Vishnu Varadan
21-957-725

Data-Driven Control Methods for Beam Wire Scanners

Master Thesis

SY-BI-PM, CERN
Automatic Control Laboratory
Swiss Federal Institute of Technology (ETH) Zurich

Supervision

Jonathan Emery   Prof. Dr. Florian Dörfler

14 December 2023

Contents

Abstract

This thesis delves into the complexities of a prototype Beam Wire Scanner (BWS) with a Tubular Linear Magnetic Coupling (TLMC), a critical instrument in particle accelerators and beam physics experiments. Employing nonlinear system identification techniques, particularly neural networks, the TLMC system is characterized, revealing its suitability for accurately calculating wire position without significantly impacting beam size accuracy. Optimal reference trajectory generation methods are explored, demonstrating their efficacy in minimizing wire vibration and enhancing the repeatability and reliability of the BWS. Recognizing the impact of evolving mechanical conditions on the Permanent Magnet Synchronous Motor (PMSM), the thesis focuses on novel data-driven motor drive control strategies. Simulations showcase that the proposed Deep Predictive Control (DeePC) architecture adapts well to changing motor parameters, ensuring high performance and adaptability in BWS operations.

Acknowledgement

I would like to express my deepest gratitude to my supervisors, Jonathan Emery and Prof. Dr. Florian Dörfler, for their unwavering guidance, expertise, and support throughout this journey. To Jonathan, I want to express my heartfelt thanks for providing me with the incredible opportunity to work at CERN and engage with Beam Wire Scanners. His mentorship, support, and assistance in setting up the experimental bench have played a pivotal role in realizing a dream and making this work possible. I would like to thank Prof. Dr. Dörfler, for being my tutor and guiding me for the past two years, throughout my master's journey, and for enabling me to carry out this work. His insights and encouragement have always been invaluable. I would also like to thank all my colleagues at the SY-BI Group at CERN, for making this a fun and delightful experience. Finally, I want to express my heartfelt gratitude to my family, for their consistent support and encouragement throughout this academic journey. Their love and understanding have been the pillars of strength, making this a truly enriching and fulfilling endeavour.

List of Figures

List of Tables

Chapter 1

Introduction

This chapter briefly introduces the European Organization for Nuclear Research (CERN) and its particle accelerators, and the constituents of Beam Wire Scanners, the instrument that is the major focus of this thesis. Later, we formulate the problem that this thesis aims to resolve and conclude the chapter with an outline of the thesis.

1.1 CERN and Particle Accelerators

The European Organization for Nuclear Research (In French: Organisation Européene pour la Recherche Nucléaire), CERN (formerly, Conseil Européen pour la Recherche Nucléaire) is the largest particle physics laboratory in the world. Established in 1954, it is spread across the French-Swiss border in the suburbs of Geneva. Starting off with 12 Member States, CERN now has grown to consist of 23 Full-Member States and 10 Associate Member States.

1.1.1 LHC Injectors Chain

The major focus at CERN is to study the interactions between different subatomic particles and for this reason, it houses around a dozen particle accelerators that accelerate protons, heavy ions and other subatomic particles to very high energies. The Large Hadron Collider (LHC) is the largest particle accelerator in the world and one of the largest machines ever built by humans. Built 100 m underground, with about 27 Km of circumference, it is designed to reach collision energies of 2×7 TeV for protons nominally.
However, the journey of a particle to achieve such high collision energies starts from a bottle of hydrogen gas injected into a plasma chamber. An electric field ionises the atoms of hydrogen, stripping it of its electrons leaving the protons behind at about 100 KeV. LINAC4, a linear accelerator then accelerates these protons to an energy of about 160 MeV and the particles travel at around one-third of the speed of light. The Proton Synchrotron Booster (PSB), the smallest circular accelerator at CERN. then brings up the energy of the protons to around 2 GeV at 91.6% the speed of light. The 628m circumferential Proton Synchrotron (PS) ramps up the energy of the particles to 26 GeV while the particles travel at 99.9% the speed of light. The second largest machine at CERN, the Super Proton Synchrotron (SPS), then accelerates the beam of protons to LHC Injection Energy of 450 GeV. Finally, the beam of particles is split into two LHC rings in the opposing directions and brought up to nominal collision energy of 2×7 TeV. Apart from protons, the LHC also accelerates lead ions, at 2.8 TeV. These ions, produced in LINAC3 and accumulated in Low Energy Ion Ring (LEIR), are injected into the PS next and

then follow the same path as the protons. The complete accelerator complex can be seen in Figure 1.1.

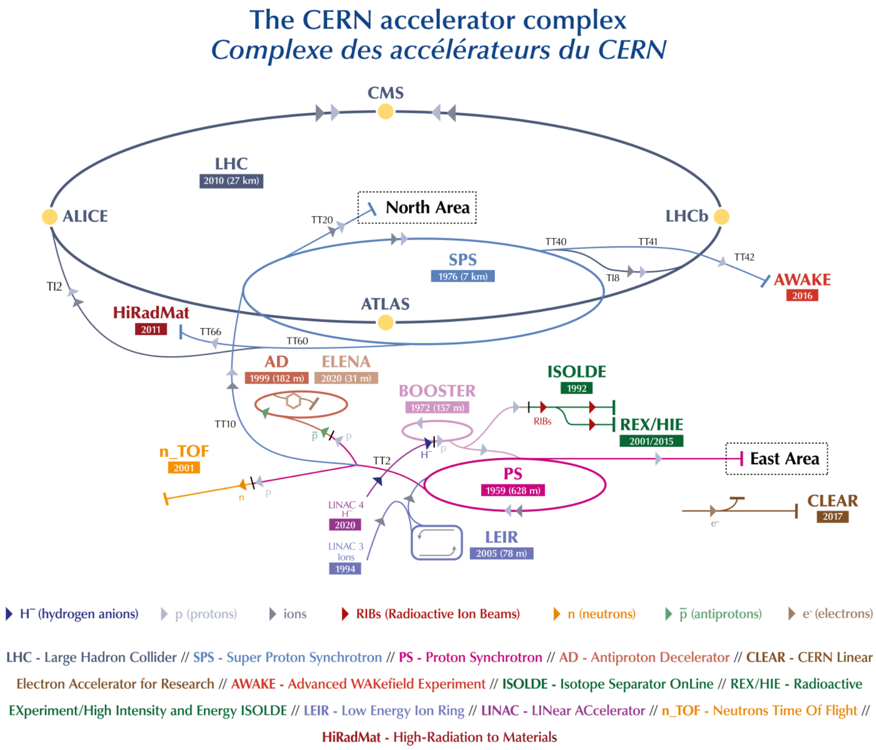


Figure 1.1: CERN Accelerator Complex and Experiments [24]

1.1.2 Experiments at CERN

The particle accelerators provide the high energy particles to different interaction points, where various experiments are located [6]. Each experiment is designed for a specific purpose. The LHC houses nine experiments that use various types of detectors to study the different particles produced by the collisions. ATLAS and CMS are the largest experiments that are general purpose detectors. ALICE is dedicated to heavy ion physics and LHCb studies b quarks. These experiments are situated in a cavern underground. TOTEM and LHCf are the smallest experiments at the LHC that focus on forward particles.

Apart from the experiments at the LHC, CERN also has fixed-target experiments, in which the accelerated beam is made to hit a fixed solid, liquid or gas target which forms the detector. The SPS provides the accelerated beam to most of these type of experiments while the PS also houses some. Two decelerators: Antiproton Decelerator (AD) and ELENA also contain numerous experiments that study the properties of antimatter.

1.2 Beam Profile Measurements

The building blocks of accelerators that guide the beam particles along a reference path are the quadrupole magnets. The magnetic field generated by these magnets affects all the particles of the beam. A series of magnets with alternating polarities is used to focus the beam through the accelerator. The focussing magnets cause the particles to form a Gaussian profile on each of the transversal planes. The rate of interaction of these particles is coupled with the size of the beam, which strongly motivates the accurate measurement of the beam profiles.
In particle physics experiments, the primary parameters are energy and the number of useful interactions. Colliding beams with high center-of-mass energy, especially in studying rare events, with a small cross-section σ_p are essential [14]. Luminosity (L) quantifies a particle accelerator's ability to produce interactions. It relates the event rate per second ($\frac{dR}{dt}$) to the cross-section σ_p as $\frac{dR}{dt} = L\sigma_p$.
When two bunches with particles n_1 and n_2 collide at machine revolution frequency f_{rev} and zero crossing angle, luminosity is given by

$$L = \frac{f_{rev} \cdot n_1 \cdot n_2}{4\pi\sigma_x\sigma_y}$$

where σ_x and σ_y are standard deviations of the Gaussian profile of the beam, also referred to as the 'beam size'.
It is now easy to understand the importance of instruments that measure transverse beam profile. Various instruments with differing principles of operation are employed for this purpose each with its own pros and cons. Depending on whether the instruments come in contact with the beam, they can be classified as interceptive and non-interceptive. Some of the major instruments used for beam profile measurement, and a short description of each instrument is given in Table 1.1.

1.3 Beam Wire Scanners

Beam Wire Scanners (BWS) stand as integral components in particle accelerators, providing crucial insights into the transverse profiles of particle beams. Employing a combination of electro-mechanical components and photoelectric detection systems, BWS plays a pivotal role in measuring and reconstructing beam density profiles with high precision and reliability.
The operational principle of BWS revolves around its ability to interact with proton beams using a slender $30\mu m$ wire, traveling at a rapid displacement speed exceeding 20 m/s within vacuum pipes. As the wire traverses the beam, secondary particle radiation is generated, detected by a scintillating material, and transformed into light. This light undergoes amplification through a photo multiplication process, resulting in a current proportional to the concentration of secondary particles at the wire's location. A rotational wire scanner and a simplified overview of the operation of BWS is shown in Figure 1.2.
BWS can be broadly categorized into two main families: linear and rotational. Linear scanners, present in facilities such as the SPS and LHC, employ bellows for motion transfer from air to vacuum and potentiometers for position measurement. However, the linear architecture limits their scan speed to 1 m/s, impacting the intensity of safely scanned beams. On the other hand, rotational scanners, can reach speeds upto 20 m/s [8]. However, the latter's complex mechanics introduces challenges such as mechanical play and increased measurement uncertainty caused by vibrations during acceleration.
A nominal scan cycle with air side and vacuum side positions is depicted in Figure 1.3. The initial movement when the wire goes from initial position 0 mm to the full stroke length 90 mm

Instrument	**Type**	**Description**
Synchrotron Light Monitor (BSRT)	Non-Interceptive	These monitors profit from the light produced on dipole magnets when highly relativistic particles are deflected by a magnetic field. Synchotron monitors are usually placed to make use of this parasitic light after a dipole and behind an "undulator" magnet, where the beam is deflected several times to enhance the photon emission.
Beam Gas Ionisation Monitor (BGI)	Non-Interceptive	Ionisation of the rest gas (or injected gas) on the vacuum chamber, the electrons generated by ionisation are accelerated by a strong electric field towards one side of the chamber, where they are detected. The footprint of the electron distribution on the detector represents the beam dimensions in one plane.
Beam Gas Vertex Detector (BGV)	Non-Interceptive	The BGV is a beam-gas interaction device. Its beam profile measurement method is based on reconstructing the track of secondary particles, produced by inelastic beam-gas collisions, exiting the beam pipe.
Secondary Emission Grids (SEM-Grids)	Interceptive	SEM-grids consists on a series of thin filaments arranged in parallel. When the beam crosses the grid, a current is generated in each of its filaments by secondary electron emission. The current in each filament is proportional to the beam intensity in the wire position.
Optical Transition Radiation (OTR) Screens	Interceptive	OTR is emitted when a charged particle beam goes through an interface with different dielectric constant. A luminous image with the footprint of the beam is generated on the material offering bi-dimensional information of the beam in a single shot.
Beam Wire Scanners (BWS)	Interceptive	A very thin carbon wire (around 30 μm) is passed through the particle beam. A shower of secondary particles is generated by the beam-wire interaction which is detected with a scintillator, and transformed into an electrical current through a Photo-Multiplier Tube (PMT). The shower intensity is proportional to the beam intensity at a given wire position. The beam profile in a single plane can be obtained by correlating the wire position with the secondaries shower intensity.

Table 1.1: Beam Profile Measurement Instruments [30]

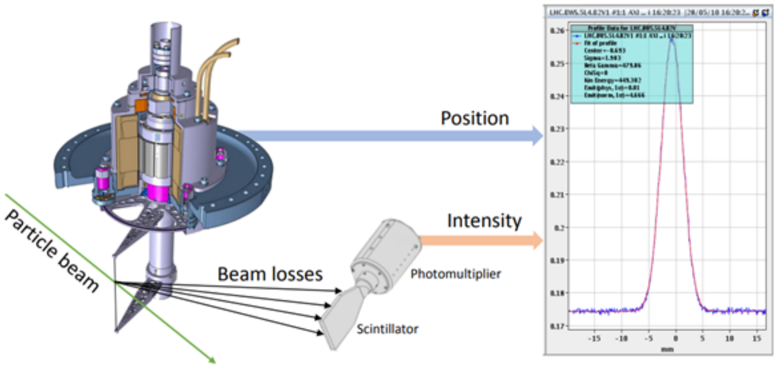


Figure 1.2: Working Principle of BWS [10]

is called the IN Stroke. When the wire comes back to its initial position from the full stroke length, the action is referred to as the OUT Stroke. The two strokes are when the wire is expected to come in contact with the beam.

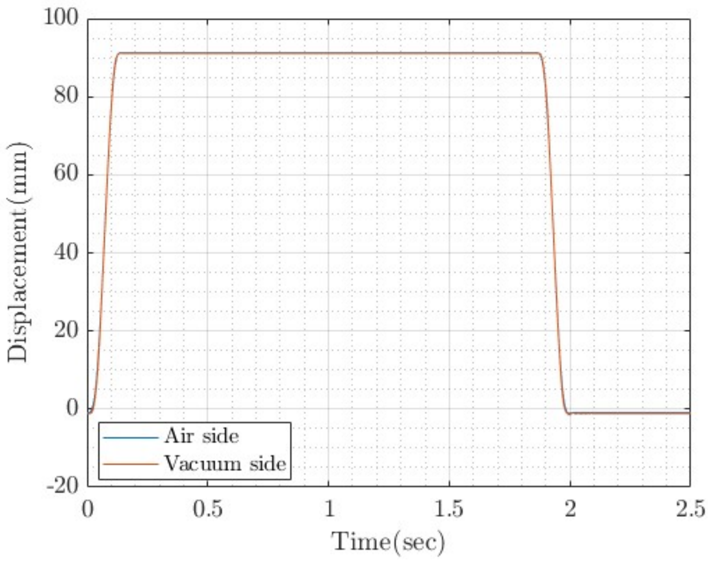


Figure 1.3: A Nominal Scan Cycle of BWS

Installing the mechanical actuator within the extreme conditions of a particle accelerator presents a significant challenge. The environment is characterized by varying pressure, temperature, and magnetic fields, necessitating careful design considerations. Typically, a bellow facilitates motion transfer from the external environment to the vacuum chamber, minimizing outgassing and protecting sensitive components from the baking procedure.
However, the bellows are subject to wear and tear over time and breakage of bellows can compromise the accelerator vacuum. Replacements for bellows, such as a Tubular-type Linear Magnetic Coupling (TLMC) are currently under investigation. This thesis revolves around the characteri-

zation and development of a control scheme for this TLMC based BWS. The following subsections provide an overview of the important components of this prototype of BWS and the experimental setup.

1.3.1 Permanent Magnet Synchronous Motor (PMSM)

One of the major components of the BWS is the motor, that generates the required torque to bring about the motion of the fork and the wire through the beam. Several types of motors have been used in the past, the ones in operation include Brushless DC (BLDC) and Permanent Magnet Synchronous Motor (PMSM). The BWS under investigation, has an AC Motor Drive system with the PMSM motor driving the moving parts.
A Permanent Magnet Synchronous Motor (PMSM) is a type of electric motor that relies on permanent magnets to generate the magnetic field required for its operation. Unlike induction motors, PMSMs have magnets embedded in the rotor, typically made of materials such as neodymium. The stator, on the other hand, carries windings through which alternating current (AC) is supplied. The interaction between the magnetic field generated by the stator windings and the permanent magnets in the rotor produces a rotating magnetic field, inducing motion in the rotor.
The operation of a PMSM is highly dependent on precise control of the current supplied to the stator windings. To achieve optimal performance, the feedback control system adjusts the amplitude and phase of the stator current, ensuring that the rotor follows the rotating magnetic field. This control is often facilitated by sensors, such as encoders or resolver devices, which provide feedback on the rotor's position and speed.
The PMSM motor used in the experimental setup is Kollmorgen AKM23F-ACC2R-00. The important parameters of the motor are listed in Table 1.2.

PMSM Parameter	**Value**
Stator resistance R_S	2.34 Ω
Stator inductance L_S	4.68 mH
Back EMF constant K_E	0.1681 V rad^{-1} s^{-1}
Torque constant K_T	0.27 Nm/A
Pole pairs p	3
DC bus voltage U_{dc}	320 V
Nominal current I_N	4.31 A
Peak current I_p	17.2 A
Rotor inertia J_{rot}	0.232×10^{-4} kg m^2
Max. mechanical speed Ω_m	837.75 rad s^{-1}
Nominal mechanical speed Ω_N	837.75 rad s^{-1}
Peak torque T_p	3.88 Nm
Nominal torque T_N	0.94 Nm
Operating temperature	-20°C to 40°C

Table 1.2: PMSM Parameters and Requirements

1.3.2 Tubular-type Linear Magnetic Coupling (TLMC)

Tubular linear magnetic couplings operate on the principle of utilizing magnetic fields to transmit torque or motion through a non-magnetic barrier, such as a wall or casing. Consisting of magnet arrangements on both sides with aligned polarities, these couplings enable the transfer

of motion without physical contact, providing advantages such as hermetic sealing and prevention of mechanical wear. The interaction of magnetic fields through the barrier facilitates the transmission of rotational or linear motion, making these couplings ideal for applications where a sealed or sterile environment must be maintained on one side while allowing mechanical motion on the other. Their use is prominent in industries such as medical, chemical, and food processing, offering an efficient solution for scenarios requiring isolation and controlled motion transfer. While using radial magnetics for torque transfer is a widely studied application, using tubular-type magnets to transmit motion is a relatively lesser studied field. The TLMCs provide a viable alternative for bellows and form an integral part of motion transfer in the BWS. Figure 1.4 shows the structure and an analytical model of the TLMC.
The experimental setup uses the Ball screw Actuated Linear Translator (BAT) drive designed by UHV Design. It has a linear breakaway thrust of $90N$ and can nominally be operated at a speed of 1 ms^{-1}

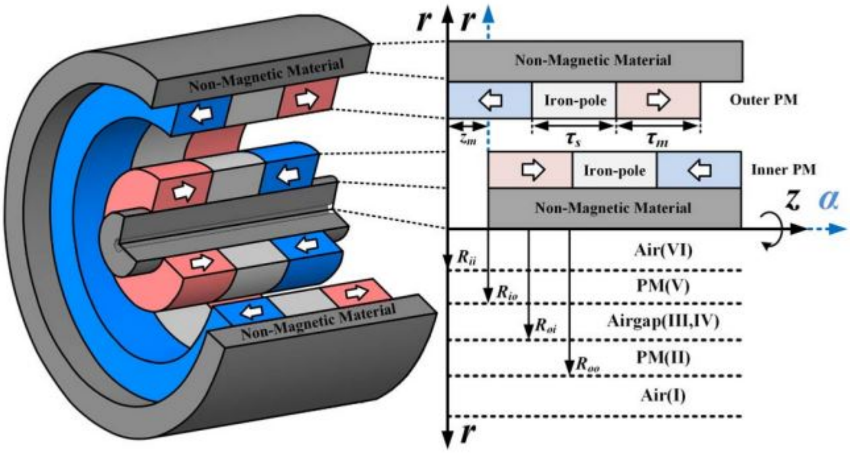


Figure 1.4: Structure and Analytical model of TLMC [18]

1.3.3 Linear Magnetic Encoder

A linear magnetic encoder operates on the principle of utilizing magnetized tracks on a linear scale to provide precise position feedback. The encoder consists of a magnetized scale, typically composed of stainless steel, and a read head containing sensors, such as Hall sensors and anisotropic magnetoresistive (AMR) sensors. As the scale moves relative to the read head, the sensors detect changes in the magnetic field generated by specific magnetization patterns on the scale. Interpolation electronics within the readhead process these variations to achieve higher resolution in position measurements, and custom logic circuitry refines the signals for transmission. The signals are then sent to an internal microcontroller unit (MCU), where specialized algorithms decode the magnetic signatures, determining the absolute position of the linear scale. The resulting position information is output to external systems or controllers, enabling precise control over the position of the associated device or machinery. The contact-less nature of this technology, along with its ability to offer high-resolution feedback, makes linear magnetic encoders suitable for applications requiring accuracy and reliability in motion control and industrial automation.

The Linear Magnetic Encoders fitted to the experimental setup are LA11 manufactured by RLS. They can be configured to work in both absolute and incremental modes and are characterized with a resolution of about $0.244\mu m$.

1.3.4 Fork Assembly and Wire

The part of BWS that travels through the beam line in high vacuum conditions are the fork assembly and the wire. A metallic fork consisting of two arms is fixed on a shaft, which is interfaced to the moving part of the instrument. The wire, made of a single carbon fibre 34 μm in diameter is stretched in the rigid fork and the tension on the wire is maintained due to its elasticity. Carbon with its high melting point, good thermal conductivity and comparatively low density make it a suitable choice of material for the wire.

1.3.5 Secondary Shower Detection

Another important component of the BWS is its detection system, where a scintillating material captures the escaping secondary particles. The scintillator properties are usually dependent on the accelerator. This scintillation-induced light is then amplified by a photo multiplier tube (PMT), converting it into a measurable current signal. A digitized signal from the detector is then processed. The entire process allows for the reconstruction of the transverse beam intensity profile, achieved by plotting the current intensity against the wire's position.

1.3.6 Experimental Setup

The experimental setup consists of the prototype of BWS composed of the PMSM, the TLMC and two Linear Magnetic Encoders, one to measure the displacement in the air side and the other to measure the displacement in the vacuum side. The setup does not contain the fork assembly and secondary particle acquisition system as of now. The motor is controlled real-time by a dSPACE MicroLabBox with a sampling frequency of 16 kHz. The power supply of the motor is developed in-house and deployed to actuate all the wire scanners in operation at CERN [8]. There is also a magnetic brake attached to the PMSM. An image of the experimental setup at CERN is shown in Figure 1.5.

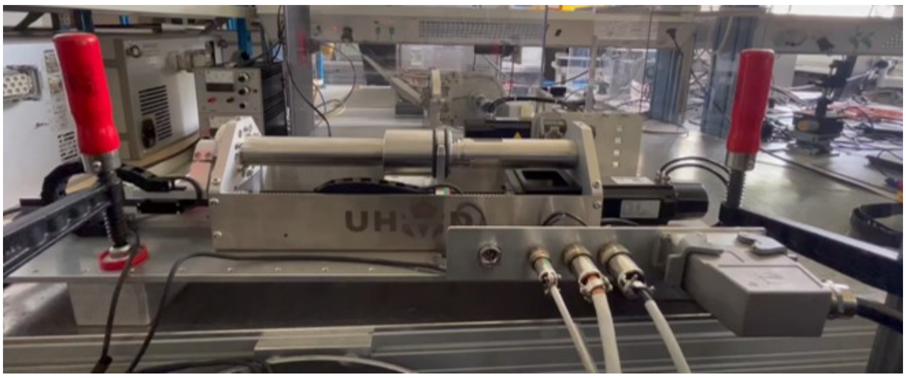


Figure 1.5: Experimental Setup

1.3.7 Operational Systems Overview

At the time of writing of this thesis, there are 17 Rotational BWS, installed in the PSB, PS and SPS as a part of LHC Injectors Upgrade [8]. In addition, there are 8 Linear BWS instruments that are in operation in the LHC.

1.4 Problem Statement

This thesis aims at the complete characterization of the TLMC based BWS prototype and the development of different control schemes that require minimal tuning of parameters, and are robust to parameter changes.
Accurate and precise wire positioning is of great importance to get an accurate profile of the particle beam. Moreover, the wire should not be subjected to vibrations when passing through the beam and possess a reproducible motion profile. Hence, we would like to identify a suitable model for the TLMC and characterize its natural frequency. Optimal Reference Trajectory Generation Methods that produce reproducible and vibration free motion are also to be studied. The current iteration of the control for PMSM drive uses FOC containing three cascade PID loops. Changes in motor parameters over time, due to change in orientation and other mechanical causes causes the control scheme to under perform thus necessitating an adaptive, self-commissioning like technique. Hence, we turn our attention towards data-driven control methods and study the feasibility and real-time application capabilities of such methods.
Summarizing, the major goals of the thesis are as follows:

1. Characterization of the TLMC using System Identification techniques
2. Implementation of Optimal S-Curve (Reference Trajectory) generation techniques and study its effects on reproducibility of motion
3. Analysis of different data-driven control methods for the PMSM drive and compare them
4. Experimental verification and validation of the above methods using the experimental setup in real-time

1.5 Thesis Outline

We start with the different system identification techniques in Chapter 2, provide the necessary theory and their applications to TLMC-BWS. We discuss and compare the results obtained from the different techniques. In Chapter 3, we present two different approaches for optimal trajectory generation, implement it on the real-time system and compare them. We move on to the control problem in Chapter 4, where we introduce the mathematical fundamentals required for the understanding of the different data-driven control methods including a review of the Model Predictive Control basics and relevant theory, notations from Behavioral Systems. Finally, in Chapter 5 we propose different data-driven control schemes for application to BWS and analyze the results obtained. We present a summary of the work done and potential directions for future research in Chapter 6.

