Real-time Economic Optimization for an Integrated Plant via a Dynamic Optimization Scheme

Thidarat Tosukhowong and Jay H. Lee
School of Chemical and Biomolecular Engineering
Georgia Institute of Technology, GA 30332-0100

***Abstract*— Conventional real-time optimizers are based on steady-state models and their effectiveness on plants with long-lived dynamics are thus limited. This is particularly true for tightly integrated plants with material recycle loops and other mass / energy integration loops, which tend to show distinct time-scale separation in their dynamic behavior. The use of steady state model limits the execution frequency of the RTO and precludes the utilization of dynamic degrees of freedom, ultimately leading to suboptimal results. Researchers have suggested to combine unit-level controls and plant-wide economic optimization into a single dynamic optimization but the demand for modeling accuracy and computation may be too high for such an approach to be feasible in practice. In this paper, we propose a two-layer architecture for dynamic plant-wide optimization and control, in which the upper layer performs a dynamic optimization of the integrated plant to determine economically optimal setpoints for the lower layer performing control functions at the unit level. To alleviate the unrealistic modeling and computational requirements, we propose the plant-wide dynamic optimization at a rate significantly lower than those of the controllers. Slow-scale plant-wide models are less "stiff" and therefore thought to be more robust to model errors. We discuss how to obtain a "slow"-scale plant-wide model for a chosen optimization frequency and the interfacing of the slow-running plant-wide dynamic optimizer with the fast running unit controllers. An example is given to compare the various approaches.**

I. INTRODUCTION

With the increasing need for improving process economics, efficiency, and quality in the globalized market environment, real-time optimization (RTO) has attracted the attention of the process industry and has been adopted sporadically [1]. The RTO system is model-based and implemented on top of unit-based multivariable controllers. The objective is to maintain the plant operation near an economic optimum in the face of disturbances and other external and internal changes. This RTO layer usually functions between the production planning/scheduling layer and the local unit-based control layer. A typical RTO strategy is based on a steady-state model of the plant and calculates setpoints for the multivariable controllers of various plant units, which steer the operating conditions of their respective units to the calculated optimal values. A general structure of the chemical plant with a steady-state RTO scheme is illustrated in Fig. I

As the domain of RTO often spans an entire plant, it can be computationally demanding to perform the optimization at a rate same as the local unit controllers. More importantly, integrated plants tend to involve dynamics of very different

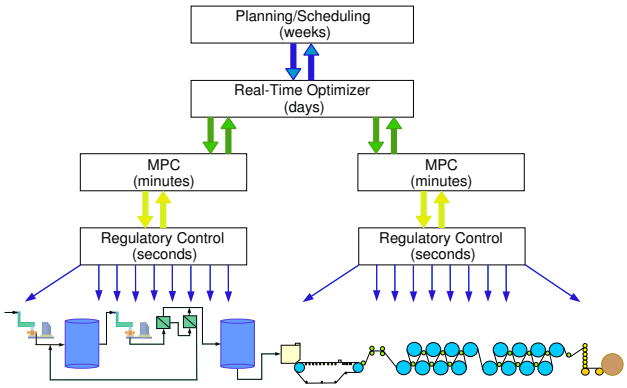

Fig. 1. Typical automation system in a chemical plant

time scales (with slow modes dominating) and therefore be extremely stiff (i.e., ill-conditioned), which can pose numerical difficulties and robustness problems. These are perhaps the reasons for the popularity of the use of a steady state model in RTO. However, for most integrated plants with long-lived dynamics, the approach can be extremely limited. In order for the assumption of steady state hold, the frequency of RTO would have to be limited to, say, once every several days. In fact, changes will invariably occur in the mean time and the plant is unlikely to reach the intended steady state ever. If the steady-state RTO is executed more frequently, however, the mismatch between the steady state model and the transient dynamics can make the solution infeasible for implementation at the local unit levels. Due to these problems, it has been observed that the on-line time of the steady-state RTO often drops with time and it even gets turned off completely [10].

