ANALYTICAL STATISTICAL STUDY OF
LINEAR PARALLEL FEEDFORWARD COMPENSATORS
FOR NONMINIMUM-PHASE SYSTEMS

Keyvan Noury, Bingen Yang
Department of Aerospace &
Mechanical Engineering
University of Southern California
Los Angeles, CA 90089, USA
Email: noury@usc.edu, bingen@usc.edu

ABSTRACT
In this work, a new parallel feedforward compensator for the feedback loop of a linear nonminimum-phase system is introduced. Then, analytical statistical arguments between the existing developed methods and the innovated method are brought. The compelling arguments suggest the parallel feedforward compensation with derivative (PFCD) method is a strong method even though at its first survey it seems to be optimistic and not pragmatic. While most of the existing methods offer an optimal integral of squared errors (ISE) for the closed-loop response of the nominal plant, the PFCD offers a finite ISE; in reality, typically, the nominal plant is not of main concern in the controller design and the performance in the presence of mismatch model, noise, and disturbance has priority. In this work, there are several arguments brought to bold the importance of the innovated PFCD design. Also, there is a closed-loop design example to show the PFCD effectiveness in action.

Keywords: nonminimum-phase (NMP); delay; stability; infinity norm; right-half plane (RHP) zero; RHP-even; RHP-odd; left-half plane (LHP) zero; parallel feedforward compensator; Nyquist diagram.

INTRODUCTION
Linear time-invariant systems with RHP-zeros are rampant and can be found everywhere such as in robotics [1], airplanes [2], and etc; consequently, they have attracted a lot of attention [3]. The phase of such systems is not minimum compared to the systems with the same magnitude and only LHP-zeros. The nonminimum-phase (NMP) systems naturally can happen when actuators and sensors of the control system are placed in different locations, and consequently, the RHP-zeros or delays can be introduced to the system because of the wave propagation and the difference between the fast and slow responses in the frequency domain [4-7]. One of the earliest NMP control implementation is the Smith’s predictor (SP) [7]. Since then, researchers have used the base idea in SP to cope with different types of NMP systems. Some important examples of these works are internal model control (IMC) and generalized Smith predictor (GSP) methodologies [8-11]. However, the introduced compensators are of orders as high as the plant’s order. Thus, there are several concerns as follows:

• An inaccurate system model, can degrade the performance significantly.
• In the presence of noise and disturbance, the developed nominal dynamic model can have limited efficiency.
• The processes that these models can cope with are somewhat restricted. For example, a system with multiple delays becomes very complicated for these methods to handle.
• Even with an exquisite dynamic modeling, the hardware can confine the “implementation”. For instance, the best way to implement a time delay is by Padé approximants. Depending on the application, the orders of polynomials in the approximation can be very high, and therefore, the controller becomes susceptible to noise.

Besides these problems, these methods are usually not generic and can provide solutions only as in a case by case study.

In this paper, a new and simpler compensator for the control of NMP systems is being sought and introduced. This method makes use of parallel feedforward compensation with derivative effort, and therefore the “PFCD” acronym is used to refer to it. The mathematical and statistical intuitions behind this method are discussed in the coming sections.

In here, a brief review of literature on linear NMP control is given. The IMC control methodology, as is shown in
FIGURE 1(a), that was first introduced by Morari et al. [8], is one of the considerable methods that copes with NMP systems in a similar fashion as in SP. However, the mismatch model and noise degrade the performance. Even though, in the next surveys some adjustments such as use of a filter in the feedback loop is considered in [9] and [10], the compensator transfer function polynomials orders can be very high. In a similar way, the GSP configuration [11], see FIGURE 1(b), has problems; however, it has a different parameterization of the system for its implementation. In the GSP method, exact pole-zero cancellations are needed, which is only possible in theorem, and in action, barley can be satisfied. Therefore, the GSP and IMC models need certain attention while implementing them. Other works, such as in [12], have similar approach, even though, there it is tried to avoid controller saturation. Hence, the mentioned problems of mismatch model, noise, and disturbance yet exist.

