Computers and Chemical Engineering 159 (2022) 107678

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

The inherent robustness of closed-loop scheduling

Robert D. McAllister[a,*], James B. Rawlings[a], Christos T. Maravelias[b]

[a] Department of Chemical Engineering, University of California, Santa Barbara, United States
[b] Department of Chemical and Biological Engineering, Princeton University, United States

ARTICLE INFO

Article history:
Received 20 April 2021
Revised 27 October 2021
Accepted 16 January 2022
Available online 20 January 2022

Keywords:
Scheduling
Robustness
Model predictive control
Closed-loop properties

ABSTRACT

In closed-loop or online scheduling the realization of uncertainty in plant operations is addressed in real-time though consistent and frequent reoptimization and rescheduling. Although some simulation studies have explored the robustness of closed-loop scheduling for specific case studies, there are no theoretical results addressing the robustness of closed-loop scheduling. In this paper, we present and justify an appropriate definition of robustness for closed-loop scheduling subject to large and infrequent disturbances such as breakdowns and delays. Assuming that a reasonable reference trajectory for the nominal system is available, we construct a novel terminal constraint and corresponding terminal cost for a general production scheduling problem. Through appropriate assumptions we establish that the proposed closed-loop scheduling algorithm is inherently robust to large, infrequent disturbances. We conclude with an example to illustrate the implications of this analysis. For this example, the proposed algorithm outperforms a typical online scheduling algorithm.

1. Introduction

Generation and execution of high-quality schedules is a crucial activity for a competitive manufacturing facility (Maravelias, 2012; Harjunkoski et al., 2014). Over the past few decades, optimization, specifically mixed-integer linear programming (MILP), has found a significant role in generating these high-quality schedules (Harjunkoski et al., 2014). Through representations such as the state-task network (STN) or resource-task network (RTN), a scheduling problem is formulated as an optimization of some performance metric of a manufacturing facility (e.g., profit) subject to the production constraints of the facility (e.g., processing times) (Kondili et al., 1993; Pantelides, 1994).

Improved modeling and optimization capabilities have received considerable attention, but generating a single schedule is not sufficient in practice. Demand variations, delays, and breakdowns ensure that the optimal schedule at one time is suboptimal or infeasible a short time later. Robust and stochastic optimization methods attempt to address this issue by generating a single schedule that accounts for uncertainty a priori (Balasubramanian and Grossmann, 2004; Bonfill et al., 2004; Lappas and Gounaris, 2016; Lappas et al., 2019; Li and Ierapetritou, 2008c; Lin et al., 2004; Sand and Engell, 2004; Shi and You, 2016; Vin and Ierapetritou, 2001). Lappas and Gounaris (2016), in particular, address the problem of uncertainty in processing times, in addition to demand variations. While accounting for uncertainty a priori is a valid approach, characterizing this uncertainty is difficult and often robust schedules are overly conservative leading to loss of nominal performance (Harjunkoski et al., 2014; Li and Ierapetritou, 2008a). Furthermore, sufficiently large endogenous disturbances, e.g., processing delays and unit breakdowns, may render even robust schedules infeasible.

An alternative approach to handling uncertainty is to react in real time to the realization of uncertainty, i.e., a *feedback* method. Often referred to as reactive or online (re)scheduling, this approach includes heuristic, optimization-based, and hybrid methods that generate a new, feasible, and, if optimization is employed, superior schedule whenever disturbances occur (Cott and Macchietto, 1989; Huercio et al., 1995; Elkamel and Mohindra, 1999; Ferrer-Nadal et al., 2007; Janak et al., 2006; Vin and Ierapetritou, 2000; Mendez and Cerdá, 2004; Chu and You, 2014; Novas and Henning, 2010; Lappas and Gounaris, 2016; Cui and Engell, 2010; Kopanos and Pistikopoulos, 2014; Li and Ierapetritou, 2008b). A natural extension of these reactive methods is to reschedule at fixed sampling intervals, regardless of whether a disturbance occurs (Subramanian et al., 2012; Gupta and Maravelias, 2016; Gupta et al., 2016). While this extension is often labeled *online* or *rolling horizon* scheduling, we employ the title *closed-loop* scheduling in this work to emphasize the parallels between closed-loop scheduling and feedback, or closed-loop, control. By reoptimizing at every time step, we gain the desirable features of feedback methods, but we also inherit

* Corresponding author.
E-mail address: rdmcallister@ucsb.edu (R.D. McAllister).

https://doi.org/10.1016/j.compchemeng.2022.107678
0098-1354/

their complexity. Methods to avoid scheduling nervousness associated with closed-loop scheduling have been proposed (McAllister et al., 2020; Dalle Ave et al., 2019; Lee et al., 2020). However, naive reoptimization at every sample time can lead to myopic behavior and poor performance even in the nominal case, i.e., no disturbances and the system model is accurate (Subramanian et al., 2012; Gupta and Maravelias, 2016; Risbeck et al., 2019).

To analyze the dynamic behavior of closed-loop scheduling, Subramanian et al. (2012) demonstrate that typical scheduling problems can be converted to dynamic state-space models. This state-space formulation fits within the framework of model predictive control (MPC). MPC uses this state-space model to predict the future states of the system and solves for the optimal trajectory of control actions (i.e., a schedule). By casting the closed-loop scheduling problem in the MPC framework, we can utilize the theoretical results associated with this control technique. These theoretical results address systems with discrete-valued constraints on inputs and economic cost functions, both essential to closed-loop scheduling (Rawlings and Risbeck, 2017; Risbeck and Rawlings, 2019; Amrit et al., 2011; Angeli et al., 2012; Ellis et al., 2014; Risbeck et al., 2019). A particularly useful theoretical result from economic MPC literature is the nominal performance guarantee afforded by terminal equality constraints. By constraining the individual open-loop optimizations to terminate along a reference trajectory, we can guarantee that the nominal closed-loop cost is no worse than that of the reference trajectory (Subramanian et al., 2012; Risbeck et al., 2019; McAllister et al., 2020).

While nominal guarantees are essential to guard against particularly poor closed-loop performance, the main purpose of closed-loop scheduling, and feedback methods in general, is to quickly respond to disturbances. Thus, the impact of uncertainty on the performance, i.e., robustness, of closed-loop scheduling is of particular importance. Gupta and Maravelias (2020) provide a detailed framework to study the impacts of uncertainty on production scheduling for individual facilities. Using this framework, the authors are able to draw useful insights for the design of closed-loop scheduling algorithms for specific case studies.

In contrast to quantitative and empirical robustness results for specific case studies, this work focuses on defining and establishing robustness properties for general closed-loop scheduling problems. The goal is to guarantee, under a set of reasonable assumptions, that a general closed-loop scheduling algorithm is inherently robust to some nonzero amount of uncertainty/disturbances. A system is deemed robust if arbitrarily “small” disturbances cannot cause large decreases in performance. The term *inherent* robustness refers to the robustness afforded by feedback and without considering uncertainty a priori through robust or stochastic optimization techniques.

For tracking MPC, inherent robustness is characterized by robust asymptotic stability of the closed-loop system subject to small, persistent disturbances (Grimm et al., 2004; Pannocchia et al., 2011; Allan et al., 2017). For closed-loop scheduling, however, the metric of interest is economic performance instead of stability. Furthermore, the class of disturbances most relevant to production scheduling are large and infrequent, such as breakdowns and delays, instead of small and persistent, like measurement noise. Thus, we use the results in Mcallister and Rawlings (2021) that establish, under suitable assumptions, the inherent robustness of economic MPC subject to large, infrequent disturbances.

