INTEGRATION OF CONTROL THEORY AND SCHEDULING METHODS FOR SUPPLY CHAIN MANAGEMENT

by

Kaushik Subramanian

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Chemical and Biological Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2012

Date of final oral examination: 12/19/12

The disseration is approved by the following members of the Final Oral Committee:
James B. Rawlings, Professor, Chemical and Biological Engineering
Christos T. Maravelias, Associate Professor, Chemical and Biological Engineering
Micheal D. Graham, Professor, Chemical and Biological Engineering
Ross E. Swaney, Associate Professor, Chemical and Biological Engineering
Jennifer Reed, Assistant Professor, Chemical and Biological Engineering
Ananth Krishnamurthy, Associate Professor, Industrial and Systems Engineering

To my teachers.

ACKNOWLEDGMENTS

I would like to express my sincere thanks to my advisers, Prof Jim Rawlings and Prof. Christos Maravelias for their invaluable help and guidance. Their depth of knowledge, attention to detail, and enthusiasm to teach and learn has always motivated me.

I would like to thank Prof Scahin Patwardhan, for his invaluable advice and guidance regarding my decision to pursue a PhD.

I also thank Prof. Ross Swaney, Prof. Jennifer Reed, Prof. Michael Graham and Prof. Ananth Krishnamurthy for taking time to be on my thesis committee.

I would like to thank Dr Jesus Flores-Cerrillo and Dr Lawrence Megan from the Advanced Controls and Optimizations group at Praxair, Tonawanda, NY, for giving me the opportunity to work in an exciting industrial research group. I have returned brimming with new ideas after my visits to Praxair.

I would like to thank Dr Aswin Venkat and Brent Miller from Bloom Energy. The experience working in a start-up company is truly unique.

I would like to thank all the members of Jim Rawlings' group. It is always a pleasure interacting with Ankur Gupta, may it be about optimization algorithms or about the Lannister clan. Rishi Srivastava has been a very good and patient friend. I enjoyed my time with Brett Stewart, whose guidance helped me a lot in understanding my research problem. Rishi Amrit has been a savior whenever I experienced computing troubles. His help in explaining Economic MPC to me has helped me immensely. I have also enjoyed interacting with Cuyler Bates, who has listened patiently to all my new ideas. I must thank Mary Diaz, for all her administrative support and the delicious cookies and cakes.

I am also thankful to all the friends I have made in Madison. The memories with Swami, Raghu, Sriram, Pavithra, Janani are something that I shall cherish for the rest of my life.

Thanks to the "Panni group" - Sashi, Shriram, Srikant, Anand. I look forward to many more New year "reunion" trips!

Finally, a big thanks to Amma, Appa and, Vidu for all their love, encouragement and support.

Kaushik - Madison, WI

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Appendix
Figure Page

ABSTRACT

A supply chain is a network of facilities and distribution options that performs the functions of procuring raw materials, transforming them to products and distributing the finished products to the customers. The modern supply chain is a highly interconnected network of facilities that are spread over multiple locations and handle multiple products. In a highly competitive global environment, optimal day-to-day operations of supply chains is essential.

To facilitate optimal operations in supply chains, we propose the use of Model Predictive Control (MPC) for supply chains. We develop:

- A new cooperative MPC algorithm that can stabilize any centralized stabilizable system
- A new algorithm for robust cooperative MPC
- A state space model for the chemical production scheduling problem

We use the new tools and algorithms to design model predictive controllers for supply chain models. We demonstrate:

- **Cooperative control for supply chains**: In cooperative MPC, each node makes its decisions by considering the effects of their decisions on the entire supply chain. We show that the cooperative controller can perform better than the noncooperative and decentralized controller and can reduce the bullwhip effect in the supply chain.
- **Centralized economic control**: We propose a new multiobjective stage cost that captures both the economics and risk at a node, using a weighted sum of an economic stage cost

and a tracking stage cost. We use Economic MPC theory (Amrit, Rawlings, and Angeli, 2011) to design closed-loop stable controllers for the supply chain.

- **Integrated supply chain**: We show an example of integrating inventory control with production scheduling using the tools developed in this thesis. We develop simple terminal conditions to show recursive feasibility of such integrated control schemes.

Chapter 1

Introduction

In today's highly competitive market, it is important that the process industries integrate their manufacturing processes with the downstream supply chain to maximize economic benefits. For example, BASF was able to generate $10 million/year savings in operating costs by performing a corporate network optimization (Grossmann, 2005). In recent years, Enterprise Wide Optimization (EWO) has become an important research area for both academia and industry. Grossmann (2005) defines EWO as

> An area that lies at the interface of chemical engineering (process systems engineering) and operations research. It involves optimizing the operations of supply, manufacturing (batch or continuous) and distribution in a company. The major operational activities include planning, scheduling, real-time optimization and inventory control.

In process control technologies like Model Predictive Control (MPC), feedback from the process, in terms of measurements of the current state of the plant, is used to improve the control performance. It has been recognized that using feedback control can be significant for supply chain optimization. Backx, Bosgra, and Marquardt (2000) highlight the importance of considering the dynamics and feedback in process integration. They say

> Future process innovation must aim at a high degree of adaptability of manufacturing to the increasingly transient nature of the marketplace to meet the challenges of global competition. Adaptation to changing environmental conditions requires feedback control mechanisms, which manipulate the quality performance

> and transition flexibility characteristics of the manufacturing processes on the basis of measured production performance indicators derived from observations of critical process variables. This feedback can be achieved by means of two qualitatively completely different approaches residing on two different time scales. The first, shorter time scale focused approach aims at the adaptation of process operations by modified planning, scheduling and control strategies and algorithms assuming fixed installations. The second approach attempts to achieve performance improvements by reengineering the plant, including process and equipment, as well as instrumentation and operation support system design.

Model predictive control is a multi-variable control algorithm which deals with operating constraints and multi-variable interactions. MPC's ability to handle constraints along with the online optimization of the control problem has made it a very popular control algorithm in the process industries (Qin and Badgwell, 2003; Morari and Lee, 1997). At the heart of MPC is a dynamic process model that is used to predict the influence of inputs (manipulated variables) on the process. Based on the prediction, an optimization problem is solved online, to find the optimal control action.

In this thesis, we propose model predictive control as a general purpose tool to aid in enterprise wide optimization. Traditionally, decision making in the process industries follows a hierarchical structure. At the top, the planning module uses a simplified model of the facility along with some knowledge of the supply chain dynamics to predict production targets and material flows. This problem is called the Planning problem. In the scheduling layer, the solution of the planning problem is used to find a detailed schedule for the plant. This problem is called the Scheduling problem. The Real Time Optimizer (RTO) uses the solution of the scheduling problem to find optimal set-points for the plant. Finally, the advanced controller regulates the plant to the predicted set-points. The main contribution of this thesis is to formulate parts of the short term planning problem (interaction of the production facility with the supply chain) and the production scheduling problem as dynamic models (also see hybrid modeling for rolling horizon approaches (Maravelias and Sung, 2009, Sec 4.4)), that can be "controlled" using MPC.

Thus, in conjunction with economic MPC (Amrit et al., 2011) that integrates the advanced control layer with the RTO; the tools developed in this thesis allows us to study the entire decision making hierarchy in the enterprise from a predictive control point of view.

We focus on two important aspects of the supply chain. First, from an operations research standpoint, we use MPC to coordinate orders and shipments in the supply chain to minimize (maximize) costs (profits). Second, from a process systems engineering standpoint, we develop tools to formulate the short term production scheduling problem as a dynamic control problem.

Overview of the thesis

Chapter 2 – Model predictive control: In this chapter, we summarize the fundamental theory for linear MPC. We state stability theorems for centralized, suboptimal and cooperative MPC. We then propose a cooperative MPC algorithm that is applicable to all centralized stabilizable systems and an algorithm for robust cooperative MPC using tube-based MPC (Rawlings and Mayne, 2009, Chapter 3).

Chapter 3 – A state-space model for chemical production scheduling: In this chapter we derive a state space model for the production scheduling problem and highlight how different scheduling disturbances can be modeled. We use ideas from MPC like the terminal region to show how the state space model can be used in iterative scheduling.

Chapter 4 – Distributed MPC for supply chain optimization: In this chapter, we show how to model a supply chain, and use the theory outlined in Chapter 2 to design centralized, distributed and, robust MPC for supply chains.

Chapter 5 – Economic MPC for supply chains: In this chapter, we briefly review economic MPC theory and show how it can be tailored for supply chains. Instead of optimizing a tracking objective, we show how to design economic and multiobjective optimization problems for supply chains. We conclude this chapter with an example of an integrated production scheduling–supply chain problem solved in a rolling horizon framework.

Chapter 6 – Conclusions and future work: We end with a summary of the contributions and recommendations for future work.

Chapter 2

Model predictive control

2.1 Introduction

Model Predictive Control (MPC) is an optimization based control algorithm in which a model of the plant is used to predict the future evolution of the plant. A constrained optimization problem is solved using these predictions to find the optimal input to the plant. In MPC, at each sampling time, the optimizer finds the next N inputs, in which N is called the prediction/control horizon. The first of these inputs is injected to the plant and the whole procedure is repeated at the next sampling time, during which, the state of the plant is estimated from measurements. In this way, the rolling horizon framework incorporates feedback. MPC is widely used in many industries like Petrochemicals, fine chemicals, food, automotive and aerospace, because of its ability to handle multiple inputs and outputs (MIMO controller) and process constraints (Qin and Badgwell, 2003).

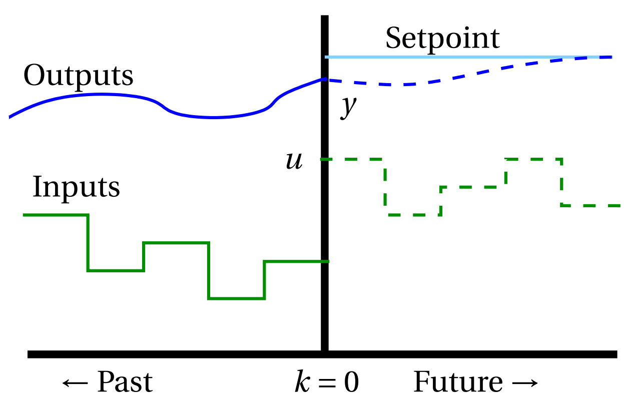

Figure 2.1: Rolling horizon optimization.

Model predictive control technology has two important aspects that are interconnected. First, is the design of the online optimization problem that is solved. The design must account for the control objectives, process constraints and dynamics. Second, is the study of the injected control moves in the plant. Since only the first input of the optimal input sequence is used, it is important to provide guarantees that the control objectives are met in the closed-loop. Stability theory provides controller design guidelines and theoretical support to ensure desirable closed-loop behavior by using the rolling horizon optimization framework.

This chapter is organized as follows. In Section 2.2, we provide an overview of centralized MPC. We discuss optimal MPC in Section 2.2.2 and suboptimal MPC in Section 2.2.3. In Section 2.3, we introduced distributed MPC; with noncooperative MPC discussed in Section 2.3.2 and cooperative MPC discussed in Section 2.3.3. An algorithm for robust cooperative control is presented in Section 2.4. In Section 2.5, we discuss related work in the field of cooperative/ distributed MPC. Since the focus of this thesis is the application of control technology for supply chain optimization, we focus our attention on linear models in this section. In Chapter 4, we show that the supply chain dynamics can be described by linear models.

2.2 Centralized MPC

2.2.1 Preliminaries

We consider the linear system

$$x^+ = Ax + Bu \tag{2.1}$$

in which $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ are the states and inputs while x^+ is the successor state.

The system is constrained by the state constraint $x \in \mathbb{X} \subseteq \mathbb{R}^n$ and input constraint $u \in \mathbb{U} \subset \mathbb{R}^m$.

For a given finite horizon N, we define the input sequence as $\mathbf{u} = (u(0), u(1), \ldots, u(N-1)) \in \mathbb{U}^N$. The state at time $j \geq 0$ for a system starting at state x at time $j = 0$, under control $\mathbf{u}$ is given by $\phi(j; x, \mathbf{u})$. If there is no ambiguity, $\phi(j; x, \mathbf{u})$ is also denoted as $x(j)$.

We define the tracking stage cost as $\ell(x,u) = 1/2(x'Qx + u'Ru); Q, R > 0$. We define an economic state cost $\ell_E(x,u)$ in Chapter 5. Without loss of generality, we assume that the MPC is designed to track (x,u) to the origin. For systems in which the steady state is not the origin, we can modify $\ell(x,u)$ by a simple variable transformation $x \leftarrow x - x_s$, in which x_s is the steady state of choice. We also define a terminal cost on the state, $V_f(x) = 1/2x'Px, P > 0$. An important feature of the MPC online optimization problem is the terminal constraint (2.3). The set $\mathbb{X}_f \subseteq \mathbb{X}$ is the terminal set.

The MPC online optimization problem is now defined as:

$$\mathbb{P}_N(x) : \min_{\mathbf{u}} V_N(x, \mathbf{u})$$

$$\text{s.t. } x(j+1) = Ax(j) + Bu(j), \qquad j = \{0,1,2,\dots,N-1\}$$

$$x(j) \in \mathbb{X} \qquad j = \{0,1,2,\dots,N\}$$

$$u(j) \in \mathbb{U} \qquad j = \{0,1,2,\dots,N-1\} \tag{2.2}$$

$$x(0) = x$$

$$x(N) \in \mathbb{X}_f \tag{2.3}$$

In the optimization problem $\mathbb{P}_N(x)$, the cost function $V_N(x,\mathbf{u})$ is given by:

$$V_N(x,\mathbf{u}) = \sum_{j=0}^{N-1} \ell(x(j), u(j)) + V_f(x(N)) \tag{2.4}$$

The set $\mathbb{Z}_N$ is defined as the set of $(x,\mathbf{u})$ for which the problem $\mathbb{P}_N(x)$ is feasible. That is,

$$\mathbb{Z}_N := \left\{(x,\mathbf{u}) \mid \phi(j;x,\mathbf{u}) \in \mathbb{X}, \phi(N;x,\mathbf{u}) \in \mathbb{X}_f, \mathbf{u} \in \mathbb{U}^N\right\} \tag{2.5}$$

The projection of set $\mathbb{Z}_N$ onto $\mathbb{X}$ is the set of admissible states, denoted by $\mathcal{X}_N$. That is,

$$\mathcal{X}_N := \left\{x \mid \exists \mathbf{u} \in \mathbb{U}^N, \text{ s.t } (x,\mathbf{u}) \in \mathbb{Z}_N\right\} \tag{2.6}$$

For a given $x \in \mathcal{X}_N$, the set of feasible inputs is given by $\mathcal{U}_N(x)$:

$$\mathcal{U}_N(x) := \{\mathbf{u} \mid (x,\mathbf{u}) \in \mathbb{Z}_N\} \tag{2.7}$$

The online optimization problem can now succinctly be expressed as:

$$\mathbb{P}_N(x) := \min_{\mathbf{u}} V_N(x,\mathbf{u}) \qquad \text{s.t. } \mathbf{u} \in \mathcal{U}_N(x)$$

2.2.2 Optimal MPC

The following assumptions are made on the system:

Assumption 1. *The centralized system* (A, B) *is stabilizable.*

Assumption 2. *The cost functions* $\ell(x, u)$ *and* $V_f(x)$ *are positive definite* [1]

Assumption 3. *The set* $\mathbb{X}_f$ *and the costs* $\ell(x, u), V_f(x)$ *are chosen such that there exists a terminal controller* $u = \kappa_f(x)$ *that satisfies:*

$$V_f(Ax + B\kappa_f(x)) - V_f(x) \leq -\ell(x, \kappa_f(x)) \qquad \forall x \in \mathbb{X}_f \tag{2.8}$$

$$Ax + B\kappa_f(x) \in \mathbb{X}_f, \kappa_f(x) \in \mathbb{U} \qquad \forall x \in \mathbb{X}_f \tag{2.9}$$

Assumption 4. *The set* $\mathbb{U}$ *is convex, closed and compact and contains the origin in its interior. The set* $\mathbb{X}$ *is convex, closed and contains the origin in its interior. The set* $\mathbb{X}_f$ *is convex, closed, compact and contains the origin in its interior.*

Remark 5. The choice of quadratic stage and terminal costs with $Q > 0, R > 0, P > 0$ automatically satisfies Assumption 2.

Remark 6. From Assumption 1, we know that there exists a linear feedback K such that $(A+BK)$ is stable. In other words, the closed-loop $x^+ = (A + BK)x$ is stable. We choose such a K as the terminal controller $\kappa_f(x)$. The terminal penalty $V_f(x) = x'Px$ is chosen as the solution to the Lyapunov equation (which exists as a consequence of Assumption 2):

$$(A + BK)'P(A + BK) + (Q + K'RK) = P$$

For the pair (P, K), we can define the control invariant region in the state-space in which $u = Kx$ does not activate any constraints as:

$$\mathbb{X}_f := \left\{x \mid x(i) = (A + BK)^i x \in \mathbb{X}_f \subseteq \mathbb{X}, Kx(i) \in \mathbb{U}, \forall i \geq 0\right\}$$

For linear systems, such sets can be easily constructed. See Gilbert and Tan (1991) for an algorithm.

---

[1] A function $f(x)$ is positive definite if $f(x) \geq 0 \forall x$ and $f(x) = 0$ if and only if $x = 0$.

The optimal solution to the optimization problem (2.2) is denoted by $\mathbf{u}^0(x)$ and the optimal objective value is given by $V_N^0(x)$. The optimal-MPC control law is now defined as $\kappa_o(x) = u^0(0;x)$ in which $u^0(0;x)$ is the first input in the optimal sequence $\mathbf{u}^0(x)$. The closed-loop evolution, under the control law $\kappa_0(x)$ is $x^+ = Ax + B\kappa_0(x)$. The centralized optimal MPC asymptotic (exponential) stability theorem is presented below. This theorem is attributed to Rawlings and Mayne (2009, Thm 2.24(b), Chap. 2).

Theorem 7 (Optimal MPC stabilty)**.** *Let Assumptions 1–4 hold. Then the origin is exponentially stable with a region of attraction $\mathcal{X}_N$ for the system $x^+ = Ax + B\kappa_0(x)$. If $\mathcal{X}_N$ is unbounded, then the region of attraction is any sublevel set of $V_N^0(\cdot)$.*

The detailed technical proof for Theorem 7 is provided in Rawlings and Mayne (2009, Chap. 2). We provide an sketch of the proof below for linear systems with positive definite stage cost. The stability proof follows by establishing that $V_N^0(\cdot)$ is a Lyapunov function for the closed-loop dynamics $x^+ = Ax + B\kappa_0(x)$.

Lyapunov stability theorem for a dynamic system $z^+ = f(z)$ states that if a function $V(z)$ exists with the following properties

$$V(z) \geq \alpha_1(|z|), \qquad \forall z \in \mathcal{Z} \tag{2.10}$$

$$V(z) \leq \alpha_2(|z|), \qquad \forall z \in \mathcal{Z} \tag{2.11}$$

$$V(z^+) - V(z) \leq -\alpha_3(|z|) \qquad \forall z \in \mathcal{Z} \tag{2.12}$$

in which $\alpha_i(\cdot), i \in \{1,2,3\}$ are $\mathcal{K}_\infty$ functions[2]; then the origin is asymptotically stable on the set $\mathcal{Z}$. Converse Lyapunov theorem states that if the dynamic system is asymptotically stable, then there exists a Lyapunov function for that system. If the $\mathcal{K}_\infty$ functions α_i are of the form $\lambda_i|x|^\sigma, \lambda_i, \sigma > 0$, then the dynamic system is exponentially stable (see Rawlings and Mayne (2009, Appendix B.) for precise statements).

[2]A function $\sigma : \mathbb{R}_+ \to \mathbb{R}_+$ belongs to the class of $\mathcal{K}_\infty$ functions if σ is continuous, strictly increasing, $\sigma(0) = 0$ and $\sigma(s) \to \infty$ as $s \to \infty$.

To show that $V_N^0(\cdot)$ is a Lyapunov function for the linear system under study [3], we first define the warm start as follows:

Definition 8 (Warm Start)**.** Let $(x, \mathbf{u})$ be a state-input vector pair such that $(x, \mathbf{u}) \in \mathbb{Z}_N$. Then the warm start for the successor initial state $x^+ = Ax + B\mathbf{u}(0; x)$ is defined as:

$$\tilde{\mathbf{u}} = (\mathbf{u}(1; x), \mathbf{u}(2; x), \ldots, \mathbf{u}(N; x), u_+)$$

in which $u_+ = \kappa_f(\phi(N; x, \mathbf{u}))$.

The lower bound (2.10) for the optimal function is established by using the fact that we choose $Q, R, P > 0$ (Assumption 2). By this choice of Q, R, P, $V_N(x, \mathbf{u})$ is positive definite, and hence $V_N^0(x) \geq \ell(x, \kappa_0(x))$. Since $\ell(x, u) = 1/2(x'Qx + u'Ru)$, we have that $\ell(x, u) \geq 1/2x'Qx \geq 1/2\underline{\lambda}_Q|x|^2$. The last inequality follows from the positive definiteness of Q with $\underline{\lambda}_Q > 0$ denoting the smallest eigen-value of Q. We denote the smallest and largets eigen-value of a matrix $\mathscr{H}$ by $\underline{\lambda}_{\mathscr{H}}$ and $\overline{\lambda}_{\mathscr{H}}$ respectively.

Following the definition of the warm start, given an optimal input sequence $\mathbf{u}^0(x)$, the warm start $\tilde{\mathbf{u}}^0$ is feasible for the successor state $x^+ = Ax + B\kappa_o(x)$, because $x(N) = \phi(N; x, \mathbf{u}^0)$ belongs to $\mathbb{X}_f$ (and hence $Ax(N) + B\kappa_f(x(N)) \in \mathbb{X}_f$ by Assumption 3). Therefore, we get the following inequality that establishes the cost-drop property in Equation (2.12).

$$V_N(x^+, \tilde{\mathbf{u}}^0) = V_N^0(x) + \underbrace{\big(V_f(Ax(N) + B\kappa_f(x(N))) + \ell(x(N), \kappa_f(x(N))) - V_f(x(N))\big)}_{\leq 0 \text{ by Assumption 3}} - \underbrace{\ell(x, \kappa_0(x))}_{\geq 0 \text{ by Assumption 2}}$$

$$V_N^0(x^+) \leq V_N(x^+, \tilde{\mathbf{u}}^0) \leq V_N^0(x) - \underline{\lambda}_Q|x|^2$$

The upper bound (2.11) is established by showing that $V_N^0(x) \leq V_f(x) \leq \overline{\lambda}_P|x|^2, \forall x \in \mathbb{X}_f$. To do so, consider $x \in \mathbb{X}_f$ and choose $u(0) = \kappa_f(x)$. Therefore, $x(1) = Ax + Bu(0)$ satisfies $V_f(x(1)) + \ell(x, u(0)) \leq V_f(x)$. Since Assumption 3 is satisfied, we can choose $u(1) = \kappa_f(x(1))$ to obtain $x(2) = Ax(1) + Bu(1)$. Therefore $V_f(x(2)) + \ell(x(1), u(1)) \leq V_f(x(1))$. So, we can conclude that $V_f(x(2)) + \ell(x(1), u(1)) + \ell(x, u(0)) \leq V_f(x(1)) + \ell(x, u(0)) \leq V_f(x)$, using the first inequality. In such a manner, we can construct an input sequence $\mathbf{u}_{\kappa_f}(x) := \{u(j) = \kappa_f(x(j))\}$,

[3] refer to Rawlings and Mayne (2009, Chap 2.) for more general cases

so that $V_N(x, \mathbf{u}_{\kappa_f}) \leq V_f(x)$. Since $\mathbf{u}_{\kappa_f}$ is a feasible input sequence, the optimal cost function $V_N(x, \mathbf{u}^0) \leq V_N(x, \mathbf{u}_{\kappa_f}) \leq V_f(x), \forall x \in \mathbb{X}_f$ (Pannocchia, Rawlings, and Wright, 2011). The upper bound is extended to $\mathcal{X}_N$ using the compactness of $\mathbb{X}_f$ (Rawlings and Mayne, 2009, Proposition 2.18).

2.2.3 Suboptimal MPC

The favorable properties of the closed-loop was established in Optimal MPC based on the optimal value function. However, in many practical applications, we might not be able to solve the optimization problem (2.2) to optimality. We might not be able to solve to optimality in the given sample time (for large problems and small sampling time) or by design (like for example, in cooperative MPC, as we show in Section 2.3.3). Hence, it is important that asymptotic stability be ensured when the online optimizations do not converge to the optimal solution. Suboptimal MPC theory is used to establish this property.

Given any feasible input sequence $\mathbf{u} \in \mathcal{U}_N(x)$ for the state x, the warm start is defined according to Definition 8, and the successor input set for the state $x^+ = Ax + Bu(0;x)$ is defined as below:

Definition 9 (Successor input set)**.** Consider $(x, \mathbf{u})$ such that $\mathbf{u}$ is feasible for $\mathbb{P}_N(x)$ (2.2). For the successor state $x^+ = Ax + Bu(0;x)$, we define the set $G(x, \mathbf{u})$

$$G(x,\mathbf{u}) = \{\mathbf{u}^+ \mid \mathbf{u}^+ \in \mathcal{U}_N(x^+), V_N(x^+,\mathbf{u}^+) \leq V_N(x,\tilde{\mathbf{u}}), V_N(x^+,\mathbf{u}^+) \leq V_f(x^+) \text{ if } x \in \mathcal{B}_r \subset \mathbb{X}_f\} \quad (2.13)$$

in which $\tilde{\mathbf{u}}$ is the warm start given by Definition 8 and $\mathcal{B}_r$ is a ball of radius $r > 0$. We choose r sufficiently small such that $\mathcal{B}_r$ is a subset of the terminal region.

Similar to optimal MPC, we inject the first input from the suboptimal sequence to the plant. The control law in the case of suboptimal MPC is therefore a set, as any input sequence in the successor input set can be used. The closed-loop analysis, consequently is on the evolution of the following system

$$x^+ = Ax + B\kappa_s(x) \tag{2.14}$$

$$\mathbf{u}^+ \in G(x, \mathbf{u}) \tag{2.15}$$

in which $\kappa_s(x)$ is the control law given by the first input in the input sequence $\mathbf{u}(x)$. The following theorem, attributed to Pannocchia et al. (2011) establishes the exponential stability of suboptimal MPC. Additionally, we make the following assumptions on the cost function $V_N(x, \mathbf{u})$.

Assumption 10. *There exist positive constants a, a_1', a_2', a_f and r, such that the cost function $V_N(x, \mathbf{u})$ satisfies:*

$$\begin{aligned} \ell(x,u) &\geq a_1' |(x,u)|^a && (x,u) \in \mathbb{X} \times \mathbb{U} \\ V_N(x,\mathbf{u}) &\leq a_2' |(x,\mathbf{u})|^a && (x,u) \in \mathscr{B}_r \\ V_f(x) &\leq a_f |x|^a && x \in \mathbb{X} \end{aligned}$$

in which $\mathscr{B}_r$ is the ball of radius r.

Note that it is easy to show that Assumption 10 is satisfied for linear systems and quadratic costs.

Theorem 11. *Let Assumptions 1 – 4 and 10 hold. For any x for which $\mathscr{U}_N(x)$ is not empty, choose $\mathbf{u} \in \mathscr{U}_N(x)$. Then, the origin of the closed-loop system* (2.14)–(2.15) *is asymptotically stable on (arbitrarily large) compact subsets of the feasible region $\mathscr{X}_N$*

We now provide a sketch of the proof for Theorem 11 for linear systems with quadratic, positive-definite stage costs. We refer the reader to Pannocchia et al. (2011) for the detailed proof for a more general case. Since in suboptimal MPC, there are multiple input sequences that satisfy (2.13), the closed-loop dynamics follows a difference inclusion, instead of a difference equation. Using the notation $z = (x, \mathbf{u})$ (called as the extended state), the closed-loop (2.14)–(2.15) can be succinctly written as:

$$z^+ \in H(z) := \left\{ (x^+, \mathbf{u}^+) \mid x^+ \in Ax + B\kappa_s(x), \mathbf{u}^+ \in G(z) \right\}$$

Analogous to the Lyapunov function described in the previous section; we can write a Lyapunov function for the difference inclusion. $V(z)$ is an exponential Lyapunov function for $z^+ \in H(z)$ on the set $\mathcal{Z}$ if the following hold, with $a_1, a_2, a_3, a \geq 0$ (Pannocchia et al., 2011, Definition 13).

$$V(z) \geq a_1|z|^a, \qquad \forall z \in \mathcal{Z} \tag{2.16}$$

$$V(z) \leq a_2|z|^a, \qquad \forall z \in \mathcal{Z} \tag{2.17}$$

$$\max_{z^+ \in H(z)} V(z^+) - V(z) \leq -a_3|z|^a, \qquad \forall z \in \mathcal{Z} \tag{2.18}$$

Exponential stability is established by showing that $V_N(x, \mathbf{u})$ is a Lyapunov function for the difference inclusion $z^+ \in H(z)$. To show that the cost function $V_N(x, \mathbf{u})$ satisfies (2.16)–(2.17), we proceed by noting that the cost function can be written as:

$$V_N(x, \mathbf{u}) = \frac{1}{2}\begin{bmatrix} x \\ \mathbf{u} \end{bmatrix}' \mathcal{H} \begin{bmatrix} x \\ \mathbf{u} \end{bmatrix} \qquad \mathcal{H} = \begin{bmatrix} \mathcal{A}'\mathcal{Q}\mathcal{A} & \mathcal{A}'\mathcal{Q}\mathcal{B} \\ \mathcal{B}'\mathcal{Q}\mathcal{A} & \mathcal{B}\mathcal{Q}\mathcal{B} + \mathcal{R} \end{bmatrix} \tag{2.19}$$

in which

$$\begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N) \end{bmatrix} = \underbrace{\begin{bmatrix} I \\ A \\ \vdots \\ A^{N-1} \end{bmatrix}}_{\mathcal{A}} x + \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \dots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}}_{\mathcal{B}} \mathbf{u}$$

and $\mathcal{Q} = \text{diag}(\underbrace{Q, Q, \dots, Q}_{N-1 \text{ times}}, P)$ and $\mathcal{R} = \text{diag}(\underbrace{R, R, \dots, R}_{N \text{ times}})$. Since $Q, R > 0$, the matrix $\mathcal{H}$ is a positive definite matrix. Therefore, $1/2\underline{\lambda}_{\mathcal{H}}|x, \mathbf{u}|^2 \leq V_N(x, \mathbf{u}) \leq \overline{\lambda}_{\mathcal{H}}|x, \mathbf{u}|^2$ Thus, (2.16) and (2.17) are satisfied. To show the cost-drop property, notice that for $x \in \mathcal{B}_r \subset \mathbb{X}_f$, we have by the property of $G(x, \mathbf{u})$ that $V_N(x, \mathbf{u}) \leq V_f(x)$. As shown in the previous section, $V_f(x) = 1/2x'Px \leq \overline{\lambda}_P|x|^2$. Hence, we have that

$$\underline{\lambda}_{\mathcal{H}}|\mathbf{u}|^2 \leq \underline{\lambda}_{\mathcal{H}}|(x, \mathbf{u})|^2 \leq V_N(x, \mathbf{u}) \leq V_f(x) \leq \overline{\lambda}_P|x|^2, \qquad x \in \mathcal{B}_r \tag{2.20}$$

From the inequality (2.20), we can conclude that

$$|\mathbf{u}| \leq d|x|, \quad x \in \mathscr{B}_r \tag{2.21}$$

in which $d = \sqrt{\frac{\overline{\lambda}_P}{\underline{\lambda}_{\mathscr{H}}}}$. Using inequality (2.21), we can establish that

$$|(x,\mathbf{u})| \leq |x| + |\mathbf{u}| \leq (1+d)|x| \leq (1+d)|(x,u(0))| \qquad x \in \mathscr{B}_r \tag{2.22}$$

As we saw in the previous section, $V_N(x^+,\tilde{\mathbf{u}}) - V_N(x,\mathbf{u}) \leq -\ell(x,u(0))$ (by choice of the warm start). Since, $\mathbf{u}^+ \in G(x,\mathbf{u})$ implies that (i) $\mathbf{u}^+$ drives the state x^+ into the terminal region in N steps and, (ii) ensures that the cost of doing so is less than $V_N(x^+,\tilde{\mathbf{u}})$; we can conclude that

$$V_N(x^+,\mathbf{u}^+) - V_N(x,\mathbf{u}) \leq -\ell(x,u(0))$$

Note that the lower bound of $\ell(x,u(0))$ is given by

$$1/2\min(\underline{\lambda}_Q,\underline{\lambda}_R)|(x,u(0))|^2 \leq \ell(x,u(0))$$

Denote $1/2\min(\underline{\lambda}_Q,\underline{\lambda}_R) = a_1'$. Using the inequality (2.22), we can then conclude that

$$V_N(x^+,\mathbf{u}^+) - V_N(x,\mathbf{u}) \leq -\ell(x,u(0)) \leq -a_1'|(x,u(0))|^2 \leq \frac{-a_1'}{(1+d)^2}|x,\mathbf{u}|^2, \qquad \forall x \in \mathscr{B}_r$$

The cost-drop property can be extended to the region of attraction using compactness of $\mathbb{U}$.

The online optimization problem being solved for suboptimal MPC is slightly modified from that of optimal MPC. In Equation (2.23), we present the optimization problem for suboptimal MPC.

$$\mathbb{P}_N(x) := \min_{\mathbf{u}} V_N(x,\mathbf{u}) \qquad \text{s.t. } \mathbf{u} \in \mathscr{U}_N(x), |\mathbf{u}| \leq d|x|, \text{if } x \in \mathscr{B}_r \tag{2.23}$$

For future reference, we define the following set:

$$\mathscr{U}_N^s(x;r) := \{u \mid u \in \mathscr{U}_N(x), |\mathbf{u}| \leq d|x|, \text{if } x \in \mathscr{B}_r\} \tag{2.24}$$

Note that the constraint on $|\mathbf{u}|$ is not enforced in practical implementations as $r > 0$ can be chosen arbitrarily small.

The advantage of using suboptimal MPC is that the online optimizations need not converge; and we can inject any suboptimal iterate generated by the optimization algorithm into the plant as long as that iterate belongs to the set $G(x, \mathbf{u})$. Another important feature that stands out from the suboptimal MPC theory is that just using the warm start at every time ensures exponential stability. Online optimization improves the open-loop prediction cost [4].

2.3 Distributed MPC

In the previous sections, we introduced centralized MPC, in which a single controller is designed for the system. The centralized controller uses system-wide information about models, constraints on the inputs and states, and objective to find a control law that has favorable properties. In many practical applications, this centralized information is distributed among many agents. For example, a chemical plant may have multiple MPC controllers running, each of which is controlling one process in the facility. In such cases, it is important to study how to coordinate information spread among multiple controllers to better control the plant. Distributed MPC is the study of various architectures for information sharing and retrieval to coordinate multiple controllers (Scattolini, 2009).

In this section, we first introduce the models and objectives of each agent or node in the system in Section 2.3.1. In Section 2.3.2, we describe the so-called noncooperative MPC, in which the nodes share information regarding their future (predicted) input moves with each other. In Section 2.3.3, we present the cooperative MPC algorithm in which the nodes not only share information about their predicted input moves with each other, but they also share (and use) model and objective functions. We show that cooperative MPC is an implementation of suboptimal centralized MPC, and hence it inherits all the desirable properties of suboptimal (centralized) MPC. We present a simple two-tank system shown in Figure 2.2 in Section 2.3.4. We use this example to illustrate the key properties of distributed MPC algorithms, namely (i) noncooperative MPC can de-stabilize a plant, and (ii) with careful design, cooperative MPC can

[4] But we cannot say anything about the closed-loop cost if we stop at suboptimal iterates

stabilize any plant that can be stabilized using centralized MPC. We choose the two-tank system because its model is a system of integrators like the supply chain model (see Chapter 4).

2.3.1 Models, constraints and objective functions

In distributed MPC, the system is assumed to be composed of several subsystems (or agents or nodes). We use the index i to denote a subsystem, and M to denote the total number of subsystems. Each subsystem $i \in \{1,2,3,\ldots,M\}$ has the following dynamics and constraints:

$$x_i^+ = A_i x_i + B_{ii} u_i + \sum_{\substack{l\in\{1,2,\ldots,M\}\\ l\neq i}} B_{il} u_l \tag{2.25}$$

$$x_i \in \mathbb{X}_i \qquad u_i \in \mathbb{U}_i \tag{2.26}$$

in which x_i, u_i are the states and inputs in subsystem i.

The stage cost for a subsystem is given by:

$$\ell_i(x_i,u_i) = 1/2(x_i' Q_i x_i + u_i' R_i u_i) \tag{2.27}$$

with the penalties $Q_i, R_i > 0$.

The centralized (system-wide) model is therefore:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix}^+ = \underbrace{\begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_M \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} B_{11} & B_{12} & \vdots & B_{1M} \\ B_{21} & B_{22} & \ldots & B_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ B_{M1} & B_{M2} & \ldots & B_{MM} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \ldots \\ u_M \end{bmatrix}}_{u} \tag{2.28}$$

$$\mathbb{X} = \mathbb{X}_1 \times \mathbb{X}_2 \times \ldots \times \mathbb{X}_M \tag{2.29}$$

$$\mathbb{U} = \mathbb{U}_1 \times \mathbb{U}_2 \times \ldots \times \mathbb{U}_M \tag{2.30}$$

The centralized stage-cost is

$$\ell(x,u) = \sum_{i=1}^{M} \ell_i(x_i,u_i) \tag{2.31}$$

We do not make any special assumptions on the local models (2.25)–(2.26). The only assumptions made are on the centralized model and stage costs (2.28)–(2.31). We assume that the centralized model and stage costs satisfies Assumptions 1–4 and 10. It is important to note that the terminal controller $\kappa_f(\cdot)$, terminal cost $V_f(\cdot)$ and, terminal set $\mathbb{X}_f$ are all defined only for the centralized system.

2.3.2 Noncooperative MPC

In noncooperative MPC, each subsystem minimizes its local objective function. Therefore, the subsystem optimization problem $\mathbb{P}^i_N(x_i, \mathbf{v}_{-i})$ is given below. For subsystem i, we use $-i$ to denote all the other subsystems, i.e.. $-i = \{1, 2, \ldots, i-1, i+1, \ldots, M\}$.

$$\begin{aligned}
\mathbb{P}^i_{N,nc}(x_i; \mathbf{v}_{-i}) : \min_{\mathbf{u}_i} & \sum_{j=0}^{N-1} \ell_i(x_i(j), u_i(j)) + V_{f,i}(x_i(N)) \\
& \text{s.t. } x_i(j+1) = A_{ii}x_i(j) + B_{ii}u_i(j) + \sum_{\substack{l\in\{1,2,\ldots,M\}\\ l\neq i}} B_{il}v_l(j) \quad j = \{0, 1, \ldots, N-1\} \\
& x_i(j) \in \mathbb{X}_i \quad j = \{0, 1, \ldots, N-1\} \\
& u_i(j) \in \mathbb{U}_i \quad j = \{0, 1, \ldots, N-1\} \\
& x_i(0) = x_i
\end{aligned} \tag{2.32}$$

We wish to bring the readers attention to two important features of the "local" optimization problem $P^i_{N,nc}(x_i; \mathbf{v}_{-i})$, namely, (i) To make accurate predictions of $x_i(j)$, and hence the cost function, subsystem i needs to know the future (predicted) inputs of all other subsystems, and (ii) No terminal constraints are enforced [5].

In noncooperative MPC (Algorithm 1), each subsystem broadcasts its input sequence prediction. Based on this prediction, every subsystem solves its local optimization problem. The final input is a convex combination of the previously broadcast inputs and the optimized inputs.

In the inner loop of Algorithm 1, each subsystem is finding its *best response* to the other subsystem inputs. It can be shown that as $\bar{p} \to \infty$, the solutions $\mathbf{u}_i^{(p)}$ converge to a point called the Nash Equilibrium (Başar and Olsder, 1999). As has been shown earlier by Venkat (2006, Sec 4.3.1), the Nash equilibrium may be unstable. We also wish to point out that we cannot make

[5] Although we include a terminal penalty, we provide no design methods to find a terminal penalty

Data: Starting state $\{x_i(0)\}$, initial guess $(\tilde{\mathbf{u}}_1(0), \tilde{\mathbf{u}}_2(0), \ldots, \tilde{\mathbf{u}}_M(0))$, $\bar{p} \geq 0$, $\omega_i \in (0,1)$ such that $\sum_{i=0}^{M} \omega_i = 1$

Result: Closed loop $x(j), u(j),\ j = \{0, 1, 2, \ldots\}$

Set $j \leftarrow 0$

while $j \geq 0$ **do**

  Set $p \leftarrow 0$

  Set $x_i \leftarrow x_i(j))$, for i in $1, 2, \ldots, M$

  Set $\tilde{\mathbf{u}}_i \leftarrow \tilde{\mathbf{u}}_i(j)$, for i in $1, 2, \ldots, M$

  Broadcast $\tilde{\mathbf{u}}_i$ to all other subsystems for i in $1, 2, \ldots, M$

  Each subsystem i creates $\tilde{\mathbf{u}}_{-i}$ from the other subsystem inputs

  while $p < \bar{p}$ **do**

    Solve $\mathbb{P}^i_{N,nc}(x_i, \mathbf{u}_{-i})$ to obtain $\mathbf{u}_i^0$ for i in $1, 2, \ldots, M$

    Set $\mathbf{u}_i^{(p+1)} \leftarrow \omega_i \mathbf{u}_i^{(p)} + (1 - \omega_i)\mathbf{u}_i^0$ for i in $1, 2, \ldots, M$

    Set $p \leftarrow p + 1$

  end

  Set input $u_i(j) = \mathbf{u}_i(0)$ for i in $1, 2, \ldots, M$ and broadcast

  Evolve state from $x_i(k)$ to $x_i(k+1)$ for the input just obtained

  Obtain feasible input $\tilde{\mathbf{u}}_i(j+1)$ for i in $1, 2, \ldots, M$

  Set $j \leftarrow j + 1$

end

Algorithm 1: Noncooperative MPC

any claims about the feasibility of $\mathbb{P}_N^i(x_i; \mathbf{v}_{-i})$ in the inner optimization loop of the noncooperative MPC algorithm. For similar reasons, we cannot make any claims about how to obtain $\tilde{\mathbf{u}}_i(j+1)$.

In Figure 2.3, we show the unstable closed-loop response using Algorithm 1 for a simple two-tank example.

2.3.3 Cooperative MPC

In this section, we tailor the distributed MPC algorithm to be an implementation of suboptimal MPC. To do so, we require that the subsystems not only share input forecasts, but also model, constraints and objective functions with each other. Under this requirement, the cooperative MPC optimization problem is written as:

$$\mathbb{P}_N(x) := \min_{\mathbf{u}} V_N(x, \mathbf{u}) \qquad \text{s.t. } \mathbf{u} \in \mathcal{U}_N^s(x; r) \tag{2.33}$$

Note that the cooperative MPC optimization problem is the same as the suboptimal MPC optimization problem. The cooperative MPC algorithm is presented in Algorithm 2.

In the inner loop in Algorithm 2, all the subsystems are solving the same optimization problem, but by fixing the decisions of other sthe ubsystems at the values broadcast in the previous iterate. The inner loop is an implementation of the Jacobi parallel optimization routine (Bertsekas and Tsitsiklis, 1989, Section 3.3.5). For convex optimization problems, the Jacobi algorithm has the property that it generates feasible iterates with non-increasing objective function values. Before presenting the cooperative MPC stability theorem (which in different forms have been stated in Stewart, Venkat, Rawlings, Wright, and Pannocchia (2010); Venkat (2006); Stewart, Wright, and Rawlings (2011); Subramanian, Rawlings, and Maravelias (2012b)), we briefly review the properties of Jacobi algorithm for convex optimization problem.

Jacobi algorithm

For the sake of simplicity, we consider only two subsystems in this section. Consider the following optimization problem (2.34), solved using Algorithm 3.

Data: Starting state $x(0)$, initial guess $(\tilde{\mathbf{u}}_1(0), \tilde{\mathbf{u}}_2(0), \ldots, \tilde{\mathbf{u}}_M(0)) \in \mathcal{U}_N^s(x(0); r)$, $\bar{p} \geq 1$ and $\omega_i \in (0,1)$ such that $\sum_{i=0}^{M} \omega_i = 1$

Result: Closed loop $(x(j), u(j))$, $j = \{1, 2, \ldots\}$

set $j \leftarrow 0$

while $j \geq 0$ **do**

  Set $p \leftarrow 0$, $x \leftarrow x(j)$

  Set $\mathbf{u}_i^{(p)} \leftarrow \tilde{\mathbf{u}}_i(j)$ for $i = 1, 2, \ldots, M$

  Broadcast current subsystem inputs $\tilde{\mathbf{u}}_i(j)$ to other subsystems

  while $p < \bar{p}$ **do**

    Solve $\min_{\mathbf{u}_i} V_N(x, \mathbf{u}) \quad \text{s.t. } \mathbf{u} \in \mathcal{U}_N^s(x; r); \mathbf{u}_{-i} = \mathbf{u}_{-i}^{(p)}$ to obtain $\mathbf{u}_i^0$ for i in $1, 2, \ldots, M$

    Set $\mathbf{u}_i^{(p+1)} \leftarrow \omega_i \mathbf{u}_i^{(p)} + (1 - \omega_i)\mathbf{u}_i^0$ for i in $1, 2, \ldots, M$

    Set $p \leftarrow p + 1$

  end

  Set $\mathbf{u} \leftarrow (\mathbf{u}_1^{(p)}, \mathbf{u}_2^{(p)}, \ldots, \mathbf{u}_M^{(p)})$ and find $x(j+N) \leftarrow \phi(N; x(j), \mathbf{u})$

  Obtain $u_+ = (u_{1+}, u_{2+}, \ldots, u_{M+}) \leftarrow \kappa_f(x(j+N))$

  Obtain warm start $\tilde{\mathbf{u}}_i(j+1) = (\mathbf{u}_i^{(p)}(1), \mathbf{u}_i^{(p)}(2), \ldots, u_{i+})$ for $i = 1, 2, \ldots, M$.

