Perfect Tracking for Non-minimum Phase Systems

Michael R. Buehner and Peter M. Young

Abstract—A controller architecture is developed that can provide perfect reference tracking for both minimum phase and non-minimum phase systems with time delays when there are no modeling errors or external disturbances. The perfect tracking property is obtained by factoring the plant into its minimum phase and non-minimum phase components. These components are used to design two feedforward controllers that share information between them. Design constraints are provided that determine both the types of signals that may be tracked perfectly and the feedforward controllers that will provide the perfect tracking. Robust analysis tools are derived that may be used to guide the design process.

A method for adapting the feedforward components in real-time is demonstrated to show the potential performance improvement gained from online system identification.

I. INTRODUCTION

The notion of “perfect tracking” of non-minimum phase systems seeks to answer the questions:

• “What output trajectories can the nominal plant actually take?”
• “What control signals will generate these output trajectories?”

By perfect tracking, we define the types of control signals and output trajectories that the nominal plant will follow when there are no external disturbances or modeling errors.

The idea of perfect tracking control has been explored in the field of robot motion planning [1], [2], [3], an application to a non-minimum phase magnetic levitation system [4], and for improved controller performance in a selective catalytic reduction (SCR) catalytic converter [5]. In robot motion planning and the magnetic levitation system, perfect tracking is achieved by using a multi-rate discrete-time controller. In these scenarios, the controller is running at a faster rate than the D/A that is providing control signals to the plant. When the plant is sampled at the slower D/A rate, the sample points match their desired values. In the SCR converter application, a data preprocessor is used to provide a reference to both a pre-filter and a feedforward controller. The feedback system then corrects for errors between the pre-filter output and the output that results from the feedforward controller. In the nominal case with no external disturbances, the plant output will track the predicted path with no error. A similar architecture was provided in [6] that quantifies the types of signals that may be perfectly tracked for minimum phase systems; however, this method is not able to handle time delays. In the work presented here, perfect tracking is achieved with both continuous-time and discrete-time controllers. These controllers may provide perfect tracking for both minimum phase and non-minimum phase systems with time delays through the use of a particular controller architecture.

The idea of perfect tracking presented here parallels some of the concepts from neuromuscular actuation systems. In humans, the neuromuscular system is the servomechanism portion that includes sensory and motor neurons at the spinal cord level and their associated muscles, joints, and receptors in the periphery [7]. When this system is used to command muscles (e.g., to command motion of a limb), it is referred to as the neuromuscular actuation system (NMAS). This system naturally describes a combination of two feedforward controllers, where the human brain decides upon an action and the NMAS takes the action. Then, a feedback mechanism corrects for errors between the desired and observed paths. As an example, consider the NMAS used to reach for a cup. In this case, the following happens:

1) the human brain calculates the desired path (feedforward prediction),
2) a ballistic response is implemented where the hand extends towards the cup (feedforward control signal), and
3) small corrections to the desired path are made based on the “observed” position of the hand (feedback control on small error signals)

The idea of modeling a neuromuscular actuation system has its roots in [7]. Since that time, the idea has been extended to feedback control systems in various forms [8], [9], [10], [11], [12]. In each of these cases, the feedback controller is used to provide either the feedforward prediction or the ballistic response, and the other piece is addressed through the use of a feedback controller. In these cases, the feedback controller is always being used for tracking reference signals, even in the nominal case. In the work presented here, both the prediction and the ballistic responses are provided by two feedforward paths, and perfect tracking is characterized by the feedback error being zero. In order get perfect tracking, the reference signal is decomposed into two signals, namely a reference signal that the ballistic response can invert and a prediction of the path that the plant output will follow based on the ballistic response. For non-minimum phase plants, this means that the predicted path includes the non-minimum phase dynamics, which separates this method from other methods (c.f., [6]). Also, this architecture is general enough such that it can be applied to a whole class
of minimum phase and non-minimum phase plants.

Another motivation for the architecture presented here is that the feedforward controllers can be nonlinear time-varying and not affect the closed-loop stability analysis. The only restriction is that the feedforward controllers themselves must have bounded energy. A particular instance of this is adaptive online model identification, which will be demonstrated later in this article.

