UNIVERSIDAD DE GUANAJUATO

CAMPUS IRAPUATO SALAMANCA

DIVISIÓN DE INGENIERÍAS

DEVELOPMENT AND CRITICAL EVALUATION OF ROBUST H_∞ FILTERS

TESIS

QUE PARA OBTENER EL GRADO:

MAESTRO EN INGENIERÍA ELÉCTRICA

PRESENTA:

ING. JUAN JOSÉ LÓPEZ SOLÓRZANO

DIRECTOR DE TESIS:

DR. YURIY S. SHMALIY

SALAMANCA, GUANAJUATO. ENERO 2023

Abstract

This work presents a discrete convolution-based H_∞ finite impulse response (H_∞ FIR) filter. The *a posteriori* H_∞ FIR filter is developed by minimizing the squared H_∞ norm of the weighted disturbance-to-error transfer function, where the weights are related to errors. Since minimization isn't reachable without numerical approximations, Linear Matrix Inequality (LMI) based algorithms are derived to compute the filter gain in batch form, also recursive (Kalman-like) forms of the filter are provided in this work. It is shown numerically and experimentally that for disturbed systems operating under measurement and initial errors, the developed H_∞ FIR filter surpasses the Kalman filter in accuracy and has almost the same robustness as an unbiased FIR filter.

Acknowledgments.

I want to thank...

To Guanajuato's University, in specific to the Electronics Engineering Division for let me complete my studies providing virtual classes during the pandemic, and giving me a space to work when we return to the classrooms.

To the Mexican Consejo Nacional de Ciencia y Tecnología (CONACyT), for the support with a scholarship.

To all the teachers that I met through the classes, for the knowledge that were transmitted to me along the master's degree.

To my advisor Dr. Yuriy S. Shmaliy, for giving me the opportunity to work with him, his advices and the time dedicated to solve all my doubts.

To all my classmates and co-workers of the postgraduate lab, for his commentaries to my work.

To my friends and colleagues from the division, Perla, Oscar, Christian, Enrique, Daniel, Armando, Gabriel and Andrés for the good times with them both in the classrooms and on the university facilities.

To my friends from the Autonomous University of the State of Morelos, Alan, Eduardo, José, Alejandro, Bryan and Lorenzo for keep following my work and supporting me during my master's degree, I wish you the best for your future.

To Dr. J. Guadalupe from the Autonomous University of the State of Morelos, because he always be available to solve any doubt that I have, for his time and his advices.

To Kenia, for being a source of inspiration throughout my master's degree, I wish you a lot of happiness.

To my friends Christian and Mariano, for their unconditional friendship and support.

To my best friend David, for our friendship since we have 6 years old and despite of the distance I know that I can count with him always.

To Mafe for her friendship, her advices and for the support during my studies.

To my aunts, Cele, Mire, Chin and Laura for their support in my move and their emotional support.

To my cousin Cynthia, for her support with the move and her good wishes.

To my cousins Leo and Toño, for the laughs and good times with them and the support to achieve my goals.

To my cousin Jillos, for his talks and advice.

To my aunt Letty, for receiving me as one of her children, her talks, her food and her unconditional support at all times.

To my godfather Miguel, for his support during my stay in Guanajuato.

To my godmother Marisol and my uncle Armando, for receiving me several times in their house and the support they gave me at all times.

To my uncle Alejandro and his children Rocío, Daniel, Alejandro, Fernando and Gerardo, for having received me several times at their home and always offering me their support.

To my cousin Luis Martín, for being like a brother to me, his support, his talks and advice, for all those moments of fun, laughter and good times.

To my cousin Ángel, for spending time with me, her talks and jokes.

To my cousin Raúl, for sharing his room with me during my stay in Guanajuato.

To my cousins Claudia and Valeria, for their talks, advice and unconditional support at all times.

To my cousins Armando, Diana and Rebeca, for the time shared with them and the good times.

Dedication.

I want to dedicate this work...

To GOD because he is everything and thanks to him I have what I have and I am what I am.

To my parents Lorena and Juan José, who with their love, affection, teaching and example, I have achieved everything I have set out to do. I love you, thank you.

To my grandparents Raquel and Gerónimo, I have always felt their support and affection, I love them very much and I send them a hug to heaven.

To my grandparents Ma. Soledad and José Luz, who are a great example of love, and who opened the doors of their house to me.

Contents

List of Figures

List of Tables

Publications.

[1] J. J. Lopez-Solorzano and Y. S. Shmaliy, "H_∞ FIR Filter Gain Computation for Disturbed Systems using Linear Matrix Inequality," in *2022 19th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, IEEE, Nov. 2022.

[2] J. J. Lopez-Solorzano and Y. S. Shmaliy, "Receding Horizon H_2-FIR Filtering for Uncertain Disturbed Systems With Initial and Data Errors," *IEEE Trans. Automation Sci. Eng.*, pp. 1–8, To Be Published.

[3] J. J. López-Solórzano, O. Ibarra-Manzano, and Y. S. Shmaliy, "Iterative Robust H_∞-FIR Filtering Algorithm for Disturbed Systems under Harsh Conditions," *IEEE Trans. Ind. Informat.*, Under Review.

[4] J. J. López-Solórzano and Y. S. Shmaliy, "H_∞-FIR Filtering of Disturbed Systems using LMI under Measurement and Initial Errors," *Int. J. of Adaptive Control and Signal Process.*, After Major Revision.

[5] J. J. López-Solórzano and Y. S. Shmaliy, "Suboptimal *a posteriori* H_2-FIR Filter Using LMI with Applications to Harmonic Models," in *2022 IEEE Int. Summer Power Meeting/Int. Meeting on Commun. and Computing (RVP-AI/ROC&C)*.

[6] J. J. López-Solórzano and Y. S. Shmaliy, "Uncertain disturbed systems: Robust H_2-OUFIR filtering under measurement errors," *Int. J. of Adaptive Control and Signal Process.*, vol. 36, no. 10, pp. 2383–2400, Oct. 2022.

Acronyms

BE backward Euler. 5, 6, 41

BRL Bounded Real Lemma. 3, 10, 41

CMN colored measurement noise. 46

CPN colored process noise. 32, 46

DLE discrete Lyapunov equation. 44, 47, 53, 54

FE forward Euler. 5

FH Finite Horizon. 9, 11

FIR finite impulse response. 2, 4, 7, 9, 12, 22, 23

GNPG Generalized Noise Power Gain. 4, 23, 24, 50–52, 54

IIR infinite impulse response. 2

KF Kalman filter. 3, 10, 26, 34, 35, 37, 40, 48, 55, 56, 62, 64, 65

LMI linear matrix inequality. 2, 3, 11, 12, 16, 17, 22–24, 43, 44, 50, 53, 54, 64, 65

LTI linear time-invariant. 1, 6, 7, 41

LTV linear time-variant. 1

ML-FIR maximum likelihood FIR. 34, 35, 64

MPC model predictive control. 1, 2

MSE mean square error. 38, 39

NPG Noise Power Gain. 22, 23

OFIR Optimal FIR. 10, 26, 34, 35, 64

OUFIR Optimal unbiased FIR. 34, 35, 64

RH Receding Horizon. 2

RMSE root mean square error. 20, 26, 28, 30, 35, 56, 57, 59, 60, 62

SFC state feedback control. 1, 2

UFIR unbiased FIR. 2, 3, 17, 20, 26, 28, 30, 32–35, 44, 48, 50, 53, 55–57, 59, 60, 62, 64, 65

WGN white Gaussian noise. 2, 4, 22, 37

Chapter 1

Introduction

1.1 Background

The practical applications of digital systems in adverse conditions, such as industrial ones, require robustness from state space estimators [8]. An estimator is considered robust if it is insensitive to disturbing factors, including its tuning factors [37]. It is known that the highest robustness is achieved if the estimation errors are minimized for the maximized perturbation and tuning factors [46]. This gives the best practical effect, although at the expense of the accuracy obtained in contrast with optimum tuning [14]. Responding to practical needs, different types of robust state space estimators have been developed during the last decades [10], [13], [18], [20] for adaptive systems, state feedback control (SFC) and model predictive control (MPC).

The most effective robust estimators were obtained in the transform domain by minimizing the estimation errors for maximized disturbances using the disturbance-to-error transfer function $\mathcal{T}$. Therefore, such observers serve linear time-invariant (LTI) systems. H_2 filtering [21], [22] minimizes the Frobenius norm of $\mathcal{T}$ in a similar way to optimal filtering. H_∞ filtering [9], [30] minimizes the induced norm of $\mathcal{T}$ in the worst error case and results in robust energy-to-energy or ℓ_2-to-ℓ_2 structures. The generalized H_2 filtering [40], [46] minimizes the energy-to-peak $\mathcal{T}$ in estimators that have the structure ℓ_2-to-ℓ_∞. ℓ_1 filtering [5], [43] minimizes the peak-to-peak $\mathcal{T}$ in ℓ_∞-to-ℓ_∞ estimators. Setting aside the transfer function approach, the game theory H_∞ approach [26], [39] provides estimators that minimize the ratio of the estimation error norm and the disturbance and errors norms over a finite horizon of data points and can be applied to linear time-variant (LTV) systems.

Robust estimators for uncertain systems were also developed, mostly using the approach proposed in [11], [13], [24], [25], but these solutions are outside of the scope of this work.

1.2 Motivation

Looking at the robust algorithms mentioned above, it can be noticed that they all have Kalman-like recursive forms. Note that recursions are mainly available for white Gaussian noise (WGN) [37], except for Gauss-Markov noise. This means that robust recursive algorithms developed for arbitrary norm-bounded disturbances ignore the correlation that inevitably causes estimation errors. By contrast, batch estimators operate with full error covariances, do not ignore correlation at finite horizons, and therefore provide estimates that are superior to recursive ones. Batch estimators are commonly considered computationally complex, but this is no longer a problem for modern computers, unlike in Kalman's days. In support, it is worth noting other important advantages of batch estimators [37]:

1. Limited input limited output stability.
2. Rejection of errors beyond the averaging horizon.
3. Reduction of numerical errors by averaging.
4. Increased robustness.

In general, batch estimators with finite impulse response (FIR) give more precision than Kalman-like recursions that have infinite impulse response (IIR). Among the available FIR estimators, the unbiased FIR (UFIR) filter developed in [35], [38] is considered the most robust, since this filter has the optimal horizon length as its only tuning factor.

Note that convolution-based filtering was originally discussed in [17], [46] and the Receding Horizon (RH) FIR approach was developed for MPC and SFC in [18]. Thereafter, different robust RH FIR filters were designed in [1]–[3], [19], [23] in batch form and using linear matrix inequality (LMI), and some other preliminary results can be found in [7], [16], [41], [44], [45]. An important limitation of the above solutions is that measurement errors and initial errors are ignored.

Although several other advanced H_2-FIR robust estimators for complex environments have been recently developed in [27]–[29], [34], H_∞ FIR filters have not yet been developed operating with initial and data errors, which motivates the present work.

1.3 Objectives

The main objective of the work is to develop and critically evaluate the performance of robust H_∞ filters, both batch and recursive forms, based on a state estimators and compare its performance with other state estimation based filters.

1.3.1 Specific Objectives

1. Solve the H_∞ FIR filtering minimization problem using LMI as numerical method to obtain the gain of the filter.
2. Propose an iterative algorithm for computation of the H_∞ FIR filter gain.
3. Test the tuning, accuracy and robustness of the H_∞ FIR filter against other FIR filters.
4. Develop a recursive (Kalman-like) algorithm for the H_∞ filter estimation computation.
5. Propose an iterative algorithm for computation of the recursive H_∞ filter bias correction gain.
6. Test the tuning, accuracy and robustness of the recursive H_∞ filter against other recursive filters.

1.4 Scope

In this work, the theory of robust *a posteriori* H_∞ filtering with initial and measurement errors is developed. Since the Bounded Real Lemma (BRL) only applies to constant matrices, the BRL is modified for the covariance of the squared error by introducing an additional variable. Furthermore, basic and iterative algorithms are provided to compute the gain of the H_∞ FIR filter using LMI. Based on numerical simulations and experimental data, it is shown that the H_∞ FIR filter outperforms the Kalman filter (KF) in precision and demonstrates almost the same robustness as the UFIR filter. Also, a recursive form for the H_∞ FIR filter is developed with basic and iterative algorithms for the computation of the bias correction gain of the filter. The main contributions are the following:

- Framework for robust *a posteriori* H_∞ FIR filtering with initial and data errors using LMI.
- LMI-based algorithm to compute the batch H_∞ FIR filter gain.

- Iterative LMI-based algorithm to compute the batch H_∞ FIR filter gain.
- Numerical and experimental evidence of the better performance of the H_∞ FIR filter compared to other state estimation FIR filters.
- Framework for recursive robust *a posteriori* H_∞ filtering with initial and data errors using LMI.
- LMI-based algorithm to compute the recursive H_∞ filter bias correction gain.
- Iterative LMI-based algorithm to compute the recursive H_∞ filter bias correction gain.
- Numerical and experimental evidence of the better performance of the recursive H_∞ filter compared to other state estimation recursive filters.

The rest of the work is organized as follows. Chapter 2 discusses the theoretical basis of state estimation, the FIR filtering with state space models, formulates the H_∞ FIR filtering problem, derives the *a posteriori* H_∞ FIR filter with measurement and data errors, presents the basic algorithm for computing the H_∞ FIR filter gain and presents a numerical example for tuning the filter under WGN.

Chapter 3 presents an iterative algorithm for computation of the $H_\infty FIR$ filter gain where the minimization is reached by taking in count the properties of the Generalized Noise Power Gain (GNPG), presents a numerical example where the filter is tuned with both algorithms under WGN to compare its accuracy, presents the robustness test to the filter and shows the performance of the H_∞ FIR filter against real data.

Chapter 4 derives the recursive *a posteriori* H_∞ filter with measurement and data errors, presents the basic algorithm for computing the recursive H_∞ filter bias correction gain and a Kalman-like algorithm for computation of the recursive H_∞ estimations and presents a numerical example for tuning the filter under colored Gauss-Markov disturbance and measurement noise.

Chapter 5 presents an iterative algorithm for computation of the recursive H_∞ filter bias correction gain where the minimization is reached by taking in count the properties of the GNPG, presents a numerical example where the filter is tuned with both algorithms under colored Gauss-Markov disturbance and measurement noise to compare its accuracy, presents the robustness test to the filter and shows the performance of the recursive H_∞ filter against real data. Finally the conclusions are indicated.

Chapter 2

The *a posteriori* H_∞ FIR Filter

2.1 State Estimation

The term "state estimation" suggests the desire to estimate the state of some process, system, or object using its measurements [37]. Since measurements are commonly conducted in the presence of noise, its desirable to have an accurate and precise estimator, preferably optimal and unbiased.

When some stochastic dynamic system (or process) appears and it is desired to predict its further behavior, its necessary to know system characteristics at each time instant. State variables describe mathematically the state of a system so it is obvious that a set of state variables should be sufficient to predict the future system behavior.

Any stochastic dynamic system can be represented with a linear or nonlinear first-order vector differential equation (in continuous time) of difference equation (in discrete time) with respect to a set of its states. Such equations are called the state equations, where the state variables are typically affected by internal noise and external disturbances, and the model can be uncertain due to mismodeling.

To estimate the state of a system with random components represented with the state equation means to evaluate the state approximately using measurements over a finite time interval or all data available. Systems and processes can be either nonlinear or linear. Accordingly, there are nonlinear and linear state space models. Linear models are represented with linear equations and Gaussian noise. A model is said to be nonlinear if it is represented with nonlinear equations or with linear equations having non-Gaussian random components.

In discrete time t_k, a system can be represented in state space with a time step $\tau = t_k - t_{k-1}$ using either the forward Euler (FE) method or backward Euler (BE) method. By the FE method, the linear discrete-time state equation appears to be predictive, and the state-space model becomes

$$x_{k+1} = F_k x_k + E_k u_k + B_k w_k, \tag{2.1}$$

$$y_k = H_k x_k + D_k w_k + v_k, \tag{2.2}$$

where $x_k \in \mathbb{R}^K$ is the state vector, $u_k \in \mathbb{R}^L$ is the input (control) vector, $y_k \in \mathbb{R}^P$ is the observation vector, $w_k \in \mathbb{R}^M$ is the system error or disturbance and $v_k \in \mathbb{R}^P$ is the observation error. The time-varing matrices of the system are given by $F_k \in \mathbb{R}^{K\times K}$ who is the process matrix, $E_k \in \mathbb{R}^{K\times L}$, $B_k \in \mathbb{R}^{K\times M}$, $H_k \in \mathbb{R}^{P\times K}$ who is the observation matrix and $D_k \in \mathbb{R}^{P\times M}$. The term with u_k is often omitted in observation equation assuming that the order of u_k is smaller than of y_k. In most of the cases, the process noise is supposed to be zero mean and white Gaussian $w_k \sim N(0, Q_k)$ with known covariance $Q_k = \mathcal{E}\left\{w_k w_k^T\right\}$. The observation noise $v_k \sim N(0, R_k)$ is also often modeled as zero mean and white Gaussian with the covariance $R_k = \mathcal{E}\left\{v_k v_k^T\right\}$ ($\mathcal{E}\left\{\cdot\right\}$ means averaging). Furthermore, many problems suggest that w_k and v_k can be considered as uncorrelated and independent processes. By the BE method, the relevant state-space model attains the form

$$x_k = F_k x_{k-1} + E_k u_k + B_k w_k, \tag{2.3}$$

$$y_k = H_k x_k + D_k w_k + v_k, \tag{2.4}$$

For LTI systems all matrices in (2.1)-(2.4) become constants as F, E, B, H, D.

2.2 Methods of Linear State Estimation

State estimation in discrete-time state-space can be conducted employing methods of optimal linear filtering based on the state-space equations in (2.3)-(2.4). Irrespective of the estimator structure, the notation $\hat{x}_{k|\tau}$ means an estimate of state x_k at time index k given observations of x_k up to and including at time index τ. A state x_k to be estimated at time index k is represented by the following standard variables:

- $\hat{x}_k^- \triangleq \hat{x}_{k|k-1}$ is the *a priori* state estimate at k given observations up to and including at time index $k-1$.

- $\hat{x}_k \triangleq \hat{x}_{k|k}$ is the *a posteriori* state estimate at k given observations up to and including at k.

- The *a priori* estimation error is defined by
  $\varepsilon_k^- \triangleq \varepsilon_{k|k-1} = x_k - \hat{x}_k^-$.
- The *a posteriori* estimation error is defined by
  $\varepsilon_k \triangleq \varepsilon_{k|k} = x_k - \hat{x}_k$.
- The *a priori* error covariance is defined by
  $P_k^- \triangleq P_{k|k-1} = \mathcal{E}\left\{\varepsilon_k^- \varepsilon_k^{-T}\right\}$.
- The *a posteriori* error covariance is defined by
  $P_k \triangleq P_{k|k} = \mathcal{E}\left\{\varepsilon_k \varepsilon_k^T\right\}$.

In what follows these definitions will be used, while deriving the H_∞ FIR filter.

2.3 Extended LTI Discrete-Time State-Space Model

Consider a LTI system represented in discrete-time state-space with the following state and observation equations, respectively,

$$x_k = F x_{k-1} + E u_k + B w_k, \tag{2.5}$$

$$y_k = H x_k + v_k, \tag{2.6}$$

where $x_k \in \mathbb{R}^K$ is the state vector, $u_k \in \mathbb{R}^L$ is the input vector, $y_k \in \mathbb{R}^P$ is the observation vector, $w_k \in \mathbb{R}^M$ is the process noise and $v_k \in \mathbb{R}^P$ is the observation noise. Assume that $F \in \mathbb{R}^{K\times K}$, $E \in \mathbb{R}^{K\times L}$, $B \in \mathbb{R}^{K\times M}$ and $H \in \mathbb{R}^{P\times K}$ are known matrices.

The model in (2.5)-(2.6) cannot be used directly in FIR filtering and requires an extension on the horizon $[m, k]$ of N points, from $m = k - N + 1$ to k. This can be done if (2.5) is rewritten using the backward-in-time solutions as

$$x_k = F x_{k-1} + E u_k + B w_k, \tag{2.7a}$$

$$x_{k-1} = F x_{k-2} + E u_{k-1} + B w_{k-1}, \tag{2.7b}$$

$$\vdots$$

$$x_{m+2} = F x_{m+1} + E u_{m+2} + B w_{m+2}, \tag{2.7c}$$

$$x_{m+1} = F x_m + E u_{m+1} + B w_{m+1}, \tag{2.7d}$$

$$x_m = x_m + E u_m + B w_m, \tag{2.7e}$$

where the initial state x_m is supposed to be known and hence $u_m = 0$ and $w_m = 0$ in (2.7e). Then substituting (2.7d) into (2.7c) to modify (2.7c) for the initial state x_m and doing so until (2.7b) and (2.7a) are also modified for x_m allow extending (2.5) on $[m, k]$. By introducing the extended vectors

$$X_{m,k} = \begin{pmatrix} x_m^T & x_{m+1}^T & \cdots & x_k^T \end{pmatrix}^T \in \mathbb{R}^{NK}, \tag{2.8}$$

$$U_{m,k} = \begin{pmatrix} u_m^T & u_{m+1}^T & \cdots & u_k^T \end{pmatrix}^T \in \mathbb{R}^{NL}, \tag{2.9}$$

$$W_{m,k} = \begin{pmatrix} w_m^T & w_{m+1}^T & \cdots & w_k^T \end{pmatrix}^T \in \mathbb{R}^{NM}, \tag{2.10}$$

and referring to (2.7), the extended state equation can be written as

$$X_{m,k} = F_N x_m + S_N U_{m,k} + D_N W_{m,k}, \tag{2.11}$$

where the extended matrices are

$$F_N = \begin{pmatrix} I & F^T & \cdots & (F^{N-2})^T & (F^{N-1})^T \end{pmatrix}^T \in \mathbb{R}^{NK \times K}, \tag{2.12}$$

$$S_N = \begin{pmatrix} E & 0 & \cdots & 0 & 0 \\ FE & E & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ F^{N-2}E & F^{N-3}E & \cdots & E & 0 \\ F^{N-1}E & F^{N-2}E & \cdots & FE & E \end{pmatrix} \in \mathbb{R}^{NK \times NL}, \tag{2.13}$$

$$D_N = \begin{pmatrix} B & 0 & \cdots & 0 & 0 \\ FB & B & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ F^{N-2}B & F^{N-3}B & \cdots & B & 0 \\ F^{N-1}B & F^{N-2}B & \cdots & FB & B \end{pmatrix} \in \mathbb{R}^{NK \times NM}. \tag{2.14}$$

Similarly, the observation equation in (2.6) can be written as

$$y_k = Hx_k + v_k, \tag{2.15a}$$

$$y_{k-1} = Hx_{k-1} + v_{k-1}, \tag{2.15b}$$

$$\vdots$$

$$y_m = Hx_m + v_m. \tag{2.15c}$$

By substituting x_k, x_{k-1}, …, x_m taken from (2.7) into (2.15) and assigning two vectors

$$Y_{m,k} = \begin{pmatrix} y_m^T & y_{m+1}^T & \cdots & y_k^T \end{pmatrix}^T \in \mathbb{R}^{NP}, \tag{2.16}$$

$$V_{m,k} = \begin{pmatrix} v_m^T & v_{m+1}^T & \cdots & v_k^T \end{pmatrix}^T \in \mathbb{R}^{NP}, \tag{2.17}$$

the extended observation equation can be obtained as

$$Y_{m,k} = H_N x_m + L_N U_{m,k} + G_N W_{m,k} + V_{m,k}, \tag{2.18}$$

in which the extended matrices are

$$H_N = \bar{H}_N F_N \in \mathbb{R}^{NP \times K}, \tag{2.19}$$

$$L_N = \bar{H}_N S_N \in \mathbb{R}^{NP \times NL}, \tag{2.20}$$

$$G_N = \bar{H}_N D_N \in \mathbb{R}^{NP \times NM}, \tag{2.21}$$

and matrix $\bar{H}_N$ is diagonal

$$\bar{H}_N = \begin{pmatrix} H & 0 & \cdots & 0 & 0 \\ 0 & H & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & H & 0 \\ 0 & 0 & \cdots & 0 & H \end{pmatrix} \in \mathbb{R}^{NP \times NK}. \tag{2.22}$$

The extended state-space equations described on (2.11) and (2.18) can be used to derive all kinds of linear convolution-based batch state estimators (filters, smoothers, and predictors) for given cost function, and the FIR filter will require a Finite Horizon (FH) $[m, k]$ of N points.

To design a FIR filter, the state x_k can be represented by the last row vector in (2.11) as

$$x_k = F^{N-1} x_m + \bar{S}_N U_{m,k} + \bar{D}_N W_{m,k}, \tag{2.23}$$

where the matrix $\bar{S}_N$ is the last row vector in S_N and so is $\bar{D}_N$ in D_N.

2.4 The H$_\infty$ FIR Filtering

Before discussing H_∞ FIR filtering, recall that the H_2 FIR filter minimizes the squared Frobenius norm of the weighted error-to-error transfer function averaged over all frequencies [29]. Thereby, it provides optimal H_2 performance, but does not guarantee that possible peaks in the transfer function $\mathcal{T}$ will also be suppressed by averaging. Moreover, if the H_2 filter is not properly tuned, the peak errors in its output may grow due to bias errors, as in the KF and Optimal FIR (OFIR) filter.

The H_∞ filtering approach was developed to minimize the H_∞ norm of the disturbance-to-error (ς-to-ε) transfer function $\left\|\mathcal{T}\right\|_\infty = \sup \sigma_{\max}[\mathcal{T}(z)]$, where $\sigma_{\max}[\mathcal{T}(z)]$ is the maximum singular value of $\mathcal{T}(z)$. A feature of the H_∞ norm is that it minimizes the highest peak value of $\mathcal{T}(z)$ in the Bode plot. In H_∞ filtering, the induced H_∞ norm

$$\left\|\mathcal{T}\right\|_\infty = \sup_{\varsigma\neq 0} \frac{\left\|\mathcal{T}\varsigma\right\|_2}{\left\|\varsigma\right\|_2} = \sup_{\varsigma\neq 0} \frac{\left\|\varepsilon\right\|_2}{\left\|\varsigma\right\|_2} \tag{2.24}$$

of the ς-to-ε transfer function $\mathcal{T}$ [14] is commonly minimized, where the squared norms of the disturbance $\left\|\varsigma\right\|_2^2 = \sum_{i=m}^{k} \varsigma_i^* \varsigma$ and the estimation error $\left\|\varepsilon\right\|_2^2 = \sum_{i=m}^{k} \varepsilon_i^* \varepsilon$ are equal to their energies on $[m, k]$. Therefore, the H_∞ approach applies in both the time domain and the transform domain. Since $\left\|\mathcal{T}\right\|_\infty^2$ represents the maximum energy gain from ς to ε, then it follows that the H_∞ norm reflects the worst estimator case and its minimization results in a robust estimator. Moreover, for stable systems the H_∞ norm coincides with the ℓ_2 induced norm of the disturbance-to-error operator [32]. Therefore, it is also referred to as $\left\|\mathcal{T}\right\|_\infty = \left\|\mathcal{T}\right\|_{2,2}$.