We summarize the subsequent sections as follows. In Section 2, we define a general scheduling problem, establish that this problem can be cast as an economic MPC problem, and review the nominal properties of economic MPC subject to specific assumptions. In Section 3, we provide a motivating example, define economic robustness to large, infrequent disturbances, and establish that, under a few additional assumptions, economic MPC satisfies this definition of robustness. In Section 4, we construct a terminal constraint and cost for our general closed-loop scheduling formulation using a suitable reference trajectory and establish, under reasonable assumptions, that the closed-loop system is economically robust to large, infrequent disturbances. We conclude with an example in Section 5 that demonstrates the implications of this analysis and compares the performance of the proposed algorithm with that of a typical online scheduling algorithm.

Notation

Let $\mathbb{I}$ denote integers and $\mathbb{R}$ denote reals. Let superscripts on these sets denote dimension and subscripts on these sets denote restrictions (e.g. $\mathbb{R}^n_{\geq 0}$ for nonnegative reals of dimension n). The set $\mathbb{T} \subseteq \mathbb{I}_{\geq 0}$ denotes discrete time points. The function $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is of class $\mathcal{K}$ if it is continuous, strictly increasing, and $\alpha(0) = 0$. We use $|\cdot|$ to denote absolute value when applied to a scalar, the Euclidean norm when applied to a vector, and the induced 2-norm when applied to a matrix. Therefore, we have $|Ax| \leq |A||x|$ in which A is a matrix and x is a vector. We use A' to denote the transpose of the matrix/vector A. Sequences are denoted in bold face. We use $\Pr(A)$ to denote the probability of event A and $\mathbb{E}[\cdot]$ to denote expected value of a random variable (or function of a random variable).

2. Closed-loop scheduling and MPC

2.1. The general scheduling problem

We define the chemical production scheduling problem for a discrete-time grid and batch processes as follows. Given (i) production facility data, (ii) production costs, (iii) material and resource availability, and (iv) production targets or orders, determine the optimal assignment of tasks and batch sizes to units at each time, i.e., a schedule. In this work, we define the minimum cost (max profit) as optimal. To represent the manufacturing facility we use the state task network (STN) representation (Kondili et al., 1993).

In the STN representation, the facility consists of tasks $i \in \mathbf{I}$, units $j \in \mathbf{J}$, and materials $k \in \mathbf{K}$. The subset of tasks i that can be carried out on unit j are denoted by $\mathbf{I}_j$. We denote the subset of materials that are considered products for the facility as $\mathbf{K}^P \subseteq \mathbf{K}$, i.e., the materials for which we have demand or that can be sold for profit. Similarly, we denote the subset of intermediate and feedstock materials as $\mathbf{K}^I = \mathbf{K} \setminus \mathbf{K}^P$. The parameters $\rho_{ik}/\bar{\rho}_{ik}$ define the mass fraction of material k produced (> 0 for produced, < 0 for consumed) by starting/completing, respectively, task i relative to the batch size of the task. The parameters τ_{ij}, $\beta^{\min}_{ij}$, and $\beta^{\max}_{ij}$ denote the processing time, minimum batch size, and maximum batch size for task i carried out on unit j. We also require that inventory of each material not exceed a maximum inventory capacity denoted by $\tilde{\psi}_k$ for each $k \in \mathbf{K}^I$ and ψ_k for each $k \in \mathbf{K}^P$.

The fixed and proportional production costs of carrying out task i on unit j are α^F_{ij} and α^P_{ij}, respectively. We consider both incoming deliveries, $\zeta_k(t)$, for each intermediate and feedstock $k \in \mathbf{K}^I$ and outgoing demand, $\xi_k(t)$, for each product $k \in \mathbf{K}^P$ at certain times t. The sales price of material $k \in \mathbf{K}$ is π_k. The inventory cost and backlog cost for product material $k \in \mathbf{K}^P$ are π^S_k and π^U_k, respectively. We assume that there are no inventory costs for intermediate and feedstock materials.

We use this STN representation to construct a state-space scheduling model similar to Subramanian et al. (2012) and Gupta and Maravelias (2016). We define the binary decision variable $W_{ij}(t)$ to be unity if task i is to start on unit j at the current time t. We also define the continuous decision variable $B_{ij}(t)$ to denote the batch size of task i on unit j at time t. To track these decisions in the state of the system we "lift" W and B with the state variable $\bar{W}_{ij}^n(t)$ and $\bar{B}_{ij}^n(t)$ for all $n \in \{0, \dots, \tau_{ij}\}$. The index n is the progress status of the task (e.g. if $\bar{W}_{ij}^n(t) = 1$, task i on unit j is n/τ_{ij} complete at time t). The state variables $\tilde{S}_k(t)$ are the inventory levels of intermediate and feedstock materials $k \in \mathbf{K}^I$ at time t. The state variables $S_k(t)$, $U_k(t)$ are the inventory and backlog (unmet demand) levels, respectively, of product material $k \in \mathbf{K}^P$ at time t. The decision variable $H_k(t)$ is the amount of material k shipped to meet demand. We assume that there is a large, but finite, upper bound on amount of material k that can be shipped to meet demand, i.e., we define $\eta_k \in \mathbb{R}_{\geq 0}$ and require $H_k(t) \leq \eta_k$.

We also add an option for the scheduler to "hold" materials in a processing unit after a task is complete. Specifically, for each task $i \in \mathbf{I}_j$, we may define a new task $i' \in \mathbf{I}_j$ that holds material in the processing unit after task i is complete. We denote the subset of hold tasks as $\mathbf{I}_j^h \subseteq \mathbf{I}_j$ and we denote the mapping of each hold task $i' \in \mathbf{I}_j^h$ to its corresponding production task $i \in \mathbf{I}_j \setminus \mathbf{I}_j^h$ as $h(i') = i$. In the scheduling model, the hold task i' consumes and produces the same set of materials produced by task $i = h(i')$, has a processing time of 1 time step, and can only be run after task i is completed or after a previous hold task i' is completed. While this option was originally proposed decades ago by Kondili et al. (1993), the importance of this action in terms of online scheduling seems underappreciated. In fact, including the option to hold materials (or some similar action) is often *necessary* to ensure that the closed-loop scheduling algorithm (or any scheduling algorithm for that matter) can satisfy maximum inventory constraints for intermediates when subject to disturbances such as delays and demand variations. Similarly, Avadiappan and Maravelias (2021) propose adding delays as optimization variables to ensure feasibility.

The decision variable $V_k(t)$ is the amount of material purchased/sold in excess of demand (> 0 for purchased, < 0 for sold). The parameters ν_k^P/ν_k^S are the maximum amount of each material that can be purchased or sold in excess of demand, respectively. Note that in some facilities, all materials have $\nu_k^P = \nu_k^S = 0$ and therefore, $V_k = 0$ and is ignored. We use ν_k^S to distinguish between cost minimization and profit maximization problems. In the former, $\nu_k^S = 0$ for all $k \in \mathbf{K}$ and we seek only to meet demand at minimum cost. In the latter, $\nu_k^S > 0$ for some $k \in \mathbf{K}$ and the minimum cost optimization problem is sometimes referred to as profit maximization. Either way, the scheduling problem may be written using the general structure presented in this section.