II. METHOD

Perfect tracking of non-minimum phase systems may be achieved by the proposed dual feedforward predictive control (DFFPC) architecture. The systems considered here are plants that are well modeled by proper linear time-invariant (LTI) systems with possible non-minimum phase components such as time delays and right-half plane (RHP) zeros. The transfer function of the plant \( G(s) \) is defined to be

\[
G(s) = \frac{K_{DC}N_{mp}(s)N_{mp}(s)}{D_{s}(s)D_{u}(s)}e^{-s\tau_d},
\]

where \( K_{DC} \) is the DC gain of the system and \( N_{mp}(s) \) and \( N_{mp}(s) \) are the non-minimum phase and minimum phase numerator polynomials, and \( D_{s}(s) \) and \( D_{u}(s) \) are the stable and unstable denominator polynomials. Here, the non-minimum phase numerator polynomial and unstable denominator polynomial have all of their roots in the closed right half plane (i.e., \( s > 0 \)). Similarly, the minimum phase numerator polynomial and stable denominator polynomial have all of their roots in open left half plane (i.e., \( s < 0 \)). Without loss of generality, each of these polynomial are equal to one when \( s = 0 \) (e.g., \( N_{mp}(0) = 1 \)). This may be achieved by choosing the scaling \( K_{DC} \) appropriately.

The transfer function \( G(s) \) may be split into a stably causal invertible part, namely \( G_i(s) \), and a part that does not have a stable causal inverse, namely \( G_{noi}(s) \). This will be referred to as the invertible/noninvertible decomposition of \( G(s) \) and will be expressed as

\[
G(s) = G_{noi}(s)G_i(s),
\]

where

\[
G_{noi}(s) = N_{mp}(s)e^{-s\tau_d}
\]

(3)

\[
G_i(s) = \frac{K_{DC}N_{mp}(s)}{D_s(s)D_u(s)}
\]

(4)

Using eqn (4), the stable causal inverse of \( G_i(s) \) is given by

\[
G_i^{-1}(s) = \frac{D_s(s)D_u(s)}{K_{DC}N_{mp}(s)}.
\]

(5)

It should be noted that \( G_{noi}(s) \) and \( G_i^{-1}(s) \) are generally not proper transfer functions by themselves, and therefore, cannot be physically realized as individual systems. This is addressed by putting design constraints on the types of signals the controller may track perfectly. Specifically, the two feedforward controllers that contain these pieces must be proper.

A. Dual Feedforward Predictive Control

DFFPC handles the prediction, design constraint, and ballistic response entirely in the two feedforward paths. The overall block diagram of DFFPC is given in Figure 1, where the blocks \( G_{noi}(s) \) and \( G_i^{-1}(s) \) are defined in eqns (3) and (5) and \( P_{des}(s) \) is a design parameter.

In order to guarantee the perfect tracking property with this structure, \( P_{des}(s) \) must satisfy the following constraints:

- At steady-state, \( r_{ff}(t) = r(t) \), which means that \( P_{des}(0)G_{noi}(0) = 1 \).
- \( FF1(s) = P_{des}(s)G_{noi}(s) \) and \( FF2(s) = P_{des}(s)G_i^{-1}(s) \) must be stable proper transfer functions.

From the earlier assumption that \( G_{noi}(0) = 1 \), \( P_{des}(0) = 1 \) will satisfy the first constraint. If the plant \( G(s) \) is proper, the second constraint is satisfied if the relative order of \( P_{des}(s) \) is greater than or equal to the relative order of \( G_i(s) \) and \( P_{des}(s) \) is stable. For brevity and to simplify notation, the dependence on \( s \) will be dropped for most transfer functions (e.g., \( G(s) \) may be written as \( G \)).

Let the standard feedback sensitivity transfer function, namely \( S \), and complementary transfer function, namely \( T \), be given by

\[
S = \frac{1}{1 + GK}
\]

(6)

Then, the overall sensitivity transfer function for DFFPC is

\[
S_{DFFPC} = P_{des}G_{noi}S - P_{des}G_i^{-1}GS
= P_{des}G_{noi}S - P_{des}G_{noi}S
= 0.
\]

(7)

In [13], nominal performance is satisfied if \( \|W_1S\|_{\infty} < 1 \). Here, performance is defined by the weight \( W_1(s) \). For DFFPC, nominal performance is perfect and the condition \( \|W_1S_{DFFPC}\|_{\infty} < 1 \) is trivially satisfied for any finite \( W_1(s) \). The feedback error signal is zero and the feedforward controller is providing perfect tracking. This is accomplished by making the feedback controller track the signal \( r_{ff}(t) \), which is a reference signal that the plant can actually follow.