  Set input as $u(j) = (\mathbf{u}_1^{(p)}(0), \mathbf{u}_2^{(p)}(0), \ldots, \mathbf{u}_M^{(p)}(0))$

  Evolve state from $x(j)$ to $x(j+1)$ under input $u(j)$

  Set $j \leftarrow j + 1$

end

Algorithm 2: Cooperative MPC

$$\mathbb{J}: \min_{y_1, y_2} J(y_1, y_2) \qquad \text{s.t. } (y_1, y_2) \in \Omega \tag{2.34}$$

in which $J(\cdot)$ is a convex function and Ω is a convex, closed and compact set.

Data: Starting guess $(y_1, y_2) \in \Omega$ and $\omega_i \in (0,1)$ such that $\sum_{i=0}^{2} \omega_i = 1$

Result: Sequence of feasible iterates $\{y^{(p)}\}$ and non-increasing objective function values $\{J(y^{(p)})\}$

Set $p \leftarrow 0$

while $p \geq 0$ **do**

Set $y_1^{(p)} \leftarrow y_1$, $y_2^{(p)} \leftarrow y_2$

Solve $\min_{y_1} J(y_1, y_2)$, s.t. $y_2 = y_2^{(p)}, (y_1, y_2) \in \Omega$ to obtain y_1^0.

Solve $\min_{y_2} J(y_1, y_2)$, s.t. $y_1 = y_1^{(p)}, (y_1, y_2) \in \Omega$ to obtain y_2^0.

Set $y_i^{(p+1)} \leftarrow \omega_i y_i^{(p)} + (1-\omega_i) y_i^0$ for i in 1, 2

Set $p \leftarrow p + 1$

end

Algorithm 3: Jacobi algorithm

The following Proposition establishes that the Jacobi algorithm generates feasible iterates that have non-increasing objective function values

Proposition 12. Let $J(y)$ be continuously differentiable and strongly convex[6] on the convex, closed and compact set Ω. Let Algorithm 3 be used to solve convex optimization problem (2.34) from an initial feasible point $y \in \Omega$. Then, (i) every iterate $y^{(p)}$ generated by the algorithm is feasible and , (ii) $J(y^{(p+1)}) \leq J(y^{(p)}), \forall p > 0$.

The proof is provided in Stewart et al. (2010).

While Proposition 12 is enough to establish stability of suboptimal MPC, we require the following proposition to establish that the optimizations in cooperative MPC converge to the optimal solution. Note that the in contrast to Proposition 12, we require a much stricter condition on the constraints in Proposition 13, namely that, the constraints be uncoupled.

[6]Strongly convex implies that $J(\lambda w + (1-\lambda)v) < \lambda J(w) + (1-\lambda) J(v), \forall \lambda \in (0,1)$

Proposition 13. Let $J(y)$ be continuously differentiable and strongly convex. Let $\Omega = \Omega_1 \times \Omega_2$, in which $y_1 \in \Omega_1$ and $y_2 \in \Omega_2$, with Ω_i convex, closed and compact. Then, as $p \to \infty$, the iterates $y^{(p)}$ converges to the y^o, in which y^o is the optimal solution to optimization problem (2.34)

This proof is provided in Stewart et al. (2010). Another proof is also provided in Bertsekas and Tsitsiklis (1989, Prop 3.9) for Gauss-Seidel algorithm which is closely related to the Jacobi algorithm.

We now present the exponential stability of the cooperative MPC algorithm.

Theorem 14. *Let Assumptions 1 – 4 and 10 hold. Choose* $r > 0$ *such that* $\mathscr{B}_r \subset \mathbb{X}_f$*. For any* x *for which* $\mathscr{U}_N^s(x;r)$ *is not empty, choose* $\mathbf{u} \in \mathscr{U}_N^s(x;r)$*. Then, the origin of the closed-loop system obtained by Algorithm 2 is exponentially stable. The region of attraction are (arbitrarily large) compact subsets of the feasible region*

$$\mathscr{X}_N := \left\{x \mid \exists u \in \mathbf{U}^N,\ s.t\, \mathscr{U}_N^s(x;r) \neq \emptyset\right\}$$

Proof. We show that the closed-loop system obtained by Algorithm 2 is an implementation of suboptimal MPC and use Theorem 11 to prove exponential stability.

We note that the optimization problem (2.33) has convex constraints and a strongly convex objective. By choice $Q, R, P > 0$, we know that the Hessian $\mathscr{H}$ (2.19) is positive definite, and hence strongly convex. From Assumption 4, the set $\mathbb{Z}_N$ is convex and hence the set $\mathscr{U}_N(x)$ is convex too. The set $\mathscr{U}_N^s(x;r)$ is the intersection of two convex sets. Hence, by Proposition 12, we know that if $(x, \tilde{\mathbf{u}})$ is feasible for (2.33) then (i) *all* the iterates generated by the inner loop in Algorithm 2 are feasible; implying $\mathbf{u}^{(p)} \in \mathscr{U}_N^s(x;r)$ and, (ii) the cost at iterate p is not greater than the cost achieved by $V_N(x, \tilde{\mathbf{u}})$; that is $V_N(x, \mathbf{u}^{(p)}) \le V_N(x, \tilde{\mathbf{u}})$.

By choice of $\tilde{\mathbf{u}}(0)$, we know that $(x(0), \tilde{\mathbf{u}}(0))$ is feasible for (2.33). Therefore, $(x(0), \mathbf{u}^{(p)}(0)) \in \mathbb{Z}_N$ with $V_N(x(0), \mathbf{u}^{(p)}(0)) \le V_N(x(0), \tilde{\mathbf{u}}(0))$. Since $\tilde{\mathbf{u}}(1)$ is the warm start constructed from $\mathbf{u}^{(p)}(0)$, we know that it is feasible and that $\tilde{\mathbf{u}}(1) \in G(x(0), \mathbf{u}^{(p)}(0))$. Therefore, by induction the closed-loop obtained by Algorithm 2 belongs to the family of closed-loop solutions for which we showed exponential stability in Theorem 11. □

The main difference between cooperative and noncooperative MPC is that in the inner optimization loop of cooperative MPC, all the subsystems minimize the centralized problem, but subject only to their inputs. Therefore, we were able to use warm start and properties of Jacobi algorithm to establish the cost-drop properties required to prove exponential stability.

The cooperative MPC algorithm has the following key properties:

1. The nominal closed-loop is exponentially stable.

2. The subsystems need to share models and objective functions with each other.

3. There is no coordinator. At the end of every cooperative MPC iteration, the subsystems just need to transfer information regarding their predicted inputs with each other.

4. There is no minimum number of iterations of the inner-loop that is required. Subsystems can choose to stop after any number of iterations.

While, we used Proposition 12 to establish recursive feasibility and cost-drop of cooperative MPC, we cannot use Proposition 13 to establish that the as $p \to \infty$, the solution obtained by the cooperative MPC optimizations converge to the centralized MPC optimization problem because the constraint set $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M) \in \mathcal{U}_N^s(x; r)$ is a coupled constraint. The optimization problem (2.33) can be written as

$$\begin{aligned}
\mathbb{P}_N(x) : 1/2 \begin{bmatrix} x \\ \mathbf{u} \end{bmatrix}' \mathcal{H} \begin{bmatrix} x \\ \mathbf{u} \end{bmatrix} & \\
\text{s.t. } x(j) \in \mathbb{X} & \qquad j \in 0, 1, \ldots, N-1 \\
u(j) \in \mathbb{U} & \qquad j \in 0, 1, \ldots, N-1 \\
\begin{bmatrix} A^{N-1}B & A^{N-2}B & \ldots & B \end{bmatrix} \mathbf{u} \in \mathbb{X}_f & \\
|\mathbf{u}| \leq d|x| & \qquad x \in \mathcal{B}_r
\end{aligned} \tag{2.35}$$

In the following sections, we describe methods to “decouple” the constraint set, so that, in addition to establishing stability, we can also establish that the inner-loop optimizations converge to the centralized optimal solution. Hence, we can establish that centralized optimal and cooperative control have the same feedback solution.

In order to do so, we make the following assumptions for cooperative MPC:

Assumption 15. *There are no state constraints. State constraints are enforced as soft-penalties by tuning the Q matrix in the stage cost.*

Assumption 16. *The input constraint space is uncoupled. That is, the input constraint set $\mathbb{U}$ is the Cartesian product of the input constraint sets of each subsystem.*

$$\mathbb{U} = \mathbb{U}_1 \times \mathbb{U}_2 \times \ldots \times \mathbb{U}_M$$

The sets $\mathbb{U}_i$ are convex, closed, compact and, contain the origin in its interior.

Using Assumption 15, the state constraints in (2.35) are removed. Assumption 16 ensures that the input constraints for subsystem $i, i = 1, 2, \ldots, M$ do not affect any other subsystems' input constraints. Although, for most practical applications, we do nor require the constraint $|\mathbf{u}| \leq d|x|$, we can easily separate the constraint by enforcing $|\mathbf{u}_i| \leq d_i|x|$ such that $\sum_i d_i \leq d$.

Stability requirements mean that we cannot remove the stability constraint $x(N) \in \mathbb{X}_f$ from the optimization problem. In the next two sections, we briefly review two techniques to “uncouple” the terminal region constraint. The advantage of using the cooperative MPC formulations without coupled terminal constraints is that we have a “performance guarantee” that the open-loop cost attained by the cooperative MPC algorithm if the inner-loop was allowed to converge would be the optimal centralized open-loop cost.

2.3.3.1 Sub-states

This relaxation was proposed by Stewart et al. (2010), to solve the terminal equality constraint problem. That is $\mathbb{X}_f = \{0\}$. The centralized optimization problem is [7]:

$$
\begin{aligned}
\mathbb{P}_N(x) : \min_{\mathbf{u}} & \; V_N(x, \mathbf{u}) \\
\text{s.t. } & x_i(j+1) = A_i x_i(j) + B_{ii} u_i(j) + \sum_{\substack{l \in \{1,2,\ldots,M\} \\ l \neq i}} B_{il} u_l(j), \quad j = \{0,1,2,\ldots,N-1\}, i = \{1,2,\ldots,M\} \\
& u_i(j) \in \mathbb{U}_i \qquad j = \{0,1,2,\ldots,N-1\}, i = \{1,2,\ldots,M\} \\
& x_i(0) = x_i \qquad i = \{1,2,\ldots,M\} \\
& x_i(N) = 0 \qquad i = \{1,2,\ldots,M\}
\end{aligned}
\tag{2.36}
$$

Notice that the only coupled constraint is $x_i(N) = 0$, as the dynamics $x^+ = Ax + Bu$ can be projected out, i.e., we use the optimization problem formulation (2.35).

We consider a non-minimal realization of the system (2.25) such that "sub-state" x_{il} in subsystem i is only influenced by input l.

$$
\tilde{x}_{il}^+ = \tilde{A}_{il} \tilde{x}_{il} + \tilde{B}_{il} u_l \tag{2.37}
$$

[7] To simplify the discussion, we enforce the constraint that all the states are zero at the end of the horizon. In Stewart et al. (2010), only the unstable states were forced to be zero at the end of the horizon

Defining $\hat{x}_i = \left[\{\tilde{x}_{il}\}, l = 1,2,\ldots,M\right]$, each subsystem model is given by (2.25). The matrices $\hat{A}_i, \hat{B}_{il}$ are used to describe the dynamics in subsystem i.

$$\underbrace{\begin{bmatrix} \tilde{x}_{i1} \\ \tilde{x}_{i2} \\ \vdots \\ \tilde{x}_{iM} \end{bmatrix}^+}_{\hat{x}_i^+} = \underbrace{\begin{bmatrix} \tilde{A}_{i1} & & & \\ & \tilde{A}_{i2} & & \\ & & \ddots & \\ & & & \tilde{A}_{iM} \end{bmatrix}}_{\hat{A}_i} \underbrace{\begin{bmatrix} \tilde{x}_{i1} \\ \tilde{x}_{i2} \\ \vdots \\ \tilde{x}_{iM} \end{bmatrix}}_{\hat{x}_i} + \underbrace{\begin{bmatrix} \tilde{B}_{i1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\hat{B}_{i1}} u_1 + \underbrace{\begin{bmatrix} 0 \\ \tilde{B}_{i2} \\ \vdots \\ 0 \end{bmatrix}}_{\hat{B}_{i2}} u_2 + \ldots \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ \tilde{B}_{iM} \end{bmatrix}}_{\hat{B}_{iM}} u_M \tag{2.38}$$

$$x_i = \underbrace{\begin{bmatrix} \tilde{C}_{i1} & \tilde{C}_{i2} & \ldots & \tilde{C}_{iM} \end{bmatrix}}_{\hat{C}_i} \underbrace{\begin{bmatrix} \tilde{x}_{i1} \\ \tilde{x}_{i2} \\ \vdots \\ \tilde{x}_{iM} \end{bmatrix}}_{\hat{x}_i} \tag{2.39}$$

We assume that the centralized states x_i can be constructed from the sub-states $\hat{x}_i$. In general, we would require that the outputs measured in subsystem i, y_i can be reconstructed from the sub-states (see Assumption 17).

The subsystem stage cost $\ell_i(x_i, u_i) = 1/2(x_i' Q_i x_i + u_i R_i u_i)$ can now be written as

$$\ell_i(x_i, u_i) = \ell_i(\hat{x}_i, u_i) = 1/2(\hat{x}_i \hat{C}_i' Q \hat{C}_i \hat{x}_i + u_i R_i u_i)$$

The centralized stage cost is $\ell(x, u) = \sum_{i=1}^{M} \ell_i(\hat{x}_i, u_i)$. Since, we use a terminal constraint that all states are zero at the end of the horizon, we do not need a terminal penalty.

We define $\underline{x}_l$ as the sub-states that are affected by input l. That is $\underline{x}_l = \left[\{\tilde{x}_{il}\}\ i = \{1,2,\ldots,M\}\right]$. Correspondingly, we define $\underline{A}_l, \underline{B}_l$ as follows:

$$\underbrace{\begin{bmatrix} \tilde{x}_{1l} \\ \tilde{x}_{2l} \\ \vdots \\ \tilde{x}_{Ml} \end{bmatrix}^+}_{\underline{x}_l^+} = \underbrace{\begin{bmatrix} \tilde{A}_{1l} & & & \\ & \tilde{A}_{2l} & & \\ & & \ddots & \\ & & & \tilde{A}_{Ml} \end{bmatrix}}_{\underline{A}_l} \underbrace{\begin{bmatrix} \tilde{x}_{1l} \\ \tilde{x}_{2l} \\ \vdots \\ \tilde{x}_{Ml} \end{bmatrix}}_{\underline{x}} + \underbrace{\begin{bmatrix} \tilde{B}_{1l} \\ \tilde{B}_{2l} \\ \vdots \\ \tilde{B}_{Ml} \end{bmatrix}}_{\underline{B}_l} u_l \tag{2.40}$$

The constraint $x_i(N) = 0$ can be equivalently written as $\tilde{x}_{il} = 0, l \in \{1, 2, \ldots, M\}$. Therefore, the centralized MPC problem can be written as:

$$
\begin{aligned}
&\mathbb{P}_N(x) : \min_{\mathbf{u}} V_N(x, \mathbf{u}) \\
&\qquad \text{s.t. } \underline{x}_l(j+1) = \underline{A} x_l(j) + \underline{B}_l(j) u_l(j) && j = \{0, 1, 2, \ldots, N-1\}, l = \{1, 2, \ldots, M\} \\
&\qquad x_i(j) = \hat{C}_i \hat{x}_i(j) && j = \{1, 2, \ldots, N\}, i = \{1, 2, \ldots, M\} \\
&\qquad u_l(j) \in \mathbb{U} && j = \{0, 1, 2, \ldots, N-1\}, l = \{1, 2, \ldots, M\} \\
&\qquad x_i(0) = x_i && i = \{1, 2, \ldots, M\} \\
&\qquad \underline{x}_l(N) = 0 && l = \{1, 2, \ldots, M\}
\end{aligned}
\tag{2.41}
$$

In the optimization problem the terminal condition is uncoupled because the dynamics of state $\underline{x}_l$ depends only on input u_l via equation (2.40). If Assumption 17 is satisfied, then Algorithm 2 can be used to stabilize the plant *with guaranteed performance* to the centralized solution of the problem (2.36) as there are no coupled constraints in the problem.

Assumption 17 ensures that (i) the input $\mathbf{u}_i$ can be used to zero all the states that u_i influences, (ii) all the substates can be reconstructed from the outputs.

Assumption 17 (Subsystem stabilizability)**.**

- *The system* $(\underline{A}_i, \underline{B}_i)$ *is stabilizable.*
- *The system* $(\hat{A}_i, \hat{C}_i)$ *is detectable.*

We now discuss the features of the aforementioned decomposition:

Applicability. The decomposition into the substate models (2.25) can be obtained from the Kalman decomposition of the original input/output y_i, u_i model (Antsaklis and Michel, 1997, p.270).

The drawback however, is that not all centralized stabilizable models have a corresponding substate non-minimal realization that satisfies Assumption 17. One such example is the system of integrators (like supply chain models).

Convergence. The decomposition ensures that if the inner optimization loop in Algorithm 2 is allowed to converge, then the solution is the centralized solution to the optimization problem (2.36).

Initialization. The optimization loop in Algorithm 2 requires an feasible starting point. The feasible starting point is ensured by the warm start. The warm start is feasible because we assume that the actual plant state at the next sampling time is equal to the model prediction for the next state. In many cases, when there are plant-model mismatches or unmodeled disturbances affecting the system, this assumption that the plant state is equal to the predicted state might break-down. In such cases, the warm start becomes infeasible and we need an distributed initialization routine.

In the substate decomposition of the model, the warm start is infeasible if $\tilde{x}_{il}(N; x, \mathbf{u}) \neq 0$ for some i, l. However, since $\tilde{x}_{il}(N; x, \mathbf{u})$ depends only on the input sequence from a single subsystem $\mathbf{u}_l$, the re-initialization routine is also decoupled.

2.3.3.2 Relaxing the terminal region

This relaxation was proposed by Rawlings, Stewart, Wright, and Mayne (2010), and has been used for nonlinear suboptimal/ distributed / economic MPC in Stewart et al. (2011), Pannocchia et al. (2011), Amrit et al. (2011) as well as linear MPC in Subramanian et al. (2012b).

In this section, we work with the centralized model (2.25). The idea here is to develop an optimization problem without terminal constraints such that every iterate generated in the inner optimization loop of Algorithm 2 lies inside a terminal region that satisfies Assumption 3. To do so, we modify (i) the terminal region (ii) the cost function and (iii) the feasible set as follows:

The terminal region is chosen as a sublevel set of the terminal cost. That is,

$$\mathbb{X}_f := \left\{ x \mid V_f(x) \leq a, a > 0 \right\} \tag{2.42}$$

For linear systems, we could use P, the solution to the Lyapunov equation as $V_f(x)$ and choose a such that all the requirements in Assumption 3 are satisfied.

The cost function is modified so that the terminal penalty is magnified. That is,

$$V_N^\beta(x,\mathbf{u}) = \sum_{j=0}^{N-1} \ell(x(i),u(i)) + \beta V_f(x(N)) \tag{2.43}$$

in which $\beta \geq 1$.

Finally, the feasible set is modified as follows:

$$\mathbb{Z}_N^\beta = \left\{(x,\mathbf{u}) \mid V_N^\beta(x,\mathbf{u}) \leq \bar{V}, \mathbf{u} \in \mathbb{U}^N\right\} \tag{2.44}$$

in which $\bar{V} \geq 0$ can be chosen arbitrarily large.

In Proposition 18, we show how the parameter β can be chosen so that if $(x,\mathbf{u}) \in \mathbb{Z}_N^\beta$, then $x(N;x,\mathbf{u}) \in \mathbb{X}_f$. Using Proposition 19 [8], we can establish that the warm start given by Definition 8 satisfies the cost-drop property, i.e., $V_N^\beta(x,\tilde{\mathbf{u}}) \leq V_N^\beta(x,\mathbf{u})$.

Proposition 18. Let Assumption 2 hold. Define the terminal region $\mathbb{X}_f$ according to (2.42) Let the cost function $V_N^\beta(x,\mathbf{u})$ be given by (2.43). For $\bar{V} \geq a$, define $\bar{\beta} := \bar{V}/a$. Then, for any $\beta \geq \bar{\beta}$ and $(x,\mathbf{u}) \in \mathbb{Z}_N^\beta$ (2.44), we have that $\phi(N;x,\mathbf{u}) \in \mathbb{X}_f$.

Proof. For sake of contradiction, assume that $(x,\mathbf{u}) \in \mathbb{Z}_N^\beta$, $\beta \geq \bar{\beta}$, but $\phi(N;x,\mathbf{u}) \notin \mathbb{X}_f$, that is $V_f\left(\phi(N;x,\mathbf{u})\right) > a$. Since $(x,\mathbf{u}) \in \mathbb{Z}_N^\beta$, we know that

$$V_N^\beta(x,\mathbf{u}) = \sum_{i=0}^{N-1} \ell(\phi(i;x,\mathbf{u}),u(i)) + \beta V_f\left(\phi(N;x,\mathbf{u})\right) \leq \bar{V}$$

From Assumption 2, we know that $\ell(x,u) \geq 0$, which implies that

$$\beta V_f(\phi(N;x,\mathbf{u})) \leq \bar{V}$$

Since $\beta \geq \bar{\beta} = \frac{\bar{V}}{a}$,

$$\frac{\bar{V}}{a} V_f\left(\phi(N;x,\mathbf{u})\right) \leq \bar{V}$$

which implies that $V_f\left(\phi(N;x,\mathbf{u})\right) \leq a$, which is a contradiction. Therefore, for $\beta \geq \bar{\beta}$, if $(x,\mathbf{u}) \in \mathbb{Z}_N^\beta$, then $\phi(N;x,\mathbf{u}) \in \mathbb{X}_f$.

□

[8] see also Rawlings and Mayne (2009, Exercise 2.11, Page 177)

Proposition 19. Let Assumption 2 hold. Choose terminal set $\mathbb{X}_f$ according to 2.42, such that it satisfies Assumption 3. Then for every $\bar{\beta} \geq 1$, any $x \in \mathbb{X}_f$, and $u = \kappa_f(x)$, the following holds:

$$\beta V_f(Ax + B\kappa_f(x)) + \ell(x, \kappa_f(x)) \leq \beta V_f(x)$$

Proof. From Assumption 3, we know that

$$V_f(Ax + B\kappa_f(x)) + \ell(x, \kappa_f(x)) \leq V_f(x)$$

Hence:

$$\beta V_f(Ax + B\kappa_f(x)) + \beta\ell(x, \kappa_f(x)) \leq \beta V_f(x)$$

From Assumption 2, $\ell(x, \kappa_f(x)) \geq 0$, $V_f(\cdot) \geq 0$. Hence $\beta\ell(x, \kappa_f(x)) \geq \ell(x, \kappa_f(x))$ and the result follows. □

Proposition 20. Let Assumptions 2–3 hold, with the $\mathbb{X}_f$ chosen according to (2.42). Choose $\bar{V} \geq a$, $\beta \geq \bar{\beta} = \bar{V}/a$ and cost function $V_N^\beta(x, \mathbf{u})$ given by (2.43). Choose $(x, \mathbf{u}) \in \mathbb{Z}_N^\beta$. Then, for the successor state $x^+ = Ax + Bu(0)$, with $u(0)$ being the first input in the sequence $\mathbf{u}$, choose the warm start $\tilde{\mathbf{u}}$ according to Definition 8. Then, $\tilde{\mathbf{u}} \in G(x, \mathbf{u})$, with the set G defined in Definition 9.

Proof. Since $(x, \mathbf{u}) \in \mathbb{Z}_N^\beta$, we can use Proposition 18 to establish that $x(N) = \phi(N; x, \mathbf{u}) \in \mathbb{X}_f$. Since Assumption 3 holds, we know that the warm start is feasible. Hence $V_N^\beta(x^+, \tilde{\mathbf{u}}) \leq \bar{V}$. By the construction of the warm start, we have that:

$$V_N(x^+, \tilde{\mathbf{u}}) = V_N(x, \mathbf{u}) + \big(\beta V_f(Ax(N) + Bu_+) + \ell(x(N), u_+) - \beta V_f(x(N))\big) - \ell(x, u(0))$$

The result follows from Proposition 19 and Assumption 2 □

The centralized MPC optimization problem is now written as:

$$\begin{aligned}
\mathbb{P}_N(x) : &\min_{\mathbf{u}} V_N^\beta(x, \mathbf{u}) \\
&\text{s.t. } x_i(j+1) = A_i x_i(j) + B_{ii} u_i(j) + \sum_{\substack{l \in \{1,2,\ldots,M\} \\ l \neq i}} B_{il} u_l(j), \quad j = \{0, 1, 2, \ldots, N-1\}, i = \{1, 2, \ldots, M\} \\
&u_i(j) \in \mathbb{U}_i \qquad j = \{0, 1, 2, \ldots, N-1\}, i = \{1, 2, \ldots, M\} \\
&x_i(0) = x_i \qquad i = \{1, 2, \ldots, M\}
\end{aligned} \tag{2.45}$$

Notice that we do not enforce any terminal constraints in problem (2.45) because they are automatically satisfied by the choice of β and $\mathbb{Z}_N^\beta$. That is, we restrict the feasible states to lie in:

$$\mathscr{X}_N^\beta := \left\{x \mid \exists \mathbf{u} \in \mathbb{U}^N \text{s.t. } (x, \mathbf{u}) \in \mathbb{Z}_N^\beta\right\}$$

Since (2.45) is a convex problem subject to uncoupled constraints (the equality constraints are projected out), for any $(x, \mathbf{u}) \in \mathbb{Z}_N^\beta$, we have that

$$V_N^{\beta,0}(x, \mathbf{u}^0) \le V_N^\beta(x, \mathbf{u}^{(p)}) \le V_N^\beta(x, \mathbf{u}) \le \bar{V}$$

in which $V_N^{\beta,0}(x, \mathbf{u}^0)$ is the optimal cost and $V_N^\beta(x, \mathbf{u}^{(p)})$ is the cost obtained after p iterations of Jacobi algorithm (Algorithm 3) applied to optimization problem (2.45). Also, as $p \to \infty$, $\mathbf{u}^{(p)} \leftarrow \mathbf{u}^0$. Hence Algorithm 2 can be used to stabilize the plant *with guaranteed performance* to the centralized solution of the problem (2.45).

We now discuss the features of the aforementioned relaxation:

Applicability. The relaxation is applicable to any stabilizable centralized system.

Convergence. As mentioned earlier, the relaxation ensures that the inner optimization loop of Algorithm 2 converges to the optimal solution of (2.45).

As shown in Pannocchia et al. (2011), for a fixed a, as we increase $\bar{V}$, the set $\mathbb{Z}_N^\beta$ approaches the following set:

$$\bar{\mathbb{Z}} = \left\{(x, \mathbf{u}) \mid \exists \mathbf{u} \in \mathbb{U}^N, \phi(N; x, \mathbf{u}) \in \mathbb{X}_f\right\}$$

Therefore, we could cover as much of the feasible space by increasing $\bar{V}$.

The main drawback of the relaxation method is that the Hessian of the objective could get ill-conditioned due to the choice of $\bar{V}$ and a.

Initialization. As mentioned in the previous section, the warm start could become infeasible. From Proposition 18, it is clear that an infeasible warm start indicates that $V_N^\beta(x, \tilde{\mathbf{u}}) > \bar{V}$. However, note that warm-start does not make the optimization problem (2.45) infeasible. Since the inner loop optimizations decrease the objective values at each iteration, there is an iteration p'

so that $V_N(x, \mathbf{u}^{(p)}) \leq \bar{V}$ for $p \geq p'$. After p' iterations of the inner optimization loop, we would have ensured that the stability requirements are satisfied. The drawback is, unlike the regulation problem with a feasible warm start, we cannot terminate the inner-loop optimizations arbitrarily after any number of iterations. For small infeasibilities, we can expect the algorithm to regain feasibility in a few iterations, but we do not have a theoretical upper-bound on the number of iterations, p', required to regain feasibility.

2.3.4 Example[9]

The system, consists of two tanks with levels x_1 and x_2. The two tanks are considered as two separate subsystems for implementing distributed MPC. Subsystem-1 controls the level x_1 and has the inputs u_{11}, u_{12}, u_{13} at its disposal. The input u_{12} drains water from the first tank into the second tank. Input u_{13} directly drains water from the first tank, but it is assumed that manipulating input u_{13} is more expensive than manipulating input u_{12}. Subsystem-2 controls the level x_2 and has the inputs u_{21} and u_{22} at its disposal. The input u_{21} recycles a fraction of the water back into the first tank according to the recycle ratio r. Similar to subsystem-1, input u_{22}, which directly drains water out from subsystem-2, is assumed to be more expensive to operate compared to input u_{21}. To add some complexity, we assume that the recycle flow from subsystem-2 to subsystem-1 introduces a further disturbance in the system,which is perfectly modeled. This disturbance introduces water into the first tank at a rate proportional to the flow out of the second tank through u_{21}. Such interactions can arise when there is tight heat and mass integration in chemical plants. The parameter r is chosen as 0.1.

The subsystem-1 model for the two-tank system is

$$x_1^+ = \underbrace{1}_{A_1} x_1 + \underbrace{\begin{bmatrix} 1 & -1 & -1 \end{bmatrix}}_{B_{11}} \underbrace{\begin{bmatrix} u_{11} \\ u_{12} \\ u_{13} \end{bmatrix}}_{u_1} + \underbrace{\begin{bmatrix} 1+r & 0 \end{bmatrix}}_{B_{12}} \underbrace{\begin{bmatrix} u_{21} \\ u_{22} \end{bmatrix}}_{u_2} \tag{2.46}$$

[9]This example is taken from Subramanian et al. (2012b).

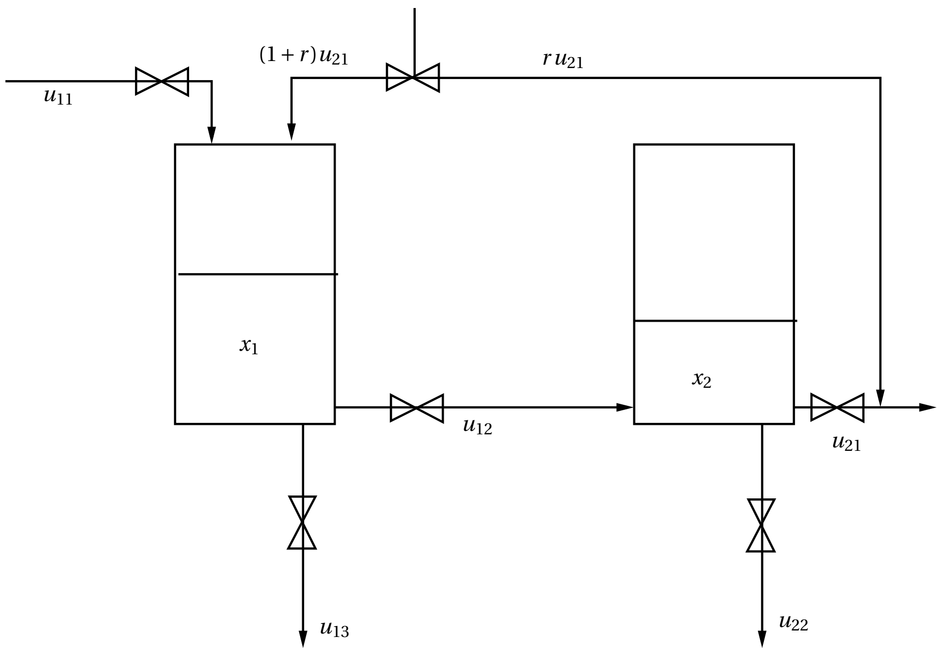

Figure 2.2: The two-tank system.

The subsystem-2 model for the two-tank system is

$$x_2^+ = \underbrace{1}_{A_2} x_2 + \underbrace{\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}}_{B_{21}} \underbrace{\begin{bmatrix} u_{11} \\ u_{12} \\ u_{13} \end{bmatrix}}_{u_1} + \underbrace{\begin{bmatrix} -1 & -1 \end{bmatrix}}_{B_{22}} \underbrace{\begin{bmatrix} u_{21} \\ u_{22} \end{bmatrix}}_{u_2} \tag{2.47}$$

The overall (centralized) model of the two-tank system is the minimum realization of

$$x^+ = \underbrace{\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}}_{B} u \tag{2.48}$$

in which $x = (x_1, x_2)$ and $u = (u_{11}, u_{12}, u_{13}, u_{21}, u_{22})$. Each input is constrained to lie between $[0, \bar{u}]$ in which 0 corresponds to the valve completely closed and $\bar{u}$ corresponds to the valve completely open. The upper bound on the valve was chosen to be arbitrarily large.

We define stage costs $\ell_1(\cdot,\cdot)$ and $\ell_2(\cdot,\cdot)$:

$$\ell_1(x_1, u_1) = x_1^2 + u_{11}^2 + u_{12}^2 + 100u_{13}^2$$
$$\ell_2(x_2, u_2) = x_2^2 + u_{21}^2 + 100u_{22}^2$$

The system starts at steady state with tank levels $(7,7)$ and all valves closed. At time $t = 0$, we change the setpoint of the two tanks to level $(3,3)$ and all valves closed.

The responses are shown in Figure 2.3. In noncooperative MPC, each subsystem uses the cheap input u_{12}, u_{21} to change the tank levels; unaware that this choice of inputs leads to instability by introducing more water into the system. The subsystems manipulate the cheap inputs because the influence of their inputs on the other subsystem is not captured in the noncooperative MPC optimization problem. At each iteration, the two subsystems, optimizing independently, harm each other because they do not want to operate the expensive valves u_{13} and u_{22}. In cooperative MPC, subsystem-2 realizes that operating valve u_{21} is not desirable, because it optimizes the overall objective function. The subsystems now judiciously use the expensive valves to maintain stability.

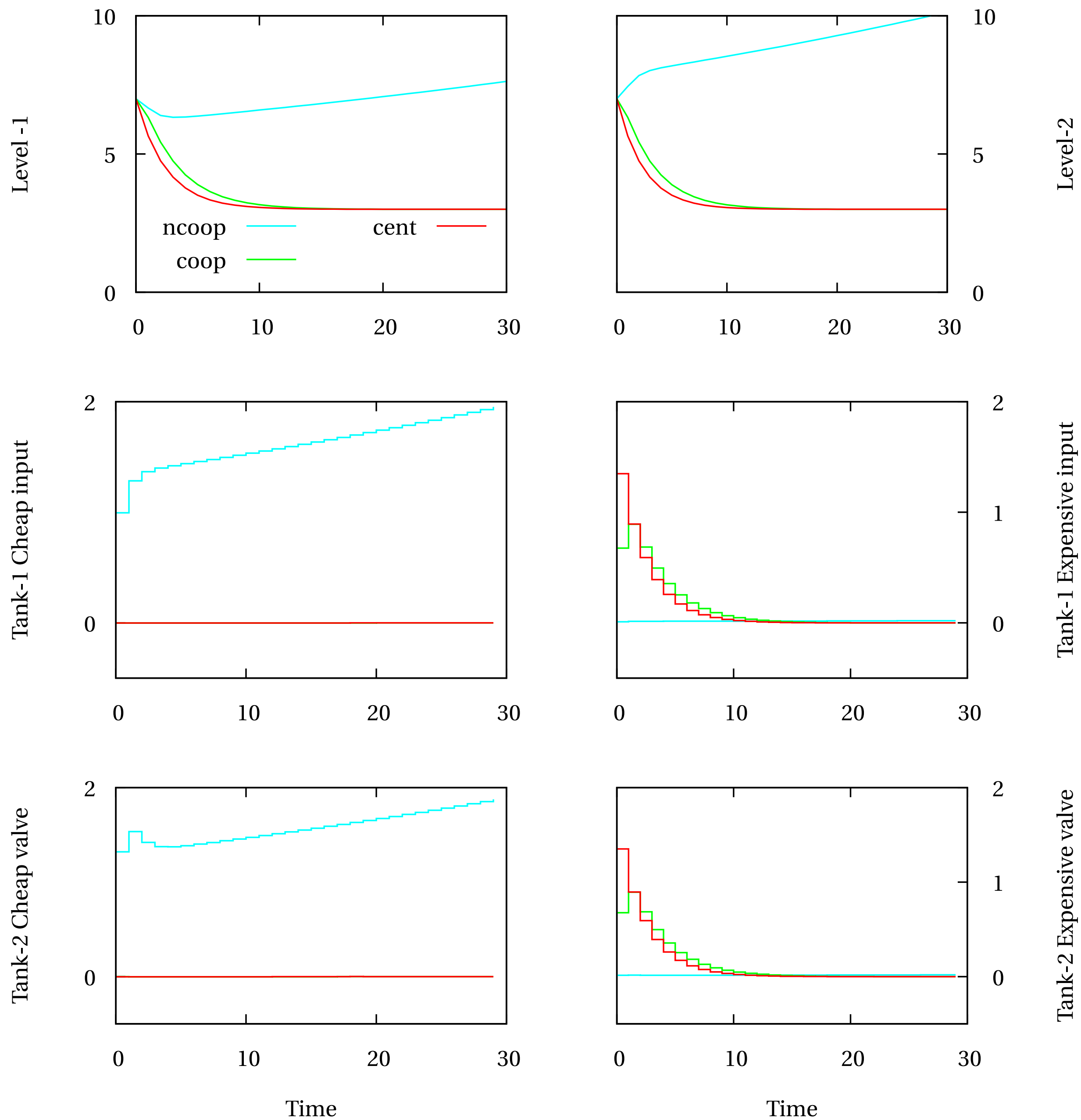

Figure 2.3: State and input profiles for two-tank system under distributed MPC (ncoop: noncooperative, coop: cooperative, cent: centralized).

2.4 Robust cooperative MPC

2.4.1 Preliminaries

We consider the centralized system (2.28) obtained from the distributed models (2.25), subject to bounded additive disturbance as follows:

$$x^+ = Ax + Bu + w \tag{2.49}$$

in which the inputs are assumed to lie in set $u_i \in \mathbb{U}_i$ as in the previous section. The assumptions on the disturbance are stated in Assumption 21

Assumption 21 (Bounded disturbance)**.** *The additive disturbance w lies in a convex, closed and, compact set $\mathbb{W}$ containing the origin in its interior.*

The nominal system, without the additive disturbance is denoted as follows, using z, v for the nominal state and input variables.

$$z^+ = Az + Bv \tag{2.50}$$

At any time k, we can write the deviation between the actual state and the nominal state as $e(k) = x(k) - z(k)$. If the inputs to both the nominal and actual system were the same, then the error dynamics can be written as:

$$e^+ = Ax + Bu + w - Az + Bu = Ae + w \tag{2.51}$$

Hence, given an initial $e(0) = 0$, the error at time k lies in the following set:

$$e(k) \in S(k) := \sum_{j=0}^{k-1} A^j \mathbb{W} = \mathbb{W} \oplus A\mathbb{W} \oplus \ldots \oplus A^{k-1}\mathbb{W} \tag{2.52}$$

in which $A^j\mathbb{W}$ indicates set multiplication. That is,

$$A^j\mathbb{W} := \left\{ A^j w \mid \forall w \in \mathbb{W} \right\}$$

The symbol $\oplus$ indicates set addition. That is,

$$\mathbb{W} \oplus A\mathbb{W} := \{w_1 + w_2 \mid w_1 \in \mathbb{W}, w_2 \in A\mathbb{W}\}$$

For stable A, it can be shown that the set $S(\infty)$ exists and is positive invariant for the system (2.51) (Kolmanovsky and Gilbert, 1998).

2.4.2 Tube based MPC

We now discuss tube based MPC (Rawlings and Mayne, 2009, Chapter 3), the basic idea for which is as follows: (i) use MPC on the nominal system to find $v(k) = \kappa(z(k))$, and (ii) based on the error at time k, $e(k)$, find the input to the plant as $u(k) = v(k) + Ke(k)$

By design, we select a K such that $A_K := A + BK$ is Hurwitz. Such a choice implies that the error dynamics in the closed-loop is:

$$e^+ = x^+ - z^+ = Ax + Bv + BK(x - z) + w - Az - Bv = A_K e + w \tag{2.53}$$

Now, since A_K is stable, we can conclude that $S_K(\infty) = \sum_{j=0}^{\infty} A_K^j \mathbb{W}$ exists and is positive invariant for (2.53).

The stability and convergence theorems are therefore based upon the following observations:(i) the origin is asymptotically stable for the nominal system $z^+ = Az + B\kappa(z)$ by design, (ii) the error is designed to lie in the set $S_K(\infty)$ by the choice of K and input $u = \kappa(z) + K(x - z)$, and (iii) the actual state $x(k); k \to \infty$ therefore belongs to the set $\{0\} \times S_K(\infty)$

In the presence of persistent disturbance, we ensure that the states lie inside a bounded set that we can compute offline. The name tube based MPC comes from the fact that at each time k, the state x lies in a "tube" defined by $x(k) \in z(k) \oplus S_K(\infty)$.

For the inputs $u = v + K(x - z)$ to remain feasible, we need to ensure that v satisfies the tighter constraints[10]:

$$\mathbb{V} := \mathbb{U} \ominus KS_K(\infty) \tag{2.54}$$

[10] If state constraints are present, they need to be tightened as well. We do not discuss state constraints because of Assumption 15

The tighter set follows from the fact that $e = (x - z) \in S_K(\infty)$.

The nominal MPC problem is defined as:

$$\begin{aligned} \tilde{\mathbb{P}}_N(z) : \min_{\mathbf{v}} \; & V_N(z, \mathbf{v}) \\ \text{s.t. } & z(j+1) = Az(j) + Bv(j) && j = \{0, 1, \ldots, N-1\} \\ & v(j) \in \mathbb{V}, && j = \{0, 1, \ldots, N-1\} \\ & z(0) = z \\ & z(N) \in \mathbb{Z}_f \end{aligned} \tag{2.55}$$

in which $\mathbb{Z}_f$ is a terminal set that satisfies Assumption 3 and $V_N(z, \mathbf{v})$ is the cost function defined by (2.4). Let $\kappa_s(z)$ denote the input law obtained by implementing a suboptimal MPC algorithm on (2.55). Then the origin is asymptotically stable for the closed loop $z^+ = Az + B\kappa_s(z)$ by Theorem 11. Now, if input $u = \kappa_s(z) + K(x - z)$ is injected to the plant, then $e \in S_K(\infty)$. Hence, we can prove that $\mathscr{A} := \{0\} \times S_K(\infty)$ is asymptotically stable for the composite system

$$z^+ = Az + B\kappa_s(z) \tag{2.56}$$

$$x^+ = Ax + B\kappa_s(z) + BK(x - z) + w \tag{2.57}$$

The region of attraction is $\mathcal{Z}_N \times \mathcal{X}_N$, in which $\mathcal{Z}_N$ is the following projection:

$$\mathcal{Z}_N := \left\{ z \mid \mathbf{v} \in \mathbb{V}^N \text{s.t. } z(N; z, \mathbf{v}) \in \mathbb{Z}_f \right\}$$

and $\mathcal{X}_N := \mathcal{Z}_N \oplus S_K(\infty)$.

2.4.3 Main results

Recall that if we use the relaxation formulation to remove the terminal region constraints in cooperative MPC, then each-time the warm start becomes infeasible, we need a warm start recovery step, which is the following projection onto convex sets problem:

$$\left\{ \tilde{\mathbf{u}}(x) \mid V_N^{\beta}(x, \tilde{\mathbf{u}}) \leq \bar{V}, \tilde{\mathbf{u}}_i \in \mathbb{U}_i \forall i \in \{1, 2, \ldots, M\} \right\}$$

Distributed algorithms for projection onto convex sets or the convex feasibility problems recover feasibility only at convergence. Hence, such algorithms are not suitable for distributed warm-start re-initialization since we cannot guarantee convergence within the sampling time.

To overcome these problems, we propose tube based robust cooperative MPC that is based on two important observations:(i) the optimization problems in tube based MPC are based on the nominal system, and (ii) by design, the warm start is feasible for z^+ if the input $v = \mathbf{v}(0; z)$ is implemented for the nominal system. Hence, we can conclude that the warm start based on the nominal MPC, $\tilde{\mathbf{v}}$, always remains feasible for the nominal problem. Furthermore, as we discussed in Section 2.3.3, cooperative MPC stabilizes the nominal system. Hence, we can use cooperative MPC for the nominal system. The only caveat is that to ensure convergence to the centralized solution, we need to have that the input sets are uncoupled. In this case, if we wish to implement cooperative MPC on the nominal system, we need to have that the set $\mathbb{V}$ be uncoupled, that is

$$\mathbb{V} = \mathbb{V}_1 \times \mathbb{V}_2 \times \ldots \times \mathbb{V}_M$$

In tube based MPC, the tightened set $\mathbb{V}$ is dependent on K, $S_K(\infty)$ and the original disturbance set $\mathbb{W}$. So, there is no guarantee that $\mathbb{V}$ does not have coupling between the inputs. Therefore, we introduce another offline calculation, which is to find a hyperbox $\tilde{\mathbb{V}}$ that lies completely inside $\mathbb{V}$.

Remark 22. As shown in Raković, Kerrigan, Kouramas, and Mayne (2003), it is not necessary to calculate $S_K(\infty)$ to obtain approximations to $\mathbb{V}$. In fact, if the input constraints are polytopic and decoupled, then the procedure mentioned in Raković et al. (2003), can be used to obtain tightened constraints that are also polytopic and decoupled.

