CHALMERS
UNIVERSITY OF TECHNOLOGY

Design of a Robust Mean-square Stabilizing Data-Driven Controller

With implementation to a 2 DoF Inverted Pendulum

Master's thesis in Systems, Control and Mechatronics

Hengyue Liang

Department of Signal and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Master's thesis EX089/2017

Design of a Robust Mean-square Stabilizing Data-Driven Controller

With implementation to a 2 DoF Inverted Pendulum

Hengyue Liang

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Signal and Systems
Division of Systems, Control and Mechatronics
Chalmers University of Technology
Gothenburg, Sweden 2017

Design of a Robust Mean-square Stabilizing Data-Driven Controller
with implementation to a 2 DoF Inverted Pendulum
Hengyue Liang

Supervisor: Jianfei Dong, Suzhou Institute of Bio-medical Engineering and Technology, Chinese Academy of Sciences
Examiner: Balázs Adam Kulcsár, Department of Signal and Systems

Master's Thesis EX089/2017
Department of Signal and Systems
Division of Systems, Control and Mechatronics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers University of Technology
Gothenburg, Sweden 2017

Design of a Robust Mean-square Stabilizing Data-Driven Controller
with implementation to a 2 DoF Inverted Pendulum
Hengyue Liang
Department of Signal and Systems
Chalmers University of Technology

Abstract

Data-Driven (DD) controllers are derived purely from measured input-output (I/O) data collected from the target plant. Compared with traditional controller design techniques (such as PID, LQ, MPC), DD controller design bypasses the first principle modelling of the system. Instead, a model in the subspace which is a simple but direct implication of the I/O relation is setup to represent the dynamic behavior of the system.
In this work, a robust mean-square stabilizing data-driven control algorithm is designed for Linear Time Invariant (LTI) systems. For LTI systems, a Vector Auto Regressive model with eXogenous(VARX) input structured subspace model is chosen as it is easy to identify by a simply linear Least Squares (LS) estimate. Different from other subspace DD controller design methods, the proposed algorithm in this work also takes into account the noise corruption in the identification together with the uncertainties brought by the finite window size of the subspace model. The proposed controller guarantees the stability of the closed-loop system in the Root Mean Square (RMS) sense based on a manually pre-defined performance output vector. Furthermore, the performance output vector can be used to tune the closed-loop dynamics.
Finally, the proposed data-driven control algorithm is tested on a real 2 Degree-of-Freedom (DoF) inverted pendulum device to show the reliability. By analyzing the experimental data, the proposed algorithm shows the potential to perform in a Plug-and-Play manner and to provide a control strategy with improved closed-loop dynamics and better energy saving performance in practice.

Keywords: data-driven control, subspace control, root mean square stability, inverted pendulum, plug-and-play

Acknowledgements

First, I would like to thank my examiner, Associate Professor Balázs Adam Kulcsár from department of Signal and System, Chalmers. I have followed his guidance in the master study since one year ago. He has taught me Linear Control System Design course, offered me a summer internship as my first experience of academic research, helped to plan the compulsory course project and finally guided me on this master degree thesis. He is undoubtedly the most important person during my master study in Chalmers. He teaches in a way that he hopes every student understand the knowledge. He is also easy to work with, and most of the time, more like a colleague rather than a teacher, a supervisor or an examiner. Balázs regards himself a researcher and he likes to do researches together with students. It is often the case that he came to the lab, having discussions about the work, offering new ideas and steering me in the right direction during the thesis work. Now my master study is coming to the end, and I can feel the progress I have made. I am thankful to all the time and efforts that Balázs has spent to help me to get such progress.

I would like to thank my supervisor, researcher Jianfei Dong, professor of Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences. Although we are in different time zones, he always replied to the email when I was in trouble and seek for help. His comments are always detailed and clear and his advice is always helpful. These all played important roles to lead this project successful.

Finally I would like to thank my school, Chalmers, who provides me a nice working environment and all the hardware support for the entire thesis work.

Hengyue Liang, Gothenburg, May 2017

Acronyms

I/O	Input and Output
DD	Data-Driven
VARX	Vector Auto Regressive model with eXogenous input
LTI	Linear Time Invariant
LS	Least Squares
RMS	Root Mean Square
DoF	Degree of Freedom
PID	Proportional–Integral–Derivative
LQ	Linear–Quadratic
LQG	Linear–Quadratic-Gaussian
MPC	Model Predictive Control
PBSID	Predictor-Based Subspace IDentification
MP	Markov Parameters
LPV	Linear Parameter Varying
PE	Prediction Error
CCW	Counter-ClockWise
ZOH	Zero-Order-Hold
DARE	Discrete-time Algebraic Riccati Equation
LMI	Linear Matrix Inequality
SDP	Semi-Definite Programming
ODE	Ordinary Differential Equation

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background and motivation

To solve a control problem, it normally starts with setting up a first principle model of the plant. Then a controller is designed to fulfill certain objectives, i.e. aiming at stabilizing the plant to a certain set point, based on the model.

The difficulties and the most time consuming tasks usually lie in the modelling of the plant. A model should be as accurate as possible, since it is the foundation of the following controller design. The model bias from the practical plant results into the different behaviors from the theoretical designed dynamics to the actual closed-loop performance. It is usually not surprising to find that a controller will fail to work if the model is poorly set up.

When it comes to the controller design, tuning the parameters is another challenging task. Considering the classical PID controllers, the parameters P, I and D are usually set by carrying out many try-and-error experiments to meet certain requirements [1]. For other controller design techniques such as LQG or MPC, the tuning usually involves one more state estimator in addition to the controller itself. Such tuning tasks are non-trivial as the controller parameters strongly affect the performance and robustness. Moreover, these tuning parameters usually have influence on each other and make the tuning process sometimes difficult. The researches of self-tuned LQG and MPC controller reveal such difficulty [2] [3] in some sense.

For most laboratory sessions in the controller design courses at school, the first principle models are usually given by mathematical equations, with pre-defined system parameters from the lab manual. Students thus mostly work on tuning the controllers, usually through simply try-and-error testing. Furthermore, it is not always successful to find a controller with a satisfactory behavior with the modelling knowledge from the manual. Thus a question is raised — Is there a way to achieve better controllers for practical plants using the data collected from such unsatisfactory initial controllers with an easy process?

The preliminary intention of this master thesis is trying to find a control algorithm to answer the above question and can solve practical problems. The derivation of the controller itself should be general. The tuning of the closed-loop dynamics should be easy and convenient. To validate the design algorithm, the obtained controller will be tested via a practical experiment, where a 2 DoF inverted pendulum lab device will be controlled. Such platform is chosen because inverted pendulum systems are commonly used for testing control algorithms for their open-loop unstable property and fast dynamics. In addition, such systems also challenge the robustness of the

designed controller as they are also highly nonlinear.

1.2 Introduction

As is mentioned in Section 1.1, the accuracy of modelling plays an important role in the controller design. This inspires the idea that an new identification can be carried out when an initial controller is running to improve the system modelling. Based on the improved model, a second controller may be derived with better behavior.
Typically there are two kinds of system identification techniques: gray box [4] and black box [5] identification. Gray box identification is the method combining a partial mathematical structure with parameters needed to decide from a collection of I/O data while a black box is the method without any prior structure.
Indeed, most of the system identification methods existed are of black box identification form. On the other hand, in order to keep the generality for the proposed controller design algorithm in this thesis work, black box identification structure is actually considered. Predictor-Based Subspace IDentification (PBSID) [6] is a well developed black box identification method for LTI state-space models. PBSID identifies the state-space model via the existence condition of a stable observer, which results in a intermediate VARX structured subspace relation of the I/O data (also called Markov Parameters) and then formulate the state-space model. One advantage for PBSID is that it can be used to consistently estimate a state-space model without any prior knowledge of the controller, thus it is able to be used in either open-loop or closed-loop identifications. Fig. 1.1 illustrates the work-flow of the PBSID method.

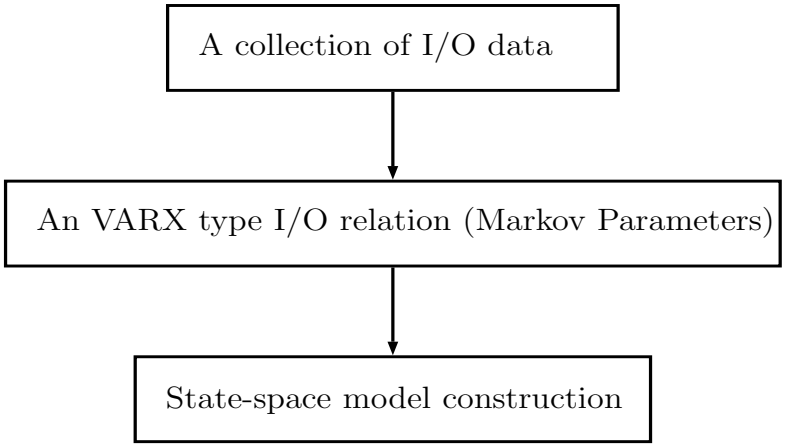

Figure 1.1: Flowchart of the PBSID method to identify state-space parameters

The first principle model in the state-space form that PBSID tries to identify takes the following form (in discrete time):

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) + D_d u(k) \end{aligned} \tag{1.1}$$

where x is the state vector, u is the input vector, y is the measurement vector, A_d, B_d, C_d and D_d are matrices with appropriate dimensions.

First of all, the system order (the number of elements in x) is a manually tuning parameter in PBSID method. Second, it is not surprising that large amount of data are required to identify all of the parameters in matrices A_d, B_d, C_d and D_d. On the other hand, the identified parameters are not always accurate[6]. Thus the controller design based on Eq. (1.1) will be restricted by the accuracy of the identified parameters. Intuitively, as PBSID goes through an intermediate step in the subspace representation before it comes to identify the state-space model parameters, there is one more possibility to introduce errors for the identification result. In another word, if a new controller can be successfully designed based on the intermediate subspace representation directly, it is likely to achieve better closed-loop performance with fewer identification errors. The different processes of the first principle model based controller design and the subspace based controller design are shown in Fig. 1.2, where the first principle modeling is bypassed for the subspace controller design.

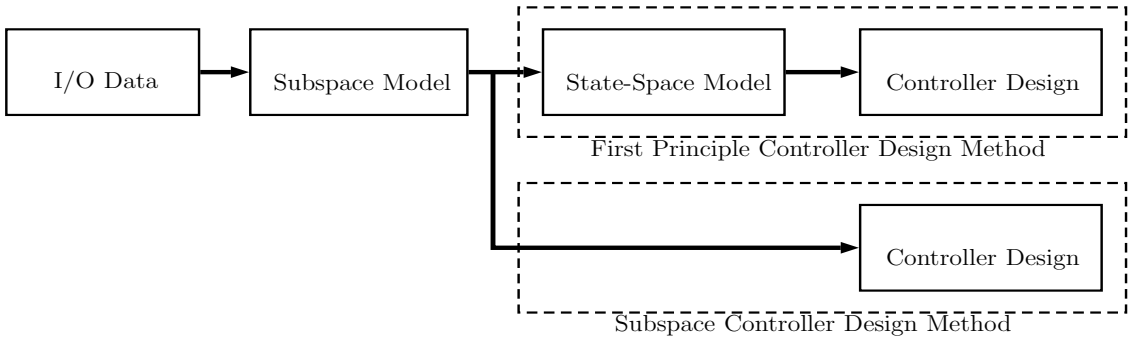

Figure 1.2: Different controller design processes of first principle model based and subspace based

However, the subspace representation will still have uncertainties from the identification process, i.e. noise corruption and model bias. Thus the problem which will be studied in this thesis is defined as:

Problem 1 *Given an I/O level dynamical model of the system in the subspace, design an control algorithm directly based on such model and such controller is able to meet stability and robustness requirements with respect to bias or disturbances.*

Current researches have studied various ways of subspace controller design techniques. [9] and [10] have proposed predictive controllers in the subspace for LTI systems based on open-loop and closed-loop identification respectively. [11] summarizes that both subspace predictive controllers are asymptotically equivalent to LQG controllers when the length of MP and the prediction horizon to be infinitely long. [12] and [13] have proposed subspace identification method for Linear Parameter Varying(LPV) systems to seek identification solutions to non-linear systems. In [14], a predictive controller in the subspace for LPV systems is given with an practical example to control a non-linear DC motor.
However, there are several aspects that we may notice. First, these approaches preassume that the past window size to construct the MP is long enough, so that the

bias caused from the finite length of MP is negligible. Otherwise, significant tracking error may occur. Second, the computation of the predictive control input involves an inversion of the Hankel matrix constructed from the identified MP, which indicates the past horizon cannot be too long in case of a poor condition of the rank.
To overcome the ambiguous choice of the past horizon and to reach a Plug-and-Play manner subspace controller, an LTI $\mathcal{H}_\infty$ style data-drive controller in the subspace is proposed in this paper. The model bias between the subspace and the true system, together with the bias term and disturbances, are designed to be rejected based on a pre-defined performance vector, induced in a Root Mean Square(RMS) level. The noise corruption on the identification of the MP is also considered and is claimed to have a mean sense stability based on the proposed controller design
The proposed controller design algorithm results in an online structure of static state feedback controller, which can be easily applied to systems with fast dynamics without any computational burden. The static feedback gain can be computed offline with a collection of excitation data from the initial controller. The proposed algorithm is proved capable to design a well functioned controller for the 2 DoF inverted Pendulum device with a performance improvement from the initial controller through the practical experiments. Few parameters need to tune to adjust the closed-loop behavior of the proposed controller. This also shows the potential of such controller to act in the Plug-and-Play manner.

The following contents of this thesis report are organized as:
In Chapter 2, the theoretic derivation of the robust mean-square stabilizing data-driven controller is presented. Such controller design method is wrapped as an algorithm and the output is a static feedback control strategy in the subspace. Chapter 3 introduces the experimental platform where the proposed data-driven controller will be implemented and tested. In addition, the steps to implement the proposed controller is briefly described. Chapter 4 illustrates the implementation result of the proposed controller .

2

Robust Mean-square Stabilizing Data-Driven Controller

This chapter introduces in detail the theoretical formulation of the subspace mean-square stabilizing data-driven controller. The proposed subspace controller takes a simple form of a static feedback controller. Such controller design approach is given as an algorithm, taking the past window size p, a collection of $N + p$ I/O data and a manually defined performance vector as input and a static feedback gain in the subspace as the output. The algorithm is formulated as an optimization problem with Linear Matrix Inequality (LMI) conditions which can be efficiently solved through existing Semi-Definite Programming (SDP) solvers.

The structure of the subspace model used in the controller design is similar to a state-space model. To clarify the notation, in what follows of this chapter, system matrices for state-space models will be denoted in capital block letters (i.e. A) while matrices related to subspace models will be put in capital calligraphic letters (i.e. $\mathcal{A}$).

