Model Predictive Control to Autonomous Helicopter Flight

P. J. H. Zillinger Molenaar

DCT 2007.35

Traineeship report

Coach(es): -

Supervisor: Prof.dr.ir. M. Steinbuch

Technische Universiteit Eindhoven
Department Mechanical Engineering
Dynamics and Control Technology Group

Eindhoven, April, 2007

Model Predictive Control to Autonomous Helicopter Flight

Philippe Zillinger Molenaar

Department of Control Engineering
Aalborg University
2006

Supervisor: Ph.D. MSc. Anders la Cour-Harbo

Contents

Chapter 1

Introduction

1.1 Helicopter control

Piloting a helicopter is a complex task, which only should be performed by a qualified operator. In this project the autonomous flight of helicopter based unmanned aerial vehicles (HUAV) is simulated. The helicopter model used, which is already available [7], has been obtained from the project helicopter. This model has been implemented in a simulation environment under Matlab's toolbox Simulink. It will be tried to design a control system to achieve autonomous flight. The goal is to control the trajectory of the flight path of this helicopter model using Model Predictive Control, with use of Matlab's MPC toolbox. First a stabilizing hover just above ground will be tried to be achieved, to try and move to more complex manoeuvres from there. To guarantee stability of the unstable non-linear helicopter model in the operating point specified, LQR control has also been used.

1.2 Structure of the work

In Chapter 2 the structure of the helicopter model will be shortly discussed. In chapter 3 the control design and necessary linearization for this are presented. In Chapter 3 stability, LQR and MPC control are explained. Chapter 4 deals with the simulations of the helicopter with MPC and LQR feedback control. Chapter 5 gives conclusions and recommendations of the work done.

Chapter 2

Helicopter model

2.1 Helicopter description

The project helicopter used is a commercial available rc helicopter. Necessary adjustments have been made to let the helicopter fly autonomously. The model derived [7] is based upon first principal modeling, using elementary laws of physics and aerodynamics. The model accounts for the dynamics a helicopter exhibits in hover and hover like flight. A non-linear model results, which has been implemented in Simulink, see Appendix A.

2.2 Structure of helicopter model

Figure 2.1 shows the basic structure of the helicopter model. The separate blocks are briefly discussed in the following paragraphs. The signals are described in more detail in the Simulink model (Appendix A) used for the simulations. All the 17 states are given in Appendix B.

Actuator Dynamics This block describes the dynamics of the servo and the linkages from the servo to the swash plate and the tailrotor pitch.

Rotary Wing Dynamics In this block the direction and magnitudes of the thrust are determined. The magnitudes are a function of the magnitude of the thrust inputs, but are are also dependent of the translatory movement of the helicopter and attitude of the helicopter body. The direction of this thrust is a function of the tilting of the swashplate, but is also influenced by factors such as translatory movement and rotation of the helicopter body.

Force and Moment Generating Process The magnitude and direction of the thrust forces created is split here into sub components of forces and moments affecting the helicopter. The output is a three dimensional force and moment vector.

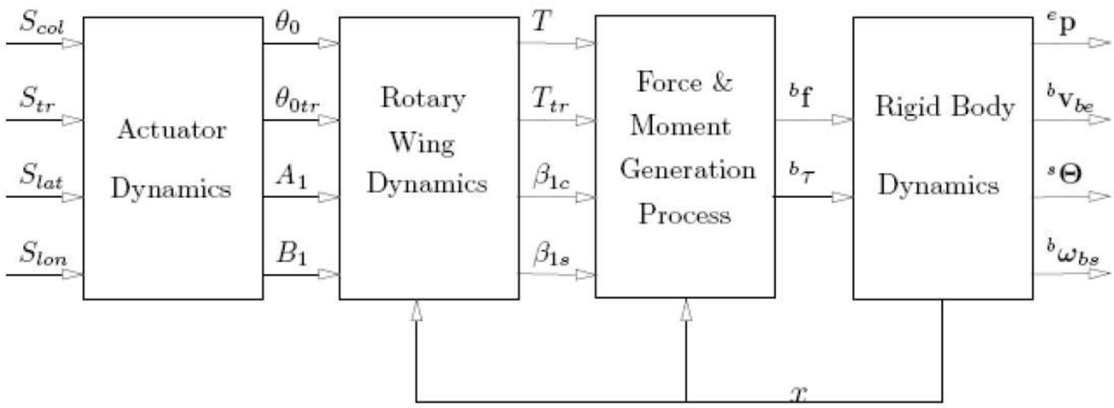

Figure 2.1: Structure of helicopter model

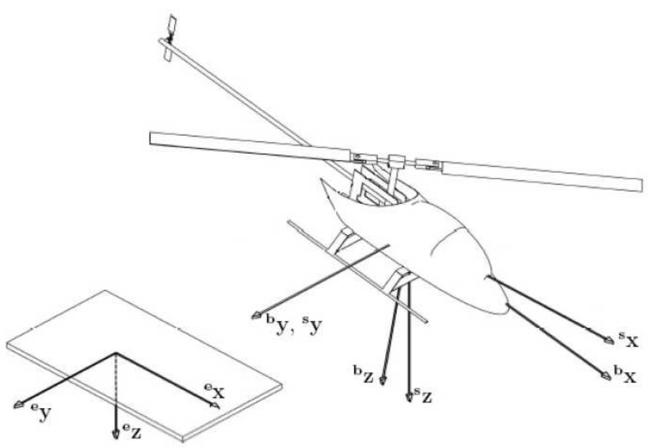

Figure 2.2: Definition of earth frame, body frame and spatial frame

Rigid Body Dynamics Here the final equations that describes the translational and rotational movement of the helicopter are derived. For this purpose the helicopter is regarded as a rigid body.

There exists a dynamic coupling between the velocity and rotation of the helicopter and the forces and moments developed, and exerted on the helicopter. This coupling is indicated in the block diagram by the feedback of the state vector $\dot{\mathbf{x}}$.

To describe the flight dynamics different basic frames are used (Figure 2.2). The body fixed reference frame (BF) is a frame with constant inertia of the helicopter, the center coincides with the center of mass from the helicopter. In order to describe the position and translateral movement of the helicopter a frame that has no acceleration is needed. This will be referred to as an earth frame (EF), which is defined with the origin at the airfield where the helicopter operates. A spatial frame (SF) is used, where resulting moment and forces described in BF are translated into rotations and translatory movement of the helicopter. The spatial frame has its origin in the helicopter center of mass like the BF and the same orientation as the EF.

