

Project Thesis

MODELLING, SIMULATION AND CONTROL OF A TWIN ROTOR MIMO-SYSTEM

at

Department of Systems Engineering and Control
Universidad Politécnica de Valencia

Christoph Gabriel
Fritz-Haber-Straße 3
93073 Neutraubling

Supervising directors: D. F. Xavier Blasco Ferragud, D. Sergio Garcia-Nieto Rodriguez

Valencia,

Acknowledgement

I would like to express my gratitude to all those who gave me the possibility to work on this project. I want to thank my supervising project directors D. F. Xavier Blasco Ferragud and D. Sergio Garcia-Nieto Rodriguez for accepting me as an exchange student to work in the department and for always being willing to listen to my problems. Furthermore I would like to thank my professor at my home university in Germany Prof. Dr.-Ing. habil. Günter Roppenecker who agreed to supervise this project for the part of the home university. Last but not least I would like to thank my family, my friends and everyone who encouraged me to dare to study one year abroad in a different language.

Abstract

The object of the project is to simulate and control the laboratory helicopter model called Twin-Rotor Multiple Input-Multiple Output System (TRMS 33-220), manufactured by Feedback Instruments Limited Company.

So the first aim was to get familiar with the physical basics of the model what implicated an intensive literature research at the beginning of the work on this project. Having built up the basic mathematical model there was to deal with adjusting the particular parameters by engaging several experiment on the real model comparing the results of the simulation done in Matlab/Simulink. This was one of the most delicate tasks during the projects since there was to deal with constants that were not directly measurable and because particular parts of the model had been changed in the past so that the constant parameters given in the manual went out of date. The control design as well was done using Matlab/Simulink

Having reached satisfactory results of the model further experiments were thenceforward on a simulator, firstly implemented in Labview, then embedded on a real-time platform called PXI-1002 manufactured by National Instruments to avoid physical damages of the real model by possible failures during the controller testing.

The two control objects were first to keep the helicopter at a certain point and second to follow a given trajectory. So the first effort to create a control was to design two independent PID-Controllers, one for the vertical axis and one for the horizontal one, considering the cross-coupling forces as disturbance. This form of control worked out quite fine as long as only one input was changed. To compensate the disturbances correctly we had to deal with a different type, called state-space-control.

The final task was to run the designed controller on the PXI-1002 controlling the physical system.

Contents

1	Introduction	1
2	Modelling of the System	2
2.1	Physical Basics	2
2.1.1	DC-Motors	2
2.1.2	Propellers	4
2.1.3	Gravity Forces	5
2.1.4	Ascending Forces	6
2.1.5	Moments of Inertia	7
2.1.6	Friction	9
2.1.7	Torsion	10
2.1.8	Electro-mechanical Coupling	10
2.1.9	Centrifugal Force	10
2.2	Coupled non-linear System	11
2.3	Validation	12
2.4	State-Space Representation	13
2.4.1	Linearised Continuous System	17
3	Control Design	18
3.1	State Control	18
3.2	Observer	22
3.3	Control Results	25
3.4	Transfer to Discrete Time Domain	27
4	Controlling the System	30
4.1	Mathematical model controlled by PXI	30
4.2	Physical model controlled by PXI	31
5	Conclusion and Future Works	32
6	Appendix	33
6.1	List of used constants	33
6.2	List of Figures	34
6.3	Literature	36

List of used symbols

Symbol	Unit	Description
$u_{m/t}$	[V]	DC-motor input voltage
$E_{m/t}$	[V]	DC-motor armature voltage
$R_{m/t}$	[ω]	DC-motor coil resistance
$L_{m/t}$	[H]	DC-motor coil inductivity
$i_{m/t}$	[A]	DC-motor current
$k_{m/t}$	[$\frac{V}{rpm}$]	DC-motor constant
$k_{bm/bt}$	[$\frac{Nm}{rpm}$]	DC-motor bearing friction constant
$k_{cm/ct}$	[$\frac{Nm}{rpm}$]	Electro-mechanical coupling constant
$k_{Am/At}$	[$\frac{Nm}{rpm^2}$]	Propeller air friction constant
$k_{fvp/fhp}$	[$\frac{N}{rpm^2}$]	Ascending force constant
k_v/h	[$\frac{Nms}{rad}$]	Viscous friction constant
k_c	[$\frac{Nm}{rad}$]	Torsion constant
$\omega_{m/t}$	[rpm]	rotational propeller speed
$J_{Am/At}$	[kgm]	DC-motor moment of inertia
$J_{Pm/Pt}$	[kgm]	Propeller moment of inertia
J_v/h	[kgm]	System moment of inertia
$M_{Mm/Mt}$	[Nm]	DC-motor torque
$M_{Bm/Bt}$	[Nm]	DC-motor bearing friction torque
$M_{Am/At}$	[Nm]	Air friction torque
M_L	[Nm]	!!loading torque!!
M_G	[Nm]	Gravity torque
$M_{Av/Av}$	[Nm]	Torque due to ascending force
$M_{Fv/Fh}$	[Nm]	Friction torque
M_T	[Nm]	Torsion torque
$M_{MCm/MCt}$	[Nm]	Torque due to electro-mechanical coupling
M_C	[Nm]	Centrifugal torque

Symbol	Unit	Description
m_t	[kg]	Mass of tail part of the beam
m_{tr}	[kg]	Mass of tail motor and rotor
m_{ts}	[kg]	Mass of tail shield
m_m	[kg]	Mass of main part of the beam
m_{mr}	[kg]	Mass of main motor and rotor
m_{ms}	[kg]	Mass of main shield
m_b	[kg]	Mass of counter-weight beam
m_{cb}	[kg]	Mass of counter-weight
l_t	[m]	length of tail part of the beam
l_m	[m]	length of main part of the beam
l_b	[m]	length of counter-weight beam
l_{cb}	[m]	distance between counterweight and pivot
K	$[\frac{1}{V}]$	PT1 Gain Factor
T	[s]	General Time Constant
T_m	[s]	Main-Rotor system time constant
T_t	[s]	Tail-Rotor system time constant
α_v	[rad]	Vertical displacement angle
α_h	[rad]	Horizontal displacement angle
Ω_v	$[\frac{rad}{s}]$	Vertical angular velocity
Ω_h	$[\frac{rad}{s}]$	Horizontal angular velocity
$\overline{\omega_m}$	[1]	Pseudo rotational speed of main rotor
$\overline{\omega_h}$	[1]	Pseudo rotational speed of tail rotor
$F_{v/h}$	[s]	Ascending force

1 Introduction

A helicopter is an aircraft that is lifted and driven by one or more rotor blades. The name "helicopter" is adapted from French "hélicoptère" and is linked to the Greek words *helix* (spiral) and *pteron* (wing).

Helicopters are classified as rotary-wing aircraft in contrast to fixed-wing aircraft since a helicopter gets its lift from rotating rotor blades around a mast. The special benefit of an helicopter is the availability to land and take off vertically without the need for runways and maintaining a steady hover in the air over a single point what makes this type of aircraft ideal for rescue and evacuation missions but also as an observer platform. Further application of helicopters can be found in fire fighting, as an aerial crane and in military matters. Nowadays there are a lot of different kinds of helicopters, the most popular one is the type with one main rotor and one anti torque tail rotor. Using this technology the main rotor is used for the lift, forward and sideways motion by varying the approach angle of the rotor blades. The tail rotor is used for rotation around the vertical axis and balance of torque around the vertical axis built up by the movement of the main rotor.

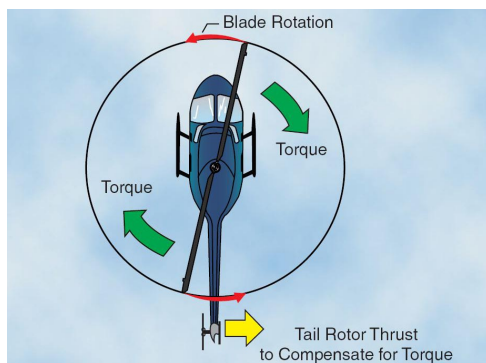


Figure 1: Torque effect on a helicopter and compensation by tail rotor

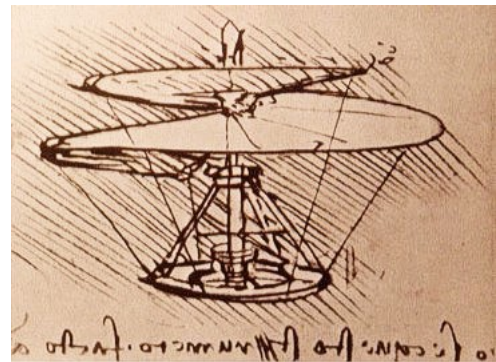


Figure 2: Early draft of an helicopter by Leonardo da Vinci

The roots of the idea of building an helicopter go far back to the 15th century when the first drafts of an helicopter were made by Leonardo da Vinci. Until the final realisation more than 400 years passed by when Paul Cornu for the first time in history took off for at least 20 seconds in 1907 using tandem-rotors driven by a 24 PS V8-engine. The first mass-produced helicopter was the german Focke-Achgelis Fa 223 in 1941.



Figure 3: A helicopter of the type Bell 205 dropping water on fire

2 Modelling of the System

The physical model we have to deal with can be described as follows:

The TRMS as shown in Fig. (1) is driven by two DC motors which are each coupled to a propeller. Like in a real helicopter the two propellers are perpendicular to each other. The connection between the two motors is



Figure 4: Photo of the model

built by a beam pivoted on its base that can move freely in both horizontal and vertical planes. Moreover there is a counterweight mounted by another beam fixed perpendicular to the main one. The beam can be moved by changing the input voltages of the motors to affect the rotational speed of the rotors.

Even though the laboratory model of the helicopter is quite different to an original one it resembles the main behaviour we have to deal with designing a control - the strong cross-coupling between the main rotor and the tail rotor.

The so introduced mathematical model tries to describe the physical model as precisely as possible by Newtonian equations.

2.1 Physical Basics

2.1.1 DC-Motors

The TRMS possesses two permanent magnet DC motors, one for the main propeller and one for the tail. In the original model these two motors were identical but since one of the motors was replaced due to a malfunction they are not to be considered as identical. Anyway the motors work with different loads since

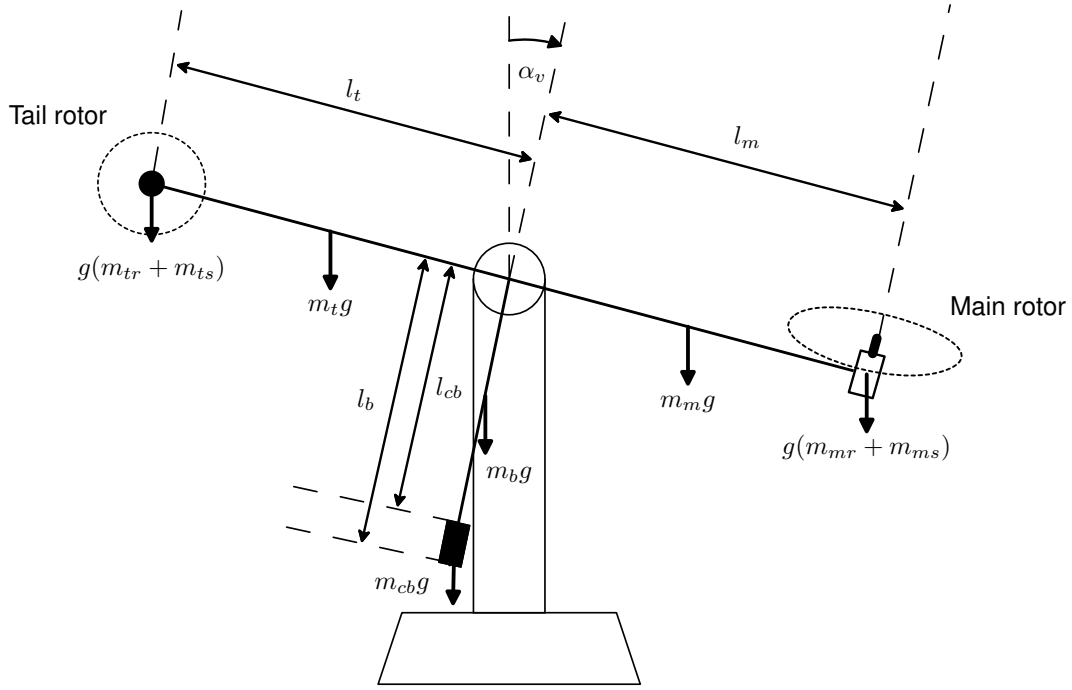


Figure 5: Schematic model in vertical plane

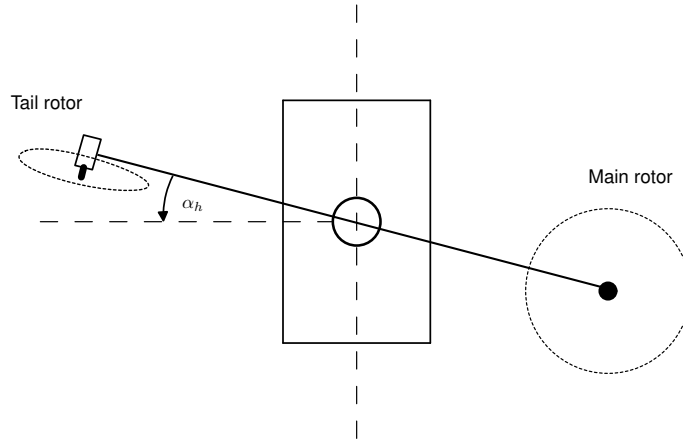


Figure 6: Schematic model in horizontal plane

the propellers are of unequal sizes.