Chapter 2

System Identification and Data-driven Characterization of TLMC

As described in Section 1.3, the rotational motion generated by the motor is translated into linear motion using a ball screw. This motion happens outside of the beam line, outside vacuum, what we term as the 'air side'. It is transmitted to a fork attached that is present in the beam line through a Tubular-type Linear Magnetic Coupling (TLMC). It is of paramount importance to know the precise position of the fork and the wire in the 'vacuum side'. However, the sensors need to be designed to be radiation hard. In this chapter, we characterize the TLMC and analyze if the vacuum side position can be predicted just with data from the air side and how the predictions ultimately affect the beam size estimations.

2.1 Wire Position Accuracy

Prior to delving into the system identification methods, it is important to establish the importance of the accuracy of the wire position measurement in the vacuum side. An extensive analysis and experimental simulations on the contribution of all the sources of error that can occur in the Beam Wire Scanner is detailed in [30]. In this section we only focus on the relevant parts, the contribution of error in position measurement to the relative beam size error.
As explained in 1.2, the nominal profile of a beam can be expected to be described by a Gaussian curve (shown in Figure 2.1) and the size of the beam is given by its standard deviation, σ_{beam}. The relative beam size error δ_{rel} is defined as,

$$\delta_{rel} = \frac{\sigma_{measured} - \sigma_{beam}}{\sigma_{beam}}$$

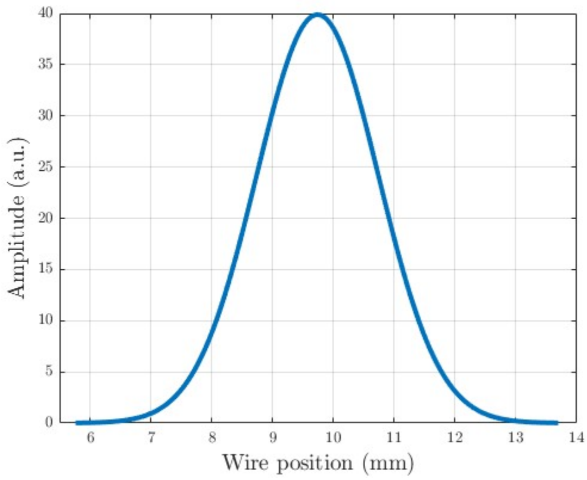


Figure 2.1: Simulation of Nominal Beam profile with $\sigma_{beam} = 1mm$

with $\sigma_{measured}$ being the size of the beam measured by the BWS. A histogram of measured beam sizes, with a sufficiently large number of entries can be approximated by a Gaussian distribution, with a mean μ_σ and a standard deviation σ_σ, as shown in Figure 2.2. The difference between the actual beam size and the mean of the histogram represents the 'systematic error', while the standard deviation of the histogram gives the 'statistic' or 'random error'. Mathematically, they can be described as,

$$\epsilon_{systematic} = \frac{\sigma_{beam} - \mu_\sigma}{\sigma_{beam}}$$
$$\epsilon_{statistic} = \frac{\sigma_\sigma}{\sigma_{beam}}$$

Studies performed in [30] show that the $\epsilon_{systematic}$ does not depend on the accuracy of the wire position but only on the wire diameter. However, the statistic error, $\epsilon_{statistic}$, depends on the error of wire position (ϵ_{SNRx}), error of the photomultiplier tube (PMT) amplitude (ϵ_{SNRy}) and the error due to number of particles crossing the detector surface (ϵ_{NTE}), and is given by,

$$\epsilon_{statistic} = \sqrt{\epsilon^2_{SNRx} + \epsilon^2_{SNRy} + \epsilon^2_{NTE}}$$

Consider n two-dimensional points representing a beam profile, $(x_{pred,i}, \bar{y}_i) \in \mathbb{R}^2, \forall i \in [n]$ where $x_{pred,i}$ represents the predicted value of wire position and $\bar{y}_i$ represents the wire amplitude from the PMTs. Let $x_{true,i}$ represent the true value of the wire position. The difference between $x_{pred,i}$ and $x_{true,i}$ gives the absolute error in the prediction. Let each $x_{pred,i}$, due to the influence of the prediction error, be a Gaussian Random variable with mean $\bar{x}_{pred,i}$ and a standard deviation $\sigma_{pred,i}$. When the mean of the random variable, for some γ, can be expressed as,

$$\bar{x}_{pred,i} = x_{true,i} + \gamma, \forall i \in [n]$$

we can write the error of the wire position as,

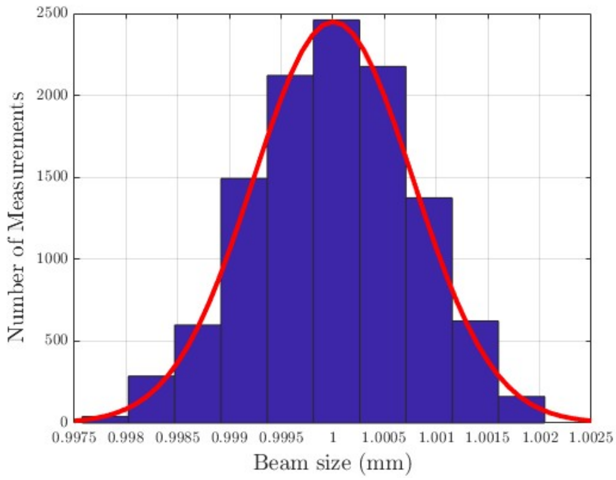
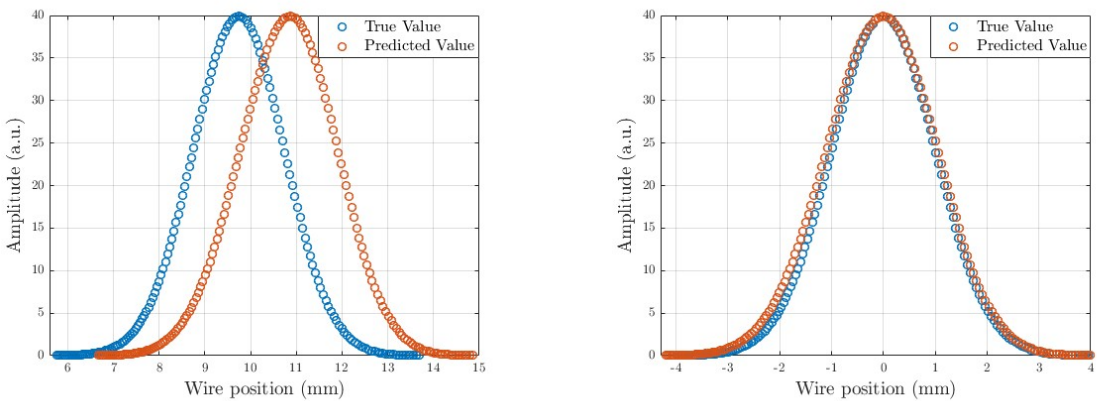


Figure 2.2: Simulation of Nominal Beam Histogram and Gaussian Approximation

$$\epsilon_{SNRx} = \frac{0.4 \max_i \sigma_{pred,i}}{f(\sigma_{beam}, v_{scan})}$$

where $f(\sigma_{beam}, v_{scan}) = \sigma_{beam}\sqrt{\frac{\sigma_{beam}}{v_{scan}} f_{rev}}$ with f_{rev}, the machine revolution frequency and v_{scan} the velocity of a scan cycle. The value γ, typically chosen to ensure that the centre of the beam lies at 0, can be imagined to be a factor that causes a shift in the beam position. The effect of the centering of the beam is pictorially represented in Figure 2.3. It however has no impact on $\epsilon_{statistic}$ and thus the beam size calculations. An experimental verification of this is presented in subsequent subsections of this chapter. Calculation of ϵ_{SNRy} and ϵ_{NTE} lie outside the scope of this document and more information can be obtained from [30].



(a) Before Centering (b) After Centering

Figure 2.3: Beam Size Comparison

For a nominal beam of $100\mu m$ at the LHC with the scanner moving at a speed of $1ms^{-1}$ and a machine revolution frequency of $11kHz$, the contribution of ϵ_{SNRy} and ϵ_{NTE} turn out to be 3.33×10^{-5} and 3.66×10^{-5} respectively. In order to measure the beam within an accuracy of 1%, the value of $\max_i \sigma_{pred,i}$ turns out to be $4\mu m$.
The major inferences from this section are,

- The inaccuracies in wire position affect the beam size calculation only through $\epsilon_{statistic}$
- Irrespective of the value of γ, $\epsilon_{statistic}$ can be reduced by bounding $\max_i \sigma_{pred,i}$ suitably, which is expected to be $4\mu m$ under typical working conditions

2.2 Linear Methods

Several works have previously been done to study the magnetic characteristics of the TLMC [18]. However, no previous works are known to take a system-theoretical approach to characterize the force and the motion associated with the TLMC to the knowledge of the author. We consider the TLMC to be a black-box and use system identification techniques to come up with a model for the TLMC that can be used to predict the position in the vacuum side, based on the position on the air side. We start by assuming the TLMC exhibits only linear characteristics, since there is no evidence suggesting otherwise. In this section, we study and analyze the effectiveness of two linear system identification techniques - one in frequency domain and one in time domain. Mathematically, we assume that the TLMC can be expressed as is,

$$y(t) = G(s)u(t) + H(s)d(t)$$

where, $y(t)$ is the vacuum side displacement, $u(t)$ is the air side displacement, $G(s)$ is the transfer function that describes the relationship between the two displacements in Laplace domain, and the term $H(s)d(t)$ denotes how all the disturbances affect the vacuum side displacement value.

2.2.1 Empirical Transfer Function Estimate

The physical interpretation of any transfer function of a system $G(s)$ the complex number given by $G(e^{i\omega})$ gives the amplitude and phase change that happens to the output of the system, when it is fed by an input sinusoidal signal at a frequency of ω. The frequency response of a system can then be obtained by choosing an input signal $u(t) = \alpha \sin \omega t, \forall t$, calculating an estimate of the amplitude and phase shift of the input signal and repeating the process for several frequencies, ω corresponding to a region of interest.
Extending the above idea to the case of multi-frequency inputs, the estimate of a transfer function given by,

$$\hat{G}_N(e^{i\omega}) = \frac{Y_N(e^{i\omega})}{U_N(e^{i\omega})}$$

$$\text{where, } Y_N(e^{i\omega}) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} y(t)e^{-i\omega t}$$

$$U_N(e^{i\omega}) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} u(t)e^{-i\omega t}$$

the ratio between the N-point Discrete Fourier Transform (DFT) of the output signal and the input signal is called the Empirical Transfer Function Estimate (ETFE) of the system.

Input Signal Selection

As in most real-world applications, there exist a number of challenges on the choice of the input signal. The most important one being that the selected input signal cannot be applied directly to the air side displacement $u(t)$, but can only be given as a reference trajectory $\theta_{ref}(t)$ to be followed by the PMSM control system. The air side displacement $u(t)$, can only be measured and not manipulated directly. Thus, we are limited by the control capabilities of the existing control system.

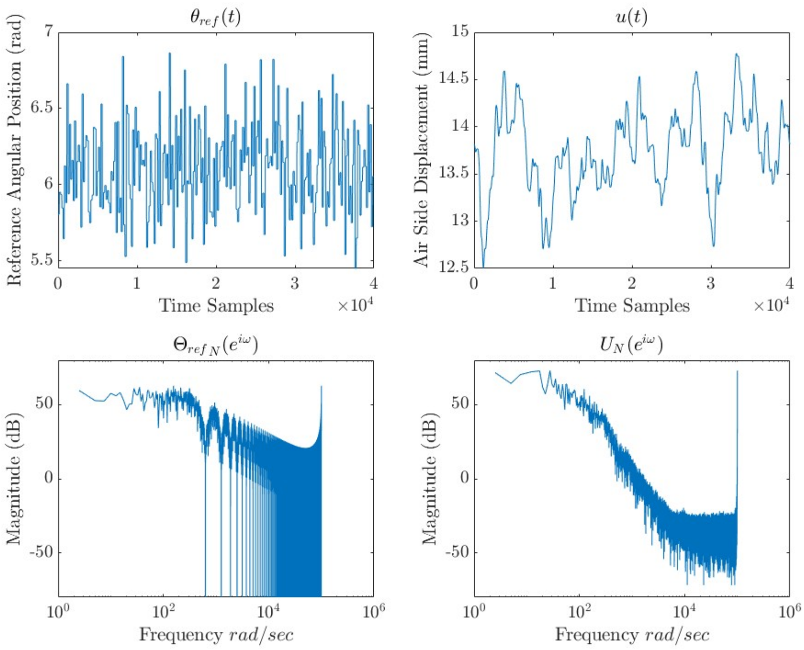


Figure 2.4: Application of Gaussian White Noise to identify TLMC

The simplest choice of an input signal would be a filtered Gaussian White Noise. Any signal spectrum can be achieved with the suitable design of filters. Figure 2.4, shows the application of a white noise signal as the reference to the control system. Clearly, the intended signal is not being translated by the control system from $\theta_{ref}(t)$ to $u(t)$. Moreover, significant changes to the frequency components of the signal can also be observed, especially after 100 rad/sec. Thus, we tend to lose the desirable properties of the white noise signal while using it to identify the TLMC. A similar result is also observed, on the application of Random Binary Signal (RBS) and Pseudo Random Binary Signal (PRBS) to the system.

Thus, we turn our attention towards sinusoidal signals. A chirp signal is a sinusoidal signal with frequencies that increase or change continuously over a certain range i.e. $\forall \omega \in [\omega_1, \omega_2]$ over a specific time range $\forall t \in [0, T]$. They can be expressed as,

$$u(t) = Acos(\omega_1 t + \frac{(\omega_2 - \omega_1)t^2}{2T})$$

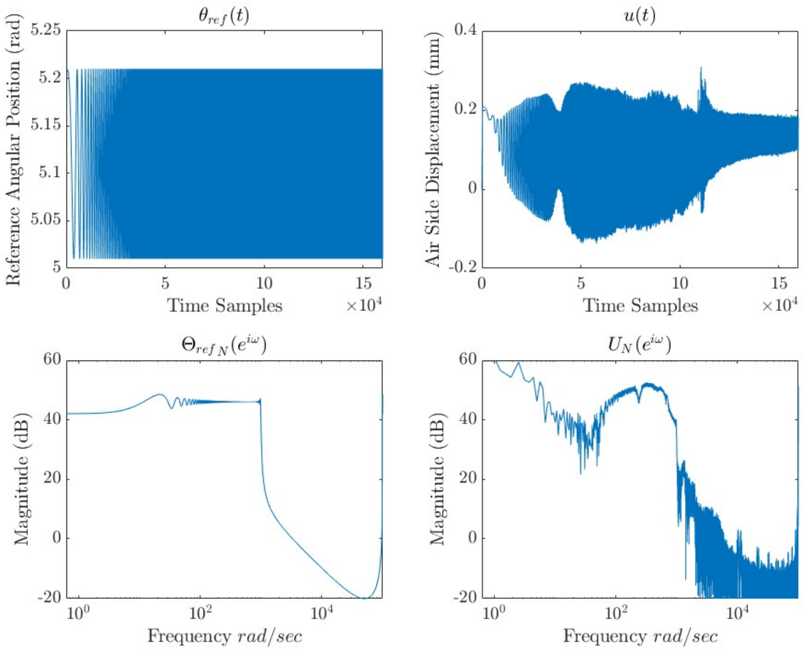


Figure 2.5: Application of chirp signal to identify TLMC

Figure 2.5, shows the application of a chirp signal with $\omega_1 = 1\ rad/sec, \omega_2 = 1000\ rad/sec, T = 10\ sec$ and $A = 0.1$ at the sampling frequency of 16 kHz as the reference to the control system. Again, we notice that the chirp signal does not completely translate from $\theta_{ref}(t)$ to $u(t)$. However, a closer look at the frequency spectra of the signals indicates a similarity in shape and equivalence in the magnitudes over the frequency range. Figure 2.6 shows the Power Spectral Density (PSD) of the chirp signal over the desired frequency range and a comparison of the PSD of the measured air side displacement signal.

Results and Model Approximation

We choose the chirp signal to perform the ETFE procedure outlined. The measured air side displacement $u(t)$ and the vacuum side displacement $y(t)$, on the application of the chirp signal is depicted in Figure 2.7.

The Bode Plot obtained after performing ETFE is shown in Figure 2.8. In the region of interest, between 1 rad/sec and 1000 rad/sec, the TLMC exhibits typical second order characteristics, with the presence of a corner frequency, where the magnitude peaks, and the phase shifts from $0°$ to $-180°$. It would not be unreasonable to assume that the TLMC is a second order system. We can estimate the parameters of the approximate second order system from the Bode Plot characteristics.

The transfer function model of a second order system is written as $\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$. in which ω_n represents the natural frequency of oscillation of the system and ζ is the damping coefficient of the system. These two parameters can be calculated from the size of the peak M_p and the frequency at which the peak occurs ω_r and the relationship is given as,

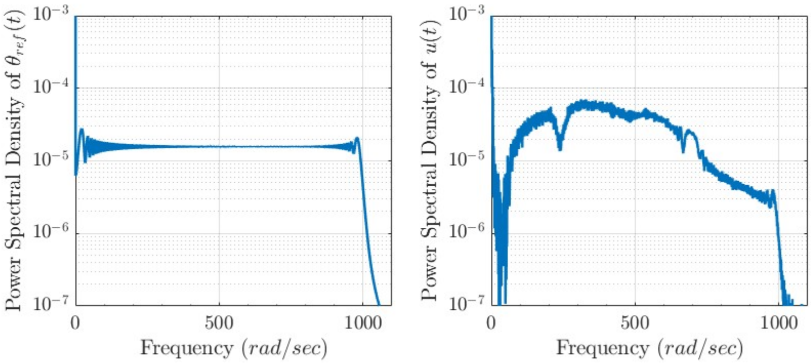


Figure 2.6: Power Spectral Density Comparison

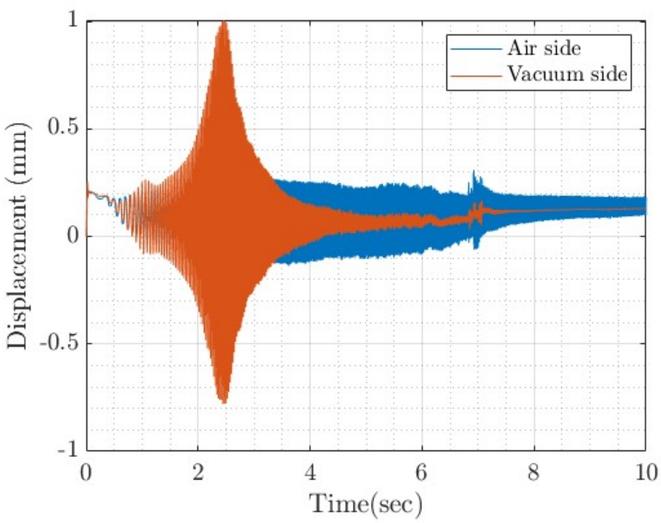


Figure 2.7: Air and Vacuum Side Displacements measured with Chirp Reference

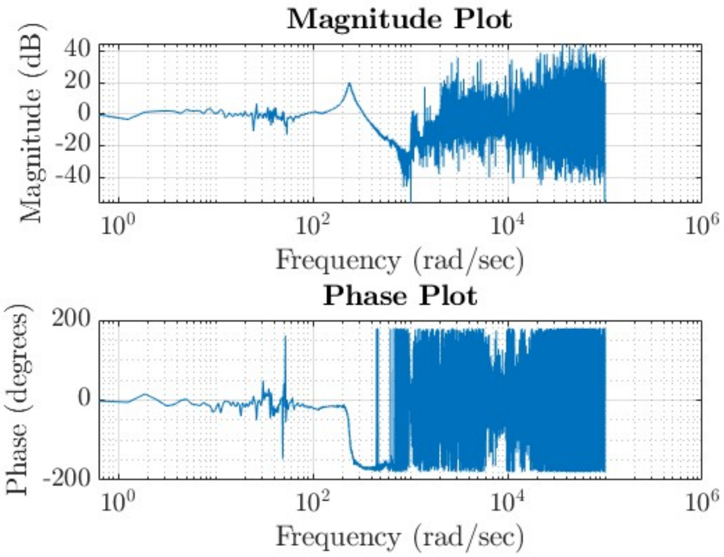


Figure 2.8: Bode Plot of TLMC

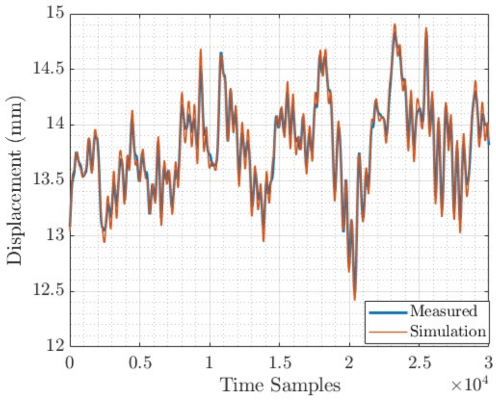
$$M_p = \frac{1}{2\zeta\sqrt{1-\zeta^2}}$$
$$\omega_n = \frac{\omega_r}{\sqrt{1-\zeta^2}}$$

A quick calculation of the parameters by plugging in the peak size $M_p = 18\ dB$ and $\omega_r = 236\ rad/sec$ results in the following second order approximation of the model,

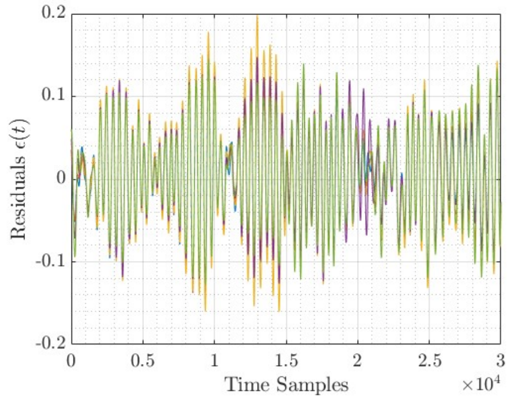
$$\hat{G}(s) = \frac{56412.6774}{s^2 + 29.88874s + 56412.6774} \tag{2.1}$$

Model Validation and Residual Analysis

Model validation involves establishing confidence in the estimated model's ability to capture essential aspects of system behavior, extending beyond the data used for its estimation. A commonly employed method for this purpose is Cross Validation. This entails creating a separate validation data set distinct from the one used for model estimation. The model is then simulated using the validation input, and the resulting output is compared with the measured validation output. Evaluation methods include visual inspection or the computation of a quantitative measure of fit. Additionally, a further validation test involves residual analysis, which examines the "leftover" portion in the model, denoted as $\epsilon(t) = y(t) - \hat{G}(s)u(t)$, representing the output unaccounted for by the model. It is crucial that these residuals are independent or at least uncorrelated with the information available during the model output formation. Residual analysis typically includes computing the auto-correlation function for the residuals and cross-correlation function between residuals and past inputs, with the expectation that these functions do not significantly deviate from zero. Cross-validation with the PMSM tracking Gaussian White Noise is shown in Figure 2.9 while 2.10 shows the results of a nominal scan cycle of the BWS.

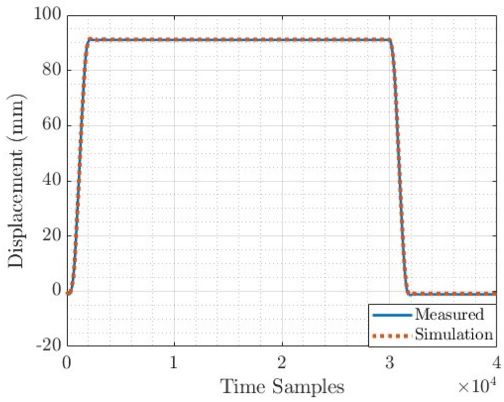


(a) Measured and Simulated Values

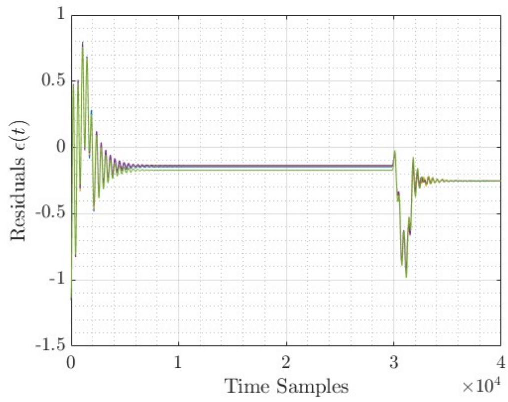


(b) Residuals over five sets of validation

Figure 2.9: ETFE Model Simulation Results - Gaussian White Noise



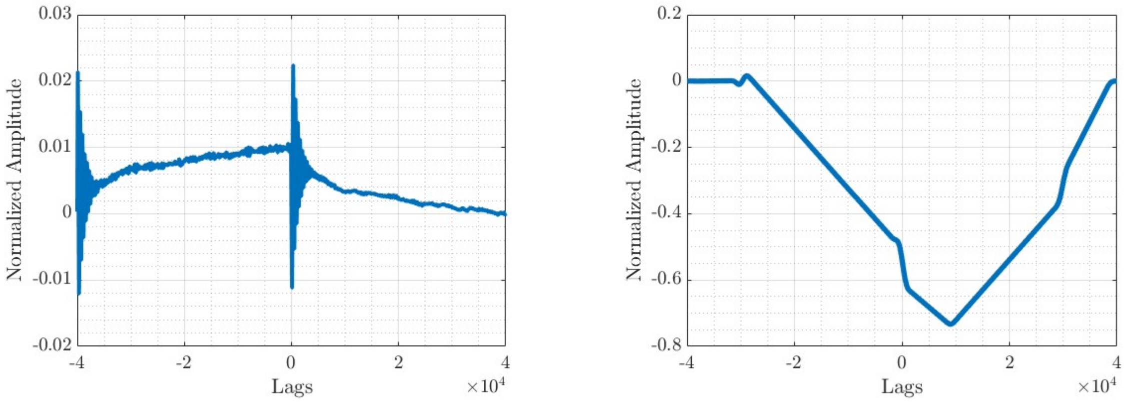
(a) Measured and Simulated Values



(b) Residuals over five sets of validation

Figure 2.10: ETFE Model Simulation Results - Nominal Scan Cycle

From Figures 2.9 and 2.10, it is evident that the second order linear model approximation calculated from ETFE captures most of the characteristics of the TLMC. However, the residuals seem to increase in magnitude with the increase in speed of the scan v_{scan}. Estimation of the static gain of the system seems reasonable. More information can be obtained by performing the cross-correlation between the input $u(t)$ and the residuals $\epsilon(t)$, shown in Figure 2.11. The cross-correlation is negligible between the residuals from Gaussian White Noise validation and the corresponding inputs. This indicates that the linear model almost completely captures everything the input has to offer in this case and it performs quite well. However, the cross-correlation in the nominal scan cycle case is quite high indicating that the linear model does not adequately perform well. This dual nature of the result, indicates discrepancies in considering the TLMC to be completely linear.



