New Development of Extremum Seeking in Parameter Estimation

Zhenyu Li

Academic Supervisors:
Prof. Iven Mareels
A/Prof. Ying Tan

Advisory Committees:
Prof. Michael Cantoni
Prof. Chris Manzie

Department of Electrical & Electronic Engineering

The University of Melbourne

This dissertation is submitted for the degree of
Master of Philosophy

June 2016

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University.

Zhenyu Li
June 2016

Abstract

Internal combustion engines have undergone a series of increasingly stringent regulations due to their environmental and energy impacts. The inherently conflicting constraints imposed on the engine performance, i.e. exhaust emission reduction with improved fuel efficiency, have been challenging the control to meet the tightening regulations.

In the conventional practice of the engine industry, most of the engine parameters that affect the emissions and the efficiency, such as fuel injection, ignition timing, boost pressure and EGR, are determined from look-up tables calibrated around each operating point in an experimental environment. In order to have better real-time performance and be more robust to disturbances, it is proposed to develop a control method that can further adjust input values based on real-time plant output and/or plant state. A difficulty in implementing this method is the practicality of sensors (high price, conflict with the need of customers). In this thesis a soft sensor is used to infer emissions and fuel efficiency through state estimation from an off-line determined model and from on-line measurements of energy output. Here it is proposed to optimize, in real time, the model parameters to improve the local validity of the model by considering how well the model works to estimate the energy output. The assumption is that a better set of model parameters, a better model, will estimate emissions and the efficiency better. Given the operating point and the optimized model, in a subsequent control action those engine parameters that influence emissions and fuel efficiency will be selected so as to further reduce emissions and improve efficiency at the given operating conditions.

Extremum seeking (ES), an on-line optimization method, is a robust (model-free) approach for system optimization. One noticeable weakness of ES is its slow convergence speed. Furthermore, the working performance of ES degrades when a time delay is present in the closed loop. Here it is proposed to use ES despite the presence of time delay in the actuatory/sensory environment, because it is possible to design a prediction-like control law to overcome the issues of time delay regarding the closed-loop stabilization scheme.

Motivated by the engine application and ES theory development, this work focuses on extending ES to have better convergence performance and to deal with delay in the context of specific engine optimization problem. Specifically, this work

1. analyses the effect of delay on the working of ES (stability and convergence aspects);
2. proposes a supervisory controller to improve ES convergence performance using the available ES tuning parameters;
3. utilizes ES to identify the time delay in a system, and removes the negative delay influence from the closed-loop stabilization scheme.

Please note that there are a number of ES algorithms in the literature and the proposed method can be used for different types of ES techniques. Here in this thesis, a standard ES algorithm whose performance properties are very well known in the literature is utilised as an example to illustrate the proposed method.

Table of contents

Chapter 1

Introduction

Internal combustion (IC) engines are described by highly nonlinear multi-input-multi-output dynamic equations. Among the outputs of an IC engine, fuel (thermal) efficiency and exhaust emissions are required to meet stringent regulations that set upper thresholds for emissions and lower threshold for fuel efficiency for different on-road operating conditions.

The main inputs/outputs for a certain type IC engine are shown in Figure 1.1. Given

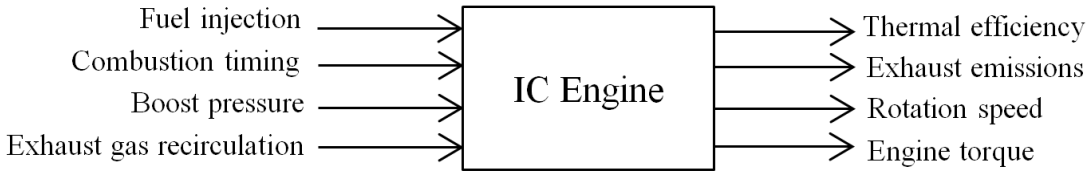

Fig. 1.1 Some inputs/outputs for a certain IC engine

the stringent nature of the regulations, engines come with sensors that can assist control strategies in measuring exhaust emissions and in measuring fuel efficiency by way of measuring a pressure or temperature. The extra cost for the relatively expensive sensors may be prohibition from the perspective of customers. For this reason, it is currently not viable to regulate the real-time performance of the engines by deploying the sensors to help measure in real-time the efficiency and the emissions.

There is another issue with regard to the measurement of the outputs. Even if some sensors would be deployed in product engines in the near future, the corresponding measurements, as is shown in Figure 1.2, would have variable time delay because the unmeasured substances (NOx, CO, fume, etc.) need to go a certain way before reaching the sensors. The velocity of the transmission of mass depends on the variable conditions like pressure and

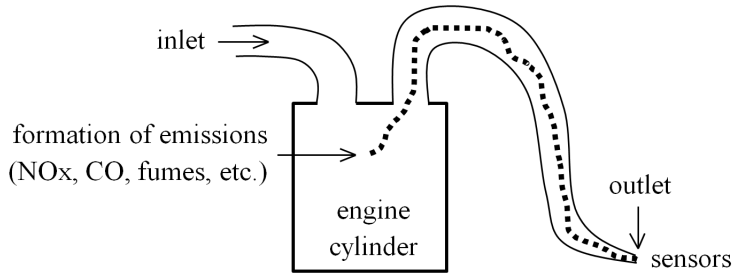

Fig. 1.2 Cause of delay - transmission of mass

temperature, which results in the variability of delay time. This variable time delay is roughly the same operating time of $50 \sim 150$ engine cycles, which not only slows down each of the control responses but also causes more troubles to the design of the control law due to its variability.

Although the sensors could not be currently installed in product engines as customers are unwilling to afford the cost of the sensors, these sensors are utilized in experimental engines in a laboratory environment, which leads to two different ways to calibrate the performance of IC engines.

There are currently two different ways [1] to optimize the performance of IC engines, i.e. off-line calibration with gain scheduling (e.g. [14–18, 22, 23]) and on-line calibration (e.g. [19–21]). Off-line calibration is about off-line, model based [1] or non-model based [14–18, 22, 23], experiments to obtain the optimal or near-optimal values of control inputs for each operating point. The operating points are typically defined by engine speed and engine load. These experimentally obtained control input values for each operating point then populate a look-up table which is stored in the engine control unit (ECU). When the engine is operated in real time (on-road operation), the control process for the engine is shown in

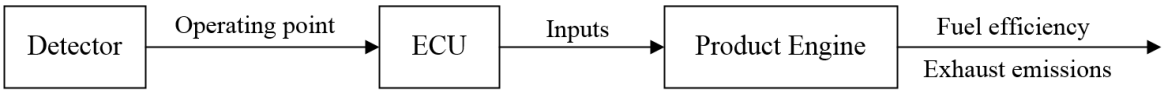

Fig. 1.3 Basic diagram of the real-time operation of a product engine

Figure 1.3. First, the current operating point (defined by engine speed and engine load) is detected and passed to the ECU. Then, based on the experimentally constructed look-up tables stored in the ECU, the corresponding input values to the engine for this operating point are obtained and the ECU will give the corresponding inputs to the plant.

One of the advantages of this calibration method is the simplicity of the control structure in real time (Figure 1.3). Moreover, in calibration conditions, the sample time for the sensors can be slow so that it is much easier to measure in calibration conditions than in real time operation. Besides, it will not have any influence on the real time operation of product engines if the experimental calibration process is long. However, good performance of the engine demands a high density of operating points that requires a large amount of experiments to be carried out, which is expensive. In addition, because the input values are obtained in an experimental environment, disturbances or deviations from on-road operating environment that are not adequately captured in the look-up table will lead to undesired engine performance. Also, aging and performance degradation of the engine cannot be easily accounted for.

In on-line calibration, non-model based optimization methods like extremum seeking (ES) are considered (see e.g. black-box ES [25–34] and grey-box ES [19, 21, 35]). The basic

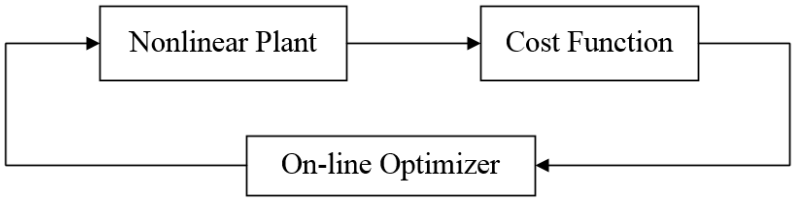

Fig. 1.4 Basic diagram of on-line optimization

diagram is shown in Figure 1.4. Extremum seeking, an on-line optimization method, is able to find near optimal inputs using on-line measurements of input and output signals for a large class of systems, without relying on a precise model. Therefore, ES is a robust (model-free) optimization approach for input design. Taking into account the fact that engine operation is a complex and highly nonlinear process, ES is an attractive idea for automotive engine performance optimization and it has been utilized in automotive engine management with a view to adjusting parameters that drive designed cost function to its minimum/maximum. For example, ES was utilized to tune PID controller parameters on-line in order to regulate engine combustion timing position [16]. Nonlinear programming based ES was used in automotive engine calibration for multi-input cases [14]. In [17], gasoline engine speed control was implemented via ES to find the optimal starting speed. The complex HCCI engine combustion timing control problem was addressed in [18] by controlling the intake air temperature and using the cylinder pressure measurement as feedback. ES algorithms were used to find an optimal set point (for maximizing the engine efficiency) as well as the optimal controller-parameters. Corti et al [22] applied ES to control gasoline engine spark timing and air-fuel ratio to optimize a cost function based on the indicated torque, exhaust

temperature and knock intensity. ES has also been used for online tuning in flex-fuel [21] and compressed natural gas (CNG) engines [20].

However, as an on-line calibration method, ES is slow in terms of convergence when optimizing a nonlinear plant with unknown dynamics. ES works for steady state input-output map optimization. It has to wait for sufficiently long so that the transient process of the plant output dies away, which is implemented by tuning parameters of ES appropriately so that the dynamic process of the system is always faster than the updating of ES process (time scale separation [8]). In on-line calibration process, most existing ES algorithms are used directly in control loops without requiring the knowledge of plants as ES is a model-free method; the corresponding convergence performance of ES is not desirable. In general, ES optimization processes are in the order of minutes, during which the engine operating condition is kept relatively constant. Thus, this method is ineffective when operating conditions change quickly during acceleration and deceleration phases, which is rather the norm for city driving.

During the past decades, regulatory constraints being imposed on engine emissions and fuel efficiency are getting increasingly stringent, which is challenging the conventional control practice for IC engines to meet the tightening regulations. Recently, a non-conventional combustion mode, called low temperature combustion (LTC), is proposed for car engines. The fundamental mechanism of low temperature combustion is to limit the in-cylinder combustion temperature below the NOx formation threshold. Its inherent drawback is that LTC affects overall efficiency in a negative manner. There exist possibilities to achieve LTC for both spark ignition (SI) engines and compression ignition (CI) engines. However, considering the fact that CI engines produce more emissions and has higher thermal efficiency, LTC is more suitable for CI engines. Besides, the after-treatment system for a CI engine is complex, which needs a minimum of pollutant formation during the combustion process.

In this thesis, we investigate two theoretical problems motivated by a diesel engine optimization project. The diesel engine (CI engine) project is about regulating the in-cylinder combustion temperature (inner state of the engine) to achieve low temperature combustion mode. The engine project will be discussed in Chapter 2. The two theoretical problems discussed in this thesis are about convergence shortening for ES and time delay identification. Since the sample-data structure and discretization design in the engine project are finished by the collaborators [24], the contribution of this thesis is that it provides a theoretical method for a general continuous time problem, and this engine management project becomes a discrete time application and verification of this general method.

Chapter 2

Problems from engine project

2.1 Overiew of engine project

The diagram of the engine optimization project is shown in Figure 2.1. Here let me first

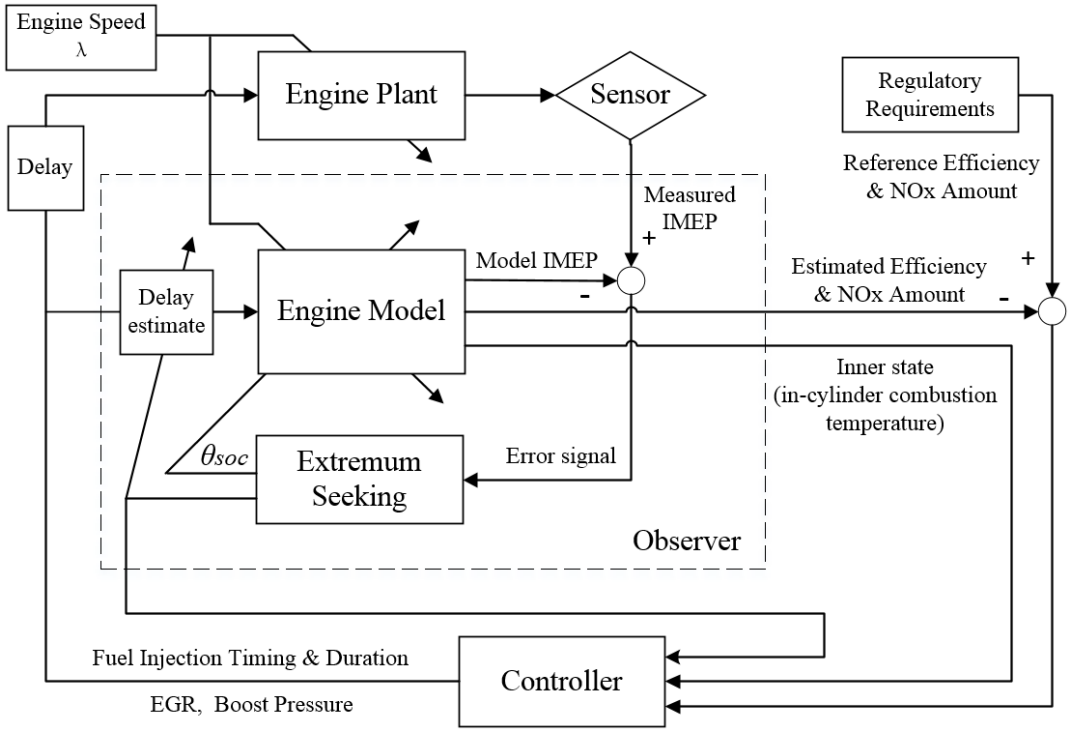

Fig. 2.1 Diagram of the engine project

explain the main idea of this engine project before explaining the corresponding symbol (IMEP), engine model, extremum seeking and delay identification. In this project, a soft sensor is used to infer the emission and fuel efficiency through state estimation from an off-line

determined model and from on-line measurements of energy output (IMEP). It is proposed to refine, in real time, the model parameter to improve the local validity of the model by considering how well the model works to estimate the energy output. The assumption is that a better set of model parameters, a better model, will estimate the emission and the efficiency better. Given the operating point and the improved model, in a subsequent control action those engine parameters that influence emissions and fuel efficiency are selected so as to further reduce emissions and improve efficiency at the given operating conditions.

The reference efficiency and NOx amount are set according to the regulatory efficiency and emission requirements. The sensor output, indicated mean effective pressure (IMEP), an indicator for engine load, is calculated as below:

$$IMEP = \frac{Engine\ work\ per\ cycle}{Cylinder\ displacement\ volume} \tag{2.1}$$

Engine work is measured by sensors. The cylinder displacement volume is an intrinsic engine parameter. Since the engine model is constructed using pressure, temperature and volume, without entropy, the in-cylinder pressure–volume diagram (Figure 2.2) is utilized to illustrate diesel engine in-cylinder combustion process. The shadowed area in Figure 2.2 (abcda)

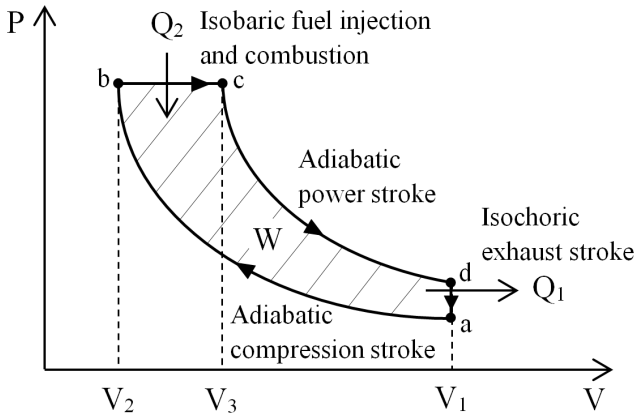

Fig. 2.2 Combustion process (P-V diagram) of diesel engine

repeats the work in each four-stroke engine cycle. The cylinder displacement volume is the difference between cylinder maximum volume V_1 and cylinder minimum volume V_2. For details of diesel cycles, please refer to [2, 3].

The engine model is provided by [24]; the model in [24] is briefly described for completeness, as follows.

State equation:

$$\begin{bmatrix} \hat{T}(\theta_k) - \hat{T}(\theta_{k-1}) \\ \hat{P}(\theta_k) - \hat{P}(\theta_{k-1}) \end{bmatrix} = \begin{bmatrix} A \cdot [Q_{hr}(\theta_{soc}) - Q_{ht} - W_{PV}] \\ \hat{P}(\theta_{k-1}) \cdot \left[(\frac{\hat{T}(\theta_k)}{\hat{T}(\theta_{k-1})})^{\frac{gamma}{gamma-1}} - 1 \right] \end{bmatrix} \tag{2.2}$$

Output equation:

$$IMEP(engine\ cycle) = \sum_{IVC}^{EVO} \frac{\hat{P}(\theta_k) \cdot (V(\theta_k) - V(\theta_{k-1})}{V_d} \tag{2.3}$$

$$NO_x(engine\ cycle) = \sum_{IVC}^{EVO} f(\hat{T}(\theta_{k-1})) \tag{2.4}$$

$$Eff(engine\ cycle) = \sum^{cycle} \frac{\hat{P}(\theta_k) \cdot (V(\theta_k) - V(\theta_{k-1})}{m_{fuel} \cdot LHV} \tag{2.5}$$

Parameters in state equation (2.2) are in discrete crank angle θ_k domain.

θ_k	engine crank angle degree, $k \in N$ ranging from 1 to 720
$\hat{T}(\theta_k)$	in-cylinder temperature estimate at crank angle (θ_k)
$\hat{P}(\theta_k)$	in-cylinder pressure estimate at crank angle (θ_k)
A	specific ratio, calculated by euqation (6) in [24], a constant
gamma	specific heat ratio
$\theta_{SOC} \in [\theta_{IVC}, \theta_{EVO}]$	crank angle location at the start of combustion
IVC	intake valve closing crank angle
EVO	exhaust valve opening crank angle
Q_{hr}	combustion heat release amount parameterized by θ_{SOC}
Q_{ht}	cylinder wall heat transfer amount
W_{PV}	piston work amount for each crank angle resolution
V_d	in-cylinder displacement volume
m_{fuel}	cyclic fuel injection amount
LHV	lower heating value
Eff	fuel efficiency

NOx formation and in-cylinder temperature relationship Equation (2.4) is represented by a simplified Zeldovich thermal NOx mechanism. For details of the model, please refer to [24].

Engine crank angle domain [2, 3] is illustrated in Figure 2.3. The crankshaft angular velocity at time:

$$\omega_r(t) = 2\pi \cdot rpm/60 \tag{2.6}$$

TDC : top dead centre
P : piston pin
N : crank pin
O : crank centre
l : rod length
r : crank radius (half stroke)
θ : crank angle (from TDC)
c : position of piston pin from crank centre

$$l^2 = r^2 + c^2 - 2rc \cdot \cos(\theta)$$

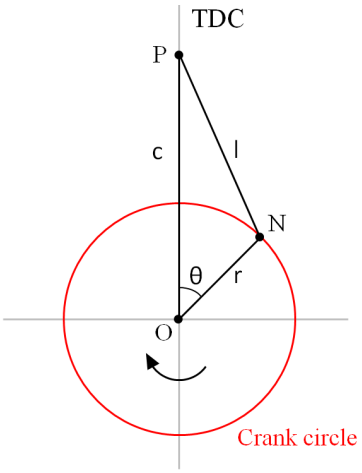

Fig. 2.3 Piston motion geometry

where engine rotation speed *rpm* is the number of revolutions per minute. The relation between crank angle (θ) domain and time (t) domain:

$$\theta = \int_{t_0}^{t_1} \omega_r \, \mathrm{d}t \tag{2.7}$$

where θ is the current crank angle value, t_0 is the starting time (the time when piston starts at top dead centre TDC and where we set θ to be zero) and t_1 is the current time value. For each four-stroke engine cycle (unit one in engine cycle domain), the overall crank angle change is 720 degrees while the time duration ($t_1 - t_0$, $720 = \int_{t_0}^{t_1} \omega_r \, \mathrm{d}t$) varies from cycle to cycle since the angular velocity ω_r varies from cycle to cycle.

Here is a simplified structure of the engine plant, as illustrated in Figure 2.4. The mathematical description of the plant and the model are formulated in crank angle (j) - engine cycle (n) domain, as are shown in Equation (2.8) - (2.13).

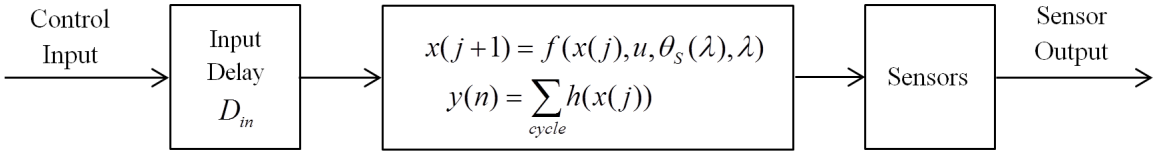

Fig. 2.4 Engine plant diagram

Engine plant in crank angle (j) - engine cycle (n) domain:

$$x(j+1) = f(x(j), u(j - D_{in}), \theta_s(\lambda), \lambda), \;\; x(0) = x_0, \; x \in R^2, \; j \in N, \theta_s \in [\underline{\theta}, \overline{\theta}], \\ \lambda \in R, u \in R^l, l \in N^+ \tag{2.8}$$

$$y(n) = \sum_{\tau=1+(n-1)\cdot T}^{\tau=n\cdot T} h(x(\tau)) \,, \qquad y \in R^3, n \in N^+, \tau \in N^+, \; T = 720 \tag{2.9}$$

Sensor output:

$$\zeta_1(n) = y_1(n), \; \zeta_1 \in R \tag{2.10}$$

$$y_2(n) \in R, \; y_3(n) \in R \textit{ are unmeasurable} \tag{2.11}$$

where

the states x represent in-cylinder temperature and pressure;

plant outputs y represent $IMEP$, NO_x emission and Eff fuel efficiency;

j is the crank angle indicator and n is engine cycle number;

λ is the engine rotation speed and u is the control input;

D_{in} is the unknown delay time that varies according to different engine rotation speed λ;

θ_s is an unknown variable parameterized by λ and it has a known data range due to the physical structure of engine cylinder. This parameter can be measured.