A suchlike motor can be mathematically represented by a simple continuous-current circuit

$$u_{m/t} = E_{m/t} + R_{m/t} i_{m/t} + L_{m/t} \frac{di_{m/t}}{dt} \quad (1)$$

The armature voltage is proportional to the rotation speed whereas $k_{m/t}$ states a motor constant.

$$E_{m/t} = k_{m/t} \omega_{m/t} \quad (2)$$

Hence the torque can be calculated by the following differential equation:

$$J_{Am/At} \frac{d\omega_{m/t}}{dt} = M_{Mm/Mt} - M_L \quad (3)$$

with $M_{Mm/t} = k_{m/t}i_{m/t}$

Since the momentum of load M_L is non-linear and does not directly come from the rotation speed $k_{m/t}$, this variable will be discussed in the propeller section.

2.1.2 Propellers

First of all there is to say that the the propellers are each hard coupled to the motors so that we get a composed moment of inertia made up of both the propeller's and motor's moments of inertia.

$$J_{m/t} = J_{Am/At} + J_{Pm/Pt} \quad (4)$$

By connecting voltage to the inputs of the DC-motor, it builds up a torque that depends on the armature voltage and hence on the input current. The movement of the propeller can be described as a first order non-linear differential equation. The particular summands that build up the torque are composed of the bearing friction M_B , the air friction M_A and the motor's driving torque $M_{Mm/Mt}$. Putting all together we get the extended equation of (3) - now with the composed moment of inertia and splitted up M_L :

$$J_{m/t} \frac{d\omega_{m/t}}{dt} = M_{Bm/Bt} + M_{Am/At} + M_{Mm/Mt} = -k_{Bm/Bt}\omega_{m/t} - k_{Am/At} \text{sgn}(\omega_{m/t})\omega_{m/t}^2 + k_{m/t}i_{m/t} \quad (5)$$

Putting together the equitations (1) to (5) in a block diagram leads to block diagram shown in figure (7).

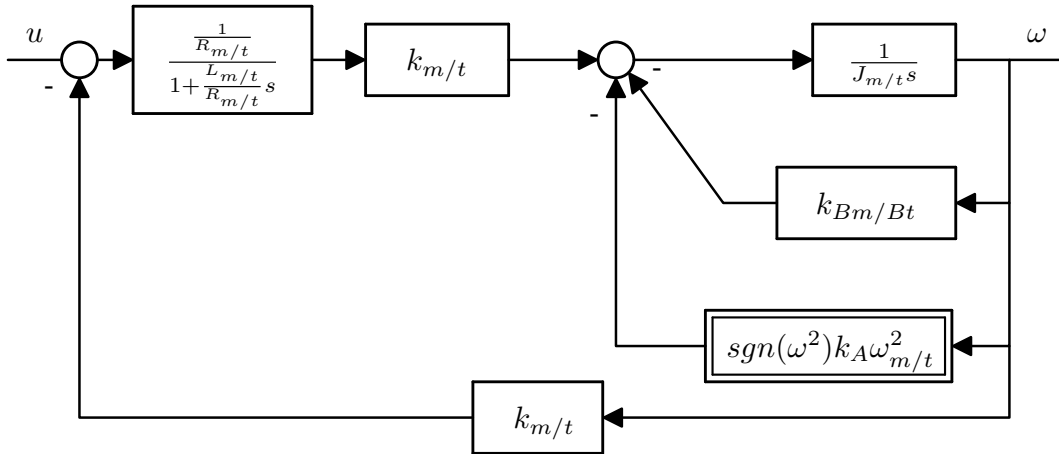


Figure 7: Block diagramm of the coupled motor-propeller system

Since we can easily see that the first-order lag element's time constant is quite small compared to the surrounding one's, we consider the motor system as a proportional element to simplify the model. Moreover we modify the diagram to that effect that we get the normal form of a first-order lag element like shown in figure (8).

As we see that the term $\frac{k_{m/t}}{J_{m/t}R_{m/t}}$ states the gain factor divided by the time constant $\frac{K}{T}$ while

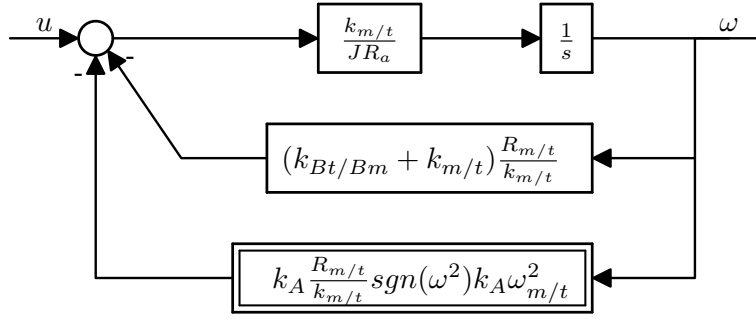


Figure 8: Block diagramm of the simplified coupled motor-propeller system considered as a first-order-lag element

$-k_{Bm/Bt}\omega_{m/t} - k_{Am/At}\text{sgn}(\omega_{m/t})\omega_{m/t}^2$ only states the gain factor it is possible to consider the whole system as a first-order lag element with a gain factor of $\frac{1}{\nu}$ with an additional non-linear gain factor like shown in figure (9). The parameters for this system were identified experimentally. The gain factor was measured by recording a characteristic curve of the rotational speed against the input voltage as shown in figures (10) to (11). The time constants were measured by recording a simple step responses of the angular velocity as shown in Fig. (12) and (13).

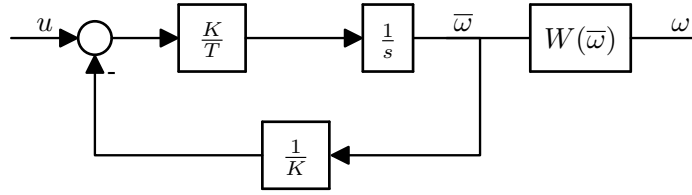


Figure 9: Block diagramm of the simplified coupled motor-propeller system considered as a first-order-lag element with additional non-linear gain

Referring to Figure (10) and (11) we use the time constants $T_m = 1,33s$ and $T_t = 0,43s$.

The static functions of the rotational speed can be approximated by the following equations like shown in Figure (8) and (9):

$$W_m(\bar{\omega}_m) = (63,23\bar{\omega}_m^6 + 518.84\bar{\omega}_m^5 - 106.16\bar{\omega}_m^4 - 1060\bar{\omega}_m^3 + 47.93\bar{\omega}_m^2 + 1166.2\bar{\omega}_m)rpm \quad (6)$$

for the main rotor and

$$W_t(\bar{\omega}_t) = (89,65\bar{\omega}_t^6 + 532.48\bar{\omega}_t^5 - 88.56\bar{\omega}_t^4 - 1027,3\bar{\omega}_t^3 + 23.91\bar{\omega}_t^2 + 1519,8\bar{\omega}_t)rpm \quad (7)$$

for the tail rotor

2.1.3 Gravity Forces

Since the Gravity Force always affects mass perpendicularly to the earth's surface the resulting torque only affects the angle α_v , the vertical one. Due to simplification all used masses are considered as point-masses.

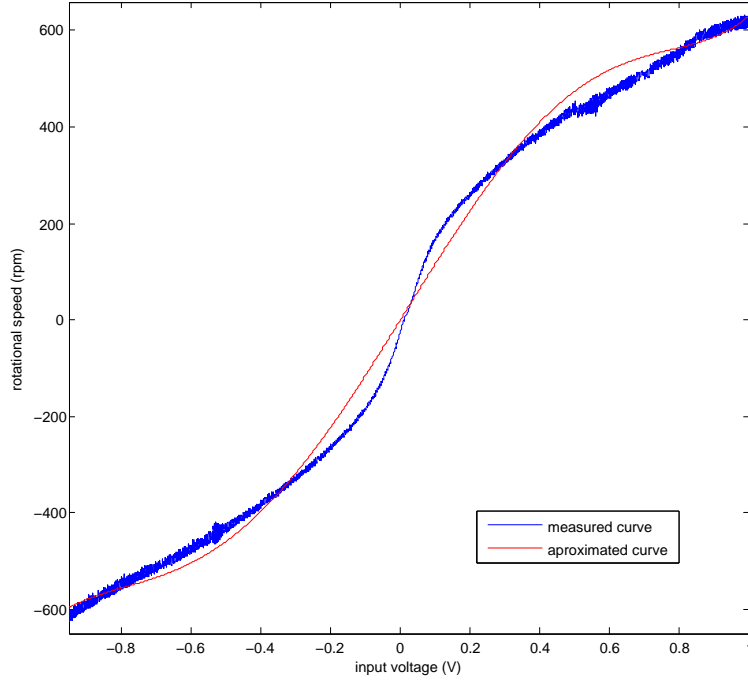


Figure 10: Rotational speed of main rotor against input voltage

$$M_G = \left\{ \left[\left(\frac{m_t}{2} + m_{tr} + m_{ts} \right) l_t - \left(\frac{m_m}{2} + m_{mr} + m_{ms} \right) l_m \right] \cos(\alpha_v) - \left(\frac{m_b}{2} l_b + m_{cb} l_{cb} \right) \sin(\alpha_v) \right\} g \quad (8)$$

or in compact form:

$$M_G = \{ (A - B) \cos(\alpha_v) - C \sin(\alpha_v) \} g \quad (9)$$

2.1.4 Ascending Forces

A helicopter produces its ascending force by a rotatory creation of pressure difference. Concerning Newton's third axiom there is a force in opposite direction as the force that pushes the air down. The rotor's face the rotational speed and the density of the surrounding medium are the main values that determine the ascending force. The simplified equation of the original characteristic force can be considered as:

$$F_{h/v} = k_{fvp/fhp} \operatorname{sgn}(\omega_{m/t}) \omega_{m/t}^2 \quad (10)$$

This approach provides satisfying results so that the density change of the surrounding air and the wing-in-ground effect does not have to be analysed in more detail.

