Distributed MPC for Dynamic Supply Chain Management

William B. Dunbar[1] and S. Desa[2]

[1] Computer Engineering, Baskin School of Engineering, University of California, Santa Cruz, 95064, USA
dunbar@soe.ucsc.edu
[2] Technology and Information Management, Baskin School of Engineering, University of California, Santa Cruz, 95064, USA

Summary. The purpose of this paper is to demonstrate the application of a recently developed theory for distributed nonlinear model predictive control (NMPC) to a promising domain for NMPC: dynamic management of supply chain networks. Recent work by the first author provides a distributed implementation of NMPC for application in large scale systems comprised of cooperative dynamic subsystems. By the implementation, each subsystem optimizes locally for its own policy, and communicates the most recent policy to those subsystems to which it is coupled. Stabilization and feasibility are guaranteed for arbitrary interconnection topologies, provided each subsystem not deviate too far from the previous policy, consistent with traditional MPC move suppression penalties. In this paper, we demonstrate the scalability and performance of the distributed implementation in a supply chain simulation example, where stages in the chain update in parallel and in the presence of cycles in the interconnection network topology. Using anticipative action, the implementation shows improved performance when compared to a nominal management policy that is derived in the supply chain literature and verified by real supply chain data.

1 Introduction

A supply chain can be defined as the interconnection and evolution of a demand network. Example subsystems, referred to as stages, include raw materials, distributors of the raw materials, manufacturers, distributors of the manufactured products, retailers, and customers. Between interconnected stages, there are two types of process flows: 1) information flows, such as an order requesting goods, and 2) material flows, i.e., the actual shipment of goods. Key elements to an efficient supply chain are accurate pinpointing of process flows and timing of supply needs at each stage, both of which enable stages to request items as they are needed, thereby reducing safety stock levels to free space and capital [3]. Recently, Braun *et al.* [2] demonstrated the effectiveness of model predictive control (MPC) in realizing these elements for management of a dynamic semiconductor chain, citing benefits over traditional approaches and robustness to model and demand forecast uncertainties. In this context, the chain is isolated from competition, and so a cooperative approach is appropriate. Limitations of their approach are that it requires acyclic interconnection network topologies,

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 607–615, 2007.
springerlink.com

and sequential updates from downstream to upstream stages. Realistic supply chains contain cycles in the interconnection network, and generally do not operate sequentially, i.e., stages typically update their policies in parallel, often asynchronously. To be effective in the general case, a distributed MPC approach should demonstrate scalability (stages are locally managed), stability, permit parallel updates, and allow for cycles in the interconnection network topology. The purpose of this paper is to demonstrate the application of a recently developed distributed implementation of nonlinear MPC (NMPC) [4, 5] to the problem of dynamic management of supply chain networks. By this implementation, each subsystem optimizes locally for its own policy, and communicates the most recent policy to those subsystems to which it is coupled. Stabilization is guaranteed for arbitrary interconnection topologies (permitting cycles), provided each subsystem not deviate too far from the previous policy. A contribution of this paper is to demonstrate the relevance and efficacy of the distributed NMPC approach in the venue of supply chain management.

2 Problem Description

A supply chain consists of all the stages involved in fulfilling a customer request [3]. A three stage supply chain network consisting of a supplier S, a manufacturer M, and a retailer R is shown in Figure 1, and will be the focus of this paper. Dell employs a "build-to-order" management strategy that is based on a version of the chain in Figure 1, where R is the customer, M is Dell, S is a chip supplier [3]. Each variable shown has a superscript denoting the corresponding stage it is