In addition to these standard decision variables, we also allow inventory (material) disposal at each time step for product materials. We define the nonnegative input variable D_k to be the amount of inventory disposed for each material $k \in \mathbf{K}^P$. We also specify upper bounds for inventory disposal for each material $k \in \mathbf{K}^P$ as μ_k. We apply significant cost for this action (π_k^D) to discourage the optimizer from using this action. However, the additional flexibility provided by these variables, as discussed in subsequent sections, is essential to construct a closed-loop scheduling algorithm that is inherently robust for a general class of scheduling problems.

Remark 1. In practice, inventory "disposal" can be treated in many ways aside from the usual interpretation (e.g., waste). For example, we can ship the material to a long-term storage facility or move the material to a separate storage unit on-site. The only requirement is that this disposal action removes the material from the inventory state and therefore removes it from the cost function.

We consider disturbances in this scheduling model. We define the binary variable $Y_j(t)$ to be unity if unit j experiences a delay of one sample time during $[t, t+1)$. We also define the binary variable $Z_j(t)$ to be unity if unit j experiences a breakdown during $[t, t+1)$ and remains down until $t+1$. Next, we define a fractional yield loss of material on unit j at time t as $L_j(t)$ which takes values in [0,1] (e.g. $L_j = 0.25$ is a 25% loss of material on unit j). Since individual units are not permitted to run more than one task at a time, we can specify our disturbance variables by only the unit affected.

With these variable and parameter definitions we construct the following discrete-time dynamic model. To streamline notation, we note that all variables on the right-hand side of the equality statements are at time t and the left-hand side at time $t+1$ denoted by $^+$. For all $i \in \mathbf{I}_j$, $j \in \mathbf{J}$ such that $\tau_{ij} \geq 2$ we have

$$\begin{aligned}
(\bar{W}_{ij}^0)^+ &= (\bar{W}_{ij}^0 + W_{ij})Y_j(1 - Z_j) \\
(\bar{W}_{ij}^1)^+ &= \left((\bar{W}_{ij}^0 + W_{ij})(1 - Y_j) + \bar{W}_{ij}^1 Y_j\right)(1 - Z_j) \\
(\bar{W}_{ij}^{\tau_{ij}})^+ &= \bar{W}_{ij}^{\tau_{ij}-1}(1 - Y_j)(1 - Z_j) \\
(\bar{B}_{ij}^0)^+ &= (\bar{B}_{ij}^0 + B_{ij})Y_j(1 - Z_j)(1 - L_j) \\
(\bar{B}_{ij}^1)^+ &= \left((\bar{B}_{ij}^0 + B_{ij})(1 - Y_j) + \bar{B}_{ij}^1 Y_j\right)(1 - Z_j)(1 - L_j) \\
(\bar{B}_{ij}^{\tau_{ij}})^+ &= \bar{B}_{ij}^{\tau_{ij}-1}(1 - Y_j)(1 - Z_j)(1 - L_j)
\end{aligned}$$

and

$$\begin{aligned}
(\bar{W}_{ij}^n)^+ &= (W_{ij}^{n-1}(1 - Y_j) + W_{ij}^n Y_j)(1 - Z_j) \\
(\bar{B}_{ij}^n)^+ &= (B_{ij}^{n-1}(1 - Y_j) + B_{ij}^n Y_j)(1 - Z_j)(1 - L_j)
\end{aligned}$$

for all $n \in \{1, \dots, \tau_{ij} - 1\}$.[1]

We note that the "variables" Y_j, Z_j, L_j enter the model as bilinear terms. These variables, however, are actually parameters (if they are considered at all) in the deterministic optimization problem solved to generate a schedule. With this formulation, we need to specify only the unit that is delayed/broken and the model representation enforces realistic behavior of all the lifted variables.

For intermediate and feedstock materials ($k \in \mathbf{K}^I$), we model the discrete-time evolution of inventory as

$$\tilde{S}_k^+ = \tilde{S}_k + \sum_{j \in \mathbf{J}} \sum_{i \in \mathbf{I}_j} \left(\bar{\rho}_{ik}\bar{B}_{ij}^{\tau_{ij}} + \rho_{ik}B_{ij}\right) + V_k + \zeta_k$$

For products ($k \in \mathbf{K}^P$), we model the discrete-time evolution of inventory and backlog as follows.

$$S_k^+ = S_k + \sum_{j \in \mathbf{J}} \sum_{i \in \mathbf{I}_j} \left(\bar{\rho}_{ik}\bar{B}_{ij}^{\tau_{ij}} + \rho_{ik}B_{ij}\right) + V_k - H_k - D_k$$

$$U_k^+ = U_k - H_k + \xi_k$$

Subsequently, we employ the methods discussed in Subramanian et al. (2012) (and further refined in Gupta and Maravelias (2020)), to represent this scheduling model as a discrete-time state-space model with associated constraints. To streamline notation, we begin by defining each variable without subscripts to indicate a column vector containing the variable at each subscript, e.g.,

$$\bar{W} := \left[(\bar{W}_{ij}^n \ \forall j \in \mathbf{J}, i \in \mathbf{I}_j, n \in \mathbb{I}_{[0,\tau_{ij}]})\right]'$$

$$S := \left[(S_k \ \forall k \in \mathbf{K}^P)\right]'$$

We define the state, input (decisions), and disturbance for the system, respectively, as

$$x = \begin{bmatrix} \bar{W} \\ \bar{B} \\ \tilde{S} \\ S \\ U \end{bmatrix} \quad u = \begin{bmatrix} W \\ B \\ V \\ H \\ D \end{bmatrix} \quad w = \begin{bmatrix} Y \\ Z \\ L \end{bmatrix}$$

[1] If instead $\tau_{ij} = 1$, we must slightly modify these equations.

Using these variables we can represent the dynamic evolution of the system as

$$x^+ = f(x, u, w, t) \tag{1}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $w \in \mathbb{R}^p$, and $t \in \mathbb{T}$. Note that for $w = 0$, we have an linear affine description of the system,

$$f(x, u, 0, t) = Ax + Bu + c(t)$$

in which $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $c(t) \in \mathbb{R}^n$.

