

Robust Design of Unconstrained Predictive Controllers

Kostas Hrissagis[§], Oscar D. Crisalle[§], and Mario Szaier[‡]

[§] Chemical Engineering Department
University of Florida
Gainesville FL 32611

[‡] Electrical Engineering Department
The Pennsylvania State University
University Park PA 16802

ABSTRACT

A systematic and complete design procedure for robust predictive controllers is proposed. The synthesis method is based on rigorous theoretical foundations, without resorting to approximations or ad hoc design guidelines, yet it remains a viable tool for practical utilization. A significant feature is that the robustified predictive controller retains the servo performance of a nominal predictive controller designed using conventional methods. In addition, the robust predictive controller can be designed to guarantee perfect steady-state rejection of asymptotically constant disturbances. The robust design method is developed for systems affected by unmodeled dynamics, and is based on solving a discrete-time model-matching problem. It is shown that the robustified controller can legitimately be classified as a predictive controller. An illustrative design example is given.

1. INTRODUCTION

Predictive control strategies have received much attention in the literature and have also found acceptance in industry. Currently there is an increasingly visible interest in the research community to revisit the predictive-control design techniques with the intention of including robustness features that guarantee stability or adequate performance when the plant model is uncertain. Zafiriou [1] uses a contraction-mapping property to derive sufficient robust-stability conditions for a quadratic DMC technique with constraints. Although the approach is rigorous, it is likely to require large computational effort. More recently, Genceli and Nikolaou [2] and Zheng and Morari [3] derive sufficient robust-stability conditions for a control technique based on a linear cost functional instead of the more classical quadratic functional of prevalence in predictive control. Robinson and Clarke [4] investigate indirectly the robustness of the GPC technique by analyzing two particular designs, namely a dead-beat and a mean-level controller. A more comprehensive approach to predictive control robustification is found in the work of Kouvaritakis *et al.* [5]. The authors are the first to make use of a Youla parametrization to robustify the closed loop with respect to unstructured perturbations, and use a simple approach to find polynomials or fixed-order transfer functions to approximate the unknown parameters.

This paper presents a systematic procedure for robustifying predictive controllers in the presence of unstructured modeling uncertainty. The approach consists of parametrizing a nominal predictive controller which is designed using conventional and well-established methods. An important feature is that the parametrization technique preserves the servo dynamics of the nominal controller. The method is applicable to unconstrained predictive control designs that use transfer-function plant models corrupted by unstructured uncertainty, and has the ability to include an integrator in the robust controller, hence guaranteeing steady-state rejection of asymptotically constant disturbances.

2. NOMINAL PREDICTIVE CONTROL DESIGN

The design of nominal predictive controllers is vastly documented in the literature. In particular, a wealth of knowledge is available to resolve crucial design issues such as nominal closed-loop stability, and parameter tuning. Typically, predictive controllers are deployed by executing at every sampling instant an algorithm that solves a quadratic optimization problem. For analysis purposes, it is desirable to represent the algorithmic controller in terms of transfer functions, thus allowing the utilization of classical z-domain tools for analyzing stability and performance. This section presents a brief review of the analysis technique discussed in [6], which casts an algorithmic predictive-control law of the GPC type into a form involving transfer-function operators. The resulting nominal controller is used later as the basis for the design of a robust controller.

Consider the nominal process model

$$A(z) y(z) = B(z) u(z) \quad (2.1)$$

where $y(z)$ and $u(z)$ are the process output and input respectively, and $A(z)$ and $B(z)$ are the coprime polynomials

$$A(z) = z^n + a_{n-1}z^{n-1} + \dots + a_0 \quad (2.2)$$

$$B(z) = b_m z^m + b_{m-1}z^{m-1} + \dots + b_0 \quad (2.3)$$

of order n and m , respectively, where $n > m$. The optimal control action is obtained by minimizing the following

quadratic performance index which includes predicted future output errors and a penalization of future control moves:

$$J(t) = \sum_{i=1}^{N_y} [r(t+i) - y(t+i)]^2 + \lambda \sum_{i=1}^{N_u} [\Delta u(t+i)]^2 \quad (2.4)$$

where $\{r(t+i)\}$ is the sequence of future values of the set point, $\{y(t+i)\}$ is the sequence of predicted future values of the output, $\{\Delta u(t+i)\}$ is the sequence of future control increments, λ is the move-suppression parameter used to penalize excessive control energy, and parameters N_y and N_u are the prediction and control horizons, respectively.