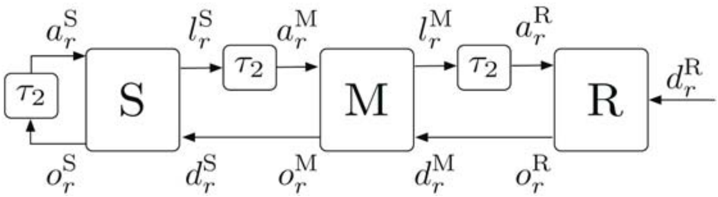

Fig. 1. Block diagram of a three stage supply chain comprised of a supplier S, a manufacturer M, and a retailer R

associated with. The classic MIT "Beer Game" [7] is used as an example three stage supply chain. In the beer game, the supplier S may be thought of as the supplier of bottles to the manufacturer M, who brews and "bottles" the beer, and then ships it to the retailer R for sale to customers. The supply chain is therefore driven by customer demand (number of cases sold per day), which then triggers a series of information flows and material flows. The *information flows* are assumed to have negligible time delays, and are represented by the three left pointing arrows in Figure 1. The *material flows* are assumed to have shipment delays, and are represented by the arrows that pass through blocks labeled τ_2, where τ_2 is a constant representing the amount of delay in days to move the goods. In the case of the supplier, the outgoing information flow (o_r^{S}) is converted through fabrication into materials, and this conversion process is modeled as a simple delay. Since material flows downstream, we say that R is *downstream* from M (likewise, M is downstream from S), while M is *upstream* from R (likewise, S is upstream from M). The customer can be thought of as a stage downstream (not shown) from R in our model.

Each stage $x \in \{\text{S,M,R}\}$ in Figure 1 is characterized by 3 state variables, defined as follows. The stock level s^x is the number of items currently available in stage x for shipment to the downstream stage. The unfulfilled order of stock o_u^x is the number of items that stage x has yet to receive from the upstream stage. The backlog of stock b^x is the number of committed items that stage x has yet to ship to the downstream stage. The exogenous *inputs* (assumed measureable) are the demand rate d_r^x, defined as the number of items per day ordered by the downstream stage, and the acquisition rate a_r^x, defined as the number of items per day acquired from the upstream stage. The *outputs* are the order rate o_r^x, defined as the number of items per day ordered from the upstream stage, and the shipment rate l_r^x, defined as the number of items per day shipped to the downstream stage. The order rate is the *decision variable* (control). By our notation, all rate variables are denoted by an r subscript. The model, state and control constraints for any stage $x \in \{\text{S,M,R}\}$ are

$$\left.\begin{aligned} \dot{s}^x(t) &= a_r^x(t) - l_r^x(t) \\ \dot{o}_u^x(t) &= o_r^x(t) - a_r^x(t) \\ \dot{b}^x(t) &= d_r^x(t) - l_r^x(t) \end{aligned}\right\}, \quad t \geq 0, \tag{1}$$

$$\text{subject to} \quad \left.\begin{aligned} 0 \leq (s^x(t), o_u^x(t), b^x(t)) &\leq \mathrm{s}_{\max} \\ 0 \leq o_r^x(t) &\leq o_{r,\max} \end{aligned}\right\}, \quad t \geq 0, \tag{2}$$

where $l_r^x(t) = d_r^x(t - \tau_1) + b^x(t)/t_b$. The dynamics of the supply chain in the present work arise either from rates of accumulation, or from one of two types of material flow delay (see [7], Chapter 11). Equation (1) describes the first-order dynamics for stock, unfulfilled orders, and backlog, each arising from rates of accumulation. The constraints on the state and control in (2) reflect that stock, unfulfilled order and backlog are independently bounded from below by zero and from above by a common constant $\mathrm{s}_{\max}$, and that the control (order rate) is non-negative and bounded by the positive constant $o_{r,\max}$. The objective of supply chain management is to minimize total costs, which includes avoiding backlog (keep near zero) and keeping unfulfilled orders and stock near desired (typically low) levels [7]. Specifically, the control objective for each stage is $(s^x(t), o_u^x(t)) \rightarrow (s_d, o_{ud}^x(t))$, where s_d is a constant desired stock (common to every stage) and $o_{ud}^x(t) = t_l l_r^x(t)$ is the desired unfulfilled order. The flow constant t_l represents the lead time from the downstream stage. Note that if the demand rate converges to a steady value $d_r^x(t) \rightarrow d_r$, then backlog will converge to zero, the shipment rate converges $l_r^x(t) \rightarrow d_r$, and the desired unfulfilled order becomes the constant $o_{ud}^x = t_l d_r$. For each stage $x \in \{\text{S,M,R}\}$, the acquisition rate $a_r^x(t)$ and the demand rate $d_r^x(t)$ are defined as follows: S: $a_r^{\mathrm{S}}(t) = o_r^{\mathrm{S}}(t-\tau_2)$, $d_r^{\mathrm{S}}(t) = o_r^{\mathrm{M}}(t)$; M: $a_r^{\mathrm{M}}(t) = l_r^{\mathrm{S}}(t-\tau_2)$, $d_r^{\mathrm{M}}(t) = o_r^{\mathrm{R}}(t)$; and R: $a_r^{\mathrm{R}}(t) = l_r^{\mathrm{M}}(t-\tau_2)$. The demand rate at the retailer $d_r^{\mathrm{R}}(t)$ is an input defined as the current/projected customer demand. After substitutions, we have the following models for each of the three stages. For the supplier stage,

