RECEDING HORIZON OPTIMIZATION IN HAAR DOMAIN FOR UNCONSTRAINED LINEAR TIME-INVARIANT SYSTEMS

By

RUSHD MD KHALED JULFIKER

Bachelor of Science in Mechanical Engineering
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh
2009

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2013

RECEDING HORIZON OPTIMIZATION IN HAAR DOMAIN FOR UNCONSTRAINED LINEAR TIME-INVARIANT SYSTEMS

Thesis Approved:

Dr. Prabhakar Pagilla

Thesis Advisor

Dr. Gary E. Young

Dr. Martin Hagan

ACKNOWLEDGMENTS

I would like to express my sincerest gratitude to my advisor and thesis committee chairman Dr. Prabhakar R. Pagilla for his supervision, guidance and inspiration. I am forever grateful to him for his motivation and technical insights, allowing me to broaden my horizon in Wavelet analysis. It was wonderful learning experience. I will certainly benefit from the knowledge I have collected working under his supervision. I would also like to thank Dr. Martin Hagan and Dr. Gary E. Young for being on my thesis committee. Their suggestions and understanding made the development of this thesis a positive learning experience.

I would like to thank Dr. Mauro Cimino for his guidance during the initial stages of this project. His knowledge and encouragement were certainly helpful during my setbacks.

I would also like to extend my gratitude towards my wonderful colleagues at Oklahoma State University; Pramad Raul, Orlando Cobos, Shyam Konduri, Emilio Gabino and Maya Wulandari for their intellectual inputs and discussion all through the time of my research.

Finally and most deeply, I would like express my appreciation to my parents and wife. Without their love, support, patience, blessings and sacrifices, I would never have come this far in my life.

Acknowledgments reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: Rushd Md Khaled Julfiker

Date of Degree: December, 2013

Title of Study: RECEDING HORIZON OPTIMIZATION IN HAAR DOMAIN FOR UNCONSTRAINED LINEAR TIME-INVARIANT SYSTEMS

Major Field: Mechanical and Aerospace Engineering

Scope and Method of Study: The focus of this study is on developing Haar wavelet based Model Predictive Controller (MPC) for linear unconstrained systems. The problem of computation load in MPC has been addressed. By utilizing the structure of the Haar transformation, the coefficients that construct the first control input in the prediction horizon of the MPC are isolated. Considering only these coefficients, Haar based optimization procedure has been modified. The performance and computational load are compared with those of a Dynamic Matrix Controller (DMC) for a velocity regulation problem of a DC motor. Using the proposed modified Haar based MPC, position and orientation control of a two link planar robot and a wheeled mobile robot are provided as examples to reinforce the discussions.

Findings and Conclusions: Modifications in the Haar based MPC reduced the amount of computation necessary to construct the first control action in the prediction horizon. Despite the modification and reduction in computation, the controller could handle sudden changes in setpoint, which was depicted in a velocity regulation of a DC motor. For increase in the size of prediction horizon beyond 2^6 time steps, Haar based MPC requires less computation than DMC. Large prediction horizons provide more stability, less aggressive control action and smoother response for the Haar based MPC.

TABLE OF CONTENTS

Chapter Page

I INTRODUCTION 1

I.1 Thesis Contributions 4

I.2 Thesis Organization 5

II TIME FREQUENCY REPRESENTATIONS 6

II.1 Time-Frequency Analysis 6

II.1.1 Short Time Fourier Transform (STFT) 7

II.1.2 Time-Frequency Resolution for STFT 9

II.1.3 Discrete Short Time Fourier Transform 11

II.1.4 Scaling in Short Time Fourier Transform 12

II.2 Wavelet Analysis 13

II.2.1 Wavelet Function 14

II.2.2 Scaling Function 16

II.2.3 Time-Frequency Resolution in Wavelet Domain 17

II.2.4 Discrete Wavelet Transform (DWT) 19

II.3 Orthogonal Wavelet Bases 21

II.3.1 Multiresolution Analysis (MRA) 21

II.3.2 Orthogonal Wavelet Transform 23

II.3.3 Frequency Bands and Time Intervals for Orthogonal Wavelets 24

II.3.4 Decomposition of a signal into Haar Wavelets 26

III OPTIMAL CONTROL OF LINEAR SYSTEMS VIA HAAR WAVELETS 33

III.1 Optimal Control Problem 33

III.1.1 Indirect Methods 34

III.1.2 Direct Method 34

III.2 Orthogonal Functions in Solving Differential Equations 35

III.3 Integration of Haar Wavelets 35

III.4 Solving Linear Time Invariant Systems via Haar Wavelets 39

III.4.1 Example of a LTI System solution via Haar Wavelets 43

III.5 Open Loop Optimal Control via Haar Wavelets 45

III.5.1 Example of Open Loop Optimal Control via Haar Wavelets . . 52

III.6 Receding Horizon Optimization 53

III.7 Closed Loop Optimal Control 55

III.8 Modified Computation for Closed Loop Optimal Control 58

III.8.1 Computation Load in Modified Closed Loop Optimization . . 67

III.9 Comparison with Dynamic Matrix Control (DMC) 67

III.9.1 DMC Control Law 68

III.9.2 Velocity Regulation of a DC motor 70

IV TRAJECTORY TRACKING 72

IV.1 Position Control of Two Link RR Planar Robot 72

IV.1.1 Position Control 74

IV.2 Trajectory Tracking of a Mobile Robot 79

IV.2.1 Robot Kinematics 79

IV.2.2 Position and Orientation Tracking of Mobile Robot 81

V CONCLUSION AND FUTURE WORK 85

REFERENCES 88

A ORTHONORMAL BASES 93

B WAVELET FAMILIES **95**

B.1 Wavelets used in Continuous Wavelet Transform 95

B.1.1 Morlet Wavelet 95

B.1.2 Gaussian Wavelet 96

B.1.3 Mexican Hat Wavelet 97

B.1.4 Meyer Wavelet 97

B.1.5 Frequency B-Spline Wavelet 98

B.2 Wavelets used in Discrete Wavelet Transform 99

B.2.1 Daubechies Wavelet 99

B.2.2 Coiflets Wavelet 100

B.2.3 Symlet Wavelet 101

C SOME PROPERTIES OF KRONECKER PRODUCT, VECTOR, TRACE AND MATRIX DERIVATIVES **103**

D WEIGHT MATRICES IN LINEAR QUADRATIC COST FUNCTION **104**

E ROBOT KINEMATICS AND DYNAMICS **106**

E.1 Forward Kinematics 106

E.2 Inverse Kinematics 107

E.3 Robot Dynamics 108

F ROBOT TRAJECTORY GENERATION **111**

LIST OF TABLES

Table Page

IV.1 Symbol definitions for kinematic and dynamic structure of 2 link planar robot 73

LIST OF FIGURES

Figure Page

II.1 Windows for STFT with analyzing functions of different frequencies (from (II.1)). Wider windows encompass more oscillations while smaller windows do fewer. 8

II.2 Time-frequency boxes representing the energy spread of two windows for different time-frequency localization. Window sizes do not change over time-frequency plane, so time-frequency resolution stay the same over the plane. 10

II.3 Low scale gives high time resolution but low frequency resolution, and vise versa. In all the cases, both time and frequency resolutions remain fixed over time-frequency plane once the window is decided. 13

II.4 Wavelets of same shape but different sizes 14

II.5 Time-Frequency Boxes representing the energy spread for different time-frequency localization in wavelet domain. Different sizes of the windows give different time-frequency resolution over the plane. . . . 17

II.6 Multi-resolution of time and frequency 18

II.7 Multiresolution analysis of $L^2(\mathbb{R})$ into subspaces. 22

II.8 For a signal sampled at 1000 Hz, temporal widths of Haar functions and their different frequency bands of projections are shown for a decomposition up to $j = 4$. Both the size of frequency band and width of the functions vary dyadically with the change of j. 25

II.9 Time evolution of $x(t)$ at different frequency ranges, starting from the frequency band at lowest frequency region. 32

III.1 Haar functions of H_4 and their integrals, defined over [0 1]s. 37

III.2 Comparison between states from analytic solution and solution via Haar wavelets. 43

III.3 Comparison between states from analytic solution and solution via Haar wavelets. 45

III.4 Open Loop Optimal Control 51

III.5 Time evolution of states and open loop optimal control via Haar wavelet. 52

III.6 Optimal control sequence u^t with fixed horizon size T_f is receding in time. Only the first control action (with the ✓mark) is applied to the system, rests are discarded. 54

III.7 Receding Horizon Optimizer acting as a closed loop controller. Control u^t is determined by minimizing the states at each time step. The first control step u_t is selected from the optimal control sequence u^t and set as $u(t) = u_t$. So $u(t + T_s) = f(x(t))$. 55

III.8 Close Loop Optimal Control 57

III.9 Modified Close Loop Optimal Control 66

III.10Desired and actual angular velocity of the motor from Haar based MPC 71

III.11Desired and actual angular velocity of the motor from DMC 71

IV.1 2 link planar robot in global x-y co-ordinate 74

IV.2 Auxiliary controls in Figures IV.2(g) and IV.2(h) are the close loop auxiliary optimal control obtained in receding horizon control fashion, via Haar wavelet. τ_1 and τ_2 in Figures IV.2(i) and IV.2(j) are torques driving the states of the links towards their desired trajectories, shown in Figures IV.2(a)-IV.2(f). Figure IV.2(k) shows the desired and actual end-effector trajectory in Cartesian space. 79

IV.3 Mobile robot kinematic model 80

IV.4 Control signals are driving the error states towards zero. The robot is initially placed outside the reference circle, then the actual trajectory merges with the reference trajectory. 83

B.1 Amplitude of real and imaginary part of Morlet wavelet for $\omega_b = 1$ Hz and $\omega_c = 1$ Hz. 96

B.2 Amplitude of real and imaginary part of Gaussian wavelet for $N = 2$. 97

B.3 Amplitude of Mexican Hat wavelet for $N = 2$. 97

B.4 Amplitude of Meyer scalar and wavelet function. 98

B.5 Amplitude of real and imaginary part of frequency B-spline wavelet for $\omega_b = 1$ Hz, $\omega_c = 1$ Hz and $p = 2$. 99

B.6 Amplitude of real and imaginary part of Shannon wavelet for $\omega_b = 1$ Hz and $\omega_c = 1$ Hz. 99

B.7 Daubechies scalar and wavelet function of order 2 and 4 respectively. 100

B.8 Coiflet scalar and wavelet function of order 2 and 4 respectively. . . . 101

B.9 Symlet scalar and wavelet function of order 2 and 4 respectively. . . . 102

F.1 Desired circular trajectory 112

CHAPTER I

INTRODUCTION

This thesis explores the potential of Haar wavelets in designing a Receding Horizon Controller with reduced computation load at each time step. Wavelets are set of orthogonal functions which form the basis of square integrable functions. They were developed separately in the field of mathematics, numerical analysis, electrical engineering, seismic geology and quantum field theory. Now they have many applications, such as image processing, data compression, noise removal, earthquake prediction, human and computer vision [1], [2].

Mathematical background of wavelets dates back to the work of Joseph Fourier (1807), which laid the foundation of frequency analysis. The first ever wavelet that can be found in the literature is the Haar wavelet, in the work of Alfred Haar [3]. This wavelet has compact support, which means it is non zero only on a finite time interval. It is not continuously differentiable, but its compact support enables it to approximate functions with discontinuities or sudden changes. In 1946, physicist Gabor proposed decomposing a signal over elementary waveforms with minimal spread in time-frequency plane. He called these waveforms the Time-Frequency atoms [4]. The algorithm for wavelet analysis dates back to the work of Stephane Mallat [5]. Following his work, Ingrid Daubechie constructed a set of wavelet orthonormal basis functions which covers the mathematical background of wavelets [6].

Wavelets represent a signal both in time and frequency domain, comparable with Short Time Fourier transformation which is shifted in time. The difference is, in wavelets, the window width changes as a function of the analyzing frequency band, whereas in Fourier transformation, the width remains the same. In the case of discrete wavelet transform, a signal can be decomposed into wavelet coefficients, which can be used to reconstruct the signal. Before reconstruction, these coefficients can be processed using filter banks, which will remove the noise corresponding to certain frequency band. Denoisoing with wavelets is useful in the case of short lived external disturbances. Data compression can also be achieved by omitting insignificant wavelet coefficients.

Due to the orthogonality of the wavelets, they can be used to form integral and delay operational matrices. Wavelet coefficients together with operational matrices are used in the analysis and controller design of linear time invariant/variant (LTI/LTV) systems with delays [7–10].

Like any other orthogonal functions, wavelets are used to find approximate solutions to the ordinary differential equations (ODEs) for a finite time interval [11]. It is done by integrating the ODEs. The states and inputs are approximated by orthogonal functions and an integral operational matrix is used in the equation to replace the integration operation, thus converting the ODE into an algebraic equation. The solution of this algebraic equation is obtained in terms of the coefficients of the orthogonal function. Different orthogonal functions can be chosen from Block pulse functions, Chebyshev polynomials, Fourier series, Hermite polynomials, Jacobi series, Laguerre series, shifted Legendre polynomials, Taylor series, Walsh functions, Bessel series and Wavelet functions [12]. The selection of the function will also determine the form of the integral and delay operational matrices. Solving ODEs in wavelet domain leads to the process of solving an optimal control problem in wavelet domain.

Optimal control is a process of determining the control input and state trajectory such that a performance index is minimized while satisfying the system equation and constraints. Due to the complexity in most applications, optimal control problems are solved numerically. These methods are divided into two broad classes: indirect and direct methods. In the indirect method, calculus of variations is used to determine a first-order optimality condition of the original optimal control problem [13]. It converts the optimal control problem into a multiple point boundary value problem (MPBVP), which is solved to determine candidate optimal trajectories known as extremals. Each of the candidate extremals are then examined to see if it is a local minimum, maximum or saddle point. The particular extremal with the lowest performance index is selected. This method involves the solution of the Hamilton-Jacobi-Bellman (HJB) equation. For linear time invariant systems, this problem reduces to solving an algebraic Riccati equation. For general systems, however, the problem remains in finding the value function that satisfies the HJB equation. Finding the value function is challenging because it requires the solution of a partial differential equation which cannot be solved explicitly.

Direct approach bypasses the tenacious job of solving the HJB equation. Here, the optimal solution is obtained by direct minimization of the cost function subjected to constraints. For this approach, various computational methods based on orthogonal wavelet functions have been developed in the past recent years [10, 14–17]. In this method, the control variable is discretized and approximated by a finite series of the known wavelet function with unknown coefficients. Then the state variables are solved for a finite time interval in terms of those unknown parameters by integrating the system state equation.

Upon getting the solution, an appropriate cost function and control law are selected. The cost function is transformed into the wavelet domain and it is subjected to the optimality criterion. The resulting equation, together with solution of the system differential equation, gives the optimal control and state trajectory. The initial states, plant parameters and the desired trajectory are what determine the control. The control can be transformed back into the time domain by using the inverse wavelet transform. This method is also known as Model Predictive Control (MPC), as it uses a model of the plant to predict the states of the system which is again optimized by minimizing a cost function. At each time step this process is repeated over a certain horizon which also recedes with time, so this process is also known as Receding Horizon Control [18].

In this thesis Haar wavelet based open loop optimization of unconstrained linear system is applied to construct a Haar based MPC. Haar is a compact basis function with a simple analytical expression. While devising the Haar based MPC, open loop optimization formulas from [16] are revised to compute only those Haar coefficients which will build the first control action from the control horizon. Thus the amount of computation involved in Haar based optimization at each time step is reduced. With this reduced computation, the performance of Haar based MPC is compared with a Dynamic Matrix Controller (DMC), which is a conventional MPC controller.

I.1 Thesis Contributions

The contributions of this thesis are summarized in the following:

- Optimization based on Haar wavelets is implemented in a Receding Horizon Control fashion to construct a closed loop controller.

- Derivations for ordinary differential solution and optimization in Haar domain are modified to isolate the Haar wavelets necessary for only the first control action and thus reducing the computation order.

- The designed closed loop optimizer is used to stabilize systems to minimize the error at each time step. This is shown by MATLAB simulations of velocity regulation of a DC motor and position and orientation tracking of a two link planar robot and a wheeled mobile robot.

I.2 Thesis Organization

The rest of the thesis report is organized as follows. Time-frequency representation of a function is discussed in Chapter II. Haar wavelet method for solving the optimal control problem for LTI systems is reviewed in Chapter III. Receding Horizon Optimization for closed loop control scheme and modifications in the derivations are shown in this chapter. Performance of the modified algorithm for velocity regulation of a DC motor is compared with DMC in this chapter. Trajectory tracking using the proposed controller are discussed with examples in Chapter IV. Conclusion and possible future works are discussed in Chapter V.

CHAPTER II

TIME FREQUENCY REPRESENTATIONS

This chapter is modest in scope, it is only a brief introduction to a vast subject that itself is small part of mathematics: how a signal can be represented in different forms. Information can be represented in different forms; a mathematical function or a physical signal may be decomposed using a *Fourier transform* or a *wavelet transform.* Even though wavelets represent a departure from Fourier analysis, the two methods clearly belong to the same family.

Most signals are represented as a sequence of numbers as a function of time. But some of their properties are better revealed in the frequency domain. Fourier analysis is useful to get global information about the frequency distribution of a signal, but it cannot localize frequency component in time domain. The localized frequency content is revealed by decomposing signals using functions which are localized in time and frequency domain. Short Time Fourier Transform and Wavelet Transform are two important techniques that can be used to localize in time the frequency content of signals. The first section provides an overview on Time-Frequency Analysis and its various forms. Section II.2 and II.3 present some background on multi-resolution frequency analysis.

II.1 Time-Frequency Analysis

In time frequency analysis, localized time-frequency information of a signal are revealed using basis function which are localized and time-frequency. The Short Time Fourier Transform (STFT) and the Wavelet analysis are two important techniques that can be used to localize in time the frequency content of signals.

II.1.1 Short Time Fourier Transform (STFT)

To localize frequency component of a signal in time, the Fourier transform is used in a short window of time. Sliding of window over the signal in time provides a time localized Fourier transform. This process is knows as Short Time Fourier Transform (STFT), first introduced by Dennis Gabor in 1946 [4].

In STFT, width of the sliding window determines how much of the signal would be taken into consideration, in both time and frequency. A real and symmetric window $g(t) = g(-t)$ is translated in time domain by τ and modulated by the frequency ξ.

$$g_{\xi,\tau}(t) = g(t-\tau)e^{2\pi i\xi t} \tag{II.1}$$

Here $g \in L^2(\mathbb{R})$ and $||g||^2 = \int |g(t)|^2 dt$. τ is the mean time in $g_{\xi,\tau}(t)$, which is given by

$$\tau = \frac{1}{||g||^2}\int_{-\infty}^{\infty} t|g_{\xi,\tau}(t)|^2 dt \tag{II.2}$$

The Fourier transformation $\hat{g}$ of g is also symmetric. So, the Fourier transform of $g_{\xi,\tau}(t)$

$$\hat{g}_{\xi,\tau}(\omega) = \hat{g}(\omega-\xi)e^{-2\pi i(\omega-\xi)\tau} \tag{II.3}$$

is a translation by ξ of the frequency window $\hat{g}$. The mean frequency ξ of $\hat{g}_{\xi,\tau}(\omega)$ is given by

$$\xi = \frac{1}{||g||^2}\int_{-\infty}^{\infty} \omega|\hat{g}_{\xi,\tau}(\omega)|^2 d\omega \tag{II.4}$$

This window can be a small analyzing function (small in time length), which envelops higher frequency oscillations in it (Figure II.1). The shape and size of the envelop is determined by the function g, and different oscillations inside the envelop are coming from $e^{-2\pi i\xi t}$. Larger window has more oscillation in it and vise versa.

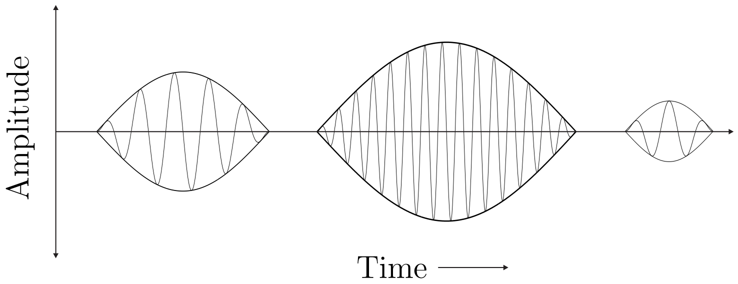


Figure II.1: Windows for STFT with analyzing functions of different frequencies (from (II.1)). Wider windows encompass more oscillations while smaller windows do fewer.

The Short Time Fourier Transform of a function is given by

$$\hat{X}(\xi,\tau) = \int_{-\infty}^{\infty} x(t) g^*_{\xi,\tau} dt = \int_{-\infty}^{\infty} x(t) g(t-\tau) e^{-2\pi i\xi t} dt \tag{II.5}$$

When $g(t) = 1$, STFT reduces to Fourier Transform. The energy of the function is conserved in time-frequency domain.

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |\hat{X}_{\xi,\tau}|^2 d\tau d\xi \tag{II.6}$$

Reconstruction of $x(t)$ is

$$x(t) = \frac{1}{||g||^2} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{X}(\xi,\tau) g(t-\tau) e^{2\pi i\xi t} d\tau d\xi \tag{II.7}$$

This reconstruction formula has redundancy because $g_{\xi,\tau}$ is redundant in $L^2(\mathbb{R})$.

II.1.2 Time-Frequency Resolution for STFT

Let σ_t is the standard deviation of $|g|^2$ and also the length of the window in time domain. Probability density of t is $\frac{1}{||g||^2}|g_{\xi,\tau}(t)|^2$. Then the variance of $|g|^2$ is

$$\sigma_t^2 = \frac{1}{||g||^2}\int_{-\infty}^{\infty}(t-\tau)^2|g_{\xi,\tau}(t)|^2 dt \tag{II.8}$$

Since g is even, the energy of $g_{\xi,\tau}(t) = g(t-\tau)e^{2\pi i\xi t}$ is centered at τ over the interval of σ_t. Therefore the variance of $|g|^2$ becomes

$$\sigma_t^2 = \frac{1}{||g||^2}\int_{\infty}^{\infty} t^2|g(t)|^2 dt \tag{II.9}$$

Equation (II.9) shows that width of the time window is independent of τ and ξ.

σ_ω is the standard deviation of $|\hat{g}|^2$ and also the spread of the window in frequency domain. Probability density of ω in the window is $\frac{1}{||g||^2}|g_{\xi,\tau}(\omega)|^2$. Then variance of $|\hat{g}|^2$ is

$$\sigma_\omega^2 = \frac{1}{||g||^2}\int_{-\infty}^{\infty}(\omega-\xi)^2|g_{\xi,\tau}(\omega)|^2 d\omega \tag{II.10}$$

Since $\hat{g}$ is symmetric, energy of $\hat{g}_{\xi,\tau}(\omega)$ is centered at ξ over the interval of σ_ω. Like σ_t, σ_ω for STFT is also independent of τ and ξ.

$$\sigma_\omega^2 = \frac{1}{||g||^2}\int_{\infty}^{\infty}\omega^2|g(\omega)|^2 d\omega \tag{II.11}$$

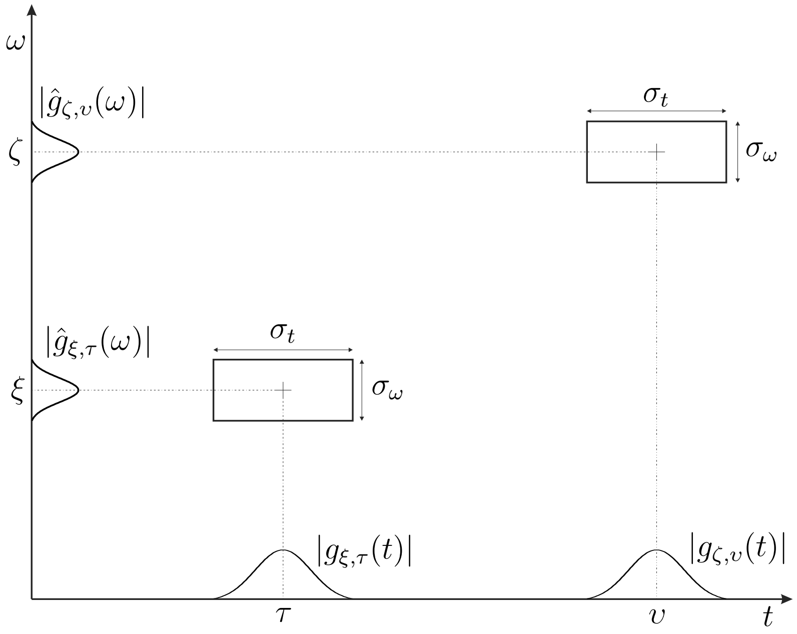


Figure II.2: Time-frequency boxes representing the energy spread of two windows for different time-frequency localization. Window sizes do not change over time-frequency plane, so time-frequency resolution stay the same over the plane.

In the time-frequency plane, this window is a rectangle, centered at (ξ, τ). Time width and frequency width are σ_t and σ_ω respectively. For STFT, they are independent of τ and ξ (equations (II.9) and (II.11)). So widths of the boxes stay the same over the time-frequency plane regardless of their positions. Sizes of these rectangles are equal over the entire time-frequency plane (Figure II.2), because they are related by Second Order Moment Heisenberg's Inequality [4, 19, 20].

$$\sigma_t \sigma_\omega \geq \frac{1}{4\pi} \tag{II.12}$$

Different values appears on the right hand side of the inequality stated above for different formulas used for Fourier transformation. It is not possible to have high resolution in both time and frequency domain, since they are related by equation (II.12).

II.1.3 Discrete Short Time Fourier Transform

Discrete STFT is derived using the same idea from Discrete Fourier Transform. Any N sample signal is assumed to be periodic with period N. The window function $g[n] = g[-n]$ is an even function of period N. It is translated in time domain by m and modulated by the frequency $\frac{2\pi}{N}k$.

$$g_{l,m}[n] = g[n-m]e^{\frac{2\pi i}{N}ln} \tag{II.13}$$

Its Fourier transform is

$$\hat{g}_{l,m}[k] = \hat{g}_{k-l}e^{-\frac{2\pi i}{N}(k-l)m} \tag{II.14}$$

In discrete form, the STFT of a signal of period N is given by

$$\hat{X}[l,m] = \sum_{n=0}^{N-1} x[n]g[n-m]e^{-\frac{2\pi i}{N}ln} \tag{II.15}$$

The energy of the function is conserved in discrete time-frequency domain.

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N}\sum_{l=0}^{N-1}\sum_{m=0}^{N-1} |\hat{X}[l,m]|^2 \tag{II.16}$$

The reconstruction formula is given by

$$x[n] = \frac{1}{N}\sum_{m=0}^{N-1} g[n-m]\sum_{l-0}^{N-1} \hat{X}[l,m]e^{\frac{2\pi i}{N}ln} \tag{II.17}$$

Here the reconstruction of $x[n]$ is achieved using N^2 terms, despite $x[n]$ contains only N points. Therefore there is a redundancy in the inverse STFT. In the past few decades, various kinds of window for STFT have been introduced, which are suitable for different purposes [21]. For instance, Gaussian window is better suited for analyzing transient signal, the Hanning and Hann windows are applicable for narrow band, random signals, the Kaiser-Bessel window for differentiating signal components with close frequencies at different amplitudes. Therefore, the choice of the window directly depends on specific application.

II.1.4 Scaling in Short Time Fourier Transform

Time-frequency localization of the window can be varied using scaling factor $s > 0$. Let a dilated version of window g is $g_s(t) = \frac{1}{\sqrt{s}} g(\frac{t}{s})$. To normalize energy of the dilated $g(\frac{t}{s})$, it is multiplied by $\frac{1}{\sqrt{s}}$. For this dilated window, a change of variables in equations (II.9) and (II.11) show that time and frequency width for g_s are $s\sigma_t$ and σ_ω / s respectively. Heisenberg boxes have equal areas in time-frequency plane (due to uncertainty inequality), but time width is dilated by s and frequency width is compressed by s. So time intervals and frequency bands, that is, resolution in both domain can be changed by scaling window function g.

An important thing to notice is, in STFT, since σ_t and σ_ω are independent of τ and ξ, time widths and frequency widths of the boxes remain the same regardless of their different positions in the time-frequency plane. So ratio of time-frequency resolutions remains the same in the plane. This implies that STFT is able to analyze a signal only at a fixed resolution in time and frequency. For wavelets, however, we will see that σ_t and σ_ω are dependent on the center position of the Heisenberg boxes, where these centers are again dependent on the scaling parameter. So, scaling window not only resizes the Heisenberg boxes, but also changes the ratio of time-frequency resolutions over the time-frequency plane. This is the reason why wavelets are able to analyze a signal at different time-frequency resolutions.