(a) Cross-correlation of residuals from Gaussian White Noise

(b) Cross-correlation of residuals from Nominal Scan Cycle

Figure 2.11: Cross-correlation Analysis

2.2.2 Auto Regressive Exogeneous

The Autoregressive with eXogenous input (ARX) model is a widely used mathematical framework in system identification and time-series analysis. In this model, the output of a system at a given time is expressed as a linear combination of its past values (auto regressive component) and the effects of external inputs (exogenous component). The ARX model is mathematically represented as:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + \ldots + a_{n_a} y(t-n_a) + b_1 u(t-1) + b_2 u(t-2) + \ldots + b_{n_b} u(t-n_b) + e(t)$$

Here, $y(t)$ denotes the system output at time t, $u(t)$ is the input to the system, a_1 to a_{n_a} are auto-regressive coefficients, b_1 to b_{n_b} are coefficients associated with the input terms, n_a is the auto-regressive order, n_b is the exogenous input order, and $e(t)$ represents the model's error or disturbance term.

ARX models are valuable for understanding and predicting the dynamic behavior of systems based on historical data and external influences. Our inferences from performing ETFE analysis indicates the presence of higher-order terms in the system or nonlinearities depending upon the velocity of scan. In order to understand this in better light and to explain the residuals better, we perform a study using ARX models of differing orders.

Estimation of ARX Parameters

Estimating the ARX parameters involves using observed input-output data to find the values of a_1 to a_{n_a} and b_1 to b_{n_b} that minimize the difference between the predicted output from the model and the actual output.
One common approach is to use the least squares method, where the objective is to minimize the sum of the squared differences between the predicted output and the actual output over all available data points. This leads to a set of linear equations that can be solved to obtain the parameter values.
The regressor matrix, often denoted as $\Phi(t)$, is a matrix that collects the input and output values for each time step and is used in the parameter estimation process. For the ARX model, the regressor matrix is constructed by stacking the lagged output and input values,

$$\Phi(t) = \begin{bmatrix} -y(t-1) & -y(t-2) & \dots & -y(t-n_a) & u(t-1) & u(t-2) & \dots & u(t-n_b) \end{bmatrix}^T \tag{2.2}$$

The parameter estimation problem is then formulated as,

$$\begin{aligned} &\Phi(t)\Theta = y(t) \\ &\text{where } \Theta = \begin{bmatrix} a_1 & a_2 & \dots & a_{n_a} & b_1 & b_2 & \dots & b_{n_b} \end{bmatrix}^T \end{aligned}$$

Data set and Validation Criteria

The input signal selection discussed in detail in subsection 2.2.1 also holds here. Hence, we use the signal shown in Figure 2.5 for ARX model identification as well. We use this data to identify different orders of ARX model, given by $\forall n_a \in \{3, 4, \dots, 20\}, \forall n_b \in [n_a]$ to ensure the causality of the identified model.
According to the input signal selection criteria given in Chapter 9.5 of [25], one of the major consideration points is the purpose of the modelling. Our purpose of the characterization and modelling of the TLMC is for simulation and prediction of the vacuum side displacement and not for designing a controller. Hence, it is objectively more important for the model to perform better in capturing the dynamics of the TLMC under nominal operating conditions rather than capturing the complete dynamics of the TLMC. Thus, we also consider a data set comprised of 170 experiments of nominal scan cycles of the BWS. An example of a nominal scan cycle is shown in 1.3. Thus the two data sets considered are,

1. D1: Chirp signal that has sinusoidal frequencies from 1 rad/sec to 1000 rad/sec
2. D2: 170 experiments of nominal scan cycles of BWS

With data set D1, we are interested in the Extrapolation behaviour of the model, i.e., how it behaves when the simulation input and output lie outside the region of the training data. On the contrary, with data set D2, we are more interested in how the model behaves between the training samples, i.e., its Interpolation behaviour. We also define two criteria for the validation of the ARX models of different orders based on Mean Squared Error (MSE) of the residuals $\epsilon(t)$, given as $MSE = \frac{1}{N}\sum_{t=1}^{N}\epsilon(t)^2$. They are,

1. C1: MSE obtained from simulating Gaussian White Noise averaged over 5 sets of validation data

2. C2: MSE obtained from simulating Nominal Scan Cycle averaged over 5 sets of validation data

Table 2.1 shows the two data sets and which behaviour of the model can be characterized from the two evaluation criteria obtained from these data sets.

	C1	C2
D1	Interpolation	Extrapolation
D2	Extrapolation	Interpolation

Table 2.1: Model Behaviour Characterization

Results and Discussion

Figure 2.12 shows the costs obtained by applying criteria C1 with data set D1 for the different models. The lowest cost obtained is for the model $(n_a, n_b) = (3, 3)$. While $n_a \leq 11$ and $n_b \geq 5$, the costs are higher while the costs remain almost similar for very high order models $n_a \geq 12$ and $n_b \geq 12$. Similar results are also observed from criteria C2 shown in Figure 2.13. Higher order models do not seem to explain the residuals well when compared to lower order models.

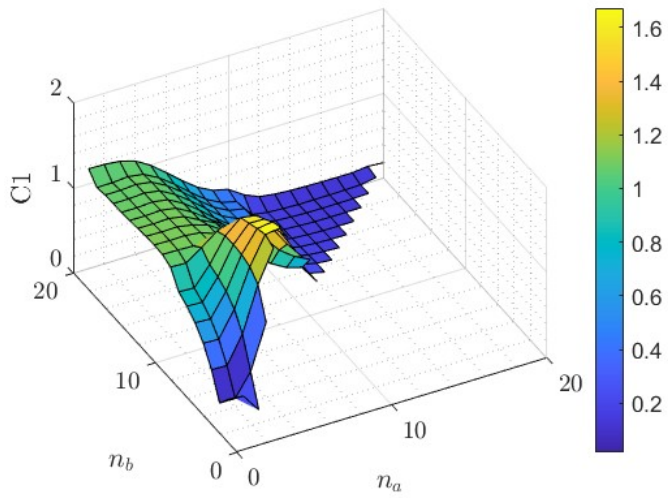


Figure 2.12: Variation of criteria C1 with n_a and n_b with data set D1

Figure 2.14 shows the residuals obtained after applying the ARX Model (3,3) to the validation data. We observe larger residuals when compared to the residuals obtained from the second order model approximated from ETFE. With data set D2, the ARX model (4,3) turned out to be the best one under both C1 and C2. Figure 2.15 shows the corresponding residuals. We see comparatively lower residuals, than the residuals obtained from D1. The model obtained from D2 exhibits significantly better interpolation and extrapolation behaviours when compared to the one obtained from D1. Other models like Auto-Regressive Moving Average Exogenous (ARMAX) and Output Error (OE) did not give better results either. We conclude this subsection by saying with the inference that higher order models cannot satisfactorily explain the TLMC. It is thus required to turn our attention towards nonlinear models.

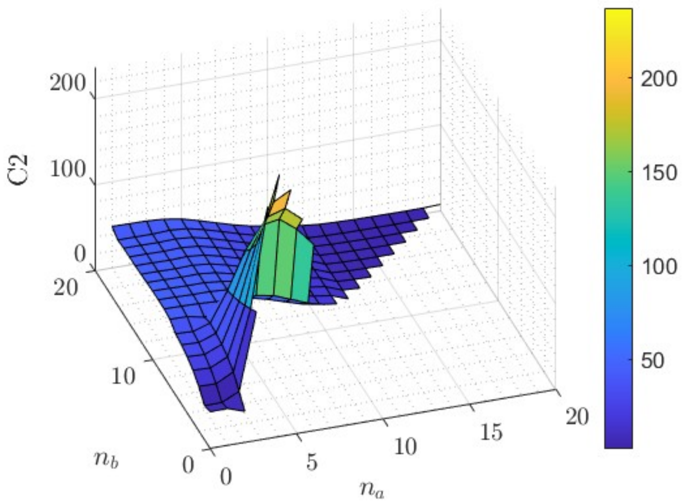
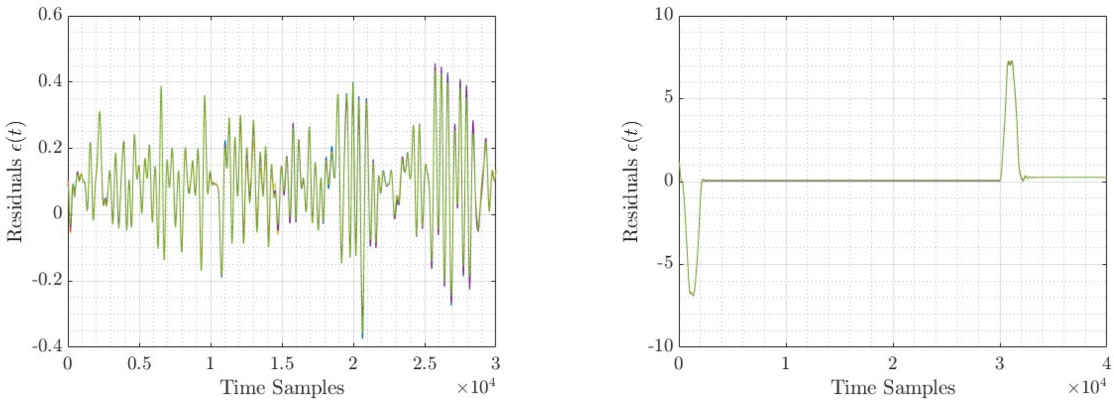


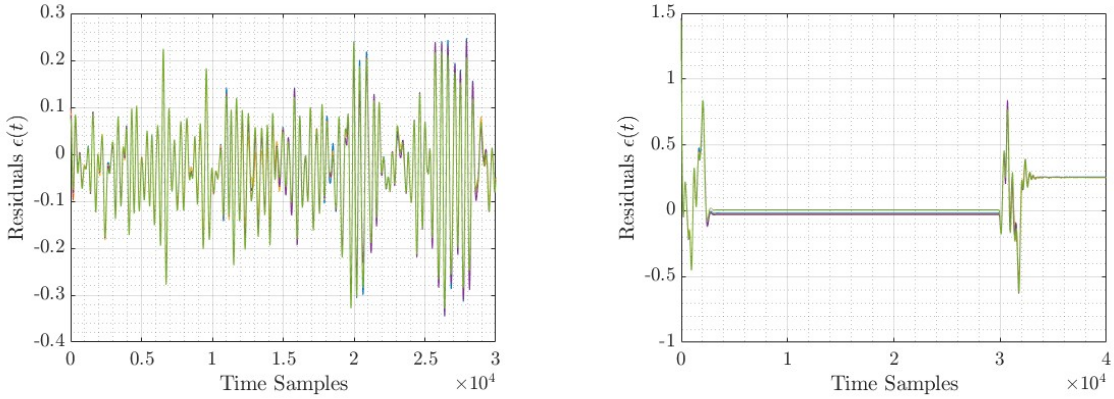
Figure 2.13: Variation of criteria C2 with n_a and n_b with data set D1



(a) Residuals from criteria C1

(b) Residuals from criteria C2

Figure 2.14: Residual Analysis of ARX (3,3) with dataset D1



(a) Residuals from criteria C1

(b) Residuals from criteria C2

Figure 2.15: Residual Analysis of ARX (4,3) with dataset D2

2.3 Nonlinear Methods

2.3.1 Nonlinear Auto Regressive Exogeneous

The Nonlinear AutoRegressive with eXogenous input (NARX) model is an extension of the traditional ARX model, introducing the capability to capture nonlinear relationships within a system. In the NARX model, the output at a given time is expressed as a nonlinear function of both its past values and external inputs. The mathematical representation can be expressed as:

$$y(t) = f(y(t-1), \dots, y(t-n_a), u(t-1), \dots, u(t-n_b)) + e(t)$$

Here, $y(t)$ is the system output, $u(t)$ is the input, n_a is the auto regressive order, n_b is the exogenous input order, f is a nonlinear function that characterizes the system's dynamics, and $e(t)$ represents the model's error or disturbance.
NARX models are particularly useful when dealing with systems exhibiting nonlinear behavior, making them applicable in various fields such as control systems, time-series prediction, and system identification. These models offer a more flexible and accurate representation of complex systems by allowing for the incorporation of nonlinear dependencies between variables. In practice, NARX models are employed to capture and analyze intricate relationships, enabling a more nuanced understanding. It is widely used for one-step ahead prediction of system responses, which makes it unsuitable for our application. However, we would like to understand if the nonlinear prediction method would explain the relationship between the displacements better than the previously described linear models.

Model Architecture, Data set and Training

We use Neural Networks composed of Multi Layer Perceptrons to learn the unknown function f. Based on the selected order of the model n_a and n_b, the number of neurons in the input layer varies. Specifically it is $n_a + n_b$ while the number of neurons in the output layer is always 1, corresponding to one-step ahead prediction. The training data set is composed of $D2$, in which each individual training data made up of each row of the equation (2.2). The output for the training is the corresponding $y(t)$. Very much similar to ARX, instead of using regression to find out linear coefficients, the neural network is trained with the data set to learn the unknown function f. We use 1 hidden layer of neurons arbitrarily with ReLU activation function. A simplified depiction of the neural network architecture used for NARX identification is shown in 2.16. Each of the arrows is associated with a weight, which is learned through back propagation in the training phase. For the one-step prediction, the corresponding u and y values are given to the network as inputs, which then outputs the prediction.

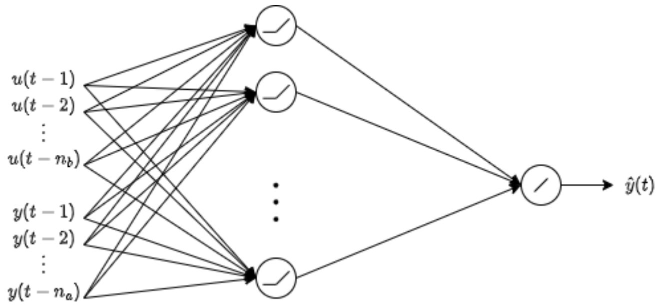


Figure 2.16: Simplified NARX Architecture

Results and Inferences

Figure 2.17 shows the average absolute prediction error in each of the one-step predictions for different orders of the model. Although the results cannot be directly translated for our application to simulate the complete system response of a scan cycle, we note that the order of prediction error obtained is much smaller than the residuals from the linear model. This motivates us to look into other nonlinear architectures that are more suitable towards our requirements.

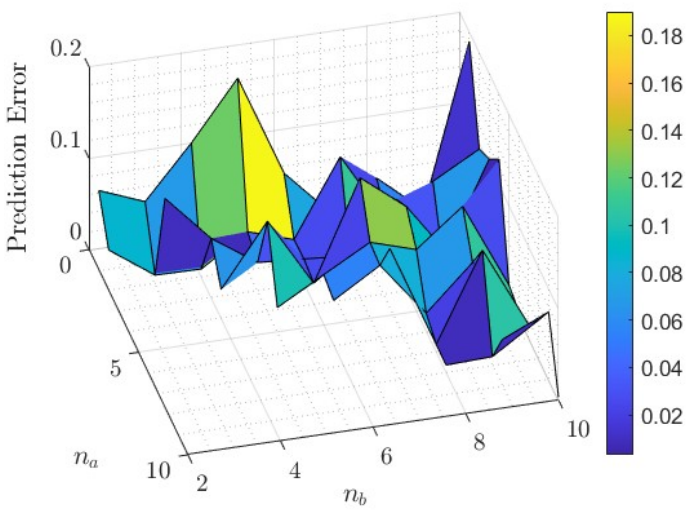


Figure 2.17: Average Absolute One-Step Prediction Error

2.3.2 Supplementary Nonlinear Model

The approach of using a nonlinear model in conjunction with a linear model to identify nonlinear systems is suggested in Chapter 19.4 of [25]. The approach outlined in this discussion offers distinct advantages. Notably, the linear model captures the dominant linear dynamics of the system, helping improve the extrapolation behaviour of the model, while the nonlinear model helps improve the interpolation behaviour. Two primary strategies for model identification, as

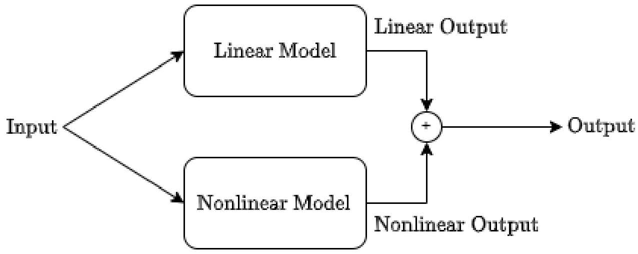


Figure 2.18: Supplementary Nonlinear Model

depicted in 2.18, can be distinguished:

1. Sequential Estimation: Initially estimate the parameters of the linear model. Subsequently, train the supplementary model using the error of the linear model while keeping the linear model fixed.
2. Simultaneous Training: Train both the supplementary and linear models concurrently.

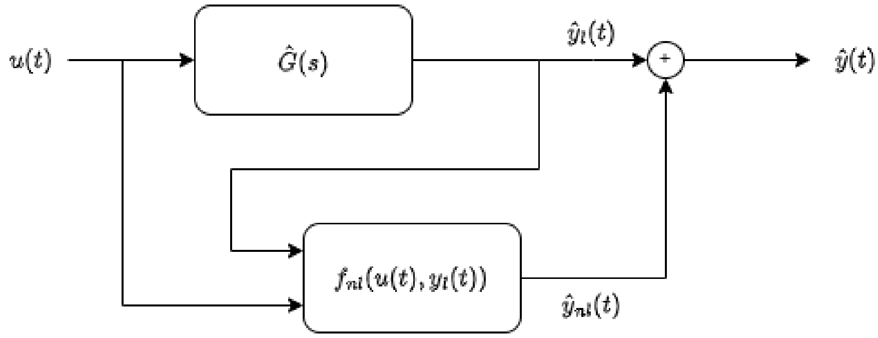
The second strategy is more flexible, as the number of effective parameters equals the sum of the parameters of both models, generally leading to higher model accuracy, especially on the training data. However, the linear model is identified only in conjunction with the nonlinear one, making it potentially less representative of the process on its own, and its extrapolation behavior may not be realistic. Instability of the linear model is also a concern.
On the other hand, the first strategy is less flexible, with a smaller number of effective parameters compared to both models combined. Unlike the second strategy, the linear model in this approach typically captures an average representation of the nonlinear process dynamics and is more likely to be stable, though not guaranteed. Consequently, the first approach is more reliable in producing realistic extrapolation behavior. Due to the availability of linear models from 2.2, and the simplicity of the approach we stick with the first strategy.

Model Architecture, Data set and Training

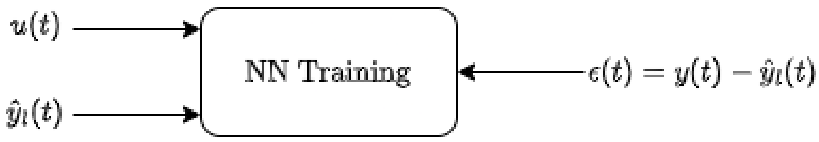
The architecture of the proposed model is depicted in Figure 2.19a. Analogous to the model explained above, we consider the Air side displacement, $u(t)$ to be the input to both the linear model $\hat{G}$ and the nonlinear model $f_{nl}(.,.)$. When passed through the estimated linear model, it generates $\hat{y}_l(t)$, which forms the other input to $f_{nl}(.,.)$. The nonlinear model takes in $u(t)$ and $\hat{y}_l(t)$ to predict $\hat{y}_{nl}(t)$. The final predicted Vacuum side displacement $\hat{y}(t)$ is given as,

$$\hat{y}(t) = \hat{G}(s)u(t) + f_{nl}(u(t), \hat{G}(s)u(t)) = \hat{y}_l(t) + \hat{y}_{nl}(t)$$

We have already obtained different candidates for $\hat{G}$: the approximated second order model from ETFE, the ARX(3,3) model obtained from D1 and the ARX(4,3) model obtained from D2. Due



(a) Architecture of the proposed model



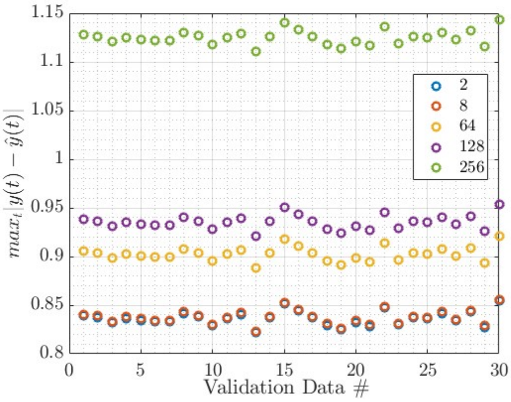
(b) Training of the Neural Network

Figure 2.19: Architecture and Training of the Nonlinear Model

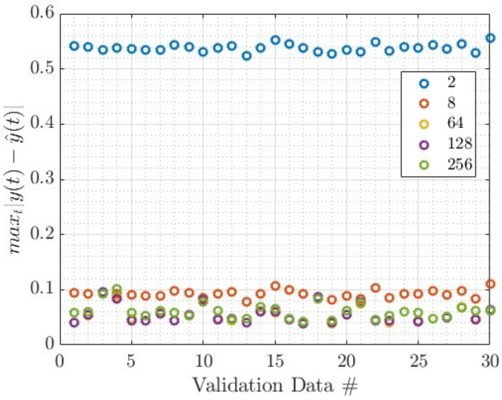
to the lower residuals obtained from the model from ETFE, we select this model as the fixed linear model $\hat{G}(s)$.
From our observations in Section 2.2, we concluded that the model obtained from D2 performed better than the model obtained from D1. Thus we select D2 comprising of 170 experiments of nominal BWS scans to be the data set of $u(t)$ to estimate $f_{nl}(.,.)$. Corresponding to the 170 experiments, we also calculate $\hat{y}_l(t)$ from the chosen $\hat{G}(s)$ to finish up the input data set needed for the estimation. The output data set is made of the residuals $\epsilon(t) = y(t) - \hat{y}_l(t)$, the difference between the measured value of the vacuum side displacement and the output of the estimated linear model. The residuals for each corresponding experiment can be calculated and it forms the output data set for the estimation. We use an MLP NN similar to the one described in subsection 2.3.1 to learn and identify $f_{nl}(.,.)$. The training architecture is given by Figure 2.19b.

Results and Discussion

The expectation from the supplementary nonlinear model is that, it would be able to decrease the residuals obtained only after the linear model and that it would reduce the cross-correlation seen between the inputs and the residuals. Using the data set D2 for the learning of the function $f_{nl}(.,.)$ we require the interpolation performance of the model to be good, since it would be used only for the prediction of vacuum side displacements in other nominal scan cycles.
We consider an input layer of dimension $2fT$, where f is the sampling frequency and T is the total lengths of the scan cycle. The output layer is of dimension fT. Initially we start off with a single hidden layer. Training is performed using the data set D2, and data is prepared for training as detailed earlier. Of the 170 sets of data, we have a training and testing split of 70% and 30% respectively. Figure 2.20a shows the value of the maximum absolute prediction error, $|y(t) - \hat{y}(t)|$ over 30 sets of validation data, for ReLU activation function, while 2.20b shows the same for sigmoid activation function for different number of neurons in the hidden layer. For the network using ReLU activation function, there aren't any significant improvements in the



(a) Variation of maximum absolute prediction error with number of neurons for ReLU activation function



(b) Variation of maximum absolute prediction error with number of neurons for Sigmoid activation function

Figure 2.20: Comparison of activation functions and size of hidden layer

maximum absolute prediction error as the number of neurons increases. In fact, the maximum absolute prediction error increases with the increase in number of neurons which seems counter-intuitive. However, the network using sigmoid activation function shows significant reduction in the maximum absolute prediction error for initial increase in number of neurons. As the number of neurons becomes large enough, the value does not change significantly. Comparing the two activation functions, it is evident that the sigmoid activation function performs better than ReLU in learning the nonlinearities.