Engine model:

$$\hat{x}(m+1) = \hat{f}(\hat{x}(m), u(m), \hat{\theta}^*(\lambda), \lambda), \;\; \hat{x}(0) = \hat{x}_0, \; \hat{x} \in R^2, \; m \in [0, 719] \\ \hat{\theta} \in [\underline{\theta}, \overline{\theta}], \lambda \in R, u \in R^l, l \in N^+ \tag{2.12}$$

$$\hat{y} = \sum_{\tau=1}^{\tau=720} \hat{h}(\hat{x}(\tau)) \,, \qquad \hat{y} \in R^3, \tau \in N^+ \tag{2.13}$$

This engine model is just a mapping, rather than an ongoing cycle-by-cycle process as Equation (2.8, 2.9). The engine-cycle based model (or called event-based model in [13, 24]) has been calibrated off-line with experimental data; the corresponding validation results are presented in [24].

The engine model is constructed based on the engine mechanism, but it is different from the real engine plant. When given the same inputs to both the plant and the model, their outputs could be very close to each other, but inside the plant and the model, they could be quite different. The value of the parameter θ_s in the plant (2.8) can be measured.

However, the corresponding parameter $\hat{\theta}^*$ in the model (2.12) could be quite different from the measured value of θ_s. Thus, when the engine is operated in reality, this model needs to be calibrated with on-line measurement of the plant output to identify the unknown parameter $\hat{\theta}^*$ in the model ($\hat{\theta}^*$is the same as the parameter θ_{soc} in (2.2)).

Assumption 2.1.1
Given the continuous model, for any control input u belonging to the possible control input range, and for any feasible engine speed λ, there exists only one $\hat{\theta}^*$ in known set $[\underline{\theta}, \overline{\theta}]$ such that $|y_1 - \hat{y}_1(\hat{\theta})|_{min}$ is achieved.

Explanation for Assumption 2.1.1
Since engine operation is cyclic and as is shown from (2.8), (2.9), (2.12) and (2.13), the output is a summation of function of states over a whole cyclic period and states are related to state initial conditon, control input u, engine speed λ and the unknown parameter $\hat{\theta}$. When the initial condition, control input and engine speed are given, the relationship among model output $\hat{y}$ and the parameter $\hat{\theta}$ is:

$$\hat{y} = g(\hat{\theta}) \tag{2.14}$$

where g is a continuous nonlinear static mapping. Suppose y_{mea} is the measured y_1 output value from the engine plant, Assumption 2.1.1 means that there exists only one $\hat{\theta}^* \in [\underline{\theta}, \overline{\theta}]$ such that

$$\hat{\theta} \to \hat{\theta}^* \;\Rightarrow\; |\hat{y}_1 - y_{mea}| \;\to\; |\hat{y}_1 - y_{mea}|_{min}$$

$|\hat{y}_1 - y_1|_{min}$ is related to the closeness of intial condtions ($\|\hat{x}_0 - x_0\|$, where $\hat{x}_0$ is pre-defined from experimental data) and closeness between the model and the plant (Assumption 2.1.2), both of which are validated by the collaboration group [24].

Assumption 2.1.2
When $|\hat{\theta} - \hat{\theta}^*|$ is small enough, $||\hat{y} - y(n_0)||$ & $||\hat{x}(m) - x(j)||$ $(n_0, m, j \in N^+, m \in [1, 720], j \in [1 + 720 \cdot (n_0 - 1), 720 \cdot n_0])$ are within permissible tolerance.

Explanation for Assumption 2.1.2
Assumption 2.1.2 means that after finishing the model calibration, a precise model (precise enough for control design) is obtained.

Here is the explanation why the variable input delay should be known to achieve the corresponding feedback control regulation. Suppose the engine speed changes from λ_0 to λ_1,

as is shown in Figure 2.5. The speed is detected at the n_1th engine cycle and the input values

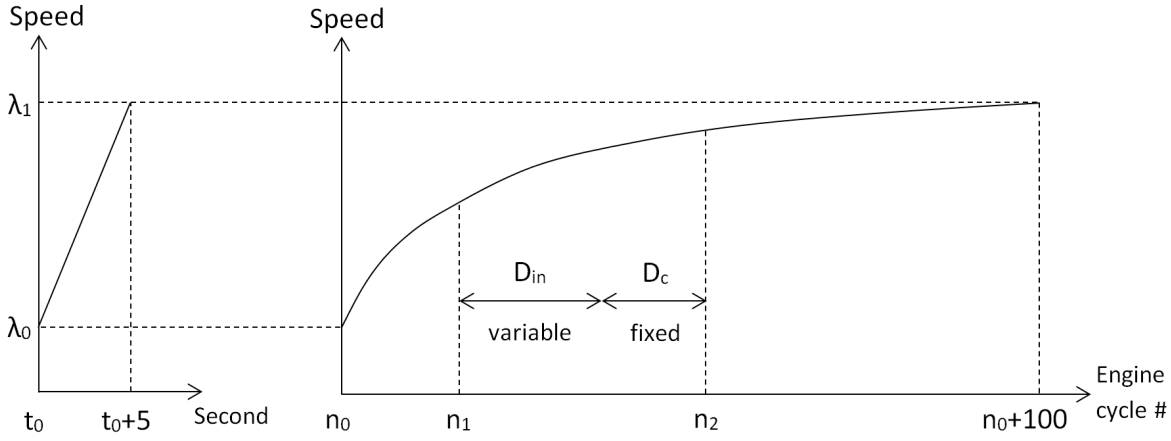

Fig. 2.5 Explanation for the effect of delay

to the engine are calculated from the engine model. But due to the existence of variable input delay D_{in} and fixed computational time D_c, the input values calculated for the n_1th engine cycle will affect the engine at n_2th cycle, which may not lead to the desired engine performance meeting the requirements. Thus, on-line identifying the variable time delay and compensating it in the control would be helpful to get a better engine performance.

The on-line optimization process is:

1. A driver gives instruction to accelerate or decelerate, and the engine speed λ changes; the corresponding $\hat{\theta}^*$ changes.

2. Mode detector gives the current speed value λ to engine control unit (ECU) where the engine model is stored; output sensor feedback the plant output y_1 (the output without delay) as a calibration reference to ECU.

3. Extremum seeking begins to minimize the error between $\hat{y}_1$ and y_1 by driving $\hat{\theta}$ converging into a small neighborhood of $\hat{\theta}^*$.

4. When the on-line model calibration process finishes, $\|\hat{y}-y\|$ & $\|\hat{x}-x\|$ (in Assumption 2.1.2) are within permissible tolerance; Extremum seeking then begins to identify the input delay utilizing the calibrated model.

5. When the on-line optimization process is done, $\hat{y}$, $\hat{x}$, $\hat{D}_{in}$ are obtained, all of which are suitable approximation for the corresponding real values. The corresponding feedback control law is:

$$u = u\left(\hat{x}(1),\hat{x}(2),...,\hat{x}(720),\ \hat{D}_{in},\ \lambda\right) \tag{2.15}$$

During Step 1,2 and 3, the feedback control input u are fixed in order to calibrate the model. Step 3 is the model calibration process and during the process, only $\hat{\theta}$ estimation loop is working while delay identification $\hat{D}_{in}$ loop does not work. When the model calibration process finishes (from Step 4), $\hat{\theta}$ estimation loop stops working and delay identification loop stops to work. The control input u begin to change from Step 4 in order to identify the delay and implement the feedback control.

Challenges

There are two main challenges for designing the on-line optimization process.

- Taking into account engine's operation in reality, this on-line optimization process (model calibration & delay identification) is supposed to finish as quickly as possible, which is closely related to the convergence speed of extremum seeking.
- Since the delay identification and feedback stabilization are working simultaneously and delay $\hat{D}_{in}$ is one of the parameters of the feedback control law (2.15), it is both interesting and essential to design the delay identification process so that this parameter updating process does not affect the stability of the feedback control system (plant + feedback controller).

The four different domains in this project is explained in Figure 2.6. T_0 represents the

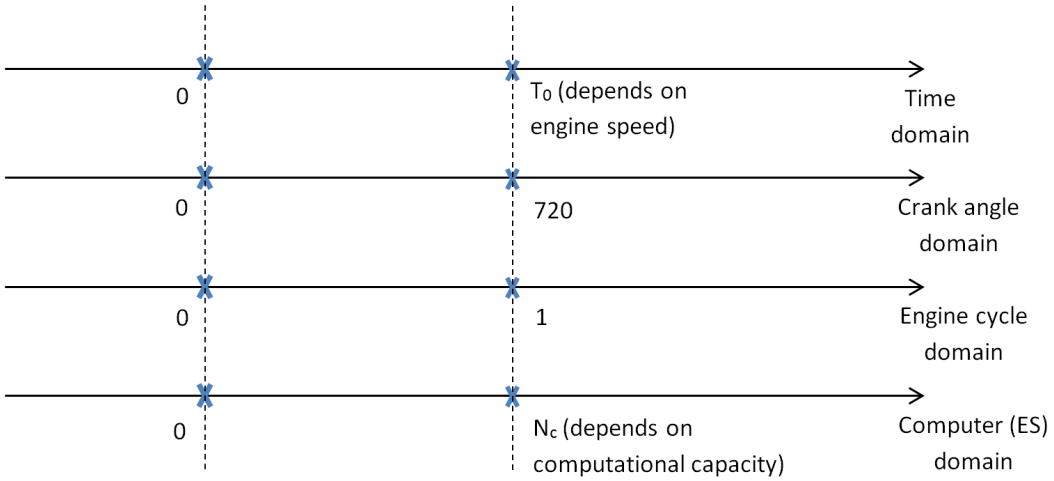

Fig. 2.6 Explanation for differnet domains

operation time for one engine cycle, which is variable (depending on the variable engine speed). The model calibration process (ES+model) is in computer domain. N_c is the number of iterative calculation, which depends on the computational capacity of the ECU.

2.2 Extremum seeking

2.2.1 Gradient algorithm

Optimization [4] deals with problems including optimality conditions and algorithms. Here let us have a look at a simple case with the fundamental optimality conditions and the widely used gradient algorithm.

Let us consider a single-input-single-output (SISO) smooth static mapping case.

Assumption 2.2.1 *There exists a unique θ^* maximizing $Q(\theta)$ and the following holds:*

$$Q'(\theta^*) = 0 \ , \ \theta^* \in R \tag{2.16}$$

$$Q'(\theta^* + \zeta)\zeta < 0 \ , \ \forall \zeta \in R - \{0\} \tag{2.17}$$

Problem Find θ^* with known Q' (the first derivative of Q).

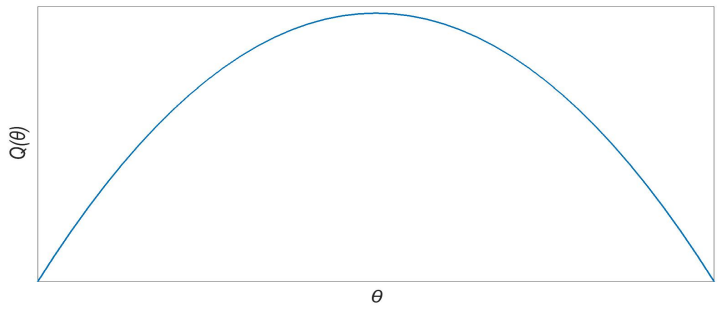

Fig. 2.7 SISO smooth static mapping $Q(\theta)$

When solving this kind of optimization problem, gradient driven alogrithms use gradient information (current gradient, or an estimate of current gradient) and the updating law of these algorithms in continuous time is of the form:

$$\dot{\hat{\theta}} = \delta \cdot Q'(\hat{\theta}) \tag{2.18}$$

or in iteration format

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \delta \cdot Q'(\hat{\theta}(k)) \tag{2.19}$$

where $\delta > 0$ is the updating step size and Q' refers to the gradient information (current gradient). When searching for a minimun, the sign of δ in (2.18) and (2.19) needs to be changed from positive to negative.

It has been proved [11, 12] that for the updating law (2.19), as long as the step size δ is smaller than a threshold $\bar{\delta}$ which is related to the mapping Q, $\hat{\theta}$ will converge into a small neighborhood of θ^*, the size of which depends on the step size δ and the mapping Q. The smaller the step size, the smaller the final residual oscillation and the longer the converging time. As for the continuous time updating law (2.18), $\hat{\theta}$ converges to θ^* without any residual oscillation.

2.2.2 Extremum seeking

Extremum seeking (ES) is an on-line optimization technique. It is a way of implementing gradient-driven algorithms [4] like gradient method or Newton's method, without having an explicit knowledge of the gradient information or dynamics of the plant. The existing literature regarding extremum seeking can be classified into three categories: adaptive ES [5], numeric ES [6] and stochastic ES [7]. Here we focus on the perturbation-based adaptive extremum seeking [5, 29]. In the ES scheme, there are two things that need to be determined, i.e. how the ES scheme gets the gradient information and what the step size is in the ES.

Static plant

Let us first consider the same single-input-single-output smooth static mapping Q. A basic diagram of perturbation-based single-input-single-output extremum seeking scheme [25, 29] is shown in Figure 2.8.

Problem Given Assumption 2.2.1 in Section 2.2.1, find θ^* without knowing Q or Q', but given the ability to measure $Q(\theta)$ for any given θ.

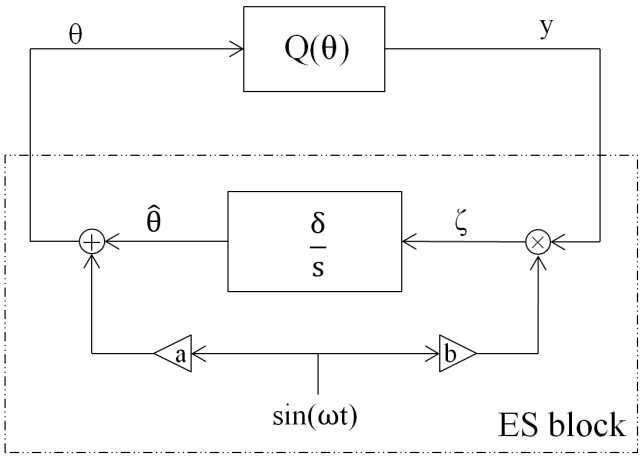

Fig. 2.8 ES for static plant

The algorithm in Figure 2.8 can be expressed as:

$$y = Q(\theta) = Q(\hat{\theta} + asin\omega t) \tag{2.20}$$

$$\zeta = b \cdot \sin \omega t \cdot y = b \sin \omega t \cdot Q\left(\hat{\theta} + a \sin \omega t\right) \tag{2.21}$$

$$\dot{\hat{\theta}} = \delta \cdot \zeta = \delta \cdot b \sin \omega t \cdot Q\left(\hat{\theta} + a \sin \omega t\right) \tag{2.22}$$

where a and b are the amplitudes of the dither, ω is the frequency of the dither and δ is the gain of the integral action.

Introduce the change of coordinates, $\tilde{\theta} = \hat{\theta} - \theta^*$ and Equation (2.22) becomes:

$$\dot{\tilde{\theta}} = \dot{\hat{\theta}} = \delta \cdot b \sin \omega t \cdot Q\left(\tilde{\theta} + \theta^* + a \sin \omega t\right) \tag{2.23}$$

Existing theorem [29]

Under the Assumption 2.2.1, the solution of System (2.23) with parameter $(a, \frac{b\delta}{\omega})$ *is semi-globally practically asymptotically (SPA) stable.*

Heuristic explanation

By using Taylor series expansion for $Q(\hat{\theta} + a \sin \omega t)$, it leads to

$$\begin{aligned} & Q(\hat{\theta} + a \sin \omega t) \\ &= Q(\hat{\theta}) + Q'(\hat{\theta}) \cdot a \sin \omega t + \frac{1}{2} \cdot Q''(\hat{\theta}) \cdot (a \sin \omega t)^2 + O(a^3) \end{aligned} \tag{2.24}$$

Substituting (2.24) into (2.22) yields

$$\dot{\hat{\theta}} = \delta \cdot \left[Q(\hat{\theta}) \cdot bsin\omega t + ab \cdot Q'(\hat{\theta}) \cdot (sin\omega t)^2 + \frac{a^2 b}{2} \cdot Q''(\hat{\theta}) \cdot (sin\omega t)^3 + O(a^3 b) \right] \tag{2.25}$$

We can re-write (2.25) into a new time scale $\sigma = \omega \cdot t$:

$$\begin{aligned} \frac{d\hat{\theta}}{dt} = \frac{d\hat{\theta}}{d\sigma} \cdot \frac{d\sigma}{dt} = \omega \cdot \frac{d\hat{\theta}}{d\sigma} &= \delta \cdot \left[Q(\hat{\theta}) \cdot bsin\sigma + ab \cdot Q'(\hat{\theta}) \cdot (sin\sigma)^2 \right] \\ &+ \delta \cdot \left[\frac{a^2 b}{2} \cdot Q''(\hat{\theta}) \cdot (sin\sigma)^3 + O(a^3 b) \right] \end{aligned} \tag{2.26}$$

$$\frac{d\hat{\theta}}{d\sigma} = \frac{\delta \cdot b}{\omega} \cdot \left[Q(\hat{\theta}) \cdot sin\sigma + a \cdot Q'(\hat{\theta}) \cdot (sin\sigma)^2 + \frac{a^2}{2} \cdot Q''(\hat{\theta}) \cdot (sin\sigma)^3 + O(a^3) \right] \tag{2.27}$$

After a transformation of coordinates, the ES description (2.27) is in standard form to apply averaging techniques [8, 9].

Averaging

The fundamental idea for using averaging techniques [8, 9] to get stability result is as follows.

- the averaged system is simpler for analysis than the original system (motivation);
- get the stability result of the averaged system;
- guarantee the closeness between the averaged system and the original one to obtain the stability result of the orignal system.

Let us introduce the averaged system for Equation (2.27), that is,

$$\begin{aligned}\frac{d\hat{\theta}_{av}}{d\sigma} &= \frac{\delta b}{\omega}\cdot\left[Q(\hat{\theta}_{av})\cdot\frac{1}{2\pi}\int_0^{2\pi} sin\sigma\,\mathrm{d}\sigma + a\cdot Q'(\hat{\theta}_{av})\cdot\frac{1}{2\pi}\int_0^{2\pi}(sin\sigma)^2\,\mathrm{d}\sigma\right]\\ &\quad+\frac{\delta b}{\omega}\cdot\left[\frac{a^2}{2}\cdot Q''(\hat{\theta}_{av})\cdot\frac{1}{2\pi}\int_0^{2\pi}(sin\sigma)^3\,\mathrm{d}\sigma + O(a^3)\right]\\ &= \frac{\delta b}{\omega}\cdot\left[a\cdot Q'(\hat{\theta}_{av})\cdot\frac{1}{2\pi}\int_0^{2\pi}\frac{1-\cos 2\sigma}{2}\,\mathrm{d}\sigma + O(a^3)\right]\\ &= \frac{ab\delta}{2\omega}\cdot[Q'(\hat{\theta}_{av}) + O(a^2)] \end{aligned} \tag{2.28}$$

Transform (2.28) into the original t time coordinate:

$$\frac{d\hat{\theta}_{av}}{dt} = \frac{d\hat{\theta}_{av}}{d\sigma}\cdot\frac{d\sigma}{dt} = \omega\cdot\frac{d\hat{\theta}_{av}}{d\sigma} = \omega\cdot\frac{ab\delta}{2\omega}\cdot[Q'(\hat{\theta}_{av}) + O(a^2)] \tag{2.29}$$

Equation (2.29) shows that there exists a^* depending on the mapping Q such that when $a < a^*$, the term $Q'(\hat{\theta}_{av})$ in (2.29) dominates over $O(a^2)$, and (2.29) can be approximated as:

$$\frac{d\hat{\theta}_{av}}{dt} = \frac{a\cdot b\cdot\delta}{2}\cdot Q'(\hat{\theta}_{av}) \tag{2.30}$$

which is asymptotically stable (please see [29] for the supporting evidence regarding the conclusion from Equation (2.29) and (2.30)). Next step is to check the closeness between the original system (2.27) and the averaged one (2.29).

Given parameter a in Equation (2.27), the terms in the bracket are always bounded. By tuning $\frac{\delta b}{\omega} > 0$ sufficiently small, $\frac{d\hat{\theta}}{d\sigma}$ could be sufficiently small and $\hat{\theta}$ changes slowly

compared with $\sin\sigma$. Therefore, the evolution of $\hat{\theta}$ in (2.27) can be approximated by the evolution of its averaged system (2.29).

As mentioned at the beginning of Section 2.2.2, ES is a way of implementing gradient driven optimization algorithm; it is clear from the deduction that ES obtains the gradient information by using the demodulation of dither signals, and the step size in this ES scheme is $\frac{a\cdot b\cdot\delta}{2}$.

To sum up for the static case, there exists $a^* > 0$ such that for any $0 < a < a^*$, there exists $(\frac{\delta b}{\omega})^*(a) > 0$ such that when $0 < \frac{\delta b}{\omega} < (\frac{\delta b}{\omega})^*$, the stability of the ES scheme is guaranteed. In other words, $\hat{\theta}$ will converge into a small neighborhood of θ^* (due to the averaging error and the approximation from Equ.(2.29) to Equ.(2.30)), the convergence speed of which depends on a, b, δ and the mapping Q. The smaller the parameters (a, b, δ), the slower the convergence speed and the longer the convergence time. Please see [29] for details of the supporting evidence for the statement.

Dynamic plant

For the dynamic plant case, there are detailed ES stability proofs (local stability [25], non-local stability [29]). The difference between the dynamic case and the static case is that singular perturbation technique [8] is used in order to handle the dynamics of the plant. The main idea of singular perturbation in the context of ES is that the tuning parameter ω, frequency of dither signal, should be sufficiently small so that when the dither signal changes slowly enough, the dynamic plant can be approximated as a static mapping, or in other words, the dynamics of the plant is always faster than the updating of ES. When the dynamic plant is approximated as a static mapping, the rest of the proof is the same as the static proof (Taylor expansion & averaging).

The tuning guidelines for the dynamic case (in sequence) are:

1. a is sufficiently small so that Euqation (2.30) is an acceptable approximation;

2. $\frac{b\cdot\delta}{\omega} > 0$ is sufficiently small (closeness between original system and the averaged system); δ is the gain of integral section and b is the amplitude of dither signal;

3. ω, frequency of dither signal, is sufficiently small compared to the dynamics of the plant (implementation of singular perturbation).

Please note that the ES scheme discussed here is used for searching a maximum; for searching a minumum, the sign of integral gain needs to be changed from "δ" to "$-\delta$" ($\delta > 0$).

Multi-variable ES

For multivariable optimum searching ($\theta^* = [\theta_1^*, \cdots, \theta_m^*]^T,\ m \in N,\ m \geq 2$), the references [5, 26] gives the tuning guideline for multivariable extremum seeking. Compared with SISO ES case, an additional requirement is added, that is, dither frequencies should satisfy:

$$\omega_i \neq \omega_j \ and \ \omega_i + \omega_j \neq \omega_k$$

for all i, j, k ranging from 1 to m, which is the requirement of P.E. condition [39–42].