As noted earlier, in robust MPC, the optimizations are performed based on the nominal state information, while the actual state could have drifted far from the nominal state because of the disturbances. We therefore use the modified version of the robust MPC algorithm presented in Rawlings and Mayne (2009, P.234).

We choose $\bar{V}, a, \beta$ such that the set

$$\mathbb{Z}_f := \left\{ z \mid V_f(z) \leq a \right\}$$

satisfies Assumption 3. We choose $\bar{V} \geq a$ and β according to Proposition 18. The controller gain K is chosen such that the centralized system $(A + BK)$ is stable and the input constraint set is tightened as:

$$\tilde{\mathbb{V}} := \tilde{\mathbb{V}}_1 \times \tilde{\mathbb{V}}_2 \times \ldots \tilde{\mathbb{V}}_M \subseteq \mathbb{V} := \mathbb{U} \ominus KS_K(\infty)$$

The centralized nominal MPC optimization problem is:

$$\begin{aligned} \tilde{\mathbb{P}}_N(z) : \min_{\mathbf{v}} \; & V_N^{\beta}(x, \mathbf{v}) \\ \text{s.t. } & z(j+1) = Az(j) + Bv(j) && j = \{0, 1, \ldots, N-1\} \\ & v(j) \in \tilde{\mathbb{V}} && j \{\in 0, 1, \ldots, N-1\} \\ & z(0) = z \end{aligned} \tag{2.58}$$

Note that the region of attraction for the cooperative nominal MPC is:

$$\mathcal{Z}_N := \left\{ z \mid \mathbf{v} \in \tilde{\mathbb{V}} \text{s.t. } V_N^{\beta}(z, \mathbf{v}) \leq \bar{V} \right\}$$

The robust cooperative MPC algorithm is presented in Algorithm 4.

The modification that we alluded to earlier is the "if condition" in Algorithm 4. The condition states that if the warm start is feasible for the actual state at time k and satisfies a cost-drop criteria, then we reset the error to zero. In this way, not only do we not lose the convergence property of the closed-loop nominal state (since the cost-drop is satisfied all the time), but we also incorporate feedback into the system. Another modification to Algorithm 4 is a slow time scale reset of the nominal state to the actual state. That is, after every T sampling times, in which T is much larger than the sampling time employed, we automatically reset the nominal trajectory. However, in this case, we need to ensure that the warm-start is feasible for the reset.

2.4.4 Example

Consider the two tank system as shown in Figure 2.4. The overall system consists of two tanks which are the two subsystems. The first subsystem (tank-1) manipulates inputs $u_1 =$

Data: Starting state $x(0)$, initial guess $(\tilde{\mathbf{u}}_1(0), \tilde{\mathbf{u}}_2(0), \ldots, \tilde{\mathbf{u}}_M(0))$ so that $V_N^\beta(x, \tilde{\mathbf{u}}) \le \bar{V}$ $\bar{p} \ge 1$ and $\omega_i \in (0,1)$ such that $\sum_{i=0}^{M} \omega_i = 1$

Result: Asymptotically stable closed loop

Offline: Perform the following computations and share with every subsystem **begin**

  Compute K so that $A + BK$ is stable

  Compute/ Approximate $S_K(\infty)$, $\mathbb{V} = \mathbb{U} \ominus K S_K(\infty)$

  Compute $\tilde{\mathbb{V}}_i$ so that $\tilde{\mathbb{V}}_1 \times \tilde{\mathbb{V}}_2 \times \ldots \times \tilde{\mathbb{V}}_M \subseteq \mathbb{V}$

end

Online: begin

  set $z(0) \leftarrow x(0); \tilde{\mathbf{v}}(0) \leftarrow \tilde{\mathbf{u}}(0)$

  set $k \leftarrow 0$

  while $k \ge 0$ **do**

    Set $p \leftarrow 0$

    Set $\mathbf{v}_i^{(p)} \leftarrow \tilde{\mathbf{v}}_i(k)$ for $i = 1, 2, \ldots, M$

    Broadcast current subsystem inputs $\tilde{\mathbf{v}}_i(k)$ to other subsystems

    while $p < \bar{p}$ **do**

      if $V_N^\beta(x(k), \tilde{\mathbf{v}}) \le V_N^\beta(z(k), \tilde{\mathbf{v}}) \le \bar{V}$ **then**

        Reset $z(k) \leftarrow x(k)$

      end

      Solve $\min_{\mathbf{v}_i} V_N^\beta(z, \mathbf{v})$s.t. $\mathbf{v}_i \in \tilde{\mathbb{V}}_i, \mathbf{v}_{-i} = \mathbf{v}_{-i}^{(p)}$ to obtain $\mathbf{v}_i^0$ for i in $1, 2, \ldots, M$. Set $\mathbf{v}_i^{(p+1)} \leftarrow \omega_i \mathbf{v}_i^{(p)} + (1 - \omega_i)\mathbf{v}_i^0$ for i in $1, 2, \ldots, M$

    end

    Set $\mathbf{v} \leftarrow (\mathbf{v}_1^{(p)}, \mathbf{v}_2^{(p)}, \ldots, \mathbf{v}_M^{(p)})$ and find $z(k+N) \leftarrow \phi(N; z(k), \mathbf{v})$

    Obtain $v_+ = (v_{1+}, v_{2+}, \ldots, v_{M+}) \leftarrow \kappa_f(z(k+N))$

    Obtain warm start $\tilde{\mathbf{v}}_i(k+1) = (\mathbf{v}_i^{(p)}(1), \mathbf{v}_i^{(p)}(2), \ldots, v_{i+})$ for $i = 1, 2, \ldots, M$.

    Set input as $v(k) = (\mathbf{v}_1^{(p)}(0), \mathbf{v}_2^{(p)}(0), \ldots, \mathbf{v}_M^{(p)}(0))$

    Evolve nominal state from $z(k)$ to $z(k+1)$ under input $v(k)$

    Set input $u(k) = v(k) + K(x(k) - z(k))$

    Evolve state from $x(k)$ to $x(k+1)$ under input $u(k)$

  end

end

Algorithm 4: Robust cooperative MPC

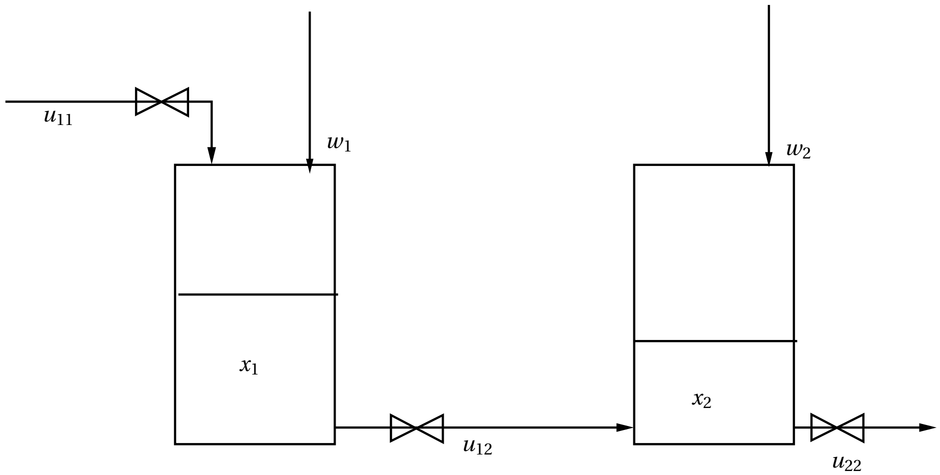

Figure 2.4: Two tank system

(u_{11}, u_{12}), while the second subsystem (tank-2) manipulates inputs $u_2 = (u_{22})$. There are two disturbances affecting the system, w_1 in the first tank and w_2 in the second tank. The state dynamics for this two tank system is given by:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u_2 + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

We assume that the nominal value of $w_{1,n} = 0.1$ and that of $w_{2,n} = 5$. The set $\mathbb{W}$ is given by $\mathbb{W} := \{w \mid 0 \le w_1 \le 0.2, 0 \le w_2 \le 10\}$.

The input constraints are given by $\mathbb{U}_1 = \{u_1 \mid 0 \le u_{11} \le 10, 0 \le u_{12} \le 10\}$ and $\mathbb{U}_2 = \{u_2 \mid 0 \le u_{22} \le 20\}$.

Note that since we have a system of integrators, any level in the tank can be stabilized as long as all the flows in the system are balanced. Therefore, we choose the steady state in the tank as $x_s = (20, 20)$ (the level in both tanks are 20). The input steady state is obtained by solving the following optimization problem

$$\min_u 1/2 u' R u \qquad \text{s.t. } Bu = -w_n; u \in \mathbb{U}$$

in which w_n is the nominal disturbance. For the choice of $R = I$ (I denotes the identity matrix), the input steady state is obtained as $u_s = (3.2667, 3.3667, 8.3667)$.

The stage cost was chosen as $\ell(x,u) = 1/2(0.1x'x + u'u)$. We solve the MPC problem in deviation variables, so that regulation to the origin implies regulation to the steady state mentioned above. Following the design procedure outlined in the previous sections, we choose (i) $V_f(x) = 1/2x'Px, \kappa_f(x) = Kx$ as the solution to the Riccati equation, (ii) $a = 1$ and the terminal region as $\{x \mid 1/2x'Px \leq 1\}$. The choice of $a = 1$ satisfies the requirements in Assumption 3, (iii) $\bar{V} = 100$, (iv) a prediction horizon of $N = 15$ and (v)the controller that corrects for the error between the nominal and actual states was as $K = \kappa_f(x)$.

For these choice of parameters, we followed the algorithm mentioned in Raković et al. (2003) to find the set $\mathbb{V}$ (we chose $N = 200$ and $\alpha = 1e-6$). Note that, since the original input set contained no coupled inputs, the tightened set also contains no coupled inputs.

In Figure 2.5, we show the closed-loop response nominal closed-loop response of the level in the second tank and the for cooperative MPC rejecting a persistent disturbance $w_k \in \mathbb{W}$. We also show the cost-function $V_N^\beta(z, \tilde{\mathbf{v}})$ and $V_N^\beta(x, \tilde{\mathbf{v}})$ to show that although the warm-start was infeasible for the actual state, it was still feasible for the nominal state and hence we could obtain the closed-loop guarantees for robust cooperative MPC. Note that, for this particular disturbance realization, we could not reset the nominal state to the actual state.

In Figure 2.6, we show the closed-loop response using a modified version of Algorithm 4. The modification we made are to reset the nominal state to the actual state at time k if the following conditions are satisfied (i)The nominal state $z(k)$ is inside $\mathbb{Z}_f$. (ii) The warm start $\tilde{\mathbf{v}}(k)$ is feasible for the actual state $x(k)$ and , (iii) The time elapsed since the last reset is greater than $\bar{T}$ time periods (we chose $\bar{T} = 10$)

2.5 Related Work

Cooperative MPC has evolved as an attractive architecture for distributed control because it solves the centralized control problem, and inherits the desirable closed-loop properties of centralized control. In the previous sections, we described cooperative MPC based on the "primal decomposition" of the centralized optimization problem. In the primal decomposition,

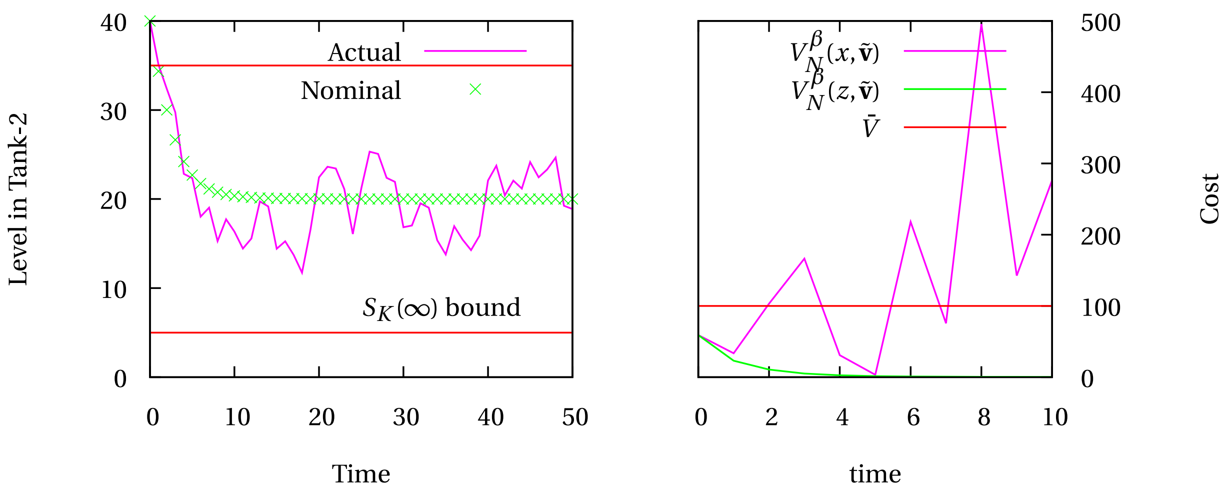

Figure 2.5: (Left) Closed-loop response (Right) Warm start rendered infeasible for actual state because of disturbance. The warm start is infeasible if $V_N^\beta(x,\tilde{\mathbf{v}}) > \bar{V}$

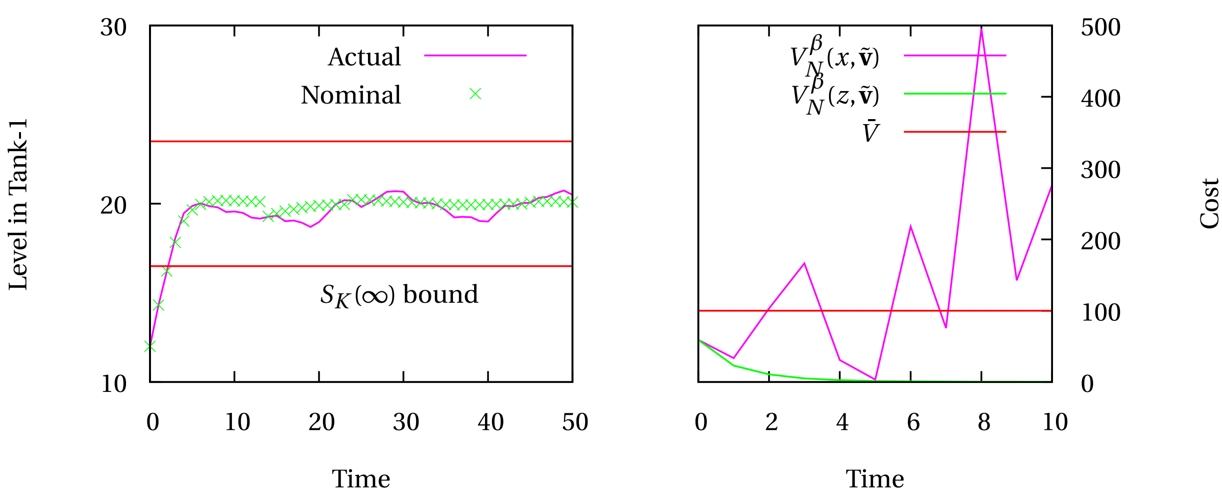

Figure 2.6: (Left) Closed-loop response. Notice that we reset the state around t = 15 (Right) Warm start rendered infeasible for actual state because of disturbance

the centralized optimization problem is solved directly using parallel optimization architectures. Liu, Chen, Muñoz de la Peña, and Christofides (2010) also use the primal decomposition to solve the centralized optimization problem for a nonlinear process model. They use a closed-form controller $u = h(x)$ for which a Lyapunov function is known as a reference controller to design their MPC optimization problem. Thus, they ensure that the MPC inherits the stability properties of $u = h(x)$. Note that $u = h(x)$ also provides a warm start, even when the actual and predicted states are different. However, since this stability constraint is a coupled constraint, there can be no guarantees about the convergence of the parallel optimization routine to the optimal solution; and hence equivalence of optimal MPC and cooperative MPC if the iterations were allowed to converge. The authors propose both a Jacobi algorithm (all subsystems optimize in parallel) and a Gauss-Seidel algorithm (subsystems optimize in sequence). In comparison, in Stewart et al. (2011), the authors propose a Jacobi algorithm for nonlinear MPC that converges to the centralized optimal solution. Since, for non-convex problems, the Jacobi optimizations does not necessarily produce a descent direction, the authors propose a sequential procedure to obtain a descent direction using the solutions obtained from each subsystem. This overhead is not equivalent to implementing a coordinator as each subsystem only calculates an objective function in the second phase of the algorithm in which a descent direction is determined. Maestre, Muñoz de la Peña, Camacho, and Alamo (2011b) propose a primal decomposition approach to cooperative MPC based on agent negotiation. The advantage of their procedure is that agents need only know models of the subsystems whose inputs affect their states. In the proposed method, each agent optimizes its local objective over all the inputs that affect its dynamics, and share the proposed solution with other agents. The other agents evaluate the proposal for cost-drop and constraint violation and communicate back to the original agent making the proposal, who can then decide to accept or reject the proposal. The authors ensure that only feasible proposals are accepted. The drawback of the proposed architecture, however, is that (i) the agents have to solve larger optimization problems (because they have to optimize over all the inputs that affect their state), and (ii) the convergence to the centralized optimal solution cannot be guaranteed. Stability is guaranteed using the warm start. Maestre, Muñoz de la Peña, and Camacho (2011a) use game-theoretic analysis to propose a distributed optimization framework. In this method, each node, optimizes its local objective over its local decisions while keeping the other subsystem decisions fixed. After completion of the optimizations, the agents compute their local objectives for all possible combinations of the overall system input (based on optimized solution of the agents and the warm start). Upon sharing the objectives, the agents then select the input that minimizes the overall cost. Thus, each agent cooperatively makes a decision. However, the proposed algorithm also fails to establish convergence to the centralized optimal on iteration. Stability is guaranteed by design of terminal region and warm-start. Müller, Reble, and Allgöwer (2012) propose a optimization algorithm based on each node optimizing over its local optimization problem. They use a terminal region which is a sub-level set of the terminal penalty. Because of the presence of coupled constraints, input directions are discarded if they are not feasible, based on a check made after the optimizations. To ensure cost-drop, the centralized objectives are also evaluated after the optimizations and inputs that do not achieve cost-drop are discarded. The model considered by the authors had coupling introduced only via the constraints (both in the objective function and the constraints). The authors also provide a method to define local time varying terminal regions, so that the coupled terminal region constraint is satisfied if each subsystem satisfies its local time varying terminal region constraint. The algorithm provided by the authors, satisfies the requirements of the optimizer for suboptimal MPC, but again, does not give any guarantee on convergence to the optimal solution. The requirement of decoupled dynamics is important in problems like multi vehicle synchronization etc. Johansson, Speranzon, Johansson, and Johansson (2006) use a primal decomposition to solve a multi-vehicle consensus problem as a MPC problem. While the dynamics are decoupled, the consensus point, similar to terminal equality constraint, is the complicating constraint. Unlike the MPC problems where the objectives are also constrained because of the dynamics, the multi-vehicle receding horizon problem falls into the category of uncoupled objective but coupled constraints. The author's use a primal decomposition which generates feasible iterate that reduce the objective function value.

However, in order to ensure that the centralized optimal solution is achieved, the authors use a coordinator, which is based on sub-gradient optimization to handle the coupled constraint.

A common theme in optimizing the centralized problem is that it is not easy to guarantee convergence to the optimal solution. However, stability can be guaranteed because every iterate is designed so that it reduces the cost while remaining feasible. In contrast, there are a lot of cooperative MPC algorithms which are based on the "dual decomposition". In the dual decomposition, the coupled constraints are relaxed by using the Lagrangian of the optimization problem. For a fixed value of the Lagrange multipliers (also called as prices or dual variables), the relaxed problem can be solved using parallel optimization methods as there are no complicating constraints. Upon achieving the solution to the relaxed problem, the Lagrange multipliers are updated. The Lagrange multiplier update is usually done by a coordinator. These algorithms often converge faster to the optimal solution. However, their main disadvantage is that they are guaranteed to produce a feasible iterate only upon convergence. Since stability theory for suboptimal MPC rely on the fact that the suboptimal iterate is feasible, cooperative MPC algorithms using dual decomposition use stability theory based on optimal MPC to ensure stability. Therefore, a common theme in dual decomposition based cooperative MPC algorithms are a coordinator layer and a requirement that the iterates converge.

The cooperative MPC algorithms using dual decomposition differ based on the technique used to update the dual variables. In Cheng, Forbes, and Yip (2007), the dual variables (prices) are updated using a sub-gradient based optimization algorithm. Sub-gradient methods are also used in Ma, Anderson, and Borrelli (2011), Wakasa, Arakawa, Tanaka, and Akashi (2008), Marcos, Forbes, and Guay (2009). Moroşan, Bourdais, Dumur, and Buisson (2011) formulate the building control problem as a MPC problem with linear objectives and use Benders decomposition to solve the problem. Benders decomposition is a widely popular parallel algorithm when by fixing the value of a complicating variable, the remaining problem can be completely separated. Scheu and Marquardt (2011) propose a dual decomposition algorithm without a coordination layer. They augment the local subsystem objective function with the sensitivity of the objectives and constraints of other subsystems to obtain updates for the dual variables along with the primal variables. However, this method generates a feasible solution only upon convergence. Giselsson, Doan, Keviczky, De Schutter, and Rantzer (2012), Giselsson and Rantzer (2010) propose a dual decomposition algorithm with a stopping criteria based on the objective value to ensure stability. They advocate the use of long prediction horizon along with results obtained in Grüne (2009) to determine bounds on the value of the objective function so that stability can be guaranteed. Doan, Keviczky, Necoara, Diehl, and De Schutter (2009) modified the Han's algorithm which is a dual decomposition based algorithm for the special structure of the MPC problem. Although the method uses communication between directly connected subsystems, stability is guaranteed only upon convergence. Necoara, Doan, and Suykens (2008) use a smoothing technique to simplify the dual problem. With the smoothing technique, the coordinator problem for finding the Lagrange multiplier updates becomes easier. The algorithm also gives bounds on the number of iterations so that the optimal solution and constraint violation are within a pre-specified limit (ϵ approximation of the centralized problem). Finally, Doan, Keviczky, and De Schutter (2011), propose a primal feasible dual gradient approach, that generates a primal feasible solution that achieves cost-drop in a finite number of iterations based on an averaging scheme of the primal variables at each iteration.

Christofides, Scattolini, de la Peña, and Liu (2012) is a recent review of different algorithms for distributed MPC. Necoara, Nedelcu, and Dumitrache (2011) provides an excellent overview of the different optimization problems and parallel solution strategies that are seen in control and estimation.

Trodden and Richards (2006, 2007) propose a tube based robust distributed MPC algorithm. In their method, at each sampling time, only one subsystem performs optimization. The subsystem optimizes only over its decision variables, keeping all other subsystem decisions fixed from the previous iteration. This method is also an example of primal decomposition. Richards and How (2004) present a robust tube-based MPC for systems with decoupled dynamics. The coupling constraints are coupled output constraints. Their algorithm is based on the Gauss-Seidel iterations.

Chapter 3

A state space model for chemical production scheduling

In Chapter 2, we discussed design of on-line optimization problems for the control of dynamic systems using MPC, so that the closed-loop has desirable properties like recursive feasibility, asymptotic convergence. In this chapter, we employ ideas from MPC to address iterative or rolling horizon scheduling problems. In Section 3.1, we provide an introduction to the problem that we wish to address. In Section 3.2, we give a brief background on chemical production scheduling and associated rescheduling problems. In Section 3.3, we derive the state space model, including four types of disturbances. In Section 3.4, we present an example illustrating the advantages of using terminal constraints in iterative scheduling.

3.1 Introduction [1]

Chemical production scheduling problems arise in a wide variety of applications, from batch production of pharmaceuticals and fine chemicals to continuous production of bulk chemicals and oil refining operations. To address these problems, research within the process systems engineering (PSE) community has primarily focused on (i) the formulation of models for a wide variety of scheduling problems, and (ii) the development of scheduling algorithms. In terms of model development, the emphasis has been on the accurate representation of problems in a range of production environments as well as the modeling of various processing characteristics and constraints (e.g.,utility constraints, changeovers, transfer operations, etc.)(Méndez, Cerdá,

[1] This text appears in Section 1 of Subramanian, Maravelias, and Rawlings (2012a)

Grossmann, Harjunkoski, and Fahl, 2006). An aspect that has received limited attention is how to design algorithms, based on these models, for iterative scheduling.

Chemical production is an inherently dynamic process. A schedule has to be revised when new information becomes available (new orders, modified due dates, raw material availability etc.), and/or production disturbances occur (e.g.,processing delays, unit breakdowns, process unit availability, etc.). However, while some of the issues arising when scheduling is performed iteratively have been discussed in contributions dealing with rescheduling, scheduling is still thought of as a *static* open-loop problem – the goal is to obtain an optimal schedule for the current state of the system based on current (and possibly some forecast) data. The development of methods (models and solution algorithms) for the closed-loop problem has received no attention. Another limitation of existing rescheduling methods, as we discuss in Section 3.2.2, is that they are model specific and rely on the solution of a rescheduling model that is generated *empirically*.

The goal of this paper is to address some of the aforementioned limitations employing ideas from the area of control and model predictive control (MPC) in particular. Model predictive control offers a natural framework for the study of dynamic problems. First, it relies on a general representation of the underlying system, including different types of disturbances, via the state space model. Second, it offers results with regard to the quality of the closed-loop performance of various control strategies. For example, with careful design of the on-line optimization problem, features such as recursive feasibility (feasibility of the optimization problem at each sampling instance) and asymptotic stability (convergence to a set-point for the nominal case) can be obtained. Interestingly, it has been shown that simple re-optimization does not necessarily lead to good closed-loop performance, as has been assumed in the scheduling literature.

Towards this goal, we first transform a general mixed-integer programming (MIP) scheduling model into a state space model (2.1). Second, we show how common scheduling disruptions can me modeled as disturbances in the state space model, and finally, we discuss how some concepts from MPC like terminal constraints can be used in scheduling.

3.2 Background

3.2.1 Chemical production scheduling problems and models[2]

Production scheduling is one of the many planning functions in a manufacturing supply chain. The interactions of scheduling with other functions along with capacity considerations determine the class of scheduling problem. The interactions with demand and production planning determine the type of scheduling problem to be solved (cyclic vs. short-term). The types of decisions made at the scheduling level are determined by the decisions made at the production planning level. Also, capacity constraints often determine the objective function (e.g.,throughput maximization vs. cost minimization). Finally, input parameters to scheduling (e.g.,raw material availability) are provided by other functions (Maravelias and Sung, 2009; Maravelias, 2012; Stadtler, 2005).

In general, scheduling problems can be classified in terms of a triplet $\alpha/\beta/\gamma$, where α denotes the *production* environment; β denotes the processing characteristics/constraints and γ denotes the objective function (Pinedo, 2008). The main production environments are *sequential, network* and *hybrid* (Maravelias, 2012). Note that different types of processing can be present in the same facility. Processing characteristics and constraints include setups, changeovers, release/due times, storage constraints, material transfers, etc. Common objective functions are the minimization of makespan, the minimization of production costs, the maximization of throughput, and the minimization of weighted lateness.

The modeling approaches to chemical production scheduling can be classified in terms of (Maravelias, 2012):

1. the decisions made at the scheduling level;
2. the entities used to express the scheduling model; and
3. the modeling of time.

[2]This text appears in Section 2.1 of Subramanian et al. (2012a)

In the most general case, scheduling involves three types of decisions: (i) batching (number and size of batches needed to satisfy demand); (ii) assignment of batches (or tasks) to processing units; and (iii) sequencing and/or timing of batches (tasks) on processing units. If the batching decisions are fixed, then scheduling problems are expressed in terms of batches (batch-based approach). If batching decisions are made at the scheduling level, then materials and material amounts are typically used to formulate the scheduling model (material-based approach). Finally, the modeling of time includes decisions at four levels: (i) selection between precedence and grid-based approach; (ii) if precedence-based, selection between local and global precedences; if time-grid based, selection between common and unit specific grids; (iii) specific assumptions regarding the precedence relationship between two tasks and the mapping of task onto time; and (iv) selection between discrete and continuous time representation (Maravelias, 2012).

In this paper we assume batching, unit-task assignment and sequencing/timing decisions are all made at the scheduling level (material-based approach). We further assume that the general scheduling problem can be expressed in terms of production *tasks, units*(unary resources), and *materials*. While this type of formalism has been traditionally used to express problems in network production environment, Sundaramoorthy and Maravelias (2010) showed that it can also be employed to represent problems in all production environments. A thorough discussion of the various scheduling problems and modeling approaches is presented in Méndez et al. (2006).

3.2.2 Reactive scheduling[3]

Rescheduling, or reactive scheduling, after observing disturbances to the nominal schedule has attracted some research attention in the past few years. Smith (1995) emphasizes the process view of the scheduling problem and outlines the following criteria for *reactive scheduling*: (i) prioritize outstanding problem; (ii) identify modifying goals; and (iii) estimate possibilities

[3]This text appears in Section 2.4 of Subramanian et al. (2012a).

for efficient and non-disruptive schedule modification. In the MIP-based approaches to reactive scheduling, a nominal schedule is used in conjunction with a MIP model to react to disturbances. On observing a disturbance, part of the schedule which has already been implemented is fixed and the remainder of the scheduling horizon is re-optimized using modifications to the original model to reflect the disturbances. Such strategies were proposed by Vin and Ierapetritou (2000); Janak, Floudas, Kallrath, and Vormbrock (2006); Relvas, Matos, Barbosa-Póvoa, and Fialho (2007), among others. Novas and Henning (2010) propose a constraint programming based approach to locally repair the nominal solution. Méndez and Cerdá (2003) also propose a local repair solution to the schedule based on a MIP formulation that considers the current "state" of the plant, a nominal schedule and new information. Motivated by rolling horizon optimization in process control, several shrinking horizon and rolling horizon approaches to the scheduling problem have also been proposed. For instance, van den Heever and Grossmann (2003) provide an example of a complex hydrogen pipeline, in which they divide the planning horizon into planning periods, and for each planning period, they solve the scheduling problem in a shrinking horizon formulation. Sand and Engell (2004) solve a two stage stochastic optimization problem to find robust schedules. They employ a moving horizon framework in which the decisions in the current time period, the first stage decisions are implemented, while the second stage decisions are embedded in a scenario tree for stochastic variables. Honkomp, Mockus, and Reklaitis (1999) use an optimizer to perform the scheduling in conjunction with a simulator to simulate stochastic scenarios. Rodrigues, Gimeno, Passos, and Campos (1996) propose a rolling horizon reactive scheduling method in which they provide a predictive framework to determine future infeasibilities that lie outside the current optimization horizon. Huercio, Espuna, and Puigjaner (1995) present heuristics for rescheduling based on shifting of task processing times and reassignment of tasks to other units. Li and Ierapetritou propose a multiparametric approach to rescheduling. Munawar and Gudi (2005) propose a three level decomposition of the problem, and motivated by process control, formulate feedback and cascade control-like solutions to reactive scheduling. Li and Ierapetritou (2008) present a review of different strategies used in reactive scheduling. Verderame, Elia, Li, and Floudas (2010) also present a review of different approaches taken in different industries.

3.3 State space scheduling model

3.3.1 General problem statement[4]

The scheduling problem we consider is stated as follows. We are given:

1. A set of processing tasks $i \in \mathbf{I}$; the processing time of task i is denoted by $\tilde{\tau}_i$, its fixed batchsize by β_i (variable batchsizes are considered in Section 3.3.6), and its production cost by γ_i. Tasks that can be performed on many units are modeled as different tasks, each one carried out only in one unit.

2. A set of equipment units $j \in \mathbf{J}$. The subset of tasks i that can be carried out in unit j is denoted by the set $\mathbf{I}_j$.

3. A set of materials $k \in \mathbf{K}$ stored in dedicated storage vessels of capacity σ_k. The unit inventory cost of k is ν_k. The set of tasks i that produce/consume k is denoted by $\mathbf{I}_k^+/\mathbf{I}_k^-$. Task i consumes/produces ρ_{ik} units of material k per unit of batchsize β_i.

4. A set of shipments, $l \in \mathbf{L}$ (deliveries of feedstocks $k \in \mathbf{K}^F \subset \mathbf{K}$ or order for products $k \in \mathbf{K}^P \subset \mathbf{K}$); $\tilde{\phi}_l$ is the release (due) time of delivery (order) l; and ζ_l is the amount delivered ($\zeta_l > 0$) or due ($\zeta_l < 0$). $\mathbf{L}_k$ is the set of shipments (deliveries or orders) of material k.

Our goal is to meet the orders for the final products at the minimum total cost. Other objective functions can also be considered.

If a task has no input or output material (e.g.,when two consecutive tasks are carried out on the same unit), dummy materials can be introduced to model the sequence of tasks. Also, if material amounts need not be monitored (e.g.,sequential processes with fixed batchsize), then we assume a nominal batchsize of 1 and unit conversion coefficients. Note that we use the term *material* instead of *state*, because the latter has a different meaning in state space models. Raw

[4] This text appears in Section 2.2 of Subramanian et al. (2012a).

time-related data, $\tilde{\tau}_i$ and $\tilde{\phi}_l$ are given in regular time units (e.g.,hours), and are represented by parameters with a tilde.

3.3.2 Scheduling MIP model[5]

We consider a discrete-time model, in which the time horizon η, is divided into T periods of fixed length $\delta = \eta/T$, defining $t+1$ time points, where period t starts(ends) at time point $t-1(t)$ (Shah, Pantelides, and Sargent, 1993). We use time index $t \in \mathbf{T}$ to denote both time point and periods. Time-related data are scaled using δ and approximated so that the resulting solutions are feasible. Specifically, processing times are rounded up, $\tau_i = \lceil \tilde{\tau}_i/\delta \rceil$; and release and due times are approximated conservatively, $\phi_l = \lceil \tilde{\phi}_l/\delta \rceil$ if $\gamma_l > 0$ and, $\phi_l = \lfloor \tilde{\phi}_l/\delta \rfloor$ if $\gamma_l < 0$. We also generate the set of shipments for material k at time t, $\mathbf{L}_{kt} = \left\{ l \in \mathbf{L}_k \mid \phi_l = t \right\}$; and then calculate the total shipment of material k at time t:

$$\xi_{kt} = \sum_{l \in \mathbf{L}_{kt}} \zeta_l, \qquad \forall k, t$$

The optimizing decisions are $W_{i,t} \in \{0,1\}$ which is one if task i is assigned to start on unit j at time point t; and $S_{k,t} \geq 0$, which is the inventory of material k during time period t. Any feasible schedule should satisfy the assignment constraint (3.1) that expresses that at most one task can be executed on a unit at a time.

$$\sum_{i \in \mathbf{I}_j} \sum_{t'=t-\tau_i+1}^{t} W_{i,t'} \leq 1, \qquad \forall j, t \tag{3.1}$$

If we assume that the orders are satisfied on time, the following material balance gives the inventory variables:

$$S_{k,t+1} = S_{k,t} + \sum_{i \in \mathbf{I}_k^+} \rho_{ik} \beta_i W_{i,t-\tau_i} + \sum_{i \in \mathbf{I}_k^-} \rho_{ik} \beta_i W_{i,t} + \xi_{kt} \leq \sigma_k, \qquad \forall k, t \tag{3.2}$$

The objective function is

$$z = \min \sum_{i,t} \gamma_i W_{i,t} + \sum_{k,t} \nu_k S_{k,t} \tag{3.3}$$

[5]This text appears in Section 2.3 of Subramanian et al. (2012a)

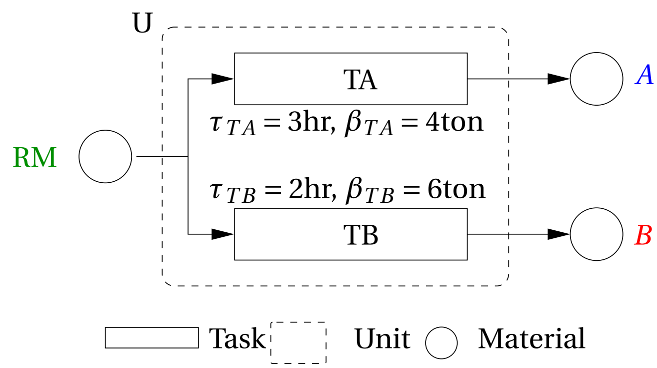

Figure 3.1: Simple scheduling problem

The basic scheduling model $\mathbf{M}^{\mathrm{SCH}}$ we consider consists of Equations (3.1)–(3.3), with $W_{i,t} \in \{0,1\}, \forall i,t$ and $S_{k,t} \in [0,\sigma_k].\forall k,t$.

If orders cannot be met on time (or we do not wish to meet them), then we can introduce backorders variable and an additional equation for its calculation.

A simple problem with one unit, two tasks and three materials, and the associated data are shown in Figure 3.1. Figure 3.2 shows a solution to this problem-a Gantt chart showing the execution of the tasks and the inventory profiles of the three materials. We use this example throughout this chapter to illustrate the basic ideas.

3.3.3 Inputs and states[6]

Since assignment variables $W_{i,t}$ are the main scheduling decisions; they are the inputs in the state space realization of $\mathbf{M}^{\mathrm{SCH}}$. Inventory levels $S_{k,t}$ in $\mathbf{M}^{\mathrm{SCH}}$ are determined by $W_{i,t}$ and the inventory balance dynamics, and hence are states. However, the variables $S_{k,t}$ do not completely describe the state of the system. Consider the solution shown in Figure 3.2. The variables $W_{\mathrm{TA},2}$ and $W_{\mathrm{TA},3}$ are both zero, but at $t = 2$, the task TA has run for one hour while at $t = 3$, the task TA has run for two hours. Therefore, to completely describe the state of the system, the history of the system should also be included in the system state. This is achieved through *lifting*. We define the new state variables $\bar{W}_{i,t}^{n}$ to carry past decisions to t. The state variable $\bar{W}_{i,t}^{n} = 1$ indicates

[6]This section corresponds to the model developed in Section 3.1 of Subramanian et al. (2012a).

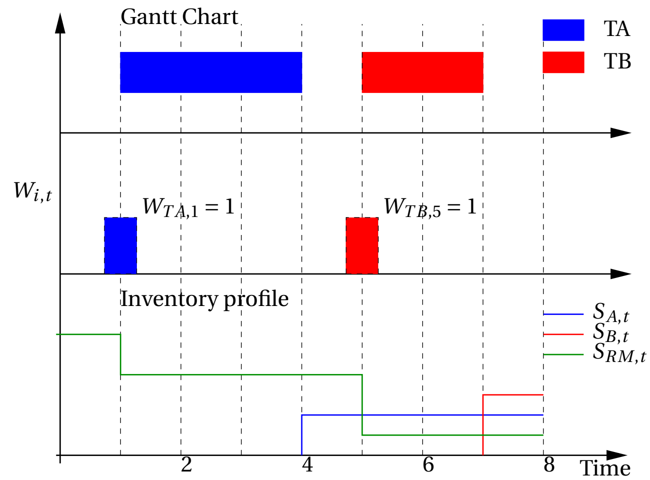

Figure 3.2: Scheduling solution

that a batch of task i started at time $t-n$. The lifted equations are given by:

$$\begin{aligned} \bar{W}^1_{i,t+1} &= W_{i,t} \\ \bar{W}^n_{i,t+1} &= \bar{W}^{n-1}_{i,t} \quad \forall n \in \{2,3,\ldots,\tau_i\} \end{aligned} \tag{3.4}$$

Using the lifted states, the inventory balance equation (3.2) can be written as:

$$S_{k,t+1} = S_{k,t} + \sum_{i\in \mathbf{I}_k^+} \rho_{ik}\beta_i \bar{W}^{\tau_i}_{i,t} + \sum_{i\in \mathbf{I}_k^-} \rho_{ik}\beta_i W_{i,t} + \xi_{k,t} \qquad \forall k,t \tag{3.5}$$

Similarly, the assignment constraint (3.1) can be written as

$$\sum_{i\in \mathbf{I}_j} W_{i,t} + \sum_{i\in \mathbf{I}_j} \sum_{n=1}^{\tau_i - 1} \bar{W}^n_{i,t} \leq 1 \qquad \forall j,t \tag{3.6}$$

Defining the state $x(t) = \left[S_{k,t}, k\in\mathbf{K}, \quad \bar{W}^n_{i,t}, i\in\mathbf{I}, n\in\{1,2,\ldots,\tau_i\}\right]$, the input $u(t) = \left[W_{i,t}, i\in\mathbf{I}\right]$ and the disturbance $d(t) = \left[\xi_{k,t}, k\in\mathbf{K}\right]$, we can write the scheduling model in the familiar state space form $x(k+1) = Ax(k)+Bu(k)+B_d d(k)$. Equations (3.4) and (3.5) express the dynamic evolution of the system, and Equation (3.6) is a joint state-input constraint. The objective function can be easily written as the sum of economic stage costs $\ell_E(x,u) = q'x + r'u$.

The dynamic evolution, constraints and stage costs for the simple scheduling model introduced in Figure 3.1 are given in Equations (3.7)–(3.9).

$$
\begin{bmatrix} S_{RM} \\ S_A \\ S_B \\ \bar{W}_{TA}^1 \\ \bar{W}_{TA}^2 \\ \bar{W}_{TA}^3 \\ \bar{W}_{TB}^1 \\ \bar{W}_{TB}^2 \end{bmatrix}_{t+1} = \underbrace{\begin{bmatrix} 1 & & & & & & & \\ & 1 & & & \beta_{TA} & & & \\ & & 1 & & & & & \beta_{TB} \\ & & & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & & & \\ & & & & & & 1 & \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} S_{RM} \\ S_A \\ S_B \\ \bar{W}_{TA}^1 \\ \bar{W}_{TA}^2 \\ \bar{W}_{TA}^3 \\ \bar{W}_{TB}^1 \\ \bar{W}_{TB}^2 \end{bmatrix}_t}_{x(t)} + \underbrace{\begin{bmatrix} -\beta_{TA} & -\beta_{TB} \\ & \\ & \\ 1 & \\ & \\ & \\ & 1 \\ & \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} W_{TA} \\ W_{TB} \end{bmatrix}_t}_{u(t)} + \underbrace{\begin{bmatrix} & \\ 1 & \\ & 1 \\ & \\ & \\ & \\ & \\ & \end{bmatrix}}_{B_d} \underbrace{\begin{bmatrix} \xi_A \\ \xi_B \end{bmatrix}_t}_{d(t)} \tag{3.7}
$$

$$
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \leq \underbrace{\begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & 1 & & 1 & \end{bmatrix}}_{E_x} \begin{bmatrix} S_{RM} \\ S_A \\ S_B \\ \bar{W}_{TA}^1 \\ \bar{W}_{TA}^2 \\ \bar{W}_{TA}^3 \\ \bar{W}_{TB}^1 \\ \bar{W}_{TB}^2 \end{bmatrix}_t + \underbrace{\begin{bmatrix} & \\ & \\ & \\ 1 & 1 \end{bmatrix}}_{E_u} \begin{bmatrix} W_{TA} \\ W_{TB} \end{bmatrix}_t \leq \begin{bmatrix} \sigma_{RM} \\ \sigma_A \\ \sigma_B \\ 1 \end{bmatrix} \tag{3.8}
$$

$$
\ell_E(x,u) = \underbrace{\begin{bmatrix} \nu_{RM} & \nu_A & \nu_B & & & & & \end{bmatrix}}_{q'} \begin{bmatrix} S_{RM} \\ S_A \\ S_B \\ \bar{W}_{TA}^1 \\ \bar{W}_{TA}^2 \\ \bar{W}_{TA}^3 \\ \bar{W}_{TB}^1 \\ \bar{W}_{TB}^2 \end{bmatrix}_t + \underbrace{\begin{bmatrix} \gamma_{TA} & \gamma_{TB} \end{bmatrix}}_{r'} \begin{bmatrix} W_{TA} \\ W_{TB} \end{bmatrix}_t \tag{3.9}
$$

3.3.4 Disturbances[7]

Events that can lead to rescheduling are modeled as disturbances. We have already discussed shipments as a disturbance in the previous section. In this section, we model three disturbances, namely, task yields, task delays and unit breakdowns.

