OPTIMAL TRACKING CONTROL FOR UNMANNED AERIAL VEHICLES

BY
PÉTER BAUER

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

SUPERVISOR
Prof. József Bokor

SYSTEMS AND CONTROL LABORATORY
Computer and Automation Research Institute

and
DEPARTMENT OF CONTROL FOR TRANSPORTATION- AND VEHICLE SYSTEMS
Budapest University of Technology and Economics

BUDAPEST, HUNGARY

May 2013.
NYILATKOZAT

Alulírott Bauer Péter kijelentem, hogy ezt a doktori értekezést magam készítettem és abban csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Budapest, 2013. május 2.

________________________________________
Bauer Péter

Az értekezésről készült bírálatok és a jegyzőkönyv a későbbiekben a Budapestei Műszaki és Gazdaságtudományi Egyetem Közlekedésmérnöki és Járműmérnöki Karának Dékáni Hivatalában elérhetőek.
I would like to express my deepest gratitude to my advisor, Professor József Bokor for his patronage, technical guidance, many suggestions and constant encouragement during this research.

I also wish to thank Balázs Kulcsár for his help at the beginning, and Bálint Vanek for his help in the last years. I would like to thank the discussions, useful ideas, revision of my articles and encouragement.

I had the pleasure of meeting Professor Gary Balas and the staff of his control group during my visits at Department of Aerospace Engineering and Mechanics at University of Minnesota or during their visits in Hungary. I’m especially thankful to Pete Seiler, Paw Yew Chai, Rohit Pandita, Andrei Dorobantu, Will Johnson and Arion Mangio for working together.

Regarding the Department of Control for Transportation- and Vehicle Systems I’am thankful to my colleagues Károly Kurutz, Géza Tarnai, István Hrivnák, Géza Szabó, Zoltán Komócsin, Károly Gyenes, Balázs Sághi, Edit Baranyi and István Varga for their support and kindest help during my work at the department. I’am also thankful to the others, Zsuzsa Preitl, Imre Károlyi, Tamás Bécsi, Szilárd Aradi, Tamás Luspay, Tamás Tettamanti, Balázs Németh, Alfréd Csikós and András Mihály for working and teaching together. I would like to express my special thanks to Erzsébet Paczona for her continuous administrative support and for the encouragement during the hard periods.

Regarding the Systems and Control Laboratory at Computer and Automation Research Institute I’m very thankful to Alexandros Soumelidis for working together on aerospace applications and making many discussions about electronics and microcontroller programming. I’m also thankful to the other colleagues Péter Gáspár, András Edelmayer and Tamás Bartha. I’am thankful to working together with Gergely Regula, Ádám Bakos, Gyula Pintér, István Gőzse, Márk Lukátsi, András Erdő and István Réti. I wish to express my thanks to my first roommates Tamás Péni, Gábor Rödönyi and András Bódis-Szomorú for their encouragement in research. I also would like to thank the friendly atmosphere and encouragement to my current roommates Zoltán Fazekas, Zsolt Biro, Krisztián Szabó and Ernő Simonyi. I wish to express my special thanks to Katalin Hargitai, Marika Farkas and Éva Thiry for their continuous administrative support and for the encouragement during the hard periods.

At last but not at least, thanks to my parents for their constant support during the first part of my work and for my wife Reka and her family for the same support during the last part of my work.

*Soli Deo Gloria*
Notations and symbols

Notations

\( a \): exponent of exponentially bounded reference signal / acceleration value

\( \mathbf{a} \): acceleration vector

\( A \): maximum bound of exponentially bounded reference signal

\( A, B, C, C_r, G, V, W \): discrete time system matrices / transformed discrete time system matrices

\( A^a, B^a \): matrices of system augmented by the observer

\( A^c, B^c, G^c, C^c, C^c_r \): continuous time augmented system matrices

\( A^c_c, B^c_c, C^c_c, C^c_r, G^c_c, V^c, W^c \): continuous time system matrices

\( A_d, B_d, C_d, D_d \): state space matrices of actuator delay dynamics

\( A_F, B_F, C_F, D_F \): state space matrices of washout filter

\( A_i, B_i, C_i, D_i \): uncertain discrete time system matrices

\( A_k, C_k \): time-varying state space matrices

\( A^d_y \): disturbance state dynamics matrix

\( A_1, A_2, A, L_1 \): Heun scheme gain matrices

\( AW \): anti windup gain

\( b \): angular rate bias vector / wingspan

\( B_d \): disturbance input matrix

\( \mathbf{B} \): augmented input matrix

\( c_L, c_N \): roll / yaw moment coefficients

\( \mathbf{C} \): projection matrix constructed from \( C_r \)

\( \mathbf{\tilde{C}}_r \): transformed \( C_r \) matrix
\( d \): deterministic disturbance vector / dimension of disturbance vector / exponent of disturbance exponential bound

\( D \): maximum bound of exponentially bounded disturbance

\( \tilde{d} \): augmented disturbance vector

\( D_E, d_E \): exponentially bounded state estimation error maximum bound and exponent

\( d_L, d_N \): roll and yaw torque disturbances

\( d_i \): exponent of exponentially bounded augmented disturbance

\( e, e_\phi \): output tracking error / roll angle tracking error

\( e_G \): unit gravity vector

\( e_I \): integral of roll angle tracking error

\( E_\phi, E_r, E_p, E_{\dot{\alpha}}, E_{\dot{\delta}} \): error and signal norm values

\( F \): steady state input to output transition matrix / augmented matrix for feedforward control design

\( G_{ac} \): aircraft dynamical model

\( G_{act} \): actuator dynamics (transfer function)

\( G_d \): disturbance transfer matrix

\( G_{delay} \): actuator delay dynamical model

\( G_{d_0}^{d} \): disturbance to estimated disturbance transfer matrix

\( G_{e_0}^{d} \): error to estimated disturbance transfer matrix

\( G_{r_0}^{d} \): reference to estimated disturbance transfer matrix

\( G_e \): estimation error transfer matrix

\( G_{est} \): estimator EKF dynamics

\( G_{efilt} \): smoothing filter for estimated disturbances

\( G_{filt} \): transfer function matrix including washout filter

\( G_r \): reference signal transfer matrix

\( G_{d_2}^{d} \): disturbance to state transfer matrix

\( G_{e_2}^{d} \): error to state transfer matrix

\( G_{r_2}^{d} \): reference to state transfer matrix
$H$: transformation matrix from reference to tracking state

$H_p, H_u$: prediction and control horizons in MPC control

$I_p, \omega_p$: engine and propeller inertia and angular rate

$J$: cost functional / Jordan matrix

$k$: time index

$K$: gain of disturbance estimator

$K_d$: gain from disturbance

$K_D$: maximum bound of exponentially bounded augmented disturbance

$K_{d\infty}$: gain from steady state disturbance

$K_I$: tracking error integral gain

$K_r$: reference feedforward gain

$K_{r1}, K_{r2}$: modified reference gains in MPC control

$K_{r\infty}$: gain from steady state reference

$K_{S1}, K_{S2}, K_{S1}$: gain matrices related to $S1/S2$ matrices

$K_x$: state feedback gain

$K_x, K_Q, K_S$: optimal input gains (minimax control)

$K_{x1}$: pre-stabilizing state feedback gain

$K_{x2}$: LQ optimal state feedback gain

$K_u$: maximum bound of exponentially bounded reference input / MPC gain from previous input

$L$: matrix of closed-loop estimation error dynamics

$L_p, L_r, L_{\delta_a}, L_{\delta_e}, L_{d_L}, L_{d_N}$: coefficients of aircraft linearized roll dynamics

$L_o$: observer gain matrix

$L_x, L_Q, L_S$: worst case disturbance gains (minimax control)

$m$: dimension of input vector

$M$: constructor matrix for the linearly dependent columns of $C_r$ / gain matrix in MPC input design / gain of disturbance estimator

$M_2, M_3, M_4, M_R$: auxiliary matrices
$M_B$: auxiliary input matrix

$M^+, M^-$: matrices of EKF state dynamics

$n$: dimension of system state space

$N$: control horizon / system noise covariance matrix

$N_p, N_r, N_{\delta_a}, N_{\delta_r}, N_{dL}, N_{dN}$: coefficients of aircraft linearized yaw dynamics

$N_x, N_u$: gains of feedforward control

$p$: dimension of measured output / roll rate

$p_0$: poles of open-loop system

$p_{LQ}, p_{MM}$: varied parameter vectors

$P$: solution of ARE / DARE

$PI$: probability of instability

$P_{Sat}$: probability of saturation

$P_\phi, I_\phi, D_\phi$: gains of PID control

$q$: dynamic pressure

$Q, Q_1, Q_2$: weighting matrices of cost function

$r$: reference vector / dimension of reference vector / yaw rate

$r_2$: $(2\cdot r \times 1)$ dimensional augmented reference vector

$R$: input weighting matrix / augmented input weighting matrix

$Ra$: angular rate noise covariance matrix

$RnH, Rna, R_{GPS}$: magnetic, acceleration and GPS measurement noise covariance matrices

$\overline{RnH}, \overline{Rna}$: basic noise covariance matrices

$R_h$: sum of reference related inputs

$R_a, R_d$: input / disturbance weighting matrices in minimax design

$s_R$: actual value of forcing function

$S$: wing surface

$S, S_1, S_2$: gain matrices in costate variable and forcing function

$T, T_S$: similarity transformation matrices
$T_{\text{act}}, \zeta_{\text{act}}$: actuator time constant and damping

$T^{BE}$: earth to body transformation (rotation) matrix

$T_d$: system time delay

$T_e$: estimator EKF time lag

$T_{s1}, T_{s2}$: settling times

$u$: input vector / exponent of exponentially bounded reference inputs

$u^r$: reference related part of input vector

$v, w$: stochastic disturbance vectors / dimensions of stochastic disturbance vectors

$v$: vector of measurements

$v^\theta$: angular rate noise vector

$V$: matrix from the linearly independent columns of $C_r$ / velocity vector

$V_w$: wind strength

$V_0, \theta_0$: trim velocity and pitch angle

$W$: measurement noise covariance matrix

$x$: state vector

$x^a$: augmented system state vector

$x^d$: state vector of actuator delay state space model / state vector of disturbance estimator

$x^e$: state estimation error vector

$x^F$: state vector of washout filter state space model

$\hat{x}$: tracking state vector

$\hat{x}$: estimated state vector

$\tilde{x}$: transformed state vector

$\bar{x}$: predicted state vector

$x^I, x^{II}$: parts of partitioned state vector

$X$: bidiagonal matrix of Jordan blocks

$X, Y, Z$: axes of coordinate systems

$y$: measured output vector
$y^r$: tracking output vector

$Y$: diagonal block of $Q$ weighting matrix

$\hat{z}$: predicted output in MPC control

$Z$ matrix from forcing function system of equations

$Z$, $T$, $\Delta U$ vectors in MPC control prediction model

$Z_1$, $Z_2$: full block matrices in DARE solvability derivation

**Greek notations**

$\beta$: angle of sideslip

$\delta_a$: aileron deflection

$\delta_r$: rudder deflection

$\Delta$: difference from steady state

$\Delta t$: discrete time step

$\varepsilon$: MPC control tracking error vector

$\phi$: roll (bank) angle

$\Phi$: stabilized state space matrix

$\Phi_1$: closed-loop state matrix

$\Phi_1$: uncertain closed-loop state matrix

$\Upsilon$, $\Theta$, $\Gamma$, $\Omega$, $\mathcal{Q}$, $\mathcal{R}$: MPC prediction model matrices

$\lambda$: costate variable / eigenvalue

$\Lambda$: matrix in DARE solvability check

$\psi$: azimuth angle

$\psi_w$: wind direction

$\rho$, $\mathcal{P}$: angular rate parameter vectors

$\theta$: pitch angle

$\kappa$: quaternion vector

$\omega$: angular velocity vector
Acronyms

ARE: algebraic Riccati equation
DARE: discrete algebraic Riccati equation
EKF: extended Kalman filter
HIL: hardware-in-the-loop simulation
IAS: indicated airspeed
LQ: linear quadratic
MM: minimax
MPC: model predictive control
PID: proportional, integral, derivative control
UAV: unmanned aerial vehicle
UDE: unknown disturbance estimator
UDEB: unknown disturbance estimator through input matrix

Symbols

()\(^+\): Moore-Penrose pseudoinverse of a matrix / superscript of matrix in EKF dynamics
()\(^E\), ()\(^B\), ()\(^N\): notation in earth, body or normal coordinate systems
\(\bar{()}\): mean value
()\(^T\): transpose of a matrix
3 Infinite horizon LQ optimal minimax tracking control

3.1 Unknown input and state estimation ........................................... 42
3.2 The infinite horizon, discrete time LQ optimal minimax tracker ........... 43
  3.2.1 Design of a stabilizing state feedback controller for \((A,B)\) ........ 44
  3.2.2 Design an optimal state and disturbance estimator for \((\Phi,C,G)\) ... 44
  3.2.3 LS optimal disturbance cancellation with the control input ............ 44
  3.2.4 Determining the solution of the zero steady state tracking error problem considering constant reference and disturbance .............. 45
  3.2.5 Derivation of the LQ optimal finite horizon solution for the centered output tracking minimax problem ........................................... 45
  3.2.6 Derivation of the LQ sub-optimal infinite horizon solution for the output tracking minimax problem ........................................... 48
  3.2.7 Summation of the control input components ............................. 48
3.3 The properties of the infinite horizon LQ optimal minimax tracker ......... 50
  3.3.1 The satisfaction of the separation principle ............................. 51
  3.3.2 No need for anti-windup compensation .................................. 52
  3.3.3 Asymptotic stability and zero steady state error for constant, finite references and disturbances ........................................... 52
  3.3.4 Finite functional value and LQ optimality for infinite horizon with constant, finite references and disturbances .......................... 54
  3.3.5 BIBO and so \(l_p\) stability with \(l_p\) time-varying references ....... 54
  3.3.6 Finite functional value (on infinite horizon) for \(l_1/l_2\) references .... 55
3.4 Comparison with other methods in Matlab simulations ....................... 56
  3.4.1 Constant unknown disturbance estimation through system input \((UDEB)\) .................................................. 56
  3.4.2 Unknown disturbance estimation \((UDE)\) ................................ 57
  3.4.3 Model predictive control \((MPC)\) with static gains \((MPC)\) .......... 58
  3.4.4 \(\mathcal{H}_\infty\) optimal control \((\mathcal{H}_\infty)\) .......................... 59
  3.4.5 Minimax tracker solution \((MM tracker)\) .............................. 60
  3.4.6 Comparison of roll doublet tracking control results .................... 60
3.5 Summary ............................................................................. 63

4 Aircraft attitude estimation for aerospace application .......................... 64
  4.1 Sensor calibration and measurement selection .............................. 65
  4.2 Working modes and switching strategies .................................. 69
  4.3 System equations and the steps of the algorithm ......................... 71
    4.3.1 Filter initialization .................................................. 71
    4.3.2 Filter equations and system observability .......................... 72
  4.4 Tuning and testing of the algorithm ...................................... 76
    4.4.1 Tuning and testing using HIL data ................................ 77
    4.4.2 Tuning and testing using real flight data .......................... 78
  4.5 Detailed HIL testing including wind effects .............................. 80
    4.5.1 Applying wind estimation and correction ........................... 82
  4.6 Summary ............................................................................. 84
5 Robustness test of the LQ and minimax tracker solutions

5.1 Parameter uncertainties in the controlled aircraft lateral-directional system model .......................................................... 86
5.2 Robustness test of the LQ tracker ...................................................... 88
5.3 Robustness test of the minimax tracker ............................................... 90
5.4 Summary ....................................................................................... 93

6 Real flight test results and comparison with the PID solution

6.1 Summary ....................................................................................... 96

7 New scientific results

8 Appendix

8.1 The applied basic methods ............................................................... 101
8.2 Introduction of the E-flite Ultrastick 25e aircraft ............................... 103
8.3 Derivation of the aircraft lateral dynamical model ............................... 104
8.4 Derivation of the finite horizon LQ optimal tracking solution with Lagrange multiplier method ...................................................... 105
8.5 The nonsingularity of $I - \Phi^T M_2$ ...................................................... 106
8.6 Evaluation of the DARE weighting matrix ......................................... 107
8.7 Upper bounds for reference signal $l_1$ and $l_2$ norms ......................... 108
8.8 Upper bound for the absolute value of reference part of input ................. 108
8.9 Finiteness of the infinite horizon cost functional for $l_1 (l_2)$ references ... 109
8.10 Interconnected system structures to test tracker methods without and with disturbances ....................................................... 113
8.11 The final gain vectors with the different design methodologies in the LQ tracker comparison ...................................................... 117
8.12 Enlarged transients from the comparison of LQ tracker solution .......... 118
8.13 Derivation of the finite horizon LQ optimal minimax tracking solution with Lagrange multiplier method .............................................. 120
8.14 Finiteness of the infinite horizon cost functional for $l_1 (l_2)$ references and disturbances ....................................................... 122
8.15 The final gain vectors with the different design methodologies in the MM tracker comparison ...................................................... 123
8.16 Enlarged transients from the comparison of MM tracker solution .......... 125
8.17 Difference between initial and final angular rate biases ...................... 127
8.18 Aircraft attitude estimator related equations and detailed derivation ........ 127
8.18.1 The detailed derivation of the filter state dynamic equation ............... 128
8.19 Enlarged figures from HIL test results N/NB/W case .......................... 130
8.20 The applied wind estimation algorithm ............................................. 131
8.21 Enlarged figures from HIL test results with wind correction N/NB/W case 132
8.22 Detailed robustness test results ........................................................ 132
8.22.1 The physical meaning of parameter uncertainties .......................... 132
8.22.2 Results with LQ tracker ............................................................... 134
8.22.3 Results with minimax tracker ....................................................... 136
8.23 Enlarged figures about real flight test results ...................................... 139
Motivation

As a member of the Group of Measurement and Control Technologies (MCT) (Systems and Control Laboratory, Computer and Automation Research Institute, Hungarian Academy of Sciences) the author deals with the modelling and reference signal tracking control of unmanned systems (USs) (mainly air and ground vehicles).

In aerospace and automotive control applications usually microcontrollers are used to implement the algorithms with limited computational capacity. The author first faced this problem in a small unmanned aircraft project ([21] and [70]). That’s why the methods developed in this thesis will be tested in aerospace applications (on unmanned aerial vehicle (UAV)) but they are also applicable in any other application. The computational burden of microcomputer systems turned the author’s attention towards to control solutions with minimum computational requirements.

These requirements can be satisfied by applying constant gain controllers instead of dynamic ones and minimizing the required number of control system states. Constant gains can be achieved both for linear time invariant (LTI) and for nonlinear and/or time-varying systems, this depends on the used design methodology. However, the use of linear, time invariant models can speed up the control design and LTI dynamical models are easy to identify from measurement data. The question is if an LTI model based controller design can well stabilize and guide the originally nonlinear system or not?

Experiences with a cascade proportional, integral, derivative (PID) control onboard a small unmanned aerial vehicle (see [70]) has shown that cascade linear controllers are able to stabilize and guide the nonlinear system in normal working modes (straight and level flight and waypoint navigation in the aerospace application). So, LTI controllers (and system models) arranged in a hierarchical structure can possibly be applied for several systems having only moderate nonlinearity.

After deciding about the considered system class, decision should be done about the considered class of reference signals and disturbances. [12] gives a good overview about this question.

"The specification of realistic inputs depends on the application. Many applications require that the output be driven to a constant reference input. For example, an airplane autopilot is typically required to maintain the airplane’s heading and altitude at desired constant values (reference inputs). These reference inputs are occasionally changed upon encountering waypoints. Abrupt changes in reference inputs can be described mathematically as step functions." ([12], p. 99)

"Short-duration disturbances can be approximated by impulse functions"

"Constant and step disturbances are also commonly encountered"

"Sinusoidal disturbances ... frequently appear in control applications" ([12], p. 100).
Considering all of these the reference signals are decided to be low frequency constant (set point control) or time-varying (tracking control during waypoint guidance) signals. The disturbances will be considered as deterministic, slowly varying (low-frequency), bounded and nonzero mean signals. Additional stochastic noise components will be considered in system and measurement equations. It is also assumed that the system is causal with unknown future reference and disturbance values. This guarantees the applicability of the developed control solutions also in user controlled systems (manned aircrafts for example, see [64], [69]).

Consequently, causal tracking controllers should be designed for LTI systems subject to low frequency reference and disturbance signals. The next question is the selection of the controller design method considering the selected system, input- and reference signal classes and the computational requirements.

A PID controller is an easily implementable LTI solution, which is capable to track references and attenuate disturbances, but it can be designed only for single input single output (SISO) subsystem models. This increases the complexity of control design because multiple input multiple output (MIMO) systems can only be controlled with cascade PID control which is tedious to be tuned. To have automated design, MIMO design methods, such as linear quadratic (LQ) optimal or $H_\infty$ techniques should be applied.

$LQ$ optimal techniques are widely studied and applied since the 1960s. In the last decades they entered into industrial applications (with LQ regulators, model predictive control (MPC) and preview controllers [16], [23], [33], [34], [60], [68], [71], [82]). The conventional state feedback LQ regulator or tracker results in constant gains and have some robustness against model uncertainties. However, in real applications one needs to use linear quadratic gaussian (LQG) control because the system states are not measured. This way the state dimension of the controller equals with the dimension of the system state vector in case of a full state observer. A drawback of LQG control is the loss of robustness (see [22]).

$H_\infty$ design solves the problem of robustness and disturbance rejection, but highly increases the state dimension of the controller because of the applied weighting functions ([12], [85], [86]). Nowadays new techniques are developed to design fixed-structure $H_\infty$ controllers with lower state dimensions (see [29]). But even in this case the need to track constant nonzero or slowly time-varying references causes a problem. These are usually non-$L_2$ signals and this violates the basic assumption in $H_\infty$ design. The third problem is caused by the low-frequency disturbances which can violate the trade-off in $H_\infty$ design: guarantee good low-frequency performance together with noise attenuation and disturbance rejection on higher frequencies. This is violated if low frequency disturbances act on the system.

So, from LQG and $H_\infty$ techniques the former one is more appropriate to achieve constant gain controllers with minimum controller state space dimension in reference tracking solutions. Another advantage is the applicability of non-$L_2$ references also. Of course the robustness of the resulting system should be carefully examined.

A drawback can be the inability to attenuate low frequency disturbances with the conventional LQ optimal design (which can be also an inability of $H_\infty$ designs). This can be overcome by the use of minimax ([29]) control design and the application of disturbance (unknown input) estimators (such as [31], [32]). The latter can make it possible to cancel
most of the effect of disturbances applying feedforward compensation. This way only the much smaller disturbance residual should be attenuated by the minimax controller and so, performance improvement can be achieved.

The requirement of causality (apply the developed control without knowledge about the future reference values) leads to the need of an infinite horizon LQ and / or minimax control solution, because finite horizon control usually assumes the knowledge of references on the whole horizon. In references [3], [4], [53], [83] finite horizon non-causal LQ trackers or infinite horizon approximate trackers (with no guarantee of asymptotically zero tracking error) are developed. So, an improvement is required in this field.

The applicability of the designed state feedback controllers requires the measurement or estimation of system states. In this thesis an aircraft control example will be introduced. In aircraft stabilization and waypoint guidance, mostly aircraft angular rates and Euler angles are applied in state feedback. Angular rates can be measured with onboard large (conventional) or even small (microelectronic) sensor units, while Euler angles can be measured only with the large conventional gimbal units. This means that Euler angles should be estimated onboard a small aircraft where the large units can not be applied. The Euler angles have a highly nonlinear relation with angular rates and so, a nonlinear estimator (such as the Extended Kalman Filter (EKF)) should be used (see [42] for example).

Referencing all the above statements, the objectives of this thesis are the following:

1. To examine the possibility of deriving an infinite horizon, causal LQ optimal reference tracking solution both for constant and slowly time-varying references considering LTI systems and using state feedback.

2. To consider disturbance estimation with a control input which cancels most of the disturbance effects using the estimated disturbance. To synthesize an infinite horizon, causal minimax reference tracking solution both for constant and slowly time-varying references considering LTI systems. This controller should attenuate the disturbance residual remaining after disturbance cancellation.

3. To apply the developed control solutions in aircraft reference signal tracking problems implementing and testing them in Matlab Simulink simulations and in real flights.

4. Aircraft related application examples inspired the design of an Euler angle state estimator which switches between the sensor measurements and is applicable during the entire flight time from take-off to landing.

The hypotheses behind this are the assumption of the possibility to derive infinite horizon LQ and LQ minimax optimal tracking solutions which give good performance meanwhile have minimal computational requirements. It is also assumed that some improvement can be achieved in the field of LQ optimal tracking control, where the infinite horizon limiting solutions are only approximated in the literature. The last assumption was the possibility to build a multi-mode attitude estimator which uses the most accurate sensor data every time by switching between the sensors.
The used methods are LQ and LQ minimax optimal output tracking control and extended Kalman filter for nonlinear systems. They are briefly described in Appendix 8.1.

The expected results are infinite horizon LQ and LQ minimax optimal (or sub-optimal) LTI trackers, which are easy to tune, causal, inherently MIMO solutions with minimum state dimension. The other expected outcome is an aircraft attitude estimator which can be used during the entire flight from take-off to landing.

The expected tracking results are applicable in any system with limited computational capability where a LTI model well describes the system dynamics. The attitude estimator is applicable in any aircraft equipped with IMU and GPS sensors and a microcomputer.

The structure of this thesis work is the following: chapter 1 introduces the considered system class together with basic notations and assumptions. Chapter 2 deals with the problem of infinite horizon LQ optimal tracking regulators. After a short literature review it describes the basic assumptions, derives the discrete time finite horizon solution and then the infinite horizon one. Then gives a solvability condition for the related discrete algebraic Riccati equation (DARE) and summarizes the properties of the derived infinite horizon LQ tracker solution. Finally shows comparison with other methods through simulation examples. Chapter 3 further improves the results of the previous chapter applying disturbance estimation and compensation with infinite horizon LQ optimal minimax control. It also lists the properties of the derived solution and compares it to other methods through simulation. Chapter 4 steps towards the real applicability of the methods by developing an aircraft attitude estimator EKF. Its properties will be examined through simulation examples. Chapter 5 does Monte Carlo simulations to explore LQ and minimax controller robustness considering the effects of the nonlinear state estimator. Chapter 6 presents real flight test results and comparison of methods with a PID control solution. Chapter 7 summarizes the achieved results in five theses and concludes the dissertation.
Chapter 1

The considered system class

This thesis considers the class of continuous time (CT), LTI systems with deterministic and stochastic disturbances described by

\[
\begin{align*}
\dot{x} &= A_c x + B_c \tilde{u} + G_c d + W_c w \\
y^r &= C_{cr} x \\
y &= C_c x + V_c v
\end{align*}
\] (1.1)

Where \( x \in \mathbb{R}^n \), \( \tilde{u} \in \mathbb{R}^m \), \( d \in \mathbb{R}^d \), \( y^r \in \mathbb{R}^r \), \( y \in \mathbb{R}^p \), \( w \in \mathbb{R}^w \), \( v \in \mathbb{R}^v \) are the system state, input, disturbance (deterministic, low frequency), tracking output, measured output, stochastic disturbance and measurement noise respectively and the matrices \( A_c, B_c, G_c, C_{cr}, C_c, W_c, V_c \) have appropriate dimensions. Considering the application example such system model can well describe the motion of an aircraft around a trim point subject to deterministic engine reaction torque and wind disturbances and to some stochastic disturbance also. \( n \geq m \geq r \) is assumed through all the thesis.

Notice that two outputs are defined. \( y^r \) should track the references (tracking output), while \( y \) is the measured output of the system used in state and unknown input estimation.

The discrete time (DT) equivalent of the above CT system can be represented as follows (the block structure of this system model can be seen in Figure 1.1): here \( z^{-1} \) is the backward time shift operator:

\[
\begin{align*}
x_{k+1} &= A x_k + B \tilde{u}_k + G d_k + W w_k \\
y^r_k &= C_r x_k \\
y_k &= C x_k + V v_k
\end{align*}
\] (1.2)

In the forthcoming controller design chapters it will be assumed that the system state vector \( x_k \) is available for feedback, \( d_k \) is a deterministic, nonzero mean, low frequency disturbance such as constant wind or engine reaction torque. \( w_k \) and \( v_k \) are assumed to be zero mean, uncorrelated, gaussian white noises.
1.1 The aircraft model used in the application example

This section describes the lateral dynamical model of the E-flite Ultrastick 25e aircraft, used in the application example. This model was derived from the model developed in [70]. A brief description of the aircraft can be found in Appendix 8.2. Besides the linear aircraft dynamics, the model contains actuator dynamics and time delay (see Figure 1.2).

$u$, $u_0$, $u_1$ are the input vectors including $\delta_a$ aileron and $\delta_r$ rudder deflections. $x$ is the state vector including $p$ roll rate, $r$ yaw rate and $\phi$ roll angle. $d$ is the disturbance vector which includes $d_L$ roll and $d_N$ yaw torque disturbances from engine and wind effects.

In the simulation model block diagram $u_0(t)$ is the reference input given by the pilot or the controller. *delay* stands for the actuator time delay caused by its electronics, gear transmission and the rod mechanism between servo and control surface. $G_{act}$ represents the dynamics of the actuator. $u(t)$ is the control surface deflection. $u_1(t)$ can not be physically represented because delay and dynamics are only theoretically distinguished (the real actuator does not have a delay and a dynamic part). In the following, the content of the three blocks in figure 1.2 is given. The CT linear dynamic equation of the system ($G_{ac}$) is:
\[
\begin{bmatrix}
\dot{p} \\
\dot{r} \\
\dot{\phi}
\end{bmatrix} =
\begin{bmatrix}
L_p & L_r & 0 \\
N_p & N_r & 0 \\
1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
p \\
r \\
\phi
\end{bmatrix}
+ \begin{bmatrix}
L_\delta_a & L_\delta_r \\
N_\delta_a & N_\delta_r \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
\delta_a \\
\delta_r
\end{bmatrix}
+ \begin{bmatrix}
L_d_L & L_d_N \\
N_d_L & N_d_N \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
d_L \\
d_N
\end{bmatrix}
\]

(1.3)

The coefficients (aircraft stability and control derivatives) in \( A^c \) and \( B^c \) were obtained in \([70]\) using system identification techniques. The output matrix is assumed to be identity \( C^c = I \) in all cases. Three different model parameter sets resulted from three flight measurements. The parameters are summarized in Table 1.1. The flight conditions were indicated airspeed (IAS) between 16-18m/s, altitude between 90-110m and throttle between 45-60%. The aircraft was trimmed in straight and level flight in all cases, then aileron and rudder doublet inputs were applied. The aircraft can fly only in mild wind conditions, so identification measurements were done in such weather.

Table 1.1: Ultrastick aircraft parameters

<table>
<thead>
<tr>
<th>Param.</th>
<th>( L_p )</th>
<th>( L_r )</th>
<th>( N_p )</th>
<th>( N_r )</th>
<th>( L_\delta_a )</th>
<th>( L_\delta_r )</th>
<th>( N_\delta_a )</th>
<th>( N_\delta_r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOD1</td>
<td>-12</td>
<td>12.7</td>
<td>0.294</td>
<td>-8.48</td>
<td>58.1</td>
<td>13.6</td>
<td>-6.58</td>
<td>-17.5</td>
</tr>
<tr>
<td>MOD2</td>
<td>-12.8</td>
<td>14.4</td>
<td>-0.448</td>
<td>-6.08</td>
<td>61.4</td>
<td>12.4</td>
<td>-3.67</td>
<td>-15</td>
</tr>
<tr>
<td>MOD3</td>
<td>-11.1</td>
<td>8.62</td>
<td>0.687</td>
<td>-4.62</td>
<td>43.3</td>
<td>8.99</td>
<td>-4.76</td>
<td>-11.9</td>
</tr>
</tbody>
</table>

The detailed, formal derivation of all of the model coefficients (\( A^c \), \( B^c \), \( G^c \) matrices) is presented in Appendix 8.3.

The considered actuator dynamics was derived by Paw Yew Chai at University of Minnesota (unfortunately there is no article about it). It is a very fast dynamics with time constant 0.04s and damping 0.7 and so, can be approximated with \( G_{act} \approx 1 \). This approximation will be considered in the control design. The real CT transfer function is the following:

\[
G_{act} = \frac{U(s)}{U_1(s)} = \frac{631.6}{s^2 + 35.2s + 631.6} \tag{1.4}
\]

The time delay in the controlled aircraft system is approximately 0.08s published in \([70]\) and verified by the authors in hardware in the loop (HIL) simulation (which includes not only the real onboard microcontroller, but also the real RS-232 communication channels with the same frequency and baudrate as onboard). But tuning the controllers for this delay gave unsatisfactory results in real flight tests so, the real delay should be larger. Examination of real flight data shown that the delay can be about 0.2s so, this value was used finally. In the simulation model this was implemented as an integer delay.
1.1.1 The model used in control design

In the control design, a simplified model was used neglecting actuator dynamics (because $G_{act} \approx 1$) and using the Padé approximation of delay (0.2s). An additional washout filter matrix ($G_{filt}$) was inserted to select the high frequency component of yaw rate (see Figure 1.3).

\[ u_0(t) \xrightarrow{\text{delay}} u(t) \xrightarrow{G_{ac}} y(t) \xrightarrow{G_{filt}} r \]

Figure 1.3: The controlled model block diagram

For the Padé approximation of delay both first and second order functions were tested. The step response of the second order one is better, because it does not start from negative value so, finally it was selected (see Figure 1.4). More detailed information about the Padé modelling of delay can be found in [80] which points out that orders of four or five give acceptable approximation of the delay (with only about 2% error). However, the requirement of low state space dimension does not makes it possible to use such orders. The detailed test results show that the second order approximation was insufficient only in case of MPC disturbance rejecting control, otherwise it was sufficient.

\[
G_{delay} = \begin{bmatrix} 0.004s^2 - 0.1s + 1 \\ 0.004s^2 + 0.1s + 1 \end{bmatrix}
\]

\[
x^d = A_dx^d + B_du_0
\]

\[
u = C_dx^d + D_du_0
\]

(1.5)

Figure 1.4: Padé step responses
In figure 1.3 $G_{filt}$ symbolizes a transfer function matrix with the following structure: 

$$G_{filt} = \frac{\pi(s)}{Y(s)} = \begin{bmatrix} 0 & \frac{s}{s+15} & 0 \end{bmatrix}.$$  

So, only the $r$ yaw rate component is transferred to $\tau$ using the washout filter from [70]. The goal with this filter is to suppress the low-frequency yaw rate and consider only the high-frequency one. The crossover frequency is 2.4 Hz. This provides the opportunity to make a steady turn with the aircraft without activating the yaw damper meanwhile higher frequency yaw rate oscillations will be well damped. Without this filter the rudder would act against the aileron activated turning of the aircraft. Its equivalent state space representation is:

$$\dot{x}^F = A_F x^F + B_F r, \quad \tau = C_F x^F + D_F r$$  \hspace{1cm} (1.6)

Here $x^F$ is filter state, while $\tau$ is the filtered yaw rate. The augmented CT controlled system can be constructed from (1.3), (1.5) and (1.6) as follows:

$$\begin{bmatrix} \dot{x} \\
\dot{x}^d \\
\dot{x}^F \end{bmatrix} = \begin{bmatrix} A & BC_d & 0 \\
0 & A_d & 0 \\
0 & B_F & 0 \end{bmatrix} \begin{bmatrix} x \\
x^d \\
x^F \end{bmatrix} + \begin{bmatrix} BD_d \\
B_d \\
0 \end{bmatrix} u + \begin{bmatrix} G \\
0 \\
0 \end{bmatrix} d$$  \hspace{1cm} (1.7)

$$y^r = \begin{bmatrix} \phi \\
\tau \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\
0 & D_F & 0 & 0 & C_F \end{bmatrix} \begin{bmatrix} p \\
p \\
\phi \\
x^d \\
x^F \end{bmatrix}$$

(1.7) shows that the selected tracking outputs are aircraft roll angle and filtered yaw rate. This will be applied throughout the thesis unless stated otherwise.
Chapter 2

Infinite horizon LQ optimal tracking control

LQ optimal tracking control is a widely covered area in control theory. Several solutions and methods exist. The first group of methods includes the predictive ([23], [34], [50], [57], [64], [65], [69]) and preview ([16], [25], [26], [28], [37], [47], [48], [54], [60], [63], [66], [68], [71]) techniques when the knowledge of future input and disturbance values are assumed. This violates the assumption of causal tracking.

The second group of methods includes the solutions without preview needs. Such derivations can be found in [2], [3], [4], [53], [67] and [83]. In [3] at first, the finite horizon LQ optimal output tracking problem for CT, time-varying linear systems is developed. The optimal control input results as sum of the well known state feedback term (which results in the LQ optimal regulator problem) and a forcing function which depends on the reference input. Both of them can be calculated backward in time starting with the terminal state. This means that the knowledge of the reference signal is required on the whole control horizon. So, unfortunately preview action is needed. This is summarized by the authors as follows:

'Certainly, the formulation of our control problem did not include the requirement of realizability'

'Is there an easy way of incorporating the requirement of realizability in the mathematical formulation of the optimal problem? The answer is no'. ([3], p. 799)

After this results, the authors give approximate relations for LTI systems with constant reference signals and very large horizons. Infinite horizon can not be considered, because then the cost functional becomes infinite and so, can not be minimized. The final conclusion is as follows:

'Unfortunately, at this time, a rigorous exposition of the limiting case \(T = \infty\) is not available' ([3], p. 803)

The derivation of the same results as in [3] for DT and CT, time-varying, linear systems can be found in [53]. After deriving the finite horizon results, some extension to infinite horizon is given. But the published solution does not guarantee the finite functional value and so, optimality is questionable (see the citation from [3] p. 803 above). For constant reference values [53] cites the solution in [3].

In [2] similar results to those of [3] and [53] are published, and a couple of other solutions are provided which are based on considering an autonomous dynamical system.
which generates the reference input. The tracking controller is then designed with augmenting the original system with the additional states. The designed controller can be applied only, if the additional states are observed from the reference output which needs the implementation of an additional observer. This increases the complexity of control which should be avoided because of the computational burden. The same results for discrete-time systems are also published. The authors in [2] also publish results with proportional integral control which is very similar to LQ Servo control (see [64]) in which the controller contains additional integrators. This highly improves the tracking performance but the implementation of integrators means additional computational load.

In [4] a system of linear algebraic equations is formulated to determine the initial condition of the forcing function resulted in [3] (CT systems). But this requires the type of the reference signal to be known and the system to have moderate dimensions (they use polynomial matrices in calculations).

In [83] a rigorous solution (according to the title of the article) of the infinite time interval LQ optimal tracking problem is given. The authors give a two step solution for the problem. The first step is the LQ optimal determination of the steady state, the second is to guide the system LQ optimally into this state. It is pointed out that the two problem result in the same control law and so, can be solved with one unified control. However, the LQ optimal steady state search does not guarantee zero steady state tracking error even for constant references. The authors give a counterexample to show this. They reference [73] in which a cost functional centered with the steady state input and state is used in the design problem. This idea is worth to be considered because possibly guarantees finite functional values in the infinite horizon problem.

Other useful ideas to solve the infinite horizon, causal LQ optimal tracking problem are published in [64]. These are to rewrite the tracking problem in error coordinates (centered functional as above) and to apply extrapolation for the unknown future references.

[67] publishes a predictive control like method with off-line optimization for the LQ optimal state tracking problem. The two step reference preview is a useful idea if one applies extrapolation, but otherwise the published method is too complicated with a large amount of off-line calculation.

All these books and articles mainly deal with CT systems. In tracking problems the Hamiltonian equation is not a homogeneous, but an inhomogeneous one because of a forcing term which includes the effect of the reference signal (see [2], [3], [4] and [53]). This can be solved in continuous time only with an infinite horizon integral relation (in the resulting forcing function). However, in DT formulation there can be chance to solve this problem easier (see the DT formulation in [53]). That is why the author decided to consider the stated problems in discrete time. Another issue is that a DT algorithm can be directly implemented on a microcomputer without any transformation.

As a summary, the goal of the thesis’s first part is to derive a DT, causal, infinite horizon LQ optimal output tracking controller for LTI systems guaranteeing zero steady state tracking error at least for constant references. The starting point of this derivation is the finite horizon DT solution in [53] mixed with the ideas coming from [2], [64] and [73]. Another outcome of this chapter is the characterization of the solvability conditions for the resulting DARE and the proposal of a weighting strategy which can guarantee solvability.
In this chapter the $d_k$ deterministic disturbance is assumed to be zero and the effects of $w_k$ and $v_k$ are assumed to be considered in the state estimation. This way the estimated $x_k$ state vector is assumed to be available for feedback and the considered system is as follows:

$$
x_{k+1} = Ax_k + B\tilde{u}_k
$$

$$
y^r_k = C_rx_k \quad y_k = Cx_k
$$

Here $y^r_k$ is the output which should track the reference signal $r_k$, the measured outputs ($y_k$) of the system can be different from it. It is assumed that, the pair $(A,B)$ is stabilizable.

### 2.1 The finite horizon discrete time LQ optimal tracker

The finite horizon output tracking problem for (2.1) can be defined with the following functional:

$$
J(y, e, u) = \frac{1}{2} \sum_{k=0}^{N-1} \left( y^T_k Q_1 y_k + e^T_k Q_2 e_k + \tilde{u}_k^T R\tilde{u}_k \right) +
\frac{1}{2} \left( y_N^T Q_1 y_N + e_N^T Q_2 e_N \right)
$$

where:

$$
e_k = y^r_k - r_k = C_r x_k - r_k
$$

$$
\overline{y}_k = \overline{C}_x x_k = \left( I - C_r^T (C_r C_r^T)^{-1} C_r \right) x_k
$$

Here, $Q_1 \geq 0$, $Q_2 \geq 0$ and $R > 0$ are symmetric weighting matrices. The term $\overline{y}^T_k Q_1 \overline{y}_k$ is optional, but it can be required to make the resulting DARE (for infinite horizon) solvable if $A$ has eigenvalue(s) on the unit circle or to weight the states not considered in $y^r_k$ ($\overline{y}_k$ is the orthogonal projection of $x_k$ onto the null space of $C_r$) (for details see section 2.3).

