Abstract
This paper presents a control scheme of engine-dynamometer system performing real gasoline engine operation in virtual driver-vehicle-road simulation conditions. The focus is on the transient behavior of the dynamometer speed control during engine-in-the-loop simulation, and a control-oriented model of the dynamometer is constructed. Based on the model, the generalized predictive controller is designed as the external control loop of the existing dynamometer controller to improve the response of the engine-dynamometer system, which reduces the synchronizing speed error between actual engine-dynamometer and the virtual driveline-vehicle model. The performance of the proposed control scheme is verified through comparison experiments including using the original control without the external control loop and a conventional proportion-differentiation control in the external control loop. The results indicate that the generalized predictive control yields significant improvement in terms of response ability. Finally, a speed following test using the proposed control scheme is conducted, which demonstrates the fast response of the engine-dynamometer system during a transient simulation process.

Key words: Engine-in-the-Loop Simulation, Dynamometer Control, Generalized Predictive Control

1. Introduction

Transient control has recently become a major concern in research into the internal combustion (IC) engine because of its tremendous influence on engine fuel consumption and emission\(^1\)\(^-\)\(^4\). To test engine control performance under different operational conditions, engine-dynamometer test benches are widely used in the automotive industry\(^5\),\(^6\). In such systems, the dynamometer is driven to follow a pre-programmed torque or speed profile, which is calculated based on the operating mode and route conditions. This test configuration is sufficient to evaluate engine performance under fixed operating profiles. However, it is not sufficient to provide a validity test for the engines operated by any drive style on any given route. The engine-in-the-loop simulation (EILS) system has been the focus of much attention, since the driver-engine-vehicle-route loop is emulated in real time. In the system, the physical engine-dynamometer system is coupled with a virtual vehicle mathematical model. The dynamometer acts as an external load generator and adjusts the engine’s operational state in real time. The key to such a system is to guarantee that the virtual vehicle model is driven by the actual torque generated by the real engine. The control issue of the dynamometer is how to achieve this testing environment by driving the dynamometer, the low-inertia electrical machine.

Nevertheless, in commercial dynamometers, the transient process during speed or torque control could generate an undesired delay in response to the speed track due to induction-to-power delays and dynamometer nonlinearities. Furthermore, there is a coupled relationship between these two control plants (gasoline engine and dynamometer), which influences the engine’s simulation performance in the closed loop control. Therefore, an accurate mathe-
matical model and a better understanding of the dynamic characteristics of this system are essential to realize precise control.

Many previous studies of the construction and control of EILS have been reported. Babbit et al. proposed the implementation details and test results for a transient engine dynamometer system\(^{(7),(8)}\). In such implementations, the transient engine performance was calibrated, and the dynamometer worked under the dynamic conditions to emulate the real road load. Hosam et al. reviewed the development and applications of the EILS system and verified the transient emission performance of conventional and hybrid powertrains\(^{(9)}\). Bunker et al. presented a robust multivariable feedback controller to avoid excessive loop interactions in this closed-loop system. In their research, an engine-dynamometer model used for controller design was developed by spectral estimation techniques and step responses calibration methods\(^{(10)}\). Jose designed a decoupled PI action controller for an engine-dynamometer system based on a transfer function model in order to improve the control response of the test bed\(^{(11)}\).

The literatures offer some inspirations and insights into methodologies to validate engine transient behavior via integration with a virtual vehicle model and road conditions. Indeed, fast response of the dynamometer control is required in these studies.

This paper mainly focuses on the problem of control design for the dynamometer such that it reduces the synchronized error between the speed response of the vehicle model and the dynamometer under the configuration described. To this end, a simplified kinetic model is presented first, and based on the model, a generalized predictive control (GPC) scheme is deduced for the dynamometer. The quick response of the dynamometer when the virtual vehicle is driven by the actual engine torque is the aim. Finally, experimental results of the dynamometer control using a conventional proportion-differentiation (PD) feed-forward control and a GPC algorithm were compared to verify the effects of the proposed control scheme, and a speed following test was demonstrated.