There are four output vectors of the rigid body dynamics part:

- Position - Relative to the earth frame
- Velocity - Relative to the earth frame
- Euler angles -Relative to the spatial frame
- Euler rates - Relative to the spatial frame

Chapter 3

Control design

3.1 MPC control

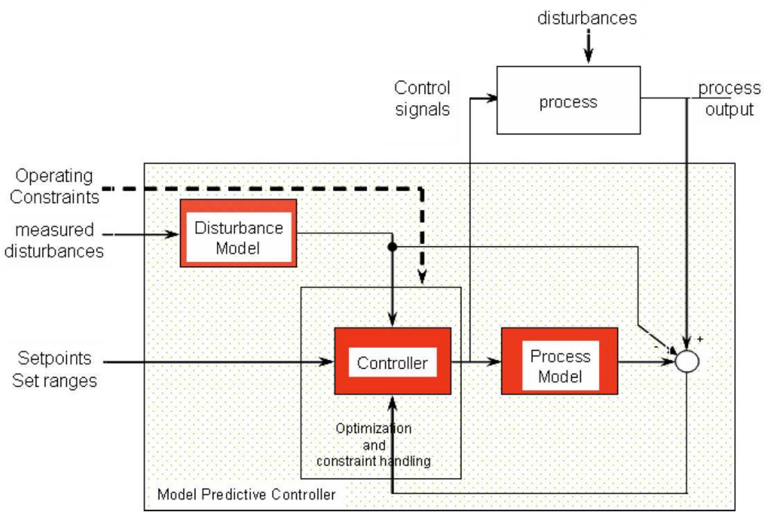

Figure 3.1: MPC control scheme

The control type used is Model Predictive control. The benefits compared to general PID controllers are it's ability to handle constraints on the control and output signals, there is no manual tuning of the controller needed and it's an effective means to deal with large multivariable systems. Which means no separate controller for every input-output combination has to be designed.

A general scheme of a Model Predictive Controller applied on a process can be seen in Figure 3.1 [1]. The MPC controller requires a linear process model, the use of a disturbance model for measured disturbances is optional and will not be used here. Operating constraints on the control signal and the process output are also defined. The model predictive control system uses model based predictions of process outputs to manipulate the process inputs in such a way that deviations from specified values (setpoints) are minimized, subject to constraints on inputs and outputs. For unmeasured disturbances the controller provides feedback compensation (relatively slowly).

The tool used here to design a MPC controller is the MPC toolbox in Matlab [2]. An MPC Toolbox design generates a discrete-time controller, one that takes action at regularly-spaced discrete time instants. The sampling instants are the times at which the controller acts. The interval separating successive sampling instants is the sampling period Δt (also called the control interval).

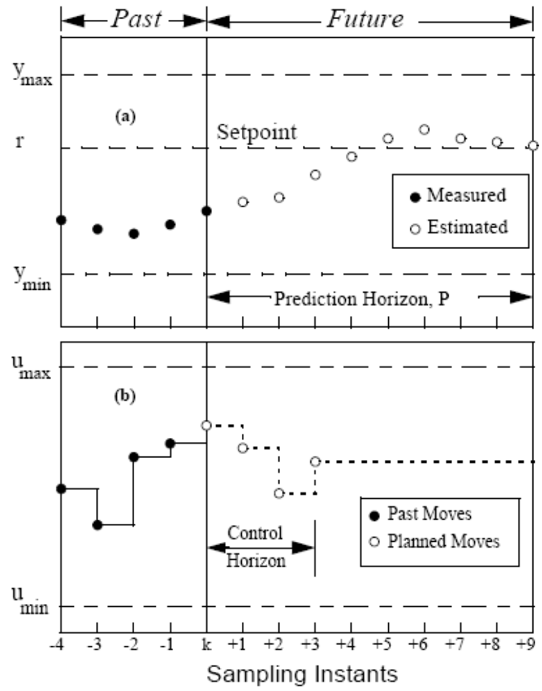

Figure 3.2: MPC horizon

The latest measured output, y_k, and previous measurements, y_{k-1}, y_{k-2}, ..., are known and are the filled circles in Figure 3.2(a). Figure 3.2(b) shows the controllerŠs previous moves, u_{k-4}, ..., u_{k-1}, as filled circles. To calculate its next move, u_k the controller operates in two phases:

1 Estimation. In order to make an intelligent move, the controller needs to know the current state. This includes the true value of the process output (y_k) and any internal variables that influence the future trend, y_{k+1},...,y_{k+P}). In

which P is the prediction horizon. To accomplish this, the controller uses all past and current measurements and the models of plant and measured disturbances (not used here).

2 Optimization. Values of setpoints, measured disturbances, and constraints are specified over a finite horizon of future sampling instants, $k+1$, $k+2$, ..., $k+P$. The controller computes M moves u_k, u_{k+1}, ..., u_{k+M-1}, where M ($\geq 1, \leq$ P), see Figure 3.2(b).

3.2 Linearization

Since the design of the controller type used requires a linear model of a system, the available non-linear model has been linearized in certain operating points. In these points a controller is designed and is implemented on both the linear and non-linear model in the specific operating points. It is expected that the behavior of the linear and non-linear model in the same operating point is the same. However, the further the system operates from the specific operating point, the bigger the differences between linear and non-linear model expected.

Simulink Control Design [5] has been used to design a linear model in a given operating point. It is tried to calculate steady-state for the specified operating point, which means that the point remains steady and constant with time; all states in the model are at equilibrium.

The default linearization method is used: block-by-block analytic linearization. This linearizes the blocks individually, using pre-programmed block linearizations and then combines the results to produce the linearization of the whole system. This method is more accurate compared to the numerical-perturbation linearization, which linearizes the whole system by numerically perturbing the system's inputs and states about the operating point.

3.3 Controllability

Given the process defined by the state space model:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{3.1}$$

,with $n \times 1$ state x, $m \times 1$ input u, $n \times n$ matrix A, $m \times m$ matrix B and $m \times n$ matrix C.