This functional can be rewritten using $\tilde{x}_k = C_r^T (C_r C_r^T)^{-1} r_k = H r_k$ (see [2]):

$$
J(x, \tilde{x}, u) = \frac{1}{2} \sum_{k=0}^{N-1} \left( (x_k - \tilde{x}_k)^T Q (x_k - \tilde{x}_k) + \tilde{u}_k^T R\tilde{u}_k \right) +
\frac{1}{2} \left( x_N - \tilde{x}_N \right)^T Q (x_N - \tilde{x}_N)
$$

where:

$$
Q = \overline{C}^T Q_1 \overline{C} + C_r^T Q_2 C_r
$$

The optimization problem stated in (2.3) is formally similar to the one defined in [53] chapter 2.6. The difference is the additional $Q_1$ weighting which can have a crucial role in problem solvability as stated above. The optimization can be solved using Lagrange multiplier method. Most of the results are derived in [53] but the solution published here gives deeper insight into the structure of the forcing variable which will make it possible
to remove the need for backward calculation (and to know the whole reference signal sequence in advance). The structure of the costate variable results as (for details see Appendix 8.4 which builds on the results of [53]).

\[ \lambda_k = P_k x_k + S_k \tilde{x}_{k+1} - Q \tilde{x}_k = P_k x_k - s_R(k) \quad (s_R(k) = Q \tilde{x}_k - S_k \tilde{x}_{k+1}) \quad (2.4) \]

From the assumed structure of the costate variable (\( \lambda_k = P_k x_k - s_R(k) \)) [53] derives the same Riccati difference equation (2.6-19), forcing function (2.6-20) and optimal control input (2.6-25, 2.6-26) as:

\[
\begin{align*}
P_N &= Q s_R(N) = QHr_N \\
P_k &= Q + A^T P_{k+1} A - A^T P_{k+1} B \left[ B^T P_{k+1} B + R \right]^{-1} B^T P_{k+1} A \\
s_R(k) &= QHr_k + A^T \left[ I + P_{k+1} B R^{-1} B^T \right]^{-1} s_R(k + 1) \quad (2.5) \\
\tilde{u}_k &= -R^{-1} B^T P_{k+1} \left[ I + B R^{-1} B^T P_{k+1} \right]^{-1} A x_k + \\
&\quad + R^{-1} B^T \left[ I + P_{k+1} B R^{-1} B^T \right]^{-1} s_R(k + 1) = -R^{-1} B^T \lambda_{k+1}
\end{align*}
\]

The second line in (2.5) is the recursive algebraic Riccati equation (ARE) from which the \( P_k \) matrix can be determined. The third line is the discrete equivalent of the forcing function resulted in CT in [2] and [3]. As can be seen the implementation of this finite horizon control needs the a-priori knowledge of the reference signal because the equations can be solved only backward (from index \( N \) to 1). In the literature unfortunately all the published methods required the backward solution of the forcing equation. Note that the optimal control at time \( k \) depends on \( r_{k+1} \) and \( r_{k+2} \) which is the same result as derived in [67] from an iterative optimized trajectory tracking procedure. But hopefully (and indeed, see section 2.2) one can overcome this difficulty considering the infinite horizon case. But care must be taken in the derivation of the infinite horizon solution, because for nonzero reference input and zero tracking error, the output and so, the input will be nonzero which means that the nonzero input drives \( J(y,e,u) \) into \( \infty \). The distinction between steady state calculation and transient control and the centralization of the functional (see [73] and [83]) will be used to solve this problem in the next section.

### 2.2 The infinite horizon, discrete time LQ optimal tracker

In this section the goal is to solve the infinite horizon version of the LQ optimal tracking problem. The solution can be constructed as a three step process:

1. Design a stabilizing state feedback controller for the pair \((A, B)\) in (2.1)

2. Determine the solution of the steady state constant reference tracking problem considering the stabilized system
3. Construct an LQ sub-optimal tracking controller for time-varying references, centering the original system with the steady state equilibrium point and the steady state reference value

### 2.2.1 Design of a stabilizing state feedback controller for \((A, B)\)

This can be solved either with pole placement or with LQ optimal regulator design. The resulting system equations can be written as follows (considering additional input to be applied later in tracking):

\[
x_{k+1} = (A - BK_{x1}) x_k + Bu_k
\]

\[
x_{k+1} = \Phi x_k + Bu_k
\]

\[y_k = C r x_k\] (2.6)

### 2.2.2 Determining the solution of the steady state constant reference tracking problem

The next step is to assume steady state with constant state, input and reference vectors and calculate the required steady input value.

\[
x_{\infty} = \Phi x_{\infty} + Bu_{\infty}
\]

\[y_{\infty} = C r x_{\infty} = r_{\infty}\]

\[(I - \Phi) x_{\infty} = Bu_{\infty} \rightarrow x_{\infty} = (I - \Phi)^{-1} Bu_{\infty}\]

\[C r (I - \Phi)^{-1} B u_{\infty} = r_{\infty}\]

\[u_{\infty} = F^+ r_{\infty}\] (2.7)

Here the existence of \((I - \Phi)^{-1}\) requires \(\Phi\) not to have eigenvalues on the unit circle that’s why it was designed to be a stable DT system matrix in the previous step. \((\cdot)^+\) denotes the Moore - Penrose pseudoinverse of a matrix. \(F\) is an \(r \times m\) matrix and its pseudoinverse (or inverse if \(r = m\)) usually exists with \(FF^+ = I\) and so guarantees reference tracking in steady state \((r \leq m\) was assumed).

### 2.2.3 Construction of the infinite horizon LQ sub-optimal output tracking controller

The required steady state input to track a constant reference signal can be calculated using (2.7). However, the control of the transient from initial state to steady state should be considered. This can be designed together with the solution of cases with time-varying references in a unified framework as follows.

The modified state dynamic equation (2.6) and the steady state system equation are:
The equations in (2.8) can be subtracted from each other, giving another modified state equation:

\[ x_{k+1} - x_\infty = \Phi (x_k - x_\infty) + B (u_k - u_\infty) \]
\[ \Delta x_{k+1} = \Phi \Delta x_k + B \Delta u_k \] (2.9)

The second equation in (2.9) gives a system dynamics around the steady state. This equation together with the centered reference signal \( \Delta r_k = r_k - r_\infty \) can be used to form an LQ optimal tracking problem for the transient (in case of constant references) or for the case with time-varying references. The formulated problem is the same as in [BKB08a] but the solution will be different. The finite horizon functional for the centralized system is similar to (2.2):

\[
J(\Delta y, \Delta e, \Delta u) = \frac{1}{2} \sum_{k=0}^{N-1} \left( \Delta y^T_k Q_1 \Delta y_k + \Delta e^T_k Q_2 \Delta e_k + \Delta u^T_k R \Delta u_k \right) + \\
+ \Delta y^T_N Q_1 \Delta y_N + \Delta e^T_N Q_2 \Delta e_N
\]
where:
\[
\Delta e_k = \Delta y^r_k - \Delta r_k = C_r \Delta x_k - \Delta r_k \\
\Delta y_k = C \Delta x_k = \left( I - C^T_r (C_r C^T_r)^{-1} C_r \right) \Delta x_k
\] (2.10)

After the same transformation a functional very similar to (2.3) can be obtained:

\[
J(\Delta x, \Delta \hat{x}, \Delta u) = \frac{1}{2} \sum_{k=0}^{N-1} \left( \left( \Delta x_k - \Delta \hat{x}_k \right)^T Q \left( \Delta x_k - \Delta \hat{x}_k \right) + \Delta u^T_k R \Delta u_k \right) + \\
+ \left( \Delta x_N - \Delta \hat{x}_N \right)^T Q \left( \Delta x_N - \Delta \hat{x}_N \right)
\] (2.11)
where:
\[
Q = \overline{C}^T Q_1 \overline{C} + C^T Q_2 C
\]

The problem represented by (2.11) has the same optimal solution as in (2.5):

\[
P_N = Q \quad s_R(N) = Q H \Delta r_N \\
P_k = Q + \Phi^T P_{k+1} \Phi - \Phi^T P_{k+1} B \left[ B^T P_{k+1} B + R \right]^{-1} B^T P_{k+1} \Phi \\
s_R(k) = Q H \Delta r_k + \Phi^T \left[ I + P_{k+1} B R^{-1} B^T \right]^{-1} s_R(k + 1) \\
\Delta u_k = - R^{-1} B^T P_{k+1} \left[ I + B R^{-1} B^T P_{k+1} \right]^{-1} \Phi \Delta x_k + \\
+ R^{-1} B^T \left[ I + P_{k+1} B R^{-1} B^T \right]^{-1} Q H \Delta r_{k+1} - \\
- R^{-1} B^T \left[ I + P_{k+1} B R^{-1} B^T \right]^{-1} S_{k+1} H \Delta r_{k+2} = - R^{-1} B^T \lambda_{k+1}
\] (2.12)
The infinite horizon solution can be constructed based-on [53] p. 118 (2.6-35 and 2.6-36). It states that the optimal infinite horizon solution can be obtained by substituting $P_\infty$ into all of the expressions. $P_\infty$ is the solution of the steady state (infinite horizon) algebraic Riccati equation called DARE:

$$P_\infty = Q + \Phi^T P_\infty \Phi - \Phi^T P_\infty B \left[ B^T P_\infty B + R \right]^{-1} B^T P_\infty \Phi$$ (2.13)

Substituting $P_\infty$ into the other expressions in (2.12) one gets:

$$s_R(k) = QH \Delta r_k + \Phi^T \left[ I + P_\infty BR^{-1}B^T \right]^{-1} \Phi \Delta x_k +$$

$$\Delta u_k = - R^{-1}B^T P_\infty \left[ I + BR^{-1}B^T P_\infty \right]^{-1} \Phi \Delta x_k +$$

$$+ R^{-1}B^T \left[ I + P_\infty BR^{-1}B^T \right]^{-1} QH \Delta r_{k+1} -$$

$$- R^{-1}B^T \left[ I + P_\infty BR^{-1}B^T \right]^{-1} S_{k+1} H \Delta r_{k+2}$$ (2.14)

These are the optimal equations according to [53]. From now, the only question is the existence of a steady state solution of the forcing function $s_R(k)$. To obtain this solution the detailed structure of $s_R(k)$ and $s_R(k+1)$ should be substituted into the forcing function with generalized terms $s_R(k) = QH \Delta x_k - S_k H \Delta x_{k+1} = S_1 \Delta r_k - S_2 \Delta r_{k+1}$.

$$s_R(k) = QH \Delta r_k + \Phi^T \left[ I + P_\infty BR^{-1}B^T \right]^{-1} s_R(k+1)$$

$$S_1 \Delta r_k - S_2 \Delta r_{k+1} = QH \Delta r_k + \Phi^T \left[ I + P_\infty BR^{-1}B^T \right]^{-1} (S_1 \Delta r_{k+1} - S_2 \Delta r_{k+2})$$ (2.15)

From (2.15) one gets the following system of equations (applying a multiplication by -1):

$$- S_1 \Delta r_k = -QH \Delta r_k$$

$$S_2 \Delta r_{k+1} = -\Phi^T \left[ I + P_\infty BR^{-1}B^T \right]^{-1} S_1 \Delta r_{k+1}$$ (2.16)

$$0 = \Phi^T \left[ I + P_\infty BR^{-1}B^T \right]^{-1} S_2 \Delta r_{k+2}$$

Unfortunately it is impossible to satisfy the third equation for nonzero $\Delta r_{k+2}$ reference values. So, the general LQ optimal solution of the problem is impossible. However, in real applications at time instant $k$ $\Delta r_{k+2}$ usually should be considered with linear extrapolation because it is not known (see [BKB08a]). Considering this fact a sub-optimal selection of $S_1$ and $S_2$ is possible:

$$\Delta r_{k+2} = 2 \Delta r_{k+1} - \Delta r_k$$

$$- S_1 \Delta r_k = -QH \Delta r_k - \Phi^T \left[ I + P_\infty BR^{-1}B^T \right]^{-1} S_2 \Delta r_k$$

$$S_2 \Delta r_{k+1} = -\Phi^T \left[ I + P_\infty BR^{-1}B^T \right]^{-1} S_1 \Delta r_{k+1} + 2 \Phi^T \left[ I + P_\infty BR^{-1}B^T \right]^{-1} S_2 \Delta r_{k+1}$$ (2.17)

From (2.17) a system of equations for $S_1$ and $S_2$ can be formulated (using $M_2 = \left[ I + P_\infty BR^{-1}B^T \right]^{-1}$):

16
\[ S_1 - \Phi^T M_2 S_2 = QH \]
\[ \Phi^T M_2 S_1 + (I - 2\Phi^T M_2) S_2 = 0 \]
\[ \begin{bmatrix} I & -\Phi^T M_2 \\ \Phi^T M_2 & I - 2\Phi^T M_2 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} QH \\ 0 \end{bmatrix} \]  \hspace{1cm} (2.18)

(2.18) can be solved if \( Z \) is invertible. Calculate its determinant using block matrix determinant lemma:

\[
det(Z) = 1 \cdot det(I - 2\Phi^T M_2 - \Phi^T M_2 (-\Phi^T M_2)) = det(I - 2\Phi^T M_2 + (\Phi^T M_2)^2) = \]
\[
= det \left((I - \Phi^T M_2)^2\right) \neq 0 \hspace{1cm} (2.19)
\]

Because \( I - \Phi^T M_2 = I - \Phi^T \left[I + P_\infty BR^{-1}B^T\right]^{-1} \) is nonsingular (see Appendix 8.5), the inverse of \( Z \) and so, the solution can be calculated using block matrix inversion:

\[
\begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} I - \Phi^T M_2 \left((I - \Phi^T M_2)^2\right)^{-1} \Phi^T M_2 & (\ldots) \\ - \left((I - \Phi^T M_2)^2\right)^{-1} \Phi^T M_2 & (\ldots) \end{bmatrix} \begin{bmatrix} QH \\ 0 \end{bmatrix} \hspace{1cm} (2.20)
\]

Finally the sub-optimal solution for \( S_1 \) and \( S_2 \) results as:

\[
S_1 = \left[I - \Phi^T M_2 \left((I - \Phi^T M_2)^2\right)^{-1} \Phi^T M_2\right] QH 
\]
\[
S_2 = - \left((I - \Phi^T M_2)^2\right)^{-1} \Phi^T M_2 QH \hspace{1cm} (2.21)
\]

This way the control input for the centralized problem can be obtained from (2.12) considering the extrapolation of the reference signal:

\[
\Delta u_k = - R^{-1} B^T P_\infty \left[I + BR^{-1}B^T P_\infty\right]^{-1} \Phi \Delta x_k + R^{-1} B^T M_2 S_1 \Delta r_{k+1} - \\
- R^{-1} B^T M_2 S_2 \Delta r_{k+2} = -K_{x2} \Delta x_k + K_{S_1} \Delta r_{k+1} - K_{S_2} \Delta r_{k+2} = \\
= - K_{x2} \Delta x_k + \left(K_{S_1} - 2K_{S_2}\right) \Delta r_{k+1} + K_{S_2} \Delta r_k \hspace{1cm} (2.22)
\]

However, we need to solve the original control problem considering the system in (2.1). The required input of the original system can be constructed considering (2.6), (2.7), (2.9), (2.22) and the estimated state \( \hat{x}_k \) instead of the real state \( x_k \):
\[ u_k - u_\infty = -K_{x_2} (x_k - x_\infty) + K_{S_1} (r_{k+1} - r_\infty) + K_{S_2} (r_k - r_\infty) \]
\[ u_k = -K_{x_2} x_k + K_{S_1} r_{k+1} + K_{S_2} r_k + K_{x_2} x_\infty - (K_{S_2} + K_{S_1}) r_\infty + u_\infty \]
\[ u_\infty = F^+ r_\infty \& x_\infty = (I - \Phi)^{-1} B u_\infty = (I - \Phi)^{-1} B F^+ r_\infty \]
\[ u_k = -K_{x_2} \hat{x}_k + K_{S_1} r_{k+1} + K_{S_2} r_k + \]
\[ + [K_{S_2} - K_{S_1} + (K_{x_2} (I - \Phi)^{-1} B + I) F^+] r_\infty \]
\[ \hat{u}_k = -K_{x_1} \hat{x}_k + u_k = - (K_{x_1} + K_{x_2}) \hat{x}_k + K_{S_1} r_{k+1} + K_{S_2} r_k + \]
\[ + [K_{S_2} - K_{S_1} + (K_{x_2} (I - \Phi)^{-1} B + I) F^+] r_\infty = \]
\[ = -K_{x_2} \hat{x}_k + K_{S_1} r_{k+1} + K_{S_2} r_k + K_{r_\infty} r_\infty \]

This way the sub-optimal solution of the infinite horizon LQ optimal output tracking problem is derived. It is sub-optimal for time-varying references because the steady state of the forcing function was only approximately derived. However, it is optimal for constant references because \( \Delta r_i = r_i - r_\infty \) satisfies the system of equations in (2.16) for any \( r_\infty \). In a real application \( r_\infty \) is usually unknown, and so it can be substituted by \( r_{k+1} \), the actual value of the reference signal in the control input \( \hat{u}_k \).

The closed loop interconnection structure of the application example system with LQ tracker can be seen in figure 2.1 (for notations see Appendix 8.10).

In the next section the solvability conditions of the related DARE in (2.13) will be examined.

### 2.3 DARE solvability conditions

In this section the solvability of the following infinite horizon DARE will be examined in detail assuming no pre-stabilization of the system (2.1).

\[
P_\infty = Q + A^T P_\infty [I + BR^{-1} B^T P_\infty]^{-1} A \]
\[
Q = \bar{C}^T Q_1 \bar{C} + C^T Q_2 C \]
\[
\bar{C} = \left( I - C_r^T (C_r C_r^T)^{-1} C_r \right) \]

Figure 2.1: Closed loop interconnection system with LQ tracker
The DARE contains a state weighting $Q$ and an input weighting $R$ matrix. Usually, $Q \geq 0$ and $R > 0$ are the given conditions for DARE solvability. In output tracking problems, the role of $Q$ is slightly changed. It represents the combination of output tracking error weighting matrix $Q_2 \geq 0$ and an additional matrix $Q_1 \geq 0$. $Q_2$ weights the tracking error related to the tracking outputs $y_k^r = C_r x_k$. The $Q$ matrix satisfies to be positive semi-definite but this is not a sufficient condition for DARE solvability if one considers open-loop systems with the poles of $A$ on the unit circle. Such systems are called type of integral (Type – I) often covering real physical phenomenons and therefore these are strictly related to real applications \[BRSB08\].

In (2.6) $C_r \in \mathbb{R}^{r \times n}$ means that $\max (\text{rank}(C_r)) = r$ ($r \leq n$ and so, it can have $r$ linearly independent rows and also columns). Considering this maximum rank, (2.6) can be transformed with a permuting similarity transformation ($\tilde{x}_k = T S x_k$) to achieve a form in which $C_r$ is the following:

$$\tilde{C}_r = [V \ V M] \text{ where } y_k^r = \tilde{C}_r \tilde{x}_k \quad (2.25)$$

In (2.25) the $V$ block contains the $r$ linearly independent columns of $C_r$ and so, it is a full rank quadratic matrix ($V \in \mathbb{R}^{r \times r}$). $M$ is an appropriately selected $r \times (n - r)$ matrix which constructs the other linearly dependent columns of $C_r$ from $V$.

In special cases, when the number of states affecting the controlled output ($y_k^r$) equals with it’s dimension $r$, $M = 0$ because all of the linearly dependent columns in $C_r$ are zero vectors.

In the rest of this chapter, $A, B, C_r$ will denote the matrices of the transformed system equations $\tilde{A} = T S A T_S^{-1}, \ B = T S B, \ \tilde{C}_r = C_r T_S^{-1}$ respectively. Now, the sufficient conditions for DARE solvability, and the possible problems for type–I open-loop systems will be examined considering these matrices.

Usually, the conditions given to obtain a solvable DARE are $Q \geq 0$ and $R > 0$ considering a DARE in the following form (which is equivalent to (2.24)):

$$P_\infty = A^T P_\infty A + Q - A^T P_\infty B (B^T P_\infty B + R)^{-1} B^T P_\infty A \quad (2.26)$$

However, textbook [14] p. 135 gives two, more accurate conditions which have to be satisfied to solve equation (2.26). These are the followings:

1. **Condition 1**: The pair $(A, BR^{-1/2})$ has to be stabilizable.

2. **Condition 2**: The pair $(Q^{1/2}, A)$ should not have unobservable modes on the unit circle.

Now examine these two conditions for type–I open-loop systems (or for systems having poles on the unit circle). The first condition is guaranteed to be satisfied, if the system matrices combined with the inverse square root of the input weight satisfy the Kalman rank condition:

$$\text{rank } \begin{bmatrix} BR^{-1/2} & ABR^{-1/2} & \ldots & A^{n-1} BR^{-1/2} \end{bmatrix} = n \quad (2.27)$$

19
However, $R^{-1/2}$ can be moved out from the brackets which results in the following condition:

$$\text{rank} \left( \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} R^{-1/2} \right) = n$$  \hspace{1cm} (2.28)

The original condition for $R$ was its positive definiteness, which also means that the matrix is full rank and invertible. So, its inverse square root is also full rank, which means that it does not modify the rank of the matrix product in (2.28). From this, it is obvious that, if the pair $(A, B)$ is controllable (it satisfies the Kalman rank condition) the pair $(A, BR^{-1/2})$ is also controllable.

However, we only assumed the stabilizability of the pair $(A, B)$. Multiplication of $B$ by a full rank matrix preserves its rank and also the properties of the pair $(A, B)$. So, the stabilizability condition ($\text{Condition 1}$), is satisfied by any plant with stabilizable $(A, B)$ pair. This means that, the first DARE solvability condition is satisfied in these cases.

The satisfaction of $\text{Condition 2}$ is not so straightforward. The Kalman rank condition can not be used to decide about unobservable modes on the unit circle, instead another condition (given in [14] p. 354) should be used. This is the following:

An eigenvalue (mode) $\lambda$ of $A$ is observable (considering the pair $(Q^{1/2}, A)$) iff

$$\text{rank} \left( \begin{bmatrix} A - \lambda I \\ Q^{1/2} \end{bmatrix} \right) = \text{rank}(A) = n$$  \hspace{1cm} (2.29)

So, the second solvability condition can be checked testing (2.29) for every eigenvalue of the DT state matrix $A$ on the unit circle. To point out the problems, assume that the $A$ matrix is transformed into Jordan canonical form with transformation matrix $T$ and contains the examined $\lambda$ eigenvalue in its first Jordan block (this can be again achieved with transformation by permuting matrices). Assume also that, the $Q$ matrix is permuted similarly (but preserve the notation $Q$). In this way the eigenvalue observability condition becomes:

$$\text{rank}(A) = \text{rank} \left( \begin{bmatrix} T(J - \lambda I)T^{-1} \\ TT^{-1}Q^{1/2}TT^{-1} \end{bmatrix} \right) = n$$  \hspace{1cm} (2.30)

where:

$$J = \begin{bmatrix}
\lambda & 1 & 0 & 0 & \ldots & 0 \\
0 & \lambda & 1 & \vdots & \vdots \\
0 & 0 & \lambda & 0 & \ldots & 0 \\
0 & \cdots & \cdots & \cdots & \cdots & X
\end{bmatrix}$$

Here $\lambda$ was assumed to be on the unit circle and have a multiplicity of three, but the method works for any multiplicity. $X$ means the bidiagonal matrix constructed from the other Jordan blocks. Considering the above Jordan canonical form, the rank condition will be the following (from (2.30)):

$$\text{rank} \left( \begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\
0 & 0 & 1 & \vdots & \vdots \\
0 & 0 & 0 & 0 & \ldots & 0 \\
0 & \cdots & \cdots & \cdots & \cdots & X
\end{bmatrix} T^{-1} \right) = n$$  \hspace{1cm} (2.31)

20
This shows that the rank of $A - \lambda I$ will be $\text{rank}(A) - 1 = n - 1$ if $\text{rank}(A) = n$ (assume this best case) because another zero column (and also row) is generated. The matrix in the rank condition is $2n \times n$ so its rank will be below $n$ if it has one or more linearly dependent column or $n + 1$ or more linearly dependent row.

The original condition for $Q$ is its positive semi definiteness which means that, it can be a zero matrix. At first, examine this case. If $Q$ is a zero matrix, then $\Lambda$ will have at least one zero column so, its rank is below $n$ and the $(Q^{1/2}, A)$ pair has unobservable mode on the unit circle. In this case $\Lambda$ has at least $n + 1$ zero rows so, its row rank (i.e. the number of linearly independent rows) is also below $n$. This means that, $Q$ should not be zero matrix (and usually it isn’t). The question is that, how close has to be $Q$ to a positive definite matrix (how many zero and nonzero eigenvalues are allowed) to satisfy Condition 2.

Of course, Condition 2 depends on the system matrix $A$. If one examines (2.31) it is obvious that, if $Q$ has even one nonzero row different from all the rows of $J - \lambda I$ then $\Lambda$ has less than $n + 1$ linearly dependent rows (if $\text{rank}(A) = n$) so, its row rank is $\geq n$. This means that only its column rank (i.e. the number of linearly independent columns) can be below $n$. Now assume $Q$ in the following form:

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & Y \end{bmatrix}$$

(2.32)

where $Y$ diagonal with elements in $\mathbb{R}$

Examine $T^{-1}Q^{1/2}T$ in (2.31). At first, consider the left multiplication with $T^{-1}$ this results in:

$$\begin{bmatrix} 0 & Z_1 \\ 0 & Z_2 \end{bmatrix} T$$

(2.33)

where $Z_1$ and $Z_2$ are full blocks

Now the column rank of the matrix in (2.31) depends on the $T$ matrix which is unpredictable without the knowledge of $A$ (will the first column of $T^{-1}Q^{1/2}T$ result as zero, or not?). This means that the assumption (2.32) can be relaxed to consider any nonzero $Q$ matrix. Experiences show that, if $Q$ has a lot of zero columns (or equivalently rows because $Q$ is symmetric), then the probability of losing column rank in (2.31) is high and this violates Condition 2.

However, if $Q > 0$ holds then $T^{-1}Q^{1/2}T$ will be full rank and the pair $(Q^{1/2}, A)$ will not have unobservable modes on the unit circle.

As a summary, it can be stated that, Condition 1 is satisfied by the considered system class, meanwhile the satisfaction of Condition 2 highly depends on the selection of the $Q$ matrix. In the following, the weighting strategy with $Q_1$ and $Q_2$ (originally proposed in [2]) will be examined, taking special attention to the positive definiteness of $Q$.

Considering $Q_1 = 0$ which means $Q = C_r^TQ_2C_r$ where $C_r \in \mathbb{R}^{r \times n}$, $Q_2 \in \mathbb{R}^{r \times r}$ (it weights the output tracking error) and so, $Q \in \mathbb{R}^{n \times n}$. If $Q_2 > 0$ it is a full rank ($\text{rank}(Q_2) = r$) invertible matrix, but $C_r$ has only rank $r$. Matrix multiplication can not increase the rank of the product, which means that $Q$ will be a matrix with rank $r$ or
below r. So, it is a singular matrix and not positive definite. This can probably result in an unsolvable DARE mainly for open-loop systems having many poles on the unit circle.

Another problem can be the lack of constraints on states unaffected by \( Q = C_r^T Q_2 C_r \). This is obvious for designs with \( M = 0 \) (\( C_r = [V \ 0] \)) where \( Q \) results as:

\[
Q = \begin{bmatrix} V^T Q_2 V & 0 \\ 0 & 0 \end{bmatrix}
\]

The unaffected states will not be constrained by the controller, which can result in poor system performance and even in system damage.

[2] proposes a weighting strategy with nonzero \( Q_1 \) which solves both of the problems (DARE solvability and state weighting) for the class of systems considered here and with output tracking design characterized by \( M = 0 \). This will be pointed out in the following.

Examine the structure of the combined state weighting matrix:

\[
Q = \overline{C}^T Q_1 \overline{C} + C_r^T Q_2 C_r
\]

(2.34)

Above, it is pointed out that, \( Q > 0 \) guarantees DARE solvability so, if (2.34) gives a positive definite matrix, the DARE in (2.24) will be solvable (for \( R > 0 \)).

\( Q \) is the state weighting matrix in LQ optimal control, so it characterizes the quadratic term for the states (or state tracking errors here) in the functional: \( x_k^T Q x_k \). However, \( x_k \) can be partitioned into two parts considering the partitioning of \( C_r = \overline{C}r \) in (2.25):

\[
x_k^T = \begin{bmatrix} x_k^T I_k^T \\ x_k^T II_k^T \end{bmatrix}
\]

(2.35)

Assume that, \( Q_1 \) and \( Q_2 \) are positive definite matrices, but partition \( Q_1 \) in the same way as \( x_k \):

\[
Q_1 = \begin{bmatrix} Q_{1,11} & 0 \\ 0 & Q_{1,22} \end{bmatrix}
\]

(2.36)

In (2.34) \( \overline{C} \) and \( C_r \) are rank deficient matrices so, \( Q \) will be only a positive semidefinite matrix if \( Q_1 \) and \( Q_2 \) are positive definite and one assumes general matrices. However, \( \overline{C}, C_r \) and \( Q_1 \) have special structure (as assumed) so, one has the possibility to have a positive definite resultant \( Q \) matrix. From these preliminaries considering (2.25) and by evaluating \( \overline{C} \) and (2.34) one gets (with algebraic manipulations) three different matrices (for details see Appendix 8.3) with \( M_3 = I + M^T M \):

\[
\begin{align*}
\overline{Q}_{1,11} &= M_3^{-T} M^T Q_{1,11} M M_3^{-1} \geq 0 \\
\overline{Q}_{1,22} &= M_3^{-T} Q_{1,22} M_3^{-1} > 0 \\
\overline{Q}_2 &= V^T Q_2 V > 0
\end{align*}
\]

\( \overline{Q}_{1,11} + \overline{Q}_{1,22} \) can be denoted as \( \overline{Q}_1 > 0 \). It can be verified that the quadratic expression \( x_k^T Q x_k \) can be equivalently written in the following form by considering (8.22) and (2.35):

\[
x_k^T Q x_k = \begin{bmatrix} x_k^T & x_k^T M & -x_k^T I^T & x_k^T M^T \end{bmatrix} \begin{bmatrix} Q_2 & 0 & 0 & \overline{Q}_2 \\ 0 & Q_1 & 0 & \overline{Q}_1 \\ 0 & \overline{Q}_1 & 0 & \overline{Q}_2 \\ \overline{Q}_2 & 0 & 0 & \overline{Q}_2 \end{bmatrix} \begin{bmatrix} x_k^T \\ M^T x_k^T \\ -x_k^T I \\ M x_k^T \end{bmatrix}
\]

(2.37)

22
In this way, the positive definiteness of $Q$ needs the positive definiteness of the matrix on the right hand side of (2.37) (if one considers $x_k$ with arbitrary nonzero $x_k^I$ and $x_k^II$ components). This property can be examined using the Sylvester criterion, which states that, a matrix is positive definite if all the determinants of its leading principal submatrices are positive. Examine the determinant of the $3 \times 3$ quadratic block of the matrix given in (2.37) using the lemma for the determinant of a block matrix and considering the properties: $Q_1 > 0$ and $Q_2 > 0$.

$$\det \begin{vmatrix} Q_2 & 0 & 0 \\ 0 & Q_1 & Q_1 \\ 0 & Q_1 & Q_1 \end{vmatrix} = \det (Q_2) \det (Q_1).$$

The determinant in (2.38) is zero, because the last term contains the determinant of a zero matrix. This means that, we have found a leading principal submatrix which has a non positive determinant. So, the matrix given in (2.37) and also $Q$ cannot be positive definite in a general case ($M \neq 0$).

However, one can also examine the special case when $M = 0$ ($C_r = [V \ 0]$). In this case $Q_1,11 = 0$, $Q_1,22 = Q_1,22 > 0$, $Q_2 = V^T Q_2 V > 0$ and $Q_1 = Q_1,22 = Q,1,22 > 0$. Finally, the expression for $x_k^T Q x_k$ in (2.37) can be simplified to:

$$x_k^T Q x_k = \left[ x_k^T \ x_k^II^T \right] \cdot \begin{bmatrix} Q_2 & 0 \\ 0 & Q_1,22 \end{bmatrix} \cdot \begin{bmatrix} x_k^I \\ x_k^II \end{bmatrix}$$

This contains a positive definite matrix which is equal with $Q$ and so, guarantees DARE solvability. Note that, here $\overline{Q}_2$ weights the states considered in $y_k^r$ (see the structure of $C_r$ above) and $Q_1,22$ weights the states unaffected by $\overline{Q}_2$.

The results of this examination can be summarized in the following theorem:

**Theorem 1 (Sufficient weighting for DARE solvability)** Consider a discrete time, linear, time invariant system with state equation: $x_{k+1} = Ax_k + Bu_k$, and controlled output $y_k^r = C_r x_k$ $(x_k \in \mathbb{R}^n, \ u_k \in \mathbb{R}^m, \ y_k^r \in \mathbb{R}^r, \ A \in \mathbb{R}^{n \times n}, \ B \in \mathbb{R}^{n \times m}, \ C_r \in \mathbb{R}^{r \times n})$. Assume that, the pair $(A, B)$ is stabilizable and all the $n - r$ linearly dependent columns of $C_r$ are zero (this means that, $C_r$ can be transformed into the form given in (2.25) with $M = 0$). If all the above conditions are satisfied, the following weighting strategy guarantees DARE solvability (irrespective of the structure of $A$) and makes it possible to weight the states considered and not considered in the controlled output separately:

State weighting matrix:

$$Q = C_r^T Q_1 C_r + C_r^T Q_2 C_r$$
where:

\[ Q_2 > 0 \]
\[ Q_1 > 0 \text{ diagonal} \]
\[ \overline{C} = I - C_r^T (C_r C_r^T)^{-1} C_r \]

Input weighting matrix:

\[ R > 0 \]

Remark: The system matrices \( A, B, C_r \) do not have to be transformed to achieve \( C_r = [V \ 0] \) (see (2.25)) but in the case they are transformed, the \( Q_1 \) matrix can be partitioned into \( Q_{1,11} \) and \( Q_{1,22} \) as in (2.36). In this case, \( Q_{1,11} = 0 \) can be set, because \( \overline{C} = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \) and so, \( Q_{1,11} \) does not have effect on \( Q \). Note that, in this case \( Q_{1,22} \) weights the states not considered in the controlled output. The transformation means the reordering of system states, so in the untransformed case this concept can also be used by nulling out the elements of \( Q_1 \) where \( C \) has zero diagonal entries.

In the following, the examination will be extended to non diagonal \( Q_1 \) matrices.

\[ Q_1 = \begin{bmatrix} Q_{1,11} & Q_{1,12} \\ Q_{1,12}^T & Q_{1,22} \end{bmatrix} > 0 \]

(2.40)

The positive definiteness of \( Q_1 \) implies constraints on its blocks considering block matrix determinant and Sylvester criterion (a matrix is positive definite if all the determinants of its leading principal submatrices are positive).

\[
\begin{align*}
\det(Q_{1,11}) > 0 & \implies Q_{1,11} > 0 \\
\det(Q_1) = \left(\det(Q_{1,11})\right) \det(Q_{1,22} - Q_{1,12}^{-1} Q_{1,11} Q_{1,12}) & > 0 \\
Q_{1,22} - Q_{1,12}^{-1} Q_{1,11} Q_{1,12} & > 0 \implies Q_{1,22} > 0
\end{align*}
\]

(2.41)

Consider now the construction of \( Q \) in a similar way as in (2.37). The same steps as presented in Appendix 8.6 lead to:

\[
\begin{align*}
\overline{Q}_{1,11} & = M_3^{-T} M^T Q_{1,11} M M_3^{-1} \geq 0 \\
\overline{Q}_{1,22} & = M_3^{-T} Q_{1,22} M_3^{-1} > 0 \\
\overline{Q}_2 & = V^T Q_2 V > 0 \\
\overline{Q}_1 & = \overline{Q}_{1,11} + \overline{Q}_{1,22} > 0 \\
\overline{Q}_{1,12} & = M^T M_2^{-T} Q_{1,12} M_3^{-1} + M_3^{-T} Q_{1,12}^T M M_3^{-1}
\end{align*}
\]

(2.42)

The matrix in (2.43) contains the same first 3x3 block as in (2.37) which violates the Sylvester criterion. This means that in the general case \( M \neq 0 \) even non diagonal \( Q_1 \) and \( Q_2 \) do not guarantee DARE solvability. However, in the special case when \( M = 0 \), the same expression as in (2.39) results and this guarantees the solvability.
\[ x_k^T Q x_k = \begin{bmatrix} x_k^T & x_k^T M & -x_k^H T & x_k^H T M & -x_k^T \end{bmatrix} \cdot \]
\[
\begin{bmatrix}
Q_2 & 0 & 0 & Q_2 & M_{2}^{-1}M_{2}Q_{1,12}M_{3}^{-1} \\
0 & Q_1 & Q_1 & 0 & 0 \\
0 & Q_1 & Q_1 & M_{3}^{-1}Q_{1,12}M_{2}^{-1} & Q_{1,12} \\
M_{3}^{-T}M_{3}Q_{1,12}M_{3}^{-1} & 0 & Q_{1,12} & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}.
\]

Remark: In the derived multi-step LQ optimal tracking solution the pre-stabilization of the system solves the possible problems with DARE solvability because the pre-stabilized \( \Phi \) system matrix will not have eigenvalues on the unit circle. However, \( Q_1 \) can be further used to weight the dynamics of states unaffected by the controlled output weighting \( Q_2 \).

### 2.4 The properties of the infinite horizon LQ optimal tracker

Considering the derived tracking solution, the following properties can be stated:

1. It satisfies the separation principle both for constant and time-varying references.
2. It does not require anti-windup compensation because of static gain memoryless control (no internal states of controller).
3. It guarantees asymptotic stability, zero steady-state tracking error, finite LQ functional value (on infinite horizon) and so, LQ optimality for constant, finite references.
4. It is sub-optimal, BIBO and so \( l_p \) stable for time-varying references. Sub-optimal here means that the infinite horizon LQ optimal tracking problem was only approximately solved and so, optimality was lost for these reference signal class as explained above during the derivation.
5. It guarantees finite cost functional value (on infinite horizon) for \( l_1/l_2 \) references.

#### 2.4.1 The satisfaction of the separation principle

This can be proven using [BKB08b], (2.1) and (2.23):

The equations of the DT actual state estimator are the following (it can be either deterministic or stochastic estimator):

...
\[
\dot{x}_{k+1} = A\hat{x}_{k-1} + Bu_{k-1}
\]

\[
x^e_{k+1} = \hat{x}_{k+1} - x_{k+1} = (I - L_oC)Ax^e_k
\]

(2.44)

Here \(\hat{x}_k\) is the estimated state and \(y_k\) is the measured output which is usually different from the controlled one. The state dynamic equations of the augmented system are as follows:

\[
\begin{bmatrix}
  x_{k+1} \\
  x^e_{k+1}
\end{bmatrix}
= \begin{bmatrix}
  A^a & -BK_x \\
  0 & (I - L_o C) A
\end{bmatrix}
\begin{bmatrix}
  x_k \\
  x^e_k
\end{bmatrix}
+ \begin{bmatrix}
  B\bar{K}S_1 r_{k+1} + BK_{S_2} r_k + BK_{r_\infty} r_\infty \\
  0
\end{bmatrix}
\]

(2.45)

From \(A^a\) the poles of the augmented system can be calculated as \(\det(zI - A^a) = \det(zI - A + BK_x)\det(zI - A + L_o CA) = 0\) and neither the reference signal, nor the system states affect the dynamic of the estimation error. So, the separation principle is satisfied.

### 2.4.2 No need for anti-windup compensation

Let’s consider the structure of the system input from (2.23) in the applicable form \(r_\infty = r_{k+1}\):

\[
\tilde{u}_k = -K_x \hat{x}_k + (\bar{K}_S_1 + K_{r_\infty}) r_{k+1} + K_{S_2} r_k
\]

(2.46)

In (2.46) only the actual value of estimated system state vector and the actual and previous reference signal values are applied. There is no integral state and so there is no need to anti-windup compensation, the resulting control input can be simply saturated.

Considering \(\bar{K}_S_1 = K_{S_1} - 2K_{S_2}\) and \(K_{r_\infty} = K_{S_2} - K_{S_1} + (K_{x2}(I - \Phi)^{-1}B + I)F^+\) from (2.23), \(\tilde{u}_k\) results as a PD-like control input:

\[
\tilde{u}_k = -K_x \hat{x}_k - K_{S_2} (r_{k+1} - r_k) + (K_{x2}(I - \Phi)^{-1}B + I)F^+ r_{k+1}
\]

(2.47)

The difference from conventional PD control is that it uses only the difference of the reference signal instead of the tracking error. This can give better noise attenuation properties because usually the reference signals are generated by the user in a noiseless way.
2.4.3 Asymptotic stability and zero steady state tracking error for constant, finite references

Consider (2.45) to derive the steady state of the system with constant \( r_{k+1} = r_k = r_\infty \):

\[
x_{k+1}^a = A^a x_k^a + B^a(k)
\]

\[
x_{\infty}^a = (A^a)^\infty x_0^a + (I - A^a)^{-1} B^a(\infty)
\]

(2.48)

The inverse \((I - A^a)^{-1}\) can be calculated applying the block matrix inversion lemma and considering (2.45) and \( \Phi_1 = A - BK_x \):

\[
I - A^a = \begin{bmatrix}
I - \Phi_1 & BK_x \\
0 & I - L
\end{bmatrix}
\]

\[
(I - A^a)^{-1} = \begin{bmatrix}
(I - \Phi_1)^{-1} & -(I - \Phi_1)^{-1} BK_x (I - L)^{-1} \\
0 & (I - L)^{-1}
\end{bmatrix}
\]

(2.49)

\[
B^a(\infty) = \begin{bmatrix}
BK_{S_1} r_\infty + BK_{S_2} r_\infty + BK_{r_\infty} r_\infty \\
0
\end{bmatrix}
\]

Multiplying \( B^a(\infty) \) with \((I - A^a)^{-1}\) results in the following steady state:

\[
\begin{bmatrix}
x_{\infty}^a \\
x_{\infty}^e
\end{bmatrix} = \begin{bmatrix}
(I - \Phi_1)^{-1} (BK_{S_1} r_\infty + BK_{S_2} r_\infty + BK_{r_\infty} r_\infty) \\
0
\end{bmatrix}
\]

(2.50)

(2.50) shows that the final state estimation error is zero and the final state depends only on \( r_\infty \) which is finite so, the asymptotic stability is guaranteed.

