Distributed Model Predictive Control for Dynamic Supply Chain Management

William B. Dunbar[†] and S. Desa[⋆]

[†] *Computer Engineering, University of California, Santa Cruz, 95064, USA,* dunbar@soe.ucsc.edu
[⋆] *Information Systems and Technology Management, University of California, Santa Cruz, 95064, USA*

Keywords : distributed control, model predictive control, supply chain management

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 and exciting future 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 stability properties of the distributed implementation in a realistic 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 (coupling) and evolution (dynamics) 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: information flows, e.g., an order requesting goods, and 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 [4]. Recently, Braun *et al.* [3, 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, 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, as well as 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) to the problem of dynamic management of supply chain networks. The theory behind the implementation for generic decoupled nonlinear dynamics and constraints and coupling in a quadratic cost function is presented by Dunbar and Murray in [6]. The theory is also extended to the case of dynamically coupled nonlinear systems [5]. 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 (consistent with traditional MPC control move suppression penalties), and that the updates happen sufficiently fast. Simulations have demonstrated performance comparable to a centralized implementation [6]. A contribution of this

paper will be to demonstrate the relevance and efficacy of the distributed NMPC approach in the venue of supply chain management. In fact, Braun *et al.* employ a heuristic version of our distributed implementation by sharing policies with downstream echelons in an acyclic interconnection network topology. In the presence of uncertainty, control deviation penalties are critical to their approach, a fact consistent with the theory in [6]. Once the supply chain problem has been defined in Section 2, other MPC-based alternative approaches will be cited and compared with our approach.

In Section 2, we define the supply chain management problem, using the classic MIT "Beer Game" [9] as the example three stage supply chain. The control approaches are then presented in Section 3, detailing a nominal feedback policy derived and studied in the supply chain literature [9], and our distributed MPC policy. Numerical experiments comparing the two approaches are then presented in Section 4, examining the response of a single stage and the full three stage chain. Finally, conclusions and extensions are then discussed in Section 5.

2 Problem Description

A supply chain consists of all the stages involved directly, or indirectly, in fulfilling a customer request [4]. 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's "build-to-order" management strategy is based on a version of the chain in Figure 1, where R is the customer, M is Dell, S is a chip supplier [4].

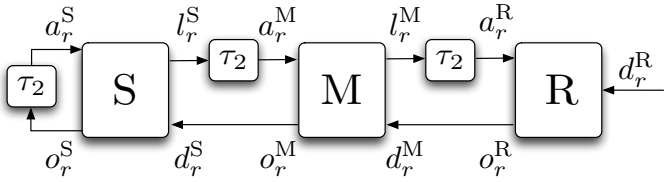


Figure 1: Block diagram of a three stage supply chain comprised of a supplier S, a manufacturer M, and a retailer R. Direct arrows depict information flows (orders) and arrows through a delay block, with time delay parameter τ_2, depict material flows (transport of goods). An exception is at left end of the chain, where the information flow from the supplier S is converted through fabrication into goods, which then flows back into the supplier S. For simplicity, this conversion is modeled as a delay.

The variables shown in the figure are defined below, and each will have a superscript denoting the corresponding stage it is associated with, e.g., o_r^{M} is the order rate of stock from the manufacturer stage.

We will use the classic MIT "Beer Game" [9] to provide a concrete context for visualizing the three stage supply chain in Figure 1. For this case, 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 units or goods sold per day), which then triggers a series of information flows and material flows, as shown in Figure 1. The *information flows* are assumed to have no (or 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 from R in our model.

An important variable in the supply chain is the inventory, or stock, in each stage of the supply chain, and a important objective in the management and control of a supply chain is to rapidly fulfill customer demand while keeping the inventory (stock) level in each stage as low as possible. For example, consider the manufacturer stage M in the supply chain. The manufacturer must respond to a demand from the retailer R from his current inventory and by ordering goods (an information flow) from the supplier S. The supplier attempts to fulfill the manufacturers order (from his inventory or by fabrication), and ships the goods (a material flow) to the manufacturer who receives (or acquires) them after a time delay that depends on the method of shipment.