![Diagram of IMC Control System](image)

**FIGURE 1:** (a) In the IMC configuration a model of the process in the parallel branch is subtracted from the process and the result is fed back as error. (b) The GSP configuration; omits the NMP part of the process ($G_{p+}$) and controls the MP part ($G_{p-}$) of the process, while the NMP part is left for the output of the closed-loop system.

This paper is presented as follows. First, two mathematical lemmas are presented that will be the background theme for this paper for stable polynomials. Then, the relation between the lemmas and PFCD control methodology is brought. Thereafter, the developed PFCD method with existing methods from a statistical point of view are compared and justified. Then, for a rather complicated transfer function, that is even very hard for the existing methods to model their parallel compensator accordingly, it has been shown how conveniently the PFCD parameter is tuned to account for the closed-loop stability and performance. The simulation results in this section are brought in the corresponding figures. The PFCD computational cost and the characteristics of the closed-loop system are discussed throughout the paper. In the end, concluding remarks are followed to support the PFCD methodology for NMP systems.

**MATHEMATICAL BACKGROUND**

In this section, it is shown that we can make the sum of a stable and unstable polynomial be stable. In the control methodology of the PFCD, this fact will be utilized.

**Stable polynomial generation**

It is not difficult to see the summation of one stable and one unstable polynomial can be unstable, e.g. see [13]. However, in here, we represent two simple polynomial summations that will result in stable polynomials.

**Lemma 1:** Assume the roots of the following polynomial

$$P(s) = kA(s) + B(s)$$

where $B(s)$ can be a quasipolynomial (see [13] for definition) or normal polynomial and $A(s)$ is a normal polynomial in which the order of $A(s)$ is at least equal to the order of $B(s)$. Then, by making $k$ large enough, $P(s)$ can be stabilized.

To show the intuition for the lemma first consider normal polynomials

$$A(s) = \prod_{i=1}^{n} (s - p_i)$$

$$B(s) = \prod_{j=1}^{m} (s - z_j)$$

where $\forall p_i \in \mathbb{R}^{-}$, $\forall z_j \in \mathbb{R}$ for $i = 1, ..., n, j = 1, ..., m$; and $s$ is a complex parameter; and $b_m > 0$. Intuitively, in Eq. (1) if the coefficient, i.e. $k$, of the stable polynomial is increased, the polynomial terms of $A(s)$ will dominate the polynomial terms of $B(s)$. Then, the sum can be written as

$$ks^n + k\left(a_{n-1} + \frac{b_{n-1}}{k}\right)s^{n-1} + \cdots + k\left(a_0 + \frac{b_0}{k}\right)$$

For $j > m$, $b_j$ coefficients can be considered to be zero. As it can be seem, as $k$ increases, the terms in the sum will be close to the terms of $A(s)$, and thus, the roots should be near to the roots of $A(s)$. A root locus sketch of the phenomena is brought in FIGURE 2. As the coefficient $k$ is increased, the roots of Eq. (1) will move from the roots of $B(s)$ to the proximity of the roots of polynomial $A(s)$. Thus, for a large value of $k$, the roots will be in the left half plane (LHP) of the root locus $s$-plane. It can be proven that for

$$k \geq \frac{|B(j\omega)|}{|A(j\omega)|}_\infty$$

(5)
the polynomial in Eq. (1) will be stable. In here, the right hand side is the infinity norm of the transfer function \( B(s)/A(s) \). In the rest, we will use \( H^{B/A}_\infty \) to represent this norm.

Through the use Lemma 1, we can transfer the RHP-zeros of a process to LHP and have a stable feedback system. Nevertheless, since the feedback is meant to improve the performance and have a zero steady state error, we introduce a second lemma that will help for this purpose.

