PB model-based FEL and its application to Driving Pattern Learning

Luka Eciolaza * Tadanari Taniguchi ** Michio Sugeno *
Dimitar Filev *** Yan Wang *** John Michelinı ***

* European Centre for Soft Computing, 33600 Mieres, Asturias, Spain (e-mail: luka.eciolaza@softcomputing.es, michio.sugeno@gmail.com).
** Tokai University, Hiratsuka, Kanagawa, 2591292 Japan (e-mail: taniguchi@tokai-u.jp).
*** Ford Research and Innovation Center, 2101 Village Road, Dearborn, MI 48103, USA (e-mail: dfilev@ford.com, ywang21@ford.com, jmichel1@ford.com).

Abstract:
A significant part of the vehicle control software is based on Look-Up-Tables (LUTs) that define the mappings from different combinations of multiple independent (input) variables to the values of one dependent (output). LUTs are used as feedforward controllers or as gain-scheduling parameters for feedback controllers. The LUT feedforward controllers can be viewed as inverse vehicle models capturing the strong nonlinearity and multimodal behaviors that can be (in many cases) formalized only by experimentally measured data under different operating conditions. The paper proposes a Feedback Error Learning (FEL) based method for adaptation of the LUT feedforward controllers in order to match the desired and actual vehicle performance. The FEL is an on-line learning strategy acquiring an inverse model of a plant through feedback control actions. In this paper we consider the driver demand LUTs as a feedforward controller defining the relationship between the accelerator pedal position, the engine speed, and corresponding brake torque and the driver as a feedback controller. The FEL scheme have been implemented through Piecewise Bilinear (PB) models which can be expressed as LUTs and are very convenient with regard to nonlinear modeling, control objective and on-line learning capability.

1. INTRODUCTION

Look-up tables (LUTs) are used in automotive engineering applications as feedforward controllers or as gain-scheduling parameters for feedback controllers. In a broad sense they represent “pseudo-equations to make up for a lack of real equations or perhaps to replace complicated equations with simpler ones” (Bernstein [2005]). For example, the engine control systems employ from 5, 500 to 6, 000 LUTs that contain calibration parameters or define engine control actions under different operating conditions (C. Bohn and Magnor [2006]). The most common LUTs are the two-dimensional (2D) tables that define the values of one dependent (output) variable for different combinations of two independent (input) variables. Some of the reasons for the wide acceptance of the LUTs in the automotive industry are the strong nonlinearity and multimodal behaviors that can be (in many cases) formalized only by experimentally measured data under different operating conditions. The LUT feedforward controllers can be viewed as inverse vehicle models capturing the entire vehicle operation under different operating conditions. Such control is designed to fit normal driver and environment and have to compromise performance otherwise.

The typical approach to addressing the problem of adaptation of the LUT based feedforward controllers is to use the difference between the desired and the actual vehicle performance to realize an adaptation of the LUTs so as to match the two performances. Because all the LUTs are built for nominal conditions if the vehicle operates under such conditions, the vehicle performance with inputs defined by the LUTs outputs (actuators) usually match the desired performance which is the input to the LUTs. Any difference of the performance indicates the deviation from the nominal conditions (driver and/or environment). It is then possible to consider an algorithm for the modification of the LUTs to improve performance matching the desired performance.

This conventional approach of adaptation is not always applicable to the vehicle feedforward LUT controllers because the driver is not an independent controller but a part of a human-in-the-loop system. Fig. 1 shows a general view of the driver-vehicle interaction (Q. Zhao and Filev [2013]). The drivers control actions that are communicated to the vehicle subsystems through the Human Machine Interaction (HMI), steering (S), accelerator (A), and brake (B) pedals are a result of a complex process driven by two main information flows - a cognitive flow including the quantitative information readily available to the driver through various sensors, control and advisory systems (solid lines) and a subjective flow including visual, perceptual, emotional, and experiential factors that are processed in the driver’s mind (broken line). While the quantitative information flow is available to both the driver and the vehicle systems, the subjective information flow is generally inaccessible to the electronic control systems. In addition, the drivers objectives (target speed, acceleration, torque, etc.) are in general not explicitly available to the vehicle control systems. Because of
these reasons it is not always easy or even possible to apply the conventional approach of developing an adaptive strategy that would adjust the feedforward LUT controllers to the actual vehicle performance.