2.2.3 Example

Let us illustrate the principle and tuning of ES via a system with a unique global maximum. Consider a system (the example is adapted from the example in [29]):

$$\dot{x}_1 = -x_1 + u_1^2 + 4u_1 \qquad x_1(0) = 2 \tag{2.31}$$

$$\varepsilon \dot{x}_2 = -x_2 + 2u_2 \qquad \varepsilon = 0 \tag{2.32}$$

$$y = -(x_1 + 4)^2 - (x_2 - 2)^2 \tag{2.33}$$

It is clear that when $[x_1,\ x_2]^T = [-4,\ 2]^T$, y reaches its global maximum $y^* = 0$. Let control input $u = [u_1,\ u_2]^T = [\theta_1,\ \theta_2]^T$, we have $[\theta_1^*,\ \theta_2^*]^T = [-2,\ 1]^T$ and $y^* = 0$. The initial value is $[\theta_1,\ \theta_2]^T = [0,\ 0]^T$, which is away from θ^*. The structure of the ES scheme is shown in Figure 2.9.

This example is interesting in that the plant is a multi-variable ES case, consisting of both dynamic example and static example. The state x_1 can be treated as the dynamic ES case while the state x_2 is regarded as static ES case. By observing the convergence difference between x_1 and x_2, we can illustrate the difference bewteen static ES case and dynamic ES case, i.e., the convergence time for the static case can always be shortened by a better set of tuning parameters while the convergence time for the dynamic case is always limited by the uncertainty of the dynamics of the plant (black-box) which results in the conservative adaptation of the ES scheme. Let us follow the tuning strategy (in sequence), as is mentioned in Chapter 3.1.2 and the specific values are shown in Table 2.1.

1. set $a_1,\ a_2$ sufficiently small ($a_1 < a_1^*$ & $a_2 < a_2^*$);
2. make $\frac{b_1 \cdot \delta_1}{\omega_1}$ and $\frac{b_2 \cdot \delta_2}{\omega_2}$ sufficiently small;

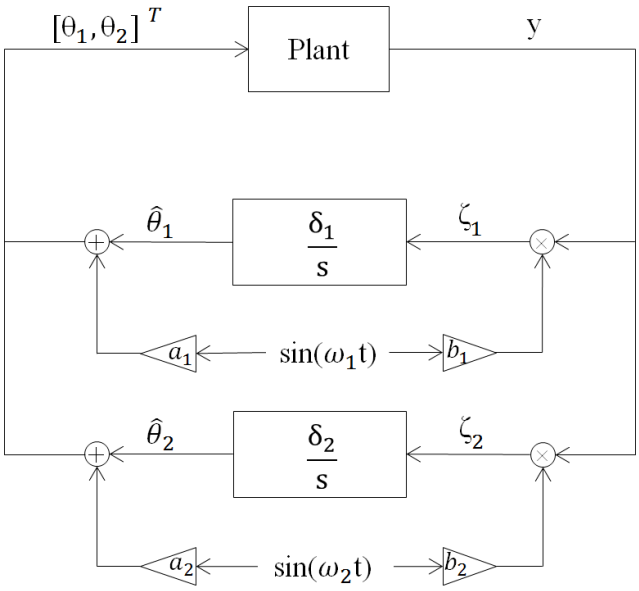

Fig. 2.9 Multi-variable ES structure

3. Choose ω_1 sufficiently small ($\omega_1 < \omega_1^*$) and $\omega_2 \neq \omega_1$.

Parameter	Value	Unit
ω_1	0.1	rad/sec
ω_2	50	rad/sec
$a_1,\ a_2$	0.1	N/A
δ_1	0.1	N/A
δ_2	0.5	N/A
$b_1,\ b_2$	1	N/A

Table 2.1 Table of ES parameters

The corresponding results are shown in Figure 2.10.

As is shown in Figure 2.10, plant output literally reaches its optimum y^* and $[\theta_1,\ \theta_2]^T$ converges into a small neighborhood of θ^*. For the convergence performance of the static case (state x_2), we could shorten the convergence by increasing δ_2 from 0.5 to 5 and ω_2 from 50 to 500. The corresponding results are shown in Figure 2.11. In Figure 2.10, ES takes around 20 seconds to drive the state x_2 into a small neighborhood of its optimality whereas for the plant, ES takes around 5 seconds (shown in Figure 2.11) to drive the state x_2 into a small region of its optimality from the same intial condition.

As is shown in Figure 2.11, the convergence speed of ES for static case can be increased by increasing step size and dither frequency, provided that $\frac{b_2\delta_2}{\omega_2}$ is still sufficiently small. For the dynamic case (state x_1), since ω_1 is limited by ω_1^* (singular perturbation), step size $\frac{a_1 b_1 \delta_1}{2}$ is also restricted by the value of $\frac{b_1\delta_1}{\omega_1}$. If forcibly increase the step size for the state x_1 like increasing δ_1 or b_1, ES bcomes unstable and the system blows up. An example is shown in Figure 2.12, when increasing δ_1 from 0.1 to 1 and ω_2 from 0.1 to 1.

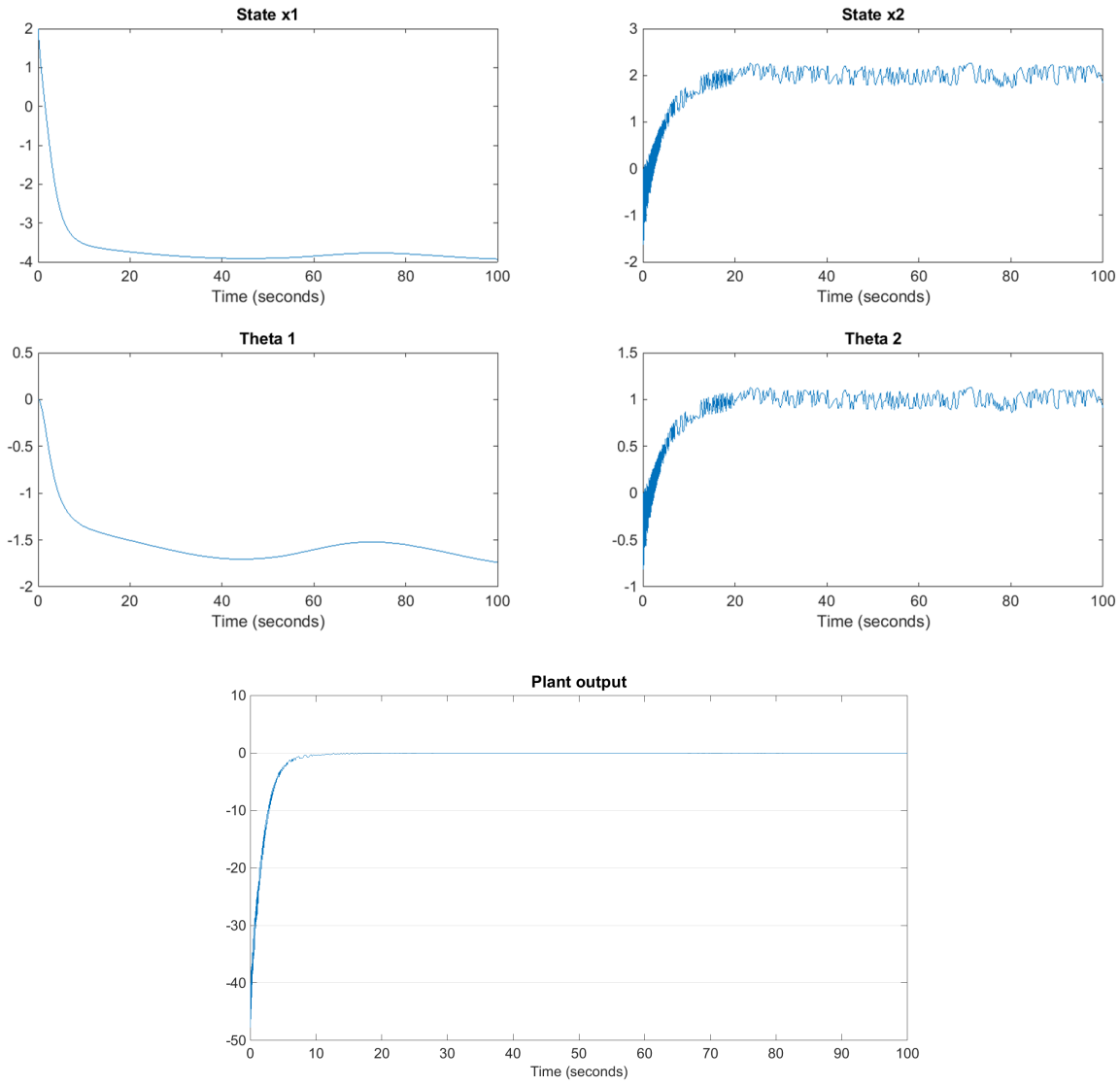

Fig. 2.10 Convergence result of the ES process

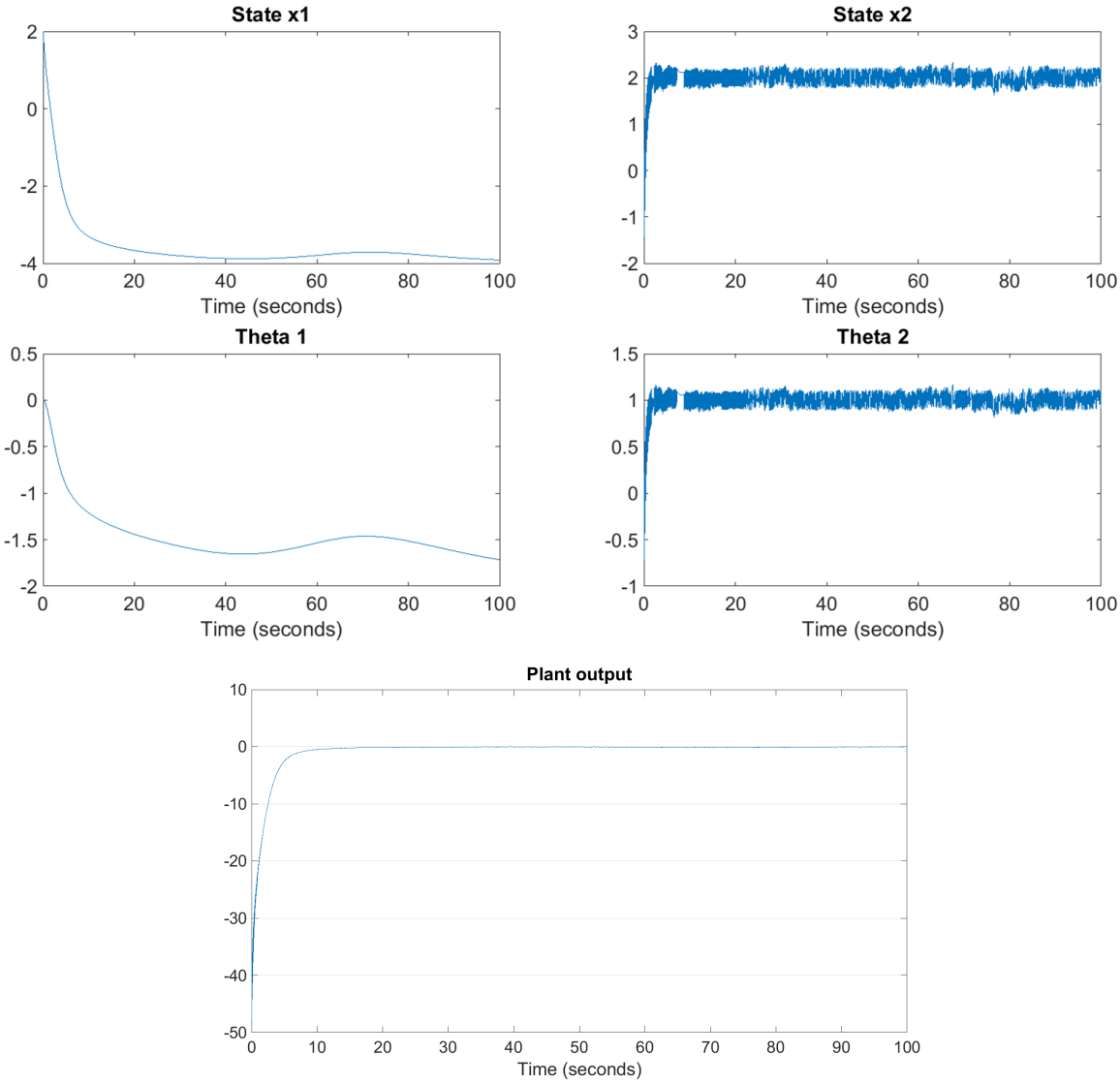

Fig. 2.11 Convergence result of the ES process

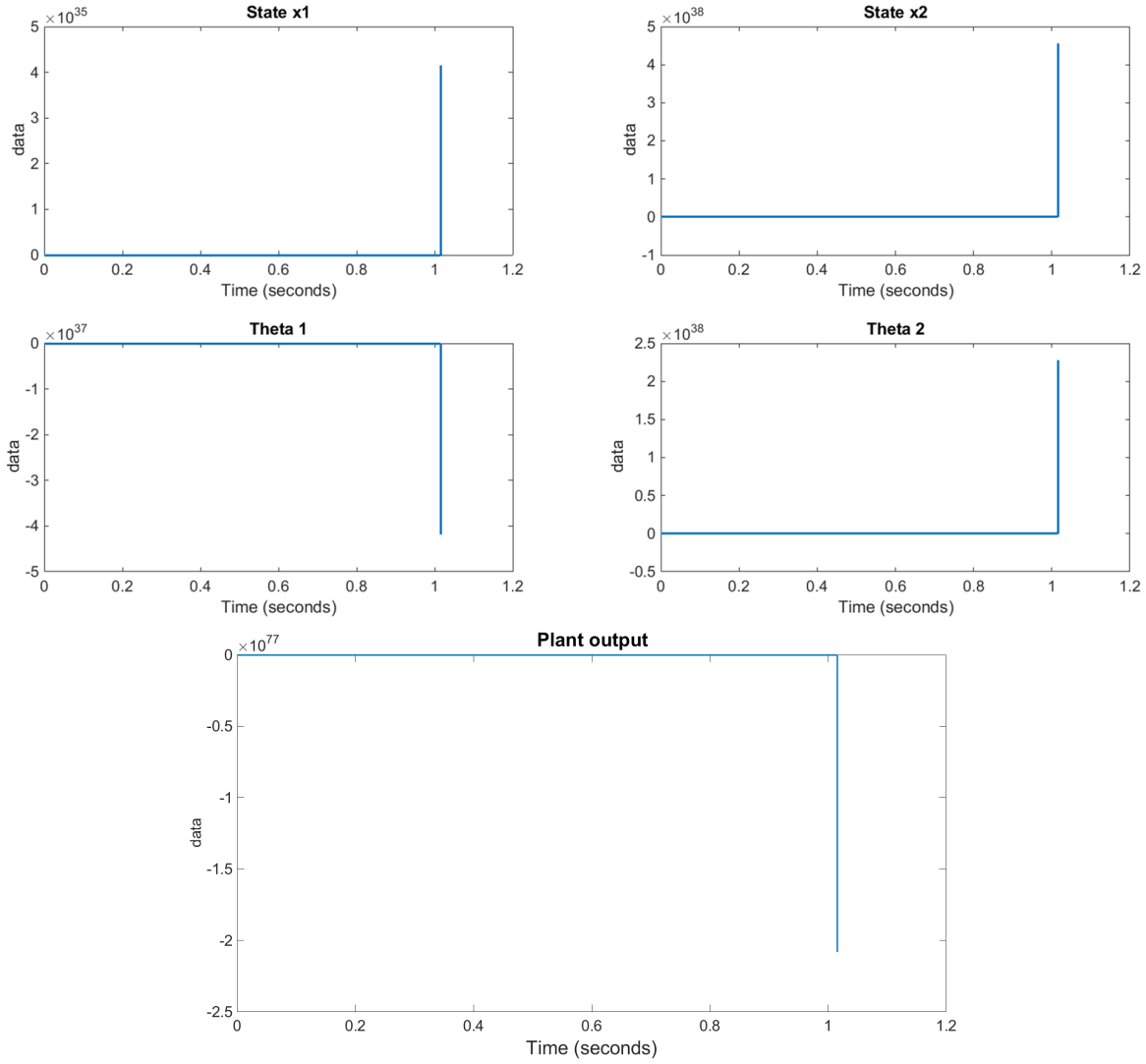

Fig. 2.12 Result of the unstable ES process

2.2.4 Application to the project

Based on the model (2.12, 2.13) and Assumption 2.2.1, this on-line model calibration problem is a static-mapping minimization problem.The fundamental structure of black-box extremum seeking for searching local optimum (e.g. [25, 29]) is shown in Figure 2.13.

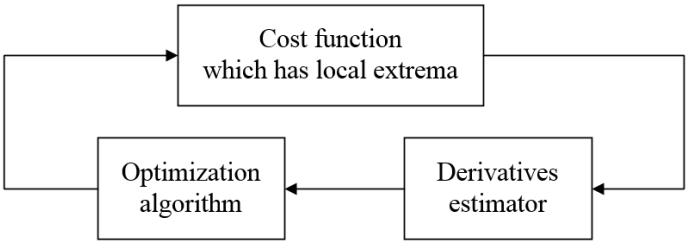

Fig. 2.13 Fundamental structure of the ES

The inputs & outputs of the cost function which depends on the engine plant may be measured but no analytical expression of the engine is available. The design of a black-box extremum seeking consists of designing an estimator of derivatives and an appropriate optimization algorithm.

This project utilizes the discrete type, perturbation based extremum seeking scheme [25, 28–31]. The structure of the extremum seeking scheme is illustrated in Figure 2.14.

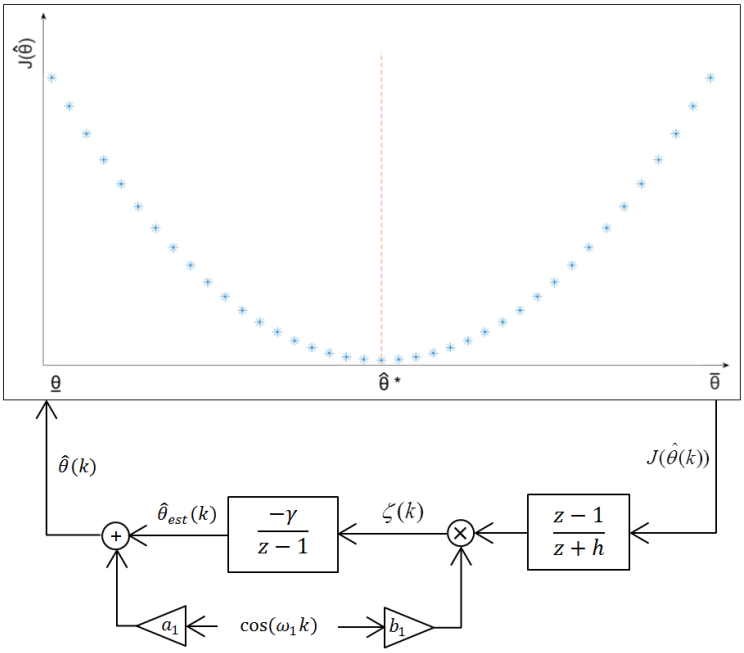

Fig. 2.14 Structure of ES scheme

The cost function is designed as below:

$$J(\theta_{SOC}(K)) = (\frac{Model\ IMEP(K) - Measured\ IMEP}{Normalized\ IMEP})^2 \tag{2.34}$$

where the parameter K means the K^{th} iteration of ES in computer (ECU) domain. Normalized IMEP is a constant used for normalizing the cost function. Measured IMEP is an averaged IMEP value over the last 10 engine cycles. During each 10-engine-cycle time interval, ES can have 40 calculation cycles in the ECU [24].

Due to the feature of this project, that is, engine control unit (ECU) is discrete controller and all computational tasks including ES algorithm are implemented in the ECU, discrete type extremum seeking is utilized for this project.

The tuning parameters of this ES optimization scheme are ω_1 (frequency of the perturbation signal), a_1, b_1 (amplitude of the perturbation signal) and δ_1 (gain of the integral action). The initial condition of θ_{SOC}, used for ES updating calculation, is pre-defined from experimental data for each operating point . It forms part of the off-line determined data look-up table. The range of $\hat{\theta}^*$ is due to the physical structure of engine cylinder, please see the explanation for the engine model in Chapter 2.1.

In order to achieve the control objective of the project, it is supposed to develop an ES method that can track the quick change of the parameter θ_{SOC}. In particular, we need to establish a time-scale separation as described in Figure 2.15: the ES dynamics are sandwiched in between the faster regular engine dynamics and the slower variation of the engine parameter.

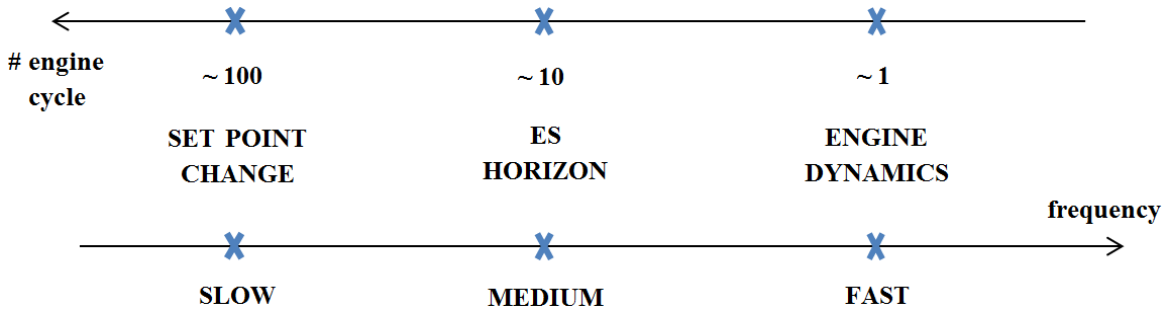

Fig. 2.15 Time-scale design in the engine project

Normal tuning practice

In order to guarantee ES stability and have small residual oscillation, the normal practice for ES tuning is to keep a_1 (dither amplitude) and $a_1 \cdot b_1 \cdot \delta_1$ sufficiently small, as the collaboration group did in [24]. The result in [24] is shown in Figure 2.16 & Figure 2.17.

Figure 2.16 and Figure 2.17 show the model calibration process for 8 operating points, with each operating point having different plant output measurement y_1. As is seen from Fig. 2.16, model output $\hat{y}_1$ (IMEP) cannot follow the engine measurement value y_1 and ES cannot finalize model calibration process in 40 searching cycles for each operating points. Based on this normal practice and their choices, ES needs 2000∼4000 cycles of iteration (25 seconds to 50 seconds in reality) to finalize the calibration process for each of the experimental engine's operating points, as is illustrated in Figure 2.17.

Deficiency

Considering the real engine operation process, the model calibration convergence speed for this normal tuning practice (results in [24]) is too slow to be applied in reality. A method that can shorten the ES convergence duration is essential to this engineering project.

