Master's Thesis

Czech
Technical
University
in Prague

F2

Faculty of Mechanical Engineering
Department of Instrumentation and Control Engineering

Control of flexible mechanical set-up with time delays in the feedback

Bc. David Osta

June 2017
Supervisor: Prof. Ing. Tomáš Vyhlídal, PhD

ČESKÉ
VYSOKÉ
UČENÍ
TECHNICKÉ
V PRAZE

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Osta** Jméno: **David** Osobní číslo: **396988**

Fakulta/ústav: **Fakulta strojní**

Zadávající katedra/ústav: **Ústav přístrojové a řídicí techniky**

Studijní program: **Strojní inženýrství**

Studijní obor: **Přístrojová a řídicí technika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Řízení flexibilní mechanické soustavy se zpožděním ve zpětné vazbě

Název diplomové práce anglicky:

Control of flexible mechanical set-up with time delays in the feedback

Pokyny pro vypracování:

- Proveďte rešerši na témata řízení flexibilních systémů pomocí tvarovačů signálu a řízení systémů s komunikačním zpožděním ve zpětné vazbě
- Vytvořte matematický model laboratorní úlohy s pružně propojenými vozíky a kyvadlem a identifikujte parametry modelu.
- Navrhněte řídicí systém s inverzním tvarovačem pro potlačení flexibilních módů kyvadla s uvažováním komunikačního zpoždění ve zpětné vazbě.
- Experimentálně ověřte navržený systém řízení
- Proveďte souhrn dosažených výsledků.

Seznam doporučené literatury:

Jméno a pracoviště vedoucí(ho) diplomové práce:

prof. Ing. Tomáš Vyhlídal Ph.D., U12110.3

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **19.04.2017** Termín odevzdání diplomové práce: **16.06.2017**

Platnost zadání diplomové práce: ____________

Podpis vedoucí(ho) práce Podpis vedoucí(ho) ústavu/katedry Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

19.4.2017

Datum převzetí zadání Podpis studenta

Acknowledgement

I would like to give special thanks to my supervisor Prof. Tomáš Vyhlídal, who pointed me in the right direction during my time in need, for his general support and encouragement throughout the course of this thesis.

I would also like to thank Ing. Milan Anderle and Ing. Jaroslav Bušek for their time and advices.

Finally I would like to thank to my parents, for their support throughout my studies.

This work was supported by CTU grant n. SGS17/176/OHK2/3T/12.

Declaration

I hereby declare that I worked out the presented thesis independently and I quoted all the sources used in this thesis in accord with Methodical instructions about ethical principles for writing academic thesis.

In Prague, 6th June 2017

....

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 6. Června 2017

....

Abstrakt / Abstract

Cílem této práce je navrhnout řízení laboratorní soustavy skládající se z aktivně poháněného vozíku s pasivně připojeným vozíkem a kyvadelem. Síťová řídící architektura je použita pro přenášení řídícího a zpětnovazebního signálu prostřednictvím internetového komunikačního protokolu. Pro laboratorní soustavu byl odvozen a identifikován matematicky model za účelem návrhu řízení soustavy. Pro potlačení vibrací flexibilních mechanických části soustavy byl použit nový typ tvarovače signálu s distribuovaným dopravním zpožděním, jenž byl v inverzní formě zapojený ve zpětné vazbě. Proporcionálně derivační regulátor pro řízení polohy vozíku je navržen s ohledem na dopravní zpoždění způosbené přenosem signálů internetem a tvarovačem signálů ve zpětné vazbě.

Klíčová slova: Potlačvání vibrací, Tvarovač signálů, Řízení přes internet, Dopravní zpoždění, Identifikace systému, Kompenzace tření, Stavová formulace

The goal of this thesis is to design a control system of a laboratory set-up that consists of an active cart with a passively connected cart and pendulum. The networked control architecture is used to transmit a control signal together with a feedback signal via Internet communication protocols. A mathematical model of the laboratory set-up is derived and subsequently identified for the control design purposes. A novel type of zero vibration shaper with distributed time-delay is applied in an inverse form in the feedback path in order to suppress oscillations of flexible parts. A classical proportional derivative controller is designed with respect to introduced time-delays caused by transmission and signal shaper.

Keywords: Vibration Suppression, Signal Shaper, Networked Control System, Time Delay, System Identification, Friction Compensation, State-Space Representation

Contents

1 Introduction 1
1.1 Motivation 1
1.2 Aims of Thesis 2
2 Modern Control Theory 3
2.1 State-Space Representation 3
2.1.1 State Space 3
2.1.2 State Space Equations 3
2.2 Controllability and Observability 4
2.3 Pole-Placement 5
2.4 State Observers 6
2.4.1 Observer based Friction Compensation 8
3 Time Delay Systems 9
3.1 Time Delay 9
3.2 Linear Time Delay Systems 10
3.3 Stability of Time-Delay Systems 12
4 Signal Shapers 14
4.1 Zero Vibration Shaper 16
4.2 Robust Zero Vibration Shapers 16
4.3 Distributed Zero Vibration Shaper 17
4.4 Shapers in Feedback Architecture 18
4.4.1 Implementation of Inverse Shaper 21
5 Internet-based Control Systems 24
5.1 Internet-Based Control Architectures 26
5.2 Networked Control System 27
5.3 Time Delay in Packet-Switched Networks 28
5.3.1 Processing Delay 29
5.3.2 Queuing Delay 29
5.3.3 Transmission Delay 30
5.3.4 Propagation Delay 30
5.3.5 Basic Communication Protocols of Transport Layer 31
6 Controlled System 32
6.1 System Description 32
6.2 Mathematical Model 33
6.2.1 Derivation of Equations of Motion 35
6.2.2 Linearization of Mathematical Model 37
6.2.3 Model in State-Space Representation 37
6.3 Compensation of System's Nonlinearities 39
6.3.1 State Observer for Friction Compensation 41
6.4 System Identification 42
6.4.1 Identification of Input Delay 48
6.4.2 Design of State Observer 48
6.5 Identification of oscillation mode of flexible subsystem 50
6.5.1 Analytical Identification 50
6.5.2 Experimental Identification 51
7 Design of Internet Control Architecture 54
7.1 Time Delay Identification 56
8 Control Design 60
8.1 Shaper Design 60
8.2 Controller Design 63
9 Experimental Results 66
9.1 Measured Results 66
9.2 Simulation results 70
10 Conclusion 71
References 72
A Source Codes 75
A.1 Matlab source code of friction compensation 75
A.2 Matlab source code of delay estimation function 76
A.3 Simulink model used at the remote computer 76
A.4 Simulink model used at the local computer 77
B Abbreviations and Symbols 78
B.1 Abbreviations 78
B.2 Symbols 79

Tables / Figures

5.1. Pros and cons of possible links between the Internet and control levels . . . 25
6.1. identified system's parameters . 42
6.2. identified oscillation mode of the pendulum . . . 52
7.1. identified time delay . . . 56
8.1. Designed shapers's parameters . . . 63
8.2. Designed controller's parameters . . . 63
8.3. Stability margins of closed loop system without inverse shaper . . . 64
8.4. Stability margins of closed loop system with inverse shaper . . . 64
9.1. Table of controller performance criteria for the controlled position of the cart . . . 69
9.2. Table of controller performance criteria for the passively controlled pendulum's deflection . . . 69

2.1. Block diagram of the Linear Time Invariant system represented in state space . . . 4
2.2. Block diagram of system and full-order state observer . . . 7
2.3. Block diagram of the friction observer based compensation scheme . . . 8
3.1. Delay distribution functions . . . 10
3.2. Graphical representation of the state and the initial conditions of the Time Delay Systems . . . 11
3.3. Example of contour mapping technique for roots finding of characteristic equation by use of QPMR algorithm . . . 13
4.1. Complex conjugate poles depicted in the complex plane . . . 15
4.2. Feedback loop with shaper as reference command filter . . . 18
4.3. Feedback loop architecture with an inverse shaper . . . 19
4.4. Coupled case of feedback loop architecture with an inverse shaper . . . 19
4.5. Spectrum of zeros of selected zero vibration shapers . . . 21
4.6. Frequency response of different Zero Vibration shapers . . . 22
4.7. Implementation of inverse vibration shapers . . . 23
5.1. The layers of Information architecture . . . 25
5.2. Control architecture with the remote controller . . . 26
5.3. Control architecture with remotely sent input signal to the local controller . . . 27
5.4. Bilateral control architecture . . 27
5.5. Block diagram of the networked control system . . . 27
5.6. The nodal delay at router A . . . 28
5.7. Dependence of average queuing delay on Traffic Intensity . . 30
6.1. Photo of the controlled system . 32

6.2. The scheme of the whole system . . . 33
6.3. Free body diagrams of the controlled system . . . 34
6.4. Graphical representation of the classical friction model . . . 39
6.5. Approximation of the nonlinearity of the transfer of the control signal to the control force . . . 40
6.6. Designed scheme to compensate nonlinearities . . . 41
6.7. Pulse response of the controlled system - 2V . . . 43
6.8. Pulse response of the controlled system - 3V . . . 44
6.9. Pulse response of the controlled system - 4V . . . 45
6.10. Pulse response of the controlled system - 5V . . . 46
6.11. Poles of the mathematical model . . . 47
6.12. Input delay for different impulse responses . . . 48
6.13. Comparison of observed outputs and measured values . . . 49
6.14. Mechanical scheme of the cart b and the pendulum p . . . 50
6.15. Poles of the reduced state matrix . . . 51
6.16. Free body diagram of the pendulum and cart . . . 51
6.17. Identification of pendulum's oscillation mode . . . 52
7.1. Signal flow of designed Networked Control System . . . 54
7.2. Block diagram of the designed Networked Control System . . . 55
7.3. Designed GUI for control of Networked Control Systems . . . 55
7.4. Link time-delay for connection established via Wi-Fi . . . 57
7.5. Link time-delay for connection established via Ethernet . . . 57

7.6. Round Trip Time for connection established via Ethernet .. 58
7.7. Effect of initial delay of UDP packets 59
8.1. Spectrum of designed signal shaper 61
8.2. Frequency response of the designed signal shaper 62
8.3. Bode diagram of the open loop configurations 64
8.4. Closed loop spectra for feedback with inverse shaper and communication delays 65
9.1. Comparison of local control with remote control, without inverse shaper 67
9.2. Comparison of local control with remote control, with inverse shaper 68
9.3. Comparison of simulated responses 69
9.4. Comparison of simulated responses 70
A.1. Simulink model used at the remote computer 76
A.2. Simulink model used at the local computer 77

Chapter 1
Introduction

Vibration suppression is an important and extensively studied subject in most engineering fields. In the most cases, vibrations are undesirable phenomena, mainly because they can substantially reduce machine life or even cause life-threating situations, e.g. machine failure. The vibration suppression control techniques can be divided into two major categories, active and passive techniques. The active approaches usually consist of the state feedback controllers, that are designed to dampen system's vibrations at desired frequencies. Whereas, the passive approach is based on the shaping of the reference signal in such a way that the energy supplied to the system does not induce system's vibrations. This technique, also known as Signal Shaping, is studied and applied for a laboratory set-up in this thesis.

Signal shaping has been applied in many control systems with flexible mechanical parts, such as cranes, industrial robots, manipulators, coordinate measuring machines or hard-drive seekers [1], [2]. The concept of signal shaping was first published as a Posicast control, by O.J. Smith in 1950's [3]. Smith's idea was revisited in 1990's by Singer, Seering and Singhose [4], [5], who proposed concepts of zero-vibration (ZV) shapers and developed more robust types of signal shapers. Recently, a novel type of signal shapers with distributed time-delay (DZV) was proposed by Vyhlídal, Kučera and Hromčík [6], [7]. Same authors also proposed a novel control architecture that incorporates inverse form of DZV signal shaper directly into the feedback path. The main benefit of this architecture is possibility to suppress vibrations excited by disturbances, that is not possible to achieve via classical architecture with signal shaper as reference command filter [8], [9]. However, any inverse form of the signal shaper in the feedback path introduces the undesirable time-delayed feedback and this aspect should be taken into account during the control design. In this thesis the novel DZV shaper and the control architecture with an inverse shaper will be studied and experimentally tested on the laboratory set-up with flexible mechanical parts.

1.1 Motivation

The motivation of this thesis is to experimentally verify novel modifications of signal shaping technique proposed in [6], [8] on the laboratory set-up, moreover, the possibility of controlling laboratory set-up remotely via Internet communication protocols will be experimentally tested. The Internet remote control has recently become more appealing in control applications since the infrastructure of the Internet network has rapidly evolved during the last two decades. The signal transmission via Internet network has improved so that its speed and reliability is now sufficient enough for most real-time control applications. It is now even possible to remotely control systems from almost any distances at any time. Remote control is especially convenient for the controlled systems with components that cannot be connected via classical point-to-point cables, however, control over long distances naturally induces undesired communication time-delays that might negatively affect control performance. Another advantage of the remote control is its flexibility, since control algorithm can be simply changed or updated on the remote computer. This comes very handy at situations when control process has different and frequently changed operational modes, for example a crane can operate with different type of loads that changes overall dynamics of the system. The control algorithm can be then simply switched to any specific operational mode.

1.2 Aims of Thesis

In this section the aims of the thesis will be presented and analyzed. The aims contain the whole process of control engineering design from mathematical modeling to system identification, control design, and their MATLAB development, which makes it a perfect topic for a diploma thesis in Control Engineering field. The structure of the thesis was designed from the following aims of thesis.

- **Study the fundamental concepts of modern control theory** : In this introductory part are studied fundamentals concepts of modern control, that are applied for system modeling of the laboratory set-up. Concepts such as state-space representation, controllability, observability and the pole placement method are discussed. This part is covered in Chapter 2.
- **Study the time-delayed systems and Internet based control architectures** : The Internet based control architectures are becoming more popular in real-time control applications. This type of architecture usually introduces a undesirable time-delay phenomena, that can degrade overall control performance. Time-delay systems and Internet based control are briefly studied in Chapters 3 and 5, respectively. Implementation of networked control architecture via program MATLAB/Simulink is presented in Chapter 7.
- **Study the signal shapers technique for vibration suppression of the flexible mechanical parts** : Signal Shaping is an effective method for vibration suppression of flexible mechanical parts. There are many types of Signal Shapers and this thesis focuses only one novel modifications of Zero Vibration shaper. This subject covers Chapter 4.
- **Derive and identify a mathematical model of the controlled system** : For a proper control design a mathematical model of the laboratory-set up has to be derived, linearized and identified. This part also deals with compensation of system's nonlinearities. This part is discussed in Chapter 6.
- **Design control system with an inverse signal shaper in the feedback together with delayed feedback by the transmission delay** : In Chapter 8 is designed controller together with signal shaper to control the laboratory set-up with communication time delays.
- **Experimentally verify designed control system and evaluate obtained results** : In final Chapter 9 are presented and discussed obtained experimental results.

Chapter 2
Modern Control Theory

In this chapter, the fundamental concepts of modern control theory based on time-domain approach for mathematical modeling of physical systems will be represented. Concepts such as state-space representation, controllability, observability together with methods for pole placement and state estimation are discussed.

2.1 State-Space Representation

The time-domain state space representation models a physical system as a set of input, output and state variables represented as vectors that are related by a set of first-order differential and algebraic equations written in matrix form. The state space representation is suitable for modeling and analyzing systems with multiple inputs, outputs and greater number of state variables. The state space representation greatly simplifies the mathematical representation of systems of equations, since the increase in the number of state variables, the number of inputs, or the number of outputs does not increase the complexity of the equations, moreover, models are not restricted by zero initial conditions and are applicable to linear and nonlinear, time invariant or time varying systems, ([10], p.29, p.648).

2.1.1 State Space

The *n-dimensional space* whose coordinate axes consists of *n-state variables* is called a *state space*, ([10], p.30). The state of a dynamic system is represented by a vector of linearly independent state variables, this vector is known as *state vector*. The state variables are the smallest possible subset of system variables that can represent the entire state of the system at any given time. The state of the system for time $t \geq t_0$ can be completely determined from knowledge of the input $u(t)$ for $t \geq t_0$ and given state variables at time $t = t_0$. Therefore, the future of the state is solely dependent on the present state. The state-space representation for a given system is not unique, except that the number of state variables is the same for any different state-space representations of the same system. The dynamic system must involve elements that memorize the values of the input for $t \geq t_1$. Integrators in continuous-time serve as memory devices, the outputs of integrators can be considered as the variables that define the internal state of the dynamic system, ([10], p.29, p.30).

2.1.2 State Space Equations

The linear, time-invariant system (LTI) with multiple-inputs m, multiple-outputs p and n state variables is described by following state equations, ([10], eq.2-14,15)

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t), \tag{2.1}$$

$$\boldsymbol{y}(t) = \mathbf{C}\boldsymbol{x}(t) + \mathbf{D}\boldsymbol{u}(t), \tag{2.2}$$

where $\mathbf{x}(t) \in \Re^n$ is called the state vector, $\mathbf{u}(t) \in \Re^m$ the input vector, $\mathbf{y}(t) \in \Re^p$ the output vector, $\mathbf{A} \in \Re^{n\times n}$ the state matrix, $\mathbf{B} \in \Re^{n\times m}$ the input matrix, $\mathbf{C} \in \Re^{p\times n}$ the output matrix, and $\mathbf{D} \in \Re^{p\times m}$ the direct transmission matrix, ([10], p.31). The state equation (2.1) and the output equation (2.2) can be described by a block diagram shown in Figure 2.1.

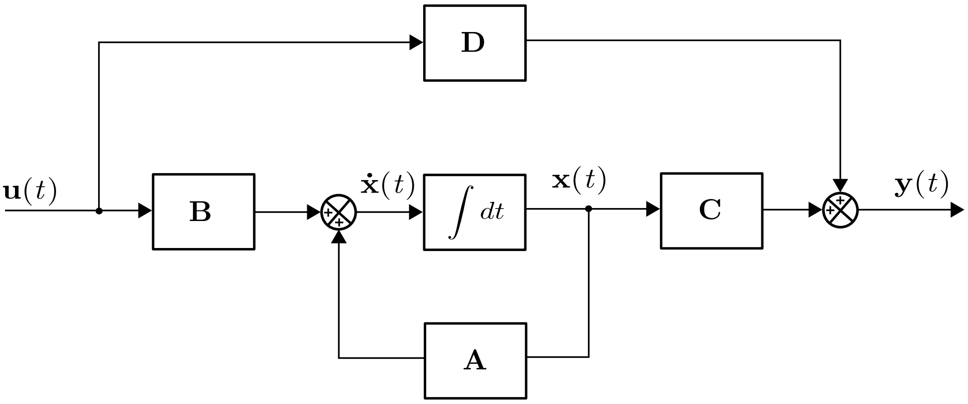


Figure 2.1. Block diagram of the LTI system represented in state space, (taken and modified from [10], Figure 2-14).

The eigenvalues of the state matrix $\mathbf{A}$ are the roots of the characteristic equation

$$M(s) = det(s\mathbf{I} - \mathbf{A}), \tag{2.3}$$

where $\mathbf{I} \in \Re^{n\times n}$ is an identity matrix. The system's stability can be determined by roots' location in the complex plane and stability criterion is defined as follows, ([11], eq. 2.3)

$$\{s_i \in C : \Re(s_i) < 0, M(s) = 0\}. \tag{2.4}$$

2.2 Controllability and Observability

Controllability plays a crucial role in stabilization of unstable systems by feedback or in optimal control problems. A system is said to be controllable at time t_0 if it is possible by means of an unconstrained control vector to transfer the system from any initial state $\mathbf{x}(t_0) = \mathbf{x_0}$ to an arbitrary final state $\mathbf{x}(t_f) = \mathbf{x_f}$ in a finite interval of time $t_f < \infty$. If every state is controllable, then the system is said to be completely state controllable and if the system is completely state controllable it is guaranteed that any state can be reached, however, that does not imply that any state of the system can be maintained for an arbitrary interval of time, ([10], p. 675, 676). A controllable matrix $\mathbf{R}(\mathbf{A}, \mathbf{B}) \in \Re^{n\times n}$ is defined as follows, ([10], eq. 9-55)

$$\mathbf{R}(\mathbf{A},\mathbf{B}) = \left(\mathbf{B} \ \vdots \ \mathbf{AB} \ \vdots \ \ldots \ \vdots \ \mathbf{A^{n-1}B} \right). \tag{2.5}$$

A continuous-time system (2.1) is controllable if the rank of the controllable matrix (2.5) equals to the number of state variables n or if column vectors of a controllable matrix are linearly independent, ([10], p.677)

$$rank(\mathbf{R}(\mathbf{A},\mathbf{B})) = n. \tag{2.6}$$

Partially controllable system has both controllable and uncontrollable modes. Such system is said to be stabilizable if the uncontrolled modes are stable and the unstable modes are controllable. An example of uncontrollable system is a system that has a subsystem that is physically disconnected from the input, ([10], p.688).

The concept of observability is very important for the state feedback control, since all system's state variables are required to construct the control signal. However, in practice some state measurement are often not accessible for direct measurement. Therefore, the concept of observability is very important for estimation of the unmeasurable state variables. The system is said to be completely observable if every state $\mathbf{x}(t_0) = \mathbf{x_0}$ can be determined from the observation of $\mathbf{y}(t) = \mathbf{y}$ over a finite time interval, $t_0 \leq t \leq t_1$, ([10], p.682). If the system is completely observable, then every transition of the state eventually affects every element of the output vector. An observable matrix $\mathbf{O}(\mathbf{A},\mathbf{B}) \in \Re^{n\times n}$ is defined as, ([10], eq. 9-65)

$$\mathbf{O}(\mathbf{A},\mathbf{C}) = \begin{pmatrix} \mathbf{C} \\ \vdots \\ \mathbf{CA} \\ \mathbf{CA^{n-1}} \end{pmatrix}. \tag{2.7}$$

A continuous-time system described by state equations (2.1) and (2.2) is observable if the rank of the observable matrix (2.7) is equal to the number of state variables n, ([10], p.688)

$$rank(\mathbf{O}(\mathbf{A},\mathbf{C})) = n. \tag{2.8}$$

Partially observable system has both observable and unobservable modes. Such system is called *detectable* if the unobservable modes are stable and unstable modes are observable, ([10], p.688).

2.3 Pole-Placement

In the pole-placement design all closed-loop poles may be placed at any desired location in the complex plane, however, placing all closed-loop poles requires availability of all state variables. First the desired closed-loop poles are determined based on the control design requirements such as transient-response, frequency-response, speed or damping ratio, ([10], p.723). Next by choosing an appropriate gain matrix $\mathbf{K} \in \Re^{\mathbf{1\times n}}$, it is possible to force the system to have closed-loop poles at the desired locations, provided that the original system is completely state controllable, ([10], p.723).

For the system described by state equations (2.1) and (2.2) the control signal u is determined by an actual state, ([10], eq. 10-2)

$$u = -\mathbf{K}\cdot\mathbf{x}. \tag{2.9}$$

By substituting Equation (2.9) into (2.1) we receive a closed-loop control system with no input and with target to maintain the zero output, ([10], p.724). This system is described as follows, ([10], eq. 10-3)

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{BK})\mathbf{x}(t), \tag{2.10}$$

$$\tilde{\mathbf{A}} = \mathbf{A} - \mathbf{BK},$$

$$\dot{\mathbf{x}}(t) = \tilde{\mathbf{A}}\mathbf{x}(t). \tag{2.11}$$

The stability and transient response are determined by the eigenvalues of new system's state matrix $\tilde{\mathbf{A}}$ and for properly chosen $\mathbf{K}$ the system (2.11) can be made asymptotically stable. If the system is not completely state controllable, but is stabilizable, then it is possible to make the entire state stable by placing the closed-loop poles at desired locations for controllable modes, ([10], p.728).

The selection of the closed-loop poles is a compromise between the speed of the response of the error vector and the sensitivity to disturbances and measurement noises. That is, if we increase the speed of error response, then the adverse effects of disturbances and measurement noises generally increase, ([10], p.732).

For SISO systems we can determine the feedback gain matrix $\mathbf{K}$ by Ackermann's formula, that is based on the Cayley-Hamilton theorem that states that every square matrix over a commutative ring satisfies its own characteristic equation, ([10], p.730, [11], p.11)

$$\phi(\tilde{\mathbf{A}}) = \tilde{\mathbf{A}}^n + \alpha_1\tilde{\mathbf{A}}^{n-1} + \ldots + \alpha_{n-1}\tilde{\mathbf{A}} + \alpha_n\mathbf{I} = \mathbf{0}, \tag{2.12}$$

Ackermann's forumula is then defined as follows, ([10], eq. 10-18)

$$\mathbf{K} = \begin{pmatrix} 0 & 0 & \ldots & 0 & 1 \end{pmatrix}\mathbf{R}^{-1}\phi(\mathbf{A}). \tag{2.13}$$

2.4 State Observers

A state observer is a subsystem that estimates the state variables based on the measurements of the output control variables. The state observer can be designed if and only if the condition of observability (2.8) is satisfied. The notation $\hat{\mathbf{x}}$ defines the observed state vector that is often used in conjunction with the state feedback to generate the desired control vector. The main purpose of state observer is to reconstruct the state vector of the plant, therefore, the observer's mathematical model is derived from the plant with an additional term that includes the observer error. The observer error is the difference between the measured output and the estimated output, ([10], p. 751, 752).

Compensation of inaccuracies caused by difference between the observer's mathematical model and the physical plant is performed by including an additional correction term $\mathbf{K_e}(y - \mathbf{C}\hat{\mathbf{x}})$ where y is the measured output , $\mathbf{C}\hat{\mathbf{x}}$ is the estimated output and the matrix $\mathbf{K_e}$ is a weighting matrix to the correction term also called as the observer gain. Correction term continuously corrects the model output and improves the performance of the observer. Observer's inputs are the plant output y and the control input u, ([10], p.752). Mathematical model of the observer for system defined by equations (2.1), (2.2) is defined as follows, ([10], eq.10-57)

$$\begin{aligned}\dot{\hat{\mathbf{x}}} &= \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}u + \mathbf{K_e}(y - \mathbf{C}\hat{\mathbf{x}}) \\ &= (\mathbf{A} - \mathbf{K_e}\mathbf{C})\hat{x} + \mathbf{B}u + \mathbf{K_e}y. \end{aligned} \tag{2.14}$$

The observer is said to be full-order if its order of the state is the same as that of the plant. Block diagram of the system and full-order observer is shown in Figure 2.2.

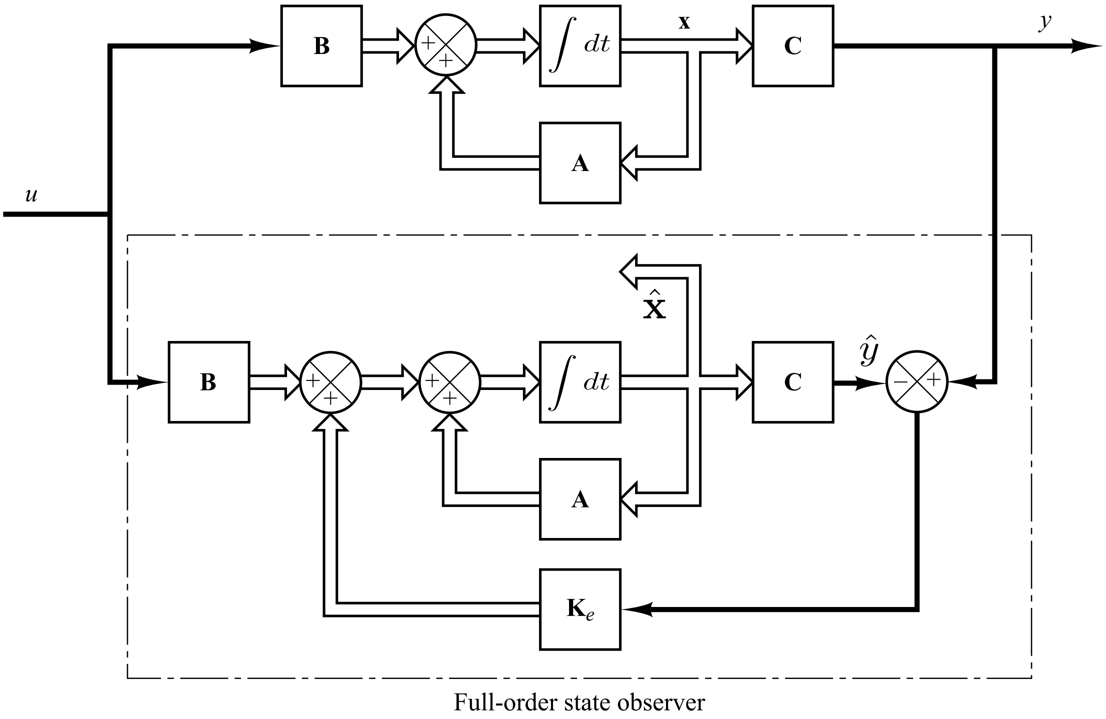


Figure 2.2. Block diagram of system and full-order state observer, when input u and output y are scalars, (taken and modified from [10], Figure 10-11).

Observer's error $\mathbf{e}$ and error's dynamics $\dot{\mathbf{e}}$ are defined by following equations respectively, ([10], eq.10-58,59)

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}, \tag{2.15}$$

$$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{K_e}\mathbf{C})\mathbf{e}. \tag{2.16}$$

If the eigenvalues of matrix $(\mathbf{A} - \mathbf{K_e}\mathbf{C})$ are chosen so that the dynamic behavior of the error vector is asymptotically stable and with sufficient speed of response, then any observer's error vector will converges to zero, however, if there are modeling errors between plant's and observer's matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ then error $\mathbf{e}$ might not approach zero value, ([10], p.754). Therefore we should choose $\mathbf{K_e}$ so that error is acceptably small, converges sufficiently fast and the observer is stable.