While the supply chain is really a discrete-time system, for the purposes of the present paper we will follow the practice of Sterman [9] by modeling the supply chain as a continuous time system. 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

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

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

$$\text{where} \quad l_r^x(t) = d_r^x(t - \tau_1) + b^x(t)/t_b. \tag{3}$$

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 ([9], Chapter 11). As an example of a rate of accumulation, the time rate-of-change (accumulation rate) of stock $\dot{s}^x(t)$ is the difference between the rate at which goods are acquired (acquisition rate $a_r^x(t)$) and the rate at which they are shipped to the customer $l_r^x(t)$. Two other accumulation variables are the rate of change of unfulfilled orders $\dot{o}_u^x(t)$ (at the upstream end of each stage) and the rate of change of backlogged orders $\dot{b}^x(t)$ (at the downstream end of each stage). 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 $o_{r,\max}$.

As stated, the dynamics of the supply chain can also arise from one of two types of material flow delay. The first type of delay, known as "pipeline" delay, occurs when the outflow of a block is the inflow with a time lag. For example, as indicated by the delay block in Figure 1, the acquisition rate a_r^x at any stage is the shipment rate l_r^x from the previous (source) stage delayed by the time it takes to ship the goods from source stage to destination stage. The second type of delay, known as the first-order material delay, occurs when there is an average delay (or flow) time associated with the outflow from a stage of material accumulated in that stage, and is equal to the accumulated material (number of units) divided by the average flow time.

The shipment rate equation (3) employs both examples of material flow delay models. A qualitative explanation of this equation is the following: orders are fulfilled based on satisfying incoming customer demand as well as clearing backlogged orders. Because of delays in both fulfilling customer demand and clearing backlog, the shipment (order fulfillment) rate is the sum of (i) a time-delayed demand rate, delayed by the time τ_1 required for processing the order (a pipeline delay), and (ii) the backlog clearance rate, which is equal to the backlog divided by the average backlog-clearance flow time (a first-order delay). On a more quantitative note, after substitution of (3) into the model for backlog in (1), it is clear that the backlog is uncontrollable. If the demand rate converges to a steady value, the backlog will converge to zero, implying stabilizability.

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 ([9], pg. 686). Specifically, the control objective for each stage is $(s^x(t), o_u^x(t)) \to (s_d, o_{ud}^x(t))$, 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 $d_r^x(t) \to d_r$, then backlog will converge to zero, the shipment rate converges $l_r^x(t) \to d_r$, 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

$$\underline{\text{S:}}\ a_r^{\text{S}}(t) = o_r^{\text{S}}(t-\tau_2),\ d_r^{\text{S}}(t) = o_r^{\text{M}}(t) \ \Big|\ \underline{\text{M:}}\ a_r^{\text{M}}(t) = l_r^{\text{S}}(t-\tau_2),\ d_r^{\text{M}}(t) = o_r^{\text{R}}(t) \ \Big|\ \underline{\text{R:}}\ a_r^{\text{R}}(t) = l_r^{\text{M}}(t-\tau_2) \quad (4)$$

and the demand rate at the retailer $d_r^{\text{R}}(t)$ is an input defined as the current/projected customer demand. For stages M and R, the acquisition rate is the shipment rate delayed by the time required to ship material from source to destination. The demand rate at stages M and S is simply the downstream order rate from R and M, respectively.

In light of the discussion above regarding the backlog dynamics in (1), and from the demand rates defined in (4), we make the following observation. By requesting order rates that are less aggressive, stages R and M can keep the backlog levels of stages M and S, respectively, lower than when more aggressive order rates are requested, where by "aggressive" we mean order rates that are fast compared to the processing delay constant τ_1. Thus, loosely speaking, less aggressive order rates are consistent with the objective of keeping backlog levels low.