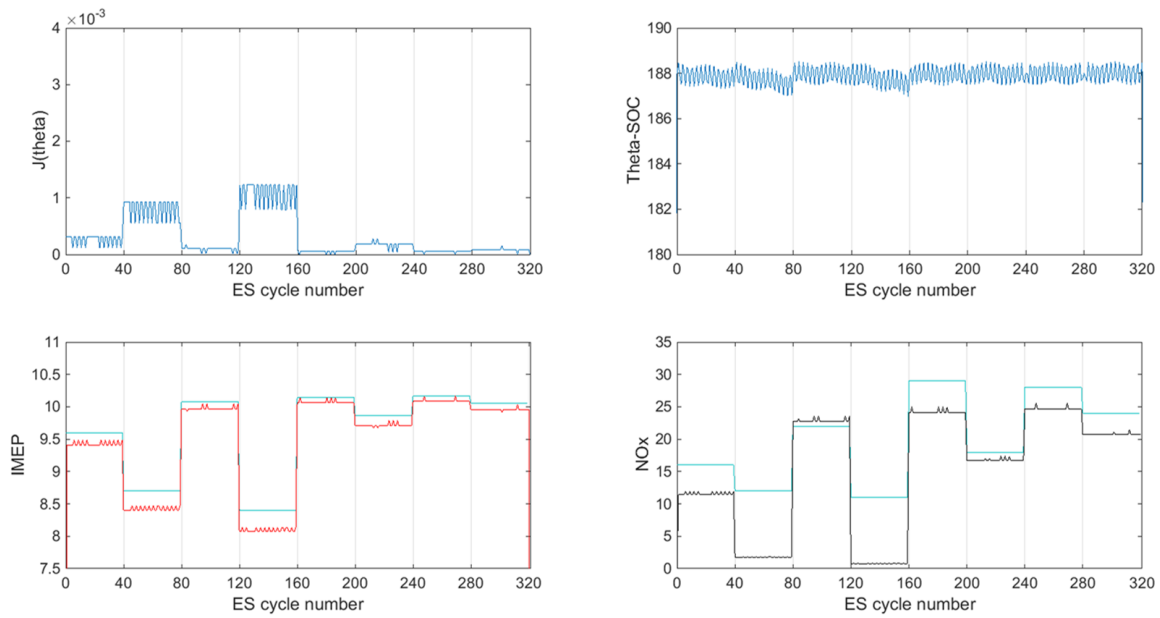

Fig. 2.16 Model calibration simulation result (40 searching cycles for each operating point)

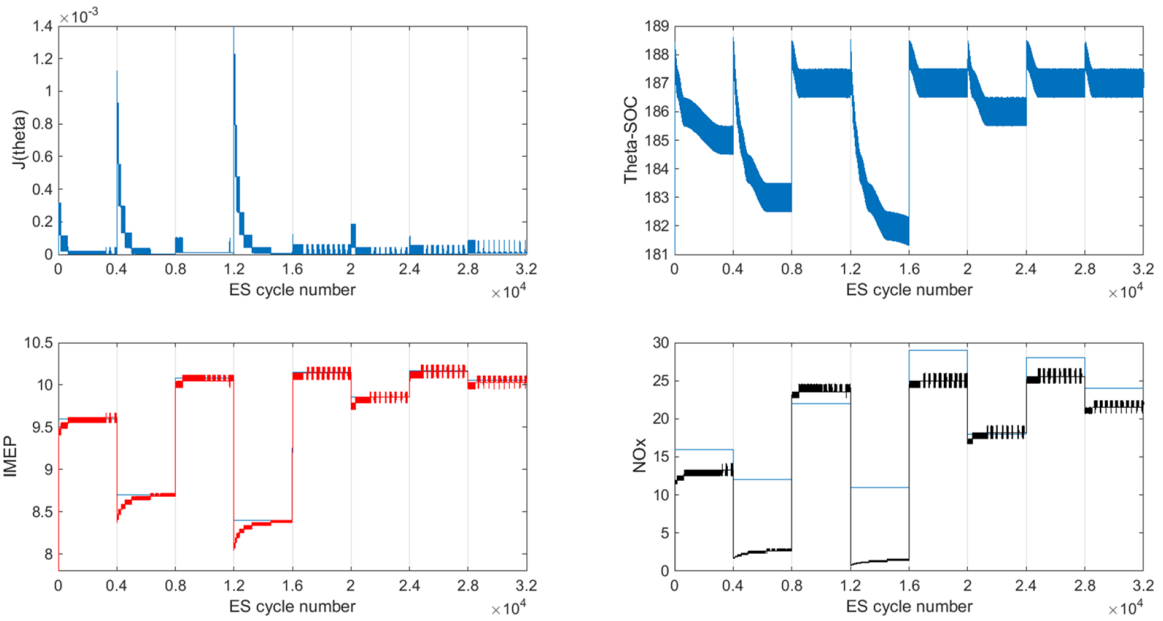

Fig. 2.17 Model calibration simulation result (4000 searching cycles for each operating point)

Existing improvement method

A existing method for shortening ES convergence duration is to use annealing-like algorithms, such as exponentially decaying dithering method, which could have faster convergence and smaller residual oscillation.

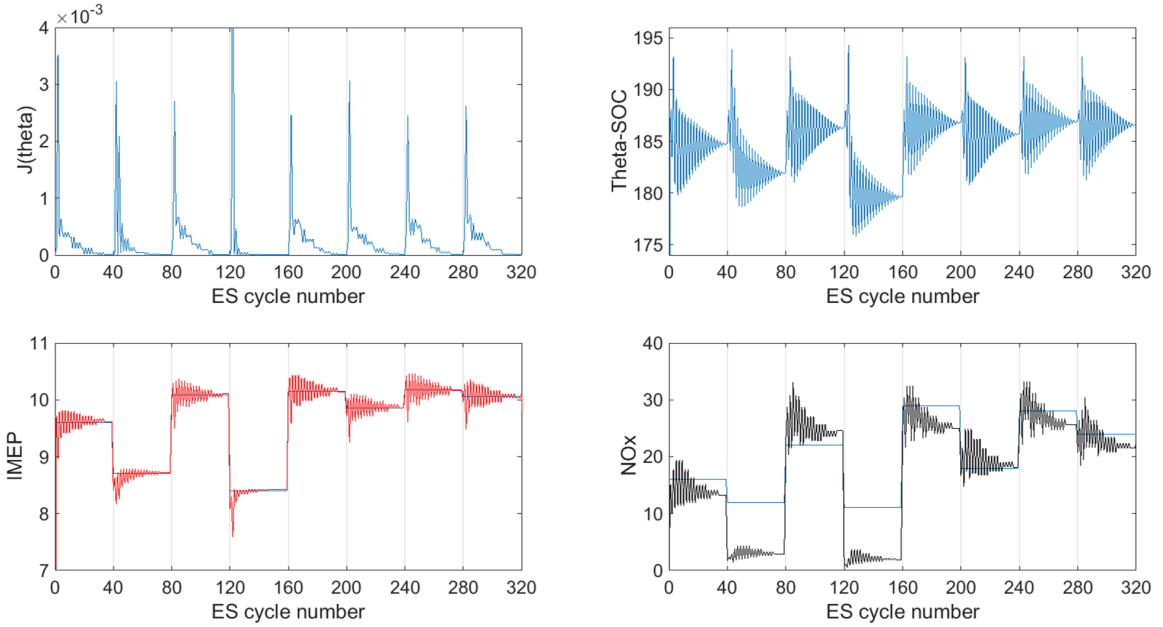

Fig. 2.18 Exponentially decaying dithering result (40 searching cycles)

The fundamental principle of this method is to start with a larger dithering amplitude $\overline{a}$ to get a reatively larger step size $\frac{a \cdot b \cdot \delta}{2}$ and faster convergence (cf. Chapter 2.2.2) and end up with a smaller dithering amplitude $\underline{a}$ to have a smaller residual oscillation. The corresponding ES convergence result is shown in Figure 2.18.

As is shown in Figure 2.18, model output IMEP can follow the engine measurement IMEP value within 40 ES cycles (half a second); the improved ES performance is much better than the original result in the paper, which helps reduce the overall control response from minutes to seconds.

Deficiency

This existing method, however, has bad convergence performance when there exists a relatively flat area in the cost function map, which often happens in practice. For example,

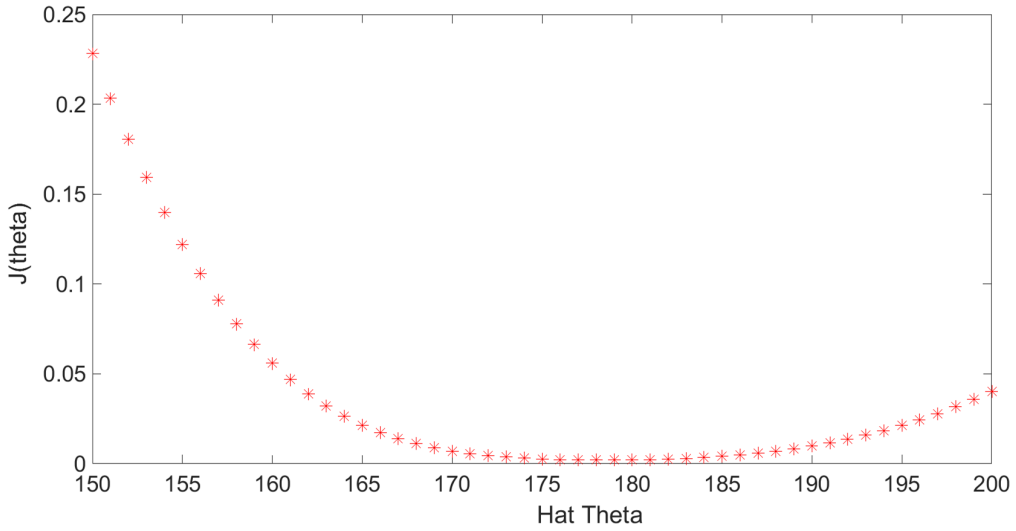

Fig. 2.19 Cost function map with flat area

Figure 2.19 is a cost function map for one of experimental engine's operating points (operating condition: Speed 1489 rpm, IMEP 9.6 bar, Torque 81.1 Nm, MAF 6.09 g/s, BSFC 85.8 g/kw-hr, Fuel Flow Rate 24.3 mg/cycle, EGR 50.3 C, Boost Pressure 1.019 bar, BMEP 20.39 bar). When using the exponentially decaying dithering method for this kind of operating condition where there exists a relatively flat area in the cost function map, the corresponding ES convergence process is illustrated in Figure 2.20.

As is shown in Figure 2.20, when using the decaying method for eight randomly select operating points whose cost function maps have a flat area (the eight operating points are

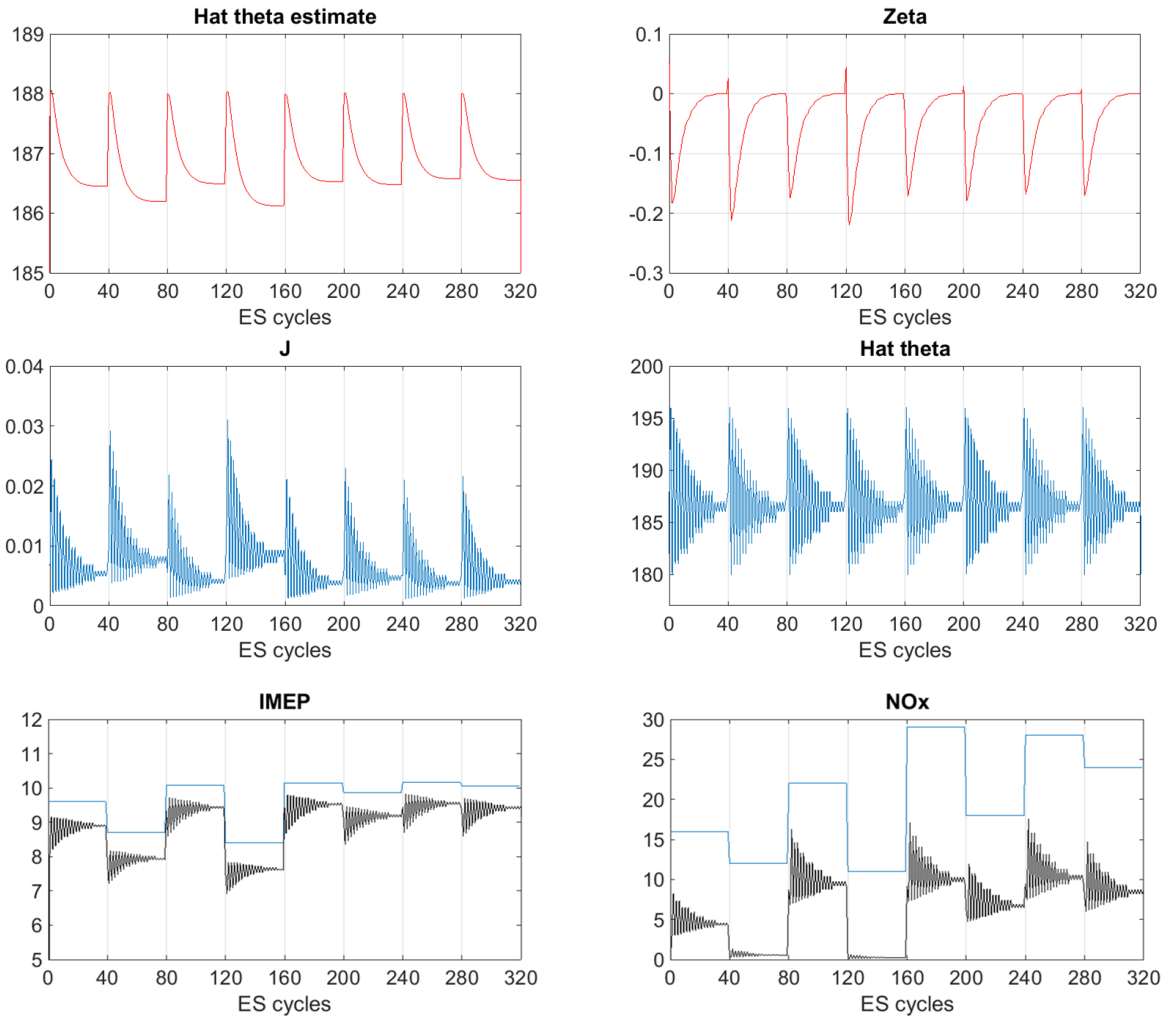

Fig. 2.20 Exponentially decaying dithering result for cost function map with flat area

all around the same speed 1500 rpm, thus their $\hat{\theta}^*$ are literally the same), $\hat{\theta}_{est}$ only reaches somewhere between 185.5 degree and 187.5 degree, while the $\hat{\theta}^*$ is between 178 degree and 179 degree (cf. cost function map Fig. 2.19 which is also obtained from around 1500 rpm). The method brings an error of 7 to 9 degrees, causing a big deviation for $\hat{y}_2$ (NOx) and $\hat{y}_3$ (Eff). Since y_2, y_3 cannot be measured on-line in product engines, which is the reason why $|y_2 - \hat{y}_2|$ and $|y_3 - \hat{y}_3|$ are not incorporated into the cost function, the existing method needs further improving in order to have better convergence performance.

Cause of deficiency

The reason why the existing improvement method does not work for this case is that ES is an implementation of gradient-driven algorithm. When there exists a flat area in the cost function map, the gradient information within that area is so weak that ES does not have adequate "energy" to drive the parameter further quickly. Considering ES updating Equation (2.30), the updating of $\hat{\theta}$ goes vey slowly when the cost function gradient is vey small, and the overall convergence of ES becomes quite slow within the flat area, as is shown in Figure 2.20 that $\hat{\theta}_{est}$ stays almost constant when ζ approaches zero ($\hat{\theta}$ enters into the flat area in Figure 2.19).

One problem is to find a method that improves the convergence of the ES for the nonlinear mapping case. This will be explained in Section 3.2.

2.3 Delay identification

2.3.1 Engine project

A noticable issue for the engine control design is the presence of input time delay (exhaust gas recirculation (EGR) delay (transmission of gas) or injection lag (transmission of fluid)). Although the delay time in the engine is subtle in time domain, the delay would be noticable in engine cycle domain since engine operation for each cycle is faster compared to the delay time, as is shown in Figure 2.5 in Section 2.1. There are existing methods with regard to compensate time delay for control purpose, but the time delay in IC engine varies according to different operating conditions. The current engine industry use the method of tabulating delay time for different operating points. However, due to the huge amount of operating conditions in reality, off-line tabulating method can be both consuming and inadequate. Furthermore, an inexact off-line approximation of delay time could cause troubles to a dynamic plant in terms of stability and oscillations [49, 54]. How to get the value of time delay efficiently and accurately would be a topic of interest for both engine industry and theory development.

2.3.2 Brief literature review

There are many articles in the literature (e.g. [43–48, 50–53, 55–68]) with regard to delay identification in the area of control engineering. This problem has drawn considerable attention in the last three decades, since for a large class of systems, like systems in biology, chemistry, physics, economics, population dynamics, as well as in engineering sciences, time delay is either included in their inner dynamics or introduced by sensors or actuators. The early approaches of control engineering, for the sake of simplicity, assume that the delay time is either priori known or approximated. The corresponding delay compensation method is then utilized for designing the feedback control law. An example of the delay compensation diagram is shown in Figure 2.21. One of the assumptions is that $\hat{D} \approx D$. However, an inexact

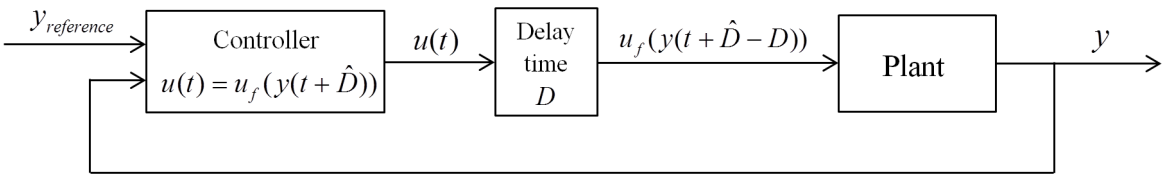

Fig. 2.21 An example of delay compensation scheme

off-line approximation of time delay in a system may cause undesired performance of the closed loop scheme in terms of stability and oscillation, as is dicussed in the two overview papers [49, 54] regarding delay effects on stability and time-delayed systems. Therefore,

on-line identification of time delay becomes a topic of interest for a wide range of researchers and engineers. An example of delay identification scheme is illustrated in Figure 2.22.

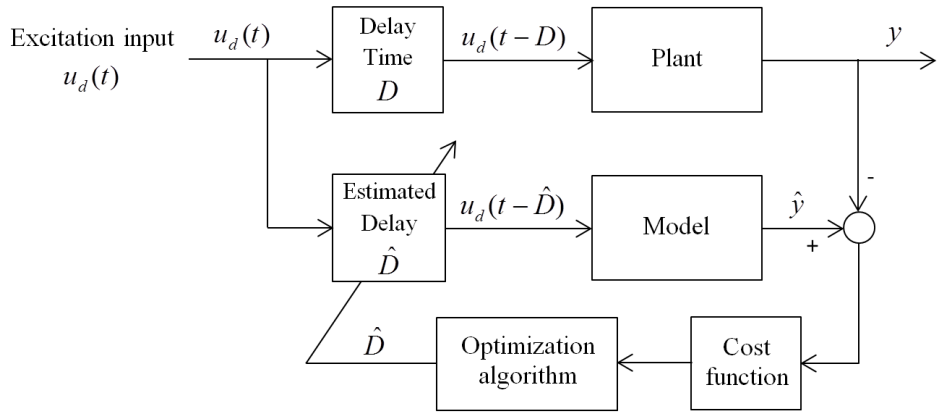

Fig. 2.22 An example of delay identification scheme

According to [68], the existing literature with regard to delay identification problem can be classified based on

(1). the type of model: continuous-time model or discrete-time model.

(2). the type of solution: linear methods based on the solution of a set of linear equations, or non-linear methods based on an optimization procedure.

(3). the type of data processing: batch-type or recursive algorithm.

The existing literature introduced and designed many different methods to on-line identify the delay in a system, such as identifying the delay by a rational transfer function (Pade approximation) [44, 45], using basic linear regression to identify delay [61], utilizing Levenberg–Marquardt algorithm (damped least-squares) to estimate delay in a nonlinear system [51], estimating the delay through correlation function analysis [53], using cross-correlation technology to identify delay in the presence of colored noise [47], designing a complex regression vector and utilizing gradient algorithm to deal with delay identification problem [68], etc.

2.3.3 Research gap

Although many efforts have been devoted into this research area during the past, there are, however, two gaps in the literatue regarding time-delayed system and time delay identifica-

tion, both of which are what we will solve in the rest of this chapter.

The first research gap is that there is few literature regarding time-delayed system that has considered combining delay identification process and closed-loop control scheme to work simultaneously. Since delay identification process is always characterized by fast transient variation and oscillation of the identification parameter $\hat{D}$ (cf. Fig. 2.22), which will cause undesired performance of the closed loop scheme (e.g. $u(t) = u_f(y(t+\hat{D}))$) in terms of stability and oscillation, the system stabilization process is normally separated from the identification process. The stabilization control law begins to work only after the delay identification process finishes, which is similar to a $0-1$ switch process. In the literature, there is one paper [57] that has considered making delay identification process and feedback stabilization scheme work simultaneously. However, the paper [57] only considers using Smith predictor as the control scheme and the corresponding controller parameters are tuned by trial-and-error. In other words, even if the parameter value of the Smith controller tuned by trial-and-error in [57] works for a large range of possible delay time and for a large class of plants, there is no design guideline for a general feedback control law, like the control law in this engineering project $u = u(\hat{x}, \hat{D}, \lambda)$, that can make closed stabilization loop work simultaneously with delay identification process.

The second observation is that few papers in the literature consider delay identification with extremum seeking. In the literature, least-square (linear regression, damped least-squares, etc.) is very popular as the optimization scheme (cf. Fig. 2.22) in delay identification problem while no one has ever used extremum seeking to solve the identification problem. Since extremum seeking and least-square are essentially the same, both of which are gradient-driven algorithm, extremum seeking should at least have the same optimization effect as least-square, or could even be better if considering the improvement method we proposed in the former subchapter regarding shortening convergence duration of the ES.

In order to fill these two gaps, the work in Chapter 3.3 utilizes extremum seeking method to on-line identify the time delay in a system and proposes a general design guideline that makes a general stabilization controller works simultaneously with the unknown delay identification process and that the identification process has little influence on the overall closed-loop control scheme in terms of stability and oscillation.

Chapter 3

Work of the thesis

3.1 The effect of delay on the ES

For perturbation-based adaptive ES, there are articles [25, 29] in the literature that proved the stability of the ES. Recently, several papers [36–38] proposed delay compensation method for the ES when a time delay is present in the closed loop of the ES. Between these two stages, there are few papers that have analysed the effect of delay on the working of the ES. Here is the mathematical explanation about the effect of delay on the working of the ES in terms of stability and convergence. Consider a smooth single-input-single-output static system $Q(\theta)$ with an unknown time delay D.

Problem Given Assumption 2.2.1, find θ^* in the presence of unknown output delay D, from only measurements of $y(t)$ and $\theta(t)$.