In the standard formulation of H_∞ filtering [14], the robust H_∞ FIR filtering problem can be formulated as follows. Find the fundamental gain $\mathcal{H}_N$ for the H_∞ FIR filter to minimize $\left\|\mathcal{T}\right\|_\infty$, given by (2.24) on the horizon $[m, k]$, by solving the following optimization problem,

$$\mathcal{H}_N = \inf_{\mathcal{H}_N} \sup_{\varsigma\neq 0} \frac{\sum_{i=m}^{k} \varepsilon_i^T P_\varepsilon \varepsilon_i}{\sum_{i=m}^{k} \varsigma_i^T P_\varsigma \varsigma_i}, \tag{2.25}$$

where P_ε and P_ς are some proper weights. Since closed-form solutions for (2.25) can be found only in some special cases, consider the following problem

$$\mathcal{H}_N \Leftarrow \sup_{\varsigma\neq 0} \frac{\sum_{i=m}^{k} \varepsilon_i^T P_\varepsilon \varepsilon_i}{\sum_{i=m}^{k} \varsigma_i^T P_\varsigma \varsigma_i} < \gamma^2, \tag{2.26}$$

which allows to define $\mathcal{H}_N$ numerically for a given small positive $\gamma > 0$ and develop suboptimal algorithms. Note that the factor γ^2, which indicates the fraction of the disturbance energy that goes into the estimator error, should preferably be small. But because γ^2 cannot be too small for stable estimators, its value should be constrained.

2.4.1 The *a posteriori* H_∞ FIR Filter

To derive the H_∞ FIR filter, the column matrix rule and the BRL are needed.

Lemma 1. ***(Column matrix rule).*** *Given a block column matrix* $Z_{m,k} = \begin{pmatrix} z_m^T & z_{m+1}^T & \cdots & z_k^T \end{pmatrix}^T$ *specified on* $[m,k]$*. Its recursive form is [19]*

$$Z_{m,k} = A_w Z_{m-1,k-1} + B_w z_k, \tag{2.27}$$

using the following strictly sparse matrices,

$$A_w = \begin{pmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, \qquad B_w = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ I \end{pmatrix}. \tag{2.28}$$

Proof. The proof is self-obvious. □ □ □.

Lemma 2. ***(Bounded real lemma (filtering)).*** *Given a state space model*

$$x_k = F x_{k-1} + B w_k, \tag{2.29}$$

$$y_k = H x_k + D w_k. \tag{2.30}$$

Let $\gamma > 0$ *and* $S = HB + D$*. If there exists a matrix* $X > 0$ *such that the following LMI is soluble,*

$$\begin{pmatrix} -X^{-1} & F & B & 0 \\ F^T & -X & 0 & F^T H^T \\ B^T & 0 & -\gamma P_w & S^T \\ 0 & HF & S & -\gamma P_y^{-1} \end{pmatrix} < 0, \tag{2.31}$$

then the following inequality holds on $[m,k]$*,*

$$\frac{\sum_{i=m}^{k} y_i^T P_y y_i}{\sum_{i=m}^{k} w_i^T P_w w_i} < \gamma^2. \tag{2.32}$$

Proof. Consider the dissipativity inequality [31] on a FH $[m,k]$*,*

$$V(x_k) - V(x_m) < \sum_{i=m}^{k} s(w_i, y_i), \tag{2.33}$$

where $V(x_k)$ *is the Lyapunov (storage) function representing the energy stored in the system at* k *and* $s(w_i, y_i)$ *is a supply function representing the energy that is supplied to the system* *at i. Choose the storage function $V(x_k) = x_k^T K x_k$ and, referring to (2.32), assign $s(w_i, y_i) = \gamma^2 w_i^T P_w w_i - y_i^T P_y y_i > 0$ to be the supply function. Then rewrite (2.33) as*

$$\sum_{i=m}^{k}(y_i^T P_y y_i - \gamma^2 w_i^T P_w w_i) + \sum_{i=m+1}^{k}(V(x_i) - V(x_{i-1})) < 0,$$

substitute y_i taken from (2.30) and x_i from (2.29), assign $S = HB + D$, go to

$$\sum_{i=m}^{k}[(HFx_{i-1} + Sw_i)^T P_y((HFx_{i-1} + Sw_i)) - \gamma^2 w_i^T P_w w_i] + \sum_{i=m+1}^{k}(x_i^T K x_i - x_{i-1}^T K x_{i-1}) < 0,$$

note that values beyond $[m, k]$, namely at $m-1$, are not available for FIR filtering, change the lower limit in the first sum to $m+1$, unite all components in one sum, and come up with

$$\sum_{i=m+1}^{k}[(HFx_{i-1} + Sw_i)^T P_y((HFx_{i-1} + Sw_i)) - \gamma^2 w_i^T P_w w_i + x_i^T K x_i - x_{i-1}^T K x_{i-1}] < 0. \tag{2.34}$$

To eliminate variables, rearrange the terms and rewrite (2.34) as

$$\sum_{m+1}^{k}\begin{pmatrix} x_{i-1} \\ w_i \end{pmatrix}^T \Theta \begin{pmatrix} x_{i-1} \\ w_i \end{pmatrix} < 0,$$

which is satisfied if the following LMI holds,

$$\Theta = \begin{pmatrix} F^T K F + F^T H^T P_y H F - K & F^T K B + F^T H^T P_y S \\ B^T K F + S^T P_y H F & B^T K B + S^T P_y S - \gamma^2 P_w \end{pmatrix} < 0. \tag{2.35}$$

Then decompose (2.35) as

$$\begin{pmatrix} -K & 0 \\ 0 & -\gamma^2 P_w \end{pmatrix} + \begin{pmatrix} F^T & F^T H^T \\ B^T & S^T \end{pmatrix}\begin{pmatrix} K & 0 \\ 0 & P_y \end{pmatrix}\begin{pmatrix} F & B \\ HF & S \end{pmatrix} < 0,$$

consider it as a Schur's complement [6], and represent with another inequality

$$\begin{pmatrix} -K & 0 & F^T & F^T H^T \\ 0 & -\gamma^2 P_w & B^T & S^T \\ F & B & -K^{-1} & 0 \\ HF & S & 0 & -P_y^{-1} \end{pmatrix} < 0. \tag{2.36}$$

Now multiply (2.36) *from the left-hand and right-hand sides with the following matrices, respectively,*

$$\begin{pmatrix} 0 & 0 & I & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \end{pmatrix}, \qquad \begin{pmatrix} 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \end{pmatrix}$$

and obtain

$$\begin{pmatrix} -K^{-1} & F & B & 0 \\ F^T & -K & 0 & F^T H^T \\ B^T & 0 & -\gamma^2 P_w & S^T \\ 0 & HF & S & -P_y^{-1} \end{pmatrix} < 0. \tag{2.37}$$

Finally multiply (2.37) *from both sides with*

$$\begin{pmatrix} \gamma^{0.5} I & 0 & 0 & 0 \\ 0 & \gamma^{-0.5} I & 0 & 0 \\ 0 & 0 & \gamma^{-0.5} I & 0 \\ 0 & 0 & 0 & \gamma^{0.5} I \end{pmatrix},$$

introduce a new variable $X = K/\gamma$, *arrive at* (2.31), *and complete the proof.* □ □ □.

To obtain the *a posteriori* H_∞ FIR filter using lemma 2, consider the model in (2.23) and (2.18) with $U_{m,k} = 0$ and define the FIR estimate as

$$\begin{aligned} \hat{x}_k &= \mathcal{H}_N Y_{m,k} \\ &= \mathcal{H}_N H_N x_m + \mathcal{H}_N G_N W_{m,k} + \mathcal{H}_N V_{m,k}. \end{aligned} \tag{2.38}$$

Now, using (2.23) with $U_{m,k} = 0$ and (2.38), the estimation error ε_k can be transformed to

$$\varepsilon_k = \mathcal{B}_N x_m + \mathcal{W}_N W_{m,k} - \mathcal{V}_N V_{m,k}, \tag{2.39}$$

where the error residual matrices are defined by

$$\mathcal{B}_N = F^{N-1} - \mathcal{H}_N H_N, \tag{2.40}$$

$$\mathcal{W}_N = \bar{D}_N - \mathcal{H}_N G_N, \tag{2.41}$$

$$\mathcal{V}_N = \mathcal{H}_N. \tag{2.42}$$

Using the following matrix forms given by lemma 1,

$$W_{m,k} = A_w W_{m-1,k-1} + B_w w_k, \tag{2.43}$$

$$V_{m,k} = A_w V_{m-1,k-1} + B_w v_k, \tag{2.44}$$

where matrices A_w and B_w are defined by (2.28).

Now introduce two augmented vectors $z_k = \begin{pmatrix} W_{m,k}^T & V_{m,k}^T & i_k^T \end{pmatrix}^T$, where $i_k = x_m$, and $\varsigma_k = \begin{pmatrix} w_k^T & v_k^T \end{pmatrix}^T$, and combine them in the following state-space model

$$z_k = \tilde{F}_\varsigma z_{k-1} + \tilde{B}_\varsigma \varsigma_k, \tag{2.45}$$

$$\varepsilon_k = \tilde{C}_\varsigma z_k \tag{2.46}$$

in which the newly introduced block matrices have the form

$$\tilde{F}_\varsigma = \begin{pmatrix} A_w & 0 & 0 \\ 0 & A_w & 0 \\ 0 & 0 & I \end{pmatrix}, \qquad \tilde{B}_\varsigma = \begin{pmatrix} B_w & 0 \\ 0 & B_w \\ 0 & 0 \end{pmatrix},$$
$$\tilde{C}_\varsigma = \begin{pmatrix} \mathcal{W}_N & -\mathcal{V}_N & \mathcal{B}_N \end{pmatrix}. \tag{2.47}$$

Note that the strictly sparse matrices $\tilde{F}_\varsigma$ and $\tilde{B}_\varsigma$ can significantly reduce computational complexity.

An important property of the model in (2.45) and (2.46) follows immediately: all error residual matrices are combined into a new observation matrix $\tilde{C}_\varsigma$, which is thus completely responsible for the H_∞ filter performance.

2.4.2 LMI Based Algorithm for $\mathbf{H_\infty}$ FIR Filter Gain Computation

The state-space equations in (2.45) and (2.46) can't be used directly on the inequality given by lemma 2 due to weight P_y in (2.31) corresponds to P_ε in the H_∞ FIR Filtering problem, and $P_\varepsilon = P_k = \mathcal{E}\{\varepsilon_k \varepsilon_k^T\}$ is given by

$$P_\varepsilon = \mathcal{B}_N \chi_m \mathcal{B}_N^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T, \tag{2.48}$$

where $\chi_m = \mathcal{E}\{x_m x_m^T\} = \hat{x}_m \hat{x}_m^T + P_m$, $\mathcal{Q}_N = \mathcal{E}\{W_{m,k} W_{m,k}^T\}$ and $\mathcal{R}_N = \mathcal{E}\{V_{m,k} V_{m,k}^T\}$, the residual matrices $\mathcal{B}_N$, $\mathcal{W}_N$, $\mathcal{V}_N$ are defined by (2.40)-(2.42). As can be seen in (2.48), P_ε is function of the gain $\mathcal{H}_N$, which is a desired variable for the minimization of γ in the inequality in (2.31), so, the inequality in (2.31) is nonlinear with respect to the gain $\mathcal{H}_N$ due to the inversion of P_ε.

To solve this problem, first rewrite P_ε in (2.48) as

$$P_\varepsilon = \mathcal{A} - \mathcal{C}\mathcal{H}_N^T - \mathcal{H}_N \mathcal{C}^T + \mathcal{H}_N \mathcal{D} \mathcal{H}_N^T, \tag{2.49}$$

where the following matrices are introduced: $\mathcal{A} = F^{N-1}\chi_m(F^{N-1})^T + \bar{D}_N\mathcal{Q}_N\bar{D}_N^T$, $\mathcal{C} = F^{N-1}\chi_m H_N^T + \bar{D}_N\mathcal{Q}_N G_N^T$, $\mathcal{D} = H_N\chi_m H_N^T + \Omega_N$, and $\Omega_N = G_N\mathcal{Q}_N G_N^T + \mathcal{R}_N$. Then decompose (2.49) as

$$P_\varepsilon = \begin{pmatrix} I & \mathcal{H}_N \end{pmatrix} \begin{pmatrix} \mathcal{A} & -\mathcal{C} \\ -\mathcal{C}^T & \mathcal{D} \end{pmatrix} \begin{pmatrix} I \\ \mathcal{H}_N^T \end{pmatrix},$$

introduce new auxiliary matrices

$$\tilde{H}_\varsigma = \begin{pmatrix} I & \mathcal{H}_N \end{pmatrix}, \tag{2.50}$$

$$P_\mathcal{J} = \begin{pmatrix} \mathcal{A} & -\mathcal{C} \\ -\mathcal{C}^T & \mathcal{D} \end{pmatrix}, \tag{2.51}$$

and rewrite P_ε as

$$P_\varepsilon = \tilde{H}_\varsigma P_\mathcal{J} \tilde{H}_\varsigma^T \tag{2.52}$$

Now replace F, B, H, P_w and P_y in (2.35) with $\tilde{F}_\varsigma$, $\tilde{B}_\varsigma$, $\tilde{C}_\varsigma$, P_ς and P_ε respectively,

$$\begin{pmatrix} \tilde{F}_\varsigma^T K\tilde{F}_\varsigma + \tilde{F}_\varsigma^T\tilde{C}_\varsigma^T P_\varepsilon\tilde{C}_\varsigma\tilde{F}_\varsigma - K & \tilde{F}_\varsigma^T K\tilde{B}_\varsigma + \tilde{F}_\varsigma^T\tilde{C}_\varsigma^T P_\varepsilon\tilde{C}_\varsigma\tilde{B}_\varsigma \\ \tilde{B}_\varsigma^T K\tilde{F}_\varsigma + \tilde{B}_\varsigma^T\tilde{C}_\varsigma^T P_\varepsilon\tilde{C}_\varsigma\tilde{F}_\varsigma & \tilde{B}_\varsigma^T K\tilde{B}_\varsigma + \tilde{B}_\varsigma^T\tilde{C}_\varsigma^T P_\varepsilon\tilde{C}_\varsigma\tilde{B}_\varsigma - \gamma^2 P_\varsigma \end{pmatrix} < 0, \tag{2.53}$$

where P_ς is given by

$$P_\varsigma = \mathcal{E}\{\varsigma_k\varsigma_k^T\} = \begin{pmatrix} Q_k & 0 \\ 0 & R_k \end{pmatrix}. \tag{2.54}$$

Replace (2.52) into (2.53) as

$$\begin{pmatrix} \tilde{F}_\varsigma^T K\tilde{F}_\varsigma + \tilde{F}_\varsigma^T\tilde{C}_\varsigma^T\tilde{H}_\varsigma P_\mathcal{J}\tilde{H}_\varsigma^T\tilde{C}_\varsigma\tilde{F}_\varsigma - K & \tilde{F}_\varsigma^T K\tilde{B}_\varsigma + \tilde{F}_\varsigma^T\tilde{C}_\varsigma^T\tilde{H}_\varsigma P_\mathcal{J}\tilde{H}_\varsigma^T\tilde{C}_\varsigma\tilde{B}_\varsigma \\ \tilde{B}_\varsigma^T K\tilde{F}_\varsigma + \tilde{B}_\varsigma^T\tilde{C}_\varsigma^T\tilde{H}_\varsigma P_\mathcal{J}\tilde{H}_\varsigma^T\tilde{C}_\varsigma\tilde{F}_\varsigma & \tilde{B}_\varsigma^T K\tilde{B}_\varsigma + \tilde{B}_\varsigma^T\tilde{C}_\varsigma^T\tilde{H}_\varsigma P_\mathcal{J}\tilde{H}_\varsigma^T\tilde{C}_\varsigma\tilde{B}_\varsigma - \gamma^2 P_\varsigma \end{pmatrix} < 0,$$

Then decompose it as

$$\begin{pmatrix} -K & 0 \\ 0 & -\gamma^2 P_\varsigma \end{pmatrix} + \begin{pmatrix} \tilde{F}_\varsigma^T & \tilde{F}_\varsigma^T\tilde{C}_\varsigma^T\tilde{H}_\varsigma \\ \tilde{B}_\varsigma^T & \tilde{B}_\varsigma^T\tilde{C}_\varsigma^T\tilde{H}_\varsigma \end{pmatrix} \begin{pmatrix} K & 0 \\ 0 & P_\mathcal{J} \end{pmatrix} \begin{pmatrix} \tilde{F}_\varsigma & \tilde{B}_\varsigma \\ \tilde{H}_\varsigma^T\tilde{C}_\varsigma\tilde{F}_\varsigma & \tilde{H}_\varsigma^T\tilde{C}_\varsigma\tilde{B}_\varsigma \end{pmatrix} < 0, \tag{2.55}$$

introduce a new matrix

$$\begin{aligned} \tilde{J}_\varsigma = \tilde{H}_\varsigma^T\tilde{C}_\varsigma &= \begin{pmatrix} \mathcal{W}_N & -\mathcal{V}_N & \mathcal{B}_N \\ \mathcal{H}_N^T\mathcal{W}_N & -\mathcal{H}_N^T\mathcal{V}_N & \mathcal{H}_N^T\mathcal{B}_N \end{pmatrix} \\ &= \begin{pmatrix} \bar{D}_N - \mathcal{H}_N G_N & -\mathcal{H}_N & F^{N-1} - \mathcal{H}_N H_N \\ \mathcal{H}_N^T\bar{D}_N - \mathcal{H}_N^T\mathcal{H}_N G_N & -\mathcal{H}_N^T\mathcal{H}_N & \mathcal{H}_N^T F^{N-1} - \mathcal{H}_N^T\mathcal{H}_N H_N \end{pmatrix}, \end{aligned} \tag{2.56}$$

and replace (2.56) into (2.55) as

$$\begin{pmatrix} -K & 0 \\ 0 & -\gamma^2 P_\varsigma \end{pmatrix} + \begin{pmatrix} \tilde{F}_\varsigma^T & \tilde{F}_\varsigma^T\tilde{J}_\varsigma^T \\ \tilde{B}_\varsigma^T & \tilde{B}_\varsigma^T\tilde{J}_\varsigma^T \end{pmatrix} \begin{pmatrix} K & 0 \\ 0 & P_\mathcal{J} \end{pmatrix} \begin{pmatrix} \tilde{F}_\varsigma & \tilde{B}_\varsigma \\ \tilde{J}_\varsigma\tilde{F}_\varsigma & \tilde{J}_\varsigma\tilde{B}_\varsigma \end{pmatrix} < 0,$$

consider it as a Schur's complement, and represent with another inequality

$$\begin{pmatrix} -K & 0 & \tilde{F}_\varsigma^T & \tilde{F}_\varsigma^T \tilde{J}_\varsigma^T \\ 0 & -\gamma^2 P_\varsigma & \tilde{B}_\varsigma^T & \tilde{B}_\varsigma^T \tilde{J}_\varsigma^T \\ \tilde{F}_\varsigma & \tilde{B}_\varsigma & -K^{-1} & 0 \\ \tilde{J}_\varsigma \tilde{F}_\varsigma & \tilde{J}_\varsigma \tilde{B}_\varsigma & 0 & -P_{\mathcal{J}}^{-1} \end{pmatrix} < 0.$$

Now multiply this inequality from the left-hand and right-hand sides with the following matrices, respectively,

$$\begin{pmatrix} 0 & 0 & I & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \end{pmatrix}, \qquad \begin{pmatrix} 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \end{pmatrix}$$

and obtain

$$\begin{pmatrix} -K^{-1} & \tilde{F}_\varsigma & \tilde{B}_\varsigma & 0 \\ \tilde{F}_\varsigma^T & -K & 0 & \tilde{F}_\varsigma^T \tilde{J}_\varsigma^T \\ \tilde{B}_\varsigma^T & 0 & -\gamma^2 P_\varsigma & \tilde{B}_\varsigma^T \tilde{J}_\varsigma^T \\ 0 & \tilde{J}_\varsigma \tilde{F}_\varsigma & \tilde{J}_\varsigma \tilde{B}_\varsigma & -P_{\mathcal{J}}^{-1} \end{pmatrix} < 0. \tag{2.57}$$

Inequality in (2.57) still nonlinear with respect to the gain $\mathcal{H}_N$ due to the quadratic terms $\mathcal{H}_N^T \mathcal{H}_N$ in $\tilde{J}_\varsigma$ defined by (2.56). To solve this problem, introduce an auxiliary matrix $\mathcal{Z}$ such that

$$\mathcal{Z} > \mathcal{H}_N^T \mathcal{H}_N,$$

and represent it with the inequality

$$\mathcal{Z} - \mathcal{H}_N^T \mathcal{H}_N > 0.$$

If the Schur complement is used, last inequality can be equivalently replaced with the LMI as:

$$\begin{pmatrix} \mathcal{Z} & \mathcal{H}_N^T \\ \mathcal{H}_N & I \end{pmatrix} > 0. \tag{2.58}$$

Now, let's redefine $\tilde{J}_\varsigma$ in (2.56) using the new variable $\mathcal{Z}$ to replace the quadratic terms,

$$\tilde{J}_\varsigma = \begin{pmatrix} \bar{D}_N - \mathcal{H}_N G_N & -\mathcal{H}_N & F^{N-1} - \mathcal{H}_N H_N \\ \mathcal{H}_N^T \bar{D}_N - \mathcal{Z} G_N & -\mathcal{Z} & \mathcal{H}_N^T F^{N-1} - \mathcal{Z} H_N \end{pmatrix} \tag{2.59}$$

Also, inequality in (2.57) is nonlinear with respect to the symmetric positive-definite matrix K. To avoid the inversion of K, pre- and post-multiply the matrix (2.57) with

$$\begin{pmatrix} K & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}$$

and obtain another form of the inequality in (2.57)

$$\begin{pmatrix} -K & K\tilde{F}_\varsigma & K\tilde{B}_\varsigma & 0 \\ \tilde{F}_\varsigma^T K & -K & 0 & \tilde{F}_\varsigma^T \tilde{J}_\varsigma^T \\ \tilde{B}_\varsigma^T K & 0 & -\gamma^2 P_\varsigma & \tilde{B}_\varsigma^T \tilde{J}_\varsigma^T \\ 0 & \tilde{J}_\varsigma \tilde{F}_\varsigma & \tilde{J}_\varsigma \tilde{B}_\varsigma & -P_{\mathcal{J}}^{-1} \end{pmatrix} < 0. \tag{2.60}$$

Inequality in (2.60) has the form of a LMI if $\tilde{J}_\varsigma$ defined by (2.59) is used.

The gain $\mathcal{H}_N$ of the *a posteriori* H_∞ FIR filter now can be determined by solving the following minimization problem,

$$\begin{gathered} \mathcal{H}_N = \min_{\mathcal{H}_N, \mathcal{Z}, K, \gamma^2} \gamma^2 \\ \text{subject to (2.58), (2.60) and } \mathcal{Z} = \mathcal{H}_N^T \mathcal{H}_N. \end{gathered} \tag{2.61}$$

The third constraint of the minimization problem in (2.61), can be achieved by using Algorithm 1 in which gamma is minimized in each iteration by increasing the trace of $\mathcal{Z}$, at the end of each iteration the trace of $\mathcal{Z}$ is compared with the trace of $\mathcal{H}_N^T \mathcal{H}_N$, if the difference between them is greater than a small threshold $\delta_0 > 0$, the routine is ended and the gain is obtained. The best candidate for initializing the minimization procedure is of course the UFIR filter gain $\hat{\mathcal{H}}_N = F^{N-1}(H_N^T H_N)^{-1} H_N^T$.

Using the gain $\mathcal{H}_N$, numerically determined by using Algorithm 1, the *a posteriori* H_∞ filtering estimate and error covariance for uncorrelated w_k, v_k, and x_m can be obtained as, respectively,

$$\hat{x}_k = \mathcal{H}_N Y_{m,k}, \tag{2.62}$$

$$P_k = \mathcal{B}_N \chi_m \mathcal{B}_N^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T, \tag{2.63}$$

where the error residual matrices are given by (2.40)-(2.42).

Algorithm 1: Algorithm for H_∞ FIR Filter Gain Computation

Data: δ_0, $\hat{\mathcal{H}}_N$, P_ς, $\mathcal{Q}_{m,k}$, $\mathcal{R}_{m,k}$

Result: $\mathcal{H}_N$

begin

  $\mathcal{H}_N = \hat{\mathcal{H}}_N$;   `/* Initialize with UFIR gain */`

  $\mathcal{Z} = \hat{\mathcal{H}}_N^T \hat{\mathcal{H}}_N$;

  while $|\mathrm{tr}(\mathcal{Z}) - \mathrm{tr}(\mathcal{H}_N^T \mathcal{H}_N)| < \delta_0$ **do**

    $\mathcal{Z}_{\text{prev}} = \mathcal{Z}$;

    $\mathcal{H}_N = \min \gamma^2$ subject to (2.58), (2.60) and $\mathrm{tr}(\mathcal{Z}) > \mathrm{tr}(\mathcal{Z}_{\text{prev}})$;

  end

end

2.5 Filter Tuning Example

Consider a radar and it is desired to measure a distance d_k in meters to a car that moves in discrete time index k with constant velocity v_k in meters by seconds. The process equations can be written as

$$d_k = d_{k-1} + \tau \mathrm{v}_k + w_{1k},$$

$$\mathrm{v}_k = \mathrm{v}_{k-1} + w_{2k},$$

where w_{1k} is a random error in the distance, w_{2k} is the error in the velocity, and $\tau = t_k - t_{k-1}$. Now assign two states. The first state is the distance $x_{1k} = d_k$, the second state is the velocity $x_{2k} = \mathrm{v}_k$. This gives the state equations:

$$x_{1k} = x_{1(k-1)} + \tau x_{2k} + w_{1k},$$

$$x_{2k} = x_{2(k-1)} + w_{2k}.$$

Next, assume that noise $w_k \sim N(0, \sigma_w^2)$ only affects the velocity and assign

$$x_k = \begin{pmatrix} x_{1k} \\ x_{2k} \end{pmatrix}, \quad F = \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} \tau \\ 1 \end{pmatrix}, \quad w_{2k} = w_k, \quad w_{1k} = 0. \tag{2.64}$$

Then write the state equation

$$x_k = F x_{k-1} + B w_k. \tag{2.65}$$

For measured distance (first state), write the observation equation

$$y_k = Hx_k + v_k, \tag{2.66}$$

where $H = \begin{pmatrix} 1 & 0 \end{pmatrix}$ and $v_k \sim N(0, \sigma_v^2)$ is the measurement noise.