The constants k_{fvp} and k_{fhp} were analysed experimentally. For the movement around the two axis we get the momentums M_v and M_h :

$$M_{Av} = F_v l_m = k_{fvp} \operatorname{sgn}(\omega_m) \omega_m^2 l_m \quad (11)$$

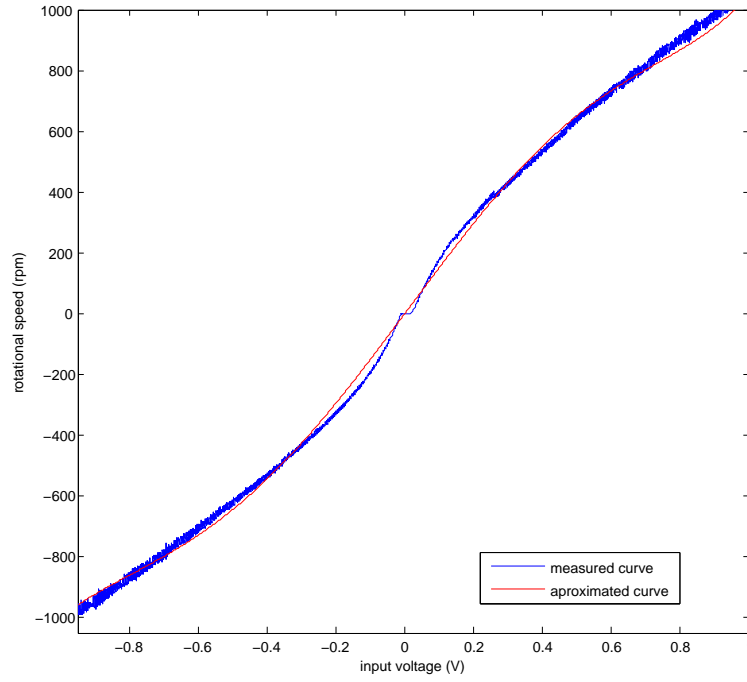


Figure 11: Rotational speed of tail rotor against input voltage

$$M_{Ah} = F_h l_t = k_{fhp} \text{sgn}(\omega_t) \omega_t^2 l_t \quad (12)$$

2.1.5 Moments of Inertia

The moment of inertia of an object is a value that describes how difficult it is to change its angular movement about an axis. The general formula is

$$J = \sum m_i r_i^2 \quad (13)$$

whereas m_i states the particular masses and r_i its distances to the centre of rotation. For some parts of the model that are not mounted symmetric, we need an additional formula to describe an object's moment of inertia that is called "parallel axis theorem". The formula is:

$$J = J_{cm} + md^2 \quad (14)$$

where:

J_{cm} denotes the moment of inertia of the object about the centre of mass

M is the object's mass

d states the perpendicular distance between the two axes

According to figure (5) we can determine the vertical components of the moment of inertia which are independent of the beam's position.

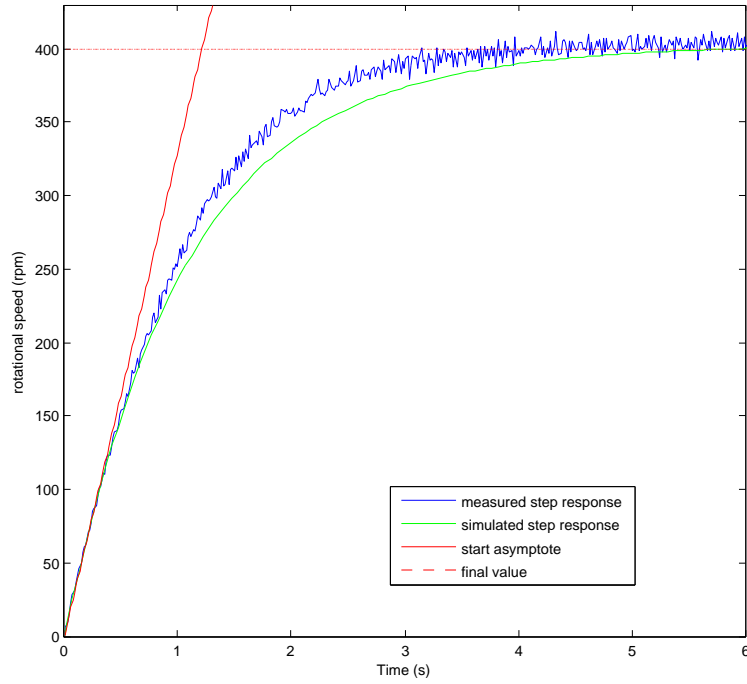


Figure 12: Step Response of the main rotor due to a step of 0,4 V

$$\begin{aligned}
 J_{v1} &= m_{mr} l_m^2 && \text{main rotor} \\
 J_{v2} &= m_m \frac{l_m^2}{3} && \text{beam of main rotor} \\
 J_{v3} &= m_{cb} l_{cb}^2 && \text{counterweight} \\
 J_{v4} &= m_b \frac{l_b^2}{3} && \text{beam of counterweight} \\
 J_{v5} &= m_{tr} l_t^2 && \text{tail rotor} \\
 J_{v6} &= m_t \frac{l_t^2}{3} && \text{beam of tail rotor} \\
 J_{v7} &= \frac{m_{ms}}{2} r_{ms}^2 + m_{ms} l_m^2 && \text{shield main rotor} \\
 J_{v8} &= \frac{m_{ts}}{2} r_{ts}^2 + m_{ts} l_t^2 && \text{shield tail rotor}
 \end{aligned}$$

The vertical moments of inertia results in

$$J_v = \sum_{i=1}^8 J_{vi} \quad (15)$$

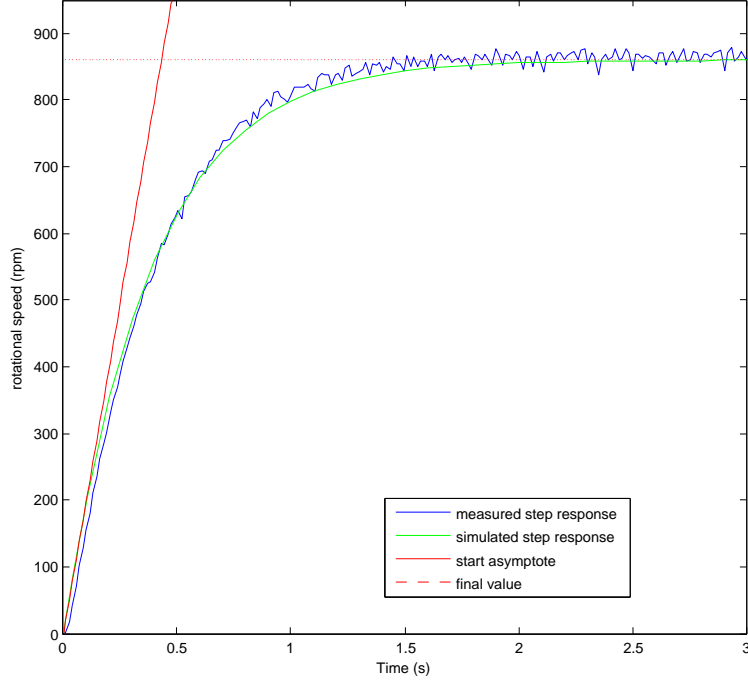


Figure 13: Step Response of the tail rotor due to a step of 0,7 V

Referring to figure (6) we also determine the horizontal components of the moment of inertia which are dependent of the angle α_v .

$$\begin{aligned}
 J_{h1} &= \frac{m_m}{3} (l_m \cos(\alpha_v))^2 && \text{beam of main motor} \\
 J_{h2} &= \frac{m_t}{3} (l_t \cos(\alpha_v))^2 && \text{beam of tail rotor} \\
 J_{h3} &= \frac{m_b}{3} (l_b \sin(\alpha_v))^2 && \text{beam of counterweight} \\
 J_{h4} &= m_{tr} (l_t \cos(\alpha_v))^2 && \text{tail rotor} \\
 J_{h5} &= m_{mr} (l_m \cos(\alpha_v))^2 && \text{main rotor} \\
 J_{h6} &= m_{cb} (l_{cb} \cos(\alpha_v))^2 && \text{counterweight} \\
 J_{h7} &= \frac{m_{ts}}{2} r_{ts}^2 + m_{ts} (l_t \cos(\alpha_v))^2 && \text{shield tail rotor} \\
 J_{h8} &= m_{ms} r_{ms}^2 + m_{ms} (l_m \cos(\alpha_v))^2 && \text{shield main rotor}
 \end{aligned}$$

Putting all horizontal moments of inertia together leads to

$$\begin{aligned}
 J_h = \sum_{i=1}^8 J_{hi} &= \frac{m_m}{3} (l_m \cos(\alpha_v))^2 + \frac{m_t}{3} (l_t \cos(\alpha_v))^2 + \frac{m_b}{3} (l_b \sin(\alpha_v))^2 + m_{tr} (l_t \cos(\alpha_v))^2 + m_{mr} (l_m \cos(\alpha_v))^2 + \\
 & m_{cb} (l_{cb} \cos(\alpha_v))^2 + \frac{m_{ts}}{2} r_{ts}^2 + m_{ts} (l_t \cos(\alpha_v))^2 + m_{ms} r_{ms}^2 + m_{ms} (l_m \cos(\alpha_v))^2 \\
 & = (D \cos(\alpha_v))^2 + E \sin(\alpha_v)^2 + F
 \end{aligned} \tag{16}$$

2.1.6 Friction

The bearing friction effects we have to deal with are to be considered as viscous ones. Hence the resulting friction torque is proportional to the angular velocity:

$$M_{Fv/Fh} = -k_{v/h}\Omega_{v/h} \quad (17)$$

2.1.7 Torsion

Even if the connecting cables for the sensors and motors are very lightweight and hang quite freely there is a torsion force in the horizontal axis that has to be considered. Whether this is very non-linear force depending of the horizontal angle α_h the following linear approximation reaches our demand of accuracy.

$$M_T = -k_c\alpha_h \quad (18)$$

2.1.8 Electro-mechanical Coupling

According to Newton's third axiom each force has a complementary force in the opposite direction. This as well counts referring the moment $M_{Mm/Mt}$ that is built up by the DC motors. The engines create a torque that respectively affect the other axis.

$$M_{MCm} = k_m i_m \cos(\alpha_v) \quad (19)$$

$$M_{MCt} = k_t i_t \quad (20)$$

Like described in the previous chapter the current is not considered in this mathematical model because the delay between voltage and current is quite small we therefore considered as proportional we introduce two new values called k_{cm} for effect of the main rotor in the vertical axis and k_{ct} for the effect of the tail rotor in the horizontal axis. This leads to two new equations:

$$M_{MCm} = k_{cm} u_m \cos(\alpha_v) \quad (21)$$

$$M_{MCt} = k_{ct} u_t \quad (22)$$

Actually the change of the angular moment has to be considered as well but since experiments showed that this effect is only notable by huge changes of the rotation speed it will not be taken into account during the further work on this model.