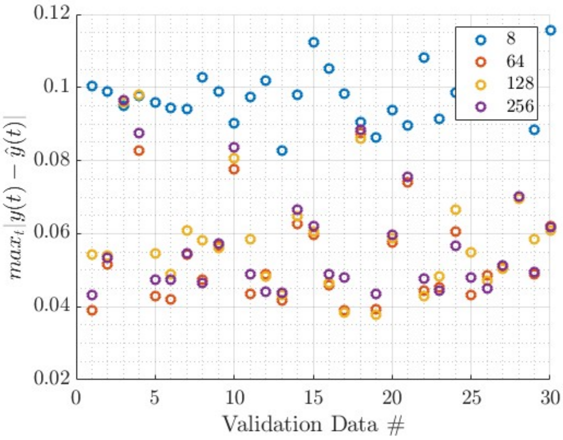
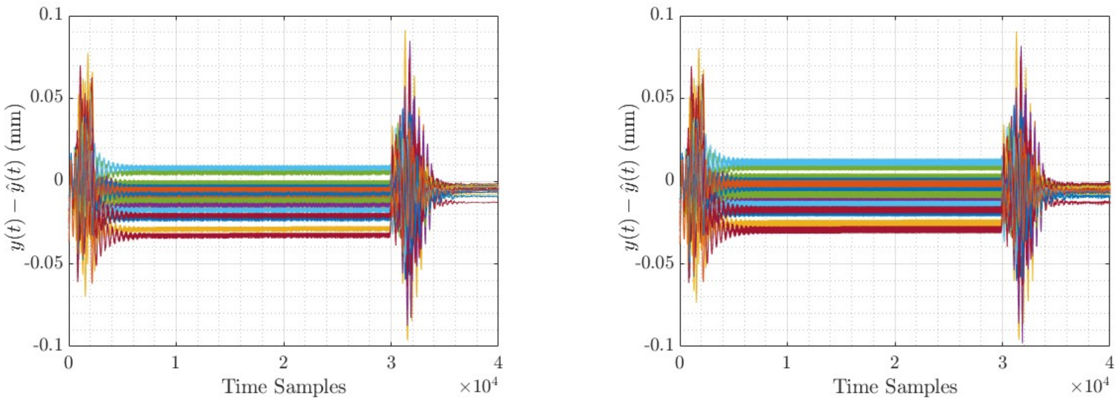


Figure 2.21: Variation of maximum absolute prediction error with number of neurons for two hidden layers

We then train the neural network with two hidden layers, the first one made of ReLU activation functions and the second hidden layer made if sigmoid activation functions and Figure 2.21 shows the variation of the maximum absolute prediction error with number of neurons for this architecture. A slight improvement from the single hidden layer architecture with sigmoid functions is noticed. We see that the optimum value of neurons would either be 64 or 128 from the data, while over fitting is possible for the network containing 256 neurons causing degradation of performance.



(a) Prediction error in mm over 30 sets of validation data for 64 neuron hidden layer

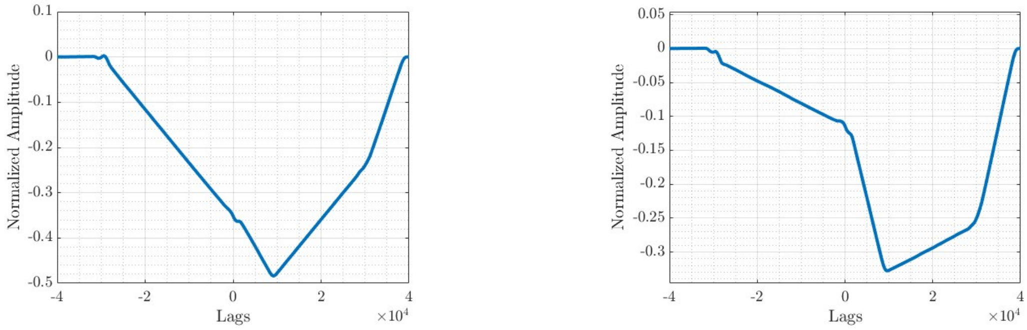
(b) Prediction error in mm over 30 sets of validation data for 128 neuron hidden layer

Figure 2.22: Comparison of Prediction Error of NN Architectures

The actual prediction error over 30 sets of validation data for two hidden layers with 64 neurons each and 128 neurons each is shown in Figure 2.22. Clearly, the nonlinear model does a better job than the linear model in capturing the residuals and the final prediction error has been brought down by almost 10 times when compared to the residuals observed from the second order approximation model from ETFE.
Plotting the cross-correlation between the prediction error and the inputs, the 128 neuron network seems to have lower correlation than the 64 neuron network. Although not close to 0, which suggests there are still some characteristics of the input to be captured by the model, the amplitude is less than half the amplitude as observed by just using the linear model from ETFE. The accuracy of the nonlinear model is significantly higher than the accuracy of the linear model, with the average prediction error lying in the order of 20 microns. Although this is not very accurate when compared to the smallest beam size of LHC which is 200 microns, as discussed in Section 2.1, the prediction error in general tends to affect the position of the beam, while the spread of the prediction error affects the beam size. Thus, in order to study how the model would behave in the estimation of beam sizes, we need to perform further simulations and is explained in the forthcoming subsection.

2.4 Beam Size Estimation and Simulations

The determination of the vacuum side displacement, which is the wire position in reality, is ultimately important in the calculation of the beam size. In this subsection, we analyze the relationship between the results that we have obtained in the prediction of vacuum side displacement using the nonlinear model and beam size estimation.



(a) Cross-correlation of prediction error and input for 64 neuron hidden layer

(b) Cross-correlation of prediction error and input for 128 neuron hidden layer

Figure 2.23: Comparison of Cross-correlation of NN Architectures

2.4.1 Steps and Implementation

The aim of the simulation is to study how the actual beam size estimated from true vacuum side displacement varies when compared to the beam size estimated from the predicted vacuum side displacement. In essence, we simulated the amplitude values of the PMT, against the measured and predicted displacement values. Under the assumption that the beam centre can lie anywhere between 30 mm and 65 mm, we try and to fit a Gaussian curve, representing the values measured by the PMT to each of these points. The steps in the simulation can be enumerated as,

1. Generate a Gaussian Functional $f_{gauss}(t) = \frac{1}{\sigma_t\sqrt{2\pi}}\exp(\frac{-(t-\mu_t)^2}{2\sigma_t^2})$ with σ_t corresponding to the desired beam size σ_{beam} and μ_t as the time instant of the beam centre.

2. Identify $y(\mu_t)$, the measured value of vacuum side displacement at time μ_t and fit $f_{gauss}(t)$ with $y(\mu_t)$ as the mean. The actual beam size, σ_{beam} which is the standard deviation of the Gaussian Fit should be the same as the desired value.

3. Identify $\hat{y}(\mu_t)$, the predicted value of vacuum side displacement at time μ_t and fit $f_{gauss}(t)$ with $\hat{y}(\mu_t)$ as the mean. The estimated beam size, $\hat{\sigma}_{beam}$ is the standard deviation of the Gaussian Fit.

4. Repeat steps 1-3 for different values of μ_t, all possible beam centres.

5. Repeat step 4 for different sets of validation data.

   The simulation can also be performed to study how the beam size is estimated just from the air side displacement values, by performing the same steps given above, but by replacing $\hat{y}(\mu_t)$ with $u(\mu_t)$, the measured value of air side displacement for comparison purposes.

2.4.2 Results and Inferences

Figure 2.24 shows the histogram of beam sizes obtained by performing the simulation for three different beam sizes, $\sigma_{beam} = 2$ mm, 1 mm, 0.2 mm as described in the previous subsection. As described in Section 2.1, we can calculate the systematic and the statistic errors in beam size calculation. The results are tabulated in Table 2.2.

In order to achieve consistent beam size estimations with an accuracy of 1%, and under the Gaussian approximation of the beam size histogram, we would require $6\epsilon_{systematic} \leq 0.01$, which

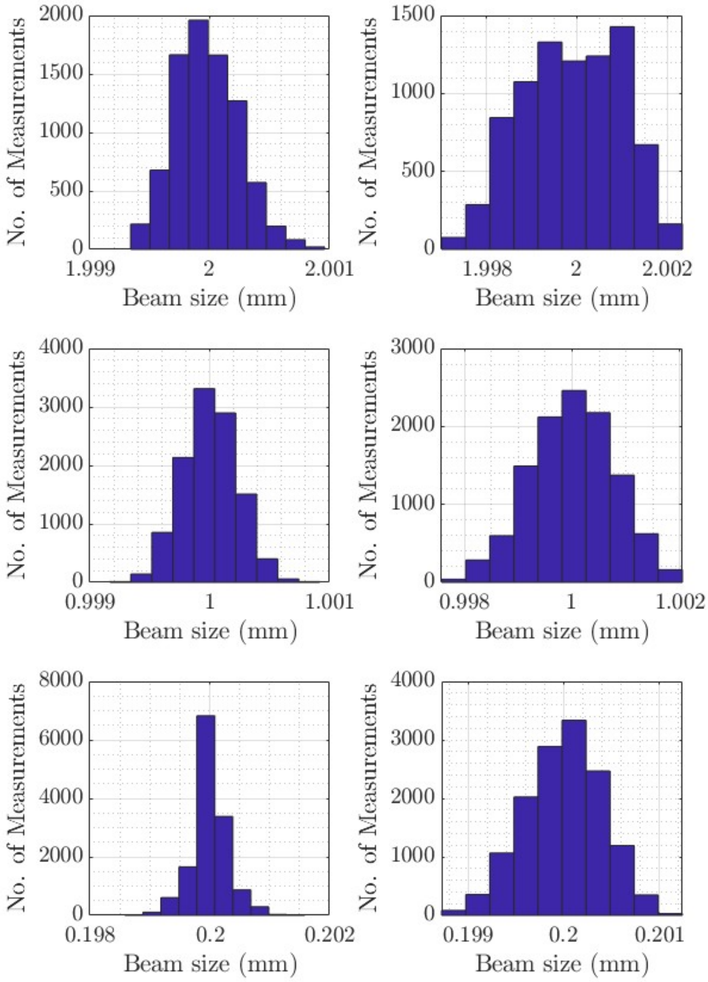


Figure 2.24: Histogram of beam sizes calculated from predictions (left) measured air side displacement (right) for 2mm, 1mm and 0.2 mm beam sizes (corresponding rows)

Beam Size	$\epsilon_{\mathbf{systematic}}$ **from** $\hat{\mathbf{y}}(\mu_\mathbf{t})$	$\epsilon_{\mathbf{statistic}}$ **from** $\hat{\mathbf{y}}(\mu_\mathbf{t})$	$\epsilon_{\mathbf{systematic}}$ **from** $\mathbf{u}(\mu_\mathbf{t})$	$\epsilon_{\mathbf{statistic}}$ **from** $\mathbf{u}(\mu_\mathbf{t})$
2 mm	1.15×10^{-5}	1.32×10^{-4}	5.7×10^{-5}	5.5×10^{-4}
1 mm	2.6×10^{-6}	2.28×10^{-4}	3.3×10^{-6}	7.74×10^{-4}
0.2 mm	0.2×10^{-5}	1.49×10^{-3}	1.5×10^{-5}	2.02×10^{-3}

Table 2.2: Systematic and Statistic Error Comparison

translates to $\epsilon_{systematic} \leq 1.67 \times 10^{-3}$, which all the histograms calculated from the predictions $\hat{y}(\mu_t)$ achieve. The accuracy of the beam size estimation progressively worsens as the size of the beam decreases, which is intuitive. The simulation results show that the predicted vacuum side displacements from the model in subsection 2.3.2 can be used in the estimation of beam sizes to give the desired accuracy. Moreover, for higher beam sizes, just the measurements of the air side displacement from the encoder would suffice to satisfy the accuracy requirements. However, this is under the assumption that the position measurement device (encoders) and the PMTs work perfectly without any noise and there is no wire vibration. Naturally, the size of the beam estimated from the predictions performs better than just estimating it from the air side displacement values, indicating that it would be more robust to the influence of errors in other components of the BWS.

Thus, we conclude this chapter having explored the importance of wire position accuracy, different data-driven system identification techniques for the TLMC, and the effect of the predicted values on beam size estimations.

Chapter 3

Optimal Reference Trajectory Generation

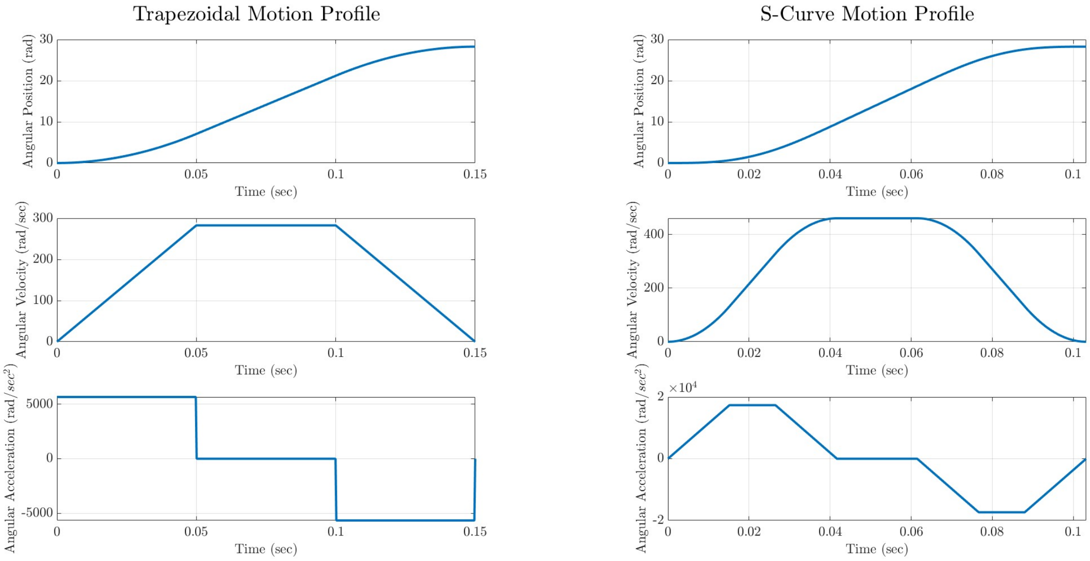
One of the major sources of uncertainties in the wire position has been attributed to vibrations in the wire attached to the fork. Several previous works have been carried out in order to characterize these vibrations, including developing a dynamic model for the wire [1]. A complete study on how wire vibrations cause inaccuracies in position measurement and a design that mitigates the effect of vibrations can be found in [15]. In this chapter, we provide a different approach to minimize the wire vibrations: by optimizing the reference trajectory to reduce vibrations in the system. In addition we also study how the velocity of the wire motion can be maximized optimally.

3.1 Time Optimal Trajectory

Pre-shaping reference trajectories to suppress vibrations, as initially proposed by Singer in 1990 [29], has found application in various systems. Overcoming the challenge of attaining rapid motions with minimal or no vibrations is crucial across different industries. While a simple trapezoidal velocity profile can achieve swift motions by accelerating the system to a constant velocity and then decelerating it to reach the desired position, it often leads to undesired oscillations due to abrupt velocity changes at the endpoints.
To address this issue, the use of S-curve velocity profiles has become prevalent. Figure 3.1 illustrates the difference between a Trapezoidal and an S-Curve Motion Profile. In the context of Beam Wire Scanners, it is imperative for the reference trajectory to meet specific objectives:

- **Final Position:** The system must reach a final position corresponding to a predefined stroke length.
- **High Speed:** While in motion, the system should attain at least a predefined minimum velocity to ensure efficiency and preferably operate at high speeds.
- **Avoid Excitation of Natural Frequency:** The identified natural frequency of the system should not be excited during its motion. This is crucial for maintaining stability and precision in position.

By integrating these goals into the design of the reference trajectory, Beam Wire Scanners can optimize their performance, achieving rapid and precise motions without inducing harmful vibrations.



(a) Trapezoidal Motion Profile (b) S-Curve Motion Profile

Figure 3.1: Motion Profiles

3.1.1 Theory and Algorithm

Among the several methods proposed for time optimal reference trajectory generation, we explore and implement the methods presented in [3] owing to the simplicity, modularity and the ease of implementation of the algorithm proposed. A brief overview of the method in [3] is given in this subsection.

S-Curve trajectories of order n are composed of polynomial segments joined to assure the continuity of derivative of order up to $n-1$. A cascade of n filters called "rectangular smoothers", each with the transfer function,

$$M_i(s) = \frac{1}{T_i}\frac{1-e^{-sT_i}}{s}$$

in which the free parameter T_i denotes the time length of the impulse response of each of the filters. The cascade arrangement of the rectangular smoothers to generate the trajectory is shown in Figure 3.2. The input to the trajectory generator is defined as $u(t) = q_{in} + q_{fin}h(t)$; with q_{in} denoting the initial position and q_{fin} denoting the final position.

We would also like to add constraints to each of the n orders i.e., position, velocity, jerk, acceleration etc. If the constraints are also symmetric, they can be formulated as,

$$-\hat{q}^{(i)}_{max} \leq q^{(i)}(t) \leq \hat{q}^{(i)}_{max}$$

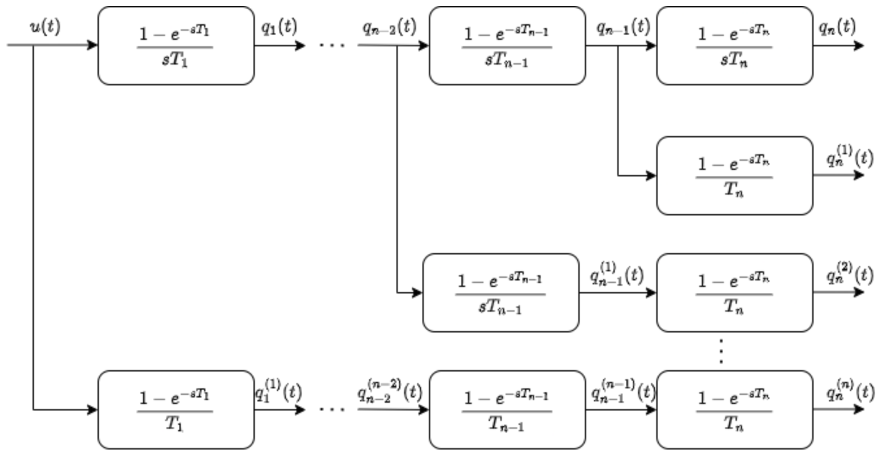


Figure 3.2: Cascade of Filters for Time Optimal Trajectory Generation

where $q^{(i)}$ denotes the i^{th} derivative of position. Finally, it is shown that the vector of optimal time parameters $\mathbf{T} = [T_1 T_2 \dots T_n] \in \mathbb{R}^n$ can be obtained by solving the following non-linear optimization problem,

$$
\begin{aligned}
& \mathbf{T}^\star = \arg\min_{\mathbf{T}} \sum_{k=1}^{n} T_k \\
\text{s.t.} \quad & \hat{q}^{(n)}_{\max} \prod_{k=1}^{n} T_k = \hat{q}^{(0)}_{\max} \\
& \hat{q}^{(n)}_{\max} \prod_{k=i+1}^{n} T_k = \hat{q}^{(i)}_{\max}, \quad \forall i \in [n-1] \\
& \mathbf{A}\mathbf{T} \leq \mathbf{0}_n
\end{aligned}
$$

where $\mathbf{A} = \begin{bmatrix} -1 & 1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & 1 & \dots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \dots & -1 \end{bmatrix}$. The vector of values obtained $\mathbf{T}^\star$ by solving the above optimization problem, gives the minimum duration to achieve the final position specified by q_{fin}, subjected to the constraints on different characteristics of motion. The other goal of the trajectory generation is to suppress the excitation of the identified natural frequency of the system. The optimization algorithm above can in principle be modified, to include the vibration suppression by adding a filter each for each of the natural frequency to be suppressed. However this finally leads to a 'sub-optimal' result with the addition of more rectangular smoothers. Thus, a different approach has been proposed and detailed in Algorithm 1.

Algorithm 1 Frequency Suppression for Time Optimal Trajectory Generation

Input Optimal Time Values $\mathbf{T}^{\star}$ and Frequencies to be suppressed $\mathbf{T}_{\omega}$
Output Optimal Time Values after frequencies are suppressed $\mathbf{T}^{\mathbf{opt}}$

for $j \leftarrow 1$ to $\min(n, m_{tot})$ **do**
    $k = \text{ceil}(\mathbf{T}^{\star}/T_{\omega_j})$
    Find l such that $k_l \in k$ minimizes $(k_l T_{\omega_j} - T_l^{\star})$
    Add $k_l T_{\omega_j}$ to $\mathbf{T}^{\mathbf{opt}}$
    Remove T_l from $\mathbf{T}^{\star}$
end for
Add the remaining elements of $\mathbf{T}^{\star}$ or $\mathbf{T}_{\omega}$ to $\mathbf{T}^{\mathbf{opt}}$

For any resonant frequency ω_i, the corresponding fundamental time period $T_{\omega_i} = \frac{2\pi}{\omega_i}$ is calculated and a vector $\mathbf{T}_{\omega} \in \mathbb{R}^{m_{tot}}$ is constructed from the m_{tot} resonant frequencies to be suppressed. The algorithm takes in as input, the vector of optimal parameters $\mathbf{T}^{\star}$ and $\mathbf{T}_{\omega}$ as input. It approximates an element of $\mathbf{T}^{\star}$, the one that minimizes the residual time $(k_l T_{\omega_j} - T_l^{\star})$ with T_{ω_j}.

3.1.2 Implementation, Results and Discussion

The outlined method and Algorithm 1 are implemented on MATLAB for the same final position, but for different constraints on the other derivatives of position. The generated trajectories are then applied to the test bench and the results are discussed in this subsection.
The specification for the final position to be reached is 28.274339 rad, which translates to a stroke length of 90 mm after the translation of motion by the bolt screw. As identified in subsection 2.2.1, we would like to suppress the natural frequency of the magnetic coupling which is around 236.9446 rad/sec. For the application on BWS, we do not have any system driven constraints on the acceleration and jerk of the trajectory. However, we would like the system to move with a scan velocity of at least 1 ms^{-1}. We do not specify this constraint explicitly on the trajectory generation algorithm. We generate different trajectories with varying level of constraints on the jerk, to study how the optimal velocity varies. Figure 3.3 shows the frequency response of the cascade of the rectangular smoothers used to generate the trajectory for position.

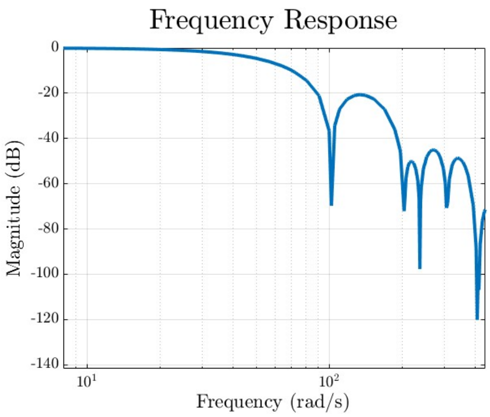


Figure 3.3: Frequency Response of the Cascade of Filters

From the figure, we can clearly see that the natural frequency a of the magnetic coupling at 236.9446 rad/sec is suppressed, since it has a magnitude of less than -90dB. The motion profile of a trajectory generated by the proposed method, with $n = 4, \hat{q}^{(1)}_{max} = 28.2743$ rad, $\hat{q}^{(2)}_{max} = 430$ rad/sec, $\hat{q}^{(3)}_{max} = 6 \times 10^4$ rad$/sec^2$, $\hat{q}^{(4)}_{max} = 2.5 \times 10^5$ rad$/sec^3$ and $T_\omega = \frac{2\pi}{236.9446}$ is shown in Figure 3.4. The total optimal time obtained by the method is 0.1563 sec. Figure 3.5 compares the DFTs of the displacements measured applying existing trajectory and time optimal trajectory to the experimental setup. A difference in magnitude of only about 10dB is noticed at the identified resonant frequency.

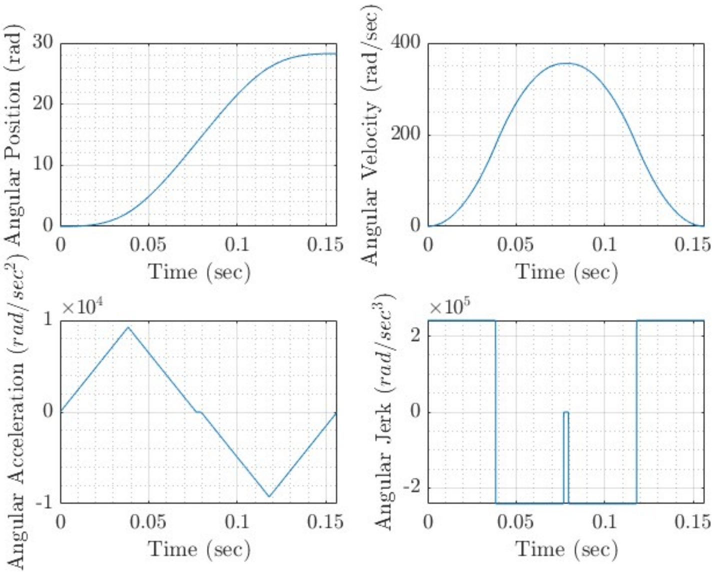


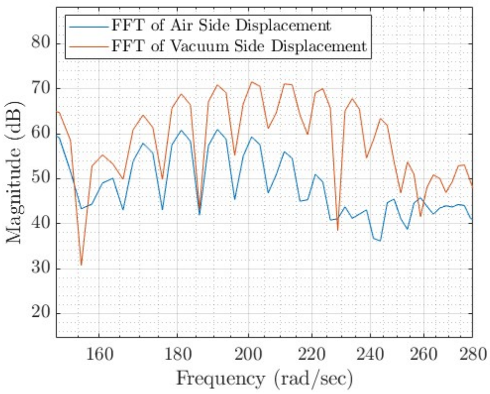
Figure 3.4: Motion Profile of a time optimal trajectory

It is interesting to see how this translates to the time domain, the actual displacement and position measured in the vacuum side. Figure 3.6 shows the IN stroke, when the wire would come into contact with the beam of the existing and the proposed trajectories for 10 different scan cycles. It is clear to see the vibration-free motion of the instrument under the proposed trajectory. Similar results are also observed in the OUT stroke of the instrument

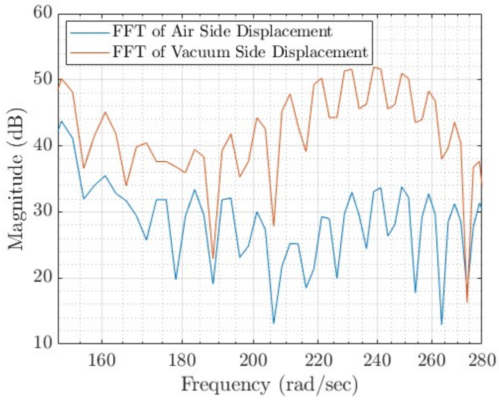
Moreover, we analyse the repeatability of the trajectories and see how the values are spread over different scan cycles. Although this is also a property of the controller in use, we use the same controller for fair comparison between the two trajectories. Over the same 10 trajectories shown above, the range of time optimal trajectory is almost 10 times lesser than the existing trajectory, indicating that it produces highly repeatable motion. This property, over a complete scan cycle, is depicted in Figure 3.7.