Now extend the state space (2.65)-(2.66) on $[m, k]$, then the *a posteriori* H_∞-FIR filter gain can now be determined numerically by using Algorithm 1 and the estimation computed as $\hat{x}_k = \mathcal{H}_N Y_{m,k}$. An example of the distance and the velocity generated with the standard deviations $\sigma_w = 12$ m/s and $\sigma_v = 10$ m and with a sample period of $\tau = 0.025$ s is shown in Fig. 2.1.

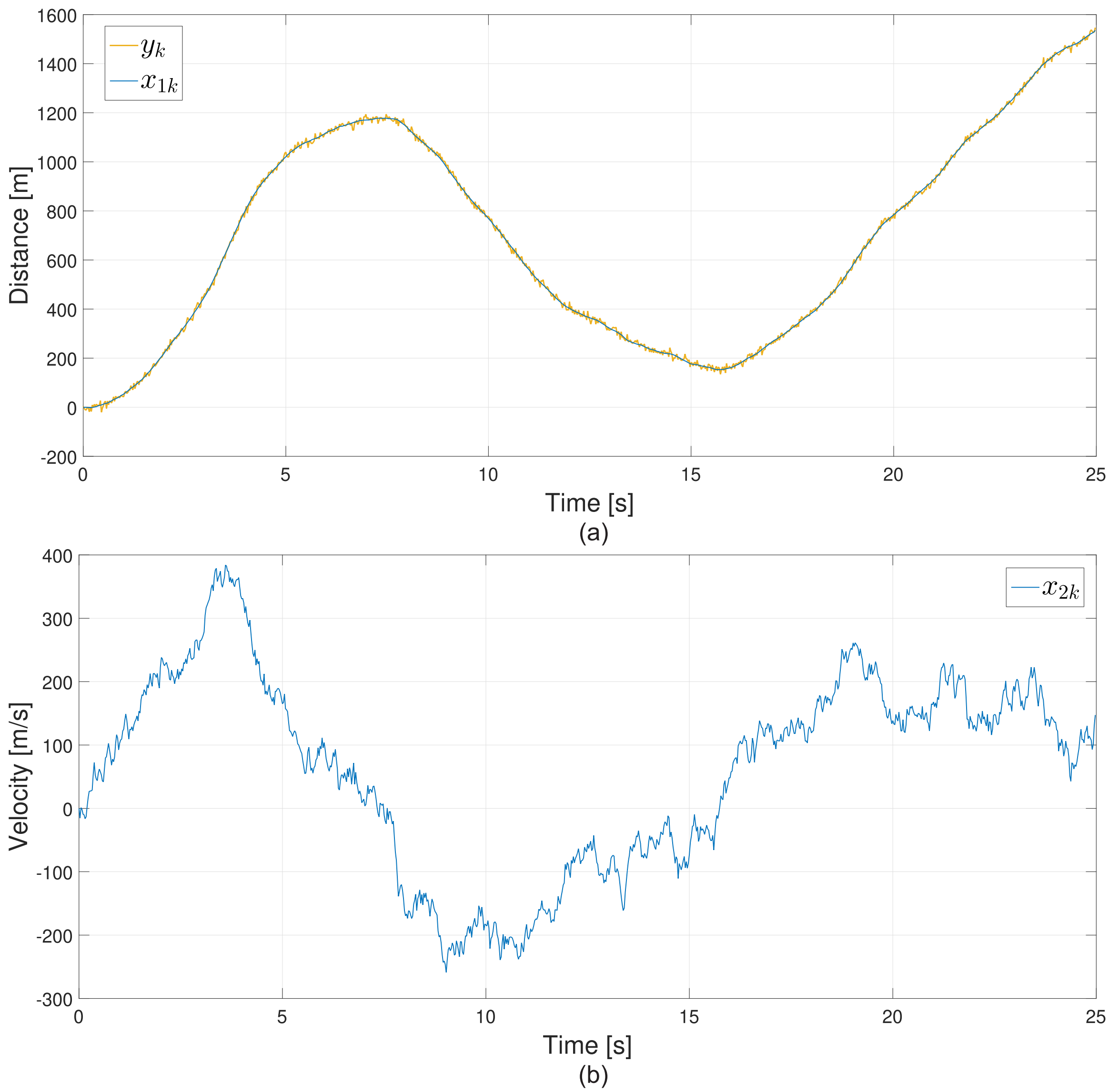


Figure 2.1: Process of the radar system generated, (a) first state measurements and first state without noise, (b) second state.

The H_∞ FIR filter gain was computed using a small threshold $\delta_0 = 0.01$, and also an estimation using the UFIR filter ($N_{\text{opt}} = 20$) was obtained as benchmark. The behavioral of Algorithm 1 could be seen graphically on Fig. 2.2. In Fig. 2.2(a) is shown the minimization of γ as the trace of $\mathcal{Z}$ is increased, as can be seen, if the trace of $\mathcal{Z}$ is increased, γ takes lower values as it reach a minimum, but Fig. 2.2(b) shows that this minimum value of gamma is not necessary the minimum that we are searching for. Fig. 2.2(b) shows the comparison of the trace of $\mathcal{Z}$ and the trace of $\mathcal{H}_N^T\mathcal{H}_N$ as the trace of $\mathcal{Z}$ is increased, in this case both graphs are very similar with lower values of the trace of $\mathcal{Z}$ but there is a point where this trace is increased and the other one diverges, this is because at this point of the algorithm $\mathcal{Z} \neq \mathcal{H}_N^T\mathcal{H}_N$ and the third restriction in (2.61) isn't satisfied, so the algorithm must be finished before both traces starts to diverge. Once the algorithm is stopped, the H_∞ FIR filter gain is obtained as the last value of $\mathcal{H}_N$ before the algorithm is stopped.

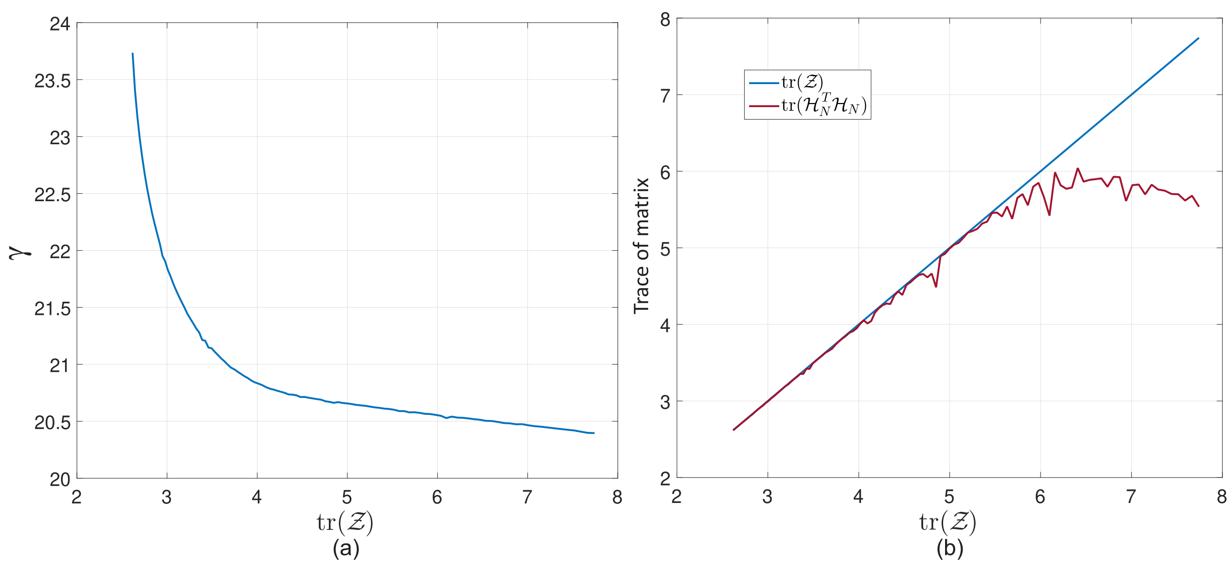


Figure 2.2: Solving the minimization problem using Algorithm 1: (a) minimizing γ as the trace of $\mathcal{Z}$ is increased and (b) comparison between the trace of matrix $\mathcal{H}_N^T\mathcal{H}_N$ and the trace of $\mathcal{Z}$ as the trace of $\mathcal{Z}$ is increased.

Typical filtering errors are shown in Fig. 2.3, it can be inferred that UFIR filter is the one who gives the less accurate estimates, while the estimation using the gain computed with Algorithm 1 looks like giving better estimates. The root mean square error (RMSE) for each filter is given in Table 2.1, and, as can be graphically seen on Fig. 2.3, the UFIR filter is the less accurate. The H_∞ FIR filter using Algorithm 1 to compute the gain has lower errors than the UFIR.

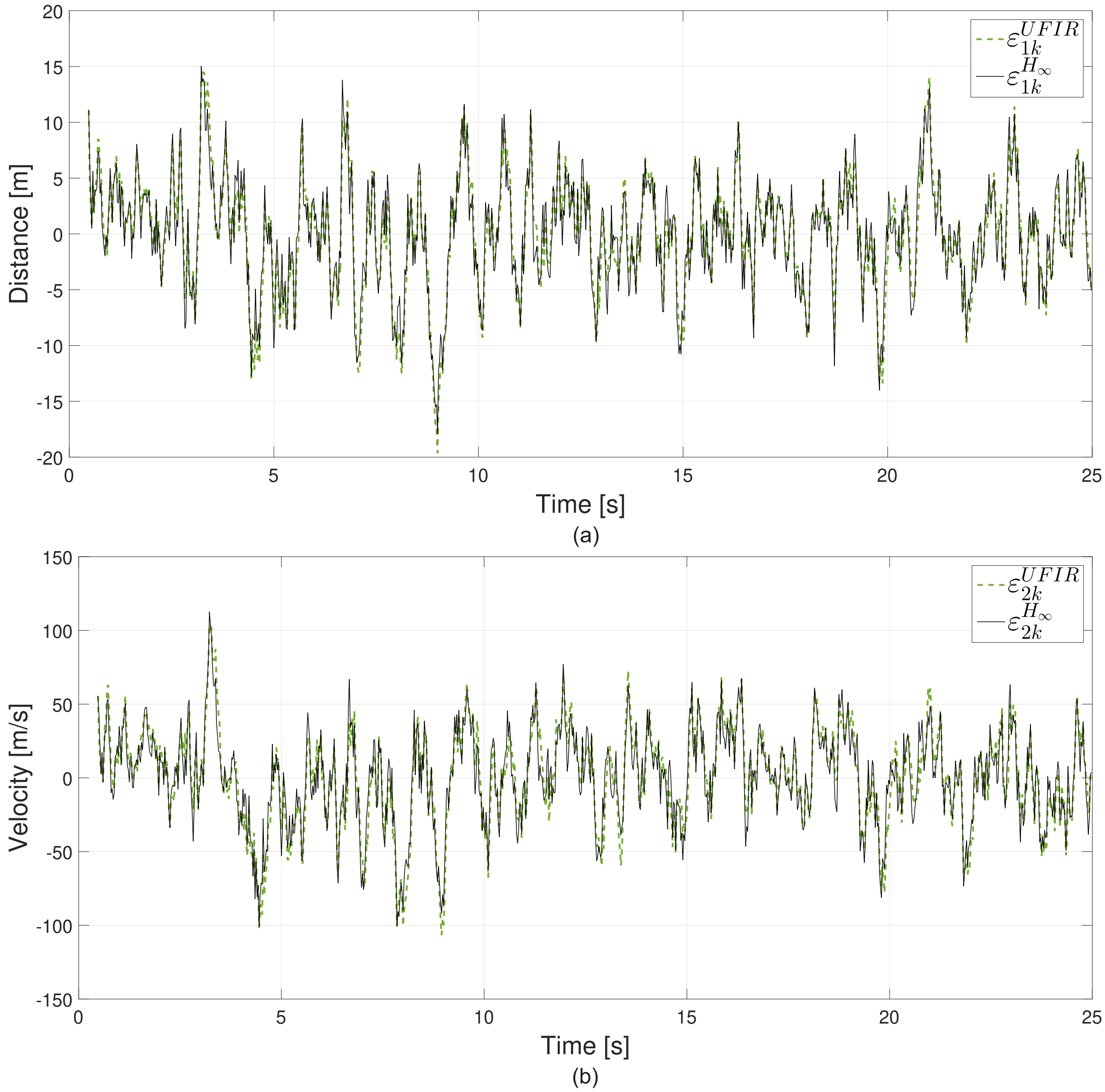


Figure 2.3: Filtering errors produced by the filters in the example system for: (a) first state and (b) second state.

Table 2.1: RMSEs produced by the filters.

Filter	RMSE
UFIR	33.2906
$\mathbf{H_\infty}$ FIR	**31.6697**

Chapter 3

Iterative Algorithm for H_∞ FIR Filter Gain Computation

The state-space equations in (2.45) and (2.46) can't be used directly on the inequality given by lemma 2 due to weight P_y in (2.31) corresponds to P_ε in the H_∞ FIR Filtering problem and $P_\varepsilon = P_k = \mathcal{E}\{\varepsilon_k \varepsilon_k^T\}$ is function of the gain $\mathcal{H}_N$, and for that reason, the inequality is nonlinear with respect to the gain $\mathcal{H}_N$. A solution for nonlinearities was proposed where the inequality in (2.31) is substituted with P_ε as function of $\mathcal{H}_N$, this solution gives an inequality where the product $\mathcal{H}_N^T \mathcal{H}_N$ makes nonlinear with respect to the gain $\mathcal{H}_N$ but introducing a new variable $\mathcal{Z}$ it can be transformed into a LMI by adding the inequality $\mathcal{Z} > \mathcal{H}_N^T \mathcal{H}_N$, then by increasing the trace of $\mathcal{Z}$ and checking if the constraint $\mathrm{tr}(\mathcal{Z}) = \mathrm{tr}(\mathcal{H}_N^T \mathcal{H}_N)$ is satisfied, the H_∞ FIR filter gain $\mathcal{H}_N$ is obtained (Algorithm 1). Another alternative to avoid the nonlinearity in the inequality in Lemma 2 is the iterative algorithm presented bellow.

3.1 Generalized Noise Power Gain

An important indicator of the effectiveness of FIR filtering is the Noise Power Gain (NPG) introduced by Trench in [42]. The NPG is the ratio of the output noise variance σ_{out}^2 to the input noise variance σ_{in}^2, which is akin to the noise figure in wireless communications. For WGN, the NPG is equal to the sum of the squared coefficients of the FIR h_k,

$$\mathrm{NPG} = \frac{\sigma_{\text{out}}^2}{\sigma_{\text{in}}^2} = \sum_{k=0}^{N-1} h_k^2, \tag{3.1}$$

which is the squared norm of h_k.

In state space, the gain $\mathcal{H}_N$ represents the coefficients of the FIR filter impulse response. Therefore, the product $\mathcal{H}_N\mathcal{H}_N^T$ plays the role of a *generalized* NPG (GNPG) [38]. The GNPG $\mathcal{G}_k$ can be written as

$$\mathcal{G}_k = \mathcal{H}_N\mathcal{H}_N^T. \tag{3.2}$$

It follows that GNPG is a symmetric square matrix $\mathcal{G}_k = \mathcal{G}_k^T \in \mathbb{R}^{K\times K}$, where the main diagonal components represent the NPGs for the system states, and the remaining components the cross NPGs. The main property of $\mathcal{G}_k$ is that its trace decreases with increasing horizon length, which provides effective noise reduction. On the other hand, an increase in N causes an increase in bias errors, and therefore $\mathcal{G}_k$ must be optimally set by choosing an optimal horizon length.

3.2 Iterative Algorithm for $\mathbf{H}_\infty$ FIR Filter Gain Computation

First, take the inequality in (2.37) and replace F, B, H, P_w and P_y with $\tilde{F}_\varsigma$, $\tilde{B}_\varsigma$, $\tilde{C}_\varsigma$, P_ς and P_ε, respectively,

$$\begin{pmatrix} -K^{-1} & \tilde{F}_\varsigma & \tilde{B}_\varsigma & 0 \\ \tilde{F}_\varsigma^T & -K & 0 & \tilde{F}_\varsigma^T\tilde{C}_\varsigma^T \\ \tilde{B}_\varsigma^T & 0 & -\gamma^2 P_\varsigma & \tilde{B}_\varsigma^T\tilde{C}_\varsigma^T \\ 0 & \tilde{C}_\varsigma\tilde{F}_\varsigma & \tilde{C}_\varsigma\tilde{B}_\varsigma & -P_\varepsilon^{-1} \end{pmatrix} < 0. \tag{3.3}$$

Inequality in (3.3) is nonlinear with respect to the symmetric positive-definite matrix K, to avoid the inversion of K, pre- and post-multiply the matrix (3.3) with

$$\begin{pmatrix} K & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}$$

and obtain another form for the inequality,

$$\begin{pmatrix} -K & K\tilde{F}_\varsigma & K\tilde{B}_\varsigma & 0 \\ \tilde{F}_\varsigma^T K & -K & 0 & \tilde{F}_\varsigma^T\tilde{C}_\varsigma^T \\ \tilde{B}_\varsigma^T K & 0 & -\gamma^2 P_\varsigma & \tilde{B}_\varsigma^T\tilde{C}_\varsigma^T \\ 0 & \tilde{C}_\varsigma\tilde{F}_\varsigma & \tilde{C}_\varsigma\tilde{B}_\varsigma & -P_\varepsilon^{-1} \end{pmatrix} < 0. \tag{3.4}$$

If P_ε is considered as constant then the inequality in (3.4) could be considered as a LMI. The weight matrix P_ε could be constant if before solving the LMI, a previous value of P_ε is computed using

$$P_\varepsilon = \mathcal{A} - \mathcal{C}\mathcal{H}_N^T - \mathcal{H}_N\mathcal{C}^T + \mathcal{H}_N\mathcal{D}\mathcal{H}_N^T, \tag{3.5}$$

where $\mathcal{A} = F^{N-1}\chi_m(F^{N-1})^T + \bar{D}_N\mathcal{Q}_N\bar{D}_N^T$, $\mathcal{C} = F^{N-1}\chi_m H_N^T + \bar{D}_N\mathcal{Q}_N G_N^T$, $\mathcal{D} = H_N\chi_m H_N^T + \Omega_N$, $\Omega_N = G_N\mathcal{Q}_N G_N^T + \mathcal{R}_N$, $\chi_m = \mathcal{E}\{x_m x_m^T\} = \hat{x}_m\hat{x}_m^T + P_{\varepsilon_m}$, $\mathcal{Q}_N = \mathcal{E}\{W_{m,k}W_{m,k}^T\}$ and $\mathcal{R}_N = \mathcal{E}\{V_{m,k}V_{m,k}^T\}$.

Define P_{prev} as the value of P_ε computed before solving the minimization problem in which, the gain that could be used for computing P_{prev} at the first time is the UFIR gain $\hat{\mathcal{H}}_N = F^{N-1}(H_N^T H_N)^{-1}H_N^T$. At this time P_{prev} is a constant so if P_ε is replaced with P_{prev}, (3.4) could be represented as the LMI

$$\begin{pmatrix} -K & K\tilde{F}_\varsigma & K\tilde{B}_\varsigma & 0 \\ \tilde{F}_\varsigma^T K & -K & 0 & \tilde{F}_\varsigma^T\tilde{C}_\varsigma^T \\ \tilde{B}_\varsigma^T K & 0 & -\gamma^2 P_\varsigma & \tilde{B}_\varsigma^T\tilde{C}_\varsigma^T \\ 0 & \tilde{C}_\varsigma\tilde{F}_\varsigma & \tilde{C}_\varsigma\tilde{B}_\varsigma & -P_{\text{prev}}^{-1} \end{pmatrix} < 0. \tag{3.6}$$

After solving the LMI in (3.6) a value for $\mathcal{H}_N$ will be obtained which will be used for computing a new P_ε using (3.5), whose trace is expected to be lower than the trace of P_{prev}. One way to guarantee that the trace of P_ε will be lower then the trace of P_{prev} is adding more constraints to the minimization problem. To do this, first define the GNPG before solving the LMI as $\mathcal{G}_{\text{prev}}$, then define a new variable $\mathcal{G}$ such that

$$\mathcal{G} > \mathcal{H}_N\mathcal{H}_N^T.$$

Now rewrite it as

$$\mathcal{G} - \mathcal{H}_N\mathcal{H}_N^T > 0.$$

If the Schur complement is used, the last inequality can be equivalently replaced with the LMI as

$$\begin{pmatrix} \mathcal{G} & \mathcal{H}_N \\ \mathcal{H}_N^T & I \end{pmatrix} > 0. \tag{3.7}$$

A important property of the GNPG is that for a fixed N, if the trace of $\mathcal{G}$ is maximized, then the value of the trace of the error covariance P_k will be minimized. Therefore, if the trace of $\mathcal{G}$ is greater than the trace of $\mathcal{G}_{\text{prev}}$, then the trace of P_ε will be lower than the trace of P_{prev}. Hence, the gain $\mathcal{H}_N$ that will be used for computing P_ε could be found numerically by solving the following minimization problem iteratively,

$$\mathcal{H}_N = \min_{\mathcal{H}_N, \mathcal{G}_2, K, \gamma^2} \gamma^2$$
$$\text{subject to (3.6), (3.7) and } \operatorname{tr}(\mathcal{G}) > \operatorname{tr}(\mathcal{G}_{\text{prev}}). \tag{3.8}$$

Using the gain $\mathcal{H}_N$ computed after solving the minimization problem in (3.8), the trace of P_ε can be computed and compared with the trace of P_{prev} until the condition $\operatorname{tr}(P_\varepsilon) < \operatorname{tr}(P_{\text{prev}})$ does not fit. A pseudo code of the *a posteriori* iterative H_∞ FIR filtering algorithm is listed as Algorithm 2.

Algorithm 2: Iterative Algorithm for H_∞ FIR Filter Gain Computation

Data: $\hat{\mathcal{H}}_N$, P_ς, $\mathcal{Q}_{m,k}$, $\mathcal{R}_{m,k}$

Result: $\mathcal{H}_N$

begin

$\mathcal{H}_N = \hat{\mathcal{H}}_N$; `/* Initialize with UFIR gain */`

$\mathcal{G} = \hat{\mathcal{H}}_N \hat{\mathcal{H}}_N^T$;

do

$P_{\text{prev}} = \mathcal{A} - \mathcal{C}\mathcal{H}_N^T - \mathcal{H}_N\mathcal{C}^T + \mathcal{H}_N\mathcal{D}\mathcal{H}_N^T$;

$\mathcal{G}_{\text{prev}} = \mathcal{H}_N\mathcal{H}_N^T$;

$\mathcal{H}_N = \min \gamma^2$ subject to (3.6), (3.7) and $\operatorname{tr}(\mathcal{G}) > \operatorname{tr}(\mathcal{G}_{\text{prev}})$;

$P_\varepsilon = \mathcal{A} - \mathcal{C}\mathcal{H}_N^T - \mathcal{H}_N\mathcal{C}^T + \mathcal{H}_N\mathcal{D}\mathcal{H}_N^T$;

while $\operatorname{tr}(P_\varepsilon) < \operatorname{tr}(P_{prev})$;

end

Using the gain $\mathcal{H}_N$, numerically determined by using Algorithm 2, the *a posteriori* H_∞ FIR filtering estimate and error covariance for uncorrelated w_k, v_k, and x_m as, respectively,

$$\hat{x}_k = \mathcal{H}_N Y_{m,k}, \tag{3.9}$$
$$P_k = \mathcal{B}_N \chi_m \mathcal{B}_N^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T, \tag{3.10}$$

where the error residual matrices are given by (2.40)-(2.42).

3.3 Numerical Example

Consider the tracking problem described in section 2.5, but for the disturbance w_k and the measurement noise v_k consider the following cases:

1. Considering both as white gaussian.

2. Considering disturbance w_k as Gauss-Markov colored and measurement noise v_k as gaussian.

3. Considering disturbance w_k as gaussian and measurement noise v_k as Gauss-Markov.

For all cases, the H_∞ FIR filter gain will be computed using Algorithm 2, and also using Algorithm 1, the filtering errors of the H_∞ FIR filter will be compared against the filtering errors of the KF, OFIR [36] and UFIR filters.

3.3.1 White Gaussian Disturbance and Measurement Noise

First, consider $w_k \sim \mathcal{N}(0, \sigma_w^2)$ and $v_k \sim \mathcal{N}(0, \sigma_v^2)$, extend the state space (2.65)-(2.66) on $[m, k]$, and compute the *a posteriori* H_∞ FIR filter gain numerically by using Algorithm 1 and Algorithm 2. The estimation can be computed as $\hat{x}_k = \mathcal{H}_N Y_{m,k}$.

The estimations were computed with the standard deviations $\sigma_w = 12$ m/s and $\sigma_v = 10$ m and with a sample period of $\tau = 0.025$ s, using this parameters, the optimal horizon for the UFIR computation is $N_{\text{opt}} = 20$.

The behavioral of Algorithm 2 could be seen graphically on Fig. 3.1, in Fig. 3.1(a) is shown the minimization of γ as the trace of $\mathcal{G}$ is increased, while Fig. 3.1(b) shows how the trace of P_ε is minimized as the trace of $\mathcal{G}$ is increased, as can be seen, the algorithm works while the trace of P_ε is minimized, if the trace starts to grew, then the algorithm is ended and the H_∞ FIR filter gain is obtained.

The RMSE for each filter is given in Table 3.1, it was obvious that KF and OFIR filter will give the most accurate estimations because they were designed to operate under white gaussian disturbance and measurement noise, so in this conditions the best option is to use KF or OFIR filter, but the errors given by the H_∞ FIR filter are near to the KF or OFIR errors. The robust UFIR filter gives the less accurate estimation than other filters, but what is really important is that both H_∞ FIR filters gives similar errors, which means that the gain is similar too, but the estimation with the gain computed using Algorithm 1 gives better estimates which means that this algorithm gives a better accuracy in the H_∞ FIR filter gain computation, but Algorithm 2 found the gain in less time than Algorithm 1, thus, the user must decide for himself which algorithm suits him best if the one more accurate or the fastest one.

