Model-based Controller Design for a Plastic Film Extrusion Process

Sung-ho Hur, Reza Katebi and Andrew Taylor

Abstract— This paper reports the development and implementation of a model-based cross-directional controller for plastic film extrusion and other web-forming processes. The controller design has a similar structure to that of internal model control (IMC) with the addition of an observer whose gain is designed to minimise process and model mis-match. The observer gain is obtained by solving a multi-objective optimisation through the application of a genetic algorithm and simulation results are presented in this paper demonstrating improvements that can be achieved by the proposed controller over two existing CD controllers.

Fig. 1. Generic web forming process

I. INTRODUCTION

As with other web forming processes, such as the papermaking and metal-rolling, plastic film extrusion employs arrays of actuators across a continuously moving web to control the cross-directional (CD) profiles of key product properties. In plastic film, the main controlled property is finished product thickness profile as measured by a scanning gauge downstream from the actuators as illustrated in Fig. 1. The main stages of the plastic film extrusion process are extrusion, where a polymer melt is fed into a die; casting, where the polymer melt is discharged through the die gap onto a rotating casting drum to form a continuous amorphous sheet; machine-directional (MD) stretching; cross-directional (CD) stretching; heat setting; and winding, where finished product is rolled. The array of actuators is located at the die gap and the scanning gauge is placed prior to winding, hence there is a significant time lag between the two as shown in Fig. 1.

CD control has received a considerable attention in the control systems community and there have been many papers published studying various CD controller designs – see [1] and references therein. In this paper, model-based CD controller design is presented and applied to a first-principles model of a plastic film extrusion process. This model is described in [2] and is used to simulate the plant throughout this paper. The proposed controller design has a similar structure to that of the controller reported in [3] since both are modifications to the internal model control (IMC), which can be regarded as a dynamic compensator (Fig. 3). Our controller design requires the solution of a quadratic programming problem online for optimal steady state performance. Model-based CD controllers require an accurate reference model and controller performance can therefore be improved by minimising the effects of process-model mismatch and disturbances. Consequently, the pro-
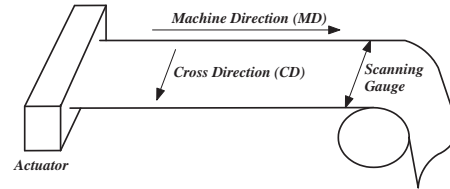
posed controller design employs an observer in place of the reference model in order to reduce the effect of process-model mismatch as well as disturbances.

The observer design is presented in Section II along with the genetic algorithm [4] that solves the resulting multi-objective optimisation problem. The dynamic compensator is discussed in Section III and Section IV illustrates how the steady state performance is achieved by solving a quadratic programming problem [3], [5]. The computational structure of the controller is summarised in Section V, and the implementation and performance of the controller are demonstrated in Section VI followed by a conclusion in Section VII.

II. OBSERVER DESIGN

A model-based controller requires a reference model. One advantage of the proposed controller design is that for this reference model there is no need to separate the dynamic response from the spatial response unlike the controller design reported in [3]. Derivation of a model in such a form requires a system identification process such as the one described in [1]. Instead, the proposed controller design utilises the System Identification Toolbox^{TM}7 in MATLAB®, which is more widely available, to derive a state space model directly from the first-principles model [2] used to simulate the plant throughout this paper. Derivation of the state space model using subspace method included in the System Identification Toolbox^{TM}7 is possible using standard functions and procedures and thus not discussed in this paper. This state space model is used to construct an observer which is then employed as the reference model for the proposed model-based controller design. Another advantage of the proposed controller design is that this observer can be designed to minimise the effects of process-model mismatch and disturbances.
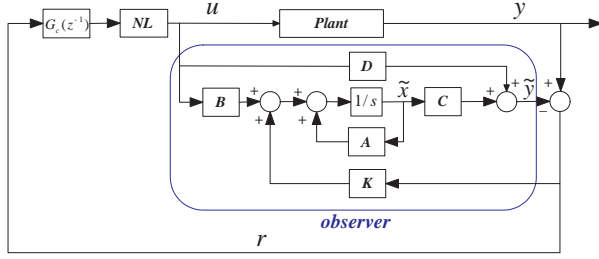
Fig. 2. Proposed controller