After substitutions, we have the following models for each of the three stages. For the supplier stage,

$$\left.\begin{array}{l} \dot{s}^{\text{S}}(t) = o_r^{\text{S}}(t-\tau_2) - o_r^{\text{M}}(t-\tau_1) - b^{\text{S}}(t)/t_b \\ \dot{o}_u^{\text{S}}(t) = o_r^{\text{S}}(t) - o_r^{\text{S}}(t-\tau_2) \\ \dot{b}^{\text{S}}(t) = o_r^{\text{M}}(t) - o_r^{\text{M}}(t-\tau_1) - b^{\text{S}}(t)/t_b \end{array}\right\}. \quad (5)$$

For the manufacturer stage,

$$\left.\begin{array}{l} \dot{s}^{\text{M}}(t) = o_r^{\text{M}}(t-\tau_1-\tau_2) + b^{\text{S}}(t-\tau_2)/t_b - o_r^{\text{R}}(t-\tau_1) - b^{\text{M}}(t)/t_b \\ \dot{o}_u^{\text{M}}(t) = o_r^{\text{M}}(t) - o_r^{\text{M}}(t-\tau_1-\tau_2) - b^{\text{S}}(t-\tau_2) \\ \dot{b}^{\text{M}}(t) = o_r^{\text{R}}(t) - o_r^{\text{R}}(t-\tau_1) - b^{\text{M}}(t)/t_b \end{array}\right\}. \quad (6)$$

For the retailer stage,

$$\left.\begin{array}{l} \dot{s}^{\text{R}}(t) = o_r^{\text{R}}(t-\tau_1-\tau_2) + b^{\text{M}}(t-\tau_2)/t_b - d_r^{\text{R}}(t-\tau_1) - b^{\text{R}}(t)/t_b \\ \dot{o}_u^{\text{R}}(t) = o_r^{\text{R}}(t) - o_r^{\text{R}}(t-\tau_1-\tau_2) - b^{\text{M}}(t-\tau_2)/t_b \\ \dot{b}^{\text{R}}(t) = d_r^{\text{R}}(t) - d_r^{\text{R}}(t-\tau_1) - b^{\text{R}}(t)/t_b \end{array}\right\}. \quad (7)$$

We say that two stages have bidirectional coupling if both of the differential equation models of each stage depend upon the state and/or input of the other stage. Equations (5)–(7) 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 (5) for S requires the order rate o_r^{M} from M, and the model (6) for M requires the backlog b^{S} from S. Cycle two: the model (6) for M requires the order rate o_r^{R} from R, and the model (7) 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.* [3, 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 [7] and later by Richards and How [8] (not in the context of supply chains), although the performance of such schemes has not been extensively evaluated.

The implementation employed here address the cycle issue in another way [5, 6]. 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 penalty on the deviation between the current (actual) prediction and the remainder of the previous prediction. A motivation

for this paper is to examine the performance of this implementation when subsystems are dynamically coupled. When subsystem are coupled through performance objective, the performance was shown to be comparable to a centralized implementation [6]. An interesting distributed MPC alternative to our implementation for dynamically coupled subsystems is given by Venket *et al.* [10, 11], though it requires that subsystems be LTI and coupled solely through the control inputs.

In the implementation and simulations here, the penalty on the deviation between the current and assumed prediction, locally within each stage, will be in the form of a move suppression term in the cost function. While proof of convergence of our implementation requires that the penalty take the form of a constraint, the simulations show excellent results while using move suppression instead of a constraint. It is well known in the MPC community that the move suppression term has the effect of making the controller less sensitive to prediction inaccuracies, although usually at the price of degrading set point tracking performance [1]. Such reduced sensitivity is precisely how one can mitigate the discrepancy between assumed and actual predictions that is intrinsic to the MPC problem for distributed systems over networks containing cycles.