Complete observability of the system is the sufficient condition for the observation of the state of the system (2.1), (2.2), then observer gain matrix $\mathbf{K_e}$ can be determined to yield the desired matrix ($\mathbf{A} - \mathbf{K_e}\mathbf{C}$). The design problem of $\mathbf{K_e}$ is mathematically equivalent to the discussed pole-placement problem. In many practical cases, the selection of the best matrix $\mathbf{K_e}$ is the trade-off between speedy response and sensitivity to disturbances and noises, ([10], p.757). Observer gain matrix can be determined by modified Ackermann's formula, ([10], eq. 10-65)

$$\mathbf{K_e} = \phi(\mathbf{A})\mathbf{O}^{-1}\begin{pmatrix} 0 & 0 & \dots & 0 & 1 \end{pmatrix}^T. \tag{2.17}$$

2.4.1 Observer based Friction Compensation

The state observer is often used in compensation schemes that are designed to compensate non-linearities of the system. Main purpose of this technique is to reduce system's nonlinearities so that the controller can be designed by means of linear control theory. The friction often causes non-linear behavior and the knowledge of velocity of moving object is necessary for effective compensation of these non-linearities. To obtain an effective friction compensation it is necessary that the velocity is either measured or estimated with a good resolution and small time delay, ([12], p.23).

The idea behind the friction compensation is that the friction force F is estimated by using a friction model that determines a compensation signal u_{comp} which is added to the control signal u to compensate the estimated friction force $\hat{F}$, however, it is important to ensure that the friction force is not overcompensated, since its overcompensation could destabilize the overall system, ([12], p.23). A practical approach how to prevent friction overcompensation is to scale down the estimated friction force $\hat{F}$ by a value that should be determined through experimental analysis, ([13], p.54). One of the possible compensation scheme is shown in Figure 2.3.

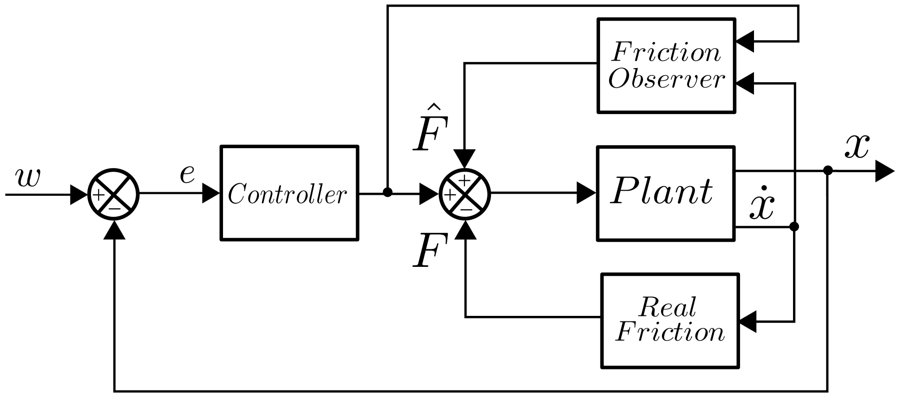


Figure 2.3. Block diagram of one of the possible friction observer based compensation scheme, (taken and modified from [12], Figure 13).

Chapter 3
Time Delay Systems

In many real physical systems the past events influence the future results, therefore, the mathematical model in form of classical state-space representation might be in some cases insufficient because the major assumption of such model is that the future state is independent of the past. Time delay systems (TDSs) represent dynamics of real plant more accurately since time-delay phenomena occurs in many systems either in the state, the control input, or the measurement. Considerable time-delay frequently occurs in systems with heat transfer, chemical processes, or remote control, ([14], p. 10).

The cost of a more authentic mathematical model is more complicated system analysis and control design. In general, every real control system operates in the presence of time-delays, due to the time it takes to acquire the information needed for decision-making (communication delay), to create control decision (computation delay), and to execute these decisions (signal conversion delay), ([15], p. 1). Time-delay is often a source of instability and this is especially true for feedback control systems, where the time-delay always worsen control conditions and quality of the control process , ([14], p. 10).

The time-delay is mathematically represented as a time shift, therefore, TDSs are usually modeled by means of differential equations with an deviating argument, these differential equations are also known as *functional differential equations* (FDEs). FDEs are infinite-dimensional and allow us to model systems that are dependent on the present and the past events, ([16], p.2).

3.1 Time Delay

A general form of the time delay can be described by using the Stieltjes integral, ([17–18], p.72, p.257)

$$r(t) = \int_0^T u(t-\tau)\, dh(\tau), \tag{3.1}$$

with the L-transform form, considering zero initial conditions

$$R(s) = \int_0^\infty e^{-s\tau} dh(\tau) \cdot X(s) = H(s)X(s), \tag{3.2}$$

where $u \in \Re$ and $r \in \Re$ are the input and output, respectively. $T \in \Re$ is the delay length and the delay distribution $h(\tau)$ is defined in the time domain as follows, ([17], p.72)

$$h(\tau) = \begin{cases} 0, & \tau < 0 \\ h_d(\tau), & \tau \in [0, T] \\ 1, & \tau > T \end{cases}, \tag{3.3}$$

the function $h_d(\tau)$ describes the delay distribution over the time interval $\tau \in [0, T]$, which can be either continuous or discontinuous, ([17], p.72). Some of the frequently used delay distribution functions are the lumped delay (LD) and, the equally distributed delay (EQD). The LD is defined as follows, ([18], p.257)

$$h_d(\tau) = H_{hs}(\tau - T), \tag{3.4}$$

$$H(s) = e^{-sT}, \tag{3.5}$$

where $H_{hs}(.)$ denotes the Heaviside step function. While the EQD has following form, ([18], p.257)

$$h_d(\tau) = \frac{1}{T}\tau, \tag{3.6}$$

$$H(s) = \frac{1 - e^{-sT}}{sT}, \tag{3.7}$$

the combination of the LD and the EQD is given as follows, ([9], p.2052)

$$h_d(\tau) = \frac{1}{(1-\alpha)T}(\tau - \alpha T), \tag{3.8}$$

$$H(s) = \frac{1}{(1-\alpha)T}\frac{e^{-s\alpha T} - e^{-sT}}{sT}, \tag{3.9}$$

where the parameter $\alpha \in [0, 1)$ determines the ratio between the length of the lumped delay and the overall delay T ([9], p.2052). A graphical representations of functions (3.4), (3.6) and (3.8) are shown in Figure 3.1.

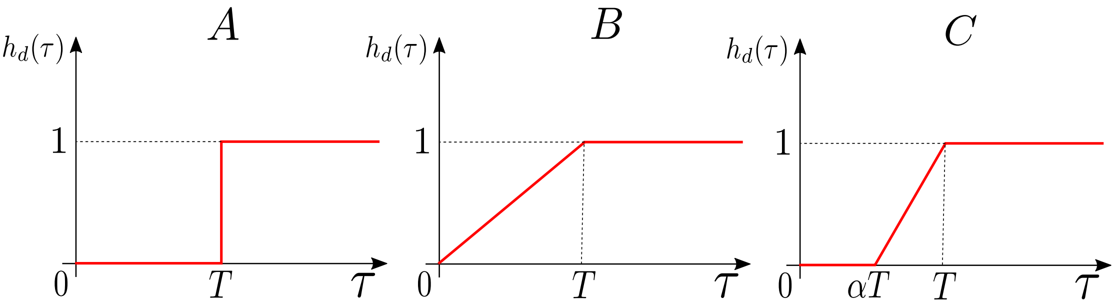


Figure 3.1. Delay distribution functions, A - the LD, B - the EQD, C - the combination of LD and EQD, (taken and modified from [7], Fig. 1).

3.2 Linear Time Delay Systems

The state-space representation of a linear system (2.1), with the time delays in any state accumulation, in the input and in the output of the retarded type is defined as follows, ([17], p.75, [19], p.27)

$$\dot{\boldsymbol{x}}(t) = \int_0^T d\mathbf{A}(\tau)\boldsymbol{x}(t-\tau) + \int_0^T d\mathbf{B}(\tau)\boldsymbol{u}(t-\tau), \tag{3.10}$$

$$\boldsymbol{y}(t) = \int_0^T d\mathbf{C}(\tau)\boldsymbol{x}(t-\tau) + \int_0^T d\mathbf{D}(\tau)\boldsymbol{u}(t-\tau), \tag{3.11}$$

where the argument of the state, the input and the output variables are shifted by a delay τ. The upper limit of integration T is the maximal system delay. The state of the system in the state-space representation (2.1) is defined by a vector of state variables $\mathbf{x}(t)$. However, the state of TDSs (3.10), is given by function segments of the state variables $\mathbf{x_t}$ on a segment of the system history, ([11], 17). The state and the initial conditions of the TDSs, respectively, are now defined as, ([20], p.6)

$$\mathbf{x_t}(\tau) = \mathbf{x}(t+\tau),\ \ \tau \in [-T, 0], \tag{3.12}$$

$$\mathbf{x_0}(\tau) = \mathbf{x}(\tau),\ \ \tau \in [-T, 0]. \tag{3.13}$$

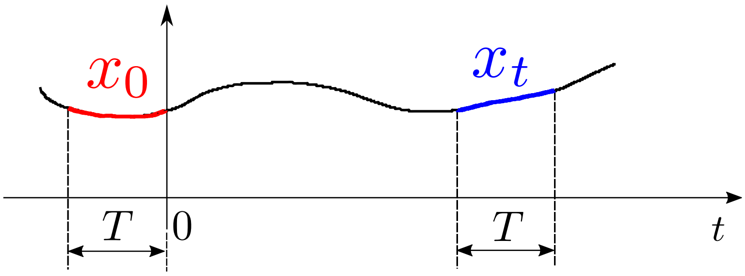


Figure 3.2. Graphical representation of the state and the initial conditions of the scalar TDSs, (taken and modified from [20], p. 6).

The equations (3.10), (3.11) are represented in s domain under zero initial conditions as follows, ([19], p.29)

$$s\mathbf{X}(s) = \mathbf{A}(s)\mathbf{X}(s) + \mathbf{B}(s)\mathbf{U}(s), \tag{3.14}$$

$$\mathbf{Y}(s) = \mathbf{C}(s)\mathbf{X}(s) + \mathbf{D}(s)\mathbf{U}(s), \tag{3.15}$$

for zero initial conditions, we can derive following equation ([19], p.29)

$$\mathbf{Y}(s) = \Big(\mathbf{C}(s)\big(s\mathbf{I} - \mathbf{A(s)}\big)^{-1}\mathbf{B}(s) + \mathbf{D}(s)\Big)\mathbf{U}(s) = \mathbf{G}(s)\mathbf{U}(s), \tag{3.16}$$

where $\mathbf{Y}(s)$ is a column vector of the outputs with m outputs, $\mathbf{U}(s)$ is a column vector of the inputs with n inputs and $\mathbf{G}(s)$ is a $m \times n$ dimensional matrix of non-rational transfer functions. Poles of the TDSs are those values of s for which $\mathbf{G}(s)$ tends to infinity, ([19], p.29). The poles can be determined by the following characteristic equation, ([20], p.7)

$$M(s) = det(s\mathbf{I} - \mathbf{A}(s)) = 0, \tag{3.17}$$

the equation (3.17) is a transcendental equation due to the presence of exponential terms $e^{-s\tau}$ in the matrix $\mathbf{A}(s)$ and has an infinite number of roots, set of all roots is also called as the spectrum of TDSs, ([19], p.29). Therefore, the state space of TDSs is infinite-dimensional and the characteristic equation is of quasi-polynomial type as opposed to a polynomial type for simple linear systems. If the spectrum of TDSs is distributed as a finite number of exponential asymptotic chains of roots, whose most dominant roots lie rightmost, then the TDSs is called *Retarded System*. The TDSs is called *Neutral System* if the spectrum distribution has no exponential character and lies between two vertical boundaries $a < \Re(s) < b$, ([11], p. 23). Note that in this case, the neutral time delay model reads as, ([11], p.17)

$$\dot{\boldsymbol{x}}(t) + \int_0^T d\mathbf{H}(\tau)\dot{\boldsymbol{x}}(t-\tau) = \int_0^T d\mathbf{A}(\tau)\boldsymbol{x}(t-\tau) + \int_0^T d\mathbf{B}(\tau)\boldsymbol{u}(t-\tau). \tag{3.18}$$

3.3 Stability of Time-Delay Systems

Since, the characteristic equation $M(s)$ is not longer polynomial but quasi-polynomial, classical algebraic methods, e.g. Routh–Hurwitz stability criterion, for stability analysis cannot be directly applied. However, the stability criterion (2.4)

$$\{s_i \in C : \Re(s_i) < 0, M(s) = 0\}, \tag{3.19}$$

can be still used for the stability analysis of TDSs. TDSs system is considered to be stable if all roots of the characteristic equation, $M(s) = 0$, lie in the open left half of complex plane, ([14], p.52). However, equation (3.17) contains an infinite number of roots due to its transcendental nature, therefore, it is impossible to determine every root of $M(s)$. Nevertheless, if the TDSs is of retarded type then the stability can be guaranteed by (2.4), moreover, the overall dynamic of the retarded system is mainly affected by the dominant roots which lie closest to the imaginary axis since the dynamics of components defined by the rest of roots, that lie further from the imaginary axis, are so fast that their influence is negligible ([14], p.52). For a stability analysis of neutral type of TDSs the reader is referred to [11].

Numerous analytical and numerical methods have been developed to compute the location of the roots of quasi-polynomial characteristic equation. One of the analytical method is based on the contour mapping of the real $R = \Re(M(s))$ and the imaginary part $I = \Im(M(s))$ of the characteristic equation (3.17). The contours are, for $s = \beta + j\omega$, defined as follows, ([20], p.8)

$$\begin{aligned} R(\beta, \omega) &= 0, \\ I(\beta, \omega) &= 0, \end{aligned} \tag{3.20}$$

the roots of $M(s)$ are determined as the intersection points of the contours (3.20). One of the algorithm based on the contour mapping technique is *Quasipolynomial Mapping Based Rootfinder* (QPMR) presented in [21].

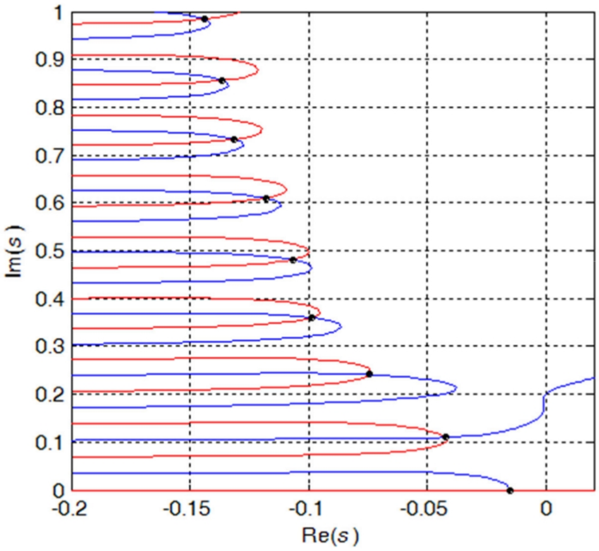


Figure 3.3. Example of contour mapping technique for roots finding of $M(s)$ by use of QPMR algorithm, red - $\Im(M(s)) = 0$, blue - $\Re(M(s)) = 0$, black - roots, (taken from [20], p. 8).

Chapter 4
Signal Shapers

Signal shapers are basically notch finite impulse response (FIR) filters, which purpose is to minimize oscillations of mechanical systems by removing excitation frequencies from the reference command. The main difference between signal shaping and traditional FIR filtering is that no frequency passband is used since there is no requirement that some frequencies must pass through the filter without significant attenuation, therefore, design procedure of signal shaping is simpler and resulted shaper will be faster than corresponding FIR filter that must pass certain frequencies. This technique is mainly used for compensation of oscillation modes of flexible mechanical parts, ([1], p.65).

Signal shapers are used mainly as a reference command filter for manipulators and cranes with flexible loads, ([1], p.29). The key idea of signal shaping is to dampen system's oscillations by delaying a part of the input signal so that the energy supplied to the system does not induce system's oscillation at its oscillatory modes. This idea, also known as posicast control, was first introduced by O.J. Smith in 1958 and was revisited in the 1990's by Singer, Seering and Singhose, who developed several signal shaping techniques based on Smith's posicast control with improved robustness for varying oscillation modes, namely, zero-vibration (ZV) shaper, zero-vibration-derivative (ZVD) shaper, extra insensitive (EI) shaper, and multi mode shapers [22]. Signal shapers are usually implemented in feed-forward control architecture and work as a reference command filter, however, disadvantage of this architecture is that they are not able to suppress oscillations excited by disturbances, ([8], p. 2050).

Shapers are designed to compensate oscillatory mode of the system defined by a complex conjugate couple of poles

$$r_{1,2} = -\beta \pm j\Omega, \tag{4.1}$$

$$\beta = \omega_0\zeta, \tag{4.2}$$

$$\Omega = \omega_0\sqrt{1-\zeta^2}, \tag{4.3}$$

where ζ is the damping ratio and ω_0 natural frequency of the mode to be compensated. A graphical representation of $r_{1,2}$ in the s plane is shown in Figure 4.1.

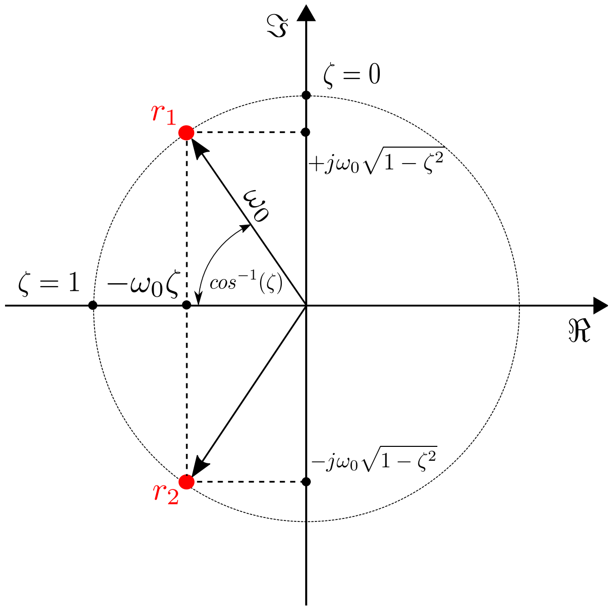


Figure 4.1. Complex conjugate poles $r_{1,2}$ depicted in the s-plane, (taken and modified from [1]), 21.04.2017).

General form of an input shaper in the time domain and the s domain, respectively, is given as follows, ([18], eq. 9, 10)

$$u(t) = A_0 w(t) + \sum_{k=1}^{n} A_k w(t - \tau_k), \tag{4.4}$$

$$S(s) = A_0 + \sum_{k=1}^{n} A_k e^{-s\tau_k}, \tag{4.5}$$

where w and u are the shaper's input and output, respectively. The parameters are gains $A_k \in \Re$ satisfying condition $\sum_{k=0}^{n} A_k = 1$, the time delays $\tau_k \in \Re^+$ and number of shapers's pulses $n \in \Re^+$.

Shaper's effectiveness depends on the precise estimation of the system's target oscillation mode determined by ζ and ω_0. Robustness to model's imperfections can be described by equation of residual vibration $V(\omega, \zeta)$, that is for shapers with lumped delays defined as follows ([1], eq. 5.1-3)

$$V(\omega, \zeta) = e^{-\zeta\omega\tau_N} \sqrt{R(\omega, \zeta)^2 + I(\omega, \zeta)^2}, \tag{4.6}$$

$$R(\omega, \zeta) = A_0 + \sum_{k=1}^{n} A_k e^{\zeta\omega\tau_k} cos(\tau_k \omega \sqrt{1 - \zeta^2}), \tag{4.7}$$

$$I(\omega, \zeta) = \sum_{k=1}^{n} A_k e^{\zeta\omega\tau_k} sin(\tau_k \omega \sqrt{1 - \zeta^2}), \tag{4.8}$$

the equation (4.6) is expressed in a non-dimensional form. It is generated by taking the amplitude of residual vibration induced by an impulse series (shaped signal) divided by the vibration amplitude caused by a single, unity-magnitude impulse (non-shaped signal) at time $t = \tau_N$ that is usually the time location of the final impulse, ([1], p.35).

[1] http://www.wikiwand.com/en/Root_locus

4.1 Zero Vibration Shaper

Zero Vibration (ZV) shaper is used in a feedforward connection with a system and is described by following equations in time domain and s domain, ([7], eq.1, 2)

$$u(t) = Aw(t) + (1 - A)w(t - \tau), \tag{4.9}$$

$$S_{ZV}(s) = A + (1 - A)e^{-s\tau}, \tag{4.10}$$

the ZV shaper has no poles but infinitely many zeros that are determined by the roots of the equation (4.10), ([7], eq.3)

$$s_{2k+1,2k+2} = -\frac{1}{\tau} ln \frac{A}{1 - A} \pm j\frac{\pi}{\tau}(2k + 1), k = 0, 1, \ldots, \infty, \tag{4.11}$$

where shaper's parameters gain A and lumped delay τ can be derived by placing the dominant zero $s_{1,2}$ at the position of oscillatory mode (4.1) and are given as follows, ([7], eq.4)

$$A = \frac{K}{1 + K}, \ K = e^{\frac{\beta\pi}{\Omega}}, \tag{4.12}$$

$$\tau = \frac{\pi}{\Omega}. \tag{4.13}$$

4.2 Robust Zero Vibration Shapers

The ZVD Shaper was implemented to address poor robustness of ZV shaper to modeling errors. ZVD shaper is designed by requiring the partial derivative of the residual vibration (4.6), with respect to the frequency, to be equal to zero at the modeling frequency, ([1], eq. 7.4)

$$\frac{\partial V(\omega, \zeta)}{\partial \omega} = 0. \tag{4.14}$$

This constraint keeps the vibration near zero as the actual frequency starts to deviate from the modeling frequency. The resulting ZVD is defines as follows, ([1], eq. 7.5)

$$u(t) = A_0 w(t) + A_1 w(t - \tau_1) + A_2 w(t - \tau_2), \tag{4.15}$$

$$S_{ZVD}(s) = A_0 + A_1 e^{-s\tau_1} + A_2 e^{-s\tau_2}, \tag{4.16}$$

where impulses A_k and time delays τ_k are

$$A_0 = \frac{1}{(1 + K)^2}, \ A_1 = \frac{2K}{(1 + K)^2}, \ A_2 = \frac{K^2}{(1 + K)^2}, \ K = e^{-\frac{\beta\pi}{\Omega}},$$

$$\tau_1 = \frac{\tau}{2}, \ \tau_2 = \tau.$$

Delay τ is defined by equation (4.13). Unlike the ZV shaper, the ZVD shaper splits the input command on three parts. Another approach how to improve robustness of ZV shaper is to combine two or more ZV input shapers in series connection with zeros placed in the neighborhood of the flexible poles. This type of shaper is called Extra Insensitive (EI). The trade-off of ZVD and EI shapers is increased rise time since the duration of the combined shaper is the sum of the individual shaper durations, ([1], p.52).

4.3 Distributed Zero Vibration Shaper

DZV shaper is a more generalized form of the ZV shaper, whose delay distribution function (4.17) can be parametrized. The delay distribution function can be described by Stieltjes integral as follows

$$u(t) = \int_0^T w(t-\eta)dh(\eta), \tag{4.17}$$

where w is the delay input, u the delay output and the function $h(\eta)$ describes the delay distribution over interval $[0, T]$. The delay distribution of the ZV shaper is of lumped type

$$h(\eta) = H(\eta - \tau), \quad \eta \in [0, \tau], \tag{4.18}$$

$$G(s) = e^{-s\tau}, \tag{4.19}$$

where $H(.)$ is the Heaviside step function and τ the lumped delay value. Another used delay distribution function for signal shapers is the trapezoidal function that distributes time delay equally

$$h(\eta) = \frac{1}{T}\eta, \quad \eta \in [0, T], \tag{4.20}$$

$$G(s) = \frac{1 - e^{-sT}}{Ts}, \tag{4.21}$$

where T is the distributed delay length. Compared to the ZV shaper, the response time of the DZV shaper with equally distributed delay is considerably longer, ([6], p.259). The response time can be improved by delaying the pulse part of the shaper signal, so that the time-delay is composed of the lumped delay τ and of the shifted equally distributed delay T. This solution was proposed in the article [7] and this type of shaper is called Distributed Zero-Vibration Shaper with lumped and equally distributed delay (D_eZV). The delay distribution function of the D_eZV shaper is defined as follows, ([18], eq. 20, 23)

$$h(\eta) = \begin{cases} 0, & \eta \in [0, \tau] \\ \frac{1}{T}(\eta - \tau), & \eta > \tau \\ 1, & \eta > T \end{cases}, \tag{4.22}$$

$$G(s) = \frac{1 - e^{-sT}}{Ts} e^{-s\tau}. \tag{4.23}$$

An analytical parametrization of D_eZV shaper was first proposed in [18]. By considering $s = -\beta + j\Omega$ the parameters of D_eZV shaper are defined as follows, ([18], eq. 24, 25)

$$T \in (0, \frac{\pi}{\Omega}], \tag{4.24}$$

$$\tau = \frac{\pi + \varphi}{\Omega}, \tag{4.25}$$

$$A = \frac{me^{\frac{\beta}{\Omega}(\pi+\varphi)}}{1 + me^{\frac{\beta}{\Omega}(\pi+\varphi)}}, \tag{4.26}$$

where,

$$m = |G(s)|, \tag{4.27}$$
$$\varphi = arg(G(s)). \tag{4.28}$$

4.4 Shapers in Feedback Architecture

Signal shapers are most used as reference command filters and classical control architecture with signal shapers is shown in Figure 4.2. The main drawback of this architecture is an inability to suppress the oscillations caused by the unmeasurable disturbance d, ([8], p. 4418).

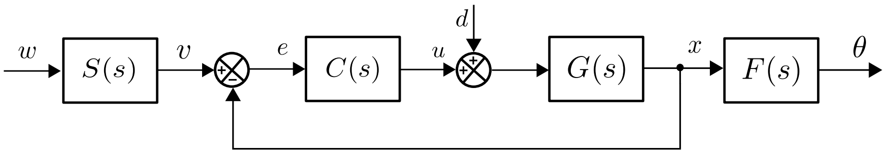


Figure 4.2. Feedback loop with shaper as reference command filter, (taken and modified from [8], Figure 2).

The transfer functions of the closed loop system, are shown in Figure 4.2, are, ([8], eq. 11, 12)

$$T_{w\theta} = \frac{C(s)G(s)S(s)}{1 + C(s)G(s)S(s)}F(s), \tag{4.29}$$
$$T_{d\theta} = \frac{G(s)}{1 + C(s)G(s)S(s)}F(s). \tag{4.30}$$

From the equation (4.29) it can be seen that the active zeros of the shaper $S(s)$ are the zeros of the closed loop system and can compensate the pole $r_{1,2}$ of the flexible system $F(s)$, therefore, oscillations induced by the reference w are suppressed. Nevertheless, from the equation (4.30), is clear that shaper's zeros does not appear in the numerator of the closed loop system, therefore, the signal shaper does not suppress oscillations caused by the disturbance d, ([8], p. 4419).

To address this problem a novel type of feedback architecture with an inverse signal shaper, was proposed by Vyhlídal, Hromčík, Kučera in articles [8, 6]. Feedback loop with an inverse shaper in the feedback path is shown in Figure 4.3.

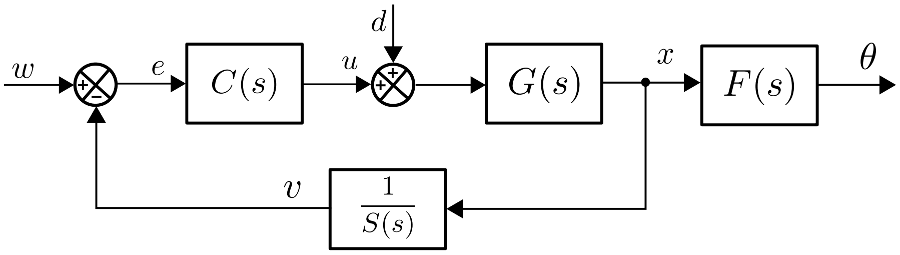


Figure 4.3. Feedback loop architecture with an inverse shaper, (taken and modified from [8], Figure 3).

The transfer functions of the closed loop system with an inverse signal shaper is defined as follows, ([8], eq. 15, 16)

$$\begin{aligned} T_{w\theta} &= \frac{C(s)G(s)}{1 + C(s)G(s)\frac{1}{S(s)}}F(s) \\ &= \frac{C(s)G(s)S(s)}{S(s) + C(s)G(s)}F(s), \end{aligned} \tag{4.31}$$

$$\begin{aligned} T_{d\theta} &= \frac{G(s)}{1 + C(s)G(s)\frac{1}{S(s)}}F(s) \\ &= \frac{G(s)S(s)}{S(s) + C(s)G(s)}F(s), \end{aligned} \tag{4.32}$$

where it can be seen that transfer function $S(s)$ appears in the numerator of both equations (4.31) and (4.32), therefore, the shaper can compensate the oscillatory mode induced by both the reference w and the disturbance d ([8], p. 4420).

The block diagrams that are shown in Figures 4.2 and 4.3 are valid only for special case where the dynamics of the system $G(s)$ and the flexible system $F(s)$ can be considered as decoupled, e.g. the mass of the system $F(s)$ is significantly smaller than the mass of the system $G(s)$. However, in reality the dynamics of $G(s)$ and $F(s)$ are always coupled as it is shown for example in Figure 4.4.