2.1.9 Centrifugal Force

Centrifugal Force is a term that refers to two different forces which are related to rotation. Both of them are oriented away from the axis of rotation. The force that is oriented towards the axis is called centripetal force. The general formula of the Centrifugal Force is the following:

$$F_{centrifugal} = m\omega^2 r_{\perp} \quad (23)$$

If so we get for the torque that comes from the Centrifugal Force

$$M_C = -\omega_h^2 \left[\left(\frac{m_t}{2} + m_{tr} + m_{ts} \right) l_t + \left(\frac{m_m}{2} + m_{mr} + m_{ms} \right) l_m + \frac{m_b}{2} l_b^2 + m_{cb} l_{cb}^2 \right] \sin(\alpha_v) \cos(\alpha_v) \quad (24)$$

or in short form

$$M_C = -\frac{1}{2} \omega_h^2 \sin(2\alpha_v) (A l_t + B l_m + \frac{m_b}{2} l_b^2 + m_{cb} l_{cb}^2) \quad (25)$$

what finally leads to

$$M_C = -\frac{1}{2} \omega_h^2 \sin(2\alpha_v) H \quad (26)$$

2.2 Coupled non-linear System

The so far described equations were all considered as totally independent of each other. For Modelling the whole system all couplings between the horizontal and the vertical system have to be considered. Putting all together we get the following composition of differential equations:

$$\frac{d\alpha_v}{dt} = \frac{1}{J_v} (M_{Av} - M_{Fv} + M_G - M_C + M_{MCt}) = \quad (27)$$

$$\frac{k_{fvpsgn}(\omega_m) \omega_m^2 l_m - k_v \Omega_v + [(A l_t - B l_m) \cos(\alpha_v) - C \sin(\alpha_v)] g - \frac{1}{2} \omega_h^2 \sin(2\alpha_v) H + k_t i_t}{J_v} \quad (28)$$

$$\frac{d\alpha_h}{dt} = \frac{1}{J_h} (M_{Ah} - M_{Fh} - M_T + M_{MCm}) = \quad (29)$$

$$\frac{k_{fhpsgn}(\omega_t) \omega_t^2 l_t \cos(\alpha_v) - k_h \Omega_h - k_c \alpha_h + k_m i_m \cos(\alpha_v)}{J_h} \quad (30)$$

$$\frac{d\bar{\omega}_m}{dt} = -\frac{1}{T_m} \bar{\omega}_m + \frac{1}{T_m} u \quad (31)$$

$$\frac{d\bar{\omega}_t}{dt} = -\frac{1}{T_t} \bar{\omega}_t + \frac{1}{T_t} u \quad (32)$$

having in mind that $M_{Av/h}$ and $\bar{\omega}_{m/t}$ are connected by non-linear relations described in equations (6), (7) and (10) to (12). This result lead to a state-space representation of 6th order described afterwards.

2.3 Validation

To verify the validity of the established model different characteristics of the real model were to compare with the mathematical model. These steps were carried out by measuring the needed step responses of the laboratory model comparing them with step responses obtained by simulations in Matlab/Simulink. Since the characteristics of the DC-motors were already verified in the modelling chapter only the behaviour of the whole system has to be checked.

Since the dynamic characteristics of the vertical system can be compared with a pendulum firstly an oscillation experiment was arranged. Letting the system oscillate freely starting at the point $\alpha_v = 0$ it performs an exponential attenuated oscillation. The consistence between the results of the model and simulation can be seen in figure (14). Later on can be seen that the idle state is at $\alpha_v = -0,3$

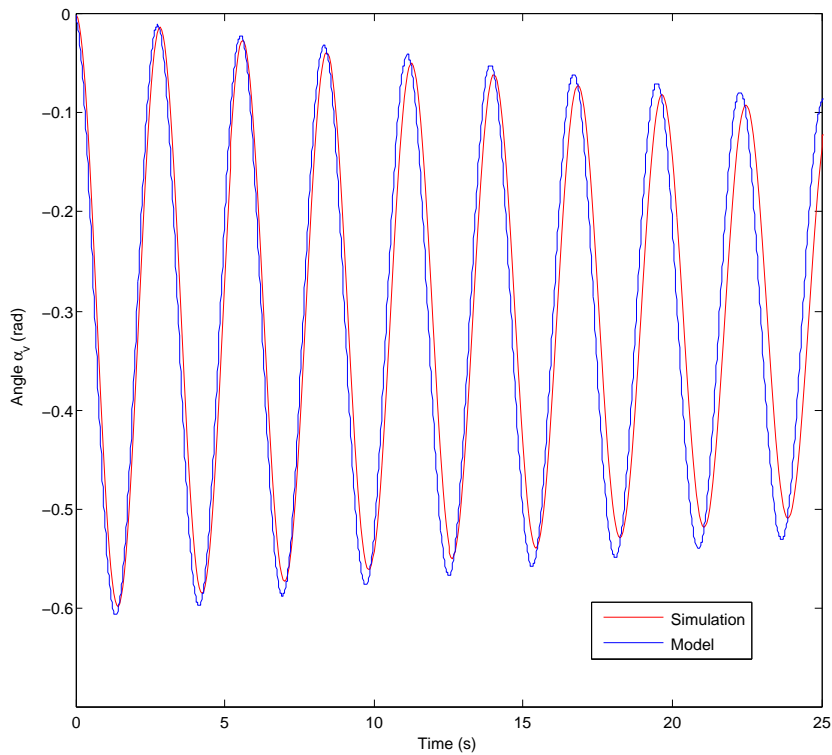


Figure 14: Oscillation experiment of the vertical part system

The next part is to verify is the behaviour of the system respective an input step of the main rotor from $u_v = 0$ to $u_v = 0,4$ that can be seen in figure (15).

The simulation shows small variances referable the model due to the approximation of the rotor system considered as first-order-lags like already mentioned beforehand. Summing up the simulation results are precise enough for designing a controller.

To validate the horizontal part system we consider as well the comparison between the step responses of the model as the simulation like seen in figure (16). Unfortunately the results are not as similar as desired due to the very non-linear effect of torsion. But for the needed purposes it is still exact enough to work with.

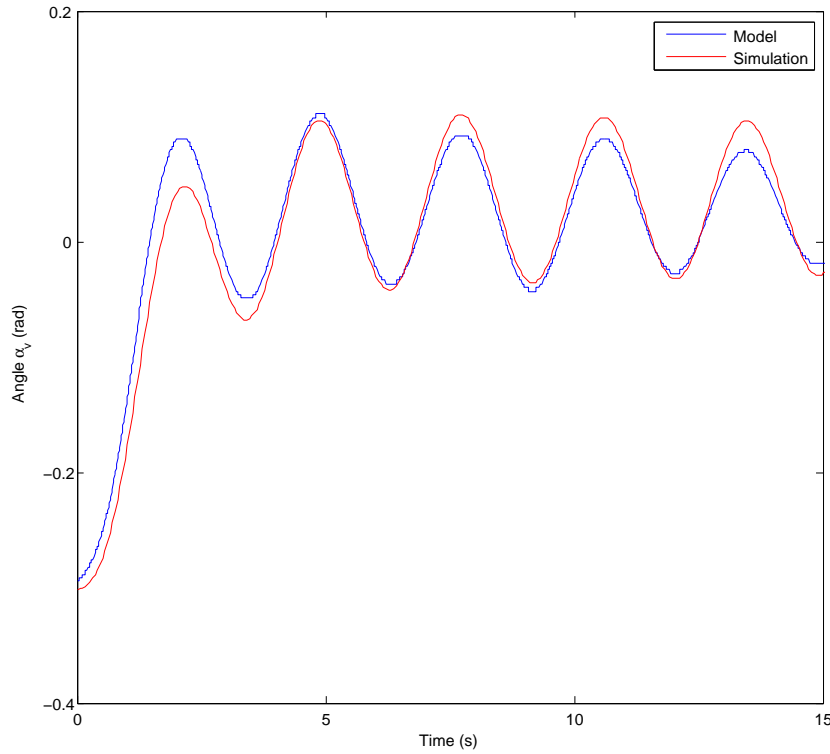


Figure 15: Main Rotor Step from $u_v = 0$ to $u_v = 0, 4$

Since now the part systems are verified there is to check the coupled effects between the horizontal and the vertical subsystem. In the first instance we compare the step response of a step to the main rotor taking effect to the horizontal angle like seen in figure (17).

Like the reader can see, after 10 seconds the two processes diverge quite hard due to the non-linearity of torsion that is approximated as linear in the working point of $\alpha_h = 0$. Due to the fact that we are dealing with an angle of α_h between 1 *rad* and 2 *rad*, that will not happen when we will work in the proximity afterwards controlling the system, the simulation are precise enough.

At last there is to verify the behaviour of the system in horizontal axis respective the input of the tail rotor. In figure (18) we see the sequence plots of both the simulation and the model concerning a step from $u_h = 0$ to $u_h = 0, 4$.

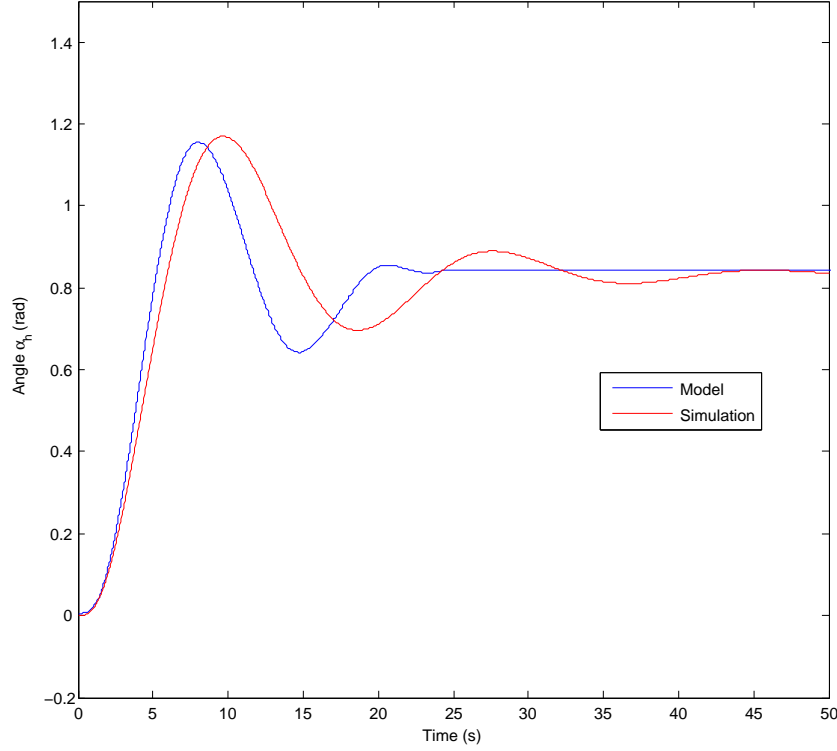
Since the influence of the tail rotor to the vertical system in general is very small the deviation between the model and the simulation will not carry authority.

Hence all characteristics of the non-linear model are validated.

2.4 State-Space Representation

A non-linear, time-invariant system in general can be described completely by two equations:

$$\dot{x} = f(x, u) = A_{nl}x + B_{nl}u \quad (33)$$

Figure 16: Tail rotor Step from $u_h = 0$ to $u_h = 0, 2$

$$y = g(x) = C_{nl}x \quad (34)$$

whereas A_{nl}, B_{nl}, C_{nl} are the system matrices, x is the state vector, u the input vector and y the output vector.