In particular, it contains the non-minimum phase dynamics of the plant that neither feedforward nor feedback systems can stably and causally invert. In relation to a NMAS, \( r_{ff}(t) \) is the feedforward prediction of what the plant will follow, \( u_{ff}(t) \) is the ballistic response, and \( u_{fb}(t) \) is the feedback
signal that corrects for the differences between the predicted output \( r_{ff}(t) \) and observed output \( y(t) \). When the filtered reference input (i.e., the output of \( P_{des} \)) is used for both \( r_{ff}(t) \) and \( u_{ff}(t) \), a similar architecture will provide perfect tracking for either minimum phase systems (i.e., \( G_{noi}(s) = 1 \)) or if the input reference \( (r(t)) \) has a right-half plane zero at the same location of the of the plant \( G \) (c.f., [6]). However, the perfect tracking of non-minimum phase systems in [6] requires an unstable pole-zero cancelation between the reference input and feedforward controller computing the plant inverse, which violates internal stability. Also, the method in [6] cannot provide perfect track for systems with time delays. Both of these limitations are overcome using the controller architecture presented here. The key to this method is to include \( G_{noi} \) in the computation of \( r_{ff}(t) \), which provides perfect tracking for non-minimum phase systems.

The intention of this method is not to remove the non-minimum phase components from the plant. Instead, it uses the two feedforward controllers to provide the feedback controller with a signal that the plant can actually track. To see this, consider the closed loop transfer function

\[
M_{DFFPC} = \frac{P_{des}G_{noi}GK}{1 + GK} + \frac{P_{des}G_{i}^{-1}G}{1 + GK} = P_{des}G_{noi} \frac{1 + GK}{1 + GK} \tag{8}
\]

Here, the non-minimum phase components appear in the closed loop transfer function as \( G_{noi} \). The feedforward design objective is to design \( P_{des} \) to get the desired closed-loop characteristics. This is a different design perspective than the standard feedback design that focuses on closed-loop characteristics by designing for a weighted error sensitivity. However, this is not a replacement for feedback design, since a feedback controller is still required to meet disturbance rejection requirements and to handle model errors in the feedforward controllers. Instead, this is an augmentation to the standard feedback configuration that is designed to improve performance.

1) Robustness Analysis with Additive Uncertainty: For robustness analysis against model uncertainty, an uncertain plant model with additive uncertainty is considered. The model \( G \) is replaced with \( \tilde{G} = G + W_{2}\Delta \), and the perturbed sensitivity function becomes

\[
\tilde{S}_{DFFPC} = \frac{P_{des}G_{noi}}{1 + (G + W_{2}\Delta)K} - \frac{P_{des}G_{i}^{-1}(G + W_{2})}{1 + (G + W_{2}\Delta)K} \tag{9}
\]

Assuming that \( \|\Delta\|_{\infty} \leq 1 \), robust stability requires \( \|\tilde{S}_{DFFPC}\|_{\infty} \leq \|W_{2}\|_{\infty} \). The robust performance condition is

\[
\|W_{2}T\|_{\infty} < 1 \quad \text{and} \quad \left\| W_{1}\frac{P_{des}G_{i}^{-1}(G + W_{2})S}{1 + W_{2}S} \right\|_{\infty} < 1, \quad \forall \Delta \tag{11}
\]

Using methods similar to that in [13], the following theorem may be proven.

**Theorem 1:** A necessary and sufficient condition for robust performance is

\[
\|W_{1}P_{des}G_{i}^{-1}W_{2}S\|_{\infty} < 1 \quad \text{and} \quad \|W_{2}KS\|_{\infty} < 1 \tag{12}
\]

**Proof:** The proof is omitted for brevity.

2) Robustness Analysis with Multiplicative Uncertainty: For robustness analysis against model uncertainty, an uncertain plant model with multiplicative uncertainty is considered. The model \( G \) is replaced with \( \tilde{G} = (1 + W_{2}\Delta)G \), and the perturbed sensitivity function becomes

\[
\tilde{S}_{DFFPC} = \frac{P_{des}G_{noi} - P_{des}G_{noi}\Delta W_{2}}{1 + (1 + \Delta W_{2})GK} = \frac{1}{1 + G \Delta W_{2}} \tag{13}
\]