3.3.4.1 Shipments

We assume that backorders are not allowed. Therefore, the shipping schedule is fixed by customer orders. Hence, shipments are treated as disturbances. We denote the nominal customer demands as $\xi^{\text{nom}}_{k,t}$. Shipment disturbances are deviations $\hat{\xi}_{k,t}$ from the nominal value. That is,

$$\xi_{k,t} = \xi^{\text{nom}}_{k,t} + \hat{\xi}_{k,t}$$

3.3.4.2 Task yields

Consumption and production disturbances are used to model changes in yields and losses during loading and unloading. We define yield disturbance variables $\beta^P_{i,k,t}$ and $\beta^C_{i,k,t}$ to denote deviation from the nominal production/consumption of material k by a batch of task i finishing/starting at time t. For example, $\beta^P_{i,k,t} < 0$ indicates lower yield than the nominal batch size. The material balance equations (3.5) is now modified as :

$$S_{k,t+1} = S_{k,t} + \sum_{i \in \mathbf{I}^+_k} \rho_{ik}\beta_i \bar{W}^{\tau_i}_{i,t} + \sum_{i \in \mathbf{I}^-_k} \rho_{ik}\beta_i W_{i,t} + \left(\xi_{k,t} + \sum_{i \in \mathbf{I}^+_k} \rho_{ik}\beta^P_{i,k,t} + \sum_{i \in \mathbf{I}^-_k} \rho_{ik}\beta^C_{i,k,t} \right) \qquad \forall k, t \tag{3.10}$$

3.3.4.3 Task delays

We introduce disturbance variable $\hat{Y}^n_{i,t}$ to model delays during the execution of a task. The variable $\hat{Y}^n_{i,t} = 1$ when an 1-period delay (δ h) of task i occurring n periods after task i started has been observed. The state equations (3.4) are corrected as:

[7]This section corresponds to the model developed in Section 3.1 of Subramanian et al. (2012a).

$$\bar{W}^1_{i,t+1} = W_{i,t} - \hat{Y}_{i,t}$$
$$\bar{W}^n_{i,t+1} = \bar{W}^{n-1}_{i,t} + \hat{Y}^n_{i,t} - \hat{Y}^{n-1}_{i,t}, \qquad \forall i,t,n \in \{2,3,\ldots,\tau_i\} \tag{3.11}$$

Equation (3.11) essentially says that the values of states $\bar{W}^n_{i,t+1}$ should be the same as $\bar{W}^n_{i,t}$ if there is a 1-period delay at t. The state equations (3.5) is corrected as:

$$S_{k,t+1} = S_{k,t} + \sum_{i\in\mathbf{I}^+_k} \rho_{ik}\beta_i \bar{W}^{\tau_i}_{i,t} + \sum_{i\in\mathbf{I}^-_k} \rho_{ik}\beta_i W_{i,t} + \xi_{k,t} - \sum_{i\in\mathbf{I}^+_k} \rho_{ik}\beta_i \hat{Y}^{\tau_i}_{i,t} \qquad \forall k,t \tag{3.12}$$

For example, consider the situation in which the task TA was started at $t = 1$. Hence $W_{\text{TA},1} = 1$. The state equation (3.4), then implies that $\bar{W}^2_{\text{TA},3} = 1$, as at $t = 3$, the task TA has been running for 2 hours. If a 1-period delay is observed at time $t = 3$, then the variable $\hat{Y}^2_{\text{TA},3} = 1$. This means that instead of finishing at $t = 4$, the task gets completed only at $t = 5$. Hence, for modeling purpose, the task started only at time $t = 2$. In rescheduling literature, a new model is written with this information, i.e., $W_{\text{TA},2} = 1, W_{\text{TA},1} = 0$. In our proposed method, such delays are handled organically by modifying the lifted states. Equation (3.11) tells us that

$$\bar{W}^3_{\text{TA},4} = \bar{W}^2_{\text{TA},3} + \hat{Y}^3_{\text{TA},3} - \hat{Y}^2_{\text{TA},3} = 1 + 0 - 1 = 0$$

and

$$\bar{W}^2_{\text{TA},4} = \bar{W}^1_{\text{TA},3} + \hat{Y}^2_{\text{TA},3} - \hat{Y}^1_{\text{TA},3} = 0 + 1 - 0 = 1$$

Hence, we can verify that the disturbance variable $\hat{Y}^n_{i,t}$ successfully models an 1-period delay.

3.3.4.4 Unit breakdowns

In contrast to a task delay, a unit breakdown leads to the termination of the task being executed on the unit at the time of the breakdown. In such an event, all production in that unit is also lost. To model a breakdown of unit j, we introduce disturbance variable $\hat{Z}^n_{i,t}$. The variable $\hat{Z}^n_{i,t} = 1$ when a breakdown (of duration 1 period) occurring n hours after task $i \in \mathbf{I}_j$ started is observed. Unlike the previous section, we force all the lifted variables affected by the shutdown to be zero as:

$$\bar{W}^1_{i,t+1} = W_{i,t} - \hat{Z}_{i,t}$$
$$\bar{W}^n_{i,t+1} = \bar{W}^{n-1}_{i,t} - \hat{Z}^{n-1}_{i,t} \qquad \forall i, n \in \{2,3,\ldots,\tau_i\} \tag{3.13}$$

The state equations (3.5) is corrected as

$$S_{k,t+1} = S_{k,t} + \sum_{i\in\mathbf{I}^+_k} \rho_{ik}\beta_i \bar{W}^{\tau_i}_{i,t} + \sum_{i\in\mathbf{I}^-_k} \rho_{ik}\beta_i W_{i,t} + \xi_{k,t} - \sum_{i\in\mathbf{I}^+_k} \rho_{ik}\beta_i \hat{Z}^{\tau_i}_{i,t} \qquad \forall k, t \tag{3.14}$$

Finally, to ensure that no tasks are assigned to unit j if it is out of order, the constraint (3.6) is modified as:

$$\sum_{i\in\mathbf{I}_j} W_{i,t} + \sum_{i\in\mathbf{I}_j}\sum_{n=1}^{\tau_i-1} \bar{W}^n_{i,t} + \sum_{i\in\mathbf{I}_j}\sum_{n=1}^{\tau_i-1} \hat{Z}^n_{i,t} + \hat{Z}_{i,t} \leq 1 \qquad \forall j, t \tag{3.15}$$

A breakdown lasting multiple periods, from t to $t+\phi$ can be modeled as consecutive 1 period breakdowns. Since the subsequent breakdowns occur while no task is executed, we introduce an idle task $\mathrm{IT}(j)\forall j$ with $\tau_{\mathrm{IT}(j)} = 1$ and use $\hat{Z}^1_{\mathrm{IT}(j),t+1} = \hat{Z}^2_{\mathrm{IT}(j),t+2} = \ldots = \hat{Z}^1_{\mathrm{IT}(j),t+\phi} = 1$.

3.3.5 Final model

The final state space scheduling model includes the state evolution described by (3.16) and the modified constraint given by (3.15).

$$\begin{aligned}
\bar{W}^1_{i,t+1} &= W_{i,t} - \hat{Z}_{i,t} - \hat{Y}_{i,t} && \forall i, t \\
\bar{W}^n_{i,t+1} &= \bar{W}^{n-1}_{i,t} - \hat{Z}^{n-1}_{i,t} + \hat{Y}^n_{i,t} - \hat{Y}^{n-1}_{i,t} && \forall i, t, n \in \{2,3,\ldots,\tau_i\} \\
S_{k,t+1} &= S_{k,t} + \sum_{i\in\mathbf{I}^+_k} \rho_{ik}\beta_i \bar{W}^{\tau_i}_{i,t} + \sum_{i\in\mathbf{I}^-_k} \rho_{ik}\beta_i W_{i,t} + \\
&\quad \left(\xi_{k,t} - \sum_{i\in\mathbf{I}^+_k} \rho_{ik}\left(\beta_i(\hat{Z}^{\tau_i}_{i,t} - \hat{Y}^{\tau_i}_{i,t}) + \beta^P_{i,k,t}\right) + \sum_{i\in\mathbf{I}^-_k} \rho_{ik}\beta^C_{i,k,t}\right) && \forall k, t
\end{aligned} \tag{3.16}$$

With the disturbance

$$d(t) = \left[\xi_{k,t}, k\in\mathbf{K}, \quad \beta^P_{i,k,t}, \beta^C_{i,k,t}, i\in\mathbf{I}, k\in\mathbf{K}, \quad \hat{Y}_{i,t}, \hat{Y}^n_{i,t}, \hat{Z}_{i,t}, \hat{Z}^n_{i,t}, i\in\mathbf{I}, n\in\{2,3,\ldots,\tau_i\}\right]$$

the final model can be written in the state space form. In the general case, we have $u \in \{0,1\}^m$ in which $m = |\mathbf{I}|$; $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_b}$ in which $n_c = |\mathbf{S}|$ and $n_b = \sum_{i\in\mathbf{I}} \tau_i$; and $d \in \mathbb{R}^{n_d} \times \{0,1\}^{2n_b}$ in which $n_d = |\mathbf{S}| + \sum_{k\in\mathbf{K}} |\mathbf{I}_k^-| + |\mathbf{I}_k^+|$. The symbol $|\cdot|$ denotes the cardinality of a set.

The state space formulation of the scheduling model is denoted as $\mathbf{M}^{\text{MPC}}$.

3.3.6 Extensions

The main ideas in transforming the discrete-time scheduling model $\mathbf{M}^{\text{SCH}}$ to the state space form $\mathbf{M}^{\text{MPC}}$ is the identification of inputs and states, and the lifting of some decision variables (inputs) so that the state vector completely describes the system. This idea can be applied to any linear discrete-time model. For example, in this section, we show how variable batchsizes, backorders and processing constraints can be modeled in the state space formulation.

3.3.6.1 Variable Batchsizes

Let $B_{i,t} \geq 0$ denote the batchsize of task i that starts at time t. The material balances in terms of $B_{i,t}$ is

$$S_{k,t+1} = S_{k,t} + \sum_{i\in\mathbf{I}_k^+} \rho_{ik} B_{i,t-\tau_i} + \sum_{i\in\mathbf{I}_k^-} \rho_{ik} B_{i,t} + \xi_{k,t} \tag{3.17}$$

$$W_{i,t} B_i^{\min} \leq B_{i,t} \leq W_{i,t} B_i^{\max}, \qquad \forall k,t \tag{3.18}$$

The parameters $B_i^{\min}$ and $B_i^{\max}$ are the lower and upper bound on the batchsize of task i.

The scheduling model now consists of (3.1), (3.17) and (3.18). To formulate it in the state space form, notice that variable $B_{i,t}$ is a decision, and hence an input. Since the state equation (3.17) requires the input from $t-\tau_i$, we lift the batch-size input to fully describe the state of the system. Hence,

$$\bar{B}_{i,t+1}^1 = B_{i,t} \tag{3.19}$$

$$\bar{B}_{i,t+1}^n = \bar{B}_{i,t}^{n-1} \qquad \forall i,t \tag{3.20}$$

The model $\mathbf{M}^{\text{MPC}}$ now consists of Equations (3.4), (3.20), (3.21) and constraints (3.6) and (3.18).

$$S_{k,t+1} = S_{k,t} + \sum_{i \in \mathbf{I}_k^+} \rho_{ik} \bar{B}_{i,t}^{\tau_i} + \sum_{i \in \mathbf{I}_k^-} \rho_{ik} B_{i,t} + \xi_{k,t} \qquad \forall k, t \tag{3.21}$$

3.3.6.2 Backorders

If the demands cannot be met at a particular sampling time, then we model it using a backorder/ backlog variable. Let $U_{k,t}$ be the backlog of the material k during period t and $V_{k,t}$ be the shipment of material k during period t. The material balance now becomes,

$$S_{k,t+1} = S_{k,t} + \sum_{i \in \mathbf{I}_k^+} \rho_{ik} \bar{B}_{i,t}^{\tau_i} + \sum_{i \in \mathbf{I}_k^-} \rho_{ik} B_{i,t} - V_{k,t} \qquad \forall k \in \mathbf{K}, t \tag{3.22}$$

while the backlog U_{kt} is calculated from

$$U_{k,t} = U_{k,t} - V_{k,t} + \xi_{k,t} \qquad \forall k, t \tag{3.23}$$

From Equations (3.22) and (3.23), it is clear that the shipments $V_{k,t}$ are the decisions, and hence inputs while, backlogs are the states.

3.3.6.3 Processing constraints[8]

Processing constraints can be modeled following the same procedure. To illustrate, we consider the modeling of tasks that require the consumption of utilities $m \in \mathbf{M}$ (e.g.,cooling water and electricity) during their execution. If ω_m is the availability of utility m, and ψ_{im} is the utility consumption of task i during its execution, then the resource constraint is written as:

$$\sum_{i \in \mathbf{I}_m} \sum_{t;=t-\tau_i+1}^{t} \psi_{im} W_{it} \le \omega_m \qquad \forall m, t \tag{3.24}$$

in which $\mathbf{I}_m$ is the set of tasks that consume resource m. Using the lifted W variables the constraint (3.24) can be written as

$$\sum_{i \in \mathbf{I}_m} W_{i,t} \psi_{im} + \sum_{i \in \mathbf{I}_m} \sum_{n=1}^{\tau_i - 1} \bar{W}_{i,t}^{n} \psi_{im} \le \omega_m \qquad \forall m, t \tag{3.25}$$

In Section 3.4, we provide an additional example of modeling changeover time between two tasks.

[8]The text in this section appears in Subramanian et al. (2012a).

3.4 Illustrative Examples[9]

3.4.1 Nominal demand

We now present the rolling horizon optimization procedure for the simple scheduling example that was introduced in Figure 3.1. We make the following modifications to the scheduling problem: (i) Variable batch size ($\beta_i^{\min} = 5, \beta_i^{\max} = 10$), (ii) A changeover time of $\text{CHT}(i, i') = 2h$ when switching between products, and (iii) Nominal demands $\xi_{k,t}^{\text{nom}} = 1.5$ ton every hour. No backlogs are allowed.

To model the changeover time, we introduce three new binary variables $Z_{i,i',t}$, $Y_{i,t}$ and $X_{i,t}$. The binary variable $Z_{i,i',t}$ is 1 when a changeover is effected from task i to task i' at time t. The binary variable $Y_{i,t}$ is 1 if the task i was started during $[t-\tau_i, t]$. The binary variable $X_{i,t}$ is 1 if the last task to be performed in the unit before time t was i. The modified assignment equations are given in (3.26) (for the example problem, hence j is omitted as there is only one uit)

$$
\begin{aligned}
&\sum_{i \in \mathbf{I}} \sum_{t'=t-\tau_i+1}^{t} W_{i,t'} + \sum_{\substack{i' \in \mathbf{I} \\ i' \neq i}} \sum_{t'=t-\text{CHT}(i,i')+1}^{t} Z_{i,i',t'} \leq 1 && \forall t \\
&\sum_{t'=t-\tau_i+1}^{t} W_{i,t'} = Y_{i,t} && \forall t, \forall i \in \mathbf{I} \\
&X_{i,t} \geq Y_{i,t} && \forall t, \forall i \in \mathbf{I} \\
&\sum_{i \in \mathbf{I}} X_{i,t} = 1 && \forall t \\
&Z_{i,i',t} \leq X_{i,t-1} && \forall t, \forall i \in \mathbf{I}, i' \in \mathbf{I}, i' \neq i \\
&Z_{i,i',t} \leq X_{i',t} && \forall t, \forall i \in \mathbf{I}, i' \in \mathbf{I}, i' \neq i \\
&Z_{i,i',t} \geq X_{i,t-1} + X_{i',t} - 1 && \forall t, \forall i \in \mathbf{I}, i' \in \mathbf{I}, i' \neq i
\end{aligned}
\tag{3.26}
$$

In the state space format, the variables $Z_{i,i',t}$ and $X_{i,t}$ are the inputs. As we see in the modified assignment equation (3.26), the state of the plant is jointly described by the inputs Z, W and X from the previous time periods and the current input. Therefore, apart from lifting W,

[9] The results in this section appears in Section 5 and Section 6.4 of Subramanian et al. (2012a).

we also lift Z and X.

$$\bar{Z}^1_{i,i',t+1} = Z_{i,i',t} \qquad \bar{Z}^2_{i,i',t+1} = \bar{Z}^1_{i,i',t}, \qquad \forall i, i' \in \mathbf{I}, t$$

and

$$\bar{X}^1_{i,t} = X_{i,t} \qquad \forall i \in \mathbf{I}, t$$

The variable Y is just a function of the lifted states $\bar{W}^n_{i,t}$ and the input.

$$Y_{i,t} = \sum_{n=1}^{\tau_i - 1} \bar{W}^n_{i,t} + W_{i,t}$$

The state space representation of the scheduling model, following the example in the previous section, can be written in the familiar format

$$x(t+1) = Ax(t) + Bu(t) + B_d d(t)$$

with constraints

$$\underline{b} \le E_x x(t) + E_u u(t) \le \bar{b}$$

and economic stage cost

$$\ell_E(x(k), u(k)) = q' x(k) + r' u(k)$$

The on-line optimization problem in its simplest form is now written as:

$$\begin{aligned}
\mathbb{P}_N(x) : \min_{\mathbf{u}} & \sum_{t=0}^{N-1} \ell_E(x(t), u(t), d^{\text{nom}}(t)) \\
& \text{s.t. } x(t+1) = Ax(t) + Bu(t) + B_d d^{\text{nom}}(t), \qquad t = \{0, 1, \ldots, N-1\} \\
& \underline{b} \le E_x x(t) + E_u u(t) \le \bar{b}, \qquad t = \{0, 1, \ldots, N-1\} \\
& x(0) = x
\end{aligned} \tag{3.27}$$

in which N is the prediction horizon and $d^{\text{nom}}(t)$ is the nominal demand, while $\mathbf{u} = (u(0), u(1), \ldots, u(N-1))$.

The Gantt chart obtained by successive re-optimization of Problem (3.27) is shown in Figure 3.3. Note that there were no disturbances to the system. As it can be seen, the optimization

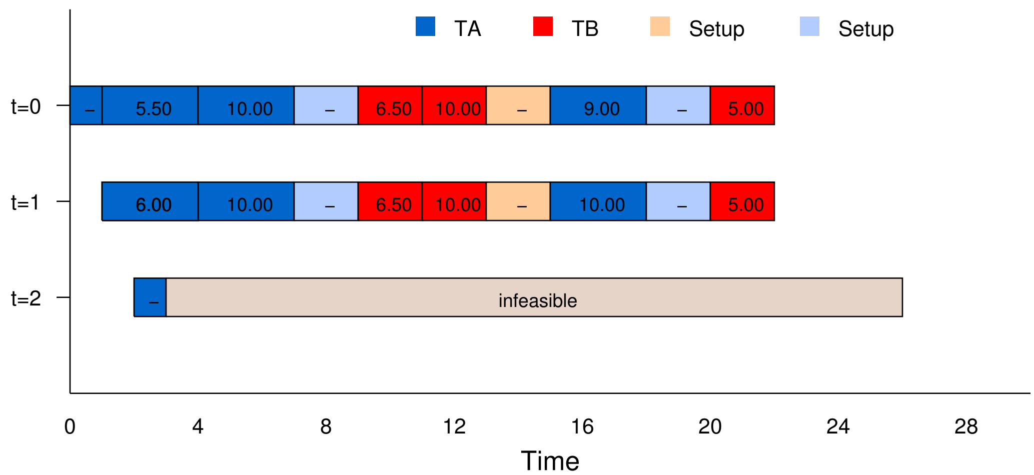

Figure 3.3: Rescheduling leads to infeasibility when no backorders are allowed

problem becomes infeasible at $t = 2$. Notice that optimization problem (3.27) aims to minimize the number of batches started as well as the inventory at each prediction time. Hence, it starts a batch of 6 ton for task TA at $t = 1$. However, for the optimization problem at time $t = 2$, there were not enough degrees of freedom to satisfy the demands observed $t = 25$, which was not considered by the problem at $t = 1$. Hence, when solved within a rolling horizon framework, Problem (3.27) does not guarantee recursive feasibility. As we show later in this section, the problem would have remained feasible if a larger batch was started at $t = 1$. In the control literature, recursive feasibility is achieved by enforcing terminal conditions, as the terminal conditions account for long term effects. To find a suboptimal infinite horizon schedule for this problem, we solve the following periodic optimization problem given in (3.28). In the periodic optimization problem, we enforce the condition that $x(0) = x(N_p)$, which says that the state of the system (including all the lifted variables) return to the starting state at the end of the period N_p. Therefore, the same schedule can be repeated at $t = N_p$. In this way, we can find an infinite horizon schedule in response to nominal demands.

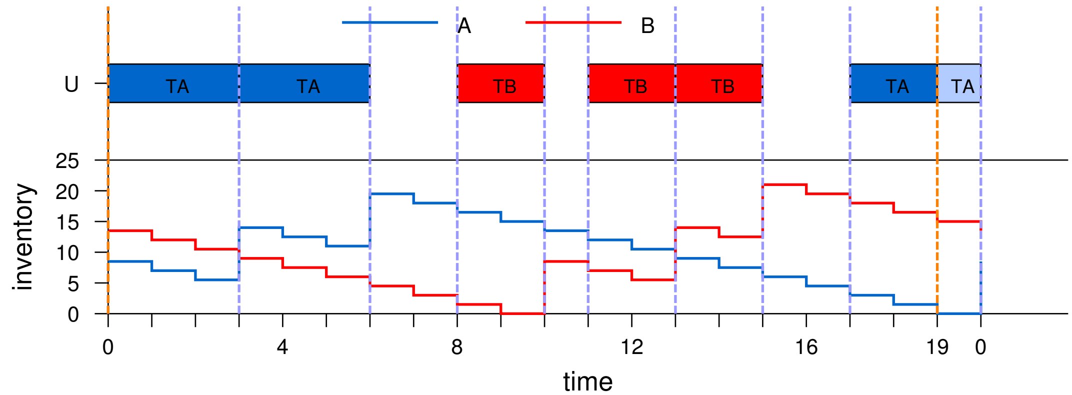

Figure 3.4: Periodic solution for the example in the absence of disturbances

$$
\begin{aligned}
\mathbb{P}_P : \min_{\mathbf{u},x(0)} & \sum_{t=0}^{N_P-1} \ell_E(x(t),u(t),d^{\text{nom}}(t)) \\
\text{s.t. } & x(t+1) = Ax(t) + Bu(t) + B_d d^{\text{nom}}(t), \qquad t = \{0,1,\dots,N-1\} \\
& \underline{b} \le E_x x(t) + E_u u(t) \le \bar{b}, \qquad t = \{0,1,\dots,N-1\} \\
& x(0) = x(N_p)
\end{aligned} \tag{3.28}
$$

The Gantt chart for the periodic solution is shown in Figure 3.4.

We now illustrate the use of terminal constraints on the states of the model which enable us to retain feasibility of the scheduling problem as we roll the horizon forward. To do so, we use the cyclic schedule found by optimization problem (3.28). Let the solution of the (3.28) be given as $(x_P^0(0), u_P^0(0), u_P^0(1), \dots, u_P^0(N_p-1))$. For this optimal periodic solution, we can calculate the corresponding state-evolution using the state evolution equation. Denote the states in this optimal periodic state evolution as $\{x_P^0(0), x_P^0(1), \dots, x_P^0(N_p-1)\}$. Then the optimization problem with terminal constraints $\mathbb{P}_N^T(x)$ can be written as:

$$\begin{aligned}
\mathbb{P}_N^T(x) : \min_{\mathbf{u}} & \sum_{t=0}^{N-1} \ell_E(x(t), u(t), d^{\text{nom}}(t)) \\
\text{s.t. } & x(t+1) = Ax(t) + Bu(t) + B_d d^{\text{nom}}(t), t = \{0, 1, \ldots, N-1\} \\
& \underline{b} \le E_x x(t) + E_u u(t) \le \bar{b}, \qquad t = \{0, 1, \ldots, N-1\} \\
& x(0) = x \\
& x(N) \in \left\{x_P^0(0), x_P^0(1), \ldots, x_P^0(N_p - 1)\right\}
\end{aligned} \tag{3.29}$$

In optimization problem $\mathbb{P}_N^T$, we enforce the condition that the terminal state be one of the states in the optimal periodic state evolution. Therefore, the solutions to (3.29) contains long-term information. This is because, we terminate at a state from which we can implement a periodic solution to respond to nominal demands. The Gantt chart obtained by successive re-optimization using problem (3.29) is shown in Figure 3.5. Due to design of the optimization problem, we remain feasible at all times. Note that the batch of TA at $t = 1$ was 8 tons, because of the information regarding future demands contained in the terminal constraint. Figure 3.6 shows the closed-loop solution over 24h.

We can use terminal constraints to reduce the computational burden in scheduling because, as shown in Figure 3.7, we can guarantee recursive feasibility using shorter prediction horizons also. In Figure 3.7, Problem (3.29) was solved with $N = 12$ and Figure 3.8 shows the closed-loop solution over a horizon of 24h using a prediction horizon of $12h$.

3.4.2 Rescheduling

In this section, we consider the same model as in Section 3.4.1, but with backlogs. That is, we introduce shipments $V_{k,t}$ and backlog $U_{k,t}$ with the new state equations defined by (3.22) and (3.23). We enforce an economic penalty on accumulating backlogs.

The following disturbances are observed: (i) Production delay of 1h at $t = 6$, (ii) breakdown for 3h from $t = 10$ to $t = 13$, (iii) unloading error at $t = 14$ (production of B is 8ton instead of 10ton) and, (iv) demand spike at $t = 16$, with $\hat{\xi}_i = 0.5$, that is the demand for both products were 2tons instead of the nominal demand of 1.5 ton. Figures 3.9 and 3.10 shows the closed-loop

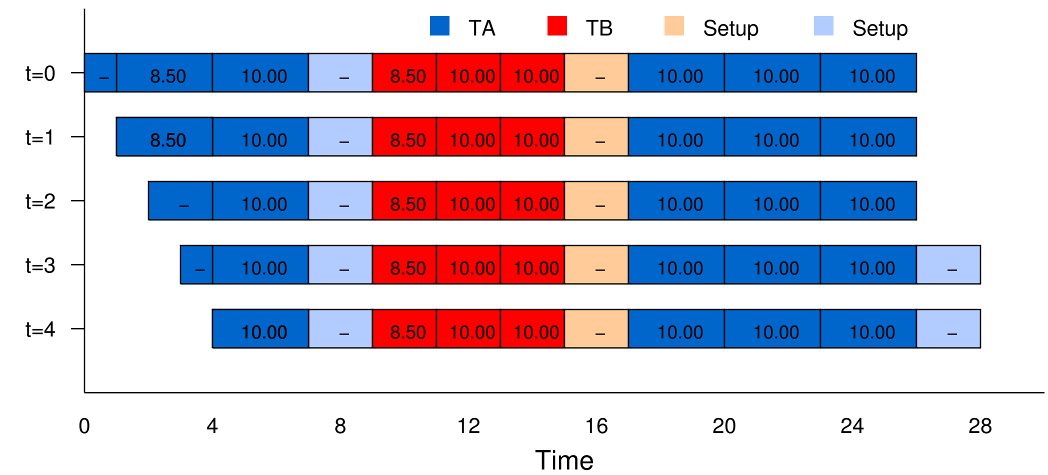

Figure 3.5: Solutions obtained by solving Problem (3.29) at $t = 0, 1, 3$ and 4. Addition of terminal constraints leads to feasible problems. Compare the schedule at $t = 0$ with 3.3. A larger batch of TA starts at $t = 1$ and there are fewer changeovers, thus larger production.

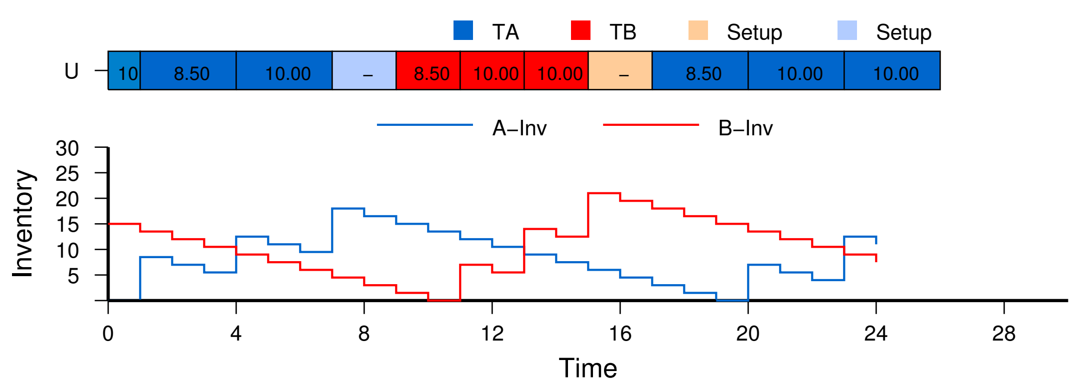

Figure 3.6: Closed-loop solution solving (3.29) with $N = 24h$

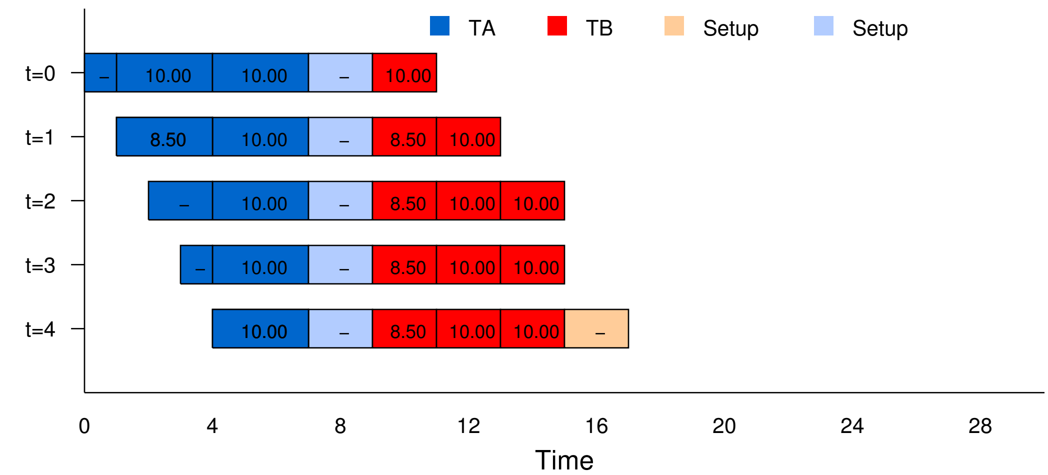

Figure 3.7: Solutions obtained by solving Problem (3.29) at $t = 0, 1, 3$ and 4. Recursive feasibility is maintained with proper choice of terminal constraints

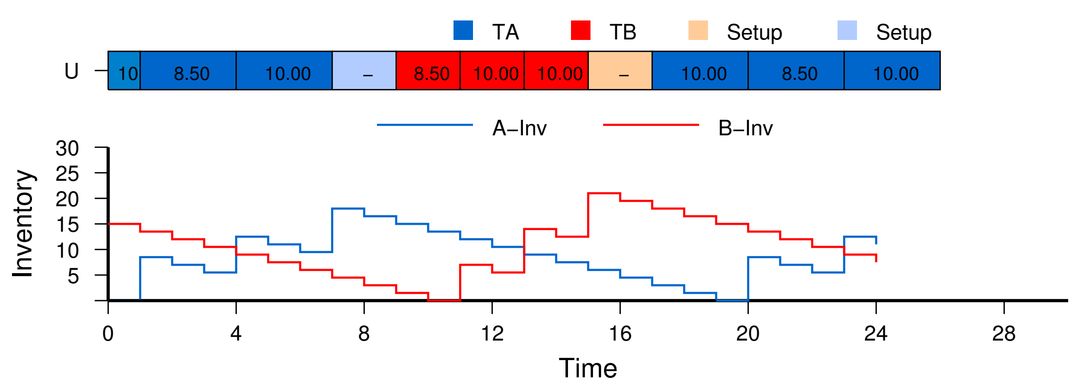

Figure 3.8: Closed-loop solution solving (3.29) with $N = 12h$

performance for an optimizer optimizing (3.27). We can observe the rescheduling that occurs naturally using the rolling horizon framework in Figure 3.9. For example, the batch sizes change between $t = 6$ and $t = 7$ and $t = 14$ and $t = 15$ after the realization of disturbances. Finally, Figure 3.11 shows the closed-loop response for the same disturbances, but when the optimizer was solving (3.29); that is, when we were enforcing terminal cyclic constraints. We observe inherent robustness of the terminal constraint formulation as the backlogs accumulated are lesser than the formulation without terminal constraints.

3.5 Discussion

3.5.1 Generality of the scheduling model[10]

Most current approaches to reactive scheduling are based on scheduling models that do not include disturbances explicitly. Thus, when an event triggers rescheduling, an empirical procedure is followed to modify the scheduling model so it (i) represents the new state of the system, and (ii) accounts for the future impact of the disturbance. One advantage of the state space model is that the same model can be directly used for rescheduling. All events that can trigger rescheduling are modeled via disturbance variables. Thus, for a resolve it is sufficient to fix the appropriate disturbance variables, which can be readily calculated from the observation at the current time. No empirical model modifications are necessary.

3.5.2 Stochastic vs. deterministic approaches[11]

In this paper, we treat scheduling as an on-line problem; we optimize to determine a single schedule based on current data and forecasts, and as new information becomes available we re-optimize to determine, again, a single solution. In each re-optimization, we solve a deterministic problem- we do not account for the fact that some data are subject to uncertainty that can be modeled. An alternative approach is to model the uncertainty in the events that can trigger rescheduling, and then generate a solution that takes into account this information

[10] This text appears in Section 3.5.1 of Subramanian et al. (2012a)
[11] This text appears in Section 3.5.2 of Subramanian et al. (2012a)

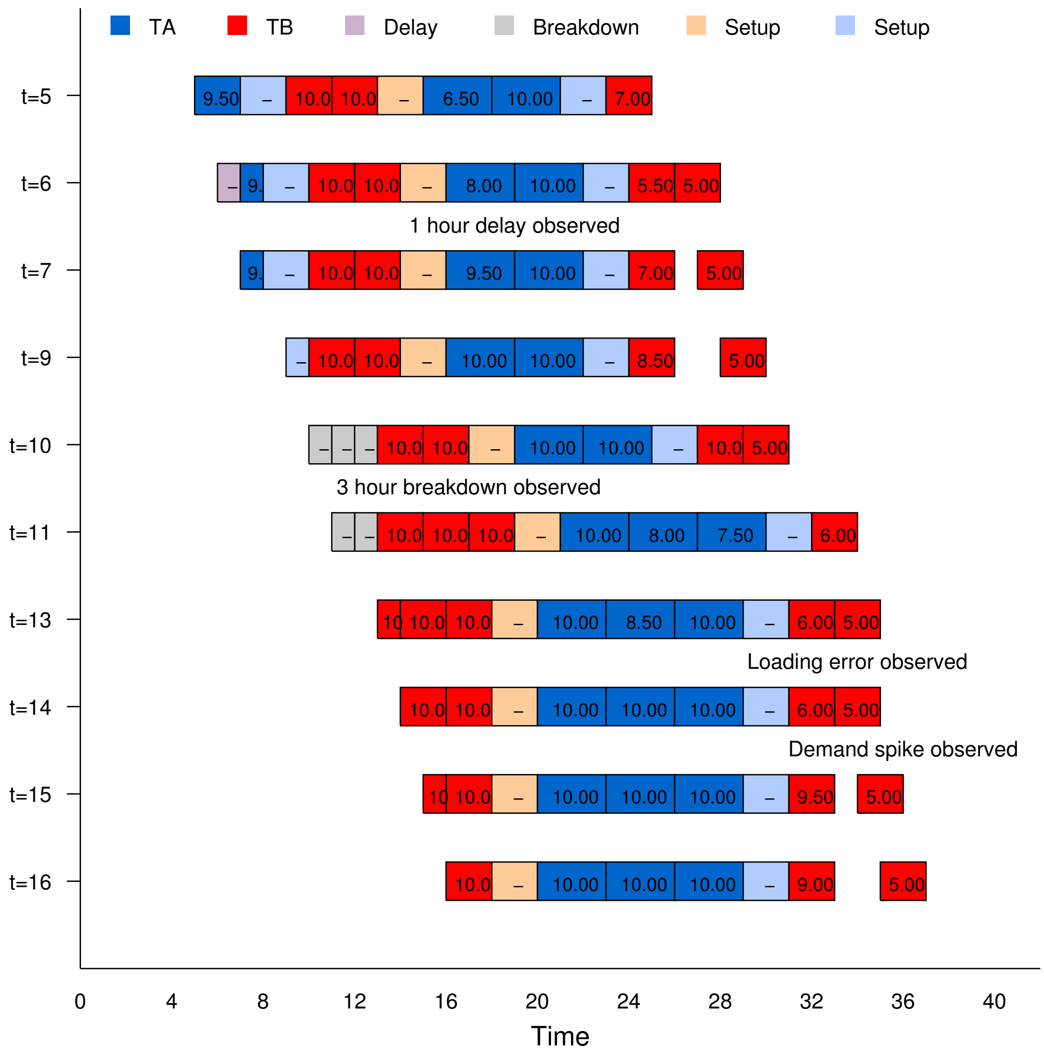

Figure 3.9: Rescheduling in the presence of disturbances

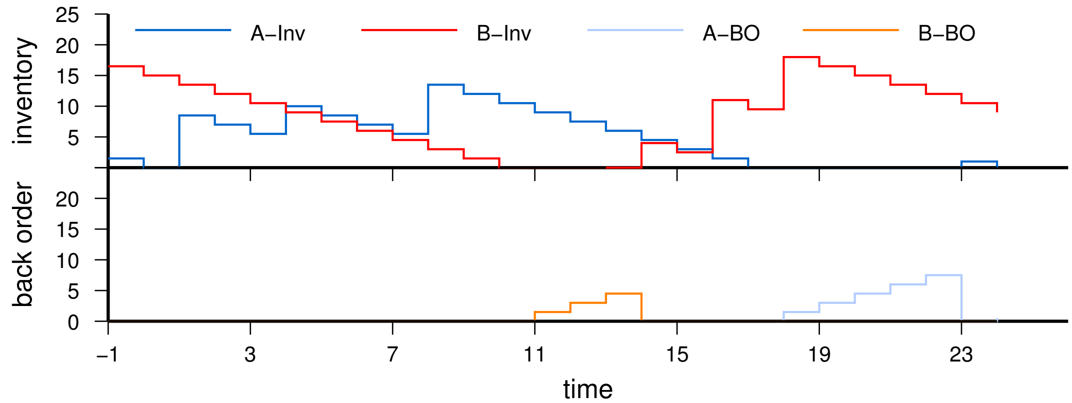

Figure 3.10: Inventory (Inv) and Backorder(BO) profiles in the closed-loop in the presence of disturbances.

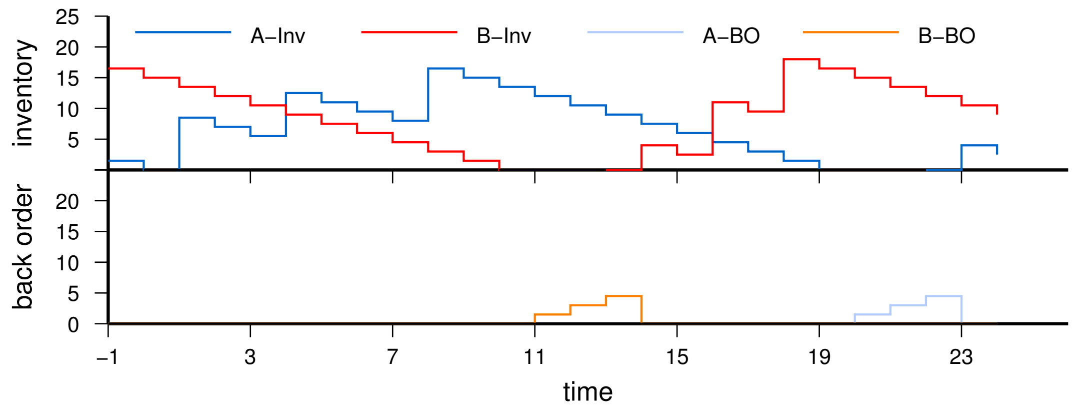

Figure 3.11: Inventory (Inv) and Backorder(BO) profiles in the closed-loop in the presence of disturbances. Compare with 3.10 to notice the inherent robustness of the terminal constraint formulation

(Li and Ierapetritou, 2008; Sahinidis, 2004). The obvious advantage of this so-called *optimization under uncertainty* approach is that, if the model is solved effectively, then it can lead to better solutions. The disadvantage is that most optimization under uncertainty methods are computationally expensive, and thus cannot be used to address practical problems. Robust optimization methods have been proposed to address this shortcoming (Ben-Tal and Nemirovski, 2002; Verderame and Floudas, 2009); they rely on solutions that is almost as hard as the deterministic problem, but leads to solutions that are conservative. The advantage of the control-inspired approach that we follow is that the deterministic optimization problem can be solved more effectively, which has three implications. First, it can lead to optimal (or near optimal) solutions that are better (even when evaluated under uncertainty) than the solution that can be obtained within the same time by a stochastic programming approach. Second, it allows us to reschedule more frequently, thus reacting faster to disturbances and thereby resulting in better closed-loop solution. Third, it allows us to consider longer scheduling horizons, which is critical and can often be more important that accounting for uncertainty.

3.5.3 Types of disturbances and uncertainties[12]

Interestingly, the types of disturbances we have discussed correspond, when treated as stochastic parameters, to different types of uncertainty. Shipment disturbances can be viewed as right-hand side (RHS) uncertainty. Production and consumption disturbances can be viewed as left-hand side (LHS) uncertainty, since they can be treated as uncertainties in the $\beta_{ik} = \rho_{ik}\beta_i$ terms:

$$S_{k,t+1} = S_{k,t} + \sum_{i \in \mathbb{I}^+} (\beta_i \bar{W}^n_{i,t} + \beta^P_{i,k,t})\rho_{ik} + \sum_{i \in \mathbb{I}^-} (\beta_i W_{i,t} + \beta^C_{i,k,t})\rho_{ik} + \zeta_{k,t} \qquad \forall k, t \tag{3.30}$$

These are two types of uncertainty that have received the most attention in stochastic optimization approaches to scheduling.

Task delays can also be treated as LHS uncertainty if the duration of a task appears only as a LHS coefficient (in the case of fixed processing times) or as a variable defined in terms

[12]This text appears in Section 3.5.3 of Subramanian et al. (2012a). Equation (3.30) has been modified to remain consistent with the notation used in this chapter

of stochastic parameters (in the case of variable processing times). Precedence-based models or time-grid based models with continuous modeling of time may result in stochastic optimization problems which lead to LHS uncertainty. However, in discrete-time formulations, the number of terms included in the summation in the LHS of the assignment constraint depends on the duration of a task. Thus, in this case, the treatment of task delays through the modeling of processing times as stochastic parameters leads to a *structural* type of uncertainty.

Finally, unit breakdowns lead to structural uncertainty since the constraints used to model resource constraints should be removed or modified. Stochastic optimization approaches cannot be used to effectively address this type of structural uncertainty because, in addition to requiring on-the-fly reformulations of the scheduling model, task delays and unit breakdowns lead to problems with either *purely endogenous* uncertainty or *exogenous uncertainty with endogenous observation* (Colvin and Maravelias, 2008, 2010; Goel and Grossmann, 2006). For example, the probability and the timing of a unit breakdown depends on the utilization of the unit, which is determined by the decision maker. The proposed approach does not suffer from this limitation.

3.5.3.1 Reverse transformation and reoptimization[13]

Since scheduling MIP models are computationally expensive, a potential disadvantage of the proposed state space modeling framework is that it leads to MIP models of larger size. For example. compared to its counterpart model, $\mathbf{M}^{\mathrm{SCH}}$, model $\mathbf{M}^{\mathrm{MPC}}$, has additional lifted variables (lifted inputs $\bar{W}_{i,t}^{n}$) and equations (3.4). However, model $\mathbf{M}^{\mathrm{MPC}}$ is not significantly slower than $\mathbf{M}^{\mathrm{SCH}}$. First we note that the addition of disturbance variables in Equations (3.16) and (3.15) leads to changes in RHS constant vector b, if the optimization model is written as $\max\left\{c'x : Ax \leq b.x \in \mathbf{X}\right\}$. In other words, they do not increase the complexity of the model. Second, the lifting equations can be used to project out variables after the state of the system is updated using $\mathbf{M}^{\mathrm{MPC}}$ and before reoptimization is performed. Commercial MIP solvers perform this type of preprocessing (variable elimination and constraint removal) automatically

[13]This text appears in Section 3.5.2 of Subramanian et al. (2012a)

(Atamtürk and Savelsbergh, 2005). Third, specific preprocessing methods can be easily developed to transform the *current* $\mathbf{M}^{\text{MPC}}$ model (i.e. the model after the injection of disturbances at t) back to a model in the form $\mathbf{M}^{\text{SCH}}$. Perprocessing based on $\mathbf{M}^{\text{MPC}}$ can also be used to automatically detect what constraints should be removed or modified. For example. in the case of a breakdown of unit $j = U$ from t_1 to t_2, the constraint (3.15) becomes:

$$\sum_{i \in \mathbf{I}_j} W_{i,t} + \sum_{i \in \mathbf{I}_j} \sum_{n=1}^{\tau_i - 1} \bar{W}_{i,t}^{n} \leq 0 \qquad \forall t \in \{t_1, \ldots, t_2 - 1\}$$

which mathematically implies that all binary variables appearing in the LHS should be fixed to zero and Equation (3.15) be removed from the model for $j = U$ and $t \in \{t_1, \ldots, t_2 - 1\}$. Not surprisingly, this is what one would do using logical arguments: if there is a breakdown in an unit, then no tasks can be started on this unit (i.e., all binaries are set to zero), and thus the corresponding assignment constraint can be removed. Model $\mathbf{M}^{\text{MPC}}$ allows us to systematically perform this type of reasoning. We note that our preliminary computational experience confirms that $\mathbf{M}^{\text{MPC}}$ is computationally comparable with model $\mathbf{M}^{\text{SCH}}$.

Finally, lifting the inputs offers insights into generation of the models for rescheduling. As we saw, the current state of the system includes past inputs from $t - \max_i \tau_i$ to t. Hence, when reformulating original model M^{SCH}, we have to fix the decisions made in the last $\max_i \tau_i$ periods. This approach has been proposed in the past (Sundaramoorthy and Maravelias, 2010). The use of state space models and input lifting formalizes this approach and makes it easy to write rescheudling model for any disturbance.

3.5.4 MPC tools[14]

The development of a state space model is a first step towards the use of MPC theory and methods to address scheduling problems. It offers a representation of scheduling problems with which the process control community is familiar. Since scheduling is a dynamic problem and can be viewed as a *production* control problem, our hope is that the proposed framework will enable MPC technology for scheduling problems.

[14]This text appears in Section 6.1 of Subramanian et al. (2012a)

Another outcome is that if facilitates the application of methods that have been developed for hybrid dynamic systems consisting of both time and event driven dynamics (Bemporad and Morari, 1999; Heemels, De Schutter, and Bemporad, 2001). Control of such systems has been the focus of many researchers in the past decade (see (Camacho, Ramirez, Limon, Muñoz de la Peña, and Alamo, 2010) for a review of MPC techniques for hybrid systems). At the same time, studying this new class of problems can lead to the development of new tools for hybrid system.

The state space model can also help to bridge the gap between scheduling and control since it allows the formulation of the integrated scheduling-control problem using a state space model. Furthermore, the unified problem can be viewed as an economic MPC problem in which the process economics (primarily determined by the scheduling decisions) are directly optimized by the controller (Diehl, Amrit, and Rawlings, 2011).