By definition, a predictive control law is an algorithm that at every sampling instant produces the control move $u(t)$ that minimizes the functional (2.4) for the prescribed set point sequence $\{r(t)\}$. The optimal control move is naturally found by differentiating (2.4), equating the result to zero, and solving for $u(t)$. Following the development in [6] it is possible to write the resulting control law in terms of transfer-function operators in the form

$$\frac{R(z)}{z^n} u(z) = T(z)r(z) - \frac{S(z)}{z^n} y(z) \quad (2.5)$$

which includes the polynomial operators

$$R(z) = z^n + r_{n-1}z^{n-1} + \dots + r_0 \quad (2.6)$$

$$S(z) = s_n z^n + s_{n-1}z^{n-1} + \dots + s_0 \quad (2.7)$$

$$T(z) = t_N z^N + t_{N-1}z^{N-1} + \dots + t_1 z \quad (2.8)$$

where

$$R(1) = 0 \quad (2.9)$$

$$T(1) = S(1) \quad (2.10)$$

and where the coefficients of the moving average polynomial $S(z)$, the regressor polynomial $R(z)$, and the set-point advancement polynomial $T(z)$ are functions of the tuning parameters N_y , N_u , and λ , and of the model polynomials $A(z)$ and $B(z)$. Note that (2.9) implies that the predictive control law (2.5) includes an integrator. A block-diagram representation of the predictive control structure is shown in Figure 1a. Further details of the derivation can be found in [6].

Note that the transfer functions operating on $u(z)$ and $y(z)$ in the nominal predictive controller (2.5) are of order n , the order of the nominal plant model. It is also significant to note that the set-point advancement polynomial $T(z)$ is of degree equal to the prediction horizon N . Since $N \geq n$ is a common tuning prescription the order of $T(z)$ exceeds the order of $R(z)$ making the control law nonproper (noncausal) with respect to the set-point signal. This noncausality is a natural consequence of the inclusion of future values of the set point in (2.4). Figure 1a shows that the advancement polynomial acts on the set point to produce the intermediate signal $w(z) = T(z)r(z)$, which has the simple time-domain representation

$$w(t) = t_N r(t+N) + t_{N-1} r(t+N-1) + \dots + t_1 r(t+1) \quad (2.11)$$

It is useful to remark that the nominal model (2.1) and the functional (2.4) are simpler versions of more elaborate formulations that improve the design performance at the expense of added complexity [7]. These design choices can be accommodated within the framework proposed in this paper through obvious modifications.

Figure 1a illustrates the closed-loop established when the nominal predictive controller (2.5) is connected to the process (2.1). Note that the servo dynamics of the closed loop are fully characterized by the equations

$$[A(z)R(z) + B(z)S(z)] y(z) = z^n B(z)T(z) r(z) \quad (2.12)$$

$$[A(z)R(z) + B(z)S(z)] u(z) = z^n A(z)T(z) r(z) \quad (2.13)$$

Therefore, the stability of the closed-loop for a given nominal predictive controller can be easily established by calculating the roots of the characteristic polynomial $A(z)R(z) + B(z)S(z)$. Furthermore, due to the presence of the integral action (2.9) in the controller and to the gain equality (2.10), the closed-loop dynamics described by (2.12) are guaranteed to realize zero-offset in the servo response. The integrator also guarantees perfect steady-state disturbance rejection for all disturbance signals that reach a steady-state.

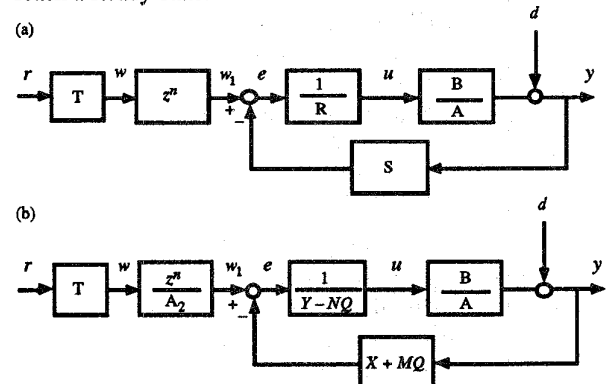


Figure 1. (a) Structure of a nominal predictive controller. (b) Structure of the parametrized predictive controller featuring the Youla parameter $Q(z)$.