2. System Overview and Problem Description

The configuration of the EILS system is shown in Fig.1. The system consists of a real engine, a driver model, a virtual driveline-vehicle dynamics model, and the low inertia electrical dynamometer. The driver inputs three signals to the system, the accelerator demand, the gear shift, and the braking demand, while engine electronic control unit (ECU) drives the engine according to the acceleration demand in real-time, and the actual torque generated by the engine forces the mathematical model of the driveline and vehicle to move under the virtual environment and load conditions. The purpose of this system is to emulate the situation in which the driver drives the vehicle with a real engine on the road. Thus, the dynamometer must be controlled to guarantee the reality of this situation, especially during the transient process.

![Fig. 1 Control structure of the engine-in-the-loop simulation system.](image-url)
In this research, the V6 gasoline engine (Toyota Motor Inc., Japan) is adopted, and the low inertia dynamometer (Horiba Inc., Japan) runs in the speed control mode, with a PID control algorithm in its controller (SPARC). By this inner control loop of SPARC, the dynamometer forces the engine to work under the virtual road conditions.

For example, when the road condition is straight and flat, the system is operated according to driver demands including the acceleration signal and braking command shown in Fig. 2(a); the actual engine torque and the vehicle speed are respectively shown in (b) and (c); and the reference engine speed (solid line in (d)) yielded from the mathematical model must be reproduced synchronously by dynamometer speed (dotted and dashed line in (d)). In fact, it is better to have a small synchronizing error between the speed of the virtual model and the actual engine-dynamometer system. However, owing to the signal propagation delay and the causality of the physical system, it is difficult to keep the synchronizing error at zero. In reality, the error between the reference speed command and the actual response of the dynamometer can reach 500 rpm, as shown in (e). Besides, the lag time of the dynamometer speed control response is about 300 ms during the transient test process, which seriously affects the precision of the simulation. Hence, an important challenge here is how to coordinate engine-dynamometer transient control responding to the commands from virtual models during the simulation process. Accurate transient control, especially for the dynamometer control system, and fast dynamic response are required to guarantee the safety and control precision of the experiments. The dynamic characteristics of the dynamometer system must be understood before testing.

Fig. 2 Example of EILS simulation test.

The problem addressed in this paper is as follows: for any given acceleration signal demanded by the driver, to find a real-time control law to provide the speed command for the dynamometer’s controller, such that the dynamometer speed tracks the speed command of the driveline-vehicle model with satisfactory accuracy and rapidity. In view of the existing control scheme in SPARC is unchangeable and the speed tracking performance is constrained by the
limited gain choice, an external control loop based on the GPC is introduced to improve the speed tracking performance. This achieves accurate simulation for the real engine transient control in the virtual driving conditions. Fig.3 represents the control diagram for speed control of the dynamometer. The reference engine speed command for SPARC given by the driveline-vehicle model will be regarded as the control input of the external control loop, and the actual rotational speed of the dynamometer is the control output.

3. Modeling

3.1. Model Structure

In the system, the engine crankshaft is connected to the dynamometer with a gear box. In order to obtain the kinematical model for the dynamometer, the engine and the dynamometer are simply regarded as two rigid bodies connected with a viscoelastic shaft, as shown in Fig.4, where \( J_e \) and \( J_d \) denote the inertia coefficient of the engine and the dynamometer, and \( k \) and \( \beta \) are the stiffness and flexibility coefficients of the equivalent shaft, respectively. \( T_e \) represents the engine torque, and \( T_d \) is the electromagnetic torque of the dynamometer.

The dynamic rotation model of the dynamometer can then be described as follows:

\[
J_d \dot{\omega}_d = k \int (\omega_e - \omega_d) dt + \beta (\omega_e - \omega_d) - T_d - \beta_d \dot{\omega}_d
\]

(1)

where, \( \beta_d \) is the rotation damping coefficient, which is determined by the internal physical structure of the dynamometer.

Given that the electromagnetic torque of the dynamometer is generated by inner-loop control (SPARC controller), the transient dynamics can be simply represented as the first-order system driven by the inner-loop control actuation, i.e., the sum of the proportional and integral of speed tracking error between command speed and the actual speed, since the dynamometer is under PI control based on the speed error. Therefore, the dynamics of the electromagnetic torque can be expressed as follows:

\[
T_d = -a T_d + b (\omega_r - \omega_d) + c \int (\omega_r - \omega_d) dt
\]

(2)

where, \( \omega_r \) is the command speed of the dynamometer, and \( a, b, \) and \( c \) are constants that relates to the response time of the system and the controller performance.

