```

35     % hvor robotten er indenfor y-båndet styres der på både x, y og
36     % theta mod 0
37     else
38         A = [0 0 0; 0 0 v0; 0 0 0];
39         B = [1 0; 0 0; 0 1];
40
41         Q = diag([q1^2 1 q2^2]);
42         R = r1^2*diag([1 r2^2]);
43         L = lqr(A,B,Q,R);
44         u=-L*[x(1);x(2);x(3)];
45     end
46 end

```

## B.3 THwaypointkontrol.m

```

1  % Bachelorprojekt for Dina Friesel (s042196)
2  % Funktion som får q=[x,y, theta] og bestemmer omega ved
3  % formlen omega=-L*(theta-thref) for 4 forskellige waypoints
4
5  function u = THwaypointkontrol(x, v0)
6
7  global punkt pNU alpha
8
9  xkoor = x(1);    %x-koordinat
10 y = x(2);       %y-koordinat
11 th = x(3);      %vinklen
12
13 %Afstanden fra robotten til referencepunktet
14 dist = sqrt((xkoor-pNU(1))^2+(y-pNU(2))^2);
15
16 %Tolerancen
17 eps=0.05;
18
19 %Beregner L-værdien
20 Q = 1;
21 R = 0.8^2;
22 A = 0;
23 B = 1;
24 L = lqr(A,B,Q,R);
25
26 %Bestemmer den direkte vinkel til referencepunktet
27 if pNU==punkt(:,4) & dist<=0.05
28     disp('4. punkt er nået! Stop simuleringen')
29     pause
30 end
31 if pNU==punkt(:,3) & dist<=eps
32     pNU = punkt(:,4);
33 end
34 if pNU==punkt(:,2) & dist<=eps
35     pNU = punkt(:,3);
36 end
37 if pNU==punkt(:,1) & dist<=eps
38     pNU = punkt(:,2);

```

```
39 end
40
41 alpha = atan2(pNU(2)-y,pNU(1)-xkoor);
42
43 if abs(th-alpha)>pi
44     if th>0
45         alpha=alpha+2*pi;
46     else
47         alpha=alpha-2*pi;
48     end
49 end
50
51 u=-L*[th-alpha];
```

## B.4 liniestyringkontrol.m

```
1 % Bachelorprojekt for Dina Friesel (s042196)
2 % Funktion som tager q=[x,y,theta] ind og generere u=[v,omega]
3 % ved hjælp af hastighedkontrol.m
4
5 function u = liniestyringkontrol(x, v0)
6
7 global punkt pNU vinkel vNU Mny st
8
9 if st >1
10 xkoor = x(1);
11 y = x(2);
12 th = x(3);
13
14 %afstanden til punktet
15 dist = sqrt((xkoor)^2+(y)^2);
16
17 eps=0.05; %tolerance
18
19 if pNU==punkt(:,1) & dist<=eps
20     disp('Punktet er nået! Stop simuleringen')
21     pause
22 end
23
24 end
25 u=hastighedkontrol(x,v0);
26 st=st+1;
```

### B.4.1 koordinatskift.m

```
1 % Bachelorprojekt for Dina Friesel (s042196)
2 % Funktion som skifter koordinater fra globalt til
3 % lokalt koordinatsystem
4
5 function koor = koordinatskift(koorg, p, v, Mny)
6
```

```
7  xg=koorg(1);
8  yg=koorg(2);
9  thg=koorg(3);
10
11 xref=p(1);
12 yref=p(2);
13 thref=v;
14
15 koor = Mny*[xg-xref; yg-yref];
16 koor = [koor; thg-thref];
```

## B.5 liniestyringkontrol2.m

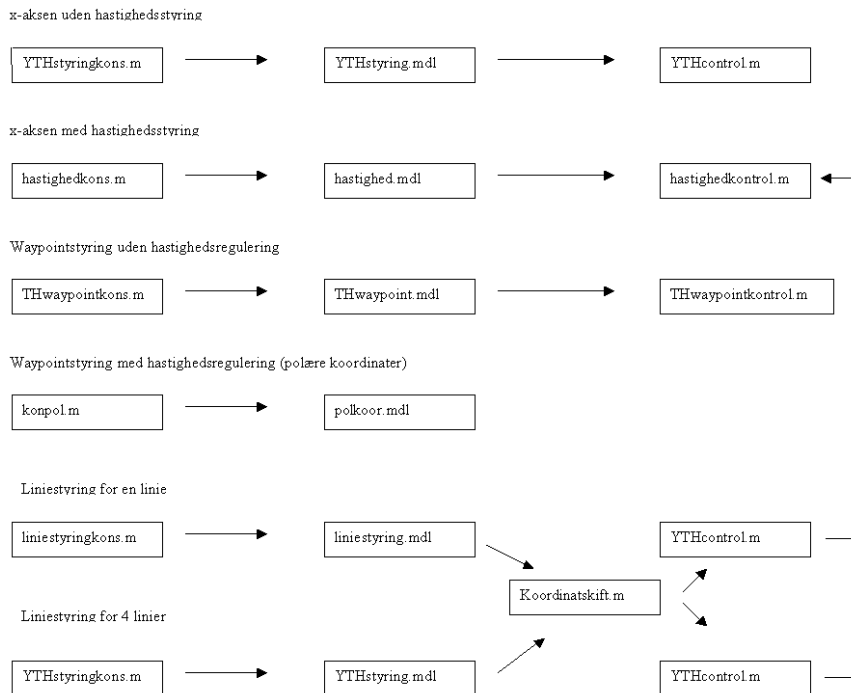
```
1  % Bachelorprojekt for Dina Friesel (s042196)
2  % Funktion som tager q=[x,y,theta] ind og generere u=[v,omega]
3  % ved hjælp af hastighedskontrol.m for 4 efterfølgende punkter
4
5  function u = liniestyringkontrol(x, v0)
6
7  global punkt pNU vinkel vNU Mny st
8
9  if st >1
10
11  xkoor = x(1);
12  y = x(2);
13  th = x(3);
14  dist = sqrt((xkoor)^2+(y)^2);
15
16  eps=0.5; %tolerance
17
18  if pNU==punkt(:,4) & dist<=0.05
19      disp('4. punkt er nået! Stop simuleringen')
20      pause
21  end
22  if pNU==punkt(:,3) & dist<=eps
23      pNU = punkt(:,4);
24      vNU = vinkel(4);
25      M=[cos(vNU), -sin(vNU); sin(vNU), cos(vNU)];
26      Mny=inv(M);
27      disp('3')
28  end
29  if pNU==punkt(:,2) & dist<=eps
30      pNU = punkt(:,3);
31      vNU = vinkel(3);
32      M=[cos(vNU), -sin(vNU); sin(vNU), cos(vNU)];
33      Mny=inv(M);
34  end
35  if pNU==punkt(:,1) & dist<=eps
36      pNU = punkt(:,2);
37      vNU = vinkel(2);
38      M=[cos(vNU), -sin(vNU); sin(vNU), cos(vNU)];
39      Mny=inv(M);
40  end
```

```

41 end
42 u=hastighedkontrol(x,v0);
43 st=st+1;

```

## B.6 Diagram over MATLAB-filerne



Figur B.1: Diagram over, hvordan de forskellige vedlagte MATLAB-filer bruger hinanden