3. CONTROLLER PARAMETRIZATION

In this section the nominal predictive controller (2.5) is parametrized in terms of a transfer function $Q(z)$ selected in the spirit of Wiener-Hopf design. However, a modification in the parametrization is introduced to achieve two important design requirements: (i) the parametrized controller must preserve the servo performance and the steady-state disturbance rejection properties of the nominal controller, and (ii) the parametrized controller must also be a predictive controller.

Consider a nominal predictive controller (2.5) that stabilizes the closed loop system (2.12)-(2.13). Because of the stability condition, the nominal closed-loop characteristic polynomial

$$A^*(z) = A(z)R(z) + B(z)S(z) \quad (3.1)$$

of degree $2n$ is Schur. In order to parametrize the controller, consider a coprime fractional representation of the nominal plant model (2.1) of the form

$$g_0(z) = \frac{B(z)}{A(z)} = \frac{N(z)}{M(z)} \quad (3.2)$$

where $N(z)$ and $M(z)$ are proper and stable transfer functions that satisfy the Diophantine equation

$$N(z)X(z) + M(z)Y(z) = 1 \quad (3.3)$$

for some pair of stable and proper transfer functions $X(z)$ and $Y(z)$. A suitable $(M(z), N(z))$ pair can be readily derived from the nominal characteristic polynomial (3.1). The procedure consists of first factoring the closed-loop characteristic polynomial in the form $A^*(z) = A_1(z)A_2(z)$, where both $A_1(z)$ and $A_2(z)$ are of degree n , and then dividing both sides of (3.1) by the factored characteristic polynomial to obtain

$$\frac{A(z)R(z)}{A_1(z)A_2(z)} + \frac{B(z)S(z)}{A_1(z)A_2(z)} = 1 \quad (3.4)$$

Finally, stable and proper factorizations that satisfy (3.3) are easily obtained by defining

$$M(z) := \frac{A(z)}{A_1(z)}, \quad N(z) := \frac{B(z)}{A_1(z)} \quad (3.5)$$

$$X(z) := \frac{S(z)}{A_2(z)}, \quad Y(z) := \frac{R(z)}{A_2(z)} \quad (3.6)$$

where $X(z)$ and $Y(z)$ are clearly stable and proper rational transfer functions. This result allows writing the nominal predictive control law (2.5) in the equivalent form

$$Y(z)u(z) = Z(z)r(z) - X(z)y(z) \quad (3.7)$$

$$Z(z) := \frac{z^n}{A_2(z)}T(z) \quad (3.8)$$

The set of all solutions to (3.3) can be written in terms of the transfer functions (3.5)-(3.6) and a proper and stable transfer-function $Q(z)$ through the well-known relations

$$X'(z) = X(z) + M(z)Q(z) \quad (3.9)$$

$$Y'(z) = Y(z) - N(z)Q(z) \quad (3.10)$$

Therefore, the set of all stabilizing controllers with the structure (3.7) is parametrized in the form

$$[Y(z) - N(z)Q(z)]u(z) = Z(z)r(z) - [X(z) + M(z)Q(z)]y(z) \quad (3.11)$$

to yield the control structure shown in Figure 1b. Clearly, setting $Q(z) = 0$ reduces the parametrized predictive

controller (3.11) to the nominal predictive controller (3.7).

Note that in contrast to the standard Youla parametrization approach, the transfer function $X(z) + M(z)Q(z)$ appears in the feedback path of Figure 1b, instead of appearing in the control block immediately preceding the plant. This deliberate departure from the standard approach, in conjunction with the factorizations (3.5) and (3.6) that make use of the nominal closed-loop polynomial, introduces highly desirable properties in the parametrized input-output maps as explained in the sequel.

Proposition 3.1. The nominal control loop of Figure 1a and the parametrized control loop of Figure 1b have identical servo transfer functions $y(z)/r(z)$ and $u(z)/r(z)$.

Proof. Carry out block-diagram algebra on each figure to derive in both cases the servo transfer functions $y(z)/r(z)$ and $u(z)/r(z)$ that are immediately obtained after a rearrangement of factors in equations (2.12) and (2.13).

Corollary 3.1. Given that the nominal controller (2.5) is a predictive controller, then the parametrized controller (3.11) is also a predictive controller.