3.2. Parameters Identification

The parameters of the model (1) and (2) can be obtained by applying the well-known identification algorithm for the linear system. To this end, we rewrite the system model as a
linear regression expression by instituting (1) into (2), which yields,
\[ J_d \ddot{\omega}_d = k(\omega_r - \omega_d) + \beta(\omega_r - \omega_d) - aT_d - b(\omega_r - \omega_d) - c(\theta_r - \theta_d) - \beta_d \omega_d \]  
where, \( \theta_d \) and \( \theta_r \) denote the rotation angle of the dynamometer shaft and the command angle, respectively. They can be regarded as the integrals of the corresponding speed variables, that is, \( \theta_d = \int \omega_d dt, \theta_r = \int \omega_r dt. \)

The model is a single-input-single-output (SISO) system, in which input is the speed command and output is the actual speed of the dynamometer. Suppose the sampling period is \( \Delta T \), the equation (3) can then be written also in the following discrete-time form with white noise item \( \varepsilon(k) \):
\[ \omega_d(k + 1) = a_1 \omega_d(k) + a_2 \omega_d(k - 1) + a_3 \omega_d(k) + a_4 \omega_d(k - 1) + a_5 \omega_d(k) + a_6 \theta_d(k) + a_7 T_d(k) + \varepsilon(k) \]  
where, \( a_1 - a_7 \) are the parameters to be estimated and they are defined as follows,
\[ a_1 = 2 + \frac{\Delta T^2}{J_d} (b - k - \frac{\beta}{\Delta T} - \frac{\beta_d}{\Delta T}), \]
\[ a_2 = \frac{\Delta T}{J_d} (\beta + \beta_d), \quad a_3 = \frac{\Delta T}{J_d} (k + \frac{\beta}{\Delta T}), \]
\[ a_4 = -\frac{\Delta T}{J_d} b, \quad a_5 = -\frac{\Delta T}{J_d} a, \]
\[ a_6 = -\frac{\Delta T^2}{J_d} c, \quad a_7 = -\frac{\Delta T^2}{J_d} a. \]

Based on the above discrete model (4), the recursive least square (RLS) method is employed to estimate these unknown parameters. The experimental data used for parameter identification can be collected via the CAN bus, and the sampling period \( \Delta T = 0.01 \) s. In order to obtain accurate estimation results, we implemented the only system input, namely the speed control command of the dynamometer \( \omega_r \) (shown in Fig.5(a)), to act like a pseudo-random binary sequence (PRBS), which can sufficiently motivate the system dynamic characteristic. The corresponding dynamometer responses include the actual speed \( \omega_d \) and electromagnetic torque \( T_d \) which are measured by the sensors inside the dynamometer, as also shown in Fig.5.

![Fig. 5 The experimental data used for model identification.](image)

Applying the RLS identification algorithm, the estimated parameters converge to constants along with the identification process, which is illustrated in Fig.6. Table 1 lists the identification results of the model parameters.

In order to evaluate the model’s accuracy, another experiment was conducted, and the comparison result between the actual speed and the model’s estimated speed is shown in
Table 1 Identification results of the dynamometer model parameters

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
<th>$a_6$</th>
<th>$a_7$</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
<th>$d$</th>
<th>$k$</th>
<th>$\beta k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5.7924</td>
<td>-9.1802</td>
<td>-1.4429</td>
<td>1.05</td>
<td>0.0076602</td>
<td>983.0082</td>
</tr>
</tbody>
</table>

Fig. 6 The convergence process of identification parameters.

Fig. 7. The estimation error between the estimated and the measured results is generally within a margin of 10%. Thus, the proposed model can be deemed acceptable for the controller design.

4. Controller Design

A key specification of EILS is real-time performance, especially in transient operation modes. From the point of controller design for the engine-dynamometer system, the actual speed of the shaft driven by both engine and dynamometer must track the speed of virtual driveline-vehicle with sufficient rapidity and accuracy when the virtual vehicle is driven by the actual engine torque. Thus, for the dynamometer speed control loop that is designed here, the reference signal is provided by the virtual model of the driveline-vehicle. It means that, during real-time simulation, the future values of the reference signal can be calculated in advance based on the mathematical model. This is an incentive to introduce predictive-based control to improve the transient performance. Moreover, GPC control has good robustness,
and is widely applied in engineering practice. In the following, a GPC control scheme is designed.