Having proven experimentally that the time optimal trajectory produces less vibrating and more repeatable motion over scan cycles, the question of selecting a suitable trajectory that reaches high velocities while still maintaining lower jerk remains. To this extent we generate different trajectories and plot the relationship between the highest velocity it reaches during the stroke and the maximum jerk it generates. A suitable trade-off needs to be found to balance the requirements.

Figure 3.8 shows the variation of the peak velocity with peak jerk and the optimum total time with peak jerk for different trajectories. The minimum linear scan velocity of 1 ms^{-1}, equates to 314.1593 rad/sec and all the trajectories that reach a peak velocity above this value would satisfy our requirements. This plot enables in the selection of a suitable trajectory that reaches

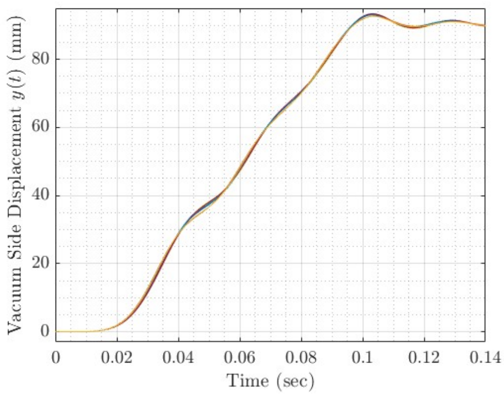


(a) DFT of displacements measured using existing trajectories

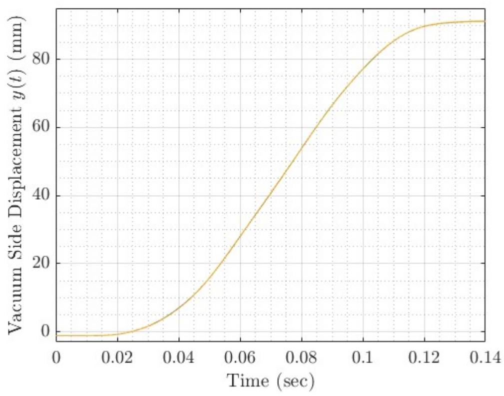


(b) DFT of displacements measured using time optimal trajectory

Figure 3.5: DFT comparison

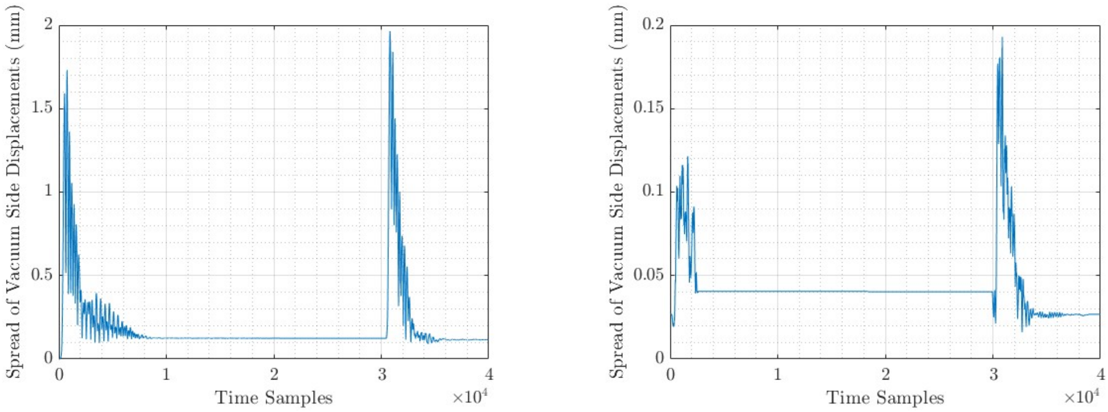


(a) Motion of existing trajectory



(b) Motion of time optimal trajectory

Figure 3.6: IN Stroke motion comparison of trajectories over 10 scan cycles



(a) Spread of existing trajectory
(b) Spread of time optimal trajectory

Figure 3.7: Spread comparison of trajectories over 10 scan cycles

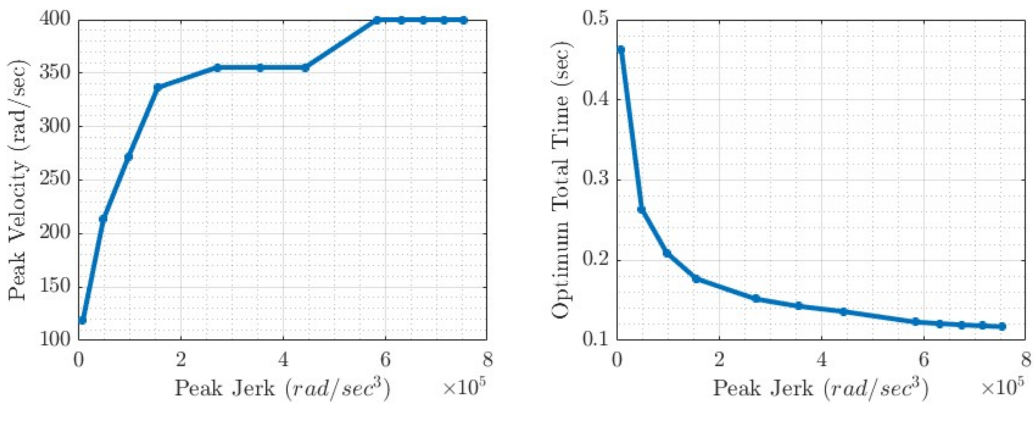


Figure 3.8: Comparison of Optimal Trajectories

a specific speed but also minimizes the peak jerk and the time taken.

3.2 Minimum Squared Jerk Trajectory

Jerk, the time derivative of acceleration is associated with rapidly changing actuator forces. A high jerk, not only has a detrimental effect on the actuators but is also associated with inducing resonant vibrations on the system and is detailed extensively in [2]. Extensive research has been done to generate trajectories that minimize the accumulated jerk over time, the maximum jerk at a point and so on [20], [19] and [13]. We investigate one such approach in this section and analyse how it performs on the BWS.

3.2.1 Theory and Methodology

We would like to minimize the total energy of the jerk, i.e.,

$$\min \quad \frac{1}{2}\int_0^{t_{fin}} (q^{(3)})^2 \, dx$$

while adhering to the constraints on the final position reached and the system stops moving. This can be formally formulated as,

$$\begin{aligned}
&q^{(0)}(0) = 0 \quad q^{(0)}(t_{fin}) = q^{(0)}_{fin} \\
&q^{(1)}(0) = 0 \quad q^{(1)}(t_{fin}) = 0 \\
&q^{(2)}(0) = 0 \quad q^{(2)}(t_{fin}) = 0
\end{aligned} \tag{3.1}$$

Several techniques and methods have been adopted to solve this trajectory generation problem including calculus of variations and various optimization methods. We cast this problem in the framework of Pontryagin [27] and apply the minimum principle to arrive at a solution for this problem.
We start by defining a simple system with three states, $q^{(0)}, q^{(1)}$and $q^{(2)}$. The jerk $q^{(3)}$ can be considered as an input and the state equations are written as,

$$\begin{aligned}
\dot{q^{(0)}}(t) &= q^{(1)}(t) \\
\dot{q^{(1)}}(t) &= q^{(2)}(t) \\
\dot{q^{(2)}}(t) &= q^{(3)}(t)
\end{aligned}$$

We define the co-state variables as $\Lambda^\top = [\lambda_1(t) \quad \lambda_2(t) \quad \lambda_3(t)]$. The final time t_{fin} is taken to be a given value for this problem. Trajectories can be generated for different values of t_{fin} such that they satisfy the minimum velocity constraint. The Hamiltonian of the problem can be written as,

$$H(q^{(1)}(t), q^{(2)}(t), q^{(3)}(t), \Lambda^\top) = \lambda_1(t)q^{(1)}(t) + \lambda_2(t)q^{(2)}(t) + \lambda_3(t)q^{(3)}(t) + \frac{1}{2}(q^{(3)}(t))^2$$

By the minimum principle, the following differential equations are the necessary conditions for a minimum to exist:

$$
\begin{aligned}
& \frac{\partial H}{\partial q^{(3)}(t)} = 0 && \implies q^{(3)} = -\lambda_3(t) \\
\dot{\lambda_1}(t) &= -\frac{\partial H}{\partial q^{(0)}(t)} = 0 && \implies \lambda_1 = c_1 \\
\dot{\lambda_2}(t) &= -\frac{\partial H}{\partial q^{(1)}(t)} = -\lambda_1(t) && \implies \lambda_2(t) = -c_1 t + c_2 \\
\dot{\lambda_3}(t) &= -\frac{\partial H}{\partial q^{(2)}(t)} = -\lambda_2(t) && \implies \lambda_3(t) = -c_1\frac{t^2}{2} - c_2 t - c_3
\end{aligned}
$$

With the above conditions, we can calculate the positon and its derivatives directly.

$$
\begin{aligned}
q^{(3)}(t) &= c_1\frac{t^2}{2} + c_2 t + c_3 \\
q^{(2)}(t) &= c_1\frac{t^3}{6} + c_2\frac{t^2}{2} + c_3 t + c_4 \\
q^{(1)}(t) &= c_1\frac{t^4}{24} + c_2\frac{t^3}{6} + c_3\frac{t^2}{2} + c_4 t + c_5 \\
q^{(0)}(t) &= c_1\frac{t^5}{120} + c_2\frac{t^4}{24} + c_3\frac{t^3}{6} + c_4\frac{t^2}{2} + c_5 t + c_6
\end{aligned}
$$

By applying Equation 3.1, it is easy to see that, $c_4 = c_5 = c_6 = 0$. We can calculate the other constants by solving the simultaneous linear equation given below.

$$
\begin{bmatrix} t_{fin}^5 & t_{fin}^4 & t_{fin}^3 \\ 5t_{fin}^4 & 4t_{fin}^3 & 3t_{fin}^2 \\ 20t_{fin}^3 & 12t_{fin}^2 & 6t_{fin} \end{bmatrix}
\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}
=
\begin{bmatrix} q_{fin}^{(0)} \\ 0 \\ 0 \end{bmatrix}
$$

The above trajectory $q^{(0)}(t)$ is the trajectory that minimizes the energy of jerk in the time period t_{fin}. As remarked earlier, the trajectory can be generated for different final time values t_{fin} and the velocities and the resonant frequency suppression capabilities can be analyzed.

3.2.2 Implementation, Results and Discussion

To foster direct comparison with the time optimal trajectory discussed in the previous subsection, we generate the minimum jerk trajectory with $t_{fin} = 0.1563$ sec, the optimum total time for the trajectory depicted in Figure 3.4. The motion profile of the resultant trajectory is shown in Figure 3.9.

We notice that the maximum velocity this trajectory reaches is only 339.1835 rad/sec while its time optimal counterpart reaches 355.4169 rad/sec. Additionally, the peak jerk produced by this trajectory is 4.4429×10^5 rad/sec^3, more than twice the peak jerk produced by the time optimal trajectory. However, as expected the total squared jerk is almost an order of magnitude less than that of the time optimal trajectory. We see if this has any impact on the repeatability of motion in Figure 3.10. The spread, although comparable to time optimal trajectory, is slightly higher.

The results prompt us to conclude the chapter with the inference that time optimal trajectories perform better than the minimum squared jerk trajectories for the same time length. In addition, it is experimentally proven to reduce vibrations in the motion and produce a more reliable motion over scan cycles at high speeds, thus satisfying the requirements for application to BWS.

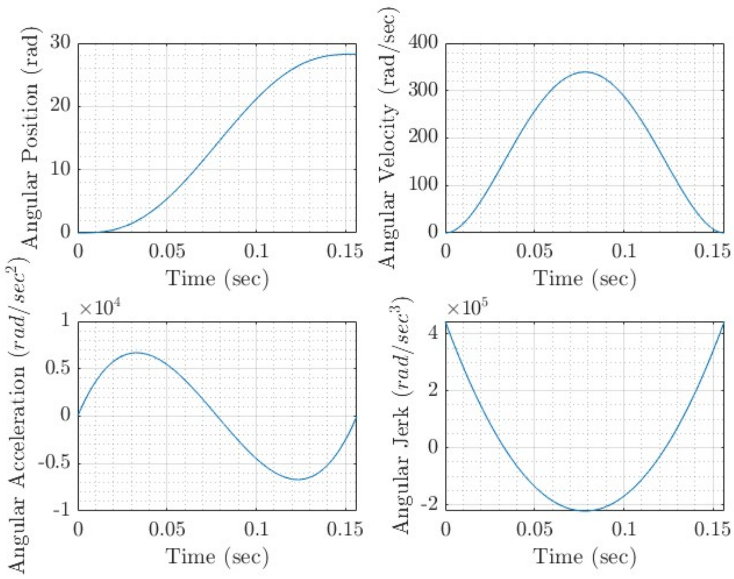


Figure 3.9: Motion Profile of a minimum squared jerk trajectory

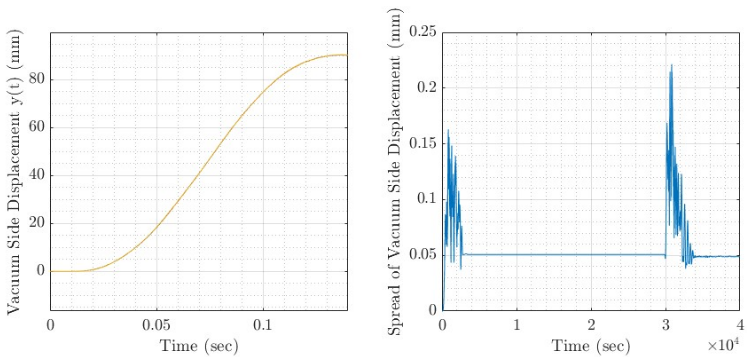


Figure 3.10: Vacuum Side Displacements and their spread over 10 scan cycles

Chapter 4

Mathematical Preliminaries of Data-driven Control Methods

The aim of this chapter is to provide the mathematical fundamentals of the two data-driven control methods namely DeePC and SPC, that form the core subject of Chapter 5 to make this thesis self-contained. The understanding of this chapter is key to build up on the insights presented in the next chapter. However, the reader may move on to the next chapter about the implementation of the methods if they have a thorough understanding of the fundamentals.

4.1 Model Predictive Control

Data Enabled Predictive Control is an adaptation of the popular Model Predictive Control (MPC), with the main difference being the fact that a parametric model of the system is replaced directly with data for DeePC. We present a brief review of the MPC Algorithm [7] in order to enable a better understanding and to build insights on how it compares with the DeePC Algorithm. Consider the discrete-time Linear Time Invariant (LTI) System described by the below equations.

$$\begin{aligned} &x(t+1) = Ax(t) + Bu(t) \\ &y(t) = Cx(t) + Du(t) \end{aligned} \tag{4.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$, denote the parameters of a system with m inputs and p outputs. The vectors $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ denote the system state, inputs and outputs at time instant $t \in \mathbb{Z}_{\geq 0}$. Model Predictive Control solves an optimization problem at each time instant t in a receding-horizon manner, with a prediction horizon of N to track the given reference trajectory $r = \{r(t), r(t+1), \dots, r(t+N)\} \in \mathbb{R}^p$ which is given as,

$$\begin{aligned} &\min_{x,u,y} \quad \sum_{k=0}^{N-1} ||y_k - r_{t+k}||_Q^2 + ||u_k||_R^2 \\ &\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k \quad \forall k \in \{0 \dots N-1\} \\ &y_k = Cx_k + Du_k \quad \forall k \in \{0 \dots N-1\} \\ &x_0 = \hat{x}(t) \end{aligned} \tag{4.2}$$

$$u_{lb} \leq u_k \leq u_{ub} \quad \forall k \in \{0 \dots N-1\}$$
$$y_{lb} \leq y_k \leq y_{ub} \quad \forall k \in \{0 \dots N-1\}$$

The vectors x_k, y_k, and u_k for all $k \in \{0 \dots N-1\}$ represent predicted values of the state, output, and control input at time instant k. The matrices $Q \in \mathbb{R}^{p \times p}$ and $R \in \mathbb{R}^{m \times m}$ denote output cost and control input cost matrices. The estimated state value at time t is denoted as $\hat{x}(t)$. The sets $u_{ub}, u_{lb} \in \mathbb{R}^m$ indicate upper and lower bound constraints on each control input u, and $y_{ub}, y_{lb} \in \mathbb{R}^p$ represent upper and lower bound constraints on each output y. The MPC algorithm solves the optimization problem given by 4.2 in a receding horizon manner while continuously updating the state estimate $\hat{x}(t)$.
The MPC algorithm relies on an accurate system model described by matrices A, B, C, and D in equation (4.1) to solve the optimization problem 4.2. However, difficulties arise when a system can't be precisely modeled or identified due to complexity and noise. DeePC addresses this challenge by enabling control without the need for an explicit model or system identification techniques.

4.2 Data Enabled Predictive Control

Traditional control methods rely on accurate parametric models, which can be challenging for complex systems. Data-driven control, leveraging the abundance of available data, eliminates the need for precise system models. Optimal trajectory tracking, a key challenge in control systems, is often addressed by Model Predictive Control (MPC), requiring accurate system models. Data-Enabled Predictive Control (DeePC), proposed in [7], offers an alternative by relying solely on data, avoiding the need for any system model. DeePC utilizes a behavioral systems perspective [22] and solves a convex optimization problem online, similar to MPC, based on Hankel matrices constructed with data exhibiting persistence of excitation [31]. This section introduces the mathematical foundations of DeePC, emphasizing its non-parametric approach to system behavior.

4.2.1 Behavioral Systems Theory

Classical Systems Theory associates a dynamic system with a specific representation, such as the state space representation, defining system properties accordingly. In contrast, Behavioral Systems Theory views a dynamic system based on the subspaces of its inputs and outputs. This section introduces definitions and properties of dynamical systems from a behavioral theory perspective, crucial for understanding DeePC [23].

Definition 4.2.1. (Dynamical System) A dynamical system is a 3-tuple $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathcal{B})$ with $\mathbb{Z}_{\geq 0}$ being the discrete-time axis, $\mathbb{W}$ the signal space and $\mathcal{B} \subseteq \mathbb{W}^{\mathbb{Z}_{\geq 0}}$ denotes the behaviour of the system.

Definition 4.2.2. (Linearity, Time Invariance and Completeness of a Dynamical System)

- The dynamical system $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathcal{B})$ is linear if $\mathbb{W}$ is a vector space and $\mathcal{B}$ is a linear subspace of $\mathbb{W}^{\mathbb{Z}_{\geq 0}}$
- The dynamical system $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathcal{B})$ is time invariant if $\mathcal{B} \subseteq \sigma\mathcal{B}$ where σ denotes the forward time shift operator given by $\sigma w(t) = w(t+1)$ and $\sigma\mathcal{B} = \{\sigma w | w \in (B)\}$
- The dynamical system $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathcal{B})$ is complete if and only if its behaviour is closed in the topology of point-wise convergence, i.e., if $w_i \in \mathcal{B}$, for $i \in \mathbb{B}$ and $w_i \to w(t) \forall t \in \mathbb{Z}_{\geq 0}$, then

$w \in \mathcal{B}$. The completeness property implies that if the dynamical system is LTI, then it is finite dimensional.

The systems in $\mathcal{L}^{m+p}$ are denoted by $\mathcal{B}$ and are characterized by linearity, time invariance, and completeness. The behavior $\mathcal{B}$ is represented as the product space $\mathcal{B}^u \times \mathcal{B}^y$, where $\mathcal{B}^u = (\mathbb{R}^m)^{\mathbb{Z}_{\geq 0}}$ and $\mathcal{B}^y = (\mathbb{R}^p)^{\mathbb{Z}_{\geq 0}}$.
The input signal's persistence of excitation is crucial, where a signal is persistently exciting if it is rich enough, and its corresponding output signal captures the system's complete behavior. The Hankel matrix of order L for an input signal u, denoted as $\mathcal{H}_L(u)$, is formed by stacking sub sequences of length $L+1$ from the input sequence.

$$\mathcal{H}_L(u) = \begin{pmatrix} u_1 & u_2 & \dots & u_{T-L+1} \\ u_2 & u_3 & \dots & u_{T-L} \\ \vdots & \vdots & \ddots & \vdots \\ u_L & u_{L+1} & \dots & u_T \end{pmatrix} \tag{4.3}$$

Definition 4.2.3. (Persistence of Excitation [31]) The input u is said to be persistently exciting of the order of L if the Hankel Matrix $\mathcal{H}_L(u)$ given by 4.3 has full row rank.

The properties controllability and observability of a behavioral system $\mathcal{B}$ are two other important preliminaries required to establish the concept of DeePC.

Definition 4.2.4. (Controllability) A system $\mathcal{B} \in \mathcal{L}^{m+p}$ is said to be controllable if $\forall T \in \mathbb{Z}_{\geq 0}$, $w^1 \in \mathcal{B}_T$, $w^2 \in \mathcal{B}$ $\exists w \in \mathcal{B}$ and $T' \in \mathbb{Z}_{\geq 0}$ such that $w_t = w_t^1$ for $1 \leq t \leq T$ and $w_t = w^2_{t-T-T'}$ for $t > T + T'$

In other words, the system $\mathcal{B}$ is controllable if any two trajectories can be patched together in finite time. The property of observability is essential to define the lag $l(\mathcal{B})$ of the system $\mathcal{B}$. Consider the equivalent parametric representation of $\mathcal{B} \in \mathcal{L}^{m+p}$ given by $\mathcal{B}(A, B, C, D)$. The parametric representation with the smallest state dimension is called the minimal representation of $\mathcal{B}$ and it's order is denoted as $n(\mathcal{B})$. We also introduce the lower triangular Toeplitz matrix of $\mathcal{B}(A, B, C, D)$ as follows:

$$\mathcal{T}_N(A,B,C,D) = \begin{pmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N-2}B & \dots & CB & D \end{pmatrix} \tag{4.4}$$

Definition 4.2.5. (Lag of a Dynamical System) The lag $l(\mathcal{B})$ of the behavioral system $\mathcal{B}$ is defined as the smallest integer l such that the observability matrix $\mathcal{O}_l(A, C) = col(C, CA, \dots, CA^{l-1})$ has rank $n(\mathcal{B})$.

Based on the above properties and definitions, we introduce two Lemmata that are central to the concept of DeePC.

Lemma 4.2.1. Let $\mathcal{B} \in \mathcal{L}^{m+p}$ be a behavioral system with minimal representation $\mathcal{B}(A, B, C, D)$. Let $T_{ini}, N \in \mathbb{Z}_{\geq 0}$ with $T_{ini} \geq l(\mathcal{B})$ and $col(u_{ini}, u, y_{ini}, y) \in \mathcal{B}_{T_{ini}+N}$. Then there exists a unique $x_{ini} \in \mathbb{R}^{n(\mathcal{B})}$ such that

$$y = \mathcal{O}_N(A, C)x_{ini} + \mathcal{T}_N(A, B, C, D)u \tag{4.5}$$

The equation 4.5 exemplifies the fact that given a sufficiently long window of data $col(x_{ini}, y_{ini})$, the state and the output to which the system $\mathcal{B}$ goes to is unique and given the matrices A, B, C, D the system state x_{ini} can be calculated.

Lemma 4.2.2. (Fundamental Lemma [22]) Consider a controllable system $\mathcal{B} \in \mathcal{L}^{m+p}$. Let $T, t \in \mathbb{Z}_{\geq 0}$ and $w = col(u, y) \in \mathcal{B}_T$. If u is persistently exciting of the order $t + n(\mathcal{B})$ then,

$$colspan(\mathcal{H}_t(w)) = \mathcal{B}_t \tag{4.6}$$

The Fundamental Lemma proves that if the data points are sufficient and persistently exciting, then the trajectory of the system $\mathcal{B}$ can be constructed using a finite number of data points. This replaces the need for a parametric model for control purposes.

4.2.2 DeePC Algorithm

We consider that the data is generated from an LTI System $\mathcal{B} \in \mathcal{L}^{m+p}$. Let $T, T_{ini}, N \in \mathbb{Z}_{\geq 0}$ such that $T \geq (m+1)(T_{ini} + N_n(\mathcal{B})) - 1$ to ensure the persistence of excitation. Assume that $u^d = col(u_1^d, u_2^d, \ldots, u_T^d) \in \mathbb{R}^{Tm}$ be the sequence of input data applied to $\mathcal{B}$ and $y^d = col(y_1^d, y_2^d, \ldots, y_T^d) \in \mathbb{R}^{Tp}$ be the corresponding output data generated by the system that are collected offline. The collected data is split into two parts as defined below.

$$\begin{pmatrix} U_p \\ U_f \end{pmatrix} = \mathcal{H}_{T_{ini}+N}(u^d) \qquad \begin{pmatrix} Y_p \\ Y_f \end{pmatrix} = \mathcal{H}_{T_{ini}+N}(y^d) \tag{4.7}$$

Here, the matrices U_p and Y_p contain the initial T_{ini} rows of the respective Hankel matrices while the matrices U_f and Y_f contain the final N rows of the respective Hankel matrices. With the help of 4.6 given by Lemma 2.1.2, we can generate trajectories of length $T_{ini} + N$ with the help of the collected data. This trajectory would belong to the LTI System $\mathcal{B}_{T_{ini}+N}$ if and only if $\exists g \in \mathbb{R}^{T-T_{ini}-N+1}$ such that

$$\begin{pmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{pmatrix} \tag{4.8}$$

Moreover, if $T_{ini} \geq l(\mathcal{B})$ then Lemma 2.1.1 through 4.5 ensures that $\exists x_{ini}$, which is unique for a unique output y. However, the state x_{ini} is implicit and is not required in any calculations. Equation 4.8 enables us to solve for a g given the initial trajectory u_{ini} and y_{ini} and a sequence of future inputs u. This can then be used to predict the future output trajectory as $y = Y_f g$. On the contrary, given a reference output value r, 4.8 can be used to calculate the control input u as well.
Furthermore, given the prediction horizon $N \in \mathbb{Z}_{\geq 0}$, the reference trajectory $r \in (\mathbb{R}^p)^{\mathbb{Z}_{\geq 0}}$, the control and output cost matrices R and Q, the past input and output data $u_{ini}, y_{ini} \in \mathcal{B}_{T_{ini}}$ and the constraints $u_{lb}, u_{ub} \in \mathbb{R}^m$ and $y_{lb}, y_{ub} \in \mathbb{R}^m$, the DeePC optimization problem can be formulated as,

$$\min_{g,u,y} ||y_k - r_{t+k}||_Q^2 + ||u_k||_R^2$$

$$
\text{such that} \begin{pmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{pmatrix} \tag{4.9}
$$
$$
u_{lb} \leq u_k \leq u_{ub}, \forall k \in \{0, 1, \ldots, N-1\}
$$
$$
y_{lb} \leq y_k \leq y_{ub}, \forall k \{0, 1, \ldots, N-1\}
$$

The formulation of the above optimization problem enables us to define the DeePC algorithm as presented in.