Several researchers suggested to use a dynamic model based RTO executed at the same frequency as the local controllers (e.g. MPC). The one-layer solution, which combines the economic and control objectives into a single dynamic optimization, has been proposed [9], [3], [11]. Extensive simulations are necessary to determine appropriate weights to be assigned to the economic objective and the control objective terms. This strategy was applied to the FCC converter used in the liquefied petroleum gas (LPG) production process involving 4 CVs and 4 MVs [3], [11]. The economic objective function they used was a nonlinear one. The optimization was solved by the sequential

quadratic programming (SQP) method and the performance was compared with that of the typical two-layer steady-state RTO approach. The results showed that the dynamic optimization indeed responded to changes faster, but as the economic and control objectives were mixed into a single term, the economic performance can unduly suffer when the process experienced large disturbances. In addition, such an approach is unlikely to be extendable to a large-scale integrated plant involving many units and recycle loops, as models encompassing dynamics of various time scales (dominant slow modes and many fast modes) would be very "stiff" and the optimization result would be extremely sensitive to model errors. Because of these reasons, we will focus on the two-layer optimization scheme, as shown in Fig. I, in this paper.

One approach designed to enhance the execution frequency of the RTO layer within the two layer structure is the single-point dynamic optimization scheme with a cross-functional coordination layer, which was proposed by Lu [6]. In this approach, RTO is performed at some predicted future point (called 'optimization point'), which is not necessary the steady state point. The execution rate of the RTO is the same as the MPCs. The cross-functional coordination layer is used to coordinate the two layers to ensure the feasibility of the setpoints sent to the MPC. Since the optimization is performed at a single time point based on a fast rate model, the performance of this scheme is highly sensitive to the choice of the 'optimization point', and we can expect that this approach too will be sensitive to plant/model mismatches in the high frequency range. It is generally very difficult to obtain 'bridge dynamics', which connect the units together, accurate up to the execution frequency of MPC controllers.

Considering this, we propose a logical middle ground where RTO based on a reduced-order "slow-scale" dynamic model of the plant is performed at a rate significantly lower than the MPC controllers in order to calculate optimal setpoints for the individual units. The slower rate should make the modeling task more tractable since the slower rate model, which retains the dominant slow modes only, should be better conditioned and more easily identifiable. Hence, we can expect that the optimization result based on this model would be more robust. In addition, computational burden should be less given the slower execution rate and the reduced order nature of the model. Finally, such a middle ground is entirely reasonable from a practical viewpoint, as most changes relevant to plant economics are low-frequency in nature.

II. DYNAMIC OPTIMIZATION USING LOW FREQUENCY MODEL

A. Model Construction

First, the frequency of the plant-wide optimization must be decided based on various factors, such as the accuracy of the plant-wide dynamic model that can be obtained, the bandwidth of external and internal changes relevant to plant economics and interaction, and computational feasibility. Once the frequency of the RTO is decided, one must develop a plant-wide model valid up to the chosen frequency. This may be done using a fundamental model or using system identification. In the former case, one typically gets a very large set of stiff DAEs. Removing the ill-conditioning ("stiffness") using the singular perturbation approach has been discussed in the literature [4] but the procedure can be extremely complicated for a large-scale nonlinear system where the state variables are not explicitly separable in terms of time scale. On the other hand, one can conceivably use numerical approaches such as the Proper Orthogonal Decomposition (POD) method coupled with residualization to identify the slow-scale model from the simulation data.