The described equations mentioned above can now be formed as State-Space equations as follows:

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & a_{25} & 0 \\ 0 & 0 & a_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & a_{54} & a_{55} & a_{56} \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 0 & b_{22} \\ b_{31} & 0 \\ 0 & 0 \\ b_{51} & 0 \\ 0 & b_{62} \end{pmatrix} u \quad (35)$$

$$y = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} x \quad (36)$$

with the states $x_1 = \alpha_v$, $x_2 = \Omega_v$, $x_3 = \bar{w}_m$, $x_4 = \alpha_h$, $x_5 = \Omega_h$, $x_6 = \bar{w}_t$ and the inputs $u_1 = u_v$ and $u_2 = u_h$. The variables describe the following equations:

$$a_{21} = \frac{1}{J_v} ((A_{lt} - B_{lm}) \cos(\alpha_v) - C \cos(\alpha_v)) g \quad (37)$$

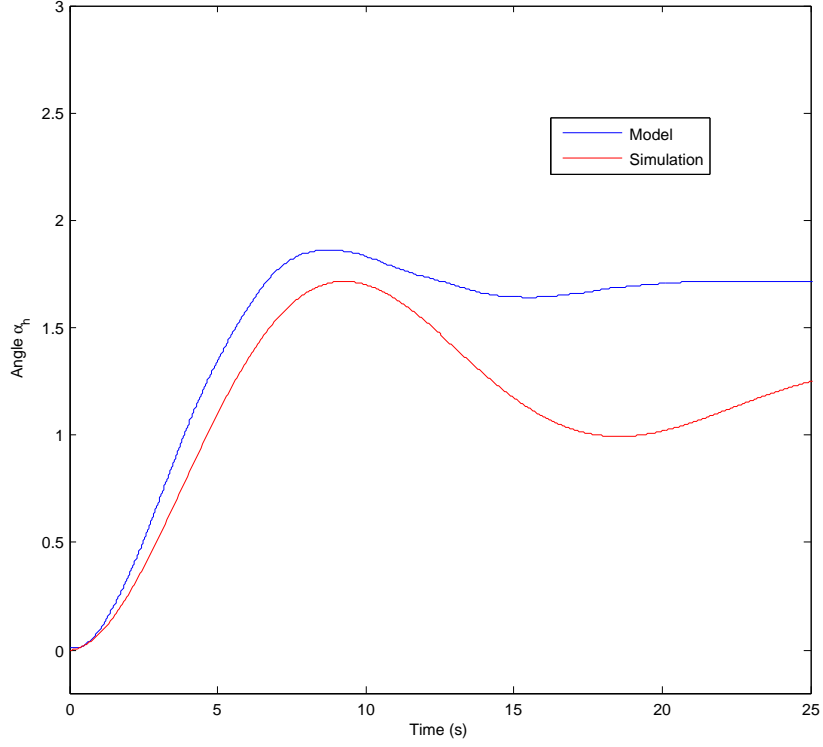


Figure 17: Behaviour of the system in the horizontal angle respective a main rotor step from $u_v = 0$ to $u_v = 0,4$

$$a_{22} = \frac{1}{J_v}(-k_v) \quad (38)$$

$$a_{23} = \frac{1}{J_v}(F_v(W_m(\bar{w}_m))) \quad (39)$$

$$a_{25} = \frac{1}{J_v}(-0,5\omega_h^2 \sin(2\alpha_v)H) \quad (40)$$

$$a_{33} = -\frac{1}{T_m} \quad (41)$$

$$a_{54} = \frac{-k_c}{D\cos(u)^2 + E\sin(u)^2 + F} \quad (42)$$

$$a_{55} = \frac{-k_h}{D\cos(u)^2 + E\sin(u)^2 + F} \quad (43)$$

$$a_{56} = \frac{F_h(W_t(\bar{w}_t)) \cos(\alpha_v)}{D\cos(u)^2 + E\sin(u)^2 + F} \quad (44)$$

$$a_{66} = -\frac{1}{T_t} \quad (45)$$

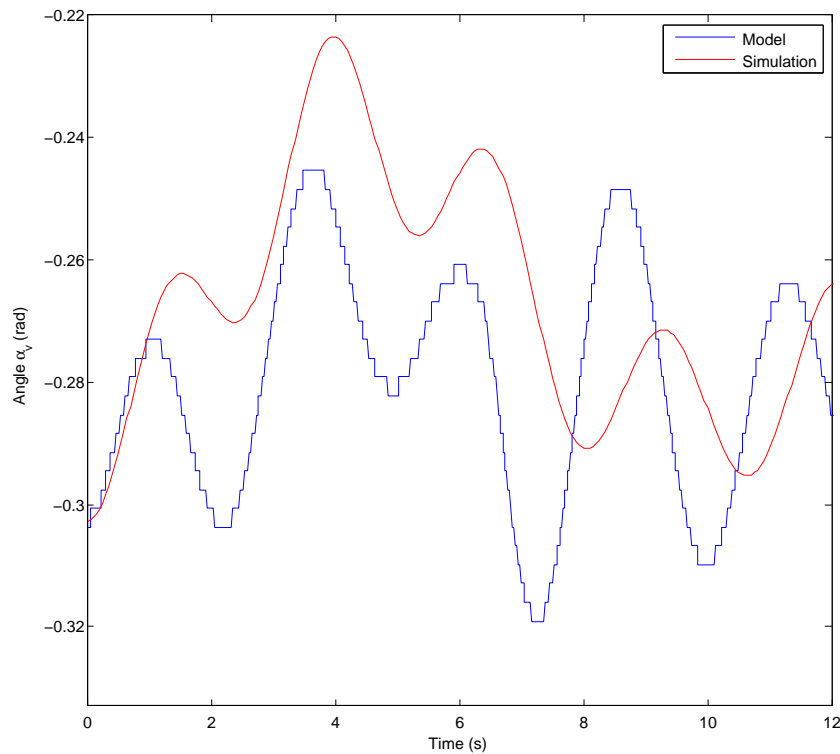


Figure 18: Behaviour of the system in the vertical angle respective a tail rotor step from $u_h = 0$ to $u_h = 0,4$

$$ba_{22} = \frac{k_{mc}}{J_v} \quad (46)$$

$$b_{31} = \frac{1}{T_m} \quad (47)$$

$$b_{51} = \frac{k_{cm} \cos(\alpha_v)}{D \cos(u)^2 + E \sin(u)^2 + F} \quad (48)$$

$$b_{52} = \frac{1}{T_t} \quad (49)$$

2.4.1 Linearised Continuous System

To use the method of designing an observer control like described in chapter 4, a linearised model, based on the original one, is needed.

As operation points of the controller the values $\alpha_v = 0$ and $\alpha_h = 0$ where chosen to hold the helicopter at the given position and to let it follow small trajectories in the proximity.

By partial derivative of the non-linear equations (35) and (36) we obtain the linear state-space representation.

$$\Delta\dot{x} = A\Delta x + B\Delta u \quad (50)$$

$$\Delta\dot{y} = C\Delta x \quad (51)$$

with

$$A = \left. \frac{\partial f}{\partial x} \right|_{x_{OP}, u_{OP}} = \left[\begin{array}{ccc} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{array} \right] \Bigg|_{x_{WP}, u_{WP}} \quad (52)$$

$$B = \left. \frac{\partial f}{\partial u} \right|_{x_{OP}, u_{OP}} = \left[\begin{array}{ccc} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_n} \end{array} \right] \Bigg|_{u_{WP}, u_{WP}} \quad (53)$$

and

$$C = \left. \frac{\partial g}{\partial x} \right|_{x_{OP}} = \left[\begin{array}{ccc} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_n} \end{array} \right] \Bigg|_{x_{WP}} \quad (54)$$

Linearising the system at each non-linear point of the system leads to the matrices as follows:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -4,738 & -0,032 & 5,660 & 0 & 0 & 0 \\ 0 & 0 & -0,752 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -0,123 & -0,186 & 1 \\ 0 & 0 & 0 & 0 & 0 & -2,326 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 0 \\ 0 & 0,239 \\ 0,752 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 2,326 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

The particular steps of the linearisation can be found in the appendix

3 Control Design

For control unit design there are a lot of different methods both in time and frequency domain. In matters of designing a control for a multiple-input multiple-output systems normally the design is done in time domain. Moreover the system is already characterised by equations in time domain so that no more conversions are needed.

The general benefit of designing a controller in time domain is to get insight into the particular states of the system. Therefore the system can be controlled with detailed eyemark on certain behaviour. By now many design methods in time domain for MIMO-Systems have been developed. The two most-established ones are control unit design by "Polvorgabe" and design according to Riccati. Concerning the method of "Polvorgabe" the circuit's eigenvalues are specifically placed to achieve the desired behaviour of the controlled circuit. Regarding the Riccati-design the feedback matrix is optimized by given loading matrices. Therefore the shift of eigenvectors can only be done indirectly using this method.

3.1 State Control

In this chapter a small overview about designing a state control using Riccati-method will be given. Beyond the later on used loading matrices will be presented.

Structure of a State-Control

Generally we deal with a linear time-invariant system that can be described as follows:

$$\dot{x} = Ax + Bu \tag{55}$$

$$y = Cx \tag{56}$$

Figure (19) shows the block structure of a general system using the equations (55) and (56).

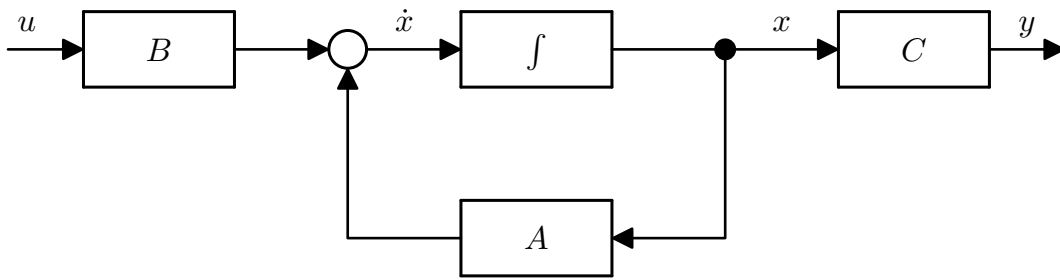


Figure 19: General state-space system

The general ansatz for a linear, time-invariant state-feedback is:

$$w = -Kx + u \quad (57)$$

The K-matrix is constant and the feedback input is added with negative sign like already known of conventional control circuits.

Combining the equations (57) with (55) we get the equation for the closed loop:

$$\dot{x} = (A - BK)x \quad (58)$$

Graphically presented we can see the the differential equation in figure (20).

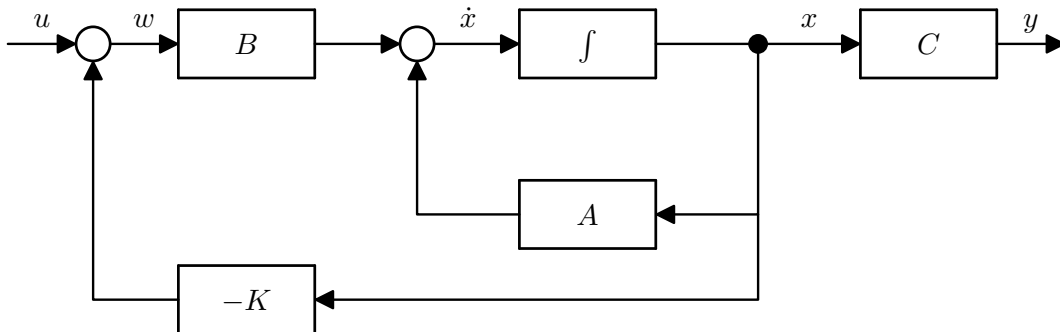


Figure 20: Controlled state-space system

Like the reader can see, the speciality of the state control is, that each state is traced back by the control-matrix K. Since $(A - BK)$ is the modified system matrix, modifying the matrix K means changing the whole system behaviour.

Like to see in figure (20) the particular states are caught within the system before passing the C-matrix. In the minority of cases it is possible to know the states due to the output vector. Neither it will always be feasible to measure the whole state-vector by sensors because of either cost reasons or physical impossibility. Therefore it is necessary to at least estimate the state vector by a Luenberger-Observer, what will be described in more detail during the next chapter.

The system is now controlled by information about the actual state but non-measurable disturbance is not

taken into account. The solution to maintain stationary accuracy is to introduce one or more states (depending on the output vector) by integrating the system output like shown in figure (21).