This system is state controllable at $t = t_0$ if it is possible to construct an unconstrained control signal that will transfer an initial state to any final state in a finite time interval $t_0 \leq t \leq t_1$. If every state is controllable, then the system is said to be completely state controllable. This requires that the $n \times n$ matrix has full rank n:

$$\begin{pmatrix} B & AB & ... & A^{n-1}B \end{pmatrix}$$

3.4 Observability

The previous is completely state observable if the entire state can be determined over any finite time interval $t_0 \leq t \leq t_1$ from complete knowledge of the system input and output over $[t_0, t_1]$, with $t_1 > t_0 > 0$. This requires that the $n \times n$ matrix has full rank n:

$$\begin{pmatrix} C & CA & ... & CA^{n-1} \end{pmatrix}$$

3.5 Stability

The MPC control design algorithm used here does not guarantee stability of the system [4], therefore LQR control is used to first stabilize the system and then design a MPC controller.

3.6 LQR control

Controllability and observability are central in the design of an LQR controller since, among other properties, they guarantee the existence of a stabilizing controller [6].

Assume the entire state x of (3.1) is measurable and a controller (regulator) $u = -Kx$ is used. Regulator design entails finding the $m \times n$ gain vector K. LQR methodology is now used to specify the gain K. Defined is:

$$J = \int_0^\infty (x^T Qx + u^T Ru)d\tau \tag{3.2}$$

The goal is to find the gain vector K to minimize this "cost function", subject to the system dynamics.

Here, assume the $n \times n$ Q matrix is diagonal with diagonal elements q_i (each providing a weight for a different element of the deviation of the state) and $m \times m$ R matrix which is also diagonal with diagonal elements r_i (providing a weight for a different element of the deviation of the control signal).

The values for Q and R are used as design parameters. How to tune them? If all the $q_i = 0$ this represents that it does not matter what type of excursions the state has while the control tries to minimize (3.2). High values of q_i relative to r_i mean that it is allowed to use lots of control energy to keep state excursions small while minimizing (3.2). Clearly $r_i = 0$ is not possible as this results in allowing infinite control energy to minimize the cost function.

Finding the K to minimize J involves solving a "Ricatti" equation and if a solution is found it results in a controlled system which is stable.

Here the Matlab "lqr" command is used to directly solve for the gain vector K given A, B, Q and R.

Chapter 4

Simulation results

4.1 Simulations

The first simulations of the helicopter model are open loop, without a controller and starting in the origin in which all the states are chosen 0, to study the dynamical behavior of the helicopter.

Thereafter simulations in the origin and just above ground (hover mode: states are all 0 except the z-coordinate, which is -1) are run, with solely a stabilizing LQR controller or MPC controller and also a combination of both controllers. The controllers both have the goal to keep the helicopter steady in the given operating point.

Since the MPC controller does not guarantee a stable system, it is not expected that only this controller will keep the helicopter in the required operating point, since the helicopter model is originally not a stable system. To design a stable system first the LQR controller will have to be used and on this new stable system the MPC controller is designed.

Since the design of the LQR and MPC controller requires a linear model of the plant in the specific operating point, it is expected that if the controllers are placed on the linear and non-linear plant, the behavior of the models with controller will be the same in the specific operating point. However, any too big deviation from the operating point in which is linearized, will cause the non-linear model not to show the same behavior as expected in the linear case.

4.1.1 Behavior in the origin

The results from the non-linear helicopter dynamics without controller, starting in the origin for t=0 s, can be seen in Figure 4.1. The simulation time is chosen such that the behavior of the helicopter around the operating point is clear in time. The figure shows that after about 30 s the helicopter starts oscillating significantly.