To conclude this section, note that for this example supply chain, the cycles could be eliminated. Specifically, if M has the initial condition and model for b^{S}, then M can compute a prediction for b^{S} locally. Likewise, if R has the initial condition and model for b^{M}, then R can compute b^{M} locally. In that case, for distributed MPC, R would not need a prediction from M, but merely the initial backlog at each update. As such, a sequential MPC could be implemented. However, from a supply chain management perspective, as well as a systems perspective in general, it is better to make backlog predictions of any stage from within that stage. One reason for this is that since the true backlog is evolving locally, that evolution can be used to detect and correct model errors. Such errors would likely be harder to detect and manage from a downstream stage, at least not without more extensive communication between the stages. As such, we treat backlogs as locally computed states within each stage. Of course, for comparison purposes, it would be useful to perform the elimination and compare the sequential approach with our approach, since no model error is assumed in this example. For space reasons, we save this for a future exercise. We now define the control approaches that will be compared in simulations.

3 Control Approaches

3.1 Nominal Feedback Control

The nominal feedback policy, derived in [9], 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. Only simulation based analysis and comparisons with real data from actual supply chains has been presented as a justification for this choice of control [9].

3.2 Distributed Model Predictive Control

For the 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 $k \in \mathbb{N} = \{0, 1, 2, ...\}$. Each model (5)–(7) is cast in the Laplace domain, and each delay is replaced by its 4th order Padé approximation. Each continuous time transfer function is then transformed into state space and discretized using the δ sample period. In the MPC problem we will refer to these models as the *discrete-time versions* of (5)–(7), although technically they are the delay-free discrete-time state space versions of the original models. 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 discrete time 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.

For each model in the MPC problem, there are two states (s^x, o^x_u), the control o^x_r and a set of other measurable inputs, depending on the stage. In line with the MPC framework presented in the MATLAB MPC toolbox manual [1], this 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 $\mathcal{D}^{\text{S}}(t_k) = \{\mathbf{b}^{\text{S}}_{\text{as}}(k), \mathbf{o}^{\text{M}}_{r,\text{as}}(k)\}$, $\mathcal{D}^{\text{M}} = \{\mathbf{b}^{\text{M}}_{\text{as}}(k), \mathbf{b}^{\text{S}}_{\text{as}}(k), \mathbf{o}^{\text{R}}_{r,\text{as}}(k)\}$ and $\mathcal{D}^{\text{R}} = \{\mathbf{b}^{\text{R}}_{\text{as}}(k), \mathbf{b}^{\text{M}}_{\text{as}}(k), d^{\text{R}}_r\}$, where $\mathbf{o}^x_{r,\text{as}}(k) = \{o^x_{r,\text{as}}(t_k;t_k), ..., o^x_{r,\text{as}}(t_{k+P};t_k)\}$ and $\mathbf{b}^x_{r,\text{as}}(k)$ 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 assumed at the outset that a customer demand $d^{\text{R}}_r(\cdot) : [0,\infty) \to \mathbb{R}$ is known well into the future and without error. 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., $b^x_{r,\text{as}}(t_k;t_k) = b^x(t_k)$ 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 MPC problem for any stage is 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 sequence $\mathbf{o}^x_{r,*}(k) \triangleq \{o^x_{r,*}(t_k;t_k), o^x_{r,*}(t_{k+1};t_k), ..., o^x_{r,*}(t_{k+M-1};t_k)\}$ satisfying

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

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 (5), (6) or (7)), and the constraints in equation (2). ■

The $W_{\delta u}$ weighted term in the cost is the move suppression penalty referred to above. The distributed MPC algorithm is now stated.

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 $o^{x,\text{nom}}_r$, set $o^x_{r,\text{as}}(t_i;t_0) = o^{x,\text{nom}}_r$, for $i = 0, ..., P$, and if $x = \text{R}$ or M, transmit $\mathbf{o}^x_{r,\text{as}}(0)$ to M or S, respectively; (b) Compute $\mathbf{b}^x_{r,\text{as}}(0)$, and if $x = \text{S}$ 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 $o^x_{r,\text{as}}(t_{j+k+1};t_{k+1}) = o^x_{r,*}(t_{j+k+1};t_k)$, for $j = 0, ..., M-2$ and $o^x_{r,\text{as}}(t_{j+k+1};t_{k+1}) = o^x_{r,*}(t_{k+M-1};t_k)$ for $i = M-1, ..., P$. If $x = \text{R}$ or M, transmit $\mathbf{o}^x_{r,\text{as}}(k+1)$ to M or S, respectively.
      ii. Compute $\mathbf{b}^x_{r,\text{as}}(k+1)$, and if $x = \text{S}$ 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 $k = k + 1$.