This chapter is organized as follows: Section 2.1 describes the equivalence of the LTI subspace MP and state-space model in representing the system dynamics, followed by the identification approach of the MP in the subspace. Section 2.2 illustrates the uncertainty components in the identified MP. The concept of the proposed data-driven controller design will be shown in section 2.3, starting with the formulation of a dynamical state-space structured subspace model, then followed by an introduction of the controller structure. At the end, the controller design approach aiming at dealing with the uncertainty component is wrapped up as solving an LMI optimization problem.

2.1 System dynamics from the LTI state-space model into I/O level subspace

This section is divided into two parts: Section 2.1.1 explains the equivalence in representing system dynamics of an LTI state-space model and a subspace MP representation. The identification of the MP can be done through a simple Least Square(LS) estimate as is shown in Section 2.1.2.

2.1.1 LTI subspace representation of system dynamics

The validity of an LTI subspace representation lies in the existence of a stable observer of an LTI discrete time state-space model. Consider the following innovation form discrete time state-space model:

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + Le(k) \tag{2.1}$$

$$y(k) = C\hat{x}(k) + e(k) \tag{2.2}$$

where k denotes the time instant; $e(k)$ is the innovation signal, assumed to be a zero-mean, white noise sequence with a non-singular covariance matrix $\Sigma_e = \mathcal{E}\mathcal{E}^T$. The dimensions of the estimated states, measurement and inputs are denoted as $\hat{x}(k) \in \mathbf{R}^{n_x}$, $y(k) \in \mathbf{R}^{n_y}$ and $u(k) \in \mathbf{R}^{n_u}$ respectively. Furthermore, an assumption of the stable observer is made:

Assumption 1 *The system written in Eq.* (2.1) *and* (2.2) *is of minimal order and its closed-loop observer* $\Phi \triangleq A - LC$ *is stable.*

Assumption 1 is non-restrictive for a stable observer Φ, as a discrete time delayed version Kalman Filter is one of the candidate methods to derive the observer gain. The assumption of observability does not introduce limitations to practical problems either as the system should be observable (on the states of interest) to have feasible solutions to the control problems.
Denote a look-back window size p as the past horizon. By letting $e(k) = y(k) - C\hat{x}(k)$ and rewriting $\hat{x}(k)$ in Eq. (2.2) with Eq. (2.1) for p times, we will get the following I/O representation of the system dynamics:

$$y(k) = \Xi z(k) + \underbrace{\overbrace{C\Phi^p\hat{x}(k-p)}^{b(k)} + e(k)}_{d(k)} \tag{2.3}$$

where $\Xi \triangleq \begin{bmatrix} CB & CL & \cdots & C\Phi^{p-1}B & C\Phi^{p-1}L \end{bmatrix}$ is the Markov Parameters (MP); $z(k) \triangleq \begin{bmatrix} u^T(k-1) & y^T(k-1) & \cdots & u^T(k-p) & y^T(k-p) \end{bmatrix}^T$ is a history of p input and output data; $b(k)$ denotes the bias term caused by the finite window size p from the "initial" condition $\hat{x}(k-p)$ outside the window; $d(k)$ is used to denote the total disturbance.
Here we can see that the LTI subspace representation of tje system dynamics in Eq. (2.3) is equivalent to Eq. (2.1) and (2.2).

2.1.2 LTI subspace identification

If we collect in total $N+p$ input and output data from an experiment of the nominal system, the following LS problem can be formulated according to Eq. (2.3):

$$Y_{k,N} = \Xi Z_{k,N} + C\Phi^p X_{t-p,N} + E_{k,N} \tag{2.4}$$

where

$$Z_{k,N} = \begin{bmatrix} z(k+N-1) & z(k+N-2) & \cdots & z(k) \end{bmatrix},$$

$$Y_{k,N} = \begin{bmatrix} y(k+N-1) & \cdots & y(k) \end{bmatrix}, \; E_{k,N} = \begin{bmatrix} e(k+N-1) & \cdots & e(k) \end{bmatrix}$$

are the history I/O data, measurement and innovation sequence respectively,

$$X_{t-p,N} = \begin{bmatrix} \hat{x}(k-p+N-1) & \cdots & \hat{x}(k-p) \end{bmatrix}$$

is a sequence of the initial states outside the look-back window p at each corresponding time instant.
With the help of Eq. (2.4), an estimate of Ξ can be easily obtained by a LS solution:

$$\hat{\Xi} \triangleq Y_{k,N} \cdot Z_{k,N}^{\dagger} \tag{2.5}$$

where † is the symbol representing the "pseudo-inverse".

Remark: From section 2.1 we can see that the subspace representation is equivalent to the discrete time state-space model in the sense of modeling the system dynamics. The formulation of a subspace model only depends on an implicit, stable observer, whose significance lies in the fact that the estimate given by the LS solution is valid for both open-loop and closed-loop identifications without any prior knowledge of the initial controller used in the identification step. Neither does it matter if the system is initially open-loop stable or not.

2.2 Markov Parameters with uncertainty component

When $N \to \infty$, the covariance of $\hat{\Xi}$ provided by the LS estimate in Eq. (2.5) tends to be zero . However, as the true Ξ is infinitely long, there is always a biased term $\hat{b}(k)$ for the estimate result $\hat{\Xi}$ in Eq. (2.3) caused by the finite window size p in practice. Furthermore, as N cannot be infinite in practice, $\hat{\Xi}$ will be inevitably noise corrupted.
Now, we would like to study such estimated error. Introducing a column vectorization operator $vec(\cdot)$, we denote the vectorized MP and its estimate to be $\Theta \triangleq vec(\Xi) \in \mathbf{R}^{n_y(n_y+n_u)p}$ and $\hat{\Theta} \triangleq vec(\hat{\Xi}) \in \mathbf{R}^{n_y(n_y+n_u)p}$ respectively. The estimate error for the MP can be expressed as:

$$\Theta - \hat{\Theta} = \delta\Theta + \Sigma_v^{1/2} v \tag{2.6}$$

$\Sigma_v^{1/2} v$ denotes the noise corruption in the estimate error, where $\Sigma_v = [Z_{k,N} Z_{k,N}]^{-1} \otimes (\Sigma_e)$, $v \in \mathbf{R}^{n_v}$ with $n_v = n_u p(n_y + n_u)$ is a zero mean random signal with identity covariance matrix. $\delta\Theta$ denotes the bias component brought by the finite window size p.

Remark: The bias term $\delta\Theta$ is related to $C\Phi^p$. As Φ is stable, the bias term can be negligible when $p \to \infty$. However, there are reasons that we would prefer a relatively short p. First, to obtain a good LS estimate, it is required that $N >> p$ and in

practice N is sometimes limited. Second, a long p will also result in low variance for each component in the Markov Parameters which will be crucial for fault estimation in subspace[23] to inverse the MP matrix. Third, the longer p, the longer time to initialize z_k for any subspace controller where unwilling transient behavior such as large overshot may occur before the data-driven controller takes its responsibility as is designed.

2.3 Robust mean-square stabilizing controller synthesis

In this section, we would like to derive a subspace controller that is functional under a relatively small p and limited size of identification data N. Therefore, both the estimated bias and the noise corruption in the MP need to be considered. Furthermore, we wrap up the controller design approach as solving an optimization problem with only a few tuning parameters. These parameters can be intuitively adjusted according to the closed-loop dynamics in order to be conveniently applied by the users.

In 2.3.1, we first build an I/O level subspace model to represent the system dynamics in a state-space-like structure. With the proposed controller structure shown in 2.3.2, 2.3.3 and 2.3.4 will show in details the stabilizing design over the uncertainty components and disturbances. Finally, a robust mean-square stabilizing controller is derived in 2.3.5.

2.3.1 Dynamic I/O model

Given the past window size p, we introduce a new state vector:

$$\chi(k) = \begin{bmatrix} y^T(k) & u^T(k-1) & y^T(k-1) & \cdots & u^T(k-p+1) & y^T(k-p+1) \end{bmatrix}^T \quad (2.7)$$

where $\chi(k) \in \mathbf{R}^{n_\chi}$ with dimension $n_\chi = n_u(p-1) + n_y p$. With Eq. (2.7), an state-space structured subspace dynamic model can be built:

$$\underbrace{\begin{bmatrix} y(k+1) \\ \hline u(k) \\ y(k) \\ u(k-1) \\ \vdots \\ u(k-p+2) \\ y(k-p+2) \end{bmatrix}}_{\chi(k+1)} = \underbrace{\left[\begin{array}{ccccc} CL & C\Phi B & C\Phi L & \cdots & C\Phi^{p-1}L \\ \hline 0_{n_u n_y} & 0_{n_u n_u} & \cdots & \cdots & 0_{n_u n_y} \\ I_{n_y} & 0_{n_y n_u} & \cdots & \cdots & 0_{n_y n_y} \\ 0_{n_u n_y} & I_{n_u} & \cdots & \cdots & 0_{n_u n_y} \\ & & \ddots & & \\ 0_{n_u n_y} & \cdots & \cdots & \cdots & 0_{n_u n_y} \\ 0_{n_y} & \cdots & I_{n_y} & 0_{n_y n_u} & 0_{n_y n_y} \end{array}\right]}_{\mathcal{A}} \underbrace{\begin{bmatrix} y(k) \\ \hline u(k-1) \\ y(k-1) \\ u(k-2) \\ \vdots \\ u(k-p+1) \\ y(k-p+1) \end{bmatrix}}_{\chi(k)}$$

$$+ \underbrace{\begin{bmatrix} CB \\ \hline I_{n_u} \\ 0_{n_y n_u} \\ \\ \vdots \\ \\ 0_{n_y n_u} \end{bmatrix}}_{\mathcal{B}_2} u(k) + \underbrace{\begin{bmatrix} I_{n_y} \\ \hline 0_{n_u n_u} \\ 0_{n_y n_y} \\ \\ \vdots \\ \\ 0_{n_y n_u} \end{bmatrix}}_{\mathcal{B}_1} d(k) \tag{2.8}$$

where $I_\star$ and $0_{\star\star}$ with $\star$ for "n_u, n_y" represent identity matrix and zero block matrix with corresponding rows and columns. $d(k) \in \mathbf{R}^{n_y}$ models the total disturbance including noise component, bias term, etc. $d(k)$ is assumed to have bounded covariance as the only restriction. $\mathcal{A} \in \mathbf{R}^{n_\chi \times n_\chi}$, $\mathcal{B}_1 \in \mathbf{R}^{n_\chi \times n_y}$ and $\mathcal{B}_2 \in \mathbf{R}^{n_\chi \times n_u}$ are the subspace model dynamic matrices.

An estimate of Ξ is given as $\hat{\Xi}$ by solving Eq .(2.5). Thus we can replace the corresponding components in $\mathcal{A}$ and $\mathcal{B}_2$ from Ξ by the ones obtained from $\hat{\Xi}$. Denote the corresponding estimated system dynamic matrices as $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}_2$, which will be later used for the future controller design. From Eq. (2.6), the noise corrupted uncertainty on $\hat{\Theta}$ is additive and can be estimated by $\Sigma_v^{1/2} v$. Thus such an stochastic uncertainty can be abstracted from the disturbance $d(k)$ into a component wise uncertainty with respect to the MP in $\hat{\mathcal{A}}$, taking the form $\sum_{i=1}^{n_v} \hat{\mathcal{A}}_i v_i$ and $\sum_{i=1}^{n_v} \hat{\mathcal{B}}_{2i} v_i$. $\hat{\mathcal{A}}_i$ can be obtained from the i^{th} column of the covariance matrix $\Sigma_v^{1/2}$ with the components with respect to $[\widehat{CL}\ \widehat{C\Phi B}\ \cdots\ \widehat{C\Phi^{p-1}L}]$ terms in $\hat{\Theta}$; Similarly, $\hat{\mathcal{B}}_{2i}$ can also be obtained from the i^{th} column of the covariance matrix $\Sigma_v^{1/2}$ with the components with respect to $[\widehat{CB}]$ term in $\hat{\Theta}$; v_i is the i^{th} entry from the random vector v. As v has identity covariance matrix, v_i are independent from each other for $i \in \{1, 2, \cdots, n_v\}$.

Therefore, we can re-organize the dynamics in the identification stage in Eq. (2.8) into the following uncertain state equation:

$$\chi(k+1) = \Big(\hat{\mathcal{A}} + \sum_{i=1}^{n_v} \hat{\mathcal{A}}_i v_i\Big)\chi(k) + \Big(\hat{\mathcal{B}}_2 + \sum_{i=1}^{n_v} \hat{\mathcal{B}}_{2i} v_i\Big)u(k) + \mathcal{B}_1 \hat{b}(k) \tag{2.9}$$

where $\hat{b}(k)$ is the deterministic bias component of $d(k)$. Here the term $\Big(\hat{\mathcal{A}} +$

$\sum_{i=1}^{n_v} \hat{\mathcal{A}}_i v_k\Big)$ and $\Big(\hat{\mathcal{B}}_2 + \sum_{i=1}^{n_v} \hat{\mathcal{B}}_{2i} v_k\Big)$ try to recover the true MP blocks $\mathcal{A}$ and $\mathcal{B}_2$ in Eq. (2.8) from the identification data.

Finally, we manually define a performance vector as the output of the subspace system to complete the model:

$$\zeta(k) = \mathcal{C}\chi(k) + \mathcal{D}_2 u(k) + \mathcal{D}_1 d(k) \tag{2.10}$$

where $\zeta(k) \in \mathbf{R}^{n_\zeta}$ denotes the user-defined performance output vector; $\mathcal{C} \in \mathbf{R}^{n_\zeta \times n_\chi}$ and $\mathcal{D}_{1,2} \in \mathbf{R}^{n_\zeta \times n_u}$ are real coefficient matrices. When $n_\zeta = n_y$ with $\mathcal{C} = [I_{n_y}\ 0_{(n_\chi - n_y)n_\chi}]$ and $\mathcal{D}_{1,2} = 0_{n_\zeta n_u}$, $\zeta(k)$ is simply the measurement output as $y(k)$ in Eq (2.2).

2.3.2 Controller structure

First, we would like to make to following assumption:

Assumption 2 *The open loop state equation defiend in Eq.* (2.9) *is stabilizable in mean-square sense with respect to the stochastic component in MP by a χ-state feedback with static gain $\kappa \in \mathbf{R}^{n_u \times n_\chi}$.*

Assumption 2 is a statement of the structure for the proposed controller which is shown in Fig. 2.1. For simplicity of the controller design, we here consider the zero stability without reference signal $r(k)$ and the performance vector (output of the subspace model) does not include direct feed-through from $d(k)$ and $u(k)$.