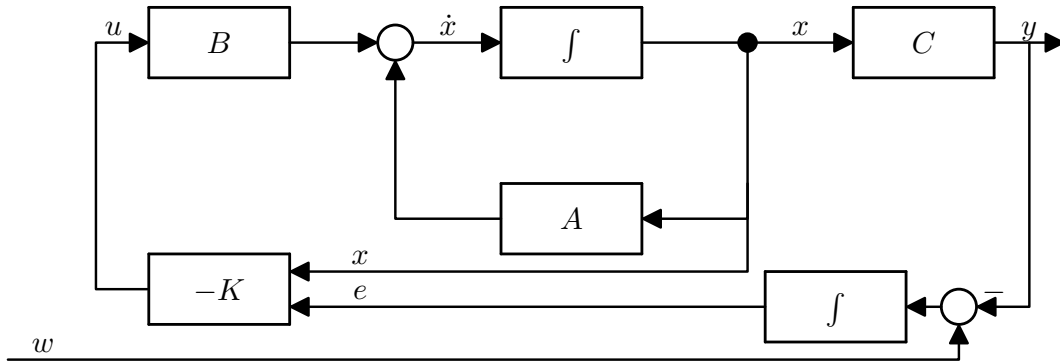


Figure 21: Controlled extended state-space system

For designing the control now by using the Riccati-method we also need an extended model, that is built as follows:

$$\dot{\tilde{x}} = \underbrace{\begin{pmatrix} A & 0 \\ -C & 0 \end{pmatrix}}_{\tilde{A}} \tilde{x} + \underbrace{\begin{pmatrix} B \\ 0 \end{pmatrix}}_{\tilde{B}} u \quad (59)$$

$$u = -\tilde{K} \underbrace{\begin{pmatrix} x \\ e \end{pmatrix}}_{\tilde{x}} \quad (60)$$

$$\dot{e} = w - y \quad (61)$$

Defining the controller matrix K by a quadratic functional

Now the design of the matrix K using the Riccati-method will be presented to get an optimal control. Firstly there is the need for a performance index, that defines what is optimal:

$$J(u) = H(x(t_e)) + \int_{t_0}^{t_e} L(x(t), u(t)) dt \quad (62)$$

Designing a control by the Riccati-method the quadratic cost functional is used:

$$G(x(t_e)) = \frac{1}{2} x^\top(t_e) S x(t_e) \quad (63)$$

und

$$L(x(t), u(t)) = \frac{1}{2} (x^\top(t) Q x(t) + u^\top(t) R u(t)) \quad (64)$$

The matrices S and Q have to be symmetric and positive semidefinite, R symmetric and positive definite.

The actuate vector has to be unrestrained, the optimization aim has to be achieved during the time $t_e - t_0$. The calculation of the controller normally is accomplished by the Hamilton-method what will not be explained in more detail in this paper. Finally it leads to a differential equations as follows:

$$-\dot{P} = Q + PA + A^T P - PBR^{-1}B^T P \quad (65)$$

with constraint

$$P(t_e) = S \quad (66)$$

For $t_e \rightarrow \infty$ the stationary solution is $x(t_e) = 0$ and hence $\dot{P} = 0$.

The equation (65) can be simplified to:

$$PBR^{-1}B^T P - PA - A^T P = Q \quad (67)$$

The quadratic performance index as well is simplified to:

$$J = \frac{1}{2} \int_{t_0}^{\infty} [x^T(t)Qx(t) + u^T(t)Ru(t)] dt \quad (68)$$

Solving this equation for P leads directly to the requested feedback matrix.

$$K = R^{-1}B^T P \quad (69)$$

Hence now only the loading matrices Q and R have to be chosen adequately.

Criteria for choosing the loading matrices

The selection of loading matrices is very difficult to that effect that clear design rules are vacant. There are only some experimental rules based on experience that are not usable generally. Therefore the only possibility is the "try and error"-method. First of all the problem specification has to be clearly formulated. In this case it was formulated as follows:

- the system does not overshoot more than 5 % in the proximity ($0 \pm 0,1$ rad) of the operating point following an input step impulse.
- the system stabilize in the requested point as fast as possible.

Then there is to modify and test the matrices Q and R as long as the desired control behaviour is reached. Q affects the system states, R the input states. It is very important to take into account the input constraints by modifying the R-matrix. For calculating K Matlab offers functions called *lqr* for continuous systems and *lqrd* for discrete ones. The simultion was carried out by Simulink.

You get the best result by using the matrices as follows:

$$Q = \begin{bmatrix} 30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 17 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 120 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 700 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 900 \end{bmatrix} \quad (70)$$

$$R = \begin{bmatrix} 16 & 0 \\ 0 & 16 \end{bmatrix} \quad (71)$$

Using the Matlab function *lqr* we obtain the control matrix:

$$K = \begin{bmatrix} 1,79 & 2,90 & 5,79 & 0,90 & -0,07 & -0,11 & -6,51 & -1,33 \\ 0,99 & -0,55 & -2,88 & 10,41 & 6,81 & 1,68 & 1,17 & -7,38 \end{bmatrix} \quad (72)$$

3.2 Observer

Like already mentioned beforehand the whole state vector has to be available to control the system. To make the vector available or at least a it's reconstruction we use the so called Luenberger-Observer. The idea is based on knowing the control process quite well so that the particular states can be calculated by running the mathematical model parallel to the physical process comparing at each time the output states to correct the observers inner states by feedbacking the observer error through the matrix L . The so obtained state vector \hat{x} is used for control. The mode of operation can be seen in figure (22).

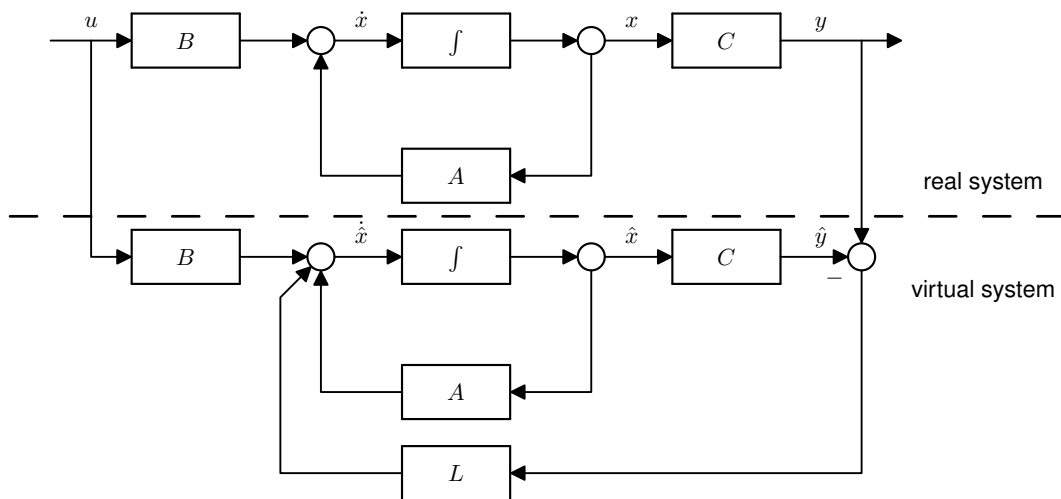


Figure 22: General Luenberger Observer

Hence for the observer we obtain the equation

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly \quad (73)$$

The obtained estimation error is $\tilde{x} = x - \hat{x}$ and should decay to 0. Having a closer look at the differential equation of the estimation error we get:

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}} = Ax + Bu - (A\hat{x} - LC\hat{x} + Bu + Ly) \quad (74)$$

with $y = Cx$ we get

$$\dot{\tilde{x}} = Ax - A\hat{x} - LC(x - \hat{x}) = (A - LC)\tilde{x} \quad (75)$$

If the eigenvalues of the dynamic matrix $(A - LC)$ are on the left side of the imaginary-axis, the estimation error will decay to 0. But since the error should be compensated even if the states are varying, the observer system has to be "faster" than the control system, what means that the eigenvalues have to be on the left side of the control system's eigenvalues. A rule of thumb does not exist since the placing of poles always depends on a lot of factors. Placing the poles too near to the system's poles the estimation error will decay very slowly, placing them too far on the left the measurement noise will be gained too much.

Since we are now designing an observer for the physical model we have to take into account that only original states and the controlling of these states are to be considered. That means that we only consider the dynamics of the equation $(A - B\tilde{K})$ whereas \hat{K} represents the first 6 columns of K .

The poles of $(A - B\tilde{K})$ lie at:

- -1,1021 + 3,0797i
- -1,1021 - 3,0797i
- -1,4014 + 2,3597i
- -1,4014 - 2,3597i
- -2,9219
- -3,4610

We got good results by placing the observer's poles around 3 times smaller than the fastest pole of the system that is $-3,4610$. So the poles were placed at

- -10,0
- -10,2

- -10,1
- -9,9
- -9,8
- -10,3

Using the Matlab function *place* we obtain the observer vector as follows:

$$L = \begin{bmatrix} 29,37 & 275,44 & 142,18 & 0,12 & 2,19 & 7,44 \\ 0,17 & 3,45 & 2,72 & 27,60 & 232,76 & 459,85 \end{bmatrix} \quad (76)$$

3.3 Control Results

Putting the now calculated parts of the control together we got quite satisfying results. Putting a step impulse on the vertical system we obtain overshooting of less than 5 % and achievement of the 5 % band within 1,9 seconds like shown in figure (23).

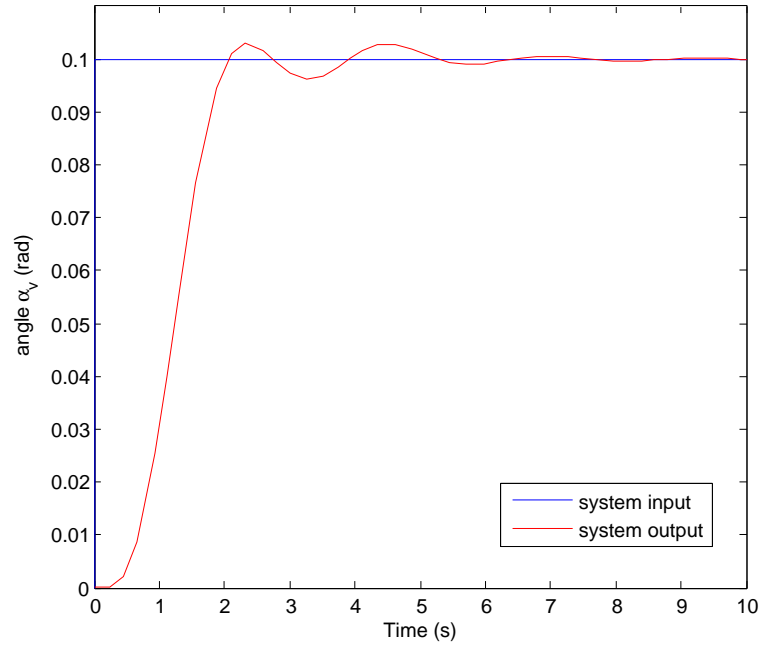


Figure 23: Vertical reference input step from 0 to 0,1

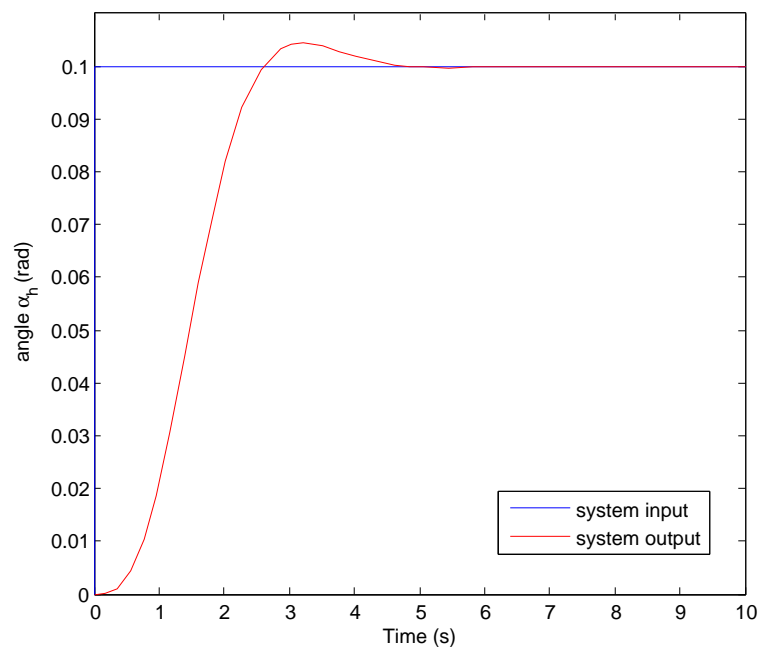


Figure 24: horizontal reference input step from 0 to 0,1

As we see in figure (24) the horizontal system as well shows overshooting of less than 5 % as well and the 5 % band is reached within 2,4 seconds. Even while controlling both systems together putting a square signal to both inputs, each one with an amplitude of 0,1 *rad* with frequencies of 0,05 *Hz* to the verical one, 0,03 *Hz* to