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

4 Numerical Experiments

The simulations were carried out in MATLAB 7.0, using Simulink 6.2 and the Model Predictive Control Toolbox 2.2. First, we compare the nominal and MPC approaches looking only at a single stage (the supplier S) responding to an initial deviation between the actual and desired stock, and then to a step increase in the demand rate d_r^{S}. Then, 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.

4.1 Single Stage Supply Chain

The two approaches show nearly identical responses, given an initial deviation between the actual and desired stock. The simulation comparison is shown in Figure 2. The control gains in the nominal approach

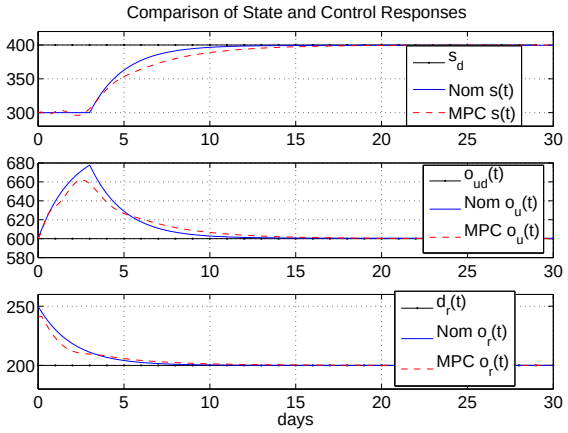
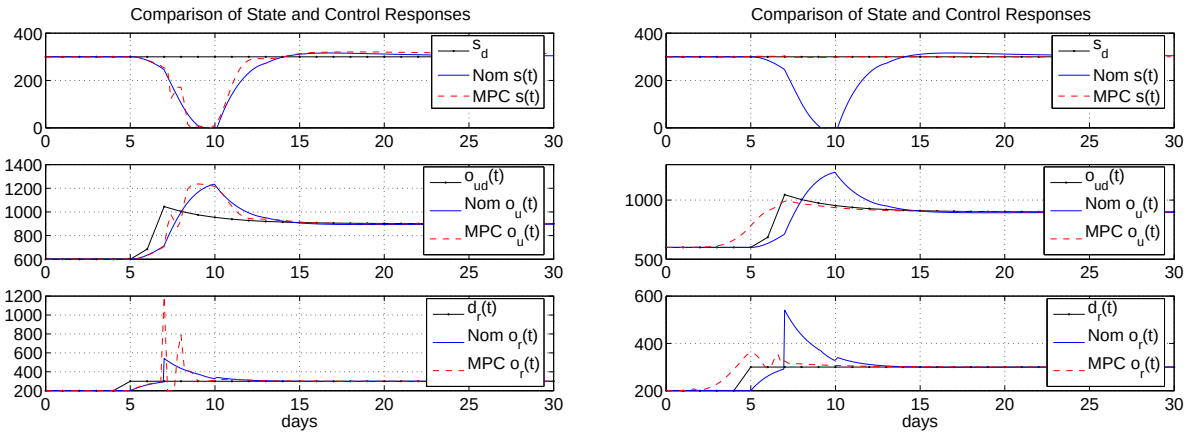


Figure 2: Response to initial deviation in stock with comparable performance between the approaches.

are $k_1 = 0.5$ and $k_2 = 0.5$. The weights used in MPC are $(W_u, W_{\delta u}, W_s, W_{o_u}) = (2, 1, 1, 1)$. The initial and desired stock are $s^{\mathrm{S}}(0) = 300$ and $s_d = 400$ cases.