More importantly. it offers a natural framework for the development of new scheduling algorithms based on MPC. As mentioned in Section 3.1, scheduling is still thought of as an open-loop problem, even though it is used in an iterative manner. As a results, concepts such as stability, recursive-feasibility, and closed-loop performance have received no attention. In Section 3.4, we showed how terminal constraints can be used to guarantee recursive feasibility.

Chapter 4

Distributed MPC for supply chain optimization

We propose cooperative model predictive control for supply chains in this chapter.

In Section 4.2, we provide a brief review of the different control theory based and distributed decision making approaches to supply chain optimization and operation. In section 4.3, we describe the dynamic modeling of supply chains. In section 4.4. we implement cooperative MPC on a single-product, two-echelon supply chain. Finally, we summarize our results in Section 4.5.

4.1 Introduction[1]

The supply chain is a system comprising organizations, decision makers, and technology decision policies that is responsible for transforming raw materials into finished products that are delivered to end customers. As expanded upon later in this paper, the supply chain is traditionally characterized by counter-current flows of information and material. Material flows from the raw material suppliers through the production and distribution facilities to the end customers, while information, in the form of demands and orders, flows from the end customers upstream to the suppliers (Backx, Bosagra, and Marquardt, 1998; Beamon, 1998).

The decisions for supply chain management can be broadly classified into three categories: strategic, tactical and operational. The strategic decisions are the long term planning decisions that may include, among others, where to locate production facilities and warehouses, and in which technologies to invest. On a medium time range, tactical decisions include selecting

[1] The text in this section appears in Section 1 of Subramanian et al. (2012b).

supply chain partners such as raw material suppliers, transportation companies, etc. The operational decisions are the short term decisions, which are related to optimally operating the supply chain. These decisions include planning and scheduling in the production facilities, and distribution decisions such as inventory management, ordering and shipping policies, etc. (Shah, 2005; Ganeshan and Harrison, 1995).

Shapiro (2004) lists the challenges in enlarging the scope of strategic planning in supply chains. Among the listed challenges are integrating manufacturing, purchase and sales decisions, multiperiod analysis and optimizing the overall supply chain profits. Stadtler (2005) is an excellent overview paper about advanced planning in supply chains. The authors emphasize linking organizational units to improve competitiveness of the supply chain. However, from an operational viewpoint, they focus on advanced planning systems (APS) that uses information and communication technology to coordinate all the flows (material, information, financial) in the supply chain to best improve customer satisfaction.

The combined strategic and operational planning is a challenging optimization problem, but researchers have made efforts to solve it; see, for instance, (Sabri and Beamon, 2000; Tsiakis, Shah, and Pantelides, 2001; You and Grossmann, 2008). The optimization problems formulated for combined strategic and operational planning typically involve selecting a supply chain network from a family of networks or a network superstructure. Recent developments in combined strategic and operational planning, including handling of uncertainties and multi-objective formulations, are described in the review paper (Papageorgiou, 2009).

At the operational level of the supply chain, the need for simultaneous decision making at the manufacturing and the distribution sites to operate a *coordinated* supply chain has been recognized. The focus of this chapter is on methods to achieve such simultaneous decisions. This simultaneous decision making is also known as enterprise wide optimization (Grossmann, 2005).

Modern supply chains operate over multiple locations and products, and are highly interconnected. In a competitive economy, neglecting these interactions may result in lower profits. A central coordinator who controls the supply chain can account for these interactions and provide optimal operation. However, centralized coordination may not always be practical for a supply chain as (i) different nodes may belong to different firms, (ii) there may be a conflict of objectives among nodes (iii) information sharing may not be perfect and (iv) a centralized decision maker is the most vital cog in a supply chain, and its failure may be catastrophic for the supply chain. Therefore, distributed coordination structures for supply chain operation is needed.

We focus on tailoring model predictive control (MPC) as a general purpose method for optimal supply chain operation. Model predictive control uses a dynamic model of the system to predict future outcomes and solves a constrained optimization problem over the predicted outcomes to find the best operational decisions. Therefore, it is well suited as a basis for supply chain operation because it makes full use of the dynamic model and knowledge of the interactions between the various nodes to predict and optimize an overall supply chain objective function.

We propose cooperative MPC as a tool for coordinating supply chains as it retains the same structure as traditional supply chains wherein each node makes its own local decisions, but instead of optimizing the local objective functions, the nodes optimize the overall supply chain objective function.

4.2 Literature survey[2]

A well defined supply chain optimization model requires a detailed dynamic description of the supply chain and an objective function that captures all the essential costs and trade-offs in the supply chain. Beamon (1998) classifies supply chain modeling in four broad categories: Deterministic models where all the parameters are known, stochastic models with at least one unknown parameter (typically demands) that follows a known probability distribution, economic game theory based models, and simulation based models. As pointed out in (Sarimveis, Patrinos, Tarantilis, and Kiranoudis, 2008), a majority of these models are steady-state models based on average performance, and hence are unsuitable for dynamic analysis. In the review

[2]The text in this section appears in Section 2 of Subramanian et al. (2012b)

of dynamic models for supply chains, Riddalls, Bennett, and Tipi (2000) classify the models as continuous time models, discrete time models, discrete event simulations, and operations research (OR) based models.

The pioneering work of "industrial dynamics" awakened the control community's interest in supply chain optimization. The industrial dynamics models are the continuous (and discrete) time dynamic models mentioned in (Angerhofer and Angelides, 2000). Industrial dynamics captures the dynamics of the supply chains using differential (or difference) equations, and therefore, control theory is a natural choice to study supply chain dynamics. In their simplest form, these models capture inventory dynamics based on the shipments and orders leaving the node

$$\mathrm{Iv}_i(k) = \mathrm{Iv}_i(k-1) - \sum_{j \in \mathrm{Dn}(i)} S_{ij}(k) + \sum_{j \in \mathrm{Up}(i)} S_{ji}(k - \tau_{ji})$$

in which $\mathrm{Iv}_i(k)$ is the inventory in node $i \in \mathscr{I}$ at discrete time k, $\mathrm{Dn}(i)$ is the set of nodes to which node i ships material and $\mathrm{Up}(i)$ is the set of nodes from which node i receives material. The shipment delay between nodes i and j is denoted τ_{ij}, and S_{ij} is the amount shipped by node i to node j.

In order to compare different methods of supply chain operation, supply chain performance has to be quantified. Beamon (1999, 1998) classify the performance measures in a supply chain as quantitative measures like cost minimization, profit maximization, customer response time minimization and, qualitative measures like customer satisfaction, flexibility etc. An important performance measure that supply chain operation strives to reduce is the *bullwhip effect*, which is defined as the amplification of demand fluctuations as one moves upstream in the supply chain. It has been observed that the orders placed by a node to its upstream nodes amplify (with respect to the customer demand) as one moves towards the supplier in a supply chain. This effect increases the cost of operating the supply chain. It has been estimated that a potential 30 billion dollar opportunity exists in streamlining the inefficiencies of the grocery supply chain, which has more than 100 days of inventory supply at various nodes in its supply chain (Lee, Padmanabhan, and Whang, 1997b,a). Among the reasons cited for the bullwhip effect is information distortion as one moves upstream in the supply chain. Information sharing has been shown to alleviate the bullwhip effect and is part of industrial practice such as vendor managed inventory (VMI), etc. The other reasons often cited for the bullwhip effect are: (i) the misunderstanding of feedback, which occurs because the nodes do not understand the dynamics of the supply chain, and (ii) the use of local optimization without global vision, in which each node tries to maximize its local profit without accounting for the effects of its decisions on the other nodes in the supply chain (Moyaux, Chain-draa, and D'Amours, 2007). Centralized operation of supply chains is best suited to mitigate bullwhip effect, as it has exact knowledge of the dynamics and complete information.

Classical control theory

The earliest applications of control theory to supply chains involved studying the transfer functions and developing single input single output (SISO) controllers for tracking inventory to its targets. Frequency domain analysis was used to analyze and evaluate alternative supply chain designs. In classical control approaches to controlling supply chain, the nodes were analyzed as linear systems using Laplace and Z-transforms. In the work of Towill (1982), a block diagram based approach to modeling a node was proposed. The single product node consisted of two integrators to capture the dynamics of inventory and backorders, while the order rate was the manipulated variable. The disturbance to the system, market demand, was incorporated in a feed-forward manner in the model. Time delays were also incorporated in the model. A feedback control law was proposed for controlling the inventory deviations from a target inventory. By varying some of these parameters like delay, controller gain etc., a family of models for a single node called as the input-output based production control system (IOBPCS) can be studied (Lalwani, Disney, and Towill, 2006). The feedback law, in its simplest form, takes the form of an order-up-to policy, that is order up-to the inventory target, if the current inventory is below its target. This policy can be viewed as a saturated proportional controller, although other forms of the controller can also be studied. Upon having a control policy and after defining other system details like delays, forecast smoothing etc, the transfer function of the node can be derived and analyzed (Dejonckheere, Disney, Lambrecht, and Towill, 2003). White (1999); Wikner, Naim, and Towill (1992) developed a PID controller without feed-forward forecasting for the node. A review of stability analysis for the IOBPCS family of models is presented in Disney, Towill, and Warburton (2006).

Classical control theory has also been studied for controlling the dynamics of the entire supply chain as well. Grubbström and Tang (2000) provides a review of the input-output modeling of supply chains and its analysis using Laplace transforms. Input-output modeling is the matrix form description of the supply chain dynamics. Burns and Sivazlian (1978); Wikner, Towill, and Naim (1991) analyzed multiechelon supply chains using the block diagram based approach. They analyzed the effect of ordering policies, delays and information availability at the nodes to analyze the supply chain response and bullwhip effect. Burns and Sivazlian (1978) used Z-transforms in their approach and found that information distortion led to bullwhip effect. Wikner et al. (1991) found out that information sharing and echelon inventory policies (in which each echelon considers inventory in all the nodes downstream to it) can mitigate bullwhip effect. Perea López, Grossmann, Ydstie, and Tahmassebi (2001); Perea López, Grossmann, Ydstie, and Tahmassebi (2000) have developed a continuous time model to describe a supply chain. The model is similar to deterministic supply chain models but uses differential equations to track dynamics. They simulated the model using a heuristic shipping policy and studied the closed-loop supply chain under three different proportional controllers for placing orders. They developed controllers to track inventory, backorder or a combination of both. The objective of the paper was to demonstrate that the model was capable of capturing the dynamics. Hence, they did not suggest any tuning methods for the controllers. Lin, Wong, Jang, Shieh, and Chu (2004) presented an approach to analyze the closed-loop stability of a supply chain and an approach for controller synthesis using a transfer function approach. The controller policy and shipping policy were similar to the Perea López et al. (2001) paper. They analyzed stability considering three extreme closed-loop scenarios: (i) high inventories and infinite replenishment from upstream nodes (infinite production),(ii) a low inventory and infinite replenishment from upstream nodes and (iii) limited production/supply. The effect of controller gains on the bullwhip effect was also analyzed. The authors proposed a controller tuning criterion based on frequency domain analysis of the Z-transfer functions. Venkateswaran and Son (2005) also studied the supply chain response using Z-transform and derived stability conditions for the supply chain. Hoberg, Bradley, and Thonemann (2007) applied linear control theory on a two-echelon supply chain and concluded that order-up-to policy based on inventory on hand can lead to instabilities. They found that the use of an echelon policy provides the best performance. Dejonckheere, Disney, Lambrecht, and Towill (2004) studied information enrichment where in, each node receives the final customer demand as well as the orders placed by its downstream nodes, using a linear control theory based approach and concluded that information enrichment is beneficial to the supply chain. Papanagnou and Halikias (2008) used a proportional controller to place orders and analyzed the bullwhip effect by estimating the state covariance matrix, for a supply chain responding to a random demand (modeled as a white noise) at the retailer node.

Sarimveis et al. (2008); Ortega and Lin (2004) provide extensive reviews of classical control approaches to supply chain design and operation.

Stochastic optimal control

Stochastic optimal control has been used to obtain ordering policies that minimize the expected costs of a node responding to random demands. We assume that the probability distribution of the demand is given. In its simplest form, the inventory control problem can be formulated as a dynamic optimization problem. The order-up-to policy is one such policy that is obtained by solving the dynamic optimization problem. The single node inventory control problem can be cast as a Markov decision problem. See Puterman (2005) for details on setting up the problem and algorithms. The order-up-to policy is optimal for independent and identically distributed demands as shown in the seminal paper by Clark and Scarf (1960). By considering set-up costs, it can be shown that the $(\sigma, \Sigma), \sigma < \Sigma$ policy, in which the node orders $\Sigma - \text{Iv}$ whenever the inventory Iv falls below σ, is the optimal policy for an infinite horizon problem; see for instance (Veinott, 1996; Iglehart, 1963; Federgruen and Zipkin, 1984). Optimality of similar policies have been shown for Markovian demands (Song and Zipkin, 1993; Sethi and Cheng, 1997), compound Poisson and diffusion demands (Bensoussan, Liu, and Sethi, 2006), etc. These results, derived for a single inventory holding facility, have been extended to multiechelon systems, (Federgruen, 1993; Shang and Song, 2003; Dong and Lee, 2003; Gallego and Özer, 2005; Chen and Song, 2001)and capacitated systems; (Levi, Roundy, Shmoys, and Truong, 2008; Federgruen and Zipkin, 1986a,b), to better capture the dynamics of modern supply chains. Chen, Drezner, Ryan, and Simchi-Levi (2000a); Chen, Ryan, and Simchi-Levi (2000b) quantify the bullwhip effect for order-up-to policy under exponential smoothing and moving average forecasts. We refer the readers to the books by Zipkin (2000) and Axsäter (2006) for more details.

Distributed decision making in supply chains

Supply chain decisions have traditionally been made by managers at each node. From a decentralized operation perspective, supply chains can be analyzed using the tools of game theory. In decentralized decision making, the payoff (profits) for each node depends not only on its decisions, but also on the decisions made by the other nodes. Therefore, supply chain operations can be viewed as a strategic game between the various nodes. Game theory based analysis can be further classified into noncooperative and cooperative game theory.

In noncooperative game theory, each node simultaneously makes decisions and then the payoff is obtained. Such games are characterized by the Nash equilibrium that is the set of game outcomes for which no node has a unilateral incentive to move away from the outcome. At the Nash Equilibrium, no node can increase its payoff by changing its decision while the choices made by the other nodes remain the same. This result is attributed to Nash in his seminal paper (Nash, 1951). Related to Nash equilibrium is the Stackelberg equilibrium attributed to the mathematician von Stackelberg. In a Stackelberg game the nodes make their decisions sequentially. We refer the reader to the excellent text by Başar and Olsder (1999) for detailed analysis into game theory tools and methods. Leng and Parlar (2005); Cachon and Netessine (2006) provide excellent reviews of game theoretic methods applied to supply chains.

If the nodes make the supply chain optimal decision in a noncooperative game, then the supply chain is said to be coordinated (Cachon and Zipkin, 1999). One of the methods to coordinate supply chains is to modify the interactions between the nodes of the supply chain (for example, by adjusting contracts) so that each node, optimizing its local objective, makes the globally optimal decision. For example, a two node newsvendor type supply chain can be coordinated using buy-back contracts. A two node newsvendor supply chain consists of a retailer and a supplier. The retailer faces a random demand with a known probability distribution at each period. In order to respond to this demand, the retailer buys product from the supplier at the beginning of the period. The supplier is assumed to ship products instantaneously. In the buy-back contract, the supplier agrees to buy back unsold stock at the end of the season from the retailer. The buy-back contract transfers some of the risk of maintaining inventory to the supplier and divides the supply chain optimal profit (the centralized profit) among the partners. In contrast, the performance of the wholesale (price only) contract, in which the supplier supplies product at a wholesale price to the retailer, can be arbitrarily poor. Under wholesale contract, the retailer takes all the risk of excess inventory and orders safely(Cachon, 2003; Cachon and Zipkin, 1999). Perakis and Roels (2007) quantified the inefficiencies in the supply chain (the ratio of the decentralized supply chain profits to that of the centralized supply chain profits) for the price only or wholesale contracts. Moses and Seshadri (2000) showed that a two-echelon supply chain can be coordinated only if the manufacturer agrees to share a fraction of the holding costs of the retailer's safety stock. Golany and Rothblum (2006) also studied linear reward/penalty as a contract modification to induce coordination in the supply chain. Li and Wang (2007) provide a survey of the various coordinating mechanisms. Axsäter (2001) studied the Stackelberg game in the supply chain. Axsäter (2001) assumed that the manufacturer is the leader in the Stackelberg game. The manufacturer minimized the system-wide costs and declared its policies to the retailers. The retailers then optimized a modified cost function that considers the policies of the manufacturer. They implemented an iterative optimization algorithm such that the policies at every iterate was better than the initial policy. The authors also noted that the iterations may not converge to the centralized solution.

On the other hand, cooperative game theory is a branch of game theory that studies the benefits of coalitions. A coalition between nodes is formed when the nodes cooperate. These studies allocate payoffs to various coalitions and these payoffs are analyzed via different techniques like Shapley value (Shapley, 1997) or nucleolus (Schmeidler, 1969). Raghunathan (2003) studied incentives for nodes to form information sharing partnerships. Leng and Parlar (2009) studied different coalitions in a three-echelon supply chain. For example, if the manufacturer and distribution center form a coalition, then it is assumed that the orders placed by the retailer are known to both the nodes. Under the grand coalition, the final customer demand is shared among all the three nodes. Leng and Parlar (2009) defined the payoff of a coalition as the cost savings obtained when extra information due to the coalition is available to the nodes. Using the payoff of all the possible coalitions, they studied the stability of different coalitions. The authors noticed that the bullwhip effect is reduced when the manufacturer and distribution center formed a coalition. Bartholdi and Kemahlioğlu-Ziya (2005) studied a two-echelon supply chain in which a manufacturer supplies to multiple retailers. They used the concepts of cooperative game theory to find profit allocation rules after cooperation. Since the value allocation was in the core of the cooperative game, it ensured that none of the participants in the coalition have incentive to leave. Nagarajan and Sošić (2008) provide a comprehensive survey of cooperative game theory applications to supply chains.

MPC for supply chains

Perea López, Ydstie, and Grossmann (2003) developed a detailed multi-product model including time delays and a mixed integer model for the manufacturing facility. They modeled the shipment rates with a "best I can do" policy that satisfies all the accumulated orders at a given time if stocks are available; otherwise it ships all of its available stock. This model was used for supply chain control using MPC maximize profit. They considered three cases in their implementation: a centralized case, and two other cases that they termed "decentralized" control. In one decentralized control scheme, they optimized the mixed integer production facility while operating the supply chain under a nominal control policy (like a proportional controller for the orders). In the other decentralized control scheme, they optimized only the orders in the supply chain subject to a nominal production schedule. The authors advocated the use of "centralized MPC". Mestan, Türkay, and Arkun (2006) developed a supply chain model using a hybrid systems approach and implemented centralized, decentralized, and noncooperative MPC as described in (Rawlings and Stewart, 2008). They compared customer satisfaction and supply chain profit for the centralized and decentralized MPC. The objective functions were chosen such that the retailer objective of maximizing customer satisfaction was in conflict with the objective of other nodes. Decentralized MPC had the highest customer satisfaction metric but the supply chain operated at a loss. The bullwhip effect was high in the decentralized approach. In centralized MPC, the supply chain found the trade-off between maximizing customer satisfaction and minimizing overall supply chain costs. The centralized approach showed a small bullwhip effect because all the shipment and order rates were determined by a central policy. The authors also noted that the performance of noncooperative MPC was much better than the performance of decentralized MPC. Dunbar and Desa (2007) solved a three-echelon, one-product supply chain using a noncooperative MPC. They developed a bidirectionally coupled model, by considering two types of delay: pipeline delay or the transportation delay and a first order material delay to quantify delays in clearing backlogs. The algorithm was found to be better than a nominal control policy. They also observed that the ordering policy was not very aggressive, indicating that the bullwhip effect may be mitigated by distributed MPC. Seferlis and Giannelos (2004) presented a two-layer MPC strategy for multiechelon supply chains. They used MPC to find shipments and orders placed to other nodes, subject to a total order constraint. The total orders placed was the manipulated variable of a PID controller to track inventory. The authors suggest that the performance can be improved by better tuning the PID controller and suggest a bi-level optimization problem in which the PID controller is replaced with an optimization-based controller. Kempf (2004) and Braun, Rivera, Carlyle, and Kempf (2002) developed a model predictive control framework for the supply chain in the semiconductor industry. They developed models that are specific to the semiconductor industry. Braun et al. (2002) implemented decentralized MPC and studied the control performance under plant model mismatch. Kempf

(2004) described a two-loop optimization technique for the supply chain optimization problem. The coarse first loop optimizer is used to generate the inventory and order setpoints (reference trajectories), while the fine inner loop MPC is used to track these setpoints. Bose and Penky (2000) also used an MPC framework. They focused on forecasting the demand signal and studied the sensitivity of the MPC framework to fluctuations in the demand signal. Maestre, D. Muñoz de la Peña, and Camacho (2009) proposed a cooperative MPC algorithm for a two-layer supply chain. In their formulation, each node minimized its local objective function, not only over its own decision space, but also over the decision spaces of the other nodes. Based on the multiple optimal objective function values (one for each node), the algorithm determined a consensus input. The drawback of the approach is that it is not scalable for large supply chains with multiple nodes. Bemporad, Di Cairano, and Giorgetti (2005) showed the applications of hybrid MPC (Bemporad and Morari, 1999) on a centralized supply chain management problem. Li and Marlin (2009) implemented robust MPC using an economic objective function on a multiechelon supply chain.

In the following section, we show that the supply chain can be modeled as a system of integrators.

4.3 Dynamic modeling of the supply chain[3]

A dynamic model is the heart of any feedback control algorithm. While developing a dynamic model of a supply chain, the components of a supply chain (like the production facility, distributor, retailer etc.) are called as nodes. The supply chain network is the vertices or arcs, which depict the connections between the various nodes. We assume that the network is fixed and given to us. We denote the set of nodes by $\mathscr{I}$. The nodes to which a particular node supplies material are called its downstream nodes, while the nodes from which a particular node obtains material are called its upstream nodes. The set of products handled in the supply chain is given by $\mathscr{P}$. For a particular node $i \in \mathscr{I}$, the set $\mathscr{P}(i)$ denotes the products handled by that

[3]This section has been modified from Section 3 of Subramanian et al. (2012b) to account for multiple products in the supply chain. Equation (4.2) has been modified to track backorders for each downstream node separately.

node. For each node $i \in \mathscr{I}$, and each product $p \in \mathscr{P}(i)$ we define the set $\mathrm{Up}(i,p)$ as the set of all nodes $j \in \mathscr{I}$ that are connected by an arc with i and are upstream to node i. These nodes supply product p to node i. Similarly, we define the set of downstream nodes to i for products p as $\mathrm{Dn}(i,p)$. For each arc in the the supply chain, material flows downstream and orders (or information) flows upstream. The supply chain in the form of nodes and arcs is shown in Figure 4.1.

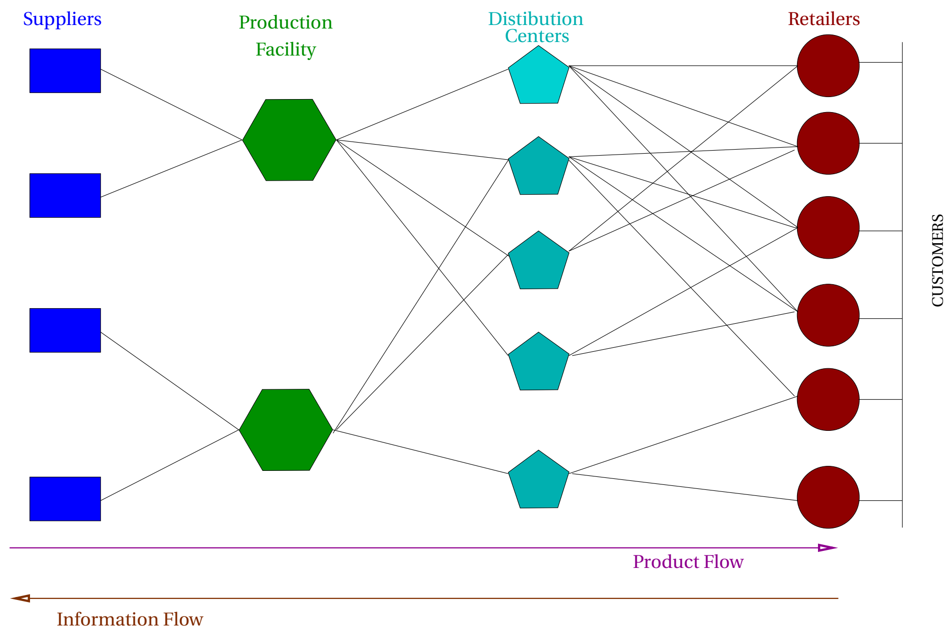

Figure 4.1: Supply chain as nodes and arcs.

From a classical chemical engineering perspective, each node can be modeled as two tanks, the inventory tank and the backorder tank. The flows out of the inventory tank are the shipments to the downstream nodes and the shipments from the upstream nodes make up the flow into the inventory tank. The flows out of the backorder tank are the shipments to the downstream nodes, which alternatively can be viewed as the orders that have been met; the flows into the backorder tank are the orders arriving at the node. For nodes that handle multiple products, we have as many inventory and backorder tanks as the number of product handled by the node. Figure 4.2 depicts the 'tanks' model of a node in the supply chain handling a single product.

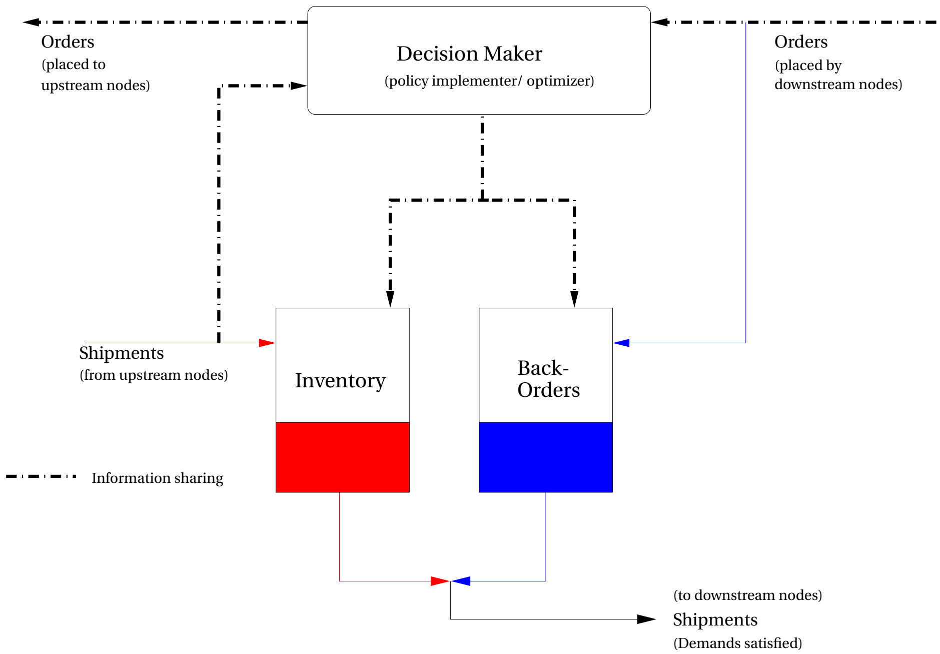

Figure 4.2: Tank analogy for modeling a node.

The states in each node i are: the inventory in the node, $\text{Iv}_{pi} \forall p \in \mathscr{P}(i)$, and the backorders in the node, $\text{BO}_{pii'} \forall p \in \mathscr{P}(i), \forall i' \in \text{Dn}(i,p)$; two inputs: the shipments made to each downstream node $j \in \text{Dn}(i,p)$, S_{pij}, and the orders placed to each upstream node $j \in \text{Up}(i,p)$, O_{pij}. The shipments coming from the upstream nodes S_{pji}, $j \in \text{Up}(i,p)$ and the orders arriving from the downstream nodes O_{pji}, $j \in \text{Dn}(i,p)$ are the disturbances arriving to the node. Denoting discrete sample time by integer k, the dynamic equations for node i can be written as

$$\text{Iv}_{pi}(k+1) = \text{Iv}_{pi}(k) + \sum_{j \in \text{Up}(i,p)} S_{pji}(k - \tau_{pji}) - \sum_{j \in \text{Dn}(i,p)} S_{pij}(k), \qquad \forall p \in \mathscr{P}(i) \tag{4.1}$$

$$\text{BO}_{pij}(k+1) = \text{BO}_{pij}(k) + O_{pji}(k) - S_{pij}(k), \qquad \forall p \in \mathscr{P}(i), \forall j \in \text{Dn}(i,p) \tag{4.2}$$

in which τ_{pji} is the transportation delay. We assume that there are no delays for order transfers between the nodes. Denoting $x_i(k) = \left[\text{Iv}_{pi}(k), p \in \mathscr{P}(i) \quad \text{BO}_{pij}(k), p \in \mathscr{P}(i), j \in \text{Dn}(i,p)\right]'$, $u_i(k) = \left[S_{pij}, p \in \mathscr{P}(i), j \in \text{Dn}(i,p) \quad O_{pij'}, p \in \mathscr{P}(i), j' \in \text{Up}(i,p)\right]'$, and by using the lifting technique described in Chapter 3, the previous dynamic equations for the nodes can be written in the familiar state space form for MPC applications

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k) + \sum_{\substack{l \in \mathscr{I} \\ l \neq i}} B_{il}u_l \tag{4.3}$$

The decision maker shown in Figure 4.2 can take several different forms:

- Each decision maker can implement a simple ordering policy that depends only on the incoming shipments and orders. Such an ordering policy could be a PI controller to control the inventory levels, or (σ, Σ) policies that are obtained from stochastic inventory control optimization. Such decision makers are implementations of classical control theory approaches to supply chain control.

- **Noncooperative MPC:** Each decision maker can implement an MPC controller to regulate its local states by optimizing a local objective function (for example, the profit function for the node). The nodes can share information regarding upstream shipments, downstream orders, etc. This form of control is termed noncooperative MPC.

- **Cooperative MPC:** Each decision maker can implement an MPC controller that considers the effect of the nodes' decision on the entire supply chain (for example, each node optimizes the supply chain profit function). The nodes still share information.

- **Centralized MPC:** We can replace all the decision makers at the nodes with a single decision maker at the supply chain level. This single decision maker makes decisions for all the nodes.

The overall supply chain dynamic model is the individual node dynamic equations collected for all nodes $i \in \mathscr{I}$. The only required change in the node dynamic equation is for the retailer and the production facility nodes.

Retailer models

For the retailer nodes $i \in \mathscr{R}$, the dynamic equations are modified as,

$$\mathrm{Iv}_{pi}(k+1) = \mathrm{Iv}_i(k) + \sum_{j \in \mathrm{Up}(i,p)} S_{pji}(k - \tau_{pji}) - S_{pic}(k), \qquad \forall p \in \mathscr{P}(i) \tag{4.4}$$

$$\mathrm{BO}_{pi}(k+1) = \mathrm{BO}_{pi}(k) + \mathrm{Dm}_{pic}(k) - S_{pic}(k), \qquad \forall p \in \mathscr{P}(i \tag{4.5}$$

in which S_{pic} is the shipment made by the retailer, and Dm_{pic} is the customer orders (demands). *The only disturbances in the overall supply chain model are the customer demands* $d = \left[\mathrm{Dm}_{pic}\right]', p \in \mathscr{P}(i), i \in \mathscr{R}$, which drive all the flows (shipments and orders) in the supply chain.

Production facility models

The production facility needs to be modeled separately because material conversion takes place in this node. In multiple product supply chains, the same production facility handles multiple products. Thus a model for the production facility needs to incorporate a scheduling model to optimize the sequence of production. In this chapter, we assume that the production facilities belong to the first echelon. We further assume an ideal supplier of raw materials to the production facilities, implying that we have infinite supply of raw materials without transportation delay.

Planning models In this chapter, we shall use an “approximate production model” to model the production facility. In the approximate production model, we replace the detailed scheduling model with convex constraints that represent the feasible region of production. This idea is similar to the process attainable region (Sung and Maravelias, 2007)-a convex region of production quantities for which there exists some feasible schedule. The process attainable region can be computed by using computational geometry tools (Sung and Maravelias, 2007; Maravelias and Sung, 2009; Sung and Maravelias, 2009) or parametric programming tools (Li and Ierapetritou, 2010). Let $\mathscr{M}$ be the set of production facility nodes. Then, for each $i \in \mathscr{M}$, the modified dynamic equations for the final products are

$$\mathrm{Iv}_{pi}(k+1) = \mathrm{Iv}_{pi}(k) + S_{pim}(k) - \sum_{j \in \mathrm{Dn}(i,p)} S_{pij}(k), \qquad \forall p \in \mathscr{P}(i)$$

$$\mathrm{BO}_{pij}(k+1) = \mathrm{BO}_{pij}(k) + O_{pji}(k) - S_{pij}(k), \qquad \forall p \in \mathscr{P}(i), \forall j \in \mathrm{Dn}(i,p) f(S_{ipm}(k)) \quad \leq 0$$

in which S_{pim} are the manipulated inputs denoting production of product p during the period. Note that, for multiproduct production facilities, each of the inputs S_{pim} for products $p \in \mathscr{P}$ are coupled by the convex production feasibility constraint. The set $\mathscr{L}$ represents the set of products.

$$f(S_{1im}(k), S_{2ip}(k), \ldots, S_{pim}(k), \ldots) \leq 0$$

Scheduling models The state task network (STN) approach is probably the most popular method to model a production facility in which multiple products are produced using shared resources (Kondili, Pantelides, and Sargent, 1993; Shah et al., 1993). As described in Chapter 3, in STN modeling, the final products, intermediates and raw materials are states that are processed using tasks like reactions, separation, etc. These tasks can be carried out in units capable of handling multiple tasks. A detailed schedule is the sequence of operation of the tasks in the units so that a production objective can be met at minimal cost without violating the scheduling constraints.

Detailed scheduling models are formulated as mixed integer linear programs (MILPs) or mixed integer nonlinear programs (MINLPs). If we chose to model the production facility using a detailed scheduling model, then the resulting supply chain MPC problems become mixed integer programs. Although, research progress has been made in the theory of MIQP and hybrid MPC (see (Bemporad and Morari, 1999)), in this chapter, we do not consider detailed scheduling models in the formulation of the supply chain model. An example using a detailed scheduling model is provided in Section 5.2.

4.3.1 Summary

In this section, we wish to bring to the readers' attention, three salient features of the supply chain dynamic model presented in this section.

Uncontrollable local models Controllability implies that there exist inputs that can move the state of the system from any initial state to any final state in finite time. Examining (4.1) and (4.2) for the inventory and backorder balance for node i, we observe that while nodes $j \in \text{Up}(i,p)$ respond to orders O_{pij} placed by node i, node i has no knowledge of the subsequent dynamics of its own orders. Therefore, we need to provide the node some model of how its orders affect the later shipments coming into the node. To do so in a noncooperative or decentralized control arrangement, we track another state (or output) Ip_{pi} termed the inventory position. The dependence of orders on incoming shipments is modeled through the function $g(\cdot)$.

$$\text{Ip}_{pi}(k+1) = \text{Ip}_{pi}(k) + \sum_{j \in \text{Up}(i,p)} g(O_{pij}(k - \tau_{pji})) - \sum_{j \in \text{Dn}(i)} S_{pij}(k) \qquad \forall p \in \mathscr{P}(i)$$

In the centralized control framework, the actual dynamics of the entire supply chain is available to the decision maker, and the relationship of the orders at node i to its subsequent incoming shipments is captured by the upstream nodes backorder balance equations and the supply chain performance metric. From Section 2.3.3, we know that cooperative MPC algorithms has complete model knowledge. Therefore, uncontrollable local models is not an issue when implementing cooperative MPC for supply chains.

Unstable models The supply chain is modeled as a system of integrators whose response to an input step change is a ramp. Such systems need to be stabilized in the closed loop, otherwise the states can keep growing (think of it as backorders keep rising as time increases). Therefore, we emphasize establishing closed-loop properties of the algorithms that we propose for supply chain optimization.

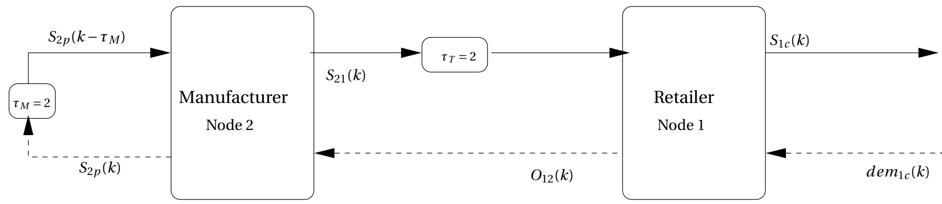

Figure 4.3: Two-stage supply chain.

Stabilizable centralized model We notice that all the nodes belonging to the manufacturing facilities are controllable because we manipulate the production rates. Therefore, the manufacturing nodes do not require an inventory position model. Since the manufacturing facility model has this property, the overall supply chain model is also controllable. The controllability of the centralized model is an important feature that we use to design closed-loop stable centralized and cooperative MPC frameworks for supply chain optimization.

4.4 Example[4]

We simulate the supply chain shown in Figure 4.3 in this section. The plant has production delay of 2 time units and a transportation delay of 2 time units and a single product. For simplicity, we drop the index on product in this section.

Production model As mentioned earlier, the production delay is 2 time units. However, we assume that the manufacturer can start a batch of the product at every sampling time. This assumption means that the manufacturer has two units that can execute the task of producing the final product.

We label the retailer node 1, with the states Iv_1 and BO_1, the inventory and backorder at the retailer. The retailer inputs u_1 consist of orders placed and the shipments made by the retailer, S_{1c} and O_{12}. We label the manufacturer node 2, with states x_2 consisting of inventory Iv_2 and backorder BO_2. The manufacturer inputs are the shipments made to the retailer S_{21} and the production eS_{2m}. The demand $d(k) = \mathrm{Dm}_{1c}$.

[4]The results this section, with the exception of Sections 4.4.2, 4.4.3 and the discussion on steady states, are slightly modified from Section 6 of Subramanian et al. (2012b) to reflect a coding error that was corrected.

Models We write a time invariant model for the supply chain that is also the process model (because we assume that a batch may start at every time) by writing the inventory and backorder balance equation. These model for the retailer is:

$$\underbrace{\begin{bmatrix} \text{Iv}_1 \\ \text{BO}_1 \end{bmatrix}_{k+1}}_{x_1(k+1)} = \underbrace{\begin{bmatrix} 1 & \\ & 1 \end{bmatrix}}_{A_1} \underbrace{\begin{bmatrix} \text{Iv}_1 \\ \text{BO}_1 \end{bmatrix}_{k}}_{x_1(k)} + \underbrace{\begin{bmatrix} -1 & 0 \\ -1 & 0 \end{bmatrix}}_{B_{11}} \underbrace{\begin{bmatrix} S_{1c} \\ O_{12} \end{bmatrix}_{k}}_{u_1(k)} + \underbrace{\begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}}_{B_{22}^{(2)}} \underbrace{\begin{bmatrix} S_{21} \\ S_{2m} \end{bmatrix}_{k-2}}_{u_2(k-2)} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B_d} \underbrace{\begin{bmatrix} \text{Dm}_{1c} \end{bmatrix}_{k}}_{d(k)} \tag{4.6}$$

The manufacturer state space is given by:

$$\underbrace{\begin{bmatrix} \text{Iv}_2 \\ \text{BO}_2 \end{bmatrix}_{k+1}}_{x_2(k+1)} = \underbrace{\begin{bmatrix} 1 & \\ & 1 \end{bmatrix}}_{A_2} \underbrace{\begin{bmatrix} \text{Iv}_2 \\ \text{BO}_2 \end{bmatrix}_{k}}_{x_2(k)} + \underbrace{\begin{bmatrix} -1 & 0 \\ -1 & 0 \end{bmatrix}}_{B_{22}} \underbrace{\begin{bmatrix} S_{21} \\ S_{2m} \end{bmatrix}_{k}}_{u_2(k)} + \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{B_{22}^{(2)}} \underbrace{\begin{bmatrix} S_{21} \\ S_{2m} \end{bmatrix}_{k-2}}_{u_2(k-2)} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}}_{B_{12}} \underbrace{\begin{bmatrix} S_{1c} \\ O_{12} \end{bmatrix}_{k}}_{u_1(k)} \tag{4.7}$$

Notice that the retailer, by just using u_1 cannot move the states from any initial condition to any final condition. We can easily verify this using the Hautus lemma. This matrix $\begin{bmatrix} I - A_1 & B_{11} \end{bmatrix}$ is rank-deficient.

Steady state From equations (4.6) and (4.7), we notice that if $B_{11}u_1(k)+B_{21}^{(2)}u_2(k-2)+B_d d(k) = 0$ and $B_{22}u_2(k) + B_{22}^{(2)}u_2(k-2) + B_{12}u_1(k) = 0$, then any inventory and backorder level can be a steady-state. From the tanks analogy, as long as all the flows in and out of the tank are equal, any level inside the tank is steady. Hence, we have a degree of freedom in choosing the steady state for the inventories and backorders. The steady states for the inputs is determined by the nominal (steady state) demand. Since, we wish to meet all demands, the steady state for backorders is zero. On the other hand, we wish to maintain a safety stock, and so we choose inventory targets to regulate around. In the discussion that follows, we use x to denote deviation from the steady state, i.e., we redefine $x \leftarrow x - x_s$ in which $x_s = (\text{Iv}_{1,t}, 0, \text{Iv}_{2,t}, 0)$, with Iv_t referring to the inventory target.

Stage cost Each node (subsystem) has a local stage cost, given by

$$\ell_1(x_1, u_1) = |x_1|_{Q_1}^2 + |u_1|_{R_1}^2, \quad \ell_2(x_2, u_2) = |x_2|_{Q_2}^2 + |u_2|_{R_2}^2$$

The overall stage cost is $\ell(x,u) = \ell_1(x_1,u_1) + \ell_2(x_2,u_2)$. The costs used are $Q_1 = Q_2 = \text{diag}(1,10)$ and $R_1 = R_2 = diag(1,1)$.

Terminal cost For centralized and cooperative MPC, following the theory outlined in Section 2.3.3.2, we chose the $P > 0, a > 0$ such that there exists a stabilizing control law $\kappa_f(x)$ in the terminal region given by:

$$\mathbb{X}_f = \{x \mid x'Px \le a\}$$

We also choose a $\bar{V} > 0$ and fix $\beta = \max(1, \bar{V}/a)$. The positive definite matrix P is of the form $\begin{bmatrix} P_{11} & P_{12} \\ P'_{12} & P_{22} \end{bmatrix}$. We choose the local terminal cost functions and the centralized terminal cost function as

$$V_f^1(x_1) = |x_1|^2_{P_{11}} \qquad V_f^2(x_2) = |x_2|^2_{P_{22}} \qquad V_f(x) = |x|^2_P$$

We now define the MPC cost functions. The subsystem cost functions are for $i \in \{1,2\}$

$$V_N^{i,\beta}(x_i(0), \mathbf{u}_i) = \sum_{j=0}^{N-1} \ell_i(x_i(j), u_i(j)) + \beta V_f^i(x_i(N))$$

while the overall cost function is:

$$V_N^{\beta}(x, \mathbf{u}) = \sum_{j=0}^{N-1} \ell(x(j), u(j)) + \beta V_f(x(N))$$

Note that since we defined the terminal costs differently for the subsystems, the overall cost function is not the sum of the subsystem cost functions. Associated with each input, we also have the input constraint set $\mathbb{U}_1$ and $\mathbb{U}_2$, which contain the minimum and maximum shipments and orders that can flow through the supply chain. The maximum shipment allowed was capped at 40 units, while any positive order could be placed (arbitrarily large constraint).

MPC implementation

Ordering policies As mentioned in Section 4.3, the local retailer model does not have knowledge of how the orders placed by the retailer affects the supply chain. Therefore, in the implementation of noncooperative and decentralized MPC, we need to incorporate an ordering policy for the retailer. Since the manufacturer reacts to the orders placed by the retailer, the closed-loop performance of the supply chain is intimately connected to the ordering policy. We study two ordering policies in this example:

1. Order-up-to policy: The order-up-to policy can be viewed as a saturated proportional controller.
$$O_{12}(k) = \begin{cases} \mathrm{Iv}_t - \mathrm{Iv}_1(k) & \text{if } \mathrm{Iv}_1 \leq \mathrm{Iv}_t \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$
in which Iv_t is the inventory target.

2. Inventory position control: In inventory position control, the retailer, instead of controlling the inventory, controls the inventory position, which is a controlled output defined as:
$$\mathrm{Ip}(k) = \mathrm{Iv}_1(k) - S_{1c}(k) + O_{12}(k) \tag{4.9}$$
Inventory position control introduces a new controlled output that is a function of the state and input. We penalize the deviations of Ip from the inventory target Iv_t in the optimizations.

Distributed MPC In decentralized and noncooperative MPC with order-up-to policy, we modify the retailer subproblem, subsystem-1 in (2.32), by adding a constraint that enforces the order-up-to policy. Similarly, for decentralized and noncooperative MPC with inventory position control, we modify the retailer objective function in the subsystem-1 problem in (2.32) by modifying the stage cost to penalize inventory position Ip. We present the optimization problem (2.32) again here for convenience (Note that (i) We do not have state constraints in this example following Assumption 15, (ii) the terminal penalty is magnified using the parameter

β.)

$$\mathbb{P}^{i}_{N,nc}(x_i;\mathbf{v}_{-i}):\min_{\mathbf{u}_i}\sum_{j=0}^{N-1}\ell_i(x_i(j),u_i(j))+\beta V_{f,i}(x_i(N))$$

$$\text{s.t. } x_i(j+1)=Ax_i(j)+Bu_i(j)+\sum_{\substack{l\in\{1,2,\ldots,M\}\\ l\neq i}}B_{li}v_l(j) \qquad j=\{0,1,\ldots,N-1\}$$

$$u_i(j)\in\mathbb{U}_i \qquad j=\{0,1,\ldots,N-1\}$$

$$x_i(0)=x_i$$

Decentralized and noncooperative MPC are implemented using Algorithm 1. In decentralized MPC, the subsystems do not share information. That is v_{-i} is assumed by each subsystem to makes its local predictions. Therefore, in the supply chain context, we add another source of inaccuracy with decentralized control (to add to using ordering policies).