Diagram of the ES

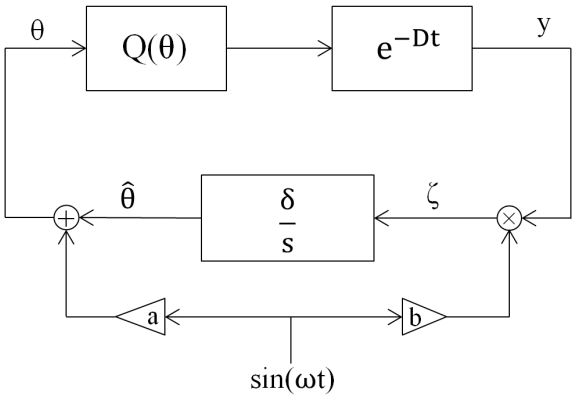

Fig. 3.1 ES for static mapping with output delay section

A simple diagram of the ES scheme with time delay is shown in Figure 3.1. It indicates that the closed loop dynamics satisfy:

$$y = Q(\theta(t-D)) = Q\left(\hat{\theta}(t-D) + asin(\omega(t-D))\right) \tag{3.1}$$

$$\zeta = b\sin(\omega t)\cdot y = b\sin(\omega t)\cdot Q\left(\hat{\theta}(t-D) + a\sin(\omega(t-D))\right) \tag{3.2}$$

$$\dot{\hat{\theta}} = \delta\cdot\zeta = \delta\cdot b\sin(\omega t)\cdot Q\left(\hat{\theta}(t-D) + a\sin(\omega(t-D))\right) \tag{3.3}$$

where a, b are the amplitudes of the dither, ω is the frequency of the dither and δ is the gain of integral action.

Result (adapted from [29])
Given any mapping Q (meeting Assumption 2.2.1) and unknown delay D, there exists positive ω^* ($\omega^* D = \frac{\pi}{2}$) such that for any $0 < \omega < \omega^*$, there exists $a^*(\omega) > 0$ such that for any $0 < a < a^*$, there exists $(\delta\cdot b)^*(a)$ such that when $0 < (\delta\cdot b) < (\delta\cdot b)^*$, the stability of ES for the delayed system is guaranteed.

The following analysis gives a more complete understanding of the result. It can be seen from the analysis that compared with the undelayed case discussed in Section 2.2.2, the convergence of ES is longer for the same system with an extra unknown time delay because the time delay further restricts the value of the ω & $\delta\cdot b$, which decreases the updating step size $\frac{ab\delta}{2}$ in time t domain and thus prolongs the convergence of ES.

Analysis
By using Taylor series expansion for $Q(\hat{\theta}(t-D) + a\sin(\omega(t-D)))$, it leads to

$$\begin{aligned} &Q(\hat{\theta}(t-D) + a\sin(\omega(t-D))) \\ &= Q(\hat{\theta}(t-D)) + Q'(\hat{\theta}(t-D))\cdot a\sin(\omega(t-D)) + \frac{1}{2}\cdot Q''(\hat{\theta}(t-D))\cdot(a\sin(\omega(t-D)))^2 \\ &\quad + O(a^3) \end{aligned} \tag{3.4}$$

Substituting (3.4) into (3.3) yields

$$\begin{aligned} \dot{\hat{\theta}} = \delta\cdot b\sin(\omega t)\cdot\left[Q(\hat{\theta}(t-D)) + a\cdot Q'(\hat{\theta}(t-D))\cdot\sin(\omega(t-D))\right] \\ + \delta\cdot b\sin(\omega t)\cdot\left[\frac{a^2}{2}\cdot Q''(\hat{\theta}(t-D))\cdot(\sin(\omega(t-D)))^2 + O(a^3)\right] \end{aligned} \tag{3.5}$$

We can re-write (3.5) into a new time scale $\sigma = \omega\cdot t$:

$$\frac{d\hat{\theta}}{dt} = \frac{d\hat{\theta}}{d\sigma}\cdot\frac{d\sigma}{dt} = \omega\cdot\frac{d\hat{\theta}}{d\sigma} = \delta\cdot\zeta = \delta\cdot b\sin\sigma\cdot Q(\hat{\theta}(\sigma-\omega D)+asin(\sigma-\omega D))$$

i.e.

$$\begin{aligned}\frac{d\hat{\theta}}{d\sigma} = {} & \frac{\delta}{\omega}\cdot b\sin\sigma\cdot\left[Q(\hat{\theta}(\sigma-\omega D))+a\cdot Q'(\hat{\theta}(\sigma-\omega D))\cdot sin(\sigma-\omega D)\right] \\ & +\frac{\delta}{\omega}\cdot b\sin\sigma\cdot\left[\frac{a^2}{2}\cdot Q''(\hat{\theta}(\sigma-\omega D))\cdot(sin(\sigma-\omega D))^2+O(a^3)\right]\end{aligned} \tag{3.6}$$

By using trigonometric identity for $\sin\sigma$, it leads to

$$\begin{aligned}\sin\sigma = {} & \sin[(\sigma-\omega D)+\omega D] \\ = {} & \sin(\sigma-\omega D)\cos\omega D+\cos(\sigma-\omega D)\sin(\omega D)\end{aligned} \tag{3.7}$$

Substituting (3.7) into (3.6) yields

$$\begin{aligned}\frac{d\hat{\theta}}{d\sigma} = {} & \frac{\delta b}{\omega}\cdot\cos\omega D\cdot\left[Q(\hat{\theta}(\sigma-\omega D))\sin(\sigma-\omega D)+a\cdot Q'(\hat{\theta}(\sigma-\omega D))\cdot(sin(\sigma-\omega D))^2\right] \\ & +\frac{\delta b}{\omega}\cdot\cos\omega D\cdot\left[\frac{a^2}{2}\cdot Q''(\hat{\theta}(\sigma-\omega D))\cdot(sin(\sigma-\omega D))^3+O(a^3)\right] \\ & +\frac{\delta b}{\omega}\cdot\sin\omega D\cdot\left[Q(\hat{\theta}(\sigma-\omega D))\cos(\sigma-\omega D)+a\cdot Q'(\hat{\theta}(\sigma-\omega D))\cdot\sin(\sigma-\omega D)\cos(\sigma-\omega D)\right] \\ & +\frac{\delta b}{\omega}\cdot\sin\omega D\cdot\left[\frac{a^2}{2}\cdot Q''(\hat{\theta}(\sigma-\omega D))\cdot(sin(\sigma-\omega D))^2\cos(\sigma-\omega D)+O(a^3)\right]\end{aligned} \tag{3.8}$$

Next, let us look at the averaged system of Equ.(3.8):

$$\frac{d\hat{\theta}_{av}}{d\sigma} = \frac{\delta b}{\omega}\cdot\left[\cos(\omega D)\cdot\frac{1}{2\pi}\int_0^{2\pi}[\,\cdot\,]\,\mathrm{d}\sigma+\sin(\omega D)\cdot\frac{1}{2\pi}\int_0^{2\pi}[\,\cdot\,]\,\mathrm{d}\sigma\right] \tag{3.9}$$

Applying trigonometric identities to (3.9), it leads to

$$\frac{d\hat{\theta}_{av}}{d\sigma} = \frac{ab\delta}{2\omega}\cdot\cos(\omega D)\cdot\left[\,Q'(\hat{\theta}_{av}(\sigma-\omega D))+O(a^2)\,\right] \tag{3.10}$$

Apply Talyer expansion for $Q'(\hat{\theta}_{av}(\sigma-\omega D))$, it leads to

$$\begin{aligned}Q'(\hat{\theta}_{av}(\sigma-\omega D)) = {} & Q'(\hat{\theta}_{av}(\sigma))+Q''(\hat{\theta}_{av}(\sigma))\cdot(\hat{\theta}_{av}(\sigma-\omega D)-\hat{\theta}_{av}(\sigma)) \\ & +O(\hat{\theta}_{av}(\sigma-\omega D)-\hat{\theta}_{av}(\sigma))^2\end{aligned} \tag{3.11}$$

Consider Equation (3.9), for any given mapping Q, delay D and $\omega, a > 0$,

$$\hat{\theta}_{av}(\sigma - \omega D) - \hat{\theta}_{av}(\sigma) = O(\frac{\delta b}{\omega} \cdot \omega D) = O(D\delta b) \tag{3.12}$$

Substituting (3.12) into (3.11) yields

$$Q'(\hat{\theta}_{av}(\sigma - \omega D)) = Q'(\hat{\theta}_{av}(\sigma)) + Q''(\hat{\theta}_{av}(\sigma)) \cdot O(D\delta b) + O(D\delta b)^2 \tag{3.13}$$

That is, for any given delay D, mapping Q and frequency ω, there exists $a^* > 0$ such that for any $0 < a < a^*$, there exists $(\delta b)^*(a)$, such that

$$Q'(\hat{\theta}_{av}(\sigma - \omega D)) + O(a^2) \approx Q'(\hat{\theta}_{av}(\sigma)) \tag{3.14}$$

Therefore, Equation(3.10) can be approximated as:

$$\frac{d\hat{\theta}_{av}}{d\sigma} = \frac{ab\delta}{2\omega} \cdot \cos(\omega D) \cdot Q'(\hat{\theta}_{av}(\sigma)) \tag{3.15}$$

Transform (3.15) into the orignal t time coordinate:

$$\frac{d\hat{\theta}_{av}}{dt} = \frac{d\hat{\theta}_{av}}{d\sigma} \cdot \frac{d\sigma}{dt} = \frac{ab\delta}{2} \cdot \cos(\omega D) \cdot Q'(\hat{\theta}_{av}(t)) \tag{3.16}$$

Providing that $0 < \omega D < \frac{\pi}{2}$, the system (3.16) is asymptotically stable.

The next step is to check the closeness between the orignal system (3.8) and the averaged system (3.9). Regarding the original system (3.8), given Q, D, ω, a, the terms in the brackets are always bounded. Thus, we can always keep the two systems as close as possible by tuning $\frac{\delta b}{\omega}$ as small as possible. In other words, given Q, D, ω & a, there exists $(\delta \cdot b)^*$ such that when $0 < (\delta \cdot b) < (\delta \cdot b)^*$, the original system can be approximated by its averaged system.

To sum up, given any mapping Q (meeting Assumption 2.2.1) and unknown delay D, there exists positive ω^* ($\omega^* D = \frac{\pi}{2}$) such that for any $0 < \omega < \omega^*$, there exists $a^*(\omega) > 0$ such that for any $0 < a < a^*$, there exists $(\delta \cdot b)^*(a)$ (meeting the averaging approximation (3.9) and the approximation (3.14)) such that when $0 < (\delta \cdot b) < (\delta \cdot b)^*$, the stability of ES for the delayed system is guaranteed.

3.2 Convergence improvement for the ES

Let us consider a general continuous-time SISO nonlinear static mapping case and this model calibration problem is a discrete-time application of this general continuous-time case. The problem formulation is as follows.

Problem Find θ^* under the given Assumption 2.2.1 and known range $\theta^* \in [\underline{\theta}, \overline{\theta}]$, without knowing Q or Q'.

3.2.1 Proposed diagram

The proposed structure of the ES is shown in Figure 3.2.

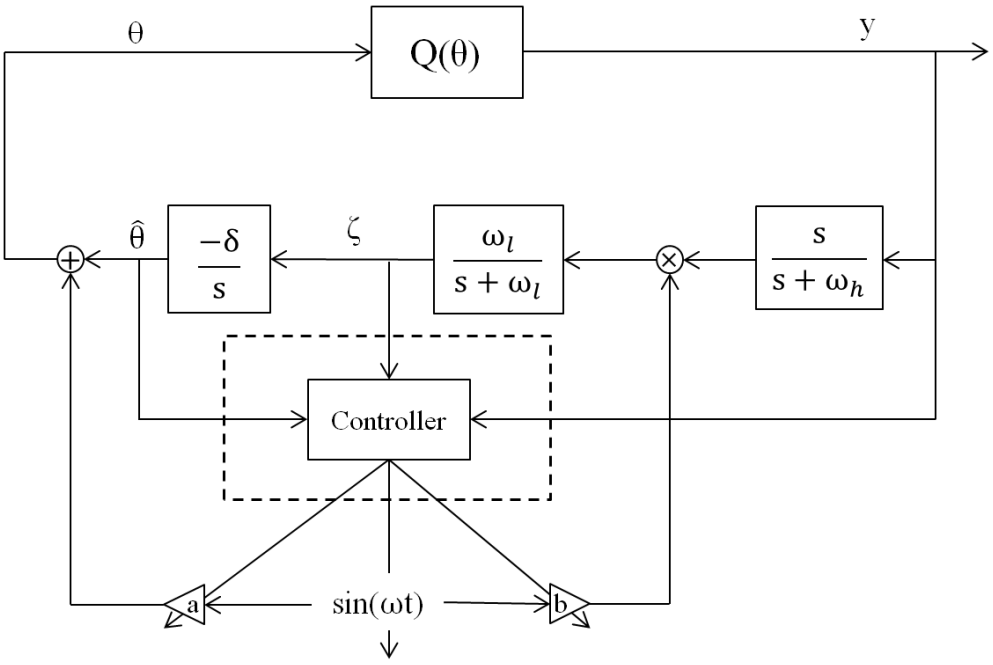

Fig. 3.2 ES with controller

Compared with the ES diagram mentioned in the former sub-chapter, this ES strucutre has three extra parts, i.e. high-pass filter, low-pass filter and a controller. There are local [25] and non-local [29] stability proof for the ES structure with low-pass and high-pass filters. The reason why we use the ES structure with filters is that the effect of environmental noise can be kept small.

My contribution is that a controller is added to the ES. The fundamental structure of the proposed diagram (Fig. 3.2) is shown in Figure 3.3.

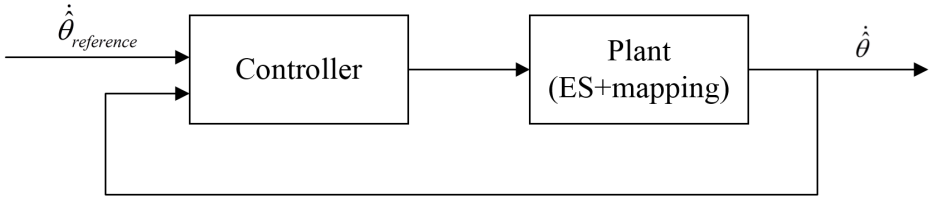

Fig. 3.3 Fundemantal structure

The output $\dot{\hat{\theta}}$ is the estimate of convergence speed of ES. The purpose of the controller is to regulate the output $\dot{\hat{\theta}}$ to a neighborhood of the reference convergence speed $\dot{\hat{\theta}}_{reference}$. The final error bound $|\dot{\hat{\theta}} - \dot{\hat{\theta}}_{reference}|$ depends on the design of the look-up table in the controller (Step 2), but this scheme could help improve the convergence performance of the ES.

3.2.2 Operation principle

The algorithm in the controller is illustrated in Figure 3.4.

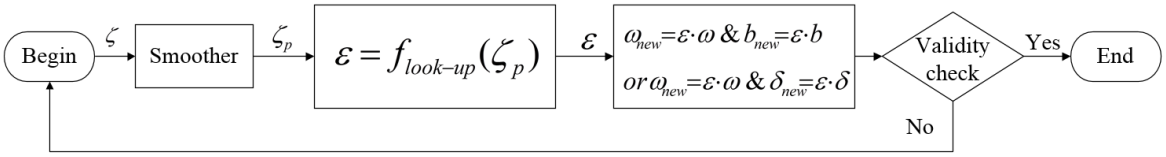

Fig. 3.4 Diagram of the algorithm

1. Observe current gradient information of the cost function, i.e. $Q'(\theta)$.
2. Give instructions to increase or decrease dither frequency ω & amplitude b or frequency ω & integral action gain δ based on the observed gradient information and contsructed look-up table.
3. Check validity. Decide whether the current value of $\hat{\theta}$ is an acceptable estimate of θ^*, i.e., whether $\hat{\theta}$ has converged into a suitable small neighborhood of θ^*. If yes, exponentially decreasing the dither amplitude a and ES process ends; otherwise, go to Step 1.

As it has already discussed in the Chapter 2.2.2,

$$\begin{aligned} y &= Q(\hat{\theta} + a\sin\omega t) \\ &= Q(\hat{\theta}) + Q'(\hat{\theta})\cdot a\sin\omega t + \frac{1}{2}\cdot Q''(\hat{\theta})\cdot(a\sin\omega t)^2 \\ &\quad + \frac{1}{6}\cdot Q'''(\hat{\theta})\cdot(a\sin\omega t)^3 + O(a^4) \end{aligned} \tag{3.17}$$

after well-designed high-pass filter ($\omega_h \leq \omega/2$), the signal is

$$Q'(\hat{\theta}) \cdot a\sin\omega t - \frac{a^2}{4} \cdot Q''(\hat{\theta}) \cdot \cos 2\omega t + \frac{1}{6} \cdot Q'''(\hat{\theta}) \cdot (a\sin\omega t)^3 + O(a^4)$$

after demodulation, the signal is

$$\begin{aligned} & ab \cdot Q'(\hat{\theta}) \cdot (\sin\omega t)^2 - \frac{a^2 b}{4} \cdot Q''(\hat{\theta}) \cdot \sin\omega t \cos 2\omega t + \frac{a^3 b}{6} \cdot Q'''(\hat{\theta}) \cdot (\sin\omega t)^4 + O(a^4 b) \\ = & \frac{ab}{2} \cdot Q'(\hat{\theta}) \cdot (1 - \cos 2\omega t) - \frac{a^2 b}{4} \cdot Q''(\hat{\theta}) \cdot \sin\omega t \cos 2\omega t \\ & + a^3 b \cdot Q'''(\hat{\theta}) \cdot (\frac{1}{16} - \frac{\cos 2\omega t}{12} + \frac{\cos 4\omega t}{48}) + O(a^4 b) \end{aligned}$$

after the low-pass filter ($\omega_l < \omega$), the signal is

$$\zeta = \frac{ab}{2} \cdot [Q'(\hat{\theta}) + \frac{a^2}{8} Q'''(\hat{\theta}) + O(a^3)] \tag{3.18}$$

which shows that ζ is a good estimate of the gradient as long as dither amplitude a is small enough, i.e.

$$\zeta \approx \frac{ab}{2} \cdot Q'(\hat{\theta}) \tag{3.19}$$

Step 1: Observe the gradient $Q'(\hat{\theta})$

As is shown in the deduction, ζ contains the gradient information. The controller will first process $\zeta(t)$ signal before accepting it as the gradient vaule.

$$\zeta_p(t) = \frac{1}{T} \int_{t-T}^{t} \zeta(\tau) d\tau \tag{3.20}$$

where T can be chosen as $4 \sim 8$ integer times the period of the dither signal. There are two

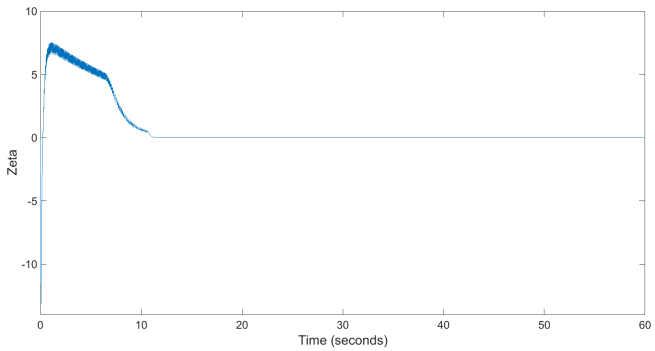

Fig. 3.5 ζ signal

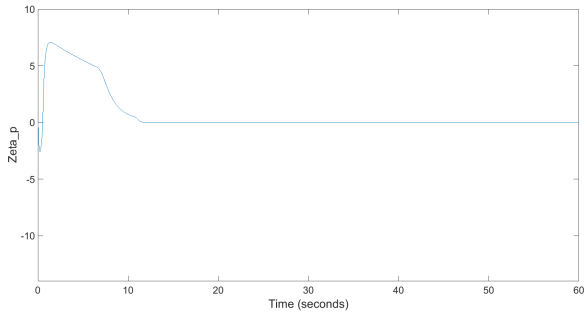

Fig. 3.6 ζ_p signal

reasons why ζ signal is averaged during a fixd-size (multi-period) moving window. Firstly, ES uses averaging technique when analysing its stability, and if ES signals are observed during each dithering period, signals are actually dithering within each period, so is ζ signal. Secondly, in the context of the averaging analysis, the averaged gradient is taken over one period only. But in the presence of measurement noise, the gradient information taken over a very short period may not be as effective as the information obtained from a relatively long period, which is reflected from empirical study. Therefore, the gradient information is acquired over multiple periods in this case. The value of T in Equation (3.20) can be chosen based on the specific application. An example of the comparison between ζ signal and ζ_p signal is shown in Figure 3.5 and 3.6.

Step 2: Adjust $\hat{\theta}$ updating speed

Based on the current value of $\zeta_p(t)$, dither amplitude b or integral gain δ is adjusted with a view to having a better updating speed of the parameter $\hat{\theta}$. Going back to Equation (3.19) and the ES structure in Fig. 3.2, the updating law of parameter $\hat{\theta}$ is:

$$\dot{\hat{\theta}} = \delta \cdot \zeta \approx \frac{ab\delta}{2} \cdot Q'(\hat{\theta}) \tag{3.21}$$

Given δ, a, b, the smaller the gradient $Q'(\hat{\theta})$, the smaller the ζ or ζ_p and the slower the parameter updating speed $\dot{\hat{\theta}}$. Therefore, when ζ_p is beyond an ideal range which shows the gradient is smaller or larger than expection, parameter b or δ can be adjusted in order to have a suitable $\hat{\theta}$ updating speed.

Based on Equation (3.21), the bound of the updating speed can be estimated. As is discussed,

$$\zeta_p \approx \zeta \approx \frac{ab}{2} \cdot Q'(\hat{\theta}) \tag{3.22}$$

When ζ_p is calculated between $[\underline{\zeta}, \overline{\zeta}]$, according to Equation (3.21), $\dot{\hat{\theta}} \in [\underline{\zeta} \cdot \delta,\ \overline{\zeta} \cdot \delta]$. Thus, if there is a desired parameter updating speed $\dot{\hat{\theta}}_d$, we can choose a new $\omega_n = \varepsilon \cdot \omega$ and $b_n = \varepsilon \cdot b$, or $\omega_n = \varepsilon \cdot \omega$ and $\delta_n = \varepsilon \cdot \delta$ such that

$$\varepsilon \cdot \underline{\zeta} \cdot \delta < \dot{\hat{\theta}}_d < \varepsilon \cdot \overline{\zeta} \cdot \delta \tag{3.23}$$

which shows that the updating speed $\dot{\hat{\theta}}$ is now adjusted into a vicinity of the ideal speed $\dot{\hat{\theta}}_d$.