Next, consider a step increase in the demand rate: $d_r^{\mathrm{S}}(t) = 200$ cases per day for $t \in [0, 5)$ and $d_r^{\mathrm{R}}(t) = 300$ for $t \in [5, \infty)$. Observe the comparison in Figure 3. The weights used in MPC without anticipation (subfigure (a)) are $(W_u, W_{\delta u}, W_s, W_{o_u}) = (2, 1, 1, 1)$, demonstrating comparable performance, although the MPC order rate is substantially more aggressive during days 7–9. We then turn on the anticipation option, meaning the MPC problem uses full future knowledge of the demand rate in predicting the stock and unfulfilled order states. The weights used in MPC with anticipation (subfigure (b)) are $(W_u, W_{\delta u}, W_s, W_{o_u}) = (0, 0.5, 10, 1)$, showing a substantial improvement in the performance of stock and unfulfilled order, with a much less aggressive order rate resulting in substantially reduced backlog levels (not shown). Thus, MPC can be tuned to be quite comparable to the nominal approach in this single stage example. Moreover, if a reliable prediction of the demand rate is available, MPC with anticipation shows substantial improvement in meeting the control objective, i.e., keeping the stock and unfulfilled orders at desired levels, while requesting orders at a rate that is not too aggressive. In a sense, figure (b) is comparing apples to oranges. A more appropriate comparison would be between MPC with anticipation and the nominal approach redefined to also incorporate a forecasted demand rate, e.g., by including an internal model that generates open-loop order-rates from the forecast. Such a comparison will be part of our future work. For now, and in the three stage experiment that follows, we demonstrate the advantage of using full knowledge of a forecast through MPC.



(a) (b)

Figure 3: State and control responses to step input in demand rate from 200 to 300 cases/day at day 5. Plot (a) shows comparable state performance between the approaches, although the MPC order rate is more aggressive during days 7 through 9. Plot (b) shows improved MPC performance as anticipative action is turned on for both the desired unfulfilled order and the demand rate. The MPC stock is observed to remain within 5 cases of the desired value of $s_d = 300$ cases.

4.2 Three Stage Supply Chain

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 $t \in [5, 15)$ in the three stage experiments. The response for the three stages under the nominal control policy is shown in Figure 4.

To implement the distributed MPC 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 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. Again, a more "apples-to-apples" comparison with the nominal approach would be to redefine the nominal approach to include internal models that would make use of the forecasted customer demand rate. The response for the three stages under the distributed MPC policy with anticipation is shown in Figure 5. 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 unfulfilled order state responses are an improvement over the nominal approach, both in keeping close to their desired values and in displaying shorter settling times. Note the nonzero steady-state error in the unfulfilled order of stages M and R using either control approach. It is also interesting to note that the familiar "bullwhip effect" [4, 9] encountered in the coordination of a multi-stage supply chain can be seen in both Figures 4 and 5. Specifically, this effect is demonstrated by the fact that the magnitude of the maximum excursion of the order rate gets larger as we move upstream in the supply chain, from retailer to supplier.

5 Conclusions and Extensions

In this paper, a realistic supply chain management problem was defined using the classic MIT "Beer Game" [9]. 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, particularly when the customer demand, an exogenous input to the supply chain, can be reliably forecasted. An appropriate extension of these results that would yield a better comparison would be to redefine the nominal approach to include internal models that would make use of the forecasted customer demand rate, while maintaining a *decentralized structure*. While we are not immediately aware of any literature supporting such an extension,