The objective of the proposed controller is to find a control law for the speed command of the dynamometer such that the actual speed of the dynamometer follows the reference engine speed \( \omega_m \) as quickly as possible, where the reference engine speed is the response of the driveline and vehicle model driven by engine torque under any given acceleration signal. Therefore, to find a desired control \( \omega_r(k) \) at the step \( k \), the following cost function is introduced:

\[
J = \sum_{i=1}^{N_p} (\omega_m(k + i) - \hat{\omega}_d(k + i \hat{\eta}k))^2 + \sum_{j=1}^{N_c} r_j \Delta \omega_r(k + j - 1)^2
\]  

where, \( N_p \) and \( N_c \) are the predictive horizon and control horizon, respectively. \( \omega_m(k + i) \) denotes the response of the driveline and vehicle model at the \( k + i(\hat{\eta}k) \) step and \( \hat{\omega}_d(k + i \hat{\eta}k) \) denotes the model-based i-step ahead prediction from step \( k \), which can be obtained by repeating the one-step ahead prediction with the system model (1) and (2). The desired control is given by the incremental form, \( \Delta \omega_r(k + j) = \omega_r(k + j) - \omega_r(k + j - 1) \). \( r_j \) is the weighting factor of the control variable.

Indeed, solving this optimization problem is a kind of receding horizon optimization process; however, the model-based generalized predictive of \( \hat{\omega}_d(k + i \hat{\eta}k) \) is used in the cost function, and it can be solved by applying the generalized predictive control (GPC) algorithm presented in [12]. To this end, the system dynamics model mentioned in Section 3 can be represented as following a continuous-time state-space equation:

\[
\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{1}{\tau_d} & -\frac{\theta_d b}{\tau_d} & -1 & 0 \\ 0 & -b & -a & c \\ 0 & -1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ b \\ 1 \end{bmatrix} \omega_r + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \xi
\]  

where, the state variables \( x = \begin{bmatrix} \theta_d & \omega_d & T_d & (\theta_r - \theta_d) \end{bmatrix}^T \), speed command \( \omega_r \) is the manipulated variable and \( \xi \) is the external disturbance, which is determined by the engine speed and can be written by the following equation:

\[
\xi = k \cdot \theta_e + \beta \cdot \omega_e.
\]

The dynamometer speed is regarded as the only output of the system; therefore, the output equation can be written as follows:

\[
y = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} x
\]  

Thus, the discrete-time state-space equations can be derived from (6) and (7) as follows:

\[
x(k + 1) = A_d x(k) + B_d \omega_r(k) + \Psi_d \xi(k)
\]

\[
y(k) = C_d x(k)
\]  

where, \( A_d, B_d, C_d \), and \( \Psi_d \) denote the discrete-time forms of the matrices \( A, B, C \), and \( \Psi \) in the continuous-time state-space equations. If \( \Delta T \) is defined as the discrete time, then they are written by following equations:

\[
A_d = e^{A \Delta T}, \quad B_d = \int_0^{\Delta T} e^{A \Delta T} dB, \quad \Psi_d = \int_0^{\Delta T} e^{A \Delta T} d\Psi, \quad C_d = C.
\]

The GPC algorithm usually consists of three basic principles including predictive model, receding optimization and feedback compensation. To solve this problem, the predictive model has to be reformulated first. The augmented predictive model can be written as follows:

\[
x_m(k + 1) = A_m x_m(k) + B_m \Delta \omega_r(k) + \Psi_m \Delta \xi(k)
\]

\[
y(k) = C_m x_m(k)
\]  

where,
\[ x_m(k) = \begin{bmatrix} x(k) - x(k-1) & y(k) \end{bmatrix}^T, \]
\[ \Delta \omega_r(k) = \omega_r(k) - \omega_r(k-1), \]
\[ \Delta \xi(k) = \xi(k) - \xi(k-1), \]
and the incremental matrices \( A_m, B_m, \Psi_m, C_m \) can be described as follows:
\[
A_m = \begin{bmatrix} A_d & O_{3 \times 1} \\ C_d A_d & 1 \end{bmatrix},
\]
\[
B_m = \begin{bmatrix} B_d \\ C_d B_d \end{bmatrix}^T,
\]
\[
\Psi_m = \begin{bmatrix} \Psi_d \\ C_d \Psi_d \end{bmatrix}^T,
\]
\[
C_m = \begin{bmatrix} O_{1 \times 4} \\ 1 \end{bmatrix}.
\]