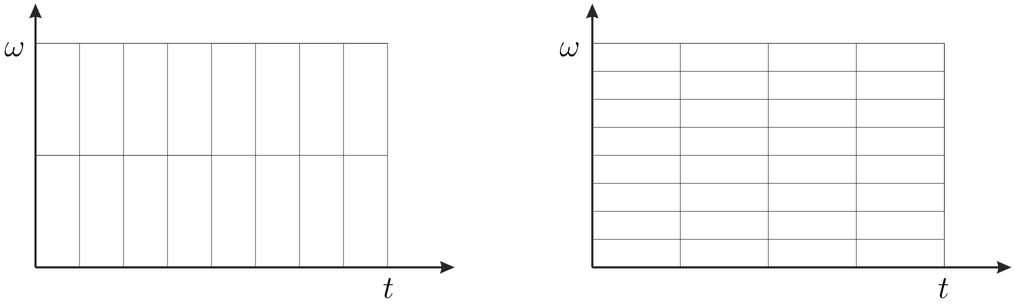


Figure II.3: Low scale gives high time resolution but low frequency resolution, and vise versa. In all the cases, both time and frequency resolutions remain fixed over time-frequency plane once the window is decided.

II.2 Wavelet Analysis

For detecting low frequency components of a signal, analyzing functions with large time support and small bandwidth are needed. On the other hand for detecting high frequency components of a signal, analyzing function with short time support and large bandwidth are needed. Therefore for detecting low and high frequency components of a signal, it is desirable to have both of these types analyzing functions forming the basis. This is not possible with the STFT where time-frequency support of the analyzing functions is fixed. However, this is possible through wavelet analysis. Wavelet analysis is a multi-resolution analysis process, where time-frequency width of the analyzing function is not the same over the time-frequency plane. Unlike STFT, windows in wavelet analysis encompass only one analyzing function. The varying width of the window stretches or compresses the analytic function inside it, thus provides varying frequencies.

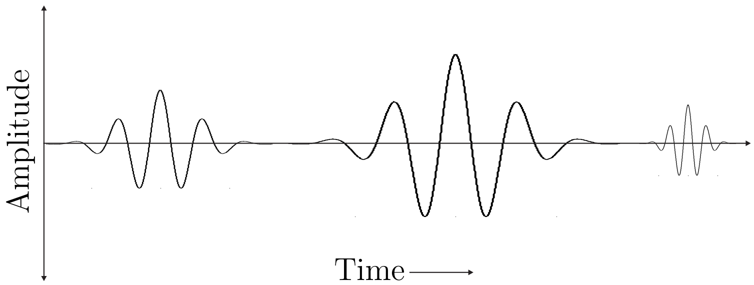


Figure II.4: Wavelets of same shape but different sizes

II.2.1 Wavelet Function

A wavelet is a function $\psi \in L^2(\mathbb{R})$ that satisfies the admissibility condition

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega < +\infty \tag{II.18}$$

Admissibility condition implies that the Fourier transform of $\psi(t)$ vanishes at zero frequency:

$$|\hat{\psi}(\omega)|^2_{\omega=0} = 0 \tag{II.19}$$

So wavelets act like band-pass filters. This also means that their average in time domain is zero

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \tag{II.20}$$

Wavelets are oscillatory in nature. It is centered in the neighborhood of $t = 0$. When ψ is selected for analyzing, scale is fixed to s_0 and the function is called the *mother wavelet* or *base wavelet*. All the other wavelets are obtained by scaling ψ by varying s from s_0 and translating by τ:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right), \quad s > 0,\ \tau \in \mathbb{R} \tag{II.21}$$

$\psi\left(\frac{t-\tau}{s}\right)$ is multiplied by $\frac{1}{\sqrt{s}}$ to normalize the energy. The center frequency of $\hat{\psi}$ is

$$\eta = \frac{1}{||\psi||^2}\int_{-\infty}^{\infty}\omega|\hat{\psi}(\omega)|^2 d\omega \tag{II.22}$$

Fourier transform of $\psi_{s,\tau}(t)$ is a dilation of $\hat{\psi}$ by $\frac{1}{s}$.

$$\hat{\psi}_{s,\tau}(\omega) = \sqrt{s}\hat{\psi}(s\omega)e^{2\pi i\omega\tau} \tag{II.23}$$

So center frequency $\psi_{s,\tau}(t)$ is η/s.

The wavelet transform of $x(t)$ at time τ and scale s is [22]:

$$Wx(s,\tau) = \int_{-\infty}^{\infty} x(t)\frac{1}{\sqrt{s}}\psi^*\left(\frac{t-\tau}{s}\right)dt \tag{II.24}$$

where ψ^* is the complex conjugate of ψ. Wavelet transformation is a linear transformation. In a convolution product form, it is given by:

$$Wx(s,\tau) = f \star \bar{\psi}_s(\tau) \tag{II.25}$$

with

$$\bar{\psi}_s(t) = \frac{1}{\sqrt{s}}\psi^*\left(\frac{-t}{s}\right) \tag{II.26}$$

As long as wavelets satisfy the admissibility condition, energy of $x(t)$ is conserved in wavelet domain.

$$\int_{-\infty}^{+\infty}|f(t)|^2 dt = \int_{-\infty}^{+\infty}\frac{1}{C_\psi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}|X(s,\tau)|^2 d\tau\frac{ds}{s^2} \tag{II.27}$$

Inverse continuous wavelet transform is given as [23]

$$x(t) = \frac{1}{C_\psi} \int_0^{+\infty} \int_{-\infty}^{+\infty} Wx(s,\tau) \frac{1}{\sqrt{s}} \psi \left(\frac{t-\tau}{s} \right) d\tau \frac{ds}{s^2} \qquad \text{(II.28)}$$

II.2.2 Scaling Function

When $Wx(s,\tau)$ is known for $s > s_0$, then we need another function which will give the remaining information of $x(t)$ from scale s to ∞. This function is called the *scaling function* ϕ (also known as the *father wavelet*), which is an aggregation of wavelets for scales larger than s. Modulus of $\hat{\phi}$ is given by

$$|\hat{\phi}(\omega)|^2 = \int_{s_0}^{+\infty} |\hat{\psi}(s\omega)|^2 \frac{ds}{s} = \int_{\omega}^{+\infty} \frac{|\hat{\psi(\xi)}|^2}{\xi} d\xi \qquad \text{(II.29)}$$

From the admissibility condition (II.18) it can be shown that

$$\lim_{\omega \to 0} |\hat{\phi}(\omega)|^2 = C_\psi \qquad \text{(II.30)}$$

Scaling function can be interpreted by as a low pass filter, and it is denoted by

$$\phi_s(t) = \frac{1}{\sqrt{s}} \phi(\frac{1}{s}) \quad \text{and} \quad \bar{\phi}_s(t) = \phi_s^*(-t) \qquad \text{(II.31)}$$

Low frequency components extracted by ϕ_s are

$$Lx(s,\tau) = \left\langle x(t), \frac{1}{\sqrt{s}} \phi \left(\frac{t-\tau}{s} \right) \right\rangle = x \star \bar{\phi}_s(\tau) \qquad \text{(II.32)}$$

With the scaling function, inverse continuous wavelet transform can be given as

$$x(t) = \frac{1}{C_\psi} \int_0^{s_0} Wx(s,\tau) \star \psi_s(\tau) \frac{ds}{s^2} + \frac{1}{C_\psi s_0} Lx(s,\tau_0) \star \phi_{s_0}(\tau) \qquad \text{(II.33)}$$

II.2.3 Time-Frequency Resolution in Wavelet Domain

With change of variable $v = \frac{t-\tau}{s}$, variance of $|\psi_{s,\tau}|^2$ is given by

$$s^2\sigma_t^2 = \frac{1}{||\psi_{s,\tau}||^2}\int_{-\infty}^{\infty}(t-\tau)^2|\psi_{s,\tau}(t)|^2dt \tag{II.34}$$

Spread of $\hat{\psi}_{s,\tau}$ around η/s is

$$\frac{\sigma_\omega^2}{s^2} = \int_{-\infty}^{\infty}(\omega-\eta/s)^2|\hat{\psi}_{s,\tau}(\omega)|^2d\omega \tag{II.35}$$

So for wavelets, σ_t and σ_ω both are dependent on scale s and center time τ.

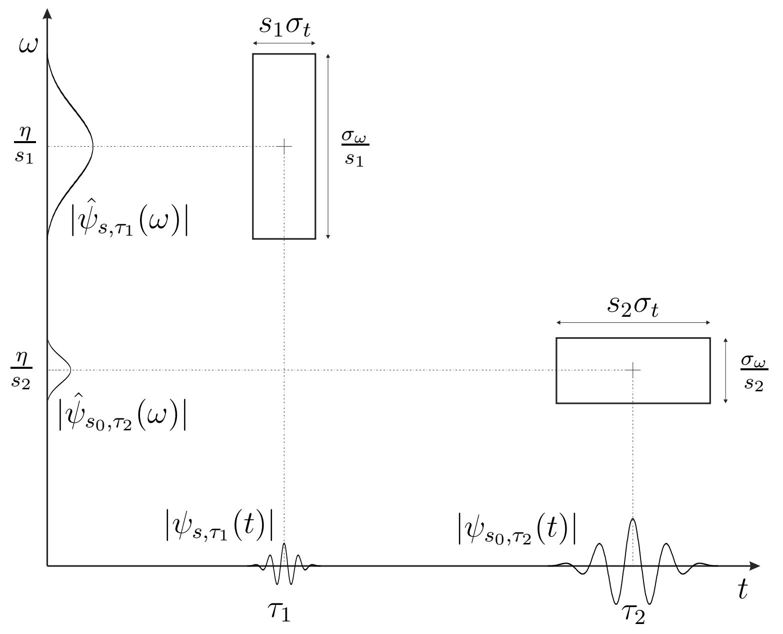


Figure II.5: Time-Frequency Boxes representing the energy spread for different time-frequency localization in wavelet domain. Different sizes of the windows give different time-frequency resolution over the plane.

According to uncertainty inequality, area of Heisenberg's boxes centered at $(\tau, \eta/s)$ is $\sigma_t\sigma_\omega$, which are equal over the time-frequency plane. But widths and heights of boxes $(\sigma_t s)$ and $(\frac{\sigma_\omega}{s})$ respectively are dependent on their centers (equations (II.34) and (II.35)), which are again dependent on the scaling parameter. So shapes of the boxes are different on the time-frequency plane, some are narrow in high frequency region, some are wide in low frequency region (Figure II.5). This implies, for wavelets, time-frequency resolutions vary over time-frequency plane. Lower scale decreases time spread but increases frequency support, which is shifted towards higher frequency. So at lower scale, frequency band of the wavelet gets larger, and time interval gets smaller. A narrow window at high frequency is able to provide better time localization, which is required to distinguish sudden changes in the signal. At higher scale, the time interval is large and the frequency band (shifted towards low frequency) is small in size. A wider window at low frequency region is providing better frequency localization, which is required to detect low frequency components, that is the trend of the signal. Narrow windows at high frequency region are providing high time resolution which is required to detect sudden changes of a signal. This shows that, in wavelet transformation, time and frequency resolution are inversely and directly proportional to the scaling parameter, respectively.

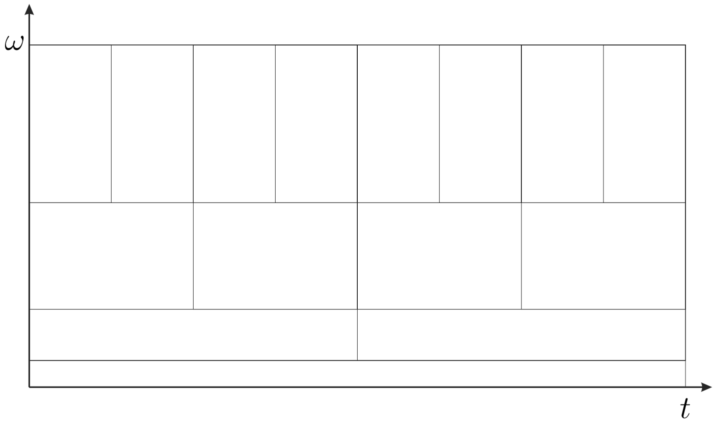


Figure II.6: Multi-resolution of time and frequency

Several commonly used wavelets for continuous wavelet transformation are shown in Appendix B.1.

II.2.4 Discrete Wavelet Transform (DWT)

Continuous variation of scale parameter s and translation parameter τ result in redundant representation of a function in wavelet domain. This redundancy can be exploited for signal denoising or extracting signal features, but they come at a cost of high computation time and memory size. For data compression, or for a concise representation of the signal in wavelet domain, orthogonal wavelet transformation is essential. The approach towards the orthogonality is to use discrete scaling and translation parameters.

In *Discrete Wavelet Transformation* (DWT), the scaling parameter s and translation parameter τ take only integer values. s is taken as integer power of some fixed $s_0 > 1$, i.e. [24]

$$s = s_0^j, \quad , s_0 > 1, \ j \in \mathbb{Z} \tag{II.36}$$

We saw that in wavelet domain, both time and frequency resolution are dependent on scaling parameter s. So, discretization of τ is dependent on j. Narrow Heisenberg's boxes (Figure II.5) mean lower scale and lower value of j, and these boxes will have to slide with smaller steps to cover the entire time span, so they will have a small translation parameter τ. For larger scale (large value of j), wider boxes have to slide with larger step to cover the time span, so they have a larger translation parameter. So discrete translation parameter is proportional of j and it is given by

$$\tau = k\tau_0 s_0^j, \quad \tau_0 > 0, \ j, k \in \mathbb{Z} \tag{II.37}$$

From equation (II.21), discrete wavelet family is

$$\psi_{k,j} = \frac{1}{\sqrt{s_0^j}} \psi \left(\frac{t - k\tau_0 s_0^j}{s_0^j} \right) \tag{II.38}$$

The discrete wavelet transform of $x(t)$ is given as

$$Wx(j,k) = \langle x(t), \psi_{j,k}(t) \rangle = \frac{1}{\sqrt{2^j}} \int_{\infty}^{+\infty} x(t)\psi^* \left(\frac{t - k\tau_0 s_0^j}{s_0^j} \right) dt \tag{II.39}$$

If the wavelet coefficients exist and they can completely describes the signal, that is, if they can be used for perfect reconstruction of the signal, then the following must hold:

$$A||x(t)||^2 \leq \sum_{j,k} |x(t), \psi_{j,k}(t)|^2 \leq B||x(t)||^2, \quad A, B \in \mathbb{R}^+ \tag{II.40}$$

Equation (II.40) is called the *wavelet frame* [23]. The bounds A, B depend on the scaling and translation parameters and the base wavelet that has been selected. If $A = B$, then the wavelet frame called *tight frame* and the reconstruction formula is given by

$$x(t) = \frac{1}{A} \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} Wx(j,k)\psi_{j,k}(t) \tag{II.41}$$

If $A \neq B$ but their difference is not large, then

$$x^{'}(t) = \frac{2}{A+B} \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} Wx(j,k)\psi_{j,k}(t) \tag{II.42}$$

Difference between $x(t)$ and $x^{'}(t)$ becomes small as B/A approaches unity. To exclude redundancy from the reconstruction formula of $x(t)$, it is necessary to have orthogonal wavelet bases.

II.3 Orthogonal Wavelet Bases

Dyadic discretization of with $s_0 = 2$ and $\tau_0 = 1$ creates a family of wavelets which constitute an orthogonal bases for $L^2(\mathbb{R})$ within the tight wavelet frame, characterized by $A = 1$.

$$\psi_{j,k} = 2^{-j/2}\psi\left(2^j t - k\right) \tag{II.43}$$

$$\phi_{j,k} = 2^{-j/2}\psi\left(2^j t - k\right) \tag{II.44}$$

The theoretical foundation for constructing orthogonal base wavelet is the *multiresolution analysis*.

II.3.1 Multiresolution Analysis (MRA)

MRA of $L^2(\mathbb{R})$ consists of successive approximation subspaces $V_j, j \in \mathbb{Z}$. Their properties are

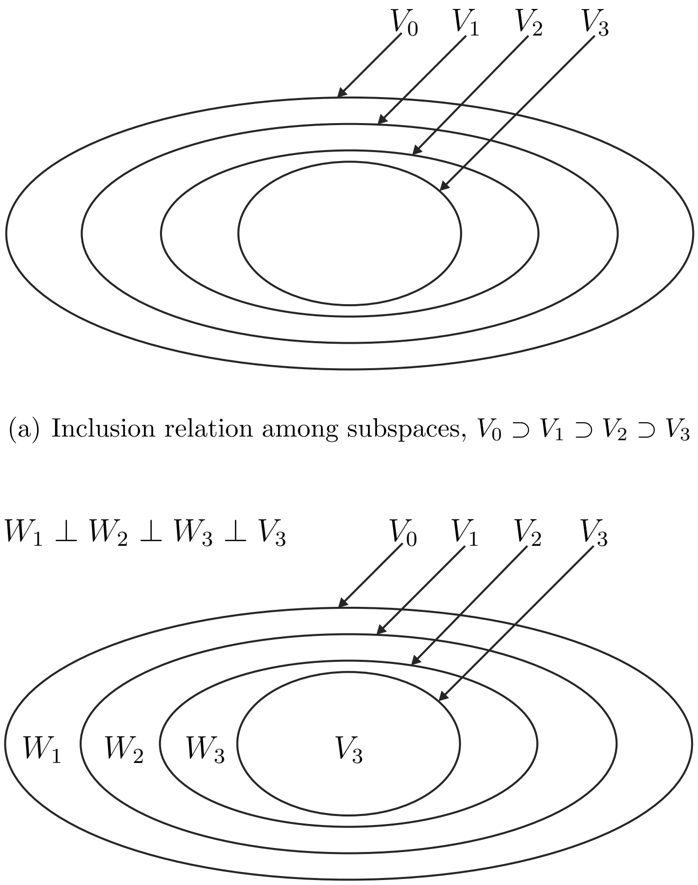
1. Monotonicity: $\cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \cdots$. This indicates the successive inclusion relationship of the closed subspaces.

2. Completeness: $\bigcap\limits_{j\in\mathbb{Z}} V_j = 0$, $\overline{\bigcup\limits_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$. This implies all the subspaces form a complete $L^2(\mathbb{R})$.

3. Dilation regularity: $x(t) \in V_j \Leftrightarrow x(2^j t) \in V_0$. This indicates the multiresolution aspect of the closed subspaces.

4. Translation invariance: $x(t) \in V_j \Rightarrow x(t - n) \in Vj0$.

5. Existence of orthogonal basis: There exists a scalar function $\phi(t) \in V_0$, so that $\{\phi(t - n)_{n\in\mathbb{Z}}\}$ form an orthogonal basis for V_0, that is, $V_0 = \overline{\operatorname{span}\phi(t - n)}$.



(a) Inclusion relation among subspaces, $V_0 \supset V_1 \supset V_2 \supset V_3$

(b) Ortogonal wavelet subspaces

Figure II.7: Multiresolution analysis of $L^2(\mathbb{R})$ into subspaces.

From Figure II.7(a), V_j have the inclusion property and they are not orthogonal, so the family of scalar functions $\phi_{j,k}(t)$ cannot be used as orthogonal bases of $L^2(\mathbb{R})$. W_j are the orthogonal complement of V_j in V_{j-1}, as shown in Figure II.7(b). These subspaces are related as

$$V_{j-1} = V_j \oplus W_j \qquad \text{(II.45)}$$

and

$$W_j \perp W_{j'}, \quad \text{for } j \neq j^{'} \qquad \text{(II.46)}$$

For $j < J$,

$$V_j = V_J \oplus \bigoplus_{k=0}^{J-j-1} W_{J-k} \tag{II.47}$$

W_j are orthonormal, and they form the $L^2(\mathbb{R})$.

$$L^2(\mathbb{R}) = \bigoplus_{j\in\mathbb{Z}} W_j \tag{II.48}$$

W_j also have the scaling property of V_j, that is

$$x(t) \in W_0 \Leftrightarrow x(2^{-j}t) \in W_j \tag{II.49}$$

So, if $W_{0,k},\ k \in \mathbb{Z}$ for is a set of orthonormal bases of $L^2(\mathbb{R})$, then $\{\psi_{j,k} = 2^{-j/2}\psi(2^{-j}t-k),\ k \in \mathbb{Z}\}$ is also a family of orthonormal bases of scale 2^j. So, all the collections of $\{\psi_{j,k},\ k, j \in \mathbb{Z}\}$ form the sets of orthonormal bases for $L^2(\mathbb{R})$.

II.3.2 Orthogonal Wavelet Transform

According to MRA of V_0 space, it can be decomposed as follows:

$$V_0 = V_1 \oplus W_1 = V_2 \oplus W_2 \oplus W_1 = V_3 \oplus W_3 \oplus W_2 \oplus W_1 = \cdots \tag{II.50}$$

$\phi_{j,k}$ is the set of orthonormal bases of the space V_j, so if function $x(t)$ is projected onto V_j, then

$$x_a^j(t) = \sum_k a_{j,k}\phi_{j,k}(t) \tag{II.51}$$

$a_{j,k}$ is called the scaler coefficients, which is given by

$$a_{j,k} = \langle x(t), \phi_{j,k}(t)\rangle \tag{II.52}$$

Similarly, if the signal $x(t)$ is projected onto space W_j, whose orthonormal base is $\psi_{j,k}$, then

$$x_d^j(t) = \sum_k d_{j,k}\psi_{j,k}(t) \tag{II.53}$$

$d_{j,k}$ is called the wavelet coefficient, which is given by

$$d_{j,k} = \langle x(t), \psi_{j,k}(t)\rangle \tag{II.54}$$

So, if $x(t)$ is decomposed up to $j = J$ and V_J and W_j are expressed in the terms of their orthonormal bases, then

$$x(t) = \sum_{j=\infty}^{J}\sum_{k=-\infty}^{+\infty} d_{j,k}\psi_{j,k}(t) + \sum_{k=-\infty}^{+\infty} a_{j,k}\phi_{j,k}(t) \tag{II.55}$$

Equation (II.55) is equivalent to equation (II.41) when $A = B = 1$. This is the case when the wavelet bases are orthonormal [6]. Equation (II.55) is the perfect reconstruction of $x(t)$ without redundancy and equations (II.52) and (II.54) are the orthogonal forward scalar and wavelet transform. Different kind of discrete orthogonal wavelets and their scalar functions are described in Appendix B.2.

II.3.3 Frequency Bands and Time Intervals for Orthogonal Wavelets

Since for orthogonal wavelets, the scale is varied in a logarithmic fashion (integer power of 2), frequency range and time span is also divided in the same way. Say a signal has 2^J discrete points and it is sampled at a constant f_s frequency. It is decomposed with a wavelet which is supported on l discrete points. At j scale, $\{\psi_{j,k}\}$ will be have a support on $2^{j-1}l$ points and it will translate with the same $2^{j-1}l$ number of points as its step. At a certain scale, translating functions do not have any overlap onto each other, since they are orthogonal to each other. $j = 1$

is the scale when we have the smallest time step of the bases functions and finest time resolution. Functions at this scale also project the function onto the coarsest frequency band, which is $\left[f_s/2 \ : \ \frac{1}{2}f_s/2\right]$ ($f_s/2$ is the maximum observable frequency, according to Nyquist sampling rate). As j increases, support of the wavelet doubles, and the frequency band of projection halves. At scale j, the frequency band is $\left[2^{-(j-1)}f_s/2 \ : \ 2^{-j}f_s/2\right]$ and the support, that is the translation step of $\{\psi_{j,k}\}$ is $2^{j-1}l$ points. When $j = J$, support of $\{\psi_{j,k}\}$ and $\{\phi_{j,k}\}$ is $2^{J-1}l$ points. Frequency band of for $\{\psi_{J,k}\}$ is $\left[2^{-(J-1)}f_s/2 \ : \ 2^{-J}f_s/2\right]$ and for $\{\phi_{J,k}\}$ it is $\left[2^{-J}f_s/2 \ : \ 0\right]$.

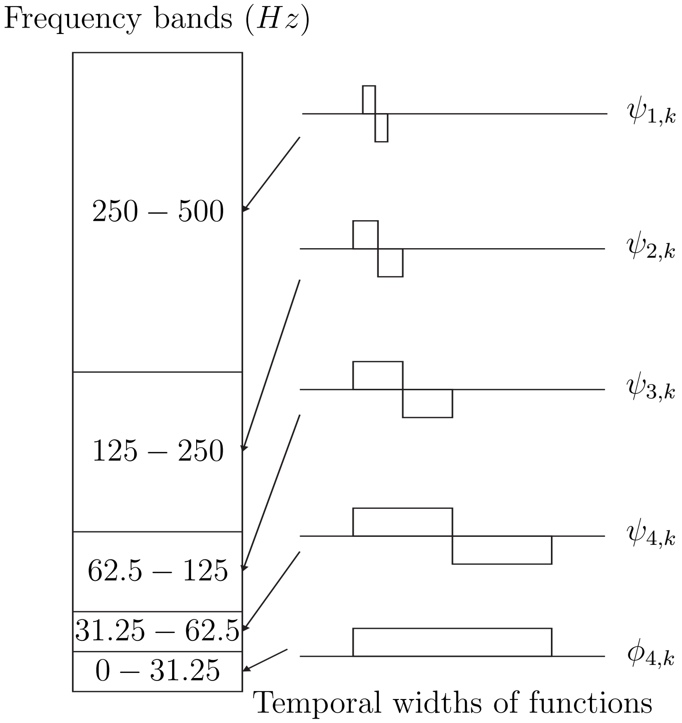


Figure II.8: For a signal sampled at 1000 Hz, temporal widths of Haar functions and their different frequency bands of projections are shown for a decomposition up to $j = 4$. Both the size of frequency band and width of the functions vary dyadically with the change of j.

In the following chapters, Haar matrix is used for solving linear differential equations and for optimal control. Before proceeding to them, a brief introduction of Haar wavelet and how it decomposes piecewise constant sequences is given in the next section.

II.3.4 Decomposition of a signal into Haar Wavelets

The oldest function ψ for which $\{\psi_{j,k}\}$ form an orthonormal basis for $L^2(\mathbb{R})$ is the Haar function. It was first introduced by Alfred Haar in 1910 [3]. It is one of the earliest example of wavelet transform which is dyadic, orthonormal and compact in support. It is also the simplest orthogonal function in mathematical expression.

$$\psi(t) = \begin{cases} 1 & \text{for } 0 \leq t < \frac{1}{2} \\ -1 & \text{for } \frac{1}{2} \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \tag{II.56}$$

Its scalar function ϕ is defined by

$$\phi(t) = \begin{cases} 1 & \text{for } 0 \leq t < \frac{1}{2} \\ 1 & \text{for } \frac{1}{2} \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \tag{II.57}$$

For a sequence with $N = m2^s$ $(N, m, s \in \mathbb{Z})$ piecewise constant points, level of decomposition with Haar wavelet is s.