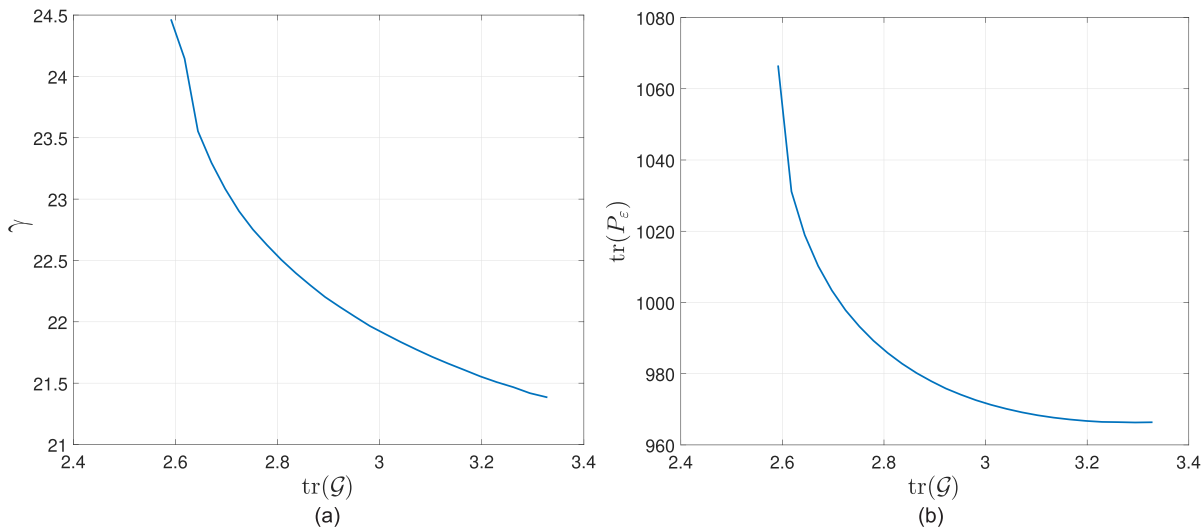


Figure 3.1: Solving the minimization problem using Algorithm 2: (a) minimizing γ as function of the trace of $\mathcal{G}$ and (b) the trace of error covariance P_ε as function of the trace of $\mathcal{G}$.

Table 3.1: RMSEs produced by the filters.

Filter	RMSE
KF	31.2242
OFIR	31.6069
UFIR	33.2906
H_∞ FIR (Algorithm 1)	31.6697
H_∞ FIR (Algorithm 2)	31.6999

3.3.2 Colored Gauss-Markov Disturbance

Consider the vehicle tracking problem described in (2.65)-(2.66), where the white Gaussian measurement noise $v_k \sim \mathcal{N}(0, \sigma_v^2)$, has the standard deviation $\sigma_v^2 = 10$ m/s, for this case the vehicle trajectory is affected by the Gauss-Markov process disturbance $w_k = \Theta w_{k-1} + \mu_k$, where the scalar disturbance factor Θ is chosen as $0 < \Theta < 1$, and the driving noise $\mu_k \sim \mathcal{N}(0, \sigma_\mu^2)$ has a standard deviation $\sigma_\mu = 12$ m/s.

To compare the estimation errors, the disturbance process is generated by changing Θ from 0.05 to 0.95 with a step 0.05. Next, the following scenarios of filter optimal tuning are considered:

1. For Θ and $N_{\text{opt}}(\Theta)$.

2. For $\Theta = 0.05$ and $N_{\text{opt}} = 17$.

3. For $\Theta = 0.95$ and $N_{\text{opt}} = 7$.

Typical tracking RMSEs are shown in Fig. 3.2 as functions of Θ, and it can be stated the following features:

- *Case 1 (theoretical): Tuning for Θ and $N_{\text{opt}}(\Theta)$.* When the filters are optimally tuned for Θ, their RMSEs reach the lowest possible values, as shown in Fig. 3.2(a). The H_∞ FIR (Algorithms 1 and 2) filter and the UFIR filter do it with a lower rate that speaks in favor of their higher robustness. Note that filter tuning to each Θ is hardly possible in practice, so this case can be considered theoretical.

- *Case 2 (regular): Tuning for $\Theta = 0.05$ and $N_{\text{opt}} = 17$.* When the disturbance is not specified, all filters are usually tuned near to white noise this is the reason this is considered the regular case. Increasing Θ causes all errors to increase at a high rate. Accordingly, all filters produce consistent and large errors when Θ reaches 0.95, as shown in Fig. 3.2(b).

- *Case 3 (robust): Tuning for $\Theta = 0.95$ and $N_{\text{opt}} = 7$.* When the disturbance boundary is known, all filter can be tuned for $\Theta = 0.95$. This drastically lowers the RMSEs in all filters compared to tuning for $\Theta = 0.05$ (Fig. 3.2(b)). It can be seen that all filters demonstrate a better robustness. However, the highest robustness is exhibited by the H_∞ FIR filter (Algorithms 1 and 2) and the UFIR filter. Among these filters, as is shown in Fig. 3.2(c), the UFIR filter looks a bit more robust, although a bit less accurate, as expected.

Comparing the RMSEs shown in Fig. 3.2(a) and Fig. 3.2(c), it can be concluded that tuning for each Θ gives the smallest errors, but can hardly be implemented practically. On the contrary, tuning for $\Theta = 0.95$ gives slightly more errors, but this case is feasible and robust. When the H_∞ FIR filter is tuned near to white noise (Fig. 3.2(b)), its behavioral is the same like the other filters, increasing the errors as the color factor is increased. By the other hand, if the H_∞ is tuned to the maximized disturbance (Fig. 3.2(c)) then the errors doesn't have a significant difference if the disturbance is decreased, which means robustness.

One way to see the comparison between of the robustness ϱ of the filters tuned in robust mode is with the ratio of the RMSE when the model has $\Theta_{\max}$ and the RMSE when the model has $\Theta_{\min}$.

$$0 < \varrho = \frac{\text{RMSE}(\Theta_{\min})}{\text{RMSE}(\Theta_{\max})} < 1 \tag{3.11}$$

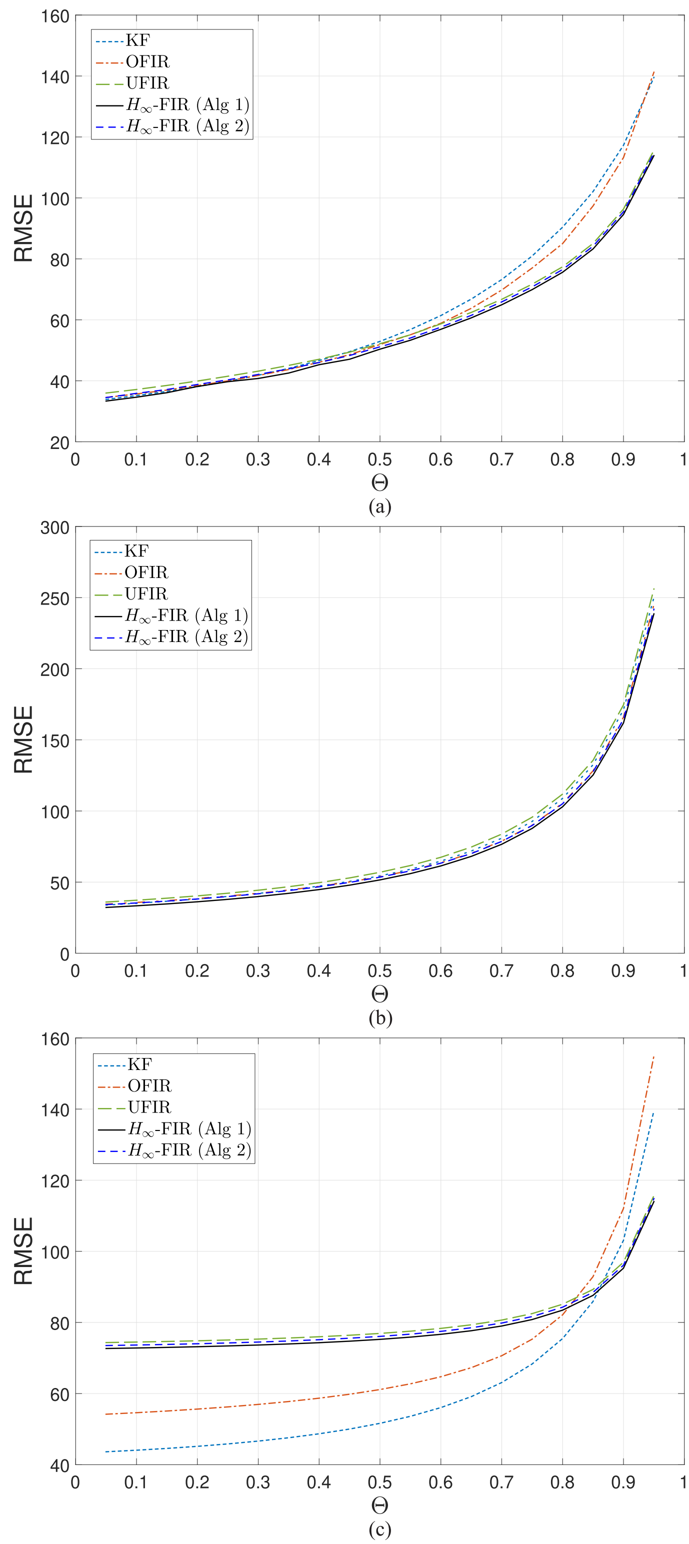


Figure 3.2: Typical RMSEs generated by the filters as functions of the process color factor $0.05 \leqslant \Theta \leqslant 0.95$ in different scenarios of tuning: (a) *theoretical:* tuning to each Θ, (b) *regular:* tuning to $\Theta = 0.05$ that corresponds to $N_{\text{opt}} = 17$, and (c) *robust:* tuning to $\Theta = 0.95$ that corresponds to $N_{\text{opt}} = 7$.

This measure suggests that the filter has the highest robustness $\varrho = 1$ when $\text{RMSE}(\Theta_{\min}) = \text{RMSE}(\Theta_{\max})$, and the lowest robustness $\varrho = 0$ when $\text{RMSE}(\Theta_{\min}) = 0$. Since these boundaries are practically unreachable, it is considered $0 < \varrho < 1$. Table 3.2 shows the robustness ϱ of each filter tuned for the robust case (Fig. 3.2(c)), it can be seen that H_∞ FIR filter has almost the same robustness than the UFIR filter.

Table 3.2: Robustness ϱ of the filters tuned in robust mode (Gauss-Markov disturbance).

Filter	ϱ
KF	0.3127
OFIR	0.3498
UFIR	0.6428
H_∞ FIR Alg. 1	0.6196
H_∞ FIR Alg. 2	0.6196

It is worth noting for practical usefulness that further maximizing the disturbance matrix $\mathcal{Q}_N$ by the factor of 4 does not change the RMSE with a noticeable difference, so the H_∞ FIR filter produces almost the same errors. This means that the robust mode is completely achievable in H_∞ FIR filtering by maximizing the errors for $\Theta_{\max}$.

3.3.3 Colored Gauss-Markov Measurement Noise

Consider the vehicle tracking problem with the white Gaussian disturbance $w_k \sim \mathcal{N}(0, \sigma_w^2)$ and the colored noise $v_k = \Psi v_{k-1} + \xi_k$, where $\xi \sim \mathcal{N}(0, \sigma_\xi^2)$ and the color factor $0 < \Psi < 1$ is chosen for stability. Using $\sigma_w = 10$ m/s, $\sigma_\xi = 12$ m/s, the same scenarios of tuning are considered:

1. For Ψ and $N_{\text{opt}}(\Psi)$.
2. For $\Psi = 0.05$ and $N_{\text{opt}} = 23$.
3. For $\Psi = 0.95$ and $N_{\text{opt}} = 66$.

Fig. 3.3 shows the RMSEs produced by filters as functions of the colored noise factor Ψ. The theoretical case (tuning for Ψ and $N_{\text{opt}}(\Psi)$) is shown in Fig. 3.3(a), as in the colored disturbance example, all filters produce consistent errors that grow and at a low rate.

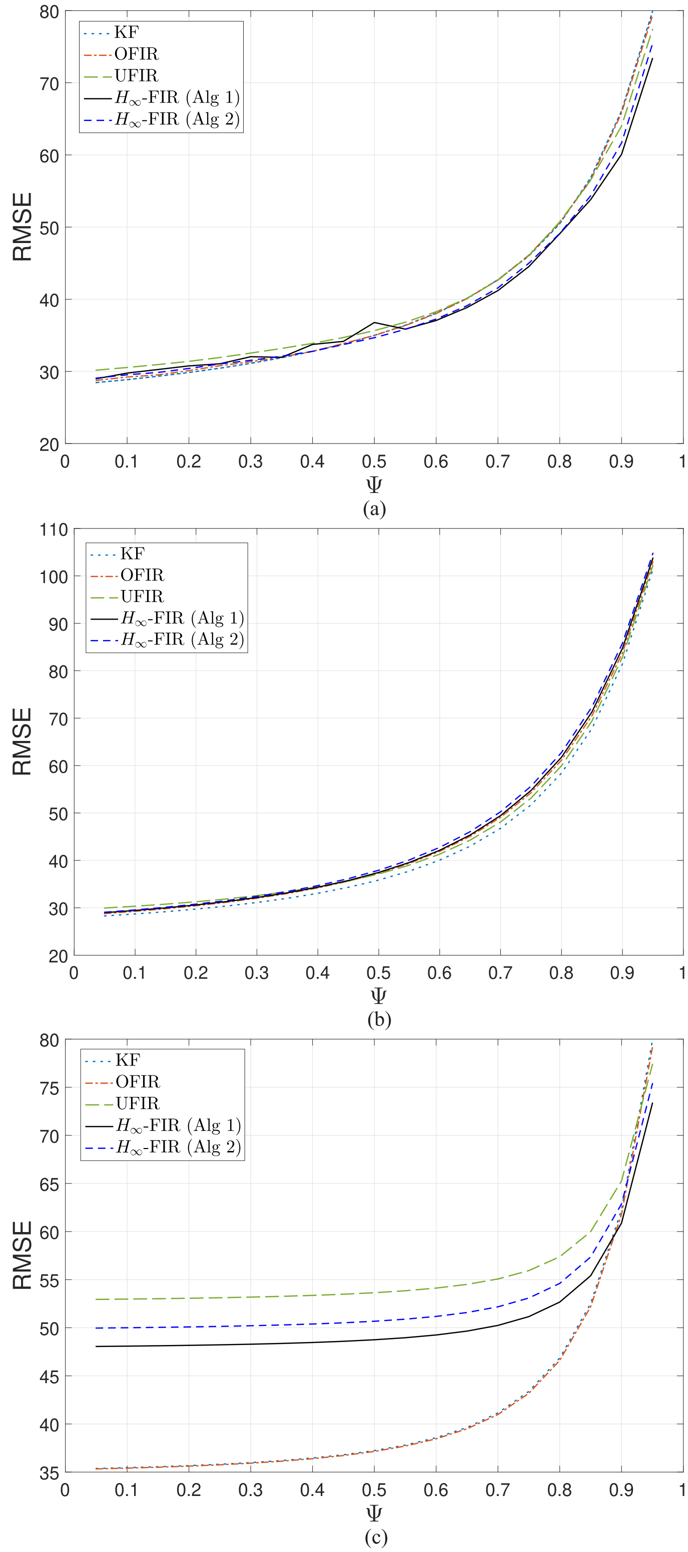


Figure 3.3: Typical RMSEs generated by the filters as functions of the colored measurement noise factor $0.05 \leqslant \psi \leqslant 0.95$ in different scenarios of tuning: (a) *theoretical*: tuning to each ψ, (b) *regular:* tuning to $\psi = 0.05$ that corresponds to $N_{\text{opt}} = 23$, and (c) *robust:* tuning to $\psi = 0.95$ that corresponds to $N_{\text{opt}} = 66$.

The regular case (tuning for $\Psi = 0.05$ and $N_{\text{opt}} = 23$) can be seen in Fig. 3.3(b) as in the colored disturbance example, increasing Ψ causes all errors to increase at a high rate. In Fig. 3.3(c) is shown the robust case (tuned for $\Psi = 0.95$ and $N_{\text{opt}} = 66$), in this case, it can be seen that all filters demonstrate a higher robustness.

The H_∞ FIR filter (Algorithms 1 and 2) and the UFIR filter demonstrate almost the same robustness, although the H_∞ is more successful in accuracy. Finally, it must be noticed that maximizing the measurement noise matrix $\mathcal{R}_N$ by the factor of 4 does not significantly change in the errors.

Analyzing the robustness ϱ of the filters tuned for the robust case (Table 3.3), it can be noticed that the H_∞ FIR filter is almost as robust as the UFIR filter.

Table 3.3: Robustness ϱ of the filters tuned in robust mode (Gauss-Markov measurement noise).

Filter	ϱ
KF	0.4429
OFIR	0.4468
UFIR	0.6843
H_∞ FIR Alg. 1	0.6552
H_∞ FIR Alg. 2	0.6630

3.4 Experimental Verification

Now, the accuracy and robustness of the H_∞ FIR filter is tested using real data from the KUKA LWR IV+ robot angular velocity measured along the coordinate x. Data available from [33] shows that a robot travels with a velocity that varies quasi periodically about the mean value, as shown in Fig. 3.4(a). If the average "slow" velocity is desired to be known, the fast variations can be considered as colored process noise (CPN). Since the original data in Fig. 3.4(a) are highly oversampled, the signal must be thinned in time, the thinning factor is selected as 56, and then the following polynomial tracking model is used,

$$x_k = F x_{k-1} + B w_k, \tag{3.12}$$

$$y_k = H x_k + D w_k + v_k, \tag{3.13}$$

where the first state is the velocity of the robot and the second state its acceleration. The model's matrices are given by

$$F = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1/\tau \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

where $\tau = 0.56$ s is the thinned sample period and D will be computed numerically.

To have a pseudo ground truth, the UFIR *a posteriori* filtering estimate $\hat{x}_k^{\mathrm{U}}$ is computed on the horizon of $N_{\mathrm{opt}} = 64$ points that corresponds to the horizon of 36 s and averages 8-9 quasi periodic variations. Then a $(q = 32)$-lag UFIR smoothing estimate $x_{k-q} = F^{-q}\hat{x}_k^{\mathrm{U}}$ is obtained, this smoothed estimation could be considered as the pseudo ground truth. The thinned data and the pseudo ground truth are shown in Fig. 3.4(b).

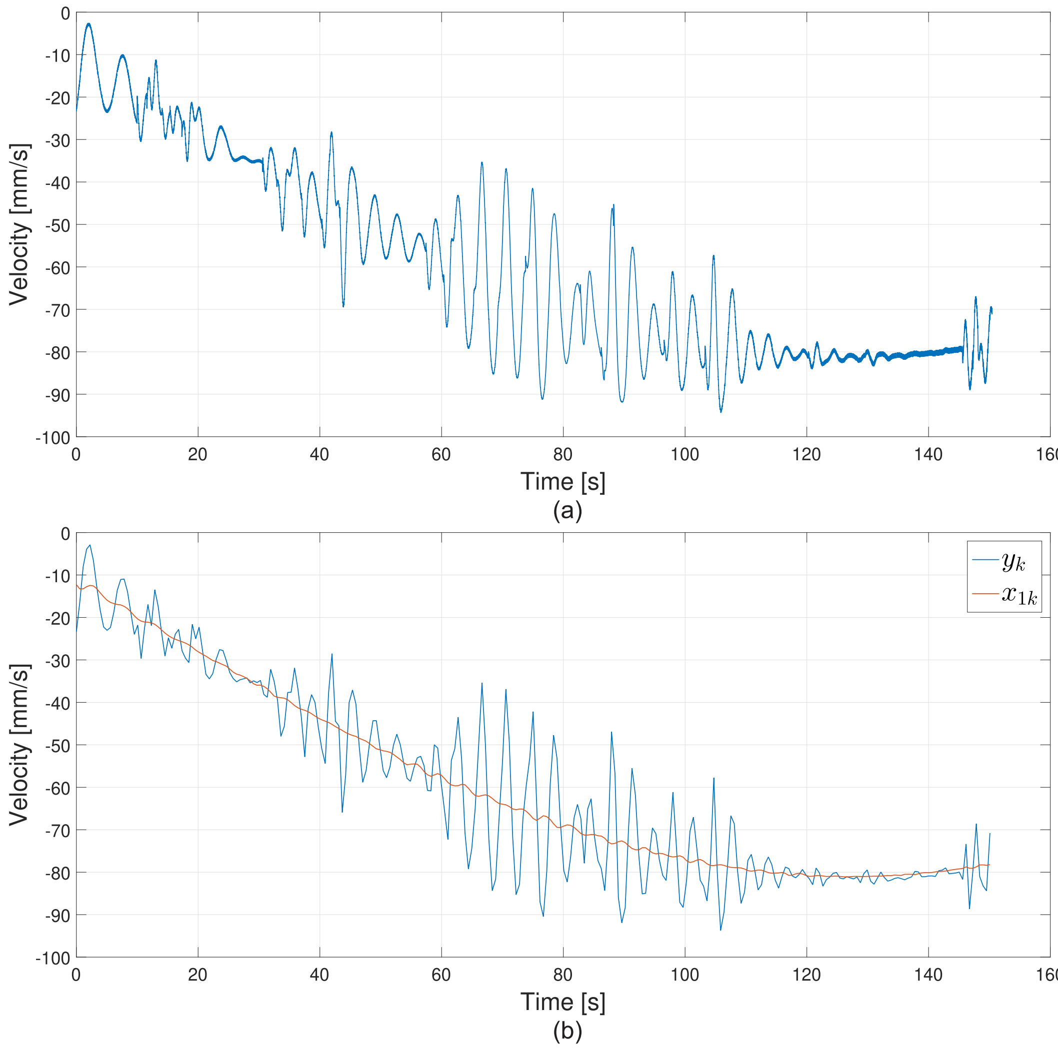


Figure 3.4: (a) Measurements of the velocity of the robot KUKA LWR IV+. (b) Thinned in time measurements and the first state of the pseudo ground truth.

In the test it will be assumed w_k as a colored Gauss-Markov disturbance and v_k as white gaussian measurement noise. The accuracy of the H_∞ FIR filter will be compared with the accuracy of the KF, OFIR, Optimal unbiased FIR (OUFIR) [37], maximum likelihood FIR (ML-FIR) [37], UFIR ($N_{\text{opt}} = 48$), H_2-OFIR [28], H_2-OUFIR filters [29].

The measurement noise v_k is computed as follows, first, rewrite (3.13) as

$$v_k = y_k - (Hx_k + Dw_k),$$

where x_k is the pseudo ground truth. At this step, the value of D is unknown, so the term $Hx_k + Dw_k$ can't be computed, but this term can be considered as the measurements average $\bar{y}_k$, then the measurement noise can be computed as,

$$v_k = y_k - \bar{y}_k,$$

the average $\bar{y}_k$ can be computed using the UFIR ($N = 10$) estimation of the oversampled measurements and then thinning in time by 56 the result. The difference $y_k - \bar{y}_k$ is shown in Fig. 3.5(a) and this difference is considered the white gaussian measurement noise v_k, the standard deviation σ_v is computed as $\mathcal{E}\{v_k v_k^T\}$, obtaining a value of 400.16 μm/s.

The disturbance can be computed as follows, first write the product Dw_k in terms of the measurements average $\bar{y}_k$,

$$Dw_k = \bar{y}_k - Hx_k,$$

and replace it on (3.12),

$$x_k = Fx_{k-1} + BD^{-1}(\bar{y}_k - Hx_k), \tag{3.14}$$

using x_k as the pseudo ground truth, D can be computed numerically by changing its value until the right-hand side of (3.14) matches with the pseudo ground truth. In this case, D was computed obtaining a value of -0.5350252. Using this value of D, the disturbance w_k can be computed as

$$w_k = D^{-1}(\bar{y}_k - Hx_k)$$

the disturbance is considered as a Gauss-Markov process, hence μ_k can be computed as $\mu_k = w_k - \Theta w_{k-1}$, where Θ is selected as the value that causes a white gaussian behavioral for μ_k. Fig. 3.5(b) shows the computed Gauss-Markov disturbance with $\Theta = 0.64$ and $\sigma_\mu = 13.03$ mm/s.

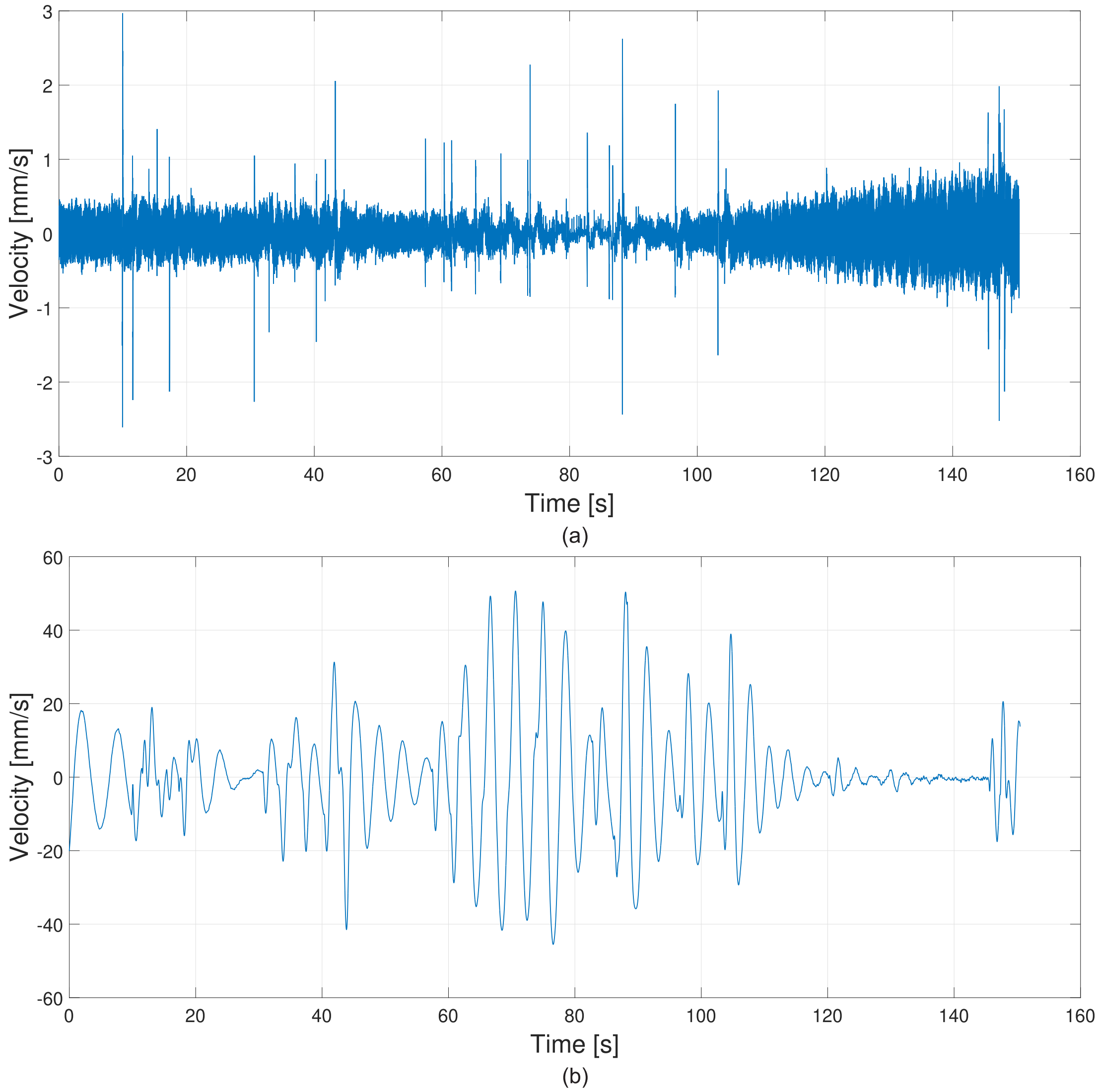


Figure 3.5: (a) White gaussian measurement noise with $\sigma_v = 400.16$ μm/s, (b) Gauss-Markov disturbance with $\Theta = 0.64$ and $\sigma_\mu = 13.03$ mm/s.