Algorithm 2 DeePC Algorithm [7]

Inputs $col(u^d, y^d), u_{ini}, y_{ini}, r, Q, R, N$
Output $u^\star$

1: Compute 4.9 for g^*
2: Calculate optimal input sequence $u^* = U_f g^*$
3: Apply the input $(u(t), \ldots, u(t+s)) = (u_0^*, \ldots, u_s^*)$ for some $s \leq N-1$
4: Set t to $t+s$ and update u_{ini} and y_{ini} to the T_{ini} most recent values
5: Iterate through steps 1 to 4

4.2.3 Quadratic Regularization on DeePC

The work [17] proves that robustness to disturbances in the data is induced by adding a quadratic regularization term to the optimization problem given by in 4.9 and the modified DeePC optimization problem is given as,

$$
\min_g ||Ag - b||_2^2 + \lambda_g ||g||_2^2
$$
$$
\text{such that } Gg - q \leq 0 \tag{4.10}
$$
$$
\text{where, } A = \begin{pmatrix} \lambda_y^{\frac{1}{2}} Y_p \\ R^{\frac{1}{2}} U_f \\ Q^{\frac{1}{2}} Y_f \end{pmatrix}, b = \begin{pmatrix} \lambda_y^{\frac{1}{2}} y_{ini} \\ 0 \\ Q^{\frac{1}{2}} r \end{pmatrix}, G = \begin{pmatrix} U_f \\ Y_f \\ -U_f \\ -Y_f \\ U_p \\ -U_p \end{pmatrix} \text{ and } q = \begin{pmatrix} u_{ub} \\ y_{ub} \\ -u_{lb} \\ -y_{lb} \\ -u_{ini} \\ u_{ini} \end{pmatrix}
$$

The values λ_u and λ_y correspond to the penalizing factor that have been added as a result of eliminating the equality constraints and adding them to the cost function directly, whereas λ_g is the regularization term that imposes a one-norm penalty on g to increase the robustness of the DeePC algorithm against uncertainty and noisy data. The optimization problem given by 4.10 would be of major interest to us as we try to tackle the problem of improving the speed of the DeePC Algorithm.

4.3 Subspace Predictive Control

Subspace Predictive Control (SPC) is another data-driven control technique, similar to DeePC, but uses the subspace identification method to predict the values of the output instead of the Fundamental Lemma which is used in DeePC. SPC was first proposed in [12].

The major idea behind subspace identification is that we can use a linear predictor to approximate the future output values, given the past inputs and outputs and the future inputs, i.e.,

$$\hat{Y}_f = L_w \begin{bmatrix} Y_p \\ U_p \end{bmatrix} + L_u U_f \tag{4.11}$$

in which the previous definitions of Hankel matrices are still valid. Thus, the first step becomes the estimation of the unknown parameters L_w and L_u as,

$$\min_{L_w, L_u} ||Y_f - \begin{bmatrix} L_w & L_u \end{bmatrix} \begin{bmatrix} Y_p \\ U_p \\ U_f \end{bmatrix} || \tag{4.12}$$

After the estimation of the matrices, the future outputs can be predicted as,

$$\hat{y}_f = L_w \begin{bmatrix} y_{ini} \\ u_{ini} \end{bmatrix} + L_u u_f \tag{4.13}$$

where u_f denotes the future inputs and $\hat{y}_f$ represents the predicted outputs. Thus, the SPC optimization problem can be formulated as,

$$\begin{aligned} &\min_{u,y} ||y_k - r_{t+k}||_Q^2 + ||u_k||_R^2 \\ &\text{such that } y_k = L_w \begin{bmatrix} y_{ini} \\ u_{ini} \end{bmatrix} + P_u u_k \\ &u_{lb} \leq u_k \leq u_{ub} \\ &y_{lb} \leq y_k \leq y_{ub} \end{aligned} \tag{4.14}$$

A receding horizon algorithm, similar to Algorithm 2 can be used for the implementation of SPC. Having introduced the mathematical fundamentals of the control algorithms to be tried out, we move on to their implementations in the subsequent chapter.

Chapter 5

Implementation of Data-driven Control Methods

An important facet of the BWS is its ability to perform and endure multiple scan cycles throughout its lifetime. Chapter 3 dealt with developing an optimal reference trajectory that would minimize vibrations on the wire improving its performance. Chapter 2 explained the importance of accurate positioning of the wire during these scan cycles in beam size estimation. The other important piece of puzzle is to make the instrument follow the reference trajectory perfectly, which is the subject of this chapter. In this chapter, we describe the system to be controlled, motivate the needs for a data-driven control scheme, describe different methods discuss, and analyze the results obtained.

5.1 PMSM Model and Field Oriented Control

A brief introduction to the working of the PMSM is given in subsection 1.3.1. The PMSM drives the wire through the beam and hence, the wire position is controlled by controlling the angular displacement θ_m of the motor. Understanding the relationship between the electrical quantities of the motor is crucial in order to design an effective control system. The following subsections define the mathematical model of PMSM and explain the existing control architecture with the associated difficulties.

5.1.1 Mathematical Model

The idea of using a synchronous frame of reference (d, q) to represent three phase electrical quantities (a, b, c) is well-researched and documented [28]. The d-q frame simplifies the analysis, provides an intuitive understanding and easier for control algorithm development. The voltages in the d-q frame of reference are given as,

$$u_d = Ri_d + L_d\frac{di_d}{dt} - \omega_e L_q i_q \tag{5.1}$$

$$u_q = Ri_q + L_q\frac{di_q}{dt} + \omega_e L_d i_d + \omega_e \psi_p \tag{5.2}$$

In the above equations, i_d and i_q are the currents in the d-q frame respectively. L_d and L_q are the inductances relative to the d-q frame, ω_e is the angular speed of rotation of the motor from the electrical frame of reference while ψ_p is the permanent magnet flux linkage defined as $\psi_p = \frac{2K_T}{3p}$. K_T is the motor torque constant and p is the number of pole pairs which are inherent properties of the motor. The relation ship between the angular speed of rotation in the electrical frame and mechanical frame of reference is defined as,

$$\omega_e = p\omega_m = p\frac{d\theta_m}{dt} \tag{5.3}$$

When the inductances, $L_d = L_q$, the torque generated by the motor can easily be written as $T_m = K_T i_q$. The mechanical equation of the motor can be written from the torque-balance equation as,

$$\frac{d\omega_m}{dt} = \frac{p}{J}(K_T i_q - \frac{B}{p}\omega_m - T_L) \tag{5.4}$$

where B and J are the viscous coefficient and the moment of inertia of the load, while T_L is the load torque. The relationship between the quantities in three phase, the abc frame of reference to the quantities in d-q frame is given by the transformation,

$$\begin{aligned}\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} &= \frac{2}{3}\begin{bmatrix} \cos(\theta_e) & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin(\theta_e) & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \\ &= K_P K_C \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}\end{aligned} \tag{5.5}$$

$$\text{where, } K_P = \sqrt{\frac{2}{3}}\begin{bmatrix} \cos(\theta_e) & \sin(\theta_e) & 0 \\ -\sin(\theta_e) & \cos(\theta_e) & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } K_C = \sqrt{\frac{2}{3}}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

where $\theta_e = p\theta_m$ is the angular position from the electrical frame of reference. In the above set of equations, K_C is the Clarke transformation that transforms abc to $\alpha\beta 0$ frame and K_P is the Park transformation that transforms $\alpha\beta 0$ to the dq frame of reference. The same equations can also be used for voltage transformation. Together, they form the Clarke-Park Transformation equations. For converting quantities in dq frame back to abc frame of reference, the Inverse Clarke-Park Transformation given by, $i_{abc} = K_C^{-1} K_P^{-1} i_{dq0}$ is used. These transformations were first proposed in [26].
The equations (5.1),(5.2), (5.3) and (5.4) describe the behaviour of the PMSM entirely. Having described the mathematical model of the PMSM that we control, we discuss the existing control strategy in place in the following subsection.

5.1.2 Field Oriented Control

Field Oriented Control (FOC), also known as Vector Control, is a sophisticated control technique widely employed in electric motor drives, particularly in applications requiring high-performance and precise control. FOC provides a means to decouple the control of the motor's magnetic field and torque components, allowing for independent and optimized control of each. By aligning the motor's magnetic field with the rotor flux, FOC minimizes torque ripple, improves efficiency, and enhances dynamic response. This control strategy involves transforming the three-phase stator currents from the stationary reference frame to a rotating reference frame, simplifying the control equations. FOC is commonly used in various industries, including robotics, electric vehicles, and industrial automation, where achieving precise control over motor behavior is critical for efficient and effective operation.

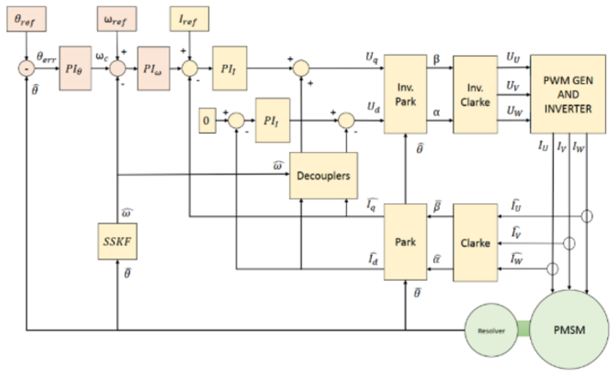


Figure 5.1: Control Architecture of FOC for PMSM in BWS Application [11]

Figure 5.1 shows the FOC architecture for the PMSM used in BWS. The fundamental idea of the architecture is to have a cascade control loop with the outermost loop regulating the angular position θ_m, which makes adjustment to the reference of the middle loop that controls the angular velocity ω_m which in turn adjusts the reference to the innermost loop that controls the quadrature current i_q. A separate control loop is used to keep the direct current i_d at 0. The manipulated variables are the voltages u_d and u_q, which is subjected to the inverse Clarke-Park transformation to get the phase voltage values before being fed to the PMSM through a PWM and Inverter. The reference trajectory for the cascade loops for each scan cycle is generated offline and then fed to the control system. The control system also includes decouplers for i_d and i_q and a Steady State Kalman Filter (SSKF) for the estimation of ω_m. A detailed account on the design of the control scheme can be found in [21]. Over the years, this control scheme has been validated on different experimental setups, and is operational on the accelerators. More information about the design and the validation methodology and an improvement to the cascade control scheme can be found in [11] and [10] respectively.

5.1.3 Drawbacks and Motivation for Data-driven Control

The design of the FOC scheme, depends to a large extent on the parameters of the PMSM as in any other control system design. However, the parameters associated with PMSM change over time due to different reasons. Wear and tear of the mechanical part of the instrument, changes to the orientation of the instrument, changes to the load torque, and harsh working conditions including high electromagnetic radiation can cause these parameter changes. The constants of the PI controllers in use right now are sensitive to these parameter changes causing a degradation in the required control performance. This requires constant tuning and re-tuning of the controller until satisfactory performance is attained.
The tuning process is made more complex by the fact that there are four controllers to be tuned, and the tuning is inter-dependent. This warrants the exploration of simpler control schemes, such as the one proposed in [10]. Again, the drawback of this scheme is the requirement of accurate system model which is more involved for real-world applications such as the BWS. A

system identification based controller design has been carried out in [9], which urges research into more advanced control schemes.
To summarize, the requirement from a new control scheme is two-fold:

1. The control scheme should be adaptive to parameter changes in the system and more robust to the operating conditions
2. The control scheme should have a simple implementation and operate in real time at high sampling rates

Although several approaches to self-commissioning of a PMSM have been proposed already, the implementation of Data-enabled Predictive Control (DeePC) and other data-driven methods to current control in synchronous motor drives in [4] and [5] motivate us in the implementation of data-driven methods for PMSM. The data-driven control methods are advantageous with their simple online implementation and can be easily adapted to parameter changes. In the subsequent sections of this chapter, we discuss these control schemes and the results obtained from their implementation.

5.2 Data-enabled Predictive Control Methods

A comprehensive overview of the DeePC algorithm and the mathematical fundamentals of DeePC has been provided in Chapter 4. In this section we focus on DeePC, and modifications to the algorithm to make it suitable for real-time application. We propose a novel architecture that controls the angular position, velocity and the current of the PMSM and analyze the control capabilities of the proposed scheme to parameter variations.

5.2.1 Closed-form Solution Derivation

It is really important that the developed control scheme works at high sampling rates in the real-time system about 16 kHz. However, DeePC requires an optimization problem to be solved at every time instant increasing its computational complexity. However, in the absence of constraints on the input and output, a closed-form solution can be derived and the solution to the optimization problem is reduced to a linear map. This solution has already been developed in [17]. We provide a detailed derivation of the closed-form solution. In the absence of constraints, the optimization problem given by 4.10 can be reduced to,

$$\begin{aligned} &\min_{g} ||Ag - b||_2^2 + \lambda_g ||g||_2^2 \\ &\text{such that } U_p g - u_{ini} = 0 \\ &\text{where, } A = \begin{pmatrix} \lambda_y^{\frac{1}{2}} Y_p \\ R^{\frac{1}{2}} U_f \\ Q^{\frac{1}{2}} Y_f \end{pmatrix}, b = \begin{pmatrix} \lambda_y^{\frac{1}{2}} y_{ini} \\ 0 \\ Q^{\frac{1}{2}} r \end{pmatrix} \end{aligned} \tag{5.6}$$

We recall that the solution to the optimization problem is $g^\star \in T - T_{ini} - N + 1$ and $u^\star = U_f g^\star$ is the optimal control input. For the sake of brevity, we define $H = T - T_{ini} - N + 1$. A closed form solution to this equality constrained optimization problem can be developed by analyzing the Karush-Kuhn-Tucker (KKT) Conditions. The Lagrangian of the optimization problem can be written as,

$$L(g, \mu) = ||Ag - b||_2^2 + \lambda_g ||g||_2^2 + \mu^\top (U_p g - u_{ini}) \tag{5.7}$$

where μ is the vector of dual variables corresponding to the equality constraints. There exists a solution $(g^\star, \mu^\star)$ to the KKT conditions of the optimization problem which are given as,

$$\frac{\partial L(g, \mu)}{\partial g} = 0 \implies (2A^\top A + 2\lambda_g \mathbb{I}_H)g + U_p^\top \mu = 2A^\top b \tag{5.8}$$

$$U_p g = u_{ini} \tag{5.9}$$

They form a pair of simultaneous equations in g and μ and the solution to the equations gives the optimal values. It is easy to see that the solution can be written as,

$$\begin{bmatrix} g^\star \\ \mu^\star \end{bmatrix} = \begin{bmatrix} 2A^\top A + 2\lambda_g \mathbb{I}_H & U_p^\top \\ U_p & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} 2A^\top b \\ u_{ini} \end{bmatrix} \tag{5.10}$$

Let $M = \begin{bmatrix} 2A^\top A + 2\lambda_g \mathbb{I}_H & U_p^\top \\ U_p & \mathbf{0} \end{bmatrix}^{-1}$. Then the optimal control input can be written as the first H rows of the expression given by,

$$u^\star = \left[U_f M \begin{bmatrix} 2A^\top b \\ u_{ini} \end{bmatrix} \right]_{(1:H,\star)} \tag{5.11}$$

$$= \left[U_f M \begin{bmatrix} 2A^\top \begin{bmatrix} \lambda_y^{\frac{1}{2}} y_{ini} \\ \mathbf{0} \\ Q^{\frac{1}{2}} r \end{bmatrix} \\ u_{ini} \end{bmatrix} \right]_{(1:H,\star)} \tag{5.12}$$

From the above equation, it is easy to see that the optimal control input $u^\star$ can be written as the linear map, $u^\star = K_y y_{ini} + K_r r + K_u u_{ini}$. Thus, we have shown that the control algorithm does not require solving of an optimization problem at every time instant but rather can be expressed by matrix vector multiplications and addition operations in the absence of constraints on the inputs and outputs. This greatly improves the real-time performance of the control algorithm. However, a question arises as to how restrictive it is to assume the absence of input and output constraints. In the application to BWS, the assumption is quite justifiable since it is supposed to track a pre-defined trajectory over and over. Precautions can be taken while generating the reference trajectory to ensure that the nominal values of inputs and outputs are always in an admissible range. Moreover, the existing control scheme does not impose any constraints but rather uses saturation blocks to prevent the application of very high or very low inputs to the system.

5.2.2 Nominal DeePC Architecture

The DeePC control algorithm consists of an offline data collection step and the online algorithm that has to be executed every time instant. We begin designing the current control loop that only regulates i_q to a pre-defined reference trajectory and tries to keep i_d close to 0. A simplified architecture of the offline design is shown in Figure 5.2. In all the architecture portrayals, the

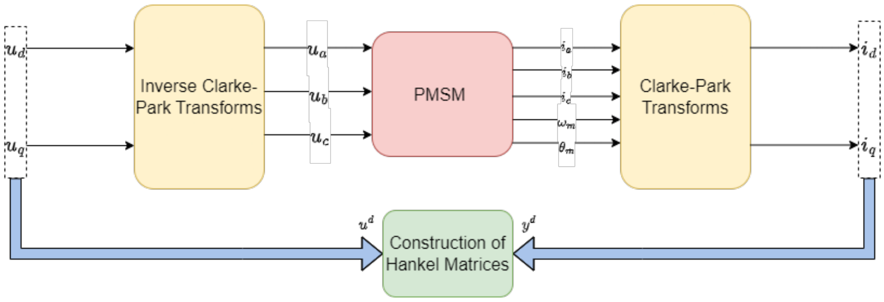


Figure 5.2: Offline Design of Nominal DeePC

Pulse Width Modulation (PWM) and Inverter components that lie between the Inverse Clarke-Park Transform and the PMSM blocks in the actual system are not shown for simplicity of the system and since the focus is on the architecture of the controller rather than the system in itself. In order to control the currents, we have to regulate the voltages u_d and u_q,evident from equations (5.1) and (5.2) which form the control inputs. Thus we collect the data matrices $u^d = [u_d(1)\ldots u_d(fT_{fin}) \quad u_q(1)\ldots u_q(fT_{fin})]$ and $y^d = [i_d(1)\ldots i_d(fT_{fin}) \quad i_q(1)\ldots i_q(fT_{fin})]$, where T_{fin} determines the length of the data collection. The Hankel matrices U_p,U_f,Y_p and Y_f can be constructed as given by 4.7. It is to be ensured that the inputs u_d and u_q satisfy the persistency of excitation condition. The offline design also includes the construction of the A matrix defined in the optimization problem 5.6. Following the offline design, we move on to the online implementation of the controller depicted in the figure 5.3.

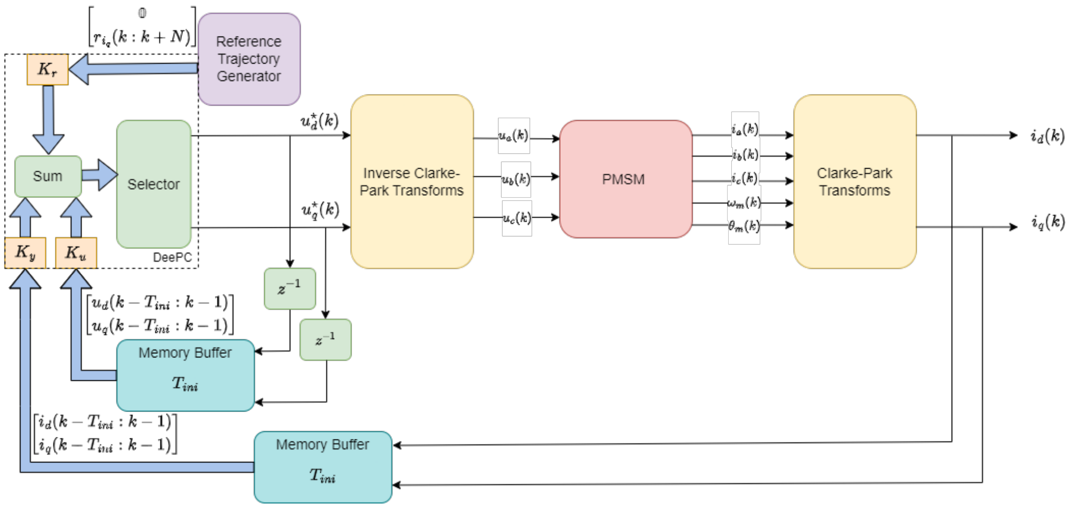


Figure 5.3: Online Control Architecture of Nominal DeePC and signals at instant k

As described in subsection 5.2.1, the controller is reduced to a linear map with K_1,K_2 and K_3 matrices calculated offline from the constructed Hankel matrices, the input and output cost matrices R and Q, and other hyperparameters. The selector is used to output the first element of the optimal vectors $u_d^\star, u_q^\star \in N$ that is to be applied to the motor. A memory buffer of length T_{ini} is used to store the previous T_{ini} values of the inputs and outputs for feeding back these values to the controller. The reference trajectory generator outputs a vector of dimension N for each of the outputs comprising of the reference in the current and future reference values.

Hyperparameter	**Value**
Data Length T_{fin}	0.02 s
Sampling Frequency f_s	16 kHz
Prediction Horizon N	15
Initial Trajectory Length T_{ini}	5
Input Cost Matrix R	$0.97\mathbb{I}_{2N}$
Output Cost Matrix Q	$\begin{bmatrix} 12\mathbb{I}_N & \mathbf{0} \\ \mathbf{0} & 28.5\mathbb{I}_N \end{bmatrix}$
Output Slack Penalization λ_y	10
Quadratic Regularization Parameter λ_g	0.1

Table 5.1: Hyperparameters for Nominal DeePC

For DeePC to perform satisfactorily, appropriate cost matrices and hyperparameters are to be selected. A note on hyperparameter tuning for DeePC is given in [16]. For the simulations, hyperparameters are tuned iteratively and are chosen based on the control performance. The hyperparameter values used for the simulation are shown in Table 5.1. Results of simulations performed on a model of the PMSM with parameters in Table 1.2 using MATLAB and Simulink are shown in Figures 5.4 and 5.5. The model of the PMSM utilized for the simulations was developed in the work [32]

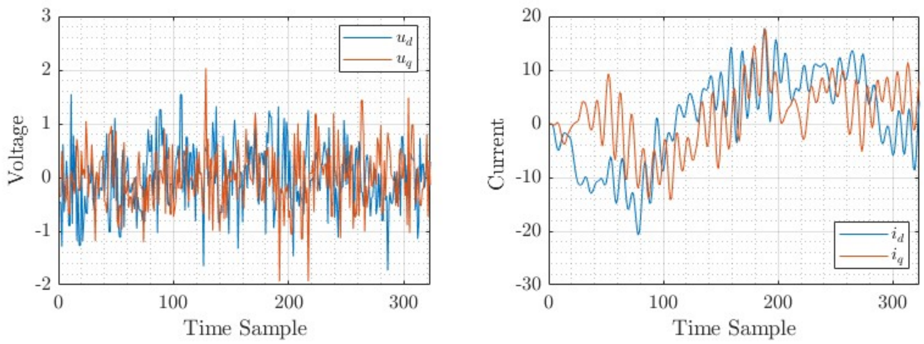


Figure 5.4: Offline data collection of voltages (left) and currents (right)

The reference trajectory for i_q is generated from the time optimal trajectory implemented in subsection 3.1.2, and applying the mechanical model of the motor given in Equation (5.4). The consequences of using the model of the system in generating the reference trajectory is explained in the following subsection. The current i_d shows significant deviation from the ideal value of 0, while we can clearly observe a steady-state error in the tracking of i_q. This behaviour of DeePC is expected since there are disturbances affecting reference tracking, most notably the back EMF induced by the magnets. The mitigation for this is introduced in the following subsection. The angular position and velocity values are not relevant here due to poor tracking of current.