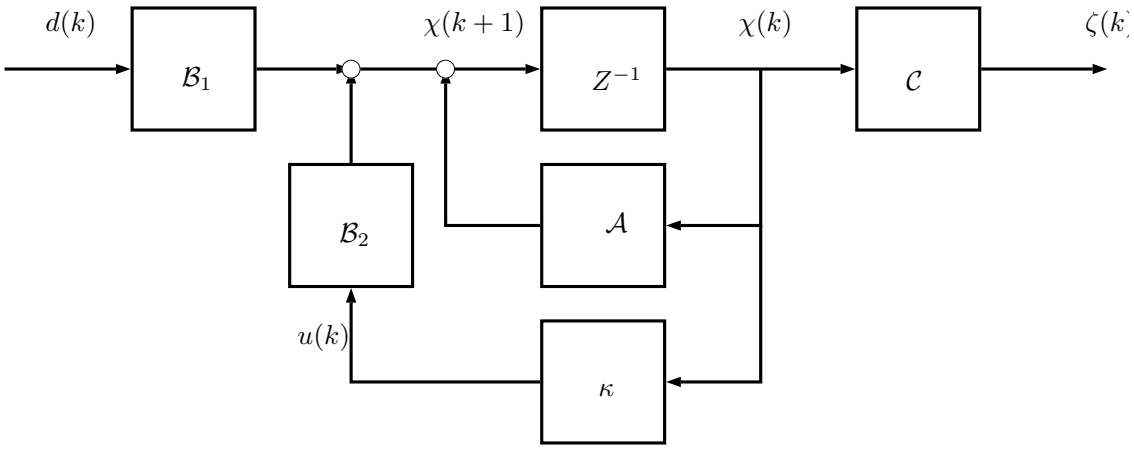

Figure 2.1: Flow chart of the proposed controller

For comparison, Fig. 2.2 illustrates a typical discrete time state-space feedback system with static feedback gain K:

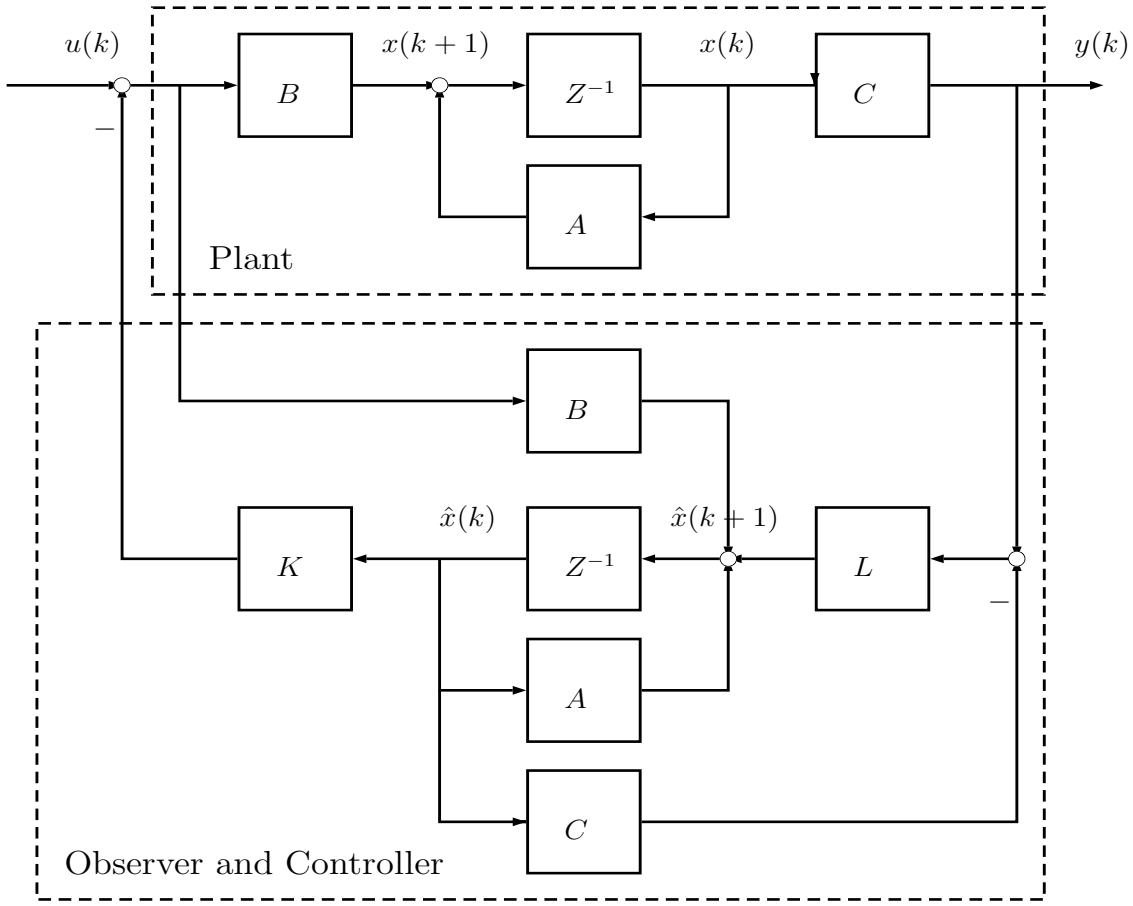

Figure 2.2: Flow chart of a typical state-space feedback controller

As is shown in Fig. 2.2, the online part of a state-space static feedback controller consists of a state estimator and a static feedback gain K. In Fig. 2.1, the proposed subspace controller with static feedback gain κ in this project works in a similar way as the state-space controller without the state estimator. With the subspace formulation, no state estimate will take place and the control input at time k will be computed simply by $u(k) = \kappa\chi(k)$, where $\chi(k)$ is defined in (2.7).
On the other hand, if a static feedback system in the original state-space is feasible (i.e. there exists an stabilizing LQ controller), Assumption 2 will not be restrictive as well. Since closed-loop identifications usually take place with the help of additive stochastic excitation signal[20], a controllable plant in the original state-space should also be controllable in the subspace as they have equivalent dynamic representations. Now it is time to define the system model that we use to design the proposed controller:

$$\begin{aligned}\chi(k+1) &= \Big(\hat{\mathcal{A}} + \sum_{i=1}^{n_v} \hat{\mathcal{A}}_i v_i\Big)\chi(k) + \Big(\hat{\mathcal{B}}_2 + \sum_{i=1}^{n_v} \hat{\mathcal{B}}_{2i} v_i\Big)u(k) + \mathcal{B}_1 d(k) \\ \zeta(k) &= \mathcal{C}\chi(k) + \mathcal{D}_2 u(k) + \mathcal{D}_1 d(k)\end{aligned} \tag{2.11}$$

Remark: We propose to use an offline, batch mode LS identification of the MP. With the batch mode identification, $d(k)$ in Eq. (2.11) is considered to be re-defined

as the future disturbance when the proposed controller is implemented, which is now different from the one that has been used in Eq. (2.8). Therefore, in the controller design problem, $\chi(k)$ and $d(k)$ only contains the data during the online implementation of the proposed controller. The random vector v, determined by the noise in the identification experiment, is thus independent from $\chi(k)$ and $d(k)$.

2.3.3 Design over the disturbance

One of the objectives of the static feedback gain κ is to deal with the deterministic bias of the system, together with unwanted noise and disturbances. To do so, we would like to introduce the following Root Mean Square (RMS) disturbance attenuation criterion:

$$\sup_{||d||_{RMS}\neq 0} \frac{||\zeta'||_{RMS}}{||d'||_{RMS}} = ||S^{\frac{1}{2}} G_{d\zeta} S^{-\frac{1}{2}}||_\infty \leq \gamma \tag{2.12}$$

where $\zeta' \triangleq S^{1/2}\zeta$ and $d \triangleq S^{-1/2}d'$. $||f||_{RMS}$ is defined as the Root Mean Square value of vector f defined as:

$$||f||_{RMS} \triangleq \Big(\mathbf{E}(f_k^T f_k)\Big)^{\frac{1}{2}} = \Big(\lim_{T\to\infty} \frac{1}{T}\sum_{k=0}^{T} f_k^T f_k\Big)^{\frac{1}{2}} < \infty \tag{2.13}$$

In Eq. (2.12), $S \succ 0 \in \mathbf{R}^{n_\zeta \times n_\zeta}$ is a positive diagonal scaling matrix; $\gamma > 0$ is the attenuation level. $G_{d\zeta}$ denotes the transfer function from d to ζ.
[22] has shown that the induced RMS criterion coincides with the component wise $\mathcal{H}_\infty$ norm according to matrix S. This indicates a successful design over κ from Eq. (2.12) will lead to a γ disturbance attenuation weighted $\mathcal{H}_\infty$ controller of the plant.

2.3.4 Design over the uncertain Markov Parameters

The other objective for the proposed controller is to deal with the stochastic uncertainty from the identified Markov Parameters. We consider to ensure the closed-loop stability using Lyapunov criteria. Define the following stochastic Lyapunov function:

$$V(k) = \chi(k)^T P \chi(k) \tag{2.14}$$

with $P = P^T \succ 0$ is a symmetrical positive definite matrix.
The following inequality of the performance objective function should hold to ensure an averaged value dissipation condition in the designed closed-loop system :

$$\begin{aligned} &\mathbf{E}_v\{V(k+1) - V(k)\} \leq \\ &\quad - \mathbf{E}_v\{\zeta'(k)^T\zeta'(k)\} + \gamma^2 \mathbf{E}_v\{d'(k)^T d'(k)\} \end{aligned} \tag{2.15}$$

Substituting Eq. (2.11), $\zeta' \triangleq S^{1/2}\zeta$ and $d \triangleq S^{-1/2}d'$ into Eq. (2.15), the following matrix inequality form will be derived:

$$\mathbf{E}_v\Bigg\{ \begin{bmatrix}\chi(k) & d(k)\end{bmatrix} [\mathcal{L}(P,v)] \begin{bmatrix}\chi(k)\\ d(k)\end{bmatrix} \Bigg\} \leq 0 \tag{2.16}$$

where

$$\mathcal{L}(P,v)=\begin{bmatrix}\tilde{\mathcal{A}}_1^T P\tilde{\mathcal{A}}_1 - P + \tilde{\mathcal{C}}_1^T S\tilde{\mathcal{C}}_1 & \tilde{\mathcal{A}}_1^T P\tilde{\mathcal{B}}_1 + \tilde{\mathcal{C}}_1^T S\mathcal{D}_1 \\ \tilde{\mathcal{B}}_1^T P\tilde{\mathcal{A}}_1 + \mathcal{D}_1^T S\tilde{\mathcal{C}}_1 & \tilde{\mathcal{B}}_1^T P\tilde{\mathcal{B}}_1 - \mathcal{D}_1^T S\mathcal{D}_1 - \gamma^2 S\end{bmatrix} \tag{2.17}$$

with

$$\begin{aligned}\tilde{\mathcal{A}}_1 &= \hat{\mathcal{A}} + \hat{\mathcal{B}}_2\kappa + \sum_{i=1}^{n_v}(\hat{\mathcal{A}}_i + \hat{\mathcal{B}}_{2i}\kappa)v_i \\ \tilde{\mathcal{B}}_1 &= \mathcal{B}_1 \\ \tilde{\mathcal{C}}_1 &= \mathcal{C} + \mathcal{D}_2\kappa\end{aligned}$$

As v is independent from $\chi(k)$ and $d(k)$ (recalling Eq. (2.11)), we only need to apply the expectation operator to the middle term $\mathcal{L}(P,v)$. If $\mathbf{E}_v\{\mathcal{L}(P,v)\}$ is negative semi-definite, the averaged value dissipation condition in Eq. (2.16) will be satisfied. By taking the expected value, we will have the following matrix inequality condition:

$$\begin{aligned}\mathcal{L}(P) = &\sum_{i=1}^{n_v}\begin{bmatrix}\tilde{\mathcal{A}}_i^T P\tilde{\mathcal{A}}_i & 0\\ 0 & 0\end{bmatrix} \\ &+\begin{bmatrix}\tilde{\mathcal{A}}^T P\tilde{\mathcal{A}} - P + \tilde{\mathcal{C}}_1^T S\tilde{\mathcal{C}}_1 & \tilde{\mathcal{A}}^T P\tilde{\mathcal{B}}_1 + \tilde{\mathcal{C}}_1^T S\mathcal{D}_1 \\ \tilde{\mathcal{B}}_1^T P\tilde{\mathcal{A}} + \mathcal{D}_1^T S\tilde{\mathcal{C}}_1 & \tilde{\mathcal{B}}_1^T P\tilde{\mathcal{B}}_1 - \mathcal{D}_1^T S\mathcal{D}_1 - \gamma^2 S\end{bmatrix} \preceq 0\end{aligned} \tag{2.18}$$

where

$$\begin{aligned}\tilde{\mathcal{A}} &= \hat{\mathcal{A}} + \hat{\mathcal{B}}_2\kappa \\ \tilde{\mathcal{A}}_i &= \hat{\mathcal{A}}_i + \hat{\mathcal{B}}_{2i}\kappa\end{aligned}$$

Detailed derivation from Eq. (2.17) to Eq. (2.18) can be found in Appendix A.

Now we have successfully formulated two key criteria, Eq. (2.12) and (2.18), for the controller design. In what follows, we will formulate an optimization problem with Linear Matrix Inequality(LMI) condition by re-organizing Eq. (2.18) in to a single block matrix. This will enable the static gain κ to be computed by a Semi-definite programming solver while trying to minimize γ.