The signal can be represented in terms of Haar wavelets as

$$x(t) = \sum_{k=0}^{m-1} \phi_{s,k}(t) + \sum_{j=1}^{s} \sum_{k=0}^{N/2^s-1} d_{j,k}\psi_{j,k}(t) \tag{II.58}$$

In rest of the report m will be assumed as 1 for simplification. If the sequence with N points is denoted by a row vector x and its corresponding Haar coefficients by a row vector X, then in matrix form, the Haar representation of sequence x is given by

$$x = XH_N \tag{II.59}$$

where H_N is the $(N \times N)$ normalized Haar matrix of order N. The recursive formula for Haar matrix is

$$H_N = \begin{bmatrix} H_{\frac{N}{2}} \otimes \phi \\ \\ I_{\frac{N}{2}} \otimes \psi \end{bmatrix} = \begin{bmatrix} H_{\frac{N}{2}} \otimes \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \\ I_{\frac{N}{2}} \otimes \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \end{bmatrix} \tag{II.60}$$

where $H_1 = 1$, I_N is the identity matrix of order N. For $N = 4$, it is

$$H_4 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} & \frac{-1}{2} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \tag{II.61}$$

X is obtained by

$$X = xH_N^{-1} \tag{II.62}$$

Since H_N is normalized, $H_N^{-1} = H_N^T$. So, X is given by

$$X = xH_N^T \tag{II.63}$$

Lets assume a sequence for $N = 4$ as

$$x = \begin{bmatrix} -3 & 0 & 4 & 1 \end{bmatrix}$$

Its Haar coefficients are

$$X = \begin{bmatrix} -3 & 0 & 4 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{-1}{\sqrt{2}} & 0 \\ \frac{1}{2} & \frac{-1}{2} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{2} & \frac{-1}{2} & 0 & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -4 & -\frac{3}{\sqrt{2}} & \frac{3}{\sqrt{2}} \end{bmatrix}$$

Each row of H_N represents a basis function, and its corresponding coefficients represent the projection of x on these functions. The signal can be reconstructed by simply multiplying the coefficients with the basis functions and adding them.

$$\begin{aligned} &\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad \times 1 \\ +&\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} & \frac{-1}{2} \end{bmatrix} \times (-4) \\ +&\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 \end{bmatrix} \times \left(\frac{-3}{\sqrt{2}} \right) \\ +&\begin{bmatrix} 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \times \left(\frac{3}{\sqrt{2}} \right) \\ =&\begin{bmatrix} -3 & 0 & 4 & 1 \end{bmatrix} \end{aligned}$$

Last two coefficients together represents the signal at frequency range $[f_s/2 \ : \ \frac{1}{2} f_s/2]$ Hz. Second coefficient represents the signal in $[\frac{1}{2} f_s/2 \ : \ \frac{1}{4} f_s/2]$ Hz, and the first coefficient in $[\frac{1}{4} f_s/2 \ : \ 0]$ Hz, f_s is the sampling frequency. Time evolution of the signal at these frequency ranges are obtained as:

$$
\begin{aligned}
[f_s/2\ :\ \frac{1}{2}f_s/2]\ \text{Hz}: &\quad \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 \end{bmatrix} \times \left(\frac{-3}{\sqrt{2}}\right) + \begin{bmatrix} 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \times \left(\frac{3}{\sqrt{2}}\right) \\
&= \begin{bmatrix} \frac{-3}{2} & \frac{3}{2} & \frac{3}{2} & \frac{-3}{2} \end{bmatrix} \\
[\frac{1}{2}f_s/2\ :\ \frac{1}{4}f_s/2]\ \text{Hz}: &\quad \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} & \frac{-1}{2} \end{bmatrix} \times (-4) \\
&= \begin{bmatrix} -2 & -2 & 2 & 2 \end{bmatrix} \\
[\frac{1}{4}f_s/2\ :\ 0]\ \text{Hz}: &\quad \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \times (1) \\
&= \begin{bmatrix} 2 & 2 & 2 & 2 \end{bmatrix}
\end{aligned}
$$

Summing the signal at these frequencies bands reconstructs x.

It can also be shown that energy is conserved in wavelet domain.

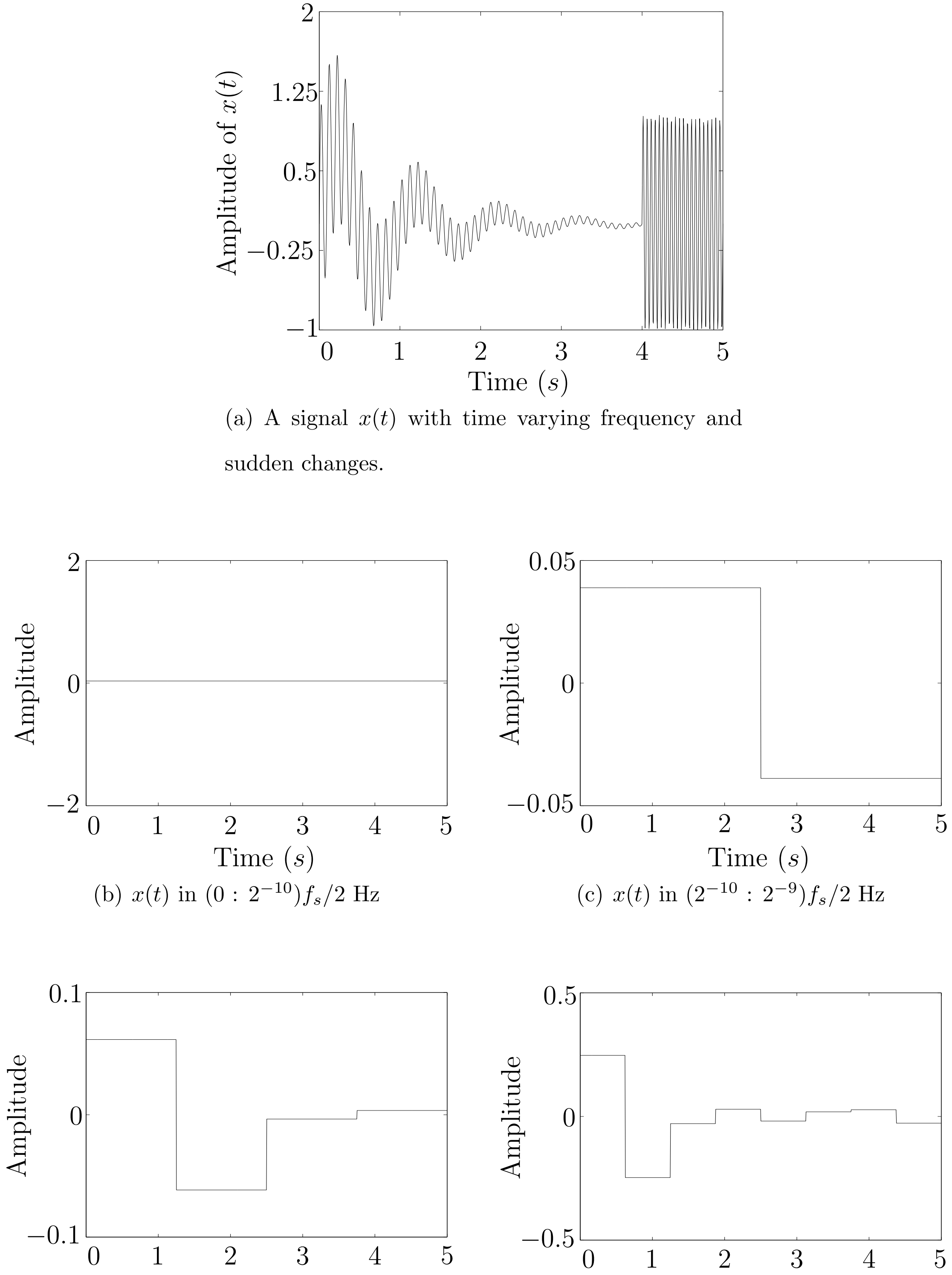
$$(-3)^2 + 0^2 + 4^2 + 1^2 = 26 = 1^2 + (-4)^2 + \left(\frac{-3}{\sqrt{2}}\right)^2 + \left(\frac{3}{\sqrt{2}}\right)^2$$

Rows of the H_N will be denoted by h_i, $i = 1,\ 2,\ \cdots,\ N$, so $H_N = [h_1\ h_2\ \cdots\ h_N]^T$. Columns of H_N will be denoted by H_λ, $\lambda = 1,\ 2,\ \cdots,\ N$. It is notable that

$$x(\lambda) = XH_N(\lambda) \tag{II.64}$$

It implies that each of the columns represents time in wavelet domain.

The signal shown in Figure II.9(a) has $N = 2^{10}$ samples, so for decomposing it with Haar wavelets, maximum level attainable is $s = 10$ (equation (II.58)). Sampling rate is $f_s = 204.6$ Hz. Time evolutions of the signal in all the frequency ranges and the reconstructed signal are shown in Figure II.9.



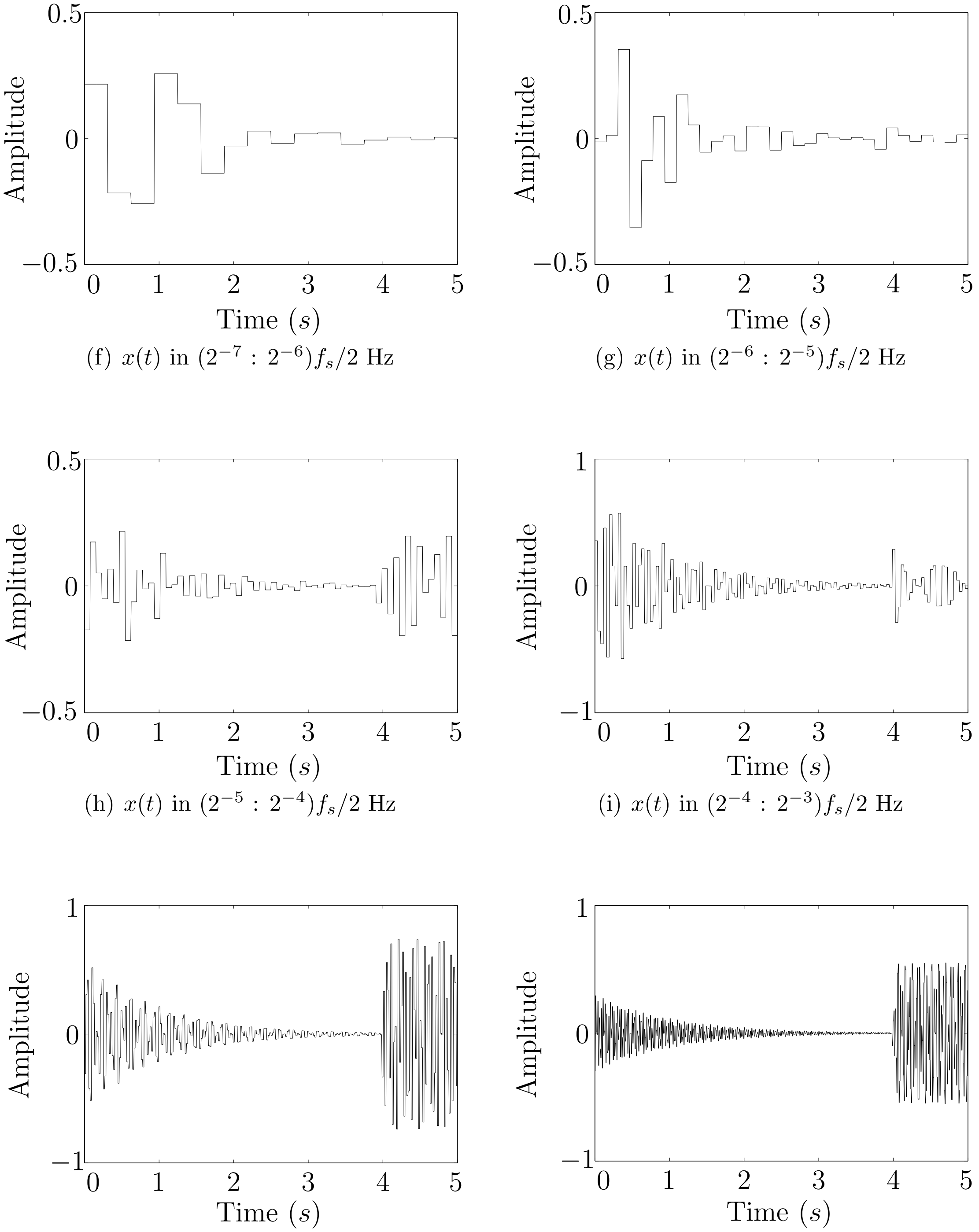
(a) A signal $x(t)$ with time varying frequency and sudden changes.

(b) $x(t)$ in $(0 : 2^{-10})f_s/2$ Hz

(c) $x(t)$ in $(2^{-10} : 2^{-9})f_s/2$ Hz

(d) $x(t)$ in $(2^{-9} : 2^{-8})f_s/2$ Hz

(e) $x(t)$ in $(2^{-8} : 2^{-7})f_s/2$ Hz



(f) $x(t)$ in $(2^{-7} : 2^{-6})f_s/2$ Hz

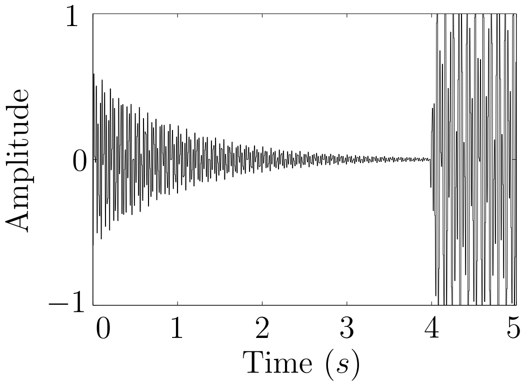
(g) $x(t)$ in $(2^{-6} : 2^{-5})f_s/2$ Hz

(h) $x(t)$ in $(2^{-5} : 2^{-4})f_s/2$ Hz

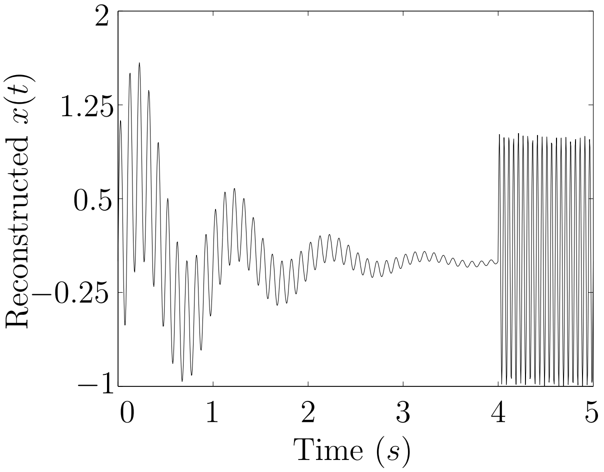
(i) $x(t)$ in $(2^{-4} : 2^{-3})f_s/2$ Hz

(j) $x(t)$ in $(2^{-3} : 2^{-2})f_s/2$ Hz

(k) $x(t)$ in $(2^{-2} : 2^{-1})f_s/2$ Hz



(l) $x(t)$ in $(2^{-1} : 2^{0})f_s/2$ Hz



(m) Reconstructed $x(t)$ from Figures II.9(b)-II.9(l)

Figure II.9: Time evolution of $x(t)$ at different frequency ranges, starting from the frequency band at lowest frequency region.

Now that we know how Haar wavelets form orthonormal bases and how it decomposes a function, we can move into solving linear differential equations via Haar wavelets [9], which will pave the way for optimal control of linear systems without constraints [16]. In the following chapter, derivation of Haar integral matrix, solution of LTI systems [9] and optimal control of unconstrained LTI systems via Haar wavelets [16] will be shown. It is important to have these derivations presented, because later we have to come back to these formulas to make modifications in them and apply them in Model Predictive Control.

CHAPTER III

OPTIMAL CONTROL OF LINEAR SYSTEMS VIA HAAR WAVELETS

This chapter proposes a method of solving the Model Predictive Control problem for unconstrained linear systems using the Haar domain. In section III.1 the optimal control problem statement and various methods for solving them are presented. Section III.2 contains how orthogonal functions can be used for solving linear differential equations [9]. Section III.3 contains Haar wavelet integral matrix from [9] which is useful in solving differential equation via Haar wavelets. Section III.4 reviews how differential equations are solved using Haar wavelets [9,16]. Using Haar domain solution, a linear quadratic cost function is minimized in Haar domain to solve an unconstrained open loop optimal control problem, which is reviewed in section III.5 [16]. Section III.6 proposes the idea of using this Haar domain optimizer in a Receding Horizon fashion to construct a Model Predictive Controller (MPC) in Haar domain. In MPC, only the first control action from the optimized control horizon is applied to the plant. So in section III.8 this Haar domain optimizer is configured in such a manner that it only produces those Haar coefficients which are used to construct the first control action only.

III.1 Optimal Control Problem

Optimization of a dynamic control system aims to find the optimal control by minimizing or maximizing a performance index under constraints. Several methods that solve optimal control problems using orthogonal function are discussed in [25].

Transforming optimization problem in wavelets appeared in [9, 15, 26]. The general statement of the optimal control problem is as follows:
Optimize the performance index

$$J = \int_{T_0}^{T_0+T_f} g\left(x(t), u(t), t\right) \tag{III.1}$$

Subjected to

$$\dot{x}(t) = f(x(t), u(t), t), \quad x(T_0) = x_0 \tag{III.2}$$

This problem is solved by one of the following approaches [27]:

- Variational method and Pontryagin's minimum principle (Euler-Lagrange equations).
- Direct methods using parameterization.

These methods are categorized as indirect methods and direct methods [15].

III.1.1 Indirect Methods

Methods based on Euler-Lagrange equations and the variational method convert the optimal control problem into two point boundary value problem. These methods result in an optimal controller which is used to control the closed-loop feedback control input $u(x(t))$.

III.1.2 Direct Method

This method consists of directly substituting the transformed control and states into the performance index without constructing the Hamiltonian for the system. Here the control and state both are approximated by a finite series of known functions with unknown parameters.

The typical direct methods require numerical procedures for solving optimality problems, specially for nonlinear systems and with constraints on states and control. Numerical procedures may produce more than one minima [15], so unique optimal solution is not guaranteed. For unconstrained linear systems though, optimized control action can be found by direct differentiation of the cost function with respect to the control. In the following sections, the control and states are transformed in Haar domain and a linear quadratic cost function is differentiated with respect to the transformed control, which in turn gives an analytic optimal control in Haar domain.

III.2 Orthogonal Functions in Solving Differential Equations

In the last few decades, orthonormal functions have been playing an important role in solving problems such as parameter estimation and optimal control. In this method the differential equation is converted into a set of algebraic equations. Chen and Hsiao [7] were the first to use the approximation method via Walsh function. However, the orthonormal functions which are globally supported face difficulty in solving systems with abrupt changes. Haar wavelet is useful in overcoming this problem, since it vanishes outside a short interval of time (equation (II.56) and (II.57)). The simple analytic form of Haar wavelets makes transformation of control, states and the cost function compact compared to other orthonormal functions. For finding the optimal control, it is necessary to have a relation between the input and the states in Haar domain, which is described by the governing equation of the system. This demands for solving the governing dynamics in Haar domain. Next section discusses the integration in Haar domain.

III.3 Integration of Haar Wavelets

It is necessary to perform integration for solving differential equations. For having a solution of a differential equation in Haar domain, it is necessary to have an integral operational matrix in Haar domain. This Haar integral matrix is used for solving differential equations for different time durations. So this Haar integral matrix is defined on a time duration of 1, so that this matrix can be used by normalizing time durations of various differential equations. For this, λ of H_N will take the values as

$$\lambda = \frac{1}{N},\ \frac{2}{N},\ \cdots,\ 1 \tag{III.3}$$

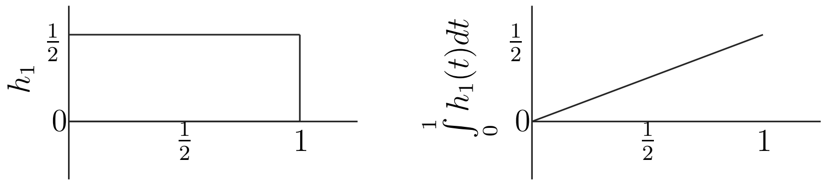
Before giving the formula of the Haar integral matrix, as an example, integrations and piecewise constant approximations of each of the Haar functions of H_4 (each row of H_4) are given [9].

$$\int_0^1 h_1(t)dt = t, \quad 0 \le t < 1 \qquad \approx \frac{1}{8}\begin{bmatrix}\frac{1}{2} & \frac{3}{2} & \frac{5}{2} & \frac{7}{2}\end{bmatrix} \tag{III.4}$$

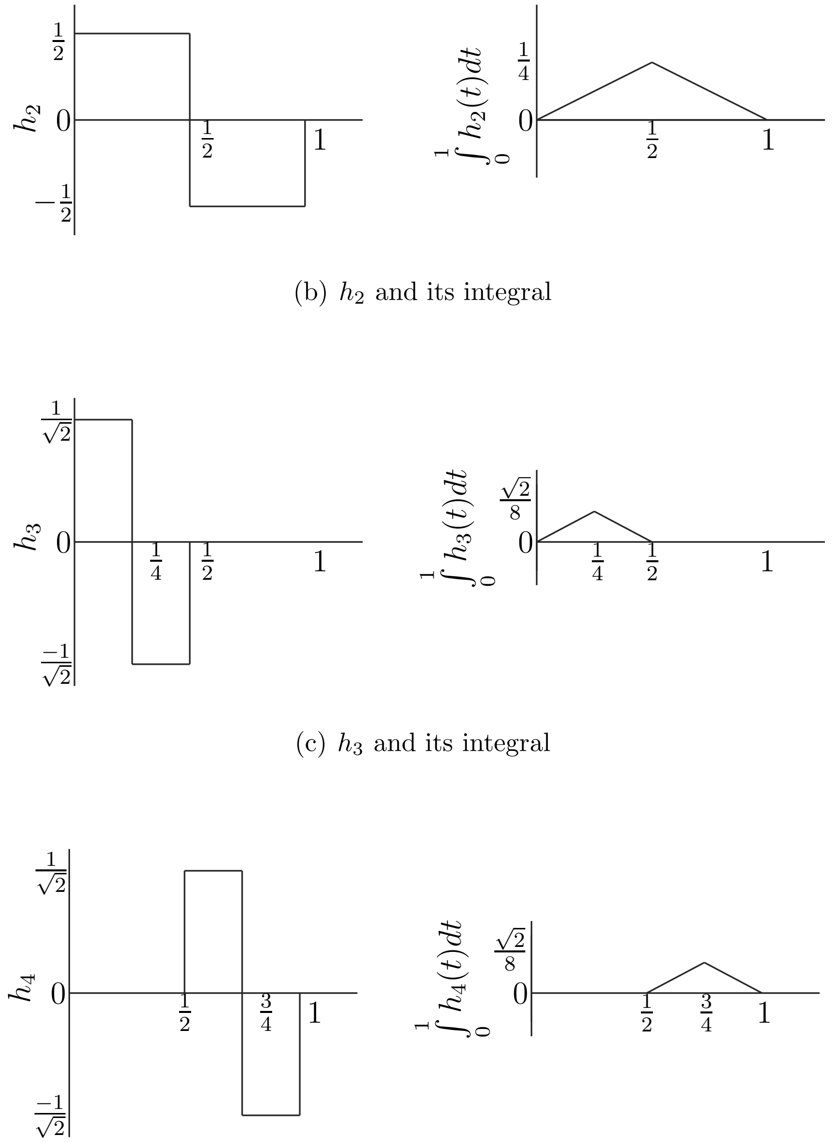
$$\int_0^1 h_2(t)dt = \begin{cases} t, & 0 \le t < \frac{1}{2} \\ 1-t, & \frac{1}{2} \le t < 1 \end{cases} \qquad \approx \frac{1}{8}\begin{bmatrix}\frac{1}{2} & \frac{3}{2} & \frac{3}{2} & \frac{1}{2}\end{bmatrix} \tag{III.5}$$

$$\int_0^1 h_3(t)dt = \begin{cases} t, & 0 \le t < \frac{1}{4} \\ \frac{1}{2}-t, & \frac{1}{4} \le t < \frac{1}{2} \end{cases} \qquad \approx \frac{1}{8}\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0\end{bmatrix} \tag{III.6}$$

$$\int_0^1 h_4(t)dt = \begin{cases} t-\frac{1}{2}, & \frac{1}{2} \le t < \frac{3}{4} \\ 1-t, & \frac{3}{4} \le t < 1 \end{cases} \qquad \approx \frac{1}{8}\begin{bmatrix}0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{bmatrix} \tag{III.7}$$



(a) h_1 and its integral



(b) h_2 and its integral

(c) h_3 and its integral

(d) h_4 and its integral

Figure III.1: Haar functions of H_4 and their integrals, defined over [0 1]s.

Writing equations (III.4) to (III.7) altogether, we obtain

$$\int_0^1 H_4(t)dt \approx \int_0^1 H_4(\lambda)d\lambda \approx \frac{1}{8}\begin{bmatrix} \frac{1}{2} & \frac{3}{2} & \frac{5}{2} & \frac{7}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{III.8}$$

If the integrals of each of the Haar functions are expanded into Haar coefficients P_4, then from equation (II.63)

$$P_4 = \left[\int_0^1 H_4(\lambda) d\lambda \right] H_4^T \tag{III.9}$$

Substituting the value of H_4, P_4 is obtained as

$$P_4 = \frac{1}{2 \times 4} \begin{bmatrix} 4 & -2 & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 2 & 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \end{bmatrix} \tag{III.10}$$

If the columns of H_4 are denoted by some other variable $\gamma = \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, 1$, the integration from 0 to any time $\lambda = \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, 1$ is

$$\int_0^\lambda H_4(\gamma) d\gamma = P_4 H_4(\gamma) \tag{III.11}$$

Following the same procedure, for H_8, the integral matrix P_8 is given by

$$P_8 = \frac{1}{2 \times 8} \begin{bmatrix} (2 \times 8) P_4 & -H_4 \\ H_4^T & \mathbf{0}_4 \end{bmatrix} \tag{III.12}$$

0_4 is the (4×4) null matrix. The recursive formula for P_N is given by [9],

$$P_N = \frac{1}{2N} \begin{bmatrix} 2N P_{N/2} & -H_{N/2} \\ H_{N/2}^T & 0_{N/2} \end{bmatrix} \tag{III.13}$$

where $H_1 = 1$, $H_1^T = 1$, $P_1 = \frac{1}{2}$ and $0_{N/2}$ is $(\frac{N}{2} \times \frac{N}{2})$ null matrix. And also

$$\int_0^{\lambda} H_N(\gamma)d\gamma = P_N H_N(\lambda) \tag{III.14}$$

III.4 Solving Linear Time Invariant Systems via Haar Wavelets

The solution of a first order linear differential equation using Haar wavelets is reviewed from [16]. Let us consider a linear time invariant system described in state space with the following differential equation:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{III.15}$$

$$y(t) = Cx(t) \tag{III.16}$$

$$x(0) = x_0 \tag{III.17}$$

where $x(t)$ is the state vector and $u(t)$ is the input vector. The coefficient matrices A and B are of order $(n \times n)$ and $(n \times l)$, respectively. The initial state is x_0. We are looking for a solution for a finite time interval of $[0 \;\; T_f]$ seconds. If we want to find the solution in Haar domain, we have to rescale $[0 \;\; T_f]$ to $[0 \;\; 1]$ seconds, since the Haar wavelet and its integral matrix are defined over time interval of $[0 \;\; 1]$ seconds; N number of samples are considered in $[0 \;\; T_f]$. Rescaling the time variable in equation (III.15) by

$$t = T_f\lambda \tag{III.18}$$

We get,

$$\dot{x}(\lambda) = T_f Ax(\lambda) + T_f Bu(\lambda) \tag{III.19}$$

Now that the differential equation has been scaled from $[0 \;\; T_f]$ to $[0 \;\; 1]$, instead of integrating in $[0 \;\; t]$, we have to integrate in $[0 \;\; \lambda]$:

$$x(\lambda) - x_0 = T_f A \int_0^{\lambda} x(\gamma) d\gamma + T_f B \int_0^{\lambda} u(\gamma) d\gamma \tag{III.20}$$

We can augment x_0 to x_0^{aug} by padding $N-1$ number of columns, where all the columns are the same x_0 initial state vector.

$$x_0^{aug} = \begin{bmatrix} x_0 & \underbrace{x_0 \; ... \; x_0}_{\text{N-1}} \end{bmatrix} \tag{III.21}$$

So (III.20) becomes

$$x(\lambda) - x_0^{aug}(\lambda) = T_f A \int_0^{\lambda} x(\gamma) d\gamma + T_f B \int_0^{\lambda} u(\gamma) d\gamma \tag{III.22}$$

If X, X_0 and U are Haar coefficients of $x(\lambda)$, $x_0^{aug}(\lambda)$ and $u(\lambda)$, respectively, then by using forward Haar transformation from equation (II.63),

$$x(\lambda) = X H_N(\lambda) \tag{III.23}$$

$$x_0^{aug}(\lambda) = X_0 H_N(\lambda) \tag{III.24}$$

$$u(\lambda) = U H_N(\lambda) \tag{III.25}$$

X_0 is given by

$$X_0 = x_0^{aug} H_N^T \tag{III.26}$$

Since all the columns in x_0^{aug} are the same, X_0 becomes

$$X_0 = \begin{bmatrix} \left(\frac{1}{\sqrt{N}}\right) x_0 & \underbrace{0 \; 0 \; ... \; 0}_{\text{N-1}} \end{bmatrix} \tag{III.27}$$

Now, equation (III.22) can be written in terms of the Haar coefficients as

$$(X - X_0)H_N(\lambda) = T_f AX \int_0^\lambda H_N(\gamma)d\gamma + T_f BU \int_0^\lambda H_N(\gamma)d\gamma \tag{III.28}$$

From (III.14),

$$(X - X_0)\, H_N(\lambda) = T_f AXP_N H_N(\lambda) + T_f BUP_N H_N(\lambda) \tag{III.29}$$

Since it is true for all $H_N(\lambda)$,

$$X - X_0 = T_f AXP_N + T_f BUP_N \tag{III.30}$$

For finding X, vec(.) operator is applied to (III.30), which stacks columns of a matrix in a single column.

$$vec(X) - vec(X_0) = T_f(P_N^T \otimes A)vec(X) + T_f(P_N^T \otimes B)vec(U) \tag{III.31}$$

Applying the property of vec(.) operator from equation (C.1) in Appendix C and after some calculations, we have

$$vec(X) = \left[I_{Nn} - T_f(P_N^T \otimes A)\right]^{-1} \left[T_f(P_N^T \otimes B)vec(U) + vec(X_0)\right] \tag{III.32}$$

$\otimes$ is the Kronecker product and I_{Nn} is the identity matrix of order $(Nn \times Nn)$. In abbreviated form, it is

$$vec(X) = K_1 vec(U) + K_2 vec(X_0) \tag{III.33}$$

where,

$$K_1 = T_f \left[I_{Nn} - T_f (P_N^T \otimes A) \right]^{-1} (P_N^T \otimes B) \tag{III.34}$$

$$K_2 = \left[I_{Nn} - T_f (P_N^T \otimes A) \right]^{-1} \tag{III.35}$$

Equation (III.33) is the transfer function of the linear systems in wavelet domain and K_1 is the state transition matrix. If we write the identity matrix of order N by denoting its rows, then

$$I_n = \begin{bmatrix} I_N(1) \\ I_N(2) \\ \vdots \\ I_N(N) \end{bmatrix} \tag{III.36}$$

Then we can find X from $vec(X)$ by

$$X = \begin{bmatrix} (I_N(1) \otimes I_N)\, vec(X) & (I_N(2) \otimes I_n)\, vec(X) & \cdots & (I_N(N) \otimes I_N)\, vec(X) \end{bmatrix} \tag{III.37}$$

Using (II.59), we can find the whole sequence x from X by

$$x = XH_N \tag{III.38}$$

And $x(\lambda)$ at different time step is given by

$$x(\lambda) = XH_N(\lambda) \tag{III.39}$$

III.4.1 Example of a LTI System solution via Haar Wavelets

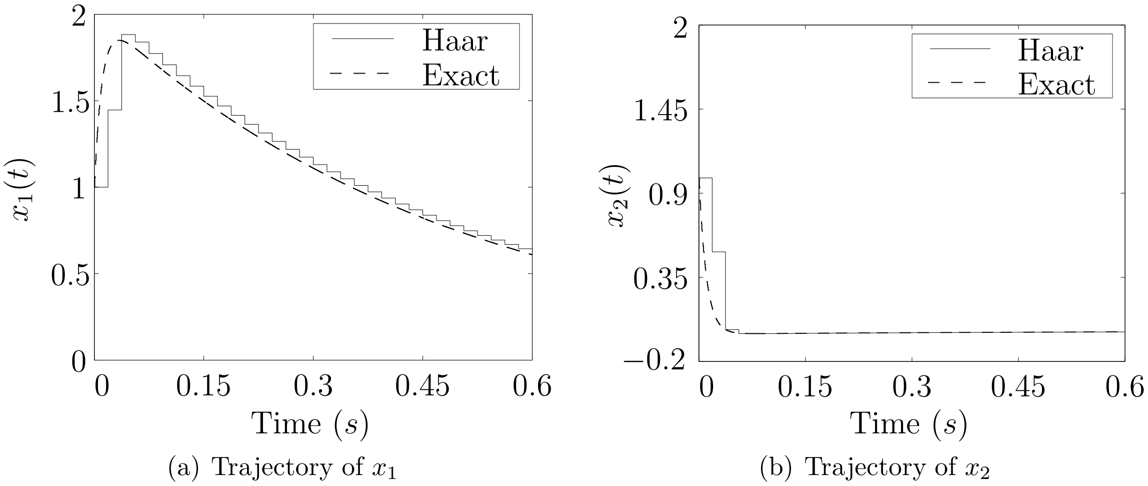
Let the system be

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} -1 & 95 \\ -1 & 97 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \quad x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{III.40}$$

Its analytic solution is given by [28]

$$\begin{aligned} x_1(t) &= \frac{1}{47}\left(95e^{-2t} - 48e^{-96t}\right) \\ x_2(t) &= \frac{1}{47}\left(48e^{-96t} - e^{-2t}\right) \end{aligned}$$

Solution via Haar wavelets are compared with the exact solutions in figure (III.2).