The mathematical description of the observer is as follows:

$$\dot{\tilde{\mathbf{x}}}(t) = \mathbf{A}\tilde{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{K}(\mathbf{y}(t) - \tilde{\mathbf{y}}(t))$$
$$= (\mathbf{A} - \mathbf{K}\mathbf{C})\tilde{\mathbf{x}}(t) + (\mathbf{B} - \mathbf{K}\mathbf{D})\mathbf{u}(t) + \mathbf{K}\mathbf{y}(t)$$
$$\tilde{\mathbf{y}}(t) = \mathbf{C}\tilde{\mathbf{x}}(t) + \mathbf{D}\mathbf{u}(t) \tag{1}$$

where $\mathbf{y}(t) \in \mathbb{R}^n$ and $\tilde{\mathbf{y}}(t) \in \mathbb{R}^n$ denote the plant measurements and model estimates, respectively, and $\mathbf{u}(t) \in \mathbb{R}^m$ represents the control action. The term involving the observer gain $\mathbf{K}$ should correct the observer estimate continuously such that $\tilde{\mathbf{y}}(t)$ follows $\mathbf{y}(t)$ more closely. This consequently means that the effects of process-model mismatch and disturbances can be reduced by optimising $\mathbf{K}$. Derivation of an optimal gain $\mathbf{K}$ is summarised in this section. $G_c(z^{-1})$ and NL in Fig. 2 are responsible for dynamic compensation and steady state performance respectively and are discussed in Section III and Section IV.

A. Observer Gain in Frequency Domain

Assuming that the plant can be represented in the state space form, process-model mismatch and disturbances may be described by additional terms $\mathbf{d}_1(t) \in \mathbb{R}^r$ and $\mathbf{d}_2(t) \in \mathbb{R}^n$ as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{d}_1(t)$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{d}_2(t) \tag{2}$$

The vectors $\mathbf{d}_1(t)$ and $\mathbf{d}_2(t)$ can represent any unknown input such as process-model mismatch and disturbances as follows:

$$\mathbf{d}_1(t) = \Delta\mathbf{A}\mathbf{x}(t) + \Delta\mathbf{B}\mathbf{u}(t) + \mathbf{E}_1\tilde{\mathbf{d}}_1(t)$$
$$\mathbf{d}_2(t) = \Delta\mathbf{C}\mathbf{x}(t) + \Delta\mathbf{D}\mathbf{u}(t) + \mathbf{E}_2\tilde{\mathbf{d}}_2(t) \tag{3}$$

where $\mathbf{E}_1$ and $\mathbf{E}_2$ are the distribution matrices and $\tilde{\mathbf{d}}_1(t)$ and $\tilde{\mathbf{d}}_2(t)$ are the disturbance vectors. $\Delta\mathbf{A}$, $\Delta\mathbf{B}$, $\Delta\mathbf{C}$ and $\Delta\mathbf{D}$ are the parameter errors or variations that represent process-model mismatch.

Subtracting $\dot{\tilde{\mathbf{x}}}(t)$ in (1) from $\dot{\mathbf{x}}(t)$ in (2), the equation for the residual $\mathbf{r}(t)$ can be derived as follows: [5]

$$\dot{\mathbf{e}}(t) = (\mathbf{A} - \mathbf{K}\mathbf{C})\mathbf{e}(t) + \mathbf{d}_1(t) - \mathbf{K}\mathbf{d}_2(t)$$
$$\mathbf{r}(t) = \mathbf{C}\mathbf{e}(t) + \mathbf{d}_2(t) \tag{4}$$

where

$$\dot{\mathbf{e}}(t) = \dot{\mathbf{x}}(t) - \dot{\tilde{\mathbf{x}}}(t) \tag{5}$$

The Laplace transform of (4) is thus

$$\mathbf{r}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A} + \mathbf{K}\mathbf{C})^{-1}\mathbf{d}_1(s)$$
$$+ (\mathbf{I} - \mathbf{C}(s\mathbf{I} - \mathbf{A} + \mathbf{K}\mathbf{C})^{-1}\mathbf{K})\mathbf{d}_2(s) \tag{6}$$

