TIP POSITION CONTROL OF SINGLE FLEXIBLE LINKS VIA PARALLEL FEEDFORWARD COMPENSATION

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

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

ABSTRACT  
Developed in this work, is a simple and innovative control method, by which a nonminimum-phase (NMP) process can be easily stabilized in a closed-loop setting. The method is named as the parallel feed-forward compensation with derivative effort (PFCD). Through use of a high order process, the control system designed by the PFCD method is shown to be less influenced by noise, disturbance, and model mismatch, compared to other methods. Moreover, the necessary data required for implementing the PFCD method are discussed. The proposed control method is illustrated on tip position control in a slewing beam as a flexible robot arm, in which the effectiveness of the PFCD method is demonstrated. In addition, the proposed control method is compared with the existing methods in terms of stability and performance. The paper is concluded with notes about the advantages.

Keywords: nonminimum-phase (NMP); stability; infinity norm; right-half plane (RHP) zero; left-half plane (LHP) zero; RHP-even; RHP-odd; open-loop (OL); closed-loop (CL); parallel feedforward compensator; root locus; time response.

INTRODUCTION  
In linear time-invariant control, nonminimum-phase (NMP) systems have gained significant attention [1]. This research interest naturally appears when a control system has dislocated actuators and sensors, by which RHP-zeros or delays may be introduced to the system [2-4]. An early practical NMP control implementation was given by introducing the Smith’s predictor (SP) [5]. Since then, there have been many SP-based methods, such as IMC and GSP [6-9], developed to cope with the nonminimum-phase systems. Because these methods create compensators of the same order as the plant’s order itself, there are several concerns as follows:

1. With inaccuracy in a system model, the performance can drop significantly.
2. Under noise or disturbance, the provided model can be very limited in action.
3. Even with perfect model estimation, the implementation of the method is limited by the hardware. For example, implementing a time delay requires a Padé approximation. The order of approximation can be very high, and thus, the controller becomes very complicated and sensitive to noise.
4. The processes that these models can deal with are somewhat limited. For example, a system with multiple delays is harder to investigate and synthesized within their methods.

Besides the above concerns, the complexity of the methods and ad-hoc solutions in the applications motivates the current effort. Presented in this paper is a new and simpler control method for NMP systems. This method makes use of a parallel feed-forward compensator with derivative effort, and therefore is called the parallel feed-forward compensation with derivative effort (PFCD). The mathematical intuition behind this method is explained in the next section.

Before we elaborate more on the new control method it is necessary to survey previous studies on control of NMP systems. The IMC model [6, 7], as shown in Figure 1(a), is one of the advanced methods that deals with NMP systems. However, the noise and mismatch model make the performance of the system susceptible. As a result, the mathematical model results are somewhat unrealistic. Although, in the later studies some modifications like a filter in the feedback loop were considered [6, 8], the method still suffers from high order controller issues. The GSP method [9], as shown in Figure 1(b), has the similar problems even though it has been reconfigured.
to be better understood and its implementation be more acceptable. Furthermore, the method requires exact pole-zero cancellation, which in reality is difficult to avoid. Consequently, this method is limited in utility and requires extreme care in its implementation. Later studies, like the work in [10], basically follow the same ideas, but with a better implementation so that the controller saturation is less likely to happen. However, the mismatch model, noise, and disturbance problems still exist.

Before moving onto the next sections, here, we lay out the foundations needed for the PFCD configuration through two relatively simple, but, fundamental lemmas.

**Stable polynomials algebra**

It is well-known that the summation of two stable polynomials may not be stable by itself, e.g. see [13]. Also it is a trivial task to find out the summation of one stable and one unstable polynomial can be unstable. In this paper, we represent two simple polynomial summations that will result into stable polynomials.

