Master's Thesis
석사 학위논문

Study on DOB based Control Systems with Saturating Actuators and Measurement Noise

Gyujin Na(나 규 진 羅 圭 鎭)

Department of Information and Communication Engineering

정보통신융합공학전공

DGIST

2017

Master's Thesis
석사 학위논문

Study on DOB based Control Systems with Saturating Actuators and Measurement Noise

Gyujin Na(나 규 진 羅 圭 鎭)

Department of Information and Communication Engineering

정보통신융합공학전공

DGIST

2017

Study on DOB based Control Systems with Saturating Actuators and Measurement Noise

Advisor: Professor Yongsoon Eun
Co-advisor: Professor Hyungbo Shim

by

Gyujin Na

Department of Information and Communication Engineering
DGIST

A thesis submitted to the faculty of DGIST in partial fulfillment of the requirements for the degree of Master of Science in the Department of Information and Communication Engineering. The study was conducted in accordance with Code of Research Ethics [1]

11. 16. 2016

Approved by

Professor Yongsoon Eun (Signature)

Professor Hyungbo Shim (Signature)

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of DGIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

Study on DOB based Control Systems with Saturating Actuators and Measurement Noise

Gyujin Na

Accepted in partial fulfillment of the requirements
for the degree of Master of Science.

11. 16. 2016

Head of Committee 은 용 순 (인)
Prof. Yongsoon Eun

Committee Member 심 형 보 (인)
Prof. Hyungbo Shim

Committee Member 손 상 혁 (인)
Prof. Sang Hyuck Son

MS/IC
201522007

나규진. Gyujin Na. Study on DOB based Control Systems with Saturating Actuators and Measurement Noise. Department of Information and Communication Engineering. 2017. 64p. Advisor Prof. Yongsoon Eun. Co-Advisor Prof. Hyungbo Shim

ABSTRACT

In chapter 1, we consider effect of measurement noise of DOB based control system with saturating actuators. Augmenting feedback control systems with disturbance observer (DOB) is a widely used technique in system design to compensate for the effect of exogenous disturbances as well as plant model uncertainties. In practice, DOB implementation has to take actuator saturation into account in order to avoid poor transient response or instability occurring due to saturating control input. In such systems, we have observed that a tracking loss may occur due to zero mean measurement noise. This phenomenon has never been reported in DOB literature. This paper reports the phenomenon, analyzes the conditions under which the tracking loss occurs, and also presents design guideline to avoid it based on the analysis. Experimental verification is also provided using a BLDC motor drive testbed.

In chapter 2, we consider heavy-duty vehicle platooning. Heavy-duty vehicle platooning has received much attention as method to reduce fuel consumption by keeping the distance between vehicles short enough to decrease aerodynamic drag. Major disturbances in the platooning are the slope in the road and uncertain mass of the vehicle. Existing method to reduce the effect of slope is to combine the vehicle position obtained from GPS and the slope database to calculate the amount of feed-forward type compensation. However, slope database is costly to acquire, and GPS signal is unreliable in some locations. We present vehicle controller based on disturbance observer (DOB) to compensate the effect of slope and mass without relying on the GPS and the slope database. Actual measurement of the road slope taken in highways of Sweden is used and simulation is conducted.

Key words : Disturbance Observer, Noise Induced Tracking Error, BLDC motor testbed, Heavy-duty vehicle, and Platooning.

사랑하는 부모님에게 이 논문을 바칩니다.

Contents

List of Figures

Notation and Symbols

$\mathbb{R}$	the set of all real numbers
$\mathbb{R}^n$	the n-dimensional Euclidean space
$\mathbb{R}^{n\times n}$	the space of $n \times n$ matrix with real entries

1 Effect of Measurement Noise

1.1 Introduction

1.1.1 Motivation

In control systems design, it is highly desired to achieve control performance robust against model uncertainties and external disturbances. Various robust control tools such as sliding mode control, backstepping control, L_1 adaptive control, H_∞ control, and so on, were introduced for this purpose. In particular, disturbance observer (DOB), which was first proposed in [1], has been widely used as a simple and powerful robust control tool. A large number of industry engineers and academic researchers have utilized DOB in application fields such as motion control [2–5], magnetic disk drive control [6], quadrotor control [7,8], electric bicycle control [9], automotive control [10, 11], BTT missile control [12], etc.

It should be noted that the performance of the control system with DOB may degrade when control input is constrained. Specifically, control systems with actuator saturation and DOB suffer from a poor transient and even instability if the saturation is not properly taken into account in the design. This is very similar to well-known integral controller windup phenomenon [13,14] occurring due to saturating actuator in PI controlled systems. The poor transient response or instability is recognized in [15, 16] and a modified DOB structure that can mitigate the undesired behavior was proposed and studied. Therefore, augmentation of DOB for systems with saturating actuators shall take the constrained control into account following the suggested method of [15,16]. We refer to this implementation as anti-windup restrained control of DOB (AWRC DOB) for convenience. Indeed, application of AWRC DOB was founded in [17, 18].

As it turns out, an unexpected behavior is observed in systems with AWRC DOB that no existing literature provides explanation. Specifically, zero mean measurement noise renders the system lose tracking: the mean of the output of the system shows constant tracking error only when measurement noise is present, although the measurement noise has zero mean. This phenomenon is similar to Noise Induced Tracking Error (NITE) reported in [19–21] for PI controlled systems with anti-windup. Since the two phenomena are closely related, we use the term NITE in the remainder of the paper to refer to noise induced tracking loss in systems with AWRC DOB.

In this paper, we analyze control systems with AWRC DOB to derive conditions under which the tracking loss occurs, and also quantify the tracking error with respect to system parameters and noise characteristics. The analysis is based on stochastic averaging theory developed in [22]. Main contribution of this work are as follows: 1) we provide the conditions under which NITE occurs and quantify tracking error and 2) based on the analysis, we propose controller design guidelines that can eliminate or reduce NITE in systems with AWRC DOB.

1.1.2 Chapter Outline

The outline of this paper is as follows: In Section 1.2, we show NITE Phenomenon in Control Systems with AWRC DOB. In Section 1.3, the control system with AWRC DOB is analyzed via stochastic averaging theory and main results are presented. Also, NITE example mentioned in section 1.3 is analyzed in detail. In Section 1.4, design guidelines to eliminate or reduce NITE are proposed. In Section 1.5, BLDC motor experiments are conducted to validate the efficacy of the proposed design guidelines. Finally, Conclusions are formulated in Section 1.6. All proofs are included in Appendix.

1.2 NITE Phenomenon in Control Systems with AWRC DOB

Consider a single input single output control system with AWRC DOB shown in Fig. 1.1. The transfer function $P(s)$, $C(s)$, $P_n(s)$, $Q(s)$ are plant, outer-loop controller, nominal plant, and low-pass filter (Q-filter) in DOB design, respectively. Signals r, e, u_c, u, v, d, y, n, d_{est}, z_1, z_2 are, respectively, reference, tracking error, controller output, control signal, saturated control input, exogenous disturbance, system output, measurement noise, estimated disturbance, and signals associated with DOB. The measurement noise n has a zero mean Gaussian distribution with standard deviation of σ_n. The saturating actuator is denoted by $\text{sat}_\alpha^\beta(u)$ with α and β being the lower and upper limits:

$$\text{sat}_\alpha^\beta(u) = \begin{cases} \alpha, & u < \alpha \\ u, & \alpha \le u \le \beta \\ \beta, & u > \beta \end{cases} \tag{1.1}$$

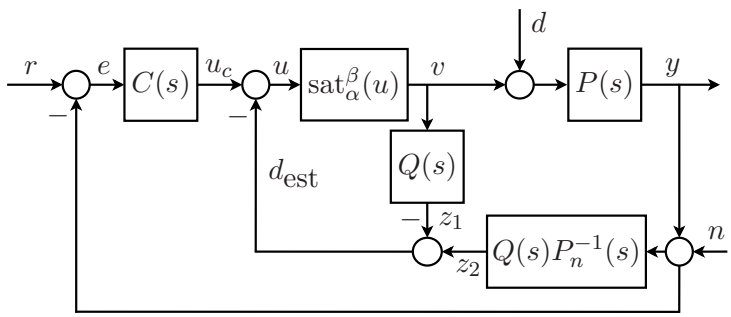

Figure 1.1: Block diagram of control systems with AWRC DOB.

For the design of DOB, $P_n(s)$ is selected to have the same relative degree as that of $P(s)$, and $Q(s)$ is selected so that $Q(s)P_n^{-1}(s)$ is a proper transfer function. For details of DOB design, see [16, 24–27, 38] and references there in.

We assume that the system of Fig. 1.1 is stable with large enough domain of attraction. Also, we assume that the reference r and the exogenous disturbance d are constants and satisfy

$$\alpha < \lim_{s\to 0} \frac{C(s)}{C(s)P(s) + P_n^{-1}(s)P(s)} r - d < \beta \tag{1.2}$$

which implies that the steady state input u_{ss} satisfies $\alpha < u_{ss} < \beta$. The derivation of (1.2) is provided in the Appendix.

Now, consider the system of Fig. 1.1 with

$$
\begin{aligned}
&P(s) = \frac{2}{s(s+3)},\ P_n(s) = \frac{1}{s(s+4)},\ Q(s) = \frac{1}{(0.1s+1)^2}, \\
&C(s) = 4,\ \alpha = -1,\ \beta = 1,\ r = 1,\ d = 0.4.
\end{aligned}
\tag{1.3}
$$

The response of the system of Fig. 1.1 with (1.3) and $n = 0$ is obtained from MATLAB/SIMULINK simulations, and shown in Fig. 1.2 (a). As expected, the system output asymptotically tracks the reference. Now the output response from simulating the dynamics of Fig. 1.1 with (1.3) and zero mean noise n with the standard deviation of $\sigma_n = 0.04$ is shown in Fig. 1.2 (b). Clearly, zero mean measurement noise induced tracking loss. The output does not track the reference anymore and shows significant bias. This phenomenon is referred to as NITE.

Analysis of NITE in feedback control systems with AWRC DOB will be carried out in subsequent section.

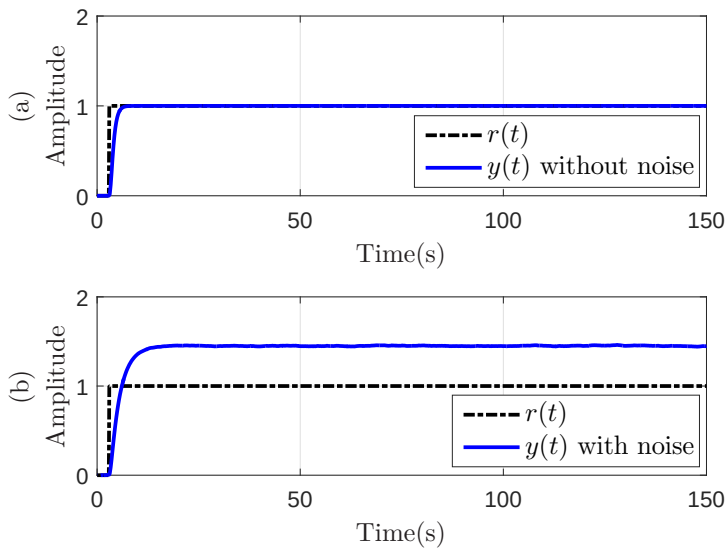

Figure 1.2: Step responses of control systems with AWRC DOB.

1.3 Analysis

1.3.1 Stochastic Averaging Theory

The analysis is based on the stochastic averaging theory [22]. A brief review is presented here. Consider the system

$$\dot{x} = f(x, n_\epsilon(t)) \tag{1.4}$$

where $x \in \mathbb{R}^k$, $f : \mathbb{R}^k \times \mathbb{R} \to \mathbb{R}^k$, and $n_\epsilon(t)$ is zero mean WSS random process with standard deviation of σ_n and bandwidth of ω_n. The subscript ϵ is intended to parameterize the noise in such a manner that $\epsilon \to 0$ as $\omega_n \to \infty$. Then, for ϵ sufficiently small, the solution $x(t)$ of (1.4) is well approximated by the solution of the averaged equation

$$\dot{\bar{x}} = \bar{f}(\bar{x}) \tag{1.5}$$

where $\bar{x} \in \mathbb{R}^k$, $\bar{f} : \mathbb{R}^k \to \mathbb{R}^k$, and $\bar{f}$ is the conditional expected value of f with respect to the distribution of $n_\epsilon(t)$, i.e.,

$$\bar{f}(\bar{x}) = E_{n_\epsilon(t)}[\bar{f}(\bar{x}, n_\epsilon(t))] \tag{1.6}$$

Since the bandwidth of the measurement noise is typically much larger than that of the closed-loop system, the behavior of (1.4) can be studied using the averaged system of (1.5).

The averaged system of Fig. 1.1 with respect to the noise process is shown in Fig. 1.3. The details of applying the stochastic averaging are included in [19–21]. For the system of Fig. 1.3, all signals are denoted by the same symbols as in Fig. 1.1, but with a bar to denote that they are the results of averaging. Notice that the measurement noise n does not appear in the system of Fig. 1.3, and the effect of n is averaged into the function $h_\alpha^\beta(\bar{u};\kappa)$ that replaces the saturating actuator $\mathrm{sat}_\alpha^\beta(u)$ in the original system.

The function $h_\alpha^\beta(\bar{u};\kappa)$ is defined as

$$\begin{aligned} h_\alpha^\beta(\bar{u};\kappa) = {} & \frac{\alpha+\beta}{2} + \frac{\bar{u}-\alpha}{2}\mathrm{erf}\left(\frac{\bar{u}-\alpha}{\sqrt{2}\kappa}\right) - \frac{\bar{u}-\beta}{2}\mathrm{erf}\left(\frac{\bar{u}-\beta}{\sqrt{2}\kappa}\right) \\ & + \frac{\kappa}{\sqrt{2\pi}}\left(\exp\left(-\frac{(\bar{u}-\alpha)^2}{2\kappa^2}\right) - \exp\left(-\frac{(\bar{u}-\beta)^2}{2\kappa^2}\right)\right) \end{aligned} \tag{1.7}$$

where the error function $\mathrm{erf}(\xi)$ and the parameter κ are, respectively, defined as

$$\begin{aligned} \mathrm{erf}(\xi) &= \frac{2}{\sqrt{\pi}} \int_0^{\xi} \exp(-t^2)\ dt, \\ \kappa &= (C_\infty + W_\infty)\sigma_n. \end{aligned} \tag{1.8}$$

For the sake of convenience, we define $Q(s)P_n^{-1}(s)$ as $W(s)$. The parameters C_∞ and W_∞ are defined as

$$\begin{aligned} C_\infty &= \lim_{s\to\infty} C(s), \\ W_\infty &= \lim_{s\to\infty} Q(s)P_n^{-1}(s). \end{aligned} \tag{1.9}$$

The derivation of $h_\alpha^\beta(\bar{u};\kappa)$ is provided in the Appendix. The properties of $h_\alpha^\beta(\bar{u};\kappa)$ are checked in the Fig. 1.4.

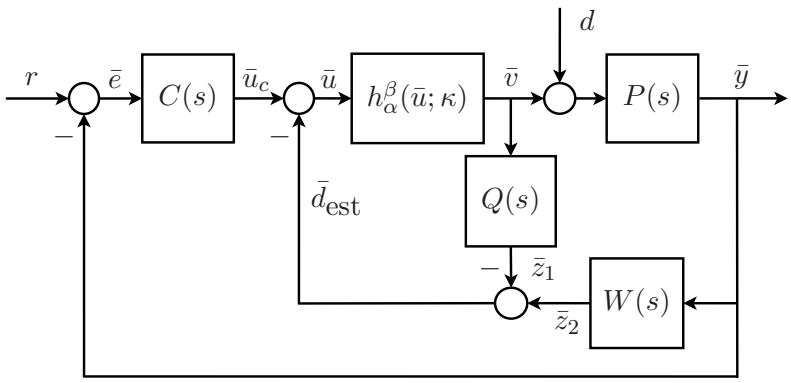

Figure 1.3: Averaged version of the feedback system of Fig. 1.1.

We now illustrate the accuracy of the averaging approach. Responses of the system in Fig. 1.3 obtained from simulations are shown in Fig. 1.5 (a). For comparison, in Fig. 1.5 (b) we show again the response of the system in Fig. 1.1 with measurement noise. The two responses are almost identical, which validates the approach of the analysis.

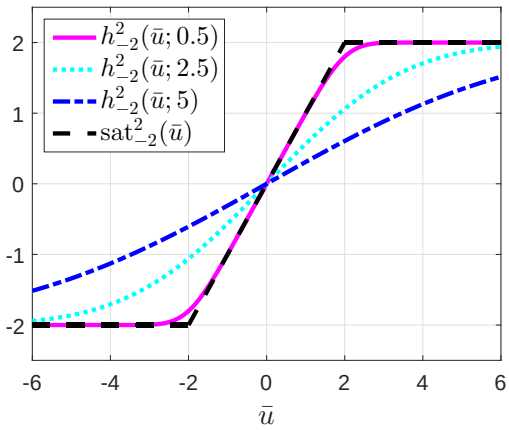

Figure 1.4: The function $\mathrm{sat}^2_{-2}(\bar{u})$ and $h^2_{-2}(\bar{u}; \kappa)$ for several values of κ.

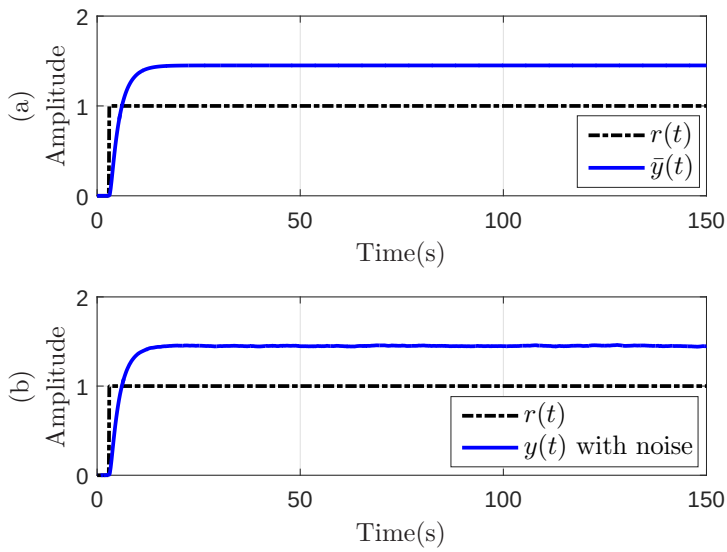

Figure 1.5: Step responses of the averaged and original control systems.