A more likely scenario in practice is to use the linear models used in the local MPCs and connect them up with 'bridge' dynamics to obtain a plant-wide model. As before, we can expect the accuracy of the bridge dynamics to be poor in the high frequency range, and therefore it must be reduced by eliminating the uncertain high frequency parts of the dynamics while retaining the dominant slow-scale dynamics. The linear models combined through bridge dynamics can be reduced using various linear model reduction methods including the frequency-weighted model reduction (FWMR)[2], which minimizes

$$\|W_o(G - G_r)W_i\|_\infty \tag{1}$$

where G is the original model, G_r is the reduced model, W_o and W_i are output and input weighting matrices, respectively. These weighting matrices make the approximation more accurate in certain ranges where W_o and W_i have larger singular values. Details of the procedure to obtain a FWMR model can be found in [12].

B. Proposed Plant-wide Automation Architecture

Once the reduced order model capturing the dominant slow-scale plant-wide dynamics is obtained, it can be used to solve the following dynamic optimization problem at the chosen optimization interval of T_{opt}:

$$\max_{\mathcal{Y}_g, \mathcal{U}_g} f_{eco} \tag{2}$$

$$\mathcal{Y}_{\min} \le \mathcal{Y}_g(k+1|k) \le \mathcal{Y}_{\max}$$
$$\mathcal{U}_{\min} \le \mathcal{U}_g(k) \le \mathcal{U}_{\max}$$

where

$$\mathcal{U}_g(k) = [u_g(k), \ldots, u_g(k+M-1)]^T,$$
$$\mathcal{Y}_g(k+1|k) = [y_g(k+1|k), \ldots, y_g(k+P|k)]^T.$$

$\mathcal{U}_g$ denotes a vector of manipulated variables with the horizon of M. $\mathcal{Y}_g$ is a vector of predicted outputs with the prediction horizon of P. This output prediction is calculated by integrating the plant model. $\mathcal{Y}_{min}$ and $\mathcal{Y}_{max}$ are the lower and the upper bounds of $\mathcal{Y}_g$, whereas $\mathcal{U}_{min}$ and $\mathcal{U}_{max}$ are the lower and the upper bounds for $\mathcal{U}_g$. Note that the above represents a receding horizon algorithm, meaning that

the same optimization is solved again at the next sample time with an updated prediction.

As the global solution from the RTO layer may not be feasible for the local controller, which does not account for interactions from the other units, we adopt the use of the following least-square coordination collar proposed by Lu (2001) [6] to find a locally feasible point closest to the global solution:

$$\min_{\mathbf{u}^j_{ls}(k)} \sum \left(\mathbf{u}^j_{ls}(k) - \mathbf{u}^j_g(k)\right)^2 \qquad (3)$$
$$\mathbf{y}^j_{\min} \le \mathbf{y}^j_{ls} \le \mathbf{y}^j_{\max}$$
$$\mathbf{u}^j_{\min} \le \mathbf{u}^j_{ls}(k) \le \mathbf{u}^j_{\max}$$

where

$$\mathbf{y}^j_{ls} = \mathbf{G}^j_x \mathbf{x}^j(k) + \mathbf{G}^j_u \mathbf{u}^j_{ls}(k) + \mathbf{G}^j_d \mathbf{d}^j(k), . \qquad (4)$$

The superscript j denotes the index of the local unit. $\mathbf{u}^j_g$ is the setpoint passed from the optimizer to the unit j. $\mathbf{u}^j_{ls}$ and $\mathbf{y}^j_{ls}$ are vectors of MVs and CVs computed by the least-square coordination collar. As this coordination layer checks the feasibility of the setpoint at the end of MPC prediction horizon, the gain matrices in Eq. (4) are for the end of MPC prediction horizon. Note that the above finds the feasible MV values that are closest to the optimal values in the least-square sense, while respecting the key constraints at the local level. Another option is to formulate the least-square problem to find feasible CV values closest to their globally optimal values. The proper choice of transfer option can be different according to the plant-wide objective.