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

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

where \( A(s) \) and \( B(s) \) can be quasipolynomials (see [13]) for its definition) or normal polynomials in which the order of \( A(s) \) is at least equal to the order of \( B(s) \). Then by increasing \( k \), eventually, \( P(s) \) can be stabilized.

The intuition for the lemma is explained for the case when both the polynomials are normal polynomials since for the rotating beam model, a rational transfer function will be developed and there will be no need when the transfer functions are of quasi type. Thus, suppose

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

\[ B(s) = b_m \prod_{j=1}^{m} (s - z_j) \]

where \( \forall p_i \in \mathbb{R}^- \) and \( \forall z_j \in \mathbb{R} \) and \( s \) is the complex variable. It is assumed that the leading coefficient of \( A(s) \) is unity and the leading coefficient of \( B(s) \) is positive, i.e. \( b_m > 0 \). Intuitively, in Eq. (1) if we increase the weight, i.e. \( k \), of the stable polynomial, the polynomial terms of \( A(s) \) will dominate the polynomial terms of \( B(s) \). This is just an intuition and the solid proof can be seen using the root locus method in Figure 2. As the weight, \( k \), is increased, the roots of Eq. (1) will go from the roots of \( B(s) \) to near the roots of \( A(s) \). This will ensure that after some positive \( k \), the roots will be in the left half plane (LHP) of the root locus plane, and consequently, the polynomial in Eq. (1) becomes stable. This positive \( k \) should be large enough. However, it is easy to see that if

\[ k \geq \left| \frac{B(j\omega)}{A(j\omega)} \right|_\infty \]

where the right hand side is the infinity norm of the transfer function \( B(s)/A(s) \), that can be indicated with \( \mathcal{H}_\infty^{B/A} \), then the polynomial in Eq. (1) will not have a solution in order intersect with the imaginary axis; thus, becomes stable.

This lemma by itself is an interesting idea to move RHP-zeros of a process near to stable LHP-poles just by using a
parallel gain in the parallel path of the process and make the closed-loop system stable. However, it was found out the performance of the system would not be satisfactory, and also, there will be some steady state error that does not vanish even by further improvement of the controller. Thus, the application of Lemma 1 is disappointing, but helps to uncover of a second lemma that will improve the performance and make the steady state error to be zero if used in a compensation sense.

**Lemma 2**: Again consider the roots of the following polynomial

\[ P(s) = ksA(s) + B(s) = 0 \]  

where \( A(s) \) and \( B(s) \) are as were defined previously. However, there is an additional requirement for \( B(s) \) that the parity of its RHP-zeros should be even. Here, the claim is that there exists a \( k_{cr} \in \mathbb{R} \) such that for \( \forall k \geq k_{cr} \), \( P(s) \) in Eq. (5) will have stable roots.

The intuition for the Lemma 2 is same as Lemma 1, i.e. an increase on the weight of the stable polynomial should make the overall polynomial stable; however, note that there is a subtle difference between Eqs. (1) and (5). Since in this paper we deal with a rational transfer function model of a rotating beam, we will not cover the polynomials of quasi type; however, such systems will be covered in a different study. Consequently, herein, we concentrate on normal polynomials. The lemma can again be proven using pure math or root locus methods. In Lemma 2, the upper-bound is different than Eq. (4) for Lemma 1, and, it can be proven the following bound will ensure all the root locus have crossed to the LHP

\[ k \geq \mathcal{H}_{\alpha_0}^{B_f/A} + \mathcal{H}_{\alpha_0}^{A_f/A} \mathcal{H}_{\alpha_0}^{B_f/A} = k_{cr} \]

We will refer to this upper-bound, i.e. \( k_{cr} \), in the next sections. It should be mentioned that this upper-bound is not a mandatory bound for \( k \) in Eq. (5) to make it a stable polynomial, but it is just a certain upper-bound that for any \( k \) greater than that, the Eq. (5) will be a stable polynomial, i.e. all the roots will be in the LHP. This is shown in Figure 2 by the direction of the root locus while increasing the weight \( k \) in Eq. (5).