5.2.3 Incremental DeePC Architecture

The effect of the steady-state error in tracking of current can be reduced by introducing an 'integral' action in the optimization problem. This is done by re-formulating the optimization

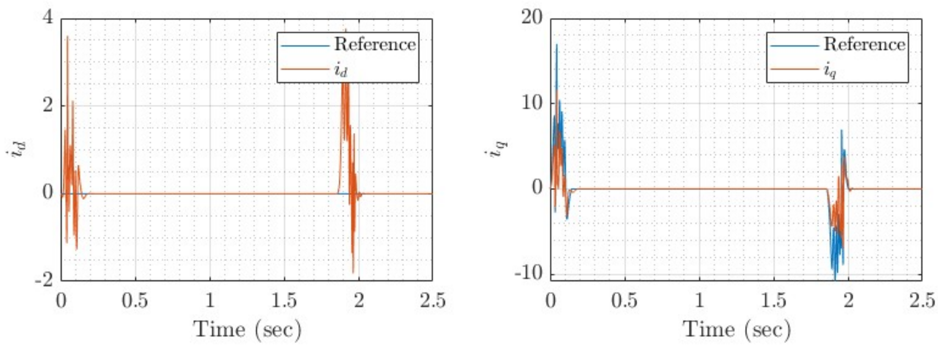


Figure 5.5: Nominal DeePC Current Control

problem 5.6, in incremental or velocity form. This solution is prevalent in MPC and is also implemented for DeePC in [5]. The optimization problem to be solved now becomes,

$$
\begin{aligned}
&\min_{g,y} ||A_1 g - b_1||_2^2 + \lambda_g ||g||_2^2 \\
&\text{such that } U_p' g - \Delta u_{ini} = 0 \\
&\text{where, } A_1 = \begin{pmatrix} \lambda_y^{\frac{1}{2}} Y_p' \\ R^{\frac{1}{2}} U_f' \\ Q^{\frac{1}{2}} Y_f' \end{pmatrix}, b_1 = \begin{pmatrix} \lambda_y^{\frac{1}{2}} \Delta y_{ini} \\ 0 \\ Q^{\frac{1}{2}} (r - y) \end{pmatrix}
\end{aligned}
\tag{5.13}
$$

The Hankel matrices U_p',U_f',Y_p' and Y_f' are constructed from the incremental data instead of the absolute values of the data collected. The values Δu_{ini} and Δy_{ini} are now made up of the incremental data of the previous time steps. The solution to the optimization problem now gives $\delta u = U_f' g$, the first instant of which is summed with the previous value in order to get the absolute value to be applied to the motor. The closed-form solution derived in subsection 5.2.1 cannot be directly applied to this optimization problem due to the presence of an additional optimization variable y. In order to be able to use the closed-form solution, a couple of extra steps are to be performed. The vector $r - y$ can be expanded as,

$$
\begin{aligned}
r - y &= \begin{bmatrix} r(k) - y(k) \\ r(k+1) - y(k+1) \\ \vdots \\ r(k+N-1) - y(k+N-1) \end{bmatrix} \\
&= \begin{bmatrix} r(k) - y(k-1) - \sum_{i=1}^{1} \Delta y(i) \\ r(k+1) - y(k-1) - \sum_{i=1}^{2} \Delta y(i) \\ \vdots \\ r(k+N-1) - y(k-1) - \sum_{i=1}^{N} \Delta y(i) \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
&= r - y(k-1)\mathbb{1}_N - \begin{bmatrix} \sum_{i=1}^{1} \Delta y(i) \\ \sum_{i=1}^{2} \Delta y(i) \\ \vdots \\ \sum_{i=1}^{N} \Delta y(i) \end{bmatrix} \\
&= r - y(k-1)\mathbb{1}_N - \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \Delta y(1) \\ \Delta y(2) \\ \vdots \\ \Delta y(N) \end{bmatrix} \\
&= r - y(k-1)\mathbb{1}_N - \mathbb{1}_{\triangle,N} Y'_f g
\end{aligned}
$$

where $\mathbb{1}_{\triangle,N}$ represents the lower triangular matrix of ones and $\mathbb{1}_N$ is the vector of ones. This simplification enables us to re-write the optimization problem as,

$$
\begin{aligned}
&\min_g ||A'g - b'||_2^2 + \lambda_g ||g||_2^2 \\
&\text{such that } U'_p g - \Delta u_{ini} = 0 \\
&\text{where, } A' = \begin{pmatrix} \lambda_y^{\frac{1}{2}} Y'_p \\ R^{\frac{1}{2}} U'_f \\ (Q^{\frac{1}{2}} + \mathbb{1}_{\triangle,N}) Y'_f \end{pmatrix}, b' = \begin{pmatrix} \lambda_y^{\frac{1}{2}} \Delta y_{ini} \\ 0 \\ Q^{\frac{1}{2}} (r - y(k-1)\mathbb{1}_N) \end{pmatrix}
\end{aligned}
\tag{5.14}
$$

This optimization problem can now be exactly solved by the closed-form solution derived in subsection 5.2.1, by replacing U_p with U'_p, u_{ini} with Δu_{ini} and A, b with A', b' respectively. The data collection step remains the same and the only change comes in constructing the Hankel matrices, which is now done using

$$
\begin{aligned}
u^d = [&u_d(1) \quad u_d(2) - u_d(1) \dots u_d(fT_{fin}) - u_d(fT_{fin} - 1) \\
&u_q(1) \quad u_q(2) - u_q(1) \dots u_q(fT_{fin}) - u_q(fT_{fin} - 1)] \\
y^d = [&i_d(1) \quad i_d(2) - i_d(1) \dots i_d(fT_{fin}) - i_d(fT_{fin} - 1) \\
&i_q(1) \quad i_q(2) - i_q(1) \dots i_q(fT_{fin}) - i_q(fT_{fin} - 1)]
\end{aligned}
$$

Thus, the offline design shown in Figure 5.2 remains the same. However, the online architecture of the control loop has some minimal changes and is shown in Figure 5.6. With the changes to the optimization problem, the controller has the same structure of a linear map with the matrices K'_r,K'_y and K'_u calculated offline with the modified matrices. The only changes occur on the inputs to the controller that have to be fed as incremental values, with the additional delay, multiplexer and difference blocks introduced for this purpose.

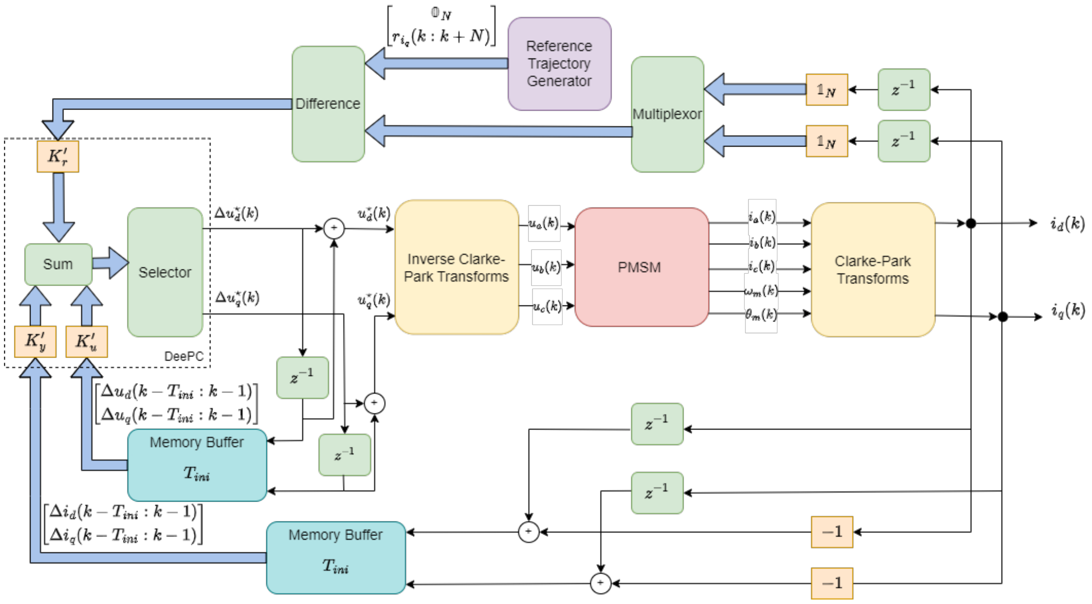


Figure 5.6: Online Control Architecture of Incremental DeePC and signals at instant k

The results of the incremental architecture in controlling the currents of the motor is shown in Figure 5.7. The parameters of the controller used are the same as tabulated in Table 5.1. The incremental DeePC controller does a much better job in the tight control of current i_q especially the rapid changes in the trajectory are tracked better when compared to the nominal DeePC controller.

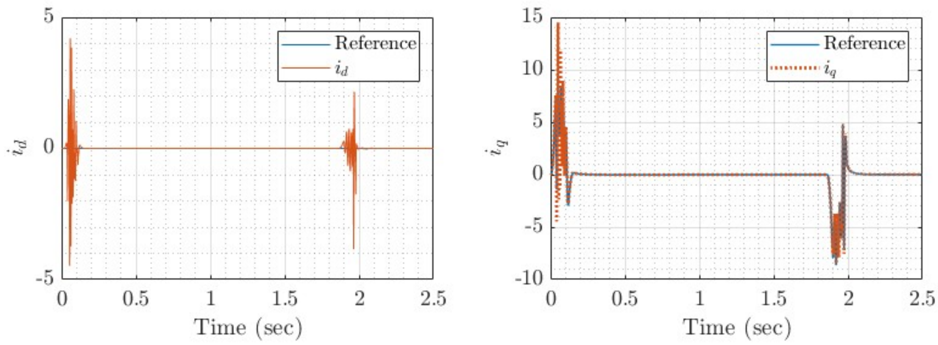


Figure 5.7: Incremental DeePC Current Control

However, a slight modification of the input cost matrix, reducing it to be $R = 0.3\mathbb{I}_{2N}$, makes the controller follow the trajectory of i_q closer than the previous result. The current i_d is also regulated better. This is seen in Figure 5.8. A comparison of the control inputs, the voltages u_d and u_q between the two input cost matrix values is shown in Figure 5.9. Counter intuitively, the voltages tend to stay closer to 0 when the cost matrix is reduced. A detailed analysis of the influence of hyper parameters and cost matrices on the stability and control performance is conducted in the forthcoming subsections.

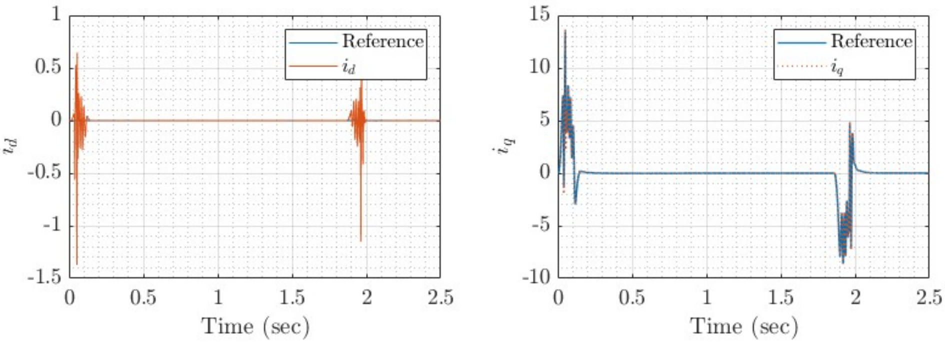


Figure 5.8: Incremental DeePC Current Control with $R = 0.3\mathbb{I}_{2N}$

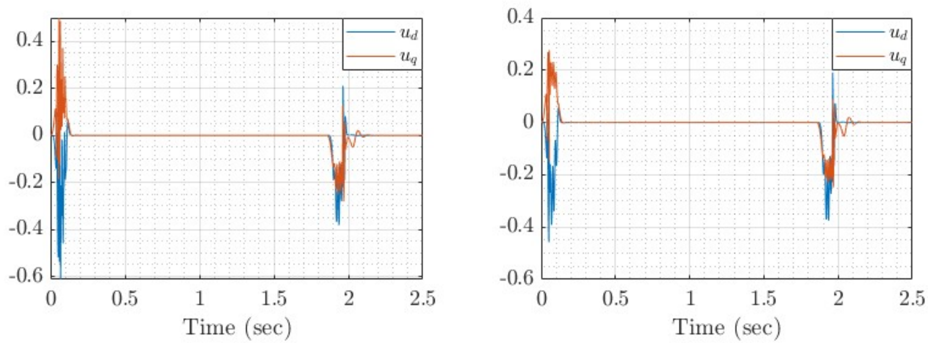


Figure 5.9: Comparison of control inputs with $R = 0.97\mathbb{I}_{2N}$ (left) and $R = 0.3\mathbb{I}_{2N}$ (right)

It is also interesting to take a look at the speed and the position of the motor and to what extent the control of current regulates these values. As mentioned, the reference trajectory for the current i_q is calculated from the required position trajectory and applying the model of the motor in equation 5.4. Figure 5.10 shows the references and the actual simulated values for speed and displacement. From the plots, it is seen that regulating only the currents is not very effective in regulating the desired values of velocity and ultimately position. This re-iterates the deficiencies of the model and the need for a data-driven controller. Moreover, the results we have obtained point out that, in order to be able to control precisely the motion of the PMSM and the wire inside the vacuum, an additional control of the velocity and/or position is required.

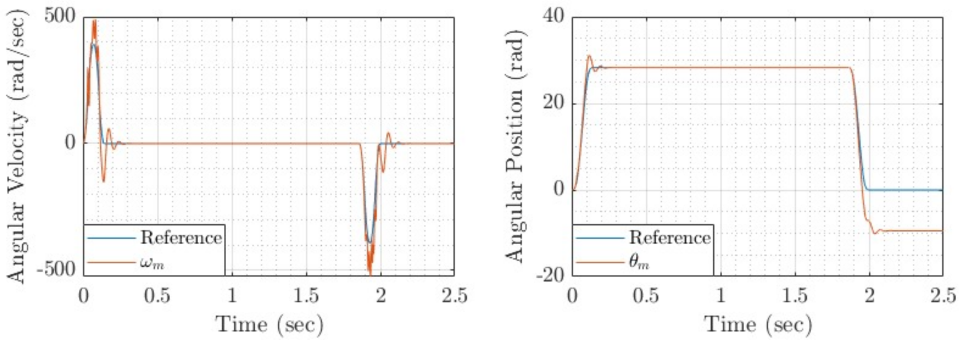


Figure 5.10: Angular velocity and position values in current control

5.2.4 Integration of Speed Control

The need for speed control for application to BWS is justified from the results shown in the previous subsection. Controlling only the currents of the motor is not effective to reach the level of precision that we desire. This motivates us to look into adding a speed control element to the controller. In this subsection, we propose a novel control architecture, applying DeePC to the speed and current control of a PMSM Drive System. This is an addition to the work done in
We continue using the incremental architecture, since it produces substantially better results than the nominal architecture. The only modification in the offline design of the controller is the additional collection of the angular velocity ω_m data in addition to the data of i_d and i_q. Thus the data collection vectors now become,

$$
\begin{aligned}
u^d = [&u_d(1) \quad u_d(2) - u_d(1) \ldots u_d(fT_{fin}) - u_d(fT_{fin} - 1) \\
&u_q(1) \quad u_q(2) - u_q(1) \ldots u_q(fT_{fin}) - u_q(fT_{fin} - 1)] \\
y^d = [&i_d(1) \quad i_d(2) - i_d(1) \ldots i_d(fT_{fin}) - i_d(fT_{fin} - 1) \\
&i_q(1) \quad i_q(2) - i_q(1) \ldots i_q(fT_{fin}) - i_q(fT_{fin} - 1) \\
&\omega_m(1) \quad \omega_m(2) - \omega_m(1) \ldots \omega_m(fT_{fin}) - \omega_m(fT_{fin} - 1)]
\end{aligned}
$$

The modified offline and online architectures are shown in Figures 5.11 and 5.12 respectively. There are no addition or deletion of blocks and the only changes are the addition of the signal ω_m to the data collection offline; and the addition of the reference for ω_m and the incremental feedback online.
Only the hyperparameters that differ from the ones mentioned in Table 5.1 are tabulated in Table 5.2. It is to be noted that the speed output is weighed almost 10 times that of the current

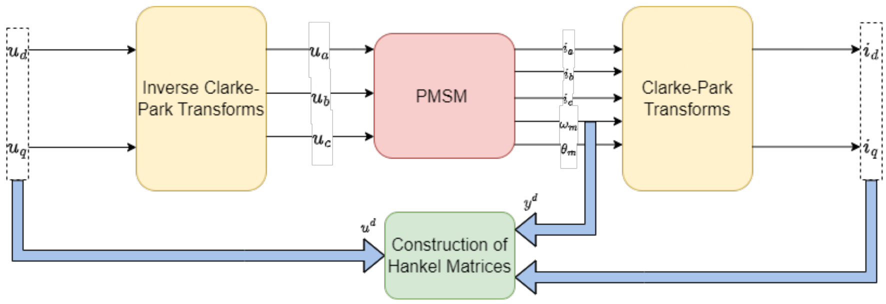


Figure 5.11: Offline Design for integration of speed control

outputs in the chosen parameters and the inputs are given much more wight when compared to the output to maintain the stability of the system.

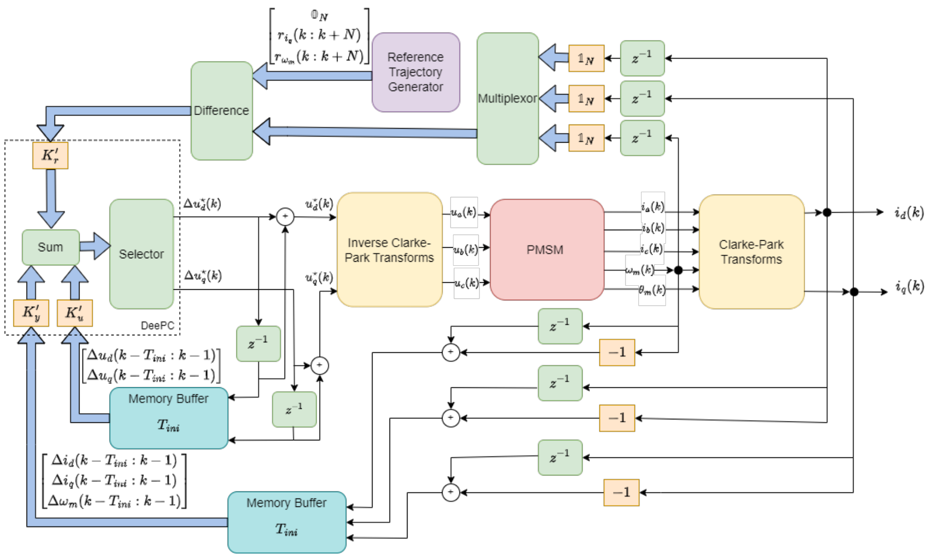


Figure 5.12: Online Architecture for integration of speed control

Hyperparameter	**Value**
Input Cost Matrix R	$0.1\mathbb{I}_{2N}$
Output Cost Matrix Q	$\begin{bmatrix} 10^{-5}\mathbb{I}_N & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 10^{-5}\mathbb{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 10^{-4}\mathbb{I}_N \end{bmatrix}$
Output Slack Penalization λ_y	50
Quadratic Regularization Parameter λ_g	1

Table 5.2: DeePC Hyperparameters for Speed Control Integration

The values of the currents i_d and i_q over time for the proposed speed integration architecture is shown in Figure 5.13. The control performance of the currents is not as good when compared to the results shown in Figure 5.8 due to the addition of another controlled variable with more precedence. The value of i_d can be maintained close to 0, by increasing its corresponding weight in the output cost matrix. Of more importance is the evolution of the angular velocity and position with time. The incorporation of speed control has improved the ability of the device to follow the reference trajectories, as expected. This is attributed to the fact that the controller doesn't just rely on the model through the reference trajectory of the current to regulate the speed, but also actively takes actions to compensate any errors in the speed. The complete speed and position trajectories of a nominal scan cycle is shown in 5.14a. A closer look of the controller performance at the IN Stroke and the OUT Stroke of the BWS, the actions in which the wire of the BWS would come in contact with the beam, is shown in Figures 5.14b and 5.14c respectively.

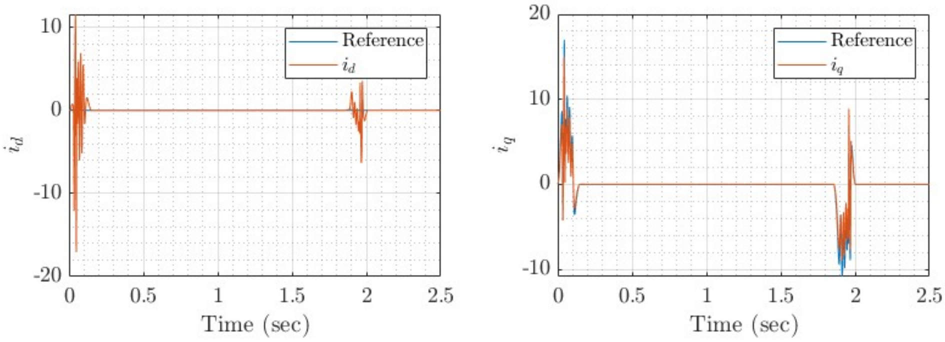
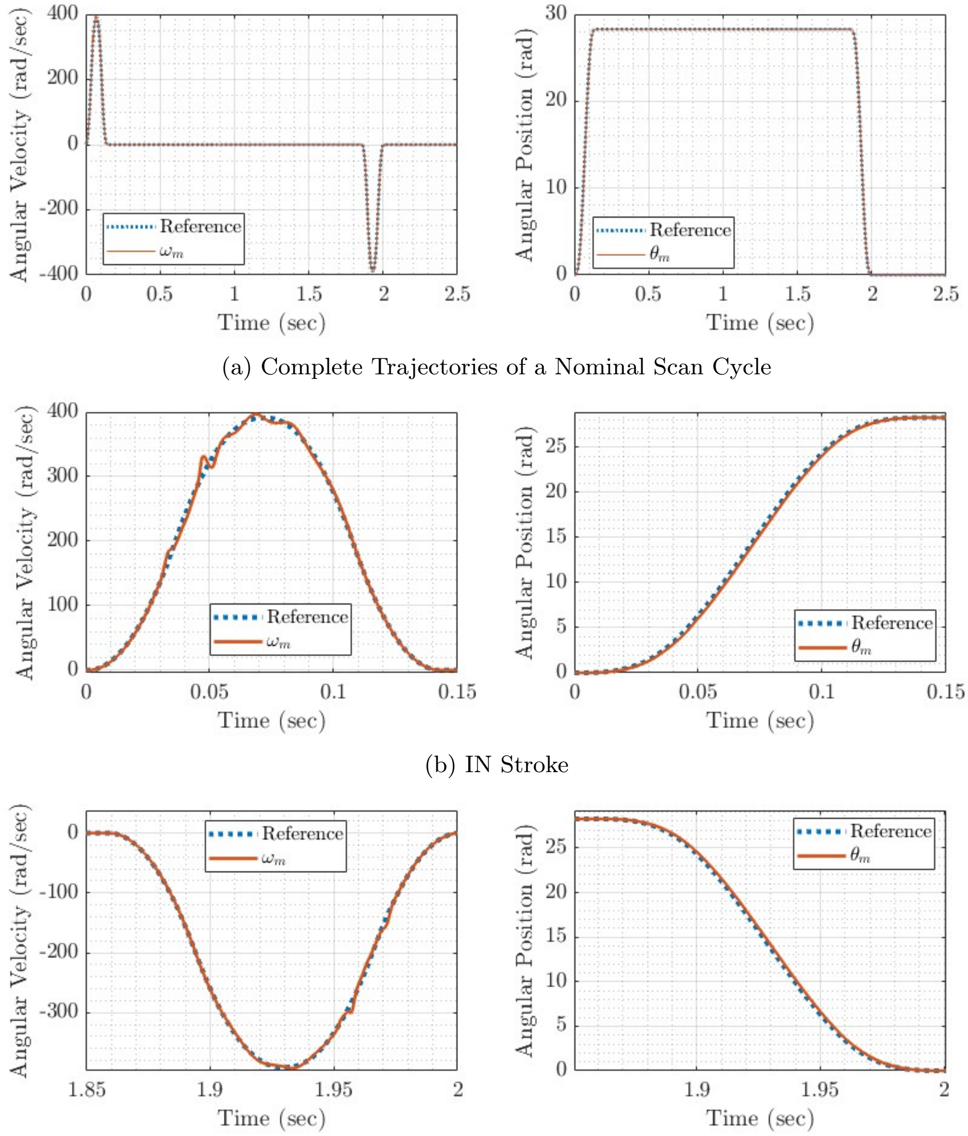


Figure 5.13: Current trajectories upon integration of speed control

The results from the integration of speed control into the incremental DeePC show that this architecture is good enough to track the angular position without any steady-state error or degradation of performance. Since the major goal of the designed controller is to be able to adapt itself to parameter changes, a comparison of the controller performance under nominal conditions is not of major interest. Instead, we compare the performance of the different architectures when subjected to parameter variation in one of the forthcoming subsections.

5.2.5 Integration of Position Control

Although the integration of speed control also provides desired level of accuracy in tracking angular position, it is still important to analyze the behaviour of the system when angular position is also added as a controlled variable. The changes to the offline and online design of the control scheme are analogous to the changes presented in the previous subsection. Hence we delve straight into the chosen values of hyper parameters and the results obtained from the integration of position control listed in Table 5.3.



(a) Complete Trajectories of a Nominal Scan Cycle

(b) IN Stroke

(c) OUT Stroke

Figure 5.14: Angular velocity and position values after integration of speed control

Hyperparameter	**Value**
Input Cost Matrix R	$0.1\mathbb{I}_{2N}$
Output Cost Matrix Q	$\begin{bmatrix} 10^{-2}\mathbb{I}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 8\times10^{-5}\mathbb{I}_N & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 8\times10^{-4}\mathbb{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 8\times10^{-5}\mathbb{I}_N \end{bmatrix}$
Output Slack Penalization λ_y	50
Quadratic Regularization Parameter λ_g	20

Table 5.3: DeePC Hyperparameters for Position Control Integration

The value of i_d stays much closer to 0, due to the increase in its corresponding weight in the cost matrix. The behaviour of the other controlled variables do not change much when compared to the results obtained with just the speed control element and no position control. We use the same hyperparameters from Table 5.3 for the comparison, with just the corresponding output cost for position removed in the control loop without position control. The metric used for comparison is Mean Absolute Error (MAE) which is defined as the average of the absolute deviation of the value from its reference at all time instants.