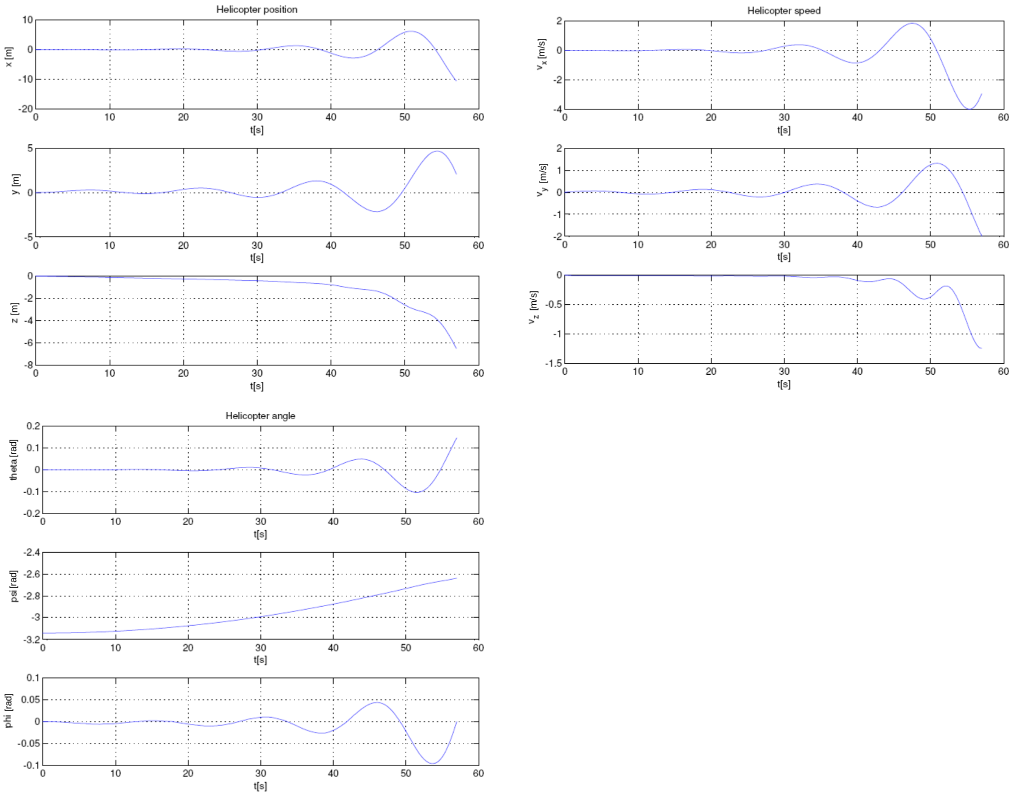

Figure 4.1: Non-linear model, position, velocity and Euler angles in the origin

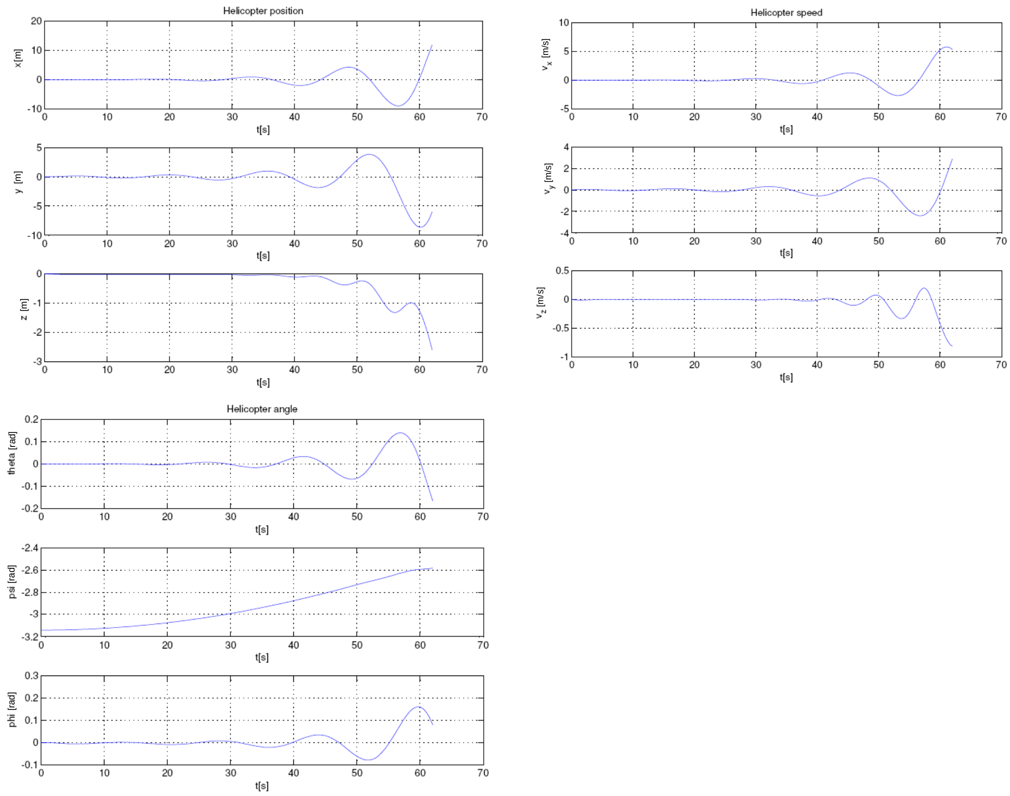

Figure 4.2: Non-linear model with MPC controller in the origin

4.1.2 MPC control in the origin

To try and keep the helicopter in the origin, a MPC controller is designed in this operating point. The goal is to see how long the helicopter will stay in the origin (since MPC does not guarantee a stable system).

To design an MPC controller the MPC toolbox in Matlab [2] has been used. The controller design requires a linear model of the plant which is to be controlled. Since the used helicopter model (Appendix A) is non-linear it is linearized in the given operating point (here the origin). Therefore Simulink Control Design [5] is used. It is tried to design a steady state linear model of the helicopter model, linearized over the three control inputs and 9 outputs (velocity, euler angles and position), with all the states set to 0. The method used is a block by block analytical linearization. A steady state value of a state is calculated by taking the time derivative of the state equal to 0. Simulink Control Design however cannot find a solution for a steady state linear model for these conditions. The reason for the steady state requirements is that a controller has to perform less control action in a steady state operating point, which makes it easier to design a (stabilizing) controller.

A MPC controller is designed in the operating point with the *cmpc* command of the MPC toolbox. No mismatch of plant and model is chosen. For the plant model a step response coefficient matrix of the plant is required. To transfer the linear state space model of the plant to this step format the *ss2step* command is used.

After closed loop simulation in the Matlab environment with the *cmpc* command the Simulink S-function for Simulink block *nlcmpc* is used to implement the MPC controller in the Simulink environment on the helicopter model. The last is done to make simulations of the MPC controller (which is thus designed using a linear model in the origin) on the non-linear helicopter model in the origin.

The designed MPC controller has the following specifications: The model simulation time in Simulink is 100 s. The plant step model is sampled with 2 Hz, this value is chosen such that the real simulation time is within a few minutes. The control horizon is set to 3 and the prediction horizon to 10. This is done by trial and error.

The reference values for the outputs of the model are all chosen equal to 0. The output weights are chosen all equal to 0; except the x, y and z position. These are chosen 1. This means that the x, y and z positions are very important to track compared to the other outputs of which the controller ignores deviations from the references completely. The output variable constraints, which are the lower and upper limits on the output variables, in the order $[V_x, V_y, Vz, x, y, z, \theta, \psi, \phi]$ are [-10 -10 -10 -10 -10 -10 -10 -10 -10 10 10 10 10 10 10 10 10 10].

The weights on the controller output signals are chosen 1, so there is no scaling. The lower limits, upper limits and rate of change (per timestep) on the controller outputs are chosen [-10 -10 -10 10 10 10 1 1 1]. The first control signal controls the tilt of the swash plate in lateral A_1 direction, the second signal controls the longitudinal direction B_1 and the third the position of the swash plate on the shaft, which is proportional to the collective pitch θ_0.

The results of the simulations from a MPC controller on the non-linear model in the origin are given in Figure 4.2.

From the results can be seen that the simulation stops after about 62 s, since then the x-position of the helicopter reaches its upper limit and the *cmpc* algorithm cannot find a solution anymore, this error is also clearly given by Simulink.

The values used for the MPC controller design in this section are mainly found by trial and error. Changing the MPC control variables, such as control and prediction horizon changes little in the extreme way the helicopter starts to move in x, y and z position. But if for example the weights on $V_x = V_y = 10$ and the weight on $V_z = 1$, the maximum values of the x and y positions decreases with approximately 50% the first 20 seconds. The maximum value of the z position hardly changes. Increasing the difference in weights more does not cause significant change.