(a) Trajectory of x_1

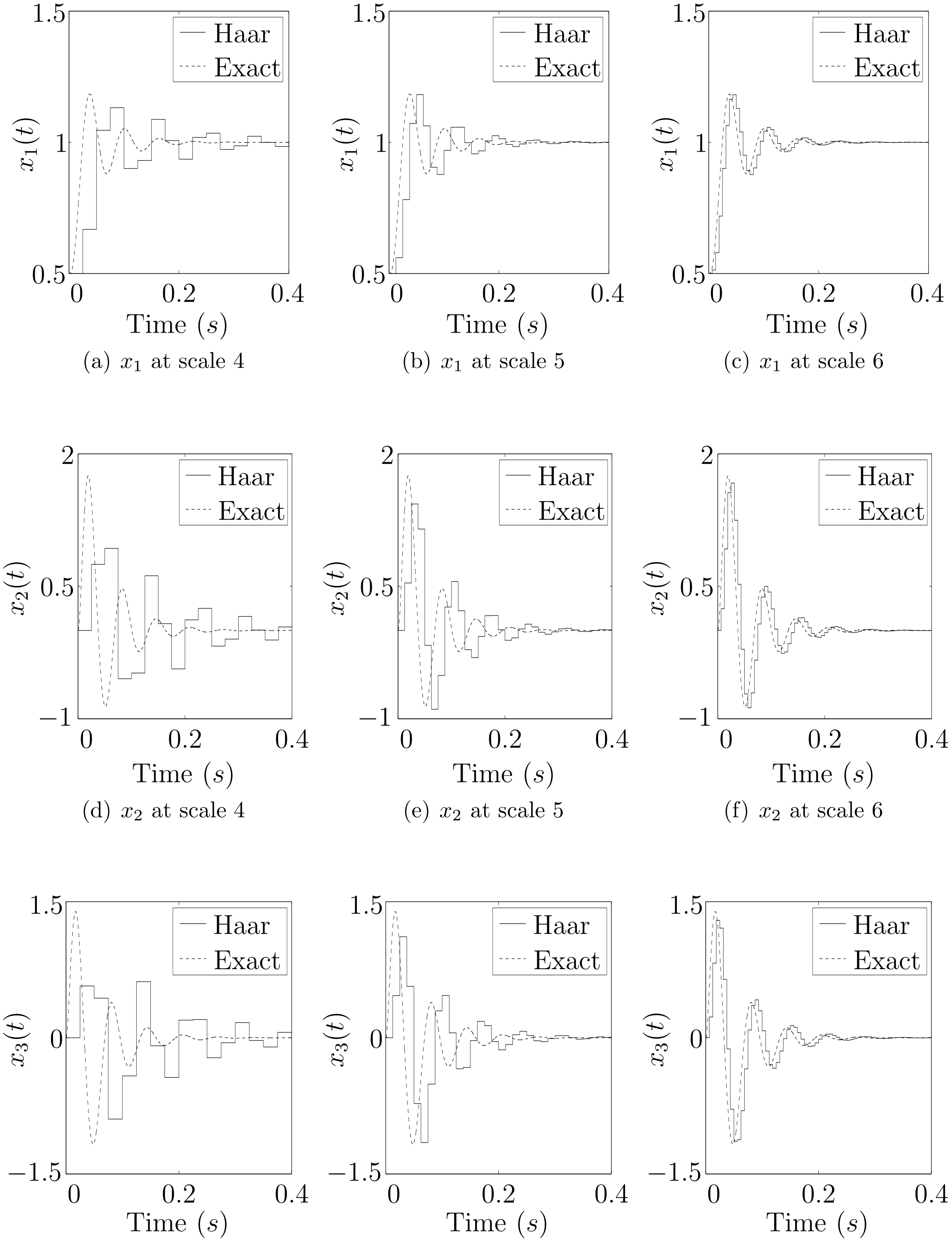
(b) Trajectory of x_2

Figure III.2: Comparison between states from analytic solution and solution via Haar wavelets.

Consider another system [9]

$$\dot{x}(t) = \begin{bmatrix} 0 & 18 & 0 & 0 \\ 0 & 0 & 25 & 250 \\ 0 & 0 & 0 & 250 \\ -390 & -92.25 & 0 & -250 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 390 \end{bmatrix} u(t), \quad x_0 = \begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{III.41}$$

Figure III.3 show the response due to step input. Increasing the scale improves the accuracy of the solution via Haar wavelets.



(a) x_1 at scale 4

(b) x_1 at scale 5

(c) x_1 at scale 6

(d) x_2 at scale 4

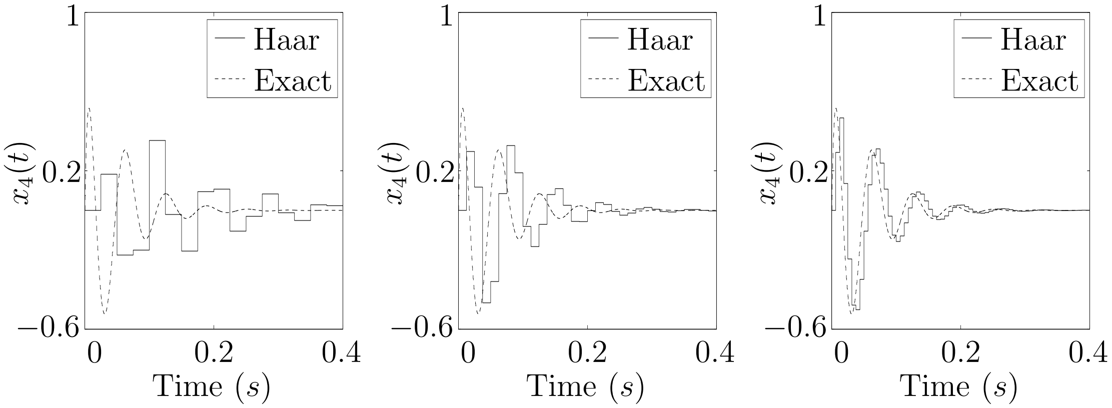
(e) x_2 at scale 5

(f) x_2 at scale 6

(g) x_3 at scale 4

(h) x_3 at scale 5

(i) x_3 at scale 6



(j) x_4 at scale 4 (k) x_4 at scale 5 (l) x_4 at scale 6

Figure III.3: Comparison between states from analytic solution and solution via Haar wavelets.

In both the examples, time duration of $[0 \;\; 0.6]s$ is used. A decomposition level of 5, that is H_5 is used for finding the solutions in Haar domain. Higher resolution gives a gives a response close to the actual response.

III.5 Open Loop Optimal Control via Haar Wavelets

In this section, optimal control via Haar wavelet is reported from [16]. It is a linear optimal open loop control, because the control depends only on the initial states of the plant. The optimal control can be obtained with respect to the optimization of a quadratic cost function as follows:

$$J = \int_0^{T_f} \left[x^T(t)Qx(t) + u^T(t)Ru(t)\right] dt \qquad \text{(III.42)}$$

subjected to

$$\dot{x} = Ax(t) + Bx(t), \quad y(t) = Cx(t), \quad x(0) = x_0 \qquad \text{(III.43)}$$

The selection of weighting matrices Q and R in the performance index is discussed in Appendix D.

Rescaling time with $t = T_f\lambda$ and $[0 \;\; T_f]$ to $[0 \;\; 1]$s, we have

$$J = T_f \int_0^1 \left[x^T(\lambda)Qx(\lambda) + u^T(\lambda)Ru(\lambda)\right] d\lambda \tag{III.44}$$

Using equation (III.23) and (III.25), we can expand the states and inputs into wavelet coefficients. Then the cost function becomes

$$J = T_f \int_0^1 \left[H_N^T(\lambda)X^TQXH_N(\lambda) + H_N^T(\lambda)U^TRUH_N(\lambda)\right] d\lambda$$

According to property from equation (C.3) in Appendix C

$$J = T_f \int_0^1 \left[tr\left(H_N(\lambda)H_N^T(\lambda)X^TQX\right) + tr\left(H_N(\lambda)H_N^T(\lambda)U^TRU\right)\right] d\lambda \tag{III.45}$$

It can be written as

$$J = T_f \left[tr\left(\int_0^1 H_N(\lambda)H_N^T(\lambda)d\lambda X^TQX\right) + tr\left(\int_0^1 H_N(\lambda)H_N^T(\lambda)d\lambda U^TRU\right)\right] \tag{III.46}$$

By defining

$$M = \int_0^1 H_N(\lambda)H_N^T(\lambda)d\lambda \tag{III.47}$$

equation (III.46) can be written as

$$J = T_f \left[tr\left(MX^TQX\right) + tr\left(MU^TRU\right)\right] \tag{III.48}$$

According to property from equation (C.2) in Appendix C

$$J = T_f \left[vec^T(X)(M^T \otimes Q)vec(X) + vec^T(U)(M^T \otimes R)vec(U)\right] \tag{III.49}$$

If we write,

$$V_1 = M^T \otimes Q \tag{III.50}$$

$$V_2 = M^T \otimes R \tag{III.51}$$

Then equation (III.49) is given by

$$J = T_f \left[vec^T(X) V_1 vec(X) + vec^T(U) V_2 vec(U) \right] \tag{III.52}$$

Now we have the cost function in terms of Haar wavelets. Since J is a function of $vec(U)$, for optimal control, J satisfies the following:

$$\frac{\partial J}{\partial vec(U)} = 0 \tag{III.53}$$

According to property from equation (C.4) in Appendix C, equation (III.52) becomes

$$\begin{aligned} \frac{\partial J}{\partial vec(U)} =& T_f \left[\frac{\partial vec^T X}{\partial vec(U)} \frac{\partial}{\partial vec(X)} \left(vec^T(X) V_1 vec(X) \right) \right] \\ &+ T_f \left[\frac{\partial vec^T(U)}{\partial vec(U)} \frac{\partial}{\partial vec(U)} \left(vec^T(U) V_2 vec(U) \right) \right] \end{aligned}$$

Using property from equation (C.5) in Appendix C

$$\frac{\partial J}{\partial vec(U)} = T_f \left[\frac{\partial vec^T(X)}{\partial vec(U)} (V_1 + V_1^T) vec(X) + (V_2 + V_2^T) vec(U) \right] \tag{III.54}$$

Now from equation (III.33)

$$\frac{\partial vec^T(X)}{\partial vec(U)} = \frac{\partial}{\partial vec(U)} \left(vec^T(U) K_1^T + vec^T(X_0) K_2^T \right)$$

Using property from equation (C.6) in Appendix C

$$\frac{\partial vec^T(X)}{\partial vec(U)} = K_1^T \tag{III.55}$$

So, equation (III.54) becomes

$$\frac{\partial J}{\partial vec(U)} = T_f \left[K_1^T (V_1 + V_1^T) vec(X) + (V_2 + V_2^T) vec(U) \right] \tag{III.56}$$

From equations (III.53) and (III.56),

$$vec(U) = -(V_2 + V_2^T)^{-1} K_1^T (V_1 + V_1^T) vec(X) \tag{III.57}$$

Using equation (III.57) in (III.33), we get

$$vec(X) = \left[I_{Nn} + K_1 (V_2 + V_2^T)^{-1} K_1^T (V_1 + V_1^T) \right]^{-1} K_2 vec(X_0) \tag{III.58}$$

Like equation (III.37), we can get the optimal X from the above $vec(X)$ by denoting I_N by its rows.

$$X = \Big[(I_N(1) \otimes I_n)\, vec(X) \quad (I_N(2) \otimes I_n)\, vec(X) \quad \cdots \quad (I_N(N) \otimes I_n)\, vec(X) \Big] \tag{III.59}$$

Then we find the whole sequence of optimal state by

$$x = X H_N \tag{III.60}$$

And $x(\lambda)$ is

$$x(\lambda) = X H_N(\lambda) \tag{III.61}$$

Putting $vec(X)$ from equation (III.58) to (III.57), we get the optimal control $vec(U)$ in terms of Haar wavelets.

$$vec(U) = \mathbf{K}vec(X_0) \tag{III.62}$$

Where

$$\begin{aligned}\mathbf{K} = &-(V_2 + V_2^T)^{-1}K_1^T(V_1 + V_1^T)\times \\ &\left[I_{Nn} + K_1(V_2 + V_2^T)^{-1}K_1^T(V_1 + V_1^T)\right]^{-1} K_2\end{aligned} \tag{III.63}$$

For reference trajectory tracking, the cost function is defined by

$$J = \int\limits_0^{T_f} \left[\left(x^{ref}(t) - x(t)\right)^T Q \left(x^{ref}(t) - x(t)\right) + u^T(t)Ru(t)\right] \tag{III.64}$$

Following the same procedure, the optimal control in Haar domain is obtained by

$$vec(U) = \mathbf{K}_4 \left(vec(X)^{ref} - K_2 vec(X_0)\right) \tag{III.65}$$

where,

$$\mathbf{K}_4 = \left[I_{Nl} + K_3K_1\right]^{-1} K_3 \tag{III.66}$$

$$K_3 = \left(V_2 + V_2^T\right)^{-1} K_1^T \left(V_1 + V_1^T\right) \tag{III.67}$$

From optimal $vec(U)$, we find U by

$$U = \left[(I_N(1)\otimes I_l)\, vec(U) \quad (I_N(2)\otimes I_l)\, vec(U) \quad \cdots \quad (I_N(N)\otimes I_l)\, vec(U)\right] \tag{III.68}$$

The optimal control input is given by

$$u = UH_N \tag{III.69}$$

And $u(\lambda)$ is

$$u(\lambda) = UH_N(\lambda) \tag{III.70}$$

So if we know the initial states of the system, we can get the optimal trajectory and optimal control in Haar domain from equations (III.58) and (III.62), respectively. This optimal control only depends on the initial states. Figure III.4 shows the algorithm of the open loop optimal control scheme.

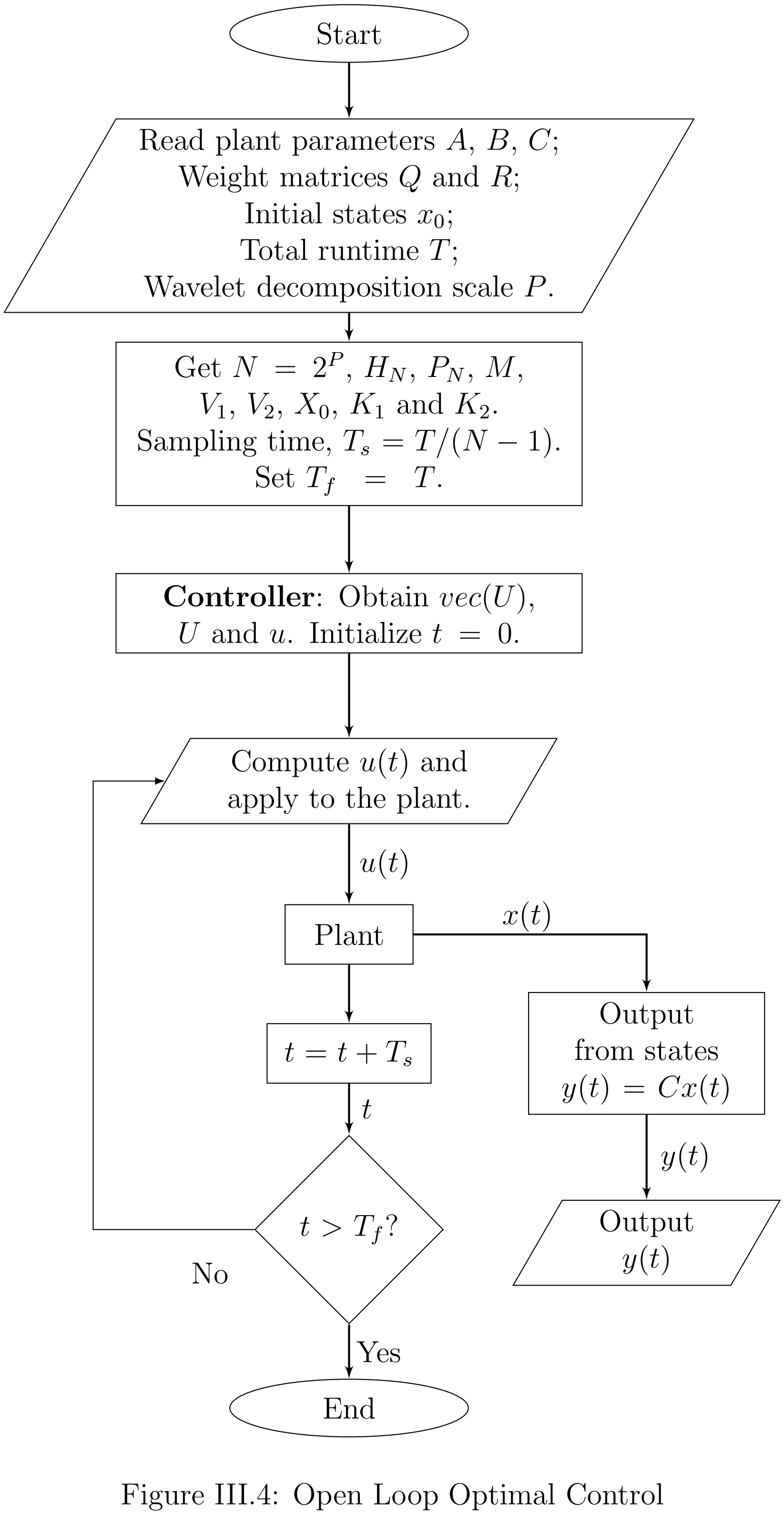

Start
Read plant parameters A, B, C;
Weight matrices Q and R;
Initial states x_0;
Total runtime T;
Wavelet decomposition scale P.
Get $N = 2^P$, H_N, P_N, M,
V_1, V_2, X_0, K_1 and K_2.
Sampling time, $T_s = T/(N-1)$.
Set $T_f = T$.
Controller: Obtain $vec(U)$,
U and u. Initialize $t = 0$.
Compute $u(t)$ and
apply to the plant.
$u(t)$
Plant
$x(t)$
Output
from states
$y(t) = Cx(t)$
$t = t + T_s$
t
$y(t)$
$t > T_f$?
No
Output
$y(t)$
Yes
End

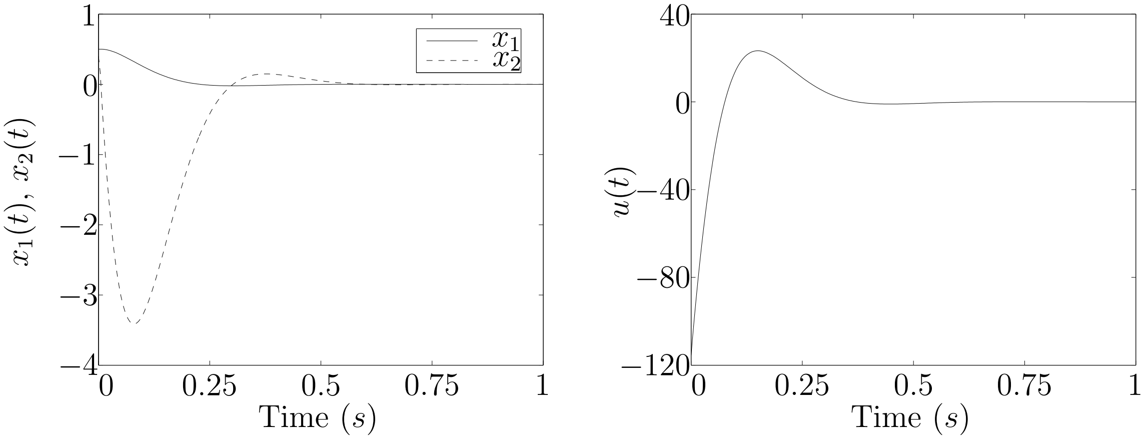
Figure III.4: Open Loop Optimal Control

III.5.1 Example of Open Loop Optimal Control via Haar Wavelets

Consider a system model given by

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \quad x_0 = \begin{bmatrix} 0.5 \\ 0.4 \end{bmatrix}$$
$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$$

We have to find out the optimal control and state trajectory for a finite time interval, which is $[0 \ \ 1.5]$ seconds. We set $T_f = 1s$. We select wavelet resolution scale $s = 8$ and get $N = 2^s$. This gives us the sampling time $T_s = T_f/N$. Next we get K_1 and K_2 from equations (III.34) and (III.35), respectively, V_1 and V_2 from equations (III.50) and (III.51), respectively. We take $Q = \begin{bmatrix} 5000 & 0 \\ 0 & 0 \end{bmatrix}$ and $R = 0.1$. Then we calculate the optimal trajectory $vec(X)$ and control $vec(U)$ from equations (III.58) and (III.62) respectively. The states and the open loop optimal control are shown in Figure III.5.



(a) Trajectory of the states

(b) Open loop optimal control

Figure III.5: Time evolution of states and open loop optimal control via Haar wavelet.

Next section reviews Receding Horizon Control, which can be used to develop a closed-loop optimal control scheme.

III.6 Receding Horizon Optimization

Receding Horizon Optimization approach is widely used in Model Predictive Control (MPC). In MPC, based on a model and the current states, future states are predicted over a horizon. Then a cost function is minimized to obtain the optimal control, which drive the predicted states towards reference. Because the optimized predicted states may change from the effect of disturbance or plant model mismatch, only the first control action from the horizon is applied to the plant. Then the horizon is receded in time, and the process is repeated. That is why the name is *Receding Horizon Control*. For a horizon size N, at time t, the control sequence u^t is obtained and the sequence ends up in future time $t + (N-1)T_s$. The horizon time NT_s is chosen to be equal or larger than the settling time of the step response of the plant.

$$u^t = \begin{bmatrix} u_t & u_{t+T_s} & \cdots & u_{t+(N-1)T_s} \end{bmatrix} \tag{III.71}$$

In Receding Horizon Control, the current control action is determined by measuring the current states and optimizing a finite horizon cost function at each sampling instant. Optimization at each sampling period yields an open loop optimal control trajectory, from which only the first portion is applied to the system until the next time step. The repetition of this procedure at each time step results in a closed loop control which mitigates uncertainties in the system. The fixed horizon recedes in time with each sampling instants, that is why the name *Receding* Horizon. It is also known as Model Predictive Control because of its use of the model to predict system behavior. The sampling period is much smaller than the horizon time.

So the idea can be summarized as follows:

i At time t , for the current state x^t, optimal control problem over a fixed time interval $[t \quad t+T_s \ \cdots \ t+(N-1)T_s]$ is solved.

ii From the resulted optimal control sequence u^t (in equation (III.71)), only the first control step u_t is applied to the system.

iii State reached at time $t + T_s$ is measured.

iv Fixed horizon optimization is repeated at time $t+T_s$ over the interval $[t+T_s \quad t+2T_s \ \cdots \ t+NT_s]$ from the *new* current states x^{t+T_s}.

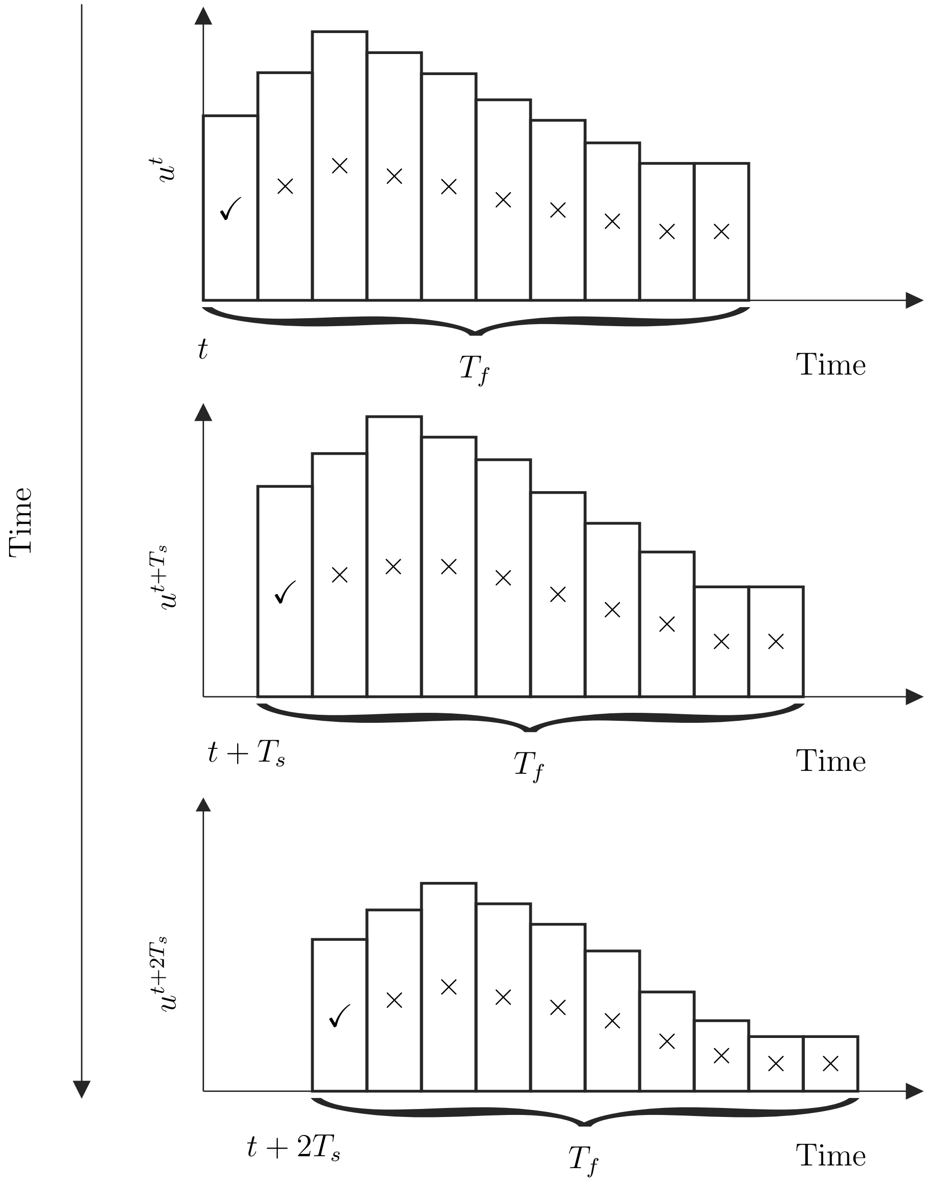


Figure III.6: Optimal control sequence u^t with fixed horizon size T_f is receding in time. Only the first control action (with the ✓mark) is applied to the system, rests are discarded.