The design guideline mentioned above is practical. For an engineering project, there is always a convergence duration limit set by the reality and the length of the parameter range $\left|\overline{\theta} - \underline{\theta}\right|$ is normally known. Therefore, we can have an ideal updating speed $\dot{\hat{\theta}}_d$ based on the information about the length and time limit. All the other information (a, b, δ, ζ_p) is known (a, b, δ are defined from normal tuning practice and current ζ_p is on-line calculated), thus ε is obtained from inequality Equations (3.23). The corresponding ε value for each bound $[\underline{\zeta},\ \overline{\zeta}]$ can be stored in the controller part, similar to the relationship of engine look-up table and engine control unit.

The look-up bound of ε could be designed in a very simple way. For example, for this engineering project, we only need to design two bound range for ε, one for the normal part, the other one for the flat area in the cost function. There is a design example for a general case in Chapter 3.2.3, explaining the design procedure and parameter setting.

Step 3: Check validity of $\hat{\theta}$

When the measured value of ζ_p approaches zero, there are two possiblilities, i.e.

- $\hat{\theta}$ converges into a very small neighborhood of θ^*,
- or $\hat{\theta}$ goes into a flat area where the gradient within the area is very small.

How to discriminate these two cases is the thing Step 3 will discuss.

Case 1:
When $\hat{\theta}$ has converged into a very small neighborhood of θ^*, what will happen if b is increased to some extent?

Let us go back to Assumption 2.2.1 and the updating law (3.21). According to the assumption, when $\hat{\theta} = \hat{\theta}^* + \Delta\theta$ $(\Delta\theta > 0, \Delta\theta \approx 0)$, $Q'(\hat{\theta}) < 0$; based on the updating law

$\dot{\hat{\theta}} = \frac{ab\delta}{2} \cdot Q'(\hat{\theta})$, $\dot{\hat{\theta}} < 0$ and $\hat{\theta}$ will go to the left side of $\hat{\theta}^*$. When $\hat{\theta} = \hat{\theta}^* - \Delta\theta$ ($\Delta\theta > 0, \Delta\theta \approx 0$), $Q'(\hat{\theta}) > 0$; based on the updating law, $\dot{\hat{\theta}} > 0$ and $\hat{\theta}$ will go to the right side of $\hat{\theta}^*$. When $\hat{\theta}$ has converged into a very small neighborhood of θ^*, $\hat{\theta}$ will oscillate around $\hat{\theta}^*$, which is commonly known in the literature.

When b is increased to some extent, according to the updating law, $|\dot{\hat{\theta}}|$ will be bigger, which will incur larger range of oscillation around $\hat{\theta}^*$; ζ will also oscillate around 0 with larger oscillation amplitude. An example is shown in Figure 3.7.

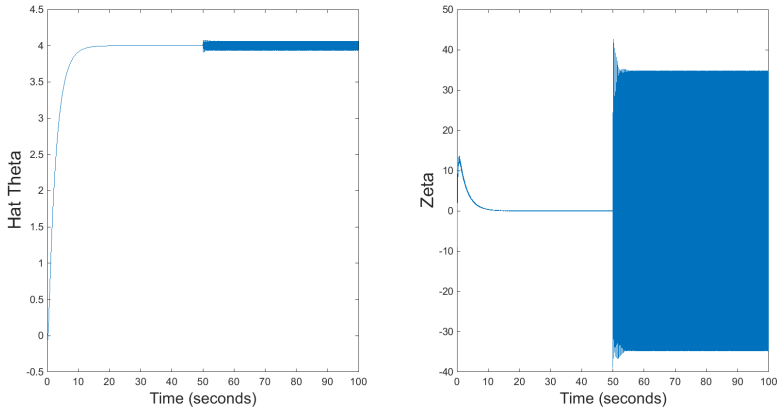

Fig. 3.7 larger oscillation with increased b

Case 2:
When $\hat{\theta}$ goes into a flat area where the gradient within the area is very small, what will happen if b is increased to some extent?

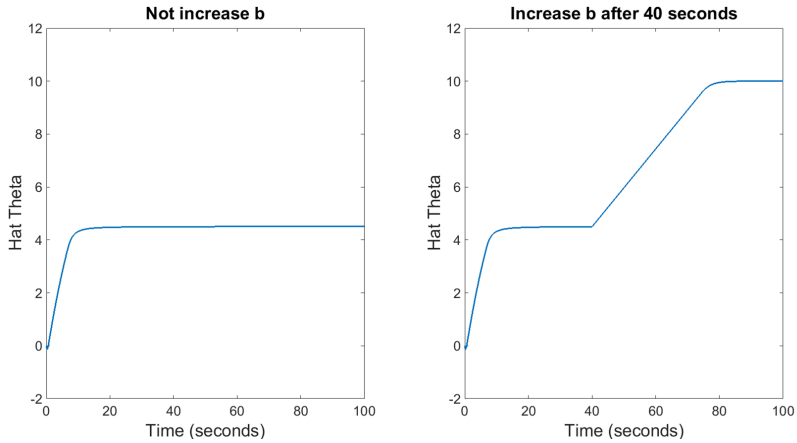

Fig. 3.8 convergence comparison between un-increased and increased b or δ

When increasing b to some extent, the updating speed $\dot{\hat{\theta}}$ will be increased to some extent, which accelerates the ES convergence process. An example is shown in Figure 3.8.

If $\hat{\theta}$ has not converged into a small neighborhood of $\hat{\theta}^*$, the ES convergence process will go on until $\hat{\theta}$ goes into a small neighborhood of $\hat{\theta}^*$. However, due to the existence of the flat area where the gradient is very small, ES convergence duration will be quite long since ES is gradient-driven method and ES convergence speed will be slow if the gradient information is weak. Or according to the updating law (3.21), when $|Q'(\hat{\theta})|$ is small, the updating speed $\dot{\hat{\theta}}$ will be small, which causes longer convergence time.

Check validity of $\hat{\theta}$

As is clearly compared between these two cases, when the measured value of ζ_p approaches zero, we could make a distinction between the two cases by increasing dither amplitude b.

- If the oscillation amplitude of $\hat{\theta}$ or ζ is larger while the averaged $\hat{\theta}$ and ζ values over integer times dithering period are still the same, $\hat{\theta}$ has converged into a small neighborhood of $\hat{\theta}^*$.
- If the oscillation amplitudes of $\hat{\theta}$ and ζ do not change too much while the averaged $\hat{\theta}$ value goes on moving, $\hat{\theta}$ is now in a flat area and it has not converged into a small neighborhood of $\hat{\theta}^*$.

The method [5] to detect oscillation amplitude is

$$\zeta = a_m sin(\omega_u t) \longrightarrow (\cdot)^2 \longrightarrow lowpass\ filter\ or\ average \longrightarrow \sqrt[2]{2 \cdot (\cdot)} \longrightarrow a_m$$

where $a_m > 0$ is the unknown oscillation amplitude, and $\omega_u > 0$ is the oscillation frequency.

The improvement method is to improve the existing ES tuning methods. In other words, the improvement method consists of normal tuning practice (for the initial value of the ES tuning paramters) and the controller design. As long as the initial value guarantees the stability of the ES, the proposed method is correct since the stability of the ES has not been changed as the value of the parameter $(a,\ \frac{\delta b}{\omega})$ has never been changed (cf. Chapter 2.2.2 Extremum seeking).

3.2.3 Simulation example

Let us illustrate the method with a simple example.

Cost function $Q(\theta)$:

$$Q(\theta) = \begin{cases} 0 - 1 * (\theta - 10)^2 & ,\text{if} \quad |\theta - 10| \geq 6 \\ -36 - 0.01 * |\theta - 10| + 0.06 & ,\text{if} \quad |\theta - 10| < 6 \end{cases} , \ \theta \in [0, 20]$$

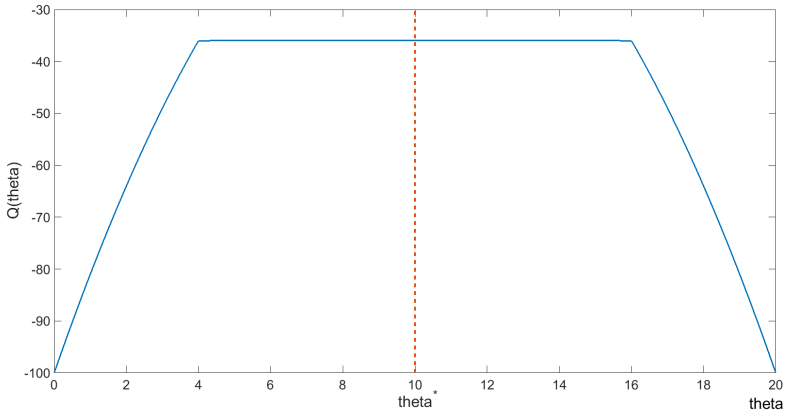

Fig. 3.9 Cost function map

Improvement Design = Normal Tuning Practice (existing) + Controller Design (contribution)

Normal tuning practice

Based on the normal ES tuning practice, the parameters for this case are tuned as below:

Parameter	Value	Meaning
a_0	0.5	dither signal amplitude a
b_0	2	dither signal amplitude b
ω	50	dither signal frequency, rad/sec
δ_0	0.1	integral section gain
ω_h	30	high-pass filter cutoff frequency, rad/sec
ω_l	5	low-pass filter cutoff frequency, rad/sec

Table 3.1 ES parameter value table

Here is the result for the normal tuning practice, as is shown in Figure 3.10.

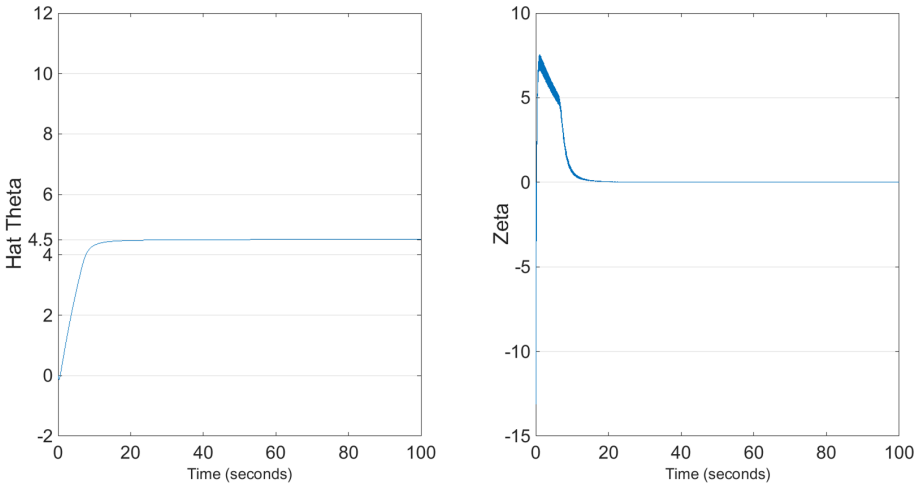

Fig. 3.10 ES result for normal tuning practice

As is shown from the result, when $\hat{\theta}$ goes into the flat area, the convergence speed is very slow; $\hat{\theta}$ stays around 4.5 and its further convergence is almost invisible.

Controller design

Suppose the maximum convergence speed $\dot{\hat{\theta}}_{max}$ is 4, which is evaluated from specific project requirements. The parameter ε can be decided as:

$$\varepsilon = \begin{cases} 1 & ,\text{if} \quad |\zeta_p| \geq 0.1 \\ 400 & ,\text{if} \quad |\zeta_p| < 0.1 \end{cases}$$

since when $|\zeta_p| < 0.1$, $|\varepsilon \cdot \delta \cdot \zeta_p| < 400 \cdot 0.1 \cdot 0.1 = 4 = \dot{\hat{\theta}}_{max}$, which meets the requirement.

Please note that there is a difference between increasing b and δ.

- When using $b_n = \varepsilon \cdot b_0$, considering Equation (3.19), ζ_p is also increased; thus ζ_p signal needs to be divided by the corresponding ε value before accepting it to give instructions to adjust parameter value.

- If δ is chosen to be changed, i.e. $\delta_n = \varepsilon \cdot \delta_0$, there is no need to do any further processing before accepting ζ_p signal to give instructions, since parameter δ is totally decoupled with the gradient signal ζ in the proposed structure (Fig. 3.2).

Result of improvement design

We still use the same initial tuning parameter vaules obtained from normal tuning practice part, and the designed controller part is incorporated into ES. Here is the result obtained from this improvement method, as is shown in Figure 3.11.

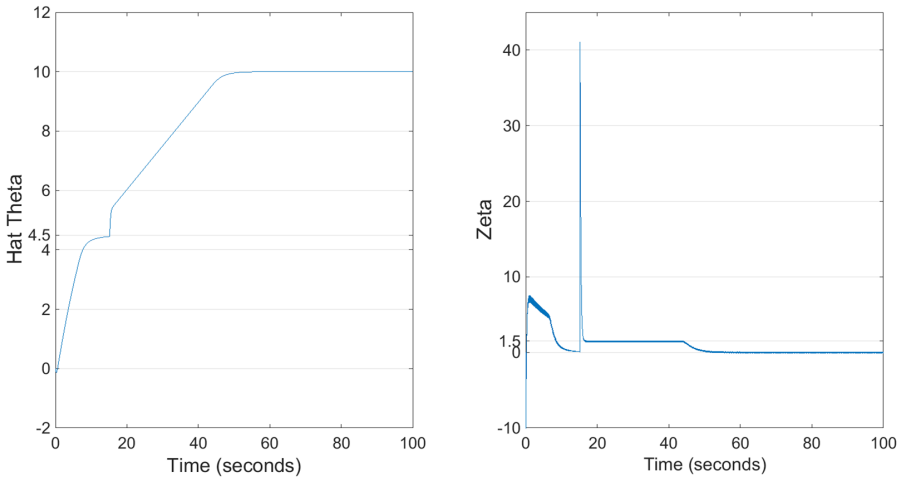

Fig. 3.11 ES result for improvement method

As is shown from the result, $\hat{\theta}$ has converged into a small neighborhood of θ^*;

How about the case without the flat area? what is the convergence result when applying this improvement method to a cost function map where there is no flat area? Let us still consider this example but the flat area is removed at this time; thus the cost function is:

$$Q(\theta) = 0 - 1 * (\theta - 10)^2 \ , \ \theta \in [0, 20]$$

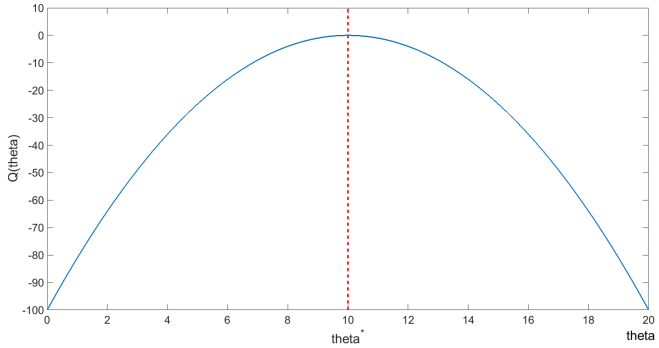

Fig. 3.12 Cost function map

Here are the results using the same improvement method with the same parameter values, as are shown in Figure 3.13, 3.14 and 3.15.

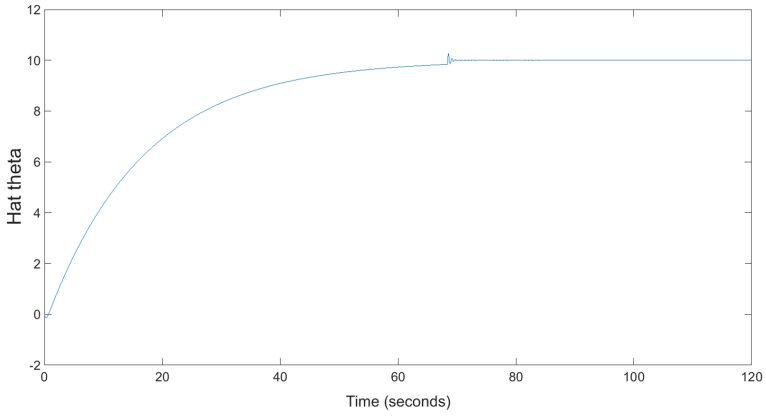

Fig. 3.13 Convergence result - $\hat{\theta}$ signal

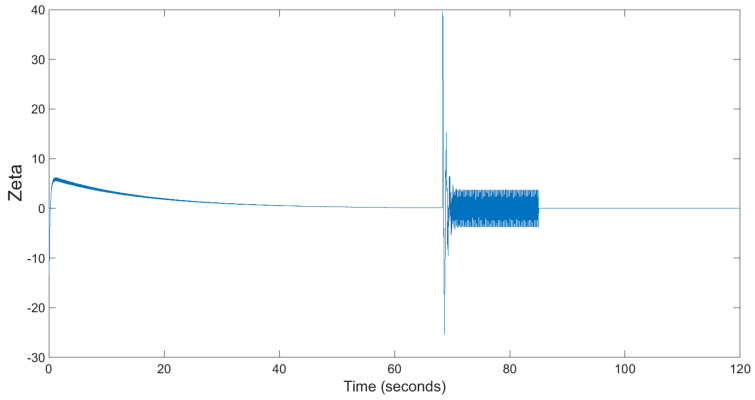

Fig. 3.14 Convergence result - ζ signal

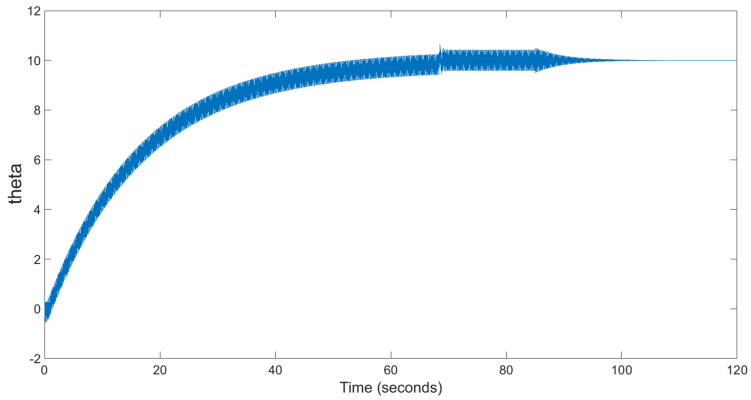

Fig. 3.15 Convergence result - θ signal

As is shown from the results, the method not only drives $\hat{\theta}$ into a small neighborhood of θ^*, it also double checks the validity of $\hat{\theta}$ and exponentially decreases the amplitude of dither signal so that the residual oscillation dies away.

The optimization process for this case is:

- At the beginning, ES is converging based on the normal tuning practice.
- Step 1: Smooth ζ signal and detect the current value of the smoothed ζ signal ($\zeta_p(t)$).
- Step 2: When $\zeta_p(t)$ is below the designed threshold 0.1, ε is changed from 1 to 400; as is clearly shown in the figures, there is a sudden jump in $\hat{\theta}$, θ and ζ signal.
- Step 3: If the averaged $\hat{\theta}$ & ζ do not have obvious change while the oscillation amplitude of $\hat{\theta}$ or ζ has obviously increased, which shows $\hat{\theta}$ has converged into a small neighborhood of θ^*, the controller will give instructions to change ε back to 1 and to decrease the dither amplitude a. Then ES optimization process finishes.

In Step 3, the checking time and the final decaying rate are what we can design. Normally this checking time is 4$\sim$10 integer times the dither period (dither period is known since ω value can be passed to the controller), and the checking time is actually very short since ω is normally a large value. In this case, in order to give a better view of this overall process, I manually prolonged the checking duration and chose a very small exponentially decaying rate so that both the checking process and amplitude decaying process are clearly viewed in Figure 3.14 and 3.15.

To sum up, the fundamental idea of this improvement method is to adjust parameter updating speed based on the gradient information of the black-box plant. The analysis shows that this proposed method is able to improve traditional ES performance by further driving the objective parameter θ into an acceptable neighborhood of the optimal value θ^* and thus helps shorten ES convergence duration. The simulation example illustrates the designing process of the proposed ES scheme, and the corresponding result shows the operating process of the design and verifies the validity of the method.

3.2.4 Application in model calibration problem

Here is the result using the improvement method, as is shown in Figure 3.16.

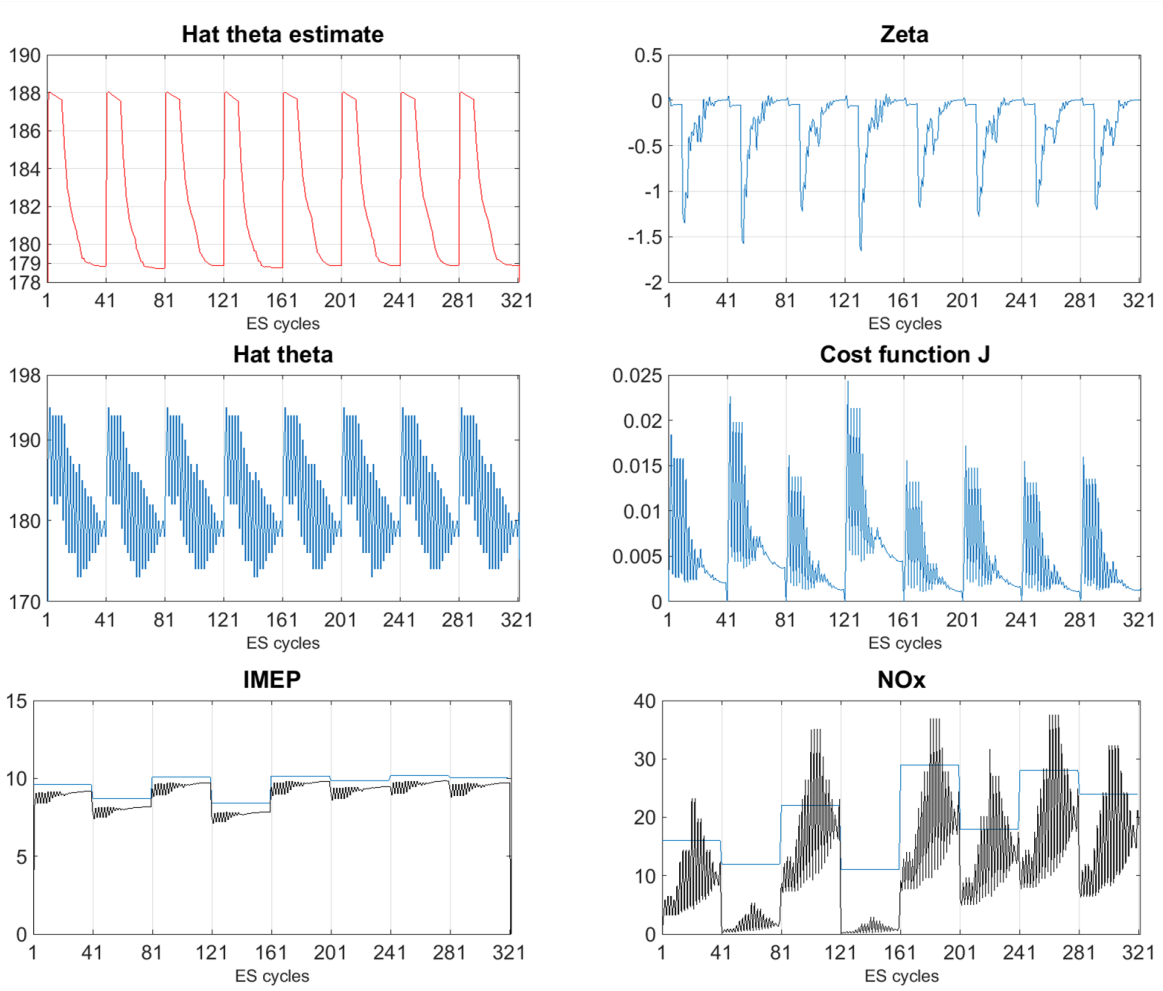

Fig. 3.16 Convergence result of the improvement method

Compared with the former result in Figure 2.20, the improvement method makes $\hat{\theta}_{est}$ arrive at somewhere between 178∼179 degree while the former method can only drive the parameter to somewhere between 185∼187 degree. The comparison of the results shows that the improvement method is more effective and it succeeds in making the parameter converge into a small neighborhood of the optimal value $\hat{\theta}^*$, which verifies the validity of the method. ($\hat{\theta}^*$ lies in somewhere between 178∼179 degree, cf. Fig. 2.19, and all the eight $\hat{\theta}^*$ are literally the same since the eight operating points are randomly selected from the same engine speed 1500 rpm.)