Assuming that \( \|\Delta\|_{\infty} \leq 1 \), robust stability requires \( \|\tilde{S}_{DFFPC}\|_{\infty} \leq \|W_{2}\|_{\infty} \). The robust performance condition is

\[
\|W_{2}T\|_{\infty} < 1 \quad \text{and} \quad \left\| W_{1}\frac{P_{des}G_{noi}\Delta W_{2}S}{1 + W_{2}S} \right\|_{\infty} < 1, \quad \forall \Delta \tag{14}
\]

By a similar method used for the additive uncertainty case, a necessary and sufficient condition for robust performance is

\[
\|W_{1}P_{des}G_{noi}W_{2}S\|_{\infty} + \|W_{2}T\|_{\infty} < 1 \tag{15}
\]

III. ILLUSTRATIVE EXAMPLES

In this section, some example problems are used to demonstrate the architecture presented here. In the first example, perfect tracking of a non-minimum phase plant is demonstrated where all of the systems are implemented as continuous-time systems. Then, an example of a discrete-time implementation that uses adaptation is presented.

A. NMP Perfect Tracking

Perfect tracking of a nominal non-minimum phase plant with a RHP zero is presented here. To begin, define the nominal non-minimum phase plant as
\[ G(s) = \frac{-s + 2}{(s + 5)(s + 10)} = \frac{-s + 1}{(s + 5)(s + 10) + 1}. \] (15)

For this plant, \( K_{DC} = 3 \), \( N_{mp}(s) = \frac{s}{s + 1} \), \( N_{mp}(s) = 1 \), \( D_L(s) = (\frac{s}{5} + 1)(\frac{s}{10} + 1) \), and \( D_u(s) = 1 \). The two feedforward models are defined to be

\[ G_{noi}(s) = \frac{-s}{2} + 1, \quad G_{i}^{-1}(s) = \frac{1}{3} \frac{s}{s(\frac{s}{5} + 1)(\frac{s}{10} + 1)}. \] (16)

In order to guarantee that the two feedforward controllers are proper, \( P_{des}(s) \) must have a relative order of two or more. For the example here, this may be accomplished by choosing

\[ P_{des}(s) = \frac{1}{(\alpha_{des} + 1)^2}, \] (17)

which satisfies the design constraint \( P_{des}(0) = 1 \). With this choice of \( P_{des}(s) \), \( FF2(s) \) is given by

\[ FF2(s) = P_{des}(s)G^{-1}_{i}(s) = \frac{1}{3} \frac{s}{(\frac{s}{5} + 1)(\frac{s}{10} + 1)} \] (18)

and \( FF1(s) \) (which is also the closed-loop transfer function) is given by

\[ FF1(s) = P_{des}(s)G_{noi}(s) = \frac{-s}{2} + 1 \] (19)

The right-half plane zero in this transfer function means that the step response will have an undershoot that is related to bandwidth of \( P_{des}(s) \) (which is \( \alpha_{des} \) rad/sec). In particular, for a faster response, a larger bandwidth is required that will result in more undershoot. This phenomenon is described in [14].

Simulation results for various \( \alpha_{des} \) are shown in Figure 2. In the top graph, the filtered reference \( r(t) \) and actual plant output \( y(t) \) are plotted for various \( \alpha_{des} \). For these cases, the plant model is perfect (nominal case) and \( y(t) = r_{ff}(t) \). Therefore, these two signals are on top of each other. This is echoed again in the bottom graph where the feedback error is zero for each case.

These simulations illustrate the design trade-offs for a system with a RHP zero. For a faster response (i.e., more bandwidth or a larger \( \alpha_{des} \)) there will be more undershoot and it will take more control authority. For the case with \( \alpha_{des} = 10 \) rad/sec, the control signal spikes up quickly and then settles in to the final value. Depending on the actuator being used, this may be undesirable. In the opposite extreme, if the bandwidth is too small (e.g., \( \alpha_{des} = 1 \) rad/sec in Figure 2), the control signal will not increase as fast, but the response will be slower. These design trade-offs are shown in Figure 2. For a good trade-off between control authority and fast response time, the design parameter \( \alpha_{des} = 5 \) rad/sec may be chosen. This design methodology may be used to design the controller for the specific plant. In general, the bandwidth \( \alpha_{des} \) should be within the bandwidth of the actuator in order to maintain the perfect tracking property.