Pre- and post- multiplying $\begin{bmatrix}P^{-1} & 0\\ 0 & I_{n_\zeta}\end{bmatrix}^T$ and $\begin{bmatrix}P^{-1} & 0\\ 0 & I_{n_\zeta}\end{bmatrix}$ respectively on the left side of Eq. (2.18) will not change the negative definite condition:

$$\begin{aligned}&\sum_{i=1}^{n_v}\begin{bmatrix}\hat{\mathcal{A}}_i P^{-1} + \hat{\mathcal{B}}_{2i}\kappa P^{-1} & 0\\ 0 & 0\end{bmatrix}^T\begin{bmatrix}P & 0\\ 0 & S\end{bmatrix}\underbrace{\begin{bmatrix}\hat{\mathcal{A}}_i P^{-1} + \hat{\mathcal{B}}_{2i}\kappa P^{-1} & 0\\ 0 & 0\end{bmatrix}}_{\mathcal{A}_{i,c}} - \begin{bmatrix}P^{-1} & 0\\ 0 & \gamma^2 S\end{bmatrix} \\ &+\begin{bmatrix}\hat{\mathcal{A}}P^{-1} + \hat{\mathcal{B}}_2\kappa P^{-1} & \mathcal{B}_1\\ \hat{\mathcal{C}}P^{-1} + \hat{\mathcal{D}}_2\kappa P^{-1} & \mathcal{D}_1\end{bmatrix}^T\begin{bmatrix}P & 0\\ 0 & S\end{bmatrix}\underbrace{\begin{bmatrix}\hat{\mathcal{A}}P^{-1} + \hat{\mathcal{B}}_2\kappa P^{-1} & \mathcal{B}_1\\ \hat{\mathcal{C}}P^{-1} + \hat{\mathcal{D}}_2\kappa P^{-1} & \mathcal{D}_1\end{bmatrix}}_{\mathcal{A}_c} \preceq 0\end{aligned} \tag{2.19}$$

The following lemma is useful to formulate a LMI block matrix:

Lemma 1 ***Shur Complement*** *Suppose matrices $A \in \mathbf{R}^{p\times p}$, $B \in \mathbf{R}^{p\times q}$, $C \in \mathbf{R}^{q\times p}$, $D \in \mathbf{R}^{q\times q}$, and D is invertable. A block matrix M takes the form $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, then the Schur complement of the block matrix D of M is: $M/D \triangleq A - BD^{-1}C$. Furthermore, if M is a positive definite, D is also positive definite.*

According to Lemma 1, the negative definite property remains if we keep applying the Shur Complement block matrix on the left hand side of Eq. (2.19).
Thus the final negative definite block matrix from Eq. (2.19) can be drawn as:

$$\mathcal{L}(\mathcal{Q}) \triangleq \begin{bmatrix} -\mathcal{Q}_\gamma & \mathcal{A}_c^T & \tilde{\mathcal{A}}_c^T \\ \mathcal{A}_c & -\mathcal{Q}_S & 0 \\ \tilde{\mathcal{A}}_c & 0 & -\mathcal{Q}_d \end{bmatrix} \preceq 0 \tag{2.20}$$

With the following blocks in Eq. 2.20 defined as:

$$\mathcal{Q} \triangleq P^{-1}$$
$$\mathcal{Q}_\gamma \triangleq diag(\mathcal{Q}, \gamma^2 S),\ \mathcal{Q}_S \triangleq diag(\mathcal{Q}, S^{-1}),\ \mathcal{Q}_d \triangleq diag(\underbrace{\mathcal{Q}_S, \cdots, \mathcal{Q}_S}_{n_v})$$
$$\mathcal{A}_c \triangleq \begin{bmatrix} \hat{\mathcal{A}}\mathcal{Q} + \hat{\mathcal{B}}_2\mathcal{Y} & \mathcal{B}_1 \\ \hat{\mathcal{C}}\mathcal{Q} + \hat{\mathcal{D}}_2\mathcal{Y} & \mathcal{D}_1 \end{bmatrix},\ \mathcal{A}_{i,c} \triangleq \begin{bmatrix} \hat{\mathcal{A}}_i\mathcal{Q} + \hat{\mathcal{B}}_{2i}\mathcal{Y} & 0 \\ 0 & 0 \end{bmatrix}$$
$$\tilde{\mathcal{A}}_c \triangleq \begin{bmatrix} \mathcal{A}_{1,c} & \cdots & \mathcal{A}_{n_v,c} \end{bmatrix},\ \mathcal{Y} = \kappa\mathcal{Q}$$

2.3.5 Robust mean-square stabilizing data-driven controller

Now it is ready to compute the subspace static feedback gain κ.
Theorem 1 *A static χ state feedback controller stabilizes the uncertain system defined by Eq.* (2.9) *and* (2.10) *in the mean-square sense with the induced RMS criterion, if the following optimization problem is feasible:*

$$\min_{\mathcal{Y},\mathcal{Q},S} \gamma \tag{2.21}$$
$$s.t. \quad \mathcal{L}(\mathcal{Q}) \preceq 0$$
$$\mathcal{Q} \succ 0$$
$$S \succ 0$$
$$\gamma > 0$$

Proof: Proof is omitted as it is clear to observe from the above equations that Eq. (2.20) is a sufficient condition to satisfy averaged value dissipation Lyapunoc condition in Eq. (2.15). ■

Remark: The optimization problem provided in Theorem 1 can be slightly tuned in order to be solved efficiently by a Semi-Definite programming solver. Instead of using the form as shown in Eq. (2.21), one can solve the LMI optimization problem given as:

$$\min_{\mathcal{Y},\mathcal{Q},S,S_\gamma} S_\gamma \tag{2.22}$$

$$\begin{aligned} s.t. \quad \mathcal{L}(\mathcal{Q}) &\preceq 0 \\ \mathcal{Q} &\succ 0 \\ S &\succ 0 \\ S_r &\succ 0 \end{aligned}$$

where S_γ is a diagonal positive matrix used to approximate $\gamma^2 S$ and block $\mathcal{Q}_\gamma$ in $\mathcal{L}(\mathcal{Q})$ in (2.22) is changed to $\mathcal{Q}_\gamma \triangleq diag(\mathcal{Q}, S_\gamma)$.

Finally, by solving the optimization problem, the static feedback gain κ can be given as $\kappa = \mathcal{Y}\mathcal{Q}^{-1}$.

2.3.6 Summary of the proposed data-driven controller design

With the discussions above, the procedures to derive the robust mean-square stabilizing data-driven controller given in Theorem 1 can be summarized as:

1. Do an identification experiment of the plant at a chosen equilibrium. If the system is unstable, do a closed-loop experiment with an empirical, initial stabilizing controller.
2. Collect a set of N I/O data from the identification experiment.
3. Select past window size p, preferably covering most dynamics of the system.
4. Compute the Markov Parameters through the LS formulation of the I/O data according to Eq. (2.5)
5. Define the performance vector ζ.
6. Solve the problem defined in Eq. (2.21) and derive the feedback gain κ.
7. Apply the derived controller as is shown in Fig. 2.1.

Remark: There are two tuning parameters in the proposed controller design, i.e. p and ζ. The choice of p only requires taking into account the major dynamic behavior of the system. In fact, as the dynamics of the subspace representation is closely related to the state-space estimator through Φ, a rough estimate on the length of p can be guided by the estimator design if there is an initial controller in the identification step. ζ is used in Eq. 2.12, explicitly related to the expected control behavior. Thus the controller is easily tuned, i.e. to be aggressive or moderate by the choice of ζ. The proposed controller can be computationally intensive. However, only step 7 needs to be carried out online. Most of the computation can be conducted offline and the online part does not require any more resources than a LQ controller, which has the ability to act in a plug-and-play style.

3

Introduction of the Experiment — Platform and Process

This chapter briefly introduces the information of the experiment — where and how the proposed data-driven controller is implemented. In section 3.1, the configuration of the 2 DoF inverted pendulum is briefly described. Section 3.2 describes the steps to implement the proposed data-driven controller on this device.

3.1 Introduction of the 2 DoF inverted pendulum lab device

Figure 3.1: Experiment setup, Quanser 2 DOF Inverted Pendulum [15]

An inverted pendulum is a special pendulum with its center of mass above its pivot. Usually it contains a rigid rod mounting on a small manipulator. The physical structure of an inverted pendulum system is typically non-linear and unstable. The laboratory device used for this thesis work is a 2 DoF inverted pendulum system where the rod can turn in 2 orthogonal directions as is shown in Fig. 3.1. The control task is to balance and keep the rod upright by moving the manipulator. For

this specific system, the pivot position is controlled by two robot arm manipulators driven by two DC motors.
A detailed configuration of the lab device can be seen in Fig. 3.2 and its home position is defined as Fig. 3.3.

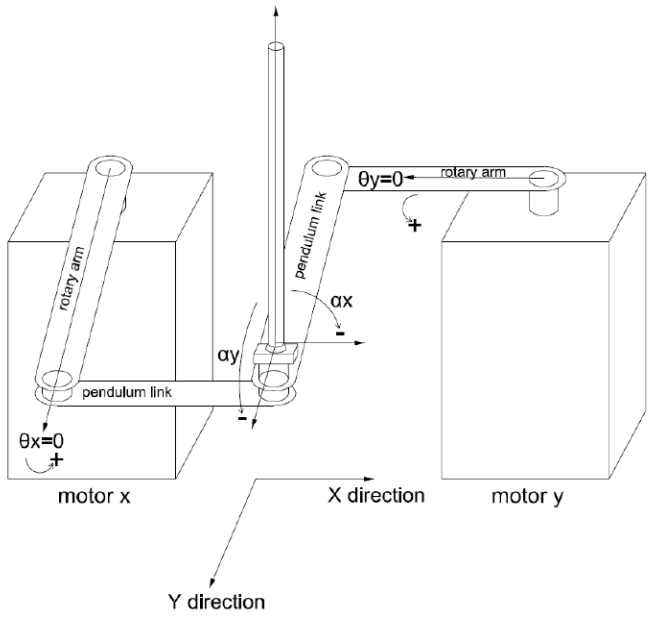

Figure 3.2: Detailed configuration of the 2 DoF Inverted Pendulum

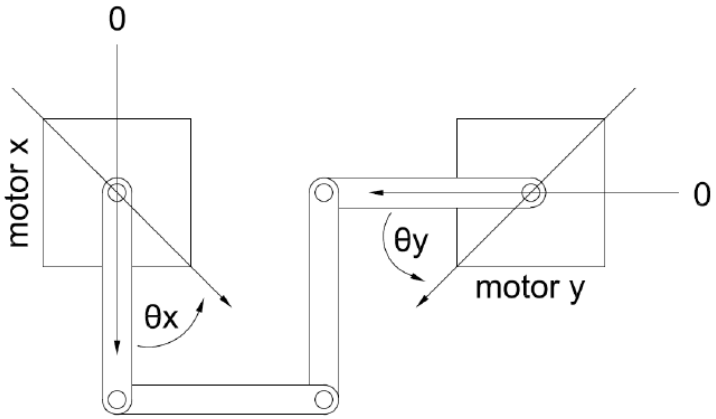

Figure 3.3: Home position of the 2 DoF Inverted Pendulum

As can be seen from Fig. 3.2, within a small region around the home position and with some approximation, the servo motor x (on the left hand side) controls the pendulum dynamics of direction x while the servo motor y controls the dynamics in direction y. The tilt of the rod can thus be split into plane x and y respectively. The links that are connected to the servo motors are two rotary arms. According to Fig. 3.2, the following parameters in Table 3.1 are defined.

Table 3.1: Parameters of the Inverted Pendulum (1)

θ_x	The rotation angle of the servo motor x
θ_y	The rotation angle of the servo motor y
α_x	The tilt angle of the pendulum in plan x
α_y	The tilt angle of the pendulum in plan y

Indeed, parameters provided in Table 3.1 are able to measure in the experiment. However, these parameters are insufficient to set-up the first principle model of the 2 DoF inverted pendulum system and more states need to be estimated through an observer. Detailed modeling steps of the pendulum set-up can be seen in Appendix B.
As Fig. 3.3 illustrates the geometry of the home position and the zero point for the rotary arms, the rotation angles θ_x, θ_y and the tilt angles α_x and α_y increase in Counter-ClockWise(CCW) direction. The servo motors are designed to have positive voltage input when turning the rotary arms CCW [15].

3.2 Experiment Process

To successfully implement the proposed data-driven controller on the 2 DoF inverted pendulum system, there are three steps that need to follow:

1. Derive an initial controller based on some knowledge of the first principle model. Do a closed-loop experiment an the initial controller with additional excitation signal. Collect I/O data and compute the Markov Parameters. Detailed information on deriving the initial controller can be found in Appendix B.
2. Compute the subspace static feedback gain κ as is shown in Chapter 2 and formulate the robust mean-square stabilizing data-driven controller according to Fig. 2.1.
3. Start a new experiment with the new data-driven controller around the equilibrium.

The reason why a closed-loop identification is carried out in step 1 is that the 2 DoF inverted pendulum system is open-loop unstable. The initial controller used in this thesis is an LQ controller as is shown in Appendix B. Also note that step 2 can be done offline completely with the collected I/O data from step 1. Only step 3 is done online for the data-driven controller, which can act in a plug-and-play manner by a simple switch from the initial LQ controller.

At the end of this chapter, it is worthy to point out again that the 2 DoF inverted pendulum system can be approximately decoupled as two same 1 DoF inverted pendulum subsystems around the home position as is shown in Fig. 3.3. Detailed explanations can also be found in Appendix B. Due to this fact, it is sufficient in the following chapter to observe only two measurements and one input from a 1 DoF subsystem during the experiment.

4

Experimental Result

The proposed robust mean-square stabilizing data-driven controller in Chapter 2 is implemented on a 2 DoF inverted pendulum lab device as is introduced in Chapter 3. In this chapter, the experimental results of the implementation will be presented. The experiments were carried out on a PC running Windows 10 system, using MATLAB®version R2015a and Quanser®driver version v2.5.1431.0. The optimization problem is solved using Sedumi 1.3 under yalmip[26] in the MATLAB®environment.

4.1 Identification Data

Among the testing experiments of the 2 DoF inverted pendulum system, a subspace model identified from a set of only 300 samples is already sufficient to formulate a well-functioned data-driven controller proposed in this thesis. However, there is a requirement on the identification data which is important for the controller design — excitation around the equilibrium. Fig. 4.1 shows two different output measurement data sets from two individual identification experiments, one on the left two plots and the other on the right. The sampling time is $Ts = 0.2\ s$ and each data set contains 300 samples. On the left side of Fig. 4.1, it is clear that the first measurement is biased from the equilibrium.

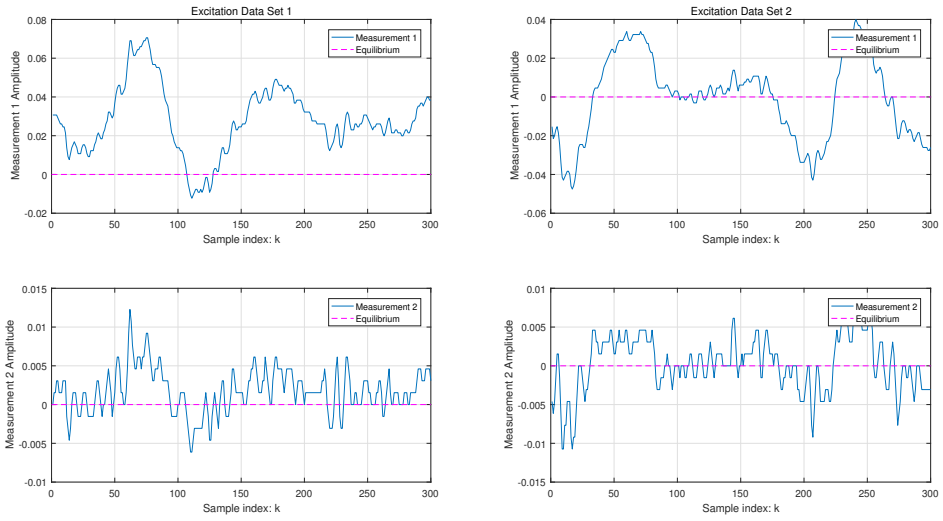

Figure 4.1: Two set of identification I/O data. Solid line - measurement, dashed line - equilibrium

In the further experiments, using the data set on the left side cannot formulate any functional data-driven controllers no matter how the past window size p and the performance vector ζ are tuned. However, the right side data successfully leads to stabilizing data-driven controllers.