Note that, throughout this paper, we call a transfer function RHP-even if it has an even number of RHP-zeros. This implies the number of real RHP-zeros is also even. A similar concept exists for RHP-odd transfer functions. It has been found out that applying the Lemma 2 on a linear stable plant that is NMP and RHP-even has benefits to stabilize the feedback loop around a plant. This is the main idea that this paper represents such that only a few information about the plant, such as infinity norm and the parity of RHP-zeros are necessary so that to obtain a stable closed-loop with a zero steady state error and relatively satisfying performance. Compared to this little required information, the other methods need more information about the process poles and zeros. In fact, in their methodology it is necessary to have a dynamic model as close as possible to the real system. Hence, it is advised to understand that the other methods are intrinsically optimistic and somewhat unrealistic.

**BEAM MODELING**

In this paper we assume the beam model, as shown in Figure 3, is designed as follows:

1. The flexible beam is mounted on a hub that has a radius of \( R \), and mass moment of inertia of \( I_0 \).
2. The radius of the hub is not significant compared to the beam’s length.
3. The junction between the non-flexible hub and the flexible beam is welded. And a damping element with a damping coefficient of \( b \) has been considered at this junction to dampen the input torque.
4. The beam has homogenous mass density \( \rho A \), length \( L \), and stiffness \( EI \).

**Extended Hamilton’s principle**

In here, the slewing beam dynamic model is obtained by using the extended Hamilton’s principle to derive the governing equations. Starting from kinetic and potential energy, ignoring the gravity effect, and considering the virtual work we can derive the Hamilton’s equation

\[
\int_{t_0}^{t_1} (\delta T - \delta V + \delta W_{nc}) dt = 0, \\
\text{and } \delta q_k = 0, k = 1, 2, ..., n; t = 0, t_1
\]

where \( \delta q_k \) are the independent generalized coordinates such that for each of them the integral is calculated independently; \( T \), \( V \), and \( W_{nc} \) are the kinetic energy, potential energy, and the
virtual work of non-conservative forces, respectively. Here, the kinetic energy will be:
\[ T = \frac{1}{2} I_0 \dot{\theta}^2 + \frac{1}{2} \int_0^L \rho A \dot{\theta} \dot{\nu} \, dx \]  
(8)
where the first term is the hub's rotational kinetic energy and the second term is the integral over all the beam's differential elements kinetic energies. The speed vector for a differential element can be written as
\[ \dot{\nu} = \dot{\theta} \hat{k} \times (x + R)i + \dot{\theta} \hat{k} \times w^j + w^j \]  
(9)
where \( x \) is the element of the beam at distance \( x \) calculated from the junction, \( \theta \) is the hub angle calculated at the junction, \( w \) is the elastic deformation of the beam compared to the reference of beam angle.

The potential energy, \( V \) (not to be mistaken with the speed vector), is due to elastic deformation and can be calculated by integration as follows
\[ V = \frac{1}{2} \int_0^L E I (\frac{d^2 w}{dx^2})^2 \, dx \]  
(10)
In Eq. (7), the initial time has been considered to be at \( t = 0 \) and the end of the rotation happens at \( t = t_1 \). The given boundary-value problem, has two governing equations (GEs) based on the chosen generalized coordinates \( \theta \) and \( \delta w \).

\( W_{nc} \) is the virtual work in the slewing beam system. However, the work only is done at the hub and the beam’s junction
\[ \delta W_{nc} = \int_{beam} F \delta \mathbf{r} = \tau \delta \theta - b \delta \theta a_{x=0} \]  
(11)
where \( F \) is the external force on an element and \( \delta \mathbf{r} \) is its virtual displacement vector. The torque, \( \tau(t) \), at the hub can be treated as an external force acting on the beam system. Also, the damping force dissipates energy proportional to the angular velocity.