Subsequently, the effects of process-model mismatch and disturbances can be minimised by minimising the following performance indices:

$$J_1(\mathbf{K}) = \left\| \mathbf{C}(s\mathbf{I} - \mathbf{A} + \mathbf{K}\mathbf{C})^{-1} \right\|_\infty \tag{7}$$
$$J_2(\mathbf{K}) = \left\| \mathbf{I} - \mathbf{C}(s\mathbf{I} - \mathbf{A} + \mathbf{K}\mathbf{C})^{-1}\mathbf{K} \right\|_\infty \tag{8}$$

where $\|.\|_\infty$ denotes L_∞ norm of a LTI model, which is different from L_∞ norm of a vector known as uniform or maximum norm. By minimising $J_1(\mathbf{K})$ and $J_2(\mathbf{K})$, the maximums of the largest singular values of $\mathbf{C}(s\mathbf{I}-\mathbf{A}+\mathbf{K}\mathbf{C})^{-1}$ and $\mathbf{I}-\mathbf{K}\mathbf{C}(s\mathbf{I}-\mathbf{A}+\mathbf{K}\mathbf{C})^{-1}$, which correspond to the peak gains of the frequency response, are minimised. Hence, the effects of process-model mismatch and disturbances reduce.

If enough information is given to determine $\Delta\mathbf{A}$, $\Delta\mathbf{B}$, $\Delta\mathbf{C}$, $\Delta\mathbf{D}$, $\mathbf{E}_1$ and $\mathbf{E}_2$, these matrices can be incorporated into (6). As a result, the effects of process-model mismatch and disturbances may be minimised even further resulting in improved controller performance. However, minimisation of more performances indices will be required in compensation – more detailed discussion on incorporating theses matrices can be found in [5].

The problem has become finding $\mathbf{K}$ such that $J_1(\mathbf{K})$ and $J_2(\mathbf{K})$ are minimised. However, it is likely that $\mathbf{K}$ causes instability. This can be prevented by parameterising $\mathbf{K}$ via the eigenstructure assignment method [5] summarised in Section II-B. Furthermore, since more than one performance index requires to be minimised, it becomes a multi-objective optimisation problem [4], [5], which may be defined as the process of simultaneously optimising two or more conflicting objectives. The genetic algorithm is a popular approach in multi-objective optimisation and is utilised to solve the multi-objective optimisation problem in Section II-C.

B. Parameterisation via Eigenstructure Assignment Method

When conducting an optimisation to minimise $J_1(\mathbf{K})$ and $J_2(\mathbf{K})$ in (7) and (8), it is important to ensure that the stability of the observer is always guaranteed, and this leads to more complex constrained optimisation problem. To guarantee the stability condition, [5] suggests the use of the eigenstructure assignment method which parameterises $\mathbf{K}$. The method has an advantage of allowing the eigenvalues in predefined regions and is summarised as follows: First it is assumed that the eigenvalues are always real for the sake of brevity. Since the observer design problem is the "dual problem" [5] of the controller design, $\mathbf{v}_i$ is the i_{th} eigenvector of $\mathbf{A}^T - \mathbf{C}^T\mathbf{K}^T$ corresponding to the i_{th} eigenvalue λ_i as follows:

$$(\mathbf{A}^T - \mathbf{C}^T\mathbf{K}^T)\mathbf{v}_i = \lambda_i\mathbf{v}_i \tag{9}$$
$$\mathbf{v}_i = -(\lambda_i\mathbf{I} - \mathbf{A}^T)^{-1}\mathbf{C}^T\mathbf{w}_i \tag{10}$$

where $\mathbf{w}_i = \mathbf{K}^T\mathbf{v}_i$. In turn, we have two design parameters $\mathbf{w}_i$ and λ_i instead of one design parameter $\mathbf{K}$. These design parameters still do not guarantee the stability of the observer.

The eigenvalues λ_i, one of the design parameters, is generally not required to be placed at a specific point in the s or z-plane but rather in a predefined region to satisfy the stability condition. This in turn provides more relaxed design freedom as follows:

$$\lambda_i \in [L_i, U_i] \tag{11}$$

where L_i and U_i ($i = 1, \ldots n$) respectively denote the upper and lower bounds. By defining an equation for the eigenvalue as

$$\lambda_i = L_i + (U_i - L_i sin^2(z_i)) \tag{12}$$

$z_i \in \mathbb{R}$ ($i = 1, \ldots, n$) becomes a design parameter instead of λ_i. Any z_i subsequently guarantees the stability condition.