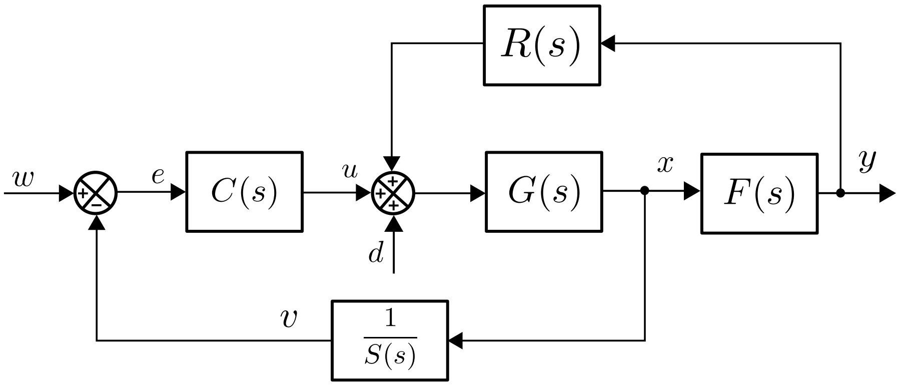


Figure 4.4. Coupled case of feedback loop architecture with an inverse shaper, (taken and modified from [9]).

where $R(s)$ is the coupling transfer function. The transfer functions of the closed-loop system with coupled dynamics are defined as follows, ([9], eq. 62, 63)

$$T_{w\theta} = \frac{C(s)G(s)}{1 + C(s)G(s)\frac{1}{S(s)} - F(s)R(s)G(s)}F(s), \tag{4.33}$$

$$T_{d\theta} = \frac{G(s)}{1 + C(s)G(s)\frac{1}{S(s)} - F(s)R(s)G(s)}F(s), \tag{4.34}$$

the system $F(s)$ can be considered decoupled on frequency range of interest if the following condition holds true, ([9], eq. 64)

$$\left|1 + C(j\omega)G(j\omega)\frac{1}{S(s)}\right| >> \left|F(j\omega)R(j\omega)G(j\omega)\right|. \tag{4.35}$$

As can be seen from (4.35), decoupled case can be obtained by an appropriate choice of the controller $C(s)$ and the shaper $S(s)$. However, if the dynamics of $G(s)$ and $F(s)$ are strongly coupled, then it is clear from (4.33) and (4.34) that the required zero-pole compensation is not guaranteed. Therefore, a different approach has to be applied in order to compensate oscillatory modes, e.g., compensation via input command shaper. The issue of mode compensation of systems with coupled dynamics via inverse shapers in feedback architecture has been recently discussed in the article [23]. In this thesis will be considered only systems with decoupled dynamics.

4.4.1 Implementation of Inverse Shaper

Transfer functions of shapers ZV, DZV and D_eZV in the inverse form are defined as follows, ([8], eq. 17, 18)

$$\frac{1}{S_{ZV}(s)} = \frac{1}{A+(1-A)e^{-s\tau}}, \tag{4.36}$$

$$\frac{1}{S_{DZV}(s)} = \frac{1}{A+(1-A)\frac{1-e^{-sT}}{sT}}, \tag{4.37}$$

$$\frac{1}{S_{D_eZV}(s)} = \frac{1}{A+(1-A)\frac{1-e^{-sT}}{sT}e^{-s\tau}}. \tag{4.38}$$

It was shown in [8] that the spectrum of zeros of ZV shaper given by equation (4.11) is of neutral type, i.e., parallel to the imaginary axis of the complex plane, therefore, ZV shaper in the inverse form is not preferred since its zeros might cause instability of the closed-loop system due to the low stability robustness against model imperfections. Thus, a shaper with retarded spectrum, i.e., zeros depart from the imaginary axis to the stable region as the moduli of roots increase, should be used especially if the oscillatory mode is very close to the imaginary axis, i.e., slightly damped system.

Inverse forms of DZV and D_eZV shapers are suitable, since their spectrum of zeros are of retarded type. Spectra of zeros of ZV, DZV and D_eZV shaper are shown in Figure 4.5 and their frequency responses in Figure 4.6.

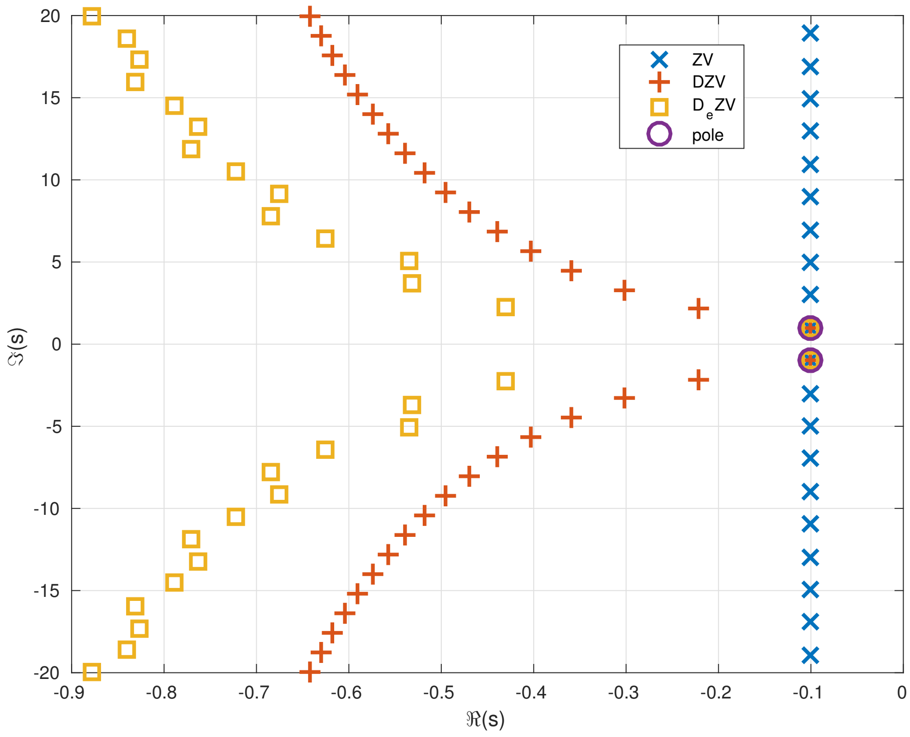


Figure 4.5. Zeros of ZV (blue), DZV (red) and D_eZV (yellow) shaper designed to compensate poles $r_{1,2} = -0.1 \pm j0.99$ (purple), source of example [18].

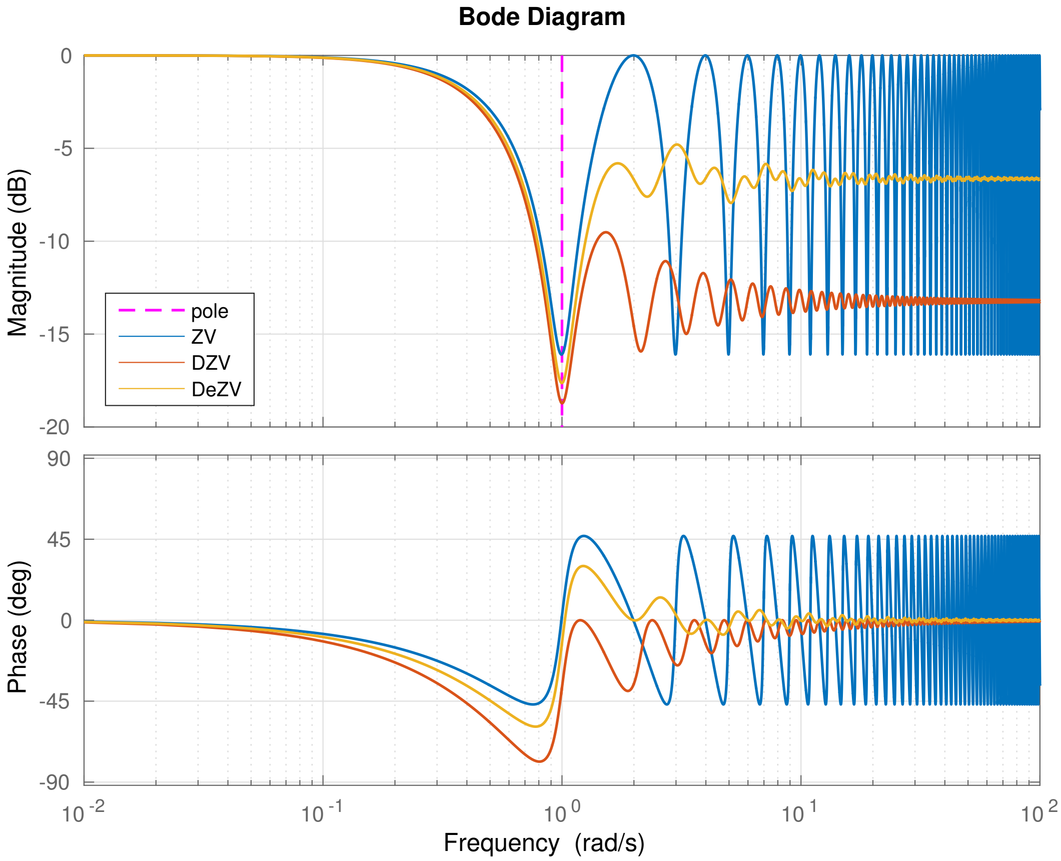


Figure 4.6. Frequency response of different Zero Vibration shapers, ZV (blue), DZV (red) and D_eZV (yellow) shaper designed to compensate poles $r_{1,2} = -0.1 \pm j0.99$ (purple), source of example [18].

Any type of inverse shaper included in the closed loop causes its infinite dimensionality and this aspect should be taken into account during the controller design $C(s)$, ([8], p. 4421).

Inverse Signal shapers can be implemented in program MATLAB/Simulink by use of block *LTI System* from *Control System Toolbox* library where the argument is shaper's transfer function, or via standard Simulink's blocks as it is shown in Figure 4.7.

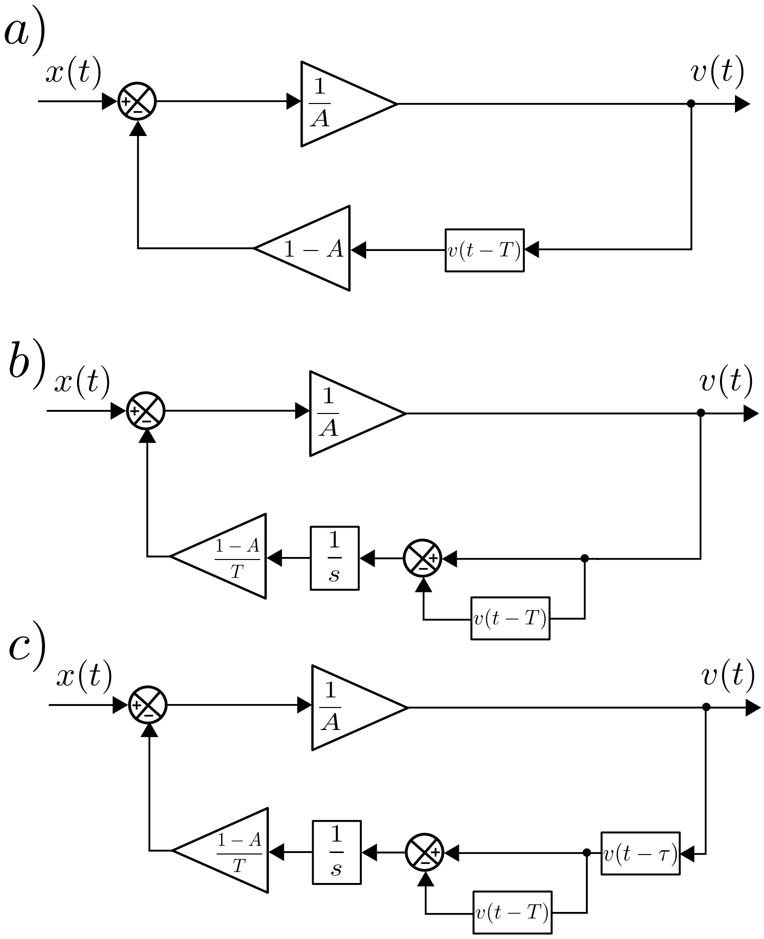


Figure 4.7. Implementation of inverse vibration shapers: a) ZV, b) DZV, c) D_eZV ,(taken and modified from [8]).

Chapter 5
Internet-based Control Systems

Process of integration of information and control plays an important role in the complex control applications of manufacturing industries. This process is often referred as Computer Integrated Manufacturing (CIM), ([24], p.3). An effective CIM improves plant energy efficiency, better monitoring, manpower savings and effective integration of the plant. Most of the process control components have RS232 connectivity, however, connecting every device via RS232 would result in a highly unmaintainable and rigid infrastructure. Thanks to the rapid development of the Internet infrastructure, the concept of Internet-based control has been receiving more attention in the recent years. This type of control system allows remote control, monitoring and adjustment of the plant over the Internet. Plants can benefit from its use by allowing world-wide control or information retrieval at anytime, ([25], p.1). The problem of device integration can be solved by designing a reasonable Information Architecture (IA) of the whole system. The IA can be separated into 4 levels from the highest to the lowest, as follows, ([25], p.4):

- Management level : commercial data, accessed by customers and managers.
- Optimization level : global databases with plant-wide information in process plants.
- Supervisory level : process database that contains real-time status of process plants.
- Regulatory level : for control of local control units (Controllers, PLC).
- Sensor/Actuator level : the lowest level for direct transmission of control and measured signals.

These levels are distinguished from each other by *4R's* principle criteria, that are defined as follows, ([25], p.4):

- Response time : Higher levels of the IA have higher tolerance level of the time-delay of the receiving data. For example, information used at the management level can be several days old, unlike at the regulatory or sensor/actuator level.
- Reliability : Required level of reliability increases with every lower level of the IA. For instance, host computers at the management level can be safely shutdown for hours, with relatively little consequence. However, this does not hold for the supervisory or the regulatory control level.
- Resolution : The higher the level, the more abstract data are preferred.
- Reparability : The reparability considers the ease with which control and computing devices can be maintained.

As it is shown in Figure 5.1, the Internet can be linked with a local computer system at any level in the IA, even at the lowest level of sensors and actuators. These links result in a range of *4R's*. For instance, if a fast response time is required, then a link should be made to the control loop, however, if only the abstract information is needed, then the Internet should be linked at a higher level in the IA, such as management or optimization level, ([25], p.5). Table 5.1 shows a simple evaluation for the possible Internet links.

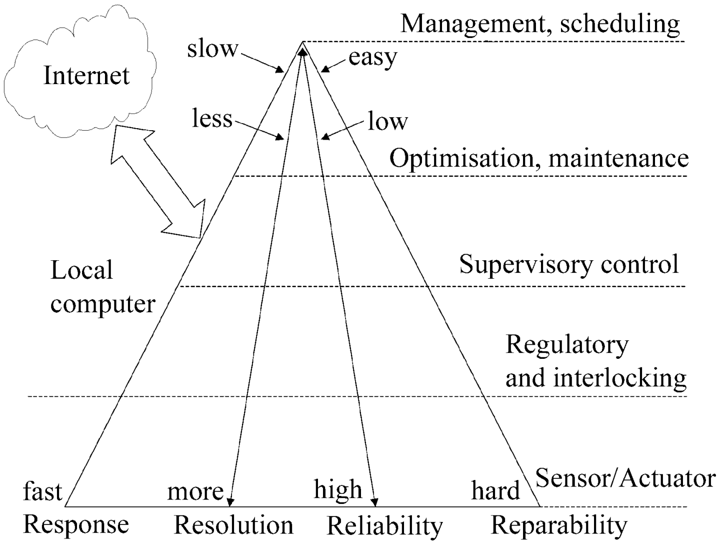


Figure 5.1. The layers of Information architecture, (taken from [25], Figure 5).

Information level	**Information exchange**	**Advantages**	**Disadvantages**
Management level	Commercial data systems	Commercial data are accessible by managers	Not suitable for real-time control tasks.
Optimization level	Global database	Easy access to the plant-wide information.	Not suitable for real-time control tasks.
Supervisory level	Process database	Easy access to the real-time status of the plant.	Lacks management information
Regulatory level	PLC, Control Unit	Direct access of the controller to the Internet.	Internet delay, safety risks
Sens./Act. level	Smart Devices	Controlling the devices directly from the Internet.	Internet delay, safety risks

Table 5.1. Pros and cons of possible links between the Internet and control levels, (taken from [25], Table 1.).

5.1 Internet-Based Control Architectures

If the Internet is linked directly with sensors or actuators, than a controller is located at a remote site, connected to an actuator and sensor via the Internet, in this architecture the Internet becomes a part of the control system as it is shown in Figure 5.2. This architecture is also known as Networked Control System (NCS) and will be described in detail in the chapter 5.2. Major disadvantage of this architecture is that the transmission delay is introduced to both the actuator and the sensor communication channel, ([25], p.5).

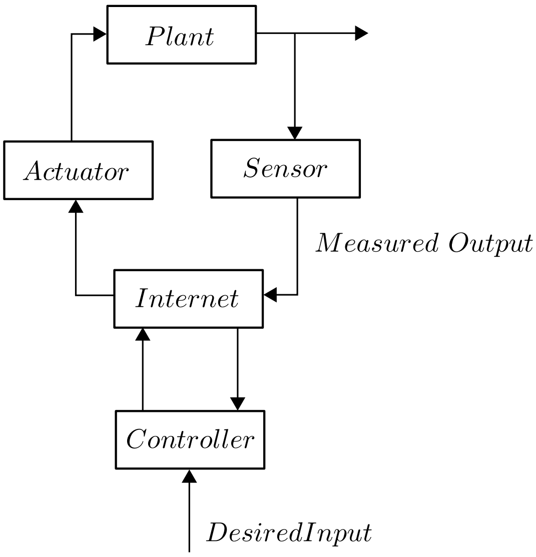


Figure 5.2. Control architecture with the remote controller, (taken and modified from [25], Figure 6).

Controllers such as programmable logic controllers (PLC) are often directly integrated with the Internet by using a Transmission Control Protocol/Internet Protocol (TCP/IP). In such a case, a desired input can be sent from the remote site to the controller located at a local site of a plant, via Internet, and in order to monitor the state of the plant over time, measured output is usually fed back to the remote site. The Internet is excluded from the closed loop of the controller and because of that Internet transmission delay will not affect the dynamics of the control system, ([25], p. 5). This architecture is shown in Figure 5.3.

Bilateral control architecture has a controller located at the plant site, and another one at the remote site linked via the Internet as it is shown in Figure 5.4. The controller at the plant site is responsible for process regulation during normal operation. However, if the performance of the controller degrades, for example due to the change of production or a environmental disturbances, the controller at the remote site can be used to retune paramters or change the desired input for the controller at the plant site. Such a control architecture is widely used in tele-robot control systems, ([25], p.6).

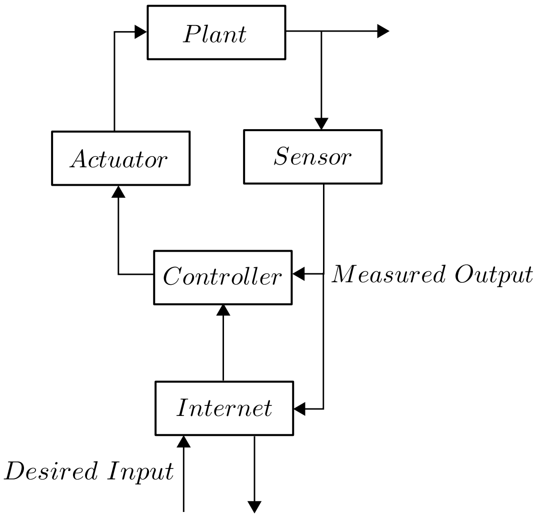


Figure 5.3. Control architecture with remotely sent input signal to the local controller, (taken and modified from [25], Figure 7).

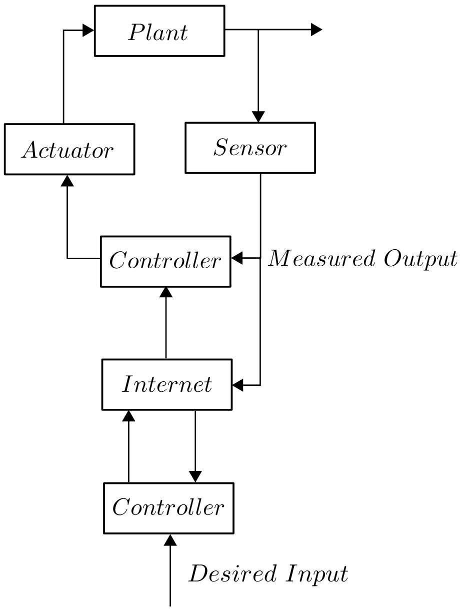


Figure 5.4. Control architecture with bilateral controllers , (taken and modified from [25], Figure 8).

5.2 Networked Control System

The Networked control systems (NCSs) basically use internet-based control architecture with the remote controller shown in Figure 5.2. The NCSs have a feedback control loop, where a plant is controlled via a communication network. The information of the plant output and the control input is exchanged using the communication network among system components (sensors, actuators, controller, etc.), ([15], p. 4). A block diagram of simple NCSs system with one sensor and one actuator network node is shown in Figure 5.5.

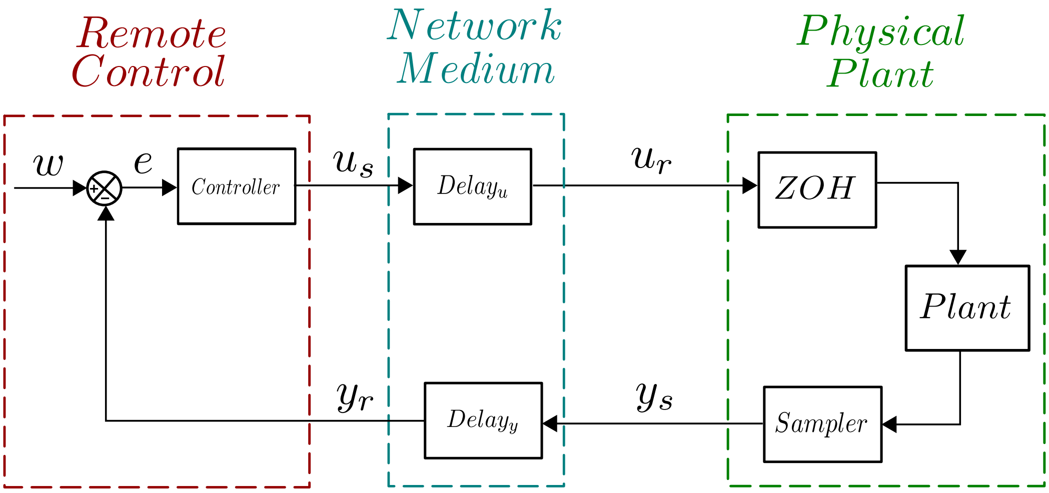


Figure 5.5. Block diagram of the networked control system, (taken and modified from [15], Fig. 1.4).

Where u_s is the control signal sent by the remote control station, u_r the control signal received by the plant, y_s is the measured output sent by the plant and y_r the measured output received by the remote control station. The performance of the NCSs is negatively affected by the imperfect transmission of signals u_s and y_s.

$$u_s(t) \neq u_r(t),$$
$$y_s(t) \neq y_r(t).$$

Transmission imperfections are mostly caused by a variable sampling intervals, variable or constant communication delays, packet loss and quantization errors caused by the finite word length of packets ([15], p. 4). The presence of these constraints can degrade the overall performance of the control loop and can even lead to instability of the controlled system ([15], p. 4).

Compared to the traditional feedback control systems, where the components are usually connected via point-to-point cables, the introduction of communication network media brings some advantages, such as low cost, simple installation, maintenance and an easily modifiable long distance control ([15], p. 4). NCSs are implemented by using digital technology and involve both a continuous-time dynamics and a discrete-time control due to an analog-to-digital converter (a sampler) and a digital-to-analog converter (a zero-order hold).

5.3 Time Delay in Packet-Switched Networks

The nodes of modern communication networks mostly utilize the method called *Packet Switching* (PS) to pass the data from a source end system, through the correct links and nodes, to a destination end system. The sent data is broken into smaller chunks of data known as *packets*. Between source and destination, each packet travels through communication links and packet switches and is transmitted at a rate equal to the full transmission rate of the link ([26], p.22). For a more general description of computer networking the reader is referred to e.g. [26].

The performance at a network node is often measured in terms of time delay and the probability of packet loss ([26], p.41). The delay of packet is a time variable, which changes with Internet traffic. Overall nodal delay consists of the nodal processing delay τ_{proc}, queuing delay τ_{queue}, transmission delay τ_{trans}, and packet propagation delay τ_{prop}, ([26], p.36), as shown illustratively in Figure 5.6.

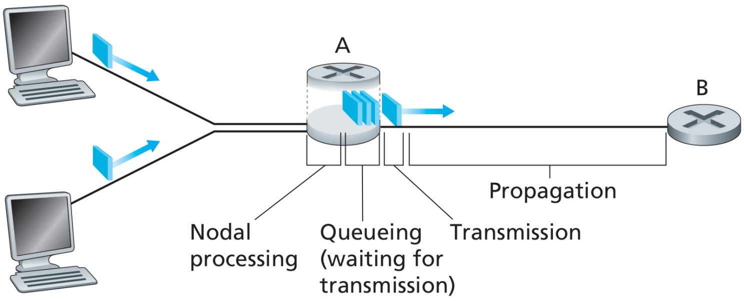


Figure 5.6. The nodal delay at router A, (taken from [26], Fig. 1.16).

The overall nodal delay is sum of individual delay components, ([26], p.40)

$$\tau_{nodal} = \tau_{proc} + \tau_{queue} + \tau_{trans} + \tau_{prop}. \tag{5.1}$$

5.3.1 Processing Delay

The processing delay is composed of the time required to examine the packet's header and determine where to direct the packet and the time needed to check for bit-level errors of arriving packet. After this nodal processing, the router directs the packet to the queue that precedes the link to the destination router. Processing delays are often negligible and in high-speed routers are typically on the order of microseconds to milliseconds, ([26], p.37).

5.3.2 Queuing Delay

The packet is delayed at the queue as it waits to be transmitted onto the link. The length of the queuing delay of a specific packet will depend on the number of earlier-arriving packets that are queued and waiting for transmission onto the link. If the queue is empty and no other packet is currently being transmitted, then queuing delay will be equal to zero. However, queuing delay increases with an increasing network traffic due to the larger amount of waiting packets, therefore, the queuing delay can vary from packet to packet. Queuing delays are usually on the order of microseconds to milliseconds, ([26], p.37).

The significance of queuing delay depends on the rate at which traffic arrives at the queue, the transmission rate of the link, and the nature of the arriving traffic, that is, whether the traffic arrives periodically or arrives in bursts. The extent of the queuing delay is often estimated by the traffic intensity (TI), that is defined as follows, ([26], p.40)

$$TI = \frac{L \cdot a}{R}, \tag{5.2}$$

where a ($packets{\cdot}s^{-1}$) is the average rate at which packets arrive at the queue, L ($bits$) is the size of the packets and R ($bits{\cdot}s^{-1}$) is the transmission rate, (for further explanation see 5.3.3). Assume that the queue is infinitely large, so that it can hold an infinite number of bits.

If the TI is close to zero, then packets arrivals are few and far between and the probability that some pockets are in the queue is small, therefore, the average queuing delay will be close to zero, ([26], p.40).

If the TI approaches 1, there will be intervals of time when the arrival rate exceeds the transmission capacity, and a queue will become larger, therefore, the average queuing delay will get larger, ([26], p.40).

And in the case of $TI > 1$, the average rate at which bits arrive at the queue exceeds the rate at which the bits can be transmitted from the queue, therefore, the queue will tend to increase without bound and the queuing delay will approach infinity, ([26], p.40). The dependence of average queuing delay on the TI is shown in Figure 5.7.

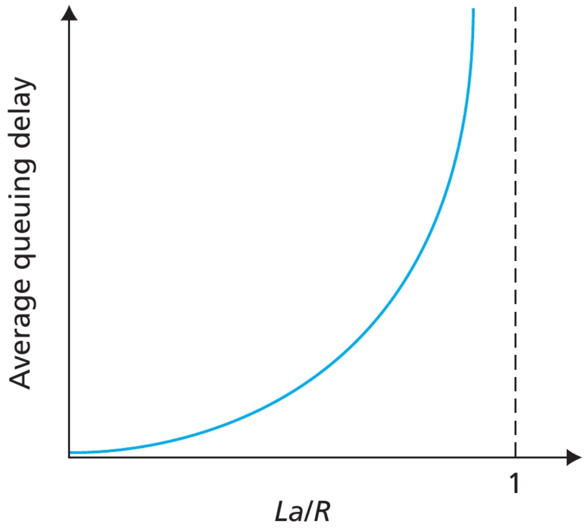


Figure 5.7. Dependence of average queuing delay on Traffic Intensity, (taken from [26], Fig. 1.18).

5.3.3 Transmission Delay

The packets, in packet-switched networks, are transmitted in a first-come-first-served manner, therefore, the packet can be transmitted only after all the packets that have arrived before it have been transmitted. The transmission delay is the amount of time required to transmit all of the packet's bits into the link and is defined as follows, ([26], p.37)

$$\tau_{trans} = \frac{L}{R}, \tag{5.3}$$

where L $(bits)$ is the length of the packet and R $(bits \cdot s^{-1})$ is the transmission rate of the link from the sending router to the destination router. Transmission delay is usually negligible for $R = 10$ $(Mbps)$ and higher (for example, for LANs), however, it can be of hundreds of milliseconds for large packets send over low-speed dial-up modem links. Transmission delays are usually on the order of microseconds to milliseconds in practice, ([26], p.37).