As an example design process, the following steps may be followed:

1) Design the feedback controller \( K(s) \) for the desired robustness to model uncertainties and for disturbance rejection.
2) Pick a structure for \( P_{des}(s) \) with tunable parameters.
3) Search over the tunable parameters of \( P_{des}(s) \) to satisfy your performance requirements (e.g., satisfy the conditions in either eqn (11) or eqn (14)).

For the previous example, \( K(s) \) was chosen to be a stabilizing controller. Given a specific \( K(s) \), the design of \( P_{des}(s) \) would require a search over \( \alpha_{des} \). Another method for designing \( P_{des}(s) \) is to use robust and optimal design methodologies. We have had some initial success with this, which is addressed later in the section on Future Directions.

B. Adaptive Feedforward Controller

The presented controller architecture may also be implemented in discrete-time. In this case, a zero-order hold equivalent discrete-time representation of the plant is used. In this setting, minimum phase zeros and stable poles lie within the open unit circle in the complex plane (i.e., \( |z| < 1 \)) and non-minimum phase zeros lie outside the closed unit circle in the complex plane (i.e., \( |z| \geq 1 \)).

For this example, a discrete-time controller is used to control a continuous-time plant. The same perfect tracking property is achieved in this case. Also, when a perturbed plant model is used, the feedforward controller is able to adapt to the plant using an online system identification routine.

For this example, a first order minimum phase system with a one second time delay is considered. This is given by

\[ G(s) = \frac{2}{10} e^{-s}. \] (20)

Here, \( G_1(s) = \frac{2}{10} e^{-s} \) and the non-minimum phase (and non-invertible) part is given by \( G_{noi}(s) = e^{-s} \). If the discrete-time controller runs at rate \( T_s = 0.01 \) seconds, the zero-order hold equivalent of the plant is given by
\[ G_{zoh}(z) = \frac{0.1903}{z - 0.9048}z^{-100}. \quad (21) \]

Now, \( G_i^{-1}(z) = \frac{z - 0.9048}{0.1903} \) and \( G_{nol}(z) = z^{-100} \). For a second order discrete-time filter that has a bandwidth of \( \omega_{\text{des}} \) radians/sec, the following discrete-time transfer function may be used

\[ P_{\text{des}}(z) = \left( \frac{1 - e^{-\alpha \omega T_s}}{z - e^{-\alpha \omega T_s}} \right)^2. \quad (22) \]

At DC (i.e., \( \omega = 0 \) or \( z = e^{j\omega T_s} = 1 \)), \( P_{\text{des}}(1) = 1 \), which satisfies the equivalent design constraint for a discrete-time system. In this case, the \( FF2(z) \) is given by

\[ FF2(z) = P_{\text{des}}(z)G_i^{-1}(z) = \frac{(1 - e^{-\alpha \omega T_s})^2}{z - \frac{0.1903}{z - 0.9048}} \cdot 0.1903, \quad (23) \]

and \( FF1(z) \) (which is also the closed-loop transfer function) is given by

\[ FF1(z) = P_{\text{des}}(z)G_{nol}(z) = \left( \frac{1 - e^{-\alpha \omega T_s}}{z - e^{-\alpha \omega T_s}} \right)^2 z^{-100}. \quad (24) \]

A feedback controller was designed using the \( \mu \)-synthesis routine described in [15]. An additive uncertainty weight of \( W_2(s) = 0.1 \) and equivalent performance weight of \( W_1(s) = \frac{s^2}{s^2 + 0.001} \) were used to create a continuous-time \( K(s) \) that was converted to a discrete-time controller using a bilinear \( Z \)-transform. This resulted in

\[ K(z) = \frac{0.077(z + 1)(z - 0.987)(z - 0.98)(z - 0.9048)}{(z - 0.99)(z - 1)(z^2 - 1.724z + 0.77)}. \quad (25) \]

The exact details of the feedback controller design are omitted for brevity. Using the weights \( W_1 \) and \( W_2 \) provided above and the final controller \( K \), \( \omega_{\text{des}} \) was chosen to satisfy the robust performance condition given in eqn (11). For this example, \( \omega_{\text{des}} = 7 \) rad/sec satisfied the robust performance condition. Simulation results with \( \omega_{\text{des}} = 7 \) rad/sec are shown in Figure 3, which shows the perfect tracking (i.e., zero error between the filtered reference input and actual output).