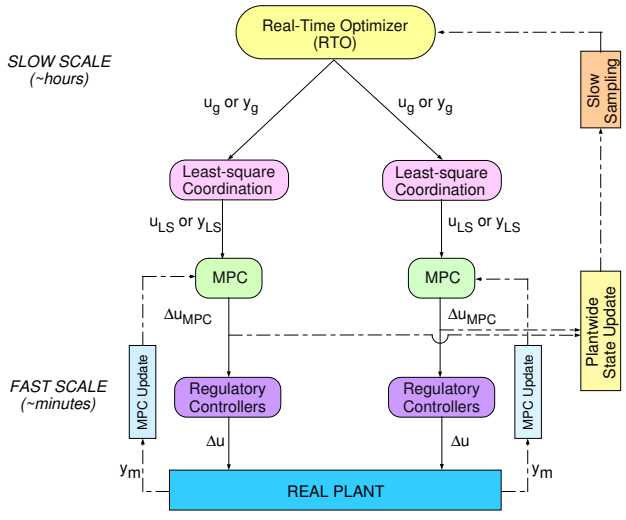

Fig. 2. Schematic representation of plant-wide dynamic optimization strategy.

Once the setpoints have been determined for each MPC, MPC calculates a profile for manipulated variables such that the errors between the future output and the given setpoint trajectories are minimum. This is repeated at each sample time of the MPC until the next execution point for the RTO is reached and new setpoints are passed down from the RTO layer. In the mean time, the state vector of the plant-wide optimizer is updated at the same sample rate as the MPC controllers, that is, every time local controller actions and disturbances are calculated, with some appropriate filtering. The architecture of the proposed scheme is illustrated in Fig. 2.

III. ILLUSTRATIVE EXAMPLES

A. Problem Description

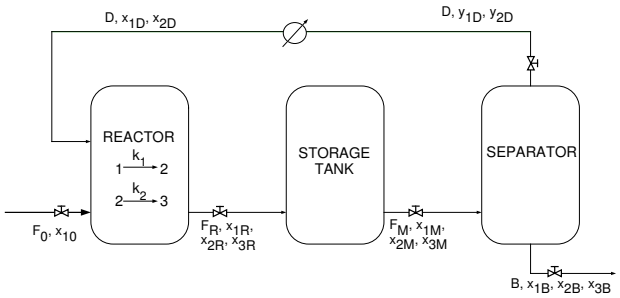

Fig. 3. Schematic of the integrated plant with reactor, storage tank, flash tank connected via a recycle stream

In this section we present an example involving a reactor, a storage tank, and a separation unit connected via a large recycle stream. The large recycle ratio introduces a time scale separation, which complicates modeling and optimization. The material balance of this system can be described by the following ordinary differential equations:

$$\begin{aligned}
\dot{H}_R &= \tfrac{1}{\rho A_R}(F_0 + D - F_R)\\
\dot{x_{1R}} &= \tfrac{1}{\rho A_R H_R - k_1 x_{1R}}[F_0(x_{10} - x_{1R}) + D(x_{1D} - x_{1R})]\\
\dot{x_{2R}} &= \tfrac{1}{\rho A_R H_R}[-F_0 x_{2R} + D(x_{2D} - x_{2R})] + k_1 x_{1R} - k_2 x_{2R}\\
\dot{x_{3R}} &= \tfrac{1}{\rho A_R H_R}[-(F_0 + D)x_{3R}] + k_2 x_{2R}\\
\dot{H}_M &= \tfrac{1}{\rho A_M}(F_R - F_M)\\
\dot{x_{1M}} &= \tfrac{F_R}{\rho A_M H_M}(x_{1R} - x_{1M})\\
\dot{x_{2M}} &= \tfrac{F_R}{\rho A_M H_M}(x_{2R} - x_{2M})\\
\dot{x_{3M}} &= \tfrac{F_R}{\rho A_M H_M}(x_{3R} - x_{3M})\\
\dot{H}_B &= \tfrac{1}{\rho A_B}(F_M - B - D)\\
\dot{x_{1B}} &= \tfrac{1}{\rho A_B H_B}[F_M(x_{1M} - x_{1B}) - D(x_{1D} - x_{1B})]\\
\dot{x_{2B}} &= \tfrac{1}{\rho A_B H_B}[F_M(x_{2M} - x_{2B}) - D(x_{2D} - x_{2B})]\\
\dot{x_{3B}} &= \tfrac{1}{\rho A_B H_B}[F_M(x_{3M} - x_{3B}) + D x_{3B}]
\end{aligned}$$