To solve the dynamics equations in an analytical way, a weak form of solution using the assumed modes method (AMM) is developed. In this paper, for the sake of simply showing the effect of the PFCD control methodology, only three modes are assumed. However, the reader will find it fairly easy to expand the PFCD idea for a higher number of modes in AMM. Further, the dynamics equations have been linearized so that a Laplace transform can be applied to those equations. At the end, we have shown that for a distributed system like the beam under survey, we can apply the PFCD implementation and get satisfactory results. The tip position of the beam is scrutinized and the time responses for different weight values in the PFCD implementation are depicted. The assumptions made about the elastic deformation are as follows
(1) Neglect the axial deformation and axial forces.
(2) Effects of shear deformation and gravity are negligible.
(3) For simplicity, it is assumed the beam is an Euler-Bernoulli beam.
(4) Elastic deflection of the beam, angular velocity and acceleration of the hub are small. Their product is close to zero and can be ignored. This helps the simplifications during the linearization step.

(5) Assume the properties of the beam are constant with length.

Using \( \delta \theta \) as a generalized coordinate, we can obtain the following governing equation
\[ \begin{align*}
(11) & \\
& + \int_0^L \rho A \dot{w} (x + R) \, dx + b \dot{\theta} = \tau(t) 
\end{align*} 
\]  
(12)
for \( t \in (0, t_1) \). And using \( \delta w \) as the second generalized coordinate, the second governing equation is as follows
\[ \rho A \dot{w} + \dot{\theta} (x + R) + E I \frac{d^4 w}{dx^4} = 0 \]  
(13)
for \( 0 < x < L \) and \( t e (0, t_1) \). Also, the boundary conditions are
\[ \begin{align*}
\frac{d^3 w}{dx^3} & = 0 \quad x = L \\
\frac{d^2 w}{dx^2} & = 0 \quad x = L \\
\frac{dw}{dx} & = 0 \quad x = 0 \\
w(x, t) & = 0 \quad x = 0 
\end{align*} 
\]  
(14)
It can be assumed that at \( t = 0 \), the system is at rest: i.e. \( w(x, t) = 0 \) and \( \theta(t) = 0 \).

Discretizing/Linearizing the governing equations
Due to the difficulty of finding an analytical solution to Eqs. (12)-(14), we can use a weaker form of solution obtained by using the assumed modes method in which the elastic response is assumed to be as follows
\[ w(x, t) = \sum_{i=1}^{n} \phi_i(x) q_i(t) \]  
(15)
where the \( \phi_i(x) \) are the mode shapes and \( q_i(t) \) are the corresponding modal coordinates for \( i = 1, 2, ..., n \). By replacing admissible functions for the mode shapes and reusing the extended Hamilton’s principle we will obtain discretized form of governing equations. Furthermore, to simplify and ignore coupling terms of the generalized coordinates (i.e. \( \theta, q_1, q_2, ... \)) around an equilibrium point, we shall ignore terms that have elements like: \( \dot{q} \dot{\theta}, \dot{q} \dot{\theta} \). The assumption is the angular velocity and elastic deformations are small, so, their product is negligible compared to other terms and their coupling terms will be neglected in a dynamic motion. In here, for the sake of space limit, we will put the dynamics of equations in the following form:
\[ M \ddot{X}(t) + C \dot{X}(t) + KX(t) = D \tau(t) \]  
(16)
where \( X(t) = [\theta(t) \quad q_1(t) \quad ... \quad q_n(t)]^T \). Here, it is needed to determine the values for \( M \) (inertia), \( C \) (damping), \( K \) (stiffness), and \( D \) (input) matrices. For example, when we only use 2 modes in Eq. (15) for the slewing beam in Figure 3, the coefficient matrices in Eq. (16) are as follows:
\[
M = \begin{bmatrix}
I_0 + \int_0^L \rho A(x + R)^2 dx & \int_0^L \rho A(x + R) \phi_1 dx & \int_0^L \rho A(x + R) \phi_2 dx \\
\int_0^L \rho A(x + R) \phi_1 dx & \int_0^L \rho A \phi_1^2 dx & \int_0^L \rho A \phi_1 \phi_2 dx \\
\int_0^L \rho A(x + R) \phi_2 dx & \int_0^L \rho A \phi_1 \phi_2 dx & \int_0^L \rho A \phi_2^2 dx
\end{bmatrix}
\]
\[
C = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix},
\quad K = \begin{bmatrix}
0 & \int_0^L E i_1 \phi_1'' dx & \int_0^L E i_1 \phi_2'' dx \\
0 & \int_0^L E i_1 \phi_1'' dx & \int_0^L E i_1 \phi_2'' dx \\
0 & \int_0^L E i_1 \phi_1'' dx & \int_0^L E i_1 \phi_2'' dx
\end{bmatrix}
\]
\[
D = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}
\]