3.3 Delay identification using the ES

3.3.1 Structure of the scheme

The overall diagram is shown in Figure 3.17.

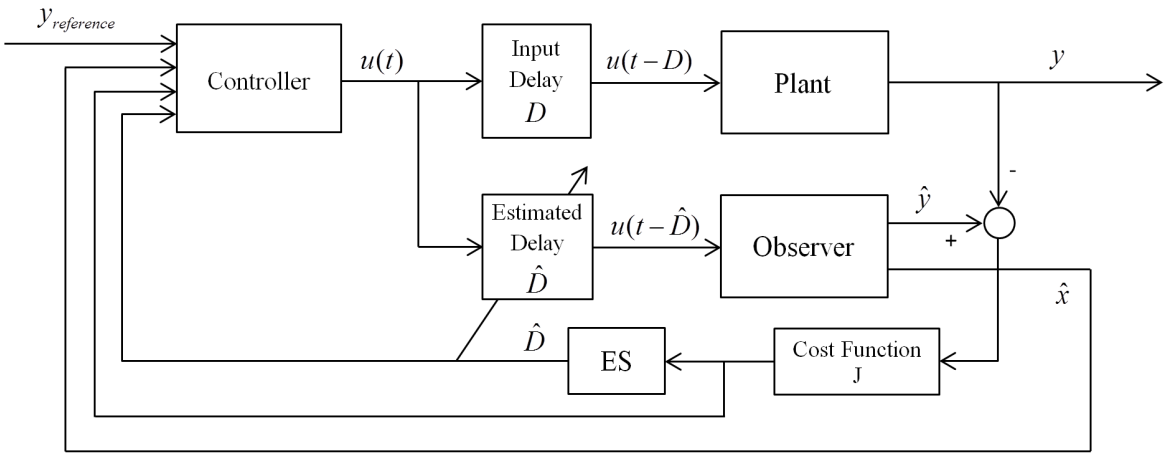

Fig. 3.17 Overall diagram

The controller output is

$$u(t) = \phi(u_d, u_f) \tag{3.24}$$

where u_d is the excitation input for the delay identification process:

$$u_d = \phi_1(t) \tag{3.25}$$

u_f is the corresponding stabilization control law:

$$u_f = \phi_2(\hat{x}(t+\hat{D})) \tag{3.26}$$

Assume the prediction-like control law (3.26) has been designed. The targets are

- identifying the time delay D, i.e. making $\hat{D} \to D$;
- designing function ϕ in (3.24) that makes the parameter (time delay) updating process have no influence on the simultaeous working process of the feedback stabilization scheme in terms of stability and oscillation.

Please note that the design for the specific control law (3.26) is not of concern in this thesis. Let us first finish the task of delay identification for this case.

3.3.2 Identification design

A successful identification design normally depends on five factors [10]; the current progress of the actuator delay identification problem is shown in Table 3.2.

Factor	Current status
Construction of model	Model has been designed and calibrated
Available information of plant	Corresponding plant output can be measured
Excitation signal	Unfinished
Criterion to be optimized	Unfinished
Optimization algorithm	Unfinished

Table 3.2 System identification factors and current status of the problem

In order to give a better view of the problem and design method, let us use a simplified problem to illustrate the overall idea.

Simplified Problem 1

Consider a monotonic (continuously increasing or decreasing) signal $u_m(t)$ and its delayed signal $u_m(t-D)$ with time delay D. The identification scheme is shown in Figure 3.18.

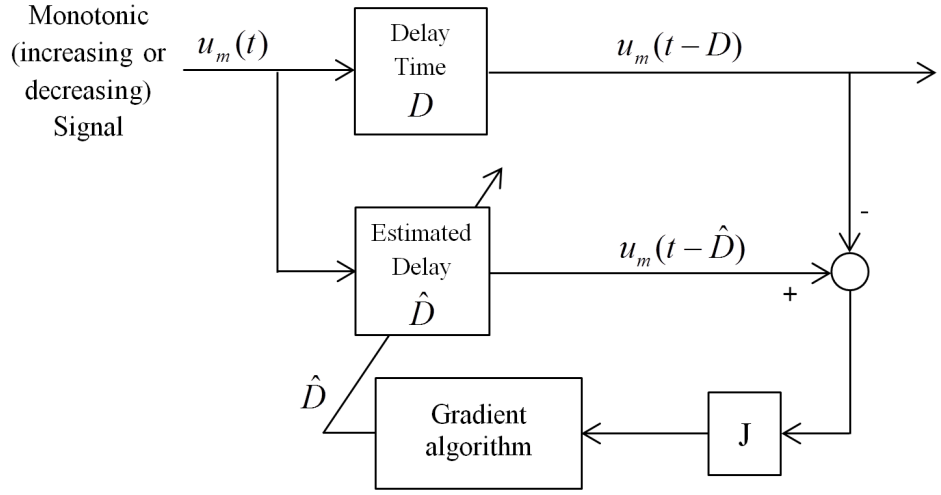

Fig. 3.18 The diagram of delay identification for monotonic signal

Statement *Considering the cost function* $J(\hat{D},t) = |u_m(t-\hat{D}) - u_m(t-D)|$*, there exists only one equilibrium* ($\hat{D} = D > 0$) *that makes* $J = 0$*; and for any feasible* t*, when* $\hat{D}$ *is approaching* D *from any side of* D (D^- *or* D^+)*, the cost function* $J(\hat{D},t) \to 0$.

Proof

Suppose $\exists\, t, D_2 \neq D \implies J(D_2,t) = 0$.
$\because u_m(t)$ is monontonic with respect to $t \therefore u_m(t-D_2) = u_m(t-D)$
$\because u_m(t)$ is monontonic with respect to $t \therefore t - D_2 = t - D$
$\implies D_2 = D$, which is contradictory the the assumption "$\exists\, D_2 \neq D \implies J = 0$".
Thus, there exists only one equilibrium $\hat{D} = D$ that makes $J(\hat{D},t) = 0$.

Suppose $u_m(t)$ is monotonically increasing with respect to t and $\hat{D}_1 < \hat{D}_2 < D$.

$$\begin{aligned} J(\hat{D}_1,t) - J(\hat{D}_2,t) &= |u_m(t-\hat{D}_1) - u_m(t-D)| - |u_m(\hat{D}_2) - u_m(t-D)| \\ &= u_m(t-\hat{D}_1) - u_m(t-D) - [u_m(t-\hat{D}_2) - u_m(t-D)] \\ &= u_m(t-\hat{D}_1) - u_m(t-\hat{D}_2) > 0 \end{aligned}$$

which shows that when $u_m(t)$ is a monotonically increasing signal and $\hat{D}$ is approaching D from the left side, the cost function $J(\hat{D},t)$ is monotonically decreasing to 0 with respect to $\hat{D}$.

So it is the same with the proof for the other three cases:

- $u_m(t)$ is monotonically increasing and $\hat{D}$ is approaching D from the right side;
- $u_m(t)$ is monotonically decreasing and $\hat{D}$ is approaching D from the left side;
- $u_m(t)$ is monotonically decreasing and $\hat{D}$ is approaching D from the right side;

Therefore, for any feasible t, when $\hat{D}$ is approaching D from any side of D (D^- or D^+), the cost function $J(\hat{D},t) \;\to\; 0$.

Q.E.D.

Problem formulation

So it comes to the question of identifying the time delay D for this case. If just considering the cost function J with respect to D, it is a simple minimization problem which can be easily solved using SISO ES for static mapping case. However, the cost function $J(D,t)$ is also influenced by the parameter t. Please note the parameter t does not change the sign of $\frac{\partial J}{\partial D}$; when $\hat{D}$ is approaching D from either side, J is decreasing to 0. What the parameter t changes is the cost function value, that is, the value of $\frac{\partial J}{\partial D}$. The problem for this case is formulated as next.

Assumption 3.3 *Consider a smooth, monotonic signal. There exists a unique D^* minimizing $J(D,t)$ and the following holds:*

$$J(D^*,\, t) < J(D^* + \zeta,\, t)\ ,\forall \zeta \in R - \{0\},\, t \in R \tag{3.27}$$

$$\frac{\partial J(D^* + \zeta,\, t)}{\partial (D^* + \zeta)} \cdot \zeta > 0\ ,\forall \zeta \in R - \{0\},\, t \in R \tag{3.28}$$

Problem Find D^*.

Method Gradient algorithm: $\dot{\hat{D}} = \begin{cases} -\delta \cdot \frac{\partial J(\hat{D},\, t)}{\partial \hat{D}} & ,\hat{D} \neq D^* \\ 0 & ,\hat{D} = D^* \end{cases}\ ,\ \forall\, t \in R,\ \delta > 0$

Stability analysis

Construct Lyapunov function

$$V = (\hat{D} - D^*)^2 \tag{3.29}$$

where $V > 0\ \forall \hat{D} \neq D^*$ and $V = 0 \iff \hat{D} = D^*$.

Then for any $\hat{D} \neq D^*$,

$$\begin{aligned} \dot{V} &= 2 \cdot (\hat{D} - D^*) \cdot \dot{\hat{D}} = 2 \cdot (\hat{D} - D^*) \cdot (-\delta \cdot \frac{\partial J(\hat{D},\, t)}{\partial \hat{D}}) \\ &= -2\delta \cdot (\hat{D} - D^*) \cdot \frac{\partial J(\hat{D},\, t)}{\partial \hat{D}} \end{aligned} \tag{3.30}$$

Let $\zeta = \hat{D} - D^*$, and take it into Equation (3.30), which leads to

$$\dot{V} = -2\delta \cdot \frac{\partial J(D^* + \zeta,\, t)}{\partial (D^* + \zeta)} \cdot \zeta \tag{3.31}$$

where $\dot{V} < 0\ \forall \zeta \neq 0$, and $\dot{V} = 0 \iff \zeta = 0\ (\hat{D} = D^*)$.

According to Lyapunov theory [8], the system is asymptotically stable and Lyapunov function V will converge to 0, that is, $\hat{D}$ converges to D^*.

Extremum seeking

Extremum seeking is a way of implementing the gradient algorithm, as is proved in [25, 29], or as is discussed in the former Chapter 2.2. Therefore, if considering the assumption that $\frac{\partial J}{\partial \hat{D}}$ exists except for one point D^*, ES can at least drive the parameter $\hat{D}$ into a small neighborhood of D^*, and the delay time for this case is thus identified.

Problem 2

Based on the simplified problem one, let us consider a further problem. Here is a linear time-invariant system

$$\begin{aligned} &\dot{x} = Ax + Bu(t-D),\ x(t_0) = x_0,\ x \in R^n, n \in N^+ \\ &y = Cx,\ y \in R \end{aligned} \tag{3.32}$$

with controllable (A,B), observable (A,C) and stable A. The plant states are unknown while the output can be measured.

Here is an observer (3.33) with exact matrixes $\hat{A} = A,\ \hat{B} = B$ and $\hat{C} = C$.

$$\begin{aligned} &\dot{\hat{x}} = A\hat{x} + Bu(t-\hat{D}),\ \hat{x}(t_0) = \hat{x}_0,\ \hat{x} \in R^n, n \in N^+ \\ &\hat{y} = C\hat{x},\ \hat{y} \in R \end{aligned} \tag{3.33}$$

The structure of the delay identification scheme is shown in Figure 3.19.

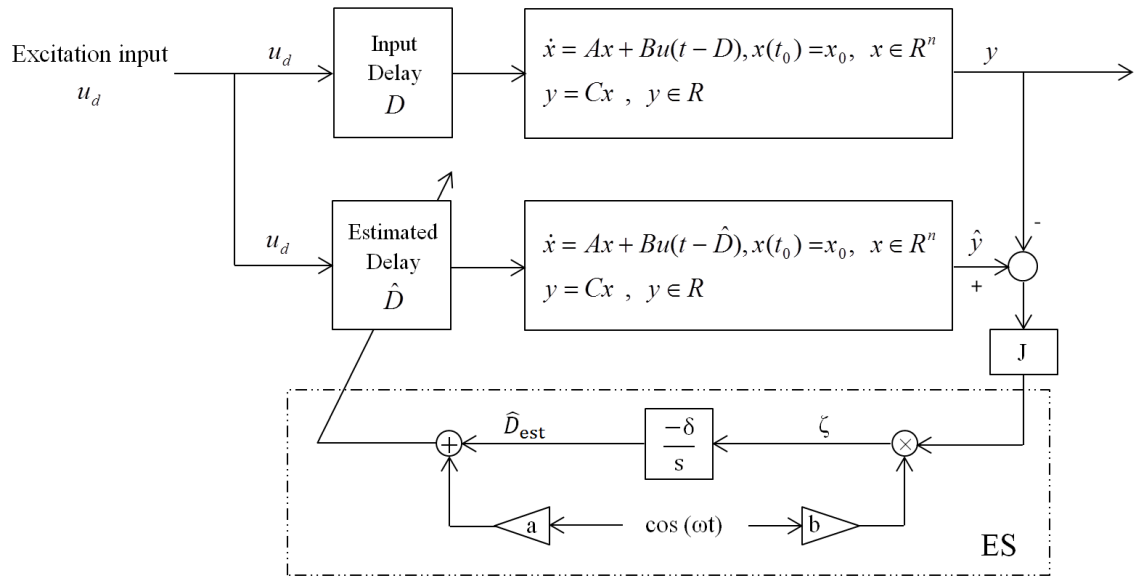

Fig. 3.19 Structure of ES for the delay identification

The cost function for extremum seeking (criterion to be optimized) is:

$$J = |\hat{y}(t) - y(t)| \tag{3.34}$$

The only remaining factor in Table 3.3 that waits designing is the excitation input. In order to give a better view of the excitation design, let us do a comparison between Problem

1 and Problem 2.

Factor	Current status
Construction of model	Model is accurate
Available information of plant	Plant output can be measured
Excitation signal	Unfinished
Criterion to be optimized	Equation (3.34)
Optimization algorithm	Extremum seeking

Table 3.3 System identification factors and current status

Comparison with Problem 1

The fundamental idea of delay identification for this LTI case is still the same as the Problem 1: there is a monotonic output signal from the model and there is the corresponding delayed monotonic output signal from the plant; optimization method like ES or least-square can be utilized to identify the delay by minimizing the absolute error between the two output signals, since there exists only one equilibrium $\hat{D} = D$ that makes the absolute error minimum, as it is proved in Problem 1. There are many papers in the literature that utilizes least-square as the optimization method to identify the delay and few has ever used extremum seeking with regard to delay identification, we will use ES to illustrate the idea.

In Problem 1, the signal is static (the input - output relationship is a static mapping) while in Problem 2, the output signal is a dynamic signal, the dynamics introduced by LTI system. For the static signal case (Problem 1), it is shown that $\hat{D}$ can converge into a suitable small neighborhood of D by utilizing ES optimization method. So the barrier of extending from Problem 1 to Problem 2 is the dynamics introduced by the system.

Handling dynamics

In order to have an good approximation of the steady state input-output mapping, singular perturbation technique [8] is utilized to handle the dynamics of the system in Problem 2. The fundamental idea is time-scale separation [8]: As long as the input to the plant changes slowly enough compared to the dynamics of the plant, the relationship between plant input and plant output can be approximated as a static mapping, i.e. steady state input-output mapping. Thus, The dynamics of the plant can be handled and Problem 2 can be approximated as Problem 1. For detailed proof of this technique, please refer to [8].

Excitation signal

The excitation signal for Problem 2 is the control input to both the plant and model. Theoretically speaking, the control input (excitation signal) can be designed as:

$$u_d = u_0 + k_s \cdot t \ , \ k_s \neq 0 \tag{3.35}$$

where u_0 is the initial control input value and k_s is the slope which is the tuning parameter.

In terms of the input to steady-output relationship of a LTI system, a monotonic input signal to the system will produce a monotonic steady-output signal (superposition principle). In order to handle the dynamics of the output signal, the input signal is supposed to change slowly enough compared with the dynamics of the system, so that the input-output relationship can be approximated as a static (input to steady-output) mapping. This task is achieved by tuning the parameter k_s in Equation (3.35). That is, as long as the absolute value of k_s is sufficiently small, the change of the input signal can be sufficiently slow compared with the dynamics of the system, and the input to output relationship can be approximated as a static mapping, thus transforming Problem 2 into Problem 1.

In reality, the control input (excitation signal) is always within a bound and thus it can be designed as:

$$u_d = u_0 + k_a \cdot \sin(\omega_s t) \ , \ k_a, \omega_s > 0 \tag{3.36}$$

where u_0 is the initial control input value and k_a, ω_s are the tuning parameters.

Compared with the control input (3.35), the input (3.36) is bounded and the corresponding parameter k_a can be adjusted so that the control input meets the boundness requirement set by the real operation condition. The tuning parameter ω_s in (3.36) is used for implementing singular perturbation technique; when ω_s is small enough, the change of the input (3.36) can be sufficiently slow compared with the dynamics of the system, and the input to output relationship can be approximated as a static mapping.

There maybe some arguments that the excitation input (3.36) is not correct since it causes mutiple equilibriums for the delay identification, that is,

$$\hat{D} = D + n \cdot \frac{2\pi}{\omega_s} \ , \ n \in Z \tag{3.37}$$

However, each equilibrium has its own domain of attraction. Considering the phase shift of a periodic signal, it is easily seen that the domain of attraction for each equilibrium is

$$\hat{D} \to D \text{ , if initial } \hat{D} \in (D - \frac{\pi}{\omega_s},\ D + \frac{\pi}{\omega_s})$$

Please note ω_s is a small value in order to implement singular perturbation technique, that is to say, $(D - \frac{\pi}{\omega_s},\ D + \frac{\pi}{\omega_s})$ is a very large range. In reality, there is some information, more or less, regarding an inexact off-line approximation of delay, which serves as the initial condition $\hat{D}(t_0)$ for the delay identification process. Few engineer in reality will choose a nonsense initial estimate value of delay that deviates far from its real value in order to make the delay identification scheme unsuccessful.

Time-scale design

A very important point of this identification design is the time-scale separation that guarantees an acceptable extension from Problem 1 to Problem 2. During the delay identification process, there are three different changing process that stays in different time scales. The time-scale design of the overall identification process is shown in Table 3.4.

Process	Time scale
Dynamics of system (plant and model)	Fast
Extremum seeking scheme	Medium
Change of excitation input	Slow

Table 3.4 Design of time-scale for each process

Tuning guideline

Regarding the ES structure in Figure 3.19 and choosing the excitation input signal (3.35) for theoretical analysis, the tuning guidelines of ES and excitation signal (3.35) are:

Excitation signal

$$\text{Given (A,B)},\ \exists\ |k_s|^* \Rightarrow |k_s| \Rightarrow k_s \tag{3.38}$$

- Tuning guideline (3.38) means that given the matrixes (A,B) of a dynamic system described in Problem 2, there exists a maximum $|k_s|^*$ such that as long as $|k_s| < |k_s|^*$, the change of control input is always slow enough compared with the dynamics of the system, meeting the requirement of time-scale separation in Table 3.4.

Extremum seeking

$$\text{Given (A,B), } \exists\ (a \cdot b \cdot \delta)^* \Rightarrow 0 < ab\delta < (ab\delta)^* \Rightarrow ab\delta \tag{3.39}$$

$$\exists\ a^* \Rightarrow 0 < a < a^* \Rightarrow a \Rightarrow (\frac{\delta b}{\omega})^*(a) \Rightarrow 0 < \frac{\delta b}{\omega} < (\frac{\delta b}{\omega})^*(a) \Rightarrow \frac{\delta b}{\omega} \tag{3.40}$$

$$\text{Given } ab\delta,\ a\ \&\ \frac{b\delta}{\omega} \Rightarrow b\delta\ \&\ \omega \tag{3.41}$$

- Tuning guideline (3.39) guarantees that given the matrixes (A,B), there exists a maximum step size of the ES $(\frac{a \cdot b \cdot \delta}{2})^*$ such that as long as $(ab\delta) < (ab\delta)^*$, the updating of ES process is always slow enough compared with the dynamics of the system. It makes little difference to the stability of the overall scheme whether ES is in the medium time-scale or slow time-scale, as long as ES updating is always slow enough compared with the dynamics of the system. However, with guaranteed stability, we always desire a shorter convergence of the process, which is the reason why we put ES updating in the medium time-scale so that the overall identification process finishes as quickly as possible.

- Tuning guideline (3.40) is a standard ES tuning, as is discussed in Chapter 2.2.2. There exists $a^* > 0$ and $0 < a < a^*$ in order to do the Approximation (2.30). Given a, there exists $(\frac{b\delta}{\omega})^*(a)$ such that $0 < (\frac{b\delta}{\omega}) < (\frac{b\delta}{\omega})^*(a)$, which is required for the averaging technique.

- Tuning guideline (3.41) is the last sequence. Based on the given value of $ab\delta,\ a\ \&\ \frac{b\delta}{\omega}$ respectively set in (3.39) and (3.40), we could get the value of $b \cdot \delta\ \&\ \omega$. Pick δ and then choose b value or pick b and then choose δ value. Thus, the setting for all tuning parameters finishes.

By this stage, all the five factors of the identification design are finished and the delay identification task is done. The next step is to design a guideline that makes the delay identification process have no influence on the simultaneous working process of the closed-loop control scheme in terms of stability and oscillation.

Simultaneous working design

Once the delay identification task is finished, the next task for us is to design the simultaneous working principle to guarantee the simultaneous working process of the identification and stabilization process.

Assume a general prediction-like control law $u_f = \phi_2(\hat{x}(t+\hat{D}))$ is constructed. When the parameter $\hat{D}$ is updating simultaneously which is normally characterized by fast transient variation and oscillation, the control input $u_f(\hat{x},\ \hat{D})$ is also oscillating and fast varying, which could cause undesired performance of the overall closed-loop scheme in terms of stability and oscillation. Furthermore, during the delay identification process, the regulation control input u_f will disrupt the identification input (excitation signal u_d), since these two processes use the same controller in Figure 3.17. It would be interesting and essential to know how to handle the simultaneous working process of the two parts and make the two working processes in harmony with each other.