Finally, the two design parameter vectors $\mathbf{W}$ and $\mathbf{Z}$ have been defined and the performance indices in (7) and (8) can be rewritten as follows:

$$J_1(\mathbf{W}, \mathbf{Z}) = \left\| C(s\mathbf{I} - \mathbf{A} + \mathbf{KC})^{-1} \right\|_\infty \tag{13}$$

$$J_2(\mathbf{W}, \mathbf{Z}) = \left\| \mathbf{I} - \mathbf{C}(s\mathbf{I} - \mathbf{A} + \mathbf{KC})^{-1}\mathbf{K}) \right\|_\infty \tag{14}$$

where

$$\mathbf{K} = [\mathbf{WV}^{-1}]^T \tag{15}$$

Having redefined the multi-objective optimisation problem as finding $\mathbf{Z}$ and $\mathbf{W}$ from finding $\mathbf{K}$ only, the stability condition is always guaranteed. Solving the multi-objective optimisation problem by utilising a genetic algorithm is described in Section II-C.

C. Multi-objective optimisation

Two most common methods for solving multi-objective optimisation problems are construction of a single aggregate objective function (AOF) and utilisation of an evolutionary algorithm. In this paper, the genetic algorithm, which is a type of an evolutionary algorithm, is exploited for multi-objective optimisation. The genetic algorithm is exploited in conjunction with the method of inequalities and the moving boundaries algorithm [6]. [5] applied a similar combination to develop a fault monitoring system for a flight model but it has never been applied for developing a CD control system before.

1) Method of Inequalities: The method of inequalities transforms the problem of the minimisation or maximisation of the performance indices to the problem of the satisfaction of a set of inequalities, That is, the problem becomes searching for a parameter set that satisfies the following inequalities:

$$J_i(\mathbf{W}, \mathbf{Z}) \leq \epsilon_i \tag{16}$$

where ϵ_i ($i = 1, 2$) is the bound on the performance index $J_i(\mathbf{W}, \mathbf{Z})$ chosen by the designer. By restricting or relaxing the bound ϵ_i, the designer can place a different emphasis. If $J_1^*(\mathbf{W}, \mathbf{Z})$ and $J_2^*(\mathbf{W}, \mathbf{Z})$ are the minimum values achieved, the designer should define ϵ_i ($i = 1, 2$) as

$$J_i^*(\mathbf{W}, \mathbf{Z}) \leq \epsilon_i \tag{17}$$

2) Moving-boundaries Algorithm: To help solving the design problem presented above, [6] suggests an algorithm which he calls the moving boundaries algorithm. More detailed description can be found in [5] and [6].

The performance indices are first normalised as follows:

$$\phi_i(\mathbf{W}, \mathbf{Z}) = J_i(\mathbf{W}, \mathbf{Z})/\epsilon_i \tag{18}$$

In turn, the problem becomes satisfying

$$\phi_i(\mathbf{W}, \mathbf{Z}) \leq 1 \tag{19}$$

To solve (19), let $\mathbf{P}_i$ be the set of parameters that satisfies the ith performance index

$$\mathbf{P}_i = \{(\mathbf{W}, \mathbf{Z}) : \phi_i(\mathbf{W}, \mathbf{Z}) \leq 1\} \tag{20}$$

and $\mathbf{P}$ be the set of parameters that satisfies both performance indices

$$\mathbf{P} = \left\{ (\mathbf{W}, \mathbf{Z}) : \max_{i=1}^{2} \{\phi_i(\mathbf{W}, \mathbf{Z}) \leq 1\} \right\} \tag{21}$$

It is now clear that the search for an optimal $\mathbf{P}$ can be achieved by solving the following optimisation problem:

$$\min \left\{ \max_{i=1}^{2} \{\phi_i(\mathbf{W}, \mathbf{Z})\} \right\} \leq 1 \tag{22}$$

In order to solve (22), let $\mathbf{P}^k$ be the set of parameters at step k, and define

$$\mathbf{P}_i^k = \left\{ (\mathbf{W}, \mathbf{Z}) : \phi_i(\mathbf{W}, \mathbf{Z}) \leq \Delta^k \right\} \tag{23}$$

where

$$\Delta^k = \max_{i=1}^{2} \left\{ \phi_i((\mathbf{W}, \mathbf{Z})^k) \right\} \tag{24}$$

Now, let the problem become finding a new parameter set $(\mathbf{W}, \mathbf{Z})$ that reduces the largest performance index Δ^k such that

$$\Delta^{k+1} \leq \Delta^k \tag{25}$$

The optimisation process terminates either when Δ^k is less than 1 or when Δ^k cannot be reduced further. If Δ^k cannot be reduced further and persists being larger than 1, the appropriate bound should be relaxed. The most difficult part of this process is the provision of a trial parameter set $\mathbf{P}^{k+1}$ given $\mathbf{P}^k$. To solve this problem, many methods have been suggested since [6] introduced this technique, but especially the incorporation of the genetic algorithm has been proven to be effective and straightforward [4], [5], and thus is a popular choice for multi-objective optimisation problems.