5.3.4 Propagation Delay

The propagation delay is the time required to propagate a bit from the beginning of the link to the destination router. The bit propagates at the propagation speed v_{prop} of the link that depends on the physical medium of the link, e.g., fiber optics, twisted-pair, copper wire, and is in the range of $2 \cdot 10^8$ $(m \cdot s^{-1})$ to $3 \cdot 10^8$ $(m \cdot s^{-1})$. The propagation delay is defined as, ([26], p.37)

$$\tau_{prop} = \frac{d}{v_{prop}}, \tag{5.4}$$

where d is the distance between the sending router and the destination router. Propagation delay is usually negligible for a link connecting two routers in the same building, however, it can be of hundreds of milliseconds for two routers interconnected by a geostationary satellite link, these delays are usually on the order of milliseconds, ([26], p.38).

5.3.5 Basic Communication Protocols of Transport Layer

Overall nodal delay can be reduced by choosing an appropriate transport layer's communication protocol used to stream data. The most popular protocols used to transmit data over the Internet are the Transmission Control Protocol/Internet Protocol (TCP/IP) and the User Datagram Protocol (UDP).

The TCP is the connection-oriented and confirmation based protocol that provides a point-to-point channel for applications that require reliable communication with no loss of information. It is a higher-level protocol that manages to robustly string together data packets, sorting them and retransmitting them, ([27], p. 128). However, in the article, [27], it was shown that the TCP protocol is not suitable for real-time control applications due to the substantial variation of the transmission time delay during connection.

The UDP is connectionless; i.e., a message can be sent from a sender to a receiver without prior arrangement, ([26], p.199). The provided communication between two applications over network, is not guaranteed because the UDP is not confirmation based protocol, therefore, the packet arrival and the order of arrival at the destination is not guaranteed, ([27], p.128). The UDP sends independent packets of data, called datagrams, from one application to another. These packets are addressed (with Internet Protocol Address and Port number) and then transmitted. The UDP is more simpler protocol then TCP and have a more raw interface to the network, therefore, the transmission delay is minimized by omitting the connection setup process, flow control, and retransmission ([28], p.1). However, this simplicity comes at the expense of some packets getting lost. The major requirement of most real-time control applications is fast, consistent and efficient data transmission. In the article [27] it was proved that the UDP provides a consistent sample rate with lower variations of the transmission delay, unlike TCP. For these reasons and its simplicity is UDP often preferred in the real-time control applications, ([29], p.1583, [27], p.129).

Chapter 6
Controlled System

In the first part of this chapter, a laboratory setup of a controlled system will be presented. Then a mathematical model of the controlled system will be derived, and finally the parameters of mathematical model will be identified.

6.1 System Description

The basic part of the laboratory set-up is a frame built of the standardized aluminum profiles that allows easy reconfigurability, ([30], p. 91). The controlled system is the electro-mechanical system which consists of carts a and b mounted on a linear ball-bearing rails, that are connected to each other via springs. A pendulum p is mounted on the cart b. The whole laboratory setup is shown in Figure 6.1.

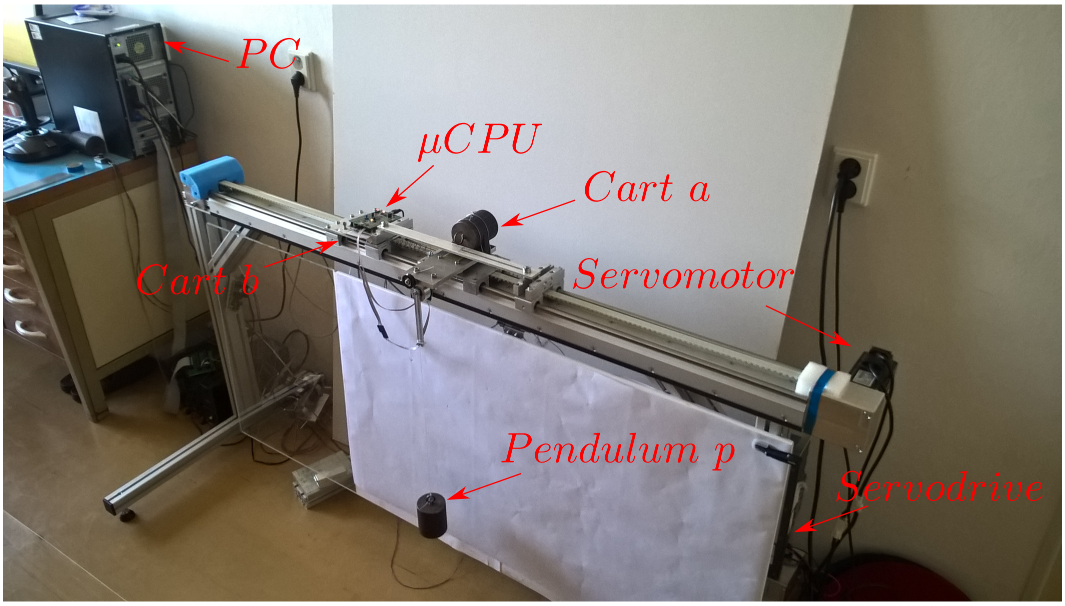


Figure 6.1. Photo of the laboratory set-up.

The active part is the cart a which is fixed to a teeth belt underneath powered by an AC servo motor (Estun-EMJ-04) controlled by a servo drive (Estun-Pronet-E-04A).

The servo drive is operated in the torque-generator regime, where the generated torque is proportional to the analog control signal, ([9], p.2058), and is commanded by an analog DC voltage signal, with the operating range $0 - 10(V)$ from an attached PC.

The PC is equipped with a data acquisition card (DAQ) (AD-622) whose 14 bit Digital/Analog converters are used to convert the digital control signal, generated by programs MATLAB/Simulink, to the analog control signal.

The measured variables are the positions of carts a, b and the deflection of the pendulum p. The position of the active cart a is determined from the feedback of the servo motor. The position of the passive cart b is measured by an incremental position sensor with integrated Hall elements (AS5304) and magnetic strip for measuring linear motion. The deflection of the pendulum p is measured by an angle position sensor (AS5048) with a 14-bit high resolution output. This sensor measures the absolute position of the magnet's rotation angle and consists of Hall sensors. All sensors' outputs are collected by a 32 bit micro-controller (STM32F407), that sends acquired data to the PC via serial communication port (COM). Received data are then processed by the program MATLAB/Simulink.

In the case of NCS setup, the feedback loop is closed through the communication network. The personal computer PC_p, that is nearby the plant, sends the measured data, via UDP communication connection, to the remote personal computer PC_c that computes the control signal and sends it back to the PC_p. The scheme of the controlling system is depicted in Figure 6.2.

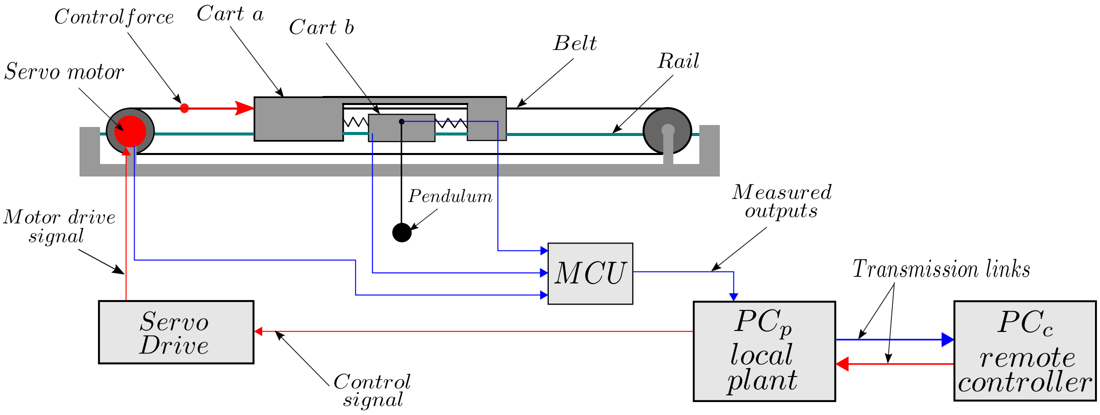


Figure 6.2. The scheme of the whole system.

6.2 Mathematical Model

The mathematical model of the controlled system is derived only for the mechanical part of the controlled system that is shown in Figure 6.2. In order to obtain a practically usable mathematical model of the system a several major assumptions had to be made. The modeling assumptions are:

- The belt's elasticity is omitted and an ideally rigid belt is assumed
- The transfer of the control signal to the control force is assumed to be without delay
- Carts and the pendulum bob are treated as point particles
- The rod on which the pendulum bob swings is inextensible
- The mass of the rod is negligible and the bob is a point mass
- The bob moves only in two dimensions and in the direction of cart's motion
- The motion of carts a and b is only of the translation type with friction
- Both springs have linear characteristic
- The air resistance is omitted

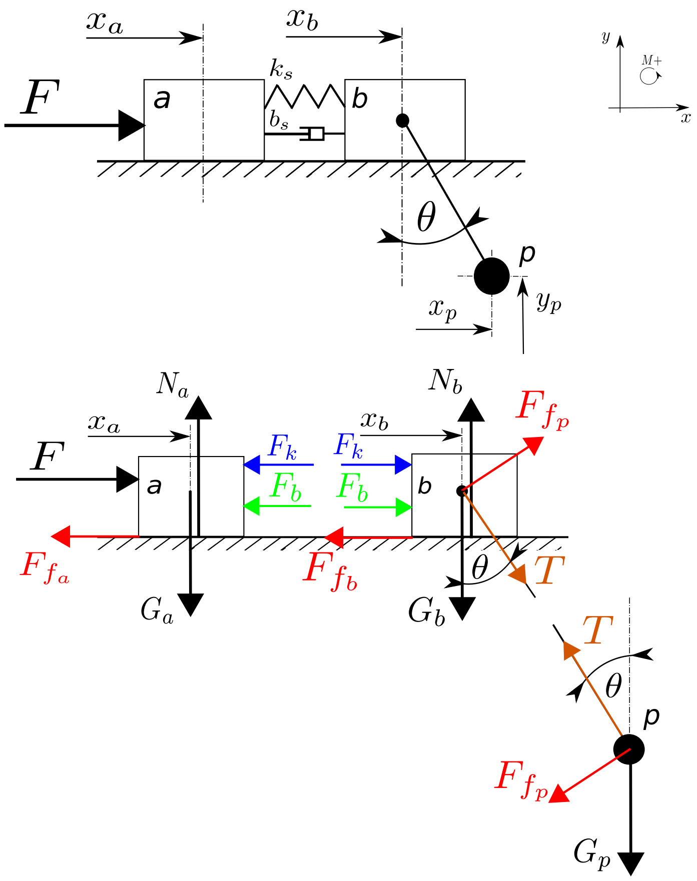


Figure 6.3. Mechanical diagram of the system with two carts and pendulum (at the top) and free body diagrams of carts and the pendulum bob (at the bottom), (taken and modified from [31]).

A mathematical model was derived by means of Newton-Euler equations of motion for free body diagrams that are shown in Figure 6.3. Indexes a, b and p denotes carts a, b and the pendulum p. $G_i = gm_i$ is a gravitational force and N_i is a corresponding normal force where m_i denotes mass. The spring force F_k and the damping force F_b are defined as follows

$$F_k = k_s \Delta x = k_s(x_b - x_a), \tag{6.1}$$
$$F_b = b_s \Delta \dot{x} = b_s(\dot{x}_b - \dot{x}_a), \tag{6.2}$$

where k_s, b_s are the stiffness and the damping of the spring. The control force F is given by equation

$$F(t) = s_a u(t), \tag{6.3}$$

where s_a is the static gain and $u(t)$ the control signal. T denotes tension force in the pendulum's rod. The same force from the friction torque is equivalent to the force applied at the end of the pendulum rod (Newton's Third law) [31], therefore, the friction force F_{f_p} is defined as follows

$$F_{f_p} = F_{f_p kin} + F_{f_p stat} = \frac{\mu_p}{l_p}\dot{\theta} + \frac{M_{s_p}(\dot{\theta})}{l_p}, \tag{6.4}$$

where μ_p is the viscous friction coefficient of the pendulum, l_p the length of the pendulum and $M_{s_p}(\dot{\theta})$ the static friction torque of the pendulum. Friction forces F_{f_a} F_{f_b} of carts a and b are given by following equitations

$$F_{f_a} = F_{f_a kin} + F_{f_a stat} = \mu_a \dot{x}_a + F_{s_a}(\dot{x}_a), \tag{6.5}$$
$$F_{f_b} = F_{f_b kin} + F_{f_b stat} = \mu_b \dot{x}_b + F_{s_b}(\dot{x}_b), \tag{6.6}$$

where μ_a and μ_b are viscous friction coefficients and $F_{s_a}(\dot{x}_a)$, $F_{s_b}(\dot{x}_b)$ are static friction forces.

6.2.1 Derivation of Equations of Motion

The Newton-Euler-Method was used to determine the dynamics equations of the system. The Newton-Euler (N-E) equations for force components in the y-axis of carts a and b are omitted since they are not necessary for derivation of equations of motion (they can be used to find normal forces N_a and N_b). The N-E equation of the cart a is

$$m_a \ddot{x}_a + (b_s + \mu_a)\dot{x}_a + k_s x_a = F + b_s \dot{x}_b + k_s x_b - F_{s_a}(\dot{x}_a), \tag{6.7}$$

N-E of the cart b

$$m_b \ddot{x}_b + (b_s + \mu_b)\dot{x}_b + k_s x_b = Tsin(\theta) + F_{fp}cos(\theta) + b_s \dot{x}_a + k_s x_a - F_{s_b}(\dot{x}_b), \tag{6.8}$$

N-E of bob of the pendulum p

$$m_p \ddot{x}_p = -Tsin(\theta) - F_{fp}cos(\theta), \tag{6.9}$$

$$m_p \ddot{y}_p = Tcos(\theta) - F_{fp}sin(\theta), \tag{6.10}$$

where $\ddot{x}_p$ and $\ddot{y}_p$ are derived from kinematic equations of the bob (6.11), (6.12), as follows

$$\begin{aligned} x_p &= x_b + l_p sin(\theta), \\ \dot{x}_p &= \dot{x}_b + l_p cos(\theta)\dot{\theta}, \\ \ddot{x}_p &= \ddot{x}_b + l_p(cos(\theta)\ddot{\theta} - sin(\theta)\dot{\theta}^2), \end{aligned} \tag{6.11}$$

$$\begin{aligned} y_p &= l_p cos(\theta), \\ \dot{y}_p &= -l_p sin(\theta)\dot{\theta}, \\ \ddot{y}_p &= -l_p(sin(\theta)\ddot{\theta} + cos(\theta)\dot{\theta}^2). \end{aligned} \tag{6.12}$$

Equation of motion of the cart b is derived by substituting $Tsin(\theta)$ from equation (6.9) to (6.8)

$$\begin{aligned} (m_b + m_p)\ddot{x}_b + (b_s + \mu_b)\dot{x}_b + k_s x_b = -l_p m_p cos(\theta)\ddot{\theta} + l_p m_p sin(\theta)\dot{\theta}^2 \\ + b_s \dot{x_a} + k_s x_a. \end{aligned} \tag{6.13}$$

Next by multiplying equation (6.10) by $sin(\theta)$, equation (6.9) by $cos(\theta)$ and by substituting term $Tsin(\theta)cos(\theta)$ from the equation (6.9) to the equation (6.10) we can receive the following equation

$$\begin{aligned} l_p m_p sin(\theta)^2 \ddot{\theta} + l_p m_p sin(\theta) cos(\theta) \dot{\theta}^2 = & -F_{fp} cos(\theta)^2 - m_p cos(\theta) \ddot{x}_b + l_p m_p cos(\theta)^2 \ddot{\theta} \\ & + l_p m_p cos(\theta) sin(\theta) \dot{\theta}^2 - F_{fp} sin(\theta)^2 \\ & - g m_p sin(\theta). \end{aligned} \tag{6.14}$$

By applying trigonometric identity $sin(\theta)^2 + cos(\theta)^2 = 1$ in the equation (6.14) and by substituting (6.4) to F_{fp}, the equation of motion of the pendulum bob is received

$$l_p m_p \ddot{\theta} + \frac{\mu_p}{l_p} \dot{\theta} + m_p cos(\theta) \ddot{x}_b + g m_p sin(\theta) = -\frac{M_{s_p}(\dot{\theta})}{l_p}. \tag{6.15}$$

In the next steps we need to isolate second derivative terms $\ddot{x}_b$ and $\ddot{\theta}$ from equations (6.15) and (6.13) so that we receive equations in the following forms

$$\begin{aligned} \ddot{x}_a &= f(x_a, \dot{x}_a, x_b, \dot{x}_b, \theta, \dot{\theta}) \\ \ddot{x}_b &= g(x_a, \dot{x}_a, x_b, \dot{x}_b, \theta, \dot{\theta}) \\ \ddot{\theta} &= h(x_a, \dot{x}_a, x_b, \dot{x}_b, \theta, \dot{\theta}) \end{aligned}$$

Equation of motion of the cart a is derived straightforwadly from the equation (6.7)

$$\ddot{x}_a = \frac{1}{m_a} \Big(F + b_s \dot{x}_b + k_s x_b - (b_s + \mu_a) \dot{x}_a - k_s x_a - F_{s_a}(\dot{x}_a) \Big). \tag{6.16}$$

For equations (6.13), (6.15) more rearrangements have to be done. First make $\ddot{\theta}$ the subject of the equation (6.15) and then substitute this subject $\ddot{\theta}$ to the equation (6.13). After rearranging the final form of equation of motion for the cart b is

$$\begin{aligned} \ddot{x}_b = \frac{1}{m_b + m_p sin^2(\theta)} \Big(& b_s \dot{x}_a + k_s x_a + m_p l_p sin(\theta) \dot{\theta}^2 + cos(\theta) (\frac{\mu_p}{l_p} \dot{\theta} + \frac{M_{s_p}(\dot{\theta})}{l_p}) \\ & + g m_p sin(\theta) cos(\theta) - (b_s + \mu_b) \dot{x}_b - k_s x_b - F_{s_b}(\dot{x}_b) \Big). \end{aligned} \tag{6.17}$$

Next, make $\ddot{x}_b$ the subject of the equation (6.13) and then substitute this subject $\ddot{x}_b$ to the equation (6.15). After rearranging the final form of equation of motion for the pendulum bob p is the following equation

$$\begin{aligned} \ddot{\theta} = \frac{1}{l_p (m_b + m_p sin^2(\theta))} \Big(& cos(\theta) ((b_s + \mu_b) \dot{x}_b + k_s x_b - b_s \dot{x}_a - k_s x_a) \\ & - (1 + \frac{m_b}{m_p}) (\frac{\mu_p}{l_p} \dot{\theta} + \frac{M_{s_p}(\dot{\theta})}{l_p}) - g(m_p + m_b) sin(\theta) - l_p m_p cos(\theta) sin(\theta) \dot{\theta}^2 \\ & + cos(\theta) F_{s_b}(\dot{x}_b) \Big). \end{aligned} \tag{6.18}$$

6.2.2 Linearization of Mathematical Model

Linear models are necessary for most control system design methods such as stability analysis or controller design, however, our mathematical model is described by nonlinear equations (6.16), (6.17) and (6.18), therefore, the model linearizion is needed for further control design and analysis.

First nonlinearity of our model is caused by a function $sign(\dot{x})$ that occurs in static friction forces F_{s_a},F_{s_b} and M_{s_p}, therefore, all static friction forces are set to zero. Another nonlinearities are caused by trigonometric functions $sin(.)$ and $cos(.)$, these functions can be linearized for assumption of small angles, $\theta = \pm 0.1 (rad)$, so that $sin(\theta) \approx \theta$, $cos(\theta) \approx 1$, $\ddot{\theta} \approx 0$ and $\dot{\theta}^2 \approx 0$. Linearized equations of motions of our model are then defined as follows

$$\ddot{x}_a = \frac{1}{m_a}\Big(F + b_s\dot{x}_b + k_s x_b - (b_s + \mu_a)\dot{x}_a - k_s x_a\Big), \tag{6.19}$$

$$\ddot{x}_b = \frac{1}{m_b}\Big(b_s\dot{x}_a + k_s x_a + \frac{\mu_p}{l_p}\dot{\theta} + g m_p \theta - (b_s + \mu_b)\dot{x}_b - k_s x_b\Big), \tag{6.20}$$

$$\ddot{\theta} = \frac{1}{l_p m_b}\Big((b_s + \mu_b)\dot{x}_b + k_s x_b - b_s\dot{x}_a - k_s x_a - (1 + \frac{m_b}{m_p})\frac{\mu_p}{l_p}\dot{\theta} - g(m_p + m_b)\theta\Big) \tag{6.21}$$

6.2.3 Model in State-Space Representation

Mathematical model can be now represented as linear time invariant (LTI) model in the state-space representation. In order to do that 2^{nd} order differential equations (6.19), (6.20) and (6.21) have to be converted into six 1^{st} order differential equations. For that it is necessary to reformulate state variables as follows

$$\begin{aligned}
x_1(t) &= x_a(t),\\
x_2(t) &= \dot{x}_a(t),\\
x_3(t) &= x_b(t),\\
x_4(t) &= \dot{x}_b(t),\\
x_5(t) &= \theta(t),\\
x_6(t) &= \dot{\theta}(t),
\end{aligned}$$

The state-space model is then given as follows

$$\dot{x}_1 = x_2, \tag{6.22}$$

$$\dot{x}_2 = \frac{1}{m_a}\Big(F + b_s x_4 + k_s x_3 - (b_s + \mu_a)x_2 - k_s x_1\Big), \tag{6.23}$$

$$\dot{x}_3 = x_4, \tag{6.24}$$

$$\dot{x}_4 = \frac{1}{m_b}\Big(b_s x_2 + k_s x_1 + \frac{\mu_p}{l_p}x_6 + g m_p x_{\mathrm{V}} - (b_s + \mu_b)x_4 - k_s x_3\Big), \tag{6.25}$$

$$\dot{x}_5 = x_6, \tag{6.26}$$

$$\begin{aligned}\dot{x}_6 = \frac{1}{l_p m_b}\Bigg(&(b_s + \mu_b)x_4 + k_s x_3 - b_s x_2 - k_s x_1 - (1 + \frac{m_b}{m_p})\frac{\mu_p}{l_p}x_6 \\ &- g(m_p + m_b)x_5\Bigg).\end{aligned} \tag{6.27}$$

The state-space representation of the derived mathematical model in the matrix form is then as follows

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{6.28}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \tag{6.29}$$

$$\mathbf{x} = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{pmatrix}^T, \tag{6.30}$$

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{k_s}{m_a} & -\frac{b_s+\mu_a}{m_a} & \frac{k_s}{m_a} & \frac{b_s}{m_a} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k_s}{m_b} & \frac{b_s}{m_b} & -\frac{k_s}{m_b} & -\frac{b_s+\mu_b}{m_b} & \frac{m_p g}{m_b} & \frac{\mu_b}{l_p m_b} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{k_s}{l_p m_b} & -\frac{b_s}{l_p m_b} & \frac{k_s}{l_p m_b} & \frac{b_s}{l_p m_b} & -\frac{(m_b+m_p)g}{l_p m_b} & \frac{-\mu_p(1+\frac{m_b}{m_p})}{l_p^2 m_b} \end{pmatrix}, \tag{6.31}$$

$$u = \begin{pmatrix} \frac{F}{s_a} \end{pmatrix}, \tag{6.32}$$

$$\mathbf{B} = \begin{pmatrix} 0 & \frac{s_a}{m_a} & 0 & 0 & 0 & 0 \end{pmatrix}^T, \tag{6.33}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \tag{6.34}$$

$$\mathbf{D} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T, \tag{6.35}$$

where $\mathbf{x}$ is the state vector, $\mathbf{A}$ the state matrix, u the input, $\mathbf{B}$ the input matrix, $\mathbf{C}$ the output matrix and $\mathbf{D}$ the direct transmission matrix for measured state variables x_a, x_b and θ.

6.3 Compensation of System's Nonlinearities

During my experiments with a laboratory setup, was it found out that the influence of the static friction of the cart a on system's dynamic is not negligible and causes a significant non-linear behavior of the controlled system. I have decided to use method *Friction Observer* (FO), described in the chapter 2.4.1 and in the article [12], in order to compensate system's nonlinearities. In the ideal case, the system's nonlinearities are compensated by FO, therefore, the system's behavior can be assumed to be almost linear.

In order to compensate friction a mathematical model of friction has to be first determined. A chosen friction model is the classical static friction model, with coulomb and stiction components. The model is graphically represented in Figure 6.4.

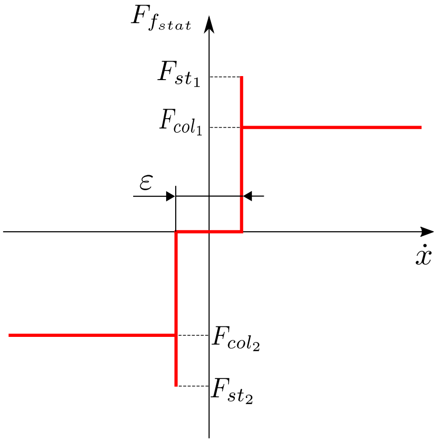


Figure 6.4. Graphical representation of the used friction model described by equations (6.36), (taken and modified from [12], Fig. 5).

And described by following equations, ([12], p. 8, eq. 5)

$$F_{f_{stat}}(\dot{x}_a, F_e) = \begin{cases} F_{col_1} \cdot sign(\dot{x}_a), & \dot{x}_a \geq \frac{\varepsilon}{2} \\ F_{st_1} \cdot sign(F_e), & \dot{x}_a \geq \frac{\varepsilon}{2} \; and \; |F_e| \geq F_{st} \\ F_{col_2} \cdot sign(\dot{x}_a), & \dot{x}_a < -\frac{\varepsilon}{2} \\ F_{st_2} \cdot sign(F_e), & \dot{x}_a \leq -\frac{\varepsilon}{2} \; and \; |F_e| \geq F_{st} \\ F_e, & \dot{x} \in (-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}) \; and \; |F_e| \leq F_{st} \end{cases}, \tag{6.36}$$

where $F_{f_{stat}}$ is the modeled static friction force, F_{col_1}, F_{col_2} the coulomb friction forces, F_{st_1}, F_{st_2} the stiction friction forces and, F_e the external force applied to the object. It is necessary to implement *Dead Zone* with an interval ε, in order to prevent excessive switching of the friction force of the friction model implemented in computer programs such as MATLAB. Parameters of the static friction model (6.36) were determined experimentally as follows

$$F_{st_1} = 4.8 \ (N), \ F_{col_1} = 0.1 \ (N),$$
$$F_{st_2} = 5.52 \ (N), \ F_{col_2} = 0.1 \ (N),$$
$$\varepsilon = 0.1 \ (m.s^{-1}).$$

During experiments with friction compensation was it found out that the controlled system is strongly influenced by another source of nonlinearity, that manifested as a significantly smaller applied control force F on the cart a for lower values of the control signal $u \in (-4, 4)$ (V).

The cause of this nonlinearity is still unknown, however, my presumption is that the transfer of the control signal u to the control force F is not linear $(F = S_a \cdot u)$ but nonlinear (that might be caused by an actuator or by transfer of actuator's torque to the longitudinal movement of carts). Experimentally was it found that this nonlinearity can be sufficiently approximated by a function that is shown in Figure 6.5.

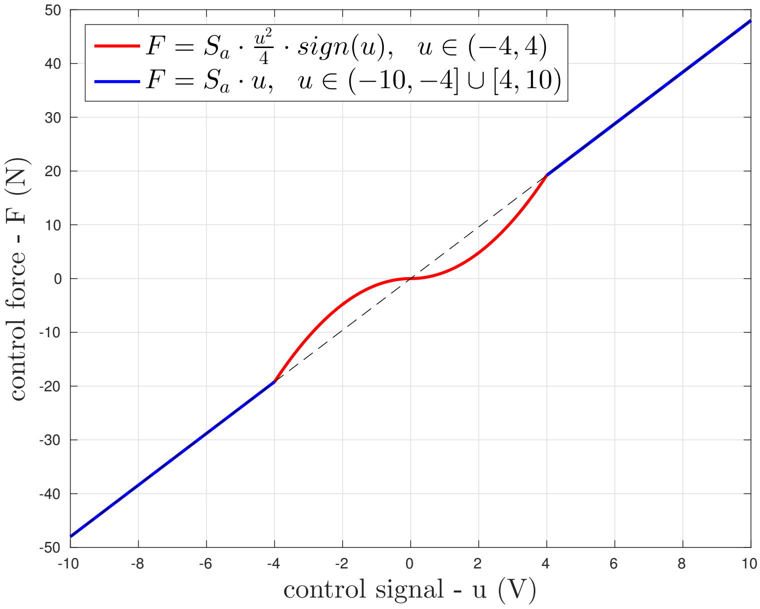


Figure 6.5. Approximation of the nonlinearity of the transfer of the control signal u to the control force F.

Additional force F_{act} to compensate nonlinearities in the interval $u \in (-4, 4)$ is defined as follows

$$F_{act}(u) = \begin{cases} S_a \cdot (|u| - \frac{u^2}{4}) \cdot sign(u), & u \in (-4, 4) \\ 0, & u \in (-10, -4] \cup [4, 10) \end{cases} \tag{6.37}$$

Both nonlinearities, caused by the friction and by the actuator, can be compensated simply by summing their parts. The total estimated compensation control force $\hat{F}_{comp}$ and the compensation control signal u_{comp} are then defined as follows

$$\hat{F}_{comp} = \hat{F}_{f_{stat}} + \hat{F}_{act}, \tag{6.38}$$

$$u_{comp} = u_{f_{stat}} + u_{act} = \frac{\hat{F}_{f_{stat}}}{S_a} + \frac{\hat{F}_{act}}{S_a}, \tag{6.39}$$

where $\hat{F}_{f_{stat}}$ and $\hat{F}_{act}$ are estimated compensation forces and S_a is the actuator's static gain. The used compensation scheme is shown in Figure 6.6, where $A(s)$ denotes the actuator and $C(s)$ the controller. The designed friction compensator is implemented in Simulink's block *MATLAB Function* its source code can be found in the appendix A.1 or enclosed CD.