The proof of corollary 3.1 is based on the fact that both controllers produce the same sequence $\{u(t)\}$. See [8] for details. Since any allowable parameter $Q(z)$ yields the same servo transfer functions $y(z)/r(z)$ and $u(z)/r(z)$, the parametrized controller has the intrinsic capability of preserving the nominal servo performance. Also note that although the terms containing $Q(z)$ effectively cancel out in the servo transfer functions, the transfer function $e(z)/w_1(z) = M(z)[Y(z) - N(z)Q(z)]$ in Figure 1b is affine in $Q(z)$, as in the standard Youla parametrization method. This allows shaping the loop sensitivity while simultaneously retaining nominal performance.

4. DESIGN OF ROBUST PREDICTIVE CONTROLLERS

When the nominal model (2.1) is not exact due to the presence of modeling errors, the plant transfer function $g(z)$ may be written in the form

$$g(z) = g_0(z) + \delta(z) \quad (4.1)$$

where $g_0(z) = \frac{B(z)}{A(z)}$ is the nominal plant model, and $\delta(z)$ is an unstructured perturbation. Without loss of generality the developments are specialized for the case of additive perturbations, where

$$|\delta(e^{j\omega})| \leq |W(e^{j\omega})| \quad \forall \omega \quad (4.2)$$

and where the uncertainty weight $W(z)$ is a stable and proper transfer function. The perturbation $\delta(z)$ is assumed to be such that $g(z)$ and $g_0(z)$ have the same number of

unstable poles. The case of multiplicative perturbations, as well as other typical unstructured uncertainty representations can be treated in an analogous way.

The objective is to design a robust predictive controller that stabilizes the closed loop for all the members of the uncertain family of plants (4.1)-(4.2), and that in the nominal case where $\delta(z)=0$ it recovers the performance of a nominal predictive controller (2.5) which is designed solely on the basis of the nominal model $g_o(z)$. The stability robustness of the closed loop shown in Figure 2, which includes the parametrized controller (3.11) and the uncertain family of plants (4.1)-(4.2), can be analyzed using H_∞ theory concepts summarized in Theorem 3.1 below. The proof follows the standard theory (c.f. [9]) and is therefore omitted.

Theorem 3.1. A necessary and sufficient condition for the robust stability of the closed-loop system of Figure 2 is the inequality condition

$$\|W(z)C(z)S_f(z)\|_\infty < 1 \quad (4.3)$$

where

$$C(z) := \frac{X(z) + M(z)Q(z)}{Y(z) - N(z)Q(z)} \quad (4.4)$$

and

$$S_f(z) := M(z)[Y(z) - N(z)Q(z)] \quad (4.5)$$

We propose a systematic procedure for solving the robust synthesis problem without resorting to approximations for the Youla parameter. A particular challenge to the design problem posed is the objective of including an integrator in the robustified controller in order to guarantee effective disturbance rejection.

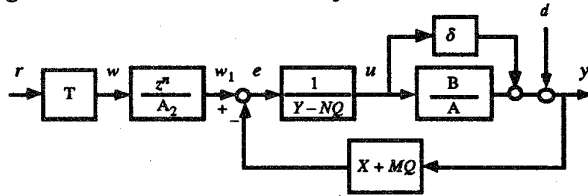


Figure 2. Structure of a robust predictive controller for a plant with an additive uncertainty.

4.1 Robust Synthesis

The details of the robust synthesis technique vary depending on whether the nominal plant model is stable or unstable, and in the latter case, depending on the presence of poles on the unit circle.

4.1.1 Base Case for Robust Design — Unstable Plant with no Poles on the Unit Circle.

Consider the robust predictive controller design problem for the case where the nominal plant model $g_o(z)$ is unstable but has no poles on the unit circle. The synthesis problem is attacked by rewriting the robust

stability condition (4.3) in the equivalent model-matching form

$$\|T_1(z) - T_2(z)Q(z)\|_\infty < 1 \quad (4.6)$$

where

$$T_1(z) = W(z)X(z)M(z) \quad (4.7)$$

$$T_2(z) = -W(z)M^2(z) \quad (4.8)$$

Inequality (4.6), which is affine in the unknown parameter $Q(z)$, is obtained by substituting equations (4.4) and (4.5) into inequality (4.3). The model-matching problem is commonly approached in the context of H_∞ control theory using the γ -iteration process, where (4.6) is substituted by the alternative inequality