$$\left.\begin{aligned} \dot{s}^{\mathrm{S}}(t) &= o_r^{\mathrm{S}}(t-\tau_2) - o_r^{\mathrm{M}}(t-\tau_1) - b^{\mathrm{S}}(t)/t_b \\ \dot{o}_u^{\mathrm{S}}(t) &= o_r^{\mathrm{S}}(t) - o_r^{\mathrm{S}}(t-\tau_2) \\ \dot{b}^{\mathrm{S}}(t) &= o_r^{\mathrm{M}}(t) - o_r^{\mathrm{M}}(t-\tau_1) - b^{\mathrm{S}}(t)/t_b \end{aligned}\right\} . \tag{3}$$

For the manufacturer stage,

$$\left.\begin{aligned} \dot{s}^{\mathrm{M}}(t) &= o_r^{\mathrm{M}}(t-\tau_1-\tau_2) + b^{\mathrm{S}}(t-\tau_2)/t_b - o_r^{\mathrm{R}}(t-\tau_1) - b^{\mathrm{M}}(t)/t_b \\ \dot{o}_u^{\mathrm{M}}(t) &= o_r^{\mathrm{M}}(t) - o_r^{\mathrm{M}}(t-\tau_1-\tau_2) - b^{\mathrm{S}}(t-\tau_2) \\ \dot{b}^{\mathrm{M}}(t) &= o_r^{\mathrm{R}}(t) - o_r^{\mathrm{R}}(t-\tau_1) - b^{\mathrm{M}}(t)/t_b \end{aligned}\right\} . \tag{4}$$

For the retailer stage,

$$\left.\begin{aligned} \dot{s}^{\mathrm{R}}(t) &= o_r^{\mathrm{R}}(t-\tau_1-\tau_2) + b^{\mathrm{M}}(t-\tau_2)/t_b - d_r^{\mathrm{R}}(t-\tau_1) - b^{\mathrm{R}}(t)/t_b \\ \dot{o}_u^{\mathrm{R}}(t) &= o_r^{\mathrm{R}}(t) - o_r^{\mathrm{R}}(t-\tau_1-\tau_2) - b^{\mathrm{M}}(t-\tau_2)/t_b \\ \dot{b}^{\mathrm{R}}(t) &= d_r^{\mathrm{R}}(t) - d_r^{\mathrm{R}}(t-\tau_1) - b^{\mathrm{R}}(t)/t_b \end{aligned}\right\} . \tag{5}$$

We say that two stages have bidirectional coupling if the differential equation models of both stages depend upon the state and/or input of the other stage. Equations (3)–(5) demonstrate the dynamic bidirectional coupling between stages S and M, and stages M and R. Due to the bidirectional coupling, there are two cycles of information dependence present in this chain. Cycle one: the model (3) for S requires the order rate o_r^{M} from M, and the model (4) for M requires the backlog b^{S} from S. Cycle two: the model (4) for M requires the order rate o_r^{R} from R, and the model (5) for R requires the backlog b^{M} from M. Cycles complicate decentralized/distributed MPC implementations, since at any MPC update, coupled stages in each cycle must *assume* predictions for the states/inputs of one another. Such predictions are different in general than the *actual* locally computed predictions for those states/inputs. When cycles are not present, life is easier, as the stages can update sequentially, i.e., stages update in order from downstream to upstream, and the actual predictions from downstream stages can be transmitted to upstream stages at each update. In accordance with the MPC approach, the first portion of these actual predictions is implemented by each stage. Thus, the absence of cycles implies that stages can transmit policies that will be implemented. The sequential update approach is taken by Braun *et al.* [2], whose supply chain example contains no cycles. When cycles are present, on the other hand, actual predictions are not mutually available. Thus, some predictions must be assumed, incurring an unavoidable discrepancy between what a stage will do and what coupled stages assume it will do. One way to address this issue is to assume that the other stages react worst case, i.e., as bounded contracting disturbances, as done first by Jia and