Based on the above augmented model, the model predictive outputs of future \( Np \) steps from the \( k - th \) step can be derived by following matrix form:
\[
Y = F x_m(k) + \Phi U + \Lambda \Xi \tag{10}
\]
where,
\[
Y = \begin{bmatrix} y(k+1|k) & y(k+2|k) & \cdots & y(k+Np|k) \end{bmatrix}^T,
\]
\[
U = [\Delta \omega_r(k) \quad \Delta \omega_r(k+1) \quad \cdots \quad \Delta \omega_r(k+Nc - 1)]^T,
\]
\[
\Xi = [\Delta \xi(k) \quad \Delta \xi(k+1) \quad \cdots \quad \Delta \xi(k+Nc - 1)]^T.
\]

Here, the disturbance variables at time \( (k+1), \ldots, (k+Nc - 1) \) are unknown. Suppose that they equals the value at \( k \) time. Therefore, \( \Xi = I \Delta \xi(k) \), \( I \) is the basic unit vector with \( Nc \times 1 \) dimensions. The predictive matrices \( F, \Phi, \) and \( \Lambda \) are written by the following forms:
\[
F = \begin{bmatrix} C_mA_m & C_mA_m^2 & \cdots & C_mA_m^{Np} \end{bmatrix}^T
\]
\[
\Phi = \begin{bmatrix} C_mB_m \\ C_mA_mB_m \\ \vdots \\ C_mA_m^{Np}B_m \\ C_mA_mA_m^{Np-1}B_m & \cdots & C_mA_m^{Np-Nc}B_m \\ \vdots \\ C_mA_m^{Np}\Psi_m & \cdots & C_mA_m^{Np-Nc}\Psi_m \\ \vdots \\ C_mA_mA_m^{Np-1}\Psi_m & \cdots & C_mA_m^{Np-Nc}\Psi_m \end{bmatrix}
\]
\[
\Lambda = \begin{bmatrix} C_mA_mB_m & 0 & \cdots & 0 \\ C_mA_mB_m & C_mA_mB_m & \cdots & 0 \\ \vdots \\ C_mA_m^{Np}B_m & C_mA_m^{Np-1}B_m & \cdots & C_mA_m^{Np-Nc}B_m \\ C_mA_m\Psi_m & C_mA_m\Psi_m & \cdots & 0 \\ \vdots \\ C_mA_mA_m^{Np-1}\Psi_m & \cdots & C_mA_m^{Np-Nc}\Psi_m \end{bmatrix}
\]

The optimal control vector \( U \) can be obtained via minimizing the cost function \( J \). From \( \partial J/\partial U = 0 \), it can be expressed as follows:
\[
U = (\Phi^T \Phi + R)^{-1}\Phi^T (R_s(k) - F x_m(k) - \Lambda I \Delta \xi(k)) \tag{11}
\]
where, the item \( R_s(k) \) is the set-point data vector that represents the response of the virtual driveline and vehicle model in the future \( Np \) steps, and \( R \) is the weighting coefficient vector that determines the closed-loop control performance. Their expressions are written as follows:
\[
R_s(k) = [\omega_r(k+1) \quad \omega_r(k+2) \quad \cdots \quad \omega_r(k+Np)]^{Np \times 1}
\]
\[
R = [r_1 \quad r_2 \quad \cdots \quad r_{Nc}]^{Nc \times 1}
\]

Based on the receding horizon control principle, we take the first element of the control vector at time \( k \) as the final incremental control variable, so the control law can be written by
\[
\Delta \omega_r(k) = K(\Phi^T \Phi + R)^{-1}\Phi^T (R_s(k) - F x_m(k) - \Lambda I \Delta \xi(k)) \tag{12}
\]
\[
\omega_r(k) = \omega_r(k-1) + \Delta \omega_r(k) \tag{13}
\]
where, \( K = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^{Nc \times 1} \).
5. Experiments

5.1. Experiment Configuration

In order to verify the control effects of the proposed GPC control scheme, validation experiments have been implemented on the EILS test bench. The configuration of the test bench is shown in Fig.8. In this system, the engine ECU provides a free communication interface and permits engine control to be taken over by the external controller. A programmable controller, called dSPACE 1006 processor board, is installed as the top level engine controller to communicate with engine ECU.