1.3.2 NITE Analysis

By analyzing the tracking error of the system of Fig. 1.3, we can quantify NITE with respect to system parameters.

Theorem 1. Assume that the outer-loop controller $C(s)$ does not include integral term, $C_0 + W_0 \neq 0$, the plant $P(s)$ does not have a pole at the origin, and the averaged version of the feedback system of Fig. 1.3 has an asymptotically stable equilibrium. The steady state error $\bar{e}_{ss}$ is defined as

$$\bar{e}_{ss} = \frac{\bar{u}_{ss} - h_\alpha^\beta(\bar{u}_{ss};\kappa) + W_0 r}{C_0 + W_0} \tag{1.10}$$

and steady state input $\bar{u}_{ss}$ is the solution of

$$\bar{u}_{ss} = C_0 r + (1 - C_0 P_0 - W_0 P_0) h_\alpha^\beta(\bar{u}_{ss};\kappa) - (C_0 P_0 + W_0 P_0) d \tag{1.11}$$

where P_0, C_0, W_0 are, respectively, the dc-gain of $P(s)$, the dc-gain of $C(s)$, and the dc-gain of $W(s)$.
Proof: See the Appendix.

The interpretations of Theorem 1 are as follows:

(1-1) Only, the level of NITE is determined by system parameters such as reference, exogenous disturbance, size of κ, dc-gain of $P(s)$, dc-gain of $C(s)$, and dc-gain of $W(s)$.

(1-2) Let us assume that reference, dc-gain of $C(s)$, dc-gain of $P_n(s)$ are fixed. As the gap between $\mathrm{sat}_\alpha^\beta(\bar{u})$ and $h_\alpha^\beta(\bar{u};\kappa)$ at a steady state input is large, the level of NITE is gradually high, i.e., $\bar{e}_{ss}$ increases as $\bar{u}_{ss} - h_\alpha^\beta(\bar{u};\kappa)$ increases.

(1-3) In the case of the control system without measurement noise, $|\bar{u}_{ss} - h_\alpha^\beta(\bar{u};\kappa)|$ must be zero. Thus, (1.10) is converted to

$$\bar{e}_{ss} = \frac{W_0 r}{C_0 + W_0} = \frac{r}{1 + C_0 P_{n0}} \tag{1.12}$$

where P_{n0} is the dc-gain of $P_n(s)$. (1.12) is a well-known mathematical formula [28] to obtain the steady state error of the unity feedback system of Fig. 1.6. This is a natural result because DOB perfectly compensates unknown disturbance and makes the real plant behave like the nominal plant in steady state.

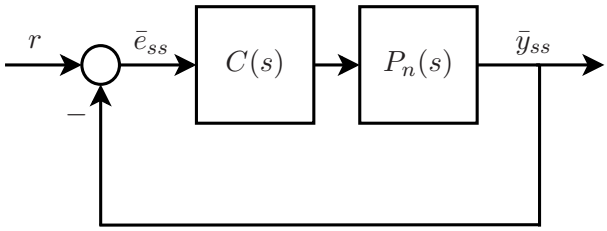

Figure 1.6: Block diagram of Fig.1.3 in the steady state.

Corollary 1. Assume that the outer-loop controller $C(s)$ does not include integral term and the plant $P(s)$ has a pole at the origin, the averaged version of the feedback system of Fig. 1.3 has an asymptotically stable equilibrium. For $C_0 + W_0 \neq 0$, the steady state error $\bar{e}_{ss}$ is defined as

$$\bar{e}_{ss} = \frac{\bar{u}_{ss} + d + W_0 r}{C_0 + W_0} \tag{1.13}$$

and steady state input $\bar{u}_{ss}$ is the solution of

$$h_\alpha^\beta(\bar{u}_{ss}; \kappa) = -d \tag{1.14}$$

where C_0, W_0 are, respectively, the dc-gain of $C(s)$ and the dc-gain of $W(s)$.
Proof: See the Appendix.

The interpretations of Corollary 1 are as follows:

(2-1) As negative disturbance $-d$ approaches to upper limit β^-, steady state error $\bar{e}_{ss}$ is close to positive infinite, i.e., $\lim_{-d\to\beta^-} \bar{e}_{ss} = +\infty$. In the case of $-d = \beta^-$, when steady state input $\bar{u}_{ss}$ is positive infinite, negative disturbance $-d$ meets with function $h_\alpha^\beta(\bar{u}_{ss}; \kappa)$, i.e., $\bar{e}_{ss} \to +\infty$ as $\bar{u}_{ss} \to +\infty$. In a similar way, as negative disturbance $-d$ approaches to the lower limit α^+, steady state error $\bar{e}_{ss}$ is close to the negative infinite, i.e., $\lim_{-d\to\alpha^+} \bar{e}_{ss} = -\infty$. In the case of $-d = \alpha^+$, when steady state input is negative infinite, negative disturbance $-d$ meets with h-function, i.e., $\bar{e}_{ss} \to -\infty$ as $\bar{u}_{ss} \to -\infty$. In other words, this means that if there exists measurement noise in the feedback system of Fig. 1.1 and negative disturbance $-d$ is placed on either lower limit α^+ or upper limit β^-, the stability of the system can be broken.

(2-2) As negative disturbance $-d$ approaches to the middle point between the upper limit and the lower limit, the steady state error is increasingly close to (1.12). i.e.,

$$\lim_{-d\to(\alpha+\beta)/2} \bar{e}_{ss} = \frac{W_0 r}{C_0 + W_0} \tag{1.15}$$

If nominal plant has a pole at the origin ($W_0 = 0$), steady state error $\bar{e}_{ss}$ is always zero, i.e., $\bar{e}_{ss} \to 0$ as $\bar{u}_{ss} \to (\alpha + \beta)/2$. This is related with properties of $h_\alpha^\beta(\bar{u}; \kappa)$ [19].

Theorem 2. In the case that outer-loop controller $C(s)$ includes integral term, the steady state error $\bar{e}_{ss}$ of the feedback system of Fig. 1.3 is always zero.
Proof: See the Appendix.

Again, consider NITE example mentioned in Section 2. Fig. 1.7 shows plot of $h_{-1}^1(\bar{u}; 4.16)$ and $\mathrm{sat}_{-1}^1(\bar{u})$ over inputs $\bar{u}$. Refer to Corollary 1. Negative disturbance $-d$ meets with $h_{-1}^1(\bar{u}; 4.16)$ at the point of $\bar{u}_{ss} = -2.2025$. Consequently, steady state error is determined as $\bar{e}_{ss} = -0.4506$. Fig. 1.8 represents steady state errors over exogenous disturbances and we can understand interpretation of Corollary 1 more deeply.

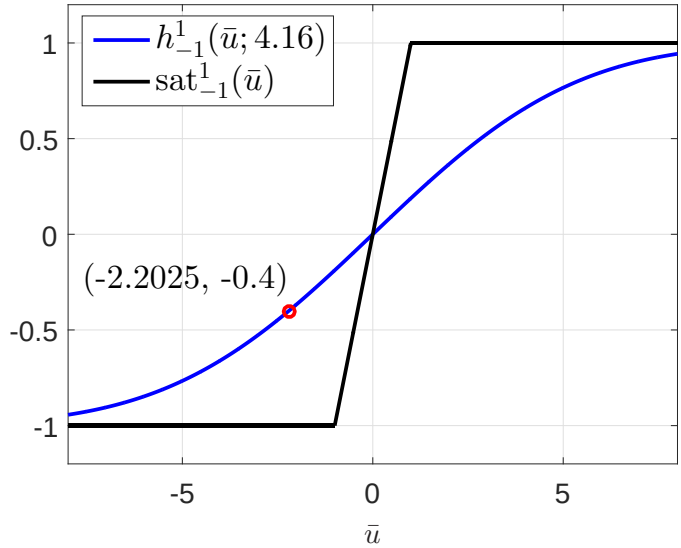

Figure 1.7: Plot of $h_{-1}^1(\bar{u}; 4.16)$ and $\mathrm{sat}_{-1}^1(\bar{u})$.

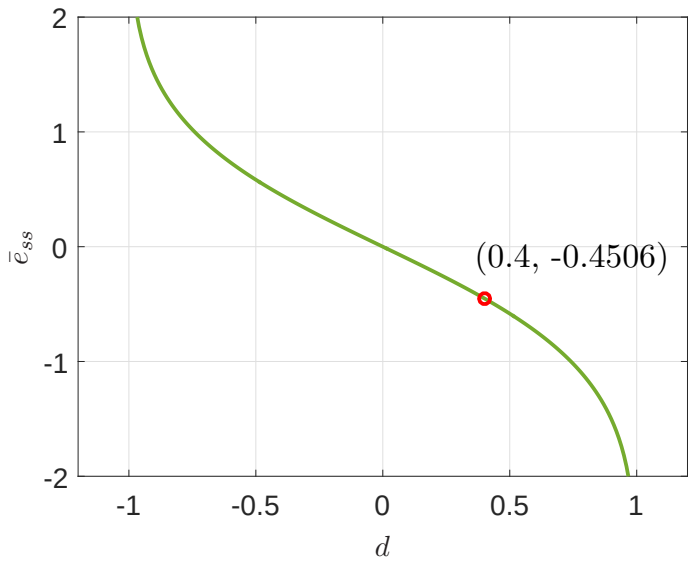

Figure 1.8: Steady state errors over several disturbances.

1.4 NITE Mitigation Method

Worthwhile design techniques that can reduce NITE are as follows:

M1: Include integral control in $C(s)$.
By Theorem 2, this makes the tracking error be zero in the steady state. However, this has a drawback in that poor transient performance caused by an unwanted pole addition at zero may be inevitable. See Fig. 1.9.

M2: Increase relative degree of $Q(s)$.
Assume that transfer functions $P_n(s)$ and $Q(s)$ are defined as follows:

$$\begin{aligned} P_n(s) &= \frac{b_p s^p + b_{p-1}s^{p-1} + \cdots + b_1 s + b_0}{a_q s^q + a_{q-1}s^{q-1} + \cdots + a_1 s + a_0} \\ Q(s) &= \frac{d_k(\tau s)^k + d_{k-1}(\tau s)^{k-1} + \cdots + d_1(\tau s) + c_0}{(\tau s)^l + c_{l-1}(\tau s)^{l-1} + \cdots + c_1(\tau s) + c_0} \end{aligned} \tag{1.16}$$

where parameters p, q, k, l are positive numbers with $q \geq p$ and $l \geq k$ and parameter τ is a positive constant, which determines cutoff frequency of lowpass filter $Q(s)$. By using (1.9) and (1.16), parameter W_∞ is defined as

$$W_\infty = \begin{cases} (a_q d_k \tau^{k-l})/b_p, & k+q = l+p \\ 0, & k+q < l+p \end{cases} \tag{1.17}$$

In the case that $C(s)$ should not include any integral term due to transient performance problem, we recommend taking κ are small as possible. This is because $h_\alpha^\beta(\bar{u};\kappa)$ is gradually similar to $\mathrm{sat}_\alpha^\beta(\bar{u})$ as κ decreases into zero. Here, W_∞ should be modulated rather than C_∞ in that C_∞ is directly related to system performance. If there is nothing wrong with implementation of DOB, we recommend relative degree of $Q(s)$ is bigger than that of $P_n(s)$. This is because W_∞ is changed into 0. Although this method can not eliminate NITE phenomenon because C_∞ is not zero, this method is quite useful in that transient performance can be somehow guaranteed and NITE is decreased. See Fig. 1.10.

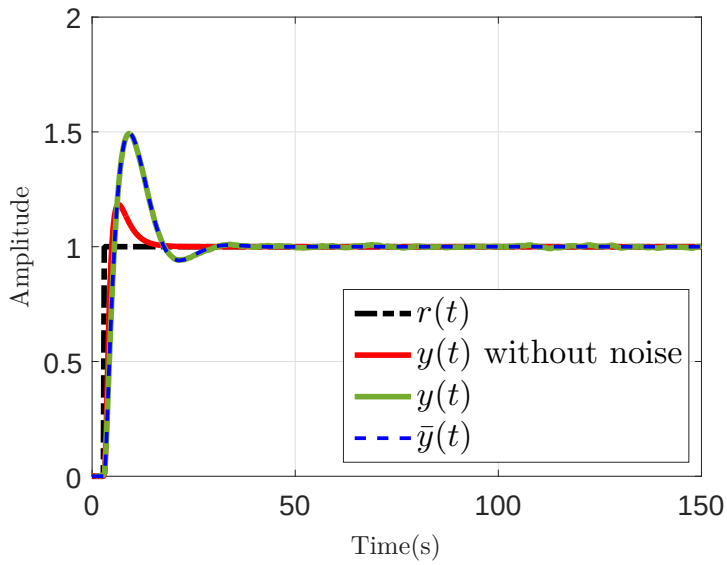

Figure 1.9: Application of the proposed design method (1) to NITE example.

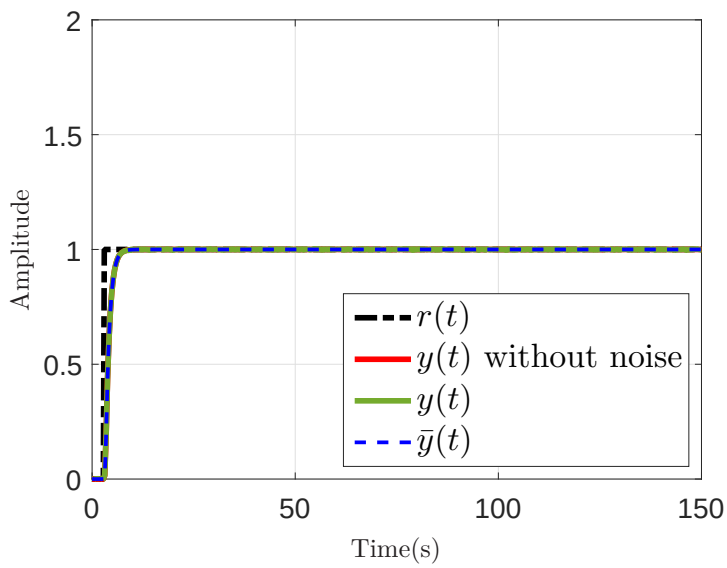

Figure 1.10: Application of the proposed design method (2) to NITE example.

1.5 DC Motor Experimental Results

We use brushless DC (BLDC) motor drive testbed to validate the effectiveness of the proposed controller design techniques. Fig.1.11 shows BLDC motor system. The experiment environment is comprised of a DSP evaluation board and a 26.2W, eight-pole BLDC motor drive. The DSP evaluation board is fitted with a Texas Instruments TMS320F28335 floating-point DSP and a 50W three phase full-bridge inverter. A built-in-Hall-effect sensor measures the speed of BLDC motor with sampling rate of 30kHz. Controllers are discretized using the Tustin approximation.

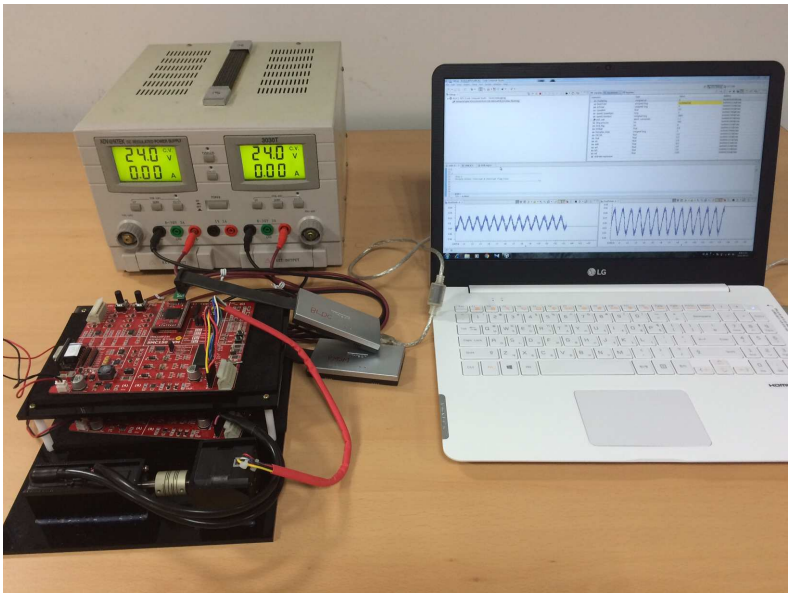

Figure 1.11: Blushless DC motor drive testbed.

The control objective is to regulate the rotative velocity of BLDC motor under the autonomously generated sensor noise and the exogenous disturbances. The speed reference of BLDC motor is set to 2000rpm. Transfer function $P_n(s)$ and $Q(s)$ are chosen as

$$P_n(s) = \frac{228}{0.01s(0.03s+1)}[\text{rpm/V}], \quad Q(s) = \frac{1}{(0.003s+1)^2} \tag{1.18}$$

The outer-loop controller is properly chosen as $C(s) = 0.0006[\text{V/rpm}]$. In order to protect

the BLDC motor, the permissible range of input voltage is set as follows.

$$\mathrm{sat}_0^{9.6}(u(t)) = \begin{cases} 0, & u(t) < 0 \\ u(t), & 0 \leq u(t) \leq 9.6 \\ 9.6, & u(t) > 9.6 \end{cases} \tag{1.19}$$

In order to observe the variation of NITE, after 5 seconds, the exogenous disturbance is induced into the motor input. Before 5 seconds, the exogenous disturbance is set to zero. Consider Fig. 1.12 and Fig. 1.13. All of subplot in the figures show the rotative velocity of BLDC motor. The first subplot of each figure shows that the motor speed does not track the reference due to NITE phenomenon. In the first subplot of each figure, we can check that NITE is increased after 5 seconds. This is because exogenous disturbance causes movement of operating point (steady state input $\bar{u}_{ss}$). As the operating point approaches to the nearby region of saturation limit, the gap between saturating actuator and h-function increasingly increases. It means that NITE increases. In the second subplot in Fig. 1.12, controller design technique (1) is applied to the motor control system. We change $C(s) = 0.0006$ into $C(s) = 0.0005(s + 4)/s$. As we mention in Section 6, by Collorary 1, NITE phenomenon is not appeared. In the second subplot in Fig. 1.13, new controller design technique (2) is applied to the motor control system. We change $Q(s) = 1/(0.003s+1)^2$ into $Q(s) = 1/(0.003s+1)^3$. NITE phenomenon is not disappeared completely. However, NITE is surely decreased. Through experimental results, we can assure that proposed design techniques to reduce NITE are quite useful.