In order to address the problem of adaptation of the vehicle feedforward LUT controllers in this paper we focus our attention on a different adaptation mechanism called the Feedback Error Learning (FEL) scheme, proposed by Kawato et al. [1987], that was originally inspired on biological control systems which usually are systems with time delays and strong nonlinearities. Fast and coordinated limb movements cannot be executed solely under feedback control, since biological feedback loops are slow and have small gains. Thus, the brain uses inverse dynamical models in order to calculate necessary feed-forward commands from desired trajectory information, see Kawato [1999]. FEL and the inverse model identification scheme represent an important role for the quick and smooth motions of the limbs in human motor control. The FEL has the feature that the learning and control can be performed simultaneously.

In most cases, human control actions are based on feedforward motions as seen in playing tennis or baseball. In fact, the brain has capability to acquire feedforward motions as seen in playing tennis or baseball. In fact, it is very effective to introduce feedforward controllers as a brain-style man-machine interface to assist driver’s control actions.

A typical schematic of the FEL control system is shown in Fig. 2. It consists of the combination of a feedback controller that ensures the stability of the system and an adaptive feedforward controller that improves control performance. An inverse model as a feedforward controller is learned by FEL so as to minimize the square error between $u_{ff}$, the output of a feedforward controller and $u_0$, the ideal feedforward control based on an inverse model. Using a conventional steepest descent method for the minimization, the parameters of a feedforward controller can be sequentially updated in proportion to $(u_{ff} - u_0)$. However, since the ideal $u_0$ is unknown, this error signal is approximated with $u_{fb}$ called feedback error in FEL. As it is shown in Fig. 2, $u$ (the control input to a plant) is equal to $u_{ff} + u_{fb}$, and $u_{ff} = u_{0}$ is replaced with $u_{ff} = u = u_{tb}$. After learning is complete, i.e., $y(t) = r(t)$, then $u_{fb}$ tends to zero and feedback control is replaced by feedforward control. Thus, $u = u_{ff}$ the feedforward controller should be serving as the inverse of the original plant.

In this paper, we have taken the approach to implement FEL based on Piecewise Bilinear (PB) models recently proposed in Eciolaza et al. [2013]. In this work we concluded the PB model represents the most convenient model to implement FEL with regard to nonlinear modeling, control objective and on-line learning capability.

FEL scheme has been successfully applied to a variety of control tasks in the past, as in Jung and Kim [2008], Ruan and Chen [2011], Nakamura et al. [2011]. However the inverse model identification was normally implemented via neural networks (NN), which are not convenient for control purpose. On the other hand, the successes of FEL attracted attention in control system community, so Miyamura and Kimura [2002], Miyamura Ideta [2006] and AlAli et al. [2006], proved FEL’s stability for SISO linear systems with a null relative degree, systems with time delay and MIMO plants respectively. However, these analyses were only made for linear systems.

The PB model, proposed in Sugeno [1999], is very effective for modeling and control of nonlinear systems. It is a fully parametric model to represent Linear/Nonlinear systems. PB model has a big number of parameters to tune, but Eciolaza and Sugeno [2012] showed that the global tuning of PB models can be made in a very simple and efficient way. PB models are very convenient for control purpose as they are interpretable (for instance as LUT), easy to compute and simple to handle.

An example of the potential of using FEL in automotive control is for learning the driver pattern during vehicle launch or tip-in, and adjusting vehicle control look-up tables to match the driver’s demand. Fig. 3 shows the pedal profiles of an aggressive driver, dashed line named Actual performance, compared to a typical driver, line named Nominal performance, during vehicle launch.

In this case, the vehicle controller (LUT) serves a feedforward controller that is the inverted vehicle plant and delivers control signals based on driver demands. The initial LUT is calibrated for a normal driver, with a nominal pedal profile. Therefore, the controller does not deliver the desired vehicle acceleration.
for an aggressive driver with a different pedal profile. It is then
desirable to have an adaptive mechanism to learn the driver’s
behavior and adjust the controller to match his/her behavior.
The difference between the nominal and actual pedal profile
will prompt the correction of the LUTs to deliver the response
the driver wants.

The algorithm adapts the fixed-parameter vehicle control sys-
tem to maintain desired performance when the vehicle is driven
at varying conditions such as, different driver, different environ-
ment, and different vehicle conditions. Such condition changes
will make vehicle performance deviating from nominal.

Sections II introduces the PB model-based FEL scheme, where
both PB models and FEL architecture are explained. Section III
describes the vehicle control application and the implementa-
tion of FEL within it. Section IV presents experimental results.
Section VVI concludes the paper with some discussion and final
remarks of the results.

2. PB MODEL-BASED FEL

As is explained in the previous section, Fig. 2 illustrates the
feedback error learning architecture. It consists of an objec-
tive plant to be controlled with a feedback controller and, in
addition, a feedforward controller to be learned by FEL. The
objective of control is to minimize the error e between the
reference signal r and the plant output y.