The virtual system consists of two independent simulation modules. One is the driver-route control model. It can be programmed in the Simulink and compiled as the C code into the MicroAutoBox II (also produced by dSPACE Inc.), which is a programmable controller. Specifically, if given the route information such as drive cycle speed, the driver model will estimate the operation commands, including acceleration, braking, and gearshift, and then drive the virtual vehicle and perform real engine tracking this reference vehicle speed during the simulation process. Another module is the driveline and vehicle dynamics model. A commercialized simulation tool (AVL InMotion) has been adopted here to provide the accurate mathematical vehicle model and real-time motion display. Further, an automatic transmission with five gears has been adopted in the driveline model.

All the necessary data transfer among these different hardwares are achieved via the CAN bus. The engine torque measured by the torque sensor will be delivered to the InMotion and consequently drag the virtual vehicle forward. Meanwhile, the reference engine speed derived from the vehicle speed will feed back to the dynamometer controller in real time. The external speed controller is set up in order to enhance the responsiveness of the speed tracking control of the engine-dynamometer system. The dynamometer then drags the engine to follow the reference engine speed command as quickly as possible. In this closed-loop control system, the minimum lag time is crucial to the engine transient tests, and therefore the proposed GPC scheme will be applied to the external speed control loop of the dynamometer.

5.2. Experimental Results

5.2.1. Case 1: Speed tracking control for dynamometer

In this research, we have implemented two kinds of experiments to verify the effects of the proposed GPC controller, where we set the predictive horizon \( N_p = 10 \) and the control horizon \( N_c = 2 \). First, the responsiveness of the speed tracking with the external speed controller will be investigated in Case 1. It provides the comparative results for validation of the speed tracking performance of the dynamometer based on an open-loop control. In this case, the virtual models are...
neglected and the engine throttle is kept constant at 8\(\text{deg}\). The predefined reference engine speed command will be delivered to the external speed controller of the dynamometer directly. Meanwhile, in order to compare the effects of the proposed GPC controller, a proportional and differential (PD) of the tracking error is also used for the feedback control of the external control loop, and a feed forward action is introduced to obtain quick response. The PD controller is expressed as follows:

\[
\omega_r(k) = k_p \cdot e(k) + k_d \cdot (e(k) - e(k - 1)) \cdot \frac{1}{\Delta T} + \omega_m(k)
\]  

where, \(e(k) = \omega_m(k) - \omega_d(k)\). In this case, the parameters \(k_p = 0.5\) and \(k_d = 0.2\). Indeed, GPC is just multi-step ahead predictive control but the PD controller can be regarded as a one-step ahead predictive control due to the function of the differential item. The speed tracking controller is shown diagrammatically in Fig. 9.

Fig. 9 Diagram of the speed tracking controller in Case 1.

![Diagram of the speed tracking controller in Case 1.](image)

Fig. 10 Speed control response test with the dynamic inputs.

![Speed control response test with the dynamic inputs.](image)

Fig. 10 shows the comparative test results when giving the system the same dynamic speed command. The reference speed command \(\omega_m\) shown as the solid line in Fig.10(a) should be followed by the actual speed of the dynamometer as quickly as possible. The dash line represents the actual response of the dynamometer speed control without the external control loop. It is obvious that there is nearly 300\(\text{ms}\) of lag time for the dynamometer to respond to the given command. However, the lag time can be reduced by the external control loop with the proposed PD controller and GPC controller. The results of the PD control shown as the dotted line can improve the speed response time by about 100 – 120\(\text{ms}\) more than where there
is no control. However, the PD controller more easily induces a large overshoot, especially when the reference speed changes sharply during 12 – 14s. However, the result of the GPC controller shows better track performance with lower overshoot compared to the PD control. The synchronizing error between the reference speed command and actual speed is shown in Fig.10(b). It also shows that using the proposed GPC controller yields an overall smaller synchronizing error. Moreover, as one of the main state variables of the system, dynamometer torque changes along with the changes of dynamometer speed, as shown in Fig.10(c).