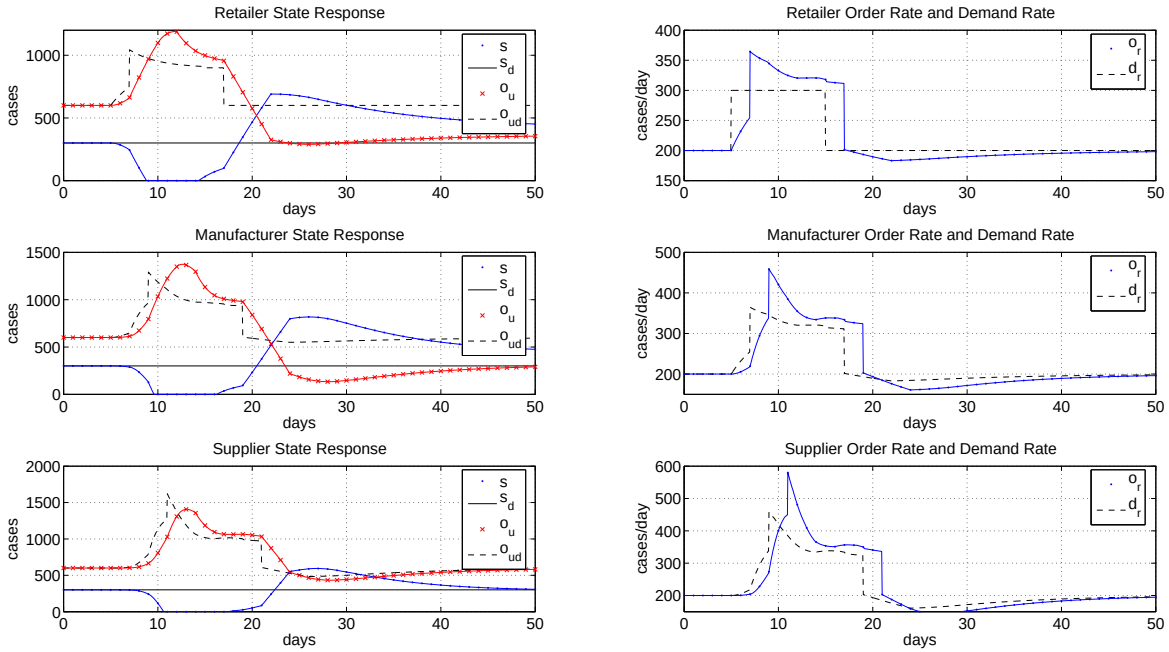


Figure 4: Nominal response to step increase at 5 days and decrease at 15 days in customer demand at the retailer. The control gains are $k_1 = 1/15$ and $k_2 = 1/30$, resulting in a feedback dominated by the shipment rate. Using higher gains, such as those used in the single stage results, gave large transients and under damped oscillations. Nonzero steady-state errors exist for the stock and unfulfilled order variables at the manufacturer and retailer, and order rates are moderately aggressive.