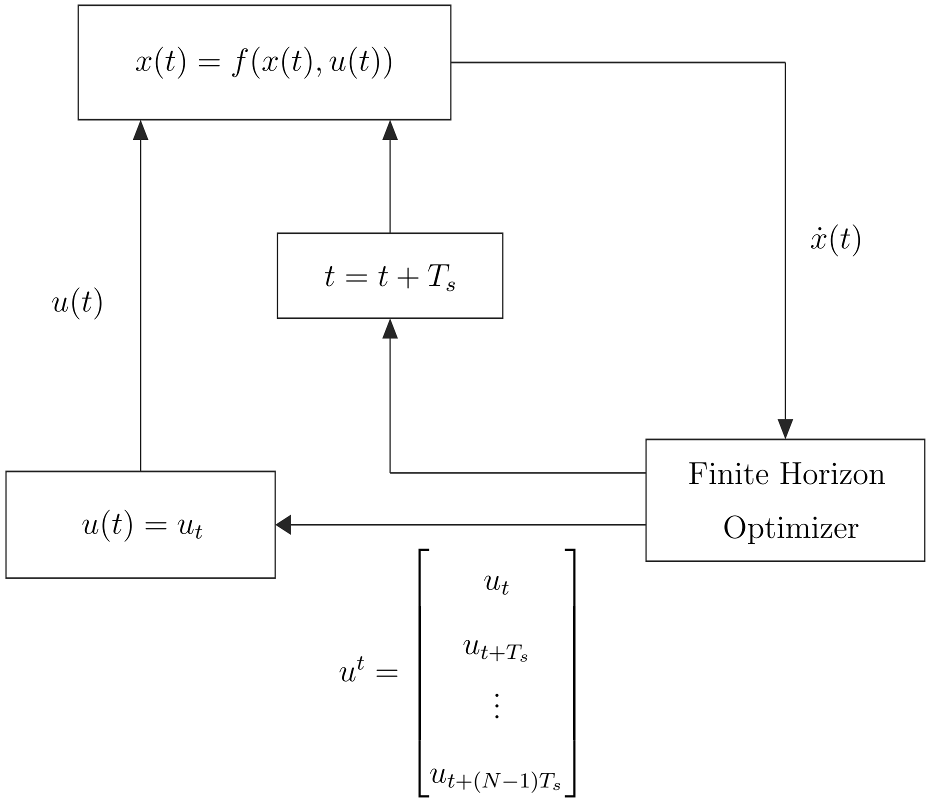


Figure III.7: Receding Horizon Optimizer acting as a closed loop controller. Control u^t is determined by minimizing the states at each time step. The first control step u_t is selected from the optimal control sequence u^t and set as $u(t) = u_t$. So $u(t + T_s) = f(x(t))$.

Computing a control sequence at each time step causes large computation load. With the advent of computing power, today it is possible to apply Receding Horizon Control to systems governed by fast dynamics. In the following section it is shown how the idea of Receding Horizon Optimization can be used for developing a closed loop control scheme using Haar wavelets.

III.7 Closed Loop Optimal Control

If we look closely at equation (III.62), we can see that the control action for T_f is dependent on the initial state x_0. A closed loop control scheme can be achieved in Receding Horizon Control (RHC) fashion. At each time step t, feedback of states is taken and set as x_0. Then an optimal control sequence u^t of N point is computed. These N points constitute the horizon T_f. This T_f will be receding in time with each time step. So the general idea would be to get the feedback of the states at every time step, set them as the new x_0^t, and calculate the new control sequence u^t for the time interval $[t \;\; (t+T_f)]$. We apply only the first control step from u^t and discard the rest $(N-1)$ steps. The same process is repeated at every T_s seconds; set the new feedback as $x_0^{t+T_s}$, calculate u^{t+T_s} for $[(t+T_s) \;\; (t+T_s+T_f)]$ interval, apply the first control step and discard the rest $(N-1)$ control steps. Figure (III.8) shows the flow chart of the closed loop control scheme.

Among offline computations, calculating K_1, K_2 and $\mathbf{K}$ (from equations (III.34), (III.35) and (III.63)) require the inversion of $(Nn \times Nn)$ matrices. During online closed loop computation, at each time step, we need to calculate $vec(U^t)$, which includes multiplication of two matrices $\mathbf{K}$ and $vec(X_0)$ with the order of $(Nl \times Nn)$ and $(Nn \times 1)$ respectively. This demands too much computation power. Moreover, we use only the first control from the sequence and discard the rest. So almost all of the computation effort at each time step is going to wasted and not necessary. To prevent this large wastage and consumption of computation power, we need to find a way to modify the computation, which will do the necessary computation only to find the first control action.

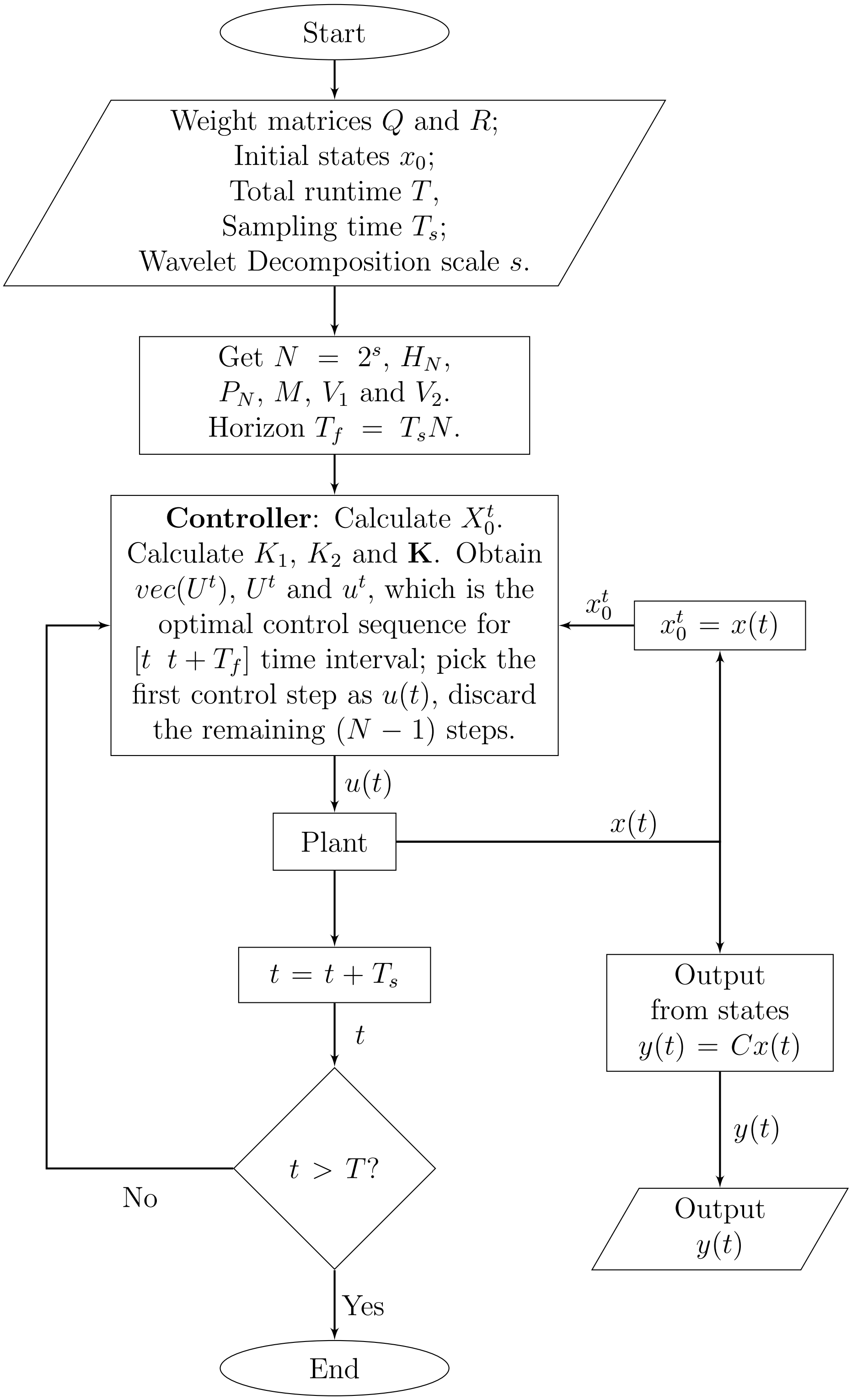

Start
Weight matrices Q and R;
Initial states x_0;
Total runtime T,
Sampling time T_s;
Wavelet Decomposition scale s.
Get $N = 2^s$, H_N,
P_N, M, V_1 and V_2.
Horizon $T_f = T_sN$.
Controller: Calculate X_0^t.
Calculate K_1, K_2 and $\mathbf{K}$. Obtain
$vec(U^t)$, U^t and u^t, which is the
optimal control sequence for
$[t \;\; t+T_f]$ time interval; pick the
first control step as $u(t)$, discard
the remaining $(N-1)$ steps.
x_0^t
$x_0^t = x(t)$
$u(t)$
Plant
$x(t)$
$t = t + T_s$
t
Output
from states
$y(t) = Cx(t)$
$y(t)$
$t > T$?
No
Output
$y(t)$
Yes
End

Figure III.8: Close Loop Optimal Control

III.8 Modified Computation for Closed Loop Optimal Control

From equations (III.39) and (III.70), we see that the first step in state and control trajectories depend only on the first column of H_N. Moreover we only need to calculate those wavelet coefficients with the same indexes of the non-zero elements in the first column of H_N. This means that the number of Haar coefficients required at every step is equal to the number of non-zero elements of the first column of H_N. This significantly reduces the amount of computation. If we select a resolution level of $s = 10$, then total number of Haar coefficients required for the whole time span becomes $N = 2^{10} = 1024$. Whereas if we consider the non-zero elements in the first column of H_N (11 only), then number of coefficients required for the first optimal state trajectory and optimal control step reduces to 11 (in general, $s+1$).

In the first column of H_N ($N = 2^s$), the number of non-zero elements is

$$N_* = s + 1 \tag{III.72}$$

Their indexes are given by an array of N_* elements, which is

$$Ind_{N_*} = \left[1, 1+2^0, 1+2^1, 1+2^2, 1+2^3, \ldots, 1+2^{s-1}\right] \tag{III.73}$$

In recursive form, it is given by

$$Ind_{N_*} = \left[Ind_{N_*-1}, 1+2^{s-1}\right] \tag{III.74}$$

In the definition of H_N (equation (II.60)), the first column of

$$H_{\frac{N}{2}} \otimes \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

gives the coordinates to Ind_{N_*-1}, and it contains 2^{s-1} number of non-zero elements.

In the first column of

$$I_{\frac{N}{2}} \otimes \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

the first element is non-zero, all other elements are zero. This gives the final non-zero element in Ind_{N*}.

For example, the first column of H_8 is (for $\lambda = 1/8$)

$$H_8\,(1/8) = \begin{bmatrix} \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \end{bmatrix}^T$$

where $N_* = 4$ non-zero elements with indexes $Ind_{8_*} = \begin{bmatrix} 1 & 2 & 3 & 5 \end{bmatrix}$. In the following, all the formulas previously introduced are revised considering that only $(s+1)$ entries of $H_N(1/m)$ are needed for control design.

Let S_{N_*} be a $(N_* \times N)$ sparse matrix whose i^{th} row has 1 in $Ind_{N_*}[i^{th}]$ column. So,

$$S_{N_*} = \begin{bmatrix} 1^{st} \text{ sparse } (1 \times N) \text{ vector with 1 in } Ind_{N_*}[1^{st}] \text{ column} \\ \vdots \\ (N_*)^{th} \text{ sparse } (1 \times N) \text{ vector with 1 in } Ind_{N_*}[(N_*)^{th}] \text{ column} \end{bmatrix} \qquad \text{(III.75)}$$

So, for H_8, S_{8_*} is given by

$$S_{8_*} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

We can construct a vector H_{N_*} which is made up with nonzero elements of the first column of H_N. It can be obtained from $H_N(1/N)$ (the first column of H_N) by premultiplying it with the sparse matrix S_{N_*}.

$$H_{N_*} = S_{N_*} H_N(1/N) \tag{III.76}$$

H_{8_*} is given by

$$\begin{aligned} H_{8_*} &= S_{8_*} H_8(1/8) \\ &= \begin{bmatrix} \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2} & \frac{1}{\sqrt{2}} \end{bmatrix}^T \end{aligned}$$

If, for example, we had the following sequence of 8 samples

$$y = [4 \;\; 7 \;\; -1 \;\; 2 \;\; 8 \;\; 12 \;\; 7 \;\; -3]$$

the corresponding Haar coefficients would be

$$Y = yH_8^T = \begin{bmatrix} 9\sqrt{2} & -3\sqrt{2} & 5 & 8 & -\frac{3}{\sqrt{2}} & -\frac{3}{\sqrt{2}} & -2\sqrt{2} & 5\sqrt{2} \end{bmatrix}$$

Y_* is made up with the columns from Y which have the indexes from Ind_{8_*}. Then

$$Y_* = \begin{bmatrix} 9\sqrt{2} & -3\sqrt{2} & 5 & -\frac{3}{\sqrt{2}} \end{bmatrix}$$

Then the first sample from the sequence y would be given by

$$Y_* H_{8_*} = \begin{bmatrix} 9\sqrt{2} & -3\sqrt{2} & 5 & -\frac{3}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2} & \frac{1}{\sqrt{2}} \end{bmatrix}^T = 4$$

So, finding the Haar coefficients with the indexes from Ind_{N_*} is sufficient for finding the first sample from the sequence.

Let us denote the first optimal state and control step at time t as x_*^t and u_*^t respectively. X_*^t and U_*^t are $(n \times N_*)$ and $(l \times N_*)$ matrices, respectively, made with the columns of X^t and U^t, respectively, which have the indexes from Ind_{N_*}. The first element of the sequences x^t and u^t are given as

$$x_*^t = X_*^t H_{N_*} \tag{III.77}$$

$$u_*^t = U_*^t H_{N_*} \tag{III.78}$$

Now we will start modifying the derivations. First we will look into the solution of linear systems via Haar wavelets, in section III.4. For obtaining the first time step of solution x in equation (III.30), X, X_0 and P_N will be replaced by X_*^t, $X_{0_*}^t$ and P_{N_*} respectively. From equation (III.27), $X_{0_*}^t$ is modified to

$$X_{0_*}^t = \left[\left(\frac{1}{\sqrt{N}}\right) x_0^t \quad \underbrace{0 \;\; 0 \;\; ... \;\; 0}_{N_*-1} \right] \tag{III.79}$$

P_{N_*} is made up from rows and columns of P_N which have the indexes from Ind_{N_*}. It is given by

$$P_{N_*} = S_{N_*} P_{N_*} S_{N_*}^T \tag{III.80}$$

That is,

$$\begin{aligned}
P_{8_*} &= S_{8_*} P_8 S_{8_*}^T \\
&= S_{8_*} \begin{bmatrix}
\frac{1}{2} & -\frac{1}{4} & -\frac{\sqrt{2}}{16} & -\frac{\sqrt{2}}{16} & -\frac{1}{32} & -\frac{1}{32} & -\frac{1}{32} & -\frac{1}{32} \\
\frac{1}{4} & 0 & -\frac{\sqrt{2}}{16} & \frac{\sqrt{2}}{16} & -\frac{1}{32} & -\frac{1}{32} & \frac{1}{32} & \frac{1}{32} \\
\frac{\sqrt{2}}{16} & \frac{\sqrt{2}}{16} & 0 & 0 & -\frac{\sqrt{2}}{32} & \frac{\sqrt{2}}{32} & 0 & 0 \\
\frac{\sqrt{2}}{16} & -\frac{\sqrt{2}}{16} & 0 & 0 & 0 & 0 & -\frac{\sqrt{2}}{32} & \frac{\sqrt{2}}{32} \\
\frac{1}{32} & \frac{1}{32} & \frac{\sqrt{2}}{32} & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{32} & \frac{1}{32} & -\frac{\sqrt{2}}{32} & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{32} & -\frac{1}{32} & 0 & \frac{\sqrt{2}}{32} & 0 & 0 & 0 & 0 \\
\frac{1}{32} & -\frac{1}{32} & 0 & -\frac{\sqrt{2}}{32} & 0 & 0 & 0 & 0
\end{bmatrix} S_{8_*}^T
\end{aligned}$$

$$= \begin{bmatrix}
\frac{1}{2} & -\frac{1}{4} & -\frac{\sqrt{2}}{16} & -\frac{1}{32} \\
\frac{1}{4} & 0 & -\frac{\sqrt{2}}{16} & -\frac{1}{32} \\
\frac{\sqrt{2}}{16} & \frac{\sqrt{2}}{16} & 0 & -\frac{\sqrt{2}}{32} \\
\frac{1}{32} & \frac{1}{32} & \frac{\sqrt{2}}{32} & 0
\end{bmatrix}$$

Equations (III.34) and (III.35) will be modified to

$$K_{1_*} = T_f \left[I_{N_* n} - T_f({P_{N_*}}^T \otimes A)\right]^{-1} ({P_{N_*}}^T \otimes B) \tag{III.81}$$

$$K_{2_*} = \left[I_{N_* n} - T_f({P_{N_*}}^T \otimes A)\right]^{-1} \tag{III.82}$$

And in equation (III.33), $vec(X)$ is replaced by

$$vec(X_*^t) = K_{1_*} vec(U_*^t) + K_{2_*} vec(X_{0_*}^t) \tag{III.83}$$

After modifying the solution via Haar wavelet, now we will modify the optimal control in section III.5. In equation (III.48), for X_*^t, M is replaced by M_*, which is

$$M_* = S_{N_*} M S_{N_*}^T \tag{III.84}$$

Consequently, V_1 and V_2 are changed to

$$V_{1_*} = {M_*}^T \otimes Q \tag{III.85}$$

$$V_{2_*} = {M_*}^T \otimes R \tag{III.86}$$

So in equation (III.52), the cost function becomes

$$J_*^t = T_f \left[vec^T(X_*^t) V_{1_*} vec(X_*^t) + vec^T(U_*^t) V_{2_*} vec(U_*^t) \right] \tag{III.87}$$

Differentiating J_*^t with respect to $vec(U_*^t)$ and setting it equal to zero for optimality, we get,

$$vec(U_*^t) = -(V_{2_*} + V_{2_*}^T)^{-1} K_{1_*}^T (V_{1_*} + V_{1_*}^T) vec(X_*^t) \tag{III.88}$$

Substituting $vec(U_*^t)$ from the above expression in equation (III.83) gives,

$$vec(X_*^t) = \left[I_{N_* n} + K_{1_*} (V_{2_*} + V_{2_*}^T)^{-1} {K_{1_*}}^T (V_{1_*} + V_{1_*}^T) \right]^{-1} K_{2_*} vec(X_{0_*}^t) \tag{III.89}$$

If we denote I_{N_*} by its rows, we can get X_*^t by

$$X_*^t = \Big[(I_{N_*}(1) \otimes I_n)\, vec(X_*^t) \quad (I_{N_*}(2) \otimes I_n)\, vec(X_*^t) \quad \cdots \quad (I_{N_*}(N_*) \otimes I_n)\, vec(X_*^t) \Big] \tag{III.90}$$

From equations (III.89) and (III.88),

$$vec(U_*^t) = \mathbf{K}_* vec X_{0_*}^t \tag{III.91}$$

where $\mathbf{K}_*$ is the feedback gain, given by

$$\begin{aligned}\mathbf{K}_* = &- (V_{2_*} + V_{2_*}^T)^{-1} {K_{1_*}}^T (V_{1_*} + V_{1_*}^T) \times \\ &\left[I_{N_* n} + K_{1_*}(V_{2_*} + V_{2_*}^T)^{-1} {K_{1_*}}^T (V_{1_*} + V_{1_*}^T)\right]^{-1} K_{2_*}\end{aligned} \tag{III.92}$$

For reference trajectory tracking, the modified formula for the first optimal control action in the horizon at time t is given by

$$vec(U_*^t) = \mathbf{K}_{4_*} \left(vec(X_*)^{ref} - K_{2_*} vec(X_{0_*}^t)\right) \tag{III.93}$$

where,

$$\mathbf{K}_{4_*} = \left[I_{N_* l} + K_{3_*} K_{1_*}\right]^{-1} K_{3_*} \tag{III.94}$$

$$K_{3_*} = \left(V_{2_*} + V_{2_*}^T\right)^{-1} K_{1_*}^T \left(V_{1_*} + V_{1_*}^T\right) \tag{III.95}$$

U_*^t is given by

$$U_*^t = \Big[(I_{N_*}(1) \otimes I_l)\, vec(U_*^t) \quad (I_{N_*}(2) \otimes I_l)\, vec(U_*^t) \quad \cdots \quad (I_{N_*}(m) \otimes I_l)\, vec(U_*^t)\Big] \tag{III.96}$$

Once we get $vec(X_*^t)$ and $vec(U_*^t)$, we get x_*^t and u_*^t from equations (III.77) and (III.78) respectively. $u(t)$ is computed as a function of the feedback from previous time step $t+1$. So $u(t)$ can be expressed as a function of the feedback of the states from the previous time step.

$$u((k+1)T_s) = f(x(kT_s)) \tag{III.97}$$

Figure III.9 shows the flow chart of the modified closed loop control scheme.

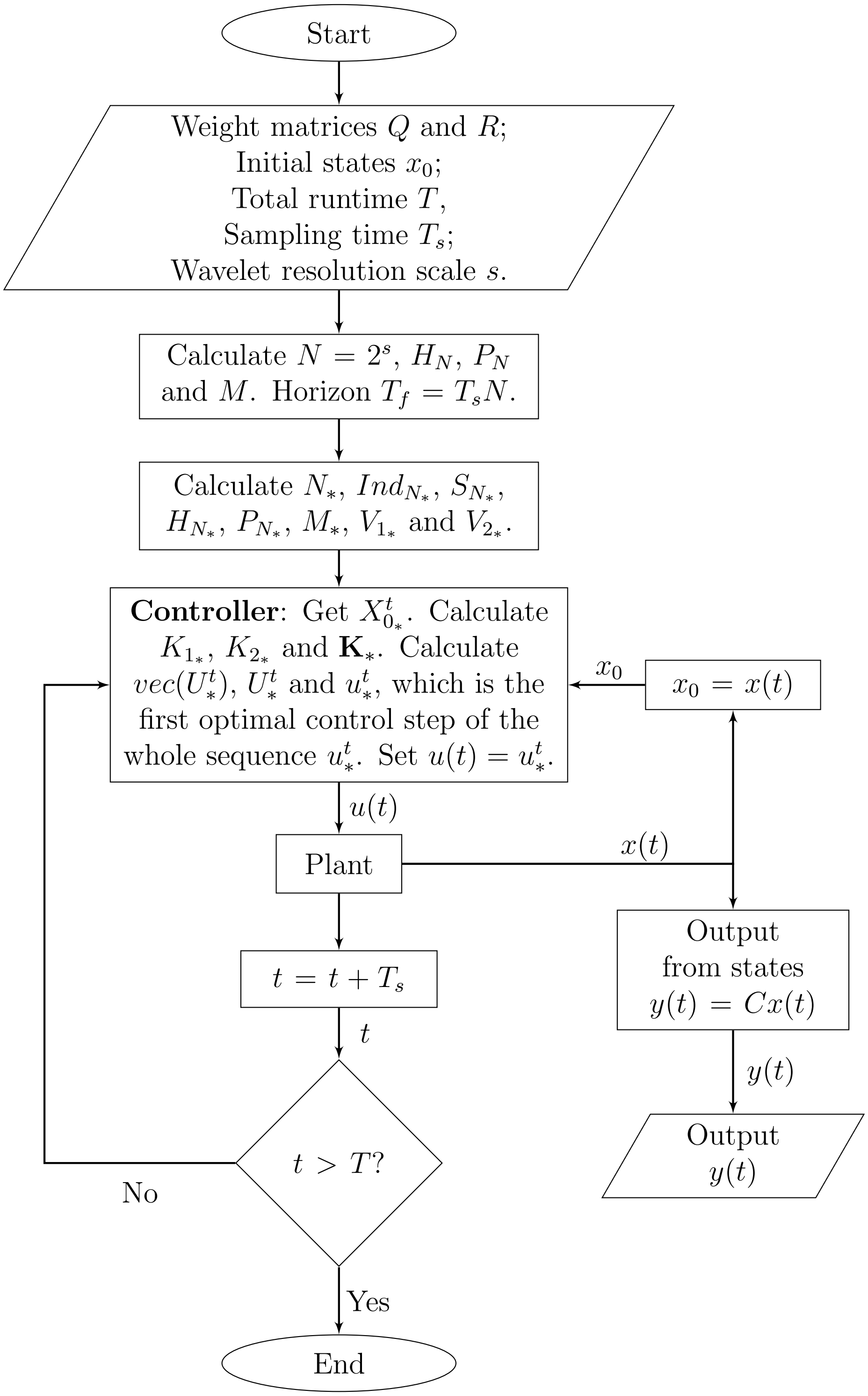

Start
Weight matrices Q and R;
Initial states x_0;
Total runtime T,
Sampling time T_s;
Wavelet resolution scale s.
Calculate $N = 2^s$, H_N, P_N and M. Horizon $T_f = T_s N$.
Calculate N_*, Ind_{N_*}, S_{N_*}, H_{N_*}, P_{N_*}, M_*, V_{1_*} and V_{2_*}.
Controller: Get $X^t_{0_*}$. Calculate K_{1_*}, K_{2_*} and $\mathbf{K}_*$. Calculate $vec(U^t_*)$, U^t_* and u^t_*, which is the first optimal control step of the whole sequence u^t_*. Set $u(t) = u^t_*$.
x_0
$x_0 = x(t)$
$u(t)$
Plant
$x(t)$
Output from states $y(t) = Cx(t)$
$t = t + T_s$
t
$y(t)$
$t > T?$
No
Output $y(t)$
Yes
End

Figure III.9: Modified Close Loop Optimal Control

III.8.1 Computation Load in Modified Closed Loop Optimization

In this modified version of closed loop control scheme, instead of calculating the whole control sequence u^t at each time step, we are calculating u^t_*, which is the first control step of u^t. It should be noticed that the receding horizon has not been reduced to a single time step. The horizon T_f is the same, we are only calculating the first control action and ignoring the rest. The calculation of K_{1_*}, K_{2_*} and $\mathbf{K}_*$ require inversion of matrices of order $(N_*n \times N_*n)$ (from equations (III.81), (III.82) and (III.92)), which were previously $(Nn \times Nn)$. $vec(U^t_*)$ is calculated from equation (III.91) or (III.93), where each of N_*l coefficients are obtained by N_*n number of arithmetic multiplication, so at each time step total number of multiplication is $N_*l \times N_*n$. Without modifications, in equation (III.62) and (III.65), total number of multiplication at each time step is $Nl \times Nn$. For example, if $s = 10$, that is $N = 2^{10} = 1024$, total number of multiplication at each time step is $(1024l \times 1024n)$, but in the modified version, that amount comes down to $(11l \times 11n)$, which is substantial savings in computation. This will allow the implementation of Receding Horizon Optimization in systems with fast dynamics.