The filters were tuned using the values of Θ, σ_μ and σ_v computed before. The errors of each filter are shown in Fig. 3.6, as can be seen, in presence of a colored Gauss-Markov disturbance, the robust filters (UFIR, H_2-OUFIR and H_∞ FIR) has the most accurate estimations, Table 3.4 shows the RMSE produced by each filters, under colored disturbance KF, OFIR, OUFIR and ML-FIR don't give an accurate estimation, UFIR filter being a robust filter, gives a good estimation, it was expected that H_2-OFIR gives better estimations than UFIR, but H_2 has larger errors than UFIR, it can be caused by the size of the batch ($N = 48$) and it needs a bigger batch, H_2-OUFIR has a good estimation, but the H_∞ FIR filter (computed with any algorithm) has less errors than any other filter.

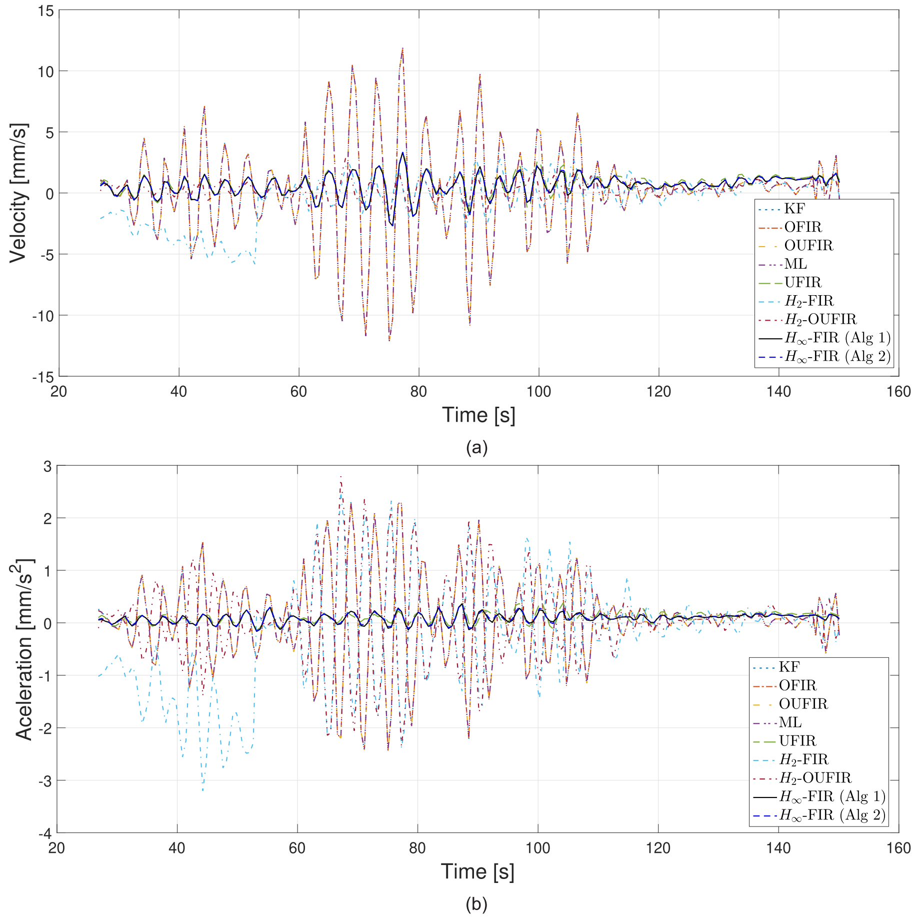


Figure 3.6: Filtering errors in the first case for (a) first state and (b) second state

Table 3.4: RMSE produced by the filters in the experiment using real data.

Filter	RMSE
KF	3.9349
OFIR	3.9346
OUFIR	3.9344
ML	3.9344
UFIR	1.1203
H_2-OFIR	2.2917
H_2-OUFIR	1.1051
$\mathbf{H_\infty}$ FIR Alg. 1	**1.0615**
H_∞ FIR Alg. 2	1.0724

Chapter 4

Recursive *a posteriori* H_∞ Filter

The batch form of the H_∞ FIR filter is computationally time consuming, especially when $N \gg 1$, due to large dimensions of all extended vectors and matrices. An efficient computation of (2.62) can be done if recursions for $\hat{x}_k$ are found into an iterative algorithm.

The objective is to find a bias correction gain for a Kalman-like algorithm such as the H_∞ norm of the ς-to-ε is minimized

$$K^{H_\infty} \Leftarrow \sup_{\varsigma \neq 0} \frac{\sum_{i=m}^{k} \varepsilon_i^T P_\varepsilon \varepsilon_i}{\sum_{i=m}^{k} \varsigma_i^T P_\varsigma \varsigma_i} < \gamma^2, \tag{4.1}$$

where K^{H_∞} is the recursive H_∞ filter bias correction gain. As in the batch form, the bias correction gain K^{H_∞} can be defined numerically for a given small positive $\gamma > 0$.

4.1 Kalman Filering Algorithm

The recursive algorithm of the Bayesian estimator associated with one-state linear models [37] can easily be extended to the K-state linear model

$$x_k = F_k x_{k-1} + E_k u_k + B_k w_k, \tag{4.2}$$

$$y_k = H_k x_k + D_k w_k + v_k, \tag{4.3}$$

where the noise vectors $w_k \sim N(0, Q_k) \in \mathbb{R}^M$ and $v_k \sim N(0, R_k) \in \mathbb{R}^P$ are uncorrelated. The corresponding algorithm was derived by Kalman in his seminal paper [15] and is now commonly known as Kalman filter (KF). Kalman derived his recursive filtering algorithm in 1960 by applying the Bayesian approach to linear processes with WGN.

The KF algorithm operates in two phases:

1. In the *prediction phase*, the *a priori* estimate and error covariance are predicted at k using using measurements at $k-1$.

2. In the *update phase*, the *a priori* values are updated at k to the *a posteriori* estimate and error covariance using measurement at k.

Consider a K-state space model (4.2)-(4.3). The Bayesian approach can be applied similarly to the one-state case, as was done by Kalman in [15].

The first thing to note is that the most reasonable prior estimate can be taken from (4.2) for the known estimate $\hat{x}_{k-1}$ and input u_k if the zero mean noise w_k is ignored. This gives the prior estimate

$$\hat{x}_k^- = F_k\hat{x}_{k-1} + E_k u_k. \tag{4.4}$$

To update $\hat{x}_k^-$, is needed to involve the observation (4.3). This can be done if the *measurement residual* is considered

$$\begin{aligned} s_k &= y_k - H_k\hat{x}_k^- \\ &= H_k x_k + v_k - H_k\hat{x}_k^- \\ &= H_k(x_k - \hat{x}_k^-) + v_k \\ &= H_k\varepsilon_k^- + v_k, \end{aligned} \tag{4.5}$$

as can be seen in (4.5), the measurement residual is the difference between data y_k and the predicted data $H_k\hat{x}_k^-$. The measurement residual covariance $S_k = S_k^T = \mathcal{E}\{s_k s_k^T\}$ can then be written as

$$\begin{aligned} S_k &= \mathcal{E}\{(H_k\varepsilon_k^- + v_k)(H_k\varepsilon_k^- + v_k)^T\} \\ &= H_k P_k^- H_k^T + R_k. \end{aligned} \tag{4.6}$$

Since s_k is generally biased, because $\hat{x}_k^-$ is generally biased, the prior estimate can be updated as

$$\hat{x}_k = \hat{x}_k^- + K_k s_k, \tag{4.7}$$

where the matrix K_k is introduced to correct the bias in $\hat{x}_k^-$. Therefore, K_k plays the role of the *bias correction gain.*

As can be seen, the estimate (4.7) will be optimal if K_k is found such that the mean square error (MSE) is minimized. To do this, refer to (4.2)-(4.3) and define the errors

$$\begin{aligned}
\varepsilon_k^- &= x_k - \hat{x}_k^-, \\
\varepsilon_k &= x_k - \hat{x}_k, \\
P_k^- &= \mathcal{E}\left\{\varepsilon_k^- \varepsilon_k^{-T}\right\}, \\
P_k &= \mathcal{E}\left\{\varepsilon_k \varepsilon_k^T\right\}
\end{aligned}$$

involving (4.4)–(4.7).

The prior estimation error ε_k^- can be transformed as

$$\begin{aligned}
\varepsilon_k^- &= F_k x_{k-1} + E_k u_k + B_k w_k - F_k \hat{x}_{k-1} - E_k u_k \\
&= F_k(x_{k-1} - \hat{x}_{k-1}) + B_k w_k \\
&= F_k \varepsilon_{k-1} + B_k w_k
\end{aligned} \tag{4.8}$$

and, for mutually uncorrelated ε_{k-1} and w_k, the prior error covariance P_k^- can be transformed to

$$P_k^- = F_k P_{k-1} F_k^T + B_k Q_k B_k^T. \tag{4.9}$$

Next, the estimation error ε_k can be represented by

$$\begin{aligned}
\varepsilon_k &= F_k x_{k-1} + E_k u_k + B_k w_k - F_k \hat{x}_{k-1} - E_k u_k - K_k s_k \\
&= \varepsilon_k^- - K_k(H_k \varepsilon_k^- + v_k) \\
&= (I - K_k H_k)\varepsilon_k^- - K_k v_k
\end{aligned} \tag{4.10}$$

and, for mutually uncorrelated ε_k^- and v_k, the *a posteriori* error covariance P_k can be transformed to

$$\begin{aligned}
P_k &= (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \\
&= P_k^- - P_k^- H_k^T K_k^T - K_k H_k P_k^- + K_k (H_k P_k^- H_k^T + R_k) K_k^T \\
&= P_k^- - P_k^- H_k^T K_k^T - K_k H_k P_k^- + K_k S_k K_k^T.
\end{aligned} \tag{4.11}$$

What is left behind is to find the optimal bias correction gain K_k that minimizes MSE. This can be done by minimizing the trace of P_k, which is a convex function with a minimum corresponding to the optimal K_k. Using the properties of the derivative of the trace of a matrix [4], [12], the minimization of $\mathrm{tr}(P_k)$ by K_k can be carried out if $\frac{\partial}{\partial K_k}\mathrm{tr}(P_k) = 0$ as

$$\begin{aligned}
\frac{\partial}{\partial K_k}\mathrm{tr}(P_k) &= \frac{\partial}{\partial K_k}\mathrm{tr}(P_k^-) - \frac{\partial}{\partial K_k}\mathrm{tr}(P_k^- H_k^T K_k^T) - \frac{\partial}{\partial K_k}\mathrm{tr}(K_k H_k P_k^-) + \frac{\partial}{\partial K_k}\mathrm{tr}(K_k S_k K_k^T) \\
&= \frac{\partial}{\partial K_k}\mathrm{tr}(K_k S_k K_k^T) - 2\frac{\partial}{\partial K_k}\mathrm{tr}(K_k H_k P_k^-) \\
&= 2K_k S_k - 2P_k^- H_k^T = 0.
\end{aligned} \tag{4.12}$$

This gives the optimal bias correction gain

$$K_k = P_k^- H_k^T S_k^{-1}. \tag{4.13}$$

The gain K_k (4.13) is known as the *Kalman gain* and its substitution on the left-hand side of the last term in (4.11) transforms the error covariance P_k into

$$\begin{aligned} P_k &= P_k^- - P_k^- H_k^T K_k^T - K_k H_k P_k^- + P_k^- H_k^T S_k^{-1} S_k K_k^T \\ &= P_k^- - K_k H_k P_k^- \\ &= (I - K_k H_k) P_k^- . \end{aligned} \tag{4.14}$$

The above derived standard KF algorithm gives an estimate at k utilizing data from zero to k. Therefore, it is also known as the *a posteriori* KF algorithm, the pseudo code of which is listed as Algorithm 3.

Algorithm 3: The *a posteriori* (standard) KF Algorithm

Data: $y_k, u_k, \hat{x}_0, P_0, Q_k, R_k$

Result: $\hat{x}_k, P_k$

begin

  for $k = 1, 2, \ldots$ **do**

    $\hat{x}_k^- = F_k \hat{x}_{k-1} + E_k u_k$; `/* a priori state estimate */`

    $P_k^- = F_k P_{k-1} F_k^T + B_k Q_k B_k^T$; `/* a priori error covariance */`

    $s_k = y_k - H_k \hat{x}_k^-$; `/* measurement residual */`

    $S_k = H_k P_k^- H_k^T + R_k$; `/* innovation covariance */`

    $K_k = P_k^- H_k^T S_k^{-1}$; `/* Kalman gain */`

    $\hat{x}_k = \hat{x}_k + K_k s_k$; `/* a posteriori state estimate */`

    $P_k = (I - K_k H_k) P_k^-$; `/* a posteriori error covariance */`

  end

end

The algorithm requires initial values $\hat{x}_0$ and P_0, as well as noise covariances Q_k and R_k to update the estimates starting at $k-1$. Its operation is quite transparent, and recursions are easy to program, fast to compute, and require little memory, making the KF algorithm suitable for many applications.

4.2 Recursive *a posteriori* $\mathbf{H}_\infty$ filter

4.2.1 Recursive Disturbance-to-Error State Space Model

One of the best ways to determine K^{H_∞} is using the BRL defined in Lemma 2. But first a state space model where is desired the disturbance ς_k as the input and the estimation error ε_k as the output. To derive this state space model, consider the BE-based state-space model of the LTI system modified as

$$x_k = Fx_{k-1} + Bw_k, \tag{4.15}$$

$$y_k = Hx_k + v_k \tag{4.16}$$

Referring to (4.4)-(4.11), the recursive state estimation $\hat{x}_k$ can be obtained as

$$\hat{x}_k = F\hat{x}_{k-1} + K^{H_\infty}(y_k - HF\hat{x}_{k-1}), \tag{4.17}$$

and the estimation error ε_k can be represented as

$$\varepsilon_k = (I - K^{H_\infty}H)F\varepsilon_{k-1} + (I - K^{H_\infty}H)Bw_k - K^{H_\infty}v_k, \tag{4.18}$$

Now, introduce a state vector $z_k = \varepsilon_k$ and an augmented vector $\varsigma_k = \begin{pmatrix} w_k^T & v_k^T \end{pmatrix}^T$, and combine them in the following state-space model

$$z_k = \tilde{F}_\varsigma z_{k-1} + \tilde{B}_\varsigma \varsigma_k, \tag{4.19}$$

$$\varepsilon_k = \tilde{C}_\varsigma z_k \tag{4.20}$$

in which the newly introduced block matrices have the form

$$\tilde{F}_\varsigma = (I - K^{H_\infty}H)F, \qquad \tilde{B}_\varsigma = \begin{pmatrix} (I - K^{H_\infty}H)B & -K^{H_\infty} \end{pmatrix}, \qquad \tilde{C}_\varsigma = I. \tag{4.21}$$

4.2.2 Algorithm for Recursive $\mathbf{H}_\infty$ Filter Bias Correction Gain Computation

The state-space equations in (4.19)-(4.20) can't be used directly on the inequality given by lemma 2 due to weight P_y in (2.31) corresponds to P_ε in the recursive H_∞ filtering problem, and referring to (4.11), $P_\varepsilon = P_k = \mathcal{E}\{\varepsilon_k \varepsilon_k^T\}$ is given by

$$P_\varepsilon = P_\varepsilon^- - P_\varepsilon^- H^T {K^{H_\infty}}^T - K^{H_\infty}HP_\varepsilon^- + K^{H_\infty}(HP_\varepsilon^- H^T + R_k){K^{H_\infty}}^T, \tag{4.22}$$

where P_ε^- is the *a priori* error covariance P_k^- in which referring to (4.9) and (4.11) is given by

$$P_\varepsilon^- = F[(I - K^{H_\infty}H)P_{k-1}^-(I - K^{H_\infty}H)^T + K^{H_\infty}R_k{K^{H_\infty}}^T]F^T + BQ_kB^T. \tag{4.23}$$

As can be seen in (4.22), P_ε is function of the bias correction gain K^{H_∞}, which is a desired variable for the minimization of γ in the inequality in (2.31), so the inequality is nonlinear with respect to the bias correction gain K^{H_∞} due to the inversion of P_ε.

To solve this problem, first rewrite P_ε in (4.22) as

$$P_\varepsilon = \begin{pmatrix} I & K^{H_\infty} \end{pmatrix} \begin{pmatrix} P_\varepsilon^- & -P_\varepsilon^- H^T \\ -HP_\varepsilon^- & HP_\varepsilon^- H^T + R_k \end{pmatrix} \begin{pmatrix} I \\ {K^{H_\infty}}^T \end{pmatrix},$$

introduce new auxiliary matrices

$$\tilde{K}_\varsigma = \begin{pmatrix} I & K^{H_\infty} \end{pmatrix}, \tag{4.24}$$

$$P_{\mathcal{J}} = \begin{pmatrix} P_\varepsilon^- & -P_\varepsilon^- H^T \\ -HP_\varepsilon^- & HP_\varepsilon^- H^T + R_k \end{pmatrix}, \tag{4.25}$$

and rewrite P_ε as

$$P_\varepsilon = \tilde{K}_\varsigma P_{\mathcal{J}} \tilde{K}_\varsigma^T \tag{4.26}$$

Now replace F, B, H, P_w and P_y in (2.35) with $\tilde{F}_\varsigma$, $\tilde{B}_\varsigma$, $\tilde{C}_\varsigma$, P_ς and P_ε respectively,

$$\begin{pmatrix} \tilde{F}_\varsigma^T K \tilde{F}_\varsigma + \tilde{F}_\varsigma^T \tilde{C}_\varsigma^T P_\varepsilon \tilde{C}_\varsigma \tilde{F}_\varsigma - K & \tilde{F}_\varsigma^T K \tilde{B}_\varsigma + \tilde{F}_\varsigma^T \tilde{C}_\varsigma^T P_\varepsilon \tilde{C}_\varsigma \tilde{B}_\varsigma \\ \tilde{B}_\varsigma^T K \tilde{F}_\varsigma + \tilde{B}_\varsigma^T \tilde{C}_\varsigma^T P_\varepsilon \tilde{C}_\varsigma \tilde{F}_\varsigma & \tilde{B}_\varsigma^T K \tilde{B}_\varsigma + \tilde{B}_\varsigma^T \tilde{C}_\varsigma^T P_\varepsilon \tilde{C}_\varsigma \tilde{B}_\varsigma - \gamma^2 P_\varsigma \end{pmatrix} < 0, \tag{4.27}$$

where P_ς is defined by (2.54).

Replace (4.26) into (4.27) as

$$\begin{pmatrix} \tilde{F}_\varsigma^T K \tilde{F}_\varsigma + \tilde{F}_\varsigma^T \tilde{C}_\varsigma^T \tilde{K}_\varsigma P_{\mathcal{J}} \tilde{K}_\varsigma^T \tilde{C}_\varsigma \tilde{F}_\varsigma - K & \tilde{F}_\varsigma^T K \tilde{B}_\varsigma + \tilde{F}_\varsigma^T \tilde{C}_\varsigma^T \tilde{K}_\varsigma P_{\mathcal{J}} \tilde{K}_\varsigma^T \tilde{C}_\varsigma \tilde{B}_\varsigma \\ \tilde{B}_\varsigma^T K \tilde{F}_\varsigma + \tilde{B}_\varsigma^T \tilde{C}_\varsigma^T \tilde{K}_\varsigma P_{\mathcal{J}} \tilde{K}_\varsigma^T \tilde{C}_\varsigma \tilde{F}_\varsigma & \tilde{B}_\varsigma^T K \tilde{B}_\varsigma + \tilde{B}_\varsigma^T \tilde{C}_\varsigma^T \tilde{K}_\varsigma P_{\mathcal{J}} \tilde{K}_\varsigma^T \tilde{C}_\varsigma \tilde{B}_\varsigma - \gamma^2 P_\varsigma \end{pmatrix} < 0.$$

Then decompose it as

$$\begin{pmatrix} -K & 0 \\ 0 & -\gamma^2 P_\varsigma \end{pmatrix} + \begin{pmatrix} \tilde{F}_\varsigma^T & \tilde{F}_\varsigma^T \tilde{C}_\varsigma^T \tilde{K}_\varsigma \\ \tilde{B}_\varsigma^T & \tilde{B}_\varsigma^T \tilde{C}_\varsigma^T \tilde{K}_\varsigma \end{pmatrix} \begin{pmatrix} K & 0 \\ 0 & P_{\mathcal{J}} \end{pmatrix} \begin{pmatrix} \tilde{F}_\varsigma & \tilde{B}_\varsigma \\ \tilde{K}_\varsigma^T \tilde{C}_\varsigma \tilde{F}_\varsigma & \tilde{K}_\varsigma^T \tilde{C}_\varsigma \tilde{B}_\varsigma \end{pmatrix} < 0, \tag{4.28}$$

introduce new matrices

$$\tilde{J}_\varsigma = \tilde{K}_\varsigma^T \tilde{C}_\varsigma \tilde{F}_\varsigma = \begin{pmatrix} (I - K^{H_\infty} H)F \\ ({K^{H_\infty}}^T - {K^{H_\infty}}^T K^{H_\infty} H)F \end{pmatrix}, \tag{4.29}$$

$$\tilde{L}_\varsigma = \tilde{K}_\varsigma^T \tilde{C}_\varsigma \tilde{B}_\varsigma = \begin{pmatrix} (I - K^{H_\infty} H)B & -K^{H_\infty} \\ ({K^{H_\infty}}^T - {K^{H_\infty}}^T K^{H_\infty} H)B & -{K^{H_\infty}}^T K^{H_\infty} \end{pmatrix}, \tag{4.30}$$

and replace (4.29) and (4.30) into (4.28) as

$$\begin{pmatrix} -K & 0 \\ 0 & -\gamma^2 P_\varsigma \end{pmatrix} + \begin{pmatrix} \tilde{F}_\varsigma^T & \tilde{J}_\varsigma^T \\ \tilde{B}_\varsigma^T & \tilde{L}_\varsigma^T \end{pmatrix} \begin{pmatrix} K & 0 \\ 0 & P_{\mathcal{J}} \end{pmatrix} \begin{pmatrix} \tilde{F}_\varsigma & \tilde{B}_\varsigma \\ \tilde{J}_\varsigma & \tilde{L}_\varsigma \end{pmatrix} < 0,$$

consider it as a Schur's complement, and represent with another inequality

$$\begin{pmatrix} -K & 0 & \tilde{F}_\varsigma^T & \tilde{J}_\varsigma^T \\ 0 & -\gamma^2 P_\varsigma & \tilde{B}_\varsigma^T & \tilde{L}_\varsigma^T \\ \tilde{F}_\varsigma & \tilde{B}_\varsigma & -K^{-1} & 0 \\ \tilde{J}_\varsigma & \tilde{L}_\varsigma & 0 & -P_{\mathcal{J}}^{-1} \end{pmatrix} < 0.$$

Now multiply the inequality from the left-hand and right-hand sides with the following matrices, respectively,

$$\begin{pmatrix} 0 & 0 & I & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \end{pmatrix}, \qquad \begin{pmatrix} 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \end{pmatrix}$$

and obtain

$$\begin{pmatrix} -K^{-1} & \tilde{F}_\varsigma & \tilde{B}_\varsigma & 0 \\ \tilde{F}_\varsigma^T & -K & 0 & \tilde{J}_\varsigma^T \\ \tilde{B}_\varsigma^T & 0 & -\gamma^2 P_\varsigma & \tilde{L}_\varsigma^T \\ 0 & \tilde{J}_\varsigma & \tilde{L}_\varsigma & -P_{\mathcal{J}}^{-1} \end{pmatrix} < 0. \tag{4.31}$$

Inequality in (4.31) still nonlinear with respect to the bias correction gain K^{H_∞} due to the presence of the quadratic term ${K^{H_\infty}}^T K^{H_\infty}$ in $\tilde{J}_\varsigma$ and $\tilde{L}_\varsigma$ defined in (4.29)-(4.30) respectively. To solve this problem, introduce an auxiliary matrix $\mathcal{Z}$ such that

$$\mathcal{Z} > {K^{H_\infty}}^T K^{H_\infty}.$$

Represent it as

$$\mathcal{Z} - {K^{H_\infty}}^T K^{H_\infty} > 0.$$

If the Schur complement is used, the last inequality can be equivalently replaced with the LMI as:

$$\begin{pmatrix} \mathcal{Z} & {K^{H_\infty}}^T \\ K^{H_\infty} & I \end{pmatrix} > 0. \tag{4.32}$$

Now. let's redefine $\tilde{J}_\varsigma$ in (4.29) and $\tilde{L}_\varsigma$ in (4.30) using the new variable $\mathcal{Z}$ to replace the quadratic terms,

$$\tilde{J}_\varsigma = \begin{pmatrix} (I - K^{H_\infty} H)F \\ ({K^{H_\infty}}^T - \mathcal{Z}H)F \end{pmatrix}, \tag{4.33}$$

$$\tilde{L}_\varsigma = \begin{pmatrix} (I - K^{H_\infty} H)B & -K^{H_\infty} \\ ({K^{H_\infty}}^T - \mathcal{Z}H)B & -\mathcal{Z} \end{pmatrix}, \tag{4.34}$$

Also, inequality in (4.31) exists another nonlinearity which is the inversion of the positive definite matrix K. To avoid the inversion of K, pre- and post-multiply the matrix (4.31) with

$$\begin{pmatrix} K & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}$$

and obtain another form of the inequality in (4.31)

$$\begin{pmatrix} -K & K\tilde{F}_\varsigma & K\tilde{B}_\varsigma & 0 \\ \tilde{F}_\varsigma^T K & -K & 0 & \tilde{J}_\varsigma^T \\ \tilde{B}_\varsigma^T K & 0 & -\gamma^2 P_\varsigma & \tilde{L}_\varsigma^T \\ 0 & \tilde{J}_\varsigma & \tilde{L}_\varsigma & -P_{\mathcal{J}}^{-1} \end{pmatrix} < 0. \tag{4.35}$$

As is desired to have only linearities in (4.35), K and K^{H_∞} can't be variables of the LMI at the same time, so the minimization problem has to be solved in two steps: The first one is to compute a value for K using a constant K^{H_∞} and the second one is to solve the minimization problem using the value of K found in step one as a constant. The best candidate for initializing the minimization procedure is of course the UFIR filter bias correction gain $K_k^{\mathrm{U}} = \mathcal{G}_k H^T$.