1.6 Conclusions

Disturbance observer (DOB) has been widely utilized in many industries. However, control systems with actuator saturation and DOB may suffer from a poor transient and even instability if the saturation is not properly taken into account in the design. For this reason, AWRC DOB was proposed in [15,16]. The AWRC DOB extends range of usage of DOB. However, while we use it, we observed the interesting phenomenon where the step response of the control system has a specific tracking error under the existence of measurement noise. we denominated such phenomenon as NITE phenomenon in AWRC DOB control system. In this paper, based on stochastic theory, we defined the mathematical formula that can numerically quantify the tracking error in the steady state and analyzed it. Also, we introduced the controller design techniques that can effectively eliminate or reduce NITE. Finally, we verified the usefulness of the proposed techniques using BLDC motor.

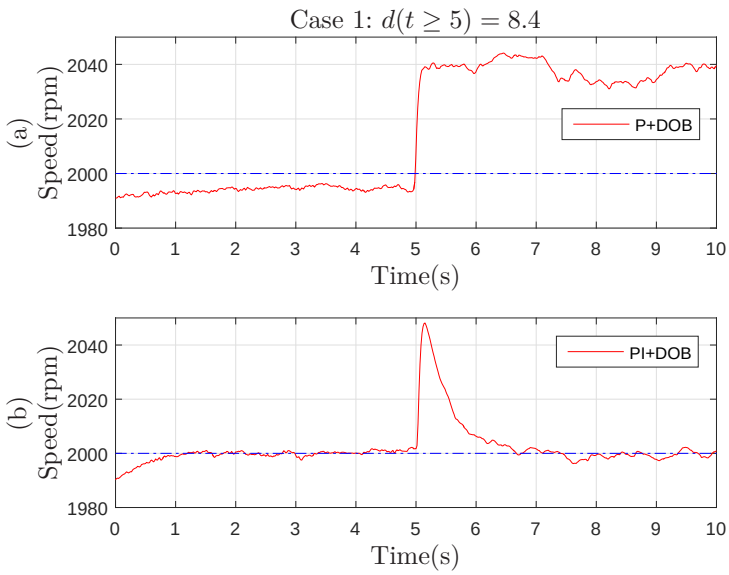

Figure 1.12: The application of controller design technique (1).

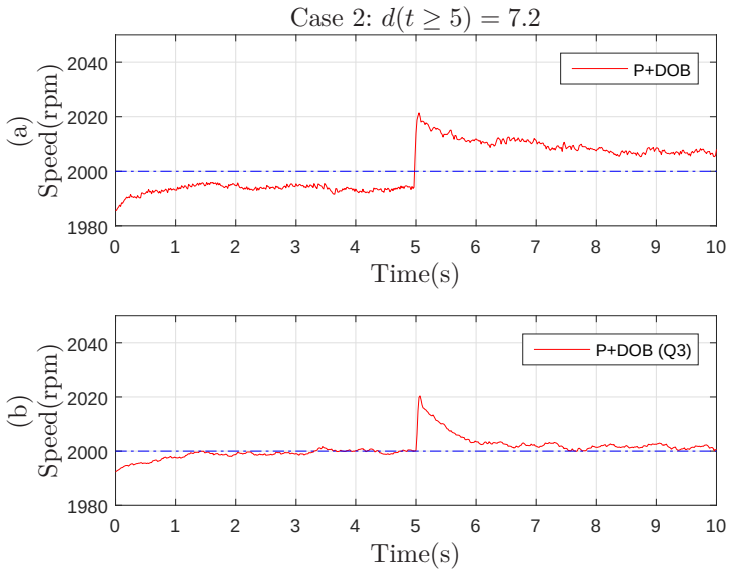

Figure 1.13: The application of controller design technique (2).

1.7 Appendix

Derivation of (1.2)

Consider Fig. 1.1. Assuming $n = 0$ and $v = u$, input u is obtained as

$$\begin{aligned} u = &\frac{C(s)}{1 - Q(s) + C(s)P(s) + Q(s)P_n^{-1}(s)P(s)} r \\ &- \frac{C(s)P(s) + Q(s)P_n^{-1}(s)P(s)}{1 - Q(s) + C(s)P(s) + Q(s)P_n^{-1}(s)P(s)} d \end{aligned} \tag{1.20}$$

By using (1.20) and $Q_0 = 1$ (For DOB design, dc-gain of $Q(s)$ should be one), steady state input u_{ss} is represented as

$$u_{ss} = \lim_{s \to 0} \frac{C(s)}{C(s)P(s) + P_n^{-1}(s)P(s)} r - d \tag{1.21}$$

(1.17) and (1.21) make (1.2).

Derivation of (1.7)

A state space realization of the feedback system in Fig. 1.1 is derived as

$$\begin{aligned} \dot{x}_p &= A_p x_p + B_p(v + d), \\ y &= C_p x_p, \\ \dot{x}_c &= A_c x_c + B_c(r - y - n), \\ u_c &= C_c x_c + D_c(r - y - n), \\ \dot{x}_q &= A_q x_q + B_q v, \\ z_1 &= C_q x_q, \\ \dot{x}_w &= A_w x_w + B_w(y + n), \\ z_2 &= C_w x_w + D_w(y + n), \end{aligned} \tag{1.22}$$

where $x_p \in \mathbb{R}^{k_p}$ is state vector of $P(s)$ and A_p, B_p and C_p are the system matrices of appropriate dimensions, $x_c \in \mathbb{R}^{k_c}$ is state vector of $C(s)$ and A_c, B_c, C_c and D_c are the system matrices of appropriate dimensions, $x_q \in \mathbb{R}^{k_q}$ is state vector of $Q(s)$ and A_q, B_q and C_q are the system matrices of appropriate dimensions, $x_w \in \mathbb{R}^{k_w}$ is state vector of $W(s)$ and A_w, B_w, C_w and D_w are the system matrices of appropriate dimensions.

$$\begin{bmatrix} \dot{x}_p \\ \dot{x}_c \\ \dot{x}_q \\ \dot{x}_w \end{bmatrix} = \begin{bmatrix} A_p & 0 & 0 & 0 \\ -B_cC_p & A_c & 0 & 0 \\ 0 & 0 & A_q & 0 \\ B_wC_p & 0 & 0 & A_w \end{bmatrix} \begin{bmatrix} x_p \\ x_c \\ x_q \\ x_w \end{bmatrix} + \begin{bmatrix} B_p & 0 & B_p & 0 \\ 0 & B_c & 0 & -B_c \\ B_q & 0 & 0 & 0 \\ 0 & 0 & 0 & B_w \end{bmatrix} \begin{bmatrix} v \\ r \\ d \\ n \end{bmatrix},$$

$$u = \begin{bmatrix} -(D_cC_p + D_wC_p) & C_c & C_q & -C_w \end{bmatrix} \begin{bmatrix} x_p \\ x_c \\ x_q \\ x_w \end{bmatrix} + \begin{bmatrix} 0 & D_c & 0 & -(D_c + D_w) \end{bmatrix} \begin{bmatrix} v \\ r \\ d \\ n \end{bmatrix} \tag{1.23}$$

According to (1.23), input u is decomposed into $u = \mu - (D_c + D_w)n$ with μ representing the portion of u not directly depending on n. The h-function denoted by $h_\alpha^\beta(\bar{u}; (D_c + D_w)\sigma_n)$ is defined as the conditional expected value of $\mathrm{sat}_\alpha^\beta(u)$ with respect to zero mean Gaussian noise n.

$$\begin{aligned} h_\alpha^\beta(\mu; (D_c + D_w)\sigma_n) &= E_n[\mathrm{sat}_\alpha^\beta(u)] \\ &= \int_{-\infty}^{\infty} \mathrm{sat}_\alpha^\beta(\mu - (D_c + D_w)n) \frac{1}{\sqrt{2\pi}\sigma_n} \exp(-\frac{n^2}{2\sigma_n^2}) dn \\ &= \frac{\alpha}{\sqrt{2\pi}\sigma_n} \int_{-\infty}^{\alpha} \exp(-\frac{n^2}{2\sigma_n^2}) dn \\ &+ \frac{1}{\sqrt{2\pi}\sigma_n} \int_{\alpha}^{\beta} (\mu - (D_c + D_w)n) \exp(-\frac{n^2}{2\sigma_n^2}) dn \\ &+ \frac{\beta}{\sqrt{2\pi}\sigma_n} \int_{\beta}^{\infty} \exp(-\frac{n^2}{2\sigma_n^2}) dn \end{aligned} \tag{1.24}$$

Calculating (1.24) makes (1.7).

Proof of Theorem 1

A state space realization of the feedback system of Fig. 1.3 is defined as

$$\begin{aligned}
\dot{\bar{x}}_p &= A_p\bar{x}_p + B_p(h_\alpha^\beta(\bar{u}; (D_c + D_w)\sigma_n) + d), \\
\bar{y} &= C_p\bar{x}_p, \\
\dot{\bar{x}}_c &= A_c\bar{x}_c + B_c(r - \bar{y}), \\
\bar{u}_c &= C_c\bar{x}_c + D_c(r - \bar{y}), \\
\dot{\bar{x}}_q &= A_q\bar{x}_q + B_q h_\alpha^\beta(\bar{u}; (D_c + D_w)\sigma_n), \\
\bar{z}_1 &= C_q\bar{x}_q, \\
\dot{\bar{x}}_w &= A_w\bar{x}_w + B_w\bar{y}, \\
\bar{z}_2 &= C_w\bar{x}_w + D_w\bar{y}, \\
\bar{u} &= \bar{u}_c - \bar{d}_{\text{est}}
\end{aligned} \tag{1.25}$$

where $\bar{x}_p$ is state vector of $P(s)$, $\bar{x}_c$ is state vector of $C(s)$, $\bar{x}_q$ is state vector of $Q(s)$, and $\bar{x}_w$ is state vector of $W(s)$. The steady state of the system is defined as

$$\begin{aligned}
A_p\bar{x}_p + B_p(h_\alpha^\beta(\bar{u}; (D_c + D_w)\sigma_n) + d) &= 0, \\
A_c\bar{x}_c + B_c(r - \bar{y}) &= 0, \\
A_q\bar{x}_q + B_q h_\alpha^\beta(\bar{u}; (D_c + D_w)\sigma_n) &= 0, \\
A_w\bar{x}_w + B_w\bar{y} &= 0
\end{aligned} \tag{1.26}$$

Since the existence of asymptotically stable equilibrium is assumed, we can obtain

$$\begin{aligned}
\bar{u}_{ss} &= \bar{u}_{c,ss} - \bar{d}_{\text{est},ss} \\
&= C_c\bar{x}_c + D_c(r - \bar{y}_{ss}) + C_q\bar{x}_q - (C_w\bar{x}_w + D_w\bar{y}_{ss})
\end{aligned} \tag{1.27}$$

where $\bar{u}_{ss}$, $\bar{u}_{c,ss}$, $\bar{d}_{\text{est},ss}$, $\bar{y}_{ss}$ are, respectively, the steady state values of $\bar{u}$, $\bar{u}_c$, $\bar{d}_{\text{est}}$, and $\bar{y}$. By using (1.26) and (1.27), we can obtain

$$\bar{u}_{ss} = (-C_cA_c^{-1}B_c + D_c)(r - \bar{y}_{ss}) + (-C_qA_q^{-1}B_q)h_\alpha^\beta(\bar{u}_{ss}; (D_c + D_w)\sigma_n) - (-C_wA_w^{-1}B_w + D_w)\bar{y}_{ss} \tag{1.28}$$

By using $C_0 = -C_cA_c^{-1}B_c + D_c$, $Q_0 = -C_qA_q^{-1}B_q$, and $W_0 = -C_wA_w^{-1}B_w + D_w$, (1.28) is rewritten by

$$\bar{u}_{ss} = C_0(r - \bar{y}_{ss}) + Q_0h_\alpha^\beta(\bar{u}_{ss}; (D_c + D_w)\sigma_n) - W_0\bar{y}_{ss} \tag{1.29}$$

where C_0, Q_0, W_0 are, respectively, the dc-gain of $C(s)$, the dc-gain of $Q(s)$, and the dc-gain of $W(s)$.

By using $\bar{e}_{ss} = r - \bar{y}_{ss}$ and $Q_0 = 1$, (1.29) is rewritten by

$$\begin{aligned} \bar{u}_{ss} &= C_0\bar{e}_{ss} + h_\alpha^\beta(\bar{u}_{ss}; (D_c + D_w)\sigma_n) - W_0(r - \bar{e}_{ss}) \\ &= (C_0 + W_0)\bar{e}_{ss} + h_\alpha^\beta(\bar{u}_{ss}; (D_c + D_w)\sigma_n) - W_0 r \end{aligned} \tag{1.30}$$

For $C_0 + W_0 \neq 0$, the steady state error $\bar{e}_{ss}$ is defined as follows.

$$\bar{e}_{ss} = \frac{\bar{u}_{ss} - h_\alpha^\beta(\bar{u}_{ss}; (D_c + D_w)\sigma_n) + W_0 r}{C_0 + W_0} \tag{1.31}$$

In order to obtain $\bar{u}_{ss}$, (1.29) is rewritten by

$$\bar{u}_{ss} = C_0 r + h_\alpha^\beta(\bar{u}_{ss}; (D_c + D_w)\sigma_n) - (C_0 + W_0)\bar{y}_{ss} \tag{1.32}$$

By using $\bar{y}_{ss} = C_p\bar{x}_p = (-C_pA_p^{-1}B_p)(h_\alpha^\beta(\bar{u}_{ss}; (D_c + D_w)\sigma_n) + d) = P_0(h_\alpha^\beta(\bar{u}_{ss}; (D_c + D_w)\sigma_n) + d)$, we can obtain

$$\begin{aligned} \bar{u}_{ss} &= C_0 r + h_\alpha^\beta(\bar{u}_{ss}; (D_c + D_w)\sigma_n) - (C_0 + W_0)P_0(h_\alpha^\beta(\bar{u}_{ss}; (D_c + D_w)\sigma_n) + d) \\ &= C_0 r + (1 - C_0P_0 - W_0P_0)h_\alpha^\beta(\bar{u}_{ss}; (D_c + D_w)\sigma_n) - (C_0P_0 + W_0P_0)d \end{aligned} \tag{1.33}$$

where P_0 is the dc-gain of $P(s)$.

Proof of Corollary 1

Since $P(s)$ has a pole at the origin, for an equilibrium to exist, the input to the plant $P(s)$ must be zero in the steady state, i.e.,

$$d + h_\alpha^\beta(\bar{u}_{ss}; (D_c + D_w)\sigma_n) = 0 \tag{1.34}$$

Therefore, by using (1.31), we can obtain

$$\bar{e}_{ss} = \frac{\bar{u}_{ss} + d + W_0 r}{C_0 + W_0} \tag{1.35}$$

where $\bar{u}_{ss}$ is the solution of (1.34).

Proof of Theorem 2

Since the outer-loop controller $C(s)$ has a pole at the origin, for an equilibrium to exist, the input to $C(s)$ must be zero in the steady state. Therefore, $\bar{e}_{ss}$ must be zero.