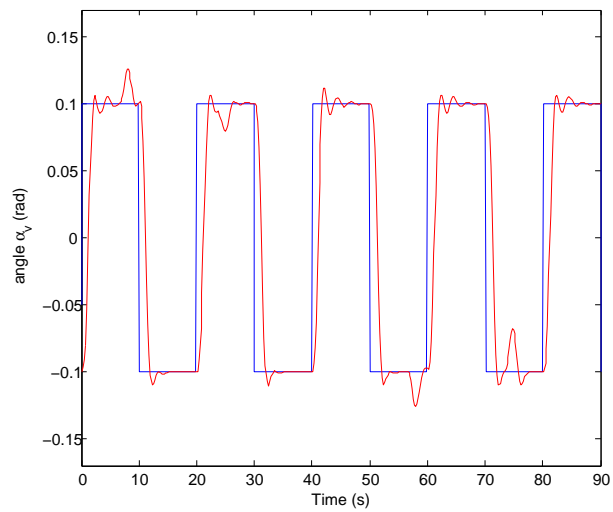


Figure 25: Behaviour in vertical angle concerning input reference of a square signal

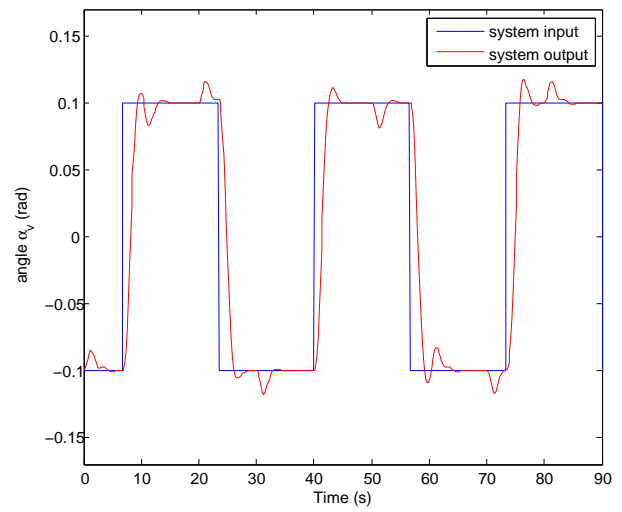


Figure 26: Behaviour in horizontal angle concerning input reference of a square signal

the horizontal one, we obtain good results with fast stabilizing times and compensation of the each other input reference like shown in figure (25) and (26).

3.4 Transfer to Discrete Time Domain

Since ultimately the physical model cannot be controlled by a continuous controller, the system has to be discretised. Controlling a system by a digital controller we always have to deal with dead time elements that come from the sampling time. Depending on the ratio between the systems's and controller's velocity there are two kinds of digital controllers. Sampling the systems's output that often that the step function is almost identical to the continuous function, the controller can be designed as a "Quasi-Continuous Controller".

"Often" in this sense means that the slowest system's time constant is 5 - 10 times greater than the the sampling time. Using this method everything is designed continuously. If the sampling time is slower, the dead time elements have to be taken into account and the design is done in the discrete z-domain.

A sampling time of 20 ms was used since it stated a well elected compromise between fastness and computational effort. Comparing the slowest time constant of the system that is the tail rotors time constant ($T_t = 0,43\text{s}$) it is easy to see that it is possible to design a "Quasi-Continuous" controller. Nevertheless since we are dealing with an observer control, the observer has to be discretised.

Transferring the linear system

Generally the linear system matrix of the type

$$\dot{x} = Ax + Bu \quad (77)$$

$$y = Cx \quad (78)$$

can be discretised to the form as follows:

$$x[k + 1] = A_d x[k] + B_d u[k] \quad (79)$$

$$y[k] = C_d x[k] \quad (80)$$

with $u[k] = u(t = kT)$, $x[k] = x(t = kT)$, $y[k] = y(t = kT)$ and $C_d = C$.

The matrices A_d and B_d are calculated by

$$A_d = e^{AT} \quad (81)$$

$$B_d = \int_0^T e^{A(T-\tau)} B d\tau \quad (82)$$

Matlab offers a function to transfer continuous state-space representations to discrete state-space representations that is called *c2d*. Using this function by passing the system matrices and the sampling time we obtain the discrete system matrices as follows:

$$A_d = \begin{bmatrix} 0,999 & 0,020 & 0,001 & 0 & 0 & 0 \\ -0,095 & 0,998 & 0,112 & 0 & 0 & 0 \\ 0 & 0 & 0,9851 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0,020 & 0 \\ 0 & 0 & 0 & -0,002 & 0,996 & 0,019 \\ 0 & 0 & 0 & 0 & 00 & 0.9546 \end{bmatrix} \quad (83)$$

$$B_d = \begin{bmatrix} 0 & 0 \\ 0,0008 & 0,0048 \\ 0,0149 & 0 \\ 0,0001 & 0 \\ 0,0077 & 0 \\ 0 & 0,0454 \end{bmatrix} \quad (84)$$

$$C_d = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (85)$$

Transferring the Controller Matrix

The computing of the the discrete Controller Matrix was carried out by the function *lqrd* by passing the continuous system matrices and the sampling time. The computing of K_d works almost in the same way as *lqr* concerning the algorithm. The only difference is that the Integral of the quadratic performance index shown in equation (68) is substitutet by the discrete summation

$$J = \frac{1}{2} \sum [x^\top(k)Qx(k) + u^\top(k)Ru(k)] \quad (86)$$

Using the function *lqrd* we obtain the control-matrix K_d

$$K_d = \begin{bmatrix} 1,64 & 2,79 & 5,67 & 0,83 & -0,08 & -0,11 & -6,24 & -1,24 \\ 0,96 & -0,53 & -2,81 & 10,08 & 6,64 & 1,64 & 1,16 & -7,10 \end{bmatrix} \quad (87)$$

Transferring the Observer Matrix

Firstly the Observer Matrix L was designed by pole shifting what means placing the new poles we get due to the dynamic matrix $(A - LC)$ that far on the left of the original poles on the real axis of the s-plane that we get satisfying results as a compromise between fastness and insensitivity against measure noise. Since we are now dealing with a discrete system the poles no longer have to be placed in the s-layer but in the z-plane. The poles generally can be transformed by the following equation:

$$z = e^{Ts} \quad (88)$$

what is the general relation between z-domain and s-domain.

Hence the beforehand approved poles are transformed and afterwards placed. As new poles in z-domain we obtain

- 0,8187
- 0,8155
- 0,8171
- 0,8204
- 0,8220
- 0,8138

To obtain the discrete observer matrix L_d the Matlab function *place* was used. The computed Matrix L_d is:

$$L_d = \begin{bmatrix} 0.53 & 4.33 & 2.10 & -0.00 & -0.03 & -0.10 \\ -0.00 & -0.05 & -0.04 & 0.50 & 3.68 & 6.68 \end{bmatrix} \quad (89)$$

Results

To test the behaviour of the designed discrete observer all controller matrices in the Simulink model were substituted by the discrete ones, the continuous integrators were substituted by discrete integrators. The simulation was carried out using a sampling time of *20 ms* both for the simulation and for the controller. The tests were again the same as before, a step-impulse of each *0,1 rad* on the reference-input. Like to see in figure (27) and (28) there is no notable difference between discrete and continuous control.

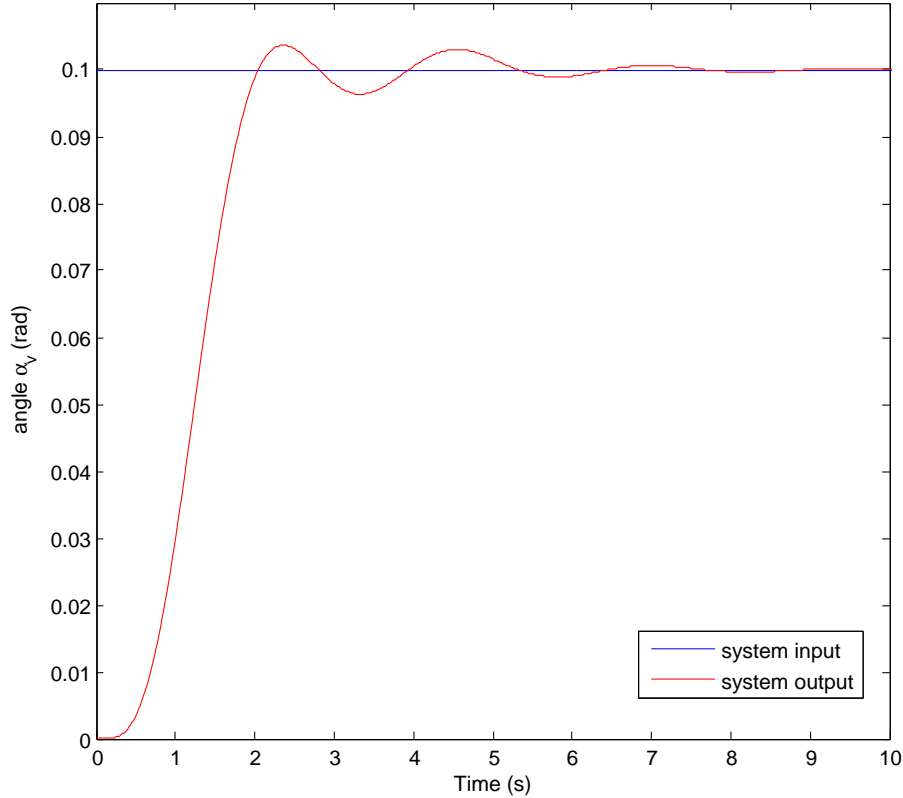


Figure 27: Vertical reference input step from 0 to 0,1 rad using discrete controller

4 Controlling the System

To avoid damages of the real helicopter model and to gain insight to the inner states of the system during the first tests of the controller, the mathematical model firstly was implemented in Labview. For further works moreover there was a PXI-1002 manufactured by National Instruments available that later on was used both to simulate the model and to control it. The PXI-1002's great benefit is the real time analog input/output cards. So it is possible to send any value of the simulated system to an analog output in real time. As well it is possible to read any input value from an analog input channel. A screenshot of the designed frontend can be seen in figure (29).

4.1 Mathematical model controlled by PXI

Firstly the discrete non-linear mathematical system of Matlab was completely transferred to Labview real-time-loop and afterwards tested by characteristic experiments. Slightly disparity results from different simulation algorithms used by the the different programs Matlab and Labview.

In the same circuit the in Matlab designed discrete controller was implemented afterwards. The connections between the two systems, model and controller, were firstly connected directly within the loop, after successful testing they were replaced by physical connections carried out by cables and AD/DA converter to take into account the measure noise. Like shown in figure (30) and (31) the results are still satisfying reaching the obligations.