Here, H_R, H_M, and H_B denote the liquid heights in the reactor, the storage tank, and the separator, respectively. Flowrates for feed, reactor outflow, storage tank outflow, recycle, and product stream are denoted by F_0, F_R, F_M, D, and B, respectively. x_{ij} denotes the molar liquid fraction of component i ($i = 1, 2, 3$) in the stream j. As the liquid level in each tank behaves as integrators, some of these flowrates must be used to stabilize the levels. According to Richardson's rule [8], the largest stream should be selected to control the liquid level in a vessel. However, if we select F_R, F_M, and D to control the levels of the reactor,

TABLE I

NOMINAL VALUES FOR THE PROCESS AND OPERATING PARAMETERS

Parameters		Value
Liquid density	$\rho = 1$	
Volatility	$\alpha_1 = 90$	$\alpha_B = 1$
Rate constant	$k_1 = 0.0167$	$k_2 = 0.0167$
Vessel area	$A_R = 5$	$A_M = 10$
		$A_B = 5$
Vessel holdup	$H_R = 20$	$H_M = 20$
		$H_B = 20$
Flowrate, hr^{-1}	$F_0 = 1.667$	$F_R = 31.33$
	$F_M = 31.33$	$B = 1.667$
	$D = 29.67$	
Mole fraction	$x_{10} = 1.00$	$x_{20} = 0$
	$x_{30} = 0$	$x_{1R} = 0.8861$
	$x_{2R} = 0.1082$	$x_{3R} = 0.0058$
	$x_{1M} = 0.8861$	$x_{2M} = 0.1082$
	$x_{3M} = 0.0058$	$x_{1B} = 0.1139$
	$x_{2B} = 0.7779$	$x_{3B} = 0.1082$
	$x_{1D} = 0.9295$	$x_{2D} = 0.0705$
Controller gains	$K_{C,R} = -10$	$K_{C,M} = -10$
	$K_{C,B} = -5$	

TABLE II

OUTPUT AND MANIPULATED VARIABLES OF UNIT 1 AND UNIT 2

Unit 1			Unit 2		
output	variables	range	output	variables	range
1	x_{1R}	[0,1]	1	x_{1B}	[0,0.15]
2	x_{2R}	[0,0.15]	2	x_{2B}	[0.75,1]
3	x_{3R}	[0,0.02]	3	x_{3B}	[0,0.15]
4	x_{1M}	[0,1]	4	B	[0.67,3]
5	x_{2M}	[0,0.15]			
6	x_{3M}	[0,0.02]			
7	F_R	[8,47]			
8	F_M	[8,47]			
MV	variables	range	MV	variables	range
1	H_R	[10, 30]	1	H_B	[10, 30]
2	H_M	[10, 30]	2	D	[8, 45]

the storage tank, and the flash tank, respectively, the three levels are not independently controllable as the MVs are all internal flow variables and are not independent. Instead, we used F_R, F_M, and B to stabilize the levels through P-only controllers. Although these flows are no longer directly available as manipulated variables for the RTO, the degrees of freedom remain the same as the level setpoint of each vessel can be used as a MV. The nominal values of the process and operating parameters are given in Table I.

The underlying process was divided into two process units with an MPC for each: Unit 1 consists of the reactor and the intermediate tank, whereas Unit 2 includes the separator. Each MPC has two manipulated variables, i.e. setpoints of the reactor level and the storage tank level for MPC 1, and the setpoint of the separator level and the recycle flow for MPC 2. Table II provides a list of output and manipulated variables of each unit as well as their constraints.