1.8 MATLAB CODE

I have saved NITE code in the common fold. To understand code information more perfectly, I recommend you to check files in common fold.

```

clear all
clc

%% Operation Time
T = 150;

%% Reference
r = 1;
rsteptime = 3;

%% Disturbance
StepTime = 3;
d = 0.4;

%% Gaussian Noise
Mean = 0;
Variance = 0.0016;
sigma = sqrt(Variance);
SampleTime = 0.0001;

%% Saturation
beta = 1;
alpha = -1;

%% PID Gain
P = 4;
I = 0;
D = 0;

%% Plant & Nominal Plant & Q Filter & QP
Pn = [2];
Pd = [1 3 0];

Pnn = [1];
Pnd = [1 4 0];

Tau = 0.1;

Qn = [1];
%Qd = [Tau^2, 2*Tau, 1];
Qd = [Tau^3, 3*Tau^2, 3*Tau, 1];

Wn = conv(Pnd, Qn);
Wd = conv(Pnn, Qd);

%% Comparision
P1 = 4;
I1 = 0;
D1 = 0;

Qn1 = [1];
Qd1 = [Tau^2, 2*Tau, 1];

Wn1 = conv(Pnd, Qn1);
Wd1 = conv(Pnn, Qd1);


%% DC Gain
dcgainP = dcgain(tf(Pn, Pd));
dcgainPn = dcgain(tf(Pnn, Pnd));
dcgainW = dcgain(tf(Wn,Wd));

%% h Function
Dc = 4;
```

```
%Dw = 100;
Dw = 0;

ubar = -10:0.01:10;
hFunction = (alpha+beta)./2 + (ubar-alpha)./2.*erf((ubar-alpha)./(sqrt(2).*(Dc+Dw).*sigma))
 - (ubar-beta)./2.*erf((ubar-beta)./(sqrt(2).*(Dc+Dw).*sigma))
 + (Dc+Dw).*sigma./sqrt(2*pi).*(exp(-((ubar-alpha).^2)./(2.*((Dc+Dw).^2.*sigma.^2)))
 -exp(-((ubar-beta).^2)./(2.*((Dc+Dw).^2*sigma.^2))));

t1 = -10:0.01:alpha;
h1 = alpha*ones(901);
t2 = alpha:0.01:beta;
h2 = t2;
t3 = beta:0.01:10;
h3 = beta*ones(901);

%% For Error Solution
j = 1;

for dValue = - 1 : 0.01 : 1

    temp0(j) = dValue;
    f = @(U)(alpha+beta)./2 + (U-alpha)./2.*erf((U-alpha)./(sqrt(2).*(Dc+Dw).*sigma))
    - (U-beta)./2.*erf((U-beta)./(sqrt(2).*(Dc+Dw).*sigma)) + (Dc+Dw).*sigma./sqrt(2*pi)
    .*(exp(-((U-alpha).^2)./(2.*((Dc+Dw).^2.*sigma.^2)))
    -exp(-((U-beta).^2)./(2.*((Dc+Dw).^2*sigma.^2)))) + dValue;
    uValue = fzero(f,[-100000,100000]);
    temp1(j) = uValue;
    eValue = (uValue+dValue+dcgainW*r)/(P + dcgainW);
    temp2(j) = eValue;
    j=j+1;

end

aaa = d;
bbb = -0.4506;
%txt1 = '$\ \Leftarrow(0.4, -0.4506)$';
txt1 = '$\mbox{(0.4, -0.4506)}$';

aaa2 = -2.2025;
bbb2 = -d;
txt2 = '$\mbox{(-2.2025, -0.4)}$';

kkkk = -d*ones(size(ubar,2));

%% Asym line
asym = -10:0.01:10;

%% Run simultion
sim('DOBpractice1');

%% Figure
figure(1);
plot(Scope.time,Scope.data(:,1),'k-.','linewidth',3); hold on;
plot(Scope.time,Scope.data(:,2),'r','linewidth',3); hold on;
plot(Scope.time,Scope.data(:,3),'linewidth',3,'Color',[0.46 0.67 0.18]); hold on;
plot(Scope.time,Scope.data(:,5),'b--','linewidth',2); hold off;
set(gca,'fontsize',15);
xlabel('$Time(s)$');
ylabel('$Amplitude$');
legend('$r(t)$','$y(t)$ without noise','$y(t)$','${\bar y}(t)$');
ylim([0,2]);
grid on;

figure(2);
plot(ubar,hFunction,'b','linewidth',2); hold on;
plot(t1,h1,'k','linewidth',2); hold on;
```

```
plot(t2,h2,'k','linewidth',2); hold on;
plot(t3,h3,'k','linewidth',2); hold on;
plot(aaa2,bbb2,'ro','linewidth',2); hold off;
set(gca,'fontsize',15);
text(aaa2,bbb2,txt2);
xlabel('$u$','fontsize',18);
legend('$h_{-1}^{1}({\bar u};4.16)$','$\mbox{sat}_{-1}^{1}({\bar u})$');
xlim([-8,8]);
ylim([-1.2,1.2]);
grid on;

figure(3);
plot(temp0,temp2,'linewidth',2,'Color',[0.46 0.67 0.18]); hold on;
plot(aaa,bbb,'ro','linewidth',2);
set(gca,'fontsize',15);
text(aaa,bbb,txt1);
xlabel('$d$','fontsize',17);
ylabel('$e$','fontsize',17);
xlim([-1.2,1.2]);
ylim([-2,2]);
grid on;

figure(4);
plot(Scope.time,Scope.data(:,1),'k','linewidth',2); hold on;
%plot(Scope.time,Scope.data(:,2),'r'); hold on;
plot(Scope.time,Scope.data(:,3),'r','linewidth',2); hold on;
%plot(Scope.time,Scope.data(:,5),'b--','linewidth',2); hold off;
set(gca,'fontsize',15);
xlabel('Time(s)','fontsize',17);
ylabel('Amplitude','fontsize',17);
legend('$r(t)$','$y(t)$');
ylim([0,2]);
grid on;

figure(5);
subplot(2,1,1)
plot(Scope.time,Scope.data(:,1),'k-.','linewidth',2); hold on;
plot(Scope.time,Scope.data(:,2),'b','linewidth',2); hold on;
set(gca,'fontsize',12);
xlabel('$Time(s)$');
ylabel({'$a$','$Amplitude$'});
legend('$r(t)$','$y(t)$ without noise');
ylim([0,2]);
grid on;

subplot(2,1,2)
plot(Scope.time,Scope.data(:,1),'k-.','linewidth',2); hold on;
plot(Scope.time,Scope.data(:,3),'b','linewidth',2); hold off;
set(gca,'fontsize',12);
xlabel('$Time(s)$');
ylabel({'$b$','$Amplitude$'});
legend('$r(t)$','$y(t)$ with noise');
ylim([0,2]);
grid on;

figure(6);
subplot(2,1,1)
plot(Scope.time,Scope.data(:,1),'k-.','linewidth',2); hold on;
plot(Scope.time,Scope.data(:,5),'b','linewidth',2); hold on;
set(gca,'fontsize',12);
xlabel('$Time(s)$');
```

```
ylabel({'$a$','$Amplitude$'});
legend('$r(t)$','${\bar y}(t)$');
ylim([0,2]);
grid on;

subplot(2,1,2)
plot(Scope.time,Scope.data(:,1),'k-.','linewidth',2); hold on;
plot(Scope.time,Scope.data(:,3),'b','linewidth',2); hold off;
set(gca,'fontsize',12);
xlabel('$Time(s)$'); ylabel({'$b$','$Amplitude$'}); legend('$r(t)$','$y(t)$ with noise');
ylim([0,2]); grid on;
```

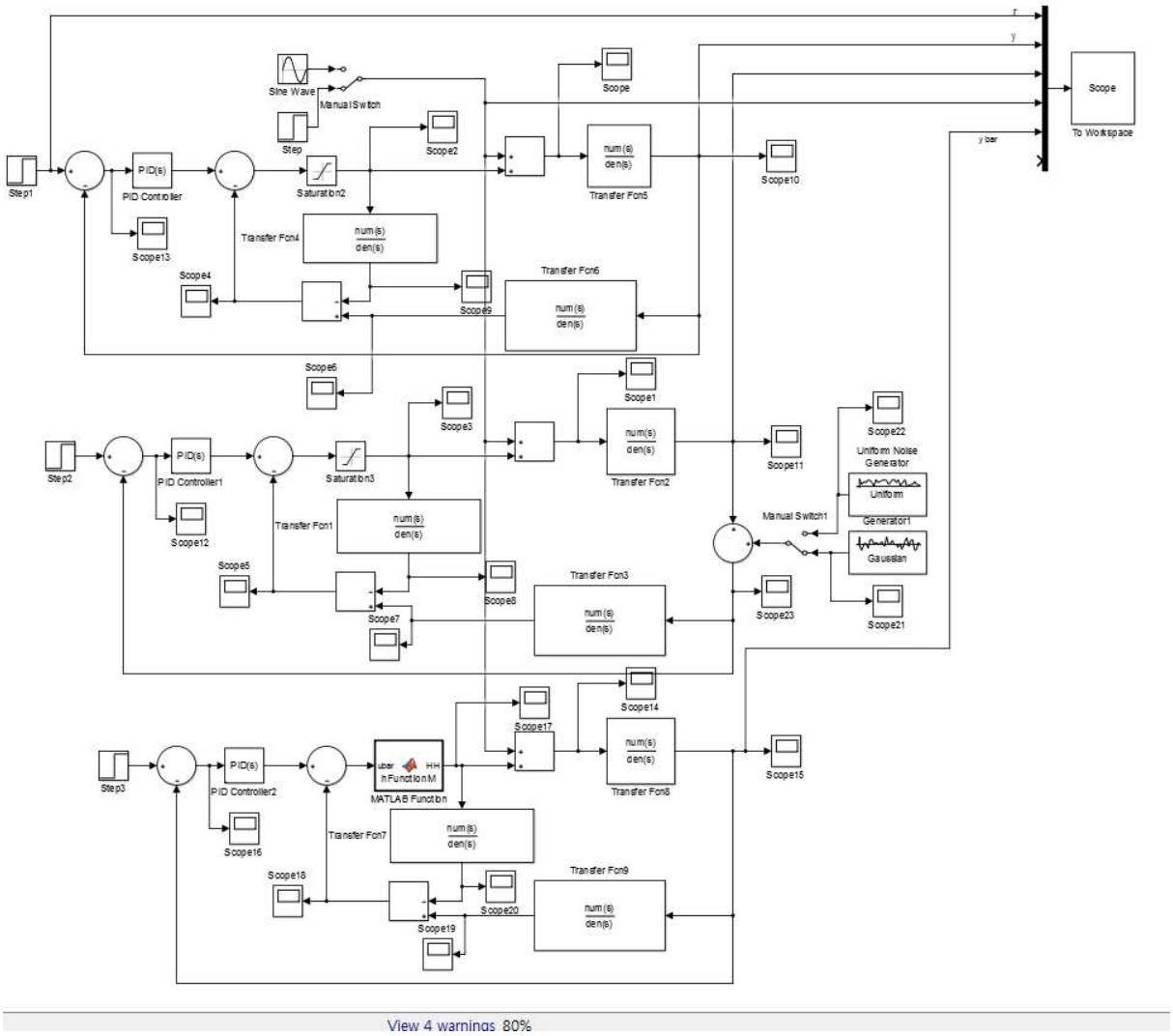

Figure 1.14: The simulink of NITE example

2 Applications to Heavy-duty Vehicle Platooning

2.1 Introduction

2.1.1 Motivation

As international economy is gradually developed, financial gain caused by goods transport has been increased. Accordingly, goods delivery services of heavy-duty vehicles has grown importance. However, long distance service for goods transport causes serious environmental disruption problem in that heavy-duty vehicles can not avoid excessive fuel consumption. For such reasons, international society has made an effort to find appropriate solution which can reduce usage of limited natural resources. As one solution, platoon technique which can reduce fuel consumption by keeping the distance between vehicles short enough to decrease aerodynamic drag has been introduced. Nowadays, platoon technique has received much attention and has been studied a lot [29–35].

Velocity reference tracking of vehicles in platoon is important issue to keep appropriate distance. However, real vehicles can not track the velocity reference due to effect of exogenous disturbances such as slope effects. For such reason, recent papers have introduced proper solution which can compensate exogenous disturbances [36, 37]. In the paper, by using model predictive control (MPC) approach based on correct vehicle model, vehicles track reference. Especially, we want to stress that all vehicles in platoon have utilized road data in order to compensate external disturbances in the paper. However, this approach may have much weakness in the case that vehicle mass is changed by loading heavy cargo (It means that vehicle dynamics as well as disturbances are changed.) or velocities of vehicles are affected by unpredictable input disturbance. Thus, it must be hard to keep short distance among vehicles in existing MPC approach. Also, in the approach, all vehicle has

used expensive road data.

By applying new vehicle controller in existing platoon approach which is proposed in [36, 37], we overcame such drawbacks several problems. Here, we presented new vehicle controller which use disturbance observer (DOB) as a sort of robust controller [3, 15–17, 38, 39]. This paper has several contribution as follows: (1) In the case that incorrect vehicle model caused by uncertain mass affects vehicle control, vehicles may not track velocity reference. However, in this proposed method, vehicles in platoon compensate effect of model uncertainty and tracks reference. (2) Vehicles may be affected by incorrect disturbances caused by change of mass, wrong slope data obtained from GPS malfunction, frequent changes of road friction and rolling coefficient, etc. If vehicles in platoon are affected by unknown exogenous disturbances, vehicles must not track velocity references. However, proposed method compensates exogenous disturbances. (3) In existing MPC approach in [36, 37], all vehicles need road data to compensate external disturbances. If all vehicles use road data, expensive fee must be paid. However, in this proposed approach, fee is considerably reduced in that only first vehicle must not need road slope data to produce velocity profile of platoon coordinator.

2.1.2 Chapter Outline

The outline of this paper is as follows: In Section 2.2, we explain platoon architecture. In Section 2.3, vehicle and platoon model is proposed. In Section 2.4, vehicle controller based on disturbance observer is presented. In Section 2.5, platoon coordinator is briefly introduced. In Section 2.6, simulations are conducted by using real road data taken in highways of Sweden. In Section 2.7, conclusion are formulated.

2.2 Platoon Architecture

Platoon architecture consists of mission planner, road data base, platoon coordinator, and vehicle controller, and this is shown in Fig. 2.1. Mission planner suggests platoon opportunities to vehicles on the roadway and sends mean velocity which vehicles in platoon should follow. Road data base sends slope data, speed limit (minimum and maximum speed) on the roadway segment to first vehicle fitted with platoon coordinator. Here, initial road data is stored by previous platoon vehicles. Thus, road which previous vehicles passed is updated consistently. See Fig. 2.4. We can check that real road data is similar estimated road data taken in previous platoon. Based on such information, platoon coordinator

uses dynamic programming formulation and produces optimal speed profile over absolute position that can reduce fuel consumption while maintaining mean velocity. After sending optimal velocity profile to each vehicle, each reference generator fitted with vehicle make new speed profile with respect to optimal speed profile obtained from platoon coordinator and speed data of preceding vehicles. In order to track new speed profile, each vehicle calculates the amount of fuel which is injected into vehicle and sprays it to engine system.

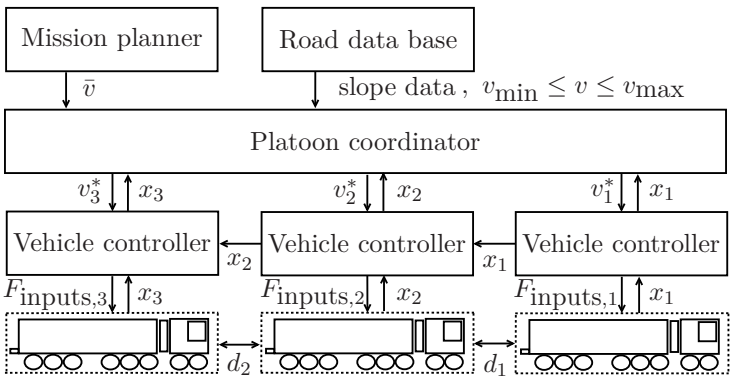

Figure 2.1: Platoon architecture.

Based on wireless sensor unit, information exchange of vehicles within platoon is operated in real time. Global positioning system (GPS) computes absolute position of each vehicle and radar device measures distance and relative speed between current vehicle and preceding vehicle. Also, engine and gear management system controls engine and brake management system controls the braking actuators. For detailed information, see [36,37].

In order to evaluate suitability of proposed platoon approach, we use actual measurement data taken in highways of Sweden shown in Fig. 2.2 and Fig. 2.3.

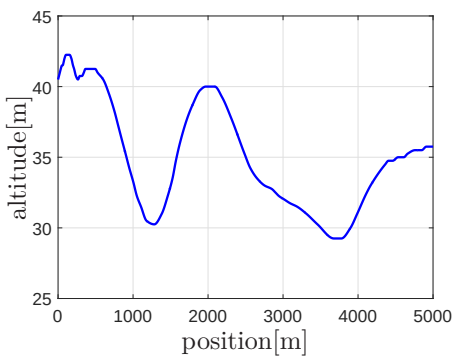

Figure 2.2: Actual altitude data taken in highways of Sweden

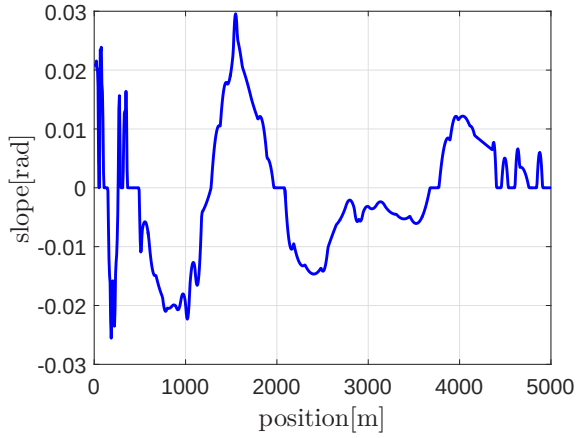

Figure 2.3: Actual slope data taken in highways of Sweden

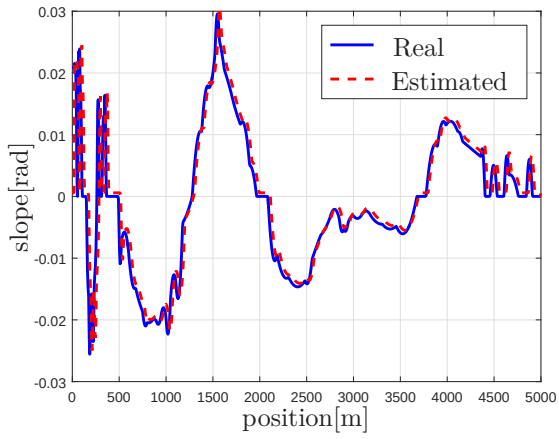

Figure 2.4: Estimated slope data taken in preceding platoon

2.3 Vehicle and Platoon model

In this section, we define single vehicle model and platoon model utilized in the two control layers. By Newton's second law, the longitudinal dynamics of single vehicle model are defined as

$$\begin{aligned} m_i \dot{v}_i &= F_{\text{engine},i} + F_{\text{brake},i} + F_{\text{ext},i} \\ v_i &= \dot{s}_i \end{aligned} \tag{2.1}$$

where m_i, v_i, s_i denote mass, speed, and longitudinal position of vehicle i, respectively, and $F_{\text{engine},i}$, $F_{\text{brake},i}$, $F_{\text{ext},i}$ are forces generated by engine system and braking system, and external force. More specifically, we assume that exogenous disturbance consists of three kinds of forces and is defined as

$$F_{\text{ext},i} = F_{\text{gravity},i}(\alpha(s_i)) + F_{\text{roll},i}(\alpha(s_i)) + F_{\text{drag},i}(v_i, d_i) \tag{2.2}$$

where $F_{\text{gravity},i}(\alpha(s_i))$, $F_{\text{roll},i}(\alpha(s_i))$, $F_{\text{drag},i}(v_i, d_i)$ are force occurring due to gravity, rolling resistance, and aerodynamic force, respectively. See Fig.2.5. We assume that engine