III.9 Comparison with Dynamic Matrix Control (DMC)

There are different types of MPCs, among them DMC is selected for comparison purpose due to its simplicity and effectiveness. It was first developed by the engineers at Shell Oil Company in the early 1970's, since then many versions of DMC have been developed. The Dynamic Matrix Control law presented in [29] is considered. This law uses the parametric plant model which yields an easy and effective controller design method for discrete time tracking controller.

III.9.1 DMC Control Law

Discrete time state space representation of a LTI system is given by

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \tag{III.98}$$

where the matrix dimensions are same as in equation (III.15). The Dynamic Matrix Controller is obtained by minimizing the quadratic cost function

$$J = \sum_{i=i}^{N} \left[e_{k+i}^T Q_i e_{k+i} + u_{k+i}^T R_i u_{k+i} \right] \tag{III.99}$$

where

$e_{k+i} = x_{k+i}^{ref} - x_{k+i}$	future tracking error vector $(n \times 1)$
x_{k+i}^{ref}	future reference trajectory vector $(n \times 1)$
x_{k+i}	future plant state vector $(n \times 1)$
u_{k+i}	future control vector $(l \times 1)$
Q_i, R_i	$(n \times n)$ and $(l \times l)$ weighting matrices
N	optimization horizon

From the state space representation in equation (III.98), future values x_{k+i} over the horizon $[k+1, \;\; k+N]$ are obtained as

$$\underbrace{\begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+N} \end{bmatrix}}_{\text{X}} = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\text{T}} x_k + \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}}_{\text{S}} \underbrace{\begin{bmatrix} u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+N} \end{bmatrix}}_{\text{U}} \tag{III.100}$$

where x_k is the state feedback at time k. T and S represents unforced and forced responses, respectively. Equation (III.99) can be written as

$$J = (X^{ref} - X)^T Q (X^{ref} - X) + U^T R U \tag{III.101}$$

where

$$Q = \begin{bmatrix} Q_1 & 0 & \cdots & 0 \\ 0 & Q_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_N \end{bmatrix} R = \begin{bmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_N \end{bmatrix} \tag{III.102}$$

The optimal control vector for the horizon is obtained by

$$U = K_{DMC}(X^{ref} - Tx_k) \tag{III.103}$$

where

$$K_{DMC} = (R + S^T Q S)^{-1} S^T Q \tag{III.104}$$

Only the first control vector u_k is applied to the plant. It is obtained by

$$u_k = K_{DMC_*}(X^{ref} - Tx_k) \tag{III.105}$$

where K_{DMC_*} is defined by the first l rows of K_{DMC}. In DMC, for finding u_k, the number of multiplication operations is $l \times Nn$, whereas in equation (III.91) or (III.93) it was $N_* l \times N_* n$. In general, the amount of multiplication at each time step in DMC and Haar based MPC are $2^s nl$ and $(s+1)^2 nl$, respectively. For $s = 5$, that is, for a horizon size of $2^5 = 32$, the number of multiplication operations in DMC at each time step is $l \times 32n = 32nl$. In Haar based MPC ($N_* = s + 1 = 6$), the number of multiplication operations at each time step is $6l \times 6n = 36nl$. But for $s = 6$, the number of multiplication operation in DMC and Haar based MPC are $64nl$ and $49nl$, respectively. So for horizon a size $N \geq 2^6$, Haar based MPC has less amount of computation than DMC. Now we will evaluate the performance of both of these controllers with the same horizon size for velocity regulation of a DC motor.

III.9.2 Velocity Regulation of a DC motor

The velocity governing equation of a DC motor is given as

$$\dot{\omega}(t) = K_t \frac{V(t) - K_e \omega(t)}{RI} \tag{III.106}$$

where ω is the angular velocity, K_t is the motor torque constant, K_e is the motor voltage constant, R is resistance, I is inertia and V is applied voltage. In this example, the motor has to follow a velocity setpoint ω^{ref}. Both Haar based MPC and DMC are applied with a same prediction horizon size.

For Haar based MPC, $V(t)$ is obtained each time step from equation (III.93). For DMC, $V(t)$ is obtained from equation (III.105).

The constants are given as

$$\begin{aligned}
&K_e = 0.575 \ \ \text{V/rpm} \\
&K_t = 48.61 \ \ \text{lb.inch/amp} \\
&I = 0.79 \ \ \text{lb.inch.s}^2 \\
&R = 4 \ \ \text{ohms}
\end{aligned}$$

The initial velocity is $\omega_0 = 0$ rpm. Total runtime $T = 2$ s and sampling time $T_s = 0.001$ s. For the first second, desired velocity is $\omega^{ref}(t) = 2$ rpm, from 1 to 2 second it is 1 rpm. Wavelet resolution scale of $s = 6$ is used. So, $N = 2^6 = 64$. The setpoint and actual velocity of the motor from both the controllers are shown in Figures (III.10) and (III.11).

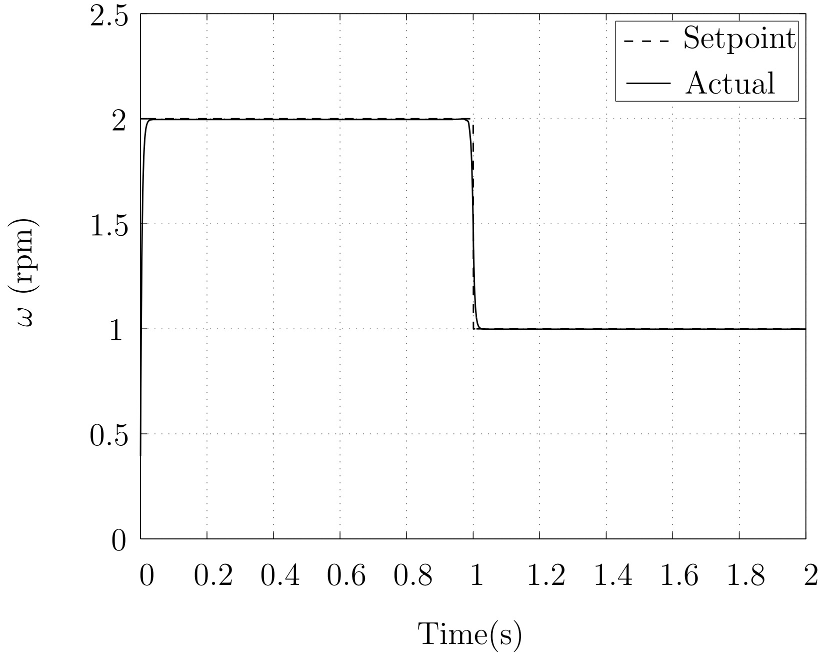


Figure III.10: Desired and actual angular velocity of the motor from Haar based MPC

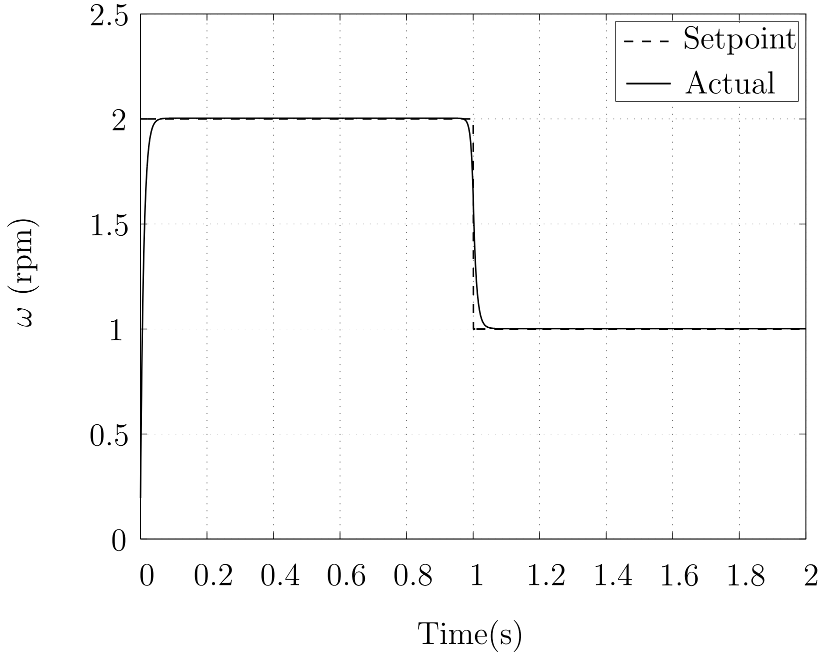


Figure III.11: Desired and actual angular velocity of the motor from DMC

For similar level of performance, Haar based MPC requires less computation. For larger horizon, the difference in computation increases. For $s = 10$, number of multiplication operations in DMC and Haar based MPC are $1024nl$ and $121nl$, respectively.

CHAPTER IV

TRAJECTORY TRACKING

In this chapter, error dynamics are stabilized using the Receding Horizon control scheme in Haar domain. Two examples are considered: position and orientation tracking of a two link planar robot and a wheeled mobile robot. In these examples, the control scheme from the previous chapter is applied on the error dynamics. In the process, an optimized auxiliary control signal is generated from the error dynamics, from which the original control for the plant is computed.

IV.1 Position Control of Two Link RR Planar Robot

The close loop control scheme described in section III.8 is applied for position control of a 2 link RR planar robot. Robot kinematics and dynamics are discussed in Appendix E. In Appendix F, desired smooth circular trajectory is generated in the workspace, and it is converted in smooth joint space trajectory for the joint variables of the robot. Position control of the robot is shown in section IV.1.1. A stylization of the kinematic structure is depicted in Figure (IV.1). Table IV.1 shows the symbols used to derive kinematic and dynamic equations of the robot. For more details on the kinematics and dynamics of manipulators, interested readers are encouraged to refer [30] and [31].

I_1	Motor 1 rotor inertia	q_1	Angle of link 1 with respect to the global x axis
I_2	Link 1 centroidal moment of inertia	q_2	Angle of link 2 with respect to link 1
M_1	Motor 1 mass	τ_1	Torque produced by motor 1
M_2	Link 1 mass	τ_2	Torque produced by motor 2
I_3	Motor 2 rotor inertia	x_{c1}	Global position vector of center of gravity of link 1
I_4	Link 2 centroidal moment of inertia	x_{c3}	Global position vector of center of gravity of motor 2
M_3	Motor 2 mass	x_{c4}	Global position vector of center of gravity of link 2
M_4	Link 2 mass	x_{cp}	Global position vector of center of gravity of payload
I_{3c}	Motor 2 stater inertia	v_{c2}	Velocity vector of center of gravity of link 2
M_p	Payload mass	v_{c3}	Velocity vector of center of gravity of motor 2
I_p	Payload moment of inertia	v_{c4}	Velocity vector of center of gravity of link 2
L_1	Length of link 1	v_p	Velocity vector of center of gravity of payload
L_2	Length of Link 2	x_{cpx}	x component of x_{cp}
L_3	Distance of center of gravity of link 1 from axis of rotation	x_{cpy}	y component of x_{cp}
L_4	Distance of center of gravity of link 2 from the axis of rotation	r	Distance of center of gravity of the payload from that of motor 1

Table IV.1: Symbol definitions for kinematic and dynamic structure of 2 link planar robot

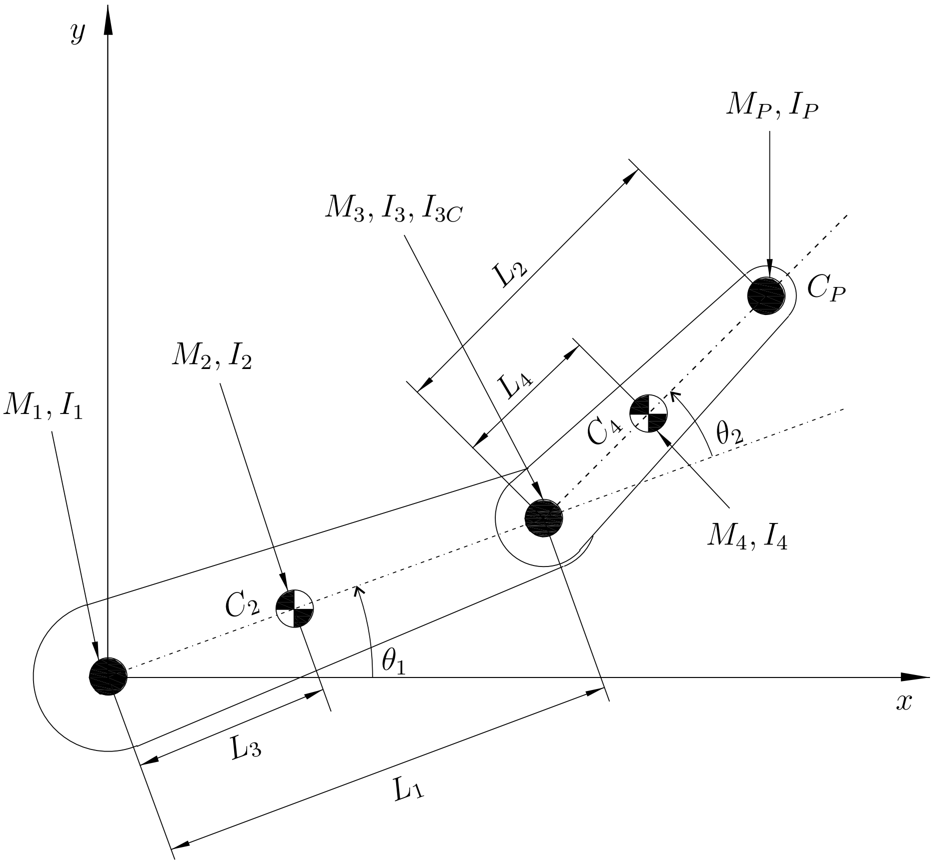


Figure IV.1: 2 link planar robot in global x-y co-ordinate

IV.1.1 Position Control

For position control, tracking error dynamics needs to constructed for the manipulator. From equation (E.22), the robot dynamics is given by

$$M(q)\ddot{q}(t) + C(q, \dot{q})\dot{q} = \tau(t) \tag{IV.1}$$

Since $q_d(t)$ and $\dot{q}_d(t)$ are defined from the previous section, position and velocity tracking errors can be obtained as

$$e(t) = q_d(t) - q(t) \tag{IV.2}$$

$$\dot{e}(t) = \dot{q}_d(t) - \dot{q}(t) \tag{IV.3}$$

Differentiating equation (IV.3) yields,

$$\ddot{e}(t) = \ddot{q}_d(t) - \ddot{q}(t) \tag{IV.4}$$

Substituting $\ddot{q}_d(t)$ in equation (IV.1),

$$\ddot{e}(t) = \ddot{q}_d(t) + M^{-1}\left(C\dot{q}(t) - \tau(t)\right) \tag{IV.5}$$

Defining the control input function

$$\tau(t) = C\dot{q}(t) - M\left(u(t) - \ddot{q}_d(t)\right) \tag{IV.6}$$

The tracking error dynamics is

$$\ddot{e}(t) = u(t) \tag{IV.7}$$

In state space form, the error dynamics becomes

$$\begin{bmatrix} \dot{e}(t) \\ \ddot{e}(t) \end{bmatrix} = \begin{bmatrix} 0_2 & I_2 \\ 0_2 & 0_2 \end{bmatrix} \begin{bmatrix} e(t) \\ \dot{e}(t) \end{bmatrix} + \begin{bmatrix} 0_2 \\ I_2 \end{bmatrix} u(t) \tag{IV.8}$$

where 0_2 and I_2 are zero and identity matrix of order 2, respectively.

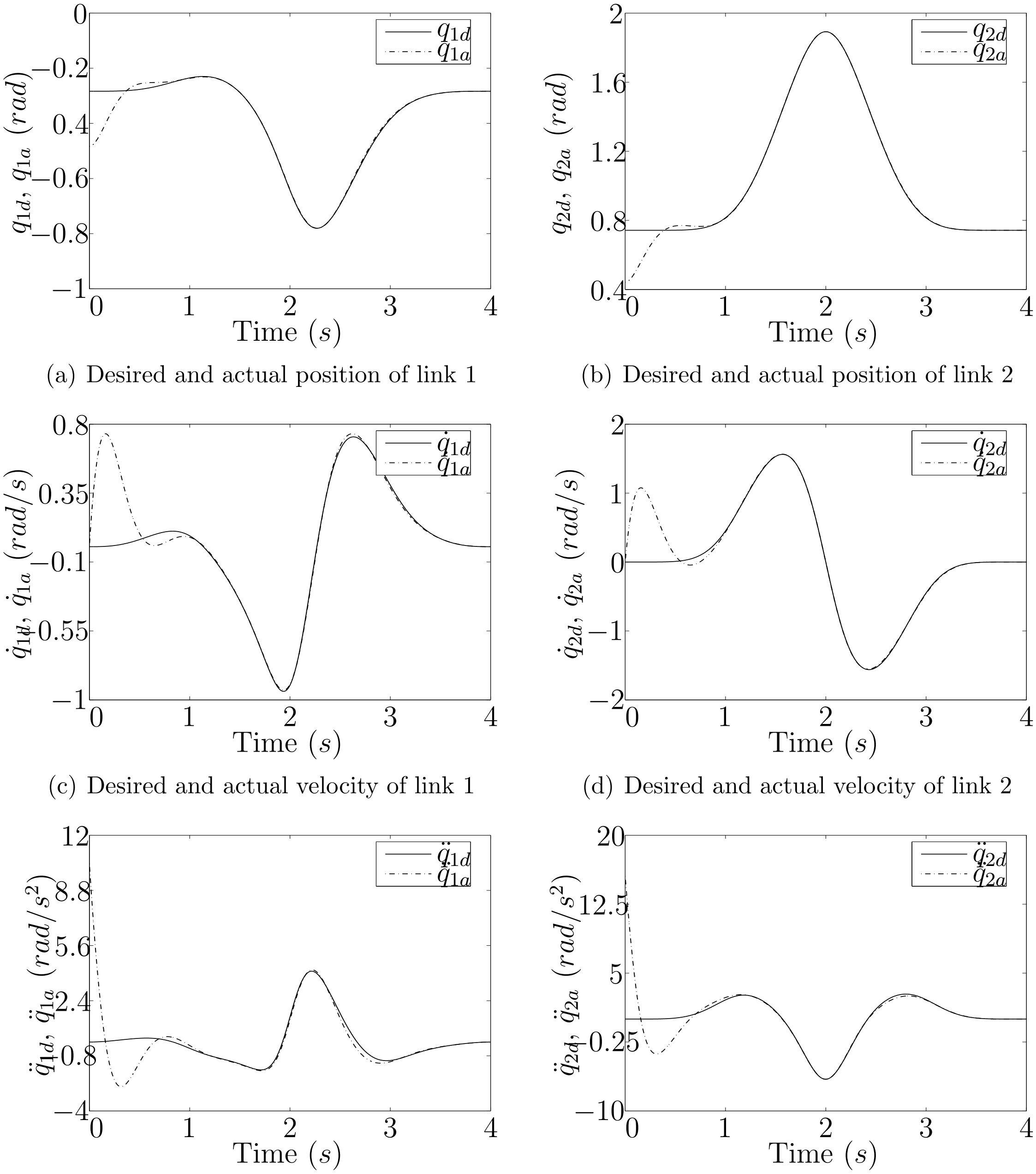
In the example, NSK 2 link robot has been taken into consideration. Its arms' lengths are $L_1 = 0.38m$ and $L_2 = 0.24m$. Under no load condition, coupled parameters are taken as $p_1 = 3.6$, $p_2 = 0.2$ and $p_3 = 0.15$ [32], [33]. Wavelet resolutions scale $s = 6$ is used, so $N = 2^6 = 64$. Receding horizon is $T_f = T_s(N-1) = 0.252s$. Weight matrices are

$$Q = \begin{bmatrix} 5000 & 0 & 0 & 0 \\ 0 & 5000 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

Here,

$$A = \begin{bmatrix} 0_2 & I_2 \\ 0_2 & 0_2 \end{bmatrix} \qquad B = \begin{bmatrix} 0_2 \\ I_2 \end{bmatrix}$$

Feedback gain matrix is obtained from equation (III.92). Every time, State feedback vector $\begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix}$ is set as x_0 and $u(t)$ is obtained according to the procedures in section III.8 and $\tau(t)$ is computed from equation (IV.6). In the simulation, the starting position of the end-effector is placed outside the desired circle to show how the controller minimize this initial error. Simulation results are shown in Figure IV.2.



(a) Desired and actual position of link 1

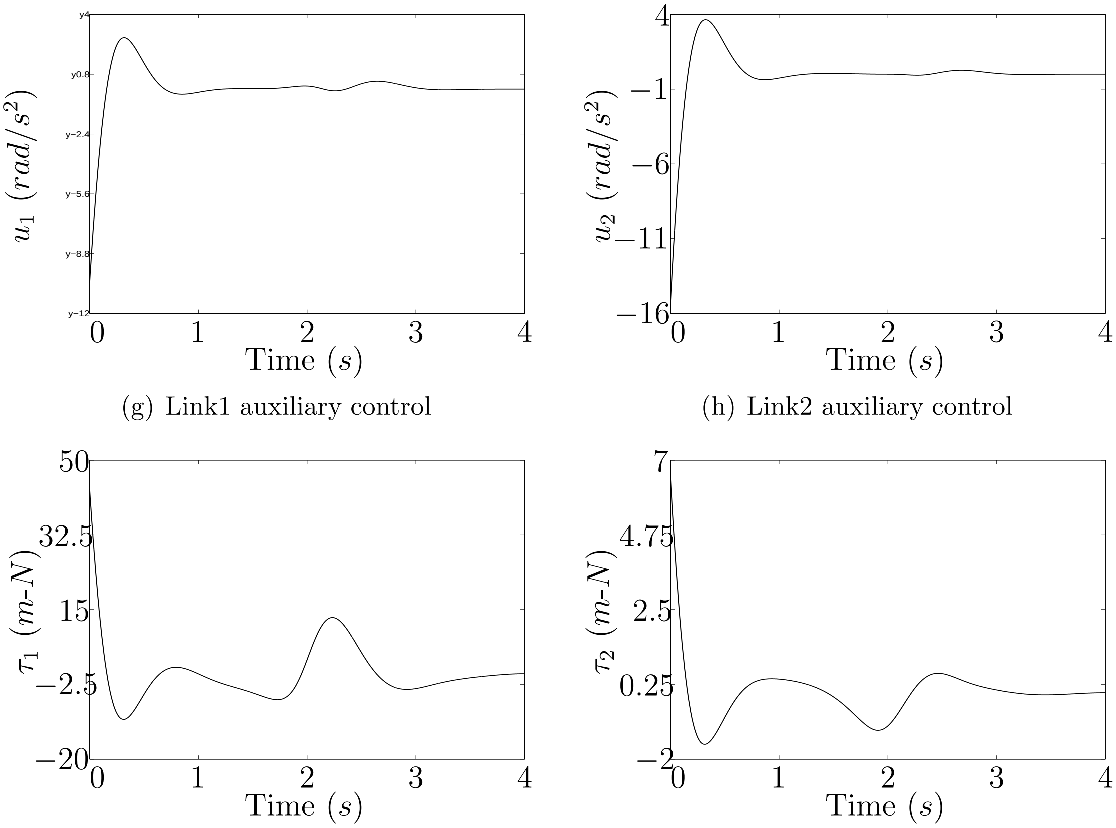
(b) Desired and actual position of link 2

(c) Desired and actual velocity of link 1

(d) Desired and actual velocity of link 2

(e) Desired and actual acceleration of link 1

(f) Desired and actual acceleration of link 2

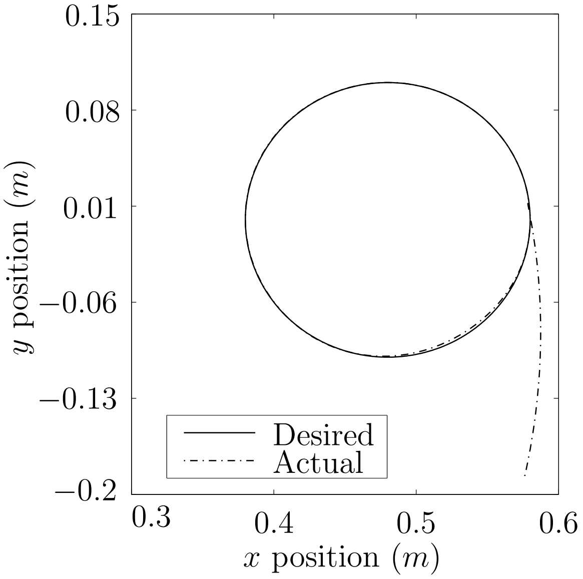


(g) Link1 auxiliary control

(h) Link2 auxiliary control

(i) Link1 torque

(j) Link2 torque



(k) End-effector cartesian position

Figure IV.2: Auxiliary controls in Figures IV.2(g) and IV.2(h) are the close loop auxiliary optimal control obtained in receding horizon control fashion, via Haar wavelet. τ_1 and τ_2 in Figures IV.2(i) and IV.2(j) are torques driving the states of the links towards their desired trajectories, shown in Figures IV.2(a)-IV.2(f). Figure IV.2(k) shows the desired and actual end-effector trajectory in Cartesian space.

IV.2 Trajectory Tracking of a Mobile Robot

A differential drive mobile robot is a nonholonomic robot. It has two independently actuated wheels on a common axis with a caster wheel. For linear motion both wheels rotate with the same velocity, to make a turn wheels rotate with different velocities, hence the name differential drive.

IV.2.1 Robot Kinematics

Position and orientation of the robot at any time is described by $q = [x, y, \theta]^T$, (x,y) is the position of C_r in the global corodinate frame and θ is the orientation with respect to the positive x axis. ω_l and ω_r are angular velocities of left and right wheels, respectively. These two angular velocities are control inputs. There are three variables to control in q, whereas we have two control inputs. So to reduce the number of variables to control in q from three to two, the following transformation is used

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{IV.9}$$

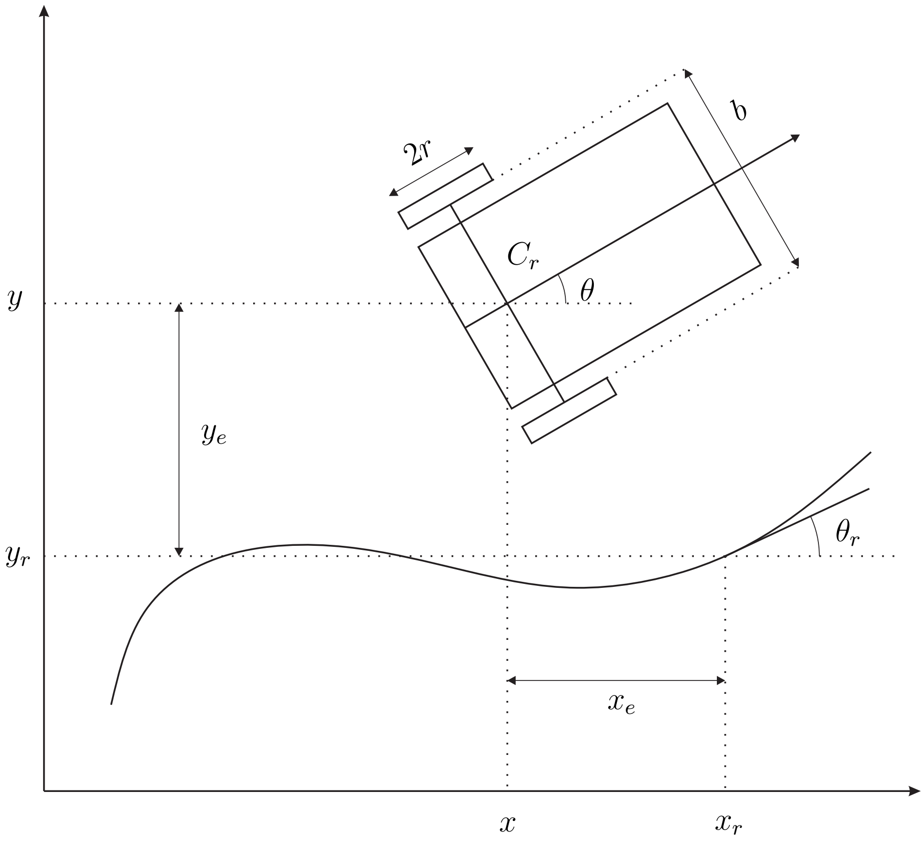


Figure IV.3: Mobile robot kinematic model

where v is linear velocity of C_r and ω is angular velocity of the robot. The velocities ω and v are given by

$$\omega = \frac{r(\omega_r - \omega_l)}{b} \tag{IV.10}$$

$$v = \frac{r(\omega_r + \omega_l)}{2} \tag{IV.11}$$

So once v and ω are obtained, desired angular velocities of the wheels can be decided using the above two equations.