Before computing a value for K, a initial value for P_ε^- is needed. In stationary mode, $k = \infty$, (4.23) becomes the Lyapunov equation

$$F(I - K^{H_\infty} H)P_\varepsilon^-(I - K^{H_\infty} H)^T F - P_\varepsilon^- + FK^{H_\infty} R_k {K^{H_\infty}}^T F^T + BQ_k B^T = 0, \tag{4.36}$$

the initial value of P_ε^- can be obtained using the initial bias correction gain K^{H_∞} by solving the discrete Lyapunov equation (DLE) in (4.36).

Now with a numerical value of P_ε^-, a prior value for K can be obtained by solving the following minimization problem with a constant K^{H_∞}

$$\begin{gathered} K = \min_{K,\gamma^2} \gamma^2 \\ \text{subject to (4.35).} \end{gathered} \tag{4.37}$$

Using a constant value of K, the recursive H_∞ filter bias correction gain K^{H_∞} can be obtained by solving the following minimization problem

$$\begin{gathered} K^{H_\infty} = \min_{K^{H_\infty},\mathcal{Z},\gamma^2} \gamma^2 \\ \text{subject to (4.32), (4.35) and } \mathcal{Z} = {K^{H_\infty}}^T K^{H_\infty}. \end{gathered} \tag{4.38}$$

The Algorithm 4 was designed for two things, the first one is to satisfy the third constraint in the minimization problem in (4.38), and the second one to update the values of K used as constant in (4.38) which will be updated using (4.37). In the algorithm, γ is minimized in each iteration by increasing the trace of $\mathcal{Z}$, at the end of each iteration the trace of $\mathcal{Z}$ is compared with the trace of ${K^{H_\infty}}^T K^{H_\infty}$, if the difference between them is greater than a small threshold $\delta_0 > 0$, the routine is ended and the gain is obtained, if not, a new value of P_ε^- and K is computed using the last value of K^{H_∞}, and the routine is repeated until the recursive H_∞ filter bias correction gain is obtained.

Algorithm 4: Algorithm for recursive H_∞ Filter Bias Correction Gain Computation

Data: δ_0, K_k^{U}, P_ς

Result: K^{H_∞}

begin
- $K^{H_\infty} = K_k^{\mathrm{U}}$; `/* Initialize with UFIR bias correction gain */`
- $\mathcal{Z} = {K^{\mathrm{U}}}^T K^{\mathrm{U}}$;
- **while** $|\mathrm{tr}(\mathcal{Z}) - \mathrm{tr}({K^{H_\infty}}^T K^{H_\infty})| < \delta_0$ **do**
  - $P_\varepsilon^- \Leftarrow$ Solution of the DLE (4.36) ; `/* a priori error covariance */`
  - $K = \min \gamma^2$ subject to (4.35) ; `/* Update K */`
  - $\mathcal{Z}_{\mathrm{prev}} = \mathcal{Z}$;
  - $K^{H_\infty} = \min \gamma^2$ subject to (4.32), (4.35) and $\mathrm{tr}(\mathcal{Z}) > \mathrm{tr}(\mathcal{Z}_{\mathrm{prev}})$;
- **end**

end

Using the bias correction gain K^{H_∞}, numerically determined by using Algorithm 4, the *a posteriori* recursive H_∞ filtering estimate and error covariance for uncorrelated w_k, v_k, and x_m can be obtained as Kalman-like recursions. A pseudo code of the recursions for H_∞ filtering is listed in Algorithm 5.

Algorithm 5: The *a posteriori* Recursive H_∞ Filter Algorithm

Data: $y_k, \hat{x}_0, P_0, Q_k, R_k, K^{H_\infty}$

Result: $\hat{x}_k, P_k$

begin

  for $k = 1, 2, \ldots$ **do**

    $\hat{x}_k^- = F\hat{x}_{k-1}$;

    $P_k^- = FP_{k-1}F^T + BQ_kB^T$;

    $s_k = y_k - H\hat{x}_k^-$;

    $\hat{x}_k = \hat{x}_k + K^{H_\infty}s_k$;

    $P_k = (I - K^{H_\infty}H)P_k^-(I - K^{H_\infty}H)^T + K^{H_\infty}R_kK^{H_\infty T}$;

  end

end

4.3 Filter Tuning Example

Consider the tracking problem described in section 2.5, but w_k is considered CPN, and it can be viewed as a Gauss-Markov process

$$w_k = \Theta w_{k-1} + \mu_k, \tag{4.39}$$

where Θ is a color factor and $\mu_k \sim N(0, \sigma_\mu^2)$.

Also, consider v_k as colored measurement noise (CMN)

$$v_k = \Psi v_{k-1} + \xi_k, \tag{4.40}$$

where Ψ is a color factor and $\xi_k \sim N(0, \sigma_\xi^2)$.

The process of the state space (2.65)-(2.66) generated with the CPN and the CMN with the standard deviations $\sigma_\mu = 12$ m/s and $\sigma_\xi = 10$ m, the color factors $\Theta = 0.2$ and $\Psi = 0.4$ and the sample period $\tau = 0.025$ s is shown in Fig. 4.1.

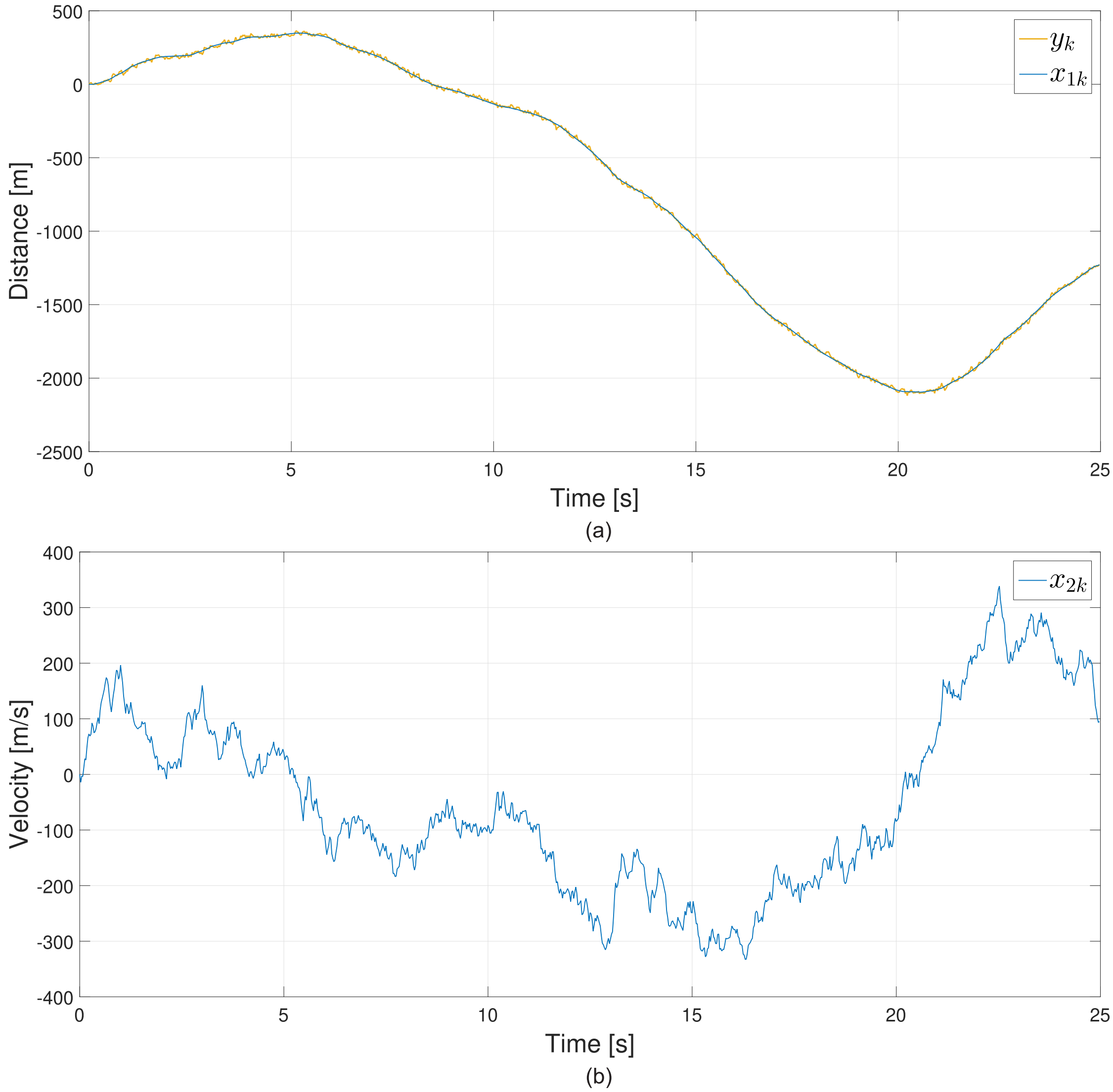


Figure 4.1: Process of the radar system generated, (a) first state measurements and first state without noise, (b) second state.

The process noise covariance $Q_k = \mathcal{E}\{w_k w_k^T\}$ is defined as

$$Q_k = \Theta Q_k \Theta^T + Q_\mu, \tag{4.41}$$

where $Q_\mu = \mathcal{E}\{\mu_k \mu_k^T\}$. The process noise covariance Q_k could be computed numerically as the solution of (4.41) written as a DLE

$$\Theta Q_k \Theta^T - Q_k + Q_\mu = 0. \tag{4.42}$$

Arguing similarly, the measurement noise covariance R_k could be obtained as the solution of the DLE

$$\Psi R_k \Psi^T - R_k + R_\xi = 0, \tag{4.43}$$

where $R_\xi = \mathcal{E}\{\xi_k \xi_k^T\}$.

To test the robustness of the filter, the noise covariances Q_k and R_k were tuned using $\Theta = \Theta_{\max} = 0.95$ and $\Psi = \Psi_{\max} = 0.95$. The recursive H_∞ filter bias correction gain were computed using Algorithm 4 with a small threshold $\delta_0 = 0.001$, then the *a posteriori* estimation is obtained using Algorithm 5, also estimations using the recursive UFIR filter and the KF [38] are obtained as benchmark.

The behavioral of Algorithm 4 could be seen graphically on Fig. 4.2. Fig. 4.2(a) shows the minimization of γ as the trace of $\mathcal{Z}$ is increased, as can be seen, if the trace of $\mathcal{Z}$ is increased, γ takes lower values as it gets to a minimum, but Fig. 4.2(b) shows that minimum value of γ is not necessary the minimum that we are searching for. Fig. 4.2(b) shows the comparison of the trace of $\mathcal{Z}$ and the trace of ${K^{H_\infty}}^T K^{H_\infty}$ as the trace of $\mathcal{Z}$ is increased, in this case both graphs are very similar with lower values of the trace of $\mathcal{Z}$ but there is a point where this trace is increased and the other one diverges, this is because at this point of the algorithm $\mathcal{Z} \neq {K^{H_\infty}}^T K^{H_\infty}$ and the third restriction in (4.38) isn't satisfied, so we have to stop the algorithm before both traces starts to diverge and obtain the recursive H_∞ filter bias correction gain.

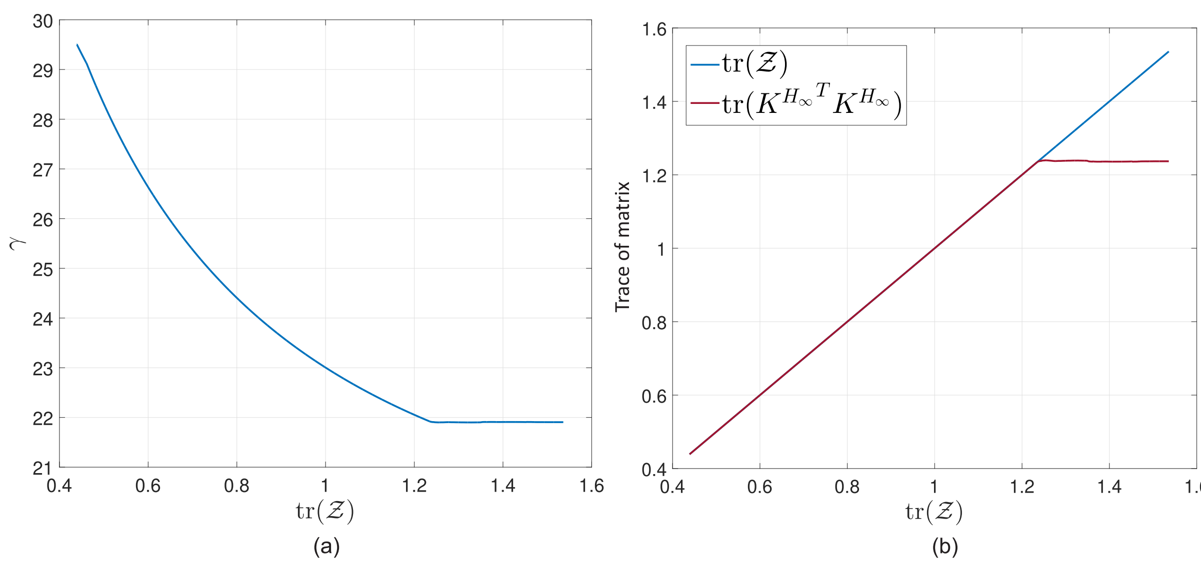


Figure 4.2: Solving the minimization problem using Algorithm 4: (a) minimizing γ as the trace of $\mathcal{Z}$ is increased and (b) comparison between the trace of matrix ${K^{H_\infty}}^T K^{H_\infty}$ and the trace of $\mathcal{Z}$ as the trace of $\mathcal{Z}$ is increased.

Typical filtering errors are shown in Fig. 4.3, the UFIR filter estimate gives larger errors than any other estimation, when there is a presence of non-Gaussian noises, KF isn't the best option, so its desired to have an estimator who gives better estimations than Kalman's, in this case, the recursive H_∞ filter gives better estimation as is shown in Table 4.1.

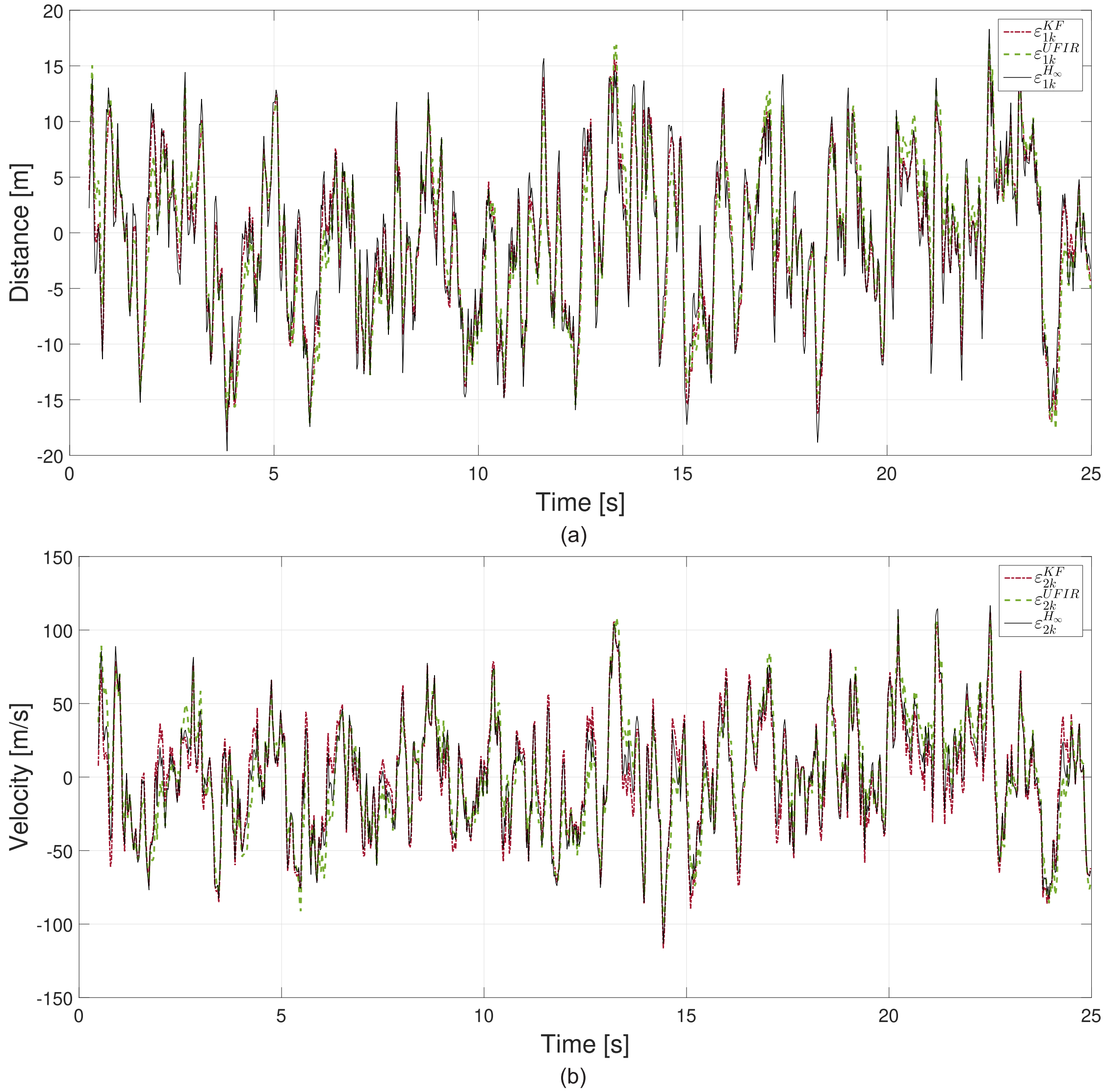


Figure 4.3: Filtering errors produced by the filters in the example system for: (a) first state and (b) second state.

Table 4.1: RMSEs produced by the filters.

Filter	RMSE
KF	38.3775
UFIR (Recursive)	39.6717
Recursive $\mathbf{H_\infty}$	**38.1486**

Chapter 5

Iterative Algorithm for Recursive H_∞ Filter Bias Correction Gain Computation

The state-space equations in (4.19)-(4.20) can't be used directly on the inequality given by lemma 2 due to weight P_y in (2.31) corresponds to P_ε in the recursive H_∞ filtering problem and $P_\varepsilon = P_k = \mathcal{E}\{\varepsilon_k \varepsilon_k^T\}$ is function of the bias correction gain K^{H_∞}, and for that reason, the inequality is nonlinear with respect to the bias correction gain K^{H_∞}. A solution for nonlinearities was proposed where the inequality in (2.31) is substituted with P_ε as function of K^{H_∞}, this solution gives an inequality where the product ${K^{H_\infty}}^T K^{H_\infty}$ makes nonlinear with respect to the bias correction gain K^{H_∞} but introducing a new variable $\mathcal{Z}$ it can be transformed into a LMI by adding the inequality $\mathcal{Z} > {K^{H_\infty}}^T K^{H_\infty}$, then by increasing the trace of $\mathcal{Z}$ and checking if the constraint $\mathrm{tr}(\mathcal{Z}) = \mathrm{tr}({K^{H_\infty}}^T K^{H_\infty})$ is satisfied, the recursive H_∞ filter bias correction gain K^{H_∞} is obtained (Algorithm 4). Another alternative to avoid the nonlinearity in the inequality in Lemma 2 is the iterative algorithm presented bellow.

5.1 Generalized Noise Power Gain in Recursive Form

If the UFIR filter gain $\mathcal{H}_N = F^{N-1}(H_N^T H_N)^{-1} H_N^T$ is used, the GNPG $\mathcal{G}_k$ can be written as

$$\begin{aligned}
\mathcal{G}_k &= F^{N-1}(H_N^T H_N)^{-1} H_N^T H_N (H_N^T H_N)^{-T} (F^{N-1})^T \\
&= F^{N-1}(H_N^T H_N)^{-1} (F^{N-1})^T, \qquad (5.1)
\end{aligned}$$

then, a convolution based estimation can be obtained as

$$\begin{aligned}\hat{x}_k &= \mathcal{H}_N Y_{m,k} \\ &= F^{N-1}(H_N^T H_N)^{-1} H_N^T Y_{m,k} \\ &= F^{N-1}(H_N^T H_N)^{-1}(F^{N-1})^T (F^{N-1})^{-T} H_N^T Y_{m,k} \\ &= \mathcal{G}_k (F^{N-1})^{-T} H_N^T Y_{m,k},\end{aligned} \tag{5.2}$$

and, by applying the decomposition

$$H_N = \begin{pmatrix} H_{N-1} \\ HF^{N-1}, \end{pmatrix} \tag{5.3}$$

the recursion for the GNPG (5.1) can be obtained if $\mathcal{G}_k^{-1}$ is transformed as

$$\begin{aligned}\mathcal{G}_k^{-1} &= (F^{N-1})^{-T} H_N^T H_N F^{-(N-1)} \\ &= (F^{N-1})^{-T} \begin{pmatrix} H_{N-1}^T & (F^{N-1})^T H^T \end{pmatrix} \begin{pmatrix} H_{N-1} \\ HF^{N-1}, \end{pmatrix} F^{-(N-1)} \\ &= (F^{N-1})^{-T} (H_{N-1}^T H_{N-1} + (F^{N-1})^T H^T H F^{N-1}) F^{-(N-1)} \\ &= (F^{N-1})^{-T} H_{N-1}^T H_{N-1} F^{-(N-1)} + H^T H \\ &= F^{-T} (F^{N-2})^{-T} H_{N-1}^T H_{N-1} F^{-(N-2)} F^{-1} + H^T H \\ &= F^{-T} \mathcal{G}_{k-1}^{-1} F^{-1} + H^T H.\end{aligned}$$

This gives the recursive form of $\mathcal{G}_{k-1}^{-1}$:

$$\mathcal{G}_{k-1}^{-1} = F^T (\mathcal{G}_k^{-1} - H^T H) F. \tag{5.4}$$

Using (5.3), (5.4) and $H_N^T = (F^{N-1})^T \mathcal{G}_k^{-1} \mathcal{H}_N$ from the UFIR filter gain, the product $H_N^T Y_{m,k}$

in (5.2) can be transformed as

$$
\begin{aligned}
H_N^T Y_{m,k} &= \begin{pmatrix} H_{N-1}^T & (F^{N-1})^T H^T \end{pmatrix} \begin{pmatrix} Y_{m,k-1} \\ y_k \end{pmatrix} \\
&= H_{N-1}^T Y_{m,k-1} + (F^{N-1})^T H^T y_k \\
&= (F^{N-2})^T \mathcal{G}_{k-1}^{-1} \mathcal{H}_{N-1} Y_{m,k-1} + (F^{N-1})^T H^T y_k \\
&= (F^{N-2})^T \mathcal{G}_{k-1}^{-1} \hat{x}_{k-1} + (F^{N-1})^T H^T y_k \\
&= (F^{N-2})^T F^T (\mathcal{G}_k^{-1} - H^T H) F \hat{x}_{k-1} + (F^{N-1})^T H^T y_k \\
&= (F^{N-1})^T (\mathcal{G}_k^{-1} - H^T H) \hat{x}_k^- + (F^{N-1})^T H^T y_k \\
&= (F^{N-1})^T (\mathcal{G}_k^{-1} \hat{x}_k^- - H^T H \hat{x}_k^- + H^T y_k) \\
&= (F^{N-1})^T (\mathcal{G}_k^{-1} \hat{x}_k^- - H^T (y_k + H \hat{x}_k^-)) \\
&= (F^{N-1})^T (\mathcal{G}_k^{-1} \hat{x}_k^- - H^T s_k).
\end{aligned} \tag{5.5}
$$

By replacing (5.5) into (5.2), the following recursion is obtained for the estimate

$$
\begin{aligned}
\hat{x}_k &= \mathcal{G}_k (F^{N-1})^{-T} (F^{N-1})^T (\mathcal{G}_k^{-1} \hat{x}_k - H^T s_k) \\
&= \mathcal{G}_k (\mathcal{G}_k^{-1} \hat{x}_k^- - H^T s_k) \\
&= \hat{x}_k^- - \mathcal{G}_k H^T s_k.
\end{aligned} \tag{5.6}
$$

Analyzing (5.6) as Kalman recursions, it can be say that the bias correction gain K_k can be related to the GNPG $\mathcal{G}_k$ by the following relationship:

$$
K_k = \mathcal{G}_k H^T. \tag{5.7}
$$

5.2 Iterative Algorithm for Recursive H_∞ Filter Bias Correction Gain Computation

First, take the inequality in (2.37) and replace F, B, H, P_w and P_y with $\tilde{F}_\varsigma$, $\tilde{B}_\varsigma$, $\tilde{C}_\varsigma$, P_ς and P_ε, respectively,

$$
\begin{pmatrix}
-K^{-1} & \tilde{F}_\varsigma & \tilde{B}_\varsigma & 0 \\
\tilde{F}_\varsigma^T & -K & 0 & \tilde{F}_\varsigma^T \tilde{C}_\varsigma^T \\
\tilde{B}_\varsigma^T & 0 & -\gamma^2 P_\varsigma & \tilde{B}_\varsigma^T \tilde{C}_\varsigma^T \\
0 & \tilde{C}_\varsigma \tilde{F}_\varsigma & \tilde{C}_\varsigma \tilde{B}_\varsigma & -P_\varepsilon^{-1}
\end{pmatrix} < 0. \tag{5.8}
$$

Inequality in (5.8) is nonlinear with respect to the symmetric positive-definite matrix K, to avoid the inversion of K, pre- and post-multiply the matrix (3.3) with

$$\begin{pmatrix} K & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}$$

and obtain another form for the inequality,

$$\begin{pmatrix} -K & K\tilde{F}_\varsigma & K\tilde{B}_\varsigma & 0 \\ \tilde{F}_\varsigma^T K & -K & 0 & \tilde{F}_\varsigma^T \tilde{C}_\varsigma^T \\ \tilde{B}_\varsigma^T K & 0 & -\gamma^2 P_\varsigma & \tilde{B}_\varsigma^T \tilde{C}_\varsigma^T \\ 0 & \tilde{C}_\varsigma \tilde{F}_\varsigma & \tilde{C}_\varsigma \tilde{B}_\varsigma & -P_\varepsilon^{-1} \end{pmatrix} < 0. \tag{5.9}$$

If P_ε is considered as constant then the inequality in (5.9) could be considered as a LMI. The weight matrix P_ε could be constant if before solving the LMI, a previous value of P_ε is obtained, to compute that value first write $P_\varepsilon = P_k = \mathcal{E}\{\varepsilon_k \varepsilon_k^T\}$ in terms of the state-space model (4.15)-(4.16) as

$$P_k = \tilde{F}_\varsigma P_{k-1} \tilde{F}_\varsigma^T + \tilde{B}_\varsigma P_\varsigma \tilde{B}_\varsigma^T, \tag{5.10}$$

where $P_\varsigma = \mathcal{E}\{\varsigma_k \varsigma_k^T\}$ is given by (2.54).