When running simulations on the linearized helicopter model, it appears that the helicopter stays in the origin and does not move at all. This is considered to be strange, since the linear model in the origin should also become unstable like in the non-linear case, without needing a disturbance to make it unstable. This since the behavior of the linear and non-linear model should be identical in the same operating point.

4.1.3 MPC control in hover mode

The next simulation is a hovermode, in which an MPC controller on the linear and non-linear model is placed. The parameters and Simulink model are the same as for the MPC controller in the origin, except the reference value for the z-position, which is now -1. The operating and linearization point is chosen such that the values for all states are 0, except z = -1 (which thus defines a hover just above ground). It appears that again a steady state linear model cannot be calculated, also it is tried to change the roll angle to a few degrees because this causes a force countering the tailrotor force and is more likely to give a more stable hovermode [3], but this does also not result in a steady state linear model.

The results of the tracking of the reference values and the control signals from the linear model are given in Figure 4.3.

The results from the same MPC controller but on the non-linear model are given in Figure 4.4.

From Figures 4.3 and 4.4, which show the position of the MPC controller on linear and non-linear helicopter model in time, it can be seen that the difference between the linear and non-linear model is very significant.

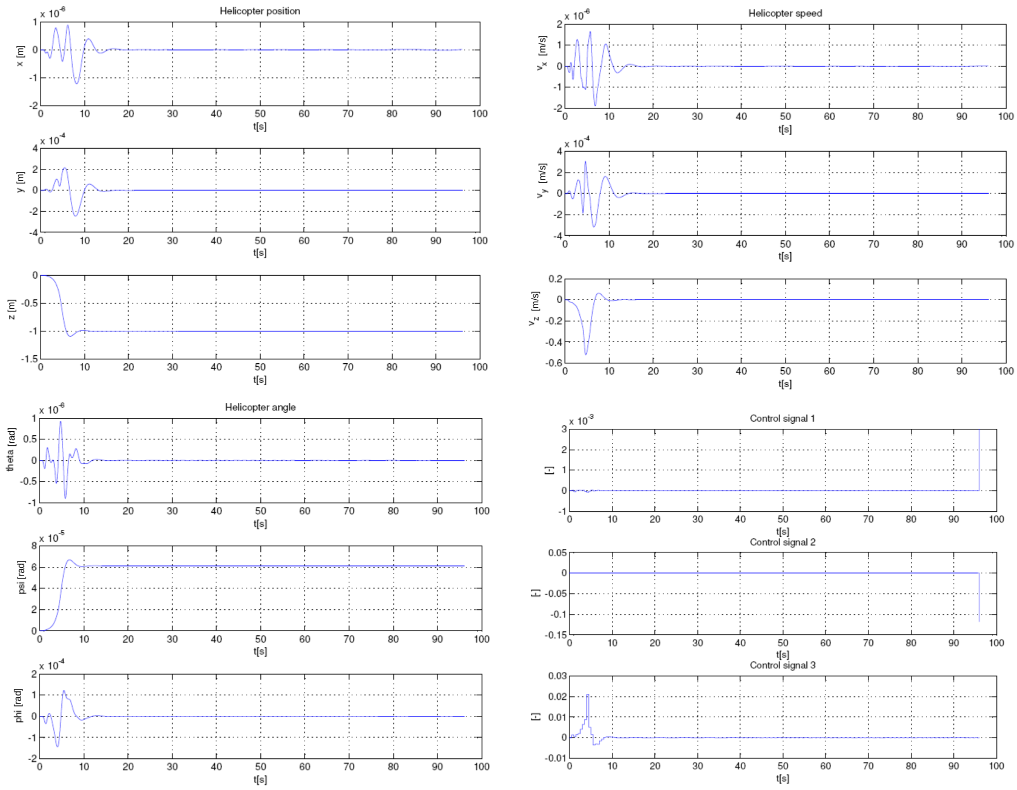

Figure 4.3: Linear model with MPC controller in hover (z = -1)

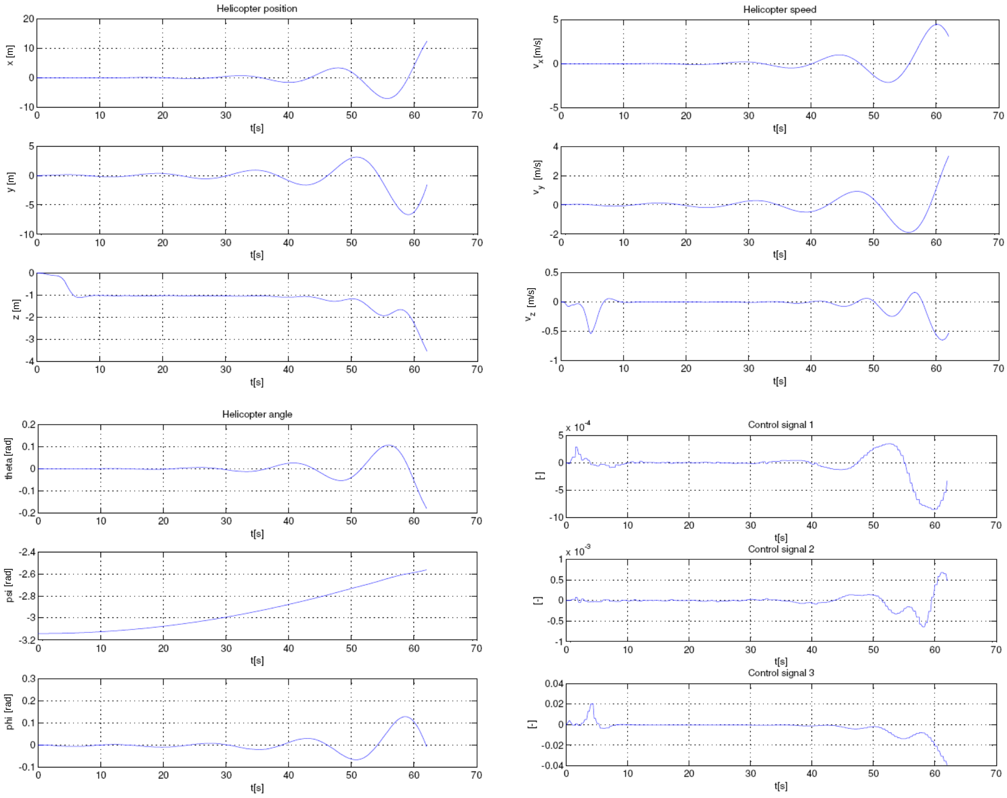

Figure 4.4: Non-linear model with MPC controller in hover (z = -1)