	Without Position Control	**With Position Control**
Position MAE during IN Stroke	0.1801 rad	0.0225 rad
Position MAE during OUT Stroke	0.1804 rad	0.0290 rad
Speed MAE during IN Stroke	1.2928 rad/sec	1.4944 rad/sec
Speed MAE during OUT Stroke	1.3153 rad/sec	1.4326 rad/sec

Table 5.4: Comparison of control schemes with and without position control

From the values in Table 5.4, we can see that the integration of position control greatly reduced the average error in angular position when the system is in motion. This only slightly decreases the performance of speed control. On the other hand, the memory and the computational complexity of the controller increases due to the addition of one more control variable. Thus it is essential to find a trade-off between control performance and complexity of the controller when deciding to integrate position control into the architecture.

5.2.6 Unified DeePC Algorithm for Application to BWS

Algorithm 3 provides the sequential steps to be followed for the application of DeePC to BWS operation in real-time. This algorithm can be followed to adapt the controller whenever necessary and used in other cases for controlling the motor in nominal scan cycles.
Initially we start off with the collection of available data u_d, u_q, i_d, i_q, ω_m and θ_m. This can be limited based on the chosen architecture of the controller. Next, we construct the Hankel matrices from the collected data and calculate the static gain matrices K'_y, K'_u, and K'_r, from the chosen hyperparameters and cost matrices which make up the controller. The first two steps of the algorithm happen offline, when no scans are required to be done or downtime of the beam. We move on to step 3, where $k = 0$ signifies the beginning of the scan cycle. We calculate the optimal inputs from the feedback of input and output values and the reference trajectory to be followed. From the calculated inputs, we apply only the first value and repeat this process until

Algorithm 3 DeePC Algorithm for BWS Operation

Inputs $T_{ini}, r, Q, R, N, \lambda_g, \lambda_y$

1: Collect data from the PMSM (u^d, y^d) corresponding to the chosen architecture.
2: Construct the matrices K'_y,K_u' and K'_r from the collected data and chosen hyperparameters given by 5.12.
3: **for** k = 0 to T_{scan} **do**
4: Calculate $u^{\star}(k : k+N-1) = K'_y \Delta y_{ini}(k-T_{ini} : k-1) + K'_r(r(k : k+N) - y(k)\mathbb{1}_N) + K'_u \Delta u_{ini}(k - T_{ini} : k - 1)$.
5: Apply the first value of the optimal inputs, $u^{\star}_d(k)$ and $u^{\star}_q(k)$ by selecting the appropriate value from $u^{\star}(k : k + N - 1)$.
6: **end for**
7: **if** Controller performance is subpar or a PMSM parameter changes **then**
8: Go to Step 1 for adapting the controller
9: **else**
10: Go to Step 3 for the next scan cycle
11: **end if**

the end of the scan cycle at $k = T_{scan}$. This forms the receding horizon approach. Once the scan cycle is complete, if the controller performance is not satisfactory or if a known change in any parameter of the PMSM occurs, we go back to the data collection step to adapt the controller to the new conditions.

5.2.7 Adaptability to Parameter Variations

One of the major purposes behind the design of a data-driven control scheme is its ability to combat parameter changes over time. Hence it is important to analyze how the proposed control schemes behave under these parameter variations. We study how good the controller is able to adapt itself when directed, by comparing the controller performances, with respect to changes in motor parameters, when it is allowed to modify and adapt itself.
The motor parameters and the extent of changes it can experience vary based on operating conditions, maintenance practices, and other factors. Changes in some of the most important motor parameters are discussed below:

1. The inertia J represents the motor's resistance to changes in angular velocity. Mechanical wear, aging, or changes in the load dynamics can influence the effective inertia over time. Changes in load characteristics or modifications to the mechanical system connected to the motor may also impact inertia.

2. Changes in resistance R due to temperature variations are typically within a few percentage points. High temperatures can lead to increased resistance, affecting the overall efficiency of the motor.

3. Inductance values L_d and L_q may experience relatively small changes under normal operating conditions.

4. The voltage constant K_E is often considered relatively stable over time. It is primarily influenced by the physical properties of the magnets used in the motor. However, changes can cause high impact to the performance of the motor.

5. Mechanical friction in the motor bearings and other components can change over time due to wear. Lubrication practices and the quality of materials used in bearings play a role in determining the rate of frictional changes.

We choose to study the effects of three parameters that are more prone to changes: the intertia J, the voltage constant K_E and the mechanical friction. Wealready know how J and K_E affect the system equations; However we still need to understand how friction acts on the motor. The friction torque $T_{friction}$ acts as a load on the motor. To calculate the friction torque, a distinction must be made between the static and dynamic cases. In the static case, the friction torque cancels out the sum of the other torques as long as it is less than the static friction constant T_{sf} . In the dynamic case, the friction torque varies as a function of the speed of rotation and opposes the direction of rotation. A detailed representation of the friction torque is given as,

$$T_{\text{friction}} = \begin{cases} T_{sf} & \text{if } \omega_m = 0 \text{ and } T_{\text{sum}} < -T_{sf} \\ -T_{sf} & \text{if } \omega_m = 0 \text{ and } T_{\text{sum}} > T_{sf} \\ -T_{\text{sum}} & \text{if } \omega_m = 0 \text{ and } -T_{sf} < T_{\text{sum}} < T_{sf} \\ -T_{df} - \mu_v \cdot \omega_m & \text{if } \omega_m > 0 \\ T_d f - \mu_v \cdot \omega_m & \text{if } \omega_m < 0 \end{cases}$$

where T_{sum} is the torque generated by the motor without any in the absence of any load, T_{sf} is the initial static friction torque, T_{df} is the initial dynamic friction torque, μ_v is the coefficient of dynamic friction. The friction torque model is discussed in detail in [32]. For the purpose of robustness analysis, we consider a change in the coefficient of dynamic friction μ_v.
The performance of the current controller introduced in subsection 5.2.3 with respect to variation in motor parameters is shown in Figure 5.15. The metric used is Mean Squared Error, which represents the average squared deviation of the value from its reference value. As expected the controller exhibits poor performance in regulation of angular position and speed. The adaptive capabilities of the controller is also poor, with the MSE in both i_d and i_q showing significant increase to variation in the parameters.
The integration of speed controller however, shows significant improvement in results as inferred from Figure 5.16. Not only does it reduce the MSE values for both angular position and speed, but the adaptability of the controller is also quite good as the magnitude of MSE for high and low Change % values does not is not very high when compared to the absence of parameter change.
The integration of a position controller indicates even better performance, which can be seen from From 5.17. Without inhibiting the performance in other control variables, it also enhances the performance of position control. The adaptability of the controller is also quite good as seen by minimal variations in MSE over a wide range of parameter changes.

5.2.8 Computational and Memory Complexity

The online implementation of the proposed architectures are all in the form of a Linear Map defined in subsection 5.2.1. Computationally, the online implementation of the controller only involves multiplication and addition operations, the complexity of which increases linearly with T_{ini}, N and the number of control variables. Similar is also the case for offline calculations of the linear map, which in addition also increases linearly with T_{fin}.
On the memory front, the online implementation requires buffers for each of the control variables and the inputs, capable of storing the past T_{ini} values. Additionally, we also need to store the complete reference trajectory for the control variables, that increases linearly with the sampling

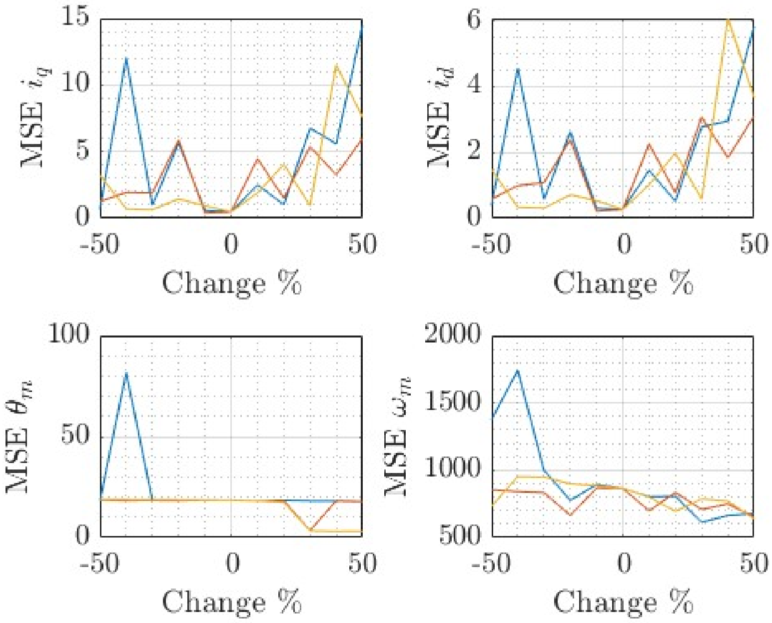


Figure 5.15: Current controller performance with parameter variation: ΔJ(blue), ΔK_E(red) and $\Delta \mu_v$(yellow)

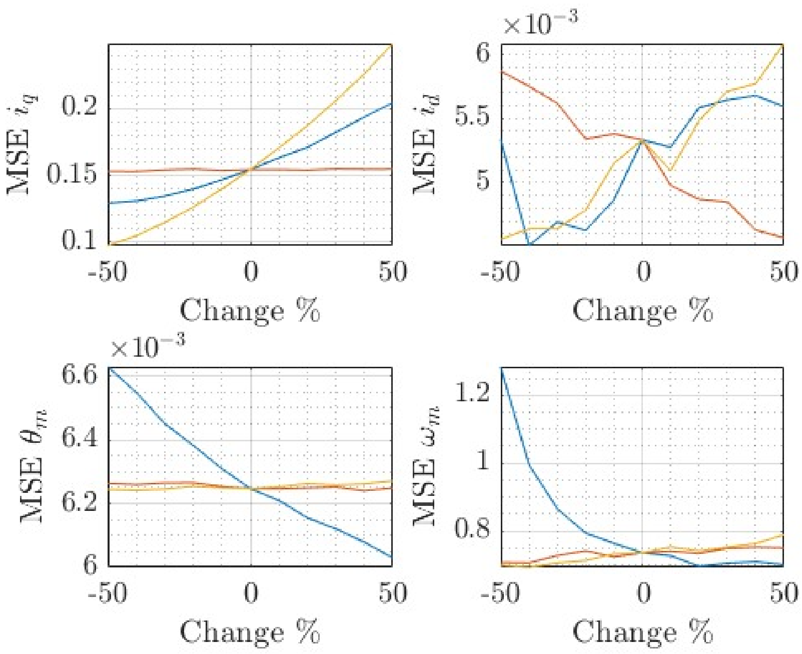


Figure 5.16: Current and speed control architecture performance with parameter variation: ΔJ(blue), ΔK_E(red) and $\Delta \mu_v$(yellow)

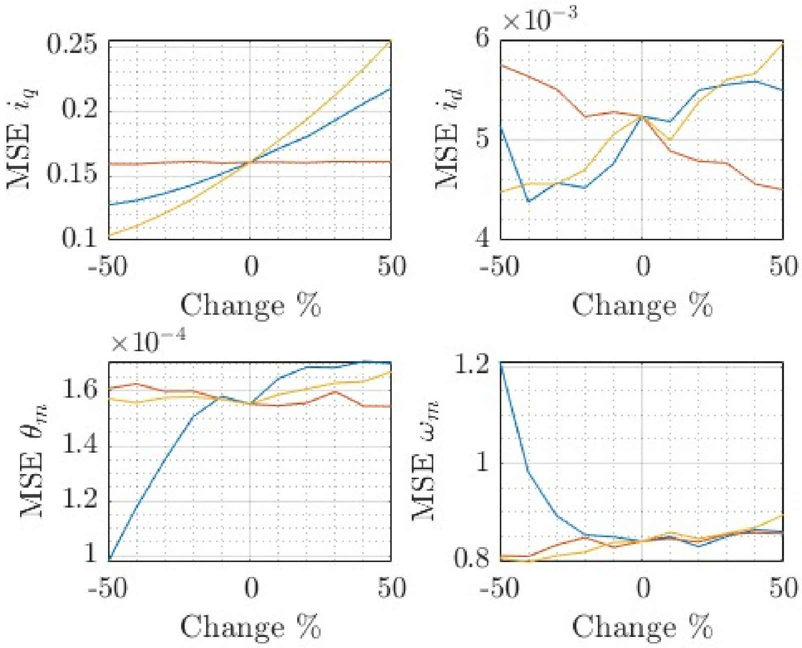


Figure 5.17: Current, speed and position control architecture performance with parameter variation: ΔJ(blue), ΔK_E(red) and $\Delta \mu_v$(yellow)

frequency f_s and scan cycle time T_{scan}. The offline storage is the data collected that again depends linearly on f_s and T_{fin}.

5.3 Subspace Predictive Control Methods

The idea of Subspace Predictive Control was introduced in the previous chapter. Similar to DeePC, in the absence of constraints on the inputs and outputs, the SPC optimization problem described by 4.14, can be reduced to a closed-form solution easier for real-time implementation. The value of the optimal inputs in this case becomes,

$$u^\star(k) = (R + L_u^\top Q L_u)^{-1} L_u^\top Q (L_w \begin{bmatrix} y_{ini} \\ u_{ini} \end{bmatrix} - r) \tag{5.15}$$

Based on our inferences from the implementation of DeePC, the incremental form provides better results, by introducing an ‘integral’ action. Thus, the analogous representation of the above closed from solution in incremental form is given by,

$$u^\star(k) = (R + L_u^\top Q L_u)^{-1} L_u^\top Q (L_w \begin{bmatrix} \Delta y_{ini} \\ \Delta u_{ini} \end{bmatrix} - (r - y(k-1)\mathbb{1}_N^\top)) \tag{5.16}$$

The data collection and the offline design remains the same as DeePC. The online architecture is also designed to be as similar to DeePC as possible except for the controller itself, that differs slightly due to the change in the linear map representation.

Hyperparameter	**Value**
Data Length T_{fin}	0.02 s
Sampling Frequency f_s	16 kHz
Prediction Horizon N	15
Initial Trajectory Length T_{ini}	5
Input Cost Matrix R	$0.08\mathbb{I}_{2N}$
Output Cost Matrix Q	$\begin{bmatrix} 10^{-5}\mathbb{I}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 2\times 10^{-6}\mathbb{I}_N & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 10^{-8}\mathbb{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 10^{-2}\mathbb{I}_N \end{bmatrix}$

Table 5.5: SPC Hyperparameters

The hyperparameters used for SPC simulation are given in Table 5.5. Other than the cost matrices, the other parameters are kept as close to DeePC as possible in order to enable direct comparisons. The results obtained are shown in Figures 5.18 and 5.19.

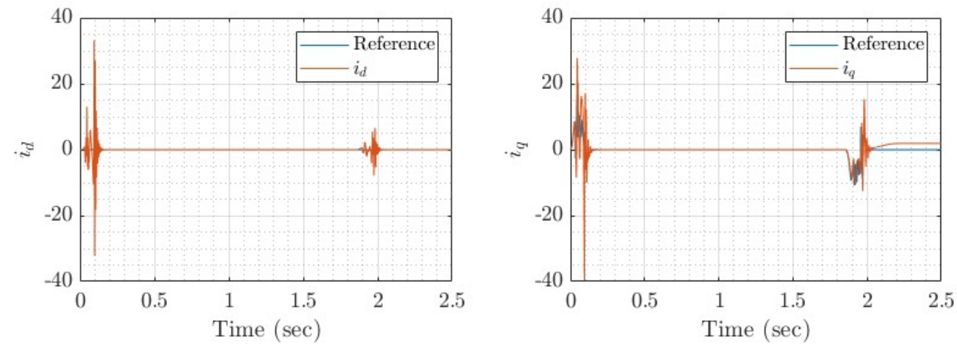


Figure 5.18: Current trajectories of SPC

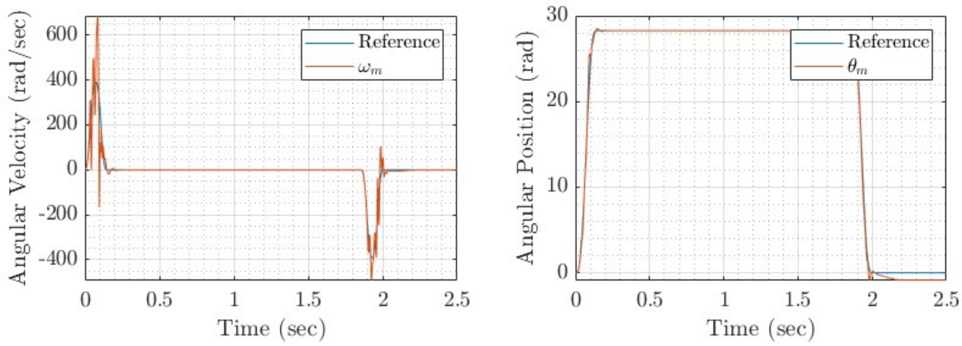


Figure 5.19: Speed and Position Trajectories of SPC

From the results obtained, SPC does not seem to perform as good as DeePC in regulating the different control variables to their respective set point values. This may be attributed to the lack of regularization and thus robustness in the algorithm. With the same prediction horizon

and initial trajectory length, no better cost matrices can be found that perform better than the potrayed results. Thus, we conclude that without increasing the computational complexity of the algorithm, SPC does not perform better than DeePC for the application to BWS.
Having explored different architectures for the application of data-driven control and compared them in simulation we conclude this chapter. At the time of writing this thesis, we are unable to test the proposed control architectures on the test bench. However, this is planned to be done soon in the forseeable future.

Chapter 6

Conclusion and Outlook

This thesis has explored and elucidated the intricate workings of a prototype of the Beam Wire Scanner based on Tubular Linear Magnetic Coupling, as a pivotal instrument in the realm of particle accelerators and beam physics experiments. The innovative design featuring a Tubular-Type Permanent Magnet Linear Magnetic Coupling has successfully been characterized with the application of nonlinear system identification techniques based on neural networks. Experiments and simulations have shown the identified model to perform well in the calculation of different beam sizes. Additionally, the identified resonant frequency of the TLMC has enabled us to generate reference trajectories that would avoid this frequency.

As part of the broader investigation, different optimal reference trajectory generation methods have been analyzed. Promising results show that their application can reduce wire vibration and improve the repeatability and reliability of the BWS.

The evolving mechanical conditions of the PMSM in the BWS over time have been identified as a potential obstacle to sustained controller performance. Recognizing this challenge, the thesis has centered its attention on the implementation and evaluation of novel data-driven drive control strategies. Simulations show that the proposed architecture of DeePC adapts itself quite well to the change in motor parameters and produces exceptional control performance. In a nutshell, the key results of this thesis are:

1. The TLMC does not significantly alter the displacement of the wire in BWS and the developed model of the TLMC can be used to calculate the wire position without affecting the accuracy of beam size considerably.

2. Reference trajectories to the BWS, specifically to the PMSM can be optimized to produce swift, repeatable and vibration-free motion of the wire.

3. Data-driven control methods, especially DeePC can be used to control the PMSM ensuring high performance and adaptability to the operating conditions of the BWS.

Building upon the groundwork laid in this thesis, several promising avenues for future research and development can be considered. Implementation of hardware-in-the-loop (HIL) simulation techniques to create a more realistic testing environment for the control system would be the logical next step. This includes adapting the architecture to run on DSpace and FPGA. This allows for comprehensive testing and validation of the proposed control architectures before deployment in the actual BWS system. Additionally this would also validate the ability of unconstrained DeePC to run on an FPGA and open it up for more real-time applications.

This thesis highlights the application of optimal control solutions for complex and dynamic systems in the ever-evolving field of accelerator science. The culmination of these efforts not only addresses the immediate challenges posed by mechanical degradation but also significantly contributes to the evolving landscape of beam diagnostics and instrumentation. By systematically assessing and implementing various advanced strategies, this research aims to set the stage for enhanced performance, longevity, and reliability of the BWS, thereby advancing the capabilities of particle beam diagnostics in accelerator facilities.

Bibliography

[1] Ana Barjau, Juan Herranz, and Bernd Dehning. Dynamical models of a wire scanner. *Journal of Vibration and Acoustics*, 138(5):051012, 2016.

[2] Pierre-Jean Barre, Richard Bearee, Pierre Borne, and Eric Dumetz. Influence of a jerk controlled movement law on the vibratory behaviour of high-dynamics systems. *Journal of Intelligent and Robotic Systems*, 42:275–293, 2005.

[3] Luigi Biagiotti and Claudio Melchiorri. Optimization of generalized s-curve trajectories for residual vibration suppression and compliance with kinematic bounds. *IEEE/ASME Transactions on Mechatronics*, 26(5):2724–2734, 2021.

[4] Paolo Gherardo Carlet, Andrea Favato, Saverio Bolognani, and Florian Dörfler. Data-driven predictive current control for synchronous motor drives. In *2020 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 5148–5154, 2020.

[5] Paolo Gherardo Carlet, Andrea Favato, Saverio Bolognani, and Florian Dörfler. Data-driven continuous-set predictive current control for synchronous motor drives. *IEEE Transactions on Power Electronics*, 37(6):6637–6646, 2022.

[6] CERN Experiments. `https://home.cern/science/experiments`.

[7] Jeremy Coulson, John Lygeros, and Florian Dörfler. Data-Enabled Predictive Control: In the Shallows of the DeePC. In *2019 18th European Control Conference (ECC)*, pages 307–312, 2019.

[8] S. Di Carlo, J. Emery, W. Andreazza, D. Belohrad, J.C. Esteban Felipe, A. Goldblatt, D. Gudkov, A. Guerrero, S. Jackson, G.O. Lacarrere, M. Martin Nieto, A.T. Rinaldi, F. Roncarolo, C. Schillinger, and R. Veness. Commissioning of the lhc injectors upgrade fast wire scanners and first experimental results. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 1053:168328, 2023.

[9] Jonathan Emery. An identification and controllers synthesis for the cern beam wire scanner. Master's thesis, HES-SO, Lausanne, 2017.

[10] Jonathan Emery, Patrik Andersson, Federico Roncarolo, and Yann Thoma. A low fluctuation control strategy for pmsm direct drive system targeting particle beam instrumentation application. In *2019 IEEE Conference on Control Technology and Applications (CCTA)*, pages 437–443, 2019.

[11] Jonathan Emery, Ana Barjau, Bernd Dehning, Juan Herranz Alvarez, Pierre-Jean Lapray, and Matteo Macchini. Design and validation methodology of the control system for a particle beam size measurement instrument at the cern laboratory. In *2017 American Control Conference (ACC)*, pages 4221–4228, 2017.

[12] Wouter Favoreel, Bart De Moor, and Michel Gevers. Spc: Subspace predictive control. *IFAC Proceedings Volumes*, 32(2):4004–4009, 1999. 14th IFAC World Congress 1999, Beijing, Chia, 5-9 July.

[13] Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, 5(7):1688–1703, 1985.

[14] Werner Herr and B Muratori. Concept of luminosity, 2006.

[15] Juan Herranz Alvarez. *Minimisation of the wire position uncertainties of the new CERN vacuum wire scanner*. PhD thesis, Barcelona, Polytechnic U., 2016. Presented 29 Feb 2016.

[16] Linbin Huang, Jeremy Coulson, John Lygeros, and Florian Dörfler. Decentralized data-enabled predictive control for power system oscillation damping, 2021.

[17] Linbin Huang, Jianzhe Zhen, John Lygeros, and Florian Dörfler. Quadratic regularization of data-enabled predictive control: Theory and application to power converter experiments, 2021.

[18] Jeong-Man Kim, Jang-Young Choi, Min-Mo Koo, Hyeon-Jae Shin, and Sung-Ho Lee. Characteristic analysis of tubular-type permanent-magnet linear magnetic coupling based on analytical magnetic field calculations. *IEEE Transactions on Applied Superconductivity*, 26(4):1–5, 2016.

[19] K.J. Kyriakopoulos and G.N. Saridis. Minimum jerk path generation. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 364–369 vol.1, 1988.

[20] Kostas J Kyriakopoulos and George N Saridis. Minimum jerk for trajectory planning and control. *Robotica*, 12(2):109–113, 1994.

[21] Matteo Macchini. Motion control design of a PMSM and FPGA implementation for the Beam Wire Scanner at CERN. Master's thesis, Pisa U., 2015. Presented 30 Apr 2015.

[22] Ivan Markovsky and Paolo Rapisarda. Data-driven simulation and control. *International Journal of Control*, 81(12):1946–1959, 2008.

[23] Ivan Markovsky, Jan C. Willems, Sabine Van Huffel, and Bart De Moor. *Exact and Approximate Modeling of Linear Systems*. Society for Industrial and Applied Mathematics, 2006.

[24] Esma Mobs. The CERN accelerator complex in 2019. Complexe des accélérateurs du CERN en 2019, 2019. General Photo.

[25] Oliver Nelles and Oliver Nelles. *Nonlinear dynamic system identification*. Springer, 2020.

[26] R. H. Park. Two-reaction theory of synchronous machines generalized method of analysis-part i. *Transactions of the American Institute of Electrical Engineers*, 48(3):716–727, 1929.

[27] Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. Routledge, 2018.

[28] Nguyen Phung Quang and Jörg-Andreas Dittrich. *Machine Models as Prerequisite to Design the Controllers and Observers*, pages 61–112. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[29] N. C. Singer and W. P. Seering. Preshaping Command Inputs to Reduce System Vibration. *Journal of Dynamic Systems, Measurement, and Control*, 112(1):76–82, 03 1990.

[30] Jose Luis Sirvent Blasco. *Beam secondary shower acquisition design for the CERN high accuracy wire scanner*. PhD thesis, Barcelona U., 2018. Presented 12 Dec 2018.

[31] Jan C. Willems, Paolo Rapisarda, Ivan Markovsky, and Bart L.M. de Moor. A note on persistency of excitation. *Systems & Control Letters*, 54:325–329, 2005.

[32] Arnaud Yersin. Modélisation et simulation en boucle fermée des systèmes Beam Wire Scanner du CERN à moteur haute performance basé sur la commande vectorielle par FPGA. Master's thesis, HES-SO, Lausanne, 2023. Presented 14 Mar 2023.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Data-Driven Control Methods for Beam Wire Scanners

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Varadan	Vishnu

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Geneva, 14.12.2023	Vih Vd

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.