IV.2.2 Position and Orientation Tracking of Mobile Robot

The velocities v and ω are obtained in the outer loop path control, and from them, ω_l and ω_r are obtained from inner loop velocity control. Here, outer loop path control is done by receding horizon optimizer in Haar domain. The reference state vector is given by

$$\dot{q}_d = \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{bmatrix} = \begin{bmatrix} v_d \cos\theta_d \\ v_d \sin\theta_d \\ \omega_d \end{bmatrix} \tag{IV.12}$$

The tracking error is written as [34]

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix} \tag{IV.13}$$

The error state dynamic model is derived as follows

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} \omega e_y - v + v_d \cos e_\theta \\ -\omega e_x + v_d \sin e_\theta \\ \omega_r - \omega \end{bmatrix} \tag{IV.14}$$

Defining the control signal

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} v_d \cos e_\theta - v \\ \omega_d - \omega \end{bmatrix} \tag{IV.15}$$

Then error state dynamics becomes

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} 0 & \omega & 0 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} + \begin{bmatrix} u_1 \\ v_d \sin e_\theta \\ u_2 \end{bmatrix} \tag{IV.16}$$

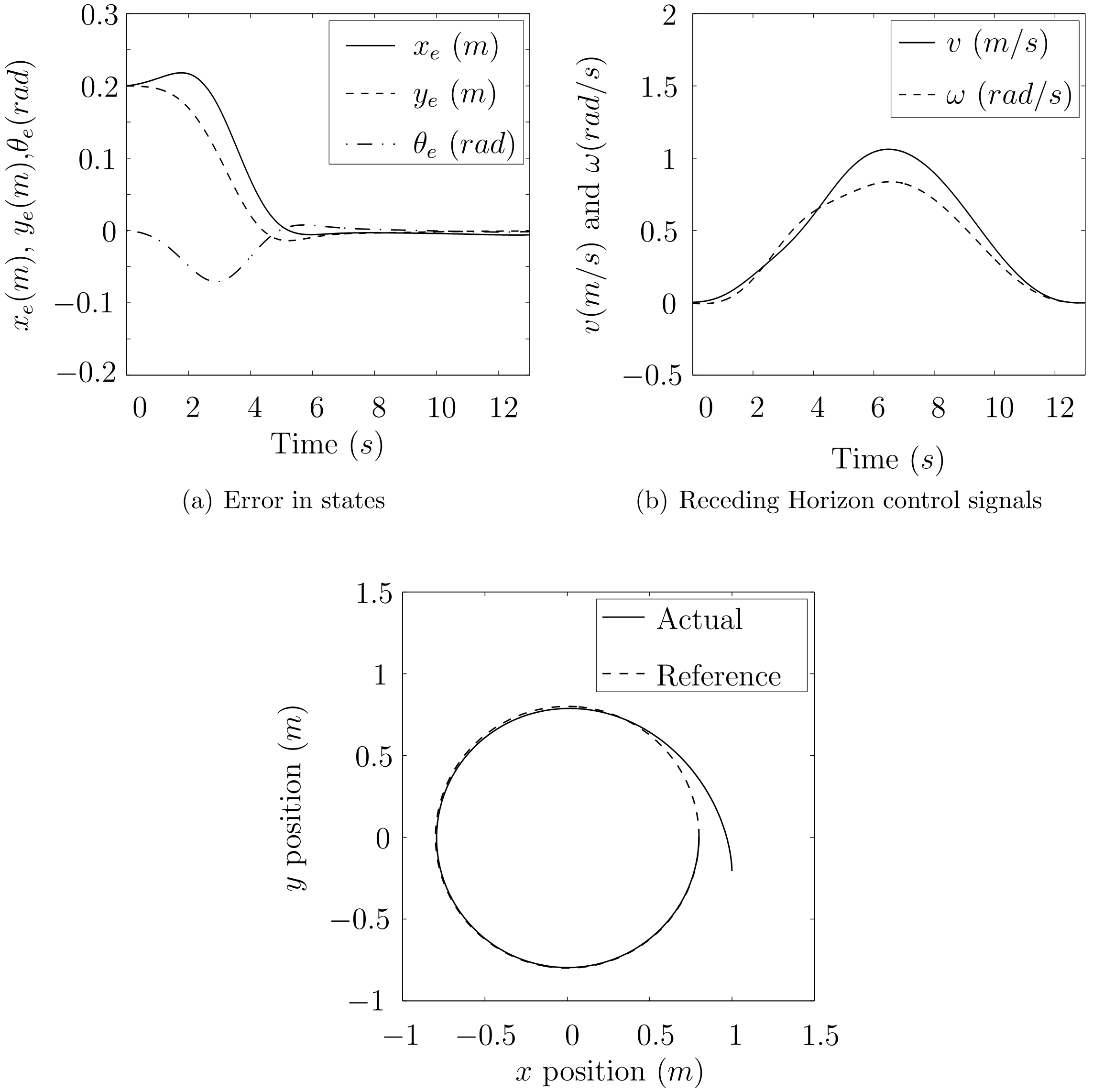
Linearized error state model is [34]

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & v_d \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{IV.17}$$

The tracking control is to find u_1 and u_2 to drive the error system towards equilibrium ($e(t) = 0$ and $u(t) = 0$). Receding horizon optimizer is used in Haar domain for finding optimal u at each time step. For simulation purposes, $b = 7in$ and $r = 1.375in$ was used. Weight matrices were

$$Q = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 0.9 & 0 \\ 0 & 0 & 0.9 \end{bmatrix} \qquad R = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix} \tag{IV.18}$$

The mobile robot is to traverse a circle of radius $0.8m$ in $13s$ without any initial and final velocity, acceleration and jerk. Sampling time is $0.01s$. Horizon size of $5s$ is used with resolution scale $s = 8$.



(a) Error in states

(b) Receding Horizon control signals

(c) Cartesian position

Figure IV.4: Control signals are driving the error states towards zero. The robot is initially placed outside the reference circle, then the actual trajectory merges with the reference trajectory.

All the simulations in this chapter were done with the Haar based Receding Horizon controller. In all the figures with comparisons between reference and actual trajectory of the states, it can be seen that actual states followed the reference very closely, and there were very less overshoot. It is because of the nature of the MPC, it could predict the future error in the error dynamics in advance. Based on where the error is going, it adjusted the control input action accordingly, so that the control input does not *overdo* and overshoots the trajectories. These results also indicate that the attempt of using Haar coefficients of only the first control input to reduce computation was successful in attaining a Model Predictive Controller; it could predict future trajectories, prevented overshoot and provided smooth trajectories following references.

CHAPTER V

CONCLUSION AND FUTURE WORK

A fixed horizon optimal controller designed in wavelet domain is investigated in this thesis. The controller optimizes a parameterized linear quadratic cost function in wavelet domain. To solve for the optimal control action for the selected weight matrices in the cost function, the governing equation of the linear system is transformed into wavelet domain.

The optimal controller minimizes the states from the initial values in the horizon, using the states' initial condition. To convert this process into a closed loop control scheme, a horizon of fixed size is receded in time by a single time step, while taking feedback of the states, setting it as the initial value for the horizon and optimizing the cost function. Only the first control action is applied to the system and the rest are discarded. This process is repeated at each time, and thus the procedure constructs a Receding Horizon Optimizer in wavelet domain.

The computational load of calculating an optimal control for a horizon at each time step poses a significant practical implementation problem. To mitigate this problem for the Haar based MPC, the fact that Haar wavelet is not globally supported is exploited. Each of the column of Haar transformation matrix represents the time component of the transformed signal. Therefore, while transforming, only the first column of the Haar matrix is used, since only the first control action is used for each horizon. Moreover, the first column of the Haar matrix is sparse. This was exploited to reduce the computational load at each time step. The corresponding modified equations are derived in this thesis. The modified algorithm was compared with DMC, a conventional MPC, for velocity regulation of a DC motor.

It was stated before that at each time step, states are driven towards zero by the optimal control input. For following desired trajectories, the error dynamics of the system is considered for the proposed Receding Horizon Optimizer in Wavelet domain, which generates an auxiliary control, which drives the error towards zero at each time step. The resulting auxiliary control is used to obtain the actual input to the system. Numerical simulation results show the successful implementation of the controller in velocity regulation of a DC motor and position and orientation control of a two-link planar robot and a wheeled mobile robot, respectively.

In the future, it might be possible to further reduce the computation load. Smooth signals have most of their information contained in lower frequency bands. So for a smooth control input, Haar wavelets corresponding to high frequency bands have lower value compared to those corresponding to lower frequency bands. If the Haar wavelets with higher frequencies can be ignored without sacrificing too much accuracy, then order of computation can be further reduced.

Equation (III.58) shows the optimal state as a function of the state feedback, which is the initial state $vec(X_0)$ of the horizon in that equation. $vec(X)$ can be thought as a forecast of the state trajectory, as a function of X_0. It shows the probable states in the future. The entire time step in the horizon need not be considered as the forecast, but two to three time steps ahead of present time can easily be considered as a forecast or as at least as the direction of the states, given that the time steps are small. This can unlock another potential use of this proposed controller, where it is necessary to know the error dynamics ahead of time.

Instead of Haar wavelet, if simple block pulse function is considered as basis function, then the transformation matrix is an identity matrix, and $N_* = 1$. This will reduce the amount of computation more, but we will loose the frequency information about the optimal control horizon, and only have time information. With the proposed Haar based MPC, it is possible to extract frequency information as well as time information with the least possible amount of computation.

Haar wavelet based MPC optimizes the system in different frequency bands. Considering separate penalizing terms for different frequency bands, control inputs in those bands can be optimized separately. This can also provide a mean for disturbance filtering.

The focus of this thesis was to reduce computation load without compromising performance. For simplicity in the derivations, time invariant systems without constraints on control and states are considered. But this concept can be extended to problems with constraints by replacing them with differentiable functions of an unconstrained parameter:

$$a \leq x \leq b \quad \Rightarrow x = \frac{b-a}{1+e^{-\tilde{x}}} + a \tag{V.1}$$

$$x \leq a \quad \Rightarrow x = e^{\tilde{x}} + a \tag{V.2}$$

For equality constraints $g(x) = 0$, $g(x)$ is introduced in the cost function by using a Lagrange multiplier:

$$J = x^T Q x + u^T R u + \lambda^T g(x) \tag{V.3}$$

These functions can be expanded in the Haar domain, and using the same procedure, constraints can be included in the derivations.

Model-plant mismatch was not considered. Plant parameter adaptation can be included to solve this problem. State delay can also be considered by including a state delay operational matrix [10] in the derivations.

REFERENCES

[1] S. Mallat, *A wavelet tour of signal processing.* Academic press, 1999.

[2] A. Graps, "An introduction to wavelets," *in Computational Science & Engineering, IEEE*, vol. 2, no. 2, pp. 50–61, 1995.

[3] A. Haar, "Zur theorie der orthogonalen funktionensysteme," *in Mathematische Annalen*, vol. 69, no. 3, pp. 331–371, 1910.

[4] D. Gabor, "Theory of communication. part 1: The analysis of information," *in Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, 1946.

[5] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[6] I. Daubechies *et al.*, *Ten lectures on wavelets*, vol. 61. SIAM, 1992.

[7] C. Chen and C. Hsiao, "A state-space approach to walsh series solution of linear systems," *in International Journal of Systems Science*, vol. 6, no. 9, pp. 833–858, 1975.

[8] C. Cheng, Y. Tsay, and T. Wu, "Walsh operational matrices for fractional calculus and their application to distributed systems," *in Journal of the Franklin Institute*, vol. 303, no. 3, pp. 267–284, 1977.

[9] C. Chen and C. Hsiao, “Haar wavelet method for solving lumped and distributed-parameter systems,” in *in IEEE Proceedings on Control Theory and Applications*, vol. 144, pp. 87–94, IET, 1997.

[10] H. Karimi, “Haar wavelet-based optimal control of time-varying state-delayed systems: A computational method,” *in Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, pp. 1528–1533, 2006.

[11] H. Jaddu, “Direct solution of nonlinear optimal control problems using quasi-linearization and chebyshev polynomials,” *in Journal of the Franklin Institute*, vol. 339, no. 4, pp. 479–498, 2002.

[12] M. A. Tavallaei and B. Tousi, “Closed form solution to an optimal control problem by orthogonal polynomial expansion,” *in American Journal of Engineering and Applied Sciences*, vol. 1, no. 2, pp. 104–109, 2008.

[13] M. Athans and P. L. Falb, *Optimal control: an introduction to the theory and its applications*. Dover Publications, 2006.

[14] C. Hsiao and W. Wang, “Optimal control of linear time-varying systems via haar wavelets,” *in Journal of optimization theory and applications*, vol. 103, no. 3, pp. 641–655, 1999.

[15] A. A. A. Haya, “Solving optimal control problem via chebyshev wavelet,” Master’s thesis, Islamic University of Gaza, 2011.

[16] H. Karimi, B. Lohmann, P. J. Maralani, and B. Moshiri, “A computational method for solving optimal control and parameter estimation of linear systems using haar wavelets,” *in International Journal of Computer Mathematics*, vol. 81, no. 9, pp. 1121–1132, 2004.

[17] H. Karimi, B. Moshiri, B. Lohmann, and P. J. Maralani, “Haar wavelet-based approach for optimal control of second-order linear systems in time domain,” *in Journal of Dynamical and Control Systems*, vol. 11, no. 2, pp. 237–252, 2005.

[18] W. B. Dunbar, *Distributed receding horizon control of multiagent systems.* PhD thesis, California Institute of Technology, 2004.

[19] B. B. Hubbard, *World According To Wavelests: The Story Of A Mathematical Technique In Making.* Universities Press, 1998.

[20] J. M. Rassias, *On the Heisenberg-Pauli-Weyl inequality.* National and Capodistrian University of Athens, Section of Mathematics and Informatics, 2004.

[21] A. V. Oppenheim, R. W. Schafer, J. R. Buck, *et al.*, *Discrete-time signal processing*, vol. 5. Prentice Hall Upper Saddle River, 1999.

[22] O. Rioul and M. Vetterli, “Wavelets and signal processing,” *in IEEE sisgnal processing*, pp. 14–38, October 1991.

[23] R. X. Gao and R. Yan, *Wavelets: Theory and applications for manufacturing.* Springer, 2011.

[24] I. Daubechies, “Orthonormal bases of compactly supported wavelets,” *in Communications on pure and applied mathematics*, vol. 41, no. 7, pp. 909–996, 1988.

[25] M. Razzaghi and G. N. Elnagar, “Linear quadratic optimal control problems via shifted legendre state parametrization,” *in International journal of systems science*, vol. 25, no. 2, pp. 393–399, 1994.

[26] M. Razzaghi and S. Yousefi, “The legendre wavelets operational matrix of integration,” *in International Journal of Systems Science*, vol. 32, no. 4, pp. 495–502, 2001.

[27] H. Jaddu, *Numerical Methods for solving optimal control problems using chebyshev polynomials.* PhD thesis, School of Information Science, Japan Advanced Institute of Science and Technology, 1998.

[28] R. Alt, “A-stable one-step methods with step-size control for stiff systems of ordinary differential equations,” *in Journal of Computational and Applied Mathematics*, vol. 4, no. 1, pp. 29–35, 1978.

[29] F. Berlin and P. Frank, “Design and realization of a mimo predictive controller for a 3-tank system,” *in Advances in model based predictive control Conference*, pp. 446–457, 1994.

[30] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control.* John Wiley & Sons New York, 2006.

[31] B. Yu, *Modeling, control design and mechatronic implementation of constrained robots for surface finishing applications.* PhD thesis, Oklahoma State University, 2000.

[32] P. R. Pagilla and Y. Zhu, “Adaptive control of mechanical systems with time-varying parameters and disturbances,” *in Journal of Dynamic Systems, Measurement, and Control*, vol. 126, no. 3, pp. 520–530, 2004.

[33] P. R. Pagilla and M. Tomizuka, “An adaptive output feedback controller for robot arms: stability and experiments,” *in Automatica*, vol. 37, no. 7, pp. 983–995, 2001.

[34] D. Gu and H. Hu, “Receding horizon tracking control of wheeled mobile robots,” *in IEEE Transactions on Control Systems Technology*, vol. 14, no. 4, July 2006.

[35] A. Grossmann and J. Morlet, "Decomposition of hardy functions into square integrable wavelets of constant shape," *in SIAM journal on mathematical analysis*, vol. 15, no. 4, pp. 723–736, 1984.

[36] A. Teolis, *Computational signal processing with wavelets.* Springer, 1998.

[37] J. B. Burl, *Linear Optimal Control: H (2) and H (Infinity) Methods.* Addison-Wesley Longman Publishing Co., Inc., 1998.

APPENDICES A

ORTHONORMAL BASES

Two real functions are said to be orthogonal to each other if their inner product is zero.

$$\langle f_1(t), f_2(t)\rangle = \int f_1(t)f_2(t)dt = 0 \tag{A.1}$$

When the functions are orthonormal, they are orthogonal to each other and their energy is 1.

$$\int f_n(t)f_m(t)dt = \begin{cases} 1 & \text{if} \quad m = n \\ 0 & \text{if} \quad m \neq n \end{cases} \tag{A.2}$$

If a signal is projected against a set of orthonormal bases, then the coefficients will be given as

$$c_n = \int x(t)f_n(t)dt \tag{A.3}$$

For orthonormal bases, there is no redundancy in forward transformation, so the transformation coefficients can be used as weight functions for reconstruction of the signal from the bases.

$$x(t) = \sum_n c_n f_n(t) \tag{A.4}$$

Since the bases are orthonormal, the coefficients are independent of each other, energy of the signal are exactly in the coefficients, and projection and reconstruction of the signal can be done for each coefficients.

APPENDICES B

WAVELET FAMILIES

Some commonly used wavelets are presented are here. In the first section, wavelets for continuous wavelet transform are presented. In the second section, wavelets for discrete wavelet transform are presented. Some wavelets are complex, some are real. Complex wavelets can separate both the amplitude and phase components of a signal. Wavelets and their associated scaling functions are also illustrated in the following sections.

B.1 Wavelets used in Continuous Wavelet Transform

These wavelets are only used for analysis. They are not compactly supported. Even though they produce perfect reconstruction in theory, they are redundant in presentation.

B.1.1 Morlet Wavelet

Morlet wavelet is defined by [35]

$$\psi(t) = \frac{1}{\sqrt{\pi\omega_b}} e^{2\pi i \omega_c t} e^{-\frac{t^2}{\omega_b}} \tag{B.1}$$

ω_b is the bandwidth parameter and ω_c is the center frequency. It can extract both the amplitude and phase information from the signal. It is obtained by windowing a sine wave by a Gaussian window.

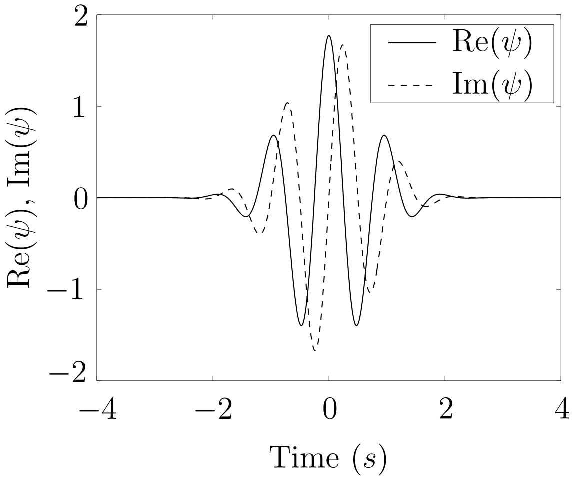


Figure B.1: Amplitude of real and imaginary part of Morlet wavelet for $\omega_b = 1$ Hz and $\omega_c = 1$ Hz.

B.1.2 Gaussian Wavelet

A Gaussian wavelet is expressed as [36]

$$\psi(t) = C_N \frac{d^N f(t)}{dt^N} \tag{B.2}$$

Where $f(t)$ is the Gaussian function defined as

$$f(t) = e^{-it} e^{-t^2} \tag{B.3}$$

$N \geq 1$ is an integer which defines the order of the Gaussian wavelet. C_N ensures that $||f^N(t)||^2 = 1$.

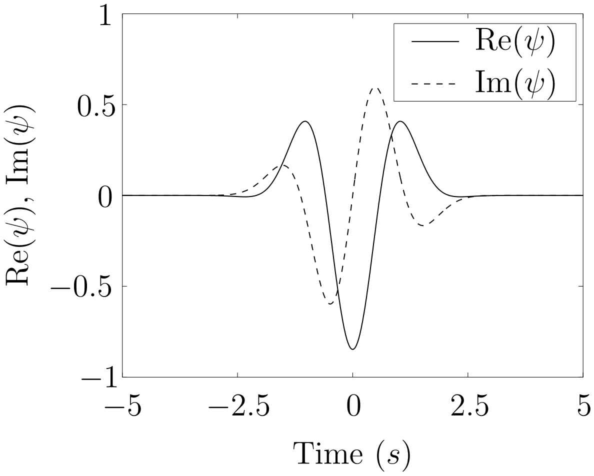


Figure B.2: Amplitude of real and imaginary part of Gaussian wavelet for $N = 2$.

B.1.3 Mexican Hat Wavelet

It is normalized, second derivative of Gaussian function, it is defined by [1, 23]

$$\psi(t) = \frac{1}{\sqrt{2\pi}\sigma^3}\left(1 - \frac{\sigma^2}{t^2}\right)e^{-t^2/2\sigma^2} \tag{B.4}$$

It is a real wavelet , it does not have any imaginary part.

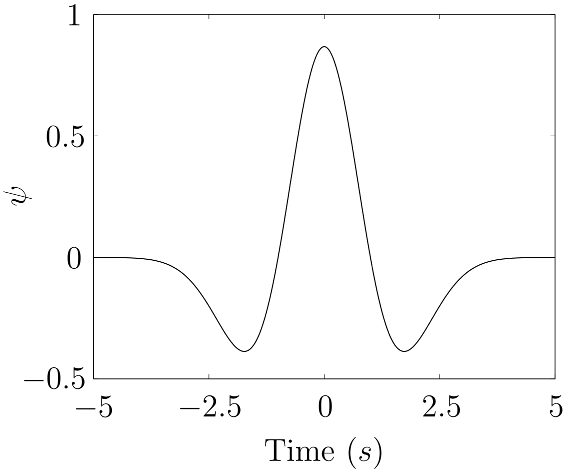
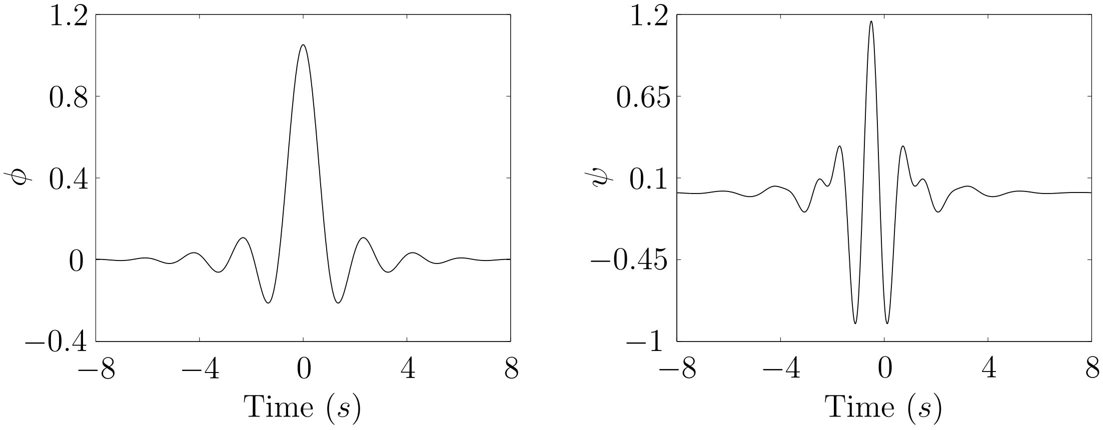


Figure B.3: Amplitude of Mexican Hat wavelet for $N = 2$.

B.1.4 Meyer Wavelet

Unlike most of the continuous wavelets, Meyer wavelet contains a scalar function.



(a) Amplitude of Meyer scalar function

(b) Amplitude of Meyer wavelet function

Figure B.4: Amplitude of Meyer scalar and wavelet function.

B.1.5 Frequency B-Spline Wavelet

Mathematical expression of Frequency B-Spline wavelet is [36]

$$\psi(t) = \sqrt{\omega_b}\left[\text{sinc}\left(\frac{\omega_b t}{p}\right)\right]^p e^{2\pi i \omega_c t} \tag{B.5}$$

ω_b is bandwidth parameter, ω_c is wavelet center frequency, and $p \geq 2$ is an integer. sinc is defined as

$$\text{sinc}(x) = \begin{cases} 1 & x = 0 \\ \frac{\sin x}{x} & \text{otherwise} \end{cases} \tag{B.6}$$

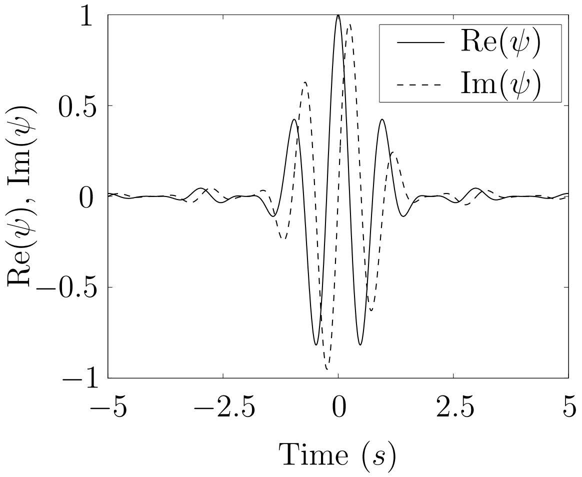


Figure B.5: Amplitude of real and imaginary part of frequency B-spline wavelet for $\omega_b = 1$ Hz, $\omega_c = 1$ Hz and $p = 2$.

Shannon wavelet is a special case of frequency B-spline wavelet for $p = 1$.

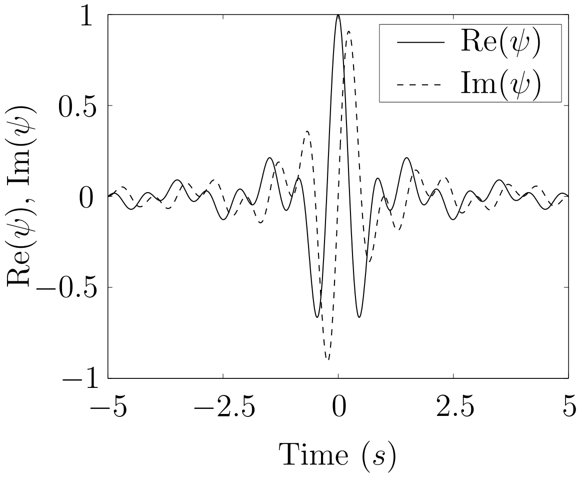
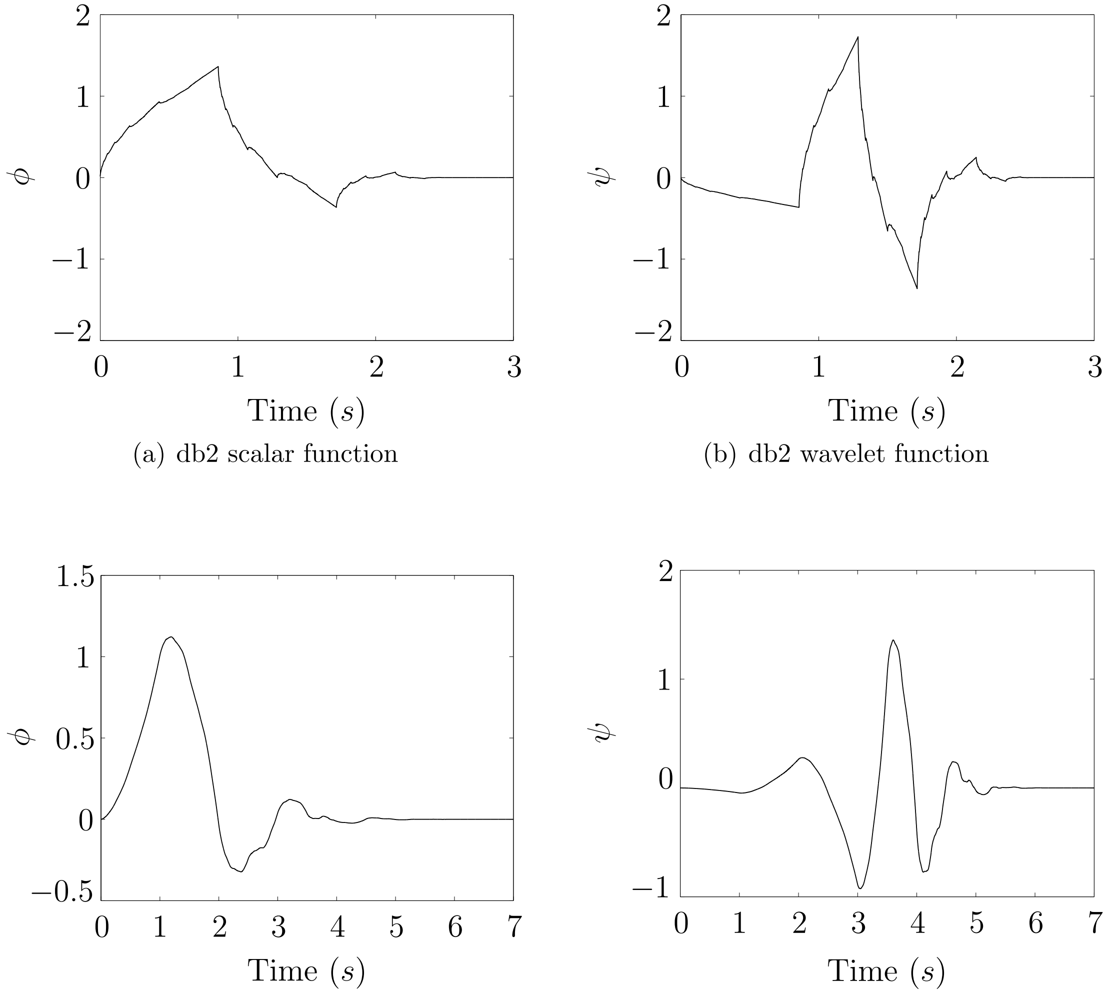


Figure B.6: Amplitude of real and imaginary part of Shannon wavelet for $\omega_b = 1$ Hz and $\omega_c = 1$ Hz.