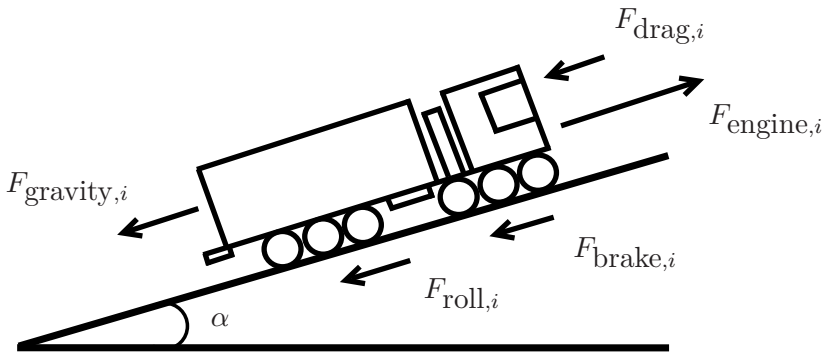

Figure 2.5: Forces acting on vehicle i.

force of vehicle i has specific saturation region represented as

$$\frac{P_{\text{min},i}}{v_i} \leq F_{\text{engine},i} \leq \frac{P_{\text{max},i}}{v_i} \tag{2.3}$$

where $P_{\text{min},i}$, $P_{\text{max},i}$ are minimum and maximum engine power of vehicle i, respectively. We assume that braking force of vehicle i is bounded by road friction and is modeled as

$$-m_i \eta_i g \mu_i \leq F_{\text{brake},i} \leq 0 \tag{2.4}$$

where μ_i, η_i are road friction coefficient and braking system efficiency, respectively. We assume that $F_{\text{gravity},i}(\alpha(s_i))$ is modeled as

$$F_{\text{gravity},i}(\alpha(s_i)) = -m_i g \sin(\alpha(s_i)) \tag{2.5}$$

where $\alpha(s_i)$ denotes road slope at position s_i and g denotes gravity acceleration. We assume that $F_{\text{roll},i}(\alpha(s_i))$ is modeled as

$$F_{\text{roll},i}(\alpha(s_i)) = -c_r m_i g \cos(\alpha(s_i)) \tag{2.6}$$

where c_r denotes rolling coefficient. We assume that $F_{\text{drag},i}$ is modeled as

$$F_{\text{drag},i}(v_i, d_i) = -\frac{1}{2}\rho A_v C_D(d_i) v_i^2 \tag{2.7}$$

where ρ, A_v are air density and cross-sectional area, respectively, and $C_D(d_i)$ is air drag coefficient defined as a function of distance between current vehicle and its preceding vehicle and is assumed to be modeled as

$$C_D(d_i) = C_{D0}(1 - \frac{C_{D1}}{C_{D2} + d_i}) \tag{2.8}$$

where parameter C_{D0} denotes coefficient determined by shape of vehicle i and also, C_{D1}, C_{D2} are coefficients obtained by regressing the experimental data and d_i denotes distance defined as

$$d_i = \begin{cases} \infty, & i = 1 \\ s_{i-1} - s_i - l_i, & 2 \leq i \leq N_v \end{cases} \tag{2.9}$$

where l_i denotes length of vehicle i. From (2.7) and (2.8), we get to know the fact that short distance between vehicles means low aerodynamic resistance and fuel efficiency is consequently enhanced. Simplified fuel model used in platoon coordinator layer is defined as

$$\delta_i = p_{1,i} F_{\text{engine},i} v_i + p_{0,i} \tag{2.10}$$

where δ_i denotes fuel flow and $p_{1,i}$, $p_{0,i}$ are coefficients related to fuel consumption rate. We assume that gear changes are not taken into account for simplicity.

2.4 Vehicle Controller

Vehicle controller architecture is consist of reference generator, and DOB based controller with braking system for safe platoon and shown in Fig. 2.6. Based on velocity profile obtained from platoon coordinator $v_i^*(z_k)$ and velocity data over position of preceding vehicle $v_{i-1}(z_k)$, reference generator makes new velocity reference $v_{\text{ref},i}$ which vehicle should track.

2.4.1 Reference Generator

This section defines new velocity reference $v_{\text{ref},i}(z_k)$ in order to compensate vehicle deviation in platoon. First vehicle follows optimal velocity profile which platoon coordinator

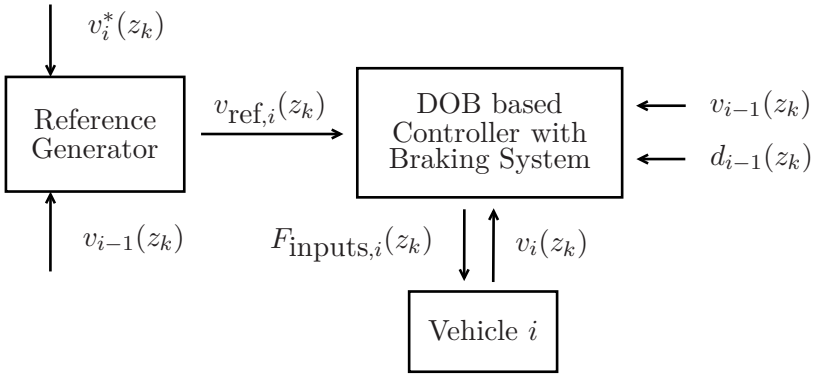

Figure 2.6: Vehicle controller architecture.

produces and following vehicles track mixed velocity reference utilizing velocity of leading vehicle and optimal velocity:

$$\begin{aligned} v_{\mathrm{ref},1}(z_k) &= v_1^*(z_k) \\ v_{\mathrm{ref},2}(z_k) &= K_2 v_2^*(z_k) + (1-K_2)v_1(z_k) \\ &\vdots \\ v_{\mathrm{ref},N_v}(z_k) &= K_{N_v} v_{N_v}^*(z_k) + (1-K_{N_v})v_{N_v-1}(z_k) \end{aligned} \tag{2.11}$$

where z_k is discretized space variable, and $K_2, \cdots, K_{N_v}$ are arbitrary positive weighting parameters less than 1.

2.4.2 DOB based Controller with Braking System

Consider a block diagram shown in Fig. 2.7. Vehicle controller uses disturbance observer (DOB) as robust control tool. Signals u_i, n, $\widetilde{F}_{\mathrm{ext}}$, F_{est} are, respectively, control signal, measurement noise, unknown external disturbance, estimated disturbance. Here, actual measurement noise reported in [40] is induced into control system.

Transfer function of uncertain vehicle model is defined as

$$\widetilde{P}_i(z) = \frac{T_s}{\widetilde{m}_i(z-1)} \tag{2.12}$$

where $\widetilde{m}_i$, T_s are uncertain mass of vehicle i and sampling time, respectively. Transfer function of nominal vehicle model is modeled as

$$P_i(z) = \frac{T_s}{m_i(z-1)} \tag{2.13}$$

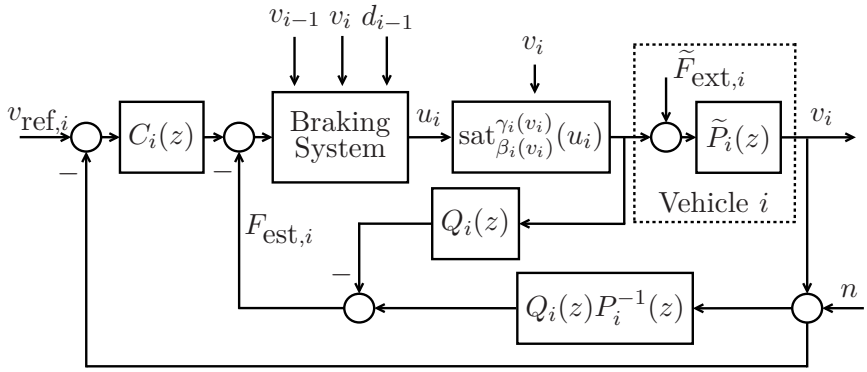

Figure 2.7: Block diagram of DOB based controller with braking system.

Transfer function $C_i(z)$ is outer-loop controller. Transfer function of Q-filter is designed as

$$Q_i(z) = \frac{c_1(z-1) + c_0}{(z-1)^2 + a_1(z-1) + a_0} \tag{2.14}$$

where $a_0 = c_0$, a_1, c_1 are arbitrary positive coefficients. For more information on discrete version of DOB, see [39]. The saturating actuator is denoted by $\text{sat}_{\beta_i}^{\gamma_i}(u_i)$ with $\beta_i(v_i)$ and $\gamma_i(v_i)$ being the lower and upper limits:

$$\text{sat}_{\beta_i(v_i)}^{\gamma_i(v_i)}(u_i) = \begin{cases} \beta_i(v_i), & u_i < \beta_i(v_i) \\ u_i, & \beta_i(v_i) \leq u_i \leq \gamma_i(v_i) \\ \gamma_i(v_i), & u_i > \gamma_i(v_i) \end{cases} \tag{2.15}$$

where $\beta_i(v_i)$, $\gamma_i(v_i)$ are defined as

$$\begin{aligned} \beta_i(v_i) &= \frac{P_{\text{min},i}}{v_i} - m_i \eta_i g \mu_i \\ \gamma_i(v_i) &= \frac{P_{\text{max},i}}{v_i} \end{aligned} \tag{2.16}$$

Unknown external disturbance is represented as

$$\begin{aligned} \widetilde{F}_{\text{ext},i} &= \widetilde{F}_{\text{gravity},i}(\alpha(s_k)) + \widetilde{F}_{\text{roll},i}(\alpha(s_k)) + \widetilde{F}_{\text{drag},i}(v_i, d_i) \\ &= -\widetilde{m}_i g \sin(\alpha(s_k)) - \widetilde{c}_{r,i} \widetilde{m}_i g \cos(\alpha(s_k)) - \frac{1}{2} \rho A_v C_D(d_i) v_i^2 \end{aligned} \tag{2.17}$$

where $\widetilde{c}_{r,i}$ is unknown rolling coefficient.

Before explaining safety constraints, we assume that vehicle speed, vehicle mass, slope, friction coefficient, rolling coefficient are bounded as follows: $0 \leq v_i \leq v_{\text{max}}$, $m_{\text{min}} \leq \widetilde{m}_i \leq m_{\text{max}}$, $|\widetilde{\alpha}_i| \leq \alpha_{\text{max}}$, $\mu_{\text{min}} \leq \widetilde{\mu}_i \leq \mu_{\text{max}}$, $c_{r,\text{min}} \leq \widetilde{c}_{r,i} \leq c_{r,\text{max}}$.

In the case that speed and position trajectories deviates significantly from the reference ones, vehicle may require safety braking system to avoid severe collision. If vehicle violates safety law, braking system of vehicle operates. The safety law is defined as follows:

$$d_i \geq d_{\text{safe},i} \tag{2.18}$$

where $d_{\text{safe},i}$ is defined as

$$\begin{aligned} d_{\text{safe},i} &= \frac{v_{i-1}^2}{2\underline{a}_{\text{min},i-1}} - \frac{v_i^2}{2\overline{a}_{\text{min},i}} \\ \underline{a}_{\text{min},i} &= -\mu_{\text{max}}\eta_{\text{max}}g - g\sin(\alpha_{\text{max}}) - c_{r,\text{max}}g - \frac{1}{2m_{\text{min}}}\rho A_v C_{D0} v_{\text{max}}^2 \\ \overline{a}_{\text{min},i} &= -\mu_{\text{min}}\eta_{\text{min}}g - c_{r,\text{min}}g \end{aligned} \tag{2.19}$$

To get detailed information about safety law, see [36, 37].

2.5 Platoon Coordinator

We use platoon coordinator algorithm which is introduced in [36,37]. Platoon coordinator is the higher layer of platoon control architecture and takes average speed requirement from mission planner and current vehicle state from vehicles. By using the available information on the planned route, it generates a unique optimal speed profile v_i^* which minimizes the fuel consumption of the whole platoon, while maintaining a certain average speed. Here, platoon coordinator layer uses a DP framework to compute optimal velocity profile. Parameters which characterize DP problem are discretization space Δs_{DP}, horizon length H_{DP}, and refresh frequency f_{DP}. Horizon space length is defined as $S_{DP} = H_{DP}\Delta s_{DP}$. In space domain, platoon coordinator layer uses a discretized version of (2.1) and is represented as

$$\begin{aligned} m_i v_i(z_k)\frac{v_i(z_k) - v_i(z_{k-1})}{\triangle s_{\text{DP}}} &= F_{\text{engine},i}(z_k) + F_{\text{brake},i}(z_k) - m_i g\sin(\alpha(z_k)) \\ &\quad - c_r m_i g\cos(\alpha(z_k)) - \frac{1}{2}\rho A_v C_D(d_i(z_k))(v_i(z_k))^2 \\ v_i(z_k)\frac{t_i(z_k) - t_i(z_{k-1})}{\triangle s_{\text{DP}}} &= 1 \end{aligned} \tag{2.20}$$

where parameters z_k, $v_i(z_k)$ denote discretized space and speed and forces $F_{\text{engine},i}(z_k)$, $F_{\text{brake},i}(z_k)$ are engine force and braking force expressed as function of discretized space,

respectively. In DP formulation, we refer to (2.20) as $v_i(z_{k-1}) = f_{v,i}(v_i(z_k), u_i(z_k))$ where $u_i(z_k)$ is the input vector defined as $u_i(z_k) = [F_{e,i}(z_k), F_{b,i}(z_k)]^T$.

Platoon model is bounded using input constraints and state constraints. According to (2.3) and (2.4), engine force and braking force are bounded by

$$\begin{aligned} \frac{P_{\text{min},i}}{v_i(z_k)} \leq F_{\text{engine},i}(z_k) \leq \frac{P_{\text{max},i}}{v_i(z_k)} \\ -m_i \eta_i g \mu_i \leq F_{\text{brake},i}(z_k) \leq 0 \end{aligned} \tag{2.21}$$

In the DP formulation, we refer to these constraints as $u_i(z_k) \in U_i(z_k)$. Considering the road speed limit obtained from road data base, vehicle speed is bounded by

$$v_{\text{min}}(z_k) \leq v_i(z_k) \leq v_{\text{max}}(z_k) \tag{2.22}$$

We refer to this constraint as $v_i(z_k) \in V(z_k)$. All the vehicles are required to track the same speed profile at same position and it is represented as

$$v_1(z_k) = v_2(z_k) = \ldots = v_{N_v}(z_k) = v(z_k) \tag{2.23}$$

This constraint reduces complex computation by reducing the search space of DP algorithm to one dimension. The cost function of platoon coordinator layer is divided into two term which are consist of J_{fuel} as overall fuel amount consumed by vehicles in platoon and J_{travel} defined as travel time over space.

$$J_{\text{DP}} = J_{\text{fuel}} + \beta J_{\text{travel}} \tag{2.24}$$

where parameter β represents weighting factor. The fuel term uses both (2.10) and kinematic energy of platoon at the end of horizon and the travel term uses (2.20).

$$\begin{aligned} J_{\text{fuel}} = & \sum_{i=1}^{N_v} \sum_{j=k}^{k+H_{DP}-1} \Delta s_{\text{DP}} (p_{1,i} F_{\text{engine},i} + \frac{p_{0,i}}{v(z_k)}) \\ & - \sum_{i=1}^{N_v} p_{1,i} \frac{m_i (v(z_h + H_{\text{DP}} - 1))^2}{2} \\ J_{\text{travel}} = & \sum_{j=k}^{k+H_{DP}-1} \frac{\Delta s_{\text{DP}}}{v(z_j)} \end{aligned} \tag{2.25}$$

The DP formulation is defined as

$$\begin{aligned} &\min_{u_i(z_j)} J_{\mathrm{DP}} \\ \text{subj. to } &v_i(z_{j-1}) = g_i(v_i(z_j), u_i(z_j)) \\ &u_i(z_j) \in U_i(z_j) \\ &v_i(z_j) = v(z_j) \in V(z_j) \\ &z_k = s_1(t) \\ &v(z_k) = v_1(t) \end{aligned} \tag{2.26}$$

for $j = k, ..., k + H_{DP} - 1$. For more information about platoon coordinator, see [36, 37].

2.6 Simulation

Various simulations are conducted to validate the effectiveness of the proposed vehicle controller and use real measurement road data. We assume that 3 vehicles are gathered for platoon. Especially, we focus on scenario where all vehicles in platoon have different masses and facts that this proposed approach does not need actual road data. (As we mentioned previous chapter, platoon coordinator must need actual road data for producing optimal velocity reference.) It means that platooning can not avoid usage of expensive road data. This may be much large disadvantage in the aspect of fee. At first, for validation of our idea, we will define parameters as follow:

$$\begin{aligned} &m_1 = m_2 = m_3 = 40\text{t}, \\ &\widetilde{m}_1 = 40\text{t}, \widetilde{m}_2 = 36\text{t}, \widetilde{m}_3 = 44\text{t}, \\ &c_{r,1} = c_{r,2} = c_{r,3} = 0.003, \\ &\widetilde{c}_{r,1} = 0.028, \widetilde{c}_{r,2} = 0.003, \widetilde{c}_{r,3} = 0.0032, \\ &\mu_1 = \mu_2 = \mu_3 = 0.8, \\ &\widetilde{\mu}_1 = 0.77, \widetilde{\mu}_2 = 0.78, \widetilde{\mu}_3 = 0.81, \end{aligned} \tag{2.27}$$

The changes of mass and several disturbances makes vehicles not to keep platoon. This means that all vehicles are affected by different size of disturbances. For compensation of this effect, we presents new vehicle controller based on disturbance observer which compensates exogenous disturbance and makes real vehicle behave like nominal vehicle in steady state. Various plots are obtained from MATLAB/SIMULINK. See Fig. 2.8, Fig. 2.9, Fig. 2.10. We can check that real vehicles are affected by several disturbances.