The subproblems for cooperative MPC are obtained by fixing the other subsystem inputs in the centralized optimization problem (reproduced here for convenience). Cooperative MPC is implemented using Algorithm 2. In centralized MPC, we solve the overall problem $\mathbb{P}^{\beta}_{N}(x)$ given in (4.4). The parameter β was chosen as 1000.

$$\mathbb{P}_N(x):\min_{\mathbf{u}} V^{\beta}_{N}(x,\mathbf{u})$$

$$\text{s.t. } x(j+1)=Ax(j)+Bu(j), \qquad j=\{1,2,\ldots,N-1\}$$

$$u(j)\in\mathbb{U} \qquad j=\{0,1,2,\ldots,N-1\}$$

$$x(0)=x$$

4.4.1 Nominal demands

We present the results of the different MPC implementations for a nominal demand of $d=8$. In each of the simulations, the retailer starts with inventory $\text{Iv}_1=47$ and the manufacturer starts with inventory $\text{Iv}_2=32$. The control objective is to keep the inventories in the nodes as close to the target inventory ($\text{Iv}_1=45$ and $\text{Iv}_2=30$)as possible while maintaining minimum backorder.

Figure 4.4 compares the results of centralized, cooperative, noncooperative and decentralized MPC in which we used the order-up-to ordering policy. Figure 4.5 compares the results of same controllers, but using the inventory control policy.

We defer discussion about the results to Section 4.5

4.4.2 Stochastic demands

In 4.4.1, we showed the response using a model predictive controller for a supply chain observing nominal demands. In this section, we show the results of implementing the robust MPC algorithm presented in Section 2.4. We consider the two node supply chain shown in Figure 4.3, but with a nominal demand of 10 units per time period. The demand is assumed to be stochastic between 5 and 15 units per time period. In this example, we choose the retailer target inventory $\mathrm{Iv}_{1,t} = 35$.

Following the procedure outlined in Section 2.4, we (i) design a stable cooperative MPC for the nominal system using the methods outlined in the previous section, but with the tightened input constraint sets to account for stochastic demands, and (ii) use the terminal controller $\kappa_f(x) = Kx$ to account for the stochasticity in demand. We used the technique outlined in Rao and Rawlings (1999) to find $\kappa_f(x) = Kx$, such that all outstanding orders (backlogged demands) from the previous time (if any) are satisfied at the current sampling time. We chose the gain K such that $(A + BK)e$ implied that error in the backorder state was zero. Recall that $e(k) = x(k) - z(k)$; the deviation between the actual state and the nominal state. Hence, there is a delay of 1 sampling time before the system reacts to the stochastic demands.

In Figure 4.6, we show the closed-loop response nominal closed-loop response of the inventory at the retailer for cooperative MPC responding to a stochastic demand signal. We also show the cost-function $V_N^{\beta}(z, \tilde{\mathbf{v}})$ and $V_N^{\beta}(x, \tilde{\mathbf{v}})$ to show that although the warm-start was infeasible for the actual state, it was still feasible for the nominal state and hence we could obtain the closed-loop guarantees for robust cooperative MPC. We used $\bar{V} = 20000$. Hence, whenever the

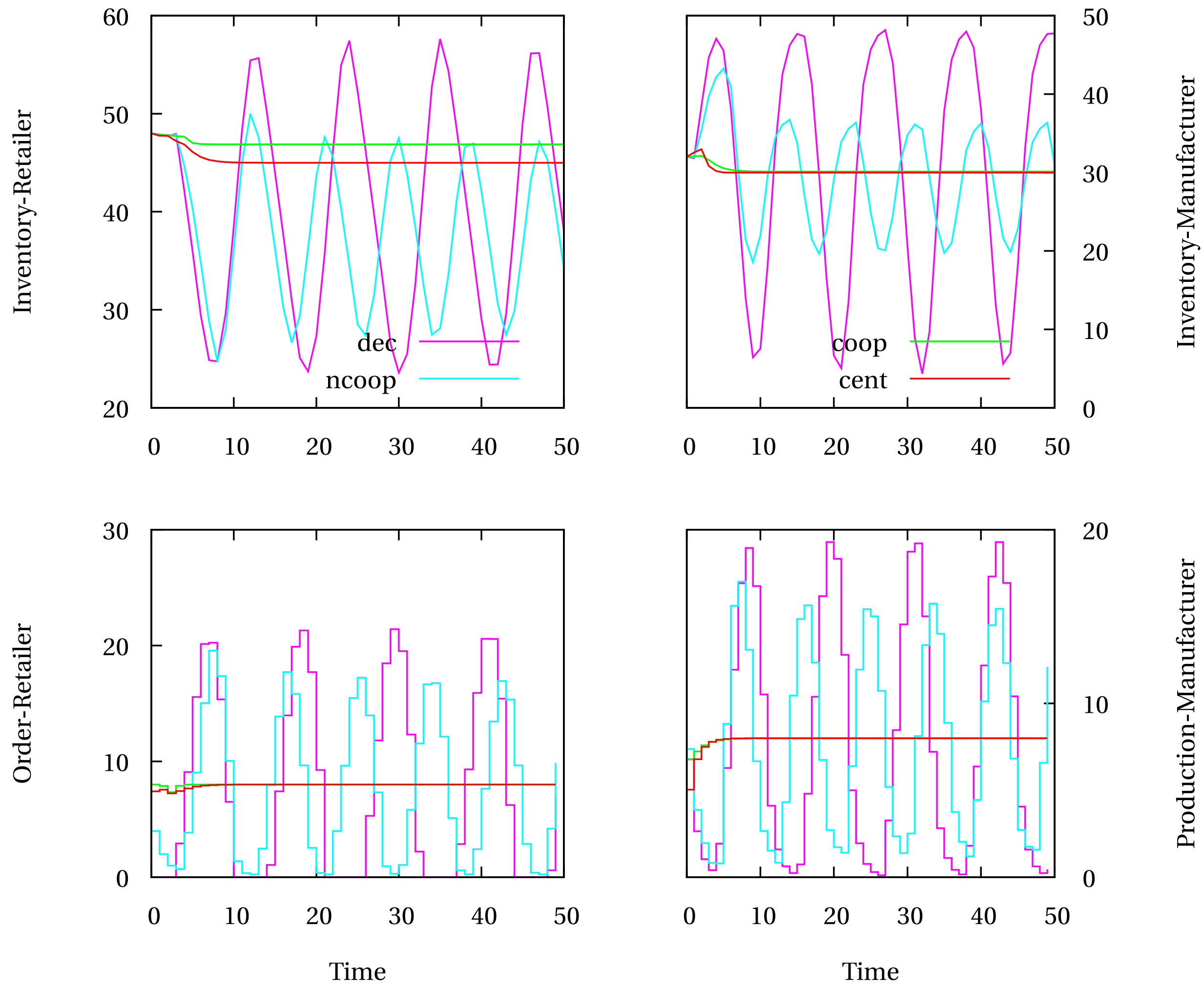

Figure 4.4: Inventories and orders placed in the supply chain: Order-up-to policy (dec: decentralized, ncoop: noncooperative, coop: cooperative, cent: centralized).

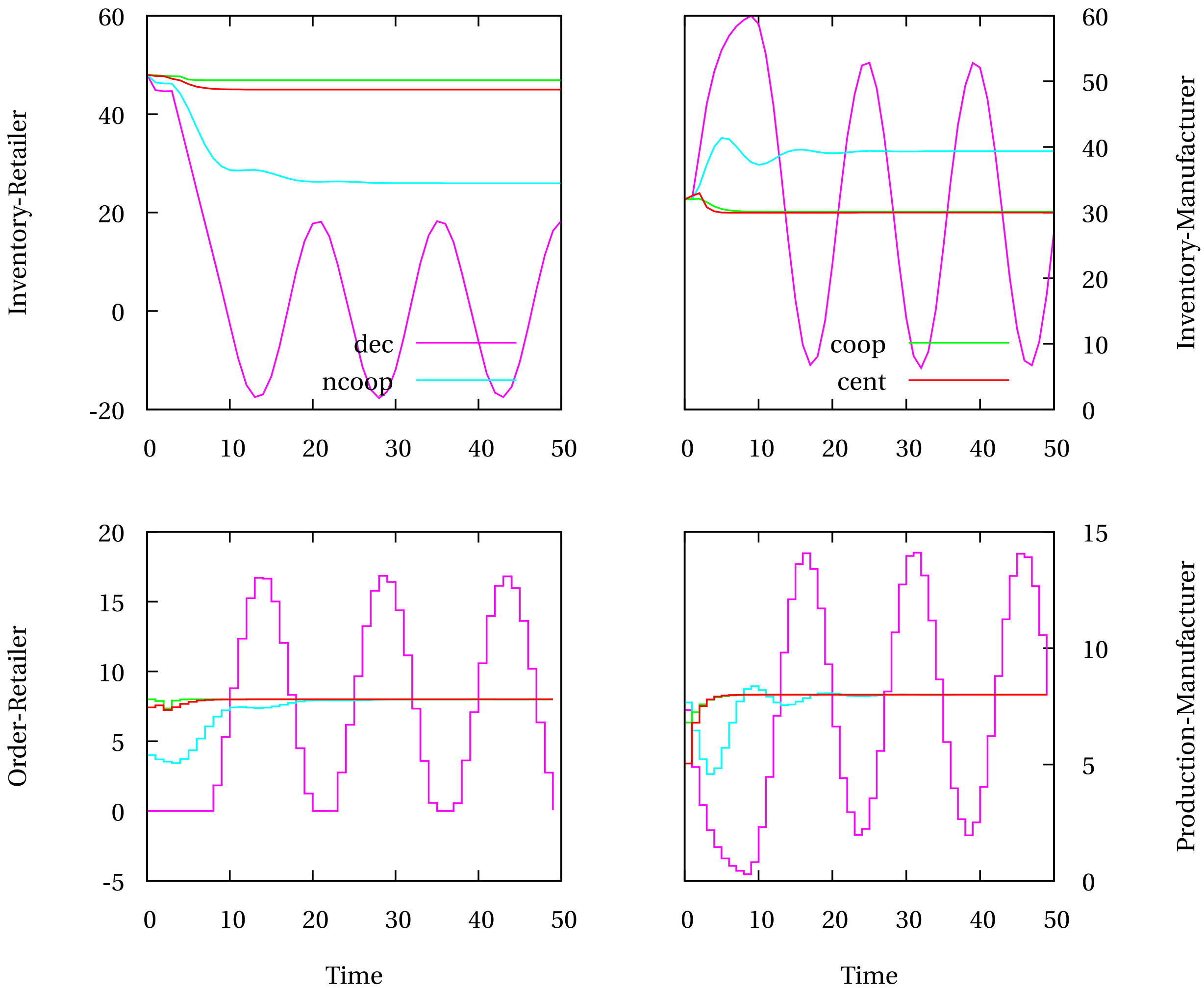

Figure 4.5: Inventories and orders placed in the supply chain: Inventory position control (dec: decentralized, ncoop: noncooperative, coop: cooperative, cent: centralized).

cost function for the actual state was greater than 20000, it meant that the warm-start was infeasible. That is, the terminal state was not inside the set $\mathbb{X}_f$. However, by design of cooperative algorithm, the warm start always remains feasible for the nominal MPC problem.

In Figure 4.7, we show the closed-loop response using a modified version of Algorithm 4 discussed in Section 2.4.

4.4.3 Multi-echelon supply chain example

A critical step in the cooperative MPC algorithm (Algorithm -2) using the Jacobi algorithm is the convex combination of the optimal input with the values at the previous iterate that is taken in the inner loop of the algorithm. The parameter ω_i limits the size of the step taken in the descent direction. Since, it is required that $\sum_{i=1}^{M} \omega_i = 1$, the step sizes generated by the Jacobi algorithm can become quite small as the number of subsystems increase (and convergence is slow). As alluded in Section 2.3.3, closely related to the Jacobi algorithm is the Gauss-Seidel parallel optimization algorithm; in which the subsystems move sequentially. The advantage in Gauss-Seidel algorithm is that the subsystems can take full steps. The Gauss Seidel algorithm for subsystem i can be written succinctly as:

$$\begin{aligned}
&\min_{\mathbf{u}_i \in \mathbb{U}_i} V_N^{\beta}(x, \mathbf{u}) \\
&\text{s.t. } \mathbf{u}_l = \mathbf{u}_l^{(p+1)}, \qquad l \in \{1, 2, \ldots, i-1\} \\
&\mathbf{u}_l = \mathbf{u}_l^{(p)}, \qquad l \in \{i+1, i+2, \ldots, M\}
\end{aligned}$$

Upon obtaining the solution $\mathbf{u}_i^0$ to the problem above, subsystem i sets its next iterate as $\mathbf{u}_i^{(p+1)} = \mathbf{u}_i^0$. As discussed in Bertsekas and Tsitsiklis (1989, Section 3.3.5), both these methods (Jacobi and Gauss-Seidel) or any combination of these algorithms (blocks of subsystem move sequentially; within every block, the subsystems move in parallel) satisfy all the properties in Proposition 12 (for convex problems) and Proposition 13 (with uncoupled constraints). Hence, depending on the application, we could choose to use Gauss-Seidel or a combination of Jacobi and Gauss-Seidel algorithm in Cooperative MPC without losing any of the guarantees of Cooperative MPC.

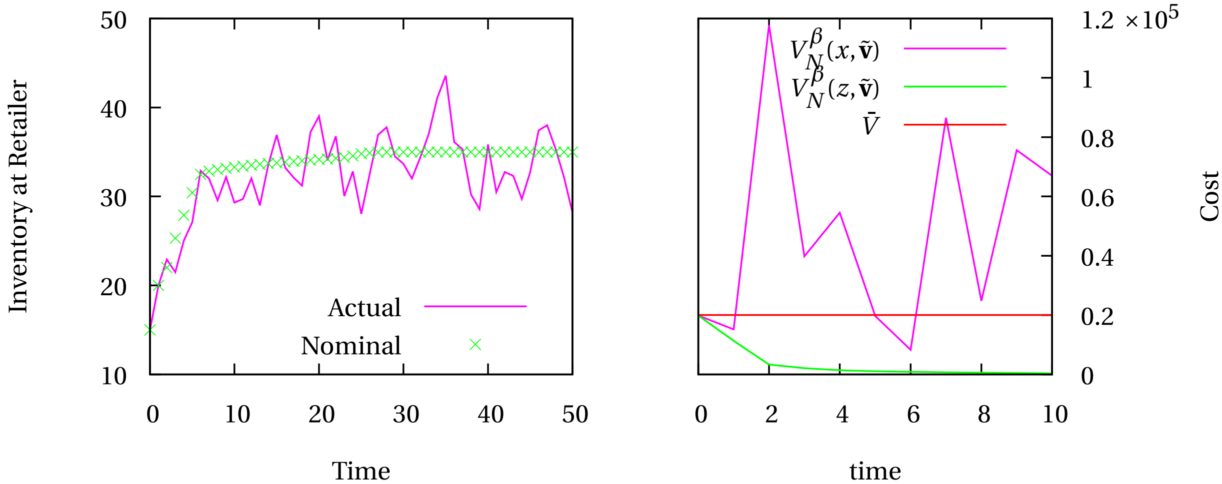

Figure 4.6: (Left) Closed-loop response (Right) Warm start rendered infeasible for actual state because of disturbance

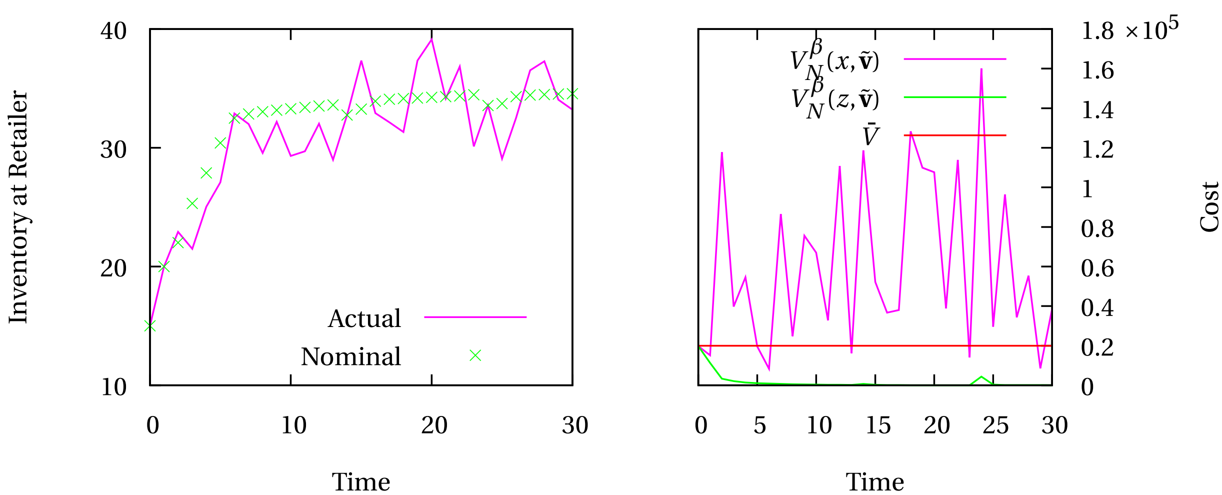

Figure 4.7: (Left) Closed-loop response. Notice that we reset the state t = 14, t = 24 when the cost is such that the warm start is feasible for the actual state (Right) Warm start rendered infeasible for actual state because of disturbance

In this section, we present a combination Gauss-Seidel and Jacobi algorithm (GSJ) that closely resembles the decision making hierarchy in supply chains. Traditionally, in supply chains, the retailers respond first to the customer demands. Upon receiving the orders from the retailers, the distributors make their decisions. Therefore, the current decision making paradigm in supply chains is a sequential one. Hence, we propose to use a mixed Gauss-Seidel and Jacobi optimization routine in cooperative MPC. The proposed optimization proceeds as follows (for a three echelon supply chain):

1. The retailers, make their decisions in parallel, by fixing the upstream nodes decisions. Since only a subset of the nodes are making their decisions in parallel, the convex combination weight $\omega_i, i \in \mathscr{R}$ scale as $|\mathscr{R}|$ (the number of retailers),

2. The distributors obtain the retailer decisions and make their choices in parallel and,

3. The manufacturers make their decisions after obtaining the decisions of both the manufacturers and retailers.

The proposed algorithm has faster convergence when compared to the Jacobi algorithm (see Figure4.9). Moreover, since the upstream nodes decisions depend on the orders placed by the downstream node, in the Jacobi iterations, the upstream nodes have a disadvantage because their optimizations are based on the previous iterate or warm-start values of the downstream orders. On the other hand, in the proposed method, the upstream nodes, optimize to react to the current iterate of the downstream demands.

In Figure 4.8, we show a 3 echelon supply chain with 7 nodes. In this example, we consider only one product. The transportation delay between nodes is 1 time unit, while the production delay is 2 time units. As with the previous example, we assume that the manufacturing unit can start a batch at every sampling time. The stage cost for each node was chosen as $\ell_i(x,u) = x'Qx + u'Ru$ with $Q = \text{diag}(1,10)$ and $R = \text{diag}(1,1)$. The objective was to regulate to the inventory targets (all backorder targets were zero). The nominal demand was 10 units every period at each retailer node. In Table 4.1, we list the starting inventory levels and the target inventory levels in each node. The maximum shipment from one node to another was 20 units. The prediction horizon was $N = 15$. We chose $\bar{V} = 5000$ and $a = 50$. We just implement one iteration of the cooperative optimization algorithm, i.e., $\bar{p} = 1$.

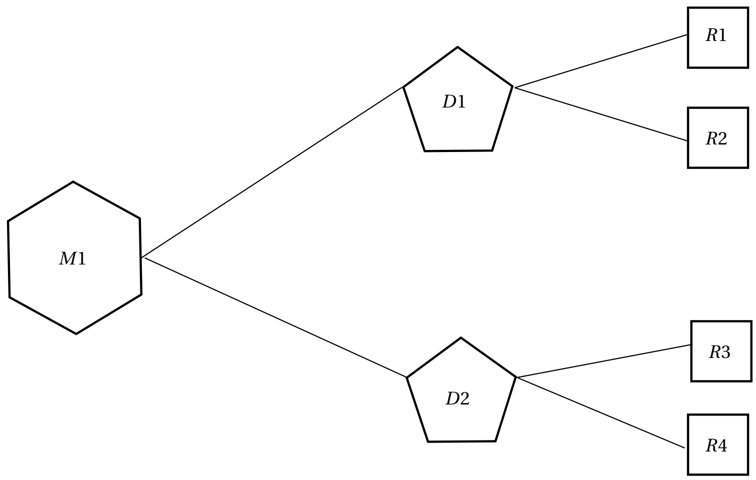

Figure 4.8: Multi-echelon supply chain studied

In Figure 4.9, we show the convergence of the three types of parallel optimization algorithms.

For this supply chain network, the Gauss-Seidel iterations converges the fastest. In Figure 4.11, we show the inventory profile for nodes $R1$ and $R2$ in the supply chain under centralized control, the proposed Gauss-Seidel-Jacobi algorithm and the Jacobi algorithm. Not much difference in the closed-loop performance can be observed. One reason for this observation could be because the cooperative MPC algorithm was initialized with the centralized optimal solution at time 0[5]. Hence, the warm start at time $t = 1$ was close to the optimal solution at time 1.

In Figure 4.12, we show the closed-loop solutions when the supply chain was initialized with a suboptimal solution at $t = 0$. In Figure 4.10, we show the open-loop cost of the supply chain at each sampling time, when the simulation was initialized with a suboptimal solution at $t = 0$.

[5]There is no guarantee that will be such small differences in if we initialize the cooperative MPC algorithm with the optimal solution at $t = 0$

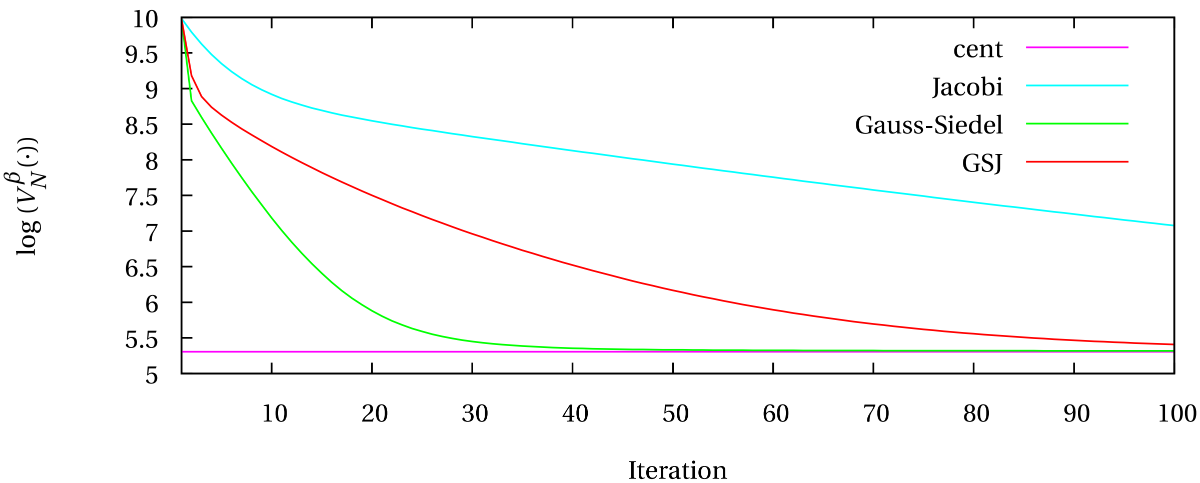

Figure 4.9: Convergence of various parallel optimization algorithms for the supply chain example

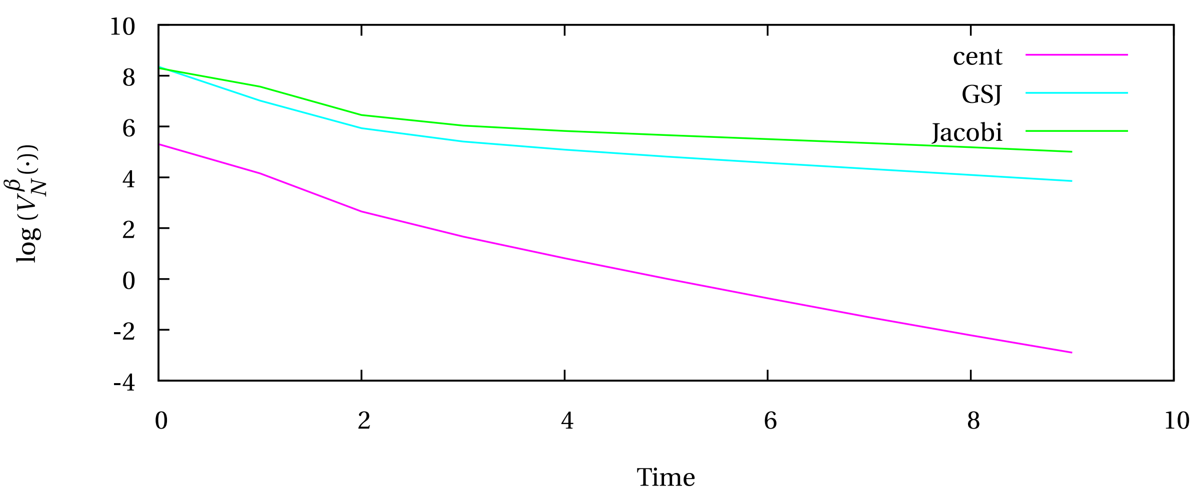

Figure 4.10: Open-loop prediction cost for cooperative MPC optimizations with 1 iteration

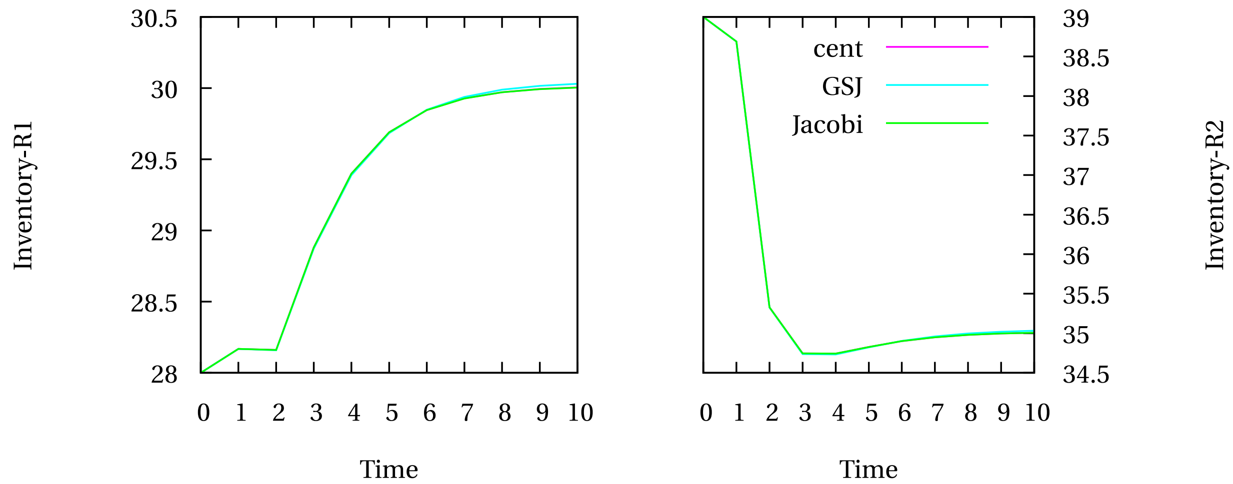

Figure 4.11: Inventories in Retailer nodes 1 and 2 when cooperative MPC is initialized with centralized optimal input at $t = 0$.

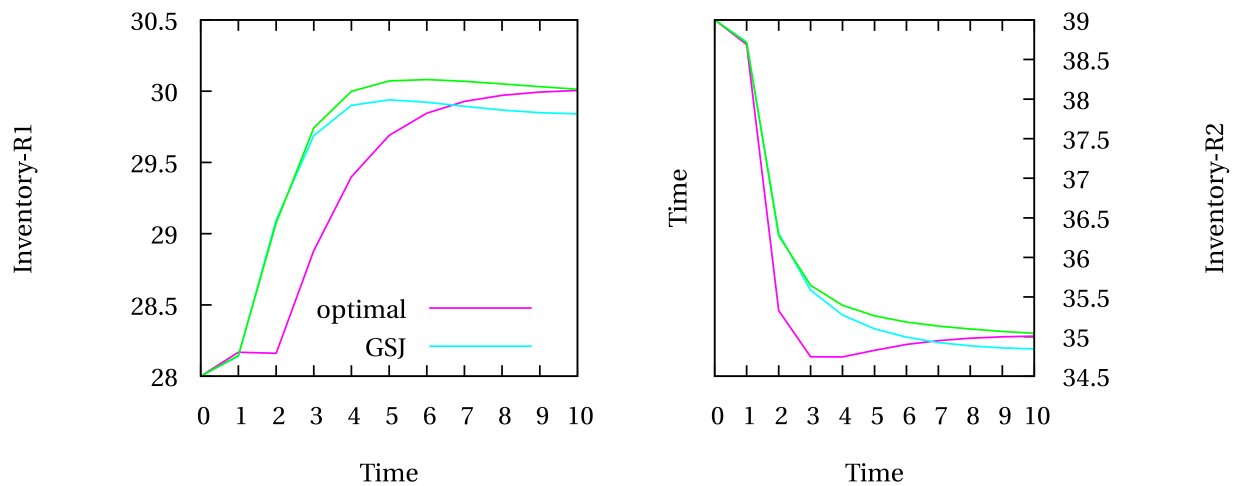

Figure 4.12: Inventories in Retailer nodes 1 and 2 when cooperative MPC is initialized with suboptimal input at $t = 0$

Table 4.1: Starting inventory and Inventory targets

	M1	D1	D2	R1	R2	R3	R4
Starting Inventory	40	37	38	28	39	29	36
Target Inventory	35	45	45	30	35	25	30

4.5 Discussion[6]

Value of information: We observe that, for both order-up-to and inventory position control, decentralized MPC produces large variations in the inventory and orders. These variations indicate a large bullwhip effect, and happen because the nodes have incomplete current information and no knowledge of the dynamics of the other nodes. At each time step, the retailer assumes some flow of materials from the manufacturer to make inventory predictions. Based on these predictions, the retailer places orders with the manufacturer. Similarly, the manufacturer knows only the current order quantity and makes some assumptions about the future orders from the retailer and makes production decisions. When the actual orders and shipments arrive at the nodes, their decisions are suboptimal.

Noncooperative MPC with the inventory position control policy able to reach a steady state, since, each node now has more information about the ordering and production plans of the other node, both are able to make better forecasts and therefore, better decisions. On the other hand, noncooperative MPC with the order-up-to policy shows sustained inventory and order oscillations. This is because each node is implementing its best-response (we used $\bar{p} = 1$) to the other nodes' decisions. In contrast,in decentralized MPC, since each node is assuming that the other node is implementing a nominal policy (all flows equal to nominal demand). Therefore, it is able to reach a steady-state as the assumed policy is actually the steady-state policy.

[6]This section, with the exception of the discussion about VMI appears in Section 6 of Subramanian et al. (2012b)

Impact of ordering policy: In noncooperative MPC with inventory position control, we observe that there are no inventory variations and the system reaches a steady state. All flows through the system settle at the nominal demand, which is the input steady state. The inventories, however, show offset from the target. In order-up-to policy, irrespective of the cost of placing large orders, the retailer is constrained to make orders if the inventory at any period falls below the target. In inventory position control, the orders placed are penalized, and therefore the retailer tends to order less, because the optimizer tries to balance ordering costs and inventory deviation costs.

Plant-model mismatch: If we compare results for cooperative and centralized MPC with noncooperative MPC, we see that, cooperative and centralized MPC reach steady state more quickly. They achieve steady state because there is no information distortion in the system. Each node in cooperative control, optimizes not only the system-wide objective, it also accounts for the dynamics of the entire supply chain. In noncooperative MPC with inventory position control, since the retailer does not know the actual supply chain dynamics, it settles at a steady state that depends on the inventory position model. Therefore, we see the value of optimizing the actual dynamics instead of introducing a mismatch between the models used by the controller and the actual dynamics by using inventory position models.

Guaranteed stability: The third important result of the analysis is that cooperative and centralized MPC have been designed to guarantee closed-loop stability. Although, we see that noncooperative MPC using inventory position control has not made the supply chain unstable, we have no stability guarantees. On the other hand, using the theory developed in Section 2.3.3, we can guarantee closed-loop stability for cooperative MPC.

Relation to echelon stock policies and VMI: Echelon stock policies are decentralized operating policies, but based on the concept of echelon stock. Echelon stock is the stock carried by the node and all its downstream nodes. Therefore, echelon stock based policies, to a certain extent, are like the noncooperative MPC because the policies for a node depend on information sharing regarding the inventories in all its downstream node.

Vendor managed inventory (VMI) is emerging as a popular tool for supply chain integration. In VMI, the buyer (retailer) authorizes the supplier (manufacturer) to maintain his inventory. VMI, therefore resembles cooperative control because, not only information regarding inventories is visible to the supplier, but also the dynamics. The supplier, in a two-stage supply chain, manages shipments between his facility and the retailer and production by observing the retailer inventory. One of the main disadvantages of VMI that has been reported is that the retailer loses control over inventory management and some knowledge gained by the retailer (like advanced forecasting) that could lead to better inventory management cannot be used. Another disadvantage is that the overall supply chain objective function is not used by the supplier (Sari, 2007). In this aspect, cooperative control can be seen as a middle ground between VMI and decentralized control. In cooperative control, each node still manages its own inventory, while optimizing the overall supply chain objective function.

Chapter 5

Economic MPC for supply chains

In this chapter, we propose an economic model predictive control (MPC) algorithm for inventory management in supply chains with guaranteed closed-loop properties. We compare the properties of the proposed controller with classical control policies like the (σ, Σ) policy. In the previous chapter, we showed centralized and cooperative MPC designed to track the states of the supply chain (inventories and backorders) to a steady state (target stocks or safety stocks). The on-line optimization problem solved in Chapter 4 did not have any knowledge about the economics of the process (e.g.,cost of shipping, production etc.). Our focus in this chapter is to leverage recent developments in Economic-MPC (Amrit et al., 2011; Diehl et al., 2011) to develop a stable controller for inventory management in which the controller directly optimizes the supply chain economics.

In Section 5.1, we review stability theory for economic MPC and propose two flavors of MPC for supply chain, (i) a pure economic objective function and (ii) a mixed objective comprising of an economic objective and a tracking objective. We implement these economic MPC policies on the two stage supply chain example introduced in Section 4.4 . In Section 5.2, we implement the controller proposed in this chapter on a multi-product, multi-echelon supply chain and compare the performance of the controller with that of the (σ, Σ) policy. In Section 5.2.2, we introduce a scheduling model for the manufacturing facility with the aim of integrating scheduling and inventory control using MPC. We present results on recursive feasibility using ideas from Chapter 3.

5.1 Economic MPC theory

As mentioned in Chapter 2, the goal of controller design using MPC is to stabilize the closed-loop. We show a supply chain example to reinforce the idea that simply solving an optimization problem at each sampling time in a rolling horizon manner can destabilize the plant.

Consider the two-stage supply chain example presented in Section 4.4. From equations (4.6) and (4.7), we can write the centralized supply chain model for the two node system as:

$$\underbrace{\begin{bmatrix} \mathrm{Iv}_1 \\ \mathrm{BO}_1 \\ \mathrm{Iv}_2 \\ \mathrm{BO}_2 \end{bmatrix}_{k+1}}_{x(k+1)} = \underbrace{\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \mathrm{Iv}_1 \\ \mathrm{BO}_1 \\ \mathrm{Iv}_2 \\ \mathrm{BO}_2 \end{bmatrix}_{k}}_{x(k)} + \underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} S_{1c} \\ O_{12} \\ S_{21} \\ S_{2m} \end{bmatrix}_{k}}_{u(k)} + \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{B^{(2)}} \underbrace{\begin{bmatrix} S_{1c} \\ O_{12} \\ S_{21} \\ S_{2m} \end{bmatrix}_{k-2}}_{u(k-2)} + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{B_d} \underbrace{\begin{bmatrix} \mathrm{Dm}_{1c} \end{bmatrix}_{k}}_{d(k)} \tag{5.1}$$

Using lifting and a slight abuse of notation, the supply chain model can be written as:

$$x(k+1) = Ax(k) + Bu(k) + B_d d(k) \tag{5.2}$$

We define the economic stage cost as

$$\ell_E(x,u) = q'x + r'u \tag{5.3}$$

in which q is a vector comprising of inventory holding q_{Iv_i} and lost sales q_{BO_i} costs, while r is a vector comprising of shipment costs $r_{S_{1c}}, r_{S_{21}}$, production costs $r_{S_{2m}}$ and ordering costs $r_{O_{12}}$.

The stage cost defined in Chapter 2, is the tracking stage cost. In Chapter 2, we defined the tracking cost to track the state to the origin. In Chapter 4, we used deviation variables $x \leftarrow x - x_s$

(x_s being the steady-state). In this chapter, we generalize the cost to track to a given target, and use absolute variables instead of tracking variables. We (re)define the tracking stage cost as $\ell_T(x,u)$ given by:

$$\ell_T(x,u,z_p) = (x-x_t)'Q(x-x_t) + (u-u_t)'R(u-u_t) \tag{5.4}$$

in which $z_p = (x_t', u_t')$ is a vector comprising of the state and input targets (x_t, u_t) respectively. The positive definite matrices Q, R penalize the deviation of the states and inputs from their targets.

In order to ensure that the cooperative MPC problems in Chapter 4 converged to the centralized optimal solution, we used Assumption 15 to "relax" the state constraints. Since, we only formulate centralized control problems in this chapter, we re-introduce state constraints. The state constraint is defined as

$$\mathbb{X} := \left\{x \mid \underline{x} \leq x \leq \bar{x}\right\} \tag{5.5}$$

Similarly, the input constraint set is:

$$\mathbb{U} := \left\{u \mid \underline{u} \leq u \leq \bar{u}\right\} \tag{5.6}$$

The inequalities in (5.5), (5.6) are componentwise. These constraints define, for example, the positivity constraints for backorders and inventories, capacities of the nodes, transportation capacities etc.

Given a planning horizon N and a demand forecast $\mathbf{d} = (d(0), d(1), \ldots, d(N-1))$, we formulate the following optimization problem:

$$\begin{aligned}
\mathbb{P}_N(x) : &\min_{\mathbf{u}} V_N(\mathbf{u}; x, \mathbf{d}, N) \\
&\text{s.t. } x(j+1) = Ax(j) + Bu(j) + B_d d(j), j = \{0,1,\ldots,N-1\} \\
&\underline{x} \leq x(j) \leq \bar{x}, \qquad j = \{0,1,\ldots,N\} \\
&\underline{u} \leq u(j) \leq \bar{u}, \qquad j = \{0,1,\ldots,N-1\} \\
&x(0) = x
\end{aligned} \tag{5.7}$$

in which $\mathbf{u} = (u(0), u(1), \dots, u(N-1))$. The cost function $V_N(\mathbf{u}; x, \mathbf{d}, N)$ is the sum of N stage costs

$$V_N(\mathbf{u}; x, \mathbf{d}, N) = \sum_{i=0}^{N-1} \ell_E(x(i), u(i))$$

Notice that in Problem (5.7), we do not have any terminal penalty or constraints. In Figure 5.1, we show the backorder BO_1 evolution for the retailer, for a controller that solved problem (5.7) with $N = 15$ and nominal demand $d(k) = d_s = 10$. The stage cost was given by

$$q = (1, 1, 1, 1)' \qquad r = (10, 1, 10, 1)'$$

We assume that the shipping costs are greater than the backorder costs. Clearly, we observe that despite implementing the optimal input at each time instance, the backorder is increasing with time, indicating that customer demand is not being met. Although, we have chosen a pathological cost vector in which production costs are greater than lost sales cost, we observe that there exists an unique steady-state for this system $x_s = \underline{x} = 0, u_s = d_s$, that is the inventory and backorders are zero and all the flows in the supply chain (shipping and ordering between nodes, production) are equal to the nominal demand. The lower bounds on inventories and backorders are zero. Notice that the choice of $u_s = d, x_s = 0$ in (5.2) is a solution to:

$$(I - A)x - B^{(2)}u - Bu = B_d d$$

Note that the steady-state for the states in the supply chain is independent of the demands and delays in the system.

For the costs mentioned in above, staying at the steady-state incurs a cost of 220 per period which is much lesser than the cost incurred per period by the rolling horizon controller, which is unbounded.

For a simple two-node supply chain, we have demonstrated that simply reoptimizing an economic objective function at each time instance could lead to undesirable closed loop performance. Although, we chose a pathological cost function to demonstrate the undesirable closed loop, for a more complex supply chain, it is difficult to apriori analyze all the interactions and judge if the rolling horizon optimization will yield a desirable closed-loop. As mentioned

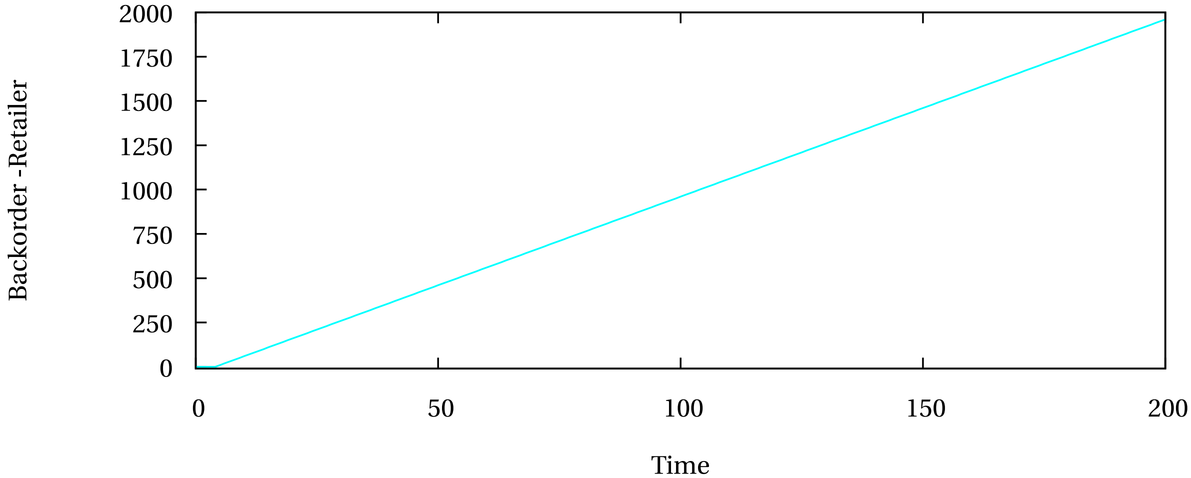

Figure 5.1: Backorder in the retailer for rolling horizon optimization without stability constraints.

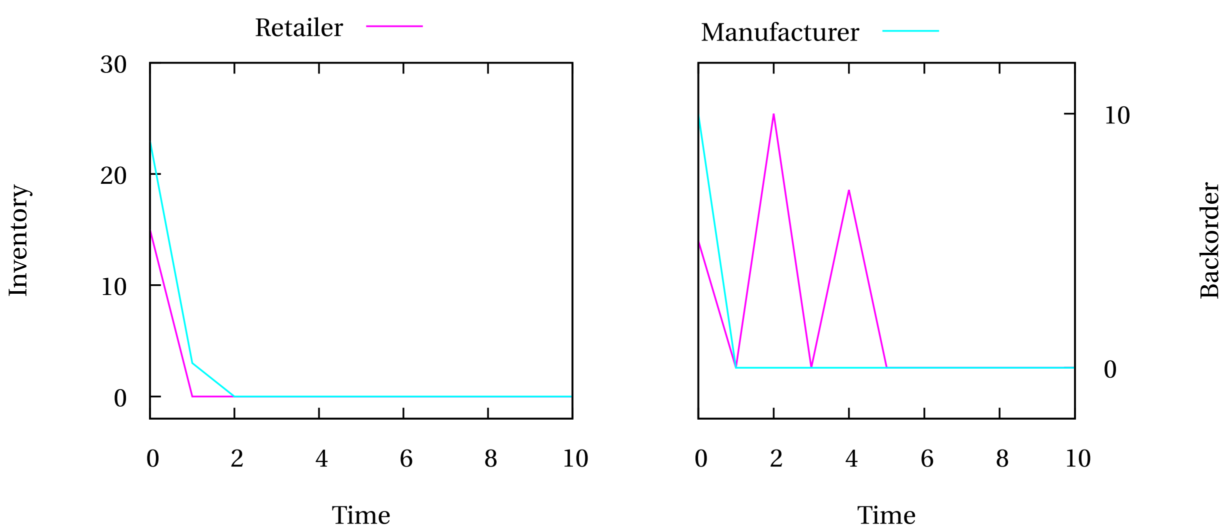

Figure 5.2: Closed loop evolution using stabilizing MPC.

in Chapter 2, stability theory for MPC gives us design guidelines on the formulation of the on-line optimization problem, so that closed-loop stability guarantees can be provided. While, we covered tracking MPC stability theory in Chapter 2, we briefly review stability theory when the stage costs are economic (or other general cost) functions. Stability theory for economic MPC is a relatively new field (Diehl et al., 2011) (Amrit et al., 2011). We state the main results in this chapter and refer the reader to the papers for more details. In Chapter 2, we ensured closed-loop properties by including (i) a terminal cost $V_f(x)$ and (ii) a terminal region $x(N) \in \mathbb{X}_f$. In the following sections, we outline the theory for economic MPC with terminal equality constraint (Diehl et al., 2011), i.e. $x(N) = x_s$ and economic MPC with terminal penalty and region (Amrit et al., 2011).