3) Multi-objective optimisation via Genetic Algorithm: This paper assumes that the readers are familiar with the genetic algorithm. Detailed introduction to the genetic algorithm can be found in many books and papers such as [5] and [7].

The multi-objective optimisation procedures that utilise the combination of the method of inequalities, the moving-boundaries algorithm, and the genetic algorithm for satisfying the performance indices are briefly summarised as follows:
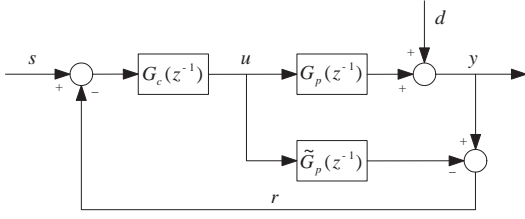
Fig. 3. Generic IMC Design

Step 1: Generation of Initial Population: The population is a row vector of length $(r \times m + r)$. First r elements of the population constitute the row vector $\mathbf{Z}$, and the rest constitute $\mathbf{W} \in \mathbb{R}^{r \times m}$.

Step 2: Evaluation: Using (15), a trial value of $\mathbf{K}$ can be found and the fitness functions (18) are evaluated. In turn, a score is given using (24).

Step 3: Reproduction: Using "ranking", which is one of the available options [7], new children are created.

Step 4: Elitism: A certain percentage of individuals in the current generation with the lowest fitness values are selected as "elites" and are passed on to form the population for the next generation.

Step 5: Recombination: Using "scattering" [7], which is one of the available options, "cross-over" children are created.

Step 6: Mutation: Mutation children are created by randomly changing the genes of parents' individuals.

Step 7: Termination Checking: Step 3 to 6 are repeated until either the stopping criteria, $\Delta^k \leq 1$, has been met or it cannot be minimised further, in which case ϵ_i $(i = 1, 2)$ should be relaxed.

Finally, having designed an observer that minimises the effects of process-model mismatch and disturbances, the observer can now be employed as a reference model necessary for model-based controllers (Figs. 2 and 3). The remaining stages of the controller design follow.

III. DYNAMIC COMPENSATION

Bump test results can be used to approximate the dynamic response of the plant as follows:

$$h(z^{-1}) = \frac{1 - \alpha}{1 - \alpha z^{-1}} z^{-k} \tag{26}$$

Recall that the proposed controller design benefits from not having to separate the spatial component from the dynamic component of the reference model. Although the bump tests provided the dynamic response of the model, the benefit claimed is still valid as the spatial component of the model is still unknown.

The generic IMC design illustrated in Fig. 3 usually designs $G_c(z^{-1})$ as the inverse of the reference model $\tilde{G}_p(z^{-1})$ so that if $\tilde{G}_p(z^{-1})$ is equal to $G_p(z^{-1})$, $y(t)$ is also equal to $s(t)$. However, $\tilde{G}_p(z^{-1})$ is usually non-invertible and should therefore be factorised into invertible and non-invertible components first as follows:

$$\tilde{G}_p(z^{-1}) = \tilde{G}_p^+(z^{-1})\tilde{G}_p^-(z^{-1}) \tag{27}$$

where the invertible component is given by

$$\tilde{G}_p^+(z^{-1}) = \frac{1 - \alpha}{1 - \alpha z^{-1}} \tag{28}$$

Subsequently, $G_c(z^{-1})$ can be designed as the inverse of $\tilde{G}_p^+(z^{-1})$. Furthermore, the effect of process-model mismatch can be minimised to improve robustness. Since mismatches generally occur at the high frequency region of the frequency response, a low-pass filter $G_f(z^{-1})$ is usually added to attenuate the effect of process-model mismatch as follows:

$$G_c(z^{-1}) = [\tilde{G}_p^+(z^{-1})]^{-1} G_f(z^{-1}) \tag{29}$$

where

$$G_f(z^{-1}) = \frac{1 - \beta}{1 - \beta z^{-1}} \tag{30}$$

Substituting (28) into (29), the equation for the controller is

$$G_c(z^{-1}) = \left(\frac{1 - \alpha z^{-1}}{1 - \alpha} \right) \left(\frac{1 - \beta}{1 - \beta z^{-1}} \right) \tag{31}$$

where β is the only tuning parameter for the proposed controller design apart from those required for conducting the genetic algorithm optimisation.