$$\|T_1(z) - T_2(z)Q(z)\|_\infty \leq \gamma \quad (4.9)$$

where γ is a positive scalar parameter selected by the designer. A robust design is obtained if $Q(z)$ is found for a specified $\gamma < 1$. The key is then to be able to synthesize a Youla parameter with a reliable algorithm. We advocate the use of a z-domain technique proposed by Rotstein and Sideris [10]. The algorithm solves the Nehari extension problem, which consists of approximating a stable transfer function $R(z)$ with an antistable (all poles outside the unit circle) transfer function $Q_R^-(z)$, where the tilde superscript denotes the conjugate operation $Q_R^-(z) := Q_R(1/z)$. Specifically, the problem calls for finding a function $Q_R^-(z)$ such that

$$\|R(z) - Q_R^-(z)\|_\infty \leq \gamma \quad (4.10)$$

The only inputs required are the transfer function $R(z)$ and the scalar γ . A necessary condition for the existence of a solution is that γ must be greater than the Hankel norm of $R(z)$, i.e., $\gamma > \|R(z)\|_H$. One solution of particular significance is the central controller, denoted $Q_{R,c}(z)$, because the set of all the solutions can then be easily found as an explicit function of the central controller. The model-matching problem (4.9) can be written in the form (4.10) through a series of norm-preserving operations. The procedure consists in first factoring $T_2(z)$ in the form

$$T_2(z) = T_{ap}(z)T_{mp}(z) \quad (4.11)$$

where $T_{ap}(z)$ is an all-pass function and $T_{mp}(z)$ is a stable minimum-phase function, and then finding $T_{ap}^-(z) = T_{ap}(1/z)$ and carrying out the decomposition

$$T_{ap}^-(z)T_1(z) = R_a(z) + R_s(z) \quad (4.12)$$

where $R_a(z)$ and $R_s(z)$ are an antistable and a stable transfer function, respectively. The transfer-function input to the algorithm is

$$R(z) = R_a^-(z) \quad (4.13)$$

Then the algorithm produces the central controller

$Q_{R,c}(z)$, and a solution to the original model-matching problem (4.6) is simply recovered as

$$Q(z) = T_{mp}^{-1}(z) \left(R_s(z) + Q_{R,c}(z) \right) \quad (4.14)$$

The final robust predictive controller design for the base case is obtained by substituting in the structure (3.11) the Youla parameter (4.14) and the factorizations (3.5)-(3.6).

4.1.2 Robust Design for the Case of an Unstable Plant with Poles on the Unit Circle

When the nominal plant model $g_0(z)$ has poles on the unit circle, the standard H_∞ control theory is no longer applicable. In addition, the factorization (4.11) is no longer possible because no minimum-phase stable transfer function can possibly satisfy the equality. This difficulty is circumvented by introducing a change of variable that maps unit-circle poles to a circle of larger radius. Let $z = \rho\zeta$, where $\rho > 1$ is a scalar, and define the operators

$$T_1'(\zeta) = T_1(z/\rho) \quad (4.15)$$

$$T_2'(\zeta) = T_2(z/\rho) \quad (4.16)$$

Then the base-case design problem (4.9) can be posed in terms of the transformed variable ζ in the form

$$\|T_1'(\zeta) - T_2'(\zeta)Q(\zeta)\|_\infty < \gamma \quad (4.17)$$

and can be solved for $Q'(\zeta)$ using the base-case algorithm as described in Section 4.1.1. The z -domain Youla parameter is simply recovered by transforming the result back to the original space, *i.e.*,

$$Q(z) = Q'(\rho\zeta) \quad (4.18)$$

The final robust predictive controller design for this case is obtained by substituting in the structure (3.11) the Youla parameter (4.18) and the factorizations (3.5)-(3.6). Using the Maximum Modulus Theorem, it follows that the transformed design problem (4.17) is related to the original problem (4.6) through the inequality

$$\|T_1'(\zeta) - T_2'(\zeta)Q(\zeta)\|_\infty \geq \|T_1(z) - T_2(z)Q(z)\|_\infty \quad (4.19)$$

Therefore, the transformed design represents only a sufficient condition for stability. If no $Q'(\zeta)$ can be found that satisfies (4.17), then a smaller value for ρ should be adopted and the design is repeated.