In addition to dynamic equations, we impose the following constraints on the state and decision variables at each time step to enforce one-task-per-unit and batch size requirements.

$$\sum_{i\in \mathbf{I}_j} \sum_{n=0}^{\tau_{ij}} \bar{W}_{ij}^n \leq 1 \quad \forall\ j \in \mathbf{J} \tag{2}$$

$$\beta_{ij}^{min} W_{ij} \leq B_{ij} \leq \beta_{ij}^{max} W_{ij} \quad \forall\ i \in \mathbf{I}_j,\ j \in \mathbf{J} \tag{3}$$

We also require that the hold tasks are only available after the appropriate production task is complete.

$$W_{ij} \leq \bar{W}_{h(i)j}^{\tau_{h(i)j}} + \bar{W}_{ij}^1 \quad \forall i \in \mathbf{I}_j^h \tag{4}$$

We enforce the appropriate variable ranges as follows.

$$\begin{aligned} & W_{ij}, \bar{W}_{ij}^n, X_j \in \{0, 1\} \quad B_{ij}, \bar{B}_{ij}^n \in [0, \beta_{ij}^{max}] \quad \forall\ i \in \mathbf{I}_j,\ j \in \mathbf{J} \\ & \tilde{S}_k \in [0, \tilde{\psi}_k] \quad \forall\ k \in \mathbf{K}^I \\ & S_k \in [0, \psi_k], U_k \geq 0 \quad \forall\ k \in \mathbf{K}^P \\ & -\nu_k^S \leq V_k \leq \nu_k^P \quad 0 \leq H_k \leq \eta_k \quad \forall\ k \in \mathbf{K} \\ & 0 \leq D_k \leq \mu_k \quad \forall\ k \in \mathbf{K} \end{aligned} \tag{5}$$

We denote the state and inputs constraints in (2), (3), and (5) using the sets $\mathbb{X}$ and $\mathbb{U}$ and require that $x \in \mathbb{X}$ and $u \in \mathbb{U}$. For the constraint in (4), we require a more general mixed constraint on the state *and* input. We combine all these constraints into the set $\mathbb{Z} \subseteq \mathbb{X} \times \mathbb{U}$ and require that $(x, u) \in \mathbb{Z}$.

Note that $\mathbb{Z}$ is closed and $\mathbb{U}$ is compact, i.e., closed and bounded. The disturbances may only take values in the compact set,

$$\mathbb{W} := \{w \in \mathbb{R}^p_{[0,1]} \mid Y_j, Z_j \in \{0, 1\}\ \forall j \in \mathbf{J}\}$$

Next, we define the stage cost as

$$\ell(x, u, t) := q'x + r'u \tag{6}$$

for $q \in \mathbb{R}^n$ and $r \in \mathbb{R}^m$. Specifically,

$$q' = \begin{bmatrix} 0 & 0 & 0 & (\pi^S)' & (\pi^U)' \end{bmatrix}$$

$$r' = \begin{bmatrix} (\alpha^F)' & (\alpha^P)' & \pi' & 0 & (\pi^D)' \end{bmatrix}$$

With this notation, we may write the schedule optimization problem starting from state x at time t as follows.

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u}} \ & \sum_{k=t}^{t+N-1} \ell(x(k), u(k), k) \\ \text{s.t.} \ & x(k+1) = f(x(k), u(k), 0, k) && \forall\ k \in \mathbb{I}_{[t,t+N-1]} \\ & (x(k), u(k)) \in \mathbb{Z} && \forall\ k \in \mathbb{I}_{[t,t+N-1]} \\ & x(t) = x,\ \ x(t+N) \in \mathbb{X} \end{aligned} \tag{7}$$

We note that this optimization problem is typical of any online scheduling problem, i.e., find the minimum cost subject to constraints on the facility and nominal operation.

2.2. Model predictive control

Next, we introduce the framework of MPC. We consider discrete, time-varying systems of the form

$$x^+ = f(x, u, w, t)$$

defined for the state $x \in \mathbb{X} \subseteq \mathbb{R}^n$, input $u \in \mathbb{U} \subseteq \mathbb{R}^m$, and disturbance $w \in \mathbb{W} \subseteq \mathbb{R}^p$ at the discrete time index $t \in \mathbb{T}$. The successor state at time $t+1$ is denoted x^+. Note that the constraints $\mathbb{X}$, $\mathbb{U}$, and $\mathbb{Z}$ may enforce integer-valued constraints on the state and input.

The nominal system is described by

$$x^+ = f(x, u, 0, t) \tag{8}$$

For the current state x and input sequence $\mathbf{u}$ at time t, the function $\hat{\phi}(k; x, \mathbf{u}, t)$ denotes the open-loop state solution to the nominal system (8) at time $k \in \mathbb{I}_{\geq t}$. We consider an MPC problem with a horizon $N \in \mathbb{I}_{\geq 0}$, initial condition x at time t, stage cost $\ell(\cdot, k) : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$, time-varying terminal constraints $\mathbb{X}_f(k) \subseteq \mathbb{X}$, and terminal cost $V_f(\cdot, k) : \mathbb{X}_f(k) \to \mathbb{R}$ for all $k \in \mathbb{I}_{\geq t}$. We define the set of admissible inputs (9), admissible initial conditions (10), and objective function (11) by

$$\begin{aligned} \mathbb{U}_N(x) &:= \{\mathbf{u} \mid (x(k), u(k)) \in \mathbb{Z}\ \forall k \in \mathbb{I}_{[t,t+N-1]}\} \\ \mathcal{U}_N(x, t) &:= \{\mathbf{u} \in \mathbb{U}_N(x) \mid x(t+N) \in \mathbb{X}_f(t+N)\} \end{aligned} \tag{9}$$

$$\mathcal{X}_N(t) := \{x \in \mathbb{X} \mid \exists \mathbf{u} \in \mathcal{U}_N(x, t)\} \tag{10}$$

$$V_N(x, \mathbf{u}, t) := \sum_{k=t}^{t+N-1} \ell(x(k), u(k), k) + V_f(x(t+N), t+N) \tag{11}$$

in which $x(k) := \hat{\phi}(k; x, \mathbf{u}, t)$.

In some scheduling problem formulations, predictions of future disturbances are considered in the model used for schedule optimization, i.e., we have some prediction of w at time $k > t$ available at time t denoted $\hat{w}(k|t)$. Although these predictions may improve the performance of the closed-loop scheduling problem, we omit these disturbance predictions to simplify the presentation of the subsequent analysis.

The optimal control problem for $x \in \mathcal{X}_N(t)$ at time $t \in \mathbb{T}$ is defined as

$$V_N^0(x, t) := \min_{\mathbf{u}\in\mathcal{U}_N(x,t)} V_N(x, \mathbf{u}, t) \tag{12}$$

and $\mathbf{u}^0(x, t)$ is the optimal input sequence for the initial condition $x \in \mathcal{X}_N(t)$ at time t. Note that the optimization problem in (12) is equivalent to (7) if $\mathbb{X}_f(t) := \mathbb{X}$ and $V_f(\cdot) = 0$. Thus, the MPC version of the optimal scheduling problem differs slightly from the standard online scheduling problem. These terminal constraints and costs, however, are essential to provide performance and robustness guarantees.

The MPC control law $\kappa_N(x, t) := u^0(t; x, t)$ is defined as the first input in $\mathbf{u}^0(x, t)$.[2] For the controlled system, the state evolves according to

$$x^+ = f_c(x, w, t) := f(x, \kappa_N(x, t), w, t) \tag{13}$$

Note that, even if $f(\cdot)$ is continuous, $f_c(\cdot)$ may be discontinuous in x since $\kappa_N(\cdot)$ may be discontinuous. We define the solution to (13) at time $k \geq t$ given the initial condition x at time t and the disturbance sequence $\mathbf{w}_k := (w(t), \dots, w(k-1))$ as $\phi(k; x, \mathbf{w}_k, t)$. In the context of production scheduling problems, $\phi(\cdot; x, \mathbf{w}_k, t)$ (and the corresponding input trajectory) represents the closed-loop (implemented) schedule for the facility subject to disturbances.