The zero steady state tracking error can be proven considering (2.1), (2.23) and (2.50):

\[
x_{\infty} = (I - \Phi_1)^{-1} B \\
\cdot (K_{S_1} - 2K_{S_2} + K_{S_2} + K_{S_2} - K_{S_1} + (K_{x_2} (I - \Phi)^{-1} B + I) F^+) r_\infty =
\]

\[
= (I - \Phi_1)^{-1} B (K_{x_2} (I - \Phi)^{-1} B + I) F^+ r_\infty =
\]

\[
= (I - \Phi_1)^{-1} (BK_{x_2} (I - \Phi)^{-1} + I) BF^+ r_\infty
\]

consider: \( I = (I - \Phi)(I - \Phi)^{-1} \) and \( I - \Phi + BK_{x_2} = I - \Phi_1 \)

\[
x_{\infty} = (I - \Phi_1)^{-1} (I - \Phi_1) (I - \Phi)^{-1} BF^+ r_\infty = (I - \Phi)^{-1} BF^+ r_\infty
\]

\[
e_{\infty} = y_{\infty} - r_\infty = C_r x_{\infty} - r_\infty
\]

\[
e_{\infty} = C_r (I - \Phi)^{-1} B(C_r (I - \Phi)^{-1} B)^+ r_\infty - r_\infty = 0
\]

2.4.4 Finite functional value and LQ optimality for infinite horizon with constant, finite references

It was pointed out that \( x_k \rightarrow x_\infty \) if \( k \rightarrow \infty \). This means that \( \Delta x_k \rightarrow 0 \). Examine now the functional given in (2.11). If \( r_{k+1} = r_k = r_\infty = \text{const} \ \forall k \geq 0 \) then \( \Delta r_k = 0 \ \forall k \),
\[ \Delta x_k = H \Delta r_k = 0 \text{ and } \Delta u_k = -K_x \Delta x_k \text{ (considering (2.22)). This way } \Delta u_k \text{ goes to zero as } \Delta x_k \to 0. \]  
In this case the functional has the following form:

\[ J(\Delta x, \Delta u) = \frac{1}{2} \sum_{k=0}^{\infty} (\Delta x_k^T Q \Delta x_k + \Delta u_k^T R \Delta u_k) \quad (2.52) \]

(2.52) is the functional of the well known infinite horizon LQ optimal regulator problem, which has an optimal solution and a finite functional value known as \( J_{opt} = \Delta x_0^T P_{\infty} \Delta x_0 \).

Here \( P_{\infty} \) is the solution of the well known steady state DARE. This way the derived solution is really LQ optimal for constant references.

### 2.4.5 BIBO and so \( l_p \) stability with \( l_p \) time-varying references

Considering (2.1) and (2.47) and \( z \) as the forward shift operator the input-output dynamics (transfer function matrices) of the system results as:

\[
\begin{align*}
\dot{x}_{k+1} &= A x_k + B \hat{u}_k \\
\hat{u}_k &= -K_x \hat{x}_k - K_{S_2} (r_{k+1} - r_k) + (K_{x_2} (I - \Phi)^{-1} B + I) F^+ r_{k+1} \\
\dot{\hat{x}}_k &= x_k + x^e_k \\
x_{k+1} &= A x_k - B K_x x_k - B K_x x^e_k - B K_{S_2} \Delta r_{k+1} + B K_r r_{k+1} \\
x_k &= -(zI - \Phi_1)^{-1} B K_x x^e_k - (zI - \Phi_1)^{-1} B K_{S_2} \Delta r_{k+1} + (zI - \Phi_1)^{-1} B K_r r_{k+1} \\
y^e_k &= C_r x_k = -C_r (zI - \Phi_1)^{-1} B K_x x^e_k - \\
&- C_r (zI - \Phi_1)^{-1} B K_{S_2} \Delta r_{k+1} + C_r (zI - \Phi_1)^{-1} B K_r r_{k+1} \\
y_k &= C x_k = -(zI - \Phi_1)^{-1} B K_x x^e_k - \\
&- C (zI - \Phi_1)^{-1} B K_{S_2} \Delta r_{k+1} + C (zI - \Phi_1)^{-1} B K_r r_{k+1}
\end{align*}
\[
(2.53)
\]

In (2.53) all the transfer function matrices from the reference signals to the outputs are stable because of the stability of \( \Phi_1 \). For a stable state estimator the \( x^e_k \) estimation error should be a bounded signal which means that the outputs will be bounded. This means the BIBO and so \( l_p \) stability of the closed loop system for any \( l_p \) time-varying reference signal.

### 2.4.6 Finite functional value (on infinite horizon) for \( l_1/l_2 \) references

The \( l_1 \) and \( l_2 \) norms for a vector sequence can be defined as follows:
\[ \|r_k\|_1 = \sum_{k=0}^{\infty} |r_k| \]
\[ \|r_k\|_2 = \sum_{k=0}^{\infty} |r_k|^2 \]

In the \(l_2\) norm the square root was omitted because it does not affect the finiteness of the norm. The existence of these norms requires that

\[ r_\infty = \lim_{k \to \infty} r_k = 0 \]

Considering this, one can have a class of \(l_1\) signals (a wide range) where the absolute value of \(r_k\) is bounded with an exponential function:

\[ |r_k| < A e^{-ak} \]  

The detailed derivation of this bound can be found in Appendix 8.7 together with the proof of that such an \(l_1\) signal is also an \(l_2\) one.

After defining the reference signal class, the value of the infinite horizon functional from (2.11) can be examined considering these type of references with \(r_\infty = 0\). \(r_\infty = 0\) means that \(u_\infty = 0\) (see (2.7)) and \(x_\infty = 0\) (see (2.8)). This way \(\Delta x_k = x_k\), \(\Delta \hat{x}_k = \hat{x}_k\), \(\Delta u_k = u_k\) and the infinite horizon functional from (2.11) can be formulated as:

\[ J(x, \hat{x}, u) = \frac{1}{2} \sum_{k=0}^{\infty} (x_k - \hat{x}_k)^T Q (x_k - \hat{x}_k) + u_k^T R u_k \]  

The system input from (2.23) can be reformulated as:

\[ u_k = -K x_k \hat{x}_k + \overline{K} S_1 r_{k+1} + K S_2 r_k = -K x_k \hat{x}_k - K x_k \hat{r}_k + \overline{K} S_1 r_{k+1} + K S_2 r_k = -K x_k \hat{x}_k + u_k^r \]  

An upper bound for the absolute value of the reference signal related part is derived in Appendix 8.8 as \(|u_k^r| < K U e^{-uk}\). Considering the state dynamics from (2.8) and the input from (2.57) the \(k\)th value of system state results as:

\[ x_1 = \Phi x_0 + B u_0 = (\Phi - B K_1 x_0 + B u_0^r) = \Phi_1 x_0 + B u_0^r \]
\[ x_2 = \Phi_1 x_1 + B u_1^r = \Phi_1^2 x_0 + \Phi_1 B u_0^r + B u_1^r \]
\[ x_3 = \Phi_1 x_2 + B u_2^r = \Phi_1^2 x_0 + \Phi_1^2 B u_0^r + \Phi_1 B u_1^r + B u_2^r \]
\[ \vdots \]
\[ x_k = \Phi_1^k x_0 + \sum_{l=0}^{k-1} (\Phi_1^l B u_{k-l-1}^r) = \Phi_1^k x_0 + R_k \]

The finiteness of the functional (2.56) is proven in Appendix 8.9 substituting (2.57), (2.58) and considering the stability of \(\Phi_1\) and \(l_1\) references.

In the next section the developed LQ optimal tracker will be compared to other methods considering the aerospace application example described in section 1.3.
2.5 Comparison with other methods in Matlab simulations

The above developed LQ optimal output tracking solution is compared to seven other control design methods assuming perfect knowledge of system model and system state vector without noise effects. The noise tolerance and robustness of the method will be examined in a separate chapter (chapter 5). In the following subsections the concurrent methods will be introduced shortly, together with the tuning for aircraft lateral model in (1.7) and then the comparison will be made. The parameters of the aircraft lateral model were the averages of the parameters of model 1 and model 2 from Table 1.1 in all test cases. The interconnection structures of each controlled system can be found in Appendix 8.10 including also the disturbance estimation and correction part which is used only in the next chapter. In the application examples the DT equivalent of the CT augmented system (1.7) is used assuming $G\alpha_c = 0 \ (d=0)$ to design the controllers. The discretization was done with $\Delta t = 0.04s$ sampling time, considering the 25Hz frequency of the control thread onboard the Ultrastick UAV in MPC555 microcontroller (for hardware system details see [70]). All the introduced methods will use the notations from (2.1) and are tuned applying trial and error by the author. The goals with controller tuning were the following:

1. To achieve the possible shortest settling time.
2. To avoid having oscillations in the system (try to avoid complex poles).
3. To have minimum overshoot.

There were no special specifications considered in the tuning. The rules of thumb for tuning were the following:

1. In the pre-stabilization designs (done by pole placement) the originally unstable system poles were moved inside the unit circle and the complex poles were modified to be purely real.
2. In the final pole placement designs real poles as close to the origin as possible were set to have as fast dynamics as possible (of course without oscillations and overshoot).
3. In the LQ optimal designs the initial weighting matrices were set using the method of inverse squares (considering the allowed tracking errors and control surface deflections). From this point the matrices were tuned to satisfy the above requirements by trial and error method.
4. In the $H_{\infty}$ control design cases the crossover frequency of the weighting functions was set based on the singular value plot of the open-loop system. The initial weights were set considering the inverse of the maximum allowable error. Then they were tuned to achieve feasibility of the $H_{\infty}$ problem and to achieve the best possible performance considering the requirements.
The considered initial weights in the LQ (and MPC) / ($\mathcal{H}_\infty$) tracking method were the following (for the $\mathcal{H}_\infty$ method the low frequency gain of the weighting functions was set to the given value):

- for $p$ roll rate: $1 / (1)$ which means about $60^\circ/s$ maximal roll rate
- for $r$ filtered yaw rate: $365 / (19)$ which means about $3^\circ/s$ maximum filtered yaw rate
- for roll tracking error $e_{\phi} = \phi - \phi_{ref}$: $3300 / (57)$ which means about $1^\circ$ maximum tracking error
- for $\int e_{\phi} dt$: $100 / (no~such~\mathcal{H}_\infty~weight)$ with no exact physical meaning because the value of the integral depends on the time horizon
- for $\delta_a$: $5 / (2.2)$ which means about $25^\circ$ maximum aileron deflection
- for $\delta_r$: $33 / (5.7)$ which means about $10^\circ$ maximum rudder deflection

All the tuning started with these values and was refined by trial and error.

These initial weights and rules are also used in the next chapter where the LQ minimax optimal control is compared to the same seven methods.

### 2.5.1 Feedforward control to track constant references ($FF$)

The method published in [9] was applied to solve the task. This method considers two feedforward components, one to set the steady state and another to set the steady nonzero input of the system to track a nonzero reference signal: $x_\infty = N_x r_\infty$ and $u_\infty = N_u r_\infty$. The control input is constructed as $\tilde{u}_k = -K(x_k - N_x r_k) + N_u r_k$. $r_k$ shows that set point changes are possible. Using this control input and the system dynamics, the following system of equations can be derived from the steady state equilibrium:

\[
x_{k+1} = Ax_k + B\tilde{u}_k = Ax_k - BK_x x_k + BK_x N_x r_k + BN_u r_k \\
y_\infty = C_r x_\infty = C_r N_x r_\infty \rightarrow C_r N_x = I \\
x_\infty = Ax_\infty - BK_x x_\infty + BK_x N_x r_\infty + BN_u r_\infty \\
N_x r_\infty = AN_x r_\infty - BK_x N_x r_\infty + BK_x N_x r_\infty + BN_u r_\infty \rightarrow (A - I)N_x r_\infty + BN_u r_\infty = 0 \\
\begin{bmatrix} A - I & B \\ C_r & 0 \end{bmatrix} \begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}
\]

Using the block matrix determinant lemma the determinant of $F$ results as $\det(F) = \det(A - I)\det(0 - C_r(A - I)^{-1}B)$. Its existence requires the nonsingularity of $A - I$ which means that $A$ should not have eigenvalues on the unit circle. If it has, a state feedback pre-compensator ($K_{x1}$) should be designed to first stabilize $A$:
\begin{equation}
\hat{u}_k = -K_{x1}x_k - K_{x2}(x_k - N_x r_k) + N_u r_k
\end{equation}
\begin{equation}
x_{k+1} = Ax_k + Bu_k = Ax_k - BK_{x1}x_k - BK_{x2}x_k + BK_{x2}N_x r_k + BN_u r_k
\end{equation}
\begin{equation}
x_{k+1} = \Phi x_k - BK_{x2}x_k + BK_{x2}N_x r_k + BN_u r_k
\end{equation}

The steps of the controller design are as follows:

1. Examine if $A$ has poles on the unit circle. Design a stabilizing state feedback $K_{x1}$ if required or take $\Phi = A$.

2. Consider the stabilized $\Phi$ and solve the constraint equation from (2.59) with $\Phi - I$ if $F$ is invertible. Note that for an exact solution $F$ should be square which means \( \dim(u) = \dim(y^r) \). If $F$ is not invertible the controller can not be designed. If \( \dim(u) < \dim(y^r) \) a least squares optimal solution can be obtained using the Moore-Penrose pseudoinverse but with tracking performance degradation.

3. Design another state feedback $K_{x2}$ for $\Phi$ to improve controller performance either with pole placement or LQ optimal regulator technique. Here, the pole placement was applied.

4. Construct the final control input in the form of:
\begin{equation}
\hat{u}_k = -K_{x1}x_k - K_{x2}x_k + K_{x2}N_x r_k + N_u r_k = -K_{x1}x_k + K_r r_k
\end{equation}

In the application example $Aa$ (the DT equivalent of $Aa^c$ from (1.7)) has a unit eigenvalue (see $p_0$ below) so, pre-stabilization with $K_{x1}$ is required. Unfortunately $F$ is rank deficient with $y^r_k = \begin{bmatrix} \phi \\ \tau \end{bmatrix}$. The solution was to remove $\tau$ from the tracking output and design $N_x$ and $N_u$ only for $\delta_a$. This way $F$ becomes quadratic and invertible. However, $K_{x2}$ is designed for the whole system and both inputs by applying pole placement. The poles of the augmented system $Aa$ are:

\begin{equation}
p_0 = \begin{bmatrix} 1 & 0.7412 & 0.6145 & 0.5488 & (0.5616 \pm 0.2291i) & \times 2 \end{bmatrix}
\end{equation}

The stabilizing poles were determined considering the given requirements:

\begin{equation}
p_1 = \begin{bmatrix} 0.98 & 0.7142 & 0.6145 & 0.6 & 0.6 & 0.5488 & 0.5616 & 0.5616 \end{bmatrix}
\end{equation}

The final controlled system poles were tuned to achieve the best tracking performance and hold control inputs between the limits ($\pm 25^\circ$ for aileron ($\pm 100\%$ deflection range), $\pm 10^\circ$ for rudder ($\pm 40\%$ deflection range)). Double poles are possible because the system has two inputs which can set two poles at the same place.

\begin{equation}
p_2 = \begin{bmatrix} 0.84 & 0.84 & 0.83 & 0.83 & 0.82 & 0.82 & 0.81 & 0.81 \end{bmatrix}
\end{equation}

The controlled system interconnection can be seen in figure 8.2
2.5.2 Integral feedforward control to track constant references ($\int FF$)

This method is also based on [9] and is very similar to the previous one. Here only the differences are described. The main difference is the augmentation of system state space with an integral state which is the integral of the tracking error. This integral term removes the need to apply $N_u$ for nonzero input but $N_x$ for steady state should be again used. The augmented system dynamics is as follows (assuming pre-stabilized $\Phi$):

\[
\begin{bmatrix}
    x_{k+1} \\
    x_I(k+1)
\end{bmatrix} =
\begin{bmatrix}
    \Phi & 0 \\
    \Delta t \cdot C_r & I
\end{bmatrix}
\begin{bmatrix}
    x_k \\
    x_I(k)
\end{bmatrix}
+ \begin{bmatrix}
    B \\
    0
\end{bmatrix} u_k
+ \begin{bmatrix}
    0 \\
    I \cdot \Delta t
\end{bmatrix} r_k
\]

The controller can be designed through the steps:

1. Examine if $A$ has poles on the unit circle. Design a stabilizing state feedback $K_{x1}$ if required or take $\Phi = A$.

2. Design $\bar{K}_x = \begin{bmatrix} K_{x2} & K_I \end{bmatrix}$ using the augmented system in (2.61) either with pole placement or LQ optimal regulator technique. Here, the pole placement was applied.

3. Solve the same constraint equation as in (2.59) but use only $N_x$.

4. Construct the final control input according to (2.61)

In the application example the same design steps were done as for the previous method ($p_0$ and $p_1$ are the same) except for the final poles which are different and the additional integral state is considered (of course the poles were tuned to achieve best tracking performance):

\[
p_2 = [0.9 \ 0.9 \ 0.89 \ 0.89 \ 0.88 \ 0.87 \ 0.87 \ 0.65]
\]

The controlled system interconnection can be seen in figure 8.3.

2.5.3 Simple LQ optimal control with tracking error feedback (Simple LQ)

If the controlled system has integral property the simple state feedback can be modified by using the state tracking error for feedback (only for the states which are integrated):

\[
\tilde{u}_k = -K_x(x_k - x_k^r)
\]

The controller gain $K_x$ can be designed either with pole placement or LQ regulator theory. Here, the latter is used. The controlled system interconnection can be seen in figure 8.4.

In the application example (for the system in (1.7)) only the $\phi$ state is modified with a nonzero reference value because it is integrated in the system dynamics. In the LQ
regulator design only $p$, $\phi$ and $\tau$ are weighted; $\phi$ and $\tau$ because of the tracking, $p$ to limit the dynamics of the roll response (roll rate limiter). The best weighting was:

$$Q \triangleq < 0.5, 0, 10, 10, 0, 0, 0 > \quad R \triangleq < 10, 20 >$$

Here, $<$ > denotes a diagonal matrix.

### 2.5.4 Simple PID control (**PID**)

The same PID roll angle tracking controller augmented with a yaw damper is used as in [70] because it was implemented and tested onboard the Ultrastick UAV which makes it possible to compare real flight test results. The roll angle tracking part:

$$e_\phi(k) = \phi_{ref}(k) - \phi_k$$
$$e_I(k) = e_I(k-1) + e_\phi(k)\Delta t \cdot AW$$
$$\delta_\phi(k) = P_\phi e_\phi(k) + I_\phi e_I(k) + D_\phi(-p(k))$$

Where $P_\phi = 0.3$, $I_\phi = D_\phi = 0.05$ and AW is an anti windup flag which can be 0 or 1. Here anti windup and saturation are applied in two cases ($\delta_{aMAX} = 25^\circ$):

1. $|I_\phi e_I(k)| > 0.2 \cdot \delta_{aMAX}$
2. $|\delta_\phi(k)| > \delta_{aMAX}$

The yaw damper part is the following including also the washout filter from (1.6):

$$\delta_r(k) = b\delta_r(k-1) + c\delta_r(k-2) + gK(-r(k-1) + ar(k-2))$$

$b = 1.207$, $c = -0.2369$, $g = 0.03$, $K = 0.7651$ and $a = 1$. Here $\delta_r(k)$ is saturated between $\pm 10^\circ$. The controlled system interconnection can be seen in figures 8.5 and 8.6.

### 2.5.5 LQ Servo control (**LQ Servo**)

This applies the same augmented system dynamics with tracking error integral as in (2.61), the gain vector $K_x = [K_x \quad K_I]$ is designed using LQ optimal regulator theory.

In the application example this controller also could not be designed both for $\phi$ and $\tau$ tracking, only for $\phi$ tracking. The system is augmented with the integral state. Only $p$, $r$ and the $\phi$ tracking error integral are weighted. The best weighting matrices are:

$$Q \triangleq < 1, 0, 2, 0, 0, 0, 120 > \quad R \triangleq < 1, 20 >$$

The controlled system interconnection can be seen in figure 8.7.

### 2.5.6 Model predictive control (**MPC**) with static gains (**MPC**)

The formulae published in [57] (pp. 74-81) are used considering static gain MPC control for LTI systems without constraints. This is the simplest possible MPC solution closest to the developed LQ and minimax controllers. Here, only the main steps of derivation are repeated, the details can be found in [57]. The functional to be minimized:
\[ J(k) = \sum_{i=1}^{H_p} \| \hat{z}(k+i|k) - r_{k+i} \|_Q^2 + \sum_{j=0}^{H_u-1} \| \Delta \hat{u}(k+j|k) \|_{R(j)}^2 \] (2.65)

Here, \( \hat{z}(k+i|k) \) is the predicted system output, \( \Delta \hat{u}(k+j|k) \) is the required control input, \( H_p \) is the prediction, \( H_u \) is the control horizon and \( Q(i) \geq 0 \) and \( R(j) > 0 \) are the tracking error and control input weighting matrices in the weighted two norm expressions \( \| \hat{z}(k+i|k) - r_{k+i} \|_Q^2 = (\hat{z}(k+i|k) - r_{k+i})^T Q(i) (\hat{z}(k+i|k) - r_{k+i}) \). In the following \( H_p = H_u = N, Q = \text{const} \) and \( R = \text{const} \) are assumed. So, the functional can be reformulated:

\[ J(k) = \| Z(k) - T(k) \|_Q^2 + \| \Delta U(k) \|_R^2 \]

\[ Z(k) = \begin{bmatrix} \hat{z}(k+1|k) \\ \vdots \\ \hat{z}(k+N|k) \end{bmatrix}, \quad T(k) = \begin{bmatrix} r_{k+1} \\ \vdots \\ r_{k+N} \end{bmatrix}, \quad \Delta U(k) = \begin{bmatrix} \Delta \hat{u}(k|k) \\ \vdots \\ \Delta \hat{u}(k+N-1|k) \end{bmatrix} \] (2.66)

\[ \bar{Q} = \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix}, \quad \bar{R} = \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \]

Considering the system dynamics and neglecting the notation \( |k \) one gets:

\[
\begin{align*}
\dot{z}_{k+1} &= C_r \dot{x}_{k+1} + C_r A x_k + C_r B \Delta \hat{u}_k + C_r B u_{k-1} \\
\dot{z}_{k+2} &= C_r A^2 x_k + C_r A B \Delta \hat{u}_k + C_r A B u_{k-1} + C_r B \Delta \hat{u}_{k+1} + C_r B \Delta \hat{u}_k + C_r B u_{k-1} \\
& \vdots \\
\dot{z}_{k+N} &= C_r A^N x_k + \sum_{j=k}^{k+N-1} \left\{ C_r \left( \sum_{l=0}^{k+N-1-j} A^l \right) B \right\} \Delta u_j + C_r \left( \sum_{l=0}^{N-1} A^l \right) B u_{k-1} 
\end{align*}
\] (2.67)

From (2.67) the following system of equations can be formulated:

\[ Z(k) = \Upsilon(k)x_k + \Theta(k) \Delta U(k) + \Gamma(k)u_{k-1} \]

\[ \Upsilon(k)(i) = C_r A^i, \quad \Theta(k)(i,j) = C_r \left( \sum_{l=0}^{k+i-1-j} A^l \right) B, \quad \Gamma(k)(i) = C_r \left( \sum_{l=0}^{i-1} A^l \right) B \] (2.68)

\[ i = 1 : N, \quad j = 0 : N - 1 \]

The goal is to achieve \( Z(k) = T(k) \) the optimal control input sequence can be designed according to [57] as:

\[
\begin{align*}
(\Theta(k)^T \bar{Q} \Theta(k) + \bar{R}) \Delta U(k) &= \Theta(k)^T \bar{Q} \varepsilon(k) \\
\Delta U(k) &= \left( \Theta(k)^T \bar{Q} \Theta(k) + \bar{R} \right)^{-1} \Theta(k)^T \bar{Q} \varepsilon(k) \\
\varepsilon(k) &= T(k) - \Upsilon(k)x_k - \Gamma(k)u_{k-1} 
\end{align*}
\] (2.69)
In (2.69) \( M \) is a \( (N \cdot \text{dim}(u)) \times (N \cdot \text{dim}(r)) \) from which the first \( \text{dim}(u) \times (N \cdot \text{dim}(r)) \) block \((M_1)\) should be used in actual control (which gives \( \Delta u_k \)):

\[
\begin{align*}
\Delta u_k &= u_k - u_{k-1} + M_1 (T(k) - \Upsilon(k)x_k - \Gamma(k)u_{k-1}) \\
&= -M_1 \Upsilon(k)x_k + (I - M_1 \Gamma(k))u_{k-1} + M_1 T(k)
\end{align*}
\]

(2.70)

Partitioning \( K_r \), considering the linear extrapolation of \( r_{k+2} = 2r_{k+1} - r_k \) (to have a similar solution as the derived LQ tracker) and assuming \( r_{k+3} = \ldots = r_{k+N} = r_{k+2} \) one gets the following expression for the reference related part of the input:

\[
K_r = \begin{bmatrix}
K_{r1} & K_{r2} & \cdots & K_{rN}
\end{bmatrix}
\]

\[
K_r T(k) = (K_{r1} + 2K_{r2} + \ldots + 2K_{rN}) r_{k+1} - (K_{r2} + \ldots + K_{rN}) r_k =
\]

\[
\begin{align*}
&= K_{r1} r_{k+1} - K_{r2} r_k \\
&= -K_{r1} x_k + K_{ru} u_{k-1} + K_{r1} r_{k+1} - K_{r2} r_k
\end{align*}
\]

(2.71)

The control input in (2.71) is structurally similar to the inputs in the derived LQ and LQ minimax optimal trackers (2.23) and (3.17) with state feedback and reference feedforward.

In the application example a two step horizon \((N=2)\) was considered. Because of the structure of the MPC problem \( p, \phi \) and \( \tau \) are all considered as outputs (in this order), the \( C_r \) matrix was constructed accordingly. The final weighting matrices with best tracking performance resulted as:

\[
Q = < 10, 230, 500 > \quad R = < 5000, 5000 >
\]

The controlled system interconnection can be seen in figure 8.8.

2.5.7 \( H_{\infty} \) optimal control (\( H_{\infty} \))

The well known (85, 86) central \( H_{\infty} \) control solution is applied here using the discrete time LMI solver. The interconnection structure is shown in figure 2.2. Nonzero \( \phi \) and zero filtered yaw rate \((\bar{\tau})\) references are considered in this design.

![Figure 2.2: System interconnection structure for \( H_{\infty} \) tracking controller design](image)
The weighting functions are given in CT and then transformed into DT. The bandwidth of the augmented system (from (1.7) denoted by $G_0$) can be read from its singular value plot. It resulted as about $2 \text{rad/s}$ and the frequency shape of the weighting functions was constructed accordingly.

The ideal transfer function $T_{ry}$ and weighting functions are as follows ($W_{e1}$ for $\phi$ tracking error, $W_{e2}$ for $\tau$ tracking error, $W_p$ for limiting roll rate, $W_u$ to bound control inputs, $W_r$ to give the magnitude of $\phi$ references and $W_n$ to give sensor noise magnitude):

$$
T_{ry} = \frac{1}{0.8s + 1} \\
W_{e1} = \frac{100(0.05s + 1)}{5s + 1} \\
W_{e2} = \frac{10(0.05s + 1)}{5s + 1} \\
W_p = \frac{1(0.05s + 1)}{5s + 1} \\
W_u = < 4, 12 > \\
W_r = 0.1 \\
W_n = < 0.02 \times 2, 0.04, 10^{-6} \times 5 >
$$

The controlled system interconnection can be seen in figure 8.9.

### 2.5.8 LQ tracker solution developed in this chapter (LQ tracker)

In this solution $\phi$ and $\tau$ are both considered as tracking outputs. The design steps described in the development are evaluated. The poles of the augmented system and the (pre-)stabilizing poles are the same as in the FF case. $C_r$ in (1.7) shows that $\phi$, $\tau$ and $r$ (through the washout filter) are affected by $Q_2$. So $Q_1$ can be applied to weight $p$ and the states of the Padé delay model. The delay model should be free, so only $p$ is weighted by $Q_1$ to provide limitation of roll rate. The final best weights of the control are:

$$
Q_1 = < 100, 0 \times 7 > \\
Q_2 = < 2000, 2 > \\
R = < 5 \times 10^3, 5 \times 10^4 >
$$

The closed loop poles are:

$$
p_2 = \begin{bmatrix} 0.1856 & 0.536 & 0.5874 \pm 0.0158i & 0.7279 & 0.8432 & 0.7304 \pm 0.2567i \end{bmatrix}
$$

The controlled system interconnection can be seen at the end of the previous section in figure 2.1.

### 2.5.9 Comparison of roll doublet tracking control results

All the methods were tested by tracking a roll doublet reference signal and holding the filtered high frequency yaw rate around zero in Matlab simulation. The roll doublet reference can be seen in figure 2.3 together with the roll ($d_L$) and yaw ($d_N$) torque disturbances applied in minimax control tests (in the next chapter).

The applied disturbances can be described with the following functions:

$$
d_L = 0.05 \cdot 1(t - 20) + 0.2 \cdot 1(t - 32) - 0.1 \cdot 1(t - 45) \\
d_N = 0.01 \cdot 1(t - 20) + 0.02 \cdot 1(t - 32) - 0.09 \cdot 1(t - 45)
$$
where $1(t)$ is the unit step function. The time transients were selected to have disturbance changes at first in steady state (20s) after the roll transient, then to have the roll reference change during a transient because of the disturbances (32s) and then to have again disturbance changes in steady state (45s).

The numerical results are summarized in table 2.1 by the following parameters: $T_{s1}$ and $T_{s2}$ are the settling times from $0^\circ$ to $20^\circ$ and from $20^\circ$ to $-20^\circ$, MEAN $|\phi_e|$ is the mean value of absolute roll angle tracking error, MEAN $|\dot{\tau}|$ is the mean absolute high-frequency yaw rate (the mean absolute error relative to the zero reference), and $\|\cdot\|_2$ are the two norms (energy) of the given signals. All the parameters are calculated from the whole 50 second simulation with the roll doublet reference. The settling times are defined as the time required until the system reaches (and then does not leave) the $0.95 \cdot y_{ref}$ range after a change in the reference signal. Here, $y_{ref}$ is the new constant value after a change of the piecewise constant reference. This definition is used for settling time throughout the whole thesis work.

Table 2.1: LQ Simulink test results

| Ctrl        | $T_{s1}$ [s] | $T_{s2}$ [s] | MEAN $|\phi_e|$ [°] | MEAN $|\dot{\tau}|$ [°/s] | $\|p\|_2$ [°/s] | $\|\delta_a\|_2$ [°] | $\|\delta_r\|_2$ [°] |
|-------------|--------------|--------------|----------------------|-----------------|------------------|-----------------|------------------|
| PID         | 4.44         | 7.88         | 1.575                | 0.149           | 389.6            | 43.46           | 1.694            |
| FF          | 1.56         | 1.68         | 1.66                 | 0.015           | 224              | 22.33           | 8.56             |
| $\int$ FF   | 1.44         | 1.52         | 1.71                 | 0.456           | 241              | 79.06           | 222.47           |
| Simple LQ   | 0.72         | 0.92         | 0.78                 | 0.119           | 381.4            | 43.937          | 1.934            |
| LQ Servo    | 1.08         | 1.24         | 1.2                  | 0.128           | 329.6            | 36.7            | 2.77             |
| MPC         | 1.36         | 2.08         | 1.65                 | 0.057           | 239.96           | 25.34           | 4.936            |
| $\mathcal{H}_\infty$ | 2.36         | 2.88         | 1.35                 | 0.14            | 338.82           | 39.12           | 3.6              |
| LQ tracker  | 0.56         | 0.76         | 0.73                 | 0.116           | 395.7            | 46.57           | 15.08            |
From Table 2.1 the following results can be read: Regarding the settling times the LQ tracker developed here is the best controller. The Simple LQ solution is very close to it, the other methods are farther. The mean \( \phi \) tracking error is the smallest again with LQ tracker and second smallest with Simple LQ. The mean filtered yaw rate error is the smallest with FF second smallest with MPC and third smallest with LQ tracker. Considering the roll dynamics (the energy of \( p \)) the LQ tracker is the worst solution with the largest energy consumption and it is almost the worst regarding \( \delta_\alpha \) and \( \delta_r \) energy also. This clearly shows that the developed tracker has superior tracking performance at the cost of larger control and system energy. But the inputs do not saturate with this tracker (see figures 2.10, 2.11, 2.12, 2.13) so it uses the system inside the allowable limits. The tracking performance is really good, the control inputs are between the saturation limits so, the method is well applicable.

Table 2.2: State dimensions of the solutions

<table>
<thead>
<tr>
<th></th>
<th>PID</th>
<th>FF</th>
<th>( \int ) FF</th>
<th>Simple LQ</th>
<th>LQ Servo</th>
<th>MPC</th>
<th>( \mathcal{H}_\infty )</th>
<th>LQ tracker</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
<td>5</td>
<td>7/6</td>
<td>5</td>
<td>7/6</td>
<td>10</td>
<td>17/9</td>
<td>7</td>
</tr>
</tbody>
</table>

Considering the required state dimensions of the tested control solutions table 2.2 summarizes the results (including the sates of Padé and washout filters applied in the control). 7/6 means that if also \( \bar{r} \) is tracked (as with LQ tracker) the solution will have 7 states, otherwise 6. The LQ tracker has larger state dimension then PID, FF and Simple LQ methods, but smaller or equal dimension as \( \int \) FF, LQ Servo, MPC and \( \mathcal{H}_\infty \). The \( \mathcal{H}_\infty \) controller has originally 12 states which can be reduced to 4 without significant performance degradation. That’s why it has 17 / 9 states considering Padé and washout filters. As a summary, it can be stated that the LQ tracker has equal state dimension as the integral tracking (\( \int \) FF and LQ Servo) solutions and smaller state dimension as the (MPC and \( \mathcal{H}_\infty \)) ones. However, it outperforms these solutions, so if it has enough robustness, it can be successfully applied instead of these methods.

The numerical results given in Table 2.1 are visualized in the following figures (from 2.4 to 2.13) showing the tracking of the first section of doublet reference to make figures readable (the first transients are further enlarged in Appendix 8.12). The controller works in a similar way on the other sections. The eight methods from Table 2.1 are grouped into two figures, the first part with FF, \( \int \) FF, Simple LQ, PID and LQ tracker and the second part with LQ Servo, MPC, \( \mathcal{H}_\infty \) and LQ tracker. Figure 2.5 also includes the result of LQ tracker simulation with \( d_L \) and \( d_N \) disturbances (LQ dist). This shows that the LQ tracker is unable to compensate disturbances and this arise the need to develop the minimax tracker solution with disturbance estimation and compensation in the next chapter.
Figure 2.4: LQ roll angle tracking, part 1

Figure 2.5: LQ roll angle tracking, part 2

Figure 2.6: LQ yaw rate tracking, part 1

Figure 2.7: LQ yaw rate tracking, part 2

Figure 2.8: LQ roll rate, part 1

Figure 2.9: LQ roll rate, part 2
2.6 Summary

In this chapter at first, the solution of the finite horizon, LQ optimal tracking problem for LTI systems is given in a formally similar way as in [53] but giving deeper insight into the reference signal effects. Then the infinite horizon, LQ optimal, causal tracking problem is solved, giving an optimal result for constant, and a sub-optimal for time-varying references. The solvability conditions of the resulting DARE are examined in detail. After stating and proving the properties of the derived infinite horizon solution it is compared to seven other methods in Matlab simulations considering the same controlled system.
Chapter 3

Infinite horizon LQ optimal minimax tracking control

The drawback of simple LQ optimal tracking control (examined in the previous chapter) is the intolerance of deterministic disturbances (see figure 2.5). For example in an UAV, engine reaction torque and the constant part of wind give such (low-frequency, deterministic) disturbances which can make the achievement of good tracking results challenging.

$H_\infty$ and minimax methods are applicable to attenuate deterministic disturbances, but $H_\infty$ method has the drawbacks listed in the motivation part. Therefore minimax method is selected to solve disturbance attenuation.

This method is capable to consider the effect of system disturbances in the LQ functional and design an optimal controller (see [49]). However, if one cancels most of the disturbances before applying the minimax controller, the controller load can be significantly decreased. This idea is applied in [13]. Applying the idea of disturbance estimation and cancellation published in [9], firstly, a least squares optimal disturbance cancellation input can be used. Then the minimax controller should be designed for the system modified by the disturbance cancellation input.

In this chapter the effects of $w_k$ and $v_k$ are assumed to be considered in the state estimation. This way the estimated $x_k$ state vector is assumed to be available for feedback and the considered system is as follows:

\[
\begin{align*}
    x_{k+1} &= Ax_k + B\tilde{u}_k + Gd_k \\
    y^r_k &= C_r x_k \\
    y_k &= C x_k
\end{align*}
\]  

(3.1)

Here $y^r_k$ is the output which should track the reference signal $r_k$, the measured outputs ($y_k$) of the system can be different from it. It is assumed that, the pair $(A, B)$ is stabilizable.

As a summary, the goal of this part is to extend the results of the previous part to the minimax case considering the disturbance compensated system, but at first a review of the unknown input (disturbance) estimation literature is given.

3.1 Unknown input and state estimation

Unknown input and state estimation is a widely researched topic since the 1970s. Part of its literature was reviewed by the author ([18], [19], [20], [27], [31], [32], [38], [43], [44], [41], [42].
These mostly include methods for state estimation in the presence of unknown inputs. Unknown input estimation is covered only by [27], [31], [32], [38], [84].

The first article [27] deals with the problem of both unknown input and state estimation for DT, deterministic, time-varying systems with unknown input direct feedthrough. They relax the usual solvability rank condition (applied in most of the publications) but their method seriously delays the estimated states and inputs. Therefore it is not applicable in real time solutions.

A combined state and unknown input estimator for DT, time-varying systems with unknown input direct feedthrough is published in [84]. The authors use $H_\infty$ hybrid estimation. Unfortunately, their solution works only if unknown input direct feedthrough is present in the system, otherwise, only the states can be estimated. Therefore it is not applicable for the considered system class (see (3.1)).

An extension and summary of the methods derived in [18], [38] and [46] can be found in [31]. It proposes an unbiased, minimum variance state and unknown input estimator without unknown input direct feedthrough. So, this method is applicable for our system (3.1) and will be implemented during the system tests.

A least squares (LS) optimal disturbance cancellation input ([9]) can be designed based on the estimated disturbance output of this filter.

3.2 The infinite horizon, discrete time LQ optimal minimax tracker

The goal is to track a prescribed constant or time-varying reference signal with maximum disturbance attenuation (minimum tracking error). The developed multi-step solution is similar to the method applied in [13]. The unknown disturbances are estimated with a filter (see [31]) and so, their effect on the system states can be decreased in a LS optimal manner. Usually this does not mean the full cancellation of the disturbances so a residual disturbance occurs. Its effect can be attenuated with an LQ minimax tracking controller design. This way the designed controller should attenuate a much smaller disturbance then the original and so its performance can be increased. A key point of the design was to guarantee LQ optimality and zero steady state tracking error for constant reference signal and disturbances. The steps of the solution are as follows:

1. Design a stabilizing state feedback control input for system (3.1).

2. Design the optimal state and disturbance estimator for the stabilized system (the method described in [18] and [31] is used for LTI systems and can be applied only for stable systems).

3. Construct the system input which cancels the disturbance effects in a LS optimal way.

4. Design another control input, which guarantees zero steady state tracking error in case of constant reference and disturbance signals.
5. Center the original system dynamics (constructed in step 1) with the steady state equilibrium point achieved in the previous step, and design an LQ optimal minimax tracker for this centered dynamics.

3.2.1 Design of a stabilizing state feedback controller for \((A, B)\)

This can be solved either with pole placement or with LQ optimal regulator design. The resulting system equations can be written as follows (considering additional input to guarantee tracking):

\[
x_{k+1} = Ax_k - BK_1 x_k + Bu_k + Gd_k; \quad \ddot{u}_k = -K_1 x_k + u_k
\]

\[
x_{k+1} = \Phi x_k + Bu_k + Gd_k
\]

\[
y^r_k = C_r x_k \quad \& \quad y_k = C x_k
\]

3.2.2 Design an optimal state and disturbance estimator for \((\Phi, C, G)\)

This can be solved applying the results of [18] and [31] for LTI systems. The disturbance and state estimation equations are as follows:

\[
\begin{align*}
time \ update: \quad \bar{x}_k &= \Phi \hat{x}_{k-1} + Bu_{k-1} \\
measurement \ updates: \quad \hat{d}_{k-1} &= M (y_k - C \bar{x}_k) \\
\hat{d}_k &= A^d_{\hat{d}_{k-1}} \hat{d}_{k-1} \\
\hat{x}_k &= \bar{x}_{k-1} + K (y_k - C \bar{x}_k)
\end{align*}
\]

Here the estimation of the disturbance does not need any information about its dynamics, but if its dynamics is known, \(\hat{d}_k\) can be used in control instead of \(\hat{d}_{k-1}\) and so, better results can be achieved. In the following steps \(\hat{d}_k \approx \hat{d}_{k-1}\) will be assumed which means constant or slowly varying disturbances.

Note that if the original system with matrix \(A\) is stable it is better to design the disturbance estimator for that system.