4.1.3 Robust Design for the Case of a Stable Plant

When the nominal plant model $g_0(z)$ is stable the robust design is straightforward. In the normal case where the weight $W(z)$ is minimum phase (*i.e.*, it does not have zeros on or outside the unit circle), a solution to the robust synthesis problem (4.6) is obtained by setting

$T_1(z) - T_2(z)Q(z) = T_3(z)$, where $T_3(z)$ is a user-specified stable biproper transfer function that satisfies the contraction condition $\|T_3(z)\|_\infty < 1$. Solving for the unknown parameter as a function of $T_3(z)$ yields the design equation

$$Q(z) = T_2^{-1}(z)(T_1(z) - T_3(z)) \quad (4.20)$$

where $Q(z)$ is clearly a stable transfer function because $T_2(z)$ is minimum phase. In the alternative case where the weight $W(z)$ is nonminimum phase, the design equation used is

$$Q(z) = T_2^{-1}(z)(T_1(z) - W(z)T_3(z)) \quad (4.21)$$

which is obtained by setting $T_1(z) - T_2(z)Q(z) = W(z)T_3(z)$, where $T_3(z)$ is a user-specified stable biproper transfer function satisfying $\|W(z)T_3(z)\|_\infty < 1$. The final robust predictive controller design results by substituting in the structure (3.11) the Youla parameter (4.20) or (4.21), along with the factorizations (3.5)-(3.6).

4.2 Steady-State Disturbance Rejection

In process control applications the controller is often required to deliver effective disturbance-rejection performance. In order to introduce integral action in the robustified controllers designed in the previous sections, it must be ensured that the parameter $Q(z)$ has a zero gain, *i.e.*, $Q(1)=0$. Details of the robust design technique that guarantees steady-state disturbance rejection are given in [8].

5. EXAMPLE

Consider the unstable second-order nominal plant model

$$g_0(z) = \frac{z + 0.2}{z^2 - 0.6z - 1.12}$$

and the uncertainty weight

$$W(z) = \frac{0.63z + 0.6174}{z + 0.5}$$

Three controllers are designed: (i) a nominal predictive controller (NPC), (ii) a robust predictive controller (RPC), and (iii) a robust predictive controller with integral action (RPCI). The nominal predictive controller is of form (2.5), and is realized using the design parameters $N_y=4$, $N_u=2$, and $\lambda=0$, to arrive at the polynomials

$$R(z) = z^2 - 0.8039z - 0.1961$$

$$S(z) = 0.8639z^2 - 1.579z + 1.0984$$

$$T(z) = 0.2914z^4 - 0.0156z^3 + 0.366z^2 + 0.3243z$$

which leads to a controller that stabilizes the closed loop when the uncertainty is neglected. However, the NPC

controller is not robustly stabilizing because it violates the robust stability condition (4.3), *i.e.* $\|W(z)C(z)S(z)\|_{\infty} = 2.9 > 1$, where $C(z)$ and $S(z)$ are calculated using $Q(z)=0$ in (4.4) and (4.5). This implies that the nominal predictive controller will fail to stabilize the closed loop for some plants belonging to the family of uncertain plants.

The RPC design is of the form (3.11). Since the unstable nominal plant has no poles on the unit circle, the design proceeds as discussed in the base case (Section 4.1.1). The transfer functions $T_1(z)$ and $T_2(z)$ are formed as prescribed in (4.7) and (4.8), using $A_2(z) = z^2$. To solve the Nehari extension problem we use $\gamma = 0.99$, which is acceptable since it exceeds the limiting Hankel norm value $\|R(z)\|_H$. The central controller found using the base-case algorithm leads to a parameter $Q(z) = N_Q(z)/D_Q(z)$ of order 8. The RPC transfer functions $Y(z) - N(z)Q(z)$, and $X(z) + M(z)Q(z)$, are of order 9 in their minimal forms. The controller is robustly stabilizing because $\|W(z)C(z)S(z)\|_{\infty} = 0.35 < 1$.

Finally, the design of the RPCI is carried out using again the specification $\gamma = 0.99$ and $\rho = 1.12$. The procedure leads to $Q(z) = (z-1)N_Q(z)/(zD_Q(z))$ of order 9. The resulting RPCI transfer functions $Y(z) - N(z)Q(z)$, and $X(z) + M(z)Q(z)$ are of order 10 in their minimal forms, and $Q(1) = 0$, as desired. The RPCI controller is robustly stabilizing because $\|W(z)C(z)S(z)\|_{\infty} = 0.49 < 1$.

Figure 3 shows the results of a closed-loop simulation test carried out to evaluate the nominal servo and regulation performances of the three control designs. The process is assumed to match exactly with the nominal model, *i.e.*, $\delta(z) = 0$. A unit-step disturbance $d(t)$ is injected at $t=12$. Figure 3a shows that during the first 12 samples, where $d(t) = 0$, all three controllers display identical dynamics. This is the expected result since Proposition 3.1 guarantees that the nominal predictive controller and the robustified controllers have identical servo transfer functions, independent of the value of $Q(z)$. Also as expected, the control-output trajectories shown in Figure 3b are also identical during this interval.