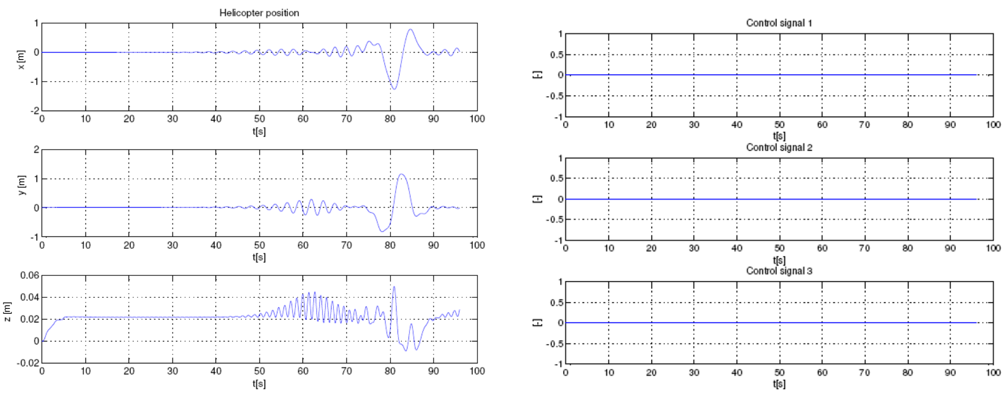

Figure 4.5: Non-linear model with LQR controller in the origin

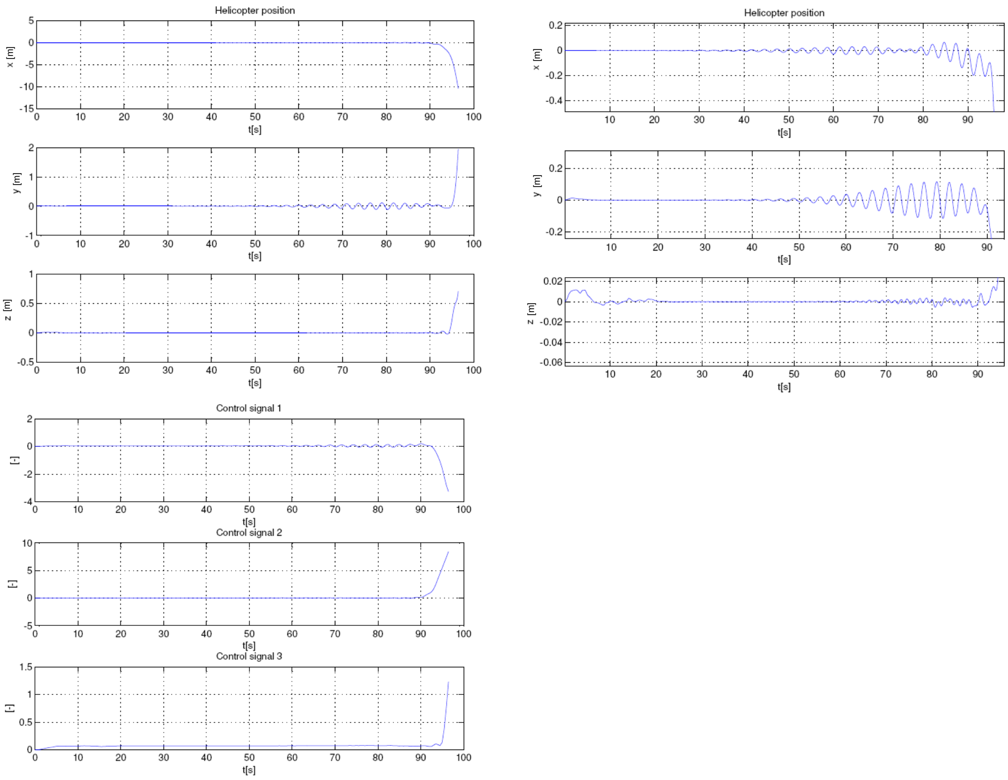

Figure 4.6: Non-linear model with LQR and MPC control in the origin

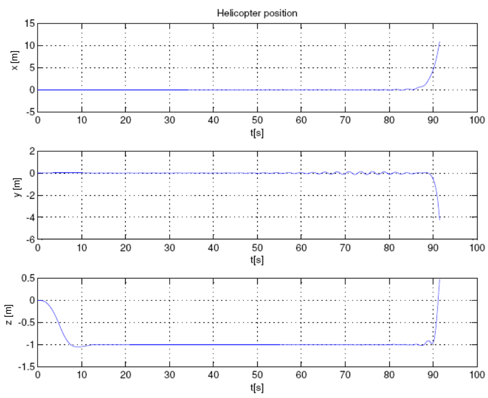

Figure 4.7: Position of helicopter: non-linear model with LQR and MPC control, in hover (z = -1)

4.1.4 LQR and MPC control in the origin

The conclusion drawn from the previous section, in which the MPC controller was used in hovermode, is that the non-linear model cannot be stabilized with solely a MPC controller. Since the MPC control design here does not guarantee stability on an unstable plant, it is decided to first design a stabilizing LQR controller on the plant and then place the MPC controller on this stabilized model.

For the design of the LQR controller it is assumed that all the states are available for the controller calculation and input.

First the results are viewed with a LQR controller on the non-linear model starting from the origin, the position and control signals can be viewed in Figure 4.5. The simulation time is 100 s. Compared to position of the helicopter with the MPC controller in the origin, see Figure 4.2, there is significant improvement.

After this the MPC controller is placed over this new plant with LQR control. The results of the position and control signals are given in Figure 4.6, the position is also zoomed in the same figure. This makes it easier to compare the results with the system in the origin with solely a LQR controller (Figure 4.5). The MPC controller shows a little improvement. The simulation time is 100 s, however the simulation stops some seconds before, this is because the x position reaches its maximum allowed value of 10 set in the MPC control design constraints, in Simulink this is given as an error message that there is no solution for the *cmpc* MPC controller design at that time.