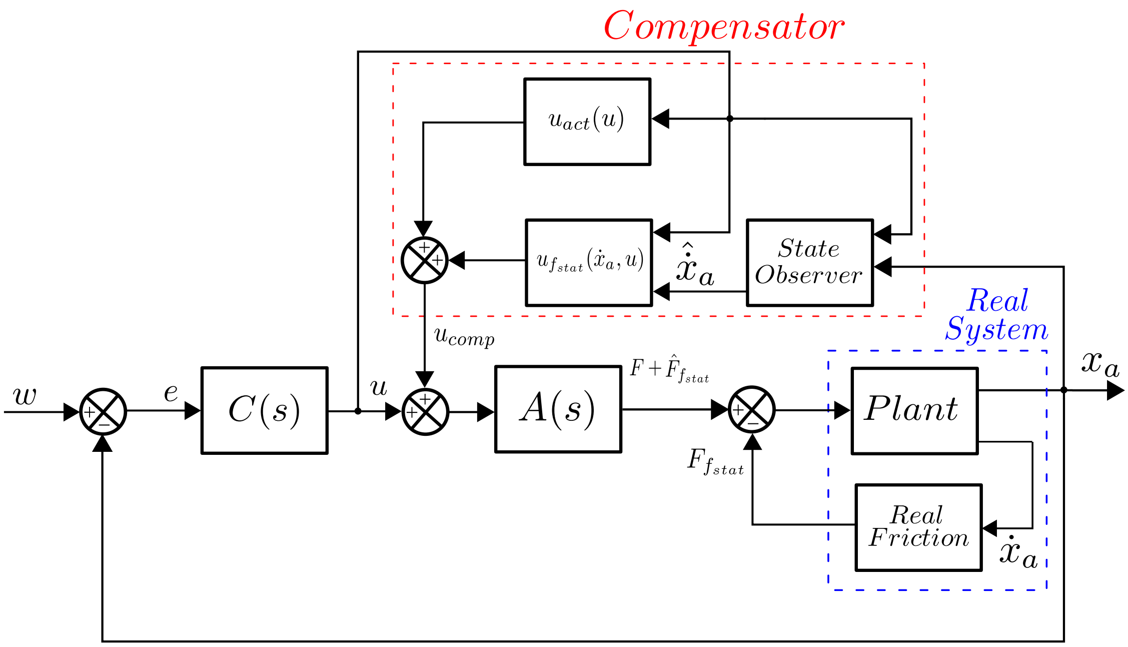


Figure 6.6. Designed scheme to compensate nonlinearities.

6.3.1 State Observer for Friction Compensation

In this chapter, the design procedure of the state observer will be outlined. The measurement of the velocity $\dot{x}_a$ of the cart a was not available and for that reason I have decided to estimate $\dot{x}_a$ by means of the state observer of the system described by equations (6.31), (6.33), (6.34) and (6.35). The state observer is described as follows

$$\begin{aligned}\dot{\hat{\mathbf{x}}} &= \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}u + \mathbf{K_e}(y - \mathbf{C}\hat{\mathbf{x}}) \\ &= (\mathbf{A} - \mathbf{K_e}\mathbf{C})\hat{x} + \mathbf{B}u + \mathbf{K_e}y,\end{aligned} \tag{6.40}$$

where $\mathbf{K_e}$ is the observer gain. Design of the observer gain was an iterative process since the parameters of the matrix of dynamics $\mathbf{A}$ (6.31) were initially unknown. In the first iteration some of the parameters of (6.31) were determined by means of physical measurement (masses of carts and pendulum, length of the pendulum), while the rest were guessed (kinematic frictions, spring stiffens and damping) and subsequently the observer gain $\mathbf{K_e}$ was calculated.

In the next iterations, new parameters of (6.31) were determined based on the results of pulse response identification of the controlled system with an applied friction observer, that was designed in the previous iteration. This process was repeated until a good correspondence between the identified model and the measured results was found. Before design of state-space observer, first the system observability has to be determined. A system is observable if its observability matrix $\mathbf{O}$ is full rank. Next we determine the observer gain matrix $\mathbf{K_e}$ so that the error dynamics $\dot{\mathbf{e}}$ are asymptotically stable with a sufficient speed of the response. The objective of observer design is then defined as follows

$$\mathbf{e}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t), \tag{6.41}$$

$$\lim_{t\to\infty} \mathbf{e}(t) = 0, \tag{6.42}$$

the error $\mathbf{e}$ will approach zero if the matrix $(\mathbf{A} - \mathbf{K_e C})$ is stable. The speed of convergence depends on the placement of poles of the matrix $(\mathbf{A} - \mathbf{K_e C})$. We can make speed of convergence faster by placing poles of the matrix $(\mathbf{A} - \mathbf{K_e C})$ further to the left from the imaginary axis. An example of observer design procedure for identified system parameters will be presented in Chapter 6.4.2.

6.4 System Identification

Parameters of the derived mathematical linear model were identified experimentally by comparing pulse responses of the linear model with a responses of the real system. Model's parameters were tuned so that dynamic properties of the linear system match the dominant dynamics of the real system. Identified system's parameters are given in Table 6.1.

Parameter	Value	Unit
m_a	1.8	(kg)
m_b	3.25	(kg)
m_p	2	(kg)
μ_a	14.5	$(N.s.m^{-1})$
μ_b	3	$(N.s.m^{-1})$
μ_p	0.01872	$(N.s.m^{-1})$
l_p	0.67	(m)
k_s	300	$(N.m^{-1})$
b_s	1	$(N.s.m^{-1})$
S_a	4.8	$(N.V^{-1})$

Table 6.1. Identified parameters of the LTI model.

The results of the system identification are presented in the following Figures 6.7, 6.8, 6.9 and 6.10, where pulse responses of the identified linear model (with parameters from Table 6.1) with pulse responses of the real system are compared. First graph shows response of the cart x_a, the response of the cart x_b is in the second graph is, the position difference between carts $x_a - x_b$ is in the third graph, the displacement of the pendulum p is in the fourth graph and the applied control signal u is in the fifth graph.

In Figures 6.7, 6.8 it can be seen that measured configuration without compensator does not correspond to the dynamics of the linear model very well. However, influence of system's nonlinearities diminishes with a higher value of the pulse signal as it can be seen in Figures 6.9 and 6.10. In all cases the designed compensator efficiently compensates system's nonlinearities for the control signal of values $u \in [0, 4)$ without overcompensation. More importantly model dynamics correspond with real system dynamics. Therefore, identified mathematical model is suitable for the control design of the controlled system.

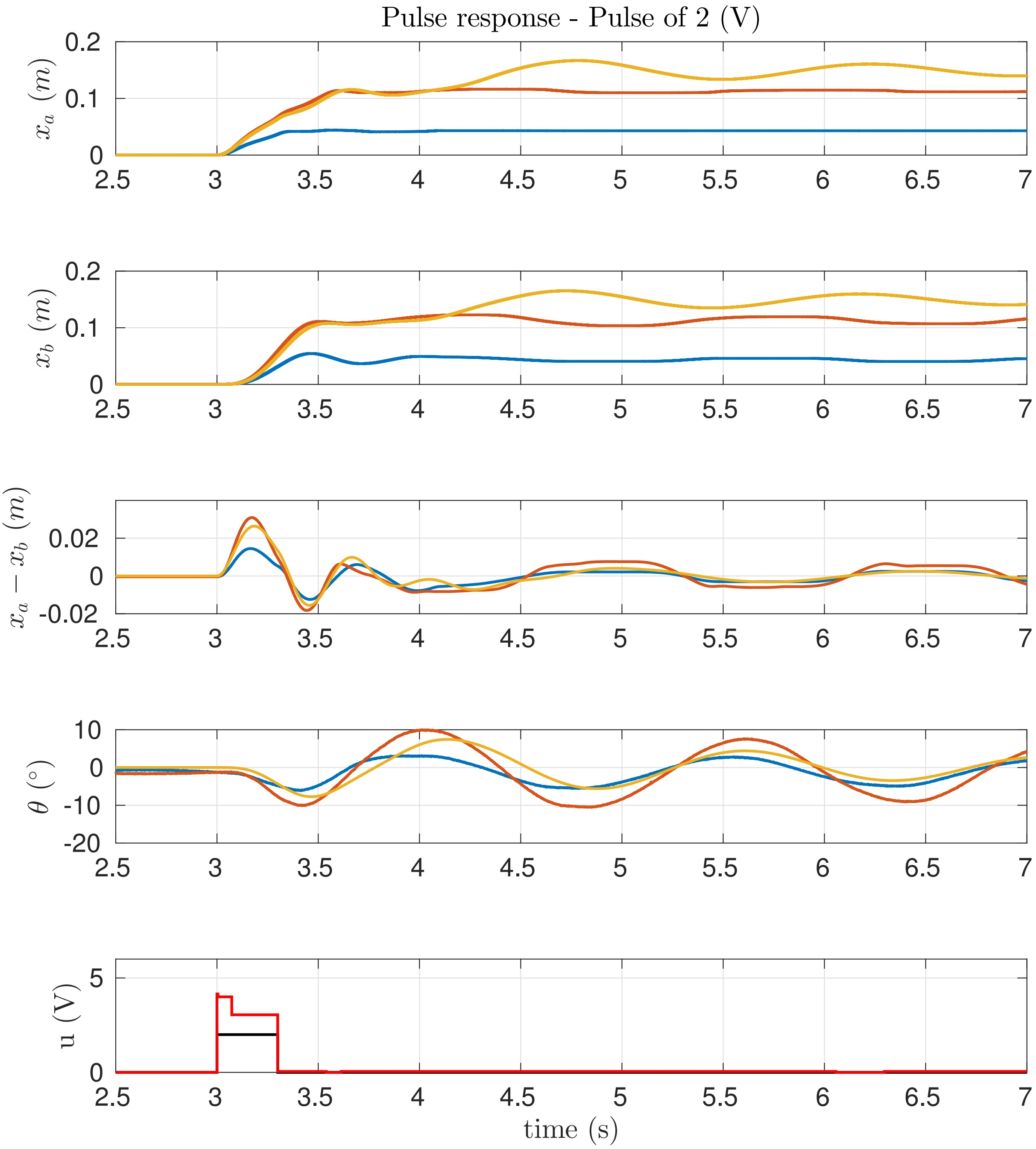


Figure 6.7. Pulse response for $u = 2$ (V), simulated (yellow), measured without compensator (blue), measured with compensator (orange), control signal (black), control signal with compensation part (red).

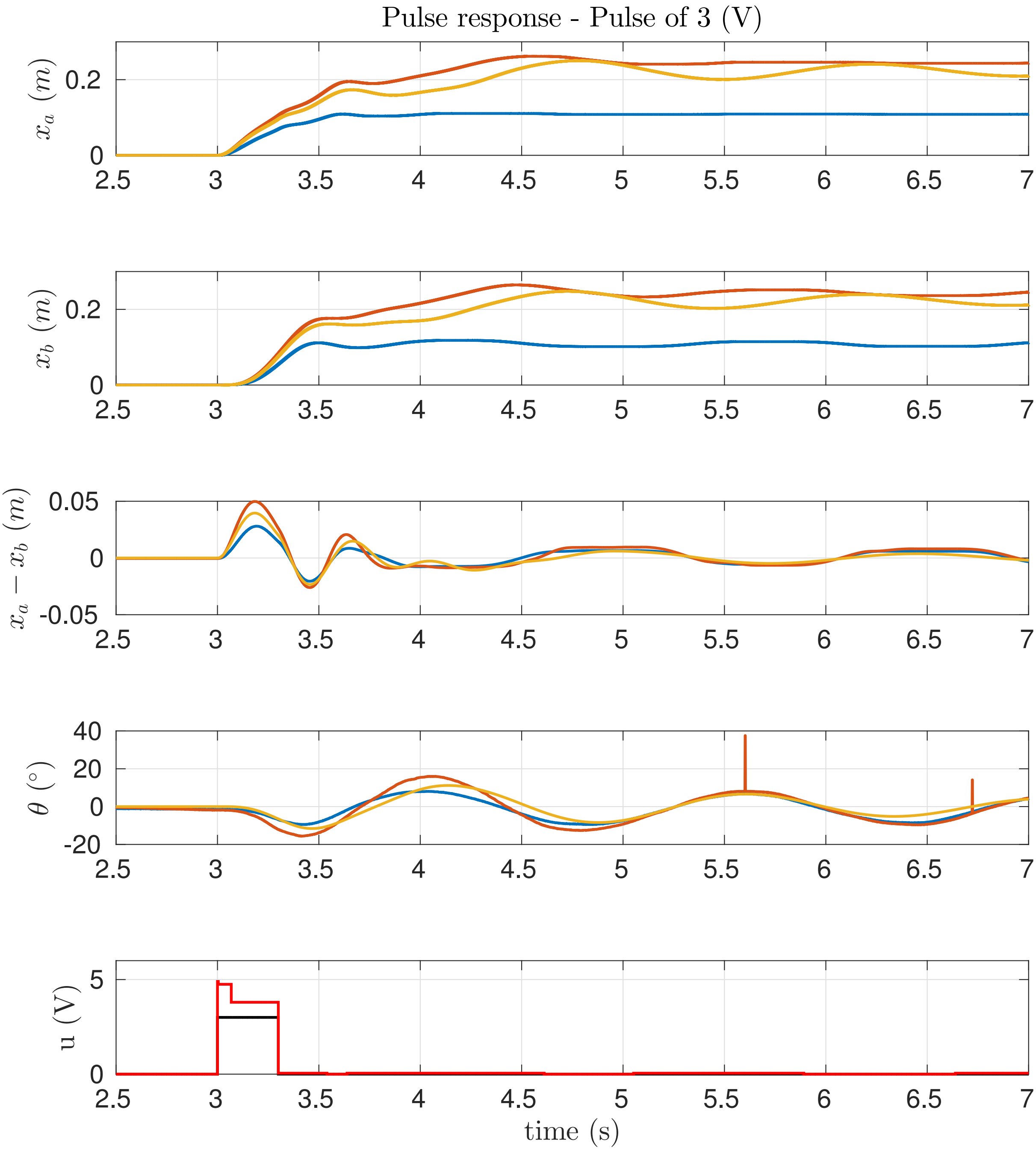


Figure 6.8. Pulse response for $u = 3$ (V), simulated (yellow), measured without compensator (blue), measured with compensator (orange), control signal (black), control signal with compensation part (red).

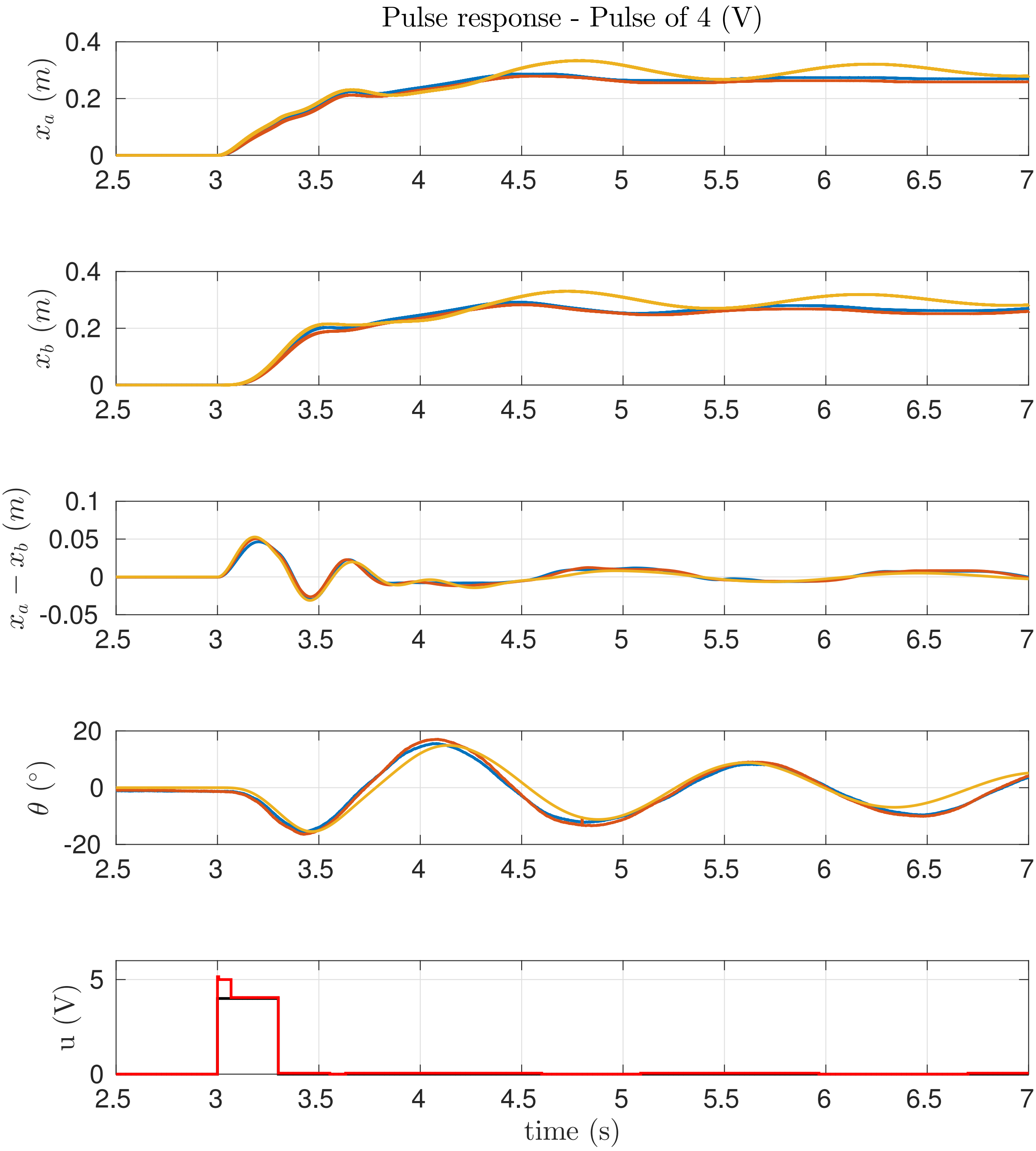


Figure 6.9. Pulse response for $u = 4$ (V), simulated (yellow), measured without compensator (blue), measured with compensator (orange), control signal (black), control signal with compensation part (red).

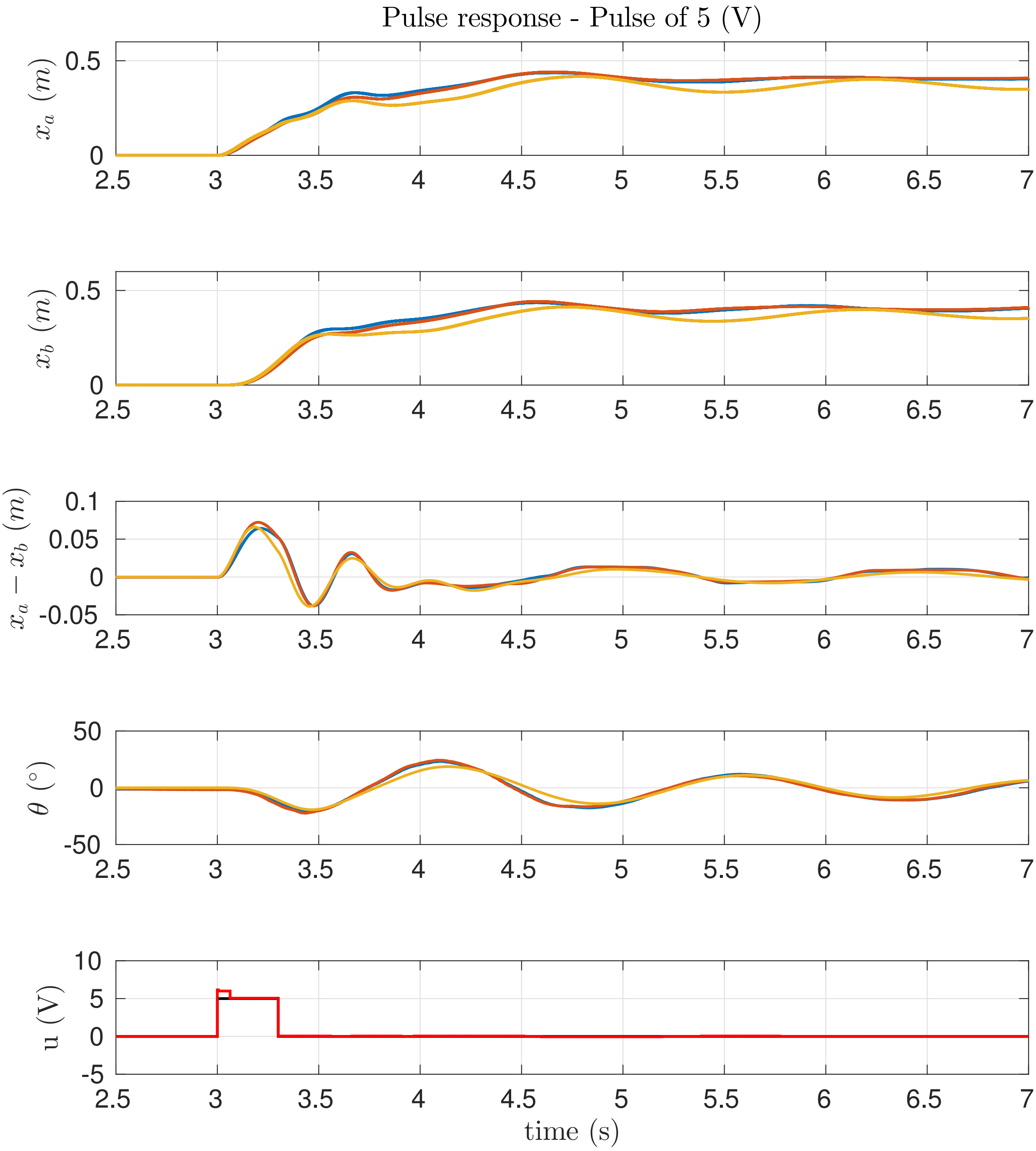


Figure 6.10. Pulse response for $u = 5$ (V), simulated (yellow), measured without compensator (blue), measured with compensator (orange), control signal (black), control signal with compensation part (red).

The static gain of the actuator S_a was determined from nominal torque $M_n = 1.27\ (N.m)$ for input of $10\ (V)$ of the servomotor Estun-EMJ-04 and from the measured diameter of the belt pulley $d_{bp} = 0.055\ (m)$. For simplicity, a linear characteristic of the transfer of u to M_n was assumed, so that $u = 1\ (V)$ corresponds to $M_n = 0.127\ (N.m)$.

$$M_n = F \cdot \frac{d_{bp}}{2},$$

$$F = \frac{M_n \cdot 2}{d_{bp}} = \frac{0.127 \cdot 2}{0.053} = 4.8\ (N),$$

the static gain is then $S_a = 4.8\ (N.V^{-1})$. The poles of the state matrix **A** (6.31) with identified parameters from Table 6.1 are

$$\begin{aligned} p_{1,2} &= -3.03 \pm 15.5i, \\ p_{3,4} &= -0.34 \pm 4.3i, \\ p_5 &= -3.12, \\ p_6 &= 0. \end{aligned}$$

The system is astatic since pole p_6 has zero value. This fact corresponds to the properties of the real system. Poles in s-plane are shown in Figure 6.11.

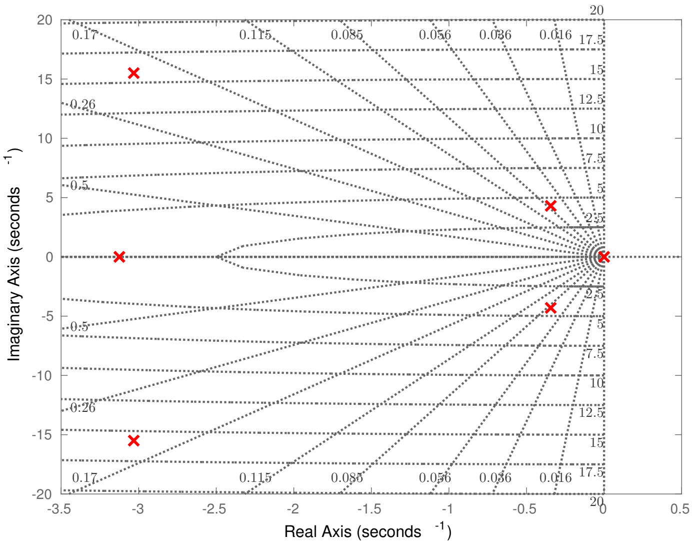


Figure 6.11. Poles of the state matrix **A** with identified parameters.

6.4.1 Identification of Input Delay

Input delay τ_{input} of the response of the cart a was identified by comparing time of the control signals u with the cart a response. Measured results are shown in Figure 6.12.

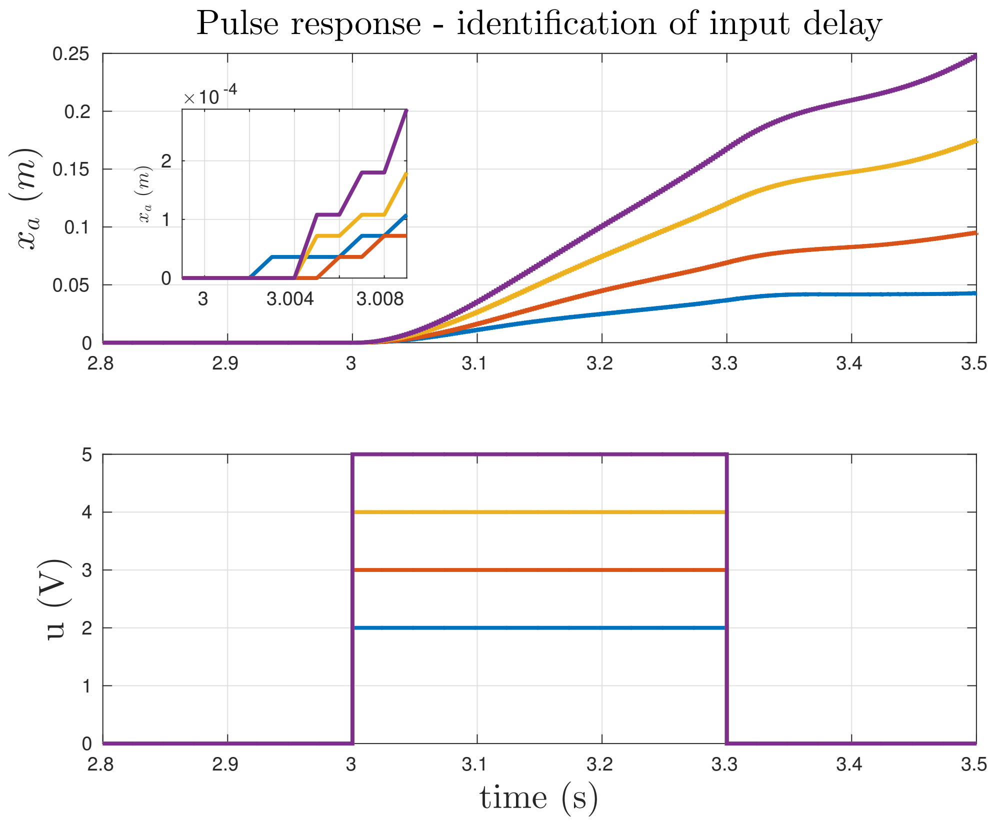


Figure 6.12. Input delay for different pulse responses, $u = 2$ (V) (blue), $u = 3$ (V) (orange), $u = 4$ (V) (yellow), $y = 5$ (V) (purple).

Based on the measured results, the input delay is assumed to be $\tau_{input} = 5$ (ms).

6.4.2 Design of State Observer

The identified parameters in Table 6.1 and equations (6.31), (6.33) were used to design observer's gain $\mathbf{K_e}$. Only the output of the measured position x_a was used for state estimation. Therefore, matrices **C** and **D** had to be modified as follows

$$\mathbf{C_{ob}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \tag{6.43}$$

$$\mathbf{D_{ob}} = \begin{pmatrix} 0 \end{pmatrix}, \tag{6.44}$$

The observability matrix was calculated and checked in program MATLAB by following commands.

```
n; % total state variables
sys_ss = ss(A,B,C,D); % define state-space model
obs = obsv(sys_ss); % calculate observability matrix
rank_of_obs = rank(ob); % calculate rank
rank_of_obs == n; % check condition, if true system is observable
```

$$rank(\mathbf{O}) = 6, \tag{6.45}$$

the rank of the matrix $\mathbf{O}$ is equal to the number of state variables $n = 6$, thus, the system is observable. The poles of the observer gain $\mathbf{K_e}$ were determined iteratively so that the observer accurately matches the dynamics of the real system, chosen poles are

$$\begin{aligned} p_{1,2} &= -3.2 \pm 16i, \\ p_{3,4} &= -0.38 \pm 4.6i, \\ p_5 &= -10, \\ p_6 &= -15, \end{aligned}$$

$$\mathbf{P} = \begin{pmatrix} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 \end{pmatrix}, \tag{6.46}$$

finally the observer's gain $\mathbf{K_e}$ was determined by MATLAB command `place` for matrices (6.46), (6.43), (6.31), by following MATLAB commands

```
P = [-p1, -p2, -p3, -p4, -p5, -p6]; % poles are at different location
K_e = place (A', C', P)';
```

$$\mathbf{K_e} = \begin{pmatrix} 21.31 & 59.6 & 22.28 & 107.97 & 32.52 & 62.07 \end{pmatrix}^T, \tag{6.47}$$

the designed observer can be cross-checked by command `eig(A - K_e *C_ob)`, this command should return a vector of poles equal to the chosen poles (6.46). Comparison of the measured position x_a with an estimated state $\hat{x}_a$ is shown in Figure 6.13.