The three controllers differ however in their regulation behavior. When the disturbance is introduced at $t=12$, the NPC rejects the disturbance, quickly returning the output to the set point, as shown in Figure 3a. In marked contrast, the RPC fails to reject the effect of the disturbance, and displays steady-state offset. The RPCI, on the other hand, succeeds in rejecting the disturbance, albeit with slower dynamics than the nominal controller. Figure 3b shows that the NPC achieves the disturbance rejection at the expense of fairly energetic control actions that follow the onset of the disturbance. On the other hand, the RPCI prescribes more conservative input adjustments, typical of robust controllers. In many practical situations, the smoother dynamics of the RPCI design may be highly preferable over the more aggressive behavior of the NPC.

As a final remark, note that all the controllers anticipate the occurrence of set-point change at $t=50$, as evidenced by the early adjustments in control action that take place starting at instant $t=46$, as shown in Figure 3b. This anticipatory behavior is a characteristic of predictive controllers. Since the prediction horizon N_y has been selected equal to four samples, the controllers naturally initiate adjustments at instant $t=46$ when the prediction horizon first permits detection of the upcoming set-point change. This observation verifies that the robustified control designs can legitimately be classified as predictive controllers, as claimed in Corollary 3.1.

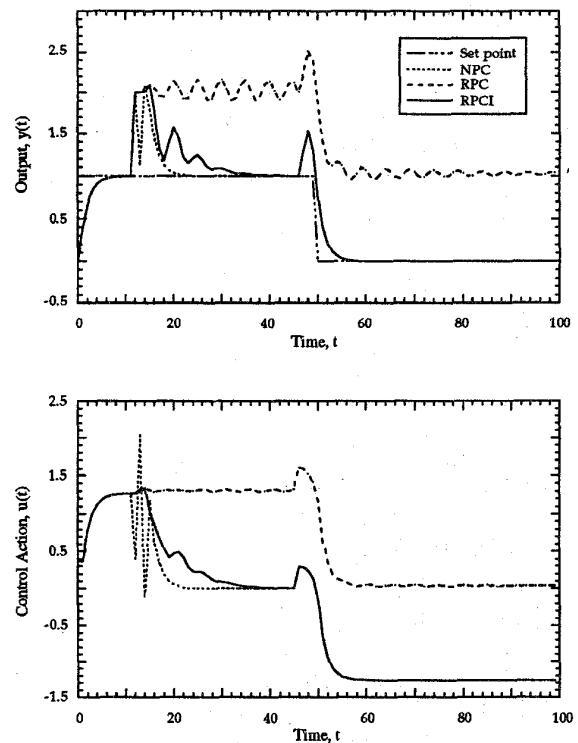


Figure 3. Comparison of the performance of the nominal predictive controller (NPC), the robust predictive controller (RPC), and the robust predictive controller with integral action (RPCI) designed for a plant with no uncertainty. The set point changes at instants $t=0$ and $t=50$. A unit-step disturbance $d(t)$ is introduced at $t=12$.

6. CONCLUSIONS

A systematic method for robustifying predictive controllers has been proposed. The technique succeeds in preserving nominal servo performance due to the unconventional feedback configuration adopted for the parametrized controller, and also to a coprime factorization that makes use of the characteristic polynomial of the nominal closed loop. A significant feature of the proposed method is its applicability to both stable and unstable plants. Another advantage of the robust synthesis technique is that it permits the

incorporation of integral action in the robustified controllers, making the resulting controllers more useful for practical applications.