4.1.5 LQR and MPC control in hover

The results of the position of the non-linear helicopter model in hover (all states equal to 0, except z = -1) with LQR and MPC controller is given in Figure 4.7.

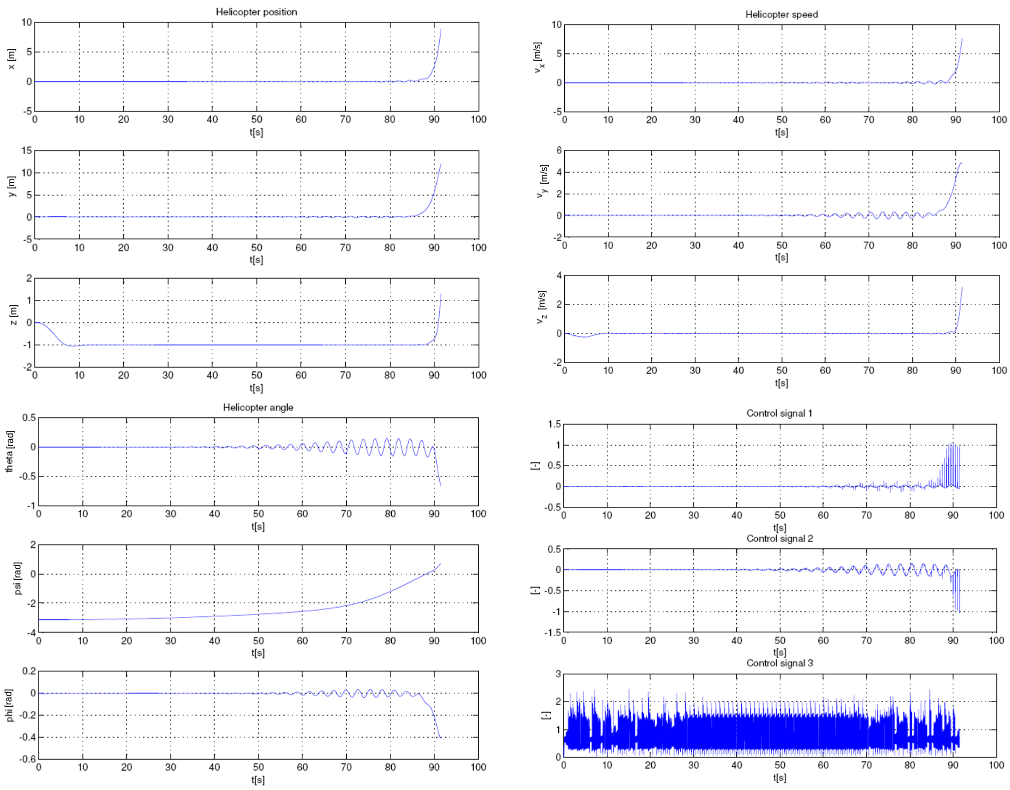

Figure 4.8: Non-linear model with LQR and MPC control, in hover (z = -1), feedback of all 17 states

The used Simulink model is the same as in the previous section. Compared to the results without LQR controller and only MPC controller (Figure 4.4) it is clear that the non-linear model stays in a stable hover for a longer time (over 40 s longer).

4.1.6 Linearization and MPC and LQR control of all 17 states, in hover mode

It is tried to see if a new linear model, which is now linearized over 3 inputs and all the 17 states as outputs and MPC control of all 17 states, causes an improvement. The states are given in Appendix B. The results for a LQR and MPC controller in hover on this new system are presented in Figure 4.8.

It can be concluded that the position of the helicopter in time does not differ much from the same system with only 9 states feedback (Figure 4.7). Except control signal number 3, which is scattering.

4.2 Stability and the tailrotor

The system appears not stable as expected after LQR control implementation, since the oscillations in the horizontal plane increase in amplitude as a function of time. An explanation for this is that the LQR algorithm in Matlab may not have calculated a stabilizing controller. The controllability of the linear helicopter model has been checked and 2 states related to the tailrotor appear not controllable: the yaw ψ and yaw-rate r. This causes the system to have two poles in the origin, which gives an unstable system. The torque generated by the tailrotor in the Simulink model, which counters the main rotor torque, causes the helicopter to be stable in static but not in dynamical situation. To make the 2 states controllable the tailrotor pitch has also been made controllable. After using the LQR algorithm in Matlab again it appears that all the poles of the system are in the left half plane, guaranteeing a stable system. However, implementing the new system in Simulink keeps generating algebraic loop errors after approximately simulating 0.01 s. This problem could not be solved by using time delay blocks with delay time of 0 s, as in the original Simulink model used successfully to solve algebraic loop problems.

Chapter 5

Discussion

Brief conclusions of the simulations are given in Chapter 4. These conclusions are further discussed here and recommendations are made.

When the helicopter model starts from the origin without using any control it starts oscillating in the horizontal plane in lateral and longitudinal direction. It oscillates with periods of about 8 s, with an increasing amplitude as a function of time. After simulating for 50 s the amplitude is $>$ 5. The height of the helicopter increases non-linear to values $<$ -5 in the same time.

Implementing solely the MPC controller on the system in the origin to try and keep it steady there, causes the helicopter model to still oscillate with a non-reduced amplitude in time, however the height of the helicopter reaches -0.2 in about 50 s. Since the amplitudes of oscillations of the movements in the horizontal plane still increase in time, the MPC controller does not stabilize the system.

Implementing solely the LQR controller on the system in the origin causes a drastic improvement in the oscillating behavior of the system compared to solely the MPC controller. The oscillations in the horizontal plane reach values $>$ 1 after only about 83 s. And the height of the helicopter $>$ -0.05 during the entire simulation time of 93 s. After this the MPC controller is designed on this new system with LQR control. Implementing this controller as well results in oscillations in the horizontal plane with amplitudes $<$ 0.2 in 93 s. The height of the helicopter then is $<$ 0.02.