The proposed controller design in Fig. 2 has a similar structure to Fig. 3. Therefore, $G_c(z^{-1})$ in Fig. 2 is also given by (31) but it is further multiplied by the identity matrix of size $n \times n$ since it is a MIMO system.

IV. STEADY STATE PERFORMANCE

Optimal steady state performance can be achieved by minimising $\|\mathbf{y}(t)\|_2 = \|\tilde{\mathbf{y}}(t) + \mathbf{r}(t)\|_2$ where $\tilde{\mathbf{y}}(t)$ is given in (1) and $\|.\|_2$ denotes L_2 norm.

In the steady state, $\dot{\tilde{\mathbf{x}}}(t)$ in (1) is equal to zero and therefore the equation becomes

$$\tilde{\mathbf{x}}(t) = -\mathbf{A}^{-1}\mathbf{B}u(t) - \mathbf{A}^{-1}\mathbf{K}\mathbf{r}(t) \tag{32}$$

Substituting (32) into the following

$$\tilde{\mathbf{y}}(t) = \mathbf{C}\tilde{\mathbf{x}}(t) + \mathbf{D}u(t) \tag{33}$$

and assuming $\mathbf{D}$ is a zero matrix, the following can be derived:

$$\tilde{\mathbf{y}}(t) = -\mathbf{C}\mathbf{A}^{-1}\mathbf{B}u(t) - \mathbf{C}\mathbf{A}^{-1}\mathbf{K}\mathbf{r}(t) \tag{34}$$

Since $\mathbf{y}(t) = \tilde{\mathbf{y}}(t) + \mathbf{r}(t)$

$$\mathbf{y}(t) = -\mathbf{C}\mathbf{A}^{-1}\mathbf{B}u(t) + (\mathbf{I} - \mathbf{C}\mathbf{A}^{-1}\mathbf{K})\mathbf{r}(t) \tag{35}$$

Hence, optimal steady state performance can be attained with

$$\mathbf{u}_{ss} = \arg\min_{\mathbf{u}} \left\| -\mathbf{C}\mathbf{A}^{-1}\mathbf{B}u(t) + (\mathbf{I} - \mathbf{C}\mathbf{A}^{-1}\mathbf{K})\mathbf{r}(t) \right\|_2 \tag{36}$$

where $\mathbf{u}(t)$ is subject to constraints, such as actuator saturation and bending constraints. Namely, the nonlinear element (NL) in Fig. 2 continuously produces the control action using (36). In order to solve the quadratic program required for (36), " fmincon" function provided by the System Identification Toolbox$^{\text{TM}}$7 was utilised.

V. COMPUTATIONAL STRUCTURE OF THE CONTROLLER

The proposed controller design can be summarised as follows (Fig. 2):

1) *Derivation of a reference model*: Using the System Identification Toolbox^{TM}7 in MATLAB®, a reference model in the state space form is derived.
2) *Construction of an observer and derivation of an optimal observer gain*, **K**: In order to find K to minimise the effects of process-model mismatch and disturbances, the genetic algorithm (Section III) is exploited for the multi-objective optimisation problem. Moreover, the stability of the observer is guaranteed by the use of the eigenstructure assignment method summarised in Section II-B.
3) *Dynamic compensation*: For dynamic compensation, the IMC design in employed (Section III). At this stage, β in (31) is determined.
4) *Steady state performance*: Online optimisation is conducted to calculate optimal control action $\mathbf{u}_{ss}$ by continuously solving (36) (Section IV).

VI. SIMULATION AND IMPLEMENTATION

A. Tuning of the genetic algorithm

In order to execute the genetic algorithm, the Genetic Algorithm and Direct Search Toolbox^{TM}2 in MATLAB®was utilised. The parameters were set as follows: population size: 20, number of generations: 100, reproduction method: ranking, elite count: 2 out of 20, cross-over fraction: 14 out of 20, cross-over function: scattering, mutation function: Gaussian, mutation fraction: 4 out of 20.