4.2 Markov Parameters Estimate

As a property of linear Least Square estimate, the result will converge to the true value as the number of samples is sufficiently large. In order to illustrate the estimate of the MP clearer, in Fig. 4.2 and 4.3, the estimated MP illustrated are computed from $10,000$ samples to alleviate the influence of noise and disturbance.
Recall that the MP takes the following form:

$$\Xi \triangleq \begin{bmatrix} CB & CL & \cdots & C\Phi^{p-1}B & C\Phi^{p-1}L \end{bmatrix} \tag{4.1}$$

and according to the identification equation Eq. (2.4), the Markov Parameters can be divided into mainly two groups: Group $C\Phi^k B$ whose components are multiplied with the input signal u and Group $C\Phi^k L$ whose components are multiplied with the output signal y. As in Eq. (2.4), the bias term $C\Phi^p X$ can be interpreted as an index of how good the MP estimate is. Since x in the experiment is closely related to the measurement y in state-space model, Fig. 4.2 illustrated $||C\hat{\Phi}^k L||$ for different past window size p so that a rough estimate of $||C\Phi^p X||$ can be approximated for each case respectively. The solid line is a computed reference from the information of the initial controller design as is shown in Appendix B. Need to note that such reference does not necessarily be the "true value". However, it is still useful to get an idea on how $||\Phi^k||$ decays.

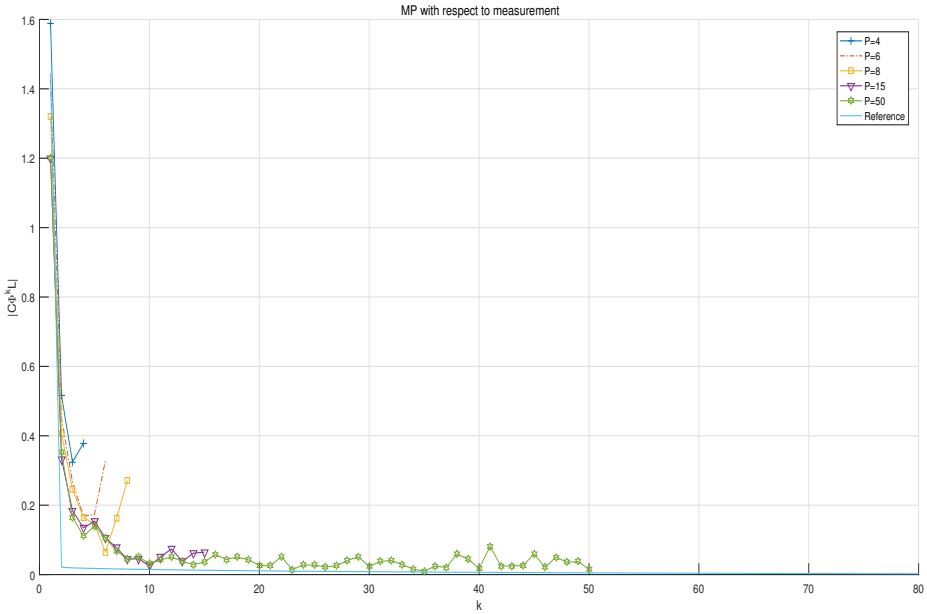

Figure 4.2: Norm of $||C\hat{\Phi}^k L||$ for different p

In Fig. 4.2, we can see that as p increases, $||C\Phi^k L||$ tends to converge to the "actual" shape (monotonically decreasing) as $||\Phi|| < 1$. However in the real experiment, when $p = 8$, a functional data-driven controller as proposed in this thesis can already be formulated. One possible reason is that it is already sufficient to capture the major dynamics with $p = 8$ and the bias term left is small enough. And the first component, $k = 1$ in the curve $||C\Phi^p L||$, of $p = 8$ is much closer to $p = 50$ comparing with the cases of $p < 8$. This may suggest that the MP provided with $p = 8$ has reached certain accuracy to be used to do the proposed controller design.

Fig. 4.3 illustrates the MP related to $||C\Phi^k B||$. Similarly, the solid line is a computed reference. However, the reference has a point $C\Phi^k B = 0.3114$ when $k = 2$. Fig. 4.3 does not include such point as it is too large to see the other results clearly. However, the reference is kept in Fig.4.3 as it is still useful to illustrate how $||\Phi^k||$ decays.

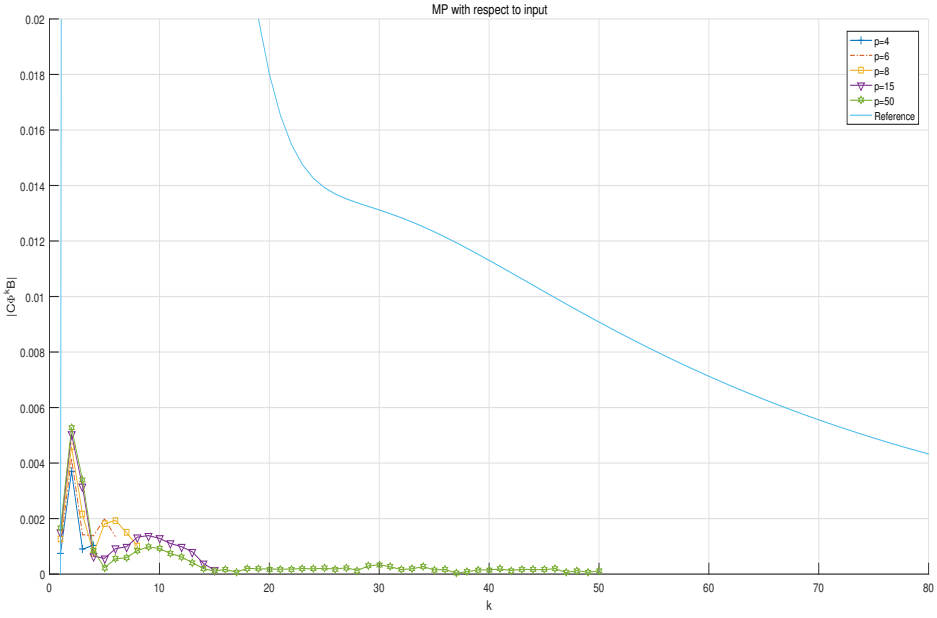

Figure 4.3: Norm of $||C\hat{\Phi}^k B||$ for different p

We can see that the estimated results for the Markov Parameters with respect to the input keeps a similar shape to the calculated reference. However, they seem to have a different order magnitude. This may be caused by the difference in the amplitudes of the measurement and input from in the identification experiment. As is shown in Fig. 4.1, measurement 1 has the order of 10^{-2} and 10^{-3} for measurement 2. However, the input is mostly between -1 to 1 as is shown in Fig. 4.4.

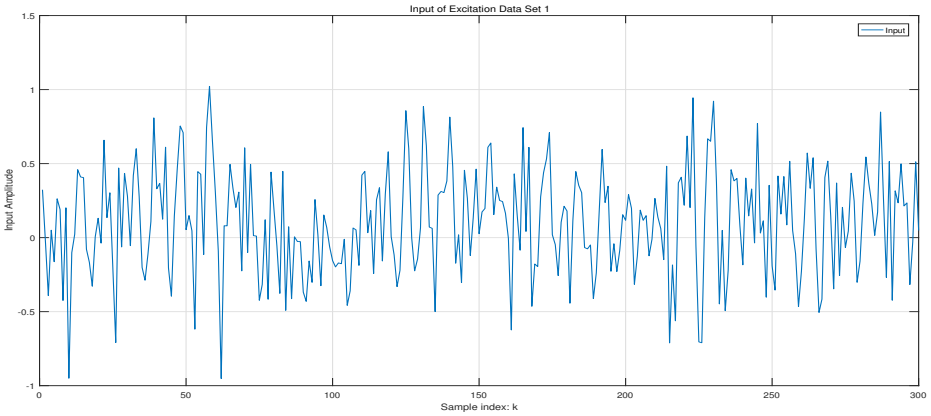

Figure 4.4: Input signal from the same identification experiment as Fig. 4.1

However, putting Fig. 4.2 and 4.3 together, Fig. 4.5 illustrates the curves of $\sqrt{||C\Phi^k B||^2 + ||C\Phi^k L||^2}$ for different choice of p.

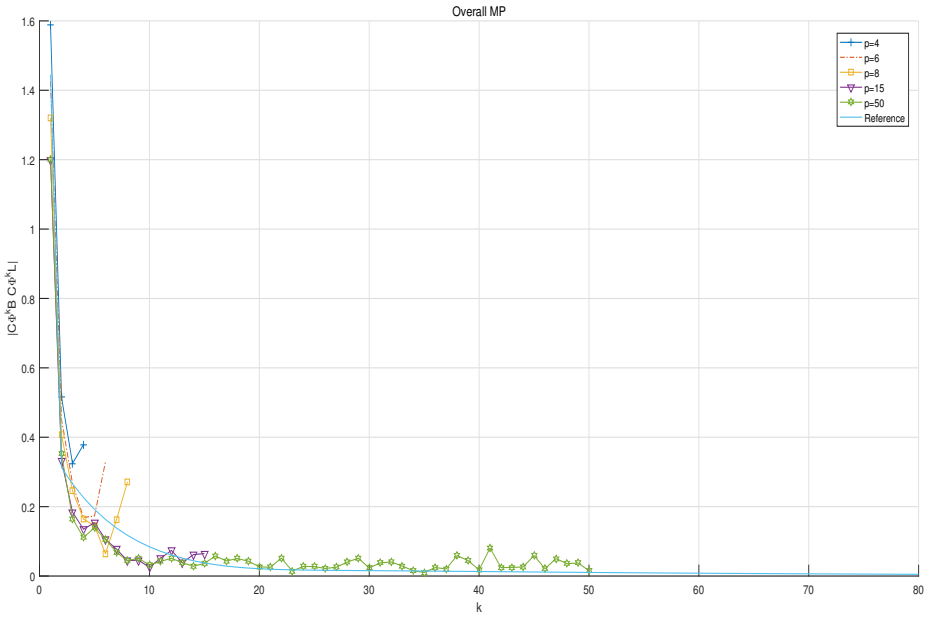

Figure 4.5: Norm of Markov Parameters

From Fig. 4.5 we can see that even if there may exist scaling problem for the MP estimate considering the output and input components respectively, the overall MP estimate does not diverge from the computed reference too much. Since it is uncertain whether the reference MP is the best representation of the real system, the difference of the identified MP from the reference may also be an improvement of the dynamic model.

4.3 Selection of the past window size

The reason has been mentioned in the previous sections why a relatively short p is preferred. From Fig. 4.5, one can also observe that the curve for $p = 50$ has the risk of over-fitting the noise by a large p as the tail of such curve starts to oscillate.

Though $p_{min} = 8$ is found as the minimal past window size for a functional proposed data-driven controller design in this particular experiment by empirical tests, it is still unclear how to determine the exact way to do the selection.

Apart from an estimate from Φ in the initial controller design and simulation, Fig. 4.6 may also provide an alternative way to give an rough guide. Recall the subspace dynamic model equation (2.8), Fig. 4.6 illustrates the sigma plot of such model, by substituting each MP component with its estimate for different past window sizes p.

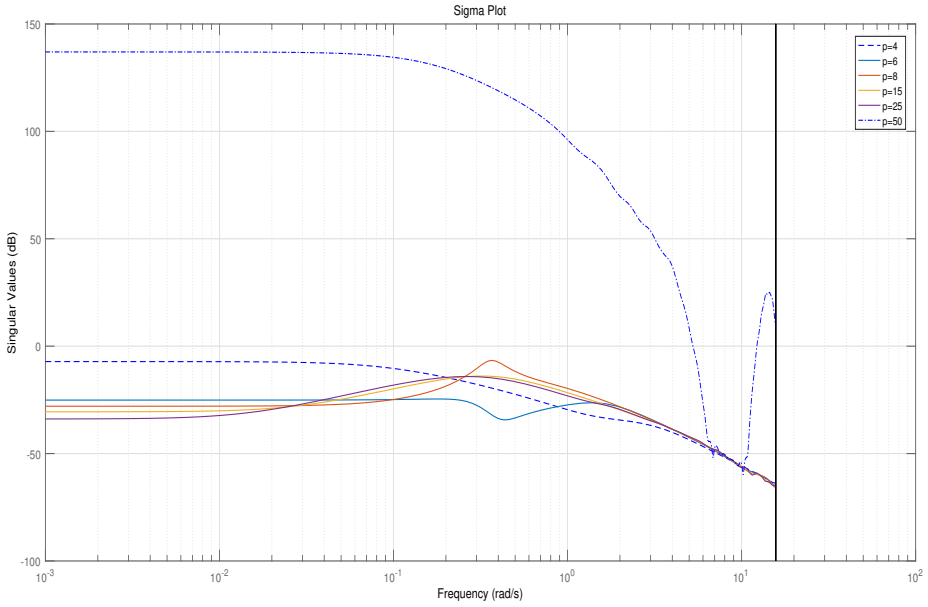

Figure 4.6: Sigma Plots for the estimated subspace dynamic model (without uncertainty component)

From $p = 3$ to $p = 50$, the level of the sigma curves first goes down and then rises up which may be useful to set a threshold for the low-frequency amplitude for p selection. It is also noted that for $p = 8, 15, 25$, the sigma curves are very similar in shape. As $p = 8$ is the minimal past window size in this experiment, this similarity in the sigma plot may also provide a rough range of p selection for the implementation and thus can be used to estimate the minimal p.

4.4 Closed-loop Performance

During the implementation, the minimum past window size is $p = 8$ which is able to lead to successful robust mean-square stabilizing data-driven controller designs. Fig. 4.7 shows an example of the closed-loop behavior of a successful design with $p = 8$. The identification of the MP uses 300 samples as is shown on the right hand side of Fig. 4.1 and Fig. 4.4. Recall that in the controller design algorithm in section 2.3.6, the performance vector ζ is a tuning parameter. In this case, $\zeta = \begin{bmatrix} 10 & 0 & 0 & \cdots & 0 \\ 0 & 200 & 0 & \cdots & 0 \end{bmatrix}$ is defined, interpreted as different penalizing weights on the actual measurement of the system. In Fig. 4.7, the closed-loop performance of the initial LQ controller is also illustrated(data collected from a separate experiment, without the additional random noise for identification purposes). Both results has a total of $10,000$ samples collected.