If the mapping from the motor command u to the sensory
outcome y is given by p : u → y, for a given reference
input r, the ideal control command, which makes the tracking
perfect, is given by the inverse u(t) = p−1(r(t)). However, the
direct inverse modeling, using I/O data, does not necessarily
ensure achieving a particular target trajectory even when the
training period is sufficiently long Kawato [1990].

In the FEL approach, the feedback controller action is converted
into motor command error and used to learn the feedforward
controller. By FEL, eventually the feedback control is replaced
by feedforward control. The learning algorithm used in FEL
is the same independent of models: NN model, linear model
and our PB model. FEL includes the features of simultaneous
learning and control, making it an adaptive controller.

2.1 PB models

The PB model is a fully parametric model to represent Lin-
ear/Nonlinear systems. It is designed to be easily applicable
for control purpose. In the model, bilinear functions are used to
regionally approximate any given function. The obtained model
is built on piecewise rectangular regions, and each region is
defined by four vertices partitioning the state space. A bilinear
function is a nonlinear function of the form
\[ y = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_n x_n \]
and any four points in the three dimensional space
are spanned with a bi-affine plane. PB system has a continuous
crossing over the piecewise regions and it is a very good general
approximator for nonlinear functions. A PB model can be ex-
pressed as a LUT which is widely used to realize controllers in
industrial applications, due to its interpretability, visibility and
simplicity. A local error does not trigger a global error and its
interpolation nature generates robust outputs.

If a general case of an affine two-dimensional nonlinear control
system is considered,

![Fig. 4. Piecewise region \( R_{\sigma \tau} \) and interpolation of \( f_i(x) \).](image)

\[
\begin{align*}
\dot{x}_1 &= f_1(x_1, x_2) \\
\dot{x}_2 &= f_2(x_1, x_2) + g(x_1, x_2) \cdot r \\
y &= h(x_1, x_2)
\end{align*}
\]
and
\[ \alpha = \frac{x_1 - d_1(\sigma)}{d_1(\sigma + 1) - d_1(\sigma)} \] (9)
\[ \beta = \frac{x_2 - d_2(\tau)}{d_2(\tau + 1) - d_2(\tau)} \] (10)
in which case \( w_1, w_2 \in [0; 1] \).

In every region of the PB models, i.e.: \( f_i(x_1, x_2) \), the values are computed through bilinear interpolation of the corresponding four vertexes as shown in Fig. 4. Note that the approximation is made by only using the values of a nonlinear function at the vertexes of \( R_{ij} \)'s in (3).

2.2 FEL: On-line Sequential Learning Algorithm

Let us express a plant model for the sake of simplicity as \( y = p(u) \) where \( y \) is output and \( u \) control input. We also denote its inverse in a similar manner as \( u = p^{-1}(y) \), assuming that a plant is invertible. We note that an inverse model used in FEL is usually written as \( u_{ff} = p^{-1}(r) \). We consider a feedforward controller expressed as \( u_{ff} = p^{-1}(r, \hat{r}) \), where \( r \) is a desired output \( y_d \) and \( \hat{r} \) is \( y_d \).

In what follows, we deal with a feedforward controller with two inputs \( r, \hat{r} \) and single output \( u_{ff} \). This feedforward controller in the FEL scheme is interpreted as a pseudo-inverse model of an objective plant. Note that a plant is, in general, not invertible unless its relative degree is 0. We will initially assume that an objective plant is unknown and its feedback controller is given. In a realistic nonlinear system control scenario, both plant identification and controller design could also be performed through PB models.

Let \( u_0 \) be an ideal feedforward control based on a pseudo-inverse model. We design a feedback controller so that \( u_{ff} = u_0 \). That is, we learn \( u_{ff} \), i.e., identify the feedforward controller parameters, to minimize the performance index
\[ I = \frac{(u_{ff} - u_0)^2}{2} \] (11)
where \( u_{ff} \) is the representation of the feedforward controller is,
\[ u_{ff} = p^{-1}(r, \hat{r}) = \sum_{i=\sigma}^{\sigma+1} \sum_{j=\tau}^{\tau+1} w_{ij}^1(r)w_{ij}^2(\hat{r})V(i, j) \] (12)
\[ I \] can be sequentially minimized using the derivative of (11)
\[ \frac{\partial I}{\partial V} = \frac{\partial u_{ff}}{\partial V}(u_{ff} - u_0) \]. (13)

However, the error \( (u_{ff} - u_0) \) is not available since \( u_0 \) is unknown. Therefore Kawato Kawato et al. [1987] suggested to use \( u_{fb} \) for \( (u_{ff} - u_0) \) since \( u = u_{fb} + u_{ff} \). This is why \( u_{fb} \) is called a feedback error playing a role as the error \( (u_{ff} - u_0) \). FEL is a learning scheme based on a signal \( u_{fb} \), a feedback error signal.