**Lemma 2:** Assume the roots of the following polynomial

\[
P(s) = ksA(s) + B(s)
\]

where \( A(s) \) and \( B(s) \) were defined in Lemma 1 with extra requirement that \( B(s) \) has even number of RHP-zeros. Then, there is a \( k_{cr} \in \mathbb{R} \) such that for \( \forall k \geq k_{cr} \), \( P(s) \) in Eq. (6) will be stable.

Again, the goal is to increase the coefficient of the stable polynomial so that the overall polynomial becomes stable; however, note that since we want a steady state error of zero there is a subtle difference between Eqs. (1) and (6). Even though, in Eq. (4) the polynomials are nominal, still the intuition is the same when either of the polynomials is quasipolynomial since the exponential term acts like a sign multiplier when \( s \) is on the imaginary axis; thus, the lemma can be proven similarly. In Lemma 2, the bound for \( k \) that guarantees that the root locus branches are all on the LHP side is

\[
k \geq \sum_{i=0}^{M} t_i H^{B_i/A}_\infty + \sum_{i=0}^{M} t_i d_i B_i H^{A_i/A}_\infty = k_{cr}
\]

where it is written in the presence of delays in \( B(s) \)

\[
B(s) = \sum_{i=0}^{M} B_i(s)e^{-d_i s}
\]

in which \( t_i \) are the delays and \( B_i \) are normal polynomials. Also in Eq. (7), \( B_i' \) and \( A_i' \) are the polynomial derivatives.

This upper-bound is denoted as \( k_{cr} \). This \( k \) bound is a “loose” upper-bound that the Eq. (6) becomes a stable polynomial, see FIGURE 2 to see how the closed-loop roots move the LHP.

In this paper we define a transfer function “RHP-even” if it has an even number of RHP-zeros. A similar definition is for “RHP-odd” transfer functions. It is easy to determine if a process is RHP-even or RHP-odd since every quasipolynomial can only have finite number of real RHP roots (the complex roots come in pairs and will not matter) and the sign of the polynomial at infinity and 0 can help to determine the parity without requiring knowing all the roots. As the reader will see in the PFCD methodology, using Lemma 2, only a few information about the process is need to be revealed for a closed-loop system control. The infinity norms in Eq. (6) and the parity of process RHP-zeros are needed to make a stable closed-loop with finite ISE (the dc-gain of 1). Compared to this few information, the other methodologies require all the poles and zeros of the process transfer functions. Thus, they are inherently not pragmatic to be implemented.

A simplified case can be considered where the polynomials are nominal polynomials, see [14] for variations in the proof.

**PFCD control methodology using Lemma 2**

From the previous lemmas, some relations between the summation of two polynomials and stabilization of a NMP system can be depicted. Since there are already some stabilization techniques for stabilizing the unstable NMP plants, e.g. see [15], in here we focus only on the stabilization of the closed-loop system for stable processes. These systems, as it is shown in the example of the next section, can have zeros at any location of the root-locus plane. Therefore, methods such as the IMC or GSP, in which pole-zero cancellations are required can fail, especially near to the imaginary axis, e.g. see [16]. The PFCD method is so that it converts the NMP open-loop (OL) system into a minimum-phase (MP) by moving the RHP-zeros into the LHP. The time response, root-locus and Nyquist diagrams are utilized to interrogate the results. Nevertheless, in the example of the next section, the root-locus plot cannot be directly used since the system has delays and therefore is of infinite dimensions and branches.

By considering the transfer function of the processes to be a numerator over a denominator polynomial as follows

\[
G_0(s) = \frac{B(s)}{A(s)}
\]

then, one can find a simple mapping between stabilizing the open-loop system in FIGURE 3 and Lemma 2 of the previous section. Since the Lemma 2 requires a \( B(s) \) that has only even number of real RHP-zeros, equivalently in the CL system of FIGURE 3, we need a RHP-even transfer function in the feedforward path; however, for a RHP-odd process, we can use an all-pass filter, i.e. \( G_f(s) = (s + z)/(s + p) \) such that \( p = -z > 0 \). Also, to have a routine root locus a sign multiplier is added to omit the negative sign that the transfer function may
have. The parallel branch of the process has a compensator of a
derivative. Using Lemma 2’s principal, when the weight in the
branch (that is parallel to the process) is increased, the closed-
loop characteristic equation will be a summation of a weighted
stable denominator polynomial with numerator polynomial that
generates a stable characteristic polynomial in the OL system.
Also, the CL that includes the parallel compensator will
become stable with a chosen coefficient in Eq. (7). The
controller can be a simple integrator, i.e. \( C(s) = 1/s \), for a
zero steady state error.