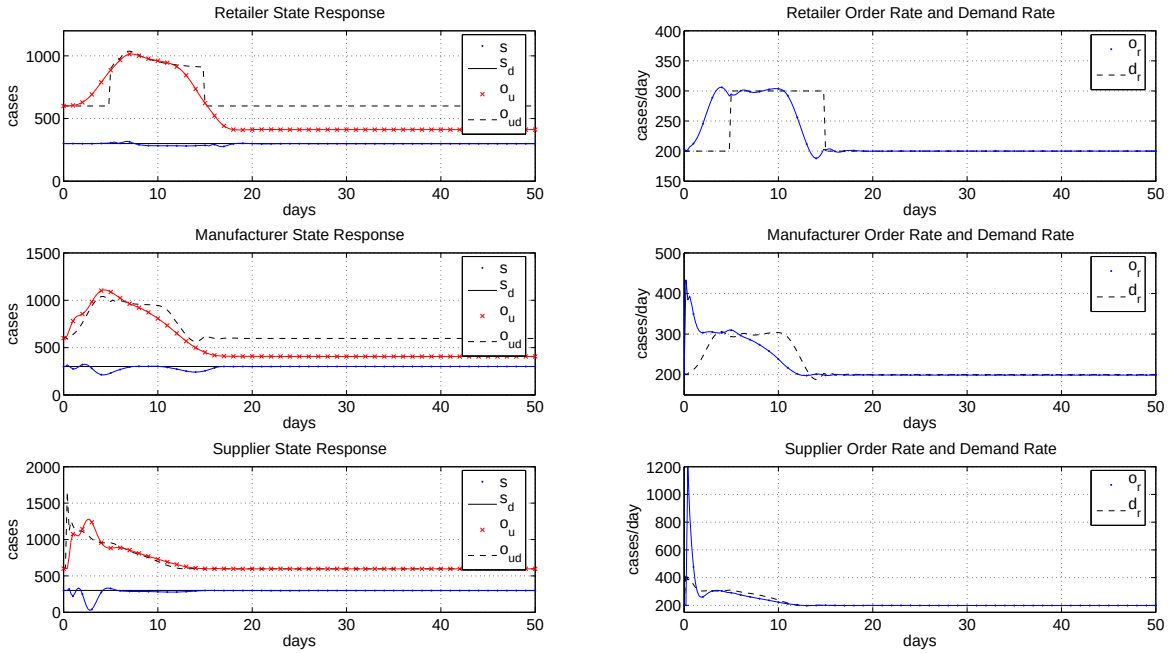


Figure 5: Distributed MPC response to the forecasted customer demand at the retailer. In comparison to the nominal response in Figure 4, the state responses are improved, and the order rates are less aggressive overall.

the ability of our implementation to make use of forecasts in a distributed way is straightforward.

For the models derived here, under both control approaches, it was found that the unfulfilled order in stages M and R exhibited nonzero steady-state error. A detailed relative degree, controllability and stabilizability analysis on the system models should reveal the source of this bias, as well as other fundamental characteristics of the system. Although the models were here presented in continuous time, in the supply chain literature, modeling is typically undertaken in the discrete time domain, a domain consistent with the MPC approach in general and the MPC Toolbox utilized in the simulation results. As part of our on going work, we will explore multi-echelon chains [4], in which at least two (and possibly many) players operate within each stage, e.g., the S stage in Dell's "build-to-order" supply chain management strategy might contain several chip suppliers such as Samsung, Intel and Micron. The decision problem becomes more complicated in these chains, since the update rates of different players in a stage are different in general. This requires an extension of the theory of distributed MPC to operate under asynchronous timing conditions.

Acknowledgments

The authors would like to thank Martin Tschoeke of Gottingen University for his assistance in the development of models and nominal control for the single and multi-stage supply chain.

References

[1] Bemporad, A., Morari, M. and Ricker, N. L., "Model Predictive Control Toolbox – for use with MATLAB," User's Guide, Version 2, The MathWorks, Inc. (2005).

[2] Braun, M. W., Rivera, D. E., Carlyle, W. M. and Kempf, K. G., "A Model Predictive Control Framework for Robust Management of Multi-Product Multi-Echelon Demand Networks," In *Proceeding of the 15th IFAC World Congress*, Barcelona, Spain (2002).

[3] 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).

[4] Chopra, S. and Meindl, P., *Supply Chain Management, Strategy, Planning, and Operations*, Second Edition, Pearson Prentice-Hall, New Jersey (2004).

[5] Dunbar, W. B., "A Distributed Receding Horizon Control Algorithm for Dynamically Coupled Nonlinear Systems," Accepted to *IEEE Conference on Decision and Control / IEE European Control Conference* (2005).

[6] Dunbar, W. B. and Murray, R. M., "Distributed Receding Horizon Control with Application to Multi-Vehicle Formation Stabilization," Accepted to *Automatica* (2004).

[7] 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).

[8] Richards, A. and How, J., "A Decentralized Algorithm for Robust Constrained Model Predictive Control," In *Proceedings of the IEEE American Control Conference*, Boston, MA (2004).

[9] Sterman, J. D., *Business Dynamics - Systems Thinking and Modelling in a Complex World*, McGraw-Hill, New York (2000).

[10] Venkat, A. N., Rawlings, J. B. and Wright, S. J., "Plant-Wide Optimal Control with Decentralized MPC," In *Proceedings of the IFAC Dynamics and Control of Process Systems Conference*, Boston, MA (2004).

[11] Venkat, A. N., Rawlings, J. B. and Wright, S. J., "Stability and Optimality of Distributed Model Predictive Control," Accepted to *IEEE Conference on Decision and Control / IEE European Control Conference* (2005).