To keep our strategy easy to implement computationally, the linearized model was used in the MPCs. Sample time of 6 minutes was used in both MPCs. As the transient dynamics last as long as 12 hours once a move is made, we used the observer-based model predictive control formulation for integrating dynamics discussed in [5], [7], which allows the step responses to be truncated well before the responses settle. Since the objective of each MPC controller is to minimize the error between the CV setpoints and future CV values, the quadratic objective function was employed. The parameters for the both MPC controllers are given in Table III, where t_{trnc} is the truncation time of the step-response model, p is the prediction horizon, m is the control horizon, Δu_{max} is the rate constraint of the input variables, Γ^y and Γ^u are the output and input weighting matrices, respectively. Note that in the MPC optimization, the output constraints were implemented as soft constraints to avoid the possibility of infeasibility.

TABLE III

PARAMETERS FOR LOCAL MPCS

Parameters	MPC 1	MPC 2
t_{trnc}	8 hrs.	8 hrs.
p	40	40
m	10	10
Δu_{max}	[0.3; 0.3]	[0.04; 0.5]
Γ^y	$\Gamma^y_{ii} = \begin{cases} 3 & \text{for } x_{1R}, \\ 7 & \text{for } x_{2R}, \\ 0 & \text{else} \end{cases}$	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$
Γ^u	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}$

To decide on the execution frequency of the RTO, we considered the eigenvalues of the system around the steady-state, which are -0.0097, -0.0167, -0.0167, -0.1567, -0.1567, -0.3133, -0.3279, -0.3458, -0.3458, -1, -1, -2. This suggests a clear separation of between the slow and fast modes at the frequency between 0.0167-0.1567 rad/min, which corresponds to the time period of 40-377 mins. Therefore, we chose to execute the RTO at the frequency of once every 60 minutes, and the FWMR method was used to obtained the slow-scale model accurate up to this optimization frequency. The optimization horizon (P) and the MV horizon (M) were chosen as 8 and 2, respectively. We also imposed the same soft output and hard input constraints as in the MPC layer. The test scenario was to increase the production throughput by 20 %. The optimization was a quadratic program (QP) to minimize the difference between this production demand and the actual product flow (B). We compared three different plant-wide optimization strategies as described below.

- *Dynamic RTO with a Slow-scale Model*: The dynamic RTO is performed at every T_{opt} time interval. The state of the dynamic optimizer is updated at each sampling time of MPC using the input and output measurements.
- *Single-point Dynamic RTO Scheme*: The objective function uses dynamic gains at the optimizing point, which is chosen to be the end of prediction horizon of the local controllers (4 hours). The optimization is

performed at the same frequency as the local MPC controllers.

- *Steady-state RTO Scheme*: The objective function of the RTO is formulated using only the steady-state gains, and the execution rate is once every 10 hours.

Note that in the first two schemes, setpoints of the MVs from the RTO are transferred to the coordination collars for the feasibility checks as in Eq. (3). Then the output trajectories were calculated and passed to the local MPCs.

B. Simulation Result

The simulation results are shown in Figs. 4-6, which include the response of the selected CVs, i.e., the production rate (B), the concentration of component 2 (x_{2B}), and the selected MVs, i.e., feed flowrate (F_0), and recycle flowrate (D). For the CV plots, the solid lines represent the output measured from the nonlinear plant, the dash lines are the output prediction from the MPCs, and the dash-dot lines represent the setpoint given to the MPC.