**Tuning the weight/gain in PFCD control**

Lemma 2 claims that there should be a finite bound for the
weight in Eq. (6) to make an overall stable polynomial.
Correspondingly, the CL stability can be assured in the PFCD,
however, the main reasoning of the feedback system is to have
a better performance; therefore, we need to search over the
coefficient range until the desired performance is achieved.
Typically, this search can be done in \([0, K_{cr}]\). In the next
section, for different \( k \)'s as in FIGURE 3, through an example
we have shown how this coefficient can affect the CL time
response and Nyquist diagram.

**FIGURE 3:** PFCD configuration. In the parallel branch we can tune
the coefficient until the desired performance is achieved. In the case of
a RHP-odd process, in the feed-forward path we should have an all-
pass filter. The derivative in the parallel branch helps to have a zero
steady state error.

**ANALYTICAL STATISTICAL PROBLEMS**

**Mismatch model**

In most of the systems, the model to represent a process
can lack certain aspects of the process. For example, for a beam
model, depending on the thicknesses and length, an Euler-
Bernoulli model can be a poor model while Timoshenko beam
can give better results as a model. Also the method to represent
a model, can affect the mismatch. For example, in the
governing equation obtained by the extended Hamilton’s
principal, 3 terms from assumed modes method (AMM) will
have a different result than 20 terms. This is especially true for
the governing equations that have transcendental solutions. So,
the truncation in the model can cause a mismatch between
the process and the model. The mismatch itself can cause problems
like the well-known phenomena of spillover [4] that will ignore
the higher frequency dynamics. Also, the mismatch can happen
when the boundary conditions or the initial state of the system
are not properly estimated. For example, for a clamped-pinned
beam, there may not be an exact estimate on how much the
clamped end is really a pure clamped. This, in part, depends on
the friction of the support. Also, for the pinned end, there are
some levels of freedom that may affect the true identity of the
support.

The mismatch can also happen for the disturbance and
noise in the system. For example a wind gust can be modeled
differently under different assumptions. Some engineers may
assume an ideal gas for the air surrounding the system, and
some engineers may use more accurate air equations. These
will all result into different models, that no one can claim an
exact model of the disturbance.

Another type of mismatch can be considered while trying
to estimate a delay with Padé approximants. Even with high
polynomial order estimates, the infinite nature of the delay is
not preserved.

**Disturbance and noise**

While the disturbance and noise mismatch is already
pointed out, there are more subtleties to this when we want to
estimate the system model in the presence of these system
associates. Note that in machine learning and statistics, the data
is considered associated with noise. Therefore, it is not peculiar
if we take a detour to make links between the control field and
statistics.

The researchers obtain data in the presence of noise and try
to fit their own model to make predictions for the future coming
data. However, it is well known that testing is absolutely
necessary to evaluate the model. While the noise and
disturbance cannot be predicted, the model derived for the
known training set is evaluated on the test set to estimate
the accuracy of the model. In here, we represent two subtleties that
may arise from high order target function estimation while
trying to model it with a low order and high order functions.

The first subtlety is in relation with the stochastic noise and
the second subtlety is in relation with deterministic noise. This
discussion is inspired from the overfitting chapter in [17].

i) **Stochastic noise:**

In here, through an example we explain why a model of
less complexity can perform better than a model of high
complexity in the presence of the noise to estimate a target
function. Assume it is desired to estimate a target function
polynomial of order 10. There are 15 data points drawn at
random in the presence of noise and we fit the two polynomial
models, see FIGURE 4(a). Then, we compute the errors of each
model to represent the 15 data points as in the \( E_{in \ sample} \) row.
The results of the two models are brought in the TABLE 1.