2.3. Nominal performance

Before addressing the behavior of the closed-loop trajectory subject to disturbances, we begin with the nominal closed-loop

[2] If multiple solutions exist for the optimization problem, we assume that some selection rule is applied to ensure that $\kappa_N(\cdot)$ is a single-valued, measurable function. All subsequent results then apply for any such selection rule.

trajectory. In this case, we assume $\mathbf{w} = \mathbf{0}$, i.e., there are no disturbances and the model of the facility is accurate. To establish a performance guarantee, we require a reference trajectory for the system (8) to properly formulate the MPC problem and to benchmark the closed-loop performance of the scheduling algorithm. Let the sequence $(\mathbf{x}_r, \mathbf{u}_r)$ denote this reference trajectory and satisfy the following assumption.

Assumption 1 (Reference trajectory)**.** The reference trajectory $(\mathbf{x}_r, \mathbf{u}_r)$ satisfies

$$x_r(t+1) = f(x_r(t), u_r(t), 0, t)$$

with $(x_r(t), u_r(t)) \in \mathbb{Z}$ for all $t \in \mathbb{T}$.

Note that, for systems with regular demand, an offline optimal periodic scheduling problem generates such a trajectory. Heuristic methods may also be used to construct such a trajectory. Indeed, even an idle facility, i.e., $\mathbf{u}_r = \mathbf{0}$ and its associated state trajectory, satisfies this assumption. While any trajectory that satisfies this assumption is permitted, the economic performance of this reference trajectory is also important as it forms the benchmark for all subsequent results. Any improvement in the performance of the reference trajectory translates to a direct improvement in the performance guarantees.

Using this reference trajectory, we define the shifted stage and optimal cost functions as follows. By shifting these values we can discuss all subsequent results in terms of their deviation from the reference trajectory.

$$\bar{\ell}(x, u, t) := \ell(x, u, t) - \ell(x_r(t), u_r(t), t)$$

$$\bar{V}_N^0(x, t) := V_N^0(x, t) - \sum_{k=t}^{t+N-1} \ell(x_r(k), u_r(k), k)$$

We also require a mild regularity assumption.

Assumption 2 (Regularity conditions)**.** The functions $f(\cdot)$, $\ell(\cdot)$ are continuous and $\ell(x, u, t)$ is uniformly bounded below for $(x, u) \in \mathbb{Z}$ and $t \in \mathbb{T}$. The set $\mathbb{Z}$ is closed and $\mathbb{U}$ is compact.

Thus far, these assumptions are mild and satisfied by almost any online scheduling formulation. These assumptions alone, however, do not guarantee anything about the performance of the closed-loop scheduling problem. As demonstrated in Risbeck et al. (2019), closed-loop scheduling problems formulated without appropriate terminal costs and constraints can produce arbitrarily poor closed-loop performance even in the nominal case. Therefore, we require terminal constraints and costs that satisfy the following assumption.

Assumption 3 (Terminal conditions)**.** The set $\mathbb{X}_f(t)$ is closed for all $t \in \mathbb{T}$. The function $V_f(x, t)$ is continuous and uniformly bounded from below for all $x \in \mathbb{X}_f(t)$ and $t \in \mathbb{T}$. For each $t \in \mathbb{T}$ and $x \in \mathbb{X}_f(t)$, the set

$$\begin{aligned}\mathbb{U}_f(x, t) : = \{u \in \mathbb{U} \mid\ & (x, u) \in \mathbb{Z}, \\ & x^+ : = f(x, u, 0, t) \in \mathbb{X}_f(t+1), \\ & V_f(x^+, t+1) \leq V_f(x, t) - \bar{\ell}(x, u, t)\}\end{aligned}$$

is nonempty and $V_f(x_r(t), t) = 0$.

We also provide a definition of sequential positive invariance.

Definition 1 (Sequential positive invariance)**.** A sequence of sets $(\mathcal{X}(t))_{t\in\mathbb{T}}$ is sequentially positive invariant for the nominal system $x^+ = f_c(x, 0, t)$, if $x \in \mathcal{X}(t)$ implies $x^+ \in \mathcal{X}(t+1)$ for all $t \in \mathbb{T}$.

If the sequence $(\mathcal{X}_N(t))_{t\in\mathbb{T}}$ is sequentially positive invariant for the nominal system $x^+ = f_c(x, 0, t)$, the optimal control problem remains feasible along the nominal closed-loop trajectory. With these assumptions, we can establish the following nominal performance guarantee for closed-loop scheduling (Risbeck and Rawlings, 2019, Thm. 1).

Theorem 1. *Let Assumptions 1–3 hold. For every initial state $x \in \mathcal{X}_N(t)$ and $t \in \mathbb{T}$, the sequence of sets $(\mathcal{X}_N(k))_{k\geq t}$ is sequentially positive invariant for the system $x^+ = f_c(x, 0, k)$ and*

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{k=t}^{t+T-1} \bar{\ell}(x(k), u(k), k) \leq 0$$

in which $x(k) := \phi(k; x, \mathbf{0}, t)$ and $u(k) := \kappa_N(x(k), k)$ denote the nominal closed-loop trajectory.

Theorem 1 ensures that, after an initial transient, the average cost of the closed-loop trajectory is better than (no worse than) the reference trajectory used to construct the terminal conditions in Assumption 3.

3. Disturbances and inherent robustness

3.1. Motivating examples

We begin with a motivating example to illustrate the implications and importance of robustness for production scheduling. Consider a simple scheduling problem involving a single unit and two tasks, T1 and T2, that produce the product M1 from an abundant source of raw material. Each task requires 2 hours to complete and there is a demand of 1 kg of M1 every 2 hours. T1 produces up to 1 kg of M1 and cost \$60 to run. T2 produces up to 1.2 kg of M1, but T2 costs \$90 to run. The cost for maintaining inventory or backlog is \$1/hr/kg or \$10/hr/kg, respectively. We also allow up to 1 kg/hr of disposal for M1 at a cost of \$10/kg. We note, however, that none of the subsequent simulations use this action. Thus, the obvious and optimal periodic schedule is to run T1 at maximum capacity and in phase with demand.

If we use trivial terminal constraints/costs, i.e., $\mathbb{X}_f = \mathbb{X}$ and $V_f(x, t) = \ell(x, 0, t)$, and initialize the system in phase with the reference trajectory, we observe that the online scheduling algorithm produces a closed-loop trajectory identical to the optimal periodic schedule. Thus, the nominal performance of the closed-loop trajectory appears to be satisfactory without terminal conditions. Nominal performance however does not imply robustness.

To probe the robustness of this scheduling algorithm, we consider the possibility of a one hour delay in the completion of T1 or T2. In Fig. 1, we plot the closed-loop schedule for the single unit facility without terminal constraints. We observe that after the task delay at $t = 2$, the scheduling algorithm never chooses to run T2 in order to produce extra M1. The consequences of this choice are severe; the total cost continues to increase relative to the reference trajectory regardless of how long we continue to run the facility. Thus, the scheduling algorithm never recovers from the disturbance even though we have a clear method to do so. Note that the lack of robustness in this example is a property of the scheduling *algorithm*, not the underlying system.

Informally, robustness implies that arbitrarily small disturbances to the system do not produce arbitrarily large losses in performance. In this example, we have subjected the system to a single disturbance, i.e., "small" disturbance, and the resulting closed-loop performance degrades significantly and permanently. Thus, the algorithm and resulting closed-loop system are not robust despite acceptable performance in the nominal case.