5.1.1 Terminal equality constraint formulation

We consider the system given by (5.2). We consider the linear economic cost given by (5.3). The states are constrained to lie in the hyperbox $\underline{x} \leq x \leq \bar{x}$ while the inputs lie in $\underline{u} \leq u \leq \bar{u}$ (The inequalities are componentwise). We assume that the sets

$$\mathbb{X} : \left\{ x \mid \underline{x} \leq x \leq \bar{x} \right\} \qquad \mathbb{U} : \left\{ u \mid \underline{u} \leq u \leq \bar{u} \right\}$$

are convex, bounded, closed, and contain the optimal steady-state defined later in (5.8). Although many of the assumptions made in this chapter are similar to the assumptions made for centralized MPC in Chapter 2, we reproduce them here again for clarity.

Note that the choice of linear models and cost function automatically satisfies Assumption 23 stated below.

Assumption 23 (Continuity)**.** *The system and the stage costs are continuous.*

We define the steady-state optimization problem as follows:

$$\min_{x,u} \ell_E(x,u;d_s)$$
$$\text{s.t. } x = Ax + Bu + B_d d_s, \tag{5.8}$$
$$x \in \mathbb{X}, u \in \mathbb{U}$$

We denote $(x_s, u_s; d_s)$ as the solution to (5.8) and make the following assumptions on $(x_s, u_s; d_s)$. The nominal demand is denoted as d_s.

Assumption 24 (Strict dissipativity)**.** *There exists $(x_s, u_s; d_s)$ and λ_s so that*

(a) $(x_s, u_s; d_s)$ *is an unique solution of* (5.8).

(b) The multiplier λ_s is such that $(x_s, u_s; d_s)$ uniquely solves (5.9)

$$\min_{x,u} \ell_E(x,u) + \lambda_s'[x - (Ax + Bu + B_d d_s)] \quad \textit{s.t. } x \in \mathbb{X}, u \in \mathbb{U} \tag{5.9}$$

(c) The system $x^+ = Ax + Bu + B_d d_s$ is strictly dissipative with respect to the supply rate $s(x,u) = \ell_E(x,u) - \ell_E(x_s,u_s)$ and storage function $\lambda(x) = \lambda_s' x$. That is, there exists a $\mathcal{K}_\infty$ function $\rho(\cdot)$ such that:

$$\lambda_s'(Ax + Bu + B_d d_s - x) \le -\rho(x - x_s) + s(x,u), \forall (x,u) \in \mathbb{X} \times \mathbb{U} \tag{5.10}$$

Because of the structure of the state space matrix A in (5.1)-(5.2), the steady-state problem (5.8) decomposes into two problems:

$$\min_x q'x \qquad \text{s.t. } \underline{x} \le x \le \bar{x} \tag{5.11}$$

and

$$\min_u r'u \qquad \text{s.t. } B^{(1)}u + B^{(2)}u + Bu + B_d d_s = 0, \underline{u} \le u \le \bar{u} \tag{5.12}$$

By choosing λ_s as the optimal Lagrange multiplier for the equality constraints in (5.8), we can establish that the $(x_s, u_s; d_s)$ is the unique solution of optimization problem (5.9).

Hence Assumption 24 is satisfied by the supply chain model.

We now define the terminal equality constraint MPC optimization problem:

$$
\begin{aligned}
\mathbb{P}_N(x) : \min_{\mathbf{u}} \; & V_N(\mathbf{u}; x, \mathbf{d}, N) \\
\text{s.t. } & x(j+1) = Ax(j) + Bu(j) + B_d d(j), j = \{0, 1, \ldots, N-1\} \\
& \underline{x} \leq x(j) \leq \bar{x}, \qquad j = \{0, 1, \ldots, N\} \qquad (5.13) \\
& \underline{u} \leq u(j) \leq \bar{u}, \qquad j = \{0, 1, \ldots, N-1\} \\
& x(0) = x \\
& x(N) = x_s \qquad (5.14)
\end{aligned}
$$

in which $\mathbf{u} = (u(0), u(1), \ldots, u(N-1))$. The cost function $V_N(\mathbf{u}; x, \mathbf{d}, N)$ is the sum of N stage costs

$$V_N(\mathbf{u}; x, \mathbf{d}, N) = \sum_{i=0}^{N-1} \ell_E(x(i), u(i))$$

Note that in contrast to problem (5.7), we have added a terminal constraint that $x(N) = x_s$ in problem (5.13). The steady state x_s is the solution to the optimization problem (5.8).

Before presenting the stability theorem, attributed to Diehl et al. (2011), we define the following sets and the control law.

The set of admissible state-input pairs $(x, \mathbf{u})$ is denoted by $\mathbb{Z}_T$ as follows (see definition (2.5)):

$$\mathbb{Z}_T := \left\{(x, \mathbf{u}) \mid \phi(i; x, \mathbf{u}, \mathbf{d}_s) \in \mathbb{X} \text{ , } \mathbf{u} \in \mathbb{U}^N, \phi(N; x, \mathbf{u}, \mathbf{d}_s) = x_s\right\} \qquad (5.15)$$

in which $\mathbf{d}_s$ is the nominal demand vector and $\phi(i; x, \mathbf{u}, \mathbf{d}_s)$ denotes the solution at time i under input $\mathbf{u}$ starting from x at time 0.

The projection of set $\mathbb{Z}_T$ onto the feasible state space $\mathbb{X}$ is called the set of admissible initial states, $\mathcal{X}_{N,T}$ (see definition (2.6))

$$\mathcal{X}_{N,T} := \left\{x \mid \exists \mathbf{u} \in \mathbb{U}^N \text{s.t. } (x, \mathbf{u}) \in \mathbb{Z}_T\right\} \qquad (5.16)$$

We denote the optimal solution of problem (5.13) as $\mathbf{u}^0(x(k), \mathbf{d}_s)$. Denoting the first input in the sequence $\mathbf{u}^0(x(k), \mathbf{d}_s)$ as $\kappa_T(x(k))$, we obtain the closed-loop dynamics of the MPC algorithm as $x^+ = Ax + B\kappa_T(x) + B_d d_s$. The control law is $\kappa_T(x(k))$. Note that, similar to nominal MPC, we prove the closed-loop properties for economic MPC only for the nominal demand d_s.

Theorem 25 (Lyapunov function with terminal equality constraint)**.** *Let Assumptions 1, 23 – 24 hold. Then the steady-state solution of the closed-loop system* $x^+ = Ax + B\kappa_T(x) + B_d d_s$ *is asymptotically stable with* $\mathcal{X}_{N,T}$ *as the region of attraction. The Lyapunov function is*

$$\tilde{V}(x) := V_N(x) + \lambda_s'[x - x_s] - N\ell_E(x_s, u_s)$$

Theorem 25 allows us to conclude that rolling horizon optimization in which we solve optimization problem (5.13) at each sampling instance steers any initial state $x(0) \in \mathcal{X}_{N,T}$ to the steady-state x_s. Therefore, in contrast to the closed-loop observed in Figure 5.1, a controller optimizing (5.13) would have never left the steady-state, thereby only incurring a cost of 220 per period. In Figure 5.2, we plot the closed-loop for a $x(0) \neq x_s$ (using the same cost vector used in the previous section).

Although economic MPC with terminal equality constraints stabilizes the supply chain system, notice that the unique steady-state for the states that is obtained by solving the linear program (5.11) is on one of the vertices of the hyperbox $\mathbb{X}$. More importantly, since the cost vector q is composed of inventory holding and lost sales costs, all its elements are strictly positive. Therefore, the solution of (5.11) is $x_s = \underline{\mathrm{x}}$. This steady state value of the state variables has important implications which motivates us to formulate the multiobjective supply chain MPC with terminal region in the next section. It is important to note that (i) $\mathcal{X}_{N,T}$ comprises of $x \geq \underline{\mathrm{x}}$ and (ii) since the steady state does not lie in the interior of $\mathbb{X}$, we cannot use the terminal penalty/ region formulation that is discussed in the next section to stabilize economic MPC.

Supply chain managers often balance economic objectives with that of risk minimization. That is, in addition to minimizing costs (or maximizing profits), the manager also tries to minimize risk by maintaining or by tracking inventory around a safety-stock level (that could determined by minimizing the probability of stock-out etc. or is the solution of stochastic inventory control algorithms like Federgruen (1993); Shang and Song (2003); Dong and Lee (2003); Gallego and Özer (2005); Chen and Song (2001)). As we stated above, stabilizing economic-MPC can only stabilize $x_s = \underline{\mathrm{x}}$. Therefore, to incorporate the managers choice to track safety stocks, we introduce a tracking stage cost (5.4) and minimize a combined economic and tracking objective in the next section.

5.1.2 Terminal region formulation

In order to allow the practitioner to implement a controller that tracks inventories to a steady-state as well as optimize the economics of the system, we use the following stage cost,

$$\ell(x,u) = \frac{\omega}{s_E}\ell_E(x,u) + \frac{(1-\omega)}{s_T}\ell_T(x,u;z_p) \tag{5.17}$$

in which $\omega \in (0,1)$ is a relative weighting chosen by the practitioner between the tracking and the economic stage costs. We use the parameters Q, R in $\ell_T(x,u)$ and ω as tuning parameters for the multiobjective MPC controller. The parameters (s_E, s_T) are scaling constants while $z_p = (x_t, u_t)$ are the tracking set-points.

We choose the input target u_t to be the "economic" input that satisfies the nominal demand, that is $u_t = u_s$ in which u_s is the solution to (5.12). The target set-point for the states is $x_t = x_{\text{safety}}$. The steady-state optimization problem now becomes

$$\begin{aligned}
&\min_{x,u} \left(\frac{\omega}{s_E}(q'x + r'u) + \frac{(1-\omega)}{s_T}((x - x_{\text{safety}})'Q(x - x_{\text{safety}}) + (u-u_s)'R(u-u_s))\right) \\
&\text{s.t. } x = Ax + Bu + B_d d_s, u \in \mathbb{U}, x \in \mathbb{X}
\end{aligned} \tag{5.18}$$

To obtain the scaling parameters s_T, s_E, we consider the utopia and nadir points of the individual stage costs $\ell_E(x,u)$ and $\ell_T(x,u;z_p)$ (Kim and De Weck, 2005). Denoting $z = (x,u)$, we obtain

$$z_E = \arg\min_{z \in \mathbb{X}\times\mathbb{U}} \ell_E(x,u), \quad z_T = \arg\min_{z\in\mathbb{X}\times\mathbb{U}} \ell_T(x,u;z_p)$$

The utopia point is

$$J^U = (\ell_E(z_E), \ell_T(z_T;z_p)) \in \mathbb{R}^2$$

The nadir point is

$$J^N = (\ell_E(z_T), \ell_T(z_E;z_p)) \in \mathbb{R}^2$$

The parameters s_T, s_E are then defined as:

$$(s_E, s_T) = J^N - J^U$$

Note that the optimization problem (5.18) is a quadratic program with a positive definite Hessian, and hence an unique solution to (5.18) exists. Based on the choice ω and tuning parameters (Q, R), the MPC controller described in this section stabilizes a different x_s. That is, the choice of weighting given to the economic and tracking objectives decide what inventories the controller is going to stabilize.

In Figure 5.3, we plot the inventory steady-state as a function of ω. The parameters used are $Q = 10\text{diag}(1,1,1,1), R = 10^{-5}\text{diag}(1,1,1,1)$, $x_{\text{safety}} = (35,0,40,0)$. The economic costs are $q = (10,10,10,1), r = (10,0.1,10,100)$. The input and state constraints were chosen as

$$0 \leq x \leq 100 \qquad 0 \leq u \leq 20$$

The nominal demand d_s was 10. Note that by choice of the state targets,the steady-state back-orders at both nodes is zero.

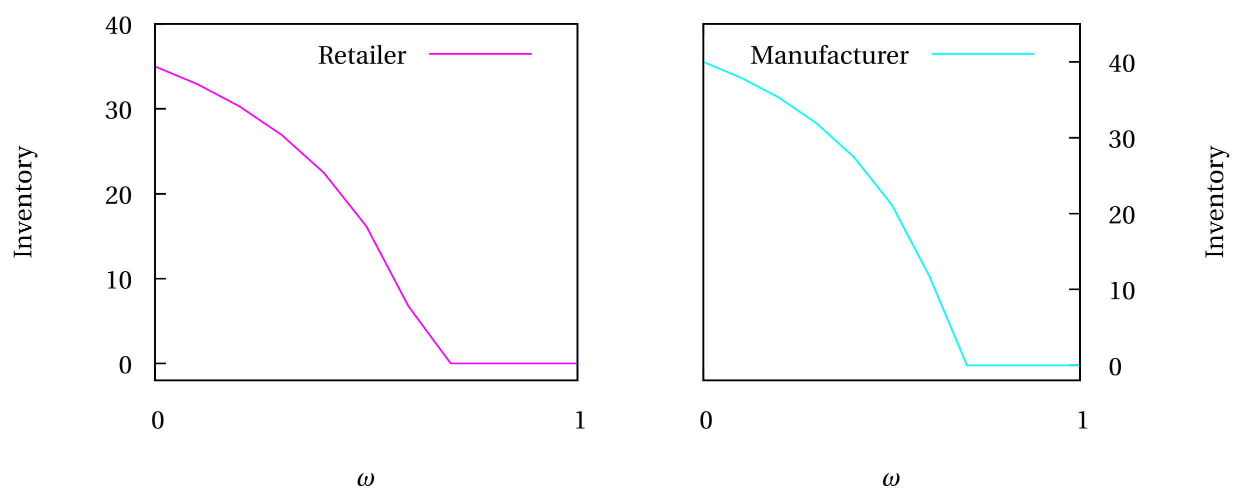

Figure 5.3: Steady-state as a function of the relative weighting between tracking and economics

Figure 5.3 shows the trade-off between choosing to rack to the safety stock and choosing to minimize the economics. As the economic weight increases, we see that the steady-state approaches the economic steady state.

Terminal region and Terminal penalty. Let Assumptions 23 and 24(a) and 24(c) hold. In Assumption 24, we use $\ell(x,u) - \ell(x_s,u_s)$ in which $\ell(x,u)$ is given by (5.17) and (x_s,u_s) is the solution of the steady-state problem (5.18) as the storage function. In addition, we make the basic stability assumption 2.8; with modifications to accommodate an economic stage cost as:

Assumption 26 (Basic stability assumption)**.** *There exists a convex, compact terminal region* $\mathbb{X}_f \subseteq \mathbb{X}$*, containing the point* x_s *in its interior and a control law* $\kappa_f : \mathbb{X}_f \to \mathbb{U}$ *and a function* $V_f : \mathbb{X}_f \to \mathbb{R}$ *such that the following holds*

$$V_f(Ax + B\kappa_f(x) + B_d d_s) \leq V_f(x) - \ell(x, \kappa_f(x)) + \ell(x_s, u_s), \forall x \in \mathbb{X}_f \tag{5.19}$$

$$Ax + B\kappa_f(x) + B_d d_s \in \mathbb{X}_f, \forall x \in \mathbb{X}_f \tag{5.20}$$

We now define the terminal penalty MPC problem:

$$\begin{aligned} \mathbb{P}_N(x) : \min_{\mathbf{u}} & \; V_N(\mathbf{u}; x, \mathbf{d}, N) \\ \text{s.t. } & x(j+1) = Ax(j) + Bu(j) + B_d d(j), j = \{0,1,\ldots,N-1\} \\ & \underline{x} \leq x(j) \leq \bar{x}, \qquad j = \{0,1,\ldots,N\} \\ & \underline{u} \leq u(j) \leq \bar{u}, \qquad j = \{0,1,\ldots,N-1\} \\ & x(0) = x \\ & x(N) \in \mathbb{X}_f \end{aligned} \tag{5.21}$$

in which $V_N(\mathbf{u}; x, \mathbf{d}, N) = \sum_{i=0}^{N-1} \ell(x(i), u(i)) + V_f(x(N))$.

Analogous to $\mathbb{Z}_T$ and $\mathcal{X}_{N,T}$, we define the sets $\mathbb{Z}_P$ and $\mathbb{X}_{N,P}$ (the subscript P refers to terminal penalty) as follows:

$$\mathbb{Z}_P := \left\{(x, \mathbf{u}) \mid \mathbf{u} \in \mathbb{U}^N, \phi(i; x, \mathbf{u}, \mathbf{d}_s) \in \mathbb{X}, \phi(N; x, \mathbf{u}, \mathbf{d}_s) \in \mathbb{X}_f\right\}$$

in which $\mathbf{d}_s$ is the nominal demand vector and $\phi(i; x, \mathbf{u}, \mathbf{d}_s)$ denotes the solution at time i under input $\mathbf{u}$ starting from x at time 0.

The projection of set $\mathbb{Z}_P$ onto the feasible state space $\mathbb{X}$ is called the set of admissible initial states, $\mathcal{X}_{N,P}$

$$\mathscr{X}_{N,P} := \left\{x \mid \exists \mathbf{u} \in \mathbb{U}^N \text{s.t. } (x, \mathbf{u}) \in \mathbb{Z}_P\right\}$$

We denote the optimal solution of problem (5.21) as $\mathbf{u}^0(x(k), \mathbf{d}_s)$. Denoting the first input in the sequence $\mathbf{u}^0(x(k), \mathbf{d}_s)$ as $\kappa_P(x(k))$, we obtain the closed-loop dynamics of the MPC algorithm as $x^+ = Ax + B\kappa_P(x) + B_d d_s$. The control law is $\kappa_P(x(k))$.

To show that the closed-loop using the control law $u = \kappa_P(x)$ is asymptotically stable, we need to make the following assumption:

Assumption 27 (Continuity of the storage function)**.** *The storage function $\lambda(\cdot)$ is continuous on* $\mathbb{X} \times \mathbb{U}$

The following theorem attributed to Amrit et al. (2011) establishes that the closed-loop $x^+ = Ax + B\kappa_P(x) + B_d d_s$ is asymptotically stable.

Theorem 28 (Lyapunov stability with terminal region)**.** *Let Assumptions 23, 24(a), 24(c), 26 and 27 hold. Then the steady-state solution x_s is an asymptotically stable equilibrium point of the system $x^+ = Ax + B\kappa_P(x) + B_d d_s$ with the region of attraction being any arbitrarily large compact subset of $\mathscr{X}_{N,P}$. The Lyapunov function is*

$$\bar{V}_N^0(\mathbf{u}; x(k), \mathbf{d}_s) = V_N^0(\mathbf{u}; x(k), \mathbf{d}_s) - N\ell(x_s, u_s) - \lambda(x_s) - V_f(x_s)$$

in which $V_N^0(\mathbf{u}; x(k), \mathbf{d}_s)$ is the optimal value function in the solution of problem (5.21).

We now discuss the choice of terminal region and terminal penalty for the supply chain model. Note that the optimal Lagrange multiplier for the equality constraints in steady-state problem (5.18) satisfies all the requirements in Assumptions 24 and 27.

For the supply chain model, we choose the terminal controller $\kappa_f(x) = Kx$, in which the gain K is such that $A_K := A + BK$ is a Hurwitz. That is, the system $\tilde{x}^+ = A_K\tilde{x}$ is asymptotically stable with the equilibrium point being x_s, in which $\tilde{x}$ is the deviation variable $x - x_s$. Note that the input is $\tilde{u} = K\tilde{x}$. Since the supply chain model (A, B) is stabilizable, such an K exists (for example, the infinite horizon unconstrained LQR gain). By the choice $\underline{x} = 0$ and backorder targets, some state constraints are active at the steady state. Hence, we find K using the technique given by Rao and Rawlings (1999). By this choice of the controller gain, we restrict the evolution of the closed-loop $(A+BK)\tilde{x}$ to the null space of the active constraints at the origin (the steady state is shifted to the origin in deviation variables). We define $Q_K = Q + K'RK$ and $q_K = q + K'r$. We choose the terminal penalty to be

$$V_f(x;x_s) = (x-x_s)'P(x-x_s) + p'(x-x_s) \tag{5.22}$$

in which the positive definite matrix P is the solution to the Lyapunov equation

$$A_K' P A_K - P = -Q_K$$

and p is the solution to

$$(A_K - I)' p = -q_K$$

In order that the basic stability assumption be satisfied, we require that

$$(x-x_s)'Q(x_s-x_t) \leq 0 \tag{5.23}$$

Therefore, we construct the terminal region $\mathbb{X}_f$ as the following set:

$$\mathbb{X}_f := \left\{x \mid A_K x + Bu_s + B_d d_s \in \mathbb{X}_f, K(x-x_s) + u_s \in \mathbb{U}, (x-x_s)'Q(x_s-x_t) \leq 0\right\} \tag{5.24}$$

Such a set can be constructed using the maximal output admissible set algorithm presented in Gilbert and Tan (1991) or by using the efficient algorithms presented in the MPT toolbox (Kvasnica, Grieder, and Baotić, 2006). In Figure 5.4, we plot the projection of the terminal region on the $\text{Iv}_1 - \text{Iv}_2$ plane. The cost functions and constraints were the same as in the steady-state calculation. The parameter ω is chosen to be 0.4.

In Figure 5.5, we plot the closed-loop response for different values of ω. We contrast the performance to a pure tracking-MPC that tracks to the steady-state solution of the multiobjective formulation using $\ell_T(x,u;z_p)$ as the stage cost. That is, we compare the closed-loop from solving problem (5.21) with the closed-loop for $\omega = 0$ and $z_p = (x_s, u_s)$. Note that, from the initial condition we chose $x(0) = (15, 10, 23, 0)$, economic-MPC that tracks to z_s is only feasible for $\omega < 0.6$ (since we need that $x(0) \geq x_s = \underline{\text{x}}$).

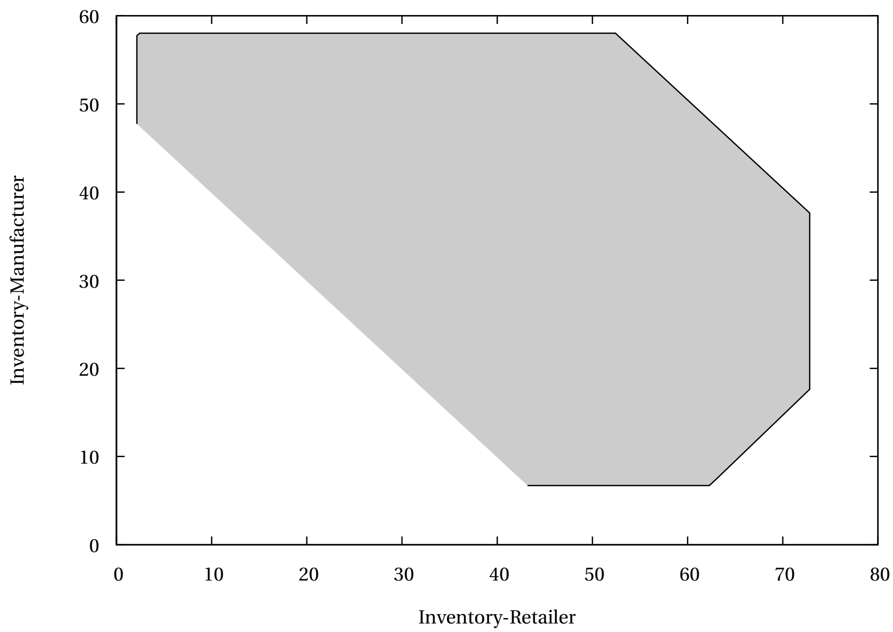

Figure 5.4: Projection of the terminal region onto the Inventory-plane for $\omega = 0.4$

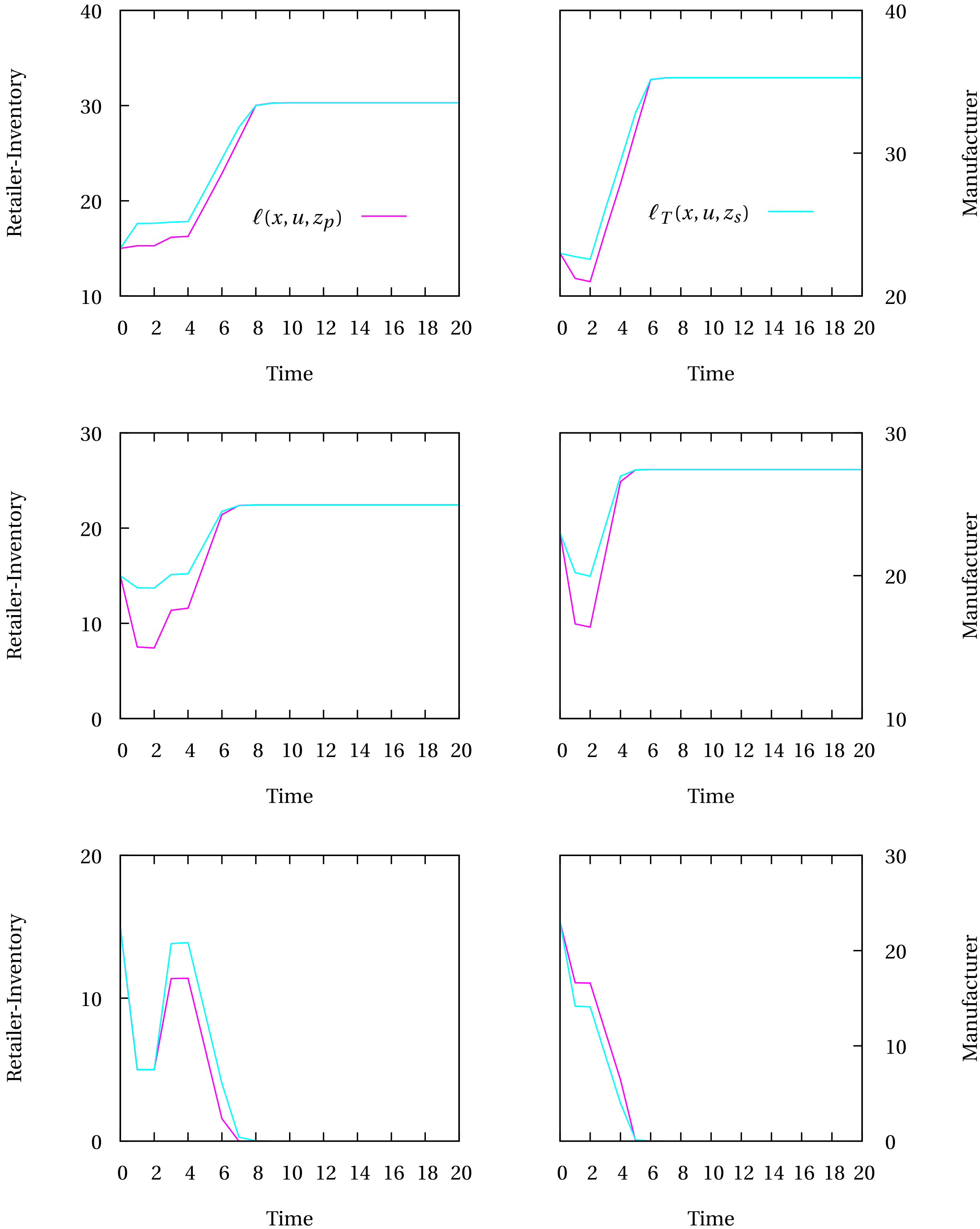

Figure 5.5: Closed-loop response for $\omega = 0.2$ (top), $\omega = 0.4$ (middle) and $\omega = 0.8$ (bottom)

In Table 5.1, we compare the economic cost incurred in using the three controllers: (i) Multi-objective MPC $\ell(x,u;z_p)$, (ii) Economic MPC $\ell_E(x,u)$ and (iii) Tracking MPC to the steady-state of the multiobjective MPC $\ell_T(x,u,z_s)$.

Table 5.1: Economic cost of implementing MPC

ω	Multiobjective$_{\times 10^4}$	Tracking$_{\times 10^4}$	Economic$_{\times 10^4}$
0	4.4342	4.4342	infeasible
0.2	4.0246	4.0645	infeasible
0.4	3.5617	3.6252	infeasible
0.8	2.2670	2.2670	2.2670
1.0	2.2670	2.2670	2.2670

We observe that as ω increases, that is, as the economic costs are given more weight, the steady-state approaches the constraint boundary. In such situations, the optimal input profile is dominated by feasibility and as such all the three stage costs incur the same economic cost in the closed loop [1]. However, for intermediate values of ω, that is, when the practitioner has comparable weighting to both tracking the safety stock and minimizing costs, multiobjective-MPC gives the best performance. While in the tracking MPC, we can design the system go to the same steady-state as multiobjective, the absence of economic knowledge in the tracking stage cost means that some economically more attractive transients are not considered by the online optimizer. While designing a controller that minimizes only the costs seems very attractive, the drawback is that for supply chains, such economic MPC can only stabilize steady states that lie on the one of the vertices of the constraint set. Therefore, we cannot use pure economic MPC to track the inventories to a target value.

[1]We optimize the open-loop cost. The numbers in Table 5.1 are just the economic cost of implementing the optimal input

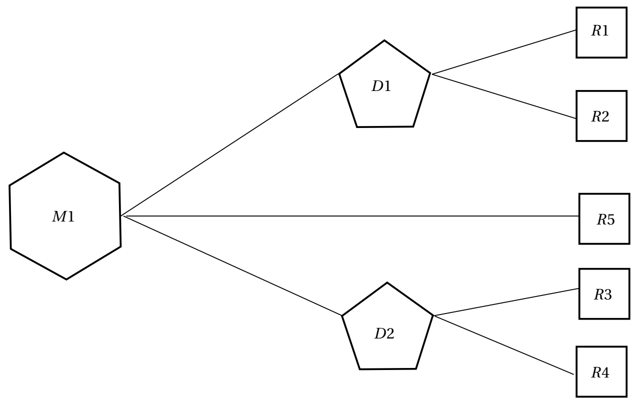

Figure 5.6: Multi-product, Multi-echelon supply chain studied

5.2 Multi-product, multi-echelon supply chain example

In this section, we follow the design procedure described in the previous section to implement model predictive control for a multi-product, multi-echelon supply chain. The supply chain that is studied is shown in Figure 5.6. It consists of a manufacturing facility $M1$ that supplies two products A, B to two distribution centers $D1, D2$ and a retailer $R5$. Distribution center $D1$ supplies the products to retailers $R1, R2$ while distribution center $D2$ supplies to $R3$ and $R4$.

We list the production lead-times at the manufacturing facility in Table 5.2. We assume that the manufacturing facility is able to produce both products simultaneously, with the only limitation being the combined storage of these products in the manufacturing node's storage facility (see Table 5.7).In Table 5.3, we list the transportation times between each node.

The retailers respond to customer demands which is assumed to arrive at each period following a normal distribution around a nominal demand. The nominal demand for each retailer node is listed in Table 5.4 while the variance of the demand signal in each retailer node for both products are listed in Table 5.5

For each node, we choose the target inventory to be the amount of product to be carried so that demands can be met for as long as the longest delay in the supply chain. The longest delay in the supply chain is 4 (the transportation time between $M1$ and $R5$). Hence, for the retailers, we choose the target inventory to be four times the nominal demand. For the distributors, the

Table 5.2: Production lead-times

Product	Lead-time
A	2
B	3

Table 5.3: Transportation lead-times

	$D1$	$D2$	$R1$	$R2$	$R3$	$R4$	$R5$
$M1$	2	1					4
$D1$			1	1			
$D2$					2	1	

Table 5.4: Nominal demand

	$R1$	$R2$	$R3$	$R4$	$R5$
A	3.0	4.5	5.0	2.0	4.0
B	4.2	3.1	1.4	2.5	4.2

Table 5.5: Variance of demand

	$R1$	$R2$	$R3$	$R4$	$R5$
A	1.1	1.3	1.1	1.2	1.4
B	1.3	1.4	1.1	1.1	1.4

target inventory is four times the nominal demand at the distributor. The nominal demand at the distributor is the sum of the demands at the retailers that is served by the distributor.

The target back-orders in each node is 0.

As mentioned earlier, each node has a combined inventory storage capacity. This capacity is listed in Table 5.7. The maximum storage is chosen to be greater than the target inventories.

The economic objective is the sum of the (i) inventory holding costs (ii) back-order costs and (iii) shipping and ordering costs. The coefficients for these costs are listed in the following tables.

In table 5.9, each entry corresponds to the shipping cost of product A and product B respectively. In addition, the cost coefficient of shipping products to the customers from the retailers is 1.

The ordering costs coefficients are all chosen to be 1, except for ordering between the $R5$ and $M1$ for which the cost-coefficient is chosen to be 0.5.

The production costs for product A is 10 per unit while that for B is 4 per unit.

The tracking objective is a weighted sum of the squares of the deviation of the inventories(and backorders) from their targets. These weights are chosen as 1 for inventory deviation (that is, we penalize $(\mathrm{Iv}-\mathrm{Iv}_t)^2$), 10 for backorder deviation $(10(\mathrm{BO}-\mathrm{BO}_t)^2)$. The inputs are penalized from their targets with a weight of 0.1 (the input targets are chosen to be the steady-state values as described below).

With the aforementioned details about the supply chain, the supply chain model can be written in the state space format (5.2) and the stage costs $\ell_E(\cdot,\cdot)$, (5.3) and $\ell_T(\cdot,\cdot)$, (5.4) is defined. Choosing an economic objective weight of 0.4, we can solve for the steady-state problem (5.18) to obtain the steady state. As discussed in the previous section, the multiobjective steady state lies between a pure tracking ($\omega = 0$) and a pure economic ($\omega = 1$) steady state. We re-iterate that the input steady-state remains the same irrespective of the objective function, because of the steady-state constraint that fixes all the flows in the supply chain in accordance with the nominal demand. In Table 5.10, we list the inventory steady-states (for product-A) for the pure economic, tracking and the multiobjective cost functions.

Table 5.6: Target inventories

	$M1$	$D1$	$D2$	$R1$	$R2$	$R3$	$R4$	$R5$
A	70	30	24	12	18	20	8	16
B	61	29.2	15.6	16.8	12.4	5.6	10	16.8

Table 5.7: Capacity constraints

	$M1$	$D1$	$D2$	$R1$	$R2$	$R3$	$R4$	$R5$
$\mathrm{Iv}_A + \mathrm{Iv}_B$	140	80	50	40	40	30	25	45

Table 5.8: State economic costs

	$M1$	$D1$	$D2$	$R1$	$R2$	$R3$	$R4$	$R5$
Inventory holding	1	1	1	1	1	1	1	1
Back-order	10	10	10	10	10	10	10	10

Table 5.9: Input costs

	$D1$	$D2$	$R1$	$R2$	$R3$	$R4$	$R5$
$M1$	(4,2)	(1,2)					(5,4)
$D1$			(1,1)	(1,1)			
$D2$					(2,2)	(1.5,1.5)	

Table 5.10: Steady state inventories for product A

	$M1$	$D1$	$D2$	$R1$	$R2$	$R3$	$R4$	$R5$
Tracking	70	30	24	12	18	20	8	16
Economic	0	0	0	0	0	0	0	0
Multiobjective($\omega = 0.4$)	57.93	17.93	11.93	0	5.93	7.93	0	3.93

(σ,Σ) **Policy.** We compare the closed-loop operation of the centralized multi-objective supply chain with that of the closed-loop dynamics due to a (σ,Σ) policy. In the (σ,Σ) policy, the nodes orders according to the following shipping and ordering policies. We denote the shipments coming from the upstream node as S^u, while the orders coming from downstream (demand for the retailer as O^d.

$$S(t) = \begin{cases} O^d(t) + \mathrm{BO}(t) & \text{if } \mathrm{Iv}(t) + S^u(t) - (O^d(t) + \mathrm{BO}(t)) \geq 0 \\ \mathrm{Iv}(t) + S^u(t) & \text{otherwise} \end{cases} \tag{5.25}$$

Having determined the shipment at time k, the node then places orders according to the inventory and backorder levels that the shipments will lead to at the next time as:

$$O(t) = \begin{cases} \Sigma - (\mathrm{Iv}(t+1) - \mathrm{BO}(t+1)) & \text{if } (\mathrm{Iv}(t+1) - \mathrm{BO}(t+1)) \leq \sigma \\ 0 & \text{otherwise} \end{cases} \tag{5.26}$$

The (σ,Σ) policy is a decentralized linear feedback policy. The retailer observes the demands, makes its ordering decisions which is then used by the distributor and so on.

5.2.1 Results

The supply chain described in the previous section was simulated for 50 days using a stochastic demand signal. The terminal condition used was that the system should be at the steady-state at the end of the prediction horizon, which was chosen as 15 days. Furthermore, we assumed that the MPC controller had perfect demand information for three days. For the remainder of the prediction horizon, we used the nominal demands as the demand forecast. For the (σ, Σ) policy, we chose $\sigma = 0.7\Sigma$ and Σ as the steady state inventory.

The initial inventories of the nodes were chosen as follows: All the backorders were 0 and

Table 5.11: Initial inventories

	$M1$	$D1$	$D2$	$R1$	$R2$	$R3$	$R4$	$R5$
A	63	14	24	0	2.1	3.1	3.1	0
B	40	12	7	0	1.2	1.2	0	5.2

the inputs were at their steady-state at the begining of the simulation.

In Figure 5.7, we report the ordering-profile in the supply chain, and compare the variance of the demands observed with the variance of the orders placed as we move upstream in the node. The variance in orders placed as we move upstream is an measure of the bullwhip effect. We see that the the MPC policy has less variance in ordering profile as we move upstream in the supply chain. This lower variance is because the MPC controller is a centralized controller that not only considers all the nodes together, but also makes predictions for two weeks into the future. Therefore, at certain nodes, the MPC controller is able to take advantage of higher inventory levels than the steady state inventory to place fewer orders in total. The (σ, Σ) policy shows the classical bullwhip effect of the variance of the orders increasing as we move upstream. In Figure 5.8, we plot the orders placed by the MPC controller and the (σ, Σ) policy controller in response to demands of product A at Retailer $R3$.

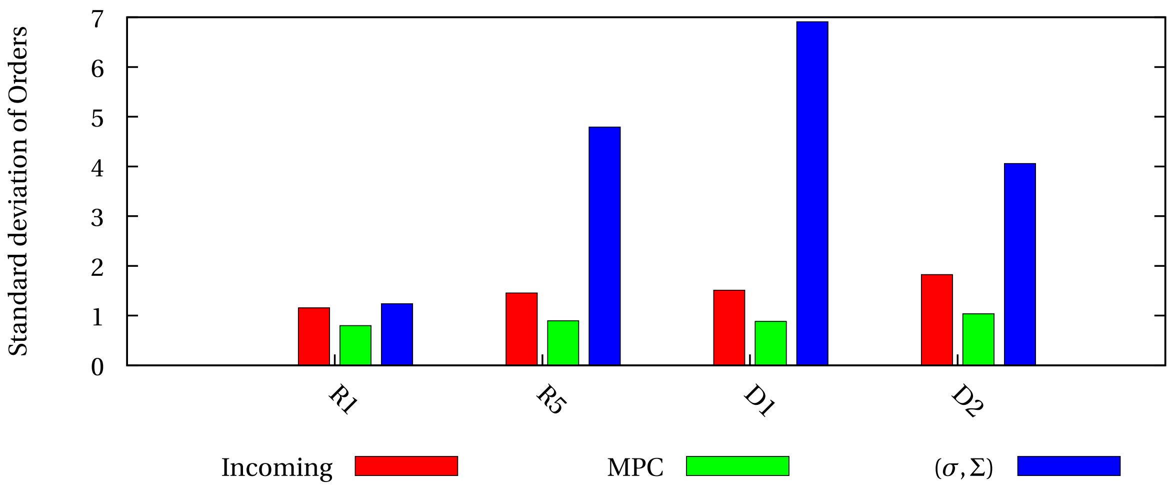

Figure 5.7: Bullwhip effect

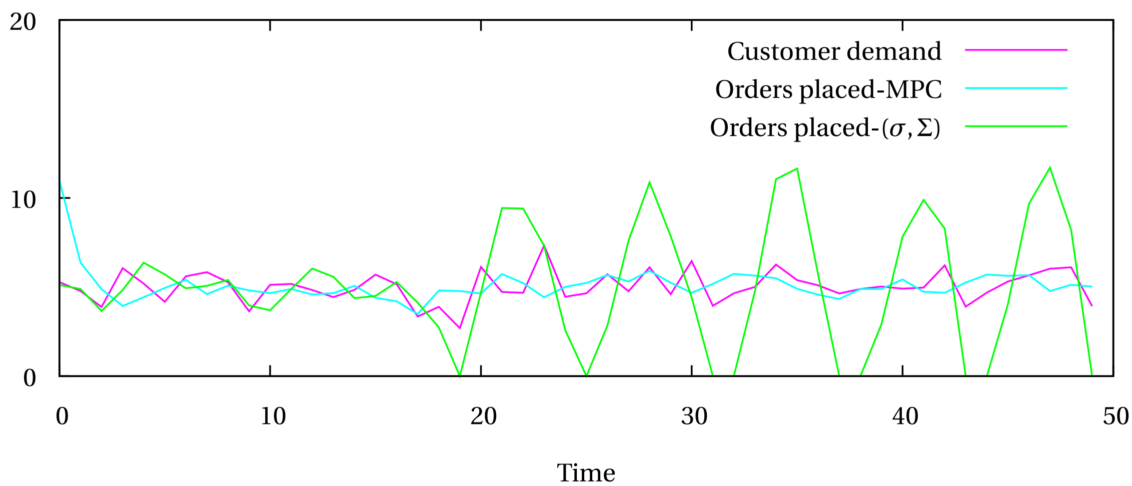

Figure 5.8: Ordering profile at $R3$

In Figure 5.9, we plot the inventory and the back-order of of the product A at retailer $R3$. Note that the steady-state for product A was 7.93.

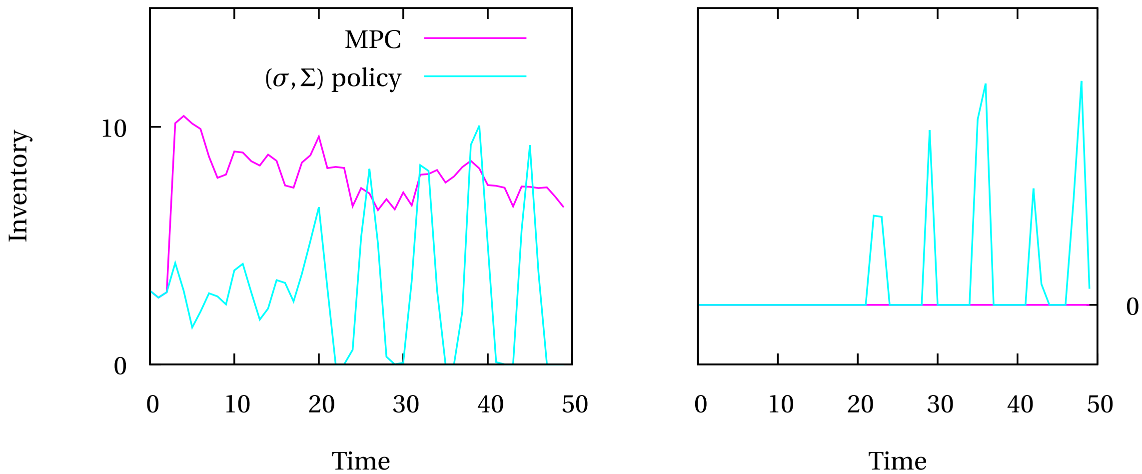

Figure 5.9: Inventory and Backorder profile at $R3$

Finally, in Table 5.12, we list the average inventory at each node for products A and B. Observe that the average inventory at the nodes for the MPC policy is much closer to the steady-state values, indicating that the MPC policy is stabilizing. This example also illustrates the inherent robustness of economic MPC as the controller was able to reject small variations in demand around the nominal demand.

5.2.2 Scheduling model

In the preceding sections, we did not consider the scheduling problem at the manufacturing unit; instead, we approximated the production delay using a constant production lead-time. In this section, we study the economic MPC closed-loop performance for a supply chain which includes a scheduling model for the manufacturing node.

We consider the same supply chain as in the previous section (Fig 5.6), but with the following modification. The manufacturing facility is assumed to have one unit U which can make both products A and B. The unit needs 3 time units to make A via the task TA and 2 time units to

Table 5.12: Average inventory for Product-A

	$M1$	$D1$	$D2$	$R1$	$R2$	$R3$	$R4$	$R5$
MPC	58.01	17.48	12.36	0.34	5.43	7.71	0.26	3.44
Policy	82.91	10.70	6.02	0.011	1.39	3.21	0.23	5.47

make B via the task TB. In addition, there is changeover time of 1 unit whenever the task is changed from A to B or vice-versa.

Denote the set $\mathbf{I} = \{\text{TA}, \text{TB}\}$, $\text{PT}(i)$ as the production lead time for each $i \in \mathbf{I}$ and $\text{CHT}(i, i')$ as the changeover time between i to i'. The scheduling constraints on the manufacturing node can now be enforced by the following inequalities (see 3.4):

$$\begin{aligned}
&\sum_{i\in\mathbf{I}} \sum_{t'=t-\tau_i+1}^{t} W_{i,t'} + \sum_{\substack{i'\in\mathbf{I}\\ i'\neq i}} \sum_{t'=t-\text{CHT}(i,i')+1}^{t} Z_{i,i',t'} \leq 1 && \forall t \\
&\sum_{t'=t-\tau_i+1}^{t} W_{i,t'} = Y_{i,t} && \forall t, \forall i \in \mathbf{I} \\
&X_{i,t} \geq Y_{i,t} && \forall t, \forall i \in \mathbf{I} \\
&\sum_{i\in\mathbf{I}} X_{i,t} = 1 && \forall t \\
&Z_{i,i',t} \leq X_{i,t-1} && \forall t, \forall i \in \mathbf{I}, i' \in \mathbf{I}, i' \neq i \\
&Z_{i,i',t} \leq X_{i',t} && \forall t, \forall i \in \mathbf{I}, i' \in \mathbf{I}, i' \neq i \\
&Z_{i,i',t} \geq X_{i,t-1} + X_{i',t} - 1 && \forall t, \forall i \in \mathbf{I}, i' \in \mathbf{I}, i' \neq i
\end{aligned} \tag{5.27}$$

To model the changeover time, we introduce three new binary variables $Z_{i,i',t}$, $Y_{i,t}$ and $X_{i,t}$. The binary variable $Z_{i,i',t}$ is 1 when a changeover is effected from task i to task i' at time t. The binary variable $Y_{i,t}$ is 1 if the task i was started during $[t-\tau_i, t]$. The binary variable $X_{i,t}$ is 1 if the last task to be performed in the unit before time t was i.