<table>
<thead>
<tr>
<th>TABLE 1: 10th order noisy target</th>
<th>2nd order</th>
<th>10th order</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_{in \ sample} )</td>
<td>0.8</td>
<td>0.05</td>
</tr>
<tr>
<td>( E_{out \ of \ sample} )</td>
<td>2</td>
<td>15</td>
</tr>
</tbody>
</table>
Since the target function is not known in advance (that the target function is a 10th order polynomial is not known), we try to fit the 10th order and 2nd order polynomials to the gathered 15 data points. To fully be able to compare the two models we need to sample more data points to find the error that each model (the 2nd and 10th order polynomials) has. Thus, we can draw more data points (20 more) to compare the errors as in the $E_{\text{out of sample}}$ row. As it can be seen from TABLE 1, even though the target function is of order 10, the polynomial estimation of order 10 performs poorly in out of sample case.

ii) Deterministic noise:

The overfitting can arise from another source and that is deterministic noise. In here, again through an example we explain why a model of less complexity can perform better than a model of high complexity to estimate a target function of high complexity even though there is no noise present in the system.

Let us assume we want to estimate a target function polynomial of order 40. Hence, 15 data points are drawn at random and we fit the two polynomial models, see FIGURE 4(b). Then, we compute the errors of each model to represent the 15 data points as in the $E_{\text{in sample}}$ row. The results of the two models are brought in the TABLE 2.

| TABLE 2: 40th order noiseless target |
|-----------------|-------|-------|
|                 | 2nd order | 10th order |
| $E_{\text{in sample}}$ | 0.95   | 0.005 |
| $E_{\text{out of sample}}$ | 3     | 85    |

Again, we try to fit the 2nd and 10th order polynomials to the gathered 15 data points. To fully be able to compare the two models we need to sample more data points to find the error that each model (the 2nd and 10th order polynomials) has. Thus again, we can draw more data points (20 more) to compare the errors as in the $E_{\text{out of sample}}$ row. As it can be seen from TABLE 2, now that the target function is of order 40, the polynomial of order 10 performs poorly in a more real scenario even though its complexity was expected to help the 10th order polynomial to model the complex target function.

Consequently, as we can see, in the presence of limited data drawn from the system, a simpler model can cope with both the stochastic and deterministic noises. This is in accordance with Ockham’s Razor principle in machine learning [17].

This result is important and can be related to the models that try to represent the processes in the control system. Trying to make a correlation between the results in this section and in control systems, we can think of the processes as the target functions and the models as polynomial ratios that are fitted to the poles and zeros of the process. These poles and zeros are drawn usually in the lower frequency range from the developed dynamic models without of paying much attention to the higher order dynamics of the system. Consequently, as it was shown here, unexpectedly, for non-nominal processes, a higher order model can drastically have a poor performance while a lower order model can do just fine even if the target process has complicated dynamics.

FIGURE 4: Fits using polynomials of order 2 (dashed line) and 10 (solid line) to 15 data points drawn from an unknown target function. (a) In the presence of noise; and, the target function being a polynomial of order 10. (b) No noise, but the target function is of order 40.

COMPARISON OF NMP METHODS

Complicated parallel feedforward compensators (CPFCs)

The existing solutions to control the NMP processes usually involve high order models in the parallel compensation branch. The IMC and GSP methodologies both require a good model of the NMP part of the process to be deducted in the OL system while dealing with the consequences when spitting it out of the CL system.

In contrast to the complicated models that need to be developed rigorously based on all the information that we can gather from the dynamics of the problem to estimate the poles, zeros, and delays, in the PFCD control methodology we only require to know certain norms and the parity of the process RHP-zeros. This means the PFCD needs a lot less data to be revealed to the control system compared to the other control methodologies. In the situations that require a rapid decision based on limited available information this is an advantage. Such scenarios can arise in the emergency conditions, e.g. combat zones, or lack of measuring instruments, e.g. in acidic conditions. On the other hand, the model adjusted based on the less data can be very robust when unpredicted situations happen to the dynamics of the not-fully-modeled process.