3.2.3 LS optimal disturbance cancellation with the control input

The task is to find a control input component which cancels most of the disturbances using their estimated value. The equation to be solved and its LS optimal solution is as follows:

\[
Bu_k = -G \hat{d}_k
\]

\[
u_k = -B^+ G \hat{d}_k
\]
The equation has an exact solution if \( G = B \) which means that the disturbances affect the system in a similar way as the inputs (this case will be examined later in detail). This way the \( u_k \) input from (3.2) can be redefined:

\[
\begin{align*}
    u_k &= \hat{u}_k - B^+ G \hat{d}_k 
\end{align*}
\]

### 3.2.4 Determining the solution of the zero steady state tracking error problem considering constant reference and disturbance

The equation to be solved can be constructed considering (3.2) and (3.5).

\[
\begin{align*}
    x_\infty &= \Phi x_\infty + Bu_\infty + BB^+ \hat{G} \hat{d}_\infty \quad \hat{d}_\infty = d_\infty \\
    x_\infty &= \Phi x_\infty + Bu_\infty + (I - BB^+) \hat{G} \hat{d}_\infty \\
    x_\infty &= (I - \Phi)^{-1} Bu_\infty + (I - \Phi)^{-1} (I - BB^+) \hat{G} \hat{d}_\infty \\
    y_\infty &= C_r x_\infty = C_r (I - \Phi)^{-1} B \hat{u}_\infty + C_r (I - \Phi)^{-1} (I - BB^+) \hat{G} \hat{d}_\infty = r_\infty \\
    F \hat{u}_\infty &= r_\infty - C_r (I - \Phi)^{-1} (I - BB^+) \hat{G} \hat{d}_\infty \\
    \hat{u}_\infty &= F^+ r_\infty - F^+ C_r (I - \Phi)^{-1} (I - BB^+) \hat{G} \hat{d}_\infty \\
\end{align*}
\]

Here the existence of \((I - \Phi)^{-1}\) requires \( \Phi \) not to have eigenvalues on the unit circle. It was designed to be a stable system matrix so, this requirement is satisfied. \( F \) is an \( r \times m \) matrix and its pseudoinverse (or inverse if \( r = m \)) exists if \( C_r (I - \Phi)^{-1} B \) has full row rank.

### 3.2.5 Derivation of the LQ optimal finite horizon solution for the centered output tracking minimax problem

The required steady state input to track a constant reference signal can be calculated using (3.6). However, the control of the transient from initial state to steady state should be considered. This can be designed together with the solution of cases with time-varying references in a unified framework as follows.

The modified state dynamic equation from (3.2) and (3.5) and the steady state system equation are:

\[
\begin{align*}
    x_{k+1} &= \Phi x_k + Bu_k + Gd_k - BB^+ \hat{G} \hat{d}_k \\
    x_\infty &= \Phi x_\infty + Bu_\infty + Gd_\infty - BB^+ \hat{G} \hat{d}_\infty 
\end{align*}
\]

The equations in (3.7) can be subtracted from each other, giving another modified state equation:
\[ x_{k+1} - x_{\infty} = \Phi (x_k - x_{\infty}) + B \left( \hat{u}_k - \hat{u}_{\infty} \right) + G (d_k - d_{\infty}) - BB^+G (\hat{d}_k - \hat{d}_{\infty}) \]

\[ \Delta x_{k+1} = \Phi \Delta x_k + B \Delta \hat{u}_k + G \Delta d_k - BB^+G \Delta \hat{d}_k \]

\[ \Delta x_{k+1} = \Phi \Delta x_k + B \Delta \hat{u}_k + \left[ \frac{G - BB^+G}{B_d} \right] \Delta \hat{d}_k \]

(3.8)

The last equation in (3.8) gives a disturbed system dynamics around the steady state. This equation together with the centered reference signal \( \Delta r_k = r_k - r_{\infty} \) can be used to form an LQ optimal minimax tracking problem for the transient (in case of constant references) or for the case with time-varying references. At first, the finite horizon solution should be derived considering the following functional:

\[ J \left( \Delta \tilde{y}, \Delta e, \Delta \hat{u}, \Delta \tilde{d}_k \right) = \]

\[ = \frac{1}{2} \sum_{k=0}^{N-1} \left( \Delta \tilde{y}_k^T Q_1 \Delta \tilde{y}_k + \Delta e_k^T Q_2 \Delta e_k + \Delta \hat{u}_k^T R_u \Delta \hat{u}_k - \gamma^2 \Delta \tilde{d}_k^T R_d \Delta \tilde{d}_k \right) + \]

\[ + \Delta \tilde{y}_N^T Q_1 \Delta \tilde{y}_N + \Delta e_N^T Q_2 \Delta e_N \]

where :

\[ \Delta e_k = y_k^r - \Delta r_k = C_r \Delta x_k - \Delta r_k \]

\[ \Delta \tilde{y}_k = \mathcal{C} \Delta x_k = \left( I - C_r^T (C_r C_r^T)^{-1} C_r \right) \Delta x_k \]

(3.9)

This functional can be rewritten using \( \Delta \tilde{x}_k = C_r^T (C_r C_r^T)^{-1} \Delta r_k = H \Delta r_k \) in a similar way as in chapter 2.

\[ J \left( \Delta x, \Delta \tilde{x}, \Delta \hat{u}, \Delta \tilde{d}_k \right) = \]

\[ = \frac{1}{2} \sum_{k=0}^{N-1} \left( \Delta x_k - \Delta \tilde{x}_k \right)^T Q \left( \Delta x_k - \Delta \tilde{x}_k \right) + \Delta \hat{u}_k^T R_u \Delta \hat{u}_k - \gamma^2 \Delta \tilde{d}_k^T R_d \Delta \tilde{d}_k \]

\[ + \left( \Delta x_N - \Delta \tilde{x}_N \right)^T Q \left( \Delta x_N - \Delta \tilde{x}_N \right) \]

where :

\[ Q = \mathcal{C}^T Q_1 \mathcal{C} + C_r^T Q_2 C_r \]

(3.10)

The stated optimization problem in (3.10) can be solved applying Lagrange multiplier method. The costate variable results as follows (for details see Appendix 8.13):

\[ \lambda_k = P_k \Delta x_k + S_k \Delta \tilde{x}_{k+1} - Q \Delta \tilde{x}_k \]

\[ \lambda_k = P_k \Delta x_k - s_R(k) \]

\[ s_R(k) = Q \Delta \tilde{x}_k - S_k \Delta \tilde{x}_{k+1} \]

(3.11)
Doing the same manipulations as in the derivation of the LQ optimal regulator (see [10]) results in the following initial (final) conditions:

\[
P_N = Q, \quad s_R(N) = Q\Delta \hat{x}_N
\]

\[
P_k = Q + \Phi^T P_{k+1} \left[ I + \bar{B} R^{-1} \bar{B}^T P_{k+1} \right]^{-1} \Phi =
\]

\[
= Q + \Phi^T P_{k+1} \Phi - \Phi^T P_{k+1} \bar{B} \left[ \bar{B}^T P_{k+1} \bar{B} + R \right]^{-1} \bar{B}^T P_{k+1} \Phi
\]

\[
s_R(k) = Q\Delta \hat{x}_k + \Phi^T \left[ I + P_{k+1} \bar{B} R^{-1} \bar{B}^T \right]^{-1} s_R(k + 1)
\]

\[
\bar{B} = \begin{bmatrix} B & B_d \end{bmatrix}
\]

\[
R = \begin{bmatrix} R_u & 0 \\ 0 & -\gamma^2 R_d \end{bmatrix}
\]

The results are formally the same as in the LQ optimal tracking case (see (2.5)) but with the modified discrete time Riccati difference equation which can be solved through \( \gamma \) iteration (see [19]).

Considering the structure of \( s_R(k) \) and \( \Delta \hat{x}_k = H \Delta r_k = C_r^T (C_r C_r^T)^{-1} \Delta r_k \) the optimal input and worst case disturbance can be written in the following form:

\[
\Delta \hat{u}_k = -R_u^{-1} B_u^T \lambda_{k+1} =
\]

\[
= -R_u^{-1} B_u^T P_{k+1} \left[ I + M_R P_{k+1} \right]^{-1} \Phi \Delta x_k + R_u^{-1} B_u^T \left[ I + P_{k+1} M_R \right]^{-1}.
\]

\[
\cdot (Q\Delta \hat{x}_{k+1} - S_{k+1} \Delta \hat{x}_{k+2}) = -K_{x_k} \Delta x_k + K_{Q_k} \Delta r_{k+1} + K_{S_k} \Delta r_{k+2}
\]

where

\[
K_{x_k} = R_u^{-1} B_u^T P_{k+1} \left[ I + \bar{B} R^{-1} \bar{B}^T P_{k+1} \right]^{-1} \Phi
\]

\[
K_{Q_k} = R_u^{-1} B_u^T \left[ I + P_{k+1} \bar{B} R^{-1} \bar{B}^T \right]^{-1} QC_r^T (C_r C_r^T)^{-1}
\]

\[
K_{S_k} = R_u^{-1} B_u^T \left[ I + P_{k+1} \bar{B} R^{-1} \bar{B}^T \right]^{-1} S_{k+1} C_r^T (C_r C_r^T)^{-1}
\]

\[
\Delta \hat{d}_k^* = \frac{1}{\gamma^2} R_d^{-1} B_d^T \lambda_{k+1} = \frac{1}{\gamma^2} R_d^{-1} B_d^T P_{k+1} \left[ I + M_R P_{k+1} \right]^{-1} \Phi \Delta x_k -
\]

\[
- \frac{1}{\gamma^2} R_d^{-1} B_d^T \left[ I + P_{k+1} M_R \right]^{-1} (Q \Delta \hat{x}_{k+1} - S_{k+1} \Delta \hat{x}_{k+2}) =
\]

\[
= L_{x_k} \Delta x_k - L_{Q_k} \Delta r_{k+1} + L_{S_k} \Delta r_{k+2}
\]

where

\[
L_{x_k} = \frac{1}{\gamma^2} R_d^{-1} B_d^T P_{k+1} \left[ I + \bar{B} R^{-1} \bar{B}^T P_{k+1} \right]^{-1} \Phi
\]

\[
L_{Q_k} = \frac{1}{\gamma^2} R_d^{-1} B_d^T \left[ I + P_{k+1} \bar{B} R^{-1} \bar{B}^T \right]^{-1} QC_r^T (C_r C_r^T)^{-1}
\]

\[
L_{S_k} = \frac{1}{\gamma^2} R_d^{-1} B_d^T \left[ I + P_{k+1} \bar{B} R^{-1} \bar{B}^T \right]^{-1} S_{k+1} C_r^T (C_r C_r^T)^{-1}
\]
This completes the derivation of the minimax tracking controller for finite horizon problems. All the calculation expressions are recursive, so they need the knowledge of the reference signal on the whole horizon in advance. This is not a useful property if the future references are not known. This difficulty can be solved considering the infinite horizon solution.

3.2.6 Derivation of the LQ sub-optimal infinite horizon solution for the output tracking minimax problem

Equations (3.12) and (3.13) are formally exactly the same as in the LQ case (see (2.12)). This way the limiting solution $P_\infty$ of the modified DARE (MDARE) instead of the time-varying one can be analogously substituted into it.

This leads to formally exactly the same expressions as for the LQ optimal tracker in (2.14), the only difference is the use of $B$ instead of $B$ and here $R$ means a block diagonal weighting matrix $R = \begin{bmatrix} R_u & 0 \\ 0 & -\gamma^2 R_d \end{bmatrix}$ which depends on the $\gamma$ tuning parameter. This means that this problem has exactly the same sub-optimal solution for $S_1$ and $S_2$ as in (2.21) with the extrapolation of the reference signal.

After determining $S_1$ and $S_2$ the optimal control input and worst case disturbance can be calculated from (3.13) and (3.14).

However, the application of the worst case disturbance in the controlled system is usually impossible because it acts through $G$ instead of $B$ so it can not be mixed with the control input (for example we can not generate and apply a worst case wind disturbance for an aircraft). This means that during the solution of the MDARE with $\gamma$ iteration the stability of the controlled system with purely the optimal input should be checked and satisfied in every step (see [49]) and the worst case disturbance should not be formulated.

The optimal control input for the centered system results as (by considering the linear extrapolation of the reference signal):

$$\Delta \hat{u}_k = - R_u^{-1} B^T P_\infty \left[ I + B R^{-1} B^T P_\infty \right]^{-1} \Phi \Delta x_k + R_u^{-1} B^T M_2 S_1 \Delta r_{k+1} -$$

$$- R_u^{-1} B^T M_2 S_2 \Delta r_{k+2} = -K x_2 \Delta x_k + K_{S_1} \Delta r_{k+1} - K_{S_2} \Delta r_{k+2} =$$

$$= -K x_2 \Delta x_k + \left( K_{S_1} - 2K_{S_2} \right) \Delta r_{k+1} + K_{S_2} \Delta r_k$$

3.2.7 Summation of the control input components

Using (3.2), (3.5) and (3.6) the final input signal required to control the system can be composed as shown in (3.16).

From the last two expressions in (3.16) the final required control input can be constructed with the proper substitution of $r_\infty$ and $\hat{d}_\infty$. These are variables only valid for constant reference signal and disturbance in steady state. However, in application $d_\infty = d_\infty$ is valid only in steady state and $r_\infty$ and $d_\infty$ are not a-priori known values. This problem can be solved substituting $r_{k+1}$ in place of $r_\infty$ and $\hat{d}_k$ in place of $d_\infty$. This re-
sults in the same steady state expression for the constant case and works for time-varying references and disturbances also.

\[ \Delta \hat{u}_k = \hat{u}_k - \hat{u}_\infty \Rightarrow \hat{u}_k = \Delta \hat{u}_k + \hat{u}_\infty \]

\[ u_k = \hat{u}_k - B^+ \hat{d}_k \]

\[ \hat{u}_k = -K_{x1} x_k + u_k = -K_{x1} x_k + \hat{u}_k - B^+ \hat{d}_k = \]

\[ = -K_{x1} x_k + \Delta \hat{u}_k + \hat{u}_\infty - B^+ \hat{d}_k \]

\[ \tilde{u}_k = -K_{x1} x_k + K_{x2} \Delta x_k + (K_{s1} - 2K_{s2}) \Delta r_{k+1} + K_{s2} \Delta r_k + F^+ r_\infty - \]

\[ - F^+ C_r (I - \Phi)^{-1} (I - BB^+) G \dot{d}_\infty - B^+ \hat{d}_k = \]

\[ = - (K_{x1} + K_{x2}) x_k + K_{x2} x_\infty + (K_{s1} - 2K_{s2}) r_{k+1} + K_{s2} r_k + (K_{s2} - K_{s1}) r_\infty + \]

\[ \Delta G \hat{d}_k \]

\[ + F^+ r_\infty - F^+ C_r (I - \Phi)^{-1} (I - BB^+) G \dot{d}_\infty - B^+ \hat{d}_k \]

\[ x_\infty = (I - \Phi)^{-1} B \hat{u}_\infty + (I - \Phi)^{-1} (I - BB^+) \dot{d}_\infty = \]

\[ = (I - \Phi)^{-1} B F^+ r_\infty + (I - \Phi)^{-1} (I - BB^+) \dot{d}_\infty - \]

\[ - (I - \Phi)^{-1} B F^+ C_r (I - \Phi)^{-1} (I - BB^+) \dot{d}_\infty \]

Another aspect is the application of the estimated state \( \hat{x}_k \) instead of \( x_k \) because the latter cannot be directly measured. This way the final expression for \( \tilde{u}_k \):

\[ \tilde{u}_k = -K_{x1} \hat{x}_k + K_{x2} x_\infty + (K_{s1} - 2K_{s2}) r_{k+1} + K_{s2} r_k + (K_{s2} - K_{s1}) r_\infty + \]

\[ + F^+ r_{k+1} - F^+ C_r (I - \Phi)^{-1} (I - BB^+) \dot{d}_k - B^+ \hat{d}_k \]

\[ x_\infty = (I - \Phi)^{-1} B F^+ r_{k+1} + (I - \Phi)^{-1} (I - BB^+) \dot{d}_k - \]

\[ - (I - \Phi)^{-1} B F^+ C_r (I - \Phi)^{-1} (I - BB^+) \dot{d}_k \]

\[ \tilde{u}_k = -K_{x1} \hat{x}_k - K_{s2} (r_{k+1} - r_k) + K_{r_\infty} r_{k+1} + K_{d_\infty} \hat{d}_k \]

where

\[ K_{r_\infty} = (K_{x2} (I - \Phi)^{-1} B + I) F^+ \]

\[ K_{d_\infty} = (K_{x2} (I - \Phi)^{-1} (I - BB^+) - \]

\[ - K_{x2} (I - \Phi)^{-1} B F^+ C_r (I - \Phi)^{-1} (I - BB^+) - \]

\[ - F^+ C_r (I - \Phi)^{-1} (I - BB^+) - B^+ G \]

This way a control solution is derived which is LQ optimal for constant references and disturbances and sub-optimal for time-varying references and disturbances (see the explanation in the LQ optimal tracker case).

The closed loop interconnection structure of the controlled system can be seen in figure 3.1 (for notations see Appendix S.10).
Notice that the input of the state and disturbance estimator should be different from \( \tilde{u}_k \) in (3.17) because it was designed for the system resulted in the second step with stable state matrix \( \Phi \). This means that \(-K_1\hat{x}_k\) should be neglected and so, the input to the estimator results as:

\[
u^e_k = -K_2\hat{x}_k - K_2 (r_{k+1} - r_k) + K_r\infty r_{k+1} + K_d\infty \hat{d}_k \tag{3.18}
\]

However, if the original system is stable and the disturbance estimator was designed to it, this modification should not be applied. In the next section the properties for constant and time-varying signals will be stated and proven.

### 3.3 The properties of the infinite horizon LQ optimal minimax tracker

Considering the derived tracking solution, the following properties can be stated:

1. It satisfies the separation principle both for constant and time-varying references and disturbances.

2. It does not require anti-windup compensation because of static gain memoryless control (no internal states of controller).

3. It guarantees asymptotic stability, zero steady-state tracking error, finite LQ functional value (on infinite horizon) and so, LQ optimality for constant, finite references and disturbances.

4. It is sub-optimal, BIBO and so \( l_p \) stable for time-varying references and disturbances. Sub-optimal here means that the infinite horizon LQ optimal minimax tracking problem was only approximately solved and so, optimality was lost for these reference signal class.

5. It guarantees finite cost functional value (on infinite horizon) for \( l_1/l_2 \) references and disturbances.
3.3.1 The satisfaction of the separation principle

At first, derive the error dynamics of the state estimation considering (3.3) and (3.18):

\[
\begin{align*}
\dot{d}_{k-1} &= M \left( y_k - C \hat{x}_{k-1} - CBu_{k-1}^c \right) \\
d_k &= \Phi \hat{x}_{k-1} + Bu_{k-1}^c + K \left( y_k - C \hat{x}_{k-1} - CBu_{k-1}^c \right) \\
x^e_k &= \hat{x}_k - x_k = (I - KC) \Phi x_{k-1}^c = L \dot{x}_{k-1} \\
d_k^c &= -A_{y_k}^d MC \Phi x_{k-1}^c
\end{align*}
\] (3.19)

In (3.19) \( L \) results as a stable matrix if one uses (3.3) to design the estimator. Construct now the augmented dynamics of the controlled system (system and estimator together) defining the augmented state as

\[
x_k^a = [x_k^T \ x_k^e T \ d_k^T]^T:
\]

\[
x_{k+1} = Ax_k + Bu_k + Gd_k = Ax_k - BKx_k \hat{x}_k - BK_S \left( r_{k+1} - r_k \right) + BK_{r_{\infty}} r_{k+1} + \\
+ BK_{d_{\infty}} \dot{d}_k - BKx_k x_k + BK_{x_k} x_k + Gd_k
\]

\[
x_{k+1} = \underbrace{(A - BKx_k)}_{\Phi_1} x_k - BK_k x_k - BK_{S_2} \left( r_{k+1} - r_k \right) + BK_{r_{\infty}} r_{k+1} + BK_{d_{\infty}} \dot{d}_k + Gd_k
\]

\[
\begin{bmatrix}
x_{k+1}^a \\
x_{k+1}^e \\
d_{k+1}
\end{bmatrix} = 
\begin{bmatrix}
\Phi_1 & -BKx_k & BK_{d_{\infty}} \\
0 & L & 0 \\
0 & -A_{y_k}^d MC \Phi & 0
\end{bmatrix}
\begin{bmatrix}
x_k \\
x_k^e \\
d_k
\end{bmatrix} +
\begin{bmatrix}
G \\
0 \\
A_{y_k}^d
\end{bmatrix} d_k +
\begin{bmatrix}
BK_{S_2} \\
B \left( K_{r_{\infty}} - K_{S_2} \right) \\
0 \\
0
\end{bmatrix}
\begin{bmatrix}
r_k \\
r_{k+1}
\end{bmatrix}
\]

\] (3.20)

(3.20) shows that the augmented system satisfies the separation principle, because neither the state system \( x_k \) nor the reference terms \( r_{k+1}, r_k \) affect the dynamics of the state and disturbance estimator. The stability of the augmented state matrix \( A^a \) can be checked using block matrix determinant lemma twice:

\[
\det \left( \begin{bmatrix}
zI - \Phi_1 & BKx_k & -BK_{d_{\infty}} \\
0 & zI - L & 0 \\
0 & A_{y_k}^d MC \Phi & zI
\end{bmatrix} \right) = \det \left( zI - \Phi_1 \right) \det \left( \begin{bmatrix}
zI - L & 0 \\
A_{y_k}^d MC \Phi & zI
\end{bmatrix} \right) =
\]

\[
= \det \left( zI - \Phi_1 \right) \det \left( zI - L \right) \det \left( zI \right) = 0
\]

(3.21)

\( \Phi_1 \) and \( L \) are stable matrices so, their poles are inside the unit circle. \( \det \left( zI_d \right) = 0 \) gives zero poles which are also stable. This means that the augmented state matrix is stable and so, the augmented system dynamics is also stable.
3.3.2 No need for anti-windup compensation

Let’s consider the structure of the system input from (3.17):

\[ \tilde{u}_k = -K_x \hat{x}_k - K_S (r_{k+1} - r_k) + K_{r,\infty} r_{k+1} + K_{d,\infty} \hat{d}_k \]  

(3.22)

In (3.22) only the actual value of estimated system state and disturbance vector and the actual and previous reference signal values are applied. There is no integral state and so there is no need for anti-windup compensation, the resulting control input can be simply saturated. The control input is again PD-like as in (2.47) with the same difference from conventional PD control in using only the difference of the reference signal instead of the tracking error. This can give better noise attenuation properties because usually the reference signals are generated by the user in a noiseless way.

3.3.3 Asymptotic stability and zero steady state error for constant, finite references and disturbances

Constant references and disturbances mean that \( r_{k+1} = r_k = r_\infty \) and \( d_{k+1} = d_k = d_\infty \) \( \rightarrow \) \( A_{y_k} = I \) these will be considered in the forthcoming calculations. The steady state of the augmented system will be (in a similar way as for the LQ optimal tracker in (2.48)):

\[
x_{a,\infty} = (A^a)^\infty x_0 + (I - A^a)^{-1} G^a d_\infty + (I - A^a)^{-1} B^a r_2(\infty)
\]  

(3.23)

As can be seen from (3.23) the augmented system is asymptotically stable for finite reference and disturbance inputs. \( (I - A^a)^{-1} \) can be calculated using twice the block matrix inversion lemma:

\[
\begin{bmatrix}
I - \Phi & BK_x & -BK_d \n
0 & I - L & 0 \\
0 & MC\Phi & I \\
\end{bmatrix}^{-1} = \begin{bmatrix}
A_{n,n} & A_{n,n+d} \\
0 & A_{n+d,n+d} \\
\end{bmatrix}^{-1} = \\
\begin{bmatrix}
A_{n,n}^{-1} & -A_{n,n}^{-1} A_{n+d,n+n+d} \\
0 & A_{n+d,n+n+d}^{-1} \\
\end{bmatrix} \\
A_{n+d,n+d}^{-1} = \begin{bmatrix}
I - L & 0 \\
MC\Phi & I \\
\end{bmatrix}^{-1} = \begin{bmatrix}
(I - L)^{-1} & 0 \\
-MC\Phi (I - L)^{-1} & I \\
\end{bmatrix} \\
(I - A^a)^{-1} = \begin{bmatrix}
(I - \Phi_1)^{-1} & - (I - \Phi_1)^{-1} [BK_x & -BK_d] A_{n+d,n+n+d}^{-1} \\
0 & A_{n+d,n+n+d}^{-1} \\
\end{bmatrix} = \\
\begin{bmatrix}
(I - \Phi_1)^{-1} & - (I - \Phi_1)^{-1} [BK_x + BK_d MC\Phi] (I - L)^{-1} (I - \Phi_1)^{-1} BK_d \\
0 & (I - L)^{-1} \\
0 & -MC\Phi (I - L)^{-1} \\
\end{bmatrix}
\]

(3.24)

Now the nonzero terms in (3.23) can be calculated considering (3.24) and (3.20):
\[(I - A^*)^{-1} G^* d_\infty = \begin{bmatrix}
(I - \Phi_1)^{-1} Gd_\infty + (I - \Phi_1)^{-1} BK_{d_\infty} d_\infty \\
0 \\
d_\infty
\end{bmatrix} \]  
(3.25)

\[(I - A^*)^{-1} B^* r_2(\infty) = \begin{bmatrix}
(I - \Phi_1)^{-1} BK_{r_\infty} r_\infty \\
0 \\
0
\end{bmatrix} \]  
(3.26)

Substituting the terms from (3.25) into (3.23) gives the steady state of the augmented system:

\[
\begin{bmatrix}
x_\infty \\
x^*_\infty \\
d_\infty
\end{bmatrix} = \begin{bmatrix}
(I - \Phi_1)^{-1} Gd_\infty + (I - \Phi_1)^{-1} BK_{d_\infty} d_\infty + (I - \Phi_1)^{-1} BK_{r_\infty} r_\infty \\
0 \\
d_\infty
\end{bmatrix} \]  
(3.26)

(3.26) shows that the state estimation error in steady state is zero and the estimated disturbance converges to the constant disturbance value. The next step is the evaluation of the steady output resulting from the steady state \(x_\infty\). This can be done using (3.26) and (3.17) and considering \(\Phi_1 = \Phi - BK_{r_2}\) and \(F = C_r (I - \Phi)^{-1} B\):

\[
y_\infty = C_r x_\infty = C_r (I - \Phi_1)^{-1} Gd_\infty + C_r (I - \Phi_1)^{-1} B [K_{x_2} (I - \Phi)^{-1} (I - B B^+) - \\
K_{x_2} (I - \Phi)^{-1} B F^+ C_r (I - \Phi)^{-1} (I - B B^+) - \\
F^+ C_r (I - \Phi)^{-1} (I - B B^+) - B^+ Gd_\infty + \\
+ C_r (I - \Phi_1)^{-1} B (K_{x_2} (I - \Phi)^{-1} B + I) F^+ r_\infty = * \\
C_r (I - \Phi_1)^{-1} (BK_{x_2} (I - \Phi)^{-1} + I) B F^+ r_\infty = \\
= C_r (I - \Phi_1)^{-1} \left( BK_{x_2} + (I - \Phi_i) \right) (I - \Phi)^{-1} B F^+ r_\infty = r_\infty
\]  
(3.27)

\[
C_r (I - \Phi_1)^{-1} B [K_{x_2} (I - \Phi)^{-1} M_B - K_{x_2} (I - \Phi)^{-1} B F^+ C_r (I - \Phi)^{-1} M_B - \\
- F^+ C_r (I - \Phi)^{-1} M_B - B^+ Gd_\infty = \\
= C_r (I - \Phi_1)^{-1} [BK_{x_2} (I - \Phi)^{-1} M_B - BB^+] - \\
- (BK_{x_2} + (I - \Phi)) (I - \Phi)^{-1} B F^+ C_r (I - \Phi)^{-1} M_B \right] \]  
(3.28)
This way the steady state output equals the constant reference signal so, the zero steady
state tracking error is guaranteed.

### 3.3.4 Finite functional value and LQ optimality for infinite horizon with constant, finite references and disturbances

It is pointed out that \( x_k \to x_\infty \) if \( k \to \infty \). This means that \( \Delta x_k \to 0 \). Examine now the functional given in (3.10). If \( r_{k+1} = r_k = r_\infty = \text{const} \ \forall k \geq 0 \) then \( \Delta r_k = 0 \ \forall k \), \( \Delta \ddot{x}_k = H \Delta \dot{r}_k = 0 \) and \( \Delta \dot{u}_k = -K_{x} \Delta x_k \) (considering (3.15)). This way \( \Delta \dot{u}_k \) goes to zero as \( \Delta x_k \to 0 \). In this case the functional has the following form:

\[
J \left( \Delta x, \Delta \ddot{x}, \Delta \dot{u}, \Delta \dot{d} \right) = \frac{1}{2} \sum_{k=0}^{\infty} \left( \Delta x_k^T Q \Delta x_k + \Delta \dot{u}_k^T R_u \Delta \dot{u}_k - \gamma^2 \Delta \dot{d}_k^T R_d \Delta \dot{d}_k \right)
\]

(3.29)

Here

\[
\Delta \dot{d}_k = G (d_k - d_\infty) - BB^+ G \left( \dot{d}_k - \dot{d}_\infty \right) \to 0 \text{ as } k \to \infty
\]

This way the functional in (3.29) describes the well known minimax regulator problem, where the states and worst case disturbances go to zero together with the control input and this way the value of the infinite horizon functional is finite and so, the solution is LQ optimal.

### 3.3.5 BIBO and so \( l_p \) stability with \( l_p \) time-varying references

Considering (3.11) and (3.17) and \( z \) as the forward shift operator the input-output dynamics of the system results as (applying the real state in the control input):

\[
x_{k+1} = A x_k + B \dot{u}_k + G d_k
\]

\[
\dot{u}_k = -K_{x} \ddot{x}_k - K_{s2} (r_{k+1} - r_k) + K_{r_\infty} r_{k+1} + K_{d_\infty} \dot{d}_k = \Delta \dot{x}_{k+1}
\]

\[
x_{k+1} = \underbrace{A x_k - BK_{x} x_k}_{\Phi_1 x_k} - BK_{s2} \Delta r_{k+1} + BK_{r_\infty} r_{k+1} + BK_{d_\infty} \dot{d}_k
\]

\[
x_k = - (z I - \Phi_1)^{-1} BK_{x} x_k - (z I - \Phi_1)^{-1} BK_{s2} \Delta r_{k+1} + (z I - \Phi_1)^{-1} BK_{r_\infty} r_{k+1} + (z I - \Phi_1)^{-1} BK_{d_\infty} \dot{d}_k
\]

\[
y_k = C r x_k = -C_r (z I - \Phi_1)^{-1} BK_{x} x_k - C_r (z I - \Phi_1)^{-1} BK_{s2} \Delta r_{k+1} + C_r (z I - \Phi_1)^{-1} BK_{r_\infty} r_{k+1} + C_r (z I - \Phi_1)^{-1} BK_{d_\infty} \dot{d}_k
\]

(3.30)
In (3.30) all the transfer functions from the reference signals and estimated disturbance to the output are stable because of the stability of the state transition matrix $\Phi$. For a stable state estimator the $x_k^e$ estimation error should be a bounded signal which means that the output will be bounded. This means the BIBO and so $l_p$ stability of the closed loop system for any $l_p$ time-varying reference signal.

### 3.3.6 Finite functional value (on infinite horizon) for $l_1/l_2$ references

$l_1$ and $l_2$ references and disturbances with similar upper bounds as derived in Appendix 8.7 are considered here:

$$\left| r_k \right| < Ae^{-\alpha k} \quad \left| d_k \right| < De^{-\beta k} \quad (3.31)$$

After defining the reference and disturbance signal class, the value of the infinite horizon functional from (3.10) can be examined considering these type of references and disturbances with $r_\infty = 0$ and $d_\infty = 0$. $d_\infty = 0$ means $\hat{d}_\infty = 0$ because of the asymptotically convergent disturbance estimator. This means that $\hat{u}_\infty = 0$ (see (3.30)) and $x_\infty = 0$ (see (3.7)). This way $\Delta x_k = x_k$, $\Delta \hat{x}_k = \hat{x}_k$, $\Delta \hat{u}_k = \hat{u}_k$, $\Delta \hat{d}_k = \hat{d}_k = \left[ \begin{array}{c} d_k \\ \hat{d}_k \end{array} \right]$ and the infinite horizon functional from (3.10) can be formulated as:

$$J(x, \hat{x}, \hat{u}, \hat{d}_k) = \frac{1}{2} \sum_{k=0}^{\infty} \left( (x_k - \hat{x}_k)^T Q (x_k - \hat{x}_k) + \hat{u}_k^T R_{\hat{u}} \hat{u}_k - \gamma^2 \hat{d}_k^T R_{\hat{d}} \hat{d}_k \right) \quad (3.32)$$

The system input from (3.15) can be reformulated as (considering the zero infinite time signal values and the estimated system state):

$$\hat{u}_k = -K_{x2} \hat{x}_k + \mathbf{K}_{S1} r_{k+1} + K_{S2} r_k = -K_{x2} x_k + K_{S2} x_k + \mathbf{K}_{S1} r_{k+1} + K_{S2} r_k = -K_{x2} x_k + u_k^r \quad (3.33)$$

This is formally the same reformulation as in the LQ optimal case in (2.57). Exactly the same upper bound for the absolute value of the reference signal related part can be derived as in Appendix 8.8 as $|u_k^r| < K_U e^{-\alpha k}$. Considering the state dynamics from (3.7) and the input from (3.33) the $k$th value of system state results as:

$$x_1 = \Phi x_0 + B \hat{u}_0 = (\Phi - BK_{x2}) x_0 + Bu_0^c + B_d \hat{d}_0 = \Phi_1 x_0 + Bu_0^c + B_d \hat{d}_0$$

$$x_2 = \Phi_1 x_1 + Bu_0^c + B_d \hat{d}_1 = \Phi_1^2 x_0 + \Phi_1 Bu_0^c + Bu_0^c + B_d \hat{d}_0 + B_d \hat{d}_1$$

$$\vdots$$

$$x_k = \Phi_1^{k-1} x_0 + \sum_{l=0}^{k-1} \left( \Phi_1^l \left[ B \quad B_d \right] \left[ \begin{array}{c} u_{k-l-1}^r \\ \hat{d}_{k-l-1} \end{array} \right] \right) = \Phi_1^k x_0 + R_k \quad (3.34)$$
This is formally exactly the same as in the LQ optimal case in (2.58). The finiteness of the functional (3.32) is proven in Appendix 8.14 substituting (3.33), (3.34) and considering the stability of Φ₁ and l₁ references and disturbances.

In the next section the developed LQ optimal minimax tracker (shortly minimax tracker) will be compared to other methods considering the aerospace application example described in section 1.1.

3.4 Comparison with other methods in Matlab simulations

The above developed minimax tracking solution was compared to the same seven other control design methods as the LQ tracker using the same augmented aircraft lateral dynamical model. The parameters of the aircraft lateral model were the averages of the parameters of model 1 and model 2 from Table 1.1 in all test cases. The tuning goals and methods were the same as in the LQ tracker case. All solutions were tuned by the author. The main difference in these cases is the application of disturbance estimation and correction. Two kind of disturbance estimators are used.

3.4.1 Constant unknown disturbance estimation through system input (UDEB)

This technique is published in [9] and assumes that the constant, unknown disturbance affects the system in the input directions through the B matrix. Considering the 'dynamics' of the disturbance the following augmented system can be constructed:

\[
\begin{align*}
    x_{k+1} &= Ax_k + Bu_k + Bd_k \\
    y_{k+1} &= Cx_{k+1} \\
    x_{k+1}^{d} &= x_k^{d} = d_k \\
    \begin{bmatrix} x_{k+1} \\ x_{k+1}^{d} \end{bmatrix} &= \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ x_k^{d} \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k \\
    y_k &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_k^{d} \end{bmatrix} 
\end{align*}
\]

(3.35)

Considering this augmented system an actual state estimator can be designed by pole placement applying duality between regulator and estimator:

\[
\begin{align*}
    \hat{x}^a_{k+1} &= F\hat{x}^a_k + Gy_{k+1} + Hu_k \\
    K &= \text{place}(A^T, A^TC^T, p) \\
    G &= K^T \\
    F &= A - GCA \\
    H &= B - GCB
\end{align*}
\]

(3.36)

From \( \hat{x}^a_k, \hat{x}^d_k = \hat{d}_k \) can be unpacked and used to compensate the disturbance effect subtracting it from the control input \( \tilde{u}_k = u_k - \hat{d}_k \) here \( u_k \) depends on the applied control method.
The simulation experiences have shown that $d_k$ can take very large values when the system disturbance $d_k$ suddenly changes and this can lead to poor performance of the controlled system. That’s why a smoothing low pass filter is applied on every $d_k$ component with the following CT transfer function (in every test cases):

$$G_{efilt} = \frac{1}{0.32s + 1} \quad (3.37)$$

This function has 0.5Hz cutoff frequency. Its DT equivalent sampled at 25Hz was used in the DT systems.

### 3.4.2 Unknown disturbance estimation (UDE)

This is the estimator described earlier in (3.3) by assuming $A_{y_k+1}^d = I$ (slowly varying disturbance). It was designed for the reduced two state lateral dynamics of the aircraft including only $p$ and $r$. This part of the system is stable, so no pre-stabilization was required for estimator design. However, with this reduced system the number of disturbances will be equal with the number of outputs ($p$ & $r$) and so, the filter from (3.3) will have a special form with $M = G^{-1}$ and $K = 0$. This way there is only state prediction, no state correction in the filter. The same low-pass filter (3.37) was applied to the estimated disturbance in this estimator also.

*Note* if the UDEB solution works well it can be also applied instead of this special estimator. But this arise the question about the application of $B\hat{d}_k$ in the minimax controller. Will this represent the case when $\hat{d}_k$ is simply subtracted from the input? Let’s consider $K_{d\infty}$ in (3.17) with $G = B$:

$$K_{d\infty} = [K_{x2} (I - \Phi)^{-1} (I - BB^+)B - K_{x2} (I - \Phi)^{-1} BF^+ C_r (I - \Phi)^{-1} (I - BB^+) B - F^+ C_r (I - \Phi)^{-1} (I - BB^+) B - B^+ B]$$

$$B^+ B = I \Rightarrow (I - BB^+) B = 0 \Rightarrow K_{d\infty} = -I \quad (3.38)$$

(3.38) shows that in this case only $\hat{d}_k$ should be subtracted from the input, which means that the generalized case with $G$ gives the trivial solution in this special case.

In the controller testing again perfect knowledge of system model and system state vector without noise effects was assumed. The noise tolerance and robustness of the minimax method will be examined in a separate chapter (chapter 5). The considered system model is the DT equivalent of (1.7) with nonzero disturbances.

For the methods FF, $\int FF$, Simple LQ, LQ Servo and PID exactly the same weights (closed-loop poles) and control gains are used but the controlled system is completed with the UDEB estimator and the negative estimated disturbance is added to the control input (as can be seen in the block structures in Appendix 8.11). It is worth to note that the disturbance acts to the system without delay (through $G$ instead of $B$) and so, only the disturbance free part $u_k$ of control input $\hat{u}_k$ should be connected to the Padé model of the delay in the controller to have acceptable performance. The Padé model is inside the control algorithm (not part of the system) so this is not a problem.
In the MPC and $\mathcal{H}_\infty$ designs the UDE estimator is used and the control design problems are restructured (also different weighting is applied) so these methods are shortly described in the next sections.

### 3.4.3 Model predictive control (MPC) with static gains (MPC)

The MPC control derivation in the previous chapter (chapter 2) should be modified because of the disturbance. It is assumed to be a known exogenous signal.

Considering the system dynamics with the disturbance:

$$\hat{z}_{k+1} = C_r x_{k+1} = C_r A x_k + C_r B \Delta \hat{u}_k + C_r B u_{k-1} + C_r G d_k$$

$$\vdots$$

$$\hat{z}_{k+N} = C_r A^N x_k + \sum_{j=k}^{k+N-1} \left\{ C_r \left( \sum_{l=0}^{k+N-j-1} A^l \right) B \right\} \Delta u_j + C_r \left( \sum_{l=0}^{N-1} A^l \right) B u_{k-1} + C_r \left( \sum_{l=0}^{N-1} A^l \right) G d_k$$

The following system of equations can be formulated:

$$Z(k) = \Upsilon(k) x_k + \Theta(k) \Delta U(k) + \Gamma(k) u_{k-1} + \Omega(k) d_k$$

$$\Upsilon(k)(i) = C_r A^i, \quad \Theta(k)(i, j) = C_r \left( \sum_{l=0}^{k+i-1-j} A^l \right) B, \quad \Gamma(k)(i) = C_r \left( \sum_{l=0}^{i-1} A^l \right) B$$

$$\Omega(k)(i) = C_r \left( \sum_{l=0}^{i-1} A^l \right) G, \quad i = 1 : N, \quad j = 0 : N - 1$$

This way $\varepsilon(k)$ in (2.69) should be modified to $\varepsilon(k) = T(k) - \Upsilon(k) x_k - \Gamma(k) u_{k-1} - \Omega(k) d_k$ otherwise the solution is the same.

The final control input should be also modified (with the same partitioning of $K_r$ as in (2.71)):

$$u_k = - \underbrace{M_1 \Upsilon(k) x_k}_{K_x} + (I - \underbrace{M_1 \Gamma(k)}_{K_u}) u_{k-1} - \underbrace{M_1 \Omega(k)}_{K_d} d_k + \underbrace{M_1 T(k)}_{K_r}$$

In the application example the Padé approximation of the delay was too inaccurate leading to system oscillations. This was the only test case where the second order Padé approximation gave unsatisfactory results. That’s why the integer delay representation was used (0.2s delay with 0.04s sampling time is exactly five steps) resulting in the following DT augmented system:
This modification highly increases the dimension of the system (and controller) model. Because of the structure of the MPC problem again $p$, $\phi$ and $\tau$ are all considered as outputs (in this order), the $C_r$ matrix was constructed accordingly. The final weighting matrices with best tracking performance resulted as:

$$Q = \begin{bmatrix} 30 & 1500 & 2 \end{bmatrix}, \quad R = \begin{bmatrix} 1000 & 10000 \end{bmatrix}$$

### 3.4.4 $H_\infty$ optimal control ($H_\infty$)

Again the well known (\cite{85,86}) central $H_\infty$ control solution is applied using the discrete time LMI solver. The interconnection structure is shown in figure 3.2. Nonzero $\phi$ and zero filtered yaw rate ($\tau$) references are considered in this design. Unfortunately the acceptable rejection of the low frequency disturbance was not possible without considering it as a known input. This proves the assumption in the motivation part about the impossibility of good low-frequency tracking performance and disturbance rejection together.