B. Simulation Results

A number of simulations have been conducted to demonstrate how the controller performs, and three of these simulations are depicted in Figs. 4, 5, 6 and 7. These simulations have been conducted not only for the proposed controller but also for the industrial controller reported in [2] and [8] as well as another model-based controller reported in [3] to provide comparison. Figs. 4, 5 and 6 show the steady state CD thickness profile measured by the scanning gauge and the corresponding actuator set-points. Plots with squares are for the proposed controller and labelled " Proposed" ; plots with circles are for the industrial controller and labelled "Industrial"; and plots with triangles are for the one reported in [3] and labelled "IMC" since it is also based on the IMC design. We assumed that the plastic film was divided into 10 lanes [2]. To simulate what happens in real-life, the edges – the first and last lanes – were not controlled and left open-loop instead. Therefore, the figures do not show the first and last lanes. The y-axes represent thickness in percentage deviation from the mean, and the x-axes denote the CD position.

The reference model used for the industrial and IMC controllers is

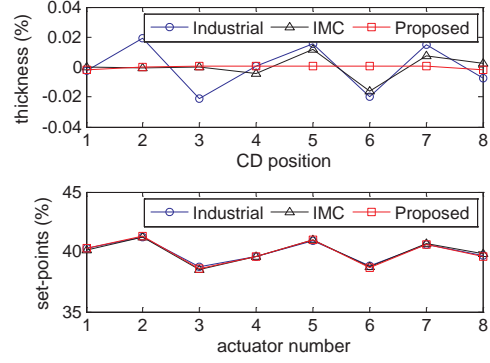$$\tilde{\mathbf{y}}(t) = h(z^{-1})k_p\mathbf{G}\mathbf{u}(t) \qquad (37)$$

Fig. 4. Simulation 1: Steady state CD thickness profile under normal operational conditions

which can be found in [8] and has been obtained using identification experiments similar to those reported in [1]. $h(z^{-1})$ is given in (26), k_p denotes a scalar gain, and G is the interaction matrix representing the steady state spatial response.

For the proposed controller design, the System Identification Toolbox^{TM}7 needs to be utilised to identify a reference model in the state-space form directly from the first-principles model, which simulates the plant. However, (37) has been identified directly from a real plant as opposed to the first-principles model. Although the first-principles model has been developed to simulate the plant, a mismatch between the plant and the first-principles model still exists. The proposed and existing controllers are all tested by application to the first-principles model and it therefore seems unfair for the existing controllers. To be fair, the reference model for the proposed controller has been identified from (37) instead of the first-principles model. If the reference model is identified from the first-principles model, improved performance of the proposed controller can be expected. It is also very important to point out that the industrial controller has been tuned to work optimally with the plant as opposed to the first-principles model, which in turn indicates that improved performance of the industrial controller can also be expected with improved tuning parameters.

Subsequently, all the controllers have been applied to the first-principles model and the simulation results are summarised as follows: In Simulation 1, the proposed controller achieves a significant improvement over the existing controllers under normal operating conditions as shown in Fig. 4. Although no disturbances are present, the set-points are not flat due to the mismatch between the reference model and the first-principles model. Although, the proposed controller design aims to minimise this mismatch, it can not be eliminated completely. Moreover, the reference model for every controller has been identified from (37) instead of the first-principles model leading to even larger model-plant mismatch.

In Simulation 2, there exists persistent (from $t = 0$) variation in fast-roll speed. The speed varies randomly within
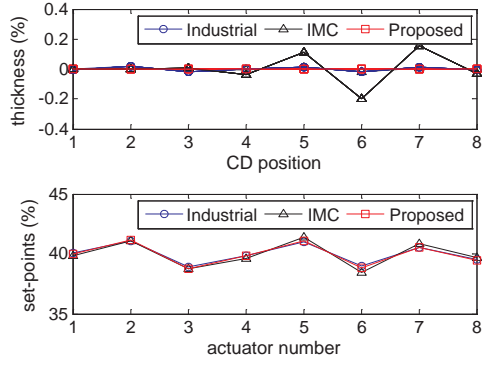
Fig. 5. Simulation 2: Steady state CD thickness profile with fast-roll speed variation from $t = 0s$; y-axis of upper plot has a different range from those in Fig. 4 and Fig. 6
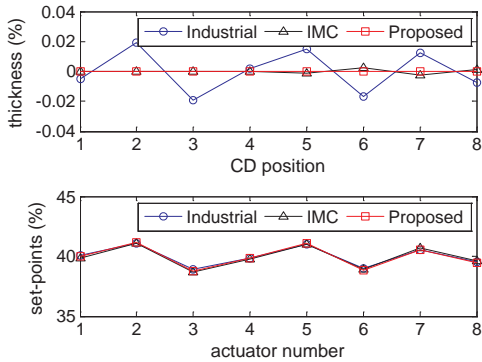

Fig. 6. Simulation 3: Steady state CD thickness profile with mass flow variation from $t = 3000s$

$\pm 10\%$ of the desired speed. The proposed controller achieves a significant improvement over the IMC controller and a slight improvement over the industrial controller.