\[ u_m = k_{p1} \cdot e_v + k_{i1} \cdot \int e_v \, dt + k_{d1} \cdot \frac{de_v}{dt}, \quad u_m \in [-100, 100] \]  

where, \( e_v \) is the speed error between the vehicle model speed \( v_{mdl} \) and reference vehicle speed \( v_{ref} \), \( e_v = v_{mdl} - v_{ref} \). The control parameters \( k_{p1}, k_{i1}, \) and \( k_{d1} \) are set as 10, 5, and 1, respectively. The driver manipulated variable \( u_m \) can be specified as the throttle or brake control according to equation (16). In order to avoid frequent changes between the throttle control and braking

**Figure 11**  EILS experimental result with driving cycle.

### 5.2.2. Case 2: EILS closed-loop control

The closed-loop EILS experiment integration with the virtual driver-vehicle-route model presented in Fig.8 is implemented in Case 2. The driver model adopts the conventional PID control scheme to control the engine throttle and brake pedal angle to make the vehicle speed tracking the given drive cycle. The driver model is described as follows,

\[ u_m = k_{p1} \cdot e_v + k_{i1} \cdot \int e_v \, dt + k_{d1} \cdot \frac{de_v}{dt}, \quad u_m \in [-100, 100] \]  

The driver manipulated variable \( u_m \) can be specified as the throttle or brake control according to equation (16). In order to avoid frequent changes between the throttle control and braking
control, an idling buffer area has been set up in this research.

\[
\begin{align*}
\text{Throttle Control} & : u_{thr} = u_m, \quad 3 \leq u_m \leq 100 \\
\text{Idling Control} & : u_{idl} = 0, \quad -2 < u_m < 2 \\
\text{Braking Control} & : u_{bra} = -u_m, \quad -100 \leq u_m \leq -2
\end{align*}
\]

(16)

In order to emulate the operation conditions of the road vehicle, the proposed GPC controller is adopted to improve the speed response of the engine-dynamometer system to the virtual model.

Fig.11 shows a typical experimental result with the drive cycle using the proposed EILS test bench. In this experiment, the engine idling speed is set as 800rpm. The engine torque used in the vehicle model is received from the torque sensor in real-time and its low limit is set as 0Nm. During the simulation process, the driver inputs such as throttle and braking signals are shown in Fig.11(a). The real engine torque (shown in (b)) corresponding to the throttle opening drives the virtual vehicle speed \(v_{mdl}\) tracking with the reference vehicle speed \(v_{ref}\) (solid line in (c)). Meanwhile, the reference engine speed \(\omega_m\) is transmitted to the GPC controller and forces the actual speed of the dynamometer \(\omega_d\) to follow it. Owing to the automatic transmission with five gears adopted in the driveline model of the InMotion software, the gearshift is controlled by the internal control algorithm of InMotion, as shown in Fig.11(e).

Fig. 12 EILS experimental result with driving cycle.

The experimental result represents the feasibility of the proposed EILS system combined with the virtual models of route-driver-driveline-vehicle. Using the external speed controller, the engines can achieve the desired work-points quickly. The local detailed performance of the experiment is extracted and shown in Fig.12. In comparison to the results without the external controller shown in Fig.2, it is obvious that the proposed GPC control scheme reduces the synchronizing error between the reference engine speed calculated by driveline model and the actual speed. It also enhances the responsiveness of the engine-dynamometer system corresponding to the virtual model in the dynamic simulation process. In addition, the test bench provides an emulational operational environment for future research on transient fuel efficiency and emission.
6. Conclusion

A modeling and control method for the engine-dynamometer simulation test bench has been proposed in this paper. In order to precisely conduct engine-in-the-loop tests, the smaller lag time for the engine-dynamometer system responding to the requirement of virtual models is crucial to the precision of the simulation. To this end, an external control loop of the dynamometer was designed, in which the GPC control scheme has been represented to reduce the synchronizing error between the reference engine speed calculated by driveline model and actual speed of the dynamometer. The test results show that the proposed controller enhances the responsiveness of the engine-dynamometer in such a closed-loop control system. Compared with PD feed forward control, the proposed GPC controller obtains better control effects and avoids an unexpected overshoot during the dynamic speed tracking control.

In addition, EILS experiments using GPC control have been introduced in this paper. With the virtual driver-vehicle-route model, the test bench can emulate the engine operation conditions as far as possible. The GPC controller improves the control performance and helps the proposed EILS test bench to achieve accurate simulation tests. It can be further used in the development of fuel efficiency technology and other engine transient control methods in the future.

References