The statement $P_\varepsilon = P_k = P_{k-1}$ is true if the stationary mode $k = \infty$ is considered, then (5.10) can be transformed in the Lyapunov equation

$$\tilde{F}_\varsigma P_\varepsilon \tilde{F}_\varsigma^T - P_\varepsilon + \tilde{B}_\varsigma P_\varsigma \tilde{B}_\varsigma^T = 0. \tag{5.11}$$

A value for P_ε can be computed by solving the DLE defined in (5.11).

Then, define P_{prev} as the value of P_ε computed before solving the minimization problem in which, the bias correction gain that could be used for computing P_{prev} at the first time is the UFIR bias correction gain $K_k^{\text{U}} = \mathcal{G}_k H^T$. At this time P_{prev} is a constant so if P_ε is replaced with P_{prev}, (5.9) could be represented as the inequality

$$\begin{pmatrix} -K & K\tilde{F}_\varsigma & K\tilde{B}_\varsigma & 0 \\ \tilde{F}_\varsigma^T K & -K & 0 & \tilde{F}_\varsigma^T \tilde{C}_\varsigma^T \\ \tilde{B}_\varsigma^T K & 0 & -\gamma^2 P_\varsigma & \tilde{B}_\varsigma^T \tilde{C}_\varsigma^T \\ 0 & \tilde{C}_\varsigma \tilde{F}_\varsigma & \tilde{C}_\varsigma \tilde{B}_\varsigma & -P_{\text{prev}}^{-1} \end{pmatrix} < 0. \tag{5.12}$$

As is desired to have only linearities in (5.12), K and K^{H_∞} can't be variables of the LMI at the same time, so the minimization problem has to be solved in two steps: The first one is to compute a value for K using a constant K^{H_∞} and the second one is to solve the minimization problem using the value of K found in step one as a constant.

A prior value for K can be obtained by solving the following minimization problem with a constant K^{H_∞}

$$K = \min_{K,\gamma^2} \gamma^2$$
$$\text{subject to (5.12).} \tag{5.13}$$

After solving the LMI in (5.12) with a constant value for K, a value for K^{H_∞} will be obtained which will be used for computing a new P_ε solving the DLE (5.11), whose trace is expected to be lower than the trace of P_{prev}. One way to guarantee that the trace of P_ε will be lower then the trace of P_{prev} is adding more constraints to the minimization problem. To do this, first define the GNPG before solving the LMI as $\mathcal{G}_{\text{prev}}$, then define a new variable $\mathcal{G}$ such that

$$\mathcal{G}H^T > K^{H_\infty}.$$

Now rewrite it as the LMI

$$\mathcal{G}H^T - K^{H_\infty} > 0. \tag{5.14}$$

A important property of the GNPG is that if the trace of $\mathcal{G}$ is maximized, then the value of the trace of the error covariance P_k will be minimized. Therefore, if the trace of $\mathcal{G}$ is greater than the trace of $\mathcal{G}_{\text{prev}}$, then the trace of P_ε will be lower than the trace of P_{prev}. Hence, the bias correction gain K^{H_∞} that will be used for computing P_ε could be found numerically by solving the following minimization problem iteratively,

$$K^{H_\infty} = \min_{K^{H_\infty},\mathcal{G},\gamma^2} \gamma^2$$
$$\text{subject to (5.12), (5.14) and } \operatorname{tr}(\mathcal{G}) > \operatorname{tr}(\mathcal{G}_{\text{prev}}). \tag{5.15}$$

Using the bias correction gain K^{H_∞} computed after solving the minimization problem in (5.15), the trace of P_ε can be computed and compared with the trace of P_{prev} until the condition $\operatorname{tr}(P_\varepsilon) < \operatorname{tr}(P_{\text{prev}})$ does not fit.

A pseudo code of the recursive *a posteriori* iterative recursive H_∞ filter bias correction gain computstion is listed as Algorithm 6.

Algorithm 6: Iterative Algorithm for recursive H_∞ Filter Bias Correction Gain Computation

Data: $\hat{\mathcal{H}}_N$, P_ς

Result: K^{H_∞}

begin

$\mathcal{H}_N = \hat{\mathcal{H}}_N \hat{\mathcal{H}}_N^T H^T$; `/* Initialize with UFIR bias correction gain */`

$\mathcal{G} = \hat{\mathcal{H}}_N \hat{\mathcal{H}}_N^T$;

do

$P_{\text{prev}} \Leftarrow$ Solution of the DLE (5.11) ;

$K = \min \gamma^2$ subject to (5.12) ; `/* Update K */`

$\mathcal{G}_{\text{prev}} = \mathcal{G}$;

$K^{H_\infty} = \min \gamma^2$ subject to (5.12), (5.14) and $\text{tr}(\mathcal{G}) > \text{tr}(\mathcal{G}_{\text{prev}})$;

$P_\varepsilon \Leftarrow$ Solution of the DLE (5.11) ;

while $\text{tr}(P_\varepsilon) < \text{tr}(P_{prev})$;

end

Using the bias correction gain K^{H_∞}, numerically determined by using Algorithm 6, the *a posteriori* recursive H_∞ filtering estimate and error covariance for uncorrelated w_k, v_k, and x_m can be obtained as Kalman-like recursions using Algorithm 5.

5.3 Numerical Example

Consider the tracking problem described in section 2.5, but for the disturbance w_k and the measurement noise v_k consider the following cases:

1. Considering both as Gauss-Markov.
2. Considering disturbance w_k as Gauss-Markov colored and measurement noise v_k as gaussian.
3. Considering disturbance w_k as gaussian and measurement noise v_k as Gauss-Markov.

For all cases, the recursive H_∞ filter bias correction gain will be computed using Algorithm 6, and also using Algorithm 4, the filtering errors of the recursive H_∞ filter will be compared against the filtering errors of the KF and recursive UFIR filter.

5.3.1 Gauss-Markov Disturbance and Measurement Noise

First, consider $w_k = \Theta w_{k-1} + \mu_k$ and $v_k \Psi v_{k-1} + \xi_k$, where $\mu_k \sim N(0, \sigma_\mu^2)$ and $\xi_k \sim N(0, \sigma_\xi^2)$, now extend the state space (2.65)-(2.66) on $[m, k]$, and compute the *a posteriori* recursive H_∞ filter bias correction gain numerically by using Algorithm 4 and Algorithm 6. The estimation can be computed using Algorithm 5.

The estimations were computed with the standard deviations $\sigma_\mu = 12$ m/s and $\sigma_\xi = 10$ m and with a sample period of $\tau = 0.025$ s, the color factors where selected as $\Theta = 0.2$ and $\Psi = 0.4$, using this parameters, the optimal horizon for the UFIR computation is $N_{\text{opt}} = 20$.

The behavioral of Algorithm 6 could be seen graphically on Fig. 5.1, in Fig. 5.1(a) is shown the minimization of γ as the trace of $\mathcal{G}$ is increased, while Fig. 5.1(b) shows how the trace of P_ε is minimized as the trace of $\mathcal{G}$ is increased, as can be seen, the algorithm works while the trace of P_ε is minimized, if the trace starts to grew, then the algorithm is ended and the recursive H_∞ filter bias correction gain is obtained.

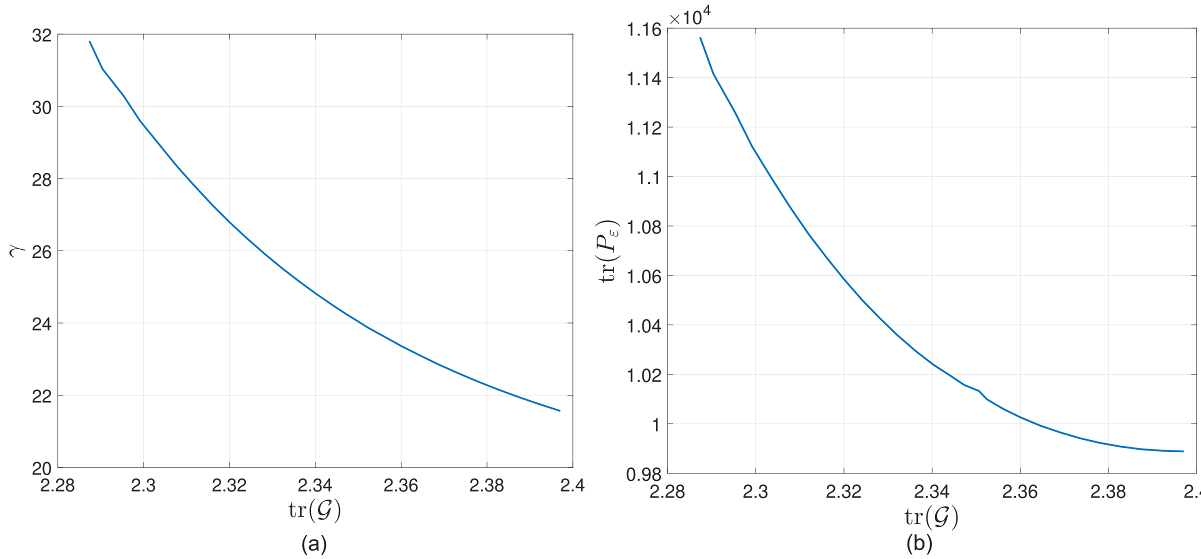


Figure 5.1: Solving the minimization problem using Algorithm 6: (a) minimizing γ as function of the trace of $\mathcal{G}$ and (b) the trace of error covariance P_ε as function of the trace of $\mathcal{G}$.

The RMSE for each filter is given in Table 5.1, in this case the process has lower values for color factor and it is expected that KF give the most accurate estimations, but the recursive H_∞ filter was the one with the lower errors.

Table 5.1: RMSEs produced by the filters.

Filter	RMSE
KF	38.3775
Recursive UFIR	39.6717
Recursive H_∞ FIR (Alg. 4)	38.1486
Recursive H_∞ FIR (Alg. 6)	38.3265

5.3.2 Colored Gauss-Markov Disturbance

Consider the vehicle tracking problem described in (2.65)-(2.66), where the white Gaussian measurement noise $v_k \sim \mathcal{N}(0, \sigma_v^2)$ has the standard deviation $\sigma_v^2 = 10$ m/s, for this case the vehicle trajectory is affected by the Gauss-Markov process disturbance $w_k = \Theta w_{k-1} + \mu_k$, where the scalar disturbance factor Θ is chosen as $0 < \Theta < 1$, and the driving noise $\mu_k \sim \mathcal{N}(0, \sigma_\mu^2)$ has a standard deviation $\sigma_\mu = 12$ m/s.

To compare the estimation errors, the disturbance process is generated by changing Θ from 0.05 to 0.95 with a step 0.05. Next, the following scenarios of filter optimal tuning are considered:

1. For Θ.
2. For $\Theta = 0.05$.
3. For $\Theta = 0.95$.

Typical tracking RMSEs are shown in Fig. 5.2 as functions of Θ, and it can be stated the following features:

- *Case 1 (theoretical): Tuning for* Θ. When the filters are optimally tuned for Θ, their RMSEs reach the lowest possible values, as shown in Fig. 5.2(a). The recursive H_∞ (Algorithms 4 and 6) filter and the recursive UFIR filter do it with a lower rate that speaks in favor of their higher robustness. Note that filter tuning to each Θ is hardly possible in practice, so this case can be considered theoretical.

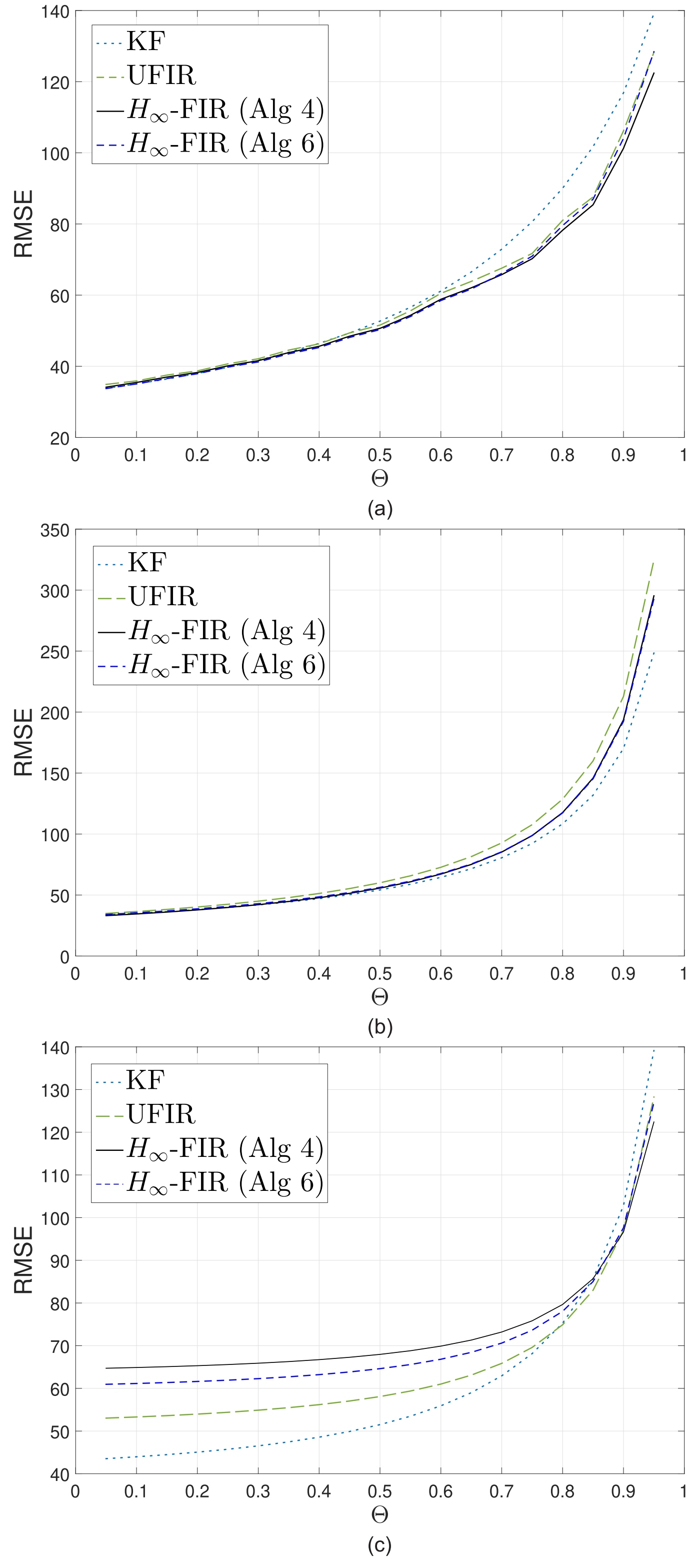


Figure 5.2: Typical RMSEs generated by the filters as functions of the process color factor $0.05 \leqslant \Theta \leqslant 0.95$ in different scenarios of tuning: (a) *theoretical:* tuning to each Θ, (b) *regular:* tuning to $\Theta = 0.05$, and (c) *robust:* tuning to $\Theta = 0.95$

- *Case 2 (regular): Tuning for* $\Theta = 0.05$. When the disturbance is not specified, all filters are usually tuned near to white noise this is the reason this is considered the regular case. Increasing Θ causes all errors to increase at a high rate. Accordingly, all filters produce consistent and large errors when Θ reaches 0.95, as shown in Fig. 5.2(b).

- *Case 3 (robust): Tuning for* $\Theta = 0.95$. When the disturbance boundary is known, all filter can be tuned for $\Theta = 0.95$. This drastically lowers the RMSEs in all filters compared to tuning for $\Theta = 0.05$ (Fig. 5.2(b)). It can be seen that all filters demonstrate a better robustness. However, the highest robustness is exhibited by the recursive H_∞ filter (Algorithms 4 and 6) and the recursive UFIR filter. Among these filters, as is shown in Fig. 5.2(c), the recursive H_∞ filter (Alg. 4) looks a bit more robust.

Comparing the RMSEs shown in Fig. 5.2(a) and Fig. 5.2(c), it can be concluded that tuning for each Θ gives the smallest errors, but can hardly be implemented practically. On the contrary, tuning for $\Theta = 0.95$ gives slightly more errors, but this case is feasible and robust. When the recursive H_∞ filter is tuned near to white noise (Fig. 5.2(b)), its behavioral is the same like the other filters, increasing the errors as the color factor is increased. By the other hand, if the H_∞ is tuned to the maximized disturbance (Fig. 5.2(c)) then the errors doesn't have a significant difference if the disturbance is decreased, which means robustness.

Analyzing the robustness ϱ of the filters tuned for the robust case (Table 5.2), it can be seen that the recursive H_∞ filter using either Alg. 4 or 4 has more robustness than the recursive UFIR filter.

Table 5.2: Robustness ϱ of the filters tuned in robust mode (Gauss-Markov disturbance).

Filter	ϱ
KF	0.3127
UFIR	0.4135
Recursive H_∞ Alg. 4	0.5287
Recursive H_∞ Alg. 6	0.4874

It is worth noting that the robustness of the recursive filters are lower than the batch form filters, and also are less accurate, but the estimation computation takes considerably less time than in batch form.

5.3.3 Colored Gauss-Markov Measurement Noise

Consider the vehicle tracking problem with the white Gaussian disturbance $w_k \sim \mathcal{N}(0, \sigma_w^2)$ and the colored noise $v_k = \Psi v_{k-1} + \xi_k$, where $\xi \sim \mathcal{N}(0, \sigma_\xi^2)$ and the color factor $0 < \Psi < 1$ is chosen for stability. Using $\sigma_w = 10$ m/s, $\sigma_\xi = 12$ m/s, the same scenarios of tuning are considered:

1. For Ψ.
2. For $\Psi = 0.05$.
3. For $\Psi = 0.95$.

Fig. 5.3 shows the RMSEs produced by filters as functions of the colored noise factor Ψ. The theoretical case (tuning for Ψ and $N_{\text{opt}}(\Psi)$) is shown in Fig. 5.3(a), as in the colored disturbance example, all filters produce consistent errors that grow and at a low rate. The regular case (tuning for $\Psi = 0.05$) can be seen in Fig. 5.3(b) as in the colored disturbance example, increasing Ψ causes all errors to increase at a high rate. In Fig. 5.3(c) is shown the robust case (tuned for $\Psi = 0.95$), in this case, it can be seen that the recursive UFIR filter demonstrates more robustness than any other filter, but also the recursive H_∞ filter has higher robustness.

The recursive H_∞ filter (Algorithms 4 and 6) and the recursive UFIR filter demonstrate high robustness, although the H_∞ is more successful in accuracy. Analyzing the robustness ϱ of the filters tuned for the robust case (Table 5.3), it can be noticed that the recursive H_∞ filter has a high robustness considering that isn't a filter in batch form.

Table 5.3: Robustness ϱ of the filters tuned in robust mode (Gauss-Markov measurement noise).

Filter	ϱ
KF	0.4430
UFIR	0.8185
Recursive H_∞ Alg. 4	0.6373
Recursive H_∞ Alg. 6	0.6244

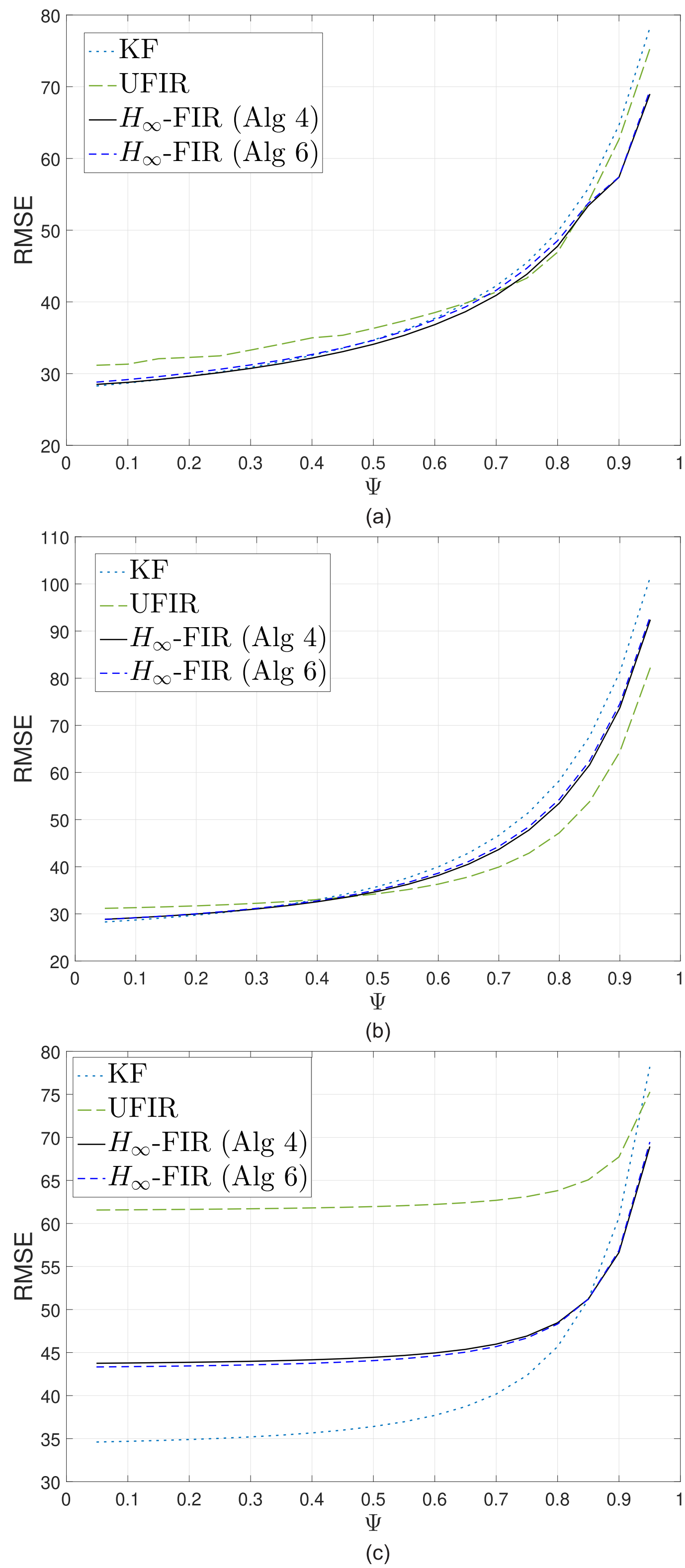
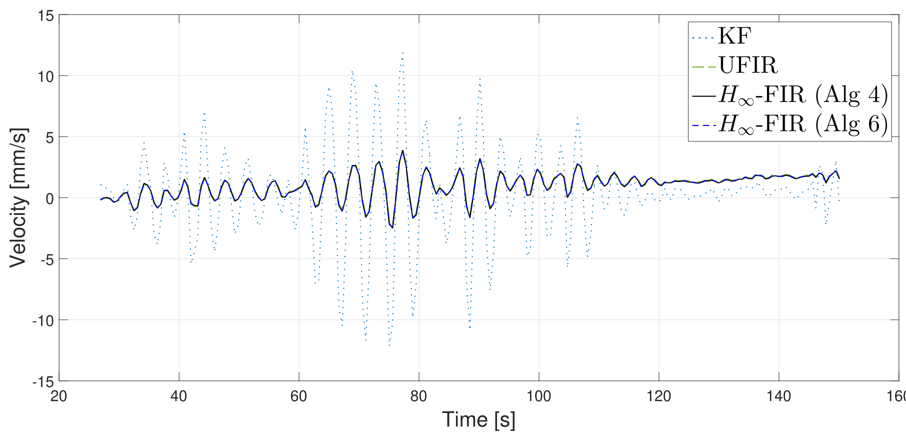


Figure 5.3: Typical RMSEs generated by the filters as functions of the colored measurement noise factor $0.05 \leqslant \psi \leqslant 0.95$ in different scenarios of tuning: (a) *theoretical*: tuning to each ψ, (b) *regular:* tuning to $\psi = 0.05$, and (c) *robust:* tuning to $\psi = 0.95$.

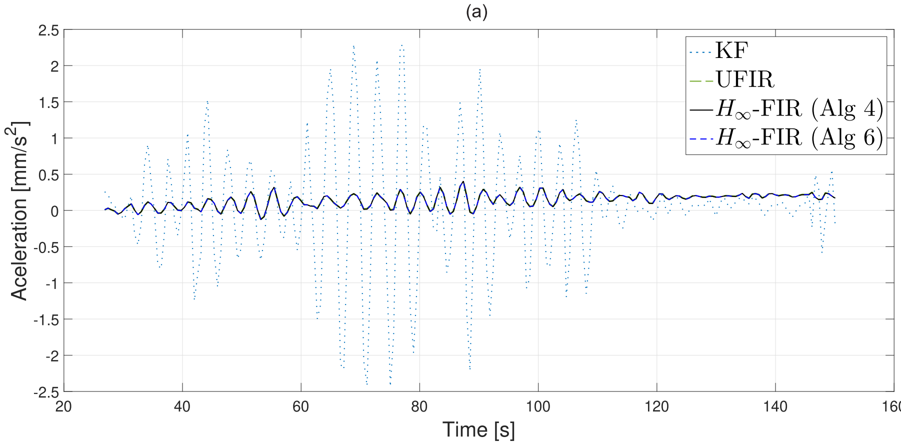
5.4 Experimental Verification

Now, the accuracy and robustness of the recursive H_∞ filter is tested using the same real data that was used to test the batch form of the H_∞ FIR filter in Section 3.4.

The filters were tuned using the using the state-space equations (3.12)-(3.13) and the values of Θ, σ_μ and σ_v computed in Section 3.4. The errors of each filter are shown in Fig. 5.4, as can be seen, in presence of a colored Gauss-Markov disturbance, the robust filters (recursive UFIR and recursive H_∞) has the most accurate estimations, Table 5.4 shows the RMSE produced by each filters, under colored disturbance KF doesn't give an accurate estimation, UFIR filter being a robust filter, gives a good estimation, the recursive H_∞ filter (computed with any algorithm) estimation is near to the UFIR estimation but a little more accurate.



(a)



(b)

Figure 5.4: Filtering errors in the first case for (a) first state and (b) second state

Table 5.4: RMSE produced by the filters in the experiment using real data.

Filter	RMSE
KF	3.9349
Recursive UFIR	1.4310
Recursive $\mathbf{H}_\infty$ Alg. 4	**1.4164**
Recursive H_∞ Alg. 6	1.4233

Conclusions

The H_∞ FIR filter was developed to suppress all peaks in the disturbance-to-error transfer function $\mathcal{T}$ by minimizing the H_∞ norm of $\mathcal{T}$, the H_∞ norm reflects the worst estimator case and its minimization results in a robust estimator. The cost function to minimize in the H_∞ FIR filter problem can't be minimized as a convex function so numerical approaches are needed.

In this work, a LMI-Based algorithm for computation of the H_∞ FIR filter gain was proposed, in which the quadratic constraints and nonlinearities in the variables of the inequality are avoid. In this case, the H_∞ FIR gain is obtained one time without initial parameters working as an unbiased FIR filter. A numerical example were provide to show how to tune the H_∞ FIR filter using the LMI-Based algorithm.