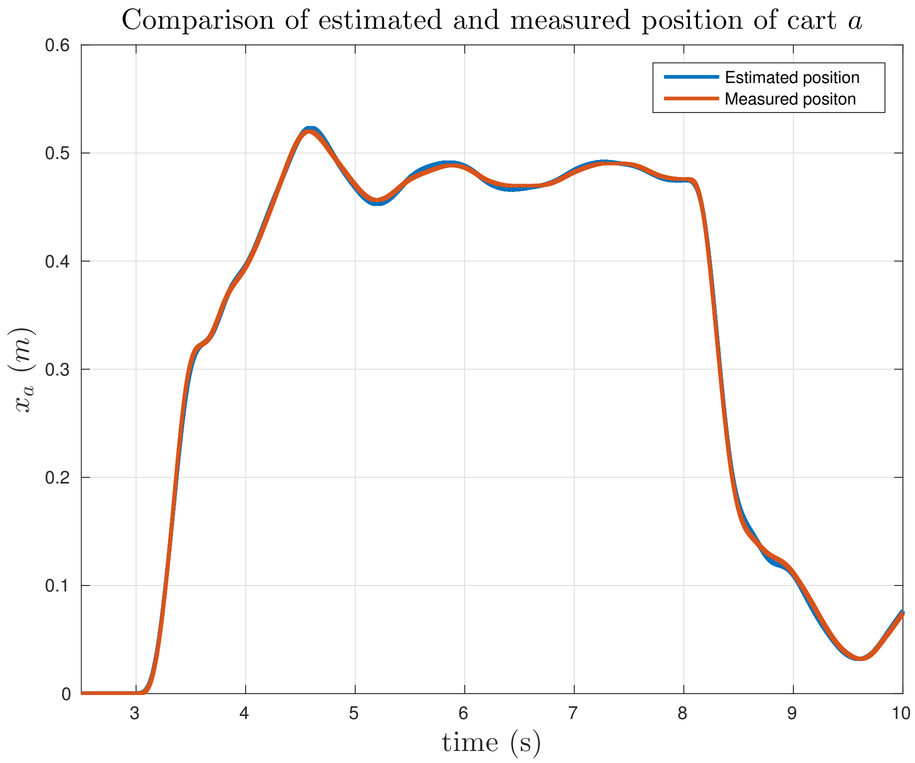


Figure 6.13. Comparison of observed outputs and measured values, observed (blue), measured (orange).

The measured position x_a and estimated position $\hat{x}_a$ are almost identical and the designed observer meets criteria of stability and fast error convergence.

6.5 Identification of oscillation mode of flexible subsystem

In order to design an effective signal shaper first it is important to precisely identify the dominate oscillation mode of the flexible system, because the effectiveness of the designed signal shaper solely depends on the accuracy of the match between the identified mode and the real mode. The oscillation mode was experimentally identified by fixing position of the cart a and measuring excited oscillation of the pendulum connected to the cart b. A mechanical scheme of the measured flexible part is shown in Figure 6.14.

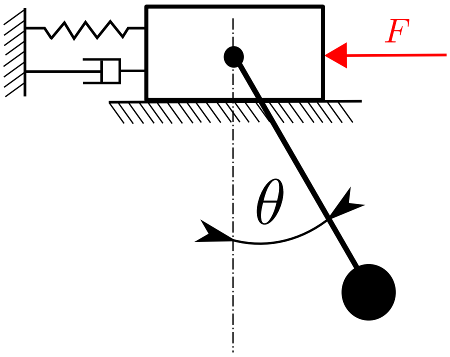


Figure 6.14. Mechanical scheme of the cart b and the pendulum p.

6.5.1 Analytical Identification

The pendulum's oscillation mode to be compensated was analytically determined from the adjusted state matrix $\mathbf{A}_{\mathsf{red}}$ that does not contain states of the cart a and is derived from (6.31) with identified parameters in Table 6.1. This matrix is defined as follows

$$\mathbf{A}_{\mathsf{red}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_s}{m_b} & -\frac{b_s+\mu_b}{m_b} & \frac{m_p g}{m_b} & \frac{\mu_p}{l_p m_b} \\ 0 & 0 & 0 & 1 \\ \frac{k_s}{l_p m_b} & \frac{b_s}{l_p m_b} & -\frac{(m_b+m_p)g}{l_p m_b} & \frac{-\mu_p(1+\frac{m_b}{m_p})}{l_p^2 m_b} \end{pmatrix}, \tag{6.48}$$

$$\mathbf{A}_{\mathsf{red}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{300}{3.25} & -\frac{1+3}{3.25} & \frac{29.81}{3.25} & \frac{0.01872}{0.673.25} \\ 0 & 0 & 0 & 1 \\ \frac{300}{0.673.25} & \frac{1}{0.673.25} & -\frac{(3.25+2)9.81}{0.673.25} & \frac{-0.01872(1+\frac{3.25}{2})}{0.67^2 3.25} \end{pmatrix}, \tag{6.49}$$

the mode $r_{m_{1,2}}$ is a pair of dominant poles of $\mathbf{A}_{\mathsf{red}}$. This mode was determined by MATLAB's command `eig(A_red)` and is shown in Figure 6.15.

$$r_{m_{1,2}} = -0.065 \pm j3.63. \tag{6.50}$$

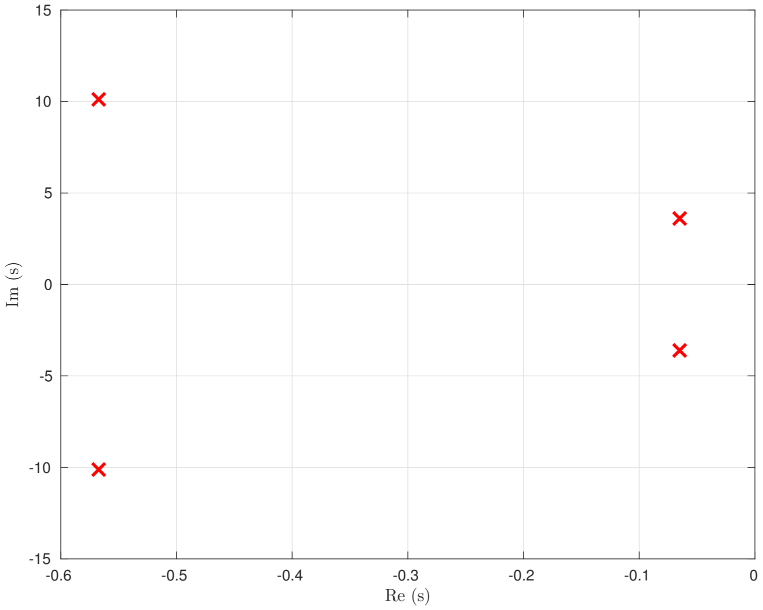


Figure 6.15. Poles of the reduced state matrix $\mathbf{A}_{\text{red}}$.

6.5.2 Experimental Identification

Experimental model identification was performed, in order to obtain more accurate parameters (ω_0, ζ) of the pendulum oscillatory mode $r_{1,2}$. First pulse response of the adjusted system (that is shown in Figure 6.14) was measured. Measured data were then used to identify parameters of the estimated pendulum's transfer function structure. The free body diagram of the pendulum is shown in Figure 6.16.

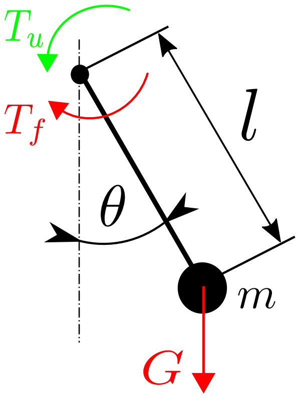


Figure 6.16. Free body diagram of the pendulum.

where T_u is the input torque, $T_f = \mu_f \dot{\theta}$ the friction torque and $G = mg$ the gravitational force. The pendulum p is a flexible subsystem $F(s)$ with an oscillation mode $r_{1,2}$ to be compensated. The structure of the flexible system's transfer function $F(s)$ for small angles $sin(\theta) \approx \theta$, was derived as follows

$$\begin{aligned} ml^2\ddot{\theta}(t) + T_f + Gsin(\theta(t)) &= T_u \\ ml^2\ddot{\theta}(t) + \mu_f\dot{\theta}(t) + mg\theta(t) &= T_u, \end{aligned} \tag{6.51}$$

The transfer function $F(s)$ of pendulum's mathematical model is of second order and is described as follows

$$F(s) = \frac{b_0}{a_2 s^2 + a_1 s + a_0}, \tag{6.52}$$

system's parameters a_2, a_1, a_0 and b_0 were determined by a MATLAB's function `s = tftest(DATA, NP, NZ)`, where the first argument `DATA` is a `iddata` object containing the input and the measured output (in the time domain), and arguments `NP` (number of poles) and `NZ` (number of zeros) define model's structure. The output `sys` is the transfer function of the estimated model. The structure of model (6.52) has two poles `NP = 2` and no zeros `NZ = 0`. The results of identification are given in Figure 6.17 and in Table 6.2.

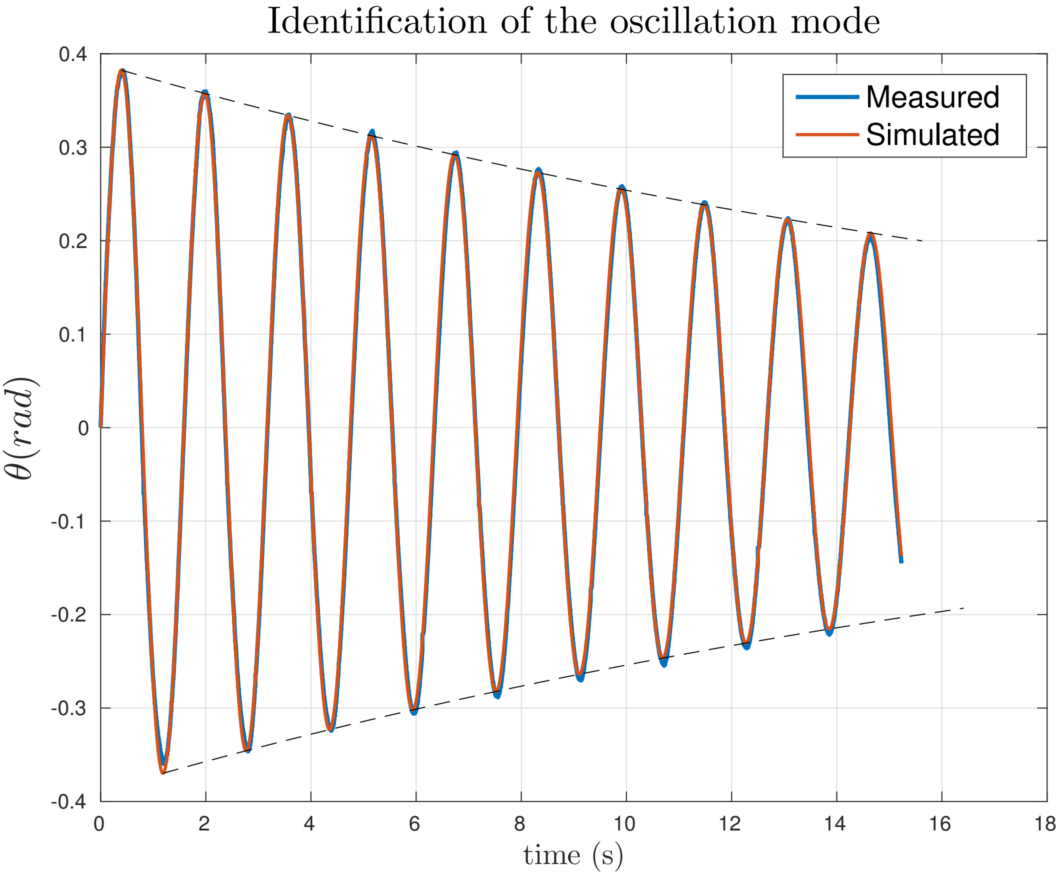


Figure 6.17. Identification of pendulum's oscillation mode, red - result of simulation, blue - result of measurement.

a_0	a_1	a_2	b_0
1	0.08526	15.75	1.544

Table 6.2. Identified parameters of the model (6.52).

The model's fit to estimation data is 96.23 (%) that gives fair agreement between simulated and measured responses. The natural frequency and damping ratio can be calculated as follows

$$\omega_0 = \sqrt{a_2} = \sqrt{15.75} = 3.9683 \; (rad.s^{-1}), \tag{6.53}$$

$$\zeta = \frac{a_1}{2\omega_0} = \frac{0.08526}{2 \cdot 3.96} = 0.0107 \; (1), \tag{6.54}$$

and the pendulum's oscillation mode $r_{1,2}$ is derived as follows

$$\beta = \omega_0 \cdot \zeta = 3.96 \cdot 0.0107 = 0.042 \; (rad.s^{-1}), \tag{6.55}$$

$$\Omega = \omega_0 \cdot \sqrt{1 - \zeta^2} = 3.9683 \cdot \sqrt{1 - 0.0107^2} = 3.968 \; (rad.s^{-1}), \tag{6.56}$$

$$r_{1,2} = -\beta \pm j\Omega = -0.042 \pm j3.968. \tag{6.57}$$

The pendulum's mode can also be obtained directly by the MATLAB's command `r_12 = pole(tftest(iddata,2,0))`.

It can be seen that there is a slight mismatch between analytically identified mode $r_{m_{1,2}} = -0.065 \pm j3.63$ and experimentally identified mode $r_{1,2} = -0.042 \pm j3.968$. This disagreement is probably caused by approximation properties of the linear model that is able to capture only dominant components of the real non-linear system. Experimentally identified mode will be used to design signal shaper since it describes real mode more accurately.

Chapter 7
Design of Internet Control Architecture

The Networked Control System for the controlled laboratory set-up, described in the chapter 6.1, was implemented according to the outlined theory in Chapter 5.2. Basic diagram of NCS's signal flow is shown in Figure 7.1.

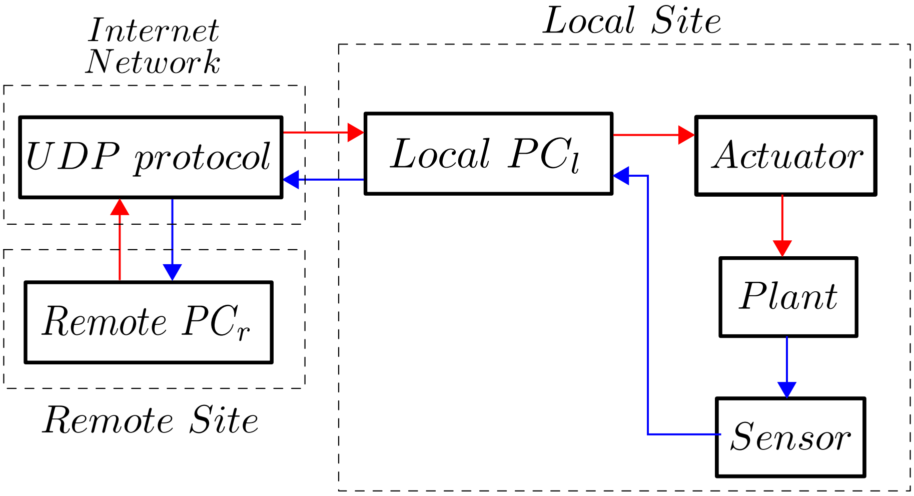


Figure 7.1. Signal flow of the designed NCS, feedback siganl (blue), control signal (red).

The signals are transmitted between the local computer PC_l and the remote computer PC_r via the UDP communication protocol, this means that the feedback loop is closed through the Internet network. Signals are received, sent and processed on both computers by program MATLAB/Simulink. The simulink program on the remote computer acts as a controller, i.e., computes and sends a control signal while simultaneously receives a feedback signal. Meanwhile, the simulink program on the local computer calculates value of the friction compensation signal and transmits the resultant control signal, e.g., sum of received and compensation control signal, to the plant while simultaneously sends collected measured signals.

Program Simulink offers several libraries for the network communication with various communication protocols. The UDP was chosen as the transport layer's communication protocol for the reasons already discussed in the chapter 5.3. The following Simulink's libraries supporting UDP communication were tested (library/blocks):

- DSP System Toolbox (DSPST) / UDP Send, UDP Receive
- Instrument Control Toolbox (ICT) / UDP Send, UDP Receive
- Simulink Desktop Real-Time (SDRT) / Packet Output, Packet Input (for the Simulink versions 8.3/R2014a and newer)
- Real Time Windows Target (RTWT) / Packet Output, Packet Input (for the Simulink versions 8.2/R2013b and older)

The libraries SDRT and RTWT are compatible with each other, i.e., communication via blocks *Packet Input/Output* between the versions 8.2/R2013b and 8.3/R2014a have been established. Surprisingly, the performance in terms of time delay, packet loss and stability of Simulink's libraries were not consistent. That is probably due to their different implementation and purposes.

After several experiments with different combinations of the above mentioned blocks and libraries, I have concluded that blocks *Packet Input/Output* (libraries SDRT and RTWT) are, in terms of connection quality, the most reliable, stable and appropriate for the purposes of NCS. Moreover, these blocks have already included the Simulink Real-Time Kernel that is necessary for a real-time control of the laboratory set-up.

The block diagram of designed NCS is shown in Figure 7.2,

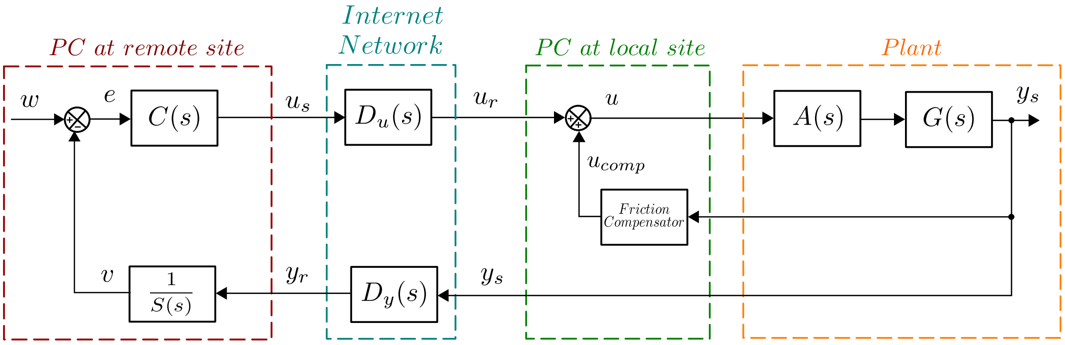
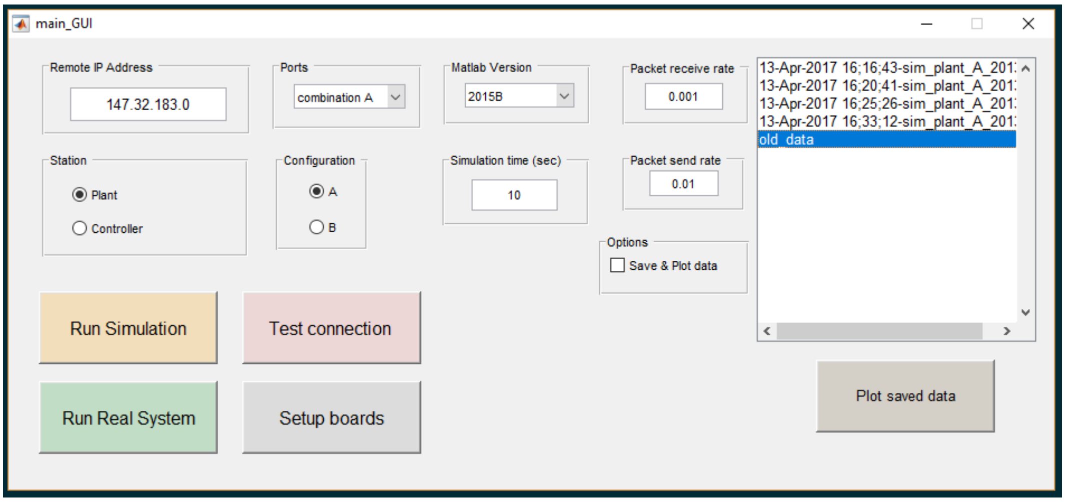


Figure 7.2. Block diagram of the designed NCS.

where $C(s)$ and $\frac{1}{S(s)}$ is the controller and the inverse shaper at the controller site, $D_u(s)$ / $D_y(s)$ are delays at the actuator / sensor node, $A(s)$ is the actuator and $G(s)$ is the controlled system.

A graphical user interface (GUI) was developed MATLAB, for a more convenient connection setup and for identification of communication delays. Designed GUI is shown in Figure 7.3.


main_GUI

Remote IP Address: 147.32.183.0

Ports: combination A

Matlab Version: 2015B

Packet receive rate: 0.001

Station: ● Plant ○ Controller

Configuration: ● A ○ B

Simulation time (sec): 10

Packet send rate: 0.01

Options: □ Save & Plot data

13-Apr-2017 16;16;43-sim_plant_A_201:
13-Apr-2017 16;20;41-sim_plant_A_201:
13-Apr-2017 16;25;26-sim_plant_A_201:
13-Apr-2017 16;33;12-sim_plant_A_201:
old data

Figure 7.3. Designed GUI for control of Networked Control Systems.

The GUI allows to setup Internet Protocol (IP) address of the remote PC, port number, Matlab version, simulation time, model configuration and the rate of sent/received packets. To run NCS in simulation mode; i.e. a simulink model is used instead of the real plant, press button *Run Simulation*, to run NCS in real mode press button *Run*

Real System. The button *Test connection* is for connection test between two devices and the button *Setup board* is used to setup manually blocks *Packet Input/Output*. Measured data are saved and can be displayed by pressing the button *Plot saved data*.

The GUI works for MATLAB versions R2013b, R2014b and R2015b and the GUI's repository is provided on the enclosed CD.

7.1 Time Delay Identification

The packets' transmission delay in the round-trip between the PC_r and PC_l, also known as Round Trip Time (RTT), ([32], p.50), was identified experimentally. The experimental procedure of the time-delay identification consists of sending large number of packets through the network from PC_l to PC_r and back from PC_r to PC_l. The source code of implemented MATLAB function for delay estimation can be found in Appendix A.2. The delay is determined by comparing time-stamp from the received packet (sent from receiver) with current time-stamp. The maximal resolution of time delay measurement is 0.005 (sec).

The program Wireshark [1] was used to determine the length of the packet L_p sent from Simulink's blocks *Packet Input* and *Packet Output*. The packet length was found to be $L_P = 56$ $(byte)$. In order to reduce packet losses, the packet exit period T_{PE} has to be sufficiently larger than the simulation sampling period T_S , ([33], p. 1418). The packet exit period corresponds to the parameter *Sample time* in the block *Packet Output*. For the fixed parameter $T_S = 0.001$ (s), the packet exit period was experimentally determined to be $T_{PE} = 0.01$ (s). The transmission rate of the Ethernet link was assumed to be $R = 10$ $(Mbps)$. The traffic intensity (TI) was then calculated as follows:

$$a = \frac{1}{T_{PE}} = \frac{1}{0.01} = 100\ (packets \cdot s^{-1}),$$

$$TI = \frac{L_P \cdot a}{R} = \frac{56 \cdot 8 \cdot 100}{80 \cdot 10^6} = 5.6 \cdot 10^{-4}\ (1). \tag{7.1}$$

The value of TI is close to zero, therefore, the danger of queuing delay is minimized. Identification of the time-delay in receiving link τ_u and in sending link τ_y (in the case of plant) was based on the value of RTT. For the sake of simplicity the symmetric link was assumed, i.e., same connection conditions in the sending link as in the receiving link. Thus, time-delays τ_u and τ_y are equal to the measured time delay τ_{link} and can be obtained as follows:

$$\tau_{link} = \tau_u = \tau_y = \frac{RTT}{2}. \tag{7.2}$$

The measured results for the connection established between two PC via Wi-Fi and via Ethernet are given in Table 7.1, Figure 7.4, Figure 7.5 and Figure 7.6.

type of connection	$\mu_{\tau_{link}}$ (ms)	$\sigma_{\tau_{link}}$ (ms)	$\tau_{max_{link}}$ (ms)	LP (%)
Wi-Fi	18.37	4.21	100	8.9
Ethernet	15	0.05	15.5	6.65

Table 7.1. Results of identification of link time-delay.

[1] `https://www.wireshark.org`

Interpretation of Table 7.1 is as follows, $\mu_{\tau_{link}}$ is the mean of the measured link delay, $\sigma_{\tau_{link}}$ the standard deviation of the measured link delay, $\tau_{max_{link}}$ the maximal measured link delay and LP the percentage of lost packets.

From the results that are shown in Table 7.1 it is obvious that Ethernet connection is superior to Wi-Fi connection in every compared aspect. This result is not surprising at all, since it is well known that connection via physical medium, such as twisted pair (Ethernet), is more reliable than via wireless technologies (Wi-Fi). Therefore, Ethernet connection is more suitable for real-time control applications and will be used for data transmission in NCS.

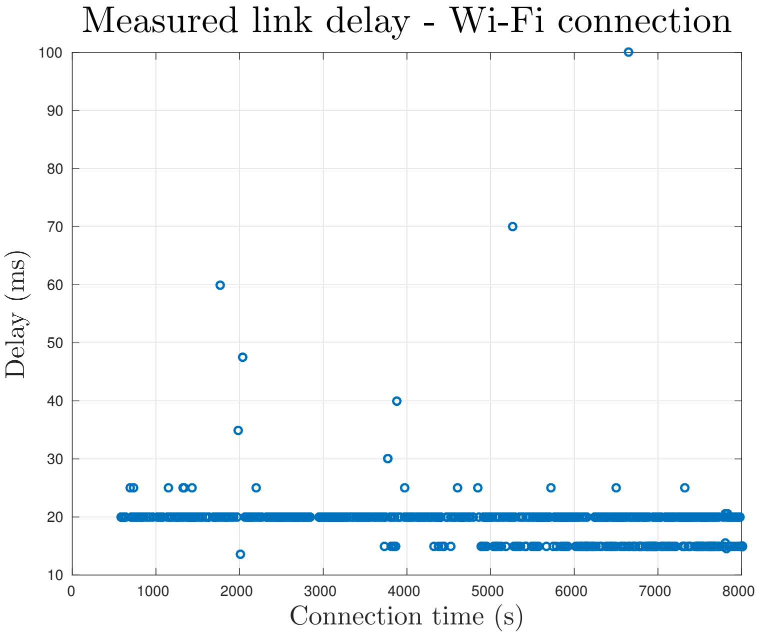


Figure 7.4. Link time-delay for connection established via Wi-Fi.

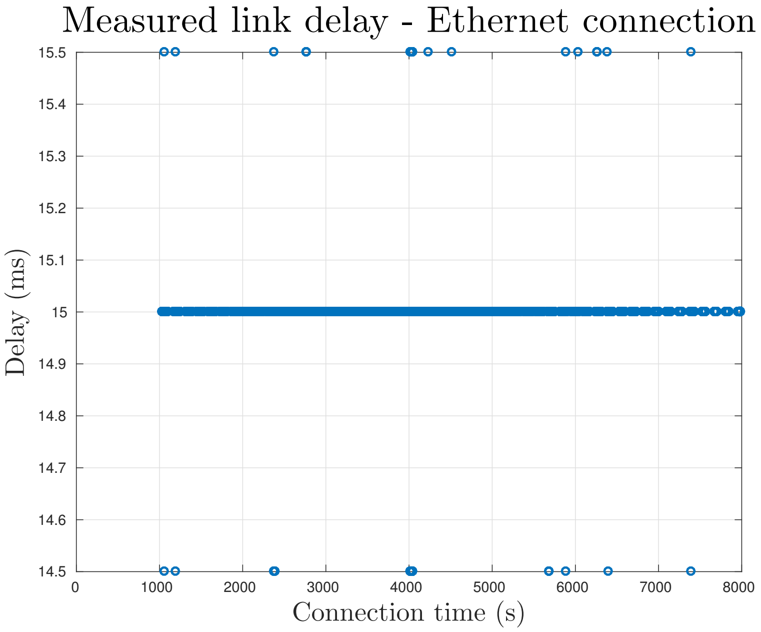


Figure 7.5. Link time-delay for connection established via Ethernet.

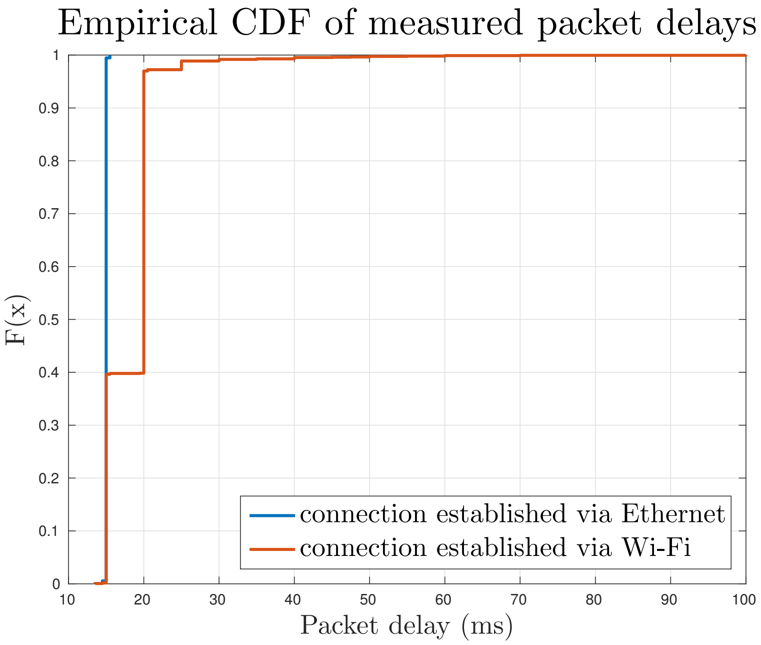


Figure 7.6. Round Trip Time for connection established via Ethernet.