Then, by the Laplace transform, the equation to be solved is
\[
M \mathbf{\ddot{x}}(s) + C \mathbf{x}(s) + K \mathbf{x}(s) = D \mathbf{\ddot{\tau}}(s)
\]
where \( \mathbf{\ddot{x}}(s) \) is the Laplace transform of \( \ddot{x}(t) \) from Eq. (16). The solution to the system parameters in the Laplace domain will be as follows
\[
\mathbf{\ddot{x}}(s) = \begin{bmatrix} \ddot{\theta}(s) \\ \ddot{q}_1(s) \\ \ddot{q}_2(s) \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \mathbf{\ddot{\tau}}(s)
\] (18)

where
\[
x_{11} = s^2 \left( I_0 + \rho AL \left( \frac{L^2}{3} + RL + R^2 \right) \right) + sb
\]
\[
x_{21} = x_{12} = s^2 \int_0^L \rho A(x + R) \phi_1 dx
\]
\[
x_{31} = x_{13} = s^2 \int_0^L \rho A(x + R) \phi_2 dx
\]
\[
x_{22} = s^2 \int_0^L \rho A \phi_1'^2 dx + \int_0^L E i_1 \phi_1'' dx
\]
\[
x_{23} = x_{32} = s^2 \int_0^L \rho A \phi_1 \phi_2 dx + \int_0^L E i_1 \phi_2'' dx
\]
\[
x_{33} = s^2 \int_0^L \rho A \phi_2'^2 dx + \int_0^L E i_1 \phi_2'' dx
\]

and for \( i = 1,2 \)
\[
\phi_i'' = \frac{d^2 \phi_i}{dx^2}
\]

It is easy to find the trend of the solution in Eq. (18) while using a higher number of modes in Eq. (15).

Then, by solving for the desired transfer function, which in here will be the tip position of the beam (a combination of the transfer functions in Eq. (18)), we can define our process transfer function; and, try to stabilize it using the control method that will be introduced in the next section based on Lemma 2 in the previous section. A numerical example with the numeric beam parameters is given in the next section.

Note that for the pinned end, an admissible function should be chosen such that satisfies the essential boundary conditions of zero displacement and zero slope, i.e. the existing essential (geometric) boundary conditions at \( x = 0 \).

**Assumed modes method (AMM)**

By using the assumed modes method, we replace the elastic deformation by a series of modes and give weights to each one such that they estimate the real displacement as closely as possible. The more accurate chosen mode shape functions are, the better the final displacement estimation will be by using a truncated series. So, in here, by a good approximation, we replaced the mode shapes as the ones that happen for a beam without rotation. These series of mode shapes are as follows
\[
\phi_k(x) = \cosh(\beta_k x) - \cosh(\beta_k x)
\]
\[
- \alpha_k \left( \sinh(\beta_k x) - \sin(\beta_k x) \right)
\] (19)
for \( k = 1,2,3,...,n \), where,
\[
\beta_k = \mu_k L, \quad k = 1,2,3,...,n
\] (20)
where \( L \) is the beam length and
\[
\begin{cases}
\mu_1 = 1.875 \\
\mu_2 = 4.694 \\
\mu_3 = 7.855
\end{cases}
\] (21)
and
\[
\alpha_k = \frac{\cosh(\mu_k) + \cos(\mu_k)}{\sinh(\mu_k) + \sin(\mu_k)}
\]