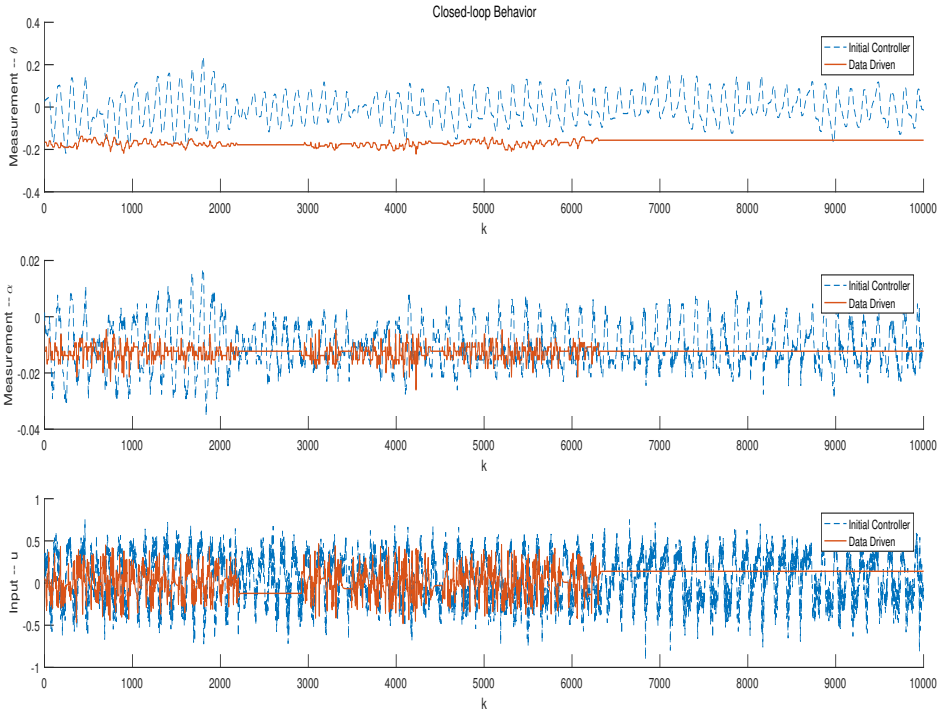

Figure 4.7: Closed-loop behavior for $10,000$ samples. Solid line – proposed data-driven controller; dashed line – initial LQ controller without additional noise signal

We can see in Fig. 4.7 that the proposed data-driven controller achieves a better stationary closed-loop behavior by having smaller oscillations both for the measurements and the input signal. The cause of the steady-state tracking error for the proposed data-driven controller is probably the non-linearity of the real system, as the formulation in this experiment does not include an integrator. In order to compare the results better, we take the Discrete Fourier transform for both closed-loop dynamics in Fig. 4.7. The result of the amplitude spectrum is shown in Fig. 4.8

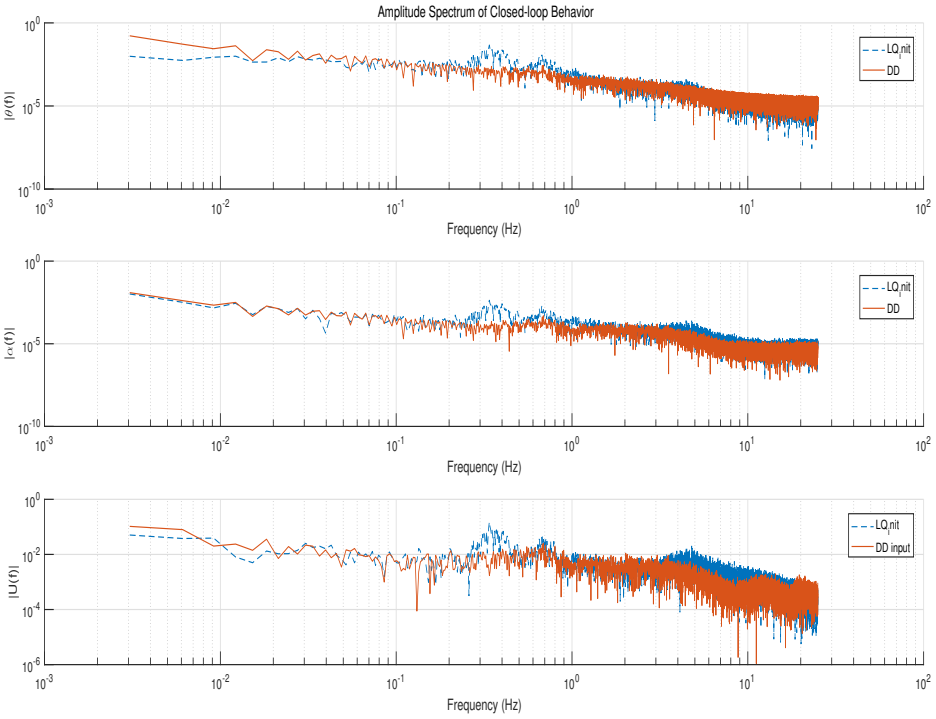

Figure 4.8: Amplitude spectrum of Fig. 4.7. Solid line – proposed data-driven controller; dashed line – initial LQ controller.

From Fig. 4.8, we can conclude that the proposed data-driven controller outperforms the initial LQ controller in a way of flattening the high frequency amplitude for both measurement signal and input. This is intuitively understandable. As in the proposed data-driven controller design, the state χ is augmented to be a history of I/O data. Having more states usually means having more freedom to control. The computation of the controller is in some sense similar to a p size FIR filter moving with time on the I/O data. Furthermore, the data-driven controller does not involve speed estimate, and speed information is likely to bring high frequency component. In addition, if we compute the total energy of the inputs and outputs for both controllers through Parseval Theorem:

$$E = \int_{-\infty}^{\infty} |X(f)|^2 \mathrm{d}f = \int_{-\infty}^{\infty} |x(t)|^2 \mathrm{d}x \tag{4.2}$$

where E is the total energy of an arbitrary signal x. We will have the following result for this case:

Table 4.1: Total energy for the closed-loop control behaviors in Fig. 4.8

	$E(\theta)$	$E(\alpha)$	$E(u)$
Initial LQ Controller	0.01826	0.000783	0.27076
Data-driven Controller	0.14877	0.000841	0.07859

Due to the steady-state tracking error, the data-driven controller has larger energy for both measurements θ and α within the illustrated $10,000$ samples. However, the

total energy of the control move computed by the proposed data-driven controller is significantly smaller than the one computed by the initial LQ controller to keep the pendulum upright. This suggests that the proposed data-driven controller can have very good ability to achieve energy saving performance in practical fixed-point stabilizing tasks.

4.5 Tuning of the Closed-loop Dynamics

Given a feasible past window size p, the only tuning parameter left is the performance vector ζ. For convenience, we can always define ζ to correspond to the actual measurement of the real system with different penalizing weights respectively. For the 2 DoF inverted pendulum system in this thesis, the performance vector takes the following form:

$$\zeta = \begin{bmatrix} W_1 & 0 & 0 & \cdots & 0 \\ 0 & W_2 & 0 & \cdots & 0 \end{bmatrix} \chi(k)$$

as there are two outputs for each decoupled 1 DoF inverted pendulum subsystem. W_1 and W_2 are manually defined penalizing weights on each measurement respectively. Fig. 4.9 shows $10,000$ samples of the closed-loop responses of two data-driven controller with different W_1 and W_2. In the figure, such weights are denoted as $[W_1\ W_2]$ in the legend. Apart from the weights, both controller share the same identification data and the past window size $p = 8$.

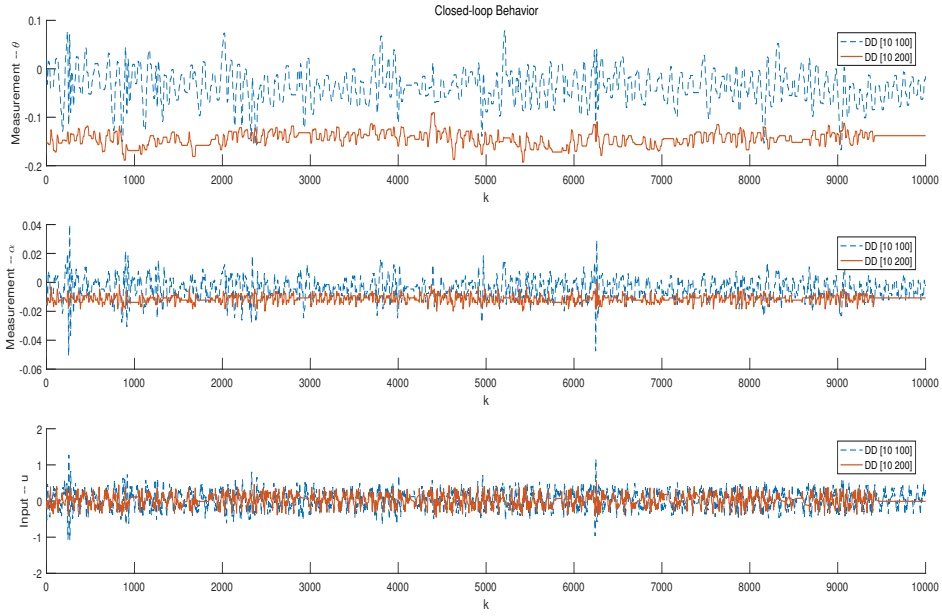

Figure 4.9: Closed-loop data-driven responses. Solid line – with $W_1 = 10,\ W_2 = 200$ in ζ; dashed line – with $W_1 = 10,\ W_2 = 100$ in ζ.

Neglecting the steady-state tracking error in Fig. 4.9 as the initial condition differs in the real experiment and is hard to control, more weights on ζ design, less oscillation

will occur in the real implementation.
Similar to Eq. (4.2), the total energy of the input signal $E(u)$ is calculated for both case:

Table 4.2: Total input energy for the closed-loop control behaviors in Fig. 4.9

	$E(u)$
$W_1 = 10,\ W_2 = 100$	0.18936
$W_1 = 10,\ W_2 = 200$	0.07508

As a matter of smaller output oscillations with larger penalizing weights, such a data-driven controller also achieves better energy saving behavior for the input signal.

4.6 Summary

The identification of the LTI I/O level subspace model used in the proposed data-driven controller design algorithm is relatively convenient and easy. The simple LS estimate enables the identification of the system dynamics with only a small number of I/O samples.
The proposed data-driven control algorithm aims at providing a robust mean-square stabilizing subspace controller with a short past horizon size p. Though how to give the minimum p explicitly in practice remains a question to solve, there is a way to make an estimate. If the system identification is done in a closed-loop style, the first principle model and the state estimator information used in the initial controller design can be a useful guide to compute Φ and simulate the data-driven controller under different choice of p.
Once p is found, the only tuning parameter for the proposed control algorithm is the performance vector ζ, which in general can be selected as the actual system output of interest with penalizing weights. The closed-loop control behavior can thus be tuned by addressing such weights. In the 2 DoF inverted pendulum experiment, a better equilibrium stabilizing behavior can be achieved with larger penalizing weights. However, very large weights will lead to too aggressive static feedback gain κ which, on the other hand, will cause instability of the system.
The proposed data-driven controller can be very good at equilibrium stabilizing tasks. In the experiment, it shows good ability at flatting the control moves and reducing control efforts to stabilize the system, while maintaining good control performances. Therefore, this indicates the potential application of such kind of controllers in real life with energy saving respect.
One possible way to eliminate the steady-state tracking error is to augment state the $\chi(k)$ with the integral of the measurements. Due to time limit of this project, this work and experimental tests will remain as a future work.
As the data-driven controller design algorithm in this thesis is based on an closed-loop identification with additional input noise in the identification experiment, the derived data-driven controller therefore is able to be robust against the input noise of the same type and level. However, it risks failing to work if the initial condition of the pendulum is too far away from the designed equilibrium (home position).

Comparing with the initial LQ controller, the proposed data-driven controller is better in terms of the stabilizing dynamic behavior, but worse in tolerating position bias from the designed equilibrium. This fact may be due to the highly non-linear property of the inverted pendulum system which causes the local I/O level model fail to represent the system dynamics.

A

Derivation of the Matrix Inequality Condition

Recall $\mathcal{L}(P, v)$ defined in Eq. (2.17):

$$\mathcal{L}(P,v) = \begin{bmatrix} \tilde{\mathcal{A}}_1^T P \tilde{\mathcal{A}}_1 - P + \tilde{\mathcal{C}}_1^T S \tilde{\mathcal{C}}_1 & \tilde{\mathcal{A}}_1^T P \tilde{\mathcal{B}}_1 + \tilde{\mathcal{C}}_1^T S \mathcal{D}_1 \\ \tilde{\mathcal{B}}_1^T P \tilde{\mathcal{A}}_1 + \mathcal{D}_1^T S \tilde{\mathcal{C}}_1 & \tilde{\mathcal{B}}_1^T P \tilde{\mathcal{B}}_1 - \mathcal{D}_1^T S \mathcal{D}_1 - \gamma^2 S \end{bmatrix} \tag{A.1}$$

with

$$\begin{aligned} \tilde{\mathcal{A}}_1 &= \hat{\mathcal{A}} + \hat{\mathcal{B}}_2 \kappa + \sum_{i=1}^{n_v} (\hat{\mathcal{A}}_i + \hat{\mathcal{B}}_{2i} \kappa) v_i \\ \tilde{\mathcal{B}}_1 &= \mathcal{B}_1 \\ \tilde{\mathcal{C}}_1 &= \mathcal{C} + \mathcal{D}_2 \kappa \end{aligned}$$

Operator $\mathbf{E}_v\{\mathcal{L}(P, v)\}$ takes the expected value with respect to v, thus we only need to consider the expected value of the terms that contain $\tilde{\mathcal{A}}_1$ as the others will be treated as constants.
Denote $\tilde{\mathcal{A}} \triangleq \hat{\mathcal{A}} + \hat{\mathcal{B}}_2 \kappa$ and $\tilde{\mathcal{A}}_i \triangleq (\hat{\mathcal{A}}_i + \hat{\mathcal{B}}_{2i} \kappa)$ for $i \in \{1, 2, \cdots, n_v\}$, then we have $\tilde{\mathcal{A}}_1 = \tilde{\mathcal{A}} + \sum_{i=1}^{n_v} \tilde{\mathcal{A}}_i v_i$. Consider the following:

$$\mathbf{E}_v\{\tilde{\mathcal{A}}_1^T P \tilde{\mathcal{A}}_1\} = \mathbf{E}_v\{(\tilde{\mathcal{A}} + \sum_{i=1}^{n_v} \tilde{\mathcal{A}}_i v_i)^T P (\tilde{\mathcal{A}} + \sum_{i=1}^{n_v} \tilde{\mathcal{A}}_i v_i)\} \tag{A.2}$$

As v_i is zero-mean, the expected value of the first order term with respect to v_i is zero after expanding Eq. (A.2). Thus we will have:

$$\mathbf{E}_v\{\tilde{\mathcal{A}}_1^T P \tilde{\mathcal{A}}_1\} = \mathbf{E}_v\{\tilde{\mathcal{A}}^T P \tilde{\mathcal{A}}\} + \mathbf{E}_v\{(\sum_{i=1}^{n_v} \tilde{\mathcal{A}}_i v_i)^T P (\sum_{i=1}^{n_v} \tilde{\mathcal{A}}_i v_i)\} \tag{A.3}$$