The LMI-Based algorithm requires considerable computational time when the size of the batch has a lot of discrete points, an iterative algorithm for computation of the H_∞ FIR filter gain was provided to reduce the computation time. The iterative algorithm is less accurate to compute the H_∞ FIR filter gain than the LMI-Based algorithm which means that it has a little more errors, but as it can compute the filter gain in less time, the user can choice which algorithm to use, if the one who gives better accurate estimations or the one who takes less computation time to find the filter gain.

The robustness of the H_∞ FIR filter was tested, where it was found that the filter is almost as robust as the UFIR filter, but giving more accurate estimations. Another remarkable result in this work is that, there isn't necessary to know the exact value of the disturbance and error covariances to have accurate estimation in H_∞ FIR filtering, another proof of the robustness of the filter.

The H_∞ FIR filter was tested against real data considered under Gauss-Markov disturbance. The H_∞ FIR filter estimation was the more accurate among estimations of KF, OFIR, OUFIR, ML-FIR, UFIR, H_2-OFIR and H_2-OUFIR filters.

The batch form of the H_∞ FIR filter is computationally time consuming, especially when

$N \gg 1$, due to large dimensions of all extended vectors and matrices. One solution to this is using recursive algorithms to compute the estimation. In this work was presented a method to obtain H_∞ filtering recursively using a Kalman-like algorithm.

The way of obtaining the bias correction gain of the recursive H_∞ filter is by using an algorithm in which all nonlinearities of the minimization problem are disappeared and the minimization problem can be solved using LMI. The recursive H_∞ filter is almost robust as the UFIR filter, and it can works with less error than KF if there are colored noises in the system.

If a faster computation of the recursive H_∞ filter bias correction gain is needed an iterative algorithm for computation of bias correction gain was provided, this new iterative algorithm gives a little more errors than the LMI-based algorithm, but it can compute the filter gain in less time.

The robustness of the recursive H_∞ filter was tested, where we can find that the filter is almost as robust as the UFIR filter. The H_∞ FIR filter in recursive mode has largest errors than the batch form, but it has almost the same robustness and it can compute the estimation takeing considerably less time.

The recursive H_∞ filter was tested against real data under Gauss-Markov disturbance, and its estimation has lower errors than the recursive UFIR filter and KF, which makes the recursive H_∞ filter a good alternative if is desired to use a recursive filter.

Bibliography

[1] C. K. Ahn, "Strictly passive FIR filtering for state-space models with external disturbance," *AEU - Int. J. of Electron. and Commun.*, vol. 66, no. 11, pp. 944–948, Nov. 2012.

[2] C. K. Ahn, Y. S. Shmaliy, S. Zhao, and H. Li, "Continuous-time deadbeat H_2 FIR filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 8, pp. 987–991, Aug. 2017.

[3] C. K. Ahn, S. Zhao, Y. S. Shmaliy, and H. Li, "On the ℓ_2-ℓ_∞ and H_∞ Performances of the Continuous-Time Deadbeat H_2 FIR Filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 11, pp. 1798–1802, Nov. 2018.

[4] R. Bellman, *Introduction to Matrix Analysis, Second Edition*. Soc. for Ind. and Appl. Math., Jan. 1997.

[5] M. Benosman, "Model-based vs data-driven adaptive control: An overview," *Int. J. of Adaptive Control and Signal Process.*, vol. 32, no. 5, pp. 753–776, Mar. 2018.

[6] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Soc. for Ind. and Appl. Math., Jan. 1994.

[7] B.-S. Chen and J.-C. Hung, "Fixed-order H_2 and H_∞ optimal deconvolution filter designs," *Signal Process.*, vol. 80, no. 2, pp. 311–331, Feb. 2000.

[8] J. Doyle, "Robust and optimal control," in *Proc. 35th IEEE Conference on Decision and Control*, IEEE, 1997.

[9] J. Doyle, K. Glover, P. Khargonekar, and B. Francis, "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Trans. Autom. Control*, vol. 34, no. 8, pp. 831–847, 1989.

[10] S. Fekri, M. Athans, and A. Pascoal, "Issues, progress and new results in robust adaptive control," *International Journal of Adaptive Control and Signal Processing*, vol. 20, no. 10, pp. 519–579, 2006.

[11] M. Fu, C. E. de Souza, and L. Xie, "H_∞ Estimation for uncertain systems," *Int. J. of Robust and Nonlinear Control*, vol. 2, no. 2, pp. 87–105, Aug. 1992.

[12] F. R. Gantmacher, J. L. Brenner, D. W. Bushaw, S. Evanusa, and P. M. Morse, "Applications of the theory of matrices," *Physics Today*, vol. 13, no. 10, pp. 56–56, Oct. 1960.

[13] E. Gershon, U. Shaked, and I. Yaesh, "H_∞ Control and Estimation of State-Multiplicative Linear Systems," *Berlin: Springer*, vol. 318, Sep. 2005.

[14] B. Hassibi, A. H. Sayed, and T. Kailath, *Indefinite-Quadratic Estimation and Control*. Soc. for Ind. and Appl. Math., Jan. 1999.

[15] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[16] B. Kwon, Z. Quan, and S. Han, "A robust fixed-lag receding horizon smoother for uncertain state space models," *Int. J. of Adaptive Control and Signal Process.*, vol. 29, no. 11, pp. 1354–1366, Feb. 2015.

[17] O. K. Kwon, W. H. Kwon, and K. S. Lee, "FIR filters and recursive forms for discrete-time state-space models," *Automatica*, vol. 25, no. 5, pp. 715–728, Sep. 1989.

[18] W. H. Kwon, *Receding horizon control, model predictive control for state models*. Springer, 2005, p. 380.

[19] Y. S. Lee, S. H. Han, and W. H. Kwon, "H_2/H_∞ FIR filters for discrete-time state space models," *Int. J. of Control Automation and Syst.*, vol. 4, no. 5, pp. 645–652, Oct. 2006.

[20] F. L. Lewis, L. Xie, and D. Popa, *Optimal and Robust Estimation With an Introduction to Stochastic Control Theory, Second Edition, With an Introduction to Stochastic Control Theory, Second Edition*. Taylor & Francis Group, 2017, p. 552.

[21] X. Lihua, "On robust H_2 estimation," *Acta Automat. Sinica*, vol. 31, no. 1, pp. 1–12, 2005.

[22] X. Lihua, Y. C. Soh, D. Chunling, and Y. Zou, "Robust h2 estimation and control," *J. Control Theory and Applicat.*, vol. 2, no. 1, pp. 20–26, Feb. 2004.

[23] K.-Z. Liu and T. Sato, "LMI solution to singular H_2 suboptimal control problems," *IFAC Proc. Volumes*, vol. 32, no. 2, pp. 3011–3016, Jul. 1999.

[24] J. J. Lopez-Solorzano and Y. S. Shmaliy, "Receding Horizon H_2-FIR Filtering for Uncertain Disturbed Systems With Initial and Data Errors," *IEEE Trans. Automation Sci. Eng.*, pp. 1–8, To Be Published.

[25] J. J. López-Solórzano and Y. S. Shmaliy, "Uncertain disturbed systems: Robust H_2-OUFIR filtering under measurement errors," *Int. J. of Adaptive Control and Signal Process.*, vol. 36, no. 10, pp. 2383–2400, Oct. 2022.

[26] C. Martin and M. Mintz, "Robust filtering and prediction for linear systems with uncertain dynamics: A game-theoretic approach," *IEEE Trans. Autom. Control*, vol. 28, no. 9, pp. 888–896, Sep. 1983.

[27] J. Ortega-Contreras, E. Pale-Ramon, Y. S. Shmaliy, and Y. Xu, "A Novel H_2 Approach to FIR Prediction Under Disturbances and Measurement Errors," *IEEE Signal Process. Lett.*, vol. 28, pp. 150–154, 2021.

[28] J. Ortega-Contreras, Y. S. Shmaliy, J. Andrade-Lucio, and O. Ibarra-Manzano, "Robust H_2-OFIR Filtering: Improving Tracking of Disturbed Systems under Initial and Data Errors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 5, pp. 4761–4770, Oct. 2022.

[29] E. G. Pale-Ramon, Y. S. Shmaliy, J. A. Andrade-Lucio, and L. J. Morales-Mendoza, "Bias-Constrained H_2 Optimal Finite Impulse Response Filtering for Object Tracking Under Disturbances and Data Errors," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 4, pp. 1782–1789, Jul. 2021.

[30] P. Rawicz, P. Kalata, K. Murphy, and T. Chmielewski, "Explicit formulas for two state kalman, H_2 and H_∞ target tracking filters," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 1, pp. 53–69, Jan. 2003.

[31] C. Scherer and S. Weiland, "Linear matrix inequalities in control," *Lecture Notes, Dutch Inst. for Syst. and Control, Delft, The Netherlands*, vol. 3, no. 2, 2000.

[32] I. C. Schick and S. K. Mitter, "Robust Recursive Estimation in the Presence of Heavy-Tailed Observation Noise," *The Ann. Stat.*, vol. 22, no. 2, Jun. 1994.

[33] Z. Shareef, P. Mohammadi, and J. Steil, "Improving the inverse dynamics model of the KUKA LWR IV+ using independent joint learning," *IFAC-PapersOnLine*, vol. 49, no. 21, pp. 507–512, 2016.

[34] Y. Shmaliy, Y. Xu, J. Andrade-Lucio, and O. Ibarra-Manzano, "Predictive Tracking Under Persistent Disturbances and Data Errors Using H_2 FIR Approach," *IEEE Trans. Ind. Electron.*, vol. 69, no. 6, pp. 6121–6129, Jun. 2022.

[35] Y. S. Shmaliy, "An iterative kalman-like algorithm ignoring noise and initial conditions," *IEEE Trans. Signal Process.*, vol. 59, no. 6, pp. 2465–2473, Jun. 2011.

[36] Y. S. Shmaliy and O. Ibarra-Manzano, “Time-variant linear optimal finite impulse response estimator for discrete state-space models,” *Int. J. of Adaptive Control and Signal Process.*, vol. 26, no. 2, pp. 95–104, Sep. 2011.

[37] Y. S. Shmaliy and S. Zhao, *Optimal and Robust State Estimation, Finite Impulse Response and Kalman Approaches*. Wiley & Sons, Limited, John, 2022.

[38] Y. S. Shmaliy, S. Zhao, and C. K. Ahn, “Unbiased finite impluse response filtering: An iterative alternative to kalman filtering ignoring noise and initial conditions,” *IEEE Control. Syst. Mag.*, vol. 37, no. 5, pp. 70–89, Oct. 2017.

[39] D. J. Simon, “Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches,” in John Wiley & Sons, Inc., Jan. 2006, pp. 331–371.

[40] R. E. Skelton, T. Iwasaki, and K. Grigoriadis, *A Unified Algebraic Approach To Control Design*. Routledge, Nov. 2017.

[41] Z. Tan, Y. C. Soh, and L. Xie, “Envelope-constrained H_∞ filter design: An LMI optimization approach,” *IEEE Trans. Signal Process.*, vol. 48, no. 10, pp. 2960–2963, 2000.

[42] W. F. Trench, “A general class of discrete time-invariant filters,” *J. Soc. for Ind. and Appl. Math.*, vol. 9, no. 3, pp. 405–421, Sep. 1961.

[43] M. Vidyasagar, “Further results on the optimal rejection of persistent bounded disturbances,” *IEEE Trans. Autom. Control*, vol. 36, no. 6, pp. 642–652, Jun. 1991.

[44] S. Wang, L. Xie, and C. Zhang, “H_2 optimal inverse of periodic FIR digital filters,” *IEEE Trans. Signal Process.*, vol. 48, no. 9, pp. 2696–2700, 2000.

[45] S. Wang, L. Xie, and C. Zhang, “Mixed H_2/H_∞ deconvolution of uncertain periodic FIR channels,” *Signal Process.*, vol. 81, no. 10, pp. 2089–2103, Oct. 2001.

[46] D. Wilson, “Convolution and hankel operator norms for linear systems,” *IEEE Trans. Autom. Control*, vol. 34, no. 1, pp. 94–97, 1989.

Appendix A

MATLAB Code for H_∞ FIR Filter Gain Computation (Alg. 1)

```
function [H_inf_gain,gamma] = H_inf_FIR_Filter_Gain(N,F,D_N,H_N,G_N,Q_N,R_N,d_0)
%% UFIR gain computation
C_N = H_N/(F^(N-1));
UFIR_gain = (C_N'*C_N)\C_N';

%% Sparce matrices generation
Aw = zeros(N,N);
Aw(1:end-1,2:end) = eye(N-1);

Bw = zeros(N,1);
Bw(end) = 1;

%% Disturbance-to-Error state space matrices
st = size(F,1); % Number of states
F_c = [Aw zeros(N,N) zeros(N,st);
    zeros(N,N) Aw zeros(N,st);
    zeros(st,N) zeros(st,N) eye(st)];

B_c = [Bw zeros(N,1);
    zeros(N,1) Bw;
    zeros(st,1) zeros(st,1)];

%% Disturbance covariance
Q = Q_N(1,1); % Process noise covariance
```

```
R = R_N(1,1); % Measurement noise covariance
P_c = diag([Q R]);

%% Auxiliar variables
x_m = zeros(st,st);
A = (F^(N-1))*x_m*(F^(N-1))' + D_N(end-1:end,:)*Q_N*D_N(end-1:end,:)';
C = (F^(N-1))*x_m*H_N' + D_N(end-1:end,:)*Q_N*G_N';
Omega_mk = G_N*Q_N*G_N' + R_N;

P_J = [A -C;
    -C' (H_N*x_m*H_N'+Omega_mk)];

%% Algorithm 1
H_inf_gain = sdpvar(st,N);
assign(H_inf_gain,UFIR_gain); % Initialice with UFIR gain

Z = sdpvar(N,N);
assign(Z,UFIR_gain'*UFIR_gain); % Initialice with UFIR gain

gamma2 = sdpvar(1);
K = sdpvar(N*st+st);

while (abs(trace(value(Z))-trace(value(H_inf_gain)'*value(H_inf_gain))) < d_0)
    H_inf_gain_prev = value(H_inf_gain);
    Z_prev = value(Z);
    gamma_prev = sqrt(value(gamma2));

    J_c = [D_N(end-1:end,:)-H_inf_gain*G_N, ...
        -H_inf_gain F^(N-1)-H_inf_gain*H_N;
        H_inf_gain'*D_N(end-1:end,:)-Z*G_N, ...
        -Z H_inf_gain'*(F^(N-1))-Z*H_N];

    M1 = [-K K*F_c K*B_c zeros(N*st+st,N+st);
        F_c'*K -K zeros(N*st+st,2) F_c'*J_c';
        B_c'*K zeros(2,N*st+st) -gamma2*P_c B_c'*J_c';
        zeros(N+st,N*st+st) J_c*F_c J_c*B_c -inv(P_J)];

    M2 = [Z H_inf_gain';
        H_inf_gain eye(2)];
```

```
    obj = gamma2;

    const = [M1<=0,M2>=0,gamma2>=0,K>=0,trace(Z)==1.01*trace(Z_prev)];
    opt = sdpsettings;
    opt.solver = 'mosek';
    optimize(const,obj,opt);
end

gamma = gamma_prev;
H_inf_gain = H_inf_gain_prev;
```

Code A.1: H_inf_FIR_Filter_Gain.m

```
function [x_e_pos,P_pos] = FIR_estimation(y,N,F,D_N,H_N,G_N,Q_N,R_N,FIR_gain)
K = size(F,1); % Number of states

x_e_pos = zeros(K,length(y));
P_pos = zeros(K,K,length(y));

m = 0;
for k = N:length(y)-1
    m = m+1;

    Y_mk = transpose(y(m+1:k+1)); % Batch of N points of the measurements

    % Initial value covariance
    x_e_pos_m = (x_e_pos(:,m+1))*x_e_pos(:,m+1)';
    x_m = x_e_pos_m + P_pos(:,:,m+1);

    % Convolution-based FIR filtering
    x_e_pos(:,k+1) = FIR_gain*Y_mk;

    % Error covariance matrix
    B_N = (F^(N-1)) - FIR_gain*H_N;
    W_N = D_N(end-1:end,:)-FIR_gain*G_N;
    V_N = FIR_gain;
    P_pos(:,:,k+1) = B_N*x_m*B_N' + W_N*Q_N*W_N' + V_N*R_N*V_N';
end
```

Code A.2: FIR_estimation.m

Appendix B

MATLAB Code for H_∞ FIR Filter Gain Computation (Alg. 2)

```
function [H_inf_gain,gamma] = H_inf_FIR_Filter_Gain_it(N,F,D_N,H_N,G_N,Q_N,R_N)
%% UFIR gain computation
C_N = H_N/(F^(N-1));
UFIR_gain = (C_N'*C_N)\C_N';

%% Sparce matrices generation
Aw = zeros(N,N);
Aw(1:end-1,2:end) = eye(N-1);

Bw = zeros(N,1);
Bw(end) = 1;

%% Disturbance-to-Error state space matrices
st = size(F,1); % Number of states
F_c = [Aw zeros(N,N) zeros(N,st);
    zeros(N,N) Aw zeros(N,st);
    zeros(st,N) zeros(st,N) eye(st)];

B_c = [Bw zeros(N,1);
    zeros(N,1) Bw;
    zeros(st,1) zeros(st,1)];

%% Disturbance covariance
Q = Q_N(1,1); % Process noise covariance
```

```
R = R_N(1,1); % Measurement noise covariance
P_c = diag([Q R]);

%% Auxiliar variables
x_m = zeros(st,st);
A = (F^(N-1))*x_m*(F^(N-1))' + D_N(end-1:end,:)*Q_N*D_N(end-1:end,:)';
C = (F^(N-1))*x_m*H_N' + D_N(end-1:end,:)*Q_N*G_N';
Omega_mk = G_N*Q_N*G_N' + R_N;

%% Algorithm 2
H_inf_gain = sdpvar(st,N);
assign(H_inf_gain,UFIR_gain); % Initialice with UFIR gain

% Disturbance-to-Error state space matrix
C_c = ([D_N(end-1:end,:)-H_inf_gain*G_N -H_inf_gain ...
    (F^(N-1))-H_inf_gain*H_N]);

GNPG = sdpvar(st);
assign(GNPG,UFIR_gain*UFIR_gain') % Initialice with UFIR gain

gamma2 = sdpvar(1);
K = sdpvar(N*st+st);

while 1
    P_prev = A - C*value(H_inf_gain)' - value(H_inf_gain)*C' + ...
        value(H_inf_gain)*(H_N*x_m*H_N'+Omega_mk)*value(H_inf_gain)';
    H_inf_gain_prev = value(H_inf_gain);
    gamma_prev = sqrt(value(gamma2));
    GNPG_prev = value(H_inf_gain)*value(H_inf_gain)';

    M1 = [-K K*F_c K*B_c zeros(N*st+st,st);
        F_c'*K -K zeros(N*st+st,2) F_c'*C_c';
        B_c'*K zeros(2,N*st+st) -gamma2*P_c B_c'*C_c';
        zeros(st,N*st+st) C_c*F_c C_c*B_c -inv(P_prev)];

    M2 = [GNPG H_inf_gain;
        H_inf_gain' eye(N)];

    obj = gamma2;
```

```
    const = [M1<=0,M2>=0,gamma2>=0,K>=0,0.99*trace(GNPG)<=trace(GNPG_prev)];
    opt = sdpsettings;
    opt.solver = 'mosek';
    optimize(const,obj,opt);

    P_e = A - C*value(H_inf_gain)' - value(H_inf_gain)*C' + ...
        value(H_inf_gain)*(H_N*x_m*H_N'+Omega_mk)*value(H_inf_gain)';

    if ~(trace(trace(P_e)) < trace(trace(P_prev)))
        break;
    end
end

gamma = gamma_prev;
H_inf_gain = H_inf_gain_prev;
```

Code B.1: H_inf_FIR_Filter_Gain_it.m

Appendix C

MATLAB Code for Recursive H_∞ Filter Bias Correction Gain Computation (Alg. 4)

```
function [K_H_inf,gamma] = rec_H_inf_Filter_BCG(K_UFIR,F,B,H,Q,R,d_0)
st = size(F,1); % Number of states

%% Disturbance covariance matrix
P_c = diag([Q R]);

%% Algorithm 4
K_H_inf = sdpvar(st,1);
assign(K_H_inf,K_UFIR); % Initialice with recursive UFIR

Z = sdpvar(1,1);
assign(Z,K_UFIR'*K_UFIR); % Initialice with recursive UFIR

gamma2 = sdpvar(1);
while (abs(trace(value(Z)) - trace(value(K_H_inf)'*value(K_H_inf))) < d_0)
    % a priori error covariance update
    K_H_inf_prev = value(K_H_inf);
    gamma_prev = sqrt(value(gamma2));
    P = dlyap(F*(eye(st)-K_H_inf_prev*H),...
        F*K_H_inf_prev*R*K_H_inf_prev'*F'+B*Q*B');
    P_J = [P -P*H';
```

```
        -H*P H*P*H'+R];

    % Computation of K
    F_c = (eye(st)-K_H_inf_prev*H)*F;
    B_c = [(eye(st)-K_H_inf_prev*H)*B -K_H_inf_prev];

    J_c = [(eye(st)-K_H_inf_prev*H)*F;
        (K_H_inf_prev'-K_H_inf_prev'*K_H_inf_prev*H)*F];
    L_c = [(eye(st)-K_H_inf_prev*H)*B -K_H_inf_prev;
        (K_H_inf_prev'-K_H_inf_prev'*K_H_inf_prev*H)*B ...
            -K_H_inf_prev'*K_H_inf_prev];

    K = sdpvar(st);

    M1 = [-K K*F_c K*B_c zeros(st,st+1);
        F_c'*K -K zeros(st,2) J_c';
        B_c'*K zeros(2,st) -gamma2*P_c L_c';
        zeros(st+1,st) J_c L_c -inv(P_J)];

    obj = gamma2;

    const = [M1<=0,gamma2>=0,K>=0];
    opt = sdpsettings;
    opt.solver = 'mosek';
    optimize(const,obj,opt);

    K = value(K); % K as constant

    % H_inf bias correction gain computation
    Z_prev = value(Z);

    F_c = (eye(st)-K_H_inf*H)*F;
    B_c = [(eye(st)-K_H_inf*H)*B -K_H_inf];

    J_c = [(eye(st)-K_H_inf*H)*F;
        (K_H_inf'-Z*H)*F];
    L_c = [(eye(st)-K_H_inf*H)*B -K_H_inf;
        (K_H_inf'-Z*H)*B -Z];

    M1 = [-K K*F_c K*B_c zeros(st,st+1);
```

```
        F_c'*K -K zeros(st,2) J_c';
        B_c'*K zeros(2,st) -gamma2*P_c L_c';
        zeros(st+1,st) J_c L_c -inv(P_J)];

    M2 = [Z K_H_inf';
        K_H_inf eye(st)];

    obj = gamma2;

    const = [M1<=0,M2>=0,gamma2>=0,0.99*trace(Z)==trace(Z_prev)];
    opt = sdpsettings;
    opt.solver = 'mosek';
    optimize(const,obj,opt);
end

gamma = gamma_prev;
K_H_inf = K_H_inf_prev;
```

Code C.1: rec_H_inf_Filter_BCG.m

```
function [x_e_pos,P_pos] = recursive_estimation(y,F,B,H,Q,R,K)
st = size(F,1); % Number of states
x_e_pos = zeros(st,length(y)); % a posteriori estimation
x_e_pri = zeros(st,length(y)); % a priori error covariance
P_pos = zeros(st,st,length(y)); % a posteriori estimation
P_pri = zeros(st,st,length(y)); % a priori error covariance

for k = 1:length(y)-1
    x_e_pri(:,k+1) = F*x_e_pos(:,k);
    P_pri(:,:,k+1) = F*P_pos(:,:,k)*F' + B*Q*B';
    s = y(k+1) - H*x_e_pri(:,k+1);
    x_e_pos(:,k+1) = x_e_pri(:,k+1) + K*s;
    P_pos(:,:,k+1) = (eye(st)-K*H)*P_pri(:,:,k+1)* + ...
        (eye(st)-K*H)' + K*R*K';
end
```

Code C.2: recursive_estimation.m

Appendix D

MATLAB Code for Recursive H_∞ Filter Bias Correction Gain Computation (Alg. 6)

```
function [K_H_inf,gamma] = rec_H_inf_Filter_BCG_it(K_UFIR,GNPG_UFIR,F,B,H,Q,R)
st = size(F,1); % Number of states

%% Disturbance covariance matrix
P_c = diag([Q R]);

%% Disturbance-to-Error state space matrix
C_c = eye(st);

%% Algorithm 6
K_H_inf = sdpvar(st,1);
assign(K_H_inf,K_UFIR);

GNPG = sdpvar(2);
assign(GNPG,GNPG_UFIR)

gamma2 = sdpvar(1);

while 1
    K_H_inf_prev = value(K_H_inf);
    gamma_prev = sqrt(value(gamma2));
```

```
F_c = (eye(st)-K_H_inf_prev*H)*F;
B_c = [(eye(st)-K_H_inf_prev*H)*B -K_H_inf_prev];

P_prev = dlyap(F_c,B_c*P_c*B_c');
GNPG_prev = value(GNPG);

% Computation of K
K = sdpvar(st);
M1 = [-K K*F_c K*B_c zeros(st,st);
    F_c'*K -K zeros(st,2) F_c'*C_c';
    B_c'*K zeros(2,st) -gamma2*P_c B_c'*C_c';
    zeros(st,st) C_c*F_c C_c*B_c -inv(P_prev)];

obj = gamma2;

const = [M1<=0,gamma2>=0,K>=0];
opt = sdpsettings;
opt.solver = 'mosek';
optimize(const,obj,opt);

K = value(K); % K as constant

% H_inf bias correction gain computation
F_c = (eye(st)-K_H_inf*H)*F;
B_c = [(eye(st)-K_H_inf*H)*B -K_H_inf];

M1 = [-K K*F_c K*B_c zeros(st,st);
    F_c'*K -K zeros(st,st) F_c'*C_c';
    B_c'*K zeros(st,st) -gamma2*P_c B_c'*C_c';
    zeros(st,st) C_c*F_c C_c*B_c -inv(P_prev)];

M2 = GNPG*H' - K_H_inf;

obj = gamma2;

const = [M1<=0,M2>=0;gamma2>=0,GNPG(1,1)==1.1*GNPG_prev(1,1)];
opt = sdpsettings;
opt.solver = 'mosek';
optimize(const,obj,opt);
```

```
    P_e = dlyap(value(F_c),value(B_c*P_c*B_c'));

    if ~(trace(trace(P_e)) < trace(trace(P_prev)))
        break;
    end
end

gamma = gamma_prev;
K_H_inf = K_H_inf_prev;
```

Code D.1: rec_H_inf_Filter_BCG_it.m