Especially, we can check that gravity force by changes of slope has biggest values in those graphs. We also show exogenous disturbance and estimated disturbance. They are very similar and this means that controller compensates effect of disturbances.

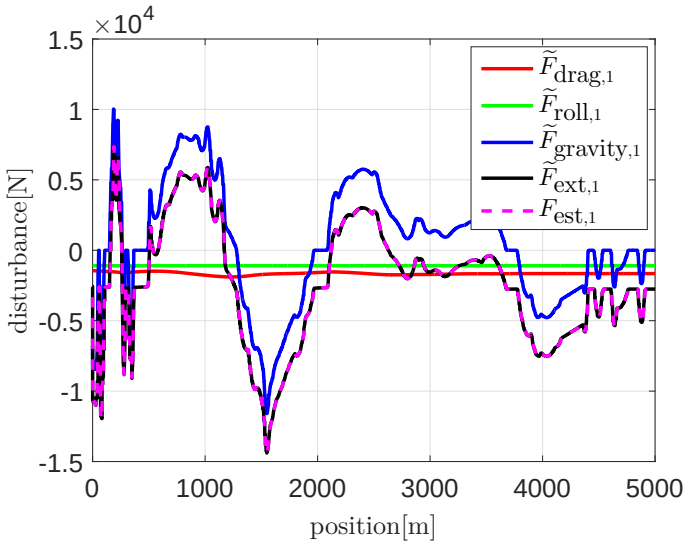

Figure 2.8: Exogenous disturbance and estimated disturbance of vehicle 1

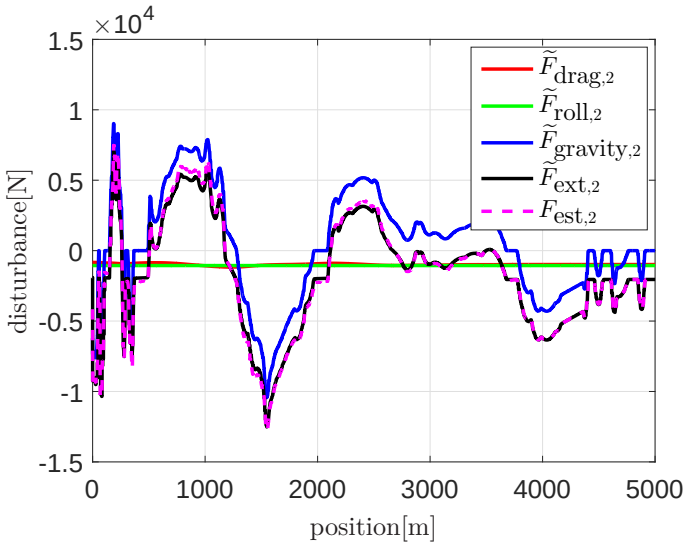

Figure 2.9: Exogenous disturbance and estimated disturbance of vehicle 2

Fig. 2.11 shows velocities of vehicles over position. Each vehicle in platoon tracks

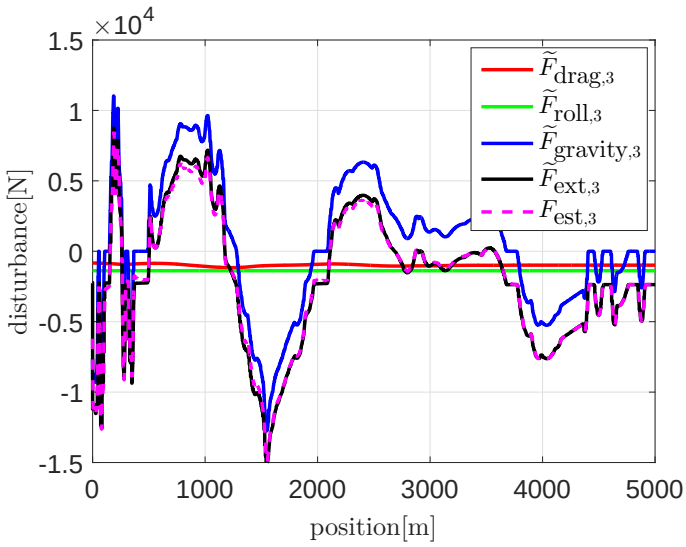

Figure 2.10: Exogenous disturbance and estimated disturbance of vehicle 3

reference produced by reference generator and also vehicles has almost same speed.

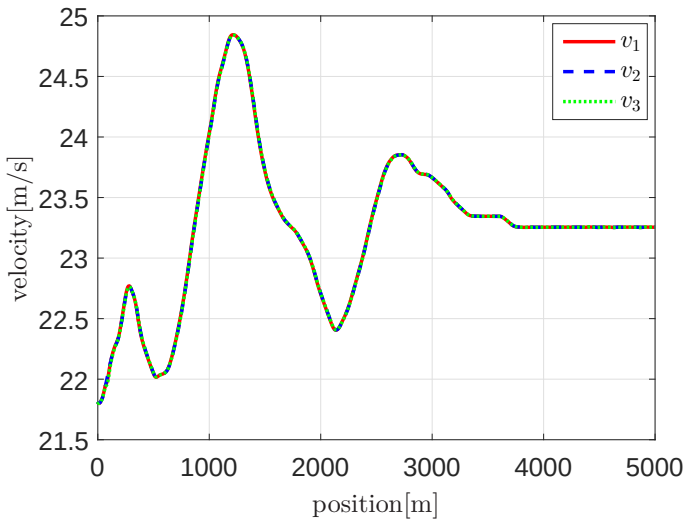

Figure 2.11: Velocities of vehicles

The distance between vehicles and safe distance are plotted in Fig. 2.12. If distance between vehicles violates safe distance, braking systems will be operated until they satisfy gap over safe distance again.

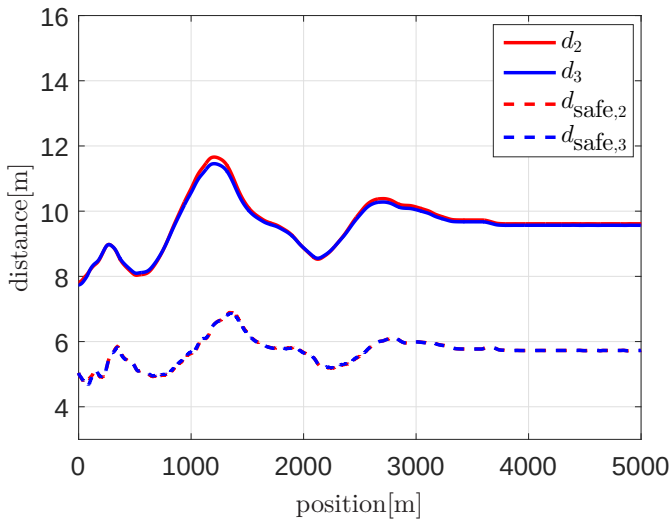

Figure 2.12: Distance over position

See Fig. 2.13 and this shows aerodynamic drag of 3 vehicles in platoon. We can check that second and third vehicles are affected by small aerodynamic drag rather than first vehicle, although first vehicle takes largest aerodynamics force of 3 vehicles. Here, we can know effect of platoon.

We show inputs of vehicles. They have different size of input at the same position because of different mass. Also, we can check that all input are bounded by upper limits.

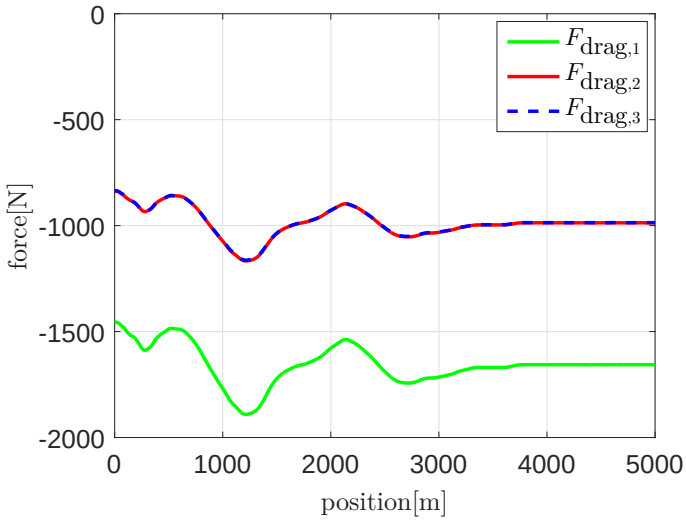

Figure 2.13: Aerodynamic drag of 3 vehicles in platoon

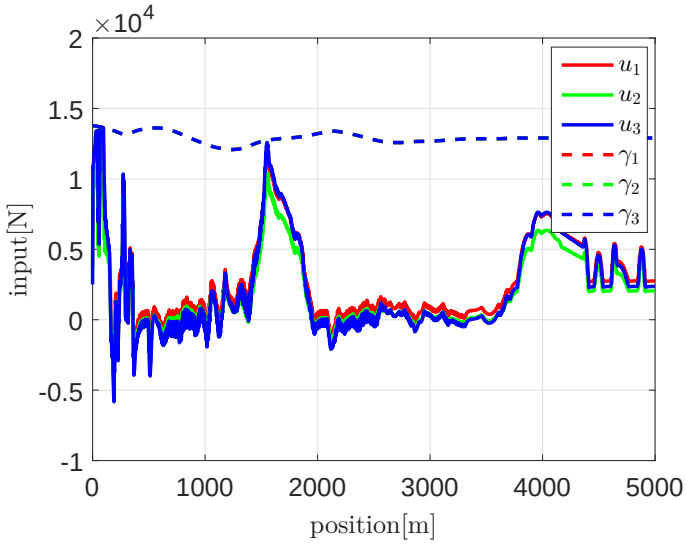

Figure 2.14: Inputs of vehicles

Fig. 2.15 shows fuel consumption of vehicles. Red bar means vehicles in the roadway operated without road data, it means vehicle does not use platoon coordinator. So Red bar is highest in 3 bars. Green bar means vehicles in the roadway use platoon coordinator. This approach is best method, but vehicles need expensive road data. However, in the case of using Disturbance Observer, vehicles use estimated road data, and then, vehicles reduce fuel consumption.

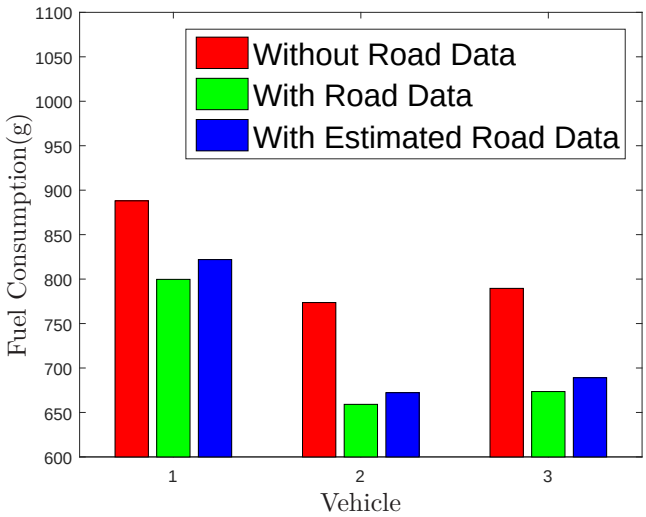

Figure 2.15: Fuel consumption of vehicles

Through simulations, we can know that proposed vehicle controller are quite useful.

2.7 Conclusions

By keeping the distance between vehicles short enough to decrease aerodynamic drag, heavy-duty vehicle platooning can reduce fuel consumption. Major disturbances in the platooning are the slope in the road and uncertain mass of the trucks. Existing method to reduce the effect of slope is to combine the vehicle position obtained from GPS and the slope database to calculate the amount of feed-forward type compensation. However, slope database is costly to acquire, and GPS signal is unreliable in some locations. We present vehicle controller based on disturbance observer to compensate the effect of slope and mass without relying on the GPS and the slope database. Actual measurement of the road slope taken in highways of Sweden was used and simulation was conducted.

2.8 MATLAB CODE

I have saved platoon code in the common fold. To understand code information more perfectly, I recommend you to check files in common fold.