![Figure 3.2: System interconnection structure to design disturbance rejecting $H_\infty$ tracking controller](image)

The weighting functions are given again in CT and the bandwidth resulted again about $2\text{rad/s}$.

The ideal transfer function $T_{\tau y}$ and weighting functions are as follows (an additional shaping filter for low-frequency disturbance $W_d$, $W_{e1}$ for $\phi$ tracking error, $W_{e2}$ for $\tau$ tracking error, $W_p$ for limiting roll rate, $W_n$ to bound control inputs, $W_r$ to give the magnitude of $\phi$ references and $W_n$ to give sensor noise magnitude):
\[ T_{\text{gy}} = \frac{1}{0.8s + 1} \quad W_d = \frac{0.12}{0.02s + 1} \]
\[ W_{e1} = \frac{20(0.05s + 1)}{5s + 1} \quad W_{e2} = \frac{10(0.05s + 1)}{5s + 1} \quad W_p = \frac{10(0.05s + 1)}{5s + 1} \]
\[ W_u = 10, 12 \quad W_r = 0.12 \quad W_n = 0.02 \times 2, 0.04, 10^{-6} \times 5 \]

### 3.4.5 Minimax tracker solution (MM tracker)

In this solution again both \( \phi \) and \( \tau \) are considered as tracking outputs. The design steps described in the development are evaluated. The poles of the augmented open-loop system and the (pre-)stabilizing poles are the same as in the FF case. The weighting strategy is similar to the LQ tracker. The final best weights of the control are:

\[ Q_1 = 500, 0 \times 7 \quad Q_2 = 500, 2 \quad R_u = 21000, 30000 > R_d = I_4 \cdot 10^{12} \]

The final \( \gamma \) value is \( 1.19 \cdot 10^{-4} \) which promises good disturbance rejection, the closed loop poles are:

\[ p_2 = [0.1624, 0.9265, 0.7219 \pm 0.2687i, 0.7228, 0.5894 \pm 0.0226i, 0.5317] \]

### 3.4.6 Comparison of roll doublet tracking control results

All the methods were tested by tracking the same roll doublet reference signal as in figure 2.3 and holding the filtered high frequency yaw rate around zero in Matlab simulation.

The numerical results are summarized in table 3.1 with the same parameters as in the LQ tracker test case. All the parameters are calculated from the whole 50 second simulation with the roll doublet reference.

**Table 3.1: MM Simulink test results**

| Ctrl       | \( T_{s1} \) [s] | \( T_{s2} \) [s] | MEAN \( |\phi_e| \) [°] | MEAN \( |\tau| \) [°/s] | \( \|p\|_2 \) [°/s] | \( \|\delta_a\|_2 \) [°] | \( \|\delta_r\|_2 \) [°] |
|------------|------------------|------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| PID        | 4.44             | –                | 1.616           | 0.18            | 372.1           | 71.52           | 38.54           |
| FF         | 1.52             | 1.68             | 2.096           | 0.065           | 242.35          | 63              | 40.68           |
| \( \int \) FF | 1.36             | 3                | 2.16            | 0.637           | 271.97          | 111.239         | 226.05          |
| Simple LQ  | 1.08             | 1.6              | 2.096           | 0.065           | 204.24          | 77.2            | 7.76            |
| LQ Servo   | 1.36             | 1.52             | 1.36            | 0.16            | 385.365         | 73.3            | 38.6            |
| MPC        | 2.88             | 3.68             | 2.49            | 0.09            | 345.18          | 70.72           | 38.03           |
| \( H_\infty \) | 2.4              | 3.12             | 1.63            | 0.12            | 287.87          | 78.69           | 3.71            |
| MM tracker | 2.8              | 4.16             | 2.295           | 0.192           | 275.5           | 67.9            | 39.71           |

Table 3.1 shows that not the MM tracker is the best solution, but there is no best solution. Some solutions are good regarding some parameters and worse regarding others.
Table 3.2: State dimensions of the solutions

<table>
<thead>
<tr>
<th></th>
<th>PID</th>
<th>FF</th>
<th>∫ FF</th>
<th>Simple LQ</th>
<th>LQ Servo</th>
<th>MPC</th>
<th>$\mathcal{H}_\infty$</th>
<th>MM tracker</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14/13</td>
<td>12/11</td>
<td>14/12</td>
<td>12/11</td>
<td>14/12</td>
<td>20</td>
<td>23/13</td>
<td>11</td>
</tr>
</tbody>
</table>

Considering the required state dimensions of the tested control solutions table 3.2 summarizes the results (including the states of Padé and washout filters and the state estimators applied in the control). $n_1/n_2$ cases are because of the difference in reference inputs (two or one) and the state dimension of the disturbance estimator $UDEB$ which can be possibly decreased from 5 to 4. The state dimension of $UDE$ is 2. Regarding the data, the MM tracker has the lowest required state space dimension. If computational time and memory limits are tight this can be an advantage.

The eight methods from the table are grouped into two figures, the first part with $FF$, $\int FF$, Simple LQ, PID and MM tracker and the second part with LQ Servo, MPC, $\mathcal{H}_\infty$ and MM tracker in 3.3 - 3.12. Considering the figures the methods can be better compared. For the convenience of the reader enlarged transients are shown in Appendix 8.16.

Regarding the tracking of the roll reference signal the MM tracker is slower than all of the methods. In the attenuation of the disturbance at 20 seconds its peak tracking error is similar to the others, but it is slower.

Regarding the filtered yaw rate the MM tracker has about the same convergence as the other methods but it has larger decrease rate.

Considering the pitch rate during the reference signal changes its convergence is slower (as the convergence of $\phi$) but during the disturbance changes it is faster. So it has faster disturbance attenuation.

Regarding the aileron and rudder deflections the MM tracker has larger decrease rates and the maximum deflections are inside the given limits.

As a summary, it can be stated that the MM tracker works similarly well and has the smallest state dimension so, it can be a potential concurrent of other methods.

Note the dynamics of the MM tracker can be made faster with re-tuning, but this weighting give the best results in real flight tests. With faster dynamics it has a tendency to oscillate, and possibly the other methods would also oscillate with their current weighting (except for the PID control which is also flight tested).
Figure 3.3: MM roll angle tracking, part 1

Figure 3.4: MM roll angle tracking, part 2

Figure 3.5: MM filtered yaw rate tracking, part 1

Figure 3.6: MM filtered yaw rate tracking, part 2

Figure 3.7: MM roll rate, part 1

Figure 3.8: MM roll rate, part 2
3.5 Summary

This chapter first gives a short literature review about the existing unknown input and state estimation methods and selects the real time applicable one. Then the infinite horizon, LQ optimal minimax, causal tracking problem is solved giving an optimal solution for constant, and a sub-optimal one for time-varying references and disturbances. In the solution, the estimated disturbance is used to cancel most of the disturbance effects. After stating and proving the properties of the derived method it is compared to the same seven methods as the LQ tracker again in Matlab simulations but considering also the disturbance effects.
Chapter 4

Aircraft attitude estimation for aerospace application

Both LQ and minimax design techniques lead to a state feedback controller completed with a reference (and disturbance) feedforward part. State feedback requires the knowledge of system states. In the thesis an aerospace application example (aircraft lateral dynamics) is used to examine the properties of the developed methods. This arose the question of aircraft state estimation. In a UAV application mostly angular rates and Euler angles should be used to design stabilization and tracking controllers (in the application example the roll- and yaw rate and roll angle is used). Angular rates are measured by onboard sensors but Euler angles have to be estimated with some algorithm.

The E-flite Ultrastick 25e test aircraft is equipped with strapdown inertial measurement unit (IMU) (with three dimensional angular rate, acceleration and magnetic measurements, and Pitot tube and static pressure measurements) and global positioning system (GPS) receiver (for details see [21]). It is a standard task to design an attitude estimator algorithm based on a strapdown IMU and GPS receiver. Many commercial-off-the-shelf solutions are available (IMUs with attitude output) but their internal software is fixed and unknown, which is a disadvantage in research. Otherwise, future research plans involve the design of automatic take-off and landing algorithms which require correct attitude information also on the ground and near to it. That’s why the goal of attitude estimator development is to design an estimator which can be used both on ground and in air from take-off to landing.

There are numerous publications on the topic of attitude estimation using different assumptions and methods ([11, 7, 8, 24, 55, 36, 42, 51, 55, 58, 59, 62]). [62] estimates the attitude and rate gyro bias of vertical take-off and landing UAVs in hover and near hover situations. Angular rate, acceleration and magnetic data is used assuming that the acceleration sensors measure the direction of Earth’s gravity vector (the same assumption is applied in [8, 58]). [42] uses the same data and assumption but for the estimation of aircraft attitude. This case the assumption about the acceleration is incorrect because of usually high dynamical acceleration effects. This problem was pointed out doing simulation tests with this algorithm applied on our Ultrastick aircraft simulation model. A possible correction to this problem can be found in [55] where velocity measurements are used to correct the measured acceleration with the angular rate - velocity cross product term. This way the corrected acceleration vector is closer to Earth’s gravity
vector and results are more accurate, but this is also only an approximation. Recent publications such as [56, 40] use angular rate, acceleration and GPS velocity measurements considering the aircraft’s whole dynamical acceleration compared to the acceleration derived from GPS velocity. This approach does not include any approximation and so, can give more accurate results. However, there can be two problems. The first is that the GPS velocity measurements are useless onboard an aircraft standing or slowly moving on the ground. The second is the effect of wind disturbance which can seriously modify the GPS velocity vector compared to aircraft air relative velocity (this problem is also mentioned in [56] but is not solved). This way the second goal should be to make the developed algorithm wind disturbance tolerant.

4.1 Sensor calibration and measurement selection

Onboard the Ultrastick 25e aircraft a µNAV sensor unit [17] is used. It is a strapdown IMU with angular rate, magnetic (magn.), acceleration (acc.), static and dynamic pressure data completed with a GPS receiver. It provides IMU data with 50Hz and GPS with 4Hz. The calibration procedure of the sensor is published in [BCI+11a BCI+11b].

The performance of an estimator strongly depends on the quality of measurements. The used coordinate axes are shown in figure 4.1 (considering the above mentioned strap-down IMU and GPS).

![Coordinate Systems](image)

Figure 4.1: The used coordinate systems with the possible measurements following standard aerospace notation [73]

\((X^B, Y^B, Z^B)\) is the moving body, \((X^E, Y^E, Z^E)\) is the inertial earth and \((X^N, Y^N, Z^N)\) is the normal coordinate system (coord. sys.). The latter has parallel axes with the earth coord. sys. but it is centered at the origin of the body coord. sys. The earth coord. sys. is assumed to be a fixed, non rotating frame because small UAVs
can not fly large distances (non rotating flat earth model). This can be done if one uses low-cost gyros, because gyro error is much more larger then the effect of earth rotation and motion according to [30] p. 154. It is assumed that $X^E, Y^E, Z^E$ coincides with the NED (North-East-Down) frame (which is also fixed). For A/C attitude estimation at least two independent vector measurements are required (as also stated in [5, 39]).

The Earth’s magn. vector ($h^E \in \mathbb{R}^3$) is assumed to be constant during the 10-15 min flight time of a small UAV and its measurement ($h^B$) is affected only by noises if the sensor is far from any power electronics (such as induction motors).

The gravity vector ($e^E_G \in \mathbb{R}^3$) is measured with the acceleration sensor, but the measured body acceleration ($a^B$) is corrupted not only with noises but also with dynamic effects in the air (see (4.1)). In the equation and the calibration procedure it was assumed that the accelerometers measure the gravity with positive sign, meanwhile the international convention is to use negative sign.

\[
a^B = -\left(\frac{dV^B}{dt} + \omega^B \times V^B\right) + e^B_G
\]

In (4.1) $V^B$ is the A/C Earth relative velocity vector in body frame, $\omega^B$ is the A/C angular velocity vector in body frame and $e^B_G$ is Earth’s gravity vector in body frame. Hence $e^B_G$ is corrupted with two terms in non stationary flight. This is illustrated in figure 4.2 where the absolute value of acc. ($|a^B|$) is only constant before take off and after landing. So, acc. can be used only on the ground. It is worth to note that measuring aircraft IAS makes it possible to apply an approximate $\omega^B \times V^B$ correction on the measured acceleration (assuming no wind disturbance) to get closer to $e^B_G$ similarly as in [53].

![Figure 4.2: GPS azimuth angles and absolute acceleration](image)

Considering the above statements a third measurement is required which can be used in more dynamic situations. In figure 4.1 $V^B$ can be transformed into earth frame assuming
known transformation matrix (the transformed vector $V^E$ can be measured by the GPS receiver) and projected onto the horizontal plane ($X^E - Y^E \to V^E$). This projection gives the Earth relative horizontal velocity which characterizes the $\chi$ course angle (GPS azimuth angle) of the A/C. So, GPS azimuth angle can be calculated from the GPS velocity measurements as in (4.2).

$$\chi = \psi_{GPS} = \arctan \left( \frac{v_E}{v_N} \right)$$

Figure 4.3: The horizontal wind triangle

It should be noted that this azimuth angle differs from the true azimuth angle $\psi$ by the angle of sideslip ($\beta$) and wind disturbance as shown in figure 4.3 (see also [6] chapter 2). $V'_a$ and $V'_w$ are the horizontal projections of A/C air relative velocity and wind velocity respectively in earth frame and $\psi$ is A/C azimuth angle. It can be seen that the GPS azimuth angle is the sum of real azimuth angle, sideslip angle and wind disturbance effect. Later in section 4.5 it will be pointed out that this corrupted azimuth angle also corrupts the roll and pitch angles in the filter so some correction should be applied. In section 4.5 wind estimation and correction will be introduced to solve this problem.

The drawback of using GPS velocity for azimuth angle measurement is the need of sufficient velocity as figure 4.2 shows that (acceptable $\psi_{GPS}$ is obtained only in air between take-off and landing).

Thus acc. and GPS azimuth angle can be used as complementary measurements on ground and in air. This leads to the concept of a multi-mode estimator, which switches
between the different measurements representing a loosely coupled INS/GPS solution [30]. According to the above discussion, the estimator will use the following measurement vectors with the given dimensions [dim]:

1. A/C angular rate [3]
2. earth magnetic vector [3]
3. A/C acceleration [3]
4. GPS azimuth angle from velocity [1]
5. IAS [1]

Two measurement configurations will be considered:

1. angular rate + magnetic vector + acceleration
2. angular rate + magnetic vector + GPS velocity

Dealing with measurements also means that one has to account for the possible disturbances on the measured quantities. The most probable errors are sensor bias and noise, scale factor errors can also be considered [78]. In the project [21] the sensors are well calibrated including also temperature compensation so, it can be assumed that the scale factor errors are negligible. A decision should be made about the consideration of other errors. Angular rate biases are usually considered and estimated together with the consideration of rate sensor noise [10, 42, 56, 58, 62]. So, angular rate bias estimation will be considered. Magnetometers are calibrated considering bias effects [BCI+11b] and it is found that these effects does not change with time, so magnetometer biases are neglected. It is pointed out in [58] that the accelerometer biases are not observable from acceleration and magnetic measurements which is one of the filter’s working modes. So they can not be considered in the estimator. In [56] it is stated that the velocity values from the GPS are drift free so there is also no need to estimate GPS velocity bias values. The structure of the IAS sensor calibration guarantees that the initial bias is removed from it [BCI+11b]. There is no time dependent bias experienced in IAS measurement, so bias estimation is not required. Finally the considered measurement disturbances are the following (for the noise terms, the zero mean Gaussian white noise assumption is used):

1. A/C angular rate: bias and Gaussian white noise
2. earth magnetic vector: Gaussian white noise
3. A/C acceleration: Gaussian white noise
4. GPS velocity: Gaussian white noise
5. IAS: a low-pass filter is applied on it, before using it in the correction of acceleration
Considering the angular rate biases an examination is done to check if they change highly during a flight. If not, their estimation is not required because they can be determined during the initialization of the filter (see the next section). Table 8.1 in Appendix 8.17 summarizes the change of bias data from 8 manual flights. The table includes the changes of bias value during the flight considering the before take-off value as 100% and publishing the percentage changes relative to it (calculated after landing). It is obvious from these values that the yaw rate bias changes are much more smaller than the others, so the yaw rate bias can assumed to be constant. This way the number of the required filter states and so the filter computational demand can be decreased. That’s why only the roll and pitch rate biases are considered in the filter equations.

Another advantage is the avoidance of problems related to flying circular paths. In this case the filter presented in [42] tends to estimate the whole yaw rate of the aircraft as bias and gives false Euler angles. This can possibly happen in other estimator solutions also if the yaw rate bias is considered.

Following the decision on measurements and sensor models, the working modes of the multi-mode estimator and the switching strategies between them are defined.

4.2 Working modes and switching strategies

The proposed working modes of the filter are defined as follows considering the applicable measurements and emergency mode (without GPS) also:

1. MODE 1: Initialization of Euler angles based on magn. and acc. measurements (when aircraft stands undisturbed on the ground). This is required, to have as accurate initial values as possible. The calculation of Earth’s magnetic vector components (in earth frame) is also done to obtain their actual values. They are assumed to be constant and required also in the subsequent modes. The rate gyro bias values are also initialized.

2. MODE 2: Ground mode before take-off. Continuous update of estimates based on magn. and acc. measurements.

3. MODE 3: Aerial mode with GPS. Continuous update of estimates based on magn. and GPS measurements during flight.

4. MODE 4: Aerial mode without GPS. Continuous update of estimates based on magn. and IAS corrected acc. measurements during flight.

5. MODE 5: Ground mode after landing. Continuous update of estimates based on magn. and acc. measurements.

The described estimator modes of operation represent a loosely coupled INS/GPS solution. The switching conditions between the different modes are defined as follows:

1. MODE 1 to MODE 2: MODE 2 starts automatically after initialization which lasts for 10 seconds. During 10 seconds 500 data points can be collected with the 50Hz sensor, which is a proper amount to calculate mean values from it.
2. MODE 2 to MODE 3: The switching happens when \( a_x \) negative acceleration is below and \( \delta_{th} \) is above a threshold, and the estimator is in MODE 2. Here the goal is to establish the time of take-off. This is characterized by large longitudinal acceleration (in absolute value) together with full throttle position. The fact of take-off is also characterized by IAS reaching the region over the stall speed, but from the acceleration the take-off can be detected earlier as figure 4.4 shows that. The figure is enlarged to better show the acceleration values that’s why the larger IAS values can not be seen (but the change of IAS from zero to nonzero can be well detected).

![Figure 4.4: Longitudinal acceleration and IAS profiles during take-off](image)

The throttle threshold is selected to be 80% considering cases without maximum throttle. The acc. switching threshold can be selected by examining the longitudinal acc. \( a_x \) pattern of several flight tests. Figure 4.4 shows one \( a_x \) pattern during take-off. It can be seen that before take-off, the \( a_x \) acceleration is about \(-1.96 m/s^2\) (-0.2g) because of the approximately 9deg pitch angle of the tail wheeled Ultrastick aircraft. During take-off large negative acceleration values occur. A threshold should be selected which is surely above the noises and ground handling disturbances. Such thresholds are summarized in Table 4.1 for eight different flights. From the table \(-0.32g\) \((-3.14 m/s^2\) can be selected as threshold. If \( a_x \) is below this, the take-off is in progress. This is a good threshold also for the case in figure 4.4. The third condition (checking of MODE 2 operation) is required to avoid in-flight switching from MODE 4 when aircraft does maneuvers with large \( a_x \) acc. and maximum throttle.

3. MODE 3 to MODE 4: This is required if there is no GPS signal or measurements are corrupted for at least 3 seconds (dwell time constraint). The validity of GPS
signal is examined in multiple steps. At first the range of valid GPS measurement is checked. The second possible type of GPS error is to give constant data for a long time, so GPS data are considered to be valid if the actual latitude differs from the previous value or the actual longitude differs from the previous one. The dwell time constraint is required to avoid frequent switching. It is selected with respect to the divergence properties of the filter with purely magn. measurements which are the only used measurements before switching to MODE 4. It is well known that the system is not observable from one vector measurement, so the filter will diverge. If the user waits too long with the switching it takes a long time to converge again in MODE 4 as simulation tests point out.

4. MODE 4 to MODE 3: If valid GPS data arrives again.

5. MODE 3 to MODE 5: For this change, the absolute GPS velocity and flight time are considered. If the velocity is below \(0.2 \text{ m/s}\) and flight time is above \(120 \text{ s}\) the system switches. The flight time limit is required because otherwise the system switches into MODE 5 during take-off (when the velocity is still low). The velocity limit is selected from several manual flight tests.

Unfortunately, it is not possible to switch from MODE 4 to MODE 5 because GPS is not available in MODE 4. Later this can be solved considering the IAS and vertical acceleration instead of GPS velocity.

After defining the modes and the switching rules, the system equations are constructed in the next section.

### 4.3 System equations and the steps of the algorithm

In this section the system equations will be derived and listed including the decision about the representation of A/C orientation. The first part includes the equations used in the initialization step (MODE 1). The second part derives the filter equations used in the subsequent modes and checks the observability of the system.

#### 4.3.1 Filter initialization

In the initialization step (MODE 1), the initial Euler angles and rate gyro biases are calculated together with the components of Earth’s magnetic vector. The mean measured values of acc., magn. and angular rate data are used to do this meanwhile aircraft stands undisturbed on the ground. These values should be calculated using a recursive formula to avoid unnecessary memory usage to store the input values.
Here, \( \mathbf{v} \) represents the vector of measured data, \( \bar{\mathbf{v}} \) is the computed mean and \( k \) is the time index of a sampled variable. In the following steps it is assumed that the IMU is perfectly aligned in body frame and there are no biases on the measured acc. and magn. values.

The initial angular rate sensor biases are the calculated mean angular rate values (the real A/C angular rate is zero during data acquisition neglecting the Earth rotation rate). The initial roll (\( \phi \)) and pitch (\( \theta \)) angles can be calculated from the mean acc. values as follows considering the fact that the measured acc. in body frame is the Earth’s gravity vector:

\[
\phi = \arctan \left( \frac{\bar{a}_y}{\bar{a}_z} \right), \quad \theta = \arcsin (-\bar{a}_x)
\]  

(4.4)

Here, \( \bar{a}_x, \bar{a}_y, \bar{a}_z \) represent the components of the mean body acc. vector \( \bar{\mathbf{a}}^B \). Using these angles the mean body magnetic vector (\( \bar{\mathbf{h}}^B \)) can be transformed into earth frame (\( \bar{\mathbf{h}}^E \)) assuming zero yaw angle. This way, the magnetic yaw angle (\( \psi_h \)) can be calculated from the horizontal magnetic components in earth frame. Finally, the magnetic yaw angle corrected with magnetic declination will be the real yaw angle (\( \psi \)) of the aircraft (assuming that magnetic north plus declination gives geographic north). The magnetic declination \( D \) is a fixed parameter in the autopilot program through a year. It is obtained and fixed from World Magnetic Model at the beginning of every year (flights are conducted every time at the same place and the yearly change of declination is about \( 0.1^\circ \) at Hungary’s longitude and latitude).

\[
\bar{\mathbf{h}}^E = T^T_{\psi}|_{\psi=0}T^T_\theta T^T_\phi \bar{\mathbf{h}}^B
\]

\[
\psi_h = \arctan \left( \frac{-\bar{\mathbf{h}}^E(2)}{\bar{\mathbf{h}}^E(1)} \right), \quad \psi = \psi_h + D
\]  

(4.5)

In (4.5) finally, the Earth’s magnetic vector (\( \mathbf{h}^E \)) is calculated in earth frame. This is used later in the measurement equations (see (4.12)) and is assumed to be constant throughout the whole flight of the aircraft. \( T_\phi, T_\theta, T_\psi \) are the rotation matrices related to the roll, pitch and yaw Euler angles respectively (see Appendix 8.18 for details). \( ()^T \) denotes the transpose of a matrix. \( \bar{\mathbf{h}}^E(1) \) and \( \bar{\mathbf{h}}^E(2) \) are the first and second coordinates of the \( \bar{\mathbf{h}}^E \) vector, \( D \) is the magnetic declination.

### 4.3.2 Filter equations and system observability

The first stage of estimator design is to select the theoretical framework. The Extended Kalman Filter (EKF) is a well known and widely used [36, 56, 58, 77, 81] method for A/C state estimation. [36] points out that in attitude estimation it is at least as accurate as other (unscented and particle filter) algorithms but computationally less expensive. So, the EKF framework is selected.
The choice of orientation representation has large impact on the solution. In aerospace applications, mostly Euler angles [1, 24, 56, 77], quaternions [42, 58, 81] or representations involving the direction cosine matrix (DCM) [5, 39, 55] are used. The decision between them should be made considering their effects on accuracy. In this work, the decision between Euler angles and quaternions is done, the DCM possibility is not considered. The differential equations for Euler angles and quaternion representations are repeated here from [79] considering measurement noise and bias in the angular rate (of the body frame) components:

$$\begin{bmatrix}
\dot{\phi} \\
\dot{\theta} \\
\dot{\psi}
\end{bmatrix} = \begin{bmatrix}
\bar{p} + \tan \theta \sin \phi \cdot \bar{q} + \tan \theta \cos \phi \cdot \bar{r} \\
\cos \phi \cdot \bar{q} - \sin \phi \cdot \bar{r} \\
\frac{\sin \phi}{\cos \theta} \bar{q} + \frac{\cos \phi}{\cos \theta} \bar{r}
\end{bmatrix} = f(\bar{p}, \phi, \theta) \quad (4.6)
$$

In (4.6), \(\bar{p}\) represents the real roll rate which means the measured roll rate corrected with the bias and zero mean Gaussian white noise of the sensor, that is, \(\bar{p} = p - b_p - v_p\). The same convention is used for the pitch (\(q\)) and yaw (\(r\)) rate variables and \(\rho\) denotes the parameter vector \(\rho = [\bar{p} \quad \bar{q} \quad \bar{r}]^T\).

$$\dot{\varrho} = -\frac{1}{2} \begin{bmatrix}
0 & \bar{p} & \bar{q} & \bar{r} \\
-\bar{p} & 0 & -\bar{q} & \bar{r} \\
-\bar{q} & \bar{r} & 0 & -\bar{p} \\
-\bar{r} & -\bar{q} & \bar{p} & 0
\end{bmatrix} \varrho = A(\bar{p})\varrho \quad (4.7)
$$

In (4.7) \(\varrho = [\varrho_0 \quad \varrho_1 \quad \varrho_2 \quad \varrho_3]^T\) is the unit quaternion representing the same rotation as the Euler angles above.

(4.6) shows that Euler angles have a singularity property at \(\theta = \pi/2 + k\pi \in \mathbb{Z}^+\). This is usually avoided in non-aerobatic flight, but means a risk in the estimation. Another issue is the implementation of the estimator which requires as simple equations as possible to provide real time applicability on microcontrollers as was considered in the development of the control algorithms also. In (4.6) the trigonometric functions of Euler angles are applied, while in (4.7) only the simple linear combinations of the angular rate components with the quaternion elements are taken.

From the continuous time (CT) nonlinear differential equations (4.6), (4.7) a discrete time (DT) linearized dynamics should be created to be applicable in a discrete time EKF. [61] points out that the use of Heun scheme (trapezoidal integration) can improve the accuracy of EKF. For a nonlinear, state (\(x\)) and parameter (\(\rho\)) dependent and noise (\(n\)) affected system, the Heun scheme results as follows [61]:

$$\dot{x} = f(x, \rho, n)$$

$$x_{k+1} \approx x_k + \frac{\dot{x}_k + \dot{x}_{k+1}}{2} \Delta t = x_k + f(x_k, \rho_k, n_k) + f(x_{k+1}, \rho_{k+1}, n_{k+1}) \Delta t \quad (4.8)$$

Here, \(\Delta t\) represents the discrete time step. Equation (4.8) is usually hard to solve because of the nonlinear relation with the unknown \(x_{k+1}\) state through \(f(x, \rho, n)\). This
is the case with the Euler angle dynamics in (4.6). However, considering the quaternion dynamics (4.7), a nice closed form solution can be derived as follows.

(4.7) can be reorganized expanding \( \bar{p}, \bar{q} \) and \( \bar{r} \) to select \( p, q, r \) measured angular rates and the bias and noise terms:

\[
\dot{x} = A_1(\rho) x + A_2(\rho) b + L_1(\dot{\rho}) v^e = A(\rho, b) x + L_1(\dot{\rho}) v^e
\]

where

\[
\rho = \begin{bmatrix} p & q & r - b_r \end{bmatrix}^T, \quad b = \begin{bmatrix} b_p & b_q \end{bmatrix}^T, \quad v^e = \begin{bmatrix} v_p & v_q & v_r \end{bmatrix}^T
\]

(4.9)

Here, \( A_1(\rho), A_2(\rho), A(\rho, b) \) and \( L_1(\dot{\rho}) \) are presented in Appendix 8.18. In (4.9) \( \rho \) is the measured parameter vector (\( b_r \) is known from initialization), \( b \) is the vector of roll and pitch rate sensor biases and \( v^e \) is the vector of sensor Gaussian noises. The dynamics of the slowly varying (theoretically constant) biases can be modelled as a system driven by zero-mean Gaussian white noise:

\[
\dot{b} = v^b
\]

where

\[
v^b = \begin{bmatrix} v_{b_p} & v_{b_q} \end{bmatrix}^T
\]

(4.10)

Considering (4.8), (4.9) and (4.10) and applying some manipulation the discrete time dynamic equations of the quaternion and bias dynamics result as:

\[
\begin{bmatrix} \dot{\rho}_{k+1} \\ \dot{b}_{k+1} \end{bmatrix} = \begin{bmatrix} \left(M^+_{k+1}\right)^{-1} M^-_k & \frac{\Delta t}{2} \left(M^+_{k+1}\right)^{-1} A_2(\dot{\rho}_k) \\ 0 & I_2 \end{bmatrix} \begin{bmatrix} \rho_k \\ b_k \end{bmatrix} + \begin{bmatrix} \Delta t \left(M^+_{k+1}\right)^{-1} L_1(\dot{\rho}_{k+1}) & 0 \\ 0 & \Delta t I_2 \end{bmatrix} \begin{bmatrix} \dot{v}^e_{k+1} \\ \dot{v}^b_{k+1} \end{bmatrix}
\]

(4.11)

Here, the state of the system consists of the quaternion and bias vector (\( x_k \in \mathbb{R}^6 \)). For the details of the derivation see Appendix 8.18. If a rotational transformation is represented by a quaternion, there is a unit norm constraint on it. The satisfaction of this constraint should be provided by normalization in every time step, because system dynamics does not guarantee unit norm. After deriving the state equations, the output equations for the different measurements should be derived together with the Jacobian matrices required in EKF dynamics.

For the magn. measurement, the nonlinear equation and its Jacobian are as follows:

\[
y^H = h^B = T^{BE}(\rho) h^E = h_1(\rho, h^E), \quad C_1 = \frac{\partial h_1(\rho, h^E)}{\partial \rho}
\]

(4.12)

Here, \( T^{BE}(\rho) \) represents the quaternion based rotation matrix from earth to body frame (see Appendix 8.18) (\( y^H \in \mathbb{R}^3, \ C_1 : \mathbb{R}^4 \rightarrow \mathbb{R}^3 \)).

For the measurement of Earth’s gravity vector in body frame (static case):

\[
y^a = a^B \approx c^B_G = T^{BE}(\rho) c^E_G = h_2(\rho), \quad C_2 = \frac{\partial h_2(\rho)}{\partial \rho}
\]

(4.13)
\( (y^a \in \mathbb{R}^3, \ C_2 : \mathbb{R}^4 \to \mathbb{R}^3) \). In the dynamic case (MODE 4) \( y^a \) is approximately corrected with the IAS according to (4.1) applying \( \epsilon_G^B \approx \bar{a} + \omega^B \times \begin{bmatrix} IAS \end{bmatrix} \).

The nonlinear output equation for GPS azimuth angle and the Jacobian are:

\[
y^{GPS} = \psi_{GPS} = \arctan\left(\frac{v_E}{v_N}\right) = \arctan\left(\frac{T^{BE}(1,2)}{T^{BE}(1,1)}\right) = h_3(\varphi), \quad C_3 = \frac{\partial h_3(\varphi)}{\partial \varphi}
\]

(4.14)

Here, \( T^{BE}(1,i) \) are selected elements from the rotation matrix (\( y^{GPS} \in \mathbb{R}, \ C_3 : \mathbb{R}^4 \to \mathbb{R} \)).

The general structure of the linearized, DT output equation is shown below (with \( w \) zero-mean Gaussian measurement noise vector).

\[
y_{k+1} = C_{k+1}x_{k+1} + w_{k+1}
\]

(4.15)

The elements of this output equation in the different modes are listed below

- **MODE 2,4,5:** \( y^{(2,4,5)} = \left[(y^H)^T \ (y^g)^T\right]^T \in \mathbb{R}^6 \) and \( C^{(2,4,5)} = \begin{bmatrix} C_1 & 0 \\ C_2 & 0 \end{bmatrix} : \mathbb{R}^6 \to \mathbb{R}^6 \)

- **MODE 3/A:** aerial mode with magn. & GPS correction \( y^{(3A)} = \left[(y^H)^T \ y^{GPS}\right]^T \in \mathbb{R}^4 \) and \( C^{(3A)} = \begin{bmatrix} C_1 & 0 \\ C_3 & 0 \end{bmatrix} : \mathbb{R}^6 \to \mathbb{R}^4 \)

- **MODE 3/B:** aerial mode with only magn. correction if GPS is invalid for short duration or not available (due to the different update rates of IMU and GPS). \( y^{(3B)} = y^H \in \mathbb{R}^3 \) and \( C^{(3B)} = \begin{bmatrix} C_1 & 0 \end{bmatrix} : \mathbb{R}^6 \to \mathbb{R}^3 \)

After deriving the state and output equations, the observability of the system should be examined. This is done by checking the observability of the nonlinear system according to [41].

The observability of the nonlinear system for every applied input (measurement) combination is checked considering the observability distribution in several points of the state space. A set of Euler angles is used to grid the quaternion space (using Euler angle to quaternion conversion) to provide a systematic test:

\[
\phi = \begin{bmatrix} -90 & -80 & -70 & \ldots & 90 & -45 & 45 \end{bmatrix} [\text{deg}]
\]

\[
\theta = \begin{bmatrix} -60 & -50 & -40 & \ldots & 60 & -45 & 45 \end{bmatrix} [\text{deg}]
\]

\[
\psi = \begin{bmatrix} -180 & -170 & -160 & \ldots & 180 & -45 & 45 \end{bmatrix} [\text{deg}]
\]

These values are selected to cover the flight envelope of a UAV with limited aerobatic capability (no loops and inverted flight). The observability distribution can be calculated according to [41] pages 69-76. The method involves symbolically calculating the
observability distribution, then to calculate its numeric values with the given gridding of the state space (the pure symbolic calculations could not include zero values and so, the system is always observable from any measurement). The following measurement configurations are examined (with the six dimensional state space (4 quaternion elements and two biases)):

- **MODE 3/B**: with nonzero angular rates and biases the rank of the observability distribution (od.) is 5 in every state space point, so the system is not observable from pure magnetic measurements
- **MODE 2,4,5**: with nonzero and also with zero angular rates and biases the rank of od. is 6 in every state space point, so the system is observable from magnetic and acceleration measurements
- **MODE 3/A**: with nonzero and also with zero angular rates and biases the rank of od. is 6 in every state space point, so the system is observable from magnetic and GPS measurements

Consequently, the system is not observable from the pure magnetic measurements according to the nonlinear observability test. The last test done is to delete the bias values from the state vector and check the observability of the resulting four state system from pure magnetic measurements. Nonzero angular rates and biases are substituted, the rank is 3 in every state space point, so the reduced system is also not observable.

As a summary, it can be stated that the system is not observable from pure magnetic measurements and, so the GPS or acc. measurement is also required to provide convergence. Between the GPS corrections some divergence can be observed (because pure magnetic measurement is used), but the 4Hz GPS frequency is enough to successfully limit the possible errors.

A hybrid automata representation of the algorithm was constructed in [BB11a]. After deriving the system and measurement equations and checking observability, the tuning and testing of the algorithm can be done.

### 4.4 Tuning and testing of the algorithm

Tuning and testing is performed as an iterative process. The final filter weights (noise covariance values) are selected through several iterations, and it has turned out that MODE 4 requires magnetic and acceleration noise covariances different from MODE 2 and 5. This is obvious considering the different function of these MODEs.

At first, tuning and testing of the aerial part using hardware-in-the-loop (HIL) simulation data with added noise and bias (but without wind disturbance) is done. Then GPS data sections from real flight data are artificially deleted and results are compared with results from fully available GPS data. Considering the differences MODE 4 aerial mode is tuned. The next step is the $C - mex$ (Matlab) file implementation of the code originally coded in .m Matlab files. Finally, implementation onboard the microcontroller of the Ultrastick 25e aircraft is done and HIL and real flight tests are performed.
4.4.1 Tuning and testing using HIL data

The HIL simulation environment used for tuning is similar to the one presented in [70]. The noise covariance weighting matrices of the EKF are \( N \) for system and \( W \) for measurement noise. \( N \) should be formulated from the noise covariance matrix of angular rate measurements \( R_a \) and the fictitious noise covariance of angular rate sensor biases \( k_b \) (same for roll and pitch) in the following form:

\[
N = \begin{bmatrix}
R_a & 0_{3 \times 1} & 0_{3 \times 1} \\
0_{1 \times 3} & k_b & 0 \\
0_{1 \times 3} & 0 & k_b
\end{bmatrix}
\]

Here, \( 0_{i \times j} \) denotes an \( i \times j \) zero matrix. \( W \) depends on the actual used measurements of the filter and is as follows:

- **MODE 2, 4, 5**: \( W = \begin{bmatrix} R_{nH} & 0_{3 \times 3} \\ 0_{3 \times 3} & R_{na} \end{bmatrix} \)
- **MODE 3**: \( W = \begin{bmatrix} R_{nH} & 0_{3 \times 1} \\ 0_{1 \times 3} & R_{GPS} \end{bmatrix} \)

The basic noise covariance matrices obtained from \( \mu NAV \) sensor data are listed in Appendix [8.18]. These basic matrices are scaled during the tuning process of the EKF to get more accurate results. They are determined leaving the sensor undisturbed and collecting data. Then from the collected data the means are removed and the noise covariance matrices are calculated. \( R_a \) is for the angular rate sensors considering \( rad/s \) measurement unit. \( R_{nH} \) is for the magnetic values considering the normalized measured magnetic vector, since it is also normalized in the estimator. \( R_{na} \) is for the acc. measurements normalized with \( g \) gravitational constant. \( R_{GPS} \) is for the GPS azimuth angle calculated from GPS velocity measurements. The noise covariance of the calculated azimuth angle is determined from undisturbed sensor data. \( k_b \) is used as a tuning parameter.

After the tuning of the parameters the resulting noise covariance matrix components in aerial mode are \( R_a, R_{nH} = R_{nH} \cdot 2000, R_{na} = R_{na} \cdot 100, R_{GPS} \) and \( k_b \). Tests with the tuned algorithm are conducted by adding white noises to all, and biases to the angular rate components in HIL test data (the noises are generated considering the noise covariance matrices of the real \( \mu NAV \) sensor). The applied bias values are the following: \( b_p = -0.01 \ rad/s, b_q = 0.008 \ rad/s, b_r = 0.011 \ rad/s \). The initial Euler angle errors are zero in the first test, but nonzero in the second (the nonzero errors are \( -1^\circ \) for \( \phi \), \( +1.25^\circ \) for \( \theta \) and \( -2^\circ \) for \( \psi \)).

The initial estimated biases are different from the real ones in all two cases (\( b_{\phi 0} = -0.007 \ rad/s, b_{\theta 0} = 0.004 \ rad/s, b_{\psi 0} = 0.011 \ rad/s \)) except for \( b_r \). Because the results on the plots are very similar to the HIL results published in the next section (Section 4.5) only the convergence of the estimated angular rate biases is plotted in Fig. 4.5. The maximum absolute estimation errors for the two cases are in Table 4.2.

Table 4.2 shows that the errors are larger with nonzero initial estimation errors, that’s why it is important to do as accurate initialization as possible. The azimuth angle maximum errors are about twice as large as for the other angles. Compared with \[81\] where the azimuth angle error is about five times larger, this filter shows better performance.