Table 5.13: Production costs

	Cost
Start a batch of TA	10
Start a batch of TB	6
Changeover from TA to TB	10
Changeover from TB to TA	5

In addition to these constraints, the batch-size is also controlled using $W_{i,t}$ and the maximum and minimum batch-sizes. For the supply chain studied, the maximum batch size was chosen to be 200 for both the products. The cost function corresponding to the new variables included a cost to start a batch and a cost to effect a changeover (see Table 5.13).

Following the procedure outlined in Chapter 3, we can write the supply chain dynamics with the binary variables in the state space format (5.2). We denote the state space model of the manufacturer scheduling problem as

$$x_M^+ = A_M x_M + B_M u_M + B_{d,M} u_{-M} \tag{5.28}$$

in which u_{-M} are the inputs of the other nodes in the supply chain, that is, the shipping and ordering among $(D1, D2, R1 - R5)$ as well as the orders placed by them to $M1$. The input u_M consists of shipments sent by the manufacturer to the other nodes, and the binary decisions given in (5.27).

Therefore, we can easily write the whole supply chain dynamics model in the state space form (5.2). We only note that the inputs now include the binary choices at the manufacturer and the constraint set, instead of being just $x \in \mathbb{X}, u \in \mathbb{U}$, becomes

$$x \in \mathbb{X}, u \in \mathbb{U} \tag{5.29}$$

$$\underline{b} \leq E_x x(t) + E_u u(t) \leq \overline{b} \tag{5.30}$$

with the second inequality representing the assignment constraint (5.27) in the state space form.

We now define the periodic optimization problem that is used to find a sub-optimal infinite horizon schedule and shipping/ordering policies in response to the nominal demand given in Table 5.4. In the previous section, we found the steady-state shipping and ordering so that the supply chain returns to the same state at the next sampling instance. In this section, we find a periodic policy so that after N_p sampling periods, the supply chain returns to the starting state. The interpretation is that, if we observe the nominal demands over an infinite horizon, then we can remain feasible with the periodic policy (by repeating it infinitely). The optimization problem solved is [2]:

$$\begin{aligned}
\mathbb{P}_p : \min_{\mathbf{u}, x(0)} & \sum_{i=0}^{N_p - 1} \ell_E(x(i), u(i), d_s(i)) \\
\text{s.t. } & x(i+1) = Ax(i) + Bu(i) + B_d d_s(i), i = \{0, 1, \ldots, N_p - 1\} && (5.31) \\
& \text{Constraints(5.29)} \qquad i = \{0, 1, \ldots, N_p - 1\} \\
& x(0) = x(N_p) \\
& && (5.32)
\end{aligned}$$

in which N_p is the period. We denote the solution to (5.31) by $(\mathbf{u}_p^0, x(0)_p^0)$, The solution to (5.31) gives us the periodic state-profile

$$\mathbb{X}_p = \left\{ x_p^0(0), x(1; x_p^0(0), \mathbf{u}_p^0, \mathbf{d}_s), \ldots, x(N_p; x_p^0(0), \mathbf{u}_p^0, \mathbf{d}_s) \right\} \tag{5.33}$$

In Figure 5.10, we show the Gantt chart for the periodic schedule with $N_p = 24$.

5.2.2.1 Dynamic Response

In this section, we show the dynamic response to a stochastic demand signal.

[2]Note that we are solving a purely economic problem

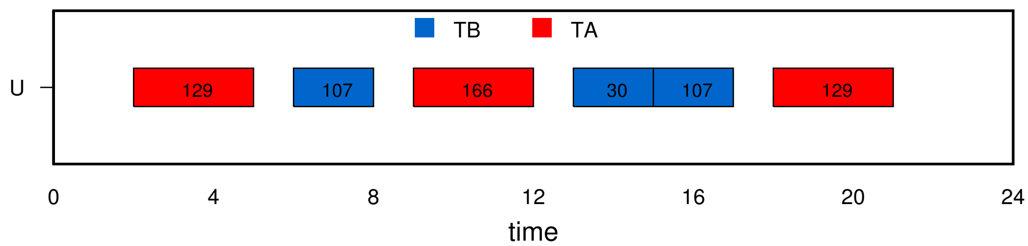

Figure 5.10: Periodic production schedule to respond to nominal demands

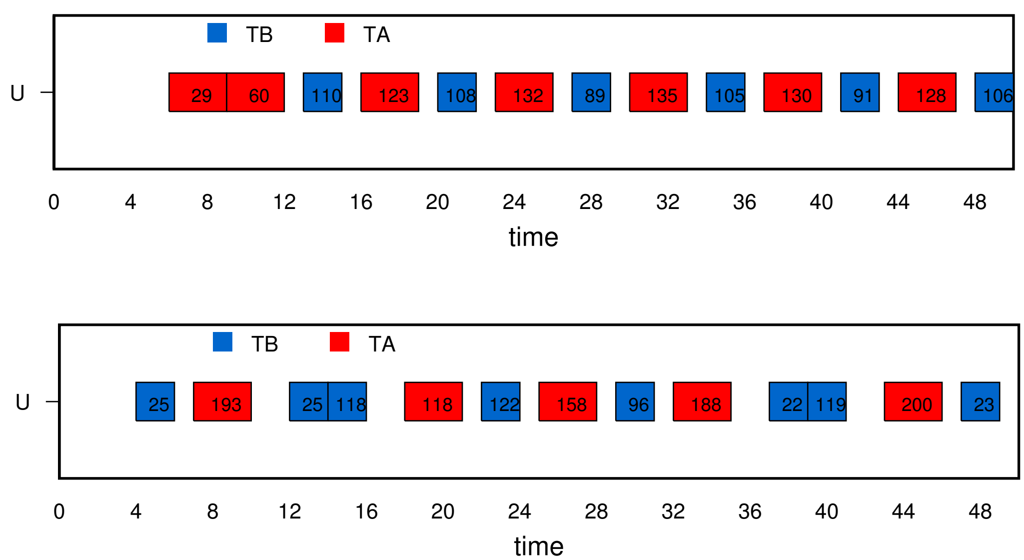

Figure 5.11: Production schedule for the MPC without terminal constraints that optimized (5.34) (Top) compared with production schedule for the MPC with terminal constraints that optimized (5.36). Note how larger batches are made for the problem with terminal constraints

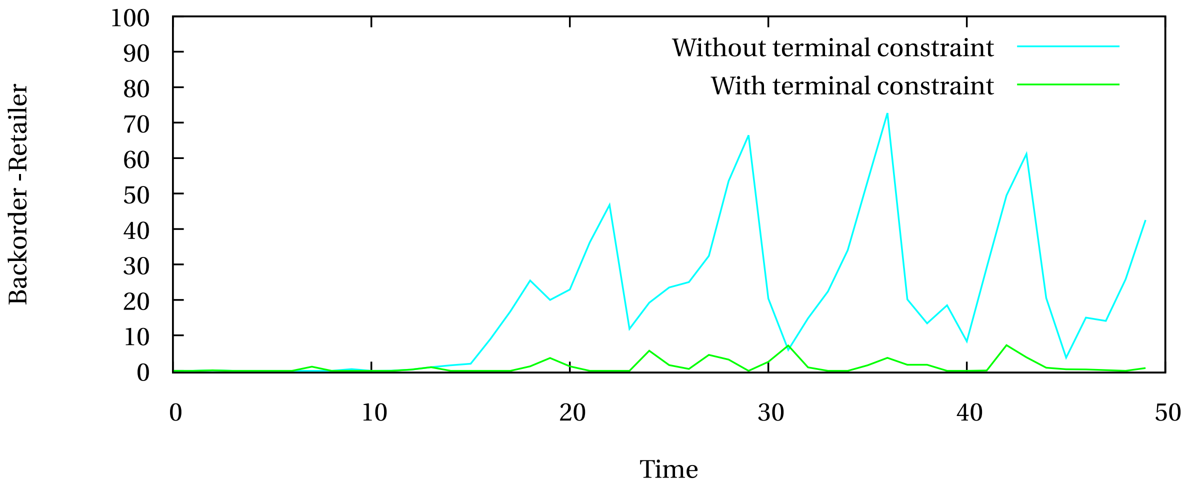

Figure 5.12: Combined backorder at all the retailer nodes

We first consider the closed-loop solution, in which, at each sampling time optimization problem (5.34) is solved, and compare it with the closed-loop solution in which the optimization problem (5.36) is solved. In (5.36), we force the final state to lie on the periodic profile $\mathbb{X}_p$. Hence, at the end of the planning horizon N, the supply-chain is at a state from which it can respond to the nominal demand. In contrast, for the optimization problem (5.34), the terminal state is chosen without considering future demands that arrive after the planning horizon N. Therefore, the solutions to (5.34) have (i) fewer production and (ii) lesser inventory at nodes. This leads to increasing backorders when new demands are observed at the next sampling time. In Figure 5.12, we compare the backorders observed in the closed-loop when the MPC optimizer solved (5.34) and (5.36). In Figure 5.10, we show the Gantt chart for the implemented schedule by the two MPCs. The planning horizon used was $N = 12$. Note that, under the presence of persistent disturbance (that is different from the nominal disturbance), the MPC design has to be robust (Rawlings and Mayne, 2009, Ch 3.). In this example, we have not designed robust-MPC but instead the results show the inherent-robustness of nominal MPC to reject small deviations from the nominal demand.

$$
\begin{aligned}
\mathbb{P}_N(x) : \min_{\mathbf{u}} & \sum_{i=0}^{N-1} \ell_E(x(j), u(j), d_s(j)) \\
& \text{s.t. } x(j+1) = Ax(j) + Bu(j) + B_d d_s(j), j = \{0, 1, \ldots, N-1\} \qquad (5.34) \\
& \text{Constraints(5.29)} \qquad j = \{0, 1, \ldots, N-1\} \\
& x(0) = x
\end{aligned}
\tag{5.35}
$$

$$
\begin{aligned}
\mathbb{P}_N(x) : \min_{\mathbf{u}} \sum_{i=0}^{N-1} & \ell_E(x(j), u(j), d_s(j)) \\
& \text{s.t. } x(j+1) = Ax(j) + Bu(j) + B_d d_s(j), j = \{0, 1, \ldots, N-1\} && (5.36) \\
& \text{Constraints(5.29)} \qquad j = \{0, 1, \ldots, N-1\} \\
& x(0) = x \\
& x(N) \in \mathbb{X}_P && (5.37)
\end{aligned}
$$

Chapter 6

Conclusions and Future work

We conclude with a summary of contributions and suggest possible directions for further research.

Contributions

Cooperative MPC for linear systems: In Chapter 2 we provided an overview of cooperative MPC for linear systems. The main contribution in Chapter 2 were (i) The extension of the class of systems for which cooperative MPC is applicable to all centralized stabilizable systems and, (ii) Tube based robust cooperative MPC to avoid centralized restarts if the warm start fails.

State-space models for scheduling: In Chapter 3, we provided a state-space model for scheduling. We expressed the scheduling problem as a dynamic problem for iterative scheduling. We also modeled a variety of scheduling disturances so that rescheduling occurs "naturally" in iterative scheduling. Finally, we used tools from MPC to demonstrate design of closed-loop scheduling problems with guaranteed recursive feasibility.

MPC for supply chains: A goal of this thesis was to use MPC as a general purpose tool for enterprise wide optimization. We demonstrated MPC design for dynamic supply chain models. The main contribution of this thesis is to complement the research in MPC/ Rolling horizon optimization frameworks for supply chain management by (i) showing the desirable properties of algorithms that guarantee closed-loop stability and, (ii) demonstrating the design of such control policies for supply chains. The main message of the thesis is that future researchers should appreciate the importance of considering the closed-loop dynamics of the supply chain as a result of the input actions taken.

1. In order to appeal to the distributed nature of decision making in supply chains, we demonstrated cooperative MPC for supply chains in which each node makes its local decisions but with a global vision in Chapter 4. We proposed a new cooperative MPC iteration scheme which closely resembles the current decision making hierarchy in linear supply chains.

2. Since supply chains directly optimize the economics, we demonstrated design of Economic MPC for supply chains in Chapter 5. We proposed a multiobjective stage cost, that not only accounts for supply chain costs, but also for supply chain risks. The supply chain is stabilized at a steady-state that reflects the managers' choice, i.e., risk seeking or risk averse.

Integration of scheduling and control: We demonstrated a supply chain example with an integrated scheduling model for the manufacturing plant. We showed the integration of the MPC design tools from Chapter 2 and Chapter 5 along with the state-space scheduling model from Chapter 3 to guarantee recursive feasibility for the integrated supply chain model. We also showed the inherent robustness of the proposed approach to small deviations from the nominal demand.

Future work

Terminal conditions for the scheduling problem: In Chapter 3, we showed recursive feasibility by using a cyclic schedule as the terminal condition. In many cases, we might not be able to find any cyclic schedule for the scheduling problem. In such cases, we have to find other suitable terminal conditions. One such idea that we are currently exploring is to find safety constraints on inventory from a scheduling point of view. Methods of finding terminal conditions for the scheduling problem is an important area for future research.

Hybrid control theory: The scheduling state space model comprises both of continuous variables (like the inventories, batch sizes, etc.) and discrete variables (like assignment, changeover etc.). For such systems, we need to study the stability theory for hybrid systems. There are methods that have been developed for hybrid dynamic systems consisting of both time and event driven dynamics (Bemporad and Morari, 1999; Morari, Baotic, and Borrelli, 2003) etc. More recently, Lyapunov stability theory for hybrid dynamic systems also have been studied (Lazar and Heemels, 2009; Lazar, Heemels, and Teel, 2009). Application and development of hybrid theory to prove stability of scheduling models is a challenging research problem.

Impact of forecast: Supply chains are described as "pull" systems because the dynamics is activated when the customer pulls products from the supply chain. As such, the supply chain is sensitive to customer demands, price signals, etc. MPC theory has been mostly built around dynamic models trying to "reject" external disturbances (e.g.,the nominal case is when there is no disturbance affecting the system). The impact of demand/ price forecast the supply chain steady state; performance, etc. are yet to be studied.

Robust terminal conditions: In this thesis, we developed algorithms based on a nominal demand signal. That is, the stability and convergence guarantees; and specially, the design of terminal region/ constraints were based on the nominal demand. In practice, it is desirable to design the terminal constraint so that we are robust to some known distribution of demands. Design of such terminal regions and integration with MPC technology remains an avenue of future work.

Cooperative game theory: The cooperative MPC tools have been developed for process industries to coordinate multiple MPC's in a single plant. Therefore, it is reasonable to assume that all the subsystems can share models and objectives with each-other. In a supply chain, however,the nodes could be owned by different companies. Hence, we need to study the incentives to cooperate from a cooperative game theory point of view.

Implementation: The ultimate test for any new tool is practical implementation. An avenue of future research is the implementation of the tools described in this thesis for a large scale supply chain with real data. Not only, would such a study help validate the idea of using MPC for supply chains, it would also help us uncover new research topics.

Bibliography

Rishi Amrit, James B. Rawlings, and David Angeli. Economic optimization using model predictive control with a terminal cost. *Annual Rev. Control*, 35:178–186, 2011.

Bernhard J. Angerhofer and Marios C. Angelides. System dynamics modelling in supply chain management: Research review. In *Simulation Conference Proceedings, 2000. Winter*, volume 1, pages 342–350, 2000.

Panos J. Antsaklis and Anthony N. Michel. *Linear Systems*. McGraw-Hill, New York, 1997.

A. Atamtürk and M.W.P. Savelsbergh. Integer-programming software systems. *Ann. Oper. Res.*, 140(1):67–124, 2005.

S. Axsäter. A framework for decentralized multi-echelon inventory. *IIE Trans.*, 33:91–97, 2001.

S. Axsäter. *Inventory control*. Springer Verlag, 2006.

T. Backx, O. Bosagra, and W. Marquardt. Towards intentional dynamics in supply chain conscious process operations. In *FOCAPO*, 1998.

T. Backx, O. Bosgra, and W. Marquardt. Integration of model predictive control and optimization of processes. In *International Symposium on Advanced Control of Chemical Processes (ADCHEM 2000)*, volume 1, pages 249–260, June 2000.

J. J. Bartholdi and E. Kemahlioğlu-Ziya. Using shapley value to allocate savings in a supply chain. *Supply Chain Optim.*, 98:169–208, 2005.

Tamer Başar and Geert Jan Olsder. *Dynamic Noncooperative Game Theory*. SIAM, Philadelphia, 1999.

B. M. Beamon. Measuring supply chain performance. *Int. J. Oper. Prod. Manage.*, 19(3):275–292, 1999.

Benita M. Beamon. Supply chain design and analysis: Models and methods. *Int. J. Prod. Econ.*, 55(3):281–294, 1998.

A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, 1999.

A. Bemporad, S. Di Cairano, and N. Giorgetti. Model predictive control of hybrid systems with applications to supply chain management. In *Congresso ANIPLA (Associazione Nazionale Per L'Automazione)*, 2005.

A. Ben-Tal and A. Nemirovski. Robust optimization–methodology and applications. *Math. Prog.*, 92(3):453–480, 2002.

A. Bensoussan, RH Liu, and S. P. Sethi. Optimality of an (s, S) policy with compound poisson and diffusion demands: A quasi-variational inequalities approach. *SIAM J. Cont. Opt.*, 44(5): 1650–1676, 2006.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1989.

S. Bose and J. F. Penky. A model predictive framework for planning and scheduling problems: A case study of consumer goods supply chain. *Comput. Chem. Eng.*, 24:329–335, 2000.

M. W. Braun, D. E. Rivera, W. M. Carlyle, and K. G. Kempf. A model predictive control framework for robust management of multi-product, multiechelon demand networks. In *IFAC, 15th Triennial World Congress*, 2002.

J. F. Burns and BD Sivazlian. Dynamic analysis of multiechelon supply systems. *Comput. Ind. Eng.*, 2(4):181–193, 1978.

G. P. Cachon. Supply chain coordination with contracts. *Handbooks Oper. Res. Manage. Sci.*, 11: 229–340, 2003.

G. P. Cachon and S. Netessine. Game theory in supply chain analysis. *Tutorials in Operations Research: Models, Methods, and Applications for Innovative Decision Making*, 2006.

G. P. Cachon and P. H. Zipkin. Competitive and cooperative inventory polices in a two stage supply chain. *Manage Sci.*, 45:936–953, 1999.

EF Camacho, DR Ramirez, D. Limon, D. Muñoz de la Peña, and T. Alamo. Model predictive control techniques for hybrid systems. 34(1):21–31, 2010.

F. Chen and J. S. Song. Optimal policies for multiechelon inventory problems with Markov-modulated demand. *Oper. Res.*, 49(2):226–234, 2001.

F. Chen, Z. Drezner, J. K. Ryan, and D. Simchi-Levi. Quantifying the bullwhip effect in a simple supply chain: The impact of forecasting, lead times, and information. *Manage Sci.*, 46:436–443, 2000a.

F. Chen, J. K. Ryan, and D. Simchi-Levi. The impact of exponential smoothing forecasts on the bullwhip effect. *Naval Res. Logist.*, 47(4):269–286, 2000b.

R. Cheng, J. F. Forbes, and W. S. Yip. Price-driven coordination method for solving plant-wide MPC problems. *J. Proc. Cont.*, 17(5):429–438, 2007.

P.D. Christofides, R. Scattolini, D.M. de la Peña, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Comput. Chem. Eng.*, 2012.

A. J. Clark and H. Scarf. Optimal policies for a multiechelon inventory problem. *Manage Sci.*, 6: 475–490, 1960.

M. Colvin and C.T. Maravelias. A stochastic programming approach for clinical trial planning in new drug development. *Comput. Chem. Eng.*, 32(11):2626–2642, 2008.

M. Colvin and C.T. Maravelias. Modeling methods and a branch and cut algorithm for pharmaceutical clinical trial planning using stochastic programming. *Eur. J. Oper. Res.*, 203(1): 205–215, 2010.

J. Dejonckheere, S. M. Disney, M. R. Lambrecht, and D. R. Towill. Measuring and avoiding the bullwhip effect: A control theoretic approach. *Eur. J. Oper. Res.*, 147(3):567–590, 2003.

J. Dejonckheere, S. M. Disney, M. R. Lambrecht, and D. R. Towill. The impact of information enrichment on the bullwhip effect in supply chains: A control engineering perspective. *Eur. J. Oper. Res.*, 153(3):727–750, 2004.

Moritz Diehl, Rishi Amrit, and James B. Rawlings. A Lyapunov function for economic optimizing model predictive control. *IEEE Trans. Auto. Cont.*, 56(3):703–707, 2011.

Stephen M. Disney, Denis R. Towill, and Roger D. H. Warburton. On the equivalence of control theoretic, differential, and difference equation approaches to modeling supply chains. *Int. J. Prod. Econ.*, 101(1):194 – 208, 2006.

D. Doan, T. Keviczky, I. Necoara, M. Diehl, and B. De Schutter. A distributed version of han's method for dmpc using local communications only. *Contr. Eng. and App. Info.*, 11(3):6–15, 2009.

M.D. Doan, T. Keviczky, and B. De Schutter. A dual decomposition-based optimization method with guaranteed primal feasibility for hierarchical mpc problems. In *18th IFAC World Congress*, 2011.

L. Dong and H. L. Lee. Optimal policies and approximations for a serial multiechelon inventory system with time-correlated demand. *Oper. Res.*, 51:969–980, 2003.

William B. Dunbar and S. Desa. Distributed model predictive control for dynamic supply chain management. In *Assessment and Future Directions of Nonlinear Model Predictive Control*. Springer, 2007.

A. Federgruen. Centralized planning models for multiechelon inventory systems under uncertainty. *Handbooks Oper. Res. Manage. Sci.*, 4:133–173, 1993.

A. Federgruen and P. Zipkin. Computational issues in an infinite-horizon, multiechelon inventory model. *Oper. Res.*, 32:818–836, 1984.

A. Federgruen and P. Zipkin. An inventory model with limited production capacity and uncertain demands II. The discounted-cost criterion. *Math. Oper. Res.*, 11(2):208–215, 1986a.

A. Federgruen and P. Zipkin. An inventory model with limited production capacity and uncertain demands I. The average-cost criterion. *Math. Oper. Res.*, 11(2):193–207, 1986b.

G. Gallego and O. Özer. A new algorithm and a new heuristic for serial supply systems. *Oper. Res. Lett.*, 33(4):349–362, 2005.

R. Ganeshan and T. P. Harrison. An introduction to supply chain management. Technical report, Department of Management Scicence and Information Systems, The Pennsylvania State University, University Park, PA, 1995.

Elmer G. Gilbert and Kok Tin Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Trans. Auto. Cont.*, 36(9):1008–1020, September 1991.

P. Giselsson and A. Rantzer. Distributed model predictive control with suboptimality and stability guarantees. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 7272–7277. IEEE, 2010.

P. Giselsson, MD Doan, T. Keviczky, B. De Schutter, and A. Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, 2012.

V. Goel and I.E. Grossmann. A class of stochastic programs with decision dependent uncertainty. *Math. Prog.*, 108(2):355–394, 2006.

B. Golany and U. G. Rothblum. Inducing coordination in supply chains through linear reward schemes. *Naval Res. Logist.*, 53(1):1–15, 2006.

I. E. Grossmann. Enterprise-wide optimization: A new frontier in process systems engineering. *AIChE J.*, 51:1846–1857, 2005.

R. W. Grubbström and O. Tang. An overview of input-output analysis applied to production-inventory systems. *Econ. Sys. Res.*, 12(1):3–25, 2000.

L. Grüne. Analysis and design of unconstrained nonlinear mpc schemes for finite and infinite dimensional systems. *SIAM J. Cont. Opt.*, 48(2):1206–1228, 2009.

W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, 2001.

Kai Hoberg, James R. Bradley, and Ulrich W. Thonemann. Analyzing the effect of the inventory policy on order and inventory variability with linear control theory. *Eur. J. Oper. Res.*, 176(3): 1620 – 1642, 2007.

SJ Honkomp, L. Mockus, and GV Reklaitis. A framework for schedule evaluation with processing uncertainty. *Comput. Chem. Eng.*, 23(4-5):595–609, 1999.

A. Huercio, A. Espuna, and L. Puigjaner. Incorporating on-line scheduling strategies in integrated batch production control. *Comput. Chem. Eng.*, 19:609–614, 1995.

D. L. Iglehart. Optimality of (s, S) policies in the infinite horizon dynamic inventory problem. *Manage Sci.*, pages 259–267, 1963.

S.L. Janak, C.A. Floudas, J. Kallrath, and N. Vormbrock. Production scheduling of a large-scale industrial batch plant. ii. reactive scheduling. *Ind. Eng. Chem. Res.*, 45(25):8253–8269, 2006.

B. Johansson, A. Speranzon, M. Johansson, and K.H. Johansson. Distributed model predictive consensus. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, pages 2438–2444, 2006.

Karl G. Kempf. Control oriented approaches to supply chain management in semiconductor manufacturing. In *Proceedings of the 2004 American Control Conference*, July 2004.

I.Y. Kim and OL De Weck. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Strucrual and Multidisciplinary optimization*, 29(2):149–158, 2005.

Ilya Kolmanovsky and Elmer G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Math. Probl. Eng.*, 4(4):317–367, 1998.

E. Kondili, C. C. Pantelides, and R. Sargent. A general algorithm for short term scheduling of batch operations–I. MILP formulation. *Comput. Chem. Eng.*, 17:211–227, 1993.

M. Kvasnica, P. Grieder, and M. Baotić. *Multi-Parametric Toolbox (MPT)*, 2006. URL `http://control.ee.ethz.ch/ mpt/`.

C. S. Lalwani, S. M. Disney, and D. R. Towill. Observable and controllable state space representations of a generalized order-up-to policy. *Int. J. Prod. Econ.*, 101(1):173–184, 2006.

M. Lazar and W. P. M. H. Heemels. Predictive control of hybrid systems: Input-to-state stability results for sub-optimal solutions. *Automatica*, 45(1):180–185, 2009.

M. Lazar, W. P. M. H. Heemels, and A. R. Teel. Lyapunov functions, stability and input-to-state stability subtleties for discrete-time discontinuous systems. *IEEE Trans. Auto. Cont.*, 54(10): 2421–2425, 2009.

H. L. Lee, V. Padmanabhan, and S. Whang. Information distortion in a supply chain: The bullwhip effect. *Manage Sci.*, 43:546–558, 1997a.

Hau L. Lee, V. Padmanabhan, and Seungjin Whang. The bullwhip effect in supply chains. *Sloan Manage. Rev.*, 38:93–102, 1997b.

M. Leng and M. Parlar. Game theoretic applications in supply chain management: A review. *INFOR*, 43(3):187–220, 2005.

M. Leng and M. Parlar. Allocation of cost savings in a three-level supply chain with demand information sharing: A cooperative-game approach. *Oper. Res.*, 57(1):200–213, 2009.

R. Levi, R. O. Roundy, D. B. Shmoys, and V. A. Truong. Approximation algorithms for capacitated stochastic inventory control models. *Oper. Res.*, 56(5):1184–1199, 2008.

X. Li and Q. Wang. Coordination mechanisms of supply chain systems. *Eur. J. Oper. Res.*, 179(1): 1–16, 2007.

Xiang Li and Thomas E. Marlin. Robust supply chain performance via model predictive control. *Comput. Chem. Eng.*, 33(12, Sp. Iss. SI):2134–2143, Dec. 2009.

Z. Li and M. Ierapetritou. Process scheduling under uncertainty: Review and challenges. *Comput. Chem. Eng.*, 32(4):715–727, 2008.

Z. Li and M. G. Ierapetritou. Rolling horizon based planning and scheduling integration with production capacity consideration. *Chem. Eng. Sci.*, 2010.

Zukui Li and Marianthi G. Ierapetritou. Reactive scheduling using parametric programming. *AIChE J.*, 54(10). ISSN 1547-5905.

Pin-Ho Lin, David Shan-Hill Wong, Shi-Shang Jang, Shyan Shu Shieh, and Ji-Zheng Chu. Controller design and reduction of bullwhip for a model supply chain system using z-transform analysis. *J. Proc. Cont.*, 14:487–499, September 2004.

Jinfeng Liu, X. Chen, David Muñoz de la Peña, and Panagiotis D. Christofides. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. *AIChE J.*, 56(5):2137–2149, 2010.

Y. Ma, G. Anderson, and F. Borrelli. A distributed predictive control approach to building temperature regulation. In *American Control Conference (ACC), 2011*, pages 2089–2094. IEEE, 2011.

J. M. Maestre, D. Muñoz de la Peña, and E. F. Camacho. Distributed MPC: A supply chain case study. In *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, 2009.

J. M. Maestre, D. Muñoz de la Peña, and E. F. Camacho. Distributed model predictive control based on a cooperative game. *Optimal Cont. Appl. Meth.*, 32(2):153–176, 2011a.

JM Maestre, D. Muñoz de la Peña, EF Camacho, and T. Alamo. Distributed model predictive control based on agent negotiation. *J. Proc. Cont.*, 21(5):685–697, 2011b.

Christos T. Maravelias and Charles Sung. Integration of production planning and scheduling: Overview, challenges and opportunities. *Comput. Chem. Eng.*, 33:1919–1930, 2009.

C.T. Maravelias. General framework and modeling approach classification for chemical production scheduling. *AIChE J.*, 2012.

Natalia I. Marcos, J. Fraser Forbes, and Martin Guay. Coordination of distributed model predictive controllers for constrained dynamic processes. In *ADCHEM 2009, International Symposium on Advanced Control of Chemical Processes*, Istanbul, Turkey, July 12-15, 2009.

C.A. Méndez and J. Cerdá. Dynamic scheduling in multiproduct batch plants. *Comput. Chem. Eng.*, 27(8):1247–1259, 2003.

C.A. Méndez, J. Cerdá, I.E. Grossmann, I. Harjunkoski, and M. Fahl. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput. Chem. Eng.*, 30 (6):913–946, 2006.

Esen Mestan, Metin Türkay, and Yaman Arkun. Optimization of operations in supply chain systems using hybrid systems approach and model predictive control. *Ind. Eng. Chem. Res.*, 45:6493–6503, August 2006.

M. Morari and J. H. Lee. Model predictive control: past, present and future. In *Proceedings of joint 6th international symposium on process systems engineering (PSE '97) and 30th European symposium on computer aided process systems engineering (ESCAPE 7)*, Trondheim, Norway, 1997.

M. Morari, M. Baotic, and F. Borrelli. Hybrid systems modeling and control. *Eur. J. Control*, 9 (2-3):177–189, 2003.

P.D. Moroşan, R. Bourdais, D. Dumur, and J. Buisson. A distributed mpc strategy based on bendersâĂŹ decomposition applied to multi-source multi-zone temperature regulation. *J. Proc. Cont.*, 21(5):729–737, 2011.

M. Moses and S. Seshadri. Policy mechanisms for supply chain coordination. *IIE Trans.*, 32(3): 245–262, 2000.

Thierry Moyaux, Brahim Chain-draa, and Sophie D'Amours. Information sharing as a coordination mechanism for reducing the bullwhip effect in a supply chain. *IEEE T. Syst. Man Cy. C*, 37(3):396–409, 2007.

M.A. Müller, M. Reble, and F. Allgöwer. Cooperative control of dynamically decoupled systems via distributed model predictive control. *Int. J. Robust and Nonlinear Control*, 2012.

S.A. Munawar and R.D. Gudi. A multilevel, control-theoretic framework for integration of planning, scheduling, and rescheduling. *Ind. Eng. Chem. Res.*, 44(11):4001–4021, 2005.

M. Nagarajan and G. Sošić. Game-theoretic analysis of cooperation among supply chain agents: Review and extensions. *Eur. J. Oper. Res.*, 187(3):719–745, 2008.

J. Nash. Noncooperative games. *Ann. Math.*, 54:286–295, 1951.

I. Necoara, V. Nedelcu, and I. Dumitrache. Parallel and distributed optimization methods for estimation and control in networks. *J. Proc. Cont.*, 21(5):756–766, 2011.

Ion Necoara, Dang Doan, and J. A. K Suykens. Application of the proximal center decomposition method to distributed model predictive control. In *Proceedings of the IEEE Conference on Decision and Control*, Cancun, Mexico, December 9-11 2008.

J.M. Novas and G.P. Henning. Reactive scheduling framework based on domain knowledge and constraint programming. *Comput. Chem. Eng.*, 34(12):2129–2148, 2010.

M. Ortega and L. Lin. Control theory applications to the production–inventory problem: A review. *Int. J. Prod. Res.*, 42:2303–2322, 2004.

Gabriele Pannocchia, James B. Rawlings, and Stephen J. Wright. Conditions under which suboptimal nonlinear MPC is inherently robust. *Sys. Cont. Let.*, 60:747–755, 2011.

Lazaros G. Papageorgiou. Supply chain optisation for the proces industries: Advances and opportunities. *Comput. Chem. Eng.*, 33:1931–1938, 2009.

CI Papanagnou and GD Halikias. Supply-chain modelling and control under proportional inventory-replenishment policies. *Int. J. Sys. Sci.*, 39(7):699–711, 2008.

G. Perakis and G. Roels. The price of anarchy in supply chains: Quantifying the efficiency of price-only contracts. *Manage Sci.*, 53(8):1249–1268, 2007.

Edgar Perea López, I. Grossmann, E. Ydstie, and T. Tahmassebi. Dynamic modeling and classical control theory for supply chain management. *Comput. Chem. Eng.*, 24:1143–1149, 2000.

Edgar Perea López, Ignacio E. Grossmann, B. Erik Ydstie, and Turaj Tahmassebi. Dynamic modeling and decentralized control of supply chains. *Ind. Eng. Chem. Res.*, 40:3369–3383, June 2001.

Edgar Perea López, B. Erik Ydstie, and Ignacio E. Grossmann. A model predictive control strategy for supply chain optimization. *Comput. Chem. Eng.*, 27(8-9):1201–1218, February 2003.

M.L. Pinedo. *Scheduling: theory, algorithms, and systems.* Springer, 2008.

M. L. Puterman. *Markov Decision Process: Discrete Stochastic Dynamic Programing.* John Wiley and Sons, 2005.

S. Joe Qin and Thomas A. Badgwell. A survey of industrial model predictive control technology. *Control Eng. Prac.*, 11(7):733–764, 2003.

S. Raghunathan. Impact of demand correlation on the value of and incentives for information sharing in a supply chain. *Eur. J. Oper. Res.*, 146(3):634–649, 2003.

S. V. Raković, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne. Approximation of the minimal robustly positively invariant set for discrete-time LTI systems with persistent state disturbances. In *Proceedings 42nd IEEE Conference on Decision and Control*, volume 4, pages 3917–3918, Maui, Hawaii, USA, December 2003.

Christopher V. Rao and James B. Rawlings. Steady states and constraints in model predictive control. *AIChE J.*, 45(6):1266–1278, 1999.

James B. Rawlings and David Q. Mayne. *Model Predictive Control: Theory and Design.* Nob Hill Publishing, Madison, WI, 2009. 576 pages, ISBN 978-0-9759377-0-9.

James B. Rawlings and Brett T. Stewart. Coordinating multiple optimization-based controllers: New opportunities and challenges. *J. Proc. Cont.*, 18:839–845, 2008.

James B. Rawlings, Brett T. Stewart, Stephen J. Wright, and David Q. Mayne. Suboptimal MPC: On replacing the terminal contraint by a terminal cost, 2010. Internal communication.

S. Relvas, H.A. Matos, A.P.F.D. Barbosa-Póvoa, and J. Fialho. Reactive scheduling framework for a multiproduct pipeline with inventory management. *Ind. Eng. Chem. Res.*, 46(17):5659–5672, 2007.

Arthur Richards and Jonathan How. A decentralized algorithm for robust constrained model predictive control. In *Proceedings of the American Control Conference*, Boston, Massachusetts, June 2004.

C. E. Riddalls, S. Bennett, and N. S. Tipi. Modelling the dynamics of supply chains. *Int. J. Sys. Sci.*, 31:969–976, 2000.

MTM Rodrigues, L. Gimeno, CAS Passos, and MD Campos. Reactive scheduling approach for multipurpose chemical batch plants. *Comput. Chem. Eng.*, 20:S1215–S1220, 1996.

Ehap H. Sabri and Benita M. Beamon. A multi-objective approach to simultaneous strategic and operational planning in supply chain design. *Omega-Int. J. Manage. Sci.*, 28:581–598, 2000.

N.V. Sahinidis. Optimization under uncertainty: state-of-the-art and opportunities. *Comput. Chem. Eng.*, 28(6):971–983, 2004.

G. Sand and S. Engell. Modeling and solving real-time scheduling problems by stochastic integer programming. *Comput. Chem. Eng.*, 28(6):1087–1103, 2004.

K. Sari. Exploring the benefits of vendor managed inventory. *Int. J. Phy. Dist. Logist. Mangage.*, 37(7):529–545, 2007.

Haralambos Sarimveis, Panagiotis Patrinos, Chris D. Tarantilis, and Chris T. Kiranoudis. Dynamic modeling and control of supply chains: A review. *Comput. Oper. Res.*, 35:3530–3561, 2008.

Riccardo Scattolini. Architectures for distributed and hierarchical model predictive control - a review. *J. Proc. Cont.*, 19(5):723–731, May 2009. ISSN 0959-1524.

H. Scheu and W. Marquardt. Sensitivity-based coordination in distributed model predictive control. *J. Proc. Cont.*, 21(5):715–728, 2011.

D. Schmeidler. The nucleolus of a characteristic function game. *SIAM J. Appl. Math.*, 17(6): 1163–1170, 1969.

P. Seferlis and N. F. Giannelos. A two-layered optimization-based control strategy for multi-echelon supply chain networks. *Comput. Chem. Eng.*, 28:1121–1129, 2004.

S. Sethi and F. Cheng. Optimality of (s, S) policies in inventory models with markovian demand. *Oper. Res.*, 45(6):931–939, 1997.

N Shah, C. C. Pantelides, and R. Sargent. A general algorithm for short term scheduling of batch operations–II. Computational issues. *Comput. Chem. Eng.*, 20:229–244, 1993.

Nilay Shah. Process industry supply chains: Advances and challenges. *Comput. Chem. Eng.*, 29: 1225–1235, April 2005.

K. H. Shang and J. S. Song. Newsvendor bounds and heuristic for optimal policies in serial supply chains. *Manage Sci.*, 49:618–638, 2003.

J. F. Shapiro. Challenges of strategic supply chain planning and modeling. *Comput. Chem. Eng.*, 28(6-7):855–861, 2004.

L. S. Shapley. A value for n-person games. *Classics in game theory*, page 69, 1997.

S.F. Smith. Reactive scheduling systems. *Intelligent scheduling systems*, pages 155–192, 1995.

J. S. Song and P. Zipkin. Inventory control in a fluctuating demand environment. *Oper. Res.*, 43: 351–370, 1993.

H. Stadtler. Supply chain management and advanced planning–basics, overview and challenges. *Eur. J. Oper. Res.*, 163(3):575–588, 2005.

Brett T. Stewart, Aswin N. Venkat, James B. Rawlings, Stephen J. Wright, and Gabriele Pannocchia. Cooperative distributed model predictive control. *Sys. Cont. Let.*, 59:460–469, 2010.

Brett T. Stewart, Stephen J. Wright, and James B. Rawlings. Cooperative distributed model predictive control for nonlinear systems. *J. Proc. Cont.*, 21:698–704, 2011.

Kaushik Subramanian, Christos T. Maravelias, and James B. Rawlings. A state-space model for chemical production scheduling. Proceedings of Foundations of Computer-AidedProcess Operations (FOCAPO) 2012 and Chemical Process Control (CPC) VIII, Savannah, GA, 2012a.

Kaushik Subramanian, James B. Rawlings, and Christos T. Maravelias. Integration of control theory and scheduling methods for supply chain management. Proceedings of Foundations of Computer-AidedProcess Operations (FOCAPO) 2012 and Chemical Process Control (CPC) VIII, Savannah, GA, 2012b.

A. Sundaramoorthy and C.T. Maravelias. A general framework for process scheduling. *AIChE J.*, 57(3):695–710, 2010.

C. Sung and C. T. Maravelias. A projection-based method for production planning of multiproduct facilities. *AIChE J.*, 55(10):2614–2630, 2009.

Charles Sung and Christos T. Maravelias. An attainable region approach for production planning of multiproduct proceses. *AIChE J.*, 53:1298–1315, 2007.

D. R. Towill. Dynamic analysis of an inventory and order based production control system. *Int. J. Prod. Res.*, 6:671–687, 1982.

P. Trodden and A. Richards. Robust distributed model predictive control using tubes. In *American Control Conference, 2006*, pages 6–pp. IEEE, 2006.

P. Trodden and A. Richards. Robust distributed model predictive control with cooperation. In *Proceedings of the European Control Conference, 2007*, pages 2172–2178, 2007.

P. Tsiakis, N. Shah, and C. C. Pantelides. Design of multi-echelon supply chain networks under demand uncertainty. *Ind. Eng. Chem. Res.*, 40:3585–3604, 2001.

S.A. van den Heever and I.E. Grossmann. A strategy for the integration of production planning and reactive scheduling in the optimization of a hydrogen supply network. *Comput. Chem. Eng.*, 27(12):1813–1839, 2003.

A. Veinott. On the optimality of (s,S) inventory policies: New conditions and a new proof. *SIAM J. Appl. Math.*, 14:1067–1083, 1996.

Aswin N. Venkat. *Distributed Model Predictive Control: Theory and Applications.* PhD thesis, University of Wisconsin–Madison, October 2006. URL `http://jbrwww.che.wisc.edu/theses/venkat.pdf`.

J. Venkateswaran and Y. J. Son. Information synchronization effects on the stability of collaborative supply chain. In *Simulation Conference Proceedings, 2005. Winter*, pages 1668–1676, 2005.

P.M. Verderame and C.A. Floudas. Operational planning framework for multisite production and distribution networks. *Comput. Chem. Eng.*, 33(5):1036–1050, 2009.

P.M. Verderame, J.A. Elia, J. Li, and C.A. Floudas. Planning and scheduling under uncertainty: A review across multiple sectors. *Ind. Eng. Chem. Res.*, 49(9):3993–4017, 2010.

Jeetmanyu P. Vin and Marianthi G. Ierapetritou. A new approach for efficient rescheduling of multiproduct batch plants. *Ind. Eng. Chem. Res.*, 39(11):4228–4238, 2000. doi: 10.1021/ie000233z. URL `http://pubs.acs.org/doi/abs/10.1021/ie000233z`.

Y. Wakasa, M. Arakawa, K. Tanaka, and T. Akashi. Decentralized model predictive control via dual decomposition. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 381–386. IEEE, 2008.

AS White. Management of inventory using control theory. *Int. J. Tech. Manage.*, 17(7):847–860, 1999.

J. Wikner, D. R. Towill, and M. Naim. Smoothing supply chain dynamics. *Int. J. Prod. Res.*, 22(3): 231–248, 1991.

J. Wikner, M. Naim, and D. R. Towill. The system simplification approach in understanding the dynamic behaviour of a manufacturing supply chain. *J. Sys. Eng.*, 2:164–178, 1992.

Fengqi You and Ignacio E. Grossmann. Design of responsive supply chains under demand uncertainty. *Comput. Chem. Eng.*, 32:3090–3111, 2008.

P. H. Zipkin. *Foundations of Inventory Management*. McGraw-Hill Boston, MA, 2000.

INTEGRATION OF CONTROL THEORY AND SCHEDULING METHODS FOR SUPPLY CHAIN MANAGEMENT

Kaushik Subramanian

Under the supervision of Professors James B. Rawlings & Christos T. Maravelias
At the University of Wisconsin-Madison

A supply chain is a network of facilities and distribution options that performs the functions of procuring raw materials, transforming them to products and distributing the finished products to the customers. The modern supply chain is a highly interconnected network of facilities that are spread over multiple locations and handle multiple products. In a highly competitive global environment, optimal day-to-day operations of supply chains is essential.

To facilitate optimal operations in supply chains, we propose the use of Model Predictive Control (MPC) for supply chains. We develop:

- A new cooperative MPC algorithm that can stabilize any centralized stabilizable system
- A new algorithm for robust cooperative MPC
- A state space model for the chemical production scheduling problem

We use the new tools and algorithms to design model predictive controllers for supply chain models. We demonstrate:

- **Cooperative control for supply chains**: In cooperative MPC, each node makes its decisions by considering the effects of their decisions on the entire supply chain. We show that the cooperative controller can perform better than the noncooperative and decentralized controller and can reduce the bullwhip effect in the supply chain.
- **Centralized economic control**: We propose a new multiobjective stage cost that captures both the economics and risk at a node, using a weighted sum of an economic stage cost and a tracking stage cost. We use Economic MPC theory (Amrit et al., 2011) to design closed-loop stable controllers for the supply chain.

- **Integrated supply chain**: We show an example of integrating inventory control with production scheduling using the tools developed in this thesis. We develop simple terminal conditions to show recursive feasibility of such integrated control schemes.

James B. Rawlings & Christos T. Maravelias