This choice of mode shapes for a rotary beam was also selected in [15-17]. These mode shapes are the exact solutions for a clamped-free non-rotating beam [15].

**Tip position transfer function**

The transfer functions between the input torque, \( \bar{\tau}(s) \), and output tilt angle, \( \bar{\theta}(s) \), or elastic displacement, \( \bar{w}(x,s) \), can be found by solving the equations of motion for \( \bar{\theta}(s) \), and \( \bar{q}_k(s) \), for \( k = 1,2,...,n \).

Therefore, one can find the elastic displacement, \( \bar{w}(x,s) \), through sum of the assumed modes function in Eq. (22). When \( \theta \) angle is small, for any \( x \in [0,L] \), the position of the \( \tilde{y} \) coordinate can be found as in
\[
\bar{w}(x,s) = \sum_{k=1}^{n} \phi_k(x) \bar{q}_k(s)
\] (22)
\[
\bar{y}(x,s) = (x + R) \bar{\theta}(s) + \bar{w}(x,s)
\] (23)
Then, the tip position transfer function with the torque input will be as follows
\[
\frac{\bar{y}(L,s)}{\bar{\tau}(s)} = (L + R) \frac{\bar{\theta}(s)}{\bar{\tau}(s)} + \sum_{k=1}^{n} \phi_k(L) \frac{\bar{q}_k(s)}{\bar{\tau}(s)}
\] (24)
The transfer function that we considered for control is the tip position to torque in Eq. (24).
PFCD control methodology for the beam

Investigating transfer functions for the slewing beam has become a trivial task in the literature, e.g. see [2], [14-16], [17] and [18]. Nowadays, researchers are familiar with possible outcomes from different dynamical models that are offered for simple distributed systems. The mutual outcome in these models is that distributed systems usually tend to have a NMP behavior due to wave propagation speed and the difference between slow and fast frequency responses; thus, the closed-loop feedback control should be carefully implemented. Also, as it is shown in the example of the next section, the inherently stable poles are very close to the imaginary axis for a slightly damped system. Consequently, a traditional method, such as the IMC or GSP method, that involves pole-zero cancellation, may have an unsuccessful implementation, e.g. see [16]. Here, we show the PFCD method gives an easy solution to convert the NMP behavior of the open-loop (OL) system into a MP behavior; also, makes the closed-loop (CL) stable. The time response and root locus plots are simulated to verify the efficacy of this method.

Applying the PFCD for the beam transfer function

If we consider the process to be controlled to be the tip position of the beam, then

\[ G_0(s) = \frac{\hat{y}(L, s)}{\hat{f}(s)} \]  

(25)

However, to be general in this section, we will assume that we are dealing with an arbitrary NMP, but with a stable transfer function. Since the feed-forward path of the PFCD configuration in Figure 4 requires a RHP-even transfer function, in the case that there is a RHP-odd plant, we need to add an all-pass filter, i.e. \( C_f(s) = (s + z)/(s + p) \) where \( p = -z > 0 \). Also, the root locus approach will not have a traditional meaning if the numerator and denominator have different signs, so we have added a sign multiplier that will take care of neutralizing the possible negative sign that the process may carry. The branch with a compensator that only includes a simple derivative and gain is parallel to the feed-forward path. Based on Lemma 2’s math, by increasing the gain in the parallel branch, the characteristic equation similar to Eq. (5), will turn into a summation of a weighted stable denominator polynomial with unstable numerator polynomial which produces a stable polynomial in the open-loop system, and consequently, the closed-loop system that includes the parallel compensator will become stable with a maximum possible weight in Eq. (6).