Krogh [6]. The implementation employed here address the cycle issue in another way [4, 5]. Coupled stages receive the *previously* computed predictions from one another prior to each update, and rely on the remainder of these predictions as the assumed prediction at each update. To bound the unavoidable discrepancy between assumed and actual predictions, each stage includes a local move suppression penalty on the deviation between the current (actual) prediction and the remainder of the previous prediction.

3 Control Approaches

The nominal feedback policy, derived in [7], is given by

$$o_r^x(t) = l_r^x(t) + k_1[s_d - s^x(t)] + k_2[o_{ud}^x(t) - o_u^x(t)], \quad k_1, k_2 \in (0, \infty).$$

In the simulations in Section 4, the state and control constraints (2) are enforced by using saturation functions. The nominal control is decentralized in that the feedback for each stage depends only on the states of that stage. Simulation-based analysis and comparisons with real data from actual supply chains is presented as a justification for this choice of control in [7].

For the distributed MPC approach, the continuous time models are first discretized, using the discrete time samples $t_k = k * \delta$, with $\delta = 0.2$ days as the sample period, and $k \in \mathbb{N} = \{0, 1, 2, ...\}$. The prediction horizon is $T_p = P * \delta$ days, with $P = 75$, and the control horizon is $T_m = M * \delta$ days, with $M = 10$. For all three stages, the stock s^x and unfulfilled order o_u^x models are included in the MPC optimization problem. The backlog b^x, on the other hand, is not included in the optimization problem, as it is uncontrollable. Instead, the backlog is computed locally at each stage using the discretized model, the appropriate exogenous inputs that the model requires, and the saturation constraint in (2). For update time t_k, the *actual* locally predicted stock defined at times $\{t_k, ..., t_{k+P}\}$ is denoted $\{s^x(t_k; t_k), ..., s^x(t_{k+P}; t_k)\}$, using likewise notation for all other variables. The *true* stock at any time t_k is simply denoted $s^x(t_k)$, and so $s^x(t_k) = s^x(t_k; t_k)$, again using likewise notation for all other variables. In line with the notational framework in the MATLAB MPC toolbox manual [1], the set of measurable inputs are termed measured disturbances (MDs). By our distributed MPC algorithm, the MDs are *assumed* predictions. The set of MDs for each stage $x \in \{\text{S,M,R}\}$ is denoted $\mathcal{D}^x(t_k)$, associated with any update time t_k. The MDs for the three stages are $\mathcal{D}^{\text{S}}(t_k) = \{\mathbf{b}_{\text{as}}^{\text{S}}(k), \mathbf{o}_{r,\text{as}}^{\text{M}}(k)\}$, $\mathcal{D}^{\text{M}} = \{\mathbf{b}_{\text{as}}^{\text{M}}(k), \mathbf{b}_{\text{as}}^{\text{S}}(k), \mathbf{o}_{r,\text{as}}^{\text{R}}(k)\}$ and $\mathcal{D}^{\text{R}} = \{\mathbf{b}_{\text{as}}^{\text{R}}(k), \mathbf{b}_{\text{as}}^{\text{M}}(k), d_r^{\text{R}}\}$, where $\mathbf{o}_{r,\text{as}}^x(k) = \{o_{r,\text{as}}^x(t_k; t_k), ..., o_{r,\text{as}}^x(t_{k+P}; t_k)\}$ and $\mathbf{b}_{r,\text{as}}^x(k)$ is defined similarly using the assumed predicted backlog. The $(\cdot)_{\text{as}}$ subscript notation refers to the fact that, except for the demand rate at the retailer d_r^{R}, all of the MDs contain *assumed* predictions for each of the associated variables. It is presumed at the outset that a customer demand $d_r^{\text{R}}(\cdot) : [0, \infty) \to \mathbb{R}$ is known well into the future and without error. As this is a strong assumption, we are considering stochastic demand rates in our more recent work. Although it is locally computed, each stage's backlog is treated as an MD since it relies on the assumed demand rate prediction from the downstream stage. Note that the initial backlog is always the true backlog, i.e., $b^x_{r,\mathrm{as}}(t_k;t_k) = b^x(t_k)$ for each stage x and at any update time t_k. Let the set $\mathcal{X}^x(t_k) = \{s_d, o^x_{ud}(t_k;t_k), ..., o^x_{ud}(t_{k+P};t_k)\}$ denote the desired states associated with stage x and update time t_k. Using the equations from the previous section, the desired unfulfilled order prediction $o^x_{ud}(\cdot;t_k)$ in $\mathcal{X}^x(t_k)$ can be computed locally for each stage x given the MDs $\mathcal{D}^x(t_k)$. By our distributed MPC implementation, stages update their control in parallel at each update time t_k. The optimal control problem and distributed MPC algorithm for any stage are defined as follows.