B.2 Wavelets used in Discrete Wavelet Transform

B.2.1 Daubechies Wavelet

Daubechies wavelets are orthogonal asymmetric wavelets. It is compactly supported with a support width of $2^p - 1$, p is the order of the scalar function [6]. With the increase of order of the scalar function, wavelets become smoother, which leads to better frequency localization. Daubechies wavelets of order 1 is actually Haar wavelet, which has been discussed in detail II.3.4. Daubechies scalar and wavelet functions of order 2 and 4 are shown in the following figure.



(a) db2 scalar function

(b) db2 wavelet function

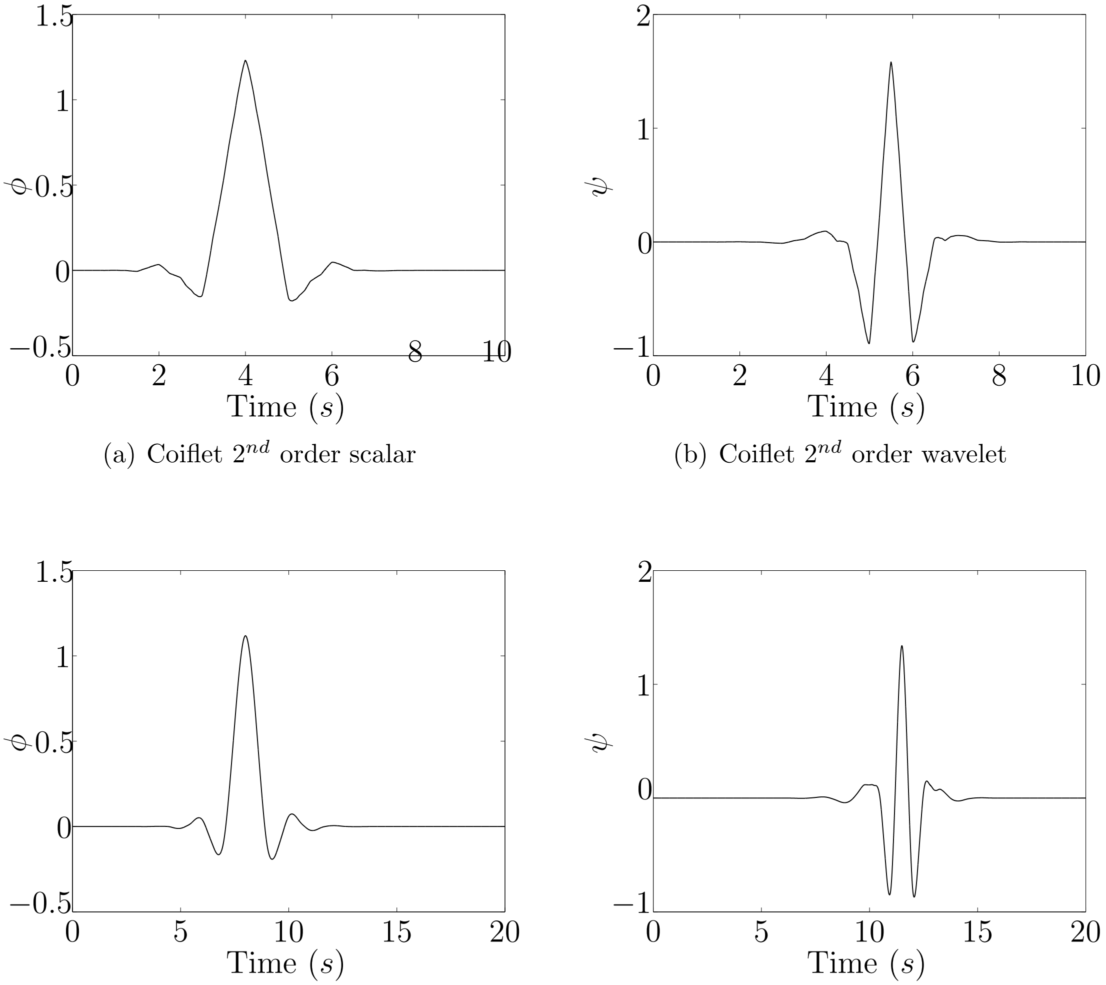
(c) db4 scalar function

(d) db4 wavelet function

Figure B.7: Daubechies scalar and wavelet function of order 2 and 4 respectively.

B.2.2 Coiflets Wavelet

Coiflets wavelets are orthogonal and near symmetric [6]. They produce $2p$ number of vanishing moment for both scalar and wavelet function of order p, with a given support width of $6p - 1$.



(a) Coiflet 2^{nd} order scalar

(b) Coiflet 2^{nd} order wavelet

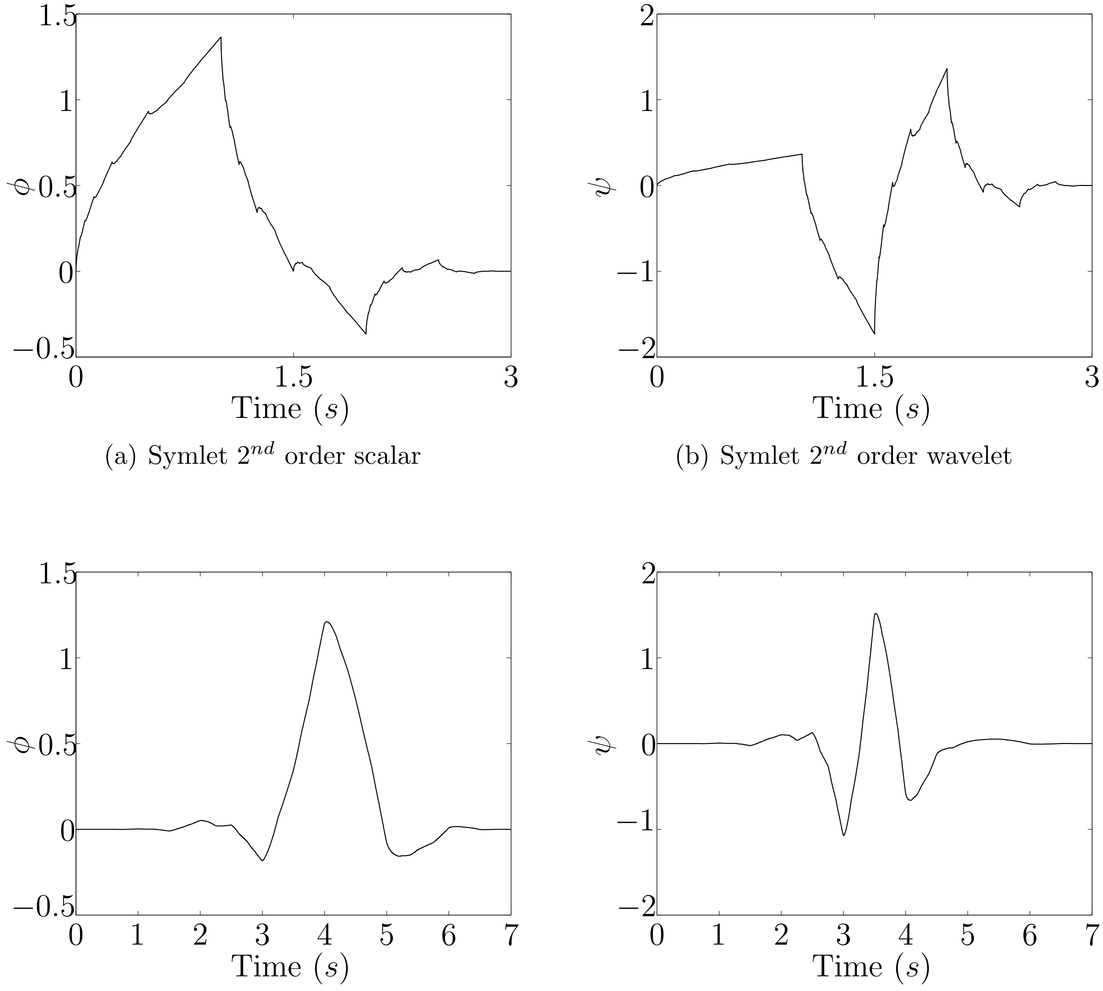
(c) Coiflet 4^{th} order scalar

(d) Coiflet 4^{th} order wavelet

Figure B.8: Coiflet scalar and wavelet function of order 2 and 4 respectively.

B.2.3 Symlet Wavelet

Symlet wavelets are orthogonal and near symmetric [6]. A Symlet wavelet of p order has p vanishing moment for a support of $2p - 1$.



(a) Symlet 2^{nd} order scalar

(b) Symlet 2^{nd} order wavelet

(c) Symlet 4^{th} order scalar

(d) Symlet 4^{th} order wavelet

Figure B.9: Symlet scalar and wavelet function of order 2 and 4 respectively.

APPENDICES C

SOME PROPERTIES OF KRONECKER PRODUCT, VECTOR, TRACE AND MATRIX DERIVATIVES

Let $A : p \times q$, $B : q \times r$, $C : r \times s$, $D : s \times p$, $E : m \times n$ and $F : m \times m$ be fixed matrices, and $x : m \times 1$, $y(x) : n \times 1$ be vectors.

$$vec(ABC) = (C^T \otimes A)vec(B) \tag{C.1}$$

$$tr(ABCD) = vec(B^T)^T(A^T \otimes C)vec(D) \tag{C.2}$$

$$E^T FE = Tr(EE^T F) \tag{C.3}$$

$$\frac{\partial f(y(x))}{\partial x} = \frac{\partial y(x)^T}{\partial x}\frac{f(y)}{\partial y} \tag{C.4}$$

$$\frac{\partial(x^T Ax)}{\partial x} = (A + A^T)x \tag{C.5}$$

$$\frac{\partial x^T A}{\partial x} = A \tag{C.6}$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1} \tag{C.7}$$

APPENDICES D

WEIGHT MATRICES IN LINEAR QUADRATIC COST FUNCTION

[37] Consider a quadratic cost function

$$J(x(t_0), u(t), t) = \int_{t_0}^{T_f} (u^T Ru + x^T Qx)dt \tag{D.1}$$

Subjected to constraints

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(t_0) = x_0, \quad y(t) = Cx(t) \tag{D.2}$$

Q and R are nonnegative definite and positive definite matrices, respectively.

A natural extension of the cost function will be

$$J(x(t_0), u(t), t) = \int_{t_0}^{T_f} (u^T Ru + y^T Qy)dt \tag{D.3}$$

A smooth trajectory of $y(t)$ is also desired. So the cost function can be generalized as follows:

$$J(x(t_0), u(t), t) = \int_{t_0}^{T_f} (u^T Ru + x^T Q_1 x + y^T Q_2 y)dt \tag{D.4}$$

Q_1 and Q_2 are nonnegative definite symmetric matrices.

$x^T Q_1 x$ minimizes the states of the system, thus minimizes the derivatives of the output and try to produce a smooth response. But this also make $y(t) = Cx(t)$ small.

This may opposes $y(t)$ to go near zero states, which is enforced by y^TQ_2y term. To avoid this complexity, Q_1 is given by

$$Q_1 = [I - C^T(CC^T)^{-1}C]^TQ_3[I - C^T(CC^T)^{-1}C] \tag{D.5}$$

for an arbitrary nonnegative definite symmetric matrix Q_3. Finally, Q can be obtained by

$$Q = Q_1 + C^TQ_2C \tag{D.6}$$

APPENDICES E

ROBOT KINEMATICS AND DYNAMICS

E.1 Forward Kinematics

Let,

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \qquad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{E.1}$$

From Figure IV.1, the position and velocity of center of mass of link 1 are:

$$x_{c2} = \begin{bmatrix} L_3 cos(q_1) \\ L_3 sin(q_1) \end{bmatrix} \qquad v_{c2} = \begin{bmatrix} -L_3 \dot{q}_1 sin(q_1) \\ L_3 \dot{q}_1 cos(q_1) \end{bmatrix} \tag{E.2}$$

The position and velocity of center of mass of motor 2 are:

$$x_{c3} = \begin{bmatrix} L_1 cos(q_1) \\ L_1 sin(q_1) \end{bmatrix} \qquad v_{c3} = \begin{bmatrix} -L_1 \dot{q}_1 sin(q_1) \\ L_1 \dot{q}_1 cos(q_1) \end{bmatrix} \tag{E.3}$$

The position and velocity of center of mass of link 2 are:

$$\begin{aligned} x_{c4} &= \begin{bmatrix} L_1 cos(q_1) + L_4 cos(q_1 + q_2) \\ L_1 sin(q_1) + L_4 sin(q_1 + q_2) \end{bmatrix} \\ v_{c4} &= \begin{bmatrix} -L_1 \dot{q}_1 sin(q_1) - L_4(\dot{q}_1 + \dot{q}_2) sin(q_1 + q_2) \\ L_1 \dot{q}_1 cos(q_1) + L_4(\dot{q}_1 + \dot{q}_2) cos(q_1 + q_2) \end{bmatrix} \end{aligned} \tag{E.4}$$

The position and velocity of center of mass of payload are:

$$\begin{aligned}
x_{cp} &= \begin{bmatrix} x_{cpx} \\ x_{cpy} \end{bmatrix} = \begin{bmatrix} L_1 cos(q_1) + L_2 cos(q_1 + q_2) \\ L_1 sin(q_1) + L_2 sin(q_1 + q_2) \end{bmatrix} \\
v_{cp} &= \begin{bmatrix} \dot{x}_{cpx} \\ \dot{x}_{cpy} \end{bmatrix} = \begin{bmatrix} -L_1 \dot{q_1} sin(q_1) - L_1(\dot{q_1} + \dot{q_2}) sin(q_1 + q_2) \\ L_1 \dot{q_1} cos(q_1) + L_2(\dot{q_1} + \dot{q_2}) cos(q_1 + q_2) \end{bmatrix}
\end{aligned} \tag{E.5}$$

v_{cp} can be written as

$$v_{cp} = J\dot{q} \tag{E.6}$$

The matrix J is called the Jacobian of the manipulator. It is given by

$$J = \begin{bmatrix} -L_1 sin(q_1) - L_2 sin(q_1 + q_2) & -L_2 sin(q_1 + q_2) \\ L_1 cos(q_1) + L_2 cos(q_1 + q_2) & L_2 cos(q_1 + q_2) \end{bmatrix} \tag{E.7}$$

E.2 Inverse Kinematics

For a given x_{cp}, joint angles q_1 and q_2 can be found out by inverse kinematics. q_2 is obtained by

$$\begin{aligned}
& r^2 = x_{cpx}^2 + x_{cpy}^2 \\
or, \quad & x_{cpx}^2 + x_{cpy}^2 = L_1^2 + L_2^2 + 2L_1L_2cos(q_2) \\
Let, \quad & D_2 := cos(q_2) = \frac{x_{cpx}^2 + x_{cpy}^2 - L_1^2 - L_2^2}{2L_1L_2} \\
\therefore \quad & q_2 = atan2(\pm\sqrt{1 - D_2^2}, D_2)
\end{aligned} \tag{E.8}$$

In the above equations, ± shows the presence of two possible configurations of the robot. q_1 is obtained as [30]

$$q_1 = tan^{-1}\left(\frac{x_{cpy}}{x_{cpx}}\right) - tan^{-1}\left(\frac{L_2 sin(q_2)}{L_1 + L_2 cos(q_2)}\right) \tag{E.9}$$

From equation (E.9), it can be seen that q_1 is determined by q_2. So for a given x_{cp}, there will be two pairs of q_1 and q_2, and these two configurations are known as elbow up and elbow down.

The joint velocities are found from the end effector velocities via the inverse Jacobian

$$\dot{q} = J^{-1} v_{cp} \tag{E.10}$$

J^{-1} is given by [30]

$$J^{-1} = \frac{1}{L_1 L_2 sin(q_2)} \begin{bmatrix} L_2 cos(q_1 + q_2) & L_2 sin(q_1 + q_2) \\ -L_1 cos(q_1) - L_2 cos(q_1 + q_2) & -L_1 sin(q_1) - L_2 sin(q_1 + q_2) \end{bmatrix} \tag{E.11}$$

The determinant of the Jacobian in equation (E.7) is $L_1 L_2 sin(q_2)$. Therefore, J^{-1} does not exist when $q_2 = 0$ or $q_2 = \pi$. This configuration is known as singular configuration.

E.3 Robot Dynamics

The robot dynamics are derived using the following Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i \tag{E.12}$$

where $i = 1, 2$. $L = T - P$ is the Lagrangian, T is the kinetic energy and P is the potential energy of the robot. It is a planar robot, gravitational weights do not affect the motion of the links, so $P = 0$. The kinetic energy is given by

$$\begin{aligned} T = \frac{1}{2}[(I_1 + I_2 + I_{3c})\dot{q_1}^2 + (I_3 + I_4 + I_p)(\dot{q_1} + \dot{q_2})^2 \\ + M_2 v_{c2}^T vc2 + M_3 v_{c3}^T v_{c3} + M_4 v_{c4}^T v_{c4} + M_p v_p^T v_p] \end{aligned} \tag{E.13}$$

From equations (E.2), (E.3), (E.4) and (E.5), substituting v_{c2}, v_{c3}, v_{c4} and v_{cp} in equation (E.13), we get

$$\begin{aligned} T = \frac{1}{2}[(I_1 + I_2 + I_{3c} + M_3 L_1^2 + M_4 L_1^2 + M_p L_1^2 + M_2 L_3^2)\dot{q_1}^2 \\ + (I_3 + I_4 + I_p + M_4 L_4^2 + M_p L_2^2)(\dot{q_1} + \dot{q_2})^2 \\ + 2(M_4 L_1 L_4 + M_p L_1 L_2)\dot{q_1}(\dot{q_1} + \dot{q_2})\cos(q_2)] \end{aligned} \tag{E.14}$$

Now,

$$\begin{aligned} \frac{\partial T}{\partial \dot{q_1}} =& (I_1 + I_2 + I_{3c} + M_3 L_1^2 + M_4 L_1^2 + M_p L_1^2 + M_2 L_3^2)\dot{q_1} \\ &+ (I_3 + I_4 + I_p + M_4 L_4^2 + M_p L_2^2)(\dot{q_1} + \dot{q_2}) \\ &+ (M_4 L_1 L_4 + M_p L_1 L_2)(2\dot{q_1} + \dot{q_2})\cos(q_2) \end{aligned} \tag{E.15}$$

$$\begin{aligned} \frac{\partial T}{\partial \dot{q_2}} =& (I_3 + I_4 + I_p + M_4 L_4^2 + M_p L_2^2)(\dot{q_1} + \dot{q_2}) \\ &+ (M_4 L_1 L_4 + M_p L_1 L_2)\dot{q_1}\cos(q_2) \end{aligned} \tag{E.16}$$

$$\frac{\partial T}{\partial q_1} = 0 \tag{E.17}$$

$$\frac{\partial T}{\partial q_2} = -(M_4 L_1 L_4 + M_p L_1 L_2)\dot{q_1}(\dot{q_1} + \dot{q_2})\sin(q_2) \tag{E.18}$$

We can define the coupled parameters as

$$\begin{aligned} p_1 =& I_1 + I_2 + I_{3c} + I_3 + I + 4 + I_p \\ & + (M_3 + M_4 + M_p)L_1^2 + M_2L_3^2 + M_4L_4^2 + M_pL_2^2 \end{aligned} \tag{E.19}$$

$$p_2 = I_3 + I_4 + I_p + M_4L_4^2 + M_pL_2^2 \tag{E.20}$$

$$p_3 = M_4L_1L_4 + M_pL_1L_2 \tag{E.21}$$

Therefore,

$$\begin{aligned} \frac{d}{dt}\frac{\partial T}{\partial \dot{q_1}} =& p_1\ddot{q_1} + p_2\ddot{q_2} + p_3(2\ddot{q_1} + \ddot{q_2})\cos(q_2) - p_3(2\dot{q_1} + \dot{q_2})\dot{q_2}\sin(q_2) \\ \frac{d}{dt}\frac{\partial T}{\partial \dot{q_2}} =& p_2\ddot{q_1} + p_2\ddot{q_2} + p_3\ddot{q_1}\cos(q_2) - p_3\dot{q_1}\dot{q_2}\sin(q_2) \\ \frac{\partial T}{\partial q_1} =& 0 \\ \frac{\partial T}{\partial q_2} =& -p_3\dot{q_1}(\dot{q_1} + \dot{q_2})\sin(q_2) \end{aligned}$$

Substituting these into equation (E.12) yields,

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = \tau(t) \tag{E.22}$$

where $M(q)$ is the inertia matrix given by

$$M(q) = \begin{bmatrix} p_1 + 2p_3\cos(q_2) & p_2 + p_3\cos(q_2) \\ p_2 + p_3\cos(q_2) & p_2 \end{bmatrix} \tag{E.23}$$

and $C(q, \dot{q})$ is the Coriolis matrix given by

$$C(q, \dot{q}) = \begin{bmatrix} -p_3\dot{q_2}\sin(q_2) & -p_3(\dot{q_1} + \dot{q_2})\sin(q_2) \\ p_3\dot{q_1}\sin(q_2) & 0 \end{bmatrix} \tag{E.24}$$

APPENDICES F

ROBOT TRAJECTORY GENERATION

This section generates smooth reference trajectories in the joint space, given a desired trajectory in the workspace for the end effector. In particular, what is shown in this section, is the generation process of smooth time history of desired joint position $q_d(t)$, velocity $\dot{q}_d(t)$, acceleration $\ddot{q}_d(t)$ and jerk $\dddot{q}_d(t)$ under some constraints. In the example, the robot end effector is required to traverse a circular trajectory in the workspace in a time duration of $[t_0 \;\; T]$=$[0 \;\; 4]$ seconds. This trajectory corresponds to a circle of radius $r_c = 0.10m$ and center $c = (0.48, 0)$. Let $\Theta^d(t)$ is the desired angular displacement of a point on the circle. Then desired position and velocity of the end effector in Cartesian co-ordinates are,

$$\begin{aligned}
x^d_{cpx}(t) =& x_c + r_c \cos(\Theta(t) + \Theta_0) \\
x^d_{cpy}(t) =& y_c + r_c \sin(\Theta(t) + \Theta_0) \\
\dot{x}^d_{cpx}(t) =& - r_c \dot{\Theta}(t) \sin(\Theta(t)) \\
\dot{x}^d_{cpy}(t) =& r_c \dot{\Theta}(t) \cos(\Theta(t))
\end{aligned} \tag{F.1}$$

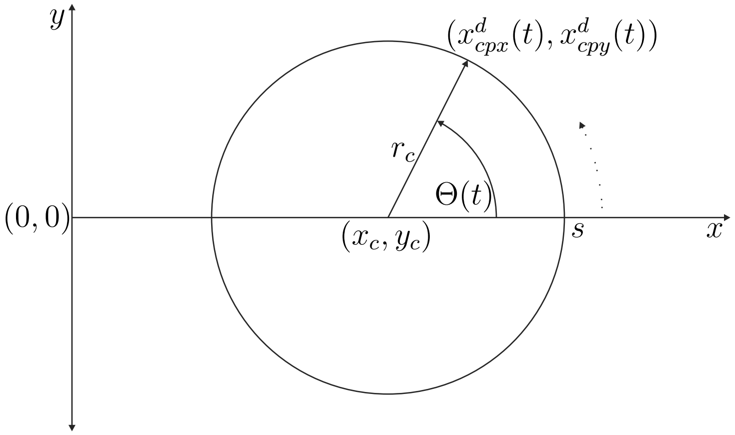


Figure F.1: Desired circular trajectory

Where Θ_0 is the angle at t_0 from which the end effector starts traversing the circle. In our example, the desired starting position of the end effector is $s = (0.58, 0)$, that is $\Theta_0 = 0$. Putting the values in equation (F.1),

$$\begin{aligned} x^d_{cpx}(t) =&0.48 + 0.1\cos(\Theta(t)) \\ x^d_{cpy}(t) =&0.1\sin(\Theta(t)) \\ \dot{x}^d_{cpx}(t) =&-0.1\dot{\Theta}(t)\sin(\Theta) \\ \dot{x}^d_{cpy}(t) =&0.1\dot{\Theta}(t)\cos(\Theta) \end{aligned} \tag{F.2}$$

It is also required that there are no initial and terminal velocity, acceleration and jerk. So, the joint space trajectory $q_d(t)$ to be designed are required to satisfy the following constraints:

$$\begin{aligned} &\Theta(t_0) = 0 \quad \Theta(T) = 2\pi \\ &\dot{\Theta}(t_0) = 0 \quad \dot{\Theta}(T) = 0 \\ &\ddot{\Theta}(t_0) = 0 \quad \ddot{\Theta}(T) = 0 \\ &\dddot{\Theta}(t_0) = 0 \quad \dddot{\Theta}(T) = 0 \end{aligned} \tag{F.3}$$

Since the trajectory satisfies these eight constraints, $\Theta(t)$ is required to be expressed by a polynomial with eight independent coefficients that can be chosen to satisfy these constraints.

$$\Theta(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_7 t^7 \tag{F.4}$$

By differentiating equation (F.4) four times and plugging the eight constraints from equation (F.4), the coefficients can be found by following:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 & t_0^6 & t_0^7 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 & 6t_0^5 & 7t_0^6 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 & 30t_0^4 & 42t_0^5 \\ 0 & 0 & 0 & 6 & 24t_0 & 60t_0^2 & 120t_0^3 & 210t_0^4 \\ 1 & T & T^2 & T^3 & T^4 & T^5 & T^6 & T^7 \\ 0 & 1 & 2T & 3T^2 & 4T^3 & 5T^4 & 6T^5 & 7T^6 \\ 0 & 0 & 2 & 6T & 12T^2 & 20T^3 & 30T^4 & 42T^5 \\ 0 & 0 & 0 & 6 & 24T & 60T^2 & 120T^3 & 210T^4 \end{bmatrix}^{-1} \begin{bmatrix} \Theta(0) \\ \dot{\Theta}(0) \\ \ddot{\Theta}(0) \\ \dddot{\Theta}(0) \\ \Theta(T) \\ \dot{\Theta}(T) \\ \ddot{\Theta}(T) \\ \dddot{\Theta}(T) \end{bmatrix} \tag{F.5}$$

By substituting the values of initial and terminal time and the constraints, the coefficients can be found from equation (F.5).

Once the coefficients are obtained, $\Theta(t)$ can be found out from equation (F.4). Then desired Cartesian position and velocity of the end-effector are found out from equation (F.2). Finally, using the inverse kinematics, q_d and $\dot{q}_d$ are obtained from equations (E.8), (E.9) and (E.10). For the example, sampling time is, $T_s = 0.004$s and $t = kT_s$, where $k = 0, 1, \cdots, 1000$.

VITA

Rushd Md Khaled Julfiker

Candidate for the Degree of

Master of Science

Thesis: RECEDING HORIZON OPTIMIZATION IN HAAR DOMAIN FOR UNCONSTRAINED LINEAR TIME-INVARIANT SYSTEMS

Major Field: Mechanical and Aerospace Engineering

Biographical:

Education:

Received the B.Sc degree from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2009, in Mechanical Engineering. Completed the requirements for the degree of Master of Science with a major in Mechanical and Aerospace Engineering at Oklahoma State University in December, 2013.

Experience:

Research Assistant at Oklahoma State University from August 2012 to November 2012. Teaching Assistant from January 2011 to May 2012.