We now address the reason for this myopic behavior. Since the cost to address the backlog by running T2 is larger than the cost to retain the backlog for 24 hours of operation, the optimal choice for this horizon is to never deal with the backlog. Without terminal costs/constraints, the open-loop optimization is unaware of the

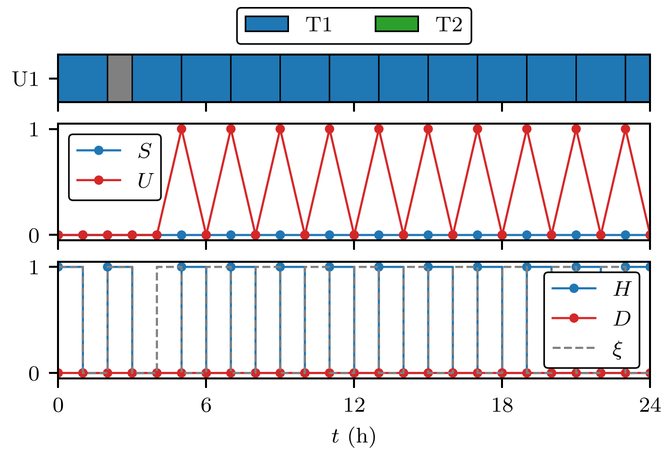

Fig. 1. Closed-loop schedule for the single unit facility with a horizon length of 24 hours, no terminal conditions, and subject to a 1 h task delay at $t = 2$. The top plot is a Gantt chart of the facility with gray blocks indicating a task delay. The middle and bottom plots show the closed-loop inventory (S), backlog (U), shipments to meet demand (H), inventory disposal (D), and demand (ξ) trajectories.

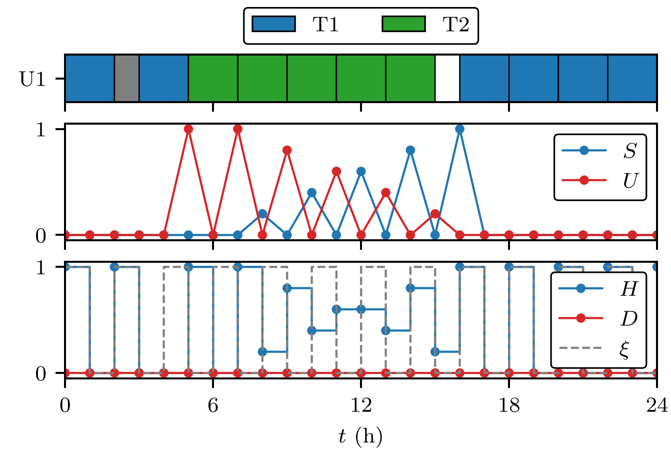

Fig. 2. Closed-loop schedule for the single unit facility with a horizon length of 8 hours, with terminal conditions, and subject to a 1 h task delay at $t = 2$. The top plot is a Gantt chart of the facility with gray blocks indicating a task delay. The middle and bottom plots show the closed-loop inventory (S), backlog (U), shipments to meet demand (H), inventory disposal (D), and demand (ξ) trajectories.

costs/consequences of any actions taken after its 24 hour horizon. In other words, the open-loop optimization problem does not accurately approximate the infinite horizon cost of the system and therefore selects a myopic control action.

One potential solution to this problem is to use a longer horizon to better approximate the infinite horizon problem. Indeed, for industrial implementations of tracking MPC this approach is often employed. However, the horizon length necessary to accurately approximate the infinite horizon cost and avoid myopic behavior is often unclear and may be prohibitively large. For the example in Fig. 1, we selected a 24 h horizon for this simple problem and still failed to properly approximate the infinite horizon cost. If we scale this to a larger facility, the horizon required to produce a robust closed-loop scheduling algorithm may be intractable for online optimization. And even with this excessive horizon length, we still cannot *guarantee* that the algorithm is robust.

We emphasize that this behavior can occur with any online or rolling horizon scheduling algorithm that reoptimizes the schedule based on feedback from the manufacturing facility, even if these algorithms are not explicitly written in state-space form. Unless properly designed, these feedback algorithms are not guaranteed to be inherently robust to disturbances. Furthermore, verifying the robustness of a scheduling algorithm for a specific application through simulation studies requires exhaustive testing of every possible combination of states and disturbances the facility may encounter.

In contrast, we propose a more direct and universal solution: use appropriate terminal constraints and costs that guarantee inherent robustness of the closed-loop scheduling algorithm. By selecting appropriate terminal constraints and costs, we guarantee that the open-loop optimization problem adequately approximates the infinite horizon cost. With these terminal conditions, we no longer require excessively long horizons to ensure nominal performance and robustness of the facility.

For the single unit example, we construct such a terminal cost and constraint based on the approaches detailed in Section 4. We select a horizon length of only *8 hours* and plot the closed-loop schedule in Fig. 2. The resulting closed-loop schedule chooses to run T2 and recovers from the disturbance. Thus, after a single disturbance the performance degrades, but eventually recovers. This behavior is an example of robustness. If we continue to run the simulations forward in time, the closed-loop schedule with terminal conditions outperforms the closed-loop schedule without terminal conditions and the gap between their performance continues to grow. We note that the implemented schedule in Fig. 2 is identical to the schedule generate by using a 26 h horizon without terminal constraints. Thus, proper terminal costs and constraints with a 8 h horizon achieve the same nominal performance and robustness that requires a 26 h horizon without terminal conditions.

3.2. Robustness to large, infrequent disturbances

We now rigorously define the term robustness for closed-loop scheduling. We begin by characterizing the types of disturbances that occur in closed-loop scheduling. In typical control problems, disturbances are small and persistent (e.g., measurement errors, small perturbations, and model inaccuracies). In production scheduling problems however, disturbances are large, often binary valued, and infrequent (e.g., task delays, breakdowns).

Even though task delays may be small and continuous-valued if modeled exactly, the discrete modeling of time maps the exact delay to an inherently binary disturbance. For example, if we assume the exact time delay is modeled as a uniform distribution on $[-1, 1]$ (with positive values indicating delay and negative values indicating early completion), then 50% of the time we experience a delay that requires rescheduling, i.e., any exact delay in (0,1] must be treated as $Y = 1$ in the scheduling model. In summary, any continuous distribution of exact time delay for a task is mapped to a binomial distribution by the discrete-time grid used in this closed-loop scheduling model.

Although production losses L are not necessarily large in terms of $|L|$, arbitrarily small losses in intermediate material production may render the remainder of the incumbent schedule infeasible (e.g., inventory of an intermediate drops below the minimum batch size for the subsequent operation). Consequently, we are unable to treat these disturbances using results from control theory for small persistent disturbances. We note that Gupta and Maravelias (2020) also treat production losses as infrequent disturbances.

Hence, we treat task delays, breakdowns, and production losses as large and infrequent disturbances. We characterize the "size" of these disturbances by the probability that they occur. Specifically, we consider that the probability a disturbance occurs, i.e., $\Pr(|w| > 0)$, is equal to some constant $\varepsilon \in (0, 1)$. The stochastic nature of these disturbances suggests a stochastic definition of robustness is appropriate.