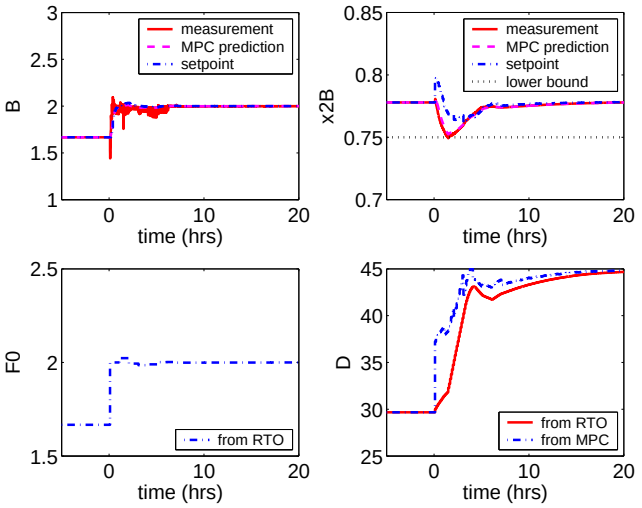

Fig. 4. Simulation results from the slow-scale dynamic RTO: (*top*) CVs and (*bottom*) MVs.

From the simulation results, the slow-scale dynamic RTO showed superior performance over the other schemes. This is due to fact that this scheme allows multiple prediction and control horizons. So the MVs were moved in a more gradual manner so as to prevent large process interactions among the process units as shown in Fig. 4. Note that the reason why the actual change of the product flow B (solid line) was slightly jumpier than the MPC prediction is because B was used to control the holdup in the flash tank and it was a small flow compared to the recycle, the reactor and the storage outflows. So during the transient period, the percentage change in this product stream can be large compared to the other streams.

In contrast, the single-point dynamic RTO and the steady-state RTO optimize the plant based on a prediction at a single point, and ignore the rest of the dynamic information.

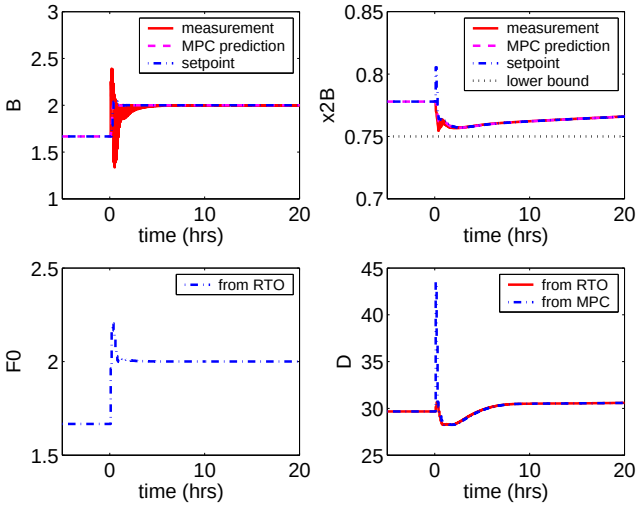

Fig. 5. Simulation results from the single-point dynamic RTO: (*top*) CVs and (*bottom*) MVs.

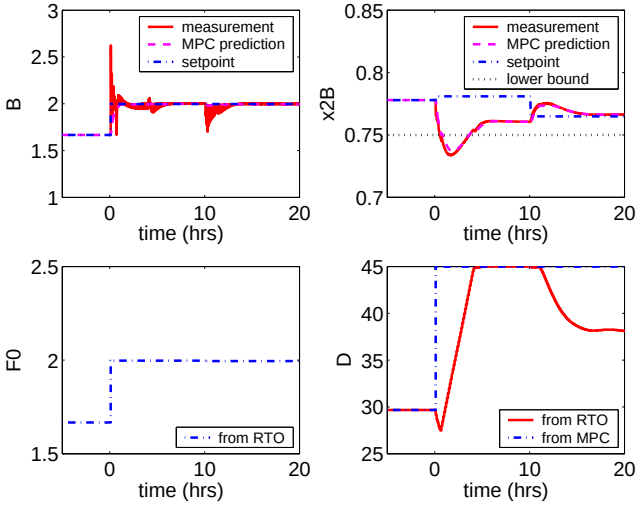

Fig. 6. Simulation results from the steady-state RTO: (*top*) CVs and (*bottom*) MVs.