Then we have
\[ \frac{\partial I}{\partial V} = \frac{\partial u_{ff}}{\partial V} u_{fb} = \frac{\partial p^{-1}(r, \hat{r})}{\partial V} u_{fb} \] (14)

The sequential learning of each vertex of a region is made using the following algorithm:
\[ V_{new(i,j)} = V_{old(i,j)} - \delta \frac{\partial u_{ff}}{\partial V(i, j)} u_{fb} \] (15)
where \( \delta \) is an adjustable parameter as a learning rate. This is the conventional steepest descent algorithm to minimize a performance index. If learning is successfully completed, i.e., \( V_{new} = V_{old} \), then \( u_{fb} \) must become zero, and only \( u_{ff} \) works.

In the case of a two dimensional PB model, if we develop (12), with (8) and (9), we have:
\[ u_{ff} = (1 - \alpha)(1 - \beta)V_{\sigma, \tau} + (1 - \alpha)\beta V_{\sigma, \tau+1} + \alpha(1 - \beta)V_{\sigma+1, \tau} + \alpha\beta V_{\sigma+1, \tau+1} \] (16)
\[ (V_{new}, V_{old}) \] refer to the values of 4 vertexes of a region and as the function is linear, the calculation of partial derivatives \( \nabla p^{-1} \) is straightforward. The equations of change for each vertex of a region will be:
\[ \begin{aligned}
V_{new,\sigma,\tau} &= V_{old,\sigma,\tau} - \delta(1 - \alpha)(1 - \beta)u_{fb}, \\
V_{new,\sigma,\tau+1} &= V_{old,\sigma,\tau+1} + \delta(1 - \alpha)\beta u_{fb}, \\
V_{new,\sigma+1,\tau} &= V_{old,\sigma+1,\tau} - \delta(1 - \beta)u_{fb}, \\
V_{new,\sigma+1,\tau+1} &= V_{old,\sigma+1,\tau+1} - \delta\alpha\beta u_{fb}.
\end{aligned} \] (17)

How to choose the value of \( \delta \) will be discussed in the experimentation section.

3. DRIVER PATTERN LEARNING

In this paper we will demonstrate the proposed FEL scheme for driving pattern learning. For that, a simulink model for vehicle speed control has been used. The model simulates the performance of a car for various given speed profiles (i.e.: city profile, highway profile).

The model has a controller implemented in order to make the vehicle follow the target speed at each point. The feedback action of the controller is divided in both pedal and brake actions, where pedal and brake actions are always positive and within \([0,100]\).

The full speed profile \( V_{target} \) is given for the simulations and at every sample the speed error \( (V_{target} - V_{actual}) \) is known and the controller tends to minimize the error.

3.1 FEL for Driver Pattern recognition:

In this application of the vehicle speed control system, the implementation of the FEL algorithm changes with respect to Eciolaza et al. [2013]. In this case, the FEL diagram is designed so that both Pedal and Brake actions are learned. For that, an unification of positive pedal input and negative brake input is done. The input \( u \) to the car takes both positive and negative values where positive value is realized by pedal and negative by brake, mechanically. If \( u_{ff} \) is positive then activate pedal and if negative then activate brake. Therefore we use pedal \( (u_{fbp}) \) and brake \( (u_{fbb}) \) actions, which are unified as \( u_{fb} = u_{fbp} - u_{fbb} \) theoretically.

Pedal-off is the engine brake and so there is no theoretical border between pedal-on, pedal-off and braking. In such a way, pedal action and brake action are continuous and we have to reflect this fact in FEL for both learning and feedforward controller. Fig. 5 shows the diagram of how FEL has been implemented in the application.

The two inputs to the car, pedal and brake, control the acceleration, positive or negative, of the car:

- Pushing pedal increases the acceleration: ++
• Releasing (weakening) pedal relatively decreases the acceleration:
• Pedal-off is the so-called engine brake to decrease acceleration:
• Brake is to make pedal-off more effective by creating negative input: --

So the vehicle input \( u \) for acceleration is considered as
\[
\begin{align*}
\dot{u} &= Pedal - Brake, \\
\end{align*}
\]
where either pedal or brake is used at each time instant and pedal means positive \( u \) and brake means negative \( u \).