Problem 1. For any stage $x \in \{\text{S,M,R}\}$, and at any update time t_k, $k \in \mathbb{N}$: Given: the current state $(s^x(t_k), o^x_u(t_k))$, the MDs $\mathcal{D}^x(t_k)$, the desired states $\mathcal{X}^x(t_k)$, the non-negative weighting constants $(W_s, W_{o_u}, W_u, W_{\delta u})$, and a non-negative target order rate o^{targ}_r,

Find: the optimal control $\mathbf{o}^x_{r,*}(k) \triangleq \{o^x_{r,*}(t_k;t_k), o^x_{r,*}(t_{k+1};t_k), ..., o^x_{r,*}(t_{k+M-1};t_k)\}$ satisfying

$$\mathbf{o}^x_{r,*}(k) = \arg\min \Bigg\{ \sum_{i=1}^{P} W_s \left[s^x(t_{k+i};t_k) - s_d\right]^2 + W_{o_u} \left[o^x_u(t_{k+i};t_k) - o^x_{ud}(t_{k+i};t_k)\right]^2$$
$$+ \sum_{j=0}^{M-1} W_u \left[o^x_r(t_{k+j};t_k) - o^{\mathrm{targ}}_r\right]^2 + W_{\delta u} \left[o^x_r(t_{k+j};t_k) - o^x_r(t_{k+j-1};t_k)\right]^2 \Bigg\},$$

where $o^x_r(t_{k-1};t_k) \triangleq o^x_{r,*}(t_{k-1};t_{k-1})$, subject to the discrete-time version of the appropriate model (equation (3), (4) or (5)), and the constraints in equation (2). ■

Algorithm 1. The distributed MPC law for any stage $x \in \{\text{S,M,R}\}$ is as follows:

Data: Current state: $(s^x(t_0), o^x_u(t_0), b^x(t_0))$. Parameters: δ, M, P, $(W_s, W_{o_u}, W_u, W_{\delta u})$, and o^{targ}_r.
Initialization: At initial time $t_0 = 0$, generate $\mathcal{D}^x(t_0)$ as follows: (a) Choose a nominal constant order rate $o^{x,\mathrm{nom}}_r$, set $o^x_{r,\mathrm{as}}(t_i;t_0) = o^{x,\mathrm{nom}}_r$, for $i = 0, ..., P$, and if $x =$ R or M, transmit $\mathbf{o}^x_{r,\mathrm{as}}(0)$ to M or S, respectively; (b) Compute $\mathbf{b}^x_{r,\mathrm{as}}(0)$, and if $x =$ S or M, transmit to M or R, respectively. Compute $\mathcal{X}^x(t_0)$ and solve Problem 1 for $\mathbf{o}^x_{r,*}(0)$.
Controller:

1. Between updates t_k and t_{k+1}, implement the current control action $o^x_{r,*}(t_k;t_k)$.
2. At update time t_{k+1}:
   a) Obtain $(s^x(t_{k+1}), o^x_u(t_{k+1}), b^x(t_{k+1}))$.
   b) Generate $\mathcal{D}^x(t_{k+1})$ as follows:
      i. Set $o^x_{r,\mathrm{as}}(t_{j+k+1};t_{k+1}) = o^x_{r,*}(t_{j+k+1};t_k)$, for $j = 0, ..., M-2$ and $o^x_{r,\mathrm{as}}(t_{j+k+1};t_{k+1}) = o^x_{r,*}(t_{k+M-1};t_k)$ for $i = M-1, ..., P$. If $x =$ R or M, transmit $\mathbf{o}^x_{r,\mathrm{as}}(k+1)$ to M or S, respectively.
      ii. Compute $\mathbf{b}^x_{r,\mathrm{as}}(k+1)$, and if $x =$ S or M, transmit to M or R, respectively.
   c) Compute $\mathcal{X}^x(t_{k+1})$ and solve Problem 1 for $\mathbf{o}^x_{r,*}(k+1)$.
3. Set $k = k+1$ and return to step 1. ■

By this algorithm, each stage initially computes an optimal order rate policy assuming neighboring stages employ a nominal constant order rate. For every subsequent update, each stage computes an optimal order rate policy, assuming that the MDs are based on the remainder of the previously computed policies computed of neighboring stages.

4 Numerical Experiments

The simulations were carried out in MATLAB 7.0, using Simulink 6.2 and the Model Predictive Control Toolbox 2.2. The nominal and distributed MPC approaches are compared on the full three stage problem, given a step increase and decrease in the customer demand rate at the retailer. For simulation purposes, we choose $d_r^{\mathrm{R}}(t) = 200$ cases/day for $t \in [0, \infty) \setminus [5, 15)$ and $d_r^{\mathrm{R}}(t) = 300$ for $t \in [5, 15)$. The response for the three stages under the nominal control policy ($k_1 = 1/15$, $k_2 = 1/30$) is shown in Figure 2. To implement the distributed MPC

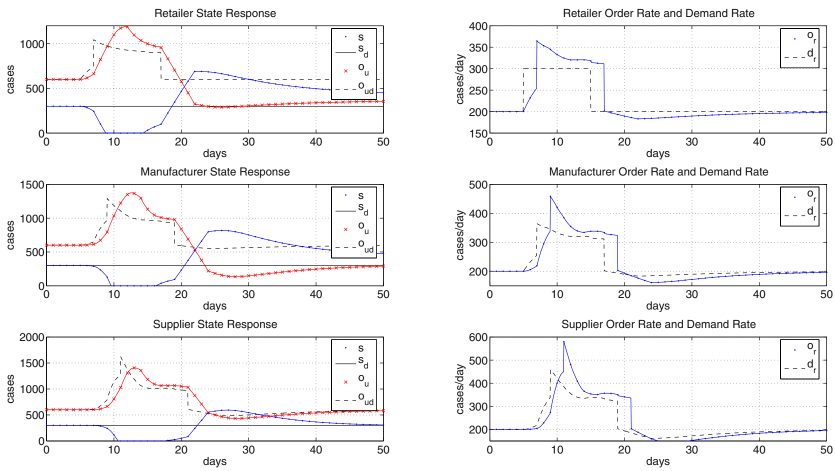

Fig. 2. Nominal response to step increase at 5 days and decrease at 15 days in retailer customer demand rate d_r^{R}

Algorithm 1, the anticipative action of the MPC Toolbox is employed so that each entire assumed prediction can be used. Recall that the assumed predictions are not the actual predictions, although the move suppression terms ($W_{\delta u}$ weighted) in the cost are used to ensure that these predictions are not too far apart. The forecasted demand rate at the retailer is also used with the anticipation option turned on. A more "apples-to-apples" comparison would be to incorporate internal models with the nominal approach that use the forecasted customer demand rate. The response for the three stages under the distributed MPC policy with anticipation is shown in Figure 3. The weights used in MPC for each stage are $(W_u, W_{\delta u}, W_s, W_{o_u}) = (1, 5, 5, 1)$. The stock and unfilled