2.8.1 Initialization

```

clear all
clc

%% Load files
load road_profile_Mariefred_Eskilstuna.mat;
%load Mariefred_Eskilstuna.mat; % sampled road data
load powertrain_model;

%slope = SLOPE;
%distance = DISTANCE;
%altitude = ALTITUDE;

%% Vehicle Operation Time(s)
OperationTime = 300;

%% Platoon coordinator
n_vehicles                      = 3;
nominal_mass                    = 40000;   %% Nominal mass of the vehicles(kg)
nominal_length                  = 18;
nominal_cr                      = 0.003;
nominal_Pmax                    = 300000;
nominal_Pmin                    = -8994.5;
nominal_engine_fuel_coeff       = [0; fuelp_coeff1(1); 0];
nominal_mass_vec                = nominal_mass*ones(1,n_vehicles);
nominal_length_vec              = nominal_length*ones(1,n_vehicles);
nominal_cr_vec                  = nominal_cr*ones(1,n_vehicles);
nominal_Pmin_vec                = nominal_Pmin*ones(1,n_vehicles);
nominal_Pmax_vec                = nominal_Pmax*ones(1,n_vehicles);
nominal_engine_fuel_coeff_vec   = repmat(nominal_engine_fuel_coeff,1,n_vehicles);

open_loop_pc            = 1;
% 1 to use the platoon coordinator in openloop, note:
```

```
% fix the variable s_pred_DP inside the platoon coordinator block
if open_loop_pc
    dt_pc               = 40000;
    % [s] refresh time of the platoon coordinator block, large value for using it in open loop
    s_pred_DP           = 8000;
    % [m] length of the prediction horizon
else
    dt_pc               = 4;
    s_pred_DP           = 4000;
end

CLAC                    = 1;
% 1 for clac, 0 for lac (optimization consider the ful consumption of only the first vehicle)
ave_v_ref               = 22.00;
max_v_ref               = 25;
% [m/s] max speed
max_err_v               = 0.02;
fixed_end_v             = 0;
% 1 for fixing the speed at the end of the horizon of the dynamic programming
end_v                   = 22;
% final speed at the end of the horizon of the dynamic programming  (if fixed_end_v = 1)
beta_tuning             = 0;
% 1 for the in block tuning of beta in order to obtain the reference average speed
k_beta                  = 0.01;
s_discre_DP             = 6;
hp_DP                   = floor(s_pred_DP/s_discre_DP);
% number of steps of the prediction horizon

% setting beta_init, i.e, the initial value for the trade-off parameter between fuel and time
if n_vehicles == 2
    if CLAC
        beta_init = 7.3873e-3;
    else
        beta_init = 4.2878e-3;
    end
elseif  n_vehicles == 3
    if CLAC
        beta_init = 10.3056e-3;
    else
        beta_init = 7.3873e-3;
    end
elseif  n_vehicles == 4
    if CLAC
        beta_init = 13.9990e-3;
    else
        beta_init = 7.3873e-3;
    end
end

InitialPosition = -300; % Initial position of the vehicle(m)
Vehicle1Length = 18; % Length of the vehicle(m)
Vehicle2Length = 18;
Vehicle3Length = 18;
InitialDistanceBetweenVehicles12 = 8; % Initial distance between vehicles(m)
InitialDistanceBetweenVehicles23 = 8;
time_gap(2)             = (InitialDistanceBetweenVehicles12+Vehicle1Length)/22;
time_gap(3)             = (InitialDistanceBetweenVehicles23+Vehicle2Length)/22;
InitialPosition1        = InitialPosition+InitialDistanceBetweenVehicles12
+Vehicle1Length+InitialDistanceBetweenVehicles23+Vehicle2Length;
v_init                  = 22;
Xref_pc_init            = [InitialPosition1+6*(-(hp_DP+40-1):0)' , v_init*ones(hp_DP+40,1)];

road_s_vec              = (0:6:distance(end))';
road_h_vec              = interp1(distance,altitude,road_s_vec,'pchip');
road_vmax_vec           = max_v_ref*ones(size(road_s_vec));

road_info               = [road_s_vec road_h_vec road_vmax_vec];
vehicles_info           = [nominal_mass_vec;
```

```
                            nominal_length_vec;
                            nominal_cr_vec];
engine_info              = [nominal_Pmin_vec;
                            nominal_Pmax_vec;
                            nominal_engine_fuel_coeff_vec];

%% Vehicle controller & Plant(Vehicles)
T = 0.05; % Sampling Time

dBm = -120;
Noisepower = 10.^((dBm-30)/10); % Band-limited white noise

g = 9.81; % Gravity acceleration

nominal_Eb = 1;
Vehicle1Eb = 1; % Braking efficiency of vehicle 1
Vehicle2Eb = 0.98; % Braking efficiency of vehicle 2
Vehicle3Eb = 0.99;

nominal_Cf = 0.8;
Vehicle1Cf = 0.77; % Friction coefficient of vehicle 1
Vehicle2Cf = 0.78; % Friction coefficient of vehicle 2
Vehicle3Cf = 0.81;

Vehicle1Cr = 0.0028; % Rolling resistance of vehicle 1
Vehicle2Cr = 0.0030; % Rolling resistance of vehicle 2
Vehicle3Cr = 0.0032;

Cd0 = 0.5271; % Coefficient Cd0 of aerodynamic drag
Ad = 1.225; % Air density
Vehicle1Av = 9.487; % Av of vehicle 1
Vehicle2Av = 9.487; % Av of vehicle 2
Vehicle3Av = 9.487;

Vehicle1Pmax = 300000; % Pmax, Pmin
Vehicle1Pmin = -8994.5;
Vehicle2Pmax = 300000;
Vehicle2Pmin = -8994.5;
Vehicle3Pmax = 300000;
Vehicle3Pmin = -8994.5;

P = 40000; % Outer controller gain of PID
I = 0;
D = 0;

DOB_switch = 1; % Discrete-time DOB switch
DOB_type = 0; % Control gain of discrete-time DOB

if DOB_type == 0
    Qn_z = 1;
    Qd_z = [1 0];
elseif DOB_type == 1
    Qn_z = [2 -1];
    Qd_z = [1 0 0];
elseif DOB_type == 2
    Qn_z = [3 -3 1];
    Qd_z = [1 0 0 0];
end

Pnn_z = T;
Pnd_z = nominal_mass*[1 -1];

QPnn_z = conv(Pnd_z,Qn_z);
QPnd_z = conv(Pnn_z,Qd_z);

M1 = 40000; % Real mass of the vehicle1(kg)
M2 = 36000; % Real mass of the vehicle2(kg)
M3 = 44000;
```

```
%M1 = 40000; % Real mass of the vehicle1(kg)
%M2 = 40000; % Real mass of the vehicle2(kg)
%M3 = 40000;

Ac1 = 0; % Plant(vehicle1,vehicle2, vehicle3), state space form
Bc1 = 1;
Cc1 = 1/M1;
Dc1 = 0;

Ac2 = 0;
Bc2 = 1;
Cc2 = 1/M2;
Dc2 = 0;

Ac3 = 0;
Bc3 = 1;
Cc3 = 1/M3;
Dc3 = 0;

InitialVehicle1Velocity = M1*22;
% Plant(vehicle1,vehicle2, vehicle3), Initial value of state space form
InitialVehicle2Velocity = M2*22;
InitialVehicle3Velocity = M3*22;

%% vRef parameter
vRefParameter = 0.9;

%% saturation-region
sat_region_on = 0;
% (sat_region_on = 1 for sat_min = 0) , (sat_region_on = 0 for sat_min = sat_min)

%% The range of vehicle parameters(v,m,slope,distance,eb,cr)
%These are necessary parameters to set the braking safe system

Vehicle_Eb_max = 1;
Vehicle_Eb_min = 0.97;
Vehicle_v_max = max_v_ref;
Vehicle_v_min = 0;
Vehicle_m_max = 45000;
Vehicle_m_min = 35000;
Vehicle_Cf_max = 0.83;
Vehicle_Cf_min = 0.77;
Vehicle_Cr_max = 0.0032;
Vehicle_Cr_min = 0.0028;
if abs(max(slope)) >= abs(min(slope))
    Vehicle_slope_max = abs(max(slope));
    Vehicle_slope_min = 0;
else
    Vehicle_slope_max = abs(min(slope));
    Vehicle_slope_min = 0;
end
Av = 9.487;
Vehicle_Aerodynamic_max = 1/(2*Vehicle_m_min)*Ad*Av*Cd0*(1)*(Vehicle_v_max^2);
Vehicle_Aerodynamic_min = 1/(2*Vehicle_m_max)*Ad*Av*Cd0*(1-14.666./26.666)*(Vehicle_v_min^2);

soft_braking_coeff = 1;
% Min_MinimumAcceleration
min_min_a = -Vehicle_Cf_max*Vehicle_Eb_max*g - g*sin(Vehicle_slope_max)
- Vehicle_Cr_max*g*cos(Vehicle_slope_min) - Vehicle_Aerodynamic_max;
% Max_MinimumAcceleration
max_min_a = -Vehicle_Cf_min*Vehicle_Eb_min*g - g*sin(Vehicle_slope_min)
- Vehicle_Cr_min*g*cos(Vehicle_slope_max) - Vehicle_Aerodynamic_min;

%% Run simulation
sim('platooningDOB');
%sim('platooningDOB_rev1');

%% Plot
```

```
minxlimit = 0; % xlimit of graph
maxxlimit = 5000;

% Slope graph from Mariefred to Eskilstuna
figure(1);
plot(distance,slope,'b','linewidth',2);
set(gca,'fontsize',15);
xlim([minxlimit,maxxlimit]);
xlabel('$position[m]$','fontsize',15);
ylabel('$slope$','fontsize',15);
grid on;

% Altitude graph from Mariefred to Eskilstuna
figure(2);
plot(distance,altitude,'b','linewidth',2);
set(gca,'fontsize',15);
xlim([minxlimit,maxxlimit]);
xlabel('$position[m]$','fontsize',15);
ylabel('$altitude[m]$','fontsize',15);
grid on;

% Velocity plot
figure(3);
plot(ScopeData.data(:,1),ScopeData.data(:,4),'r','linewidth',2); hold on;
plot(ScopeData.data(:,2),ScopeData.data(:,5),'b--','linewidth',2); hold on;
plot(ScopeData.data(:,3),ScopeData.data(:,6),'g:','linewidth',2); hold off;
set(gca,'fontsize',15);
xlim([minxlimit,maxxlimit]);
ylim([21.5,25]);
xlabel('$position[m]$','fontsize',15);
ylabel('$velocity$','fontsize',15);
legend('$v_{1}$','$v_{2}$','$v_{3}$');
grid on;

% Distance between vehicles graph
figure(4)
plot(ScopeData.data(:,2),ScopeData.data(:,8),'r','linewidth',2); hold on;
plot(ScopeData.data(:,3),ScopeData.data(:,9),'b','linewidth',2); hold on;
plot(ScopeData.data(:,2),ScopeData.data(:,10),'r--','linewidth',2); hold on;
plot(ScopeData.data(:,3),ScopeData.data(:,11),'b--','linewidth',2); hold off;
set(gca,'fontsize',15);
xlim([minxlimit,maxxlimit]);
ylim([3,16]);
xlabel('$position[m]$','fontsize',15);
ylabel('$distance[m]$','fontsize',15);
legend('$d_{2}$','$d_{3}$','$d_{\mbox{safe},2}$','$d_{\mbox{safe},3}$');
grid on;

% u1, u2, u3
figure(5);
plot(ScopeData.data(:,1),ScopeData.data(:,18),'r','linewidth',2); hold on;
plot(ScopeData.data(:,2),ScopeData.data(:,19),'g','linewidth',2); hold on;
plot(ScopeData.data(:,3),ScopeData.data(:,20),'b','linewidth',2); hold on;
plot(ScopeData.data(:,1),ScopeData.data(:,21),'r--','linewidth',2); hold on;
plot(ScopeData.data(:,2),ScopeData.data(:,22),'g--','linewidth',2); hold on;
plot(ScopeData.data(:,3),ScopeData.data(:,23),'b--','linewidth',2); hold off;
set(gca,'fontsize',15);
xlim([minxlimit,maxxlimit]);
ylim([-10000,20000]);
xlabel('$position[m]$','fontsize',15);
```

```
ylabel('$input[N]$','fontsize',15);
legend('$u_{1}$','$u_{2}$','$u_{3}$','$\gamma_{1}$','$\gamma_{2}$','$\gamma_{3}$');
grid on;

% Airdrag Force
figure(6);
plot(ScopeData.data(:,1),ScopeData.data(:,24),'g','linewidth',2); hold on;
plot(ScopeData.data(:,2),ScopeData.data(:,25),'r','linewidth',2); hold on;
plot(ScopeData.data(:,3),ScopeData.data(:,26),'b--','linewidth',2); hold off;
set(gca,'fontsize',15);
xlim([minxlimit,maxxlimit]);
ylim([-2000,0]);
xlabel('$position[m]$','fontsize',15);
ylabel('$force[N]$','fontsize',15);
legend('${F}_{\mbox{drag},1}$','${F}_{\mbox{drag},2}$','${F}_{\mbox{drag},3}$');
grid on;

figure(7);
plot(ScopeData.data(:,1),ScopeData.data(:,24),'r','linewidth',2); hold on;
plot(ScopeData.data(:,1),ScopeData.data(:,27),'g','linewidth',2); hold on;
plot(ScopeData.data(:,1),ScopeData.data(:,30),'b','linewidth',2); hold on;
plot(ScopeData.data(:,1),ScopeData.data(:,12),'k','linewidth',2); hold on;
plot(ScopeData.data(:,1),ScopeData.data(:,15),'m--','linewidth',2); hold off;
set(gca,'fontsize',15);
xlim([minxlimit,maxxlimit]);
ylim([-15000,15000]);
xlabel('$position[m]$','fontsize',15);
ylabel('$disturbance[N]$','fontsize',15);
legend('$\widetilde{F}_{\mbox{drag},1}$','$\widetilde{F}_{\mbox{roll},1}$'
,'$\widetilde{F}_{\mbox{gravity},1}$','$\widetilde{F}_{\mbox{ext},1}$','${F}_{\mbox{est},1}$');
grid on;

figure(8);
plot(ScopeData.data(:,2),ScopeData.data(:,25),'r','linewidth',2); hold on;
plot(ScopeData.data(:,2),ScopeData.data(:,28),'g','linewidth',2); hold on;
plot(ScopeData.data(:,2),ScopeData.data(:,31),'b','linewidth',2); hold on;
plot(ScopeData.data(:,2),ScopeData.data(:,13),'k','linewidth',2); hold on;
plot(ScopeData.data(:,2),ScopeData.data(:,16),'m--','linewidth',2); hold off;
set(gca,'fontsize',15);
xlim([minxlimit,maxxlimit]);
ylim([-15000,15000]);
xlabel('$position[m]$','fontsize',15);
ylabel('$disturbance[N]$','fontsize',15);
legend('$\widetilde{F}_{\mbox{drag},2}$','$\widetilde{F}_{\mbox{roll},2}$'
,'$\widetilde{F}_{\mbox{gravity},2}$','$\widetilde{F}_{\mbox{ext},2}$','${F}_{\mbox{est},2}$');
grid on;

figure(9);
plot(ScopeData.data(:,3),ScopeData.data(:,26),'r','linewidth',2); hold on;
plot(ScopeData.data(:,3),ScopeData.data(:,29),'g','linewidth',2); hold on;
plot(ScopeData.data(:,3),ScopeData.data(:,32),'b','linewidth',2); hold on;
plot(ScopeData.data(:,3),ScopeData.data(:,14),'k','linewidth',2); hold on;
plot(ScopeData.data(:,3),ScopeData.data(:,17),'m--','linewidth',2); hold off;
set(gca,'fontsize',15);
xlim([minxlimit,maxxlimit]);
ylim([-15000,15000]);
xlabel('$position[m]$','fontsize',15);
ylabel('$disturbance[N]$','fontsize',15);
legend('$\widetilde{F}_{\mbox{drag},3}$','$\widetilde{F}_{\mbox{roll},3}$'
,'$\widetilde{F}_{\mbox{gravity},3}$','$\widetilde{F}_{\mbox{ext},3}$','${F}_{\mbox{est},3}$');
```

```
grid on;
```

2.8.2 Platoon Coordinator

```
% assumtion: trucks have the same powertrain!

function Xref_pc   = platoon_coordinator(road_info, vehicles_info, engine_info,
k_beta, beta_tuning, max_v_ref, ave_v_ref, time_gap, CLAC,...
fixed_end_v, end_v, max_err_v, X1, beta_init, Xref_pc_init)

persistent beta Xref_pc_per
t_curr=0;
if isempty(beta)
    beta=beta_init;
end
if isempty(Xref_pc_per)
    Xref_pc_per=Xref_pc_init;
end

coder.extrinsic('legend')
coder.extrinsic('tic')
coder.extrinsic('toc')

%% Platoon coordinator parameter
n_DP                = 1389;         % number of discratization speed points
%n_DP                = 2000;
dv_DP               = 0.005;        % [m/s] speed discatization size
ds_DP               = 6;            % [m] discratization lenght
s_pred_DP           = 8000;        % [m] lenght of the prediction horizon
hp_DP               = floor(s_pred_DP/ds_DP);

%% reading parameters
% reading road information
road_s_vec          = road_info(:,1);       % spacial coordinate
road_h_vec          = road_info(:,2);       % road altitude

% reading vehicle info
n_vehicles          = 3;                    % number of vehicles
vehicle_m_vec       = vehicles_info(1,:);  % vehicles masses
vehicle_length_vec  = vehicles_info(2,:);  % vehicles lengths
vehicle_cr_vec      = vehicles_info(3,:);  % vehicles roll resistance

% reading engine info
engine_Pmin_vec     = engine_info(1,:);
engine_Pmax_vec     = engine_info(2,:);
engine_fuel_coeff_vec= engine_info(3:5,:);

%% current state
s1_curr             = X1(2);
v1_curr             = X1(1);

%% vehicle parameters
%ro_v=1.2;
ro_v=1.225;
%A_v=10;
A_v=9.487;
g=9.81;
%Cd0=.6;
Cd0=0.5271;
%Cd2=23*.30/.35;
Cd2=26.666;
%Cd1=.65*Cd2;
Cd1=14.666;

%% controller initialization
v_DP=max_v_ref+((-(n_DP-1)*dv_DP):dv_DP:0);
k_init=find(v_DP>=v1_curr,1);
```

```
v_DP=v_DP-(v_DP(k_init)-v1_curr)*ones(size(v_DP));

beta_old=beta;
beta_low=NaN;
beta_high=NaN;
v_ave=0;

%% DP
% initializtions
s_opt=zeros(hp_DP,1);
k_opt=zeros(hp_DP,1);
v_opt=zeros(hp_DP,1);
t_opt=zeros(hp_DP,1);
feb_opt=zeros(hp_DP,n_vehicles);
Peb_opt=zeros(hp_DP,n_vehicles);
big_number=Inf;

Cd_vec=zeros(n_vehicles,length(v_DP));
Cd_vec(1,:)=ones(size(v_DP))*Cd0;
for i=2:n_vehicles
    Cd_vec(i,:)=Cd0*(1-Cd1*ones(size(v_DP))./(Cd2+(v_DP*time_gap(i)-vehicle_length_vec(i-1))));
end

% DP routine
while (abs(v_ave-ave_v_ref)>max_err_v) && (beta_old>0 || v_ave<=22)
    tic
    J=zeros(hp_DP,n_DP);
    J_temp=zeros(n_DP,1);
    k_fut=zeros(hp_DP,n_DP);
    if fixed_end_v
        J(hp_DP,1:n_DP)=big_number;
        J(hp_DP,find(v_DP>=end_v,1))=0;
    else
        J(hp_DP,1:n_DP)=sum((1/2*vehicle_m_vec./engine_Pmax_vec(1)
        *polyval(engine_fuel_coeff_vec(:,1),engine_Pmax_vec(1)))'*(ave_v_ref^2-v_DP.^2));
    end
    k_fut(hp_DP,1:n_DP)=NaN;
    s_opt(:)=s1_curr+ds_DP*(0:(hp_DP-1));
    Feb_temp=zeros(1,n_vehicles);
    Peb_temp=zeros(1,n_vehicles);
    for j=hp_DP-1:-1:1
        dh=interp1(road_s_vec,road_h_vec,s_opt(j+1))-interp1(road_s_vec,road_h_vec,s_opt(j));
        for i=1:n_DP
            % 1 - same speed
            k=i;
            % used model (V-v)*V/ds*m=Fe-1/2*ro_v*A_v*Cd*V^2-m*g*sin(alpha(x(V))
            Feb_temp(1,:)=vehicle_m_vec'/ds_DP*v_DP(k)*(v_DP(k)-v_DP(i))+1/2
            *ro_v*A_v*Cd_vec(:,k)*v_DP(k)^2+g*vehicle_cr_vec'.*vehicle_m_vec'
            +vehicle_m_vec'*g*dh/ds_DP;
            Peb_temp(1,:)=Feb_temp*v_DP(k);
            if prod(Peb_temp(1,:)<=engine_Pmax_vec)
               fuelp_temp=polyval(engine_fuel_coeff_vec(:,1),max(engine_Pmin_vec,Peb_temp));
               if CLAC==0
                   fuelp_temp(2:end)=0;
               end
               J_temp(k)=J(j+1,k)+sum(fuelp_temp)*ds_DP/v_DP(k)+beta*ds_DP/v_DP(k);
               else
                J_temp(k)=big_number;
            end

            % 2 - from higher speed
            k=i;
            looping=true;
            while looping && k<n_DP
```

```
                k=k+1;
                Feb_temp(1,:)=vehicle_m_vec'/ds_DP*v_DP(k)*(v_DP(k)-v_DP(i))
                +1/2*ro_v*A_v*Cd_vec(:,k)
                *v_DP(k)^2+g*vehicle_cr_vec'.*vehicle_m_vec'+vehicle_m_vec'*g*dh/ds_DP;
                Peb_temp(1,:)=Feb_temp*v_DP(k);
                if prod(Peb_temp<=engine_Pmax_vec)
                    fuelp_temp=polyval(engine_fuel_coeff_vec(:,1)
                    ,max(engine_Pmin_vec,Peb_temp));
                    if CLAC==0
                        fuelp_temp(2:end)=0;
                    end
                    J_temp(k)=J(j+1,k)+sum(fuelp_temp)*ds_DP/v_DP(k)+beta*ds_DP/v_DP(k);
                else
                    J_temp(k)=big_number;
                    if not(prod(Peb_temp<=engine_Pmax_vec))
                        looping=false;
                    end
                end
            end
            k_max=k;

            % 3 - from lower speed
            k=i;
            looping=true;
            while looping && k>1
                k=k-1;
                Feb_temp(1,:)=vehicle_m_vec'/ds_DP*v_DP(k)*(v_DP(k)
                -v_DP(i))+1/2*ro_v*A_v*Cd_vec(:,k)
                *v_DP(k)^2+g*vehicle_cr_vec'.*vehicle_m_vec'+vehicle_m_vec'*g*dh/ds_DP;
                Peb_temp(1,:)=Feb_temp*v_DP(k);
                if min(1,sum(Peb_temp>=engine_Pmin_vec))
                && prod(Peb_temp<=engine_Pmax_vec)
                    fuelp_temp=polyval(engine_fuel_coeff_vec(:,1)
                    ,max(engine_Pmin_vec,Peb_temp));
                    if CLAC==0
                        fuelp_temp(2:end)=0;
                    end
                    J_temp(k)=J(j+1,k)+sum(fuelp_temp)*ds_DP/v_DP(k)
                    +beta*ds_DP/v_DP(k);
                else
                    J_temp(k)=big_number;
                    if not(min(1,sum(Peb_temp>=engine_Pmin_vec)))
                        looping=false;
                    end
                end
            end
            k_min=k;

            % optimal at step j,i
            [J(j,i),k_best]=min(J_temp(k_min:k_max));
            k_best=k_best+k_min-1;
            k_fut(j,i)=k_best;
        end
    end

    %% Optimal trajectory
    k_init=find(v_DP>=v1_curr,1);
    k_opt(1)=k_init;
    v_opt(1)=v_DP(k_init);
    t_opt(1)=t_curr;
    feb_opt(end,:)=NaN;
    Peb_opt(end,:)=NaN;
    for j=2:1:hp_DP
        k_opt(j)=k_fut(j-1,k_opt(j-1));
```

```
        v_opt(j)=v_DP(k_fut(j-1,k_opt(j-1)));
        t_opt(j)=t_opt(j-1)+ds_DP/v_opt(j);
    end

    for j=2:1:hp_DP
        dh=interp1(road_s_vec,road_h_vec,s_opt(j))
        -interp1(road_s_vec,road_h_vec,s_opt(j-1));
        feb_opt(j-1,:)=(vehicle_m_vec'/ds_DP*v_opt(j)*(v_opt(j)
        -v_opt(j-1))+1/2*ro_v*A_v*Cd_vec(:,k_opt(j))
        *v_opt(j)^2+g*vehicle_cr_vec'.*vehicle_m_vec'
        +vehicle_m_vec'*g*dh/ds_DP)./vehicle_m_vec';
        Peb_opt(j-1,:)=feb_opt(j-1,:).*vehicle_m_vec*v_opt(j);
    end
    t_end=t_opt(end)-t_curr;
    v_ave=(s_opt(end)-s_opt(1))/t_end;
    v_err=v_ave-ave_v_ref
    if v_ave<ave_v_ref
        beta_low=beta;
    else
        beta_high=beta;
    end
    beta_old=beta;
    if isnan(beta_low) || isnan(beta_high)
        beta=beta+k_beta*(ave_v_ref-v_ave);
    else
        beta=mean([beta_low beta_high]);
    end
    beta_low
    beta_high
    J_trajectory=J(1,k_init);
    if beta_tuning==0;
        v_ave=ave_v_ref;
    end
    toc
end

%% output
s_pc=NaN*ones(hp_DP+40,1);
v_pc=NaN*ones(hp_DP+40,1);
s_pc_new=s_opt(1:hp_DP);
v_pc_new=v_opt(1:hp_DP);
s_pc_old=Xref_pc_per(:,1);
v_pc_old=Xref_pc_per(:,2);
idex_old_temp=find(s_pc_old<s_pc_new(1),1,'last');
if all(idex_old_temp<40) || (isempty(idex_old_temp))
    idex_old=41
else
    idex_old=idex_old_temp(1);
end
s_pc(1:40)=s_pc_old(idex_old-39:idex_old);
v_pc(1:40)=v_pc_old(idex_old-39:idex_old);
s_pc(41:hp_DP+40)=s_pc_new;
v_pc(41:hp_DP+40)=v_pc_new;

Xref_pc_per=[s_pc, v_pc];
Xref_pc=Xref_pc_per;
```