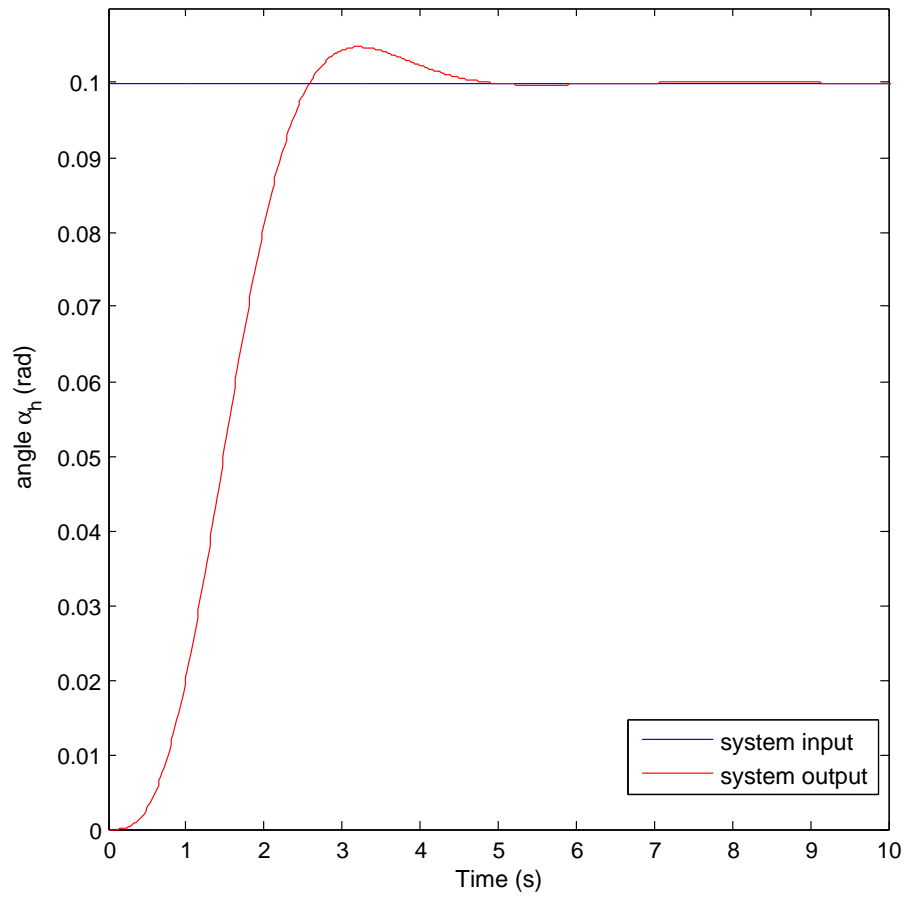


Figure 28: Horizontal reference input step from 0 to 0,1 rad using discrete controller

4.2 Physical model controlled by PXI

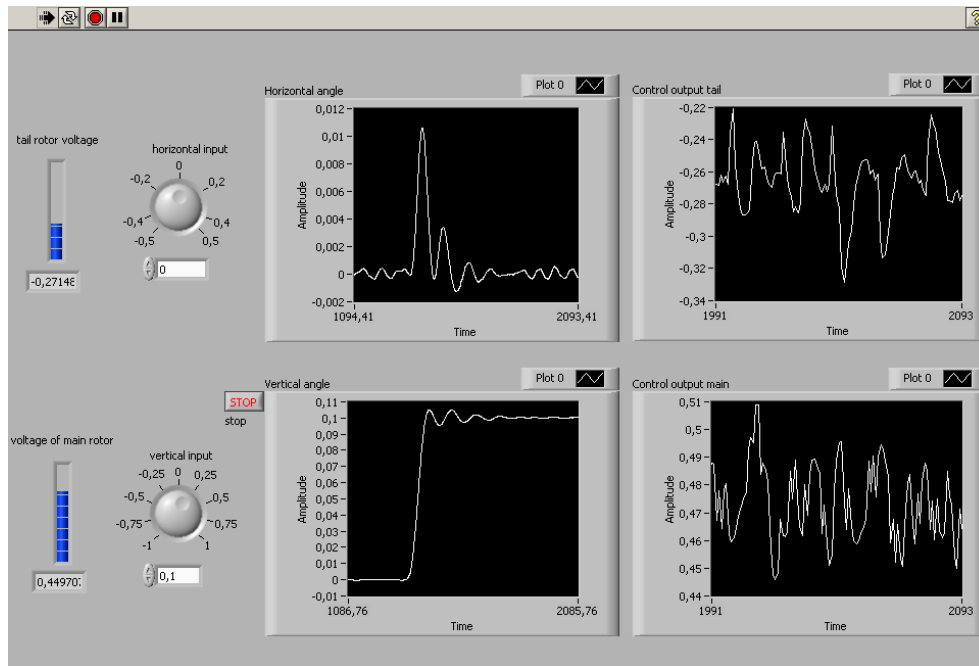


Figure 29: Screenshot of the frontend for controlling the model by LabView

5 Conclusion and Future Works

To come to an conclusion it was finally possible to control the helicopter model by a state-space controller around an operating point. The particular steps were building up the physical model, linearising it, creating the state-space representation, designing the control and the observer and finally discretising it. Since these are the general steps for designing a control this project work presents quite well the procedure and approach of working with an control-circuit even if the model that was to control was more or less an artificial one designed for students education matters. Ideas for future works on the model to improve the control action would be analyse the mechanical system in more detail to take into account the non-linearities referring for example the torsion force that were neglected in this paper. Moreover the measure noise could be compensated to get a faster engaging observer by implementing a Kalman-Filter. Another idea would be to take a complete different approach of controlling by implementing for example a Model Predictive Control. In addition the discrete controller could be implemented on a stand-alone-system like on a microcontroller. But these ideas would have all gone beyond the scope of this project work that was limited to period of four months.

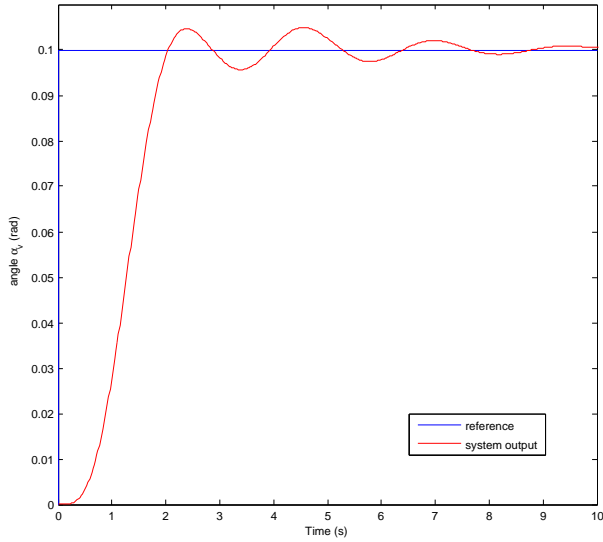


Figure 30: Vertical reference step from 0 rad to 0,1 rad

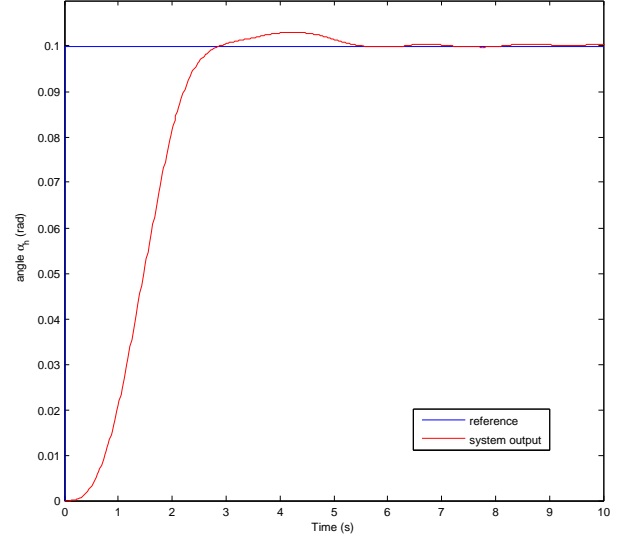


Figure 31: Horizontal reference step from 0 rad to 0,1 rad

6 Appendix

6.1 List of used constants

Parameter	Value	Unit	Parameter	Value	Unit
m_{mr}	0,125	kg	l_m	0,228	m
m_{ms}	0,15	kg	l_t	0,282	m
m_{tr}	0,11	kg	l_b	0,260	m
m_{ts}	0,09	kg	l_{cb}	0,230	m
m_m	0,014	kg	r_{ms}	0,155	m
m_{cb}	0,068	kg	r_{ts}	0,1	m
m_b	0,022	kg			
m_t	0,015	kg			
k_{cm}	0,0134	$\frac{N}{A}$	k_{ct}	0,009	$\frac{N}{A}$
T_m	1,33	s	T_t	0,43	s
g	9,81	$\frac{N}{kg}$	J_v	0,0376	kgm
k_{fvp}	$1,71 \times 10^{-6}$	$\frac{N}{rpm^2}$	k_{fhp}	$1,508 \times 10^{-7}$	$\frac{N}{rpm^2}$
A	0.0585	kgm	B	0.0643	kgm
C	0.0185	kgm	D	0.0308	kgm^2
E	0.0041	kgm^2	F	0.0041	kgm^2
H	0.0355	kgm^2			

The parameters $R_{m/t}$, $L_{m/t}$, $k_{m/t}$, $k_{Bm/Bt}$, $k_{Am/At}$, $J_{m/t}$, $J_{Am/At}$, $J_{Pm/Pt}$, do not appear at this place since the part of the model wherein these constants take place was substituted like described in chapter 2.1 .

6.2 List of Figures

List of Figures

1	Torque effect on a helicopter and compensation by tail rotor	1
2	Early draft of an helicopter by Leonardo da Vinci	1
3	A helicopter of the type Bell 205 dropping water on fire	1
4	Photo of the model	2
5	Schematic model in vertical plane	3
6	Schematic model in horizontal plane	3
7	Block diagramm of the coupled motor-propeller system	4
8	Block diagramm of the simplified coupled motor-propeller system considered as a first-order-lag element	5
9	Block diagramm of the simplified coupled motor-propeller system considered as a first-order-lag element with additional non-linear gain	5
10	Rotational speed of main rotor against input voltage	6
11	Rotational speed of tail rotor against input voltage	7
12	Step Response of the main rotor due to a step of 0,4 V	8
13	Step Response of the tail rotor due to a step of 0,7 V	9
14	Oscillation experiment of the vertical part system	12
15	Main Rotor Step from $u_v = 0$ to $u_v = 0,4$	13
16	Tail rotor Step from $u_h = 0$ to $u_h = 0,2$	14
17	Behaviour of the system in the horizontal angle respective a main rotor step from $u_v = 0$ to $u_v = 0,4$	15
18	Behaviour of the system in the vertical angle respective a tail rotor step from $u_h = 0$ to $u_h = 0,4$	16
19	General state-space system	19
20	Controlled state-space system	19
21	Controlled extended state-space system	20
22	General Luenberger Observer	22
23	Vertical reference input step from 0 to 0,1	25
24	horizontal reference input step from 0 to 0,1	25
25	Behaviour in vertical angle concerning input reference of a square signal	26
26	Behaviour in horizontal angle concerning input reference of a square signal	26
27	Vertical reference input step from 0 to 0,1 rad using discrete controller	30
28	Horizontal reference input step from 0 to 0,1 rad using discrete controller	31
29	Screenshot of the frontend for controlling the model by LabView	32

30	Vertical reference step from 0 rad to 0,1 rad	33
31	Horizontal reference step from 0 rad to 0,1 rad	33

6.3 Literature

References

- [1] Josep Tornero Montserrat, Leopoldo Armesto Ángel: *Control Óptimo*, Editorial UPV, 2007.
- [2] Günther Roppenecker: *Written notes of the lecture "Regelungstechnik"*, lecture at "Friedrich-Alexander-Universität Erlangen-Nürnberg", WS 2006/2007
- [3] Günther Roppenecker: *Written notes of the lecture "Synthese linearer Regelung"*, lecture at "Friedrich-Alexander-Universität Erlangen-Nürnberg", SS 2007
- [4] Thomas Moor: *Written notes of the lecture "Modellbildung und Simulation"*, lecture at "Friedrich-Alexander-Universität Erlangen-Nürnberg", WS 2006/2007
- [5] Thomas Moor: *Written notes of the lecture "Prozessautomatisierung"*, lecture at "Friedrich-Alexander-Universität Erlangen-Nürnberg", SS 2007
- [6] A. Rahideh, M. H. Shaheed: *Mathematical dynamic modelling of a twin-rotor multiple input - multiple output system*, Department of Engineering, Queen Mary, University of London, London, UK, 2006
- [7] Ralf Geßlein: *Entwurf von Reglern reduzierter Ordnung und Entwicklung einer Workbench*, dissertation at "Universität der Bundeswehr", Munich, 2002.