7. REFERENCES

- [1] Zafiriou, E., "Robust Model Predictive Control of Processes with Hard Constraints", *Computers Chem. Engin.*, 14, (1990).
- [2] Genceli, H., M. Nikolaou, "Robust stability analysis of constrained 11-norm model predictive control", *AIChE Journal*, 39, 12, (1993).
- [3] Zheng, Z. Q., and M. Morari, "Robust stability of constrained model predictive control", *American Control Conference*, San Francisco CA, (1993).
- [4] Robinson, B. D. and D. W. Clarke, "Robustness effects of a prefilter in generalized predictive control", *IEE Proceedings-Part D*, 138, 1, (1991).
- [5] Kouvaritakis, B., Rossiter J. A, and A. Chang, "Stable generalized predictive control: an algorithm with guaranteed stability", *IEE Proceedings*, (July 1992).
- [6] Crisalle, O.D., D. E. Seborg, and D. A. Mellichamp, "Theoretical analysis of long-range predictive controllers", *Am. Control Conf.*, Pittsburgh, (1989).
- [7] Demircioglu, H., and D.W. Clarke, "Generalized predictive control with end-point state weighting", *IEE Proceedings-Part D*, (July 1993).
- [8] Hrissagis, K., O. D. Crisalle, "Robust predictive control design with guaranteed nominal performance", Technical Report ChE-PCL 9401, University of Florida, (1994).
- [9] Francis, B. A., *A Course in H_∞ Control Theory*, Springer Verlag, (1987).
- [10] Rotstein, H., and A. Sideris, "Discrete-time H_∞ control: The one-block case", *Proc. IEEE Conf. Dec. Control*, December (1992).

8. APPENDIX

This section presents specific design equations used to synthesize a nominal predictive controller following the approach in [6]. The design equations for polynomials (2.6)-(2.8) that appear in the predictive control law (2.5) are:

$$R(z) = z^n \left[1 + z^{-1} \sum_{i=1}^{N_y} k_i \Gamma_i(z^{-1}) \right] \quad (\text{A.1})$$

$$S(z) = z^n \left[\sum_{i=1}^{N_y} k_i F_i(z^{-1}) \right] \quad (\text{A.2})$$

$$T(z) = \sum_{i=1}^{N_y} k_i z^i \quad (\text{A.3})$$

where the design operators $F_i(z^{-1})$ and $\Gamma_i(z^{-1})$ and the coefficients k_i , $i=1, 2, \dots, N_y$ are determined from the process model according the following procedure. First rewrite the nominal plant model (2.1)-(2.3) in the equivalent form

$$A_1(z^{-1}) y(z) = z^{-1} B_1(z^{-1}) u(z) \quad (\text{A.4})$$

involving inverse powers of z , where $A_1(z^{-1})$ and $B_1(z^{-1})$ are related to (2.2) and (2.3) in an obvious manner and are

of the form

$$A_1(z^{-1}) = 1 + a_{1,1}z^{-1} + a_{1,2}z^{-2} + \dots + a_{1,n_s}z^{-n_s} \quad (\text{A.5})$$

$$B_1(z^{-1}) = b_{1,0} + b_{1,1}z^{-1} + \dots + b_{1,n_b}z^{-n_b} \quad (\text{A.6})$$

To obtain the design operators $F_i(z^{-1})$, which are polynomials of degree n (the order of the plant (2.1)), solve the set of Diophantine equations

$$E_i(z^{-1})\Delta(z^{-1})A_1(z^{-1}) + z^{-i}F_i(z^{-1}) = 1, \quad i = 1, 2, \dots, N_y \quad (\text{A.7})$$

which also yields the intermediate polynomials $E_i(z^{-1})$ of degree $i-1$. The design operators, $\Gamma_i(z^{-1})$ are of degree n , and are obtained by decomposing the product $E_i(z^{-1})B(z^{-1})$ in the form

$$E_i(z^{-1})B_1(z^{-1}) = G_i(z^{-1}) + z^{-i}\Gamma_i(z^{-1}) \quad (\text{A.8})$$

where the polynomials $G_i(z^{-1})$ of degree $i-1$, known as the *dynamic polynomials*, are characterized by the fact that their coefficients are the sampled values of the step-response of the plant (A.4). In turn, the coefficients of the dynamic polynomials are used to define the nonzero elements of the Toeplitz matrix G_{N_y} known as the *truncated dynamic matrix*, which contains N_u columns, and N_y rows. Finally, the coefficients k_i , $i=1, 2, \dots, N_y$, are obtained as the components of the gain vector $k^T = [k_1 \ k_2 \ \dots \ k_{N_y}]$, which is calculated from the expression

$$k^T = [1 \ 0 \ \dots \ 0] (G_{N_y}^T G_{N_y} + \lambda I)^{-1} G_{N_y}^T \quad (\text{A.9})$$

ACKNOWLEDGMENTS

The numerical calculations for the example made use of H_∞-synthesis software made available by Dr. Hector Rotstein. The first two authors gratefully acknowledge support received from the National Science Foundation under NSF Grant No. CTS-9309659. The third author acknowledges support received from the National Science Foundation under NSF Grant No. ECS- 9211169.