When there are modeling errors, \( e(t) \neq 0 \) and the feedback controller will use \( u_{fb}(t) \) to correct the total control signal \( u(t) \). The feedback controller acts on the non-minimum phase plant, which may lead to a serious degradation in performance when the modeling errors are large. To see this, consider a mismatch in the time delay where the actual plant time delay is 0.05 seconds longer than the original model. The resulting controllers performance is shown Figure 4.

The output is very oscillatory even though the feedback controller is trying to correct on a small error signal. This results from the fact that the plant has a 1.05 second time delay, whereas \( FF1(z) \) is modeling a 1 second time delay. This oscillatory behavior may be reduced (and in some cases eliminated) by adapting the two feedforward controllers. These feedforward controllers were designed based on the discrete-time transfer function given in (21). If the finite history of the plant input (i.e., \( u[n] \)), which is the discretized version of \( u(t) \) and plant output (i.e., \( y[n] \)), which is the discretized version of \( y(t) \) time series are stored in memory, a system identification routine may be used to determine the new transfer function (e.g., \( G_{zoh}^{\text{new}}(z) \)). This new transfer function may be split into its invertible and non-invertible parts, which may be used to update the two feedforward controllers. For the particular case here, the system identification tools were used to identify the new system \( G_{zoh}^{\text{new}}(z) \) as

\[ G_{zoh}^{\text{new}}(z) = \frac{0.1903}{z - 0.9048}z^{-105}. \quad (26) \]

This new plant is used to update the feedforward controllers in Figure 3. The result after the feedforward controllers have been updated is shown in Figure 5.

In Figure 5, the perfect tracking property is again achieved for this particular system. This type of adaptation is particularly useful for systems that may develop a non-minimum phase component as they change, since it may be captured in the \( FF1 \) path. Also, the feedforward paths do not need to be fixed order controllers, which means that they can easily change order to best capture the dynamics of the plant.

The advantage to this architecture is that the adaptation is not taking place inside the feedback loop, which means internal stability is guaranteed while the controllers adapt
in real-time, provided the two feedforward controllers stay stable. For this architecture and adaptation scheme, this is guaranteed for all plants since the non-minimum phase components of the plant are always stably accounted for in the FF1 controller and the causal and stably invertible parts of the plant appear in FF2.

This example demonstrates that an accurate model of the system is required for good performance and that this model may either come from a priori knowledge or through adaptation. In ongoing research, other adaptation methods will be explored.

IV. Future Directions

In the feedforward controllers, $P_{\text{des}}(s)$ is the only design parameter that is not directly tied to the plant model. In the work presented here, $P_{\text{des}}(s)$ was parameterized and a search over the free parameters was performed. However, $P_{\text{des}}(s)$ does not need to be designed this way. In our ongoing research, we have looked at using robust and optimal control design methodologies to synthesize $FF2(s)$. These methods build off of the techniques provided in [16]. From experience with the tools used, it is possible to get the robust and optimal designs to be of the form $FF2(s) = P_{\text{des}}(s)G_i(s)$. This provides a method for synthesizing $P_{\text{des}}(s)$ directly. The details of this will be presented in future publications.

The idea of using two feedforward controllers to achieve perfect tracking for non-minimum phase systems may also be applied to a specific Smith predictor. In this case, a part of the prediction is done inside the feedback loop instead of completely in the $FF1(s)$ controller. The details of this method will be presented in future publications.

While the discussion here was restricted to SISO plants, the methods do generalize to MIMO systems, which will be presented in future publications.

An adaptation scheme was demonstrated that used a standard system identification routine to update the models based on stored historical data. In many applications, this may require too much overhead. Instead, an online optimization routine such as reinforcement learning [17] may be more appropriate. Various methods will be studied and discussed in future publications.

V. Conclusions

A methodology for achieving perfect tracking on non-minimum phase SISO plants was presented based on using two feedforward controllers that share information between them. Robustness tools were developed that may be used to guide the design process. The performance of these controllers depends upon the accuracy of the models used. In order to achieve good performance, accurate plant models are required. These models must either be known a priori or learned via adaptation. For the latter method, an example based on system identification was provided. Adaptive model learning is a topic of ongoing research.

VI. Acknowledgments

This work was partially supported via a grant from Solix Biofuels.

REFERENCES