Here is the design to achieve the goal. Since the overall closed-loop controller has two tasks to fulfil, i.e., feedback stabilization and delay identification, the corresponding changing law of the controller can be designed as:

$$u = (1-\lambda_u)\cdot u_f\left(\hat{x},\ \hat{D}\right) + \lambda_u \cdot u_d \tag{3.42}$$

where

$$\lambda_u = 1 - \frac{1}{1+\alpha\cdot|\hat{y}-y|}\ ,\ \alpha > 0 \tag{3.43}$$

The first part of the control law (3.42) is for feedback stabilization, the second part for delay identification.

λ_u is an adaptively varying coefficient that guarantees the asymptotical separation of the two working processes. It has two roles:

- ensuring the delay identification process starting first;
- making the delay identification process fade away by itself; when the identification process finishes, the excitation signal (control input for identifying the delay) vanishes.

The control process is: at the beginning (the control input begins to change), $\lambda_u \approx 1$ due to the existence of error between $\hat{y}$ and y; at this time, the control input $\lambda_u \cdot u_d$ dominates over $(1-\lambda_u)\cdot u_f$ and the error between $\hat{y}$ and y starts to decrease. During the delay identification porcess, $\hat{D}$ is approaching the real delay D and $|\hat{y}-y|$ is minimizing. When $|\hat{D}-D|$ is below a threshold, that is, $|\hat{y}-y|$ is below a threshold, λ_u starts to decrease sharply and just after a very short time, $(1-\lambda_u)\cdot u_f$ dominates over $\lambda_u \cdot u_d$ and the system starts to be stabilized.

α is a tuning parameter; the larger the parameter α, the better the accuracy of delay identification, and the longer the identification process. Given the system (plant and model)

and the control input, there exists a monotonically increasing function F_d such that $|\hat{y}-y| = F_d(|\hat{D}-D|)$. When $\alpha \cdot |\hat{y}-y|$ is below a fixed threshold η_{th} (around 2.5 in Equation (3.43)), λ_u begins to decrease sharply and delay identification input begins to vanish. We can always play with the parameter α, based on the relation $|\hat{y}-y| = F_d(|\hat{D}-D|) < \frac{\eta_{th}}{\alpha}$, so that the bound of $|\hat{D}-D|$ can be adjusted. The larger the parameter α, the smaller the bound of $|\hat{D}-D|$, and the longer the identification process.

Application in the engine problem

The engine model, after calibration, is a fixed non-linear model. The feedback regulation design is to linearize the model around each operation point, thus the model being a linearized fixed model (linear time-invarient). Based on the Assumption 2.1.2 in Chapter 2.1, the model can approximate the engine plant with sufficient accuracy, and the engine problem becomes a close approximation of Problem 2.

Problem 2 is an approximation of the real engine actuator delay identification problem. In the problem, the model is assumed to be 100% accurate (i.e. $\hat{A} = A$, $\hat{B} = B$, $\hat{C} = C$), and there exists error in reality due to this ideal approximation. A further problem extending from this work is that what the bound of identification error $|\hat{D}-D|$ is when there exists an upper error bound between the model and plant.

3.3.3 Simulation example

Let us illustrate the method with a simple example.

Set $A = \hat{A} = \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix}$, $B = \hat{B} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $C = \hat{C} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

Controllable matrix $Q_c = \begin{bmatrix} 1 & -2 \\ 2 & -2 \end{bmatrix}$, the rank of which is full, i.e., controllable.

Observable matrix $Q_b = \begin{bmatrix} 1 & 1 \\ -2 & -1 \end{bmatrix}$, the rank of which is full, i.e., observable.

Initial condition for plant states $x_o = \begin{bmatrix} 10 * rand + 5 \\ 10 * rand + 5 \end{bmatrix}$, and observer states $\hat{x}_o = \begin{bmatrix} 2 * rand \\ 2 * rand \end{bmatrix}$.

Plant input delay D is set as follows: the delay will change after each 15-second period, imitating the change of operating conditions; $D = 20$ for the first 15 seconds and $D = 25$ for the next 15 seconds, the total simulation time is 30 seconds for convenience. Initial condition for delay identification $\hat{D} = 10 * rand$.

Based on the structure in Figure 3.19 and the tuning guideline mentioned in the former Chapter 3.3.2, all the other parameters in the simulation are tuned as below:

Process	Parameter	Value	Meaning
ES	a	0.5	dither signal amplitude a
	b	1	dither signal amplitude b
	ω	150	dither signal frequency, rad/sec
	δ	0.5	integral section gain
	T_s	0.01	discrete sampling time, sec
$u_d = u_0 + k_a \sin(\omega_s t)$	u_0	0	initial condition of input
	k_a	100	amplitude of excitation signal
	ω_s	0.01	frequency of excitation signal, rad/sec
$u_f = \hat{D}$	u_f	$\hat{D}$	example to illustrate the separation
Separation design	α	1	tuning parameter in (3.43)

Table 3.5 Table of relevant parameters

Here is the corresponding result, as is shown in Figure 3.20.

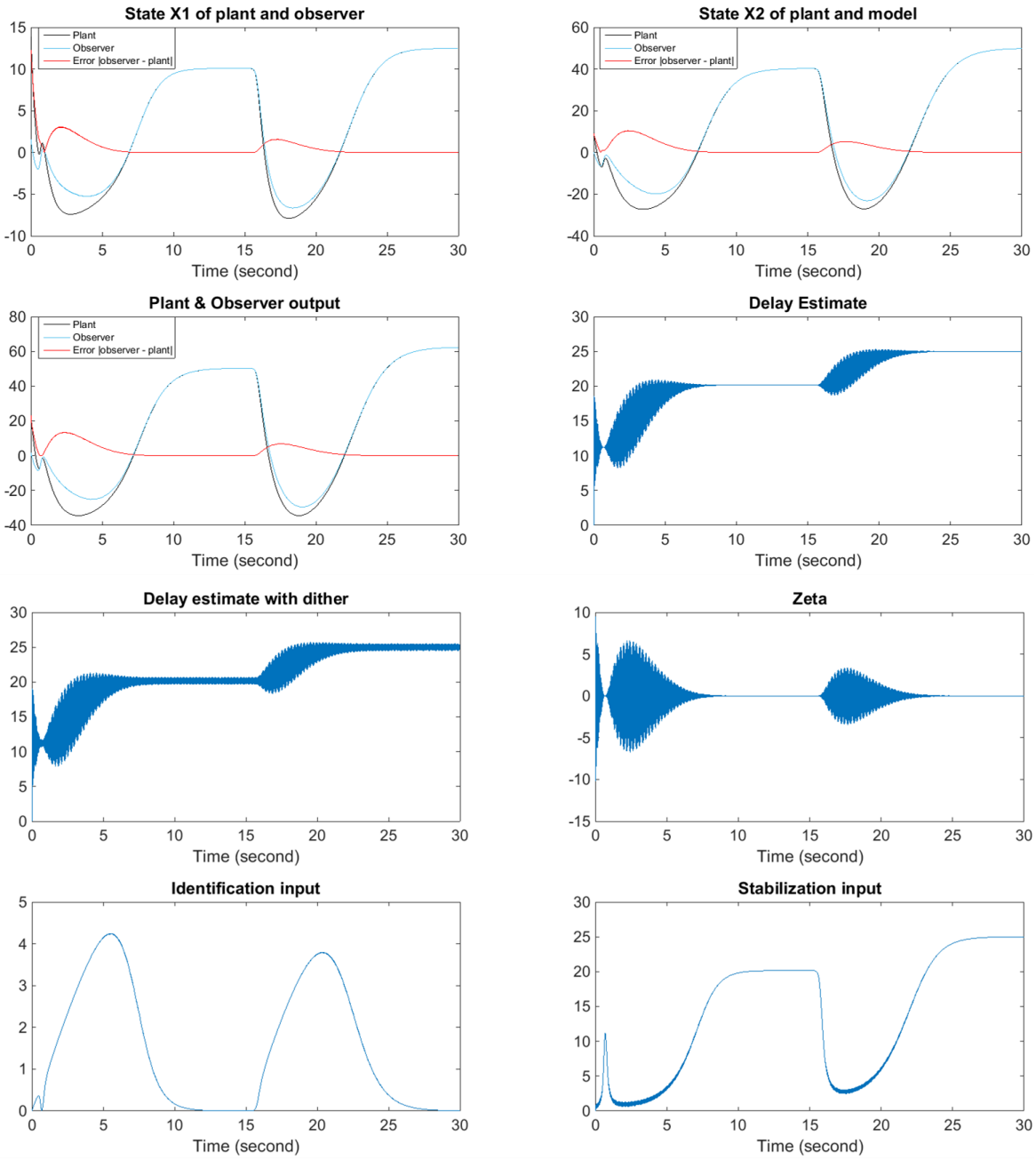

Fig. 3.20 Convergence result of the overall process

As is shown from the result, the delay time is identified accurately and the identification process fades away by itself, which have little influence on the simultaneous working process of the stabilization control in terms of stability and oscillation.

Chapter 4

Summary of Work

4.1 Contribution

1. illustrated the effect of delay on the working of ES (stability and convergence aspects).

2. Proposed a controller looking at ES convergence performance and adjusting the right ES tuning parameters to improve the convergence performance of ES. Note that in our proposed method, the parameter a (amplitude of dither on the left side) has never been increased during ES operation process, and it is supposed not to be increased in order to guarantee ES stability and keep the approximation error small.

3. Extended ES to the area of delay identification, utilized singular perturbation technique to design excitation signal and comed up with an asymptotically separation method that makes the delay identifcation process fades away by itself, causing little influence on the corresponding feedback stabilization control in terms of stability and oscillation.

4.2 Extension

1. The proposed ES improvement method can be extended from SISO ES to multivariable ES case.

2. Relax the assumption that the model is 100% accurate. A further interesting problem extending from this work is to investigate the relationship between the upper error bound of the model and the upper bound of identification error $|\hat{D} - D|$.

Bibliography

[1] Harald Waschl, Ilya Kolmanovsky, Maarten Steinbuch, and Luigi Del Re. *Optimization and Optimal Control in Automotive Systems*, volume 455. Springer, 2014.

[2] John B Heywood. *Internal combustion engine fundamentals*, volume 930. Mcgraw-hill New York, 1988.

[3] Colin R Ferguson and Allan T Kirkpatrick. *Internal combustion engines: applied thermosciences*. John Wiley & Sons, 2015.

[4] Elijah Polak. *Optimization: algorithms and consistent approximations*, volume 124. Springer Science & Business Media, 2012.

[5] Kartik B Ariyur and Miroslav Krstic. *Real-time optimization by extremum-seeking control*. John Wiley & Sons, 2003.

[6] Chunlei Zhang and Raúl Ordóñez. *Extremum-seeking control and applications: a numerical optimization-based approach*. Springer Science & Business Media, 2011.

[7] Shu-Jun Liu and Miroslav Krstic. *Stochastic averaging and stochastic extremum seeking*. Springer Science & Business Media, 2012.

[8] Hassan K Khalil. *Nonlinear systems*. New Jersey: Prentice hall, 2002.

[9] Jan A Sanders, Ferdinand Verhulst, and James A Murdock. *Averaging methods in nonlinear dynamical systems*, volume 59. Springer, 2007.

[10] Torsten Söderström and Petre Stoica. *System identification*. Prentice-Hall, Inc., 1988.

[11] Philip Wolfe. Convergence conditions for ascent methods. *SIAM review*, 11(2):226–235, 1969.

[12] Alan W. Johnson and Sheldon H. Jacobson. On the convergence of generalized hill climbing algorithms. *Discrete Applied Mathematics*, 119(1):37–57, 2002.

[13] Urs Christen, Katie J Vantine, and Nick Collings. Event-based mean-value modeling of di diesel engines for controller design. Technical report, SAE Technical Paper, 2001.

[14] Dobrivoje Popović, Mrdjan Janković, Steve Magner, and Andrew R Teel. Extremum seeking methods for optimization of variable cam timing engine operation. *Control Systems Technology, IEEE Transactions on*, 14(3):398–407, 2006.

[15] Ibrahim Haskara, Guoming G Zhu, and Jim Winkelman. Multivariable egr/spark timing control for IC engines via extremum seeking. In *American Control Conference, 2006*, pages 6–pp. IEEE, 2006.

[16] Nick J Killingsworth and Miroslav KrstiĆ. PID tuning using extremum seeking: online, model-free performance optimization. *Control Systems, IEEE*, 26(1):70–79, 2006.

[17] Shinya Kitazono, Shigehiro Sugihira, and Hiromitsu Ohmori. Starting speed control of SI engine based on extremum seeking control. In *Proceedings of the 17th IFAC World congress*, pages 1036–1041, 2008.

[18] Nick J Killingsworth, Salvador M Aceves, Daniel L Flowers, Francisco Espinosa-Loza, and Miroslav Krstić. HCCI engine combustion-timing control: Optimizing gains and fuel consumption via extremum seeking. *Control Systems Technology, IEEE Transactions on*, 17(6):1350–1361, 2009.

[19] Alireza Mohammadi, Chris Manzie, and Dragan Nesic. Extremum seeking methods for online optimization of spark advance in alternative fueled engines. In *Engine and Powertrain Control, Simulation and Modeling*, volume 3, pages 1–8, 2012.

[20] Eric Hellstrom, Donghoon Lee, Li Jiang, Anna G Stefanopoulou, and Hamza Yilmaz. On-board calibration of spark timing by extremum seeking for flex-fuel engines. *Control Systems Technology, IEEE Transactions on*, 21(6):2273–2279, 2013.

[21] Alireza Mohammadi, Chris Manzie, and Dragan Nešić. Online optimization of spark advance in alternative fueled engines using extremum seeking control. *Control Engineering Practice*, 29:201–211, 2014.

[22] Enrico Corti, Claudio Forte, Giorgio Mancini, and Davide Moro. Automatic combustion phase calibration with extremum seeking approach. *Journal of Engineering for Gas Turbines and Power*, 136(9):091402, 2014.

[23] Qingyuan Tan, Prasad Divekar, Xiang Chen, Ming Zheng, and Yonghong Tan. Exhaust gas recirculation control through extremum seeking in a low temperature combustion diesel engine. In *American Control Conference (ACC), 2014*, pages 1511–1516. IEEE, 2014.

[24] Prasad Divekar, Qingyuan Tan, Ying Tan, Xiang Chen, and Ming Zheng. Nonlinear model reference observer design for feedback control of a low temperature combustion diesel engine. In *American Control Conference (ACC), 2015*, pages 13–18. IEEE, 2015.

[25] Miroslav Krstić and Hsin-Hsiung Wang. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36(4):595–601, 2000.

[26] Mario Rotea et al. Analysis of multivariable extremum seeking algorithms. In *American Control Conference, 2000. Proceedings of the 2000*, volume 1, pages 433–437. IEEE, 2000.

[27] Andrew R Teel and Dobrivoje Popović. Solving smooth and nonsmooth multivariable extremum seeking problems by the methods of nonlinear programming. In *American Control Conference, 2001. Proceedings of the 2001*, volume 3, pages 2394–2399. IEEE, 2001.

[28] Joon-Young Choi, Miroslav Krstić, Kartik B Ariyur, and Jin S Lee. Extremum seeking control for discrete-time systems. *Automatic Control, IEEE Transactions on*, 47(2): 318–323, 2002.

[29] Ying Tan, Dragan Nešić, and Iven Mareels. On non-local stability properties of extremum seeking control. *Automatica*, 42(6):889–903, 2006.

[30] Ying Tan, Dragan Nešić, Iven MY Mareels, and Alessandro Astolfi. On global extremum seeking in the presence of local extrema. *Automatica*, 45(1):245–251, 2009.

[31] Dragan Nesic. Extremum seeking control: Convergence analysis. In *Control Conference (ECC), 2009 European*, pages 1702–1715. IEEE, 2009.

[32] D Nešić, Ying Tan, William H Moase, and Chris Manzie. A unifying approach to extremum seeking: Adaptive schemes based on estimation of derivatives. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 4625–4630. IEEE, 2010.

[33] Y Tan, WH Moase, C Manzie, D Nešić, and IMY Mareels. Extremum seeking from 1922 to 2010. In *Control Conference (CCC), 2010 29th Chinese*, pages 14–26. IEEE, 2010.

[34] William H Moase, Ying Tan, Dragan Nešić, and Chris Manzie. Non-local stability of a multi-variable extremum-seeking scheme. In *Australian Control Conference (AUCC), 2011*, pages 38–43. IEEE, 2011.

[35] Dragan Nesic, Arash Mohammadi, and Chris Manzie. A framework for extremum seeking control of systems with parameter uncertainties. *Automatic Control, IEEE Transactions on*, 58(2):435–448, 2013.

[36] Tiago Roux Oliveira, and Miroslav Krstić. Gradient extremum seeking with delays. *IFAC-PapersOnLine*, 48(12):227–232, 2015.

[37] Tiago Roux Oliveira, and Miroslav Krstić. Newton-based Extremum Seeking under Actuator and Sensor Delays. *IFAC-PapersOnLine*, 48(12):304–309, 2015.

[38] Tiago Roux Oliveira, Miroslav Krstić, and Daisuke Tsubakino. Multiparameter extremum seeking with output delays. In *American Control Conference, 2015*, pages 152–158. IEEE, 2015.

[39] Iven Mareels. Sufficiency of excitation. *Systems & control letters*, 5(3):159–163, 1984.

[40] IMY Mareels, RR Bitmead, M Gevers, CR Johnson, RL Kosut, and MA Poubelle. How exciting can a signal really be? *Systems & control letters*, 8(3):197–204, 1987.

[41] IMY Mareels and M Gevers. Persistency of excitation criteria for linear, multivariable, time-varying systems. *Mathematics of Control, Signals and Systems*, 1(3):203–226, 1988.

[42] WA Sethares, Iven MY Mareels, Brian Anderson, C Richard Johnson Jr, and Robert R Bitmead. Excitation conditions for signed regressor least mean squares adaptation. *Circuits and Systems, IEEE Transactions on*, 35(6):613–624, 1988.

[43] Charles H Knapp and G Clifford Carter. The generalized correlation method for estimation of time delay. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 24(4):320–327, 1976.

[44] H Kurz and W Goedecke. Digital parameter-adaptive control of processes with unknown dead time. *Automatica*, 17(1):245–252, 1981.

[45] PJ Gawthrop and MT Nihtilä. Identification of time delays using a polynomial identification method. *Systems & control letters*, 5(4):267–271, 1985.

[46] Ashraf Elnaggar, Guy Dumont, A-L Elshafei, et al. New method for delay estimation. In *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*, pages 1629–1630. IEEE, 1990.

[47] Wei-Xing Zheng and Chun-Bo Feng. Identification of stochastic time lag systems in the presence of colored noise. *Automatica*, 26(4):769–779, 1990.

[48] Gianni Ferretti, Claudio Maffezzoni, and Riccardo Scattolini. Recursive estimation of time delay in sampled systems. *Automatica*, 27(4):653–661, 1991.

[49] VB Kolmanovskii, Silviu-Iulian Niculescu, and Keqin Gu. Delay effects on stability: A survey. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, volume 2, pages 1993–1998. IEEE, 1999.

[50] Qing-Guo Wang and Yong Zhang. Robust identification of continuous systems with dead-time from step responses. *Automatica*, 37(3):377–390, 2001.

[51] Su Whan Sung and In-Beum Lee. Prediction error identification method for continuous-time processes with time delay. *Industrial & engineering chemistry research*, 40(24): 5743–5751, 2001.

[52] Yury Orlov, Lotfi Belkoura, Jean-Pierre Richard, and Michel Dambrine. On identifiability of linear time-delay systems. *Automatic Control, IEEE Transactions on*, 47(8): 1319–1324, 2002.

[53] Ahmad B Rad, Wai Lun Lo, and KM Tsang. Simultaneous online identification of rational dynamics and time delay: A correlation-based approach. *Control Systems Technology, IEEE Transactions on*, 11(6):957–959, 2003.

[54] Jean-Pierre Richard. Time-delay systems: an overview of some recent advances and open problems. *automatica*, 39(10):1667–1694, 2003.

[55] Tao Zhang and Yuan-chun Li. A fuzzy smith control of time-varying delay systems based on time delay identification. In *Machine Learning and Cybernetics, 2003 International Conference on*, volume 1, pages 614–619. IEEE, 2003.

[56] Yury Orlov, Lotfi Belkoura, Jean-Pierre Richard, and Michel Dambrine. Adaptive identification of linear time-delay systems. *International Journal of Robust and Nonlinear Control*, 13(9):857–872, 2003.

[57] Wei Gao, Yuanchun Li, Guangjun Liu, and Tao Zhang. An adaptive fuzzy smith control of time-varying processes with dominant and variable delay. In *American Control Conference, 2003. Proceedings of the 2003*, volume 1, pages 220–224. IEEE, 2003.

[58] Wei Gao, Miao-Lei Zhou, Yuan-Chun Li, and Tao Zhang. An adaptive generalized predictive control of time-varying delay system. In *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, volume 2, pages 878–881. IEEE, 2004.

[59] P. T. Chan X. M. Ren, A. B. Rad and W. L. Lo. Online identification of continuous-time systems with unknown time delay. *IEEE Transactions on Automatic Control*, 50(9): 1419, 2005.

[60] El-Kébir Boukas. Discrete-time systems with time-varying time delay: stability and stabilizability. *Mathematical Problems in Engineering*, 2006, 2006.

[61] S Ahmed, B Huang, and SL Shah. Parameter and delay estimation of continuous-time models using a linear filter. *Journal of Process Control*, 16(4):323–331, 2006.

[62] Sergey V Drakunov, Wilfrid Perruquetti, J-P Richard, and Lotfi Belkoura. Delay identification in time-delay systems using variable structure observers. *Annual reviews in control*, 30(2):143–158, 2006.

[63] Sreten B Stojanović, Dragutin Lj Debeljković, and Ilija Mladenović. A lyapunov-krasovskii methodology for asymptotic stability of discrete time delay systems. *Serbian Journal of Electrical Engineering*, 4(2):109–117, 2007.

[64] Oscar Gomez, Yury Orlov, and Ilya V Kolmanovsky. On-line identification of SISO linear time-invariant delay systems from output measurements. *Automatica*, 43(12): 2060–2069, 2007.

[65] Tao Zhang and Yan-qiu Cui. A bilateral control of teleoperators based on time delay identification. In *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, pages 797–802. IEEE, 2008.

[66] M De la Sen. Robust adaptive control of linear time-delay systems with point time-varying delays via multiestimation. *Applied mathematical modelling*, 33(2):959–977, 2009.

[67] Saïda Bedoui, Majda Ltaief, Kamel Abderrahim, and Ridha Ben Abdennour. Representation and control of time delay system: Multimodel approach. In *Systems, Signals and Devices (SSD), 2011 8th International Multi-Conference on*, pages 1–6. IEEE, 2011.

[68] Saïda Bedoui, Majda Ltaief, and Kamel Abderrahim. New results on discrete-time delay systems identification. *International Journal of Automation and Computing*, 9(6): 570–577, 2012.