The standard deviation of the link delay is of small value $\sigma_{\tau_{link}} = 0.05$ (ms) and more than 95 (%) measured packet delays are equal to 15 (ms) (see cumulative distribution function of measured delay in Figure 7.6). Therefore, constant $\dot{\tau}_u = \dot{\tau}_y = 0$ and equal $\tau_u = \tau_y$ time-delays in both links are assumed in order to simplify the control design process. The transfer functions of delay of both links are then given as follows:

$$D_u(s) = D_y(s) = e^{-\tau_u s} = e^{-\frac{15.5}{1000}s}. \tag{7.3}$$

During connection tests it was found out that packets sent during an initial phase of the connection (roughly during first two seconds) were affected by a larger delay then packets sent later. My opinion is that first packets are probably more affected by queuing delay caused by Simulink's block *Packet Input.* Simple solution to this problem is to start process received packets after roughly first three seconds of connection, as it is shown in Figure 7.7.

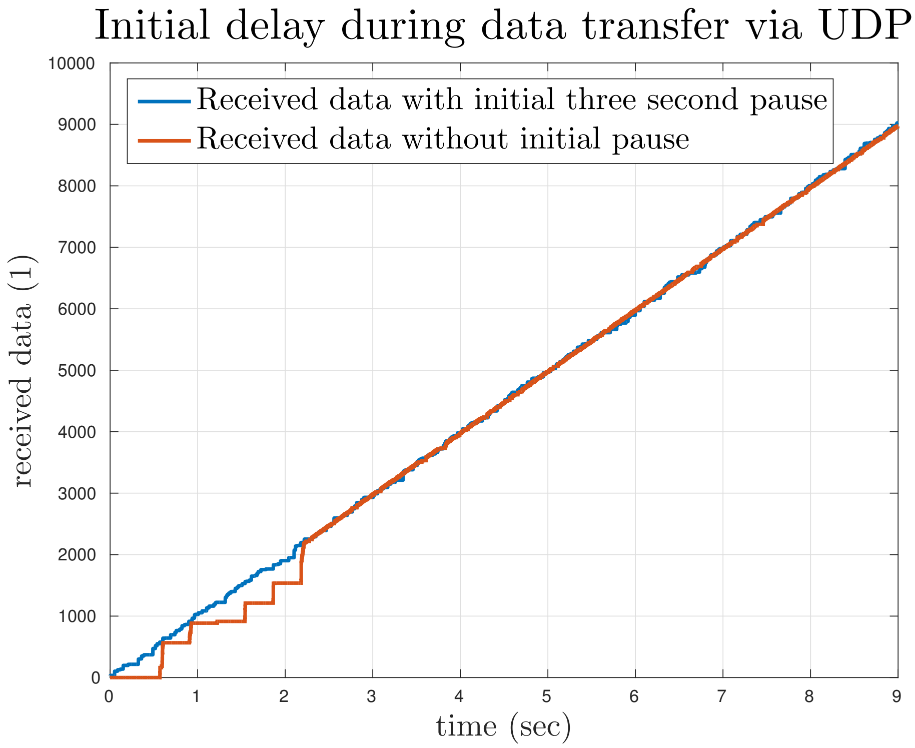


Figure 7.7. Effect of initial delay of UDP packets.

Chapter 8
Control Design

In this chapter overall control design procedure will be presented. The control part consists of an inverse shaper in the feedback and the controller. The control objective is to suppress oscillation of the flexible subsystem via signal shaper and to control position of the cart a by controller that guarantees stability of the closed loop with input and feedback time delays introduced by both communication delays and signal shaper delays.

8.1 Shaper Design

The signal shaper is designed to compensate oscillatory mode $r_{1,2}$ of the pendulum p, that gets excited whenever the controlled cart a is operating. The pendulum oscillatory mode was identified in Chapter 6.5, see (6.57)

$$r_{1,2} = -0.042 \pm j3.968. \tag{8.1}$$

In order to ensure ensure stability of the feedback loop with inverse shaper, it is necessary to design shaper with a retarded spectrum of zeros, as was discussed in Chapter 4. The chosen type of shaper is D_eZV shaper, whose spectrum of zeros is of retarded type

$$\frac{1}{S_{D_eZV}(s)} = \frac{1}{A + \frac{1-A}{T}\frac{1-e^{-sT}}{s}e^{-s\tau}}. \tag{8.2}$$

For chosen interval of equally distributed delay $T \in (0, \frac{\pi}{\Omega}]$ and the target oscillatory mode $s = -\beta + j\Omega = -0.042 + j3.968$, parameters A and τ were calculated as follows

$$T = \frac{\pi}{\Omega} = \frac{\pi}{3.968} = 0.7917\ (s), \tag{8.3}$$

$$m = \left|\frac{1-e^{-sT}}{Ts}\right| = \left|\frac{1-e^{-(-0.042+j3.968)0.7917}}{0.7917(-0.042+j3.968)}\right| = 0.6475, \tag{8.4}$$

$$\varphi = arg\left(\frac{1-e^{-sT}}{Ts}\right) = arg\left(\frac{1-e^{-(-0.042+j3.968)0.7917}}{0.7917(-0.042+j3.968)}\right) = -1.5815\ (rad), \tag{8.5}$$

$$K = me^{\frac{\beta}{\Omega}(\pi+\varphi)} = 0.6475e^{\frac{-0.042}{3.968}(\pi-1.581)} = 0.6584, \tag{8.6}$$

$$A = \frac{K}{1+K} = \frac{0.6584}{1+0.6584} = 0.397, \tag{8.7}$$

$$\tau = \frac{\pi+\varphi}{\Omega} = \frac{\pi - 1.5815}{3.968} = 0.393\ (s). \tag{8.8}$$

The designed inverse signal shaper in s domain is

$$\frac{1}{S_{D_eZV}(s)} = \frac{1}{0.39 + \frac{1-0.39}{0.759}\frac{1-e^{-s0.759}}{s}e^{-s0.3793}}, \tag{8.9}$$

and its spectrum of zeros is shown in Figure 8.1.

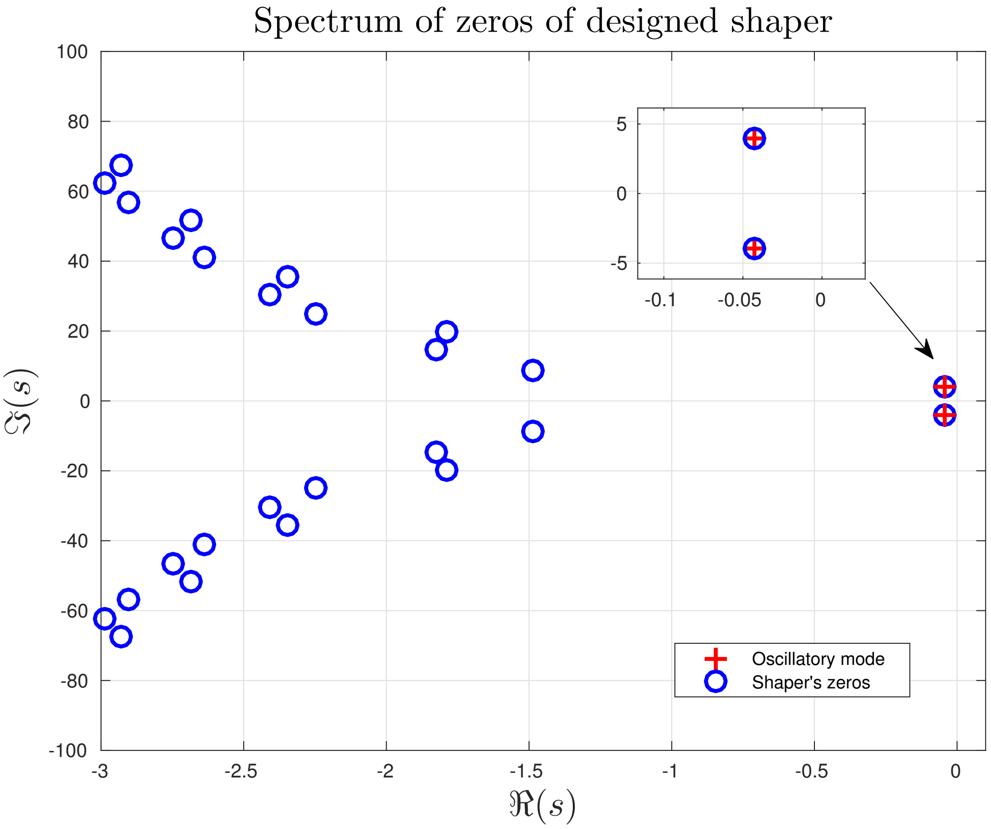


Figure 8.1. Spectrum of designed D_eZV shaper with parameters 8.1 (blue +), oscillatory mode to be compensated (red o).

From Figure 8.1 it can be seen that most dominant couple of zeros of designed shaper compensates identified oscillatory mode of the pendulum (8.1).

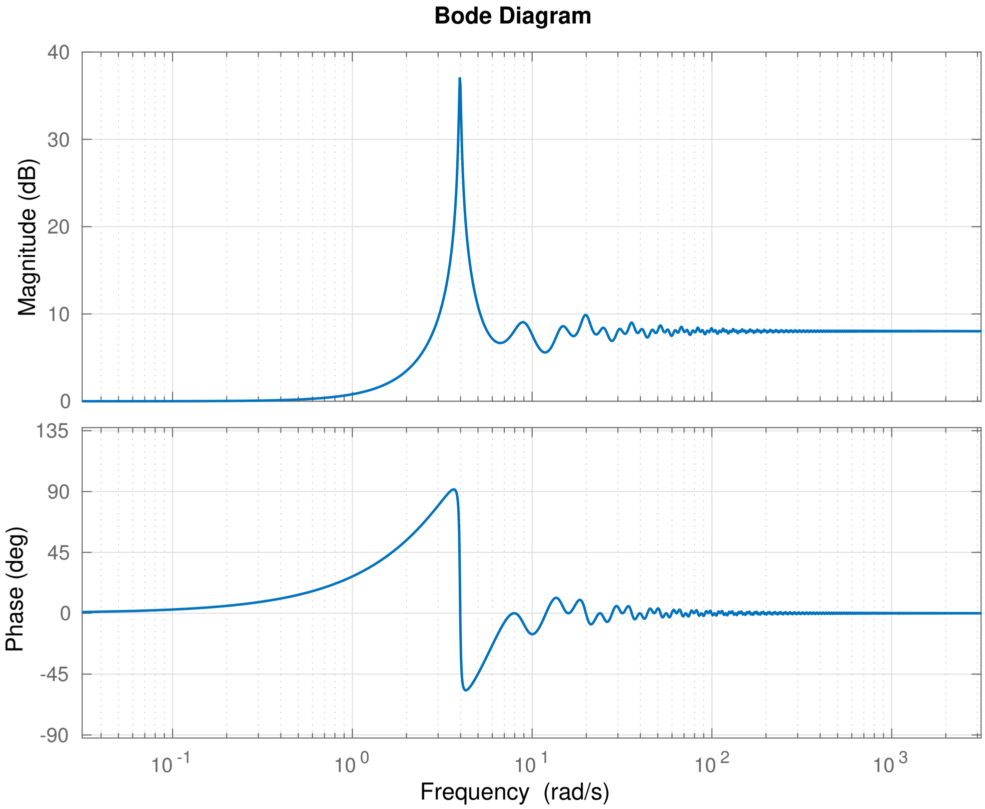


Figure 8.2. Frequency response of the designed signal shaper.

Frequency response of the designed inverse shaper (8.9) is shown in Figure 8.2. The main frequency properties of the designed inverse shaper were obtained by program MATLAB and are defined as follows

$$\lim_{\omega\to 0}\left|\frac{1}{S_{D_eZV}(j\omega)}\right| = 20\cdot\log(1) = 0\ (dB), \tag{8.10}$$

$$\lim_{\omega\to\infty}\left|\frac{1}{S_{D_eZV}(j\omega)}\right| = 20\cdot\log\left(\frac{1}{A}\right) = 20\cdot\log\left(\frac{1}{0.3899}\right) = 8.2\ (dB), \tag{8.11}$$

$$\lim_{\omega\to 0} arg\left(\frac{1}{S_{D_eZV}(j\omega)}\right) = 0\ (deg), \tag{8.12}$$

$$\lim_{\omega\to\infty} arg\left(\frac{1}{S_{D_eZV}(j\omega)}\right) = 0\ (deg), \tag{8.13}$$

the resonant peak is given for the target mode frequency $\omega = \omega_0 = 4.138(rad.s^{-1})$

$$\left|\frac{1}{S_{D_eZV}(j\omega_0)}\right| = 20\cdot\log(72.67) = 37.23\ (dB). \tag{8.14}$$

The signal shaper for vibration suppression of the mathematical mode is designed analogously. The target oscillation mode of the mathematical model $r_{m_{1,2}}$ was derived in Chapter 6.5.1, $r_{m_{1,2}} = -0.065 \pm j3.63$. The Table 8.1 shows designed D_eZV shapers for real system and mathematical model.

System	A (1)	τ (s)	T (s)
Real	0.397	0.393	0.7917
Mathematical	0.3911	0.4324	0.8664

Table 8.1. Designed shapers's parameters.

8.2 Controller Design

The controller was first designed for system with communication delays without included inverse shaper in the feedback path. Communication delays in the feedback and input were assumed to be constant, lumped and equal to the maximal identified delay of Ethernet connection, that is $\tau_{max_{link}} = 15.5$ (ms). Transfer functions of the communication delay in the actuator node $D_a(s)$ and of the delay in the sensor node $D_s(s)$ are defined as follows

$$D_a(s) = e^{-s(\tau_{max_{link}} + \tau_{input})} = e^{-s\frac{15.5+5}{1000}}, \tag{8.15}$$

$$D_s(s) = e^{-s\tau_{max_{link}}} = e^{-s\frac{15.5}{1000}}. \tag{8.16}$$

Proportional derivative (PD) controller with a filter was chosen to control astatic system. The controlled variable is the position of cart x_a. The transfer function $T_{ux_a}(s)$ of the input signal u and the position of cart a was obtained from state-space model via MATLAB command `tf(state_space_model)`

$$T_{ux_a}(s) = \frac{2.67s^4 + 3.37s^3 + 309.3s^2 + 53.19s + 3604}{s^6 + 9.876s^5 + 293.4s^4 + 1127s^3 + 5465s^2 + 13140s}, \tag{8.17}$$

and the structure of chosen controller $C(s)$ is defined as follows:

$$C(s) = \frac{r_d s + r_0}{Ns + 1}, \tag{8.18}$$

where r_d is the derivative constant, r_0 the proportional constant and N the filter constant. The controller's parameters were chosen so that the closed-loop system is stable with sufficient delay margin and fast time response. The proposed parameters are:

r_0 (1)	r_d (s)	N (s)
16	1	0.02

Table 8.2. Designed controller's parameters.

The bode diagram of the open loop transfer functions are shown in Figure 8.3.

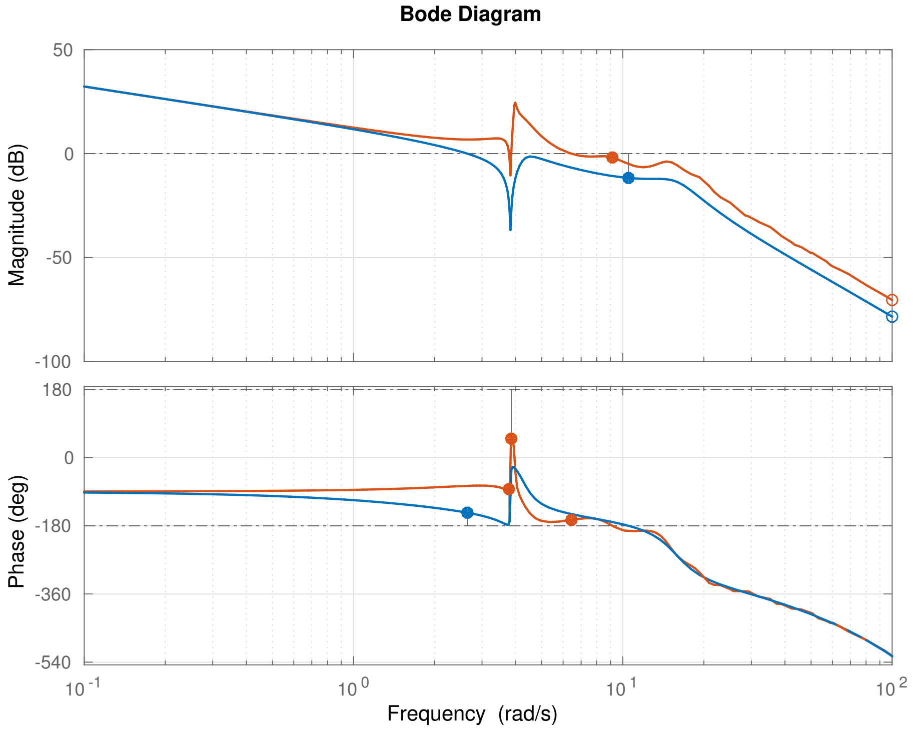


Figure 8.3. Open loop Bode plots: without inverse shaper $(C(s)D_a(s)G(s)D_s(s))$ (blue), with inverse D_eZV shaper $(S^{-1}(s)C(s)D_a(s)G(s)D_s(s))$ (red).

From the inspection of bode diagram 8.3 can be seen that the open loop gain with an inverse shaper is raised at the target frequency, due to the resonance peak of the inverse shaper dynamics, and also for higher frequencies by $20\log\frac{1}{A}$. Gain G_m and phase P_m margins were obtained by the program MATLAB. Delay margin D_m can be calculated as follows:

$$D_m = \frac{P_m}{\omega_c}, \tag{8.19}$$

where ω_c is corresponding phase or gain crossover frequency. Stability margins of the closed loop without inverse shaper are shown in Table 8.3 and the smallest margins of the closed loop with inverse shaper are shown in Table 8.4.

Margin	Value	ω_c $(rad.s^{-1})$
G_m	11.7 (dB)	10.5
P_m	34.5 (deg)	2.65
D_m	0.227 (sec)	2.65

Table 8.3. Stability margins of closed loop system: $C(s)D_a(s)G(s)D_s(s)$.

Margin	Value	ω_c $(rad.s^{-1})$
G_m	1.86 (dB)	9.15
P_m	15.5 (deg)	6.45
D_m	0.0419 (sec)	6.45

Table 8.4. Stability margins of closed loop system: $S^{-1}(s)C(s)D_a(s)G(s)D_s(s)$.

Condition of stability for the closed loop is also confirmed by the closed loop spectra for feedback, see Figure 8.4, since every pole lies in the stable region, i.e., left of the imaginary axis, system is stable. Closed loop spectra was determined via QPMR algorithm that was briefly discussed in Chapter 3.3.

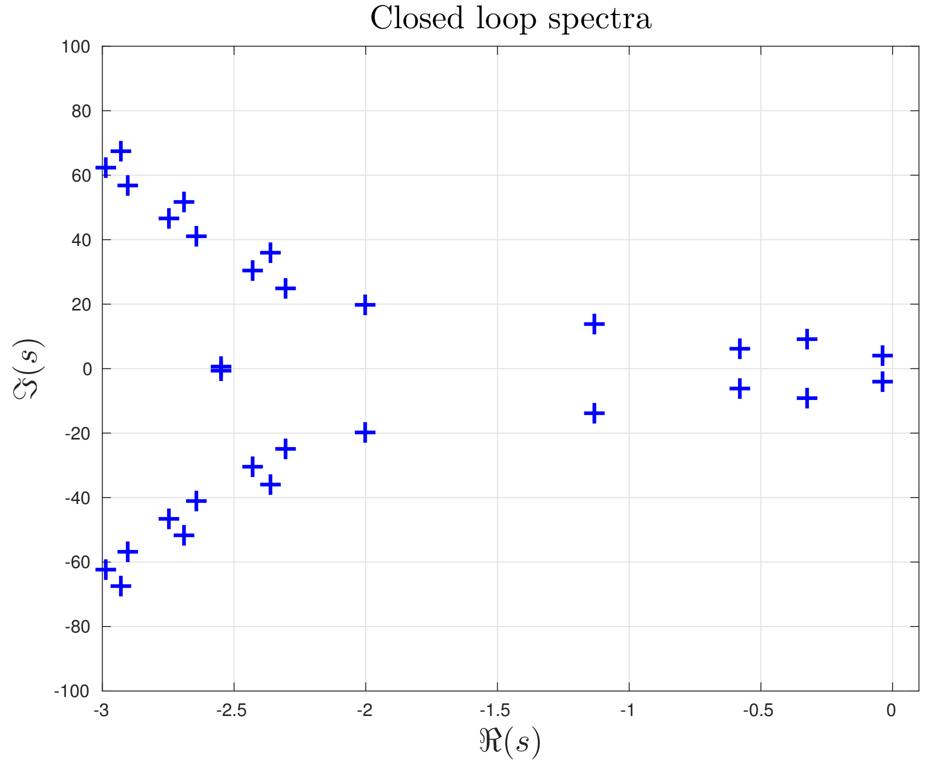


Figure 8.4. Closed loop spectra for feedback with inverse shaper and communication delays, $(S(s) + C(s)D_a(s)G(s)D_s(s))$.

Both closed loop with and without inverse shaper are stable. However, the closed loop with inverse shaper has lower robustness since inclusion of the inverse shaper into the feedback path raises the open loop gain and decreases all stability margins. Overall robustness can be increased by decreasing controller's proportional gain r_0, however, the trade-off of this solution is slower control process.

Chapter 9
Experimental Results

In this chapter results of the performed experiments with the laboratory set-up are presented. Designed PD controller, inverse shaper and NCS architecture in previous chapters were applied. Different experimental configurations were tested, in order to determine effect of NCS architecture on the overall control performance. Tested configurations are:

- **Configuration A** : System is controlled locally by the PD controller without the inverse shaper.
- **Configuration B** : System is controlled locally by the PD controller with the inverse shaper.
- **Configuration C** : System is controlled remotely by the PD controller without the inverse shaper.
- **Configuration D** : System is controlled remotely by the PD controller with the inverse shaper.

All MATLAB programs and Simulink models that were used for performed experiments are provided on the enclosed CD.

9.1 Measured Results

The main used parameters to quantify the control performance of cart are:

- **Rise time** : The time required to rise from 0 to 100 % of the reference value.
- **Peak overshoot** : The difference between the time response peak and the steady state.
- **Settling time** : The time required for the response of the cart to reach and stay within a tolerance band of its final value.

The control performance of the flexible subsystem is quantified by the following parameters:

- **Settling time** : The time required for the response of the pendulum to reach and stay within a tolerance band of its final value.
- **Maximal deflection** : The maximal deflection of the pendulum.

The measured results are presented in Tables 9.1, 9.2 and in Figures 9.1, 9.2 and 9.3 where the top graph shows the position of the cart a; the middle, the pendulum's displacement; the bottom, applied control signal. In all graphs is setpoint indicated as a dash-dotted red line and tolerance zone as a dashed black line, (± 5 (%) of the setpoint value for cart a and ± 5 (°) for pendulum).

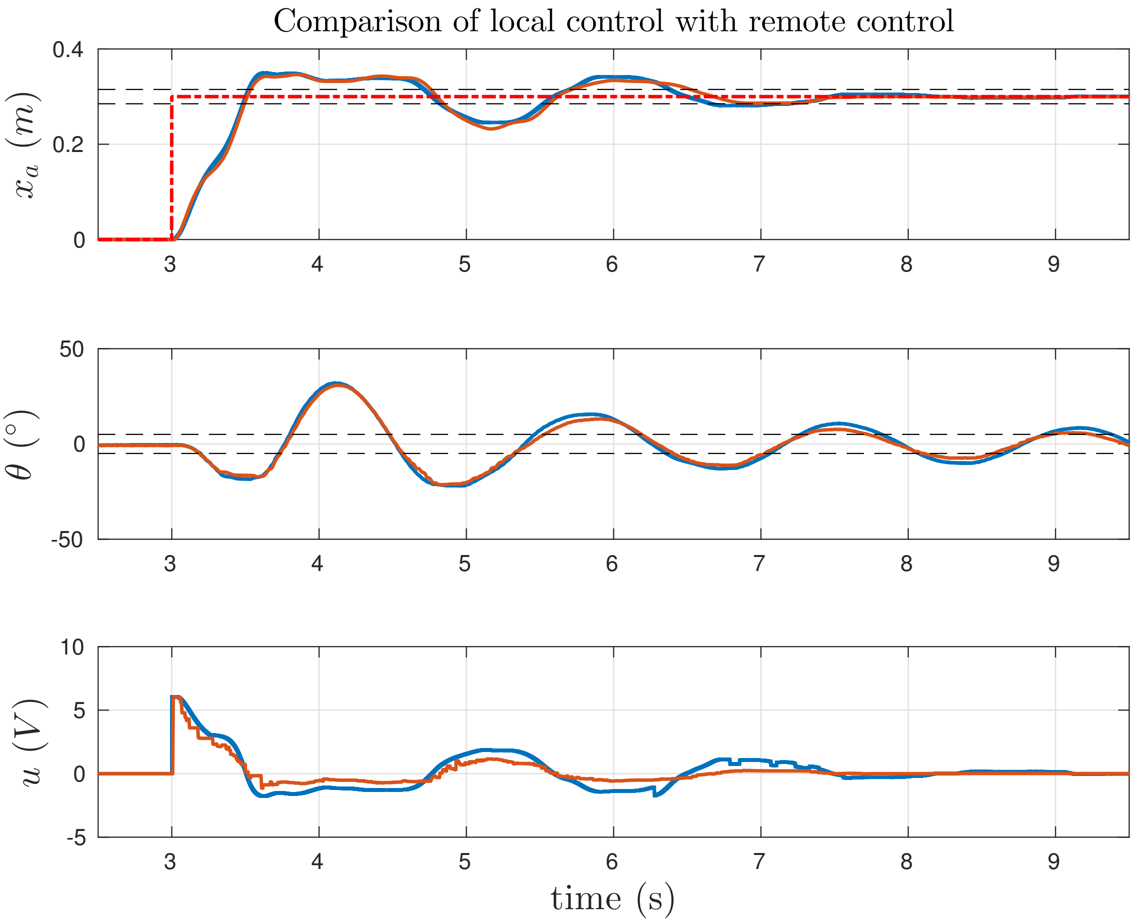


Figure 9.1. Comparison of measured setpoint responses without applied inverse signal shaper, local control - configuration A (blue), remote control - configuration C (orange).

In Figure 9.1 are compared results of the remote and the local control of the system without applied inverse shaper. The control loop without the inverse shaper has a higher robustness. Therefore, communication delays caused by the Internet network had almost no effect on system's dynamics. It can be seen that pendulum's oscillations are not suppressed by using a control architecture with only PD controller.

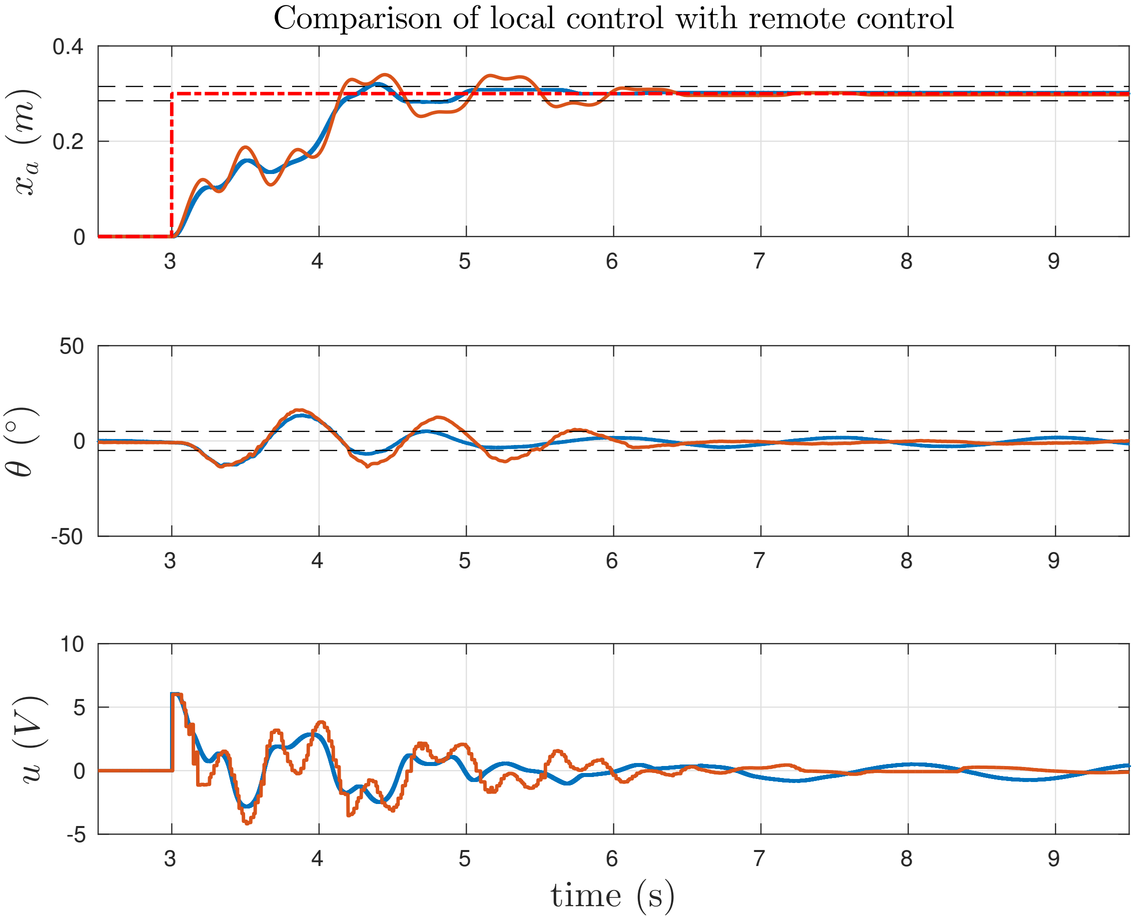


Figure 9.2. Comparison of measured setpoint responses with applied inverse signal shaper, local control - configuration B (blue), remote control - configuration D (orange).