Problems of the noise in CPFCs

The results of the previous section can be expanded to investigate the effectiveness of the methods to control NMP systems. As mentioned, the control methods like IMC and GSP require high order models of the process in the parallel branch to be deducted from the feedforward path and fed back to the system. Here, the problem becomes very similar to the
deterministic noise, that even in the case that there is no stochastic noise present in the system, the models will perform poorly to estimate the dynamic frequency response in the whole frequency region.

![Diagram](Image)

**FIGURE 5:** For different control methodologies, we can replace a parallel compensator as for X. The PFCD methodology places a simpler block as for X while the X in IMC and GSP methods are very complicated and can have statistical problems.

Consequently, the property that most of the NMP systems have about the integral of the squared error (ISE) to be optimal is not a good measure to evaluate the system response in a real scenario. However, the PFCD can claim the error that the system has in the nominal case will be similar to the case that the noise and disturbance are introduced to the system, hence; the system is robust. Thus, by using PFCD, with a simpler block X in FIGURE 5 a more robust system has been obtained.

**SIMULATION AND RESULTS**

In the rest, the PFCD implementation to control the response of a relatively complicated transfer function is considered.

It should be pointed out that this particular transfer function does not correspond to a real world example, but is a good example to see the PFCD methodology can handle even very complicated transfer functions when required. A later study has considered practical transfer functions to be controlled by PFCD method. However, in here we consider a more complicated transfer function. Let

\[ G_0(s) = \frac{B_0(s) - B_1(s)e^{-0.5s} + B_2(s)e^{-4s}}{s^4 + 19s^3 + 21s^2 + 15s + 6} \]  

(9)

where

\[ B_0(s) = s^3 - 44.7s^2 + 623.2s - 2689.6 \]
\[ B_1(s) = s^2 + 9s - 9 \]
\[ B_2(s) = 3s^3 - 10s^2 - 40s + 3 \]

Note that in here, \( B_i(s) \), for \( i = 0,1,2 \), are all unstable polynomials, and besides that, for \( B_1(s) \) and \( B_2(s) \) terms, a delay term exists. For \( G_0 \), there are only 3 real RH0 zeros, \( z_1 = 8.626 \), \( z_2 = 14.429 \), and \( z_3 = 21.665 \). Therefore, the plant is RHP-odd; and it is required to add an extra real RHP-zero to convert the feed-forward path into a RHP-even type. The added RHP-zero is better to be chosen at \( z = 200 \) since it is far from the rest of the zeros and will not affect the time response. Therefore, \( C_f(s) = (s - 200)/(s + 200) \) and \( \text{sgn} = \text{sgn}(1) = 1. \) After applying this methodology, \( G'(s) \) in the feed-forward path will be

\[ G'(s) = G_0(s) \frac{s - 200}{s + 200} \]

To determine \( k_{cr} \) in Eq. (7), we need to be careful to modify the \( B_i(s) \) terms in there, since now they are each multiplied by \( (s - 200) \). \( A(s) \) is also multiplied by \( (s + 200) \). As we see, the transfer function of \( G'(s) \) in the feed-forward path now has become RHP-even which makes the root locus to not cross the imaginary axis through the origin.

Simulations for the configuration in FIGURE 3 are performed. From Eq. (7), \( k_{cr} \) is calculated to be 3081.8. However, this value is a loose upper-bound on \( k \) in the parallel branch that makes the characteristic equation and the closed-loop system stable. An integrator is used as for the controller block \( C(s) \) to make the steady state error zero.