Since v is a zero mean random signal with identity covariance matrix and v_i is the i^{th} entry from the random vector v for $i \in \{1, 2, \cdots, n_v\}$, v_i are zero-mean and independent from each other. Furthermore, $\mathbf{E}\{v_i^2\} = 1$. Eq. (A.3) will thus become:

$$\begin{aligned} \mathbf{E}_v\{\tilde{\mathcal{A}}_1^T P \tilde{\mathcal{A}}_1\} &= \mathbf{E}_v\{\tilde{\mathcal{A}}^T P \tilde{\mathcal{A}}\} + \sum_{i=1}^{n_v} \mathbf{E}_v\{(\tilde{\mathcal{A}}_i v_i)^T P (\tilde{\mathcal{A}}_i v_i)\} \\ &= \tilde{\mathcal{A}}^T P \tilde{\mathcal{A}} + \sum_{i=1}^{n_v} \tilde{\mathcal{A}}_i^T P \tilde{\mathcal{A}}_i \end{aligned} \tag{A.4}$$

Similarly to $\mathbf{E}_v\{\tilde{\mathcal{A}}_1^T P \tilde{\mathcal{A}}_1\}$, we can have:

$$\mathbf{E}_v\{\tilde{\mathcal{A}}_1^T P \tilde{\mathcal{B}}_1\} = \tilde{\mathcal{A}}^T P \tilde{\mathcal{B}}_1 \tag{A.5}$$

$$\mathbf{E}_v\{\tilde{\mathcal{B}}_1^T P \tilde{\mathcal{A}}_1\} = \tilde{\mathcal{B}}_1^T P \tilde{\mathcal{A}} \tag{A.6}$$

Therefore, $\mathbf{E}_v\{\mathcal{L}(P, v)\} =$

$$\begin{bmatrix} \tilde{\mathcal{A}}^T P \tilde{\mathcal{A}} + \sum_{i=1}^{n_v} \tilde{\mathcal{A}}_i^T P \tilde{\mathcal{A}}_i - P + \tilde{\mathcal{C}}_1^T S \tilde{\mathcal{C}}_1 & \tilde{\mathcal{A}}^T P \tilde{\mathcal{B}}_1 + \tilde{\mathcal{C}}_1^T S \mathcal{D}_1 \\ \tilde{\mathcal{B}}_1^T P \tilde{\mathcal{A}} + \mathcal{D}_1^T S \tilde{\mathcal{C}}_1 & \tilde{\mathcal{B}}_1^T P \tilde{\mathcal{B}}_1 - \mathcal{D}_1^T S \mathcal{D}_1 - \gamma^2 S \end{bmatrix}$$

which is the $\mathcal{L}(P)$ given in Eq. (2.18).

B

First Principle Modeling of the Inverted Pendulum and Initial Controller Design

In this section, a discrete time LTI state-space model will be derived for the 2 DoF inverted pendulum as is used to design the initial controller. We will first write down the system dynamic equations based on the physical principles of the pendulum. With the dynamical equations, a continuous LTI state-space model is derived by linearizing the equations at the home position. Then the continuous LTI state-space model will be discretized. Finally the state observer and the initial LQ controller will be designed based on such discretized state-space model.

B.1 Mathematical Equations of the System Dynamics

In Fig. 3.3, within a close neighbour of the home position, a 2 DoF inverted pendulum system is decoupled in x and y plane. The dynamics can be considered as a combination of 2 independent 1 DoF inverted pendulum subsystems. Thus for simplicity, we approximate the entire system by two same set of 1 DoF system dynamic equations.
The parameters of the lab device shown in Table B.1 are available in [15]:

Table B.1: Parameters of the Inverted Pendulum (2)

M_p	Pendulum mass	$0.1270kg$
L_r	Length of rotary arm	$0.1270m$
L_p	Length of the pendulum	$0.3111m$
J_r	Equivalent inertia with the 4-bar linkage	$0.0083kg * m^2$
J_p	Pendulum inertia around CoG	$0.0012kg * m^2$
D_r	Arm viscous damping coefficient	$0.0690N * m * s/rad$
C_o	Voltage convert coefficient	$0.1285N * m * s/rad$
g	Gravitational constant	0.981 kg*s^2

If we neglect the contribution of the friction, the following nonlinear equations of

motion can be built for the decoupled 1 DoF inverted pendulum system:

$$\dot{\theta}(t) = \frac{\mathrm{d}\theta}{\mathrm{d}t} \triangleq f_1(t) \tag{B.1}$$

$$\dot{\alpha}(t) = \frac{\mathrm{d}\alpha}{\mathrm{d}t} \triangleq f_2(t) \tag{B.2}$$

$$\ddot{\theta}(t) = \frac{\mathrm{d}\dot{\theta}(t)}{\mathrm{d}t} \triangleq f_3(t) \tag{B.3}$$

$$\ddot{\theta}(t) = \frac{\mathrm{d}\dot{\alpha}(t)}{\mathrm{d}t} \triangleq f_4(t) \tag{B.4}$$

where

$$\begin{aligned} f_3(t) = & \frac{-\frac{1}{2}((4M_pL_p^2\alpha(t)\dot{\theta}(t)\dot{\alpha}(t) - 8C_oV_m(t) + 8D_r\dot{\theta}(t))J_p + M_p^2L_p^4\alpha(t)\dot{\theta}(t)\dot{\alpha}(t))}{((4J_r + 4M_pL_r^2)J_p + M_pL_p^2J_r)} \\ & - \frac{-\frac{1}{2}((M_p^2L_p^3L_r\dot{\theta}^2(t) + 20M_p^2L_p^2L_rg)\alpha(t) - 2M_pL_p^2D_r\dot{\theta}(t) + 2M_pL_p^2C_oV_m(t))}{((4J_r + 4M_pL_r^2)J_p + M_pL_p^2J_r)} \end{aligned} \tag{B.5}$$

$$\begin{aligned} f_4(t) = & \frac{((M_p^2L_p^2L_r^2 + M_pL_p^2J_r)\dot{\theta}^2(t) + 20J_rM_pL_pg + 20M_p^2L_r^2L_pg)\alpha(t)}{((4J_r + 4M_pL_r^2)J_p + M_pL_p^2J_r)} \\ & + \frac{2M_pL_rL_pC_oV_m(t) - 2M_pL_rL_pD_r\dot{\theta}(t) - M_p^2L_p^3L_r\alpha(t)\dot{\theta}(t)\dot{\alpha}(t)}{((4J_r + 4M_pL_r^2)J_p + M_pL_p^2J_r)} \end{aligned} \tag{B.6}$$

$V_m(t)$ is the control input to the servo motor (in voltage). $\theta_x(t)$, $\theta_y(t)$, $\alpha_x(t)$ and $\alpha_y(t)$ can be found in Table 3.1.

B.2 LTI State-Space Model of the Inverted Pendulum System

B.2.1 LTI State-Space Model

In modern control theory [16], LTI state-space model is a standard model form to simulate the system dynamics and to solve controller design problems. An LTI state-space model takes the following form:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{B.7}$$

$$y(t) = Cx(t) + Du(t) \tag{B.8}$$

$t \geq 0$, where $x(t) \in \mathbf{R}^{n_x}$ denotes the state vector containing all the n_x states necessary to represent the system dynamics, $u(t) \in \mathbf{R}^{n_u}$ is the input vector and $y(t) \in \mathbf{R}^{n_y}$ is the output vector. Each entry in matrices $A \in \mathbf{R}^{n_x \times n_x}$, $B \in \mathbf{R}^{n_x \times n_u}$, $C \in \mathbf{R}^{n_y \times n_x}$, $D \in \mathbf{R}^{n_y \times n_u}$ are restricted to be a static scalar for LTI systems.

B.2.2 Linearization of the Non-linear Dynamic Equations

Eq. (B.1), (B.2), (B.3) and (B.4) reveal the fact that each of the decoupled 1 DoF inverted pendulums is also a non-linear system. If we would like to take advantage of the LTI state-space model to design the controller, we can linearize Eq. (B.1), (B.2), (B.3) and (B.4) at the home position to derive a linear model taking the form as Eq. (B.7) and (B.8).
By observing Eq. (B.1), (B.2), (B.3) and (B.4), the state vector:

$$x(t) = \begin{bmatrix} x_1(t) & x_2(t) & x_3(t) & x_4(t) \end{bmatrix}^T = \begin{bmatrix} \delta\theta(t) & \delta\alpha(t) & \delta\dot{\theta}(t) & \delta\dot{\alpha}(t) \end{bmatrix}^T \tag{B.9}$$

and input:

$$u(t) = \delta V_m(t)$$

can be chosen to build the state-space model of the 1 DoF inverted pendulum subsystem. δ here is used to denote the small neighbour around the home position. With the same δ, denote the LTI state-space model representation within the neighbour:

$$\dot{x}(t) = A_\delta x(t) + B_\delta u(t) \tag{B.10}$$

$$y(t) = C_\delta x(t) + D_\delta u(t) \tag{B.11}$$

where $y(t)$ is the measurement output.
The linearized matrix A_δ and B_δ can be obtained by calculating the Jacobian at the home position :

$$A_\delta = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_4} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_4}{\partial x_1} & \cdots & \frac{\partial f_4}{\partial x_4} \end{bmatrix} \Bigg|_{(\theta=\alpha=\dot{\theta}=\dot{\alpha}=0)} \tag{B.12}$$

$$B_\delta = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} \\ \vdots \\ \frac{\partial f_4}{\partial u_1} \end{bmatrix} \Bigg|_{(\theta=\alpha=\dot{\theta}=\dot{\alpha}=0)} \tag{B.13}$$

where $f_1(t)$, $f_2(t)$, $f_3(t)$, $f_4(t)$, $x_1(t)$, $x_2(t)$, $x_3(t)$, $x_4(t)$ and $u(t)$ are defined in Eq. (B.1), (B.2), (B.3), (B.4), (B.9) and (B.10)
According to the lab device manual [15], the 2 DoF inverted pendulum device only has measurement on two angles θ and α, thus C_δ and D_δ are defined as:

$$C_\delta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \; D_\delta = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{B.14}$$

Plug in the parameters from Table B.1 into Eq. (B.12) and Eq. (B.13), the continuous LTI state-space model for the 1 DoF inverted pendulum is derived:

$$A_\delta = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 12.8209 & -7.7744 & 0 \\ 0 & 52.8828 & -4.5648 & 0 \end{bmatrix} \tag{B.15}$$

$$B_\delta = \begin{bmatrix} 0 \\ 0 \\ 14.4784 \\ 8.5012 \end{bmatrix} \tag{B.16}$$

B.2.3 Discrete LTI State-Space Model

There are two reasons that the continuous state-space model should be discretized before we move on to the controller design. First, it is common to use digital controllers today which run in discrete time. Second and more importantly, the proposed subspace controller in this paper is based on the assumption of the existence of a discrete LTI state-space model.
A discrete LTI state-space model takes the following form:

$$x(k+1) = A_d x(k) + B_d u(k) \tag{B.17}$$

$$y(k) = C_d x(k) + D_d u(k) \tag{B.18}$$

where $k \in \mathbf{N},\ k \geq 0$, A_d, B_d, C_d and D_d are with the same dimension as A_δ, B_δ, C_δ and D_δ respectively.
With the help of a continuous LTI state-space system model in Eq. (B.10), a discrete LTI state-space model in Eq. (B.17) can be obtained by discretizing Eq. (B.10) with a certain sampling time T_s and a certain integral approximation method.
To select an appropriate sampling frequency, observing the frequency response of the continuous LTI state-space model is helpful. Fig. B.1 illustrates the amplitude of the sigma plot for system showed in Eq. (B.10) and (B.11).

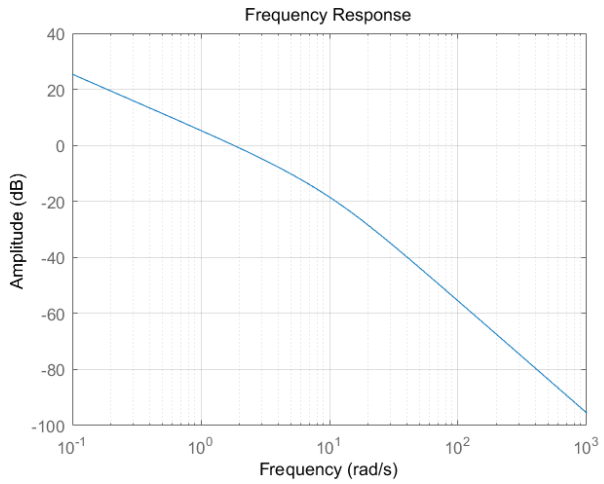

Figure B.1: The Amplitude of the sigma plot for the 1 DoF inverted pendulum model

From Fig. B.1, we can observe that for frequency $\omega_c = 20rad/s$, the corresponding amplitude in dB is zero. This frequency ω_c is called the "crossover frequency".

Although Nyquist–Shannon sampling theorem [17] suggests the minimal sampling frequency to be $2\omega_c$, it is practical in engineering problems to select the minimal sampling frequency to be $\omega_s^* = 10\omega_c$ to guarantee a good dynamic approximations of the discretized system and to provide convenience for further controller design. Hence, we can compute the minimal sampling frequency f_s^* (in Hz) from the following equation:

$$f_s^* = \frac{\omega_s^*}{2\pi} = \frac{10\omega_c}{2\pi} \approx 31.84\ Hz \tag{B.19}$$

which gives the maximal sampling time to be:

$$T_s^* = \frac{1}{f_s} \approx 0.03\ s$$

In the following project, the actual sampling time used is:

$$T_s = 0.02\ s \tag{B.20}$$

Choosing the zero-order-hold (ZOH) approximation method, the following discrete state-space model for the 1 DoF inverted pendulum can be yield:

$$A_d = \begin{bmatrix} 1 & 0.0024 & 0.0185 & 0 \\ 0 & 1.0105 & -0.0009 & 0.0201 \\ 0 & 0.2383 & 0.8559 & 0.0024 \\ 0 & 1.0502 & -0.0849 & 1.0105 \end{bmatrix} \tag{B.21}$$

$$B_d = \begin{bmatrix} 0.0028 \\ 0.0016 \\ 0.2683 \\ 0.1580 \end{bmatrix} \tag{B.22}$$

$$C_d = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},\ D_d = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{B.23}$$

with states:

$$x(k) = \begin{bmatrix} x_1(k) & x_2(k) & x_3(k) & x_4(k) \end{bmatrix}^T = \begin{bmatrix} \theta(k) & \alpha(k) & \dot{\theta(k)} & \dot{\alpha(k)} \end{bmatrix}^T \tag{B.24}$$

and input:

$$u(k) = \delta V_m(k)$$

B.3 State Estimate — Kalman Filter Design

In the manual[15], it is mentioned that only angles θ_x, θ_y, α_x and α_y are measured by embedded encoders. This fact indicates that in each 1 DoF inverted pendulum state-space model, only two of all four states needed for the state-space model in Eq. (B.24) can be measured. Thus a state estimator is necessary to reconstruct state $\dot{\theta}(k)$ and $\dot{\alpha}(k)$. Kalman Filter is a good candidate estimator as is able to provide an unbiased estimate for linear systems. Detailed information about Kalman filter is given in [18]. Here only a brief introduction of the discrete Kalman Filter that is used in this project is presented.