Definition 2 (Economically robust to large, infrequent disturbances)**.** The closed-loop system $x^+ = f_c(x, w, t)$, $w \in \mathbb{W}$ in which $\varepsilon = \Pr(|w| > 0)$ is said to be economically robust to large, infrequent disturbances relative to the reference trajectory $(\mathbf{x}_r, \mathbf{u}_r)$ if there exists $\delta \in (0, 1]$ and $\gamma(\cdot) \in \mathcal{K}$ such that the closed-loop system satisfies

$$\limsup_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{k=t}^{t+T-1} \bar{\ell}(x(k), u(k), k)\right] \le \gamma(\varepsilon) \tag{14}$$

in which $x(k) := \phi(k; x, \mathbf{w}_k, t)$ and $u(k) = \kappa_N(x(k), k)$, for all $\varepsilon \in [0, \delta]$, $x \in \mathcal{X}_N(t)$, and $t \in \mathbb{T}$.

Consider the motivating example again without terminal constraints/costs. We assume that this breakdown has an arbitrarily small probability of occurring that we denote ε. But, as $T \to \infty$, this disturbance occurs at least once with probability one. Consequently, if we use a horizon $N \le 24$ without terminal conditions, the behavior in Fig. 1 eventually occurs and

$$\limsup_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{k=t}^{t+T-1} \bar{\ell}(x(k), u(k), k)\right] \ge \$5/hr$$

even as $\varepsilon \to 0$. In other words, no matter how reliable you make the unit ($\varepsilon \to 0$), the performance of the system never returns to the nominal performance of the system. The guarantee provided by (14), in contrast, ensures that as the reliability of a plant increases, i.e., $\varepsilon \to 0$, we approach the nominal performance of the system. While $\gamma(\cdot) \in \mathcal{K}$ is often too conservative to be a useful quantitative bound, the fact that such a bound exists prevents the particularly poor performance observed in the motivating example. For closed-loop scheduling applications, robustness means that infrequent disturbances do not cause arbitrarily large and permanent losses in the expected value of performance. The behavior in Fig. 2 is an example of robustness.

Next, we provide conditions that guarantee the closed-loop system is robust in this context. Since the optimal control law $\kappa_N(\cdot)$ is defined by an optimization problem, we must first ensure that the optimization problem remains feasible for the perturbed system. We define this property precisely using robust sequential positive invariance and the following assumption.

Definition 3. The sequence of sets $(\mathcal{X}(t))_{t\in\mathbb{T}}$ is robustly sequentially positive invariant for the perturbed system $x^+ = f_c(x, w, t)$, $w \in \mathbb{W}$, if $x(t) \in \mathcal{X}(t)$ implies $x^+ \in \mathcal{X}(t+1)$ for all $w \in \mathbb{W}$ and $t \in \mathbb{T}$.

Assumption 4 (Robust recursive feasibility)**.** The sequence $(\mathcal{X}_N(t))_{t\in\mathbb{T}}$ is robustly sequentially positive invariant for the system $x^+ = f_c(x, w, t)$, $w \in \mathbb{W}$, i.e., the optimal control problem is robustly recursively feasible.

We note that assuming robust recursive feasibility for MPC is sometimes inappropriate, particularly if state constraints are included in the optimization problem. However, as we subsequently discuss in Section 4.1, Assumption 4 does indeed hold for most production scheduling applications with properly formulated terminal constraints.

We also require a bound on the optimal cost increase for the system subject to any disturbance $w \in \mathbb{W}$.

Assumption 5 (Max cost increase)**.** For the perturbed system $x^+ = f_c(x, w, t)$, $w \in \mathbb{W}$, there exists finite $b_1, b_2 \in \mathbb{R}_{\ge 0}$ such that

$$\bar{V}_N^0(f_c(x, w, t), t+1) \le \bar{V}_N^0(x, t) + b_1|\ell(x, \kappa_N(x, t), t)| + b_2$$

for all $x \in \mathcal{X}_N(t)$, $w \in \mathbb{W}$, and $t \in \mathbb{T}$.

Although verifying Assumption 5 is nontrivial for a general nonlinear system, we establish in the subsequent section that closed-loop scheduling problems, with the terminal costs and constraints proposed in this work, satisfy this assumption. With these assumptions we have the following result (Mcallister and Rawlings, 2021, Thm. 6).

Theorem 2. *Let Assumptions 1–5 hold. Then the closed-loop system $x^+ = f_c(x, w, t)$, $w \in \mathbb{W}$ is economically robust to large, infrequent disturbances relative to the reference trajectory $(\mathbf{x}_r, \mathbf{u}_r)$.*

Remark 2. The value of δ in Theorem 2 is defined as $\delta < 1/(1 + b_1)$ in which b_1 is defined in Assumption 5. Thus, as $b_1 \to \infty$, $\delta \to 0$, and the system is no longer robust to large disturbances with nonzero probability of occurring. In contrast, as $b_1 \to 0$, $\delta \to 1$ and the system is robust to disturbances that occur with any probability.

4. Robustness of closed-loop scheduling

In this section, we establish that the proposed closed-loop scheduling algorithm is economically robust to large, infrequent disturbances, e.g., delays, breakdowns, and yield losses. As noted in the previous section, Assumptions 1 and 2 are easily satisfied for production scheduling applications. A reference trajectory that satisfies Assumption 1 may be generated by a periodic optimization problem for the nominal system ($w = 0$) or any heuristic scheduling method employed by a production facility. Furthermore, the model $f(\cdot)$ and stage cost $\ell(\cdot)$ defined in this work are continuous, the set $\mathbb{Z}$ is closed, $\mathbb{U}$ is compact, and $\ell(x, u, t)$ is bounded from below because $\pi^S, \pi^U \ge 0$. In the following subsections, we construct a terminal constraint, a corresponding terminal cost, and establish that Assumptions 3–5 also hold for the proposed closed-loop scheduling algorithm.

4.1. Robust recursive feasibility

To ensure that any optimization-based controller is robust, we must first ensure that the optimal control problem remains feasible for any state x that we may reach due to disturbances, i.e., the closed-loop scheduling algorithm is robustly recursively feasible. Specifically, we need to satisfy Assumption 4.

We note that Assumption 4 implicitly requires that the sequence $(\mathbb{X})_{t\in\mathbb{T}}$ is robustly sequentially positive invariant for the system $x^+ = f_c(x, w, t)$, $w \in \mathbb{W}$ as well. This assumption is reasonable for scheduling problems, but requires that state constraints are used only to enforce physical constraints on the system (e.g. nonnegative inventory, one-task-per-unit) and not desired goals (e.g. maintain a safety stock of material $k \in \mathbf{K}$). Desired goals of a scheduling problem should be addressed in the objective function, not state constraints. In general, closed-loop scheduling formulations, including the one presented in this work, ensure that $(\mathbb{X})_{t\in\mathbb{T}}$ is robustly sequentially positive invariant as this requirement is synonymous with a well-posed online scheduling problem. However, we also require terminal costs/constraints that satisfy Assumption 3 to establish both nominal and robust performance guarantees for closed-loop scheduling. These terminal constraints, if overly restrictive, can render Assumption 4 invalid.

One proposed formulation to satisfy Assumption 3 is to set $\mathbb{X}_f(t) := \{x_r(t)\}$ and $V_f(\cdot, t) = 0$ for all $t \in \mathbb{T}$. The benefits of this approach are its simplicity; we guarantee that $u_r(t) \in \mathbb{U}_f(x_r(t), t)$ through Assumption 1. The drawbacks are its restrictiveness; we significantly reduce the size of the feasible set $\mathcal{X}_N(t)$ compared to $\mathbb{X}$. Thus, this terminal equality constraint may render the optimization problem infeasible for relevant disturbances. By constructing an appropriate terminal cost, however, we can expand this terminal equality constraint to a terminal region and thereby ensure robust recursive feasibility of the optimization problem for relevant disturbances.