![Diagram](Image)

**FIGURE 6:** \( G_M(s) \)'s Nyquist diagrams for different values of \( k \) in the PFCD design. The -1 point (shown with a red cross) comes out of encirclement as \( k \) increases. (a) \( k = 0 \), (b) \( k = 800 \), and (c) \( k = 1200 \).
The $G_M(s)$’s Nyquist diagrams for different $k$’s in the parallel branch can be compared and concluded that with the PFCD implementation, the stability is easily guaranteed. As it is shown in FIGURE 6, when $k$ is increased, the Nyquist diagram starts stretching around the critical -1 point, and there is no encirclement with further increment. This phenomenon is helping with the stabilization of the characteristic equation and consequently the stability of the closed-loop.

For the time response of the CL system, in FIGURE 7, using a few different weights it seems for $k=2500$ a suitable performance can be obtained. This $k$ gives an overshoot of 7% while the settling time (within 5% of the final value) is within 12 seconds.

![FIGURE 7: Time responses of the system for different values of $k$ in the PFCD design; if $k$ increases, the responses become slower but the overshoots decrease. For $k=2500$, the performance of the system is great with respect to the speed and quality of the response.](image)

To show the effectiveness of the method in the case that there are noise, mismatch, or disturbance existing in the system, assume there is 30% inaccuracy in evaluating the delays and largest coefficients of the numerator and denominator. Also, assume the disturbance function in FIGURE 3 is

$$d(s) = \frac{0.5}{3s+1} \quad (10)$$

Then, the time response was plotted with different $k$’s in the parallel branch to check for possible different behavior. This result for the perturbed plant is plotted in FIGURE 8. However, it can be seen that the response stays relatively robust to when there was no noise, mismatch, or disturbances in the system. In particular, for $k=2500$, the overshoot is now 17% and the settling time is about 13 seconds. This implies the method can be extended well to non-nominal cases. Nevertheless, it is deemed that more studies are needed to be performed on such systems. Therefore, some future studies will be devoted to closed-loop perturbed plants within PFCD configuration.

![FIGURE 8: Time responses of the PFCD control system in the presence of disturbance and mismatch model. In here, 30% inaccuracy for the available delays and the largest coefficients of the transfer function in Eq. (9) are considered. Also, a disturbance function as in Eq. (10) is assumed to perturb the system.](image)

**CONCLUSIONS**

In this work, a new method to control the nonminimum-phase systems is introduced. We investigated that if the weight of a higher order stable polynomial is increased compared to an unstable polynomial then the sum can be stabilized. Then, we formed a loose upper-bound on this weight in relation to the ratio of the polynomials infinity norm and present delays to guarantee the sum is stable. Correspondingly, we utilized this mathematical principle in a control system and named it the PFCD implementation. Throughout the paper, it has been pointed out the PFCD method can be more robust to noise compared to the ones given in IMC and GSP methods. From the statistics point of view, while in other methods, to implement a control system, we need much information about the NMP plant dynamics, i.e. the location of all the poles and zeros, are needed, the only required information in the PFCD design is the process parity of RHP-zeros and some of the infinity norms of the plant transfer functions. If we could plot the root-locus (even though a system might be of infinite dimensions), the feedforward path being RHP-even assures that the root locus is not crossed at its origin and with a proper weight in the parallel feedforward branch, the closed-loop poles lie in the LHP. Considering the given example in the previous section and its results, there are a few comments to bring in here:

1. For an absolutely stable process, the derivative in the parallel branch is to make the steady state error be zero in the presence of an integrator controller.
(2) With a RHP-even feed-forward path, there will be no "initial" undershoot; see more on the initial undershoots of system with odd and even RHP-zeros in [6].
(3) We can alleviate the time response overshoots at the expense of a slower speed by increasing the gain in the parallel branch.
(4) The response undershoot is small and not sensed significantly in the response.
(5) In the presence of noise, disturbance, or mismatch in the model, to bring the system back to stability, we just need to adjust the weight of the parallel branch which can be done with ease as opposed to adjusting the entire model in other methods. Usually a binary search with some tolerance can be done in the range of $[0, k_{cr}]$ to obtain a suitable response.

REFERENCES