The system with LQR and MPC combined, in which the setpoints in the MPC controller are used to bring the helicopter from the origin in a hover mode (all states 0, except z = -1), shows very similar behavior to the same system in which the stays in the origin. The best results achieved are: 10 s for the helicopter to reach hover mode. Up to 85 s: the oscillations in the horizontal plane are $<$ 0.2 and z $>$ -1.01

To improve the last system it is tried to feedback all 17 states to the MPC controller in stead of the previous 9. This shows no significant improvement.

However, the system with LQR control is not stable as expected, since the oscillations in the horizontal plane increase in amplitude in time. Two states related to the tailrotor appear not controllable causing two poles in the origin, giving an unstable system. To make the 2 states controllable the tailrotor pitch has also been made controllable. This gives a stable system, with all poles in the left half plane. However, implementing the new system in Simulink keeps generating algebraic loop errors after approximately simulating 0.01 s. These problems could not be solved.

It can be concluded that combining LQR and MPC control gives good control of the helicopter in hover mode within certain ranges in certain time. However, the system seems not able to be stabilized with the LQR controllers used to simulate. In the project time it has not been possible to look at more complex maneuvers besides the flight from the helicopter to hover mode in a predefined point, as stated in the introduction.

A big drawback of the system is that a sample frequency of 2 Hz is used for the MPC step model, since the dynamics of the helicopter has a bandwidth of at least 50 Hz. A reason for the unstable behavior may also partly be found here. The sample frequency could not be increased much more, since the Simulink simulation time would become too large ($>$ 30 min per simulation) to test the controllers.

Appendix A

Simulink model description

The simulink model given in Figure A.1 has the same basic structure as Figure 2.1. The actuator dynamics part describes the dynamics of the servo and the linkage from the servo to the swash plate. The 3 inputs are the generated signals from the controller(s) used. The outputs are the tilt of the swash plate in lateral A_1 and longitudinal direction B_1. And also the position of the swash plate on the shaft, which is proportional to the collective pitch θ_0. The dynamics of the tailrotor are not modeled in the Simulink model, since data for this is not available. The force generated by the tailrotor T_{tr} is therefore calculated through a linear feedback of the moment around the z-axis N. Therefore direct control of the tailrotor is not possible in this model.

Since model helicopter dynamics are faster than a full sized helicopter a control rotor part in the implemented model is used. The effect of this is that the overall sensitivity and bandwidth of the output response are reduced compared to the dynamics of the full size helicopter model which has been first derived.

The main rotor flapping part gives the direction of the thrust β_{1s} and β_{1c}, which is a function of the tilting of the swash plate described by A_1 and B_1. Also the translatory movement and attitude of the helicopter determine the direction of the thrust.

The magnitude of the thrust force T_{mr} created is calculated in the main rotor thrust block. The forces and moments acting on the helicopter are calculated in the force generation and moment generation block.

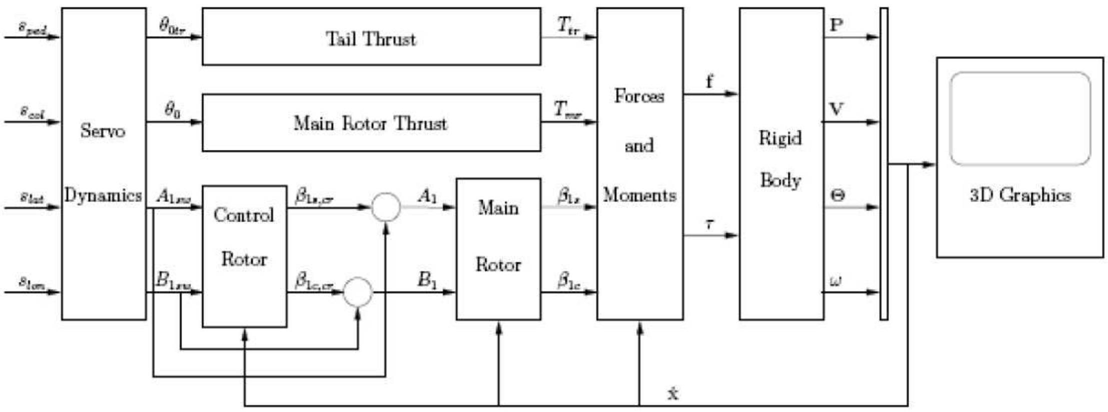

Figure A.1: Structure of Simulink model of helicopter dynamics

Appendix B

Full state feedback

To control the helicopter a full state feedback has been used. The state vector x is given by:

$$\begin{pmatrix} b_{1s} \\ b_{1c} \\ \theta_{S1s} \\ \theta_{S1c} \\ \theta_{S_0} \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \\ u_b \\ v_b \\ w_b \\ x_e \\ y_e \\ z_e \end{pmatrix} \tag{B.1}$$

, with θ_{S1s}, θ_{S1c} and θ_{S_0} the servo angles corresponding with the servo input signals $S1s$, $S1c$ and S_0. Which control the tilt of the swash plate in lateral A_1 and longitudinal direction B_1 and the collective pitch θ_0. b_{1s} and b_{1c} are the corresponding control rotor flapping outputs. ϕ, θ and ψ are the pitch, roll and yaw. p, q and r give the corresponding rotational rates. u_b, v_b and w_b the helicopter velocity components in body coordinates . x_e, y_e and z_e give the helicopter position in earth frame coordinates.

Bibliography

[1] Ton Backx and Ad van den Boom. Modelling and predictive control introduction. Lecture notes, 2005.

[2] Alberto Bemporad, Manfred Morari, and N. Lawrence Ricker. Model predictive control toolbox for use with matlab, March 2005.

[3] Ulrik B. Hald, Mikkel V. Hesselbæk, Jacob T. Holmgaard, Christian S. Jensen, Stefan L. Jakobsen, and Martin Siegumfeldt. Autonomous Helicopter Modelling and Control, chapter 10.1. Aalborg University, Department of Control Engineering, 2005.

[4] J.M. Maciejowski. Predictive Control with Constraints, chapter 1.6 Unstable plant, pages 22–25. Pearson Education, 2002.

[5] The MathWorks, Inc. Simulink Control Design, 2005.

[6] Katsuhiko Ogata. Modern control engineering, chapter 13. -, -.

[7] Roger Pettersen, Emir Mustafic, and Mads Fogh. Nonlinear control approach to helicopter autonomy. Master's thesis, Department of Control Engineering Institute of Electronic Systems Aalborg University, 2005.