The results show that the tuning is successful, the estimates are very good, including the biases, which converge after 5-10 second transient to the given real values.
Table 4.2: Maximum absolute estimation errors [deg]

<table>
<thead>
<tr>
<th>Euler errors</th>
<th>φ</th>
<th>θ</th>
<th>ψ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero initial errors</td>
<td>0.67</td>
<td>0.427</td>
<td>1.21</td>
</tr>
<tr>
<td>Nonzero initial errors</td>
<td>1.65</td>
<td>1.595</td>
<td>2.37</td>
</tr>
</tbody>
</table>

![Rate gyro biases](image)

Figure 4.5: Estimated bias from HIL data

4.4.2 Tuning and testing using real flight data

After tuning and testing based-on HIL data the algorithm is tuned and tested on real flight data. MODE 4 is tuned by deleting sections of GPS data from the measurements and comparing the resulting angles to the GPS based solution. It turned out that the \( R_{nH} \) and \( R_{na} \) weights should be modified to get better results. Their best values in MODE 4 are: \( R_{nH} = R_{nH} \cdot 20 \), \( R_{na} = R_{na} \cdot 10^5 \) this means a decrease in magnetic and an increase in acc. noise covariance. This is reasonable, because the acc. is more unreliable during flight. Unfortunately the flight data does not include Euler angles, so only comparison to Euler angle values obtained from data measured on ground can be done. A special test is conducted when the aircraft is rotated on the ground in 45 deg azimuth angle steps and leaved undisturbed for about ten seconds after every rotation before take-off. From these ten second data the Euler angles can be calculated off-line as in the initialization step of the estimator. After landing, this rotational test is not done, but the off-line angles are also calculated from the undisturbed data. The estimated angles can be compared with these off-line angles (calculated in a different way) and so checked before take-off and after landing. If the values are close and after landing there is no sudden change in the angles it can be assumed that the values during the flight are also good (based-on HIL test results where the in-flight angles are also correctly estimated as will be shown later). The Euler angle comparison is plotted in Fig. 4.6, 4.7 and 4.8.
‘Measured’ means the off-line values, the values before take-off are plotted in the upper, and the values after landing in the lower figures. The estimated angles are very close to the off-line ones. The approximately 45 deg steps can be seen in the \( \psi \) azimuth angle plot. There are no sudden changes in \( \phi \) and \( \psi \) at the end of flight, there is some change in \( \theta \), so possibly this value is not estimated with the same accuracy.

![Figure 4.6: Estimated \( \phi \) with off-line values](image)

![Figure 4.7: Estimated \( \theta \) with off-line values](image)

![Figure 4.8: Estimated \( \psi \) with off-line values](image)

After doing the \( C - mex \) file implementation and tests the \( C \) file version is created. The results of extensive HIL testing onboard the microcontroller will be summarized in the next section including comparison to the algorithm proposed in [42] and considering possible wind disturbances.

In flight comparison of Euler angles could not be done because the lack of independently measured (or calculated) Euler angle data, but successful flight tests are conducted with Euler angle tracking low level, and waypoint tracking high level controllers [BB11b, BB12]. The tested Euler angle tracking controllers are the LQ optimal and LQ minimax opatimal solutions derived here.
4.5 Detailed HIL testing including wind effects

For these tests the original HIL simulation developed in [70] is modified to include the handling of ground effects during take-off and landing based on fictitious stiffness and damping parameters and the roll and slip resistances. This makes it possible to start with an aircraft standing on the ground, do take-off and at the end of flight perform landing. So all of the modes of the estimator can be tested from initialization to ground mode after landing.

The algorithm proposed in [42] is also implemented to make it possible to compare results. This algorithm is based on magnetic and acc. measurements. It is assumed that the acc. sensors measure the direction of gravity, but IAS based correction on the measured values is applied.

Six tests are conducted with each estimator, applying sensor noises (no noise = NN, noise N), sensor biases (no bias = NB, bias = B) and wind disturbance (no wind = NW, wind = W). Independent noise on each sensor channel is assumed. The applied biases, noise covariances, and wind disturbance vector are listed in Appendix 8.18.

The estimator test results are characterized by the minimum, maximum and mean absolute errors of each estimated Euler angle relative to the real value obtained from the HIL simulation. Absolute errors are used only in the calculation of mean values. The results are summarized in Table 4.3 for the algorithm from [42] and in Table 4.4 for the present algorithm. The test flight path started with take-off, then a right turn is made, then horizontal flight with roll doublet maneuvers (using the low level autopilot), then an eight shape in horizontal plane, then again horizontal flight with roll doublet maneuvers, finally landing. This is the trajectory in all cases.

Table 4.3: HIL test results for the algorithm from [42]

<table>
<thead>
<tr>
<th>Case</th>
<th>Value [deg]</th>
<th>φ</th>
<th>θ</th>
<th>ψ</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>MIN</td>
<td>-41.23</td>
<td>-41.55</td>
<td>-49.88</td>
</tr>
<tr>
<td>NB</td>
<td>MAX</td>
<td>63</td>
<td>31.12</td>
<td>49.43</td>
</tr>
<tr>
<td>NW</td>
<td>MEAN</td>
<td>14.62</td>
<td>8.39</td>
<td>17.74</td>
</tr>
<tr>
<td>NN</td>
<td>MIN</td>
<td>-31.17</td>
<td>-54.29</td>
<td>-49.93</td>
</tr>
<tr>
<td>NB</td>
<td>MAX</td>
<td>40.75</td>
<td>20.99</td>
<td>49.92</td>
</tr>
<tr>
<td>W</td>
<td>MEAN</td>
<td>7.57</td>
<td>6.49</td>
<td>8.46</td>
</tr>
<tr>
<td>N</td>
<td>MIN</td>
<td>-35.16</td>
<td>-42.64</td>
<td>-22.82</td>
</tr>
<tr>
<td>NB</td>
<td>MAX</td>
<td>43.95</td>
<td>24.63</td>
<td>49.8</td>
</tr>
<tr>
<td>NW</td>
<td>MEAN</td>
<td>8.4</td>
<td>6.98</td>
<td>6.4</td>
</tr>
<tr>
<td>N</td>
<td>MIN</td>
<td>-34.64</td>
<td>-43.94</td>
<td>-49.8</td>
</tr>
<tr>
<td>NB</td>
<td>MAX</td>
<td>45.03</td>
<td>22.65</td>
<td>49.7</td>
</tr>
<tr>
<td>W</td>
<td>MEAN</td>
<td>8.37</td>
<td>8.63</td>
<td>9.77</td>
</tr>
<tr>
<td>N</td>
<td>MIN</td>
<td>-37.8</td>
<td>-45.09</td>
<td>-49.75</td>
</tr>
<tr>
<td>B</td>
<td>MAX</td>
<td>55</td>
<td>30.08</td>
<td>48.43</td>
</tr>
<tr>
<td>NW</td>
<td>MEAN</td>
<td>9.8</td>
<td>14.92</td>
<td>18.06</td>
</tr>
<tr>
<td>N</td>
<td>MIN</td>
<td>-26.36</td>
<td>-52.43</td>
<td>-49.92</td>
</tr>
<tr>
<td>B</td>
<td>MAX</td>
<td>42.18</td>
<td>10.81</td>
<td>49.96</td>
</tr>
<tr>
<td>W</td>
<td>MEAN</td>
<td>8.51</td>
<td>12.12</td>
<td>12.91</td>
</tr>
</tbody>
</table>
The estimation results with the present algorithm in the N/NB/W case are plotted in figures 4.9, 4.10 and 4.11 (‘Measured’ again means the value calculated in HIL simulation). In the N/B/W case the errors are too large and in the real system the biases are minimal (almost zero) that’s why this case is plotted. It can be seen that there are sometimes large differences between the real and estimated angles. Enlarged plot sections are presented in Appendix 8.19.

Regarding the tabular results the first flight in table 4.3 is somehow defective, giving larger errors then the other cases. Perhaps the maneuver accelerations are larger then otherwise. The mean absolute errors are always smaller with the present algorithm compared to [42]. The min/max values in the last two cases (noise and bias without or with wind) are similar, in the other cases they are smaller with the present algorithm. Considering the wind effects, the windy cases are always worse with the present algorithm than the non-windy ones mainly for the ψ azimuth angle. This is obvious, because the ψ measurement is calculated from the wind corrupted GPS velocity. The cases with sensor biases give unacceptably large errors which points out the importance of precise sensor calibration. With the algorithm from [42] there is no difference between windy and non-windy cases. This is again obvious because it does not uses GPS data.
Table 4.4: HIL test results for the current algorithm

<table>
<thead>
<tr>
<th>Test case</th>
<th>Value [deg]</th>
<th>φ</th>
<th>θ</th>
<th>ψ</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN MIN</td>
<td>-1.539</td>
<td>-2.248</td>
<td>-4.6</td>
<td></td>
</tr>
<tr>
<td>NB MAX</td>
<td>1.676</td>
<td>1.864</td>
<td>2.116</td>
<td></td>
</tr>
<tr>
<td>NW MEAN</td>
<td>0.29</td>
<td>0.56</td>
<td>1.353</td>
<td></td>
</tr>
<tr>
<td>NN MIN</td>
<td>-3.048</td>
<td>-3.412</td>
<td>-12.85</td>
<td></td>
</tr>
<tr>
<td>NB MAX</td>
<td>6.563</td>
<td>5.257</td>
<td>10.86</td>
<td></td>
</tr>
<tr>
<td>W MEAN</td>
<td>1.35</td>
<td>1.119</td>
<td>3.71</td>
<td></td>
</tr>
<tr>
<td>N MIN</td>
<td>-3.3</td>
<td>-5.12</td>
<td>-5.765</td>
<td></td>
</tr>
<tr>
<td>NB MAX</td>
<td>6.054</td>
<td>2.48</td>
<td>1.466</td>
<td></td>
</tr>
<tr>
<td>NW MEAN</td>
<td>0.54</td>
<td>0.369</td>
<td>1.17</td>
<td></td>
</tr>
<tr>
<td>N MIN</td>
<td>-1.796</td>
<td>-7.42</td>
<td>-15.66</td>
<td></td>
</tr>
<tr>
<td>NB MAX</td>
<td>7.54</td>
<td>4.68</td>
<td>12.47</td>
<td></td>
</tr>
<tr>
<td>W MEAN</td>
<td>1.6</td>
<td>1.346</td>
<td>4.76</td>
<td></td>
</tr>
<tr>
<td>N MIN</td>
<td>-18.04</td>
<td>-13.77</td>
<td>-49.92</td>
<td></td>
</tr>
<tr>
<td>B MAX</td>
<td>52.83</td>
<td>49.18</td>
<td>44.58</td>
<td></td>
</tr>
<tr>
<td>NW MEAN</td>
<td>9.449</td>
<td>11.13</td>
<td>3.98</td>
<td></td>
</tr>
<tr>
<td>N MIN</td>
<td>-11.4</td>
<td>-13.08</td>
<td>-46.25</td>
<td></td>
</tr>
<tr>
<td>B MAX</td>
<td>53.24</td>
<td>37.25</td>
<td>47.96</td>
<td></td>
</tr>
<tr>
<td>W MEAN</td>
<td>7.24</td>
<td>9.212</td>
<td>7.012</td>
<td></td>
</tr>
</tbody>
</table>

As an overall conclusion it can be stated that the present algorithm outperforms the other one in all cases, but it is highly dependent on the static wind disturbances and sensor biases. That’s why sensors should be calibrated (bias compensated) carefully and some wind correction should be applied to correct the errors.

4.5.1 Applying wind estimation and correction

The goal of this part is to make it possible to correct the effects of static wind on the estimated Euler angles. Three representative articles about wind estimation are considered. The article [52] uses the detailed description of aircraft dynamics under wind effects to estimate several parameters of the wind field around the aircraft (wind velocity, acceleration and gradient). On the contrary [72] uses a very simple calculation based-on estimated Euler angles and measured GPS velocity, estimating the three components of static wind in earth frame. [15] uses a method based-on the horizontal wind triangle (see figure 4.3). It estimates the direction and magnitude of static wind in the horizontal plane together with the scale factor of the Pitot tube (which is assumed to measure the magnitude of air relative velocity) using a simple EKF (for details see Appendix 8.20).

From the three methods the first is too complicated to be used for wind correction, the second could be good but it is based on Euler angles which are corrupted because of wind, so the method can not be successfully used. The third method uses available measurements and does not rely on Euler angles, so it can be used to correct the attitude estimates. The outputs of this method are the strength (\(V_W\)) and direction (\(\psi_W\)) of wind disturbance (together with Pitot tube scale factor).
The wind correction is done by correcting the GPS velocity components considering \( V_W \) and \( \psi_W \) in the GPS azimuth angle calculation:

\[
\psi_{GPS} = \text{atan} \left( \frac{v_E - V_W \sin(\psi_W)}{v_N - V_W \cos(\psi_W)} \right)
\] (4.16)

This corrected measurement is used in the correction step of the EKF. The wind estimation algorithm is tuned considering HIL simulation results and real flight data. The modified estimator is tested in the same cases in HIL simulation as before. The results are summarized in Table 4.5. Significant improvement can be seen. The min and max values are usually larger then with the non-corrected method (this is because the wind estimator needs time to converge and during this time larger deviations appear), but the mean values are usually smaller. The mean values are better or similar in the first four cases, so the wind effects are successfully compensated. In the cases with nonzero bias, the estimates of the roll and pitch angles are slightly better, while the estimation of the azimuth angle is slightly worse. However, all of the mean errors are unacceptably large in the biased cases. So, the application of wind estimation can only improve the results if a well calibrated sensory system is used. It can not compensate large bias errors in the measurements.

Table 4.5: HIL test results for the current algorithm with wind correction

<table>
<thead>
<tr>
<th>Test case</th>
<th>Value</th>
<th>( \phi )</th>
<th>( \theta )</th>
<th>( \psi )</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>MIN</td>
<td>-4.497</td>
<td>-3.072</td>
<td>-6.417</td>
</tr>
<tr>
<td>NB</td>
<td>MAX</td>
<td>3.238</td>
<td>2.5</td>
<td>3.99</td>
</tr>
<tr>
<td>NW</td>
<td>MEAN</td>
<td>0.558</td>
<td>0.5132</td>
<td>1.5293</td>
</tr>
<tr>
<td>NN</td>
<td>MIN</td>
<td>-5.3259</td>
<td>-3.8225</td>
<td>-8.0385</td>
</tr>
<tr>
<td>NB</td>
<td>MAX</td>
<td>21.89</td>
<td>35.97</td>
<td>46.7</td>
</tr>
<tr>
<td>W</td>
<td>MEAN</td>
<td>0.9571</td>
<td>1.0492</td>
<td>2.534</td>
</tr>
<tr>
<td>N</td>
<td>MIN</td>
<td>-9.8643</td>
<td>-1.3536</td>
<td>-3.7144</td>
</tr>
<tr>
<td>NB</td>
<td>MAX</td>
<td>2.3725</td>
<td>7.4929</td>
<td>26</td>
</tr>
<tr>
<td>NW</td>
<td>MEAN</td>
<td>0.5107</td>
<td>0.3682</td>
<td>1.3717</td>
</tr>
<tr>
<td>N</td>
<td>MIN</td>
<td>-5.33</td>
<td>-6.29</td>
<td>-12.497</td>
</tr>
<tr>
<td>NB</td>
<td>MAX</td>
<td>6.53</td>
<td>13.246</td>
<td>46.82</td>
</tr>
<tr>
<td>W</td>
<td>MEAN</td>
<td>0.7456</td>
<td>0.682</td>
<td>2.2848</td>
</tr>
<tr>
<td>N</td>
<td>MIN</td>
<td>-30.287</td>
<td>-25.7</td>
<td>-47.17</td>
</tr>
<tr>
<td>B</td>
<td>MAX</td>
<td>29.23</td>
<td>16.83</td>
<td>48.53</td>
</tr>
<tr>
<td>NW</td>
<td>MEAN</td>
<td>8.6927</td>
<td>5.9215</td>
<td>6.013</td>
</tr>
<tr>
<td>N</td>
<td>MIN</td>
<td>-18.47</td>
<td>-14.276</td>
<td>-46.548</td>
</tr>
<tr>
<td>B</td>
<td>MAX</td>
<td>20.74</td>
<td>26.626</td>
<td>48.798</td>
</tr>
<tr>
<td>W</td>
<td>MEAN</td>
<td>6.345</td>
<td>7.249</td>
<td>9.621</td>
</tr>
</tbody>
</table>

Similar results as in figures 4.9, 4.10 and 4.11 are plotted in figures 4.12, 4.13 and 4.14 (they represent a different HIL test). Enlarged plot sections are presented in Appendix 8.21.
The figures also show the performance improvement. As a conclusion, it can be stated that the developed estimator enhanced with wind estimation and correction has satisfactory performance and is more suitable in aircraft control if the sensors are well calibrated. This algorithm was successfully applied in all of the real flight tests of the LQ and minimax algorithms.

### 4.6 Summary

This chapter constructs a multi-mode aircraft attitude estimator to be applied in simulations and real flight tests together with the LQ and MM trackers. After summarizing sensor calibration and selecting the used measurements the working modes and switching strategy of the algorithm are defined. The filter equations are derived together with the examination of observability. After describing the tuning and test results, a wind estimation and correction method is introduced to improve accuracy.
Chapter 5

Robustness test of the LQ and minimax tracker solutions

It is pointed out in [22] that LQ control together with state estimator (Kalman filter) can lose its robustness. That’s why it is really important to examine the robustness of the developed methods together with state (and disturbance) estimators.

Another issue is the development of the methods which is based on the inversion of the steady state equilibrium of the system. With system matrices different from nominal the steady state gains in the system will be different from ideal which can result in poor tracking performance. This should be also examined.

The robustness test methodology applied here is called stochastic robustness and stochastic performance of systems. It is based on [74], [75] and [76]. It considers the possible parametric uncertainties in the LTI system by varying directly the values in the system model, similarly as in the structured singular value test formulation.

In examining the stochastic system robustness, it assumes some distribution of system parameters and generates them in Monte-Carlo simulation accordingly. The stability of the closed loop system is analytically checked with every configuration based on the system model. The stable cases are counted and from them the probability of instability (PI) is approximated:

\[ PI = 1 - \frac{\text{Number of stable cases}}{\text{All cases}} \]  

(5.1)

This is the probability from a binomial variable (stable / unstable \( \equiv \) true or false), so exact confidence interval bounds can be calculated.

Regarding the stochastic performance the method simulates the system with varied parameters and then evaluates several measures. The definition of this measures depends on the application. Probability of input saturation, probability of settling time below a limit, etc.

These methods were applied to test the robustness of LQ and minimax trackers. The next section summarizes the parameter variations considered in the system.
5.1 Parameter uncertainties in the controlled aircraft lateral-directional system model

At first, the uncertainty of parameters in the model (1.3) is considered. The Nominal model parameters are the averages of model 1 and model 2 from table 1.1. The flight conditions for the three models can be found in chapter 1. The percentage differences of the three models from the nominal can be seen in table 5.1. The maximum differences are about 30 – 40% except for $N_p$ which is a very uncertain variable (but its absolute value and so effect is much smaller than the other parameters). Thy physical meaning of these parameter uncertainties is described in Appendix 8.22.1.

<table>
<thead>
<tr>
<th>Param.</th>
<th>$L_p$</th>
<th>$L_r$</th>
<th>$N_p$</th>
<th>$N_r$</th>
<th>$L_{\delta_a}$</th>
<th>$L_{\delta_r}$</th>
<th>$N_{\delta_a}$</th>
<th>$N_{\delta_r}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOM.</td>
<td>-12.4</td>
<td>13.55</td>
<td>-0.077</td>
<td>-7.28</td>
<td>59.75</td>
<td>13</td>
<td>-5.125</td>
<td>-16.25</td>
</tr>
<tr>
<td>MOD1 [%]</td>
<td>+3.22</td>
<td>-6.27</td>
<td>+481.82</td>
<td>-16.48</td>
<td>-2.76</td>
<td>+4.62</td>
<td>-28.39</td>
<td>+7.69</td>
</tr>
<tr>
<td>MOD2 [%]</td>
<td>-3.22</td>
<td>+6.27</td>
<td>-481.82</td>
<td>+16.48</td>
<td>+2.76</td>
<td>-4.62</td>
<td>+28.39</td>
<td>+7.69</td>
</tr>
<tr>
<td>MOD3 [%]</td>
<td>+10.48</td>
<td>-36.38</td>
<td>+992.2</td>
<td>+36.54</td>
<td>-27.53</td>
<td>-30.85</td>
<td>+7.12</td>
<td>+26.77</td>
</tr>
</tbody>
</table>

The values and uncertainties of the mass moments of inertia (required to calculate $\mathbf{G}$) are obtained from physical measurements on the Ultrastick aircraft (see [11]):

\begin{align*}
I_x &= 0.062211kgm^2 \quad -2.91\% \div +9.7\% \\
I_z &= 0.167216kgm^2 \quad -1.2\% \div +0.75\% \\
I_{xz} &= 0.0011198kgm^2 \quad -11.2\% \div +10.75\%
\end{align*}

The actuator dynamics should be also uncertain. The time constant and damping of the original function given in (1.4) are: $T_{act} = 0.0398 \approx 0.04s$ and $\xi_{act} = 0.7003 \approx 0.7$. 0.04 and 0.7 were considered as nominal values.

The time delay at the system input (modelled with the Padé filter) is also an uncertain parameter. Its nominal value is $T_d = 0.2s$.

The effect of the state estimator can be considered in two ways. At first, the whole estimator can be considered, second only its effects on the estimated values. For the disturbance estimator UDE applied in the MM tracker, the whole dynamics is considered. From the state vector $p$ and $r$ can be measured by the gyroscope with biases and noises. $\phi$ should be and is estimated by the nonlinear attitude estimator developed in chapter 4. However, it is too complicated to be considered in the robustness test. That’s why only its effects on $\hat{\phi}$ will be considered. These are bias difference, noise, time lag and a saw-like component from divergence between GPS samples. This latter can be seen in figure 5.1 considering real flight data in the attitude EKF. It can be seen that this signal has a 0.25s periodicity because of the 4Hz GPS data. That’s why it was modelled as a repeated sequence from the step output of a simple transfer function $G(s) = \frac{0.00873}{0.05s+1}$ between 0 and
0.25 sec (the exponential transient part is considered). This means a maximum of 0.5° error which is larger than the about 0.1° errors in figure 5.1. The such generated signal was centered by subtracting 0.00436 from it. The final saw-like disturbance signal can be seen in the Appendix in figure 8.37.

From all the above uncertainties the biases are considered only in steady state error calculations (also for the roll- and yaw rate, see later), the noises are added in the simulation (for rates and angle) together with the saw disturbance with fixed parameters. The only considered varying parameter is the time lag of \( \dot{\phi} \) caused by the estimator \((T_e)\). It was considered by the transfer function \( G_{est} = \frac{1}{T_{est} s + 1} \). The other parameters (noise and saw-like component) were not varied because of computational time burdens. The time lag has the most important effect on the closed loop that’s why this was varied in the calculations.

Finally, the varying uncertain parameters for the LQ tracker \((p_{LQ})\) and the minimax tracker \((p_{MM})\) are:

\[
p_{LQ} = [L_p, L_r, N_p, N_r, L_{\delta_a}, L_{\delta_r}, N_{\delta_a}, N_{\delta_r}, T_{act}, \xi_{act}, T_d, T_e]
\]

\[
p_{MM} = [L_p, L_r, N_p, N_r, L_{\delta_a}, L_{\delta_r}, N_{\delta_a}, N_{\delta_r}, T_{act}, \xi_{act}, T_d, I_x, I_{xx}, I_z, T_e]
\]

All the parameters are varied between ±90\% with ±5\% steps except for \(N_p\) which was varied between ±1800\% (with ±100\% steps) and \(T_e\) varied from 0 to 0.108s with 0.012s steps. The uncertainty of \(N_p\) was increased because it originally has much larger uncertainty than the other parameters (see Appendix 8.22.1 for possible explanation).

From [74] binary uniform distributions were assumed for the parameters which means that calculations with the lower and upper bounds (minus and plus percent) are enough to get a good statistics. Of course \(T_e\) was only increased.

The calculations applied the following steps both for the LQ and the minimax tracker:

1. Vary the parameters.

![Figure 5.1: Saw-like dynamics of estimated \( \phi \)](image-url)
2. Check system stability and calculate possible steady state gains (if the system is stable).

3. If the system is stable simulate the tracking of the same doublet reference (with the same disturbances) as in figure 2.3.

4. Calculate the performance parameters from the simulation data.

**5.2 Robustness test of the LQ tracker**

The controller gains obtained during the Matlab Simulink tests were used in this test campaign. The steady state parameters can be calculated from the steady state of system dynamics. This can be derived considering (2.1) as the varied system, (2.23) and constant references ($r_\infty$):

\[
\begin{align*}
\tilde{u}_k &= -K_x \hat{x}_k + K_{S_1} r_\infty + K_{S_2} r_\infty + K_{r_\infty} r_\infty = -K_x x_k + \frac{(K_{x_2} (I - \Phi)^{-1} B + I) F^+ r_\infty}{K_r} \\
x_{k+1} &= A_i x_k + B_i \tilde{u}_k = A_i x_k - B_i K_x x_k B_i K_x x_k + B_i K_r r_\infty \\
x_\infty &= (\Phi_{1i})^\infty x_0 - (I - \Phi_{1i})^{-1} B_i K_x x^e + (I - \Phi_{1i})^{-1} B_i K_r r_\infty \\
y^r_\infty &= -C_r (I - \Phi_{1i})^{-1} B_i K_x x^e + C_r (I - \Phi_{1i})^{-1} B_i K_r r_\infty
\end{align*}
\]

(5.2)

$A_i$ and $B_i$ are the uncertain system matrices, the stability of $\Phi_{1i}$ and constant state estimation error vector $x^e$ are assumed. To have $y^r_\infty = r_\infty$, $C_r (I - \Phi_{1i})^{-1} B_i K_x = Ge = 0$ and $C_r (I - \Phi_{1i})^{-1} B_i K_r = Gr = I$ should be satisfied. From this, the examined steady state parameters are:

1. The stability of $\Phi_{1i}$ and so, the probability of instability.
2. The singular values of $Ge$ ($2 \times 8$ matrix) which well characterize the state estimation error (from measurement and estimator bias) gain.
3. The elements of $Gr$ which are the following gains: $Gr(1,1)$ from $\phi_{ref}$ to $\phi$, $Gr(2,2)$ from $\tau_{ref}$ to $\tau$, $Gr(1,2)$ from $\tau_{ref}$ to $\phi$ and $Gr(2,1)$ from $\phi_{ref}$ to $\tau$.

If the system is stable for the given parameter combination, the simulation is run. The parameters obtained from it are:

1. The lower and upper bounds for the settling times $T_{s1}$ and $T_{s2}$.
2. The saturation of control inputs. This event is counted and the probability of saturation ($P_{Sat}$) is calculated similarly to PI.
3. The lower and upper bounds for the 2-norms of $\phi$ and $\tau$ tracking errors.
4. The lower and upper bounds for the 2-norms of $p$, $\delta_a$ and $\delta_r$ (system and control energy).

5. The lower and upper bounds for the time realizations of $\phi$, $\tau$, $p$, $\delta_a$ and $\delta_r$.

One calculation in this robustness test means the consideration of all possible $\pm$ parameter value combinations with the same level of uncertainty. From all of these calculations the minimum and maximum settling time and norm values are selected and the minimum and maximum signal values at every sampling instant. This means that all the signal realizations (on a given level of uncertainty) are between the lower and upper bounds.

The probabilities of instability and input saturation can be seen in figure 5.2. This shows that system instability can occur above 35% uncertainty level (700% for $N_p$ and 0.084s for $T_e$), while input saturation occurs above 25%. So, the controlled system is robustly stable until 35% uncertainty of every considered parameter and about 0.08s estimator time lag (time constant).

![Figure 5.2: Probability of instability and input saturation according to the uncertainties (LQ case)](image)

The stochastic performance parameters are shown in Appendix 8.22. These figures show that the steady state gain from $\phi_{ref}$ to $\phi$ is acceptable (about 1) until 40% uncertainty level, the other reference to output gains are zero for any uncertainty. This is at first surprising for the $\tau$ reference, but this signal is filtered with the high-pass washout filter while steady state is related to the low (zero) frequency. It is obvious that a high-frequency component can not have an effect on a low-frequency parameter.

The maximum singular value of the gain from state estimation error to output is about 1 until 60% uncertainty level. This means that this error can be transferred to the output even with the nominal system, so the estimation error should be decreased to improve performance.

The other parameters show that the tracking performance is acceptable until 15% uncertainty level which well coincides with the probability of saturation above 25% uncertainty level. However, this 15% level was obtained considering the worst parameter
combinations (the lower and upper bounds are the worst possible signal realizations). Sim-
ulating the LQ tracker with the three real parameter sets from table 1.1 and 5.1 shows
that really good tracking performance can be achieved with higher level of uncertainties
also (see figure 5.3, for an enlarged transient see Appendix 8.22.2). To be honest it should
be noted that neither noises and saw disturbance, nor estimator time delay are considered
in this simulation. But the original 15% level means $T_e = 0.036s \approx 0.04s$ which is about
two time step delay for the 50Hz estimator and tracking was good until this level with
worst case parameters, noises and saw signal together.

As a summary, it can be stated that system robustness and acceptable performance
is possible until about 30 – 40% parameter uncertainty which is an acceptable result.

![Tracking of roll angle and yaw rate](image)

**Figure 5.3:** Good tracking results for the three different identified models (LQ case)

### 5.3 Robustness test of the minimax tracker

Here, the method was similar to the LQ tracker case, that’s why only the differences are
summarized.

Again, the controller gains obtained during the Matlab Simulink tests were used in
this test campaign. The steady state parameters are calculated from the steady state
of system dynamics considering the disturbance estimator also. The steady states can
be derived from (3.3), (3.1), (3.17) and constant references ($r_\infty$) and disturbances ($d_\infty$).
The state equation of the disturbance estimator with $K = 0$ will be (assuming perfect
measurement of $x_k$):

\[
\begin{align*}
\hat{d}_k &= M(y_{k+1} - CAx_k - CBu_k) \\
y_{k+1} &= CAx_k + CBu_k + CG_id_\infty \\
\hat{d}_k &= MC(A_ix_k + Biuk + G_id_\infty - Ax_k - Bu_k) = \\
&= MC(A_i - A)x_k + MC(B_i - B)u_k + MCG_id_\infty
\end{align*}
\] (5.3)
Here, $A_i$, $B_i$ and $G_i$ are the varied system matrices, while $A$, $B$, $G$ are the nominal ones. The $C$ matrix selects the outputs used in the disturbance estimation. The augmented (state dynamics and disturbance estimator) can be constructed as follows:

$$\begin{align*}
\hat{d}_k &= -K_x x_k - K_x x_k^e + K_{r_e} r_{\infty} + K_{d} \hat{d}_{k-1} \\
x_{k+1} &= A_i x_k - B_i K_x x_k - B_i x_k^e + B_i K_{r_e} r_{\infty} + B_i \hat{d}_{k-1} + G_i d_{\infty} \\
\hat{d}_k &= MC(A_i - A)x_k + MC(B_i - B) \left( -K_x x_k - K_x x_k^e + K_{r_e} r_{\infty} + K_{d} \hat{d}_{k-1} \right) + MCG_i d_{\infty}
\end{align*}$$

The stability of $A^a$ and constant state estimation error vector $x^e$ are assumed. To have $y^e_{\infty} = C_r x_{\infty} = r_{\infty}$, $C_r G^e_x = 0$, $C_r G^e_d = I$ and $C_r G^d_x = 0$ should be satisfied. For $d_{\infty} = d_{\infty}$, $G_d = 0$, $G^e_d = 0$ and $G^d_d = I$. From this, the examined steady state parameters are:

1. The stability of $A^a$ and so, the probability of instability.
2. The singular values of $C_r G^e_x$ and $G^e_d$ ($2 \times 8$ matrices) which well characterize the state estimation error (from measurement and estimator bias) gains.
3. The singular values of $C_r G^d_x$ and $G^d_d$ ($2 \times 2$ matrices) which well characterize the disturbance gain to system output and the reference gain to estimated disturbance.
4. The elements of $C_r G^e_d = Gr$ which are the following gains: $Gr(1,1)$ from $\phi_{ref}$ to $\phi$, $Gr(2,2)$ from $\tau_{ref}$ to $\tau$, $Gr(1,2)$ from $\tau_{ref}$ to $\phi$ and $Gr(2,1)$ from $\phi_{ref}$ to $\tau$.
5. The elements of $G^d_d = Gd$ which are the following gains: $Gd(1,1)$ from $d_L$ to $\hat{d}_L$, $Gd(2,2)$ from $d_N$ to $\hat{d}_N$, $Gd(1,2)$ from $d_N$ to $\hat{d}_L$ and $Gd(2,1)$ from $d_L$ to $\hat{d}_N$.  

$91$
If the system is stable for the given parameter combination, the simulation is run. The same parameters are obtained from it as in the LQ tracker case together with the following additional ones:

1. The lower and upper bounds for the 2-norms of disturbance estimation errors.

2. The lower and upper bounds for the time realizations of the estimated disturbances $\hat{d}_L$ and $\hat{d}_N$.

The probabilities of instability and input saturation can be seen in figure 5.4. This shows that system instability can occur above 15% uncertainty level (300% for $N_p$ and 0.036s for $T_e$) together with input saturation. So, the controlled system is robustly stable until 15% uncertainty of every considered parameter and about 0.04s estimator time delay (time constant).

![Probability of instability and control saturation](image.png)

Figure 5.4: Probability of instability and input saturation according to the uncertainties (minimax case)

The stochastic performance parameters are shown in Appendix 8.22 / 8.22.3. These figures show that the steady state gain from $\phi_{ref}$ to $\phi$ is acceptable (about 1) until 30% uncertainty level, the other reference to output gains are zero for any uncertainty similarly to the LQ tracker case.

The gains from system disturbance to the estimated disturbance start to diverge from the required values still at 5% uncertainty level. Their values are acceptable until 15% uncertainty level.

The maximum singular value of the gain from state estimation error to output is about 1 until 40% uncertainty level. This means that this error can be transferred to the output even with the nominal system, so the estimation error should be decreased to improve performance.

The maximum singular value of the gain from state estimation error to estimated disturbance is about 0 until 40% uncertainty level which is really good.
The other parameters show that the tracking performance is acceptable until 15% uncertainty level which coincides with the probability of instability and saturation bounds. However, this 15% level was obtained considering the worst parameter combinations (the lower and upper bounds are the worst possible signal realizations). Simulating the MM tracker with the three real parameter sets from table 1.1 and 5.1 shows that really good tracking performance and also stability can be achieved with higher level of uncertainties also (see figure 5.5 for an enlarged transient see Appendix 8.22.3). To be honest it should be noted that neither noises and saw disturbance, nor estimator time delay are considered in this simulation. But the original 15% level means $T_s = 0.036s \approx 0.04s$ which is about two time step delay for the 50Hz estimator and tracking was good until this level with worst case parameters, noises and saw signal together.

As a summary, it can be stated that the minimax tracker has poorer system robustness and acceptable performance properties than the LQ tracker. This can be because of the interconnection with the open loop disturbance estimator. However, acceptable simulation performance was experienced until 15% uncertainty level for both LQ and MM tracker so, the overall performance of the two methods is the same, but the MM tracker is disturbance tolerant, which is a large advantage.

In the next section real flight test results and comparison with the PID control solution will be presented.

**5.4 Summary**

This chapter does the robustness test of the LQ and MM trackers based on the stochastic robustness methodology. After describing the considered parametric uncertainties in the applied aircraft dynamical model, the detailed tests of the LQ and then the MM tracker are conducted.
Chapter 6

Real flight test results and comparison with the PID solution

Both the LQ and the minimax tracker are tested in real flight onboard the Ultrastick UAV and compared to the PID solution originally implemented onboard (see [70]). The attitude estimator developed in chapter 4 is used in all tests. The same roll doublet reference signal as in figure 2.3 was tracked several times meanwhile the yaw damper was also applied. The weighting for the LQ tracker was exactly the same as the best weighting in the Matlab simulations.

\[
Q_1 = \begin{bmatrix} 100 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 2000 \end{bmatrix}, \quad R = \begin{bmatrix} 5 \cdot 10^3, 5 \cdot 10^4 \end{bmatrix}
\]

For the minimax tracker on 13th September 2011, the weighting was exactly the same as the best in the Matlab simulations, however, this solution was too slow, that’s why the roll angle tracking error weight was increased to 600 and the rudder weight was increased to 100000 in the next test. The latter was increased to provide less oscillations in the yawing motion.

\[
Q_1 = \begin{bmatrix} 500 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 600 \end{bmatrix}, \quad R_u = \begin{bmatrix} 21000, 100000 \end{bmatrix}, \quad R_d = I_4 \cdot 10^{12}
\]

For the minimax tracker on 19th July 2012, with improved performance it was:

\[
Q_1 = \begin{bmatrix} 500 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 600 \end{bmatrix}, \quad R_u = \begin{bmatrix} 21000, 100000 \end{bmatrix}, \quad R_d = I_4 \cdot 10^{12}
\]

All the control codes were implemented on the same PhyCORE MPC555 32 bit, 40MHz floating point system on module which was used by [70]. Paw Yew Chai experienced that this module can work with a maximum of 10 to 15 dimensional controller state spaces. This shows the possible computational burden of microcontroller architecture.

The numerical test results are summarized in table 6.1 considering the two configurations of the minimax tracker together, because they gave similar results.

\[
T_{s1} \quad \text{and} \quad T_{s2} \quad \text{are the settling times to the 95\% to 105\% range.} \quad E_\phi \quad \text{and} \quad E_r \quad \text{are the mean absolute values of the roll angle and yaw rate tracking errors.} \quad E_p, \quad E_{\delta_\alpha} \quad \text{and} \quad E_{\delta_r} \quad \text{are the normalized signal two norms of the roll rate, aileron and rudder deflections defined as:}
\]

\[
94
\]
\[ \|s\| = \frac{1}{N} \sqrt{\sum_{i=1}^{N} s_i^2} \]

Where \( i \) is the time index. The normalization by \( N \) (the number of samples) is required because now signals with different lengths should be compared (the length of each test case depends on decision of the RC pilot). In the table, the minimum, maximum and mean values (from several cases) of each parameter are included.

Table 6.1: Real flight tracking parameters

<table>
<thead>
<tr>
<th>Value</th>
<th>Method</th>
<th>( T_{s1} ) [s]</th>
<th>( T_{s2} ) [s]</th>
<th>( E_\phi ) [°]</th>
<th>( E_r ) [°/s]</th>
<th>( E_p ) [°/s]</th>
<th>( E_\delta_a ) [°]</th>
<th>( E_\delta_r ) [°]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIN</td>
<td></td>
<td>0.68</td>
<td>0.34</td>
<td>1.78</td>
<td>5.03</td>
<td>0.87</td>
<td>0.06</td>
<td>0.01</td>
</tr>
<tr>
<td>MAX</td>
<td>PID</td>
<td>2.66</td>
<td>1.16</td>
<td>4.56</td>
<td>7.92</td>
<td>1.62</td>
<td>0.13</td>
<td>0.14</td>
</tr>
<tr>
<td>MEAN</td>
<td></td>
<td>1.213</td>
<td>0.72</td>
<td>3.348</td>
<td>6.5</td>
<td>1.19</td>
<td>0.088</td>
<td>0.032</td>
</tr>
<tr>
<td>MIN</td>
<td></td>
<td>0.64</td>
<td>0.68</td>
<td>3.04</td>
<td>5.57</td>
<td>0.86</td>
<td>0.05</td>
<td>0.02</td>
</tr>
<tr>
<td>MAX</td>
<td>LQ</td>
<td>4.5</td>
<td>2.08</td>
<td>4.78</td>
<td>9.06</td>
<td>1.53</td>
<td>0.2</td>
<td>0.03</td>
</tr>
<tr>
<td>MEAN</td>
<td></td>
<td>2.44</td>
<td>1.07</td>
<td>3.98</td>
<td>7.267</td>
<td>1.179</td>
<td>0.0855</td>
<td>0.027</td>
</tr>
<tr>
<td>MIN</td>
<td></td>
<td>0.72</td>
<td>2.6</td>
<td>5.52</td>
<td>3.92</td>
<td>0.43</td>
<td>0.05</td>
<td>0.06</td>
</tr>
<tr>
<td>MAX</td>
<td>MM</td>
<td>4.98</td>
<td>4.92</td>
<td>7.6</td>
<td>9.25</td>
<td>4.77</td>
<td>0.23</td>
<td>0.13</td>
</tr>
<tr>
<td>MEAN</td>
<td></td>
<td>2.9</td>
<td>3.73</td>
<td>6.263</td>
<td>5.88</td>
<td>1.248</td>
<td>0.097</td>
<td>0.087</td>
</tr>
</tbody>
</table>

Considering the table the settling times are larger both with the LQ and minimax tracker. The mean value of the \( \phi \) tracking error is about the same with the LQ, but much larger with the MM method. This clearly shows that the MM method is slower, as experienced also in the simulation tests. Regarding the error of yaw damping, the LQ mean is worse, but the MM mean is better than the PID mean. This can be closely related to the used rudder control energy. The LQ method uses the less, while the MM method uses the most rudder energy. So, the yaw damping with the MM method is better, but it uses more control energy. Considering the energy in the roll rate, the methods are about the same, the MM method uses a bit more energy. This also can be seen considering the aileron control energy.

The plots of the flight test results can be seen in figures from 6.1 to 6.12. Every plot includes time histories from several flights. Enlarged figures can be found in Appendix 8.23. Notice that in the LQ tracker cases the aircraft \( \phi \) angle is a bit below 20° while around \(-20°\). So, it tracks the negative angle easier. This is because of the negative torque disturbance resulting from electrical motor and propeller reaction torque which is negative with a propeller rotating to the right. So, this shows the worse disturbance rejection properties of the LQ tracker. As a summary it can be stated that all two methods are applicable on a real system with acceptable performance. Comparing them to the \( \mu \) synthesis based roll angle tracking controller in [70] which was tested also on Ultrastick aircraft with MPC555 microcontroller they give better results. They track the non smoothed doublet signal better than that controller tracks the smoothed one. The real flight performance of that controller is shown in figure 6.13.

Finally, a few words about the possible further application of the algorithms. The developed roll angle tracking and yaw damping algorithms can be used in aircraft waypoint...
and trajectory tracking tasks. In such problems the aircraft turn can be initiated by the banking of the aircraft through its roll angle. Waypoint and trajectory tracking algorithms were also developed and published by the author in [Bau11] and [BB12]. Two representative figures about waypoint and spatial (sinusoidal) trajectory tracking can be seen in 6.14 and 6.15.

6.1 Summary

This chapter compares real flight test results obtained with the LQ and MM trackers onboard the Ultrastick test aircraft with results obtained from a PID tracking solution (the same solution as is used in the Matlab simulation tests).