Along this consideration, we should formulate as
\[
\begin{align*}
Pedal &= u_{fpb} + u_{ffb}, \\
Brake &= u_{fbp} + u_{ffb}, \\
\end{align*}
\]
(19)
(20)
Both \( u_{fbp} \) and \( u_{fbp} \) are used for FEL but \( u_{fbp} \) does not contribute in (19), and \( u_{fbp} \) does not contribute in (19). And
\[
\begin{align*}
u_{ffb} &= \begin{cases} 
0 & \text{if } u_{ff} < 0 \\
 0 & \text{if } u_{ff} > 0
\end{cases} \\
u_{ffb} &= \begin{cases} 
-\text{FF} & \text{if } u_{ff} < 0 \\
0 & \text{if } u_{ff} > 0
\end{cases}
\end{align*}
\]
(22)
(23)

4. EXPERIMENTATION

The implementation of the FEL scheme in the application has been done successfully and learned feedforward controller representing the pseudo-inverse of the plant is stable. By FEL, feedback control is replaced by feedforward control. The \( u_{fbp} \) and \( u_{fbp} \) actions are minimized and when learning is complete they tend to zero.

Between the different parameters that must be tuned, the adequate number of vertexes and regions must be defined. In this case these values have been selected depending on different speed profiles available for the simulations.

On the other hand, the update rate \( (K_{upd}) \) of the feedforward controller must be defined. This value should depend on the sample rate of the system and the number of vertexes of the system. In this case, as the sample rate (sr) has been variable in our simulation set-up, we chose \( K_{upd} = 0.01 \times \text{sr} \).

Finally, as the analyzed system has a time-delay, we have used a pure time lag of \( \tau = 1.4 \) seconds for the implementation of the FEL scheme. Take into account that the delayed signal is only used for the learning of the feedforward controller. Correction action \( u_{fb}(t) \) corresponds to input references \( (r(t - \tau), r_{delay}(t - \tau)) \). The feedforward controller will anticipate the time delay \( (\tau) \) of the system.

Note that the origin \( (V_{target} = 0, ACC_{target} = 0) \) vertex has been set to zero.

The performance of the implemented FEL scheme is represented in figures (Fig. 6, Fig. 7, Fig. 8) where the following parameters are shown:

(a) Vehicle Speed where the desired speed \( (V_{target}) \) in blue and the real vehicle speed \( (V_{actual}) \) in red.
(b) Pedal feedback action \( (u_{fbp}) \) in blue and the Brake feedback action \( (u_{fbp}) \) in red.
(c) Feedforward controller action \( (u_{ff}) \) in blue, the total Pedal action \( (Pedal = u_{fbp} + u_{ffb}) \) in green and the total Brake action \( (Brake = u_{fbp} + u_{ffb}) \) in red.

Fig. 6 shows an example of two consecutive acceleration maneuvers of the vehicle with no FEL scheme implementation. When \( V_{actual} \) does not meet the \( V_{target} \), then pedal is pushed to accelerate the car. The car has its inertia and pedal is released when car is accelerating. The error between desired and actual vehicle speeds is not big, and the total pedal and brake actions are determined by the feedback controller.

Fig. 7 shows an example of the same two consecutive accelerations maneuvers of the vehicle working with the FEL scheme. The feedforward model is initialized as a null matrix and it can be observed, while learning, how the total pedal and brake actions eventually are shifting to be determined by the feedforward controller.

Fig. 8 shows the vehicle working with the FEL scheme. In this case the feedforward model is already learned. The total pedal and brake actions are mainly determined by the feedforward controller, and the feedback actions are minimized.

Table I shows the feedforward controller learned in the simulation presented in Fig. 8.

5. CONCLUSION

In this paper we have investigated the application of PB model based FEL for intelligent vehicle systems environment. A method to make an on-line sequential learning of a feedforward controller as a PB model has been used. The FEL scheme controls a system by adapting the pseudo-inverse piecewise bilinear model based on an output of a feedback controller, and...
adjusting operation of an actuator in response to the learned pseudo-inverse model. The lookup table provides a near exact inverse model of the plant under control when output of the feedback controller reaches zero. Consequently, an actuator or plant command output from the lookup table may cause the plant to follow a desired trajectory that was a basis for indexing data in the lookup table.

In particular, the proposed FEL scheme has been used for driving pattern learning, where learning of both the Pedal and Brake actions has been the main goal. The FEL implementation has been done successfully, and once the learning is completed the feedback control is minimized and partially replaced by the feedforward controller.

In future works, the proposed FEL method will potentially be used to address some automotive control applications. On the one hand, the target will be to improve the control performance of traditional feedback loops. On the other hand, the adaptation of the LUTs controlling the car will be made, in order to match the desired vehicle performance and the real vehicle performance.

REFERENCES