Tuning the parallel derivative gain

Since Lemma 2 states there exists a limited upper-bound for the weight in Eq. (5) that makes the summation of two polynomials stable, we conclude that the stability of the closed-loop system is guaranteed, but a better performance requires a try and error on the weight value that we pick for the parallel compensator. This weight can be further tuned to give a better possible performance for the system in Figure 4. In the example in the next section, for different \( k \)'s, we have demonstrated how the weight can influence the closed-loop time response and the root locus plot.

SIMULATION AND RESULTS

In the rest, with the given beam parameters in Table 1, the PFCD configuration to control the tip position of the slewing beam is implemented.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_0 )</td>
<td>0.5</td>
<td>kg.m²</td>
</tr>
<tr>
<td>( R )</td>
<td>0.1</td>
<td>m</td>
</tr>
<tr>
<td>( b )</td>
<td>0.2</td>
<td>N.m.s</td>
</tr>
<tr>
<td>( L )</td>
<td>2</td>
<td>m</td>
</tr>
<tr>
<td>( \rho )</td>
<td>25</td>
<td>kg.m⁻³</td>
</tr>
<tr>
<td>( A )</td>
<td>0.01</td>
<td>m²</td>
</tr>
<tr>
<td>( EI )</td>
<td>10</td>
<td>N.m²</td>
</tr>
</tbody>
</table>

The tip position transfer function is calculated as follows using only 3 terms in the AMM with the admissible functions that were defined in Eqs. (19)-(21):

\[ G_0(s) = \frac{\hat{y}(L, s)}{\hat{f}(s)} = -0.124 \left( \frac{s - 16.17}{s + 16.17} \right) \frac{1}{s(s + 0.1573)(s^2 + 0.2169s + 73.65) \left( s^2 - 150s + 6767 \right) \left( s^2 + 150s + 6767 \right)} \times \left( s^2 + 0.02s + 1277 \right)(s^2 + 0.00327s + 9594) \]

Note that \( G_0(s) \) has 3 RHP zeros, \( z_1 = 16.17 \) and \( z_{2,3} = 75 \pm j33.79 \). 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 \( sgn = sgn(-0.124) = -1 \). After applying this methodology, \( \bar{G}'(s) \) in the feed-forward path will be:
\[ G'(s) = \frac{\tilde{y}(L, s)}{u(s)} = \frac{0.124 (s - 200)(s - 16.17)(s + 16.17)}{s(s + 200)(s + 0.1573)(s^2 + 0.2169s + 73.65)} \times \frac{(s^2 - 150s + 6767)(s^2 + 150s + 6767)}{(s^2 + 0.02s + 1277)(s^2 + 0.00327s + 9594)} \]

As we see, the transfer function of \( G'(s) \) in the feed-forward path is RHP-even which makes the root locus not cross the imaginary axis through the origin.

Simulations are performed for the configuration in Figure 4. However, it is necessary to note that the Lemma 2 infers the plant should have a stable denominator in the PFCD configuration to be effective. Here, since the process transfer function is of type-1 stable and not an absolutely stable plant, we need to not add a derivate to the parallel branch and should leave it as just a gain. However, in the case that the NMP process is a stable process (all the poles to be in the LHP), we need to leave the derivative term in the parallel branch as it is. The inherent integrator (in systems of type-1) will help to have a closed-loop steady state error of zero, and consequently, a derivate in the parallel branch is not required and just a simple parallel gain will make a satisfactory stable result; however, note this is still a consequence of Lemma 2 and not Lemma 1.