![Figure 6.1: Tracking of the $\phi$ doublet reference signal on 10th December 2009. PID control](image1)

![Figure 6.2: Tracking of the $\phi$ doublet reference signal on 17th June 2010. PID control](image2)

![Figure 6.3: yaw damping on 10th December 2009. PID control](image3)

![Figure 6.4: yaw damping on 17th June 2010. PID control](image4)
Figure 6.5: Tracking of the $\phi$ doublet reference signal on 13th September 2011. LQ tracker control

Figure 6.6: Tracking of the $\phi$ doublet reference signal on 29th September 2011. LQ tracker control

Figure 6.7: Yaw damping on 13th September 2011. LQ tracker control

Figure 6.8: Yaw damping on 29th September 2011. LQ tracker control

Figure 6.9: Tracking of the $\phi$ doublet reference signal on 13th September 2011. MM tracker control

Figure 6.10: Tracking of the $\phi$ doublet reference signal on 19th July 2012. MM tracker control
Figure 6.11: Yaw damping on 13th September 2011. MM tracker control

Figure 6.12: Yaw damping on 19th July 2012. MM tracker control

Figure 6.13: Tracking performance of a $\mu$ synthesis based controller from [70] page 138. Figure 6.18 (published with the permission of the author)

Figure 6.14: Waypoint tracking

Figure 6.15: Sinusoidal trajectory tracking
Chapter 7

New scientific results

Thesis 1 I formulated and solved the problem of finite horizon, LQ optimal output tracking for DT, LTI systems with Lagrange multiplier method in a more detailed way than published in literature. This detailed solution gives deeper insight into the effect of reference signal on the system which makes it possible to derive a nice infinite horizon solution in Thesis 2. This solution could not be derived without knowing the obtained detailed structure of the costate variable.

I extended the finite horizon output tracking problem to the minimax case, when system disturbances are also considered in the functional to be minimized. A structurally similar, but otherwise different solution is derived.

My publications related to Thesis 1: [BKB08a], [Pé10], [Pé9], [BKB09b], [BKB09a].

Thesis 2 I solved the problem of infinite horizon, LQ optimal, causal reference tracking in a LQ optimal way for constant, and in a LQ sub-optimal way for time-varying references, considering DT, LTI causal systems without deterministic disturbances (2.1). The solution is based on the finite horizon result of Thesis 1 and considers reference signal extrapolation to guarantee causality and make the stated problem solvable. This extrapolation causes the sub-optimality in case of time-varying references.

My publications related to Thesis 2: [BRSB08], [Pé10], [Bau08], [Pé9], [BKB08a], [BB11b].

Thesis 3 I derived solvability conditions for the Discrete Algebraic Riccati Equation (DARE) resulting in the infinite horizon LQ tracking problem from the functional (2.11).

My publications related to Thesis 3: [BKB08b].

Thesis 4 I solved the problem of infinite horizon, LQ optimal, causal minimax reference tracking in a LQ optimal way for constant, and in a LQ sub-optimal way for time-varying references and disturbances, considering DT, LTI causal systems with deterministic disturbances (3.1). The solution is based on the finite horizon minimax result of Thesis 1 and considers reference signal extrapolation to guarantee causality and make the stated problem solvable. This extrapolation causes the sub-optimality in case of time-varying references.

My publications related to Thesis 4: [BKB09a], [BKB09b], [Pé10b], [BB11c].
Thesis 5 I developed a multi-mode attitude estimation Extended Kalman Filter. It switches between the measurements and uses wind estimation and correction to obtain as accurate estimates as possible. It calculates the integral of quaternion dynamic equation with a closed form solution of the Heun formula which can further improve accuracy. 

My publications related to Thesis 5: [Pöt10a], [BB10], [BB11a].
Chapter 8
Appendix

This appendix contains the detailed derivation of the results published in the thesis, the detailed description of some variables and detailed test results.

8.1 The applied basic methods

The thesis and the underlying work is built upon the following methods and tools from systems and control theory.

LQ optimal (output tracking) control \[3, 53\], considered in case of LTI discrete time systems with the following system model:

\[
\begin{align*}
    x_{k+1} &= Ax_k + B\tilde{u}_k \\
    y^r_k &= C_r x_k \\
    y_k &= C x_k
\end{align*}
\] (8.1)

The goal of design is to follow a prescribed reference signal \( r \) with the reference output \( y^r \) by minimizing the following functional:

\[
J(x, \bar{x}, \bar{u}) = \frac{1}{2} \sum_{k=0}^{N-1} \left( (x_k - \bar{x}_k)^T Q (x_k - \bar{x}_k) + \bar{u}_k^T R \bar{u}_k \right) +
+ (x_N - \bar{x}_N)^T Q (x_N - \bar{x}_N)
\]

where:

\[
Q = \overline{C}^T Q_1 \overline{C} + C_r^T Q_2 C_r
\]

\[
\overline{C} = \left( I - C_r^T (C_r C_r^T)^{-1} C_r \right)
\]

\[
\bar{x}_k = C_r^T (C_r C_r^T)^{-1} r_k = H r_k
\]

The standard Lagrange multiplier method can be used to minimize this functional by deriving the costate variable at first for finite, then for infinite horizons (if exists).
LQ optimal minimax (output tracking) control [13,49]. considered in case of LTI discrete time systems with the following system model:

\[
\begin{align*}
x_{k+1} &= Ax_k + B\hat{u}_k + Gd_k \\
y'_k &= C_r x_k \\
y_k &= C x_k
\end{align*}
\] (8.3)

The goal of design is to follow a prescribed reference signal \(r\) with the reference output \(y'\) together with attenuating the disturbance \(d\) and by minimizing the following functional:

\[
J(x, \hat{x}, \hat{u}, d_k) = \\
= \frac{1}{2} \sum_{k=0}^{N-1} ((x_k - \hat{x}_k)^T Q (x_k - \hat{x}_k) + \hat{u}_k^T R_u \hat{u}_k - \\
- \gamma^2 d_k^T R_d d_k) + (x_N - \hat{x}_N)^T Q (x_N - \hat{x}_N)
\]
\] (8.4)

where:

\[
Q = C^T Q_1 C + C_r^T Q_2 C_r \\
\hat{C}_k = (I - C_r (C_r C_r^T)^{-1} C_r) \\
\hat{x}_k = C_r^T (C_r C_r^T)^{-1} r_k = H r_k
\]

Here also the standard Lagrange multiplier method can be used to minimize the functional by deriving the costate variable at first for finite, then for infinite horizons (if exists).

Extended Kalman Filter for nonlinear systems [42]. Considering the following nonlinear system (linearized around the actual state):

\[
\begin{align*}
\dot{x} &= f(x) \quad \rightarrow \quad A_k = \left. \frac{\partial f(x)}{\partial x} \right|_{x_k} \\
y &= h(x) \quad \rightarrow \quad C_k = \left. \frac{\partial h(x)}{\partial x} \right|_{x_k} \\
x_{k+1} &= A_k x_k + V_k v_k \\
y_k &= C_k x_k + w_k \\
E\{v_k\} &= 0, \quad E\{w_k\} = 0, \quad N_k = E\{v_k v_k^T\}, \quad W_k = E\{w_k w_k^T\}
\end{align*}
\] (8.5)

The goal is to obtain the unbiased, minimum variance estimation of the system states from the measured output \(y\). This goal can be achieved by the following algorithm:

\[
\begin{align*}
\bar{x}_{k+1} &= f(\hat{x}_k) \\
\bar{P}_{k+1} &= A_k \bar{P}_k A_k^T + V_k N_k V_k^T \\
K_{k+1} &= \bar{P}_{k+1} C_k^T (C_k \bar{P}_{k+1} C_k^T + W_k)^{-1} \\
\hat{x}_{k+1} &= \bar{x}_{k+1} + K_{k+1} (y_{k+1} - h(\bar{x}_{k+1})) \\
P_{k+1} &= (I - K_{k+1} C_k) \bar{P}_{k+1}
\end{align*}
\] (8.6)
Here $\hat{x}$ is the estimated state of the system and $P$ is the state estimation error covariance matrix ($P_k = E((x_k - \hat{x}_k)(x_k - \hat{x}_k)^T)$).

8.2 Introduction of the E-flite Ultrastick 25e aircraft

The aircraft is originally a small RC model aircraft equipped with sensors, onboard microcomputer and wireless communication unit to be capable for autonomous flight.

![Ultrastick aircraft of MTA SZTAKI](image)

Figure 8.1: The Ultrastick aircraft of MTA SZTAKI

The technical data of the aircraft is as follows:

- Length: 1.05m
- Wingspan: 1.27m
- Empty weight: 1.5kg
- Maximum take-off weight (MTOW): 2.1kg
- Flight time: 15 to 20 min
- Cruising speed: 15 to 25 m/s
- Engine power: 600 Watt electrical DC Brushless motor

It is a shoulder wing aircraft without dihedral and with symmetric airfoil. It has limited aerobatic capability, but high bank turns can be easily flown with it.
8.3 Derivation of the aircraft lateral dynamical model

The linearized aircraft lateral dynamical model is as follows (see also (1.3)):

\[
\begin{bmatrix}
\dot{p} \\
\dot{r} \\
\dot{\phi}
\end{bmatrix}
= \begin{bmatrix}
L_p & L_r & 0 \\
N_p & N_r & 0 \\
1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
p \\
r \\
\phi
\end{bmatrix}
+ \begin{bmatrix}
L_{\delta_a} & L_{\delta_r} \\
N_{\delta_a} & N_{\delta_r} \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
\delta_a \\
\delta_r
\end{bmatrix}
+ \begin{bmatrix}
L_{d_L} & L_{d_N} \\
N_{d_L} & N_{d_N} \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
d_L \\
d_N
\end{bmatrix}
\]

This can be derived from the principle of angular momentum:

\[
\begin{bmatrix}
I_x & 0 & -I_{xz} \\
0 & I_y & 0 \\
-I_{xz} & 0 & I_z
\end{bmatrix}
\begin{bmatrix}
\dot{p} \\
\dot{q} \\
\dot{r}
\end{bmatrix}
+ \omega \times (J\omega)
= \begin{bmatrix}
L \\
M \\
N
\end{bmatrix}
\]

Here, \(I_x, I_y, I_z, I_{xz}\) are the mass moments of inertia of the aircraft obtained from measurements in [11]. Only the first and third equations should be used for aircraft lateral dynamics in the form:

\[
\begin{align*}
\dot{p} - \frac{I_{xz}}{I_x} \dot{r} &= \frac{\eta S b}{I_x} c_L - \frac{I_z}{I_x} I_y q r + \frac{I_{xz}}{I_x} q p + \frac{1}{I_x} d_L \\
\dot{r} - \frac{I_{xz}}{I_z} \dot{p} &= \frac{\eta S b}{I_z} c_N - \frac{I_y}{I_z} p q - \frac{I_{xz}}{I_z} q r - \frac{I_p}{I_z} \omega_p q + \frac{1}{I_z} d_N \\
\dot{\phi} &= p + \tan(\theta_0) r
\end{align*}
\]

Here, \(p, r\) and \(\phi\) are the roll- and yaw rate and roll angle respectively. \(I_x, I_z\) and \(I_{xz}\) are the inertias of the aircraft, \(\eta\) is the dynamic pressure, \(S\) is the wing surface, \(b\) is the wingspan, \(c_L\) and \(c_N\) are the linearized aerodynamic coefficients, \(I_p\) and \(\omega_p\) are the inertia and angular velocity of the rotating parts of the motor and propeller, \(V_0\) and \(\theta_0\) are the trim velocity and pitch angle and \(d_L\) and \(d_N\) are disturbance torques acting on the aircraft from motor reaction and wind effects for example. For the derivation of the lateral model \(q = 0\) can be assumed and so the cross terms including \(q\) can be neglected. This modelling error can be assumed to be included in the \(d_L\) and \(d_N\) disturbance torques. Assuming also \(\theta_0 = 0\) leads to \(\dot{\phi} = p\).

The parameters of the model in (8.7) were obtained from system identification in [70]. Unfortunately there the \(\beta\) angle of sideslip was not measured and so its effect could not
be identified. That’s why it is not included in the final model. Finally, the first two
equations in (8.9) should be reorganized to have only one derivative in a line on the left
hand side. Then the coefficients of the linearized model can be expressed as follows:

\[
\begin{bmatrix}
1 & -\frac{I_{xz}}{I_x} & -\frac{I_{xz}}{I_z} & I_{xz} \\
-\frac{I_{xz}}{I_x} & 1 & -\frac{I_{xz}}{I_z} & -\frac{I_{xz}}{I_x} \\
-\frac{I_{xz}}{I_z} & -\frac{I_{xz}}{I_x} & 1 & I_{xz} \\
I_{xz} & I_{xz} & I_{xz} & 1
\end{bmatrix} \begin{bmatrix}
\dot{p} \\
\dot{r}
\end{bmatrix} = \text{RHS} \rightarrow \begin{bmatrix}
\dot{p} \\
\dot{r}
\end{bmatrix} = \begin{bmatrix}
\frac{I_x I_z}{I_x} & -\frac{I_{xz}}{I_x} & -\frac{I_{xz}}{I_z} \\
-\frac{I_{xz}}{I_x} & 1 & I_{xz} \\
-\frac{I_{xz}}{I_z} & -\frac{I_{xz}}{I_x} & 1 \\
I_{xz} & I_{xz} & I_{xz} & 1
\end{bmatrix} \begin{bmatrix}
\dot{p} \\
\dot{r} \\
\dot{\lambda}_k \\
\dot{\lambda}_N
\end{bmatrix}
\]

\[
L_p = \frac{\bar{q} S b^2}{2 V_0} \left( \frac{I_1}{I_x} c_{L_p} + \frac{I_2}{I_z} c_{N_p} \right) \\
L_r = \frac{\bar{q} S b^2}{2 V_0} \left( \frac{I_1}{I_x} c_{L_r} + \frac{I_2}{I_z} c_{N_r} \right) \\
L_{\delta a} = \bar{q} S b \left( \frac{I_1}{I_x} c_{L_{\delta a}} + \frac{I_2}{I_z} c_{N_{\delta a}} \right) \\
L_{\delta r} = \bar{q} S b \left( \frac{I_1}{I_x} c_{L_{\delta r}} + \frac{I_2}{I_z} c_{N_{\delta r}} \right) \\
N_p = \frac{\bar{q} S b^2}{2 V_0} \left( \frac{I_3}{I_x} c_{L_p} + \frac{I_1}{I_z} c_{N_p} \right) \\
N_r = \frac{\bar{q} S b^2}{2 V_0} \left( \frac{I_3}{I_x} c_{L_r} + \frac{I_1}{I_z} c_{N_r} \right) \\
N_{\delta a} = \bar{q} S b \left( \frac{I_3}{I_x} c_{L_{\delta a}} + \frac{I_1}{I_z} c_{N_{\delta a}} \right) \\
N_{\delta r} = \bar{q} S b \left( \frac{I_3}{I_x} c_{L_{\delta r}} + \frac{I_1}{I_z} c_{N_{\delta r}} \right)
\]

(8.10)

8.4 Derivation of the finite horizon LQ optimal tracking solution with Lagrange multiplier method

The solution of the optimization problem stated in (2.3) is derived in [53] chapter 2.6
based on the Lagrange multiplier method. The optimal control input results as (p. 113,
(2.6-13)):

\[
\dot{u}_k = -R^{-1}B^T \lambda_{k+1}
\]

(8.11)

The costate variable results as (p. 113, (2.6-11)):

\[
\dot{\lambda}_k = Q x_k - Q \ddot{x}_k + A^T \lambda_{k+1} \\
\dot{\lambda}_N = Q x_N - Q \ddot{x}_N
\]

(8.12)

Substituting the above expressions into (2.1) gives the Hamiltonian matrix (2.6-14):

\[
\begin{bmatrix}
x_{k+1} \\
\lambda_k
\end{bmatrix} = \begin{bmatrix}
A & -BR^{-1}B^T \\
Q & A^T
\end{bmatrix} \begin{bmatrix}
x_k \\
\lambda_{k+1}
\end{bmatrix} + \begin{bmatrix}
0 \\
-QH
\end{bmatrix} r_k
\]

(8.13)

At this point, the solution in [53] makes a generalization which hides the structure of
the forcing term in the costate variable and assumes \( \lambda_k = P_k x_k - s_R(k) \) ((2.6-15) where
$S_k = P_k$ and $v_k = s_R(k)$ are used) based on the second equation (initial costate condition) in (8.12). This shows one term with the effect of $x_k$ and one with the effect of the reference signal $\tilde{x}_k = Hr_k$. But this hides the detailed reference signal effects which can be obtained as below.

The equations from (8.13) for $k = N - 1$ result as (considering $\lambda_N = Qx_N - Q\tilde{x}_N$):

$$x_N = Ax_{N-1} - BR^{-1}B^TQx_N + BR^{-1}B^TQHr_N$$
$$\lambda_{N-1} = Qx_{N-1} + A^TQx_N - A^TQHr_N - QHr_{N-1}$$

$$x_N = [I + BR^{-1}B^TQ]^{-1} (Ax_{N-1} + BR^{-1}B^TQHr_N)$$
$$\lambda_{N-1} = Qx_{N-1} + A^TQ [I + BR^{-1}B^TQ]^{-1} Ax_{N-1} +$$
$$+ A^TQ [I + BR^{-1}B^TQ]^{-1} BR^{-1}B^TQHr_N - A^TQHr_N - QHr_{N-1}$$

From the last equation in (8.14) the structure of the costate variable results as:

$$\lambda_{N-1} = P_{N-1}x_{N-1} + S_{N-1}\tilde{x}_N - Q\tilde{x}_{N-1}$$
$$\lambda_k = P_kx_k + S_k\tilde{x}_{k+1} - Q\tilde{x}_k$$

This clearly shows that the costate variable at time $k$ depends on the actual and future references instead of only the actual $s_R(k)$. This property will be used in the derivation of the infinite horizon causal solution.

8.5 The nonsingularity of $I - \Phi^T M_2$

Consider first, the state dynamics of the centralized system (2.9) and the state feedback part of its control input (2.22):

$$\Delta x_{k+1} = \Phi \Delta x_k + B \Delta u_k = \Phi \Delta x_k - BKx_2 \Delta x_k + \ldots = \Phi_1 \Delta x_k + \ldots$$

In (8.16) $\Phi_1$ is the stable system matrix of the closed loop. Considering the details in (2.22) the transpose of $\Phi_1$ can be formulated as:

$$\Phi_1 = \Phi - BKx_2 = \Phi - BR^{-1}B^T P_\infty [I + BR^{-1}B^T P_\infty]^{-1} \Phi = [I + BR^{-1}B^T P_\infty]^{-1} \Phi$$
$$\Phi_1^T = \Phi^T [I + BR^{-1}B^T P_\infty]^{-T} = \Phi^T \left( [I + BR^{-1}B^T P_\infty]^T \right)^{-1} = \Phi^T [I + P_\infty BR^{-1}B^T]^{-1}$$

This is exactly the $\Phi^T M_2$ term in $I - \Phi^T M_2 = I - \Phi^T [I + P_\infty BR^{-1}B^T]^{-1}$. This way $I - \Phi^T M_2 = I - \Phi_1^T$ will be invertible because $\Phi_1$ is a stable DT system matrix without poles on the unit circle.
8.6 Evaluation of the DARE weighting matrix

At first evaluate $\overline{C}$ considering $C_r = \overline{C}_r = [V \ V M]$ (see (2.25)).

$$
\overline{C} = \left( I - C_r^T (C_r C_r^T)^{-1} C_r \right)
$$

At first evaluate

$$
C_r C_r^T = V \left[ I + MM^T \right] V^T = VM_4 V^T
$$

$$(C_r C_r^T)^{-1} = V^{-T} M_4^{-1} V^{-1} \quad (8.18)
$$

Then evaluate $Q$ considering (2.31), (2.36), (2.25) and (8.18):

$$
Q = \overline{C}^T \begin{bmatrix} Q_{1,11} & 0 \\ 0 & Q_{1,22} \end{bmatrix} \overline{C} + \begin{bmatrix} V^T M T V^T \\ M T V^T \end{bmatrix} Q_2 \begin{bmatrix} V & V M \end{bmatrix} =
\begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} +
\begin{bmatrix} V^T Q_2 V & V^T Q_2 V M \\ M T V^T Q_2 V & M T V^T Q_2 V M \end{bmatrix}
$$

where

$$
Q_{11} = (I - M_4^{-T}) Q_{1,11} (I - M_4^{-1}) + M_4^{-T} M Q_{1,22} M T M_4^{-1}
$$

$$
Q_{12} = -(I - M_4^{-T}) Q_{1,11} M M_4^{-1} M - M_4^{-T} Q_{1,22} (I - M T M_4^{-1} M)
$$

$$
Q_{21} = -M T M_2^{-T} Q_{1,11} (I - M_4^{-1}) - (I - M T M_2^{-T} M) Q_{1,22} (I - M T M_4^{-1} M)
$$

$$
Q_{22} = M T M_2^{-T} Q_{1,11} M M_4^{-1} M + (I - M T M_2^{-T} M) Q_{1,22} (I - M T M_4^{-1} M)
$$

(8.19)

Substituting $M_4 = I + MM^T$ into the terms in (8.19), using the matrix inversion lemma and doing some algebraic manipulations leads to the following terms:

$$
Q_{11} = MM_3^{-T} M T Q_{1,11} M M_3^{-1} M T + MM_3^{-T} Q_{1,22} M_3^{-1} M T
$$

$$
Q_{12} = -MM_3^{-T} M T Q_{1,11} M M_3^{-1} M - MM_3^{-T} Q_{1,22} M_3^{-1}
$$

$$
Q_{21} = -M_3^{-T} M T Q_{1,11} M M_3^{-1} M T - M_3^{-T} Q_{1,22} M_3^{-1} M T
$$

$$
Q_{22} = M_3^{-T} M T Q_{1,11} M M_3^{-1} M T + M_3^{-T} Q_{1,22} M_3^{-1}
$$

where

$$
M_3 = I + M T M
$$

(8.20)

From (8.19) and (8.20) the following common terms can be constructed:

$$
\overline{Q}_{1,11} = M_3^{-T} M T Q_{1,11} M M_3^{-1}
$$

$$
\overline{Q}_{1,22} = M_3^{-T} Q_{1,22} M_3^{-1}
$$

$$
\overline{Q}_2 = V^T Q_2 V
$$

(8.21)
Applying these terms \( Q \) can be reformulated as:

\[
Q = \begin{bmatrix}
M\overline{Q}_{1,11}M^T + M\overline{Q}_{1,22}M^T & -M\overline{Q}_{1,11} - M\overline{Q}_{1,22}
\end{bmatrix} + \begin{bmatrix}
\overline{Q}_2 & \overline{Q}_2 M
\end{bmatrix} \begin{bmatrix}
M^T\overline{Q}_2 & M^T\overline{Q}_2 M
\end{bmatrix}
\] (8.22)

### 8.7 Upper bounds for reference signal \( l_1 \) and \( l_2 \) norms

Assuming that the absolute value of every component of the vector \( r_k \in \mathbb{R}^r \) is bounded by an exponential expression with the same exponent leads to the following bound for the absolute value of the vector:

\[
|r_k| = \begin{bmatrix} r_{1k} & r_{2k} & \ldots & r_{rk} \end{bmatrix}^T \leq A_1 e^{-ak}
\]

\[
|r_k| = \sqrt{r_{1k}^2 + r_{2k}^2 + \ldots + r_{rk}^2} \leq \sqrt{A_1^2 e^{-2ak} + A_2^2 e^{-2ak} + \ldots + A_r^2 e^{-2ak}}
\]

where

\[
|r_0| \leq A
\]

Using (8.23) the \( l_1 \) norm of \( r_k \) will be the following:

\[
\|r_k\|_1 = \sum_{k=0}^{\infty} |r_k| \leq A \sum_{k=0}^{\infty} e^{-ak} = A \frac{1}{1 - e^{-a}}
\] (8.24)

Equation (8.24) defines a geometric series which is convergent if \( e^{-a} < 1 \) this requires \( a > 0 \). This is a very tight condition for an \( l_1 \) signal so, a wide class of \( l_1 \) signals is considered.

The next question is if this \( r_k \) is also an \( l_2 \) signal? The answer is yes as follows:

If \( |r_k| \leq A e^{-ak} \) then \( |r_k|^2 \leq A^2 e^{-2ak} \) because of the strictly monotonically decreasing property of the exponential function if \(-ak \leq 0\).

\[
\|r_k\|_2 = \sum_{k=0}^{\infty} |r_k|^2 \leq A^2 \sum_{k=0}^{\infty} e^{-2ak} = A^2 \frac{1}{1 - e^{-2a}}
\] (8.25)

### 8.8 Upper bound for the absolute value of reference part of input

Here an upper bound for the absolute value of \( u_k^r \) in (2.57) is derived in the following steps by considering (8.23)

\[
u_k^r = -K_{x2}x_k^r + \overline{K}_S r_{k+1} + K_S r_k
\]

\[
|u_k^r| \leq \|K_{x2}\||x_k^r| + \|\overline{K}_S\||r_{k+1}| + \|K_S\||r_k|
\] (8.26)
In (8.26) the $|x_k^e|$ state estimation error should go to zero as the system state goes to zero because of the asymptotic stability of the estimator. So, it can be upper bounded by an exponential function $|x_k^e| < D_E e^{-d_E k}$ and this way a maximum upper bound can be derived for $u_k^r$.

$$|u_k^r| \leq \|K_x\| D_E e^{-d_E k} + \|K_{S_1}\| A e^{-a(k+1)} + \|K_{S_2}\| A e^{-a k}$$

$$A e^{-a(k+1)} < A e^{-a k} \Rightarrow$$

$$|u_k^r| \leq \|K_x\| D_E e^{-d_E k} + (\|K_{S_1}\| + \|K_{S_2}\|) A e^{-a k}$$

$$K_U = \|K_x\| D_E + (\|K_{S_1}\| + \|K_{S_2}\|) A, \ u = \min(a, d_E)$$

$$|u_k^r| < K_U e^{-u k} < \infty$$

(8.27)

### 8.9 Finiteness of the infinite horizon cost functional for $l_1$ ($l_2$) references

**Preliminary result**: the upper bound of the sum of series constructed from the norms of the $k$th powers of a stable system matrix $\sum_{k=0}^{\infty} \|\Phi_k^k\|$.

If $\Phi_1$ is stable with eigenvalues inside the unit circle, there exists an index $K$ for which $\lambda = \|\Phi_1^K\| < 1$ From this the following statements follow:

$$\|\Phi_1^{2K}\| \leq \|\Phi_1^K\| \|\Phi_1^K\| = \lambda^2 < \lambda$$

$$\vdots$$

$$\|\Phi_1^{3K-1}\| < \lambda$$

$$\|\Phi_1^{3K}\| < \lambda^2$$

$$\vdots$$

$$\|\Phi_1^{4K-1}\| < \lambda^2$$

$$\vdots$$

$$\sum_{k=2K}^{3K-1} \|\Phi_k\| < K \lambda$$

$$\sum_{k=3K}^{4K-1} \|\Phi_k\| < K \lambda^2$$

$$\vdots$$

(8.28)

Considering (8.28), the upper bound for the sum of the series results as:
Grouping the terms and applying $E$ finite. So examine the finiteness of every term considering (8.29) and (8.27).

Now substituting the control input (2.57) and the state from (2.58) into (2.52) results in:

$$J(x, \tilde{x}, u) = \frac{1}{2} \sum_{k=0}^{\infty} \left\{ (\Phi_1^k x_0 + R_k)^T Q (\Phi_1^k x_0 + R_k) - 2 (\Phi_1^k x_0 + R_k)^T \tilde{x}_k + \tilde{x}_k^T Q \tilde{x}_k + \right.$$

$$+ (-K_{x2} x_k + u_r^T) R (-K_{x2} x_k + u_r^T) \right\} =$$

$$= \frac{1}{2} \sum_{k=0}^{\infty} \left\{ (\Phi_1^k x_0 + R_k)^T Q (\Phi_1^k x_0 + R_k) - 2 (\Phi_1^k x_0 + R_k)^T \tilde{x}_k + \tilde{x}_k^T Q \tilde{x}_k + \right.$$

$$+ (-K_{x2} (\Phi_1^k x_0 + R_k) + u_r^T) R (-K_{x2} (\Phi_1^k x_0 + R_k) + u_r^T) \right\}$$

(8.30)

Grouping the terms and applying $E = Q + K_{x2}^T R K_{x2}$ the following form of the functional results:

$$J(x, \tilde{x}, u) = \frac{1}{2} \sum_{k=0}^{\infty} \{ x_0^T (\Phi_1^T)^k E \Phi_1^k x_0 + 2 x_0^T (\Phi_1^T)^k E R_k + R_k^T E R_k -$$

$$- 2 \left( x_0^T (\Phi_1^T)^k + R_k^T \right) Q \tilde{x}_k + \tilde{x}_k^T Q \tilde{x}_k - 2 \left( x_0^T (\Phi_1^T)^k + R_k^T \right) K_{x2}^T R u_k + (u_r^T)^T R u_k \}$$

(8.31)

An upper bound for the functional value can be constructed by taking the absolute values of the terms in the summation:

$$J(x, \tilde{x}, u) \leq \frac{1}{2} \sum_{k=0}^{\infty} \{ | x_0^T (\Phi_1^T)^k E \Phi_1^k x_0 | + 2 | x_0^T (\Phi_1^T)^k E R_k | + | R_k^T E R_k | +$$

$$+ 2 \left| (x_0^T (\Phi_1^T)^k + R_k^T) Q \tilde{x}_k \right| + | \tilde{x}_k^T Q \tilde{x}_k | + 2 \left| (x_0^T (\Phi_1^T)^k + R_k^T) K_{x2}^T R u_k \right| +$$

$$+ | (u_r^T)^T R u_k^T | \}$$

(8.32)

If the infinite sum of every term in (8.32) is finite, then the functional value is surely finite. So examine the finiteness of every term considering (8.29) and (8.27).
Term 1:

\[
\frac{1}{2} \sum_{k=0}^{\infty} |x_0^T (\Phi_1^T)^k E \Phi_1^k x_0| \leq \frac{1}{2} \sum_{k=0}^{\infty} |x_0^T|| (\Phi_1^T)^k ||E||\Phi_1^k||x_0| \leq \\
\leq |x_0|^2 ||E|| \frac{1}{2} \sum_{k=0}^{\infty} || (\Phi_1^T)^k ||\Phi_1^k|| \leq \frac{1}{2} |x_0|^2 ||E|| K_\Phi^2 < \infty
\]

(8.33)

Term 2:

\[
\frac{1}{2} \sum_{k=0}^{\infty} |2x_0^T (\Phi_1^T)^k ER_k| \leq \frac{1}{2} \sum_{k=0}^{\infty} 2|x_0^T|| (\Phi_1^T)^k ||E|||R_k|
\]

\[
|R_k| = \sum_{l=0}^{k-1} (\Phi_1^l Bu_{k-1-l}^r) < \sum_{l=0}^{k-1} ||\Phi_1^l||||B||u_{k-1-l}^r < \sum_{l=0}^{k-1} ||\Phi_1^l||||B|| K_\Phi e^{-u(k-1-l)}
\]

\[
\sum_{l=0}^{\infty} ||\Phi_1^l|| < \infty \Rightarrow \exists \Phi_U : ||\Phi_1^l|| \leq \Phi_U
\]

(8.34)

\[
|R_k| < \Phi_U ||B|| K_\Phi \sum_{l=0}^{k-1} e^{-u(k-1-l)} < \Phi_U ||B|| K_\Phi \frac{1}{1 - e^{-u}}
\]

\[
\frac{1}{2} \sum_{k=0}^{\infty} |2x_0^T (\Phi_1^T)^k ER_k| < |x_0||E||\Phi_U||B|| K_\Phi \frac{1}{1 - e^{-u}} \sum_{k=0}^{\infty} || (\Phi_1^T)^k || < \\
< |x_0||E||\Phi_U||B|| K_\Phi \frac{1}{1 - e^{-u}} K_\Phi < \infty
\]

Term 3: Considering that \( R_k \) exists only for \( k > 0 \).

\[
\frac{1}{2} \sum_{k=0}^{\infty} |R_k^T ER_k| \leq \frac{1}{2} \sum_{k=1}^{\infty} |R_k^T|||E||R_k| \leq \Phi_U ||B|| K_\Phi \frac{1}{1 - e^{-u}} ||E|| \frac{1}{2} \sum_{k=1}^{\infty} |R_k|
\]

\[
\sum_{k=1}^{\infty} |R_k| = \sum_{k=1}^{\infty} \sum_{l=0}^{k-1} (\Phi_1^l Bu_{k-1-l}^r) < \sum_{k=1}^{\infty} \sum_{l=0}^{k-1} ||\Phi_1^l||||B|| K_\Phi e^{-u(k-1-l)}
\]

\[
\sum_{k=1}^{\infty} \sum_{l=0}^{k-1} ||\Phi_1^l||||B|| K_\Phi e^{-u(k-1-l)} = \lim_{N \to \infty} \sum_{j=0}^{N-1} \left( \sum_{l=0}^{j} ||\Phi_1^l|| \right) ||B|| K_\Phi e^{-(N-1-j)u} < 
\]

111
\[
\lim_{N \to \infty} \sum_{j=0}^{N-1} \left( \sum_{l=0}^{\infty} \| \Phi_l \| \right) \| B \| K_U e^{-(N-1-j)u} < K_\Phi \| B \| K_U \sum_{j=0}^{\infty} e^{-ju} = K_\Phi \| B \| K_U \frac{1}{1 - e^{-u}} \\
\sum_{k=1}^{\infty} |R_k| < K_\Phi \| B \| K_U \frac{1}{1 - e^{-u}} < \infty
\]

\( (8.36) \)

\( \sum_{k=0}^{\infty} |R_k^T E R_k| < \Phi_U \| B \| K_U \frac{1}{1 - e^{-u}} \| E \| \frac{1}{2} K_\Phi \| B \| K_U \frac{1}{1 - e^{-u}} < \infty \)

Term 4:

\[
\frac{1}{2} \sum_{k=0}^{\infty} \left| 2 \left( x_0^T (\Phi_1^T)^k + R_k^T \right) Q \tilde{x}_k \right| \leq \frac{1}{2} \sum_{k=0}^{\infty} \left( 2 |x_0^T| \| \Phi_1^T \| + 2 \| R_k^T \| \right) \| Q \| \| \tilde{x}_k \| \\
|\tilde{x}_k| \| H \| |r_k| < \| H \| A e^{-ak} \]

\( (8.37) \)

\[
\frac{1}{2} \sum_{k=0}^{\infty} \left| 2 \left( x_0^T (\Phi_1^T)^k + R_k^T \right) Q \tilde{x}_k \right| \leq \left( |x_0| \Phi_U + \Phi_U \| B \| K_U \frac{1}{1 - e^{-u}} \right) \| Q \| \| H \| A \sum_{k=0}^{\infty} e^{-ak} = \\
= \left( |x_0| \Phi_U + \Phi_U \| B \| K_U \frac{1}{1 - e^{-u}} \right) \| Q \| \| H \| A \frac{1}{1 - e^{-a}} < \infty
\]

Term 5:

\[
\frac{1}{2} \sum_{k=0}^{\infty} |\tilde{x}_k^T Q \tilde{x}_k| \leq \frac{1}{2} \sum_{k=0}^{\infty} |\tilde{x}_k^T| \| Q \| |\tilde{x}_k| < \frac{1}{2} \| H \| A \| Q \| \sum_{k=0}^{\infty} \| H \| A e^{-ak} = \\
= \frac{1}{2} \| H \|^2 A^2 \| Q \| \frac{1}{1 - e^{-a}} < \infty
\]

(8.38)
Term 6: Using \((8.27)\).

\[
\frac{1}{2} \sum_{k=0}^{\infty} \Bigg| 2 \left( x_0^T \left( \Phi_1^T \right)^k + R_k^T \right) K_{x2}^T R u_k \Bigg| \leq \\
\leq \frac{1}{2} \sum_{k=0}^{\infty} \left( 2|x_0^T| \| (\Phi_1^T)^k \| + 2 \| R_k^T \| < \Phi_U \| B \| K_U \frac{1}{1 - e^{-u}} \right) \| K_{x2}^T \| \| R \| \| u_k^r \| < \\
< \bigg( |x_0| \Phi_U + \Phi_U \| B \| K_U \frac{1}{1 - e^{-u}} \bigg) \| K_{x2}^T \| \| R \| \sum_{k=0}^{\infty} K_U e^{-uk} = \\
= \bigg( |x_0| \Phi_U + \Phi_U \| B \| K_U \frac{1}{1 - e^{-u}} \bigg) \| K_{x2}^T \| \| R \| K_U \frac{1}{1 - e^{-u}} < \infty
\]

Term 7: Using \((8.27)\).

\[
\frac{1}{2} \sum_{k=0}^{\infty} \left| (u_k^r)^T R u_k \right| \leq \frac{1}{2} \sum_{k=0}^{\infty} \| R \| |u_k^r|^2 < \frac{1}{2} \| R \| K_U^2 \sum_{k=0}^{\infty} e^{-2uk} = \\
= \frac{1}{2} \| R \| K_U^2 \frac{1}{1 - e^{-2u}} < \infty
\]

All the sums in the upper bound of the functional in \((8.32)\) are finite which means that the functional itself is finite.

8.10 Interconnected system structures to test tracker methods without and with disturbances

The notations in the structures are as follows:

1. Filter: the low-pass estimated disturbance post filter from \((3.37)\)
2. EST: the unknown disturbance estimator either \(UDEB\) or \(UDE\) (see subsections \(3.4.1\) and \(3.4.2\)).
3. Padé: the Padé filter representation of system delay from \((1.5)\)
4. S: saturation of input signals
5. SDA: system input saturation, delay (integer delay 0.2s) and actuator dynamics \((1.4)\).
6. \(G_{ac}\): lateral-directional aircraft dynamical model from \((1.3)\)
7. M: memory element to store previous value of variable, can be represented by the backward shift operator \(M = z^{-1}\).
8. WO: the discrete time equivalent of the $G_{filt}$ washout filter matrix from figure 1.3

9. LQ tracker part: denotes the part of systems which is used in comparison to the LQ tracker solution without disturbances. The whole structures always represent the test configurations to compare to the minimax tracker.

10. $C_{SS}$ the state space representation of the $\mathcal{H}_\infty$ controller

The schemes of the different control interconnections can be seen in the figures. All of the interconnections were constructed considering the discrete time system models.

Figure 8.2: Controlled system with feedforward control

Figure 8.3: Controlled system with integral feedforward control
Figure 8.4: Controlled system with modified LQ optimal state feedback control

Figure 8.5: Controlled system with PID tracker and yaw damper

Figure 8.6: The detailed PID part from figure 8.5 based on equation (2.63)
Figure 8.7: Controlled system with LQ Servo controller

Figure 8.8: Controlled system with MPC controller

Figure 8.9: Controlled system with $\mathcal{H}_\infty$ controller
8.11 The final gain vectors with the different design methodologies in the LQ tracker comparison

All the given gains are derived for the DT system.