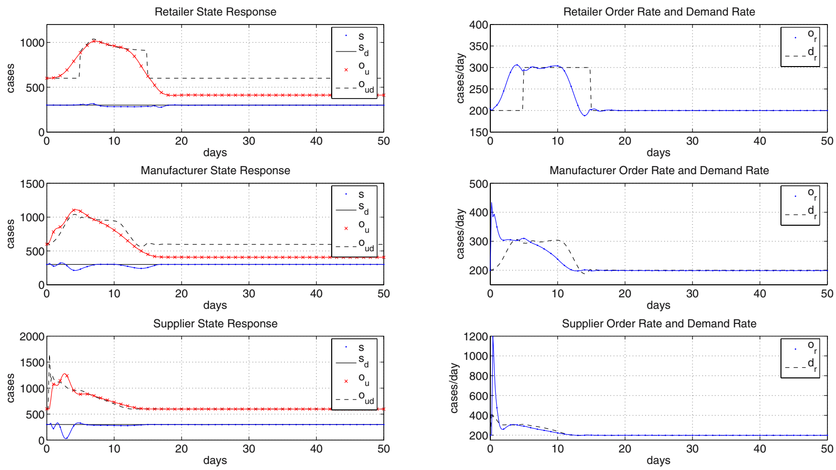

Fig. 3. Distributed MPC response to the same demand rate d_r^{R}. By using anticipation, the state responses are improved, and the order rates are smoother.

order state responses are an improvement over the nominal approach, both in terms of steady-state error and settling time. The nonzero steady-state error in the unfulfilled order and stock of stages M and R can be predicted by using system-type analysis. The well known "bullwhip effect" [3, 7] encountered in the coordination of a multi-stage supply chain is also seen in both figures, indicated by the increase in the maximum order rate excursion as one moves upstream from retailer to supplier.

5 Conclusions and Extensions

In this paper, a supply chain management problem was defined using the classic MIT "Beer Game" [7]. A nominal feedback policy, derived and experimentally validated in the supply chain literature, was then compared to a distributed MPC algorithm. The numerical experiments showed that the algorithm yielded improved performance over the nominal policy when the customer demand rate can be reliably forecasted. While one might redefine the nominal approach to include internal models that leverage forecasts, it is clear that MPC trivializes making use of forecasted inputs via anticipation, while respecting state and control constraints. As part of our on going work, we will consider a multi-echelon supply chain problem [3], in which at least two players operate within each stage. The decision problem becomes more complicated in these chains, since the update rates of different players in a stage are different in general, requiring an extension of the distributed MPC theory to asynchronous timing conditions. Additionally, we will consider stochastic (brownian) demand rate forecasts, and more realistic production models in the manufacturing stage.

References

[1] Bemporad, A., Morari, M. and Ricker, N. L., "Model Predictive Control Toolbox – for use with MATLAB," User's Guide, MathWorks, Inc. (2005).
[2] Braun, M. W., Rivera, D. E., Carlyle, W. M. and Kempf, K. G., "Application of Model Predictive Control to Robust Management of Multiechelon Demand Networks in Semiconductor Manufacturing," *Simulation*, **79** (3), pp. 139–156 (2003).
[3] Chopra, S. and Meindl, P., *Supply Chain Management, Strategy, Planning, and Operations*, Second Edition, Pearson Prentice-Hall, New Jersey (2004).
[4] Dunbar, W. B., "Distributed Receding Horizon Control of Dynamically Coupled Nonlinear Systems," Submitted to *IEEE Transaction on Automatic Control*, UCSC Baskin School of Engineering Technical Report ucsc-crl-06-03 (2006).
[5] Dunbar, W. B. and Murray, R. M., "Distributed Receding Horizon Control for Multi-Vehicle Formation Stabilization," *Automatica*, Vol. 2, No. 4 (2006).
[6] Jia, D. and Krogh, B. H., "Min-Max Feedback Model Predictive Control for Distributed Control With Communication," In *Proceedings of the IEEE American Control Conference*, Anchorage, AK (2002).
[7] Sterman, J. D., *Business Dynamics - Systems Thinking and Modelling in a Complex World*, McGraw-Hill, New York (2000).