![Figure 5. Step-input time responses of the system for different values of \( k \) in the parallel branch; by increasing \( k \), the transition to steady state becomes longer but the overshoots become smaller. For the \( k \approx 120 \), there seems to be a fair trade-off between the settling time and overshoot.](image)

In Figure 5, we have considered a few different weights in the PFCD method to pick the one that gives the best performance for the time response to a step-input. It seems for \( k \approx 120 \) a suitable performance can be achieved. This \( k \) gives an overshoot of 5% while the settling time (within 5% of the final value) is only about 24 seconds. However, depending on the expertise and application, maybe a faster response with more overshoot is desired and then a smaller weight should be chosen.

Also, we can compare the root locus plots for different weights in the parallel branch and conclude that with the PFCD implementation, the stability is guaranteed.

![Figure 6. Open-loop root locus for different values of \( k \) in the parallel compensator. The open-loop zeros move to the LHP as \( k \) increases. (a) \( k = 0 \), (b) \( k = 5 \times 10^{-6} \), and (c) \( k = 5 \).](image)

As it is shown in Figure 6, when \( k \) is increased, all the root locus branches move to the LHP where the stable poles reside. This phenomenon is helping with the stabilization of the characteristic equation and consequently the stability of the closed-loop.

Remarks: Accordingly, there are a few comments to point out in here:

1. Since the feed-forward path is RHP-even, there is no initial undershoot. This has been also discussed in [4].
2. The time response undershoot and overshoot values can be brought down at the expense of a slower speed by increasing the weight, i.e. \( k \).
3. Even though the computed \( k_{cr} \) in Eq. (6) is relatively large (in the order of thousands), as soon as all the branches cross the imaginary axis, the closed-loop system
will be stable. Thus, after a try and error it was found out that even with a much smaller weight in the parallel branch, all the branches can be transferred to the LHP. After all, the Lemma 2 is only predicting such a behavior by increasing $k$. The OL stability happens for $k \geq 0.42$.

(4) Since the system has stable poles, and, the zeros of the system are symmetric (e.g., see [2]), it can be shown the root locus branches will not return to the RHP, and consequently, it will be easier to find the weight value for which the PFCD implementation makes the characteristic equation stable. Any larger value that we pick for this weight will still satisfy the stability condition.

(5) The resultant PFCD undershoots are very small that is not sensed much in the response.

(6) Since the linearized plant has one pole at the origin (a type-1 system), a derivative for having a zero steady-state error is not required in the parallel branch; hence, only having a gain for the parallel feed-forward branch in Figure 4 suffices, i.e. $C_p(s) = k$. Also, the controller should just be a gain for the same reasons, i.e. $C(s) = 1$.

(7) 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.

(8) The selected response may seem to be slow. However, note that the poles of the system are only slightly damped and they are very close to the imaginary axis. Further study is required to justify this slow speed disadvantage.

CONCLUSIONS

In this paper, a novel control method for NMP systems is developed. The idea behind the PFCD method is based on the intuition that the sum of two polynomials, in which the higher order polynomial is stable, can be stabilized if the weight on the higher order polynomial is increased. We established a loose upper-bound on the weight of the stable polynomial in relation to the transfer function infinity norm. This weight is conservative and typically a better performances can be found using a binary search on the $[0, k_{cr}]$ interval. Then, we carried this fact to be used in a control feedback loop and called it the PFCD implementation. The closed-loop stabilization methodology using the PFCD method is much simpler than the ones given in IMC and GSP methods. In other methods, a lot of information about the NMP plant dynamics, i.e. poles and zeros, are required while the only required information in the PFCD design is the infinity norm of the plant and whether the plant RHP-zeros’ count is even or odd. The even number of RHP-zeros assures the root branches cannot enter the LHP through the origin. Note that the derivative in the parallel branch helps the steady state error to become zero. The mismatch model, noise and disturbances can affect the PFCD configuration less compared to other existing methods since in PFCD there is only one weight to adjust (i.e. $k$) as opposed to adjusting the entire model.

REFERENCES