**FF method:**

\[ K_x = \begin{bmatrix} -0.0008 & 0.0044 & 0.0237 & 0.0399 & -20.1962 & -194.3218 & -2.4201 & -10.6419 \\ 0.0008 & 0.0363 & -0.0076 & -0.2673 & -2.8990 & -7.3992 & -29.1728 & -211.4558 \end{bmatrix} \]

\[ K_r = \begin{bmatrix} 0.0237 \\ -0.0079 \end{bmatrix} \]

**IFF method:**

\[ K_x = \begin{bmatrix} 0.0024 & -0.0019 & 0.0529 & 0.0901 & -17.6787 & -179.5344 & -0.5347 & -18.7068 \\ 0.0073 & 0.0763 & 0.0208 & -0.5864 & 2.1294 & 22.6900 & -36.5452 & -203.2039 \end{bmatrix} \]

\[ K_r = \begin{bmatrix} 0.0219 \\ 0.0208 \end{bmatrix} \]

\[ K_I = \begin{bmatrix} 0.0339 \\ 0.0155 \end{bmatrix} \]

**Simple LQ method:**

\[ K_x = \begin{bmatrix} 0.0545 & 0.0727 & 0.5218 & -0.0003 & -3.7485 & 223.3756 & 2.0665 & -78.9814 \\ -0.0008 & -0.0055 & -0.0194 & -0.0056 & 0.5748 & -9.0222 & -0.0266 & 7.2788 \end{bmatrix} \]

**LQ Servo method:**

\[ K_x = \begin{bmatrix} 0.0869 & 0.1029 & 1.0053 & 0.0001 & -8.8378 & 318.4917 & 2.7243 & -87.1942 \\ 0.0003 & -0.0004 & 0.0043 & -0.0011 & -0.0018 & 0.5481 & 0.0116 & 1.2190 \end{bmatrix} \]

\[ K_I = \begin{bmatrix} 1.5040 \\ 0.0159 \end{bmatrix} \]

**MPC method:**

\[ K_x = \begin{bmatrix} 0.0066 & -0.0050 & 0.0128 & 0.2853 & -3.0457 & 4.3697 & -2.5176 & 1.8234 \\ 0.0005 & -0.0319 & 0.0008 & 0.9432 & -2.4266 & 0.6241 & -7.4291 & 3.9433 \end{bmatrix} \]

\[ K_u = \begin{bmatrix} 0.9457 & -0.0476 \\ -0.0476 & 0.8574 \end{bmatrix} \]

\[ K_{r1} = \begin{bmatrix} 0.0079 & 0.0040 & -0.0329 \\ 0.0007 & 0.0004 & -0.1088 \end{bmatrix} \]

\[ K_{r2} = \begin{bmatrix} 0.0030 & 0.0088 & -0.0032 \\ -0.0005 & 0.0004 & -0.0106 \end{bmatrix} \]
\( \mathcal{H}_\infty \) method (reduced controller state space). In figure 8.9, the state space dynamics of this controller is symbolized with \( C_{SS} \):

\[
A_c = \begin{bmatrix}
0.7418 & -0.3455 & -0.0315 & -0.0006 \\
0.3431 & -0.2249 & 0.1248 & 0.0051 \\
-0.0321 & -0.1238 & 0.9591 & -0.0071 \\
0.0045 & 0.0264 & 0.0162 & 0.9805
\end{bmatrix}
\]

\[
B_c = \begin{bmatrix}
0.0008 & 0.0043 & 0.0092 & -0.0333 & -9.2132 & -90.5167 & 2.0270 & 10.5366 & 0.0799 \\
-0.0006 & 0.0016 & -0.0206 & -0.0748 & -1.5915 & 62.8506 & 1.0224 & -5.2150 & 0.0767 \\
0.0004 & 0.0022 & -0.0518 & -0.0205 & -0.9701 & -2.5727 & 0.9290 & 0.4707 & 0.1415 \\
-0.0008 & 0.0000 & -0.0450 & -0.0717 & 0.2041 & -0.0515 & 1.5816 & -1.1176 & -0.0320
\end{bmatrix}
\]

\[
C_c = \begin{bmatrix}
-0.9167 & -0.6283 & -0.0262 & 0.0004 \\
-0.0040 & -0.0048 & 0.0097 & 0.0301
\end{bmatrix}
\]

\[
D_c = \begin{bmatrix}
-0.0276 & -0.0228 & -0.4273 & -0.1013 & 19.0981 & -342.6516 & -3.2976 & 38.3934 & 0.4352 \\
-0.0000 & 0.0002 & -0.0028 & -0.0038 & 0.0953 & 0.5955 & 0.1126 & -1.8737 & 0.0015
\end{bmatrix}
\]

**LQ tracker:**

\[
K_x = \begin{bmatrix}
0.0433 & 0.0608 & 0.4255 & 0.0001 & -4.3005 & 143.3256 & 0.4797 & -80.1696 \\
0.0012 & 0.0057 & 0.0129 & 0.0002 & 5.7886 & 87.0727 & 3.3236 & -46.3186
\end{bmatrix}
\]

\[
K_{S_1} + K_{\infty} = \begin{bmatrix}
5.1999 & -0.0001 \\
0.0646 & 0.0006
\end{bmatrix}
\]

\[
K_{S_2} = \begin{bmatrix}
-4.7738 & 0.0001 \\
-0.0562 & -0.0006
\end{bmatrix}
\]

### 8.12 Enlarged transients from the comparison of LQ tracker solution

![Tracking of roll angle](image1)

Figure 8.10: LQ roll angle tracking, part 1

![Tracking of roll angle](image2)

Figure 8.11: LQ roll angle tracking, part 2

118
Figure 8.12: LQ yaw rate tracking, part 1

Figure 8.13: LQ yaw rate tracking, part 2

Figure 8.14: LQ roll rate, part 1

Figure 8.15: LQ roll rate, part 2

Figure 8.16: LQ aileron deflections, part 1

Figure 8.17: LQ aileron deflections, part 2
8.13 Derivation of the finite horizon LQ optimal minimax tracking solution with Lagrange multiplier method

The functional (3.10) completed with the Lagrange multiplier and the centered state dynamic equation (8.8):

\[
L(\Delta x, \Delta \ddot{x}, \Delta \dot{u}, \Delta \ddot{d}, \lambda) =
\frac{1}{2} \left( Q \Delta x_0, \Delta x_0 \right) - \frac{1}{2} \left( Q \Delta \ddot{x}_0, \Delta \ddot{x}_0 \right) + \frac{1}{2} \left( R_u \Delta \dot{u}_0, \Delta \dot{u}_0 \right) - \frac{1}{2} \left( R_d \Delta \ddot{d}_0, \Delta \ddot{d}_0 \right) + \left( \lambda_0, a - \Delta x_0 \right) +
\frac{1}{2} \left( Q \Delta x_1, \Delta x_1 \right) - \frac{1}{2} \left( Q \Delta \ddot{x}_1, \Delta \ddot{x}_1 \right) + \frac{1}{2} \left( R_u \Delta \dot{u}_1, \Delta \dot{u}_1 \right) - \frac{1}{2} \left( R_d \Delta \ddot{d}_1, \Delta \ddot{d}_1 \right) + \left( \lambda_1, \Phi \Delta x_0 + B \Delta \dot{u}_0 + B_d \Delta \ddot{d}_0 - \Delta x_1 \right) +
+ \ldots +
\frac{1}{2} \left( Q \Delta x_{N-1}, \Delta x_{N-1} \right) - \frac{1}{2} \left( Q \Delta \ddot{x}_{N-1}, \Delta \ddot{x}_{N-1} \right) + \frac{1}{2} \left( R_u \Delta \dot{u}_{N-1}, \Delta \dot{u}_{N-1} \right) - \frac{1}{2} \left( R_d \Delta \ddot{d}_{N-1}, \Delta \ddot{d}_{N-1} \right) + \left( \lambda_{N-1}, \Phi \Delta x_{N-2} + B \Delta \dot{u}_{N-2} + B_d \Delta \ddot{d}_{N-2} - \Delta x_{N-1} \right) +
+ \ldots +
\frac{1}{2} \left( Q \Delta x_N, \Delta x_N \right) - \frac{1}{2} \left( Q \Delta \ddot{x}_N, \Delta \ddot{x}_N \right) + \left( \lambda_N, \Phi \Delta x_{N+1} + B \Delta \dot{u}_{N+1} + B_d \Delta \ddot{d}_{N+1} - \Delta x_N \right)
\]

The derivatives of the functional (8.41) with respect to \( \Delta \dot{u}_i \), \( \Delta x_i \), and \( \Delta \ddot{d}_i \) are the following:
\[
\begin{align*}
\frac{\partial L}{\partial \Delta \hat{u}_0} &= R_u \Delta \hat{u}_0 + B^T \lambda_1 = 0 \\
\frac{\partial L}{\partial \Delta \hat{u}_1} &= R_u \Delta \hat{u}_1 + B^T \lambda_2 = 0 \\
\ldots \\
\frac{\partial L}{\partial \Delta \hat{u}_{N-1}} &= R_u \Delta \hat{u}_{N-1} + B^T \lambda_N = 0
\end{align*}
\] (8.42)

\[
\begin{align*}
\frac{\partial L}{\partial \Delta \hat{u}_0} &= -Q \Delta \hat{x}_0 + Q \Delta x_0 - \lambda_0 + \Phi^T \lambda_1 = 0 \\
\frac{\partial L}{\partial \Delta \hat{u}_1} &= -Q \Delta \hat{x}_1 + Q \Delta x_1 - \lambda_1 + \Phi^T \lambda_2 = 0 \\
\ldots \\
\frac{\partial L}{\partial \Delta \hat{u}_{N-1}} &= -Q \Delta \hat{x}_{N-1} + Q \Delta x_{N-1} - \lambda_{N-1} + \Phi^T \lambda_N = 0 \\
\frac{\partial L}{\partial \Delta \hat{x}_N} &= -Q \Delta \hat{x}_N + Q \Delta x_N - \lambda_N = 0
\end{align*}
\] (8.43)

\[
\begin{align*}
\frac{\partial L}{\partial \Delta \tilde{d}_0} &= -\gamma^2 R_d \Delta \tilde{d}_0 + B_d^T \lambda_1 = 0 \\
\frac{\partial L}{\partial \Delta \tilde{d}_1} &= -\gamma^2 R_d \Delta \tilde{d}_1 + B_d^T \lambda_2 = 0 \\
\ldots \\
\frac{\partial L}{\partial \Delta \tilde{d}_{N-1}} &= -\gamma^2 R_d \Delta \tilde{d}_{N-1} + B_d^T \lambda_N = 0
\end{align*}
\] (8.44)

From (8.42) the optimal control input:
\[
\Delta \hat{u}_k = -R_u^{-1} B^T \lambda_{k+1}
\]

From (8.43) the costate variable:
\[
\lambda_k = Q (\Delta x_k - \Delta \hat{x}_k) + \Phi^T \lambda_{k+1}
\]

From (8.44) the worst case disturbance:
\[
\Delta \tilde{d}_k^* = \frac{1}{\gamma^2} R_d^{-1} B_d^T \lambda_{k+1}
\]

Substituting the above expressions into (3.8) one gets the Hamiltonian matrix as follows:
\[
\begin{bmatrix}
\Delta x_{k+1} \\
\lambda_k
\end{bmatrix} =
\begin{bmatrix}
\Phi & -B R_u^{-1} B^T + \frac{1}{\gamma^2} B_d R_d^{-1} B_d^T \\
Q & \Phi^T B_d R_d^{-1} B_d^T
\end{bmatrix}
\begin{bmatrix}
\Delta x_k \\
\lambda_{k+1}
\end{bmatrix} -
\begin{bmatrix}
0 \\
Q
\end{bmatrix}
\Delta \tilde{x}_k
\] (8.45)
The equations for $k = N - 1$ result as (considering $\lambda_N = Q \Delta x_N - Q \Delta \hat{x}_N$):

$$\Delta x_N = \Phi \Delta x_{N-1} - \left( BR_u^{-1}B^T - \frac{1}{\gamma^2} B_d R_d^{-1} B_d^T \right) Q (\Delta x_N - \Delta \hat{x}_N) =$$

$$= \Phi \Delta x_{N-1} - M_R Q \Delta x_N + M_R Q \Delta \hat{x}_N$$

$$\Delta x_N = [I + M_R Q]^{-1} \Phi \Delta x_{N-1} + [I + M_R Q]^{-1} M_R Q \Delta \hat{x}_N$$

$$\lambda_{N-1} = Q \Delta x_{N-1} + \Phi^T Q \Delta x_N - \Phi^T Q \Delta \hat{x}_N - Q \Delta \hat{x}_{N-1}$$

From the last equation in (8.46) the structure of the costate variable results as:

$$\lambda_k = P_k \Delta x_k + S_k \Delta \hat{x}_{k+1} - Q \Delta \hat{x}_k$$

$$\lambda_N = Q \Delta x_N - Q \Delta \hat{x}_N \quad \rightarrow \quad P_N = Q, \quad S_N = 0$$

(8.47)

### 8.14 Finiteness of the infinite horizon cost functional for $l_1$ ($l_2$) references and disturbances

Substituting the control input (8.33) and the state from (8.34) into (8.32) results in:

$$J (x, \dot{x}, u, \dot{d}) =$$

$$= \frac{1}{2} \sum_{k=0}^{\infty} \{(\Phi^k x_0 + R_k)^T Q (\Phi^k x_0 + R_k) - 2 (\Phi^k x_0 + R_k)^T Q \hat{x}_k + \hat{x}_k^T Q \hat{x}_k +$$

$$+ (-K x_2 x_k + u_k^r)^T R_a (-K x_2 x_k + u_k^r) - \gamma^2 \dot{d}_k^T R_d \dot{d}_k \} =$$

$$= \frac{1}{2} \sum_{k=0}^{\infty} \{(\Phi^k x_0 + R_k)^T Q (\Phi^k x_0 + R_k) - 2 (\Phi^k x_0 + R_k)^T Q \hat{x}_k + \hat{x}_k^T Q \hat{x}_k +$$

$$+ (-K x_2 (\Phi^k x_0 + R_k) + u_k^r)^T R_a (-K x_2 (\Phi^k x_0 + R_k) + u_k^r) - \gamma^2 \dot{d}_k^T R_d \dot{d}_k \}$$

(8.48)

Grouping the terms and applying $E = Q + K^T x_a K_{x2}$ the following form of the functional results:

$$J (x, \dot{x}, u, \dot{d}) = \frac{1}{2} \sum_{k=0}^{\infty} \{ x_0^T (\Phi^k)^T E \Phi^k x_0 + 2 x_0^T (\Phi^k)^T E R_k + R_k^T E R_k -$$

$$- 2 (x_0^T (\Phi^k)^k + R_k^T) Q \hat{x}_k + \hat{x}_k^T Q \hat{x}_k - 2 (x_0^T (\Phi^k)^k + R_k^T) Q x_k + \hat{x}_k^T Q \hat{x}_k -$$

$$- 2 (x_0^T (\Phi^k)^k + R_k^T) Q \hat{x}_k + \hat{x}_k^T Q \hat{x}_k - 2 (x_0^T (\Phi^k)^k + R_k^T) K_{x2}^T R_a u_k^r +$$

$$+ (u_k^r)^T R_a u_k^r - \gamma^2 \dot{d}_k^T R_d \dot{d}_k \}$$

(8.49)
This is exactly the same form as in (8.31) but here $R_k$ means a modified sum and an extra term $-\gamma^2 \tilde{d}_k^T R_d \tilde{d}_k$ is included. An upper bound for the functional value can be constructed by taking the absolute values of the terms in the summation:

$$J(x, \tilde{x}, u, \tilde{d}) \leq \frac{1}{2} \sum_{k=0}^{\infty} \left\{ \left| x_0^T (\Phi_1^k)^k E \Phi_1^k x_0 \right| + 2 \left| x_0^T (\Phi_1^k)^k E R_k \right| + \left| R_k^T E R_k \right| + 2 \left| x_0^T (\Phi_1^k)^k R_k^T \right| Q \tilde{x}_k + \left| \tilde{x}_k Q \tilde{x}_k \right| + 2 \left( x_0^T (\Phi_1^k)^k + R_k^T \right) K_{x2}^T R_u u_k^k \right\} + \sum_{7} + \left( u_k^T R_u u_k^k \right) + \left| \gamma^2 \tilde{d}_k^T R_d \tilde{d}_k \right| \right\} \tag{8.50}$$

If the infinite sum of every term in (8.50) is finite, then the functional value is surely finite. Because the only difference from the LQ optimal case in (8.32) is the different meaning of $R_k$ and the extra term 8 only the terms including $R_k$ and Term 8 should be examined here, for the others the proof is the same as in section 8.9. Let’s examine at first the form of $R_k$:

$$R_k = \sum_{l=0}^{k-1} \left( \Phi_1^l [B \quad B_d] \left[ \begin{array}{c} u_{k-1-l} \\ \tilde{d}_{k-1-l} \end{array} \right] \right) \tag{8.51}$$

Redefining $u_{k-1-l}^r$ as $u_{k-1-l}^r = \left[ \begin{array}{c} u_{k-1-l}^r \\ \tilde{d}_{k-1-l} \end{array} \right]$ formally the same upper bound can be derived as in (8.27) considering that $u_{k-1-l}^r$, $d_{k-1-l}$ and $\tilde{d}_{k-1-l}$ are $l_1$ signals. $B$ can be also redefined as $B = [B \quad B_d]$. From this point the proof for the terms including $R_k$ is formally exactly the same as in section 8.9. This way only the last term in (8.50) should be examined:

**Term 8**: Using $|\tilde{d}_k| < K_D e^{-d_k}$

$$\frac{1}{2} \sum_{k=0}^{\infty} \left| \gamma^2 \tilde{d}_k^T R_d \tilde{d}_k \right| \leq \frac{1}{2} \sum_{k=0}^{\infty} \gamma^2 \| R_d \| \| \tilde{d}_k \|^2 < \frac{1}{2} \gamma^2 \| R_d \| K_D^2 \sum_{k=0}^{\infty} e^{-2d_k} = \frac{1}{2} \gamma^2 \| R_d \| K_D^2 \frac{1}{1 - e^{-2d_1}} < \infty \tag{8.52}$$

All the sums in the upper bound of the functional in (8.50) are finite which means that the functional itself is finite.

### 8.15 The final gain vectors with the different design methodologies in the MM tracker comparison

In the FF, IFF, Simple LQ and LQ Servo cases exactly the same gains were used as in the comparison with the LQ tracker. So, only the MPC, $\mathcal{H}_\infty$ and MM tracker gains are collected here:
MPC method:

\[
K_x = \begin{bmatrix} K_x(1) & K_x(2) \end{bmatrix}
\]

\[
K_x(1) = \begin{bmatrix}
0.0107 & 0.0353 & 0.0892 & 0.0002 & 0.0474 & -0.0416 & 0.0628 \\
0.0002 & 0.0006 & 0.0016 & 0.0001 & 0.0008 & -0.0007 & 0.0110
\end{bmatrix}
\]

\[
K_x(2) = \begin{bmatrix}
-0.0441 & 0.0899 & -0.0426 & 0.1369 & -0.0323 & 0.2173 & -0.0050 \\
-0.0007 & 0.0014 & -0.0007 & 0.0021 & -0.0005 & 0.0033 & 0.0000
\end{bmatrix}
\]

\[
K_u = \begin{bmatrix} 0.6466 & -0.0535 \\ -0.0054 & 0.9989 \end{bmatrix}
\]

\[
K_{r1} = \begin{bmatrix} 0.1635 & 0.1784 & -0.0010 \\ 0.0024 & 0.0033 & -0.0005 \end{bmatrix}
\]

\[
K_{r2} = \begin{bmatrix} 0.0818 & 0.0892 & -0.0005 \\ 0.0012 & 0.0016 & -0.0003 \end{bmatrix}
\]

\[
K_d = \begin{bmatrix} 0.1188 & 0.0535 \\ 0.0018 & 0.0008 \end{bmatrix}
\]

\[H_{\infty} \text{ method (reduced controller state space):}\]

\[
A_c = \begin{bmatrix}
0.2878 & 0.0834 & 0.0551 & -0.0090 \\
-0.0809 & 0.9890 & -0.0127 & 0.0010 \\
-0.0523 & -0.0126 & 0.9589 & 0.0591 \\
-0.0067 & 0.0008 & 0.0392 & 0.0032
\end{bmatrix}
\]

\[
B_c = \begin{bmatrix} B_c(1) & B_c(2) \end{bmatrix}
\]

\[
B_c(1) = \begin{bmatrix}
-0.0042 & 0.0757 & -0.0781 & 0.0091 & -16.6079 \\
-0.1342 & 0.0079 & -0.4688 & 0.0132 & 0.7619 \\
-0.1351 & 0.0124 & -0.0169 & 0.0012 & 0.1618 \\
-0.0068 & -0.0056 & 0.0059 & -0.0166 & 0.8300
\end{bmatrix}
\]

\[
B_c(2) = \begin{bmatrix}
-69.8431 & 5.5057 & 17.1479 & -0.0112 & 0.0033 & 0.1023 \\
0.2475 & 1.1992 & 0.3170 & -0.0389 & 0.0042 & 0.3539 \\
0.7791 & 1.2302 & 1.1677 & -0.0196 & -0.0020 & -0.3341 \\
-3.3369 & -0.5050 & -11.9873 & -0.0008 & 0.0000 & -0.0248
\end{bmatrix}
\]

\[
C_c = \begin{bmatrix}
0.7399 & 0.0012 & -0.0116 & 0.0059 \\
-0.0217 & 0.0022 & -0.0089 & 0.1244
\end{bmatrix}
\]

\[
D_c = \begin{bmatrix} D_c(1) & D_c(2) \end{bmatrix}
\]

\[
D_c(1) = \begin{bmatrix}
-0.1134 & 0.1506 & -0.0936 & -0.0035 & 4.2331 \\
0.0055 & 0.0088 & 0.0045 & -0.0008 & -0.3578
\end{bmatrix}
\]

\[
D_c(2) = \begin{bmatrix}
-128.1477 & -1.1578 & 24.0221 & -0.0274 & -0.0189 & 0.2296 \\
11.7589 & 0.1212 & -3.0684 & 0.0010 & 0.0009 & -0.0090
\end{bmatrix}
\]

\[MM \text{ tracker:}\]

\[
K_x = \begin{bmatrix}
0.0313 & 0.0434 & 0.2191 & -0.0000 & -1.9408 & 75.0150 & -0.2801 & -80.9175 \\
0.0033 & 0.0081 & 0.0244 & 0.0003 & 5.8777 & 98.8392 & 3.5033 & -47.5533
\end{bmatrix}
\]
\[ K_{r\infty} - K_{S_2} = \begin{bmatrix} 4.2834 & 216.49 \\ 0.34746 & -1585.6 \end{bmatrix} \]

\[ K_{S_2} = \begin{bmatrix} -4.0643 & 0.0002 \\ -0.3231 & -0.0009 \end{bmatrix} \]

\[ K_{d\infty} = \begin{bmatrix} -0.1767 & -0.1048 \\ -0.0177 & 0.1250 \end{bmatrix} \]

### 8.16 Enlarged transients from the comparison of MM tracker solution

![Tracking of roll angle](image1)

*Figure 8.20: MM roll angle tracking, part 1*

![Tracking of roll angle](image2)

*Figure 8.21: MM roll angle tracking, part 2*

![Filtered yaw rate](image3)

*Figure 8.22: MM filtered yaw rate tracking, part 1*

![Filtered yaw rate](image4)

*Figure 8.23: MM filtered yaw rate tracking, part 2*
Figure 8.24: MM roll rate, part 1

Figure 8.25: MM roll rate, part 2

Figure 8.26: MM aileron deflections, part 1

Figure 8.27: MM aileron deflections, part 2

Figure 8.28: MM rudder deflections, part 1

Figure 8.29: MM rudder deflections, part 2
8.17 Difference between initial and final angular rate biases

Table 8.1: Angular rate bias value differences

<table>
<thead>
<tr>
<th>Flight</th>
<th>p</th>
<th>q</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+58%</td>
<td>-127%</td>
<td>-19%</td>
</tr>
<tr>
<td>2</td>
<td>+59%</td>
<td>-%</td>
<td>-38%</td>
</tr>
<tr>
<td>3</td>
<td>-81%</td>
<td>+142%</td>
<td>-2.4%</td>
</tr>
<tr>
<td>4</td>
<td>-30%</td>
<td>+61%</td>
<td>-8%</td>
</tr>
<tr>
<td>5</td>
<td>-20%</td>
<td>+31%</td>
<td>-3.4%</td>
</tr>
<tr>
<td>6</td>
<td>+23%</td>
<td>+34%</td>
<td>+13%</td>
</tr>
<tr>
<td>7</td>
<td>-49%</td>
<td>+8.5%</td>
<td>+2%</td>
</tr>
<tr>
<td>8</td>
<td>-27.5%</td>
<td>+29%</td>
<td>+10%</td>
</tr>
</tbody>
</table>

8.18 Aircraft attitude estimator related equations and detailed derivation

The rotation matrices with Euler angles:

\[
T_\phi = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(\phi) & \sin(\phi) \\
0 & -\sin(\phi) & \cos(\phi)
\end{bmatrix}
\] (8.53)

\[
T_\theta = \begin{bmatrix}
\cos(\theta) & 0 & -\sin(\theta) \\
0 & 1 & 0 \\
\sin(\theta) & 0 & \cos(\theta)
\end{bmatrix}
\] (8.54)

\[
T_\psi = \begin{bmatrix}
\cos(\psi) & \sin(\psi) & 0 \\
-\sin(\psi) & \cos(\psi) & 0 \\
0 & 0 & 1
\end{bmatrix}
\] (8.55)

The matrices defined in (4.19)

\[
A_1(\rho) = \begin{pmatrix}
-\frac{1}{2} & 0 \\
0 & \rho
\end{pmatrix}
\]

\[
A_2(\vartheta) = \begin{pmatrix}
-\vartheta_1 & -\vartheta_2 \\
\vartheta_0 & -\vartheta_3 \\
\vartheta_3 & \vartheta_0 \\
-\vartheta_2 & \vartheta_1
\end{pmatrix}
\] (8.57)
\[
A(\rho, b) = \left( -\frac{1}{2} \right) \cdot \begin{bmatrix}
0 & p - b_p & q - b_q & r - b_r \\
-p + b_p & 0 & -r + b_r & q - b_q \\
-q + b_q & r - b_r & 0 & -p + b_r \\
-r + b_r & -q + b_q & p - b_p & 0 \\
\end{bmatrix} \tag{8.58}
\]

\[
L_1(\rho) = \left( -\frac{1}{2} \right) \begin{bmatrix}
\rho_1 - \rho_2 & \rho_2 - \rho_3 & \rho_3 - \rho_0 \\
\rho_0 - \rho_3 & \rho_3 - \rho_2 & \rho_2 - \rho_1 \\
\rho_1 - \rho_2 & \rho_2 - \rho_1 & \rho_0 \\
\end{bmatrix} \tag{8.59}
\]

### 8.18.1 The detailed derivation of the filter state dynamic equation

A ‘steady state’ solution \(\bar{x}_{k+1}\) of the trapezoidal integration formula \(\text{(4.8)}\) can be:

\[
\bar{x}_{k+1} = x_k + f(x_k, \rho_k, n_k) + f(\bar{x}_{k+1}, \rho_{k+1}, n_{k+1}) \Delta t \tag{8.60}
\]

Substituting \(\text{(4.9)}\) and \(\text{(4.10)}\) into \(\text{(8.60)}\) (by evaluating them in discrete time steps) results in:

\[
\begin{align*}
\bar{\rho}_{k+1} & \approx \rho_k + \frac{\dot{\rho}_k + \dot{\bar{\rho}}_{k+1}}{2} \Delta t = \\
& = \rho_k + \frac{\Delta t}{2} \left( A_1(\rho_k) \rho_k + A_2(\rho_k) b_k + L_1(\rho_k) v^\rho_k \right) + \\
& + \frac{\Delta t}{2} \left( A(\rho_{k+1}, b_{k+1}) \rho_{k+1} + L_1(\rho_{k+1}) v^\rho_{k+1} \right) \\
\bar{b}_{k+1} & \approx b_k + \frac{v^b_k + v^b_{k+1}}{2} \Delta t \\
\end{align*} \tag{8.61}
\]

Assuming \(b_{k+1} = b_k\) (slowly varying bias value), considering the predicted quaternion \(\bar{\rho}_{k+1} = \rho_k + \dot{\rho}_k \Delta t\) in the \(L_1\) term, introducing fictitious noises \((\bar{v}^\rho_{k+1}, \bar{v}^b_{k+1})\) and reordering the terms in \(\text{(8.61)}\) results in:

\[
\begin{align*}
\left( I - \frac{\Delta t}{2} A(\rho_{k+1}, b_k) \right) \bar{\rho}_{k+1} & = \left( I + \frac{\Delta t}{2} A_1(\rho_k) \right) \rho_k + \\
& + \frac{\Delta t}{2} A_2(\rho_k) b_k + \Delta t L_1(\bar{\rho}_{k+1}) \bar{v}^\rho_{k+1} \\
\bar{b}_{k+1} & = b_k + \Delta t \bar{v}^b_{k+1}
\end{align*} \tag{8.62}
\]

\(\text{(8.62)}\) has a closed form solution, if \(M_{k+1}^+\) is invertible. From \(\text{(8.58)}\) and \(\text{(8.62)}\) \(M_{k+1}^+\) and its determinant results as:
\[ M_{k+1}^+ = \frac{\Delta t}{4} \begin{bmatrix} 4/\Delta t & \bar{\rho}_1 & \bar{\rho}_2 & \bar{\rho}_3 \\ -\bar{\rho}_1 & 4/\Delta t & -\bar{\rho}_2 & \bar{\rho}_3 \\ -\bar{\rho}_2 & \bar{\rho}_3 & 4/\Delta t & -\bar{\rho}_1 \\ -\bar{\rho}_3 & -\bar{\rho}_2 & -\bar{\rho}_1 & 4/\Delta t \end{bmatrix} \]

\[ \det \left( M_{k+1}^+ \right) = 1 + \frac{1}{8} |\bar{\rho}|^2 + \frac{1}{256} |\bar{\rho}|^4 > 0 \forall \bar{\rho} \]

where \( \bar{\rho} = [p - b_p \quad q - b_q \quad r - b_r]^T \)

So, it is always invertible and applying its inverse (4.11) can be obtained.

**End of derivation**

The quaternion-based transformation matrix from earth to body coord. sys.:

\[ T^{BE}(\varrho) = \begin{bmatrix} \varrho_0^2 + \varrho_1^2 - \varrho_2^2 - \varrho_3^2 & 2(\varrho_1\varrho_2 + \varrho_0\varrho_3) & 2(\varrho_1\varrho_3 - \varrho_0\varrho_2) \\ 2(\varrho_1\varrho_2 - \varrho_0\varrho_3) & \varrho_0^2 - \varrho_1^2 + \varrho_2^2 - \varrho_3^2 & 2(\varrho_2\varrho_3 + \varrho_0\varrho_1) \\ 2(\varrho_1\varrho_3 + \varrho_0\varrho_2) & 2(\varrho_2\varrho_3 - \varrho_0\varrho_1) & \varrho_0^2 - \varrho_1^2 - \varrho_2^2 + \varrho_3^2 \end{bmatrix} \]

The noise covariance matrices from \( \mu_{NAV} \) data:

\[ \text{Ra} = \begin{bmatrix} 0.1204 & -0.00435 & 0.000456 \\ -0.00435 & 0.1554 & -0.0018 \\ 0.000456 & -0.0018 & 0.10967 \end{bmatrix} / 180^2 \cdot \pi^2 \] (8.65)

\[ \text{RnH} = \bar{\text{RnH}} \cdot 100 = \]

\[ = \begin{bmatrix} 0.000436 & 0.000165 & 0.000018 \\ 0.000165 & 0.00037 & -0.000005 \\ 0.000018 & -0.000005 & 0.000105 \end{bmatrix} \] (8.66)

\[ \text{Rna} = \bar{\text{Rna}} \cdot 100 = \]

\[ = \begin{bmatrix} 0.0012 & 0.0002 & -0.00058 \\ 0.0002 & 0.00038 & -0.0001 \\ -0.00058 & -0.0001 & 0.000295 \end{bmatrix} \] (8.67)

\[ R_{GPS} = 0.00273 \] (8.68)

\[ k_0 = 1e - 10 \] (8.69)

Wind disturbance, bias and noise covariance values used in attitude estimation HIL testing:

\[ \text{Ve} = [2 \quad 4 \quad 0]^T \] (8.70)
### Table 8.2: HIL biases (Ax. = axis, vel. = velocity)

<table>
<thead>
<tr>
<th>Ax.</th>
<th>Rates</th>
<th>Magn.</th>
<th>Acc.</th>
<th>GPS vel.</th>
<th>IAS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[rad/s]</td>
<td>[nT]</td>
<td>[m/s²]</td>
<td>[m/s]</td>
<td>[m/s]</td>
</tr>
<tr>
<td>x</td>
<td>0.05</td>
<td>500</td>
<td>2</td>
<td>0.5</td>
<td>2</td>
</tr>
<tr>
<td>y</td>
<td>-0.03</td>
<td>100</td>
<td>-0.5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>z</td>
<td>0</td>
<td>2000</td>
<td>1</td>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 8.3: HIL noise covariances (Ax. = axis, vel. = velocity)

<table>
<thead>
<tr>
<th>Ax.</th>
<th>Rates</th>
<th>Magn.</th>
<th>Acc.</th>
<th>GPS vel.</th>
<th>IAS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[rad/s]</td>
<td>[nT]</td>
<td>[m/s²]</td>
<td>[m/s]</td>
<td>[m/s]</td>
</tr>
<tr>
<td>x</td>
<td>1e-5</td>
<td>1664</td>
<td>1e-3</td>
<td>1e-3</td>
<td>1e-3</td>
</tr>
<tr>
<td>y</td>
<td>1e-5</td>
<td>1664</td>
<td>1e-3</td>
<td>1e-3</td>
<td>0</td>
</tr>
<tr>
<td>z</td>
<td>1e-5</td>
<td>1664</td>
<td>1e-3</td>
<td>1e-3</td>
<td>0</td>
</tr>
</tbody>
</table>

### 8.19 Enlarged figures from HIL test results N/NB/W case

![HIL Roll Angle](image1)

**Figure 8.30:** Estimated $\phi$ with sensor noises and wind disturbance in HIL simulation

![HIL Pitch Angle](image2)

**Figure 8.31:** Estimated $\theta$ with sensor noises and wind disturbance in HIL simulation

![HIL Yaw Angle](image3)

**Figure 8.32:** Estimated $\psi$ with sensor noises and wind disturbance in HIL simulation
8.20 The applied wind estimation algorithm

The algorithm from [15] is based-on the wind triangle shown in figure 8.33.

![Wind Triangle Diagram](image)

**Figure 8.33:** The horizontal wind triangle

It assumes that the horizontal air relative velocity \( V'_a \) can be obtained from IAS via a scale factor as \( V'_a = \frac{IAS}{\sqrt{s_f}} \). It also assumes that the horizontal wind velocity \( V'_w \) and its direction \( \psi_w \) are constant. Based-on these assumptions it formulates the cosine rule for the wind triangle:

\[
\left( V'_{E'} \right)^2 + \left( V'_w \right)^2 - 2 V'_{E'} V'_w \cos(\psi_w - \psi_{GPS}) = \left( V'_a \right)^2 = \frac{IAS^2}{s_f} \tag{8.71}
\]

Using \( IAS^2 \) as a measurement an EKF can be built with states \( x = [V_w \ \psi_w \ s_f]^T \) and the following state dynamic and measurement equations:

\[
x_{k+1} = F x_k + w_k, \quad F = I
\]

\[
y_k = h(x_k) = IAS^2 = s_f \left[ \left( V'_{E'} \right)^2 + \left( V'_w \right)^2 - 2 V'_{E'} V'_w \cos(\psi_w - \psi_{GPS}) \right] + v_k \tag{8.72}
\]

\[
w_k \sim N(0, \sigma_w), \quad v_k \sim N(0, \sigma_v)
\]
8.21 Enlarged figures from HIL test results with wind correction N/NB/W case

Figure 8.34: Estimated $\phi$ with sensor noise and wind disturbance in HIL simulation with wind correction

Figure 8.35: Estimated $\theta$ with sensor noise and wind disturbance in HIL simulation with wind correction

Figure 8.36: Estimated $\psi$ with sensor noise and wind disturbance in HIL simulation with wind correction

8.22 Detailed robustness test results

8.22.1 The physical meaning of parameter uncertainties

This is studied for every coefficient separately as follows:

- $L_p = \frac{\pi S b^2}{2 V_0} \left( \frac{I_1}{I_x} c_{L_p} + \frac{I_2}{I_z} c_{N_p} \right)$ For this coefficient the uncertainty can result from the trim velocity $V_0$ (and so the dynamic pressure $\bar{\gamma}$), from the inertia terms ($I_1$, $I_2$, $I_x$ and $I_z$) and from the uncertainty of the slopes of $c_{L_p}$ and $c_{N_p}$ derivatives. So either different flight velocity, change in aircraft mass and inertias and difference in aerodynamic properties can result in different $L_p$ parameter.

- $L_r = \frac{\pi S b^2}{2 V_0} \left( \frac{I_1}{I_r} c_{L_r} + \frac{I_2}{I_z} c_{N_r} \right)$ For this coefficient the uncertainty can result from the trim velocity $V_0$ (and so the dynamic pressure $\bar{\gamma}$), from the inertia terms ($I_1$, $I_2$, $I_x$ and $I_z$) and from the uncertainty of the slopes of $c_{L_r}$ and $c_{N_r}$ derivatives. So either different flight velocity, change in aircraft mass and inertias and difference in aerodynamic properties can result in different $L_r$ parameter.
The applied saw-like disturbance of the roll angle can be seen in the following figure:

\[ I_x \text{ and } I_z \] and from the uncertainty of the slopes of \( c_{L_x} \) and \( c_{N_x} \) derivatives. So either different flight velocity, change in aircraft mass and inertias and difference in aerodynamic properties can result in different \( N_r \) parameter.

- \( L_{\delta a} = \bar{q}_s b \left( \frac{I_x}{I_r} c_{L_{\delta a}} + \frac{I_z}{I_r} c_{N_{\delta a}} \right) \) For this coefficient the uncertainty can result from the dynamic pressure \( \bar{q}_s \), from the inertia terms \( I_1, I_2, I_x \) and \( I_z \) and from the uncertainty of the slopes of \( c_{L_{\delta a}} \) and \( c_{N_{\delta a}} \) derivatives. So either different flight velocity, change in aircraft mass and inertias and difference in aerodynamic properties can result in different \( L_{\delta a} \) parameter.

- \( L_{\delta r} = \bar{q}_s b \left( \frac{I_x}{I_r} c_{L_{\delta r}} + \frac{I_z}{I_r} c_{N_{\delta r}} \right) \) For this coefficient the uncertainty can result from the dynamic pressure \( \bar{q}_s \), from the inertia terms \( I_1, I_2, I_x \) and \( I_z \) and from the uncertainty of the slopes of \( c_{L_{\delta r}} \) and \( c_{N_{\delta r}} \) derivatives. So either different flight velocity, change in aircraft mass and inertias and difference in aerodynamic properties can result in different \( L_{\delta r} \) parameter.

- \( N_p = \frac{\bar{q}_s l_{S_2}^2}{2 l_0} \left( \frac{I_x}{I_r} c_{L_p} + \frac{I_z}{I_r} c_{N_p} \right) \) For this coefficient the uncertainty can result from the trim velocity \( V_0 \) (and so the dynamic pressure \( \bar{q} \)), from the inertia terms \( I_1, I_2, I_x \) and \( I_z \) and from the uncertainty of the slopes of \( c_{L_p} \) and \( c_{N_p} \) derivatives. This is a special coefficient which is almost zero and its uncertainty is extremely large. Considering the detailed inertia terms \( I_1 \) and \( I_3 \) one can reorganize it as: \( N_p = \frac{l_{S_2}^2}{(l_1 l_2 - l_{S_2})^2 l_0} c_{L_p} + \frac{l_{S_2}^2}{(l_1 l_2 - l_{S_2})^2 l_0} c_{N_p} \). Considering \( C_{l_p} < 0 \) & \( C_{n_p} > 0 \), \(|C_{l_p}| \gg |C_{n_p}| \) and \( I_x \gg I_{xx} \), \( N_p \) can be zero, negative or positive. That’s why it is a very uncertain parameter.

- \( N_r = \frac{\bar{q}_s l_{S_2}^2}{2 l_0} \left( \frac{I_x}{I_r} c_{L_r} + \frac{I_z}{I_r} c_{N_r} \right) \) For this coefficient the uncertainty can result from the trim velocity \( V_0 \) (and so the dynamic pressure \( \bar{q} \)), from the inertia terms \( I_1, I_3, I_x \) and \( I_z \) and from the uncertainty of the slopes of \( c_{L_r} \) and \( c_{N_r} \) derivatives. So either different flight velocity, change in aircraft mass and inertias and difference in aerodynamic properties can result in different \( N_r \) parameter.

- \( N_{\delta a} = \bar{q}_s b \left( \frac{I_x}{I_r} c_{L_{\delta a}} + \frac{I_z}{I_r} c_{N_{\delta a}} \right) \) For this coefficient the uncertainty can result from the dynamic pressure \( \bar{q}_s \), from the inertia terms \( I_1, I_3, I_x \) and \( I_z \) and from the uncertainty of the slopes of \( c_{L_{\delta a}} \) and \( c_{N_{\delta a}} \) derivatives. So either different flight velocity, change in aircraft mass and inertias and difference in aerodynamic properties can result in different \( N_{\delta a} \) parameter.

- \( N_{\delta r} = \bar{q}_s b \left( \frac{I_x}{I_r} c_{L_{\delta r}} + \frac{I_z}{I_r} c_{N_{\delta r}} \right) \) For this coefficient the uncertainty can result from the dynamic pressure \( \bar{q}_s \), from the inertia terms \( I_1, I_3, I_x \) and \( I_z \) and from the uncertainty of the slopes of \( c_{L_{\delta r}} \) and \( c_{N_{\delta r}} \) derivatives. So either different flight velocity, change in aircraft mass and inertias and difference in aerodynamic properties can result in different \( N_{\delta r} \) parameter.

The applied saw-like disturbance of the roll angle can be seen in the following figure:
8.22.2 Results with LQ tracker

The steady state parameters:

![Steady state gains from reference to output](image1)

Figure 8.38: Steady state gains from references to outputs

![Sigma plot from state error to output](image2)

Figure 8.39: Steady state gains from state estimation error to output

Simulation results:

![Lower and upper tracking bounds from nominal to 15% UC](image3)

Figure 8.40: Tracking of $\phi$ reference with different uncertainty levels

![Lower and upper tracking bounds from nominal to 15% UC](image4)

Figure 8.41: Tracking of $\dot{r}$ reference with different uncertainty levels
Figure 8.42: Tracking of $\phi$ reference with different uncertainty levels (enlarged section)

Figure 8.43: Tracking of $\tau$ reference with different uncertainty levels (enlarged section)

Figure 8.44: 2-norms of tracking errors and roll rate with different uncertainty levels

Figure 8.45: 2-norms of control inputs with different uncertainty levels

Figure 8.46: Settling times with different uncertainty levels
8.22.3 Results with minimax tracker

The steady state parameters:

Figure 8.47: Good tracking results for the three different identified models (LQ case) enlarged transient
Simulation results:

Figure 8.52: Tracking of $\phi$ reference with different uncertainty levels

Figure 8.53: Tracking of $r_F$ reference with different uncertainty levels

Figure 8.54: Tracking of $\phi$ reference with different uncertainty levels (enlarged section)

Figure 8.55: Tracking of $r_F$ reference with different uncertainty levels (enlarged section)

Figure 8.56: Quality of disturbance estimation with different uncertainty levels

Figure 8.57: 2-norms of disturbance estimation errors with different uncertainty levels
Figure 8.58: 2-norms of tracking errors and roll rate with different uncertainty levels

Figure 8.59: 2-norms of control inputs with different uncertainty levels

Figure 8.60: Settling times with different uncertainty levels

Figure 8.61: Good tracking results for the three different identified models (minimax case) enlarged transient
8.23 Enlarged figures about real flight test results

Figure 8.62: Tracking of the φ doublet reference signal on 10th December 2009. PID control

Figure 8.63: Tracking of the φ doublet reference signal on 17th June 2010. PID control

Figure 8.64: Yaw damping on 10th December 2009. PID control

Figure 8.65: Yaw damping on 17th June 2010. PID control

Figure 8.66: Tracking of the φ doublet reference signal on 13th September 2011. LQ tracker control

Figure 8.67: Tracking of the φ doublet reference signal on 29th September 2011. LQ tracker control
Figure 8.68: Yaw damping on 13th September 2011. LQ tracker control

Figure 8.69: Yaw damping on 29th September 2011. LQ tracker control

Figure 8.70: Tracking of the $\phi$ doublet reference signal on 13th September 2011. MM tracker control

Figure 8.71: Tracking of the $\phi$ doublet reference signal on 19th July 2012. MM tracker control

Figure 8.72: Yaw damping on 13th September 2011. MM tracker control

Figure 8.73: Yaw damping on 19th July 2012. MM tracker control
Publications related directly to the thesis


Further publications


Bibliography