2.8.3 Reference Generator

```
function [new_s1, new_v1]= For_s_v(s1, v1, InitialPosition,
Vehicle1Length, Vehicle2Length, InitialDistanceBetweenVehicles12,
InitialDistanceBetweenVehicles23, v_init)

W = 10000;

persistent Box_s1
persistent Box_v1
persistent var1

if isempty(Box_s1)
    Box_s1 = (InitialPosition+InitialDistanceBetweenVehicles23
    +Vehicle2Length+InitialDistanceBetweenVehicles12+Vehicle1Length)*ones(W,1);
end
if isempty(Box_v1)
    Box_v1 = v_init*ones(W,1);
end
if isempty(var1)
    var1 = 1;
end

if var1 <= W
    Box_s1(var1) = s1;
    Box_v1(var1) = v1;
    var1 = var1+1;
    new_s1 = Box_s1;
    new_v1 = Box_v1;
else
    var1 = W;
    temp1_s = Box_s1;
    temp1_v = Box_v1;
    temp2_s = Box_s1;
    temp2_v = Box_v1;
    temp2_s(var1) = s1;
    temp2_v(var1) = v1;
    for i = 1 : var1-1
        temp2_s(var1-i) = temp1_s(var1-i+1);
        temp2_v(var1-i) = temp1_v(var1-i+1);
        var1 = var1+1;
    end
    Box_s1 = temp2_s;
    Box_v1 = temp2_v;
    new_s1 = Box_s1;
    new_v1 = Box_v1;
end

function vRef2 = For_vRef2(s2, new_s1, new_v1, vRefParameter)

i = 1;

while 1
    if new_s1(i) <= s2 && s2 < new_s1(i+1)
        vRef2 = (1-vRefParameter)*new_v1(i);
        break;
    end
    i = i+1;
    if i > size(new_s1, 1)
        vRef2 = (1-vRefParameter)*22;
        break;
    end
end
```

2.8.4 DOB based vehicle controller

```

function alpha = ToMakeSlope(s, distance, slope)

%When the region of slope is stable,
%if 0 <= s && s <= 1000
%    alpha = 0;
%elseif 1000 < s && s <= 1500
%    alpha = 1/20;
%elseif 1500 < s && s <= 2000
%    alpha = 0;
%elseif 2000 < s && s <= 2500
%    alpha = -1/20;
%elseif 2500 < s && s <= 3000
%    alpha = 0;
%elseif 3000 < s && s <= 3700
%    alpha = 1/15;
%elseif 3700 < s && s <= 4200
%    alpha = 0;
%elseif 4200 < s && s <= 4900
%    alpha = -1/15;
%else
%    alpha = 0;
%end

%This is RoadData From Mariefred To Eskilstuna
i = 1;
while 1
    if distance(i) <= s && s < distance(i+1)
        alpha = atan(slope(i));
        break;
    end
    i = i+1;
    if i > size(distance, 1)
        alpha = atan(0);
        break;
    end
end


function [d, AirDragForce, RollingResistanceForce, GravityForce]
= ToMakeDisturbance(v, alpha, g, Vehicle1Cr, M1, Ad, Vehicle1Av, Cd0 )

AirDragForce = - 1./2*Ad*Vehicle1Av*Cd0*v^2;
RollingResistanceForce = - Vehicle1Cr*M1*g*cos(alpha);
GravityForce = - M1*g*sin(alpha);

d = GravityForce + RollingResistanceForce + AirDragForce;

function [uout,umax,umin]= ToMakeSaturation(v, u, Vehicle1Pmax, Vehicle1Pmin,
nominal_mass, nominal_Eb, g, nominal_Cf, sat_region_on)

umax = Vehicle1Pmax./(v);
umin = Vehicle1Pmin./(v) - nominal_mass*nominal_Eb*g*nominal_Cf;

if sat_region_on == 1
    if 0 <= u && u <= umax
        uout = u;
    elseif umax <= u
        uout = umax;
    else
        uout = 0;
    end
else
    if umin <= u && u <= umax
        uout = u;
    elseif umax <= u
        uout = umax;
```

```
    else
        uout = umin;
    end
end
```

Bibliography

[1] K. Ohnishi, "New development of servo technology in mechatronics", *IEEJ Transactions on Industry Applications*, vol. 107, pp. 83-86, 1987.

[2] D. Xing, J. Su, Y. Liu, and J. Zhong, "Robust approach for humanoid joint control based on a disturbance observer", *IET Control Theory and Applications*, vol. 5, pp. 1630-1636, 2011.

[3] K. Ohnishi, M. Shibata, and T. Murakami "Motion Control for Advanced Mechatronics", *IEEE/ASME Transactions on Mechatronics*, vol. 1, pp. 56-67, 1996.

[4] H. S. Lee, and M. Tomizuka, "Robust Motion Controller Design for High-Accuracy Positioning Systems", *IEEE Transactions on Industrial Electronics*, vol. 43, pp. 48-55, 1996.

[5] H. Kobayashi, S. Katsura and K. Ohnishi, "An Analysis of Parameter Variations of Disturbance Observer for Motion Control", *IEEE Transactions on Industrial Electronics*, vol. 54, pp. 3413-3421, 2007.

[6] M. T. White, M. Tomizuka, and C. Smith, "Improved Track Following in Magnetic Disk Drives Using a Disturbance Observer", *IEEE/ASME Transactions on mechatronics*, vol. 5, pp. 3-11, 2000.

[7] W. Dong, G. Y. Gu, X. Zhu, and H. Ding, "High-performance trajectory tracking control of a quadrotor with disturbance observer", *Sensors and Actuators A: Physical*, vol. 211, pp. 67-77, 2014.

[8] S. H. Jeong, S. Jung, and M. Tomizuka, "Attitude Control of a Quad-rotor System Using an Acceleration-based Disturbance Observer: An Empirical Approach*", *IEEE/ASME International Conference on Advanced Intelligent Mechatronics(AIM)*, pp. 916-921, 2012.

[9] X. Fan, and M. Tomizuka, "Robust Disturbance Observer Design for a Power-Assist Electric Bicycle", *Proceedings of the American Control Conference*, pp. 1166-1171, 2010.

[10] B. A. Guvenc, and L. Guvenc, "Robust Two Degree-of-Freedom Add-On Controller Design for Automatic Steering", *IEEE Transactions on Control Systems Technology*, vol. 10, pp. 137-148, 2002.

[11] Y. Kim, K. S. Kim, and S. Kim, "A Novel Disturbance Observer Based Robust Current-Control for a PMSM Drive System", *Proceedings of the IEEE Conference on Decision and Control*, pp. 6043-6046, 2015.

[12] S. Li, and J. Yang, "Robust Autopilot Design for Bank-to-Turn Missiles using Disturbance Observers", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, pp. 558-579, 2013.

[13] K. J. Astrom, and T. Hagglund, "PID controllers: theory, design, and tuning", *Instrument Society of America*, Research Triangle Park, NC, 10, 1995.

[14] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, "Control system design", *Prentice Hall*, vol. 240, 2001.

[15] X. Gao, S. Komada, and T. Hori, "A wind-up Restraint Control of Disturbance Observer System for Saturation of Actuator Torque", *1999 IEEE International Conference*, vol. 1, pp. I84-I88, 1999.

[16] H. Choi, B. Kim, I. Suh, and W. Chung, "Design of Robust High-Speed Motion Controller for a Plant With Actuator Saturation", *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 122, pp. 535-541, 2000.

[17] M. Huba, "Comparing 2DOF PI and predictive disturbance observer based filtered PI control", *Journal of Process Control*, vol. 23, pp. 1379-1400, 2013.

[18] X. S. Chen, J. Yang, S. H. Li, and Q. Li, "Disturbance observer based multi-variable control of ball mill grinding circuits", *Journal of Process Control*, vol. 19, pp. 1205-1213, 2009.

[19] Y. Eun, and E. S. Hamby, "Noise Induced Loss of Tracking in Systems With Saturating Actuators and Antiwindup", *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 136, pp. 054501-1(6 pages), 2014.

[20] J. Lee, and Y. Eun, "Analysis of Noise-Induced Tracking Loss in PI controlled Systems with Anti-windup", *Proceedings of the American Control Conference*, 2016.

[21] D. Kim, K. Park, Y. Eun, S. H. Son, and C. Lu, "When Thermal Control Meets Sensor Noise: Analysis of Noise-induced Temperature Error", *Proceedings of IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 98-107, 2015.

[22] A. Skorokhod, "Asymptotic Methods in the Theory of Stochastic Differential Equations", *American Mathematical Society*, Providence, RI, 1989.

[23] H. Shim, and Y. Joo, "State Space Analysis of Disturbance Observer and a Robust Stability Condition", *Proceedings of the IEEE Conference on Decision and Control*, pp. 2193-2198, 2007.

[24] E. Sariyildiz, and K. Ohnishi, "A Guide to Design Disturbance Observer", *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 136, pp. 021011-1(10 pages), 2014.

[25] A. Mohammadi, M. Tavakoli, H. J. Marquez, and F. Hashemzadeh, "Nonlinear disturbance observer design for robotic manipulators", *Control Engineering Practice*, vol. 21, pp. 253-267, 2013.

[26] J. Back, and H. Shim, "Adding robustness to nominal output-feedback controllers for uncertain nonlinear systems: A nonlinear version of disturbance observer", *Automatica*, vol. 44, pp. 2528-2537, 2008.

[27] E. Schrijver, and J. V. Dijk, "Disturbance Observers for Rigid Mechanical Systems: Equivalence, Stability, and Design", *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 124, pp. 539-548, 2002.

[28] L. Qiu, and K. Zhou, "Introduction to Feedback Control", *International Edition*, Pearson, pp. 158-164, 2010.

[29] M. Zabat, N. Stabile, S. Farascaroli, and F. Browand, "The aerodynamic performance of platoons: A final report", California Partners for Advanced Transit and Highways (PATH), 1995

[30] A. Alam, A. Gattami, and K.H. Johansson, "An experimental study on the fuel reduction potential of heavy duty vehicle platooning", *Intelligent Transportation Systems (ITSC), 13th International IEEE Conference on*, 2010.

[31] A. Alam, B. Besselink, V. Turri, J. Martensson, and K.H. Johansson, "Heavy-Duty Vehicle Platooning for Sustainable Freight Transportation: A Cooperative Method to Enhance Safety and Efficiency", *IEEE CONTROL SYSTEMS MAGAZINE*, vol. 35, 2015.

[32] M. P. Lammert, A. Duran, J. Diez, K. Burton, and A. Nicholson, "Effect of platooning on fuel consumption of class 8 vehicles over a range of speeds, following distances, and mass", *SAE International Journal of Commercial Vehicles*, pp. 626-639, 2014.

[33] E. Hellstrom, M. Ivarsson, J. Aslund, and L. Nielsen, "Look-ahead control for heavy trucks to minimize trip time and fuel consumption", *Control Engineering Practice*, vol. 17(2), pp. 245-254, 2009.

[34] A. Alam, A. Gattami, K. H. Johansson, and C. J. Tomlin, "Guaranteeing safety for heavy duty vehicle platooning: Safe set computations and experimental evaluations", *Control Engineering Practice*, vol. 24, pp. 33-41, 2014.

[35] C. Bonnet, and H. Fritz, "Fuel consumption reduction in a platoon: Experimental results with two electronically coupled trucks at close spacing", No. 2000-01-3056, SAE Technical Paper, 2000.

[36] V. Turri, B. Besselink, J. Martensson, and K.H. Johansson, "Fuel-efficient heavy-duty vehicle platooning by look-ahead control", *Proc. of the 53th IEEE Conference on Decision and Control*, pp. 654-660, Dec, 2014.

[37] V. Turri, B. Besselink, and K.H. Johansson, "Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning", *IEEE Transaction on Control Systems Technology*, 2015.

[38] H. Shim, and Y. Joo, "State Space Analysis of Disturbance Observer and a Robust Stability Condition", *Proc. of the 46th IEEE Conference on Decision and Control*, pp. 2193-2198, Dec, 2007.

[39] G. Park, Y. Joo, C. Lee, and H. Shim, "On Robust Stability of Disturbance Observer for Sampled-Data Systems Under Fast Sampling: An Almost Necessary and Sufficient Condition", *Proc. of the 54th IEEE Conference on Decision and Control*, pp. 7536-7541, Dec, 2015.

[40] E. Choi, S. Han, and J. Choi, "Channel capacity analysis for high speed controller area network (CAN)", *Information and Communication Technology Convergence (ICTC), 2015 International Conference on. IEEE*, pp. 188-190, 2015.

요약문

포화 액츄에이터와 측정잡음을 고려한 외란관측기 기반 제어시스템 연구

첫번째로 외란관측기 기반 제어기의 측정잡음으로 인한 추종오차 현상을 연구하였습니다. 여기서 말하는 외란관측기란 모델불확실성, 외란에 의한 영향을 보상할 수 있는 강인 제어기법입니다. 이러한 이유로 현재 많은 산업체에서는 드론 제어, 로봇 제어, 차량 제어, 전기자전거 제어, 모터 제어 등의 다양한 분야에 외란관측기 기반 제어기를 적용하여 그 효과를 보고 있습니다. 하지만 저희는 외란관측기 기반 제어기를 사용하면서 센서의 측정잡음이 존재하는 경우에 측정잡음으로 인해 제어 시스템의 출력이 참조값을 따라가지 못하는 현상을 발견하였습니다. 저희는 이러한 현상을 외란관측기 기반 제어시스템에서의 나이트 현상이라 명명하였고 Stochastic averaging theory를 사용하여 나이트 현상이 나타나는 조건과 추종오차를 수학적으로 증명하였습니다. 또한 나이트 현상을 줄일 수 있는 제어기 설계 방법 2가지를 제시하였습니다. 본 연구의 타당성을 입증하기 위해 시뮬레이션과 BLDC 모터 테스트 베드를 이용하여 이를 검증하였습니다.

두번째로 자율군집주행기법을 연구하였습니다. 자율군집주행이란 차량 간에 짧은 거리를 유지하며 운행되는 주행방법으로 이를 이용하면 차량이 받는 공기저항의 영향을 줄여 연료를 절감할 수 있습니다. 하지만 화물적재로 인한 차량의 질량이 변하여 차량의 다이나믹스가 변하는 경우 혹은 차량이 외란에 의한 영향을 받을 경우, 차량은 속도 추종을 방해 받게 됩니다. 이는 자율군집주행제어에 매우 악의적인 영향임을 의미합니다. 학회에서는 이러한 문제를 어떻게 해결해야 할지를 제시하지 못하였고, 본 연구를 통해 이를 제어적인 관점에서 어떻게 속도추종을 보상하여 군집주행을 가능케 할 수 있을지에 관해 해결법을 제시하였습니다. 해결 방법으로 차량제어기에 강인제어 기법을 사용하여 제어기를 구현하였고, 시뮬레이션은 스웨덴의 Mariefred 지역에서 Eskilstuna 지역까지의 실제 도로 데이터를 이용하였으며 실제 차량과 유사한 환경으로 시뮬레이션 하였습니다.

주요어휘: 외란관측기, 포화, 측정잡음, 자율군집주행