In Figure 9.2 are compared results of the remote and the local control of the system with applied inverse shaper. In this case it is clear that communication delays negatively influenced overall system's dynamics. The remote control configuration has a higher value of peak overshoot and settling time for both cart b position and pendulum displacement. This outcome is not surprising since the inclusion of inverse shaper in the feedback path decreases all stability margins (gain, phase, and delay) of the closed loop as it was shown in Chapter 8.2. However, inverse signal shaper efficiently suppress pendulum's oscillations in both cases. The pendulum's maximal deflection has much smaller value and the settle time is reached significantly sooner than in configurations without shaper. Nevertheless, pendulum's oscillations were not suppressed perfectly, most probably because of the mismatch between identified and real oscillation mode or actuator's limitations.

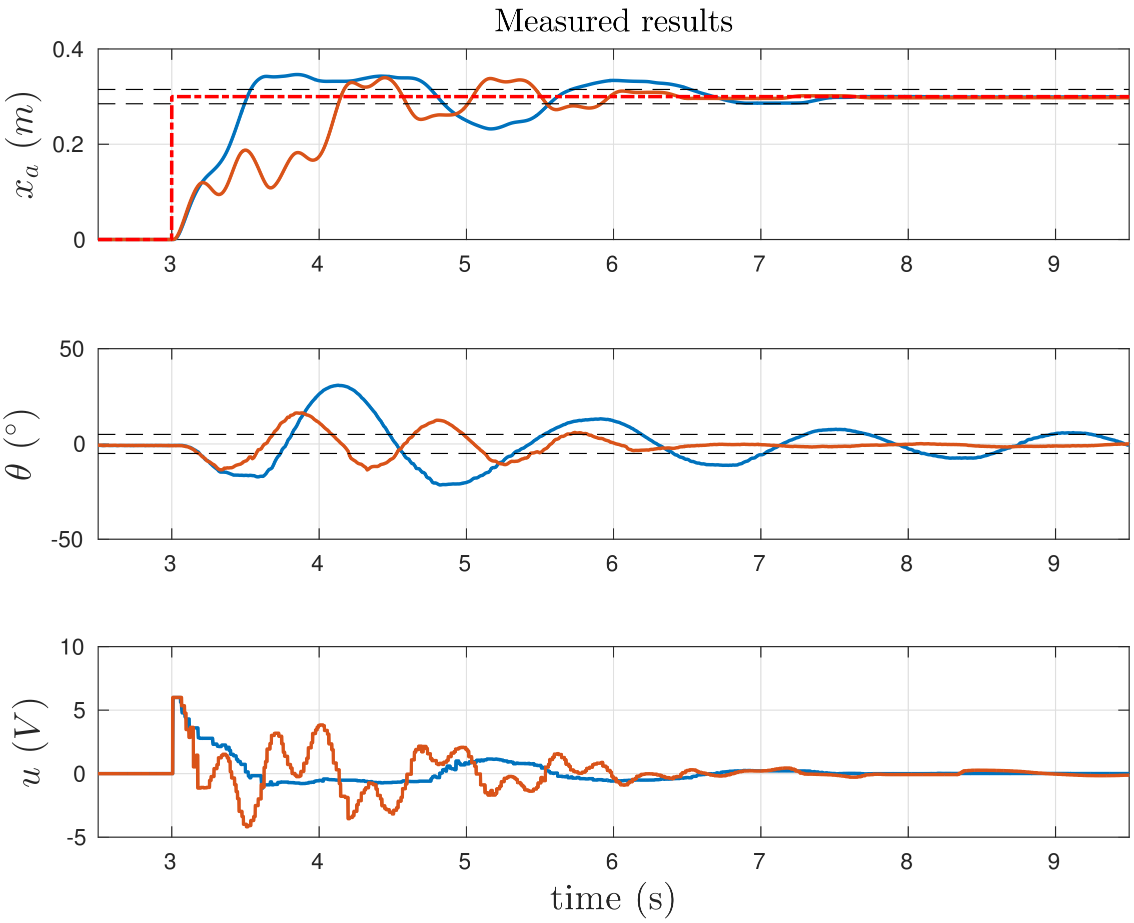


Figure 9.3. Comparison of measured setpoint responses, remote control without shaper - configuration C (blue), remote control with shaper - configuration D (orange).

Figure 9.3 compares results of remote control with and without applied inverse shaper. It can be seen that the control signal is shaped by the inverse signal shaper and the pendulum's oscillation is effectively damped in case of control architecture with inverse shaper.

Criterion \ Configuration	A	B	C	D
Rise time (s)	3.5	4.26	3.51	4.15
Settling time (s)	7.19	4.89	7.24	5.9
Peak time (s)	3.62	3.2	3.86	4.45
Peak overshoot (m)	0.0495	0.0203	0.0463	0.0398

Table 9.1. Results of controllers performance criteria for the controlled position of the cart.

Criterion \ Configuration	A	B	C	D
Settling time (s)	9.41	4.73	10.12	5.82
Maximal deflection (°)	31.91	13.38	30.64	16.28

Table 9.2. Results of controllers performance criteria for the passively controlled pendulum's deflection.

In Tables 9.1 and 9.2 are summarized measured results. Control architecture with the inverse shaper is superior in every aspect expect the rise time. Especially, the oscillation of the pendulum is significantly suppressed for both configurations with the inverse shaper. From the measured results it can be concluded that effectiveness of the inverse shaper part was experimentally verified. Communication delays influence more configurations with the inverse shaper, that is caused by the fact that the control loop with inverse shaper is less robust and has smaller time-delay margin.

9.2 Simulation results

In this section, simulation results of the derived mathematical model in Chapter 6 are presented. Signal shaper was designed to compensate model's oscillation mode since real mode and mathematical mode differs from each other. Shaper's parameters are presented in Table 6. Simulation result is presented in Figure 9.4.

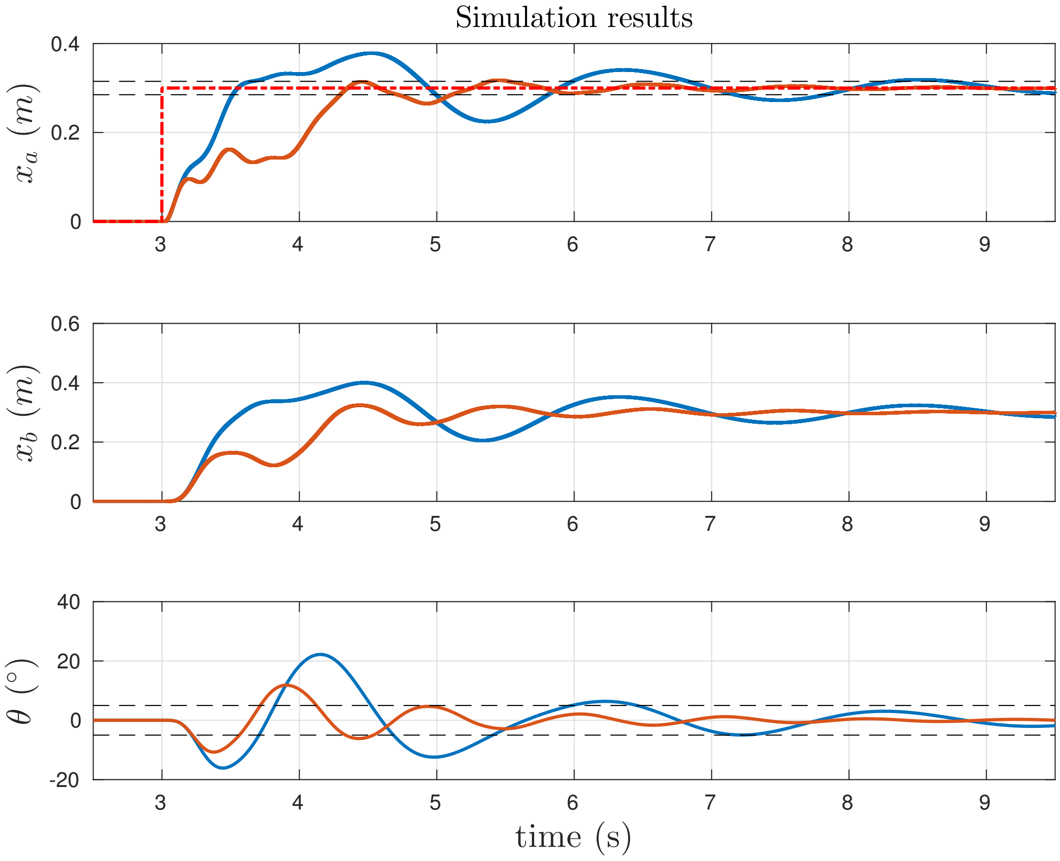


Figure 9.4. Comparison of simulated setpoint responses, control loop without signal shaper (blue), control loop with signal shaper (orange).

Simulation results again confirm effectiveness of inverse signal shaper. Good agreement between simulated and measured results confirms that derived mathematical model is adequate for the control design purposes. Pendulum's oscillations are not suppressed perfectly due to the fact that oscillation of carts a and b are not very well suppressed by used PD controller. Better results could be probably achieved by using a state space or high-order controller, that can provide a more control design options unlike PD controller that has only two design parameters (proportional and derivative gain).

Chapter 10
Conclusion

All goals of this thesis were successfully accomplished. Signal shaping method for vibration suppression of the flexible system with emphasis on recently developed novel control architecture with inverse shaper was studied in Chapter 4. The basic concepts of Internet remote control and time-delays systems, needed for thesis elaboration were described in Chapters 5 and 3, respectively.

In Chapter 6, state-space model of the laboratory setup was derived and identified. Next, in order to reduce negative influence of the friction to system responses, the friction compensation method was applied. This method has proven to be an effective solution for a simple compensation of friction forces. In Chapter 7 networked control system (NCS), used for a remote control of the laboratory setup, was designed. Moreover, a graphical user interface for a more convenient control of NCS was developed in program MATLAB.

The control of the laboratory setup was designed in Chapter 8. Classical proportional-derivate controller was used for a position control of the cart a and pendulum's oscillation was suppressed by an inverse shaper D_eZV in the feedback path. The controller was designed so that the closed feedback loop remains stable even under influence of time-delays caused by the inverse shaper and the signal transmission. However, stability margins of the closed loop with inverse shaper are of low value (gain margin $G_m = 1.86$ (dB), delay margin $D_m = 0.0419$ (s)), therefore, a more robust control design would be necessary for a real industrial application. Nevertheless, designed control loop was suitable for laboratory experiments.

Finally, designed NCS control architecture with inverse shaper was experimentally tested and results are presented in Chapter 9. The measured results confirmed effectiveness of the control architecture with inverse shaper for vibration suppression of flexible parts. However, control loop with inverse shaper in the feedback path was more prone to the destabilizing influence of communication delays. This outcome corresponds to the fact that designed control loop with inverse shaper was less robust then control loop without inverse shaper. Nevertheless, networked control worked flawlessly and designed NCS can be extended for a more sophisticated control applications in future works. Control performance could be improved by using a more sophisticated controller with more parameters, e.g. stace-space controller, high-order controller, that give us a greater freedom during control design then simple proportional-derivative controller. Therefore, future research could concentrate on the design of the control loop with more sophisticated controller and inverse shaper in the feedback path.

References

[1] William Singhose and Warren Seering. Command Generation for Dynamic Systems. William Singhose, 2011.

[2] U. Boettcher, D. Fetzer, H. Li, R. A. Callafon, F. E. Talke. Reference Signal Shaping for Closed-Loop Systems With Application to Seeking in Hard Disk Drives. IEEE Transaction on control systems technology, vol. 20, no. 2, pp. 335-345, March 2012.

[3] O. J. M. Smith, Posicast control of damped oscillatory systems. Proceedings of the IRE, Vol. 45, pp. 1249-1255, September 1957.

[4] N. C. Singer, W. P. Seering. Preshaping command input to reduce system vibration. The Journal of Dynamic System, Measurement, and Control, vol.112, pp. 76-82, 1990.

[5] W.E. Singhose, W. P. Seering, N. C. Singer. Residual vibration reduction using vector diagrams to generate shaped inputs. Journal of Mechanical Design, vol. 116, pp. 654-659, June 1994.

[6] T. Vyhlídal, V. Kučera, M. Hromčík. Signal shaper with a distributed delay: Spectral analysis and design. Elsevier, Automatica 49 (2013), pp. 3484-3489, 2013.

[7] T. Vyhlídal, V. Kučera, M. Hromčík. Zero vibration shapers with distributed delays of various types. 52nd IEEE Conference on Decision and Control, Florence, pp. 940-945, 2013.

[8] T. Vyhlídal, M. Hromčík, V. Kučera. Inverse signal shapers in effective feedback architecture. 2013 European Control Conference, Zürich, pp. 4418-4423, 2013.

[9] T. Vyhlídal, M. Hromčík, V. Kučera, M. Anderle. On feedback architectures with zero vibration signal shapers. IEEE, Transactions on Automatic Control Volume: 61, Issue: 8, pp. 2049-2064, 2015.

[10] Katsuhiko Ogata. Modern Control Engineering. Pearson, 2009.

[11] Paul McGahan. The effect of feedback latencies on the stability and design of state derivative feedback for vibration suppression problems. Czech Technical University in Prague, 2007.

[12] H. Ollson, K.J. Åström, C.Canudas de Wit, M. Gäfvert, P. Lischinsky. Friction Models and Friction Compensation. European Journal of Control 4(3), pp. 1-32, 1998.

[13] M. A. Krzeminski. Modeling Friction through the use of a Genetic Algorithm. Master thesis, University of Waterloo, 2003.

[14] P. Zítek, A.Víteček. Návrh řízení podsystémů se zpožděními a nelinearitami. Vydavatelství ČVUT, 1999.

[15] Emilia Fridman. Introduction to Time-Delay Systems. Birkhäuser, 2014.

[16] V. Kolmanovskii, A. Myshkis. Applied Theory of Functional Differential Equations. Springer, 1992.

[17] P. Zítek, R.Petrová. Matematické a simulační modely 2. Vydavatelství ČVUT, 2004.

[18] T. Vyhlídal, M. Hromčík. Parameterization of input shapers with delays of various distribution. Elsevier, Automatica 59 (2015), pp. 256-263, 2015.

[19] M.Malek-Zavarei, M.Jamshidi. Time-Delay Systems: Analysis, Optimization and Applications. North-Holland, USA, 1987.

[20] T. Vyhlídal, P. Zítek, G. Simeunovič. Úvod do problematiky systémů se zpožděnímú. ČVUT, Matematické a simulační modely, 2016.

[21] P. Zítek, T.Vyhlídal. Quasipolynomial mapping based rootfinder for analysis of time delay systems. Elsevier, Time Delay Systems 2003 – A Proceedings volume from the 4th IFAC workshop, pp. 227-232, 2003.

[22] N. C. Singer, W.P. Seering, Preshaping Command Inputs to Reduce System Vibration. J.Dyn.Syst., Meas. Control, vol. 112, pp. 76-82, 1990.

[23] M. Hromčík, T. Vyhlídal. Inverse Feedback Shapers for Coupled Multibody Systems. IEEE Tran. on Auto. Control, 2017, DOI 10.1109/TAC.2017.2688179

[24] T. Vince, I. Kováčová. Distance Control of Mechatronic Systems via Internet. Acta Electrotechnica et Informatica No.3, Vol.7, pp. 1-6. 2007.

[25] S. Yang, L. Tan, G. Liu. Architecture Design for Internet-based Control Systems. International Journal of Automation and Computing 1, pp. 1-9, 2004.

[26] J.F. Kurose, K.W. Ross. Computer Networking: A Top-Down Approach, sixth edition. Pearson, USA, 2012.

[27] S. Munir, W.J.Book. Internet-based teleoperation using wave variables with prediction. IEEE/ASME Trans. Mechatronics, vol. 7, no. 2, pp. 124-133, June 2002.

[28] H. Sawashima, Y. Hori, H. Sunahara. Characteristics of UDP Packet Loss: Effect of TCP Traffic. Japan, Nara Institute of Science and Technology, 1998.

[29] E. Delgado, M.D. Cacho, D. Bustelo, A. Barreiro. Generic Approach to Stability Under Time-Varying Delay in Teleoperation: Application to the Position-Error Control of a Gantry Crane. IEEE/ASME, Transactions on Mechatronics, vol. 18, no.5, pp. 1581-1591, 2013.

[30] D. Pilbauer, J.Bušek, V.Kučera, T.Vyhlídal. Labortory Set-up Design for Testing Vibration Suppression Algorithms with Time Delays. Technical University of Ostrava, Transactions of the VŠB, vol. LX, no. 1, pp. 87-95, 2014.

[31] E. Neumann. Cart and Pendulum, myphysicslab.com, 2017-04-07.
`https://www.myphysicslab.com/pendulum/cart-pendulum/cart-pendulum-en.html`.

[32] R. Srikant. The Mathematics of Internet Congestion Control. Birkhäuser, 2004.

[33] M.D. Cacho, A. Fernández, M. García, A. Barreiro. Internet emulation system for UDP-based teleoperation. Proc. Mediterranean Conf. Control Autom., Ajaccio, France, pp. 1417-1422, 2008.

Appendix A
Source Codes

A.1 Matlab source code of friction compensation

```
function u_comp = fcn(u,v)
%#codegen
% arg1 - control signal
% arg2 - estimated velocity
% ret1 - compensation control signal

S_a = 4.8; % (N.V-1) - static gain of actuator
dz = 0.05; % (m.s-1) - half interval of dead zone
F_s1 = 4.8; % (N) - stiction friction force for positive velocity
F_s2 = 5.52; % (N) - stiction friction force for negative velocity
F_col1 = 0.1; % (N) - coloumb friction force for positive velocity
F_col2 = 0.1; % (N) - coloumb friction force for negative velocity
v_abs = abs(v); % (m.s-1) - absolute value of the cart velocity
u_abs = abs(u); % (V) - absolute value of the control signal

% friction compensation
if(v > 0) % positive velocity
    if(v_abs <= dz && u_abs < F_s1 / S_a)
        u_f = u;
    elseif(v_abs <= dz && u_abs >= Fs1 / S_a)
        u_f = (F_s1 / S_a) * sign(u);
    else
        u_f = Fc1 / S_a;
    end
else % negative velocity
    if(v_abs <= dz && u_abs < Fs_2 / S_a)
        u_f = u;
    elseif(v_abs <= dz && u_abs >= Fs_2 / S_a)
        u_f = (Fs_2 / S_a) * sign(u);
    else
        u_f = Fc2 * -1;
    end
end
% actuator compensation
nonlin_end_point = 4;
if(u_abs < nonlin_end_point)
    u_act = (u_abs - (1/nonlin_end_point)*u.^2) * sign(u);
else
    u_act = 0;
end
```

```
% sum of compensation signals
u_comp = u_f + u_act;

end
```

A.2 Matlab source code of delay estimation function

```
function [delay,mem_new] = fcn(packet_in_time,...
                               curr_time,...
                               mem_old)
% arg1 - timestamp of arriving packet
% arg2 - timestamp of current packet
% arg3 - timestamp of previous packet
% ret1 - value of packet delay
% ret2 - timestamp of arriving packet
%#codegen

mem_new = 0;

if(packet_in_time > 0)
    mem_new = packet_in_time;
    if(mem_old ~= packet_in_time)
        delay = curr_time - packet_in_time;
    else
        delay = 0; % packet is lost
    end
else
    delay = -5; % no connection
end
```

A.3 Simulink model used at the remote computer

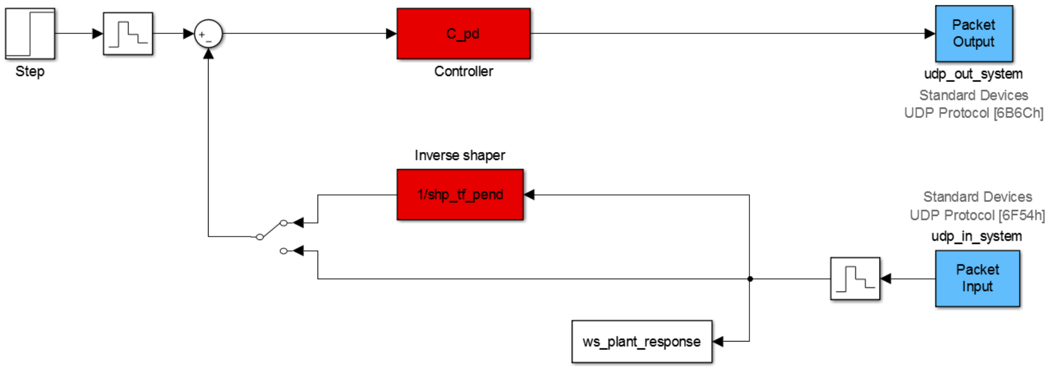


Figure A.1. Simulink model used at the remote computer

A.4 Simulink model used at the local computer

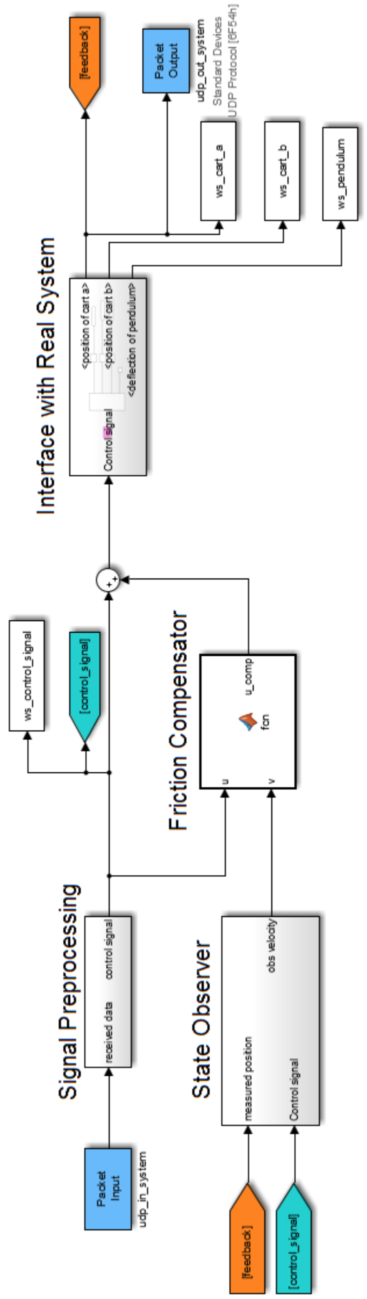


Figure A.2. Simulink model used at the local computer

Appendix B
Abbreviations and Symbols

B.1 Abbreviations

CDF	Cumulative Distribution Function
CIM	Computer Integrated Manufacturing
COM	Serial communication port
DAQ	Data Acquisition Card
DeZV	Distributed Zero-Vibration Shaper with lumped and equally distributed delay
DSPST	DSP System Toolbox
DZV	Distributed Zero-Vibration Shaper
EI	Extra Insensitive Shaper
EQD	equally distributed delay
FDEs	Functional Differential Equations
FIR	Finite Impulse Response
FO	Friction Observer
GUI	Graphical User Interface
IA	Information Architecture
ICT	Instrument Control Toolbox
IP	Internet Protocol
LAN	Local Area Network
LD	Lumped Delay
LTI	Linear Time Invariant System
NCSs	Networked Control Systems
N-E	Newton-Euler equation
PLC	Programmable Logic Controller
PC	Personal Computer
PC_p	Personal Computer of Plant
PC_c	Personal Computer of Controller
PD	Proportional-Derivative controller
QPMR	Quasipolynomial Mapping Based Rootfinder
RTWT	Real Time Windows Target
RTT	Round Trip Time
SDRT	Simulink Desktop Real-Time Library
SISO	Single Input Single Output System
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TDSs	Time Delay Systems
TI	Traffic Intensity
UDP	User Datagram Protocol

ZV	Zero Vibration Shaper
ZVD	Zero Vibration Derivative Shaper

B.2 Symbols

$\mathbf{A}$	state matrix
$\mathbf{A_{red}}$	reduced state matrix
$\tilde{\mathbf{A}}$	state matrix matrix of closed loop
$A(s)$	actuator's transfer function
a	packet average arrival rate $(m.s^{-1})$
A	shaper gain
$\mathbf{B}$	input matrix
b_s	damping of spring $(N.s.m^{-1})$
$\mathbf{C}$	output matrix
$C(s)$	controller's transfer function
$\mathbf{D}$	direct transmission matrix
$D_a(s)$	transfer function of communication delay in actuator node
D_m	delay margin (s)
$D_s(s)$	transfer function of communication delay in sensor node
$D_u(s)$	transfer function of delay of sending link
$D_y(s)$	transfer function of delay of receiving link
d_{bp}	diameter of the belt pulley (m)
$\mathbf{e}(t)$	error vector
$F(s)$	flexible system's transfer function
$\hat{F}$	estimated friction force (N)
$\hat{F}_{comp}$	estimated compensation force (N)
F_{col}	coulomb friction force (N)
F_b	damping force (N)
F_e	external force (N)
F_{f_a}	friction force of cart a (N)
F_{f_b}	friction force of cart b (N)
F_{f_p}	friction force of pendulum p (N)
F_k	spring force (N)
F_{st}	stiction force (N)
G	gravitational force (N)
G_m	gain margin (dB)
$G(s)$	system's transfer function
$\mathbf{G}(s)$	matrix of transfer functions
$H(.)$	heaviside step function of .
$h(\tau)$	delay distribution function
$\mathbf{I}$	identity matrix
$\Im(.)$	imaginary part of .
$\mathbf{K}$	feedback gain matrix
$\mathbf{K_e}$	observer gain matrix
k_s	stiffens of spring $(N.m^{-1})$
L	size of he packet $(bits)$
l_p	length of pendulum (m)
$M(s)$	characteristic equation of system
M_n	nominal torque $(N.m)$

m_a	mass of cart a (kg)
m_b	mass of cart b (kg)
m_p	mass of pendulum p (kg)
N	filter constant (s)
n	number of state variables
$\mathbf{O}$	observability matrix
P_m	phase margin (deg)
$\mathbf{R}$	controllability matrix
R	transmission rate $(bits.s^{-1})$
$R(s)$	coupling transfer function
$\Re(.)$	real part of .
r_d	derivative gain constant (s)
r_0	proportional gain constant
$r_{1,2}$	complex conjugate couple of poles
S_a	actuator's static gain $(N.V^{-1})$
$S(s)$	transfer function of shaper
$S_{ZV}(s)$	transfer function of zero vibration shaper
$S_{ZVD}(s)$	transfer function of zero vibration derivative shaper
$S_{DZV}(s)$	transfer function of distributed zero vibration shaper
$S_{D_eZV}(s)$	transfer function of distributed zero vibration shaper with lumped and equally distributed delay
s	complex variable, operator of Laplace domain
t	time (s)
T	length of distributed delay (s)
t_0	initial time (s)
$\mathbf{U}(s)$	input column vector
$\mathbf{u}(t)$	input vector
$u(t)$	input
u	control signal (V)
u_{comp}	compensation signal
$V(\omega,\zeta)$	residual vibration
v_{prop}	propagation speed $(m.s^{-1})$
x_a	position of cart a (m)
$\dot{x}_a$	velocity of cart a $(m.s^{-1})$
$\ddot{x}_a$	acceleration of cart a $(m.s^{-2})$
x_b	position of cart b (m)
$\dot{x}_b$	velocity of cart b $(m.s^{-1})$
$\ddot{x}_b$	acceleration of cart b $(m.s^{-2})$
$\mathbf{x}(t)$	state vector
$\hat{\mathbf{x}}(t)$	observed state vector
$\mathbf{Y}(s)$	output column vector
$\mathbf{y}(t)$	output vector
ϵ	dead zone interval
ζ	damping ratio (1)
θ	Pendulum's displacement (rad)
$\dot{\theta}$	Pendulum's angular velocity $(rad.s^{-1})$
$\dot{\theta}$	Pendulum's angular acceleration $(rad.s^{-2})$
μ_a	viscous friction of cart a $(N.s.m^{-1})$

μ_b	viscous friction of cart b $(N.s.m^{-1})$
μ_p	viscous friction of pendulum $(N.s.m^{-1})$
$\mu_{\tau_{link}}$	mean of the link delay (ms)
$\sigma_{\tau_{link}}$	standard deviation of link delay (ms)
τ	lumped delay (s)
τ_{nodal}	nodal delay (s)
τ_{proc}	processing delay (s)
τ_{prop}	propagation delay (s)
τ_{queue}	queuing delay (s)
$\tau_{max_{link}}$	maximal link delay (ms)
τ_{trans}	transmission delay (s)
τ_k	delay of shaper impulses (s)
τ_u	communication delay of sending link (s)
τ_y	communication delay of receiving link (s)
ϕ	argument of complex number (rad)
ω	frequency $(rad.s^{-1})$
ω_c	cross-over frequency $(rad.s^{-1})$
ω_0	natural frequency $(rad.s^{-1})$
Ω	damped natural frequency $(rad.s^{-1})$