A polymer melt is fed into the die at a certain mass flow rate [2]. In Simulation 3, mass flow rate starts to vary suddenly from $3000s$ in contrast to Simulation 2, where the variation occurs from $0s$. The purpose of this simulation is to see how the controller responds to a disturbance appearing suddenly. Mass flow rate varies randomly within $\pm 10\%$ of the desired rate. Fig. 6 shows a significant improvement achieved by the proposed controller over the industrial controller but almost none over the IMC controller. However, the dynamic response depicted in Fig. 7 indicates that the proposed controller rejects the distrubance significantly faster than the IMC controller.

Figs. 4, 5 and 6 depict that all the controllers have similar steady state actuator set-points. Despite the similar actuator set-points, the thickness profiles look quite different when they may be expected to be also similar. This is because while the set-points of the industrial controller reach the steady state with no fluctuation at all, the online optimisation, such as (36), used for the model-based controllers (i.e., proposed and IMC controllers) keeps updating the set-points,
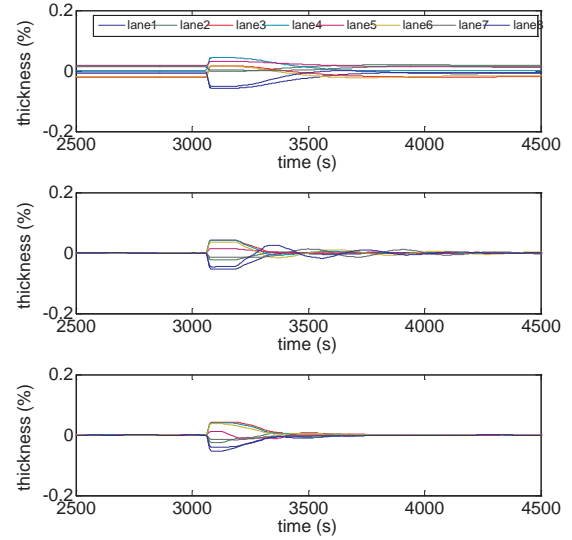

Fig. 7. Simulation 3: Plots from top to bottom: Dynamic response of industrial controller, IMC controller and proposed controller

thereby keeping improving the thickness profiles even further with small but continuous changes to the set-points even at the steady state as shown in Fig. 7.

VII. CONCLUSION

This paper has reported on the development of a new model-based CD controller design and its application to a first-principles model. Simulation results, which provided comparison between the proposed controller and other existing controllers under different disturbance scenarios, demonstrated improvements the proposed controller could possibly achieve over two other existing controllers.

REFERENCES

[1] A. Featherstone, J. VanAntwerp, and R. Braatz, *Identification and control of sheet and film processes*, ser. Advances in Industrial Control. London: Springer, 2000.

[2] S. Hur, J. Balderud, R. Katebi, and A. Taylor, "Cross directional control of a film manufacturing process using a first principle model," in *Proceedings of the European Control Conference 2009, Budapest, Hungary*, 2009, pp. 4475–4480.

[3] W. P. Heath and A. G. Wills, "Design of cross-directional controllers with optimal steady state performance," Centre for Integrated Dynamics And Control (CIDAC), University of Newcastle, NSW, 2380, Australia, Tech. Rep., 2002.

[4] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering and System Safety*, vol. 91, pp. 992–1007, 2006.

[5] J. Chen and R. J. Patton, *Robust model-based fault diagnosis for dynamic systems*, K.-Y. Cai, Ed. Kluwer Academic Publishers, 1999.

[6] V. Zakian and U. Al-Naib, "Design of dynamical and control systems by the method of inequalities," *IEE Control Theory & Application*, vol. 120(11), pp. 1421–1427, 1973.

[7] J. F. Frenzel, "Genetic algorithms," *IEEE Potentials*, vol. 12(3), pp. 21–24, 1993.

[8] A. Taylor and S. Duncan, "Actuator mapping and stability in real-life cross-directional control systems," in *Control Systems 2006, Tampere, Finland*, 2006, pp. 191–196.