B.3.1 Observability of the Discrete State-Space Model

Before any observer of the system is setup, the observability needs to be checked first. For a discrete LTI state-space model, the observability matrix O_b is defined as:

$$O_b = \begin{bmatrix} C_d \\ C_d A_d \\ \vdots \\ C_d A_d^{n_x-1} \end{bmatrix} \tag{B.25}$$

where n_x is the dimension of the state vector of the model. If $Rank(O_b) = n_x$, the discrete state-space system is said to be observable.
For the specific 1 DoF inverted pendulum model in this project, plug in the result from Eq. (B.21) and (B.23), the rank of the observability matrix is 4, which indicates the linearized inverted pendulum system is observable.

B.3.1.1 Discrete Time Delayed Kalman Filter

A discrete time "delayed" Kalman Filter is implemented in this project. Indeed, discrete time delayed Kalman Filters are widely used for discrete time controller design problems.
To model the dynamics of the inverted pendulum system, consider the following LTI discrete state-space representation with additional noise components:

$$x(k+1) = A_d x(k) + B_d u(k) + N v(k) \tag{B.26}$$

with measurement:

$$y(k) = C_d x(k) + D_d u(k) + w(k) \tag{B.27}$$

where

$$v(k) = \begin{bmatrix} v_1(k) & \cdots & v_4(k) \end{bmatrix}^T$$

is a white noise signal vector for the process noise with its covariance denoted as R_v. Matrix

$$N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.28}$$

is the noise-to-state channel and

$$w(k) = \begin{bmatrix} w_1(k) & w_2(k) \end{bmatrix} \tag{B.29}$$

is also a white noise signal vector for the measurement noise with its covariance denoted as R_w. A delayed discrete time Kalman Filter takes the following form:

$$\hat{x}(k+1) = A_d \hat{x}(k) + B_d u(k) + \bar{L}(y(k) - C\hat{x}(k)) \tag{B.30}$$

where $\hat{x}(k)$ denotes the estimate of state in Eq. (B.24). Kalman gain $\bar{L}$ can be give by the following equation:

$$\bar{L} = A\bar{P}C^T(R_w + C\bar{P}C^T)^{-1} \tag{B.31}$$

where $\bar{P}$ is a positive definite matrix calculated by:

$$\bar{P} = A\bar{P}A^T + NR_vN - A\bar{P}C^T(R_w + C\bar{P}C^T)^{-1}C\bar{P}A^T \tag{B.32}$$

In practice, covariance matrix R_v and R_w act as tuning parameters to set the "trust level" between the model prediction and the measurement correction. By setting R_v larger than R_w, the calculated $\bar{L}$ will lead to the state estimate closer to the prediction made from the state-space model and vice versa.

B.4 LQ Controller Design

In this section, an LQ controller is designed for the discrete time state-space model to act as the initial controller for the closed-loop indentification of the Markov Parameters of the inverted pendulum system.

B.4.1 Controllability of the Discrete State-Space Model

Similar to the observability, before a controller is designed, we need to check if the system is controllable. The controllability matrix for the discrete time state-space model is defined as:

$$\mathcal{C} = \begin{bmatrix} B_d & A_dB_d & \cdots & A_dB_d^{n_x-1} \end{bmatrix} \tag{B.33}$$

where n_x is the dimension of the state vector. A discrete time state-space model is said to be controllable if the rank of the controllability matrix is equal to n_x.
Plug in the result from Eq. (B.21) and (B.22), the rank of the controllability matrix is 4, which indicates the linearized inverted pendulum system is controllable.

B.4.2 Design of the Linear Quadratic (LQ) Controller

In [19], detailed information of the LQ controller design can be found. In this section, only a brief introduction of a discrete time LQ controller design which is implemented on the inverted pendulum system is introduced.
In order to ensure the excitation for the identification experiment to be carried out around the exact home position, an additional integrator is used in designing the initial LQ controller.
This is done through augmenting the state shown in Eq. (B.9) to be:

$$x_{aug}(t) = \begin{bmatrix} \delta\theta(t) & \delta\alpha(t) & \delta\dot{\theta}(t) & \delta\dot{\alpha}(t) & \int \delta\theta(t) \end{bmatrix}^T \tag{B.34}$$

and thus the corresponding continuous augmented state-space model matrices will become:

$$A_{aug} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 12.8209 & -7.7744 & 0 & 0 \\ 0 & 52.8828 & -4.5648 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B.35}$$

$$B_{aug} = \begin{bmatrix} 0 \\ 0 \\ 14.4784 \\ 8.5012 \\ 0 \end{bmatrix} \tag{B.36}$$

$$C_{aug} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \; D_{aug} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{B.37}$$

This continuous state-space model with integration of the state θ will then be discretized with the same process as is shown in section B.2.3.
Having the augmented discrete state-space model, an LQ controller considers the following infinite horizon performance index function:

$$J = \frac{1}{2} \sum_{j=0}^{\infty} (x_{aug}^T(j) Q_x x_{aug}(j) + u^T(j) Q_u u(j)) \tag{B.38}$$

while in J, $u(j)$ is the decision variable (input to be designed). Q_x and Q_u are positive definite matrices acting as the penalty weights on $x(j)$ and $u(j)$ respectively. Q_x and Q_u are tuning parameters for the LQ controller design. In the experiment, the following values are used:

$$Q_x = \begin{bmatrix} 900 & 0 & 0 & 0 & 0 \\ 0 & 900 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \; Q_u = 3.5$$

If Q_x and Q_u are set, a positive semi-definite matrix P can be calculated based on the following Discrete-time Algebraic Riccati Equation (DARE):

$$A^T P A - P - A^T P B (Q_u + B^T P B)^{-1} B P A + Q_x = 0 \tag{B.39}$$

To clarify, A and B in Eq. (B.39) denotes the corresponding discrete time state-space model matrices derived by discretizing the augmented continuous state-space model in section B.4.2. This re-definition also applies to Eq. (B.40) and (B.41).
And the optimal LQ gain $\bar{K}$ can be obtained by:

$$\bar{K} = (B^T P B + Q_u)^{-1} B^T P A \tag{B.40}$$

In this project, the reference signal for the control problem will be a zero vector as the home position is set as the equilibrium point. Thus the closed-loop discrete time state-space model for the decoupled 1 DoF inverted pendulum system under the LQ gain can be derived by setting $u(k) = -\bar{K} x(k)$ in Eq. (B.17):

$$x_{aug}(k+1) = (A - B\bar{K}) x_{aug}(k) \tag{B.41}$$

This closed-loop state-space model is used as the initial controller in this project.

C

A brief introduction on Linear Matrix Inequality

Definition 1 ***Linear Matrix Inequality*** *is a matrix inequality takes the form:*

$$F(x) \triangleq F_0 + \sum_{i=1}^{m} x_i F_i \succ 0 \tag{C.1}$$

where $x = [x_1, \cdots, x_m]^T \in \mathbf{R}^m$ *is vector variables,* $F_i = F_i^T \in \mathbf{R}^{n \times n}$ *for* $i = 1, \cdots, m$ *are given, symmetric matrices.*

The set $\{x|F(x) \succ 0\}$ is convex and does not necessarily have smooth boundaries. Strict inequality '$\succ$' (positive definite) is used mostly for convenience while inequalities of form $F(x) \succeq 0$ can also be handled.

Multiple LMIs can be expressed as a single LMI. For example, LMIs as $F_1(x) \succ 0, \cdots, F_n(x) \succ 0$ is equivalent to $\mathbf{diag}(F_1(x), \cdots, F_n(x)) \succ 0$ where **diag** is the operator to formulate a diagonal matrix.

For nonlinear (convex) matrix inequalities, it is also possible to create an equivalent LMI form using Shur Complements as stated in Lemma 1, section 2.3.4. The idea is as follows: An LMI of form:

$$F(x) = \begin{bmatrix} Q(x) & S(x) \\ S^T(x) & R(x) \end{bmatrix} \succ 0 \tag{C.2}$$

where $Q(x) = Q^T(x)$, $R(x) = R^T(x)$ and $S(x)$ is an affine function of x, is equivalent to:

$$R(x) \succ 0,\ Q(x) - S(x)R(x)^{-1}S(x)^T \succ 0 \tag{C.3}$$

It is also possible to treat problems with matrices as the variables in the problem. Under this circumstance, it is not necessary to write explicitly in the form "$F(x) \succ 0$". Instead, one can simply make clear which matrices are the variables. One of the related example can be the following Lyapunov quadratic inequality:

$$A^T P + PA + PBR^{-1}B^T P + Q \prec 0 \tag{C.4}$$

where A,B,$Q = Q^T$,$R = R^T \succ 0$ are given matrices of appropriate sizes; $P = P^T$ is the variable. Eq. (C.4) can be expressed as the LMI term:

$$\begin{bmatrix} -A^T P - PA - Q & PB \\ B^T P & R \end{bmatrix} \succ 0 \tag{C.5}$$

A bonus of Eq. (C.5) is that it reveals the quadratic inequality (C.4) is convex in P.

There are several convex and quasi-convex problems that can be efficiently solved by using LMI. One of such problems is the generalized eigenvalue problem (GEVP). The following example is an equivalent alternative form for a GEVP is:

$$\begin{aligned} &\text{minimize} && \lambda \\ &\text{subject to} && A(x,\lambda) > 0 \end{aligned} \tag{C.6}$$

where $A(x, \lambda)$ is affine in x for fixed λ and affine in λ for fixed x. This alternative form of GEVP is indeed the problem structure used in the proposed data-driven controller design in Chapter 2.
The advantage of using LMI to solve such GEVP problem is the tractability, in the sense that:

- The necessary and sufficient optimality condition can be immediately written down.
- There is a well-developed duality theory.
- LMI problems can be solved in polynomial time.

To solve LMI problems, there are efficient and powerful algorithms. Such algorithms can prove that the global optimum has been obtained within some pre-specified accuracy, or report the problem is infeasible. Algorithms such as ellipsoid algorithm and interior point methods are commonly used in solving LMI problems in practice [25].

Bibliography

[1] Åström K J, Hägglund T. PID controllers: theory, design, and tuning[J]. 1995.

[2] Marco A, Hennig P, Bohg J, et al. Automatic LQR tuning based on Gaussian process global optimization[C] Robotics and Automation (ICRA), 2016 IEEE International Conference on. IEEE, 2016: 270-277.

[3] Wojsznis W, Gudaz J, Blevins T, et al. Practical approach to tuning MPC** Based on Practical Approach to Tuning MPC by Wojsznis, Gudaz, Mehta, and Blevins, published in the Proceedings of the ISA 2001 Conference, September 10-13, 2001, Houston, TX [11][J]. ISA transactions, 2003, 42(1): 149-162.

[4] Bohlin T P. Practical grey-box process identification: theory and applications[M]. Springer Science & Business Media, 2006.

[5] Ljung L. Black-box models from input-output measurements[C] Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE. IEEE, 2001, 1: 138-146.

[6] I. Houtzager, J.W. van Wingerden, M. Verhaegen (2009) "VARMAX-based closed-loop subspace model identification", in proceedings of The 48th IEEE Conference on Decision and Control, Shanghai, China, December 2009.

[7] Chiuso A. The role of vector autoregressive modeling in predictor-based subspace identification[J]. Automatica, 2007, 43(6): 1034-1048.

[8] Simon D. Optimal state estimation: Kalman, H infinity, and nonlinear approaches[M]. John Wiley & Sons, 2006.

[9] Favoreel W, De Moor B, Van Overschee P, et al. Model-free subspace-based LQG-design[C] American Control Conference, 1999. Proceedings of the 1999. IEEE, 1999, 5: 3372-3376

[10] Dong J, Verhaegen M, Holweg E. Closed-loop subspace predictive control for fault tolerant MPC design[J]. IFAC Proceedings Volumes, 2008, 41(2): 3216-3221.

[11] Dong J. Data driven fault tolerant control: a subspace approach[D]. TU Delft, Delft University of Technology, 2009.

[12] Kulcsár B, Dong J, van Wingerden J W, et al. LPV subspace identification of a DC motor with unbalanced disc[J]. IFAC Proceedings Volumes, 2009, 42(10): 856-861.

[13] Dong J, Kulcsár B, Verhaegen M. Subspace based fault detection and identification for LPV systems[J]. IFAC Proceedings Volumes, 2009, 42(8): 336-341.

[14] Dong J, Kulcsár B, Van Wingerden J W, et al. Closed-loop subspace Predictive Control for Linear Parameter Varying systems (i)-the nominal case[C] Control Conference (ECC), 2009 European. IEEE, 2009.

[15] J Apkarian, H Lacheray, M Lévis, LABORATORY GUIDE: 2 DOF Inverted Pendulum Experiment for MATLAB®Simulink®Users, Quanser, 2012.
[16] Bubnicki Z. Modern control theory[M]. Springer Science & Business Media, 2005.
[17] Jerri A J. The Shannon sampling theorem—Its various extensions and applications: A tutorial review[J]. Proceedings of the IEEE, 1977, 65(11): 1565-1596.
[18] Welch G, Bishop G. An introduction to the Kalman filter[J]. 1995.
[19] Kwakernaak H, Sivan R. Linear optimal control systems[M]. New York: Wiley-interscience, 1972.
[20] Ljung L. System identification[M]. John Wiley & Sons, Inc., 1999.
[21] Dong J, Verhaegen M. Cautious H2 optimal control using uncertain markov parameters identified in closed loop[J]. Systems & Control Letters, 2009, 58(5): 378-388.
[22] Saberi A, Stoorvogel A A, Sannuti P. Filtering theory: with applications to fault detection, isolation, and estimation[M]. Springer Science & Business Media, 2007.
[23] Wan Y, Keviczky T, Verhaegen M, et al. Data-driven robust receding horizon fault estimation[J]. Automatica, 2016, 71: 210-221.
[24] Kulcsár B, Verhaegen M. Robust cautious data driven control with guaranteed mean square stability[C] Decision and Control (CDC), 2010 49th IEEE Conference on. IEEE, 2010: 2276-2281.
[25] Boyd S, El Ghaoui L, Feron E, et al. Linear matrix inequalities in system and control theory[M]. Society for industrial and applied mathematics, 1994.
[26] Lofberg J. YALMIP: A toolbox for modeling and optimization in MATLAB[C] Computer Aided Control Systems Design, 2004 IEEE International Symposium on. IEEE, 2004: 284-289.