Therefore, it is not surprising that both RTO schemes made more aggressive setpoint changes in the MVs as shown in Figs. 5-6, and the product flow B in both cases were affected significantly. In fact, one can possibly detune the P-controllers to make the change in B smoother when using the steady-state and the single-point RTO schemes. We did not investigate this option, since the detuning of the level loops translates into a larger MPC sample time than the pre-specified value of 6 minutes. Furthermore, in the steady-state RTO case we also observed the product concentration x_{2B} violated the lower bound constraints for a short period of time shortly after the RTO was turned on. This undesirable behavior happened due to the mismatch between the steady-state prediction and the transient response of the system.

IV. CONCLUSION

Dynamic optimization using a reduced order model capturing dominant slow modes of an integrated plant can provide a computationally efficient and robust solution that responds efficiently to external and internal changes. The key to make the proposed method practicable is a systematic method that enables the user to develop a reduced order model that accurately represents dominant slow dynamic modes and therefore is well conditioned. This coupled with the slow execution frequency makes the on-line computation feasible. The proposed architecture also emphasizes a better coordination between the RTO layer and the control layer. The use of a coordination collar is recommended to ensure the feasibility of the globally optimal setpoints within each local controller. If not, new feasible setpoints that are closest to the globally optimal setpoints are calculated and sent to the local controllers. To keep the prediction accurate, the plant-wide model's state vector is updated with filtered feedback errors at each sampling time of MPC. The plant-wide optimizer then uses this state vector for the prediction at the sample point of RTO. The suggested method is a promising alternative to the current steady-state-model-based or other single-point RTO schemes, which can be limited in terms of execution frequency and often lack the necessary robustness to realistic model errors.

REFERENCES

[1] C. R. Cutler, R. T. Perry, Real time optimization with multivariable control is required to maximize profits, *Computers and Chemical Engineering*, vol. 7(5), 1983, pp. 663.

[2] D. F. Enns, Model reduction with balance realization: An error bound and frequency weighted generalization, *Proceedings of the IEEE conference on Decision and Control*, 1984, 127.

[3] M. T. Gouvêa and D. Odloak, One-layer real time optimization of LPG production in the FCC unit: procedure, advantages and disadvantages, *Computers and Chemical Engineering*, vol. 22, 1998, pp. S191-S198.

[4] A. Kumar and P. Daoutidis, Control of nonlinear differential-algebraic-equation systems with applications to chemical processes, *Chapman & Hall/CRC*, vol. 397 of Research Notes in Mathematics Series, 1999.

[5] J. H. Lee and M. Morari and C. E. Garcia, State-space interpretation of model predictive control, *Automatica*, vol. 30(4), 1994, pp. 707-717.

[6] J. Z. Lu, Challenging control problems and emerging technologies in enterprise optimization, *Pre-Prints of 6th IFAC Symposium on Dynamics and Control of Process Systems*, 2001, pp. 29.

[7] P. Lundström and J. H. Lee and M. Morari and S. Skogestad, Limitations of dynamic matrix control, *Computers and Chemical Engineering*, vol. 19(4), 1995.

[8] W. L. Luyben, B. D. Tyréus and L. M. Luyben, *Plantwide Process Control*, McGraw-Hill, New York, NY, 1999.

[9] P. J. Vermeer, C. C. Pederson and W. M. Canney, Blend-control system all but eliminates reblends for Canadian refiner, *The Oil and Gas Journal*, vol. 95(30), 1997, pp. 77.

[10] D. C. White, Online optimization: What have we learned?, *Hydrocarbon Processing*, vol. 77(6), 1998, pp. 55.

[11] A. C. Zanin, M. T. Gouvea and D. Odloak, Industrial implementation of a real-time optimization strategy for maximizing production of LPG in a FCC unit, *Computers and Chemical Engineering*, vol. 24, 2000, pp. 525-531.

[12] K. Zhou and J. C. Doyle and K. Glover, *Robust and Optimal Control*, *Prentice-Hall*, Upper Saddle, River, NJ; 1996.