We begin by identifying the states for which the terminal equality constraint is too restrictive. For the subsequent discuss, the reference trajectory has the form

$$x_r(t) := \left[\bar{W}_r(t)', \bar{B}_r(t)', \tilde{S}_r(t)', S_r(t)', U_r(t)'\right]'$$

We postulate that, for sufficiently long horizons, requiring $\bar{W}_{ij}^n$ and $\bar{B}_{ij}^n$ to terminate exactly in phase with the reference trajectory does not affect the feasibility of the optimization problem. Similarly, we presume that, for sufficiently long horizons, requiring inventories $\tilde{S}_k$ and S_k to meet or exceed the inventories used in the reference trajectory does not affect the feasibility of the optimization problem. However, allowing any amount of inventory that exceeds the reference trajectory creates a problem when constructing a terminal cost for the system. Thus, we define the parameters

$$\tilde{\omega}_k = \min_{t\in\mathbb{T}} \tilde{\psi}_k - \tilde{S}_{r,k}(t)$$
$$\omega_k = \min_{t\in\mathbb{T}} \psi_k - S_{r,k}(t)$$

for all $k \in \mathbf{K}^I$ and $k \in \mathbf{K}^P$, respectively. The parameters $\tilde{\omega}_k$ and ω_k represent the maximum amount of inventory in excess of the reference trajectory that the system can support (without violating maximum inventory constraints) for all points in the reference trajectory. For the terminal constraint, we allow any amount of inventory that exceeds the reference trajectory, but does not exceed the reference trajectory by more than $\tilde{\omega}_k$ or ω_k (as appropriate).

For backlog U_k, the reference trajectory often requires $U_k = 0$. Thus, for any horizon length, there exists a sufficiently large initial backlog such that the terminal state cannot reach the reference trajectory. We therefore must expand the terminal region to include all values of backlog exceeding the backlog of the reference trajectory.

Based on these observations, we define our expanded terminal region as

$$\begin{aligned}\mathbb{X}_f(t) := \{x \in \mathbb{X} \mid\ & \bar{W} = \bar{W}_r(t), \bar{B} = \bar{B}_r(t)\\ & 0 \le \tilde{S} - \tilde{S}_r(t) \le \tilde{\omega}\\ & 0 \le S - S_r(t) \le \omega,\ U \ge U_r(t)\}\end{aligned} \tag{15}$$

in which '≥' denotes an element-wise comparison. Compared to the exact terminal constraints discussed previously, the terminal region in (15) ensures that Assumption 4 is satisfied for a much larger class of scheduling problems. In practice, verifying Assumption 4 is difficult. However, the expanded terminal constraint presented here and a reasonable horizon length are sufficient to satisfy Assumption 4 for most scheduling problems.

Remark 3. If we remove maximum inventory constraints, we can verify that Assumption 4 holds through a testable condition. If we do not include any intermediate or feedstock materials in the problem formulation (e.g., because we have no intermediates and the feedstock is assumed to be in abundant supply), we can also verify that Assumption 4 holds through a testable condition.

4.2. Constructing a terminal cost

Next, we must construct a terminal cost that satisfies Assumption 3 for this terminal constraint. For a tracking MPC formulation, approaches to expand the terminal region require us to determine a (suboptimal) control law that is stabilizing near the origin (or reference trajectory) for the nominal system. This control law is then used to construct an infinite horizon trajectory extending from any $x \in \mathbb{X}_f(t)$. The terminal cost is defined as the cost for this infinite horizon trajectory. Furthermore, we require an analytic function to evaluate the infinite horizon cost in an optimization problem. Often, these requirements restrict the terminal control law to a linear controller near the origin (reference trajectory), in regions for which the input constraints are not active.

For closed-loop scheduling problems, there are many discrete decisions, state constraints, and active input constraints along the reference trajectory. All of these features render the construction of a terminal control law difficult, if not intractable, for many scheduling formulations. To overcome this issue, we use the additional flexibility afforded by the inventory disposal action and place an additional restriction on the reference trajectory used to construct a terminal cost. This additional flexibility and modified reference trajectory allows us to construct an analytic terminal control law and terminal cost. Specifically, we use the following assumption.

Assumption 6 (Disposal and robust reference trajectory)**.** We can dispose of inventory for all products, i.e., $\mu_k > 0$ for all $k \in \mathbf{K}^P$. For each product, there exists $0 < \sigma_k \le \mu_k/2$ such that $\sigma_k \le D_{r,k}(t) \le \mu_k/2$ for all $t \in \mathbb{T}$ and $k \in \mathbf{K}^P$, i.e., the reference trajectory overproduces and disposes some of each product at every time step. Also, $H_{r,k}(t) + \sigma_k \le \eta_k$ for all $k \in \mathbf{K}^P$.

By requiring the reference trajectory to include disposal actions at every time step for all final products, the reference trajectory must *overproduce* all of the final products by some margin $\sigma > 0$. This overproduction ensures that the reference trajectory has some recourse available, i.e., we can use the excess production to pay down backlog. We also note that most previously proposed robust optimization approaches to scheduling (employed in an open-loop fashion) lead to final product overproduction (similar to Assumption 6) and, to our knowledge, do not discuss how the excess inventory is handled. In that respect, our approach is distinct in that we require overproduction only in the reference trajectory and (as we show in subsequent examples) do not necessarily overproduce or dispose of these products in the closed-loop schedule.

One method to construct a trajectory that satisfies Assumption 6 is to solve a finite horizon, periodic optimization problem with the lower bound $D_k \ge \sigma_k$ for all time steps and products $k \in \mathbf{K}^P$. Clearly, this reference trajectory is suboptimal for the nominal system compared to a periodic optimal solution that allows any $D_k \ge 0$. Nonetheless, $\sigma_k > 0$ may be chosen small so that the cost difference between the reference trajectory with and without overproduction is also small.[3] We note that the performance guarantee in Theorem 1 is now relative to this overproduction reference trajectory. However, this trajectory only serves as the *reference* for the scheduling algorithm and does not necessarily restrict the closed-loop performance. Indeed, the scheduling algorithm can, and often does, outperform the bounds in Theorems 1 and 2.

The intuition behind this overproduction is that we require some margin of robustness in the reference trajectory to construct a robust closed-loop scheduling algorithm. We note that in typical robustness analysis of tracking MPC, we require similar (if not stronger) assumptions about the reference trajectory to guarantee that the algorithm is inherently robust. Specifically, tracking MPC often requires the terminal region to be robustly positive invariant for the disturbance of interest, a stronger requirement than we impose for closed-loop scheduling (Allan et al., 2017). Assumption 6 is therefore well-motivated and less restrictive than the assumptions typically required to produce a inherently robust MPC algorithm.

[3] For example, we selected $\sigma = 0.01$ for the motivating example.

In addition to these restrictions on the reference trajectory, we require an assumption for the dynamics of the system. Note that the structure of A, B for production scheduling ensure that the following assumption holds.

Assumption 7 (Integrating dynamics)**.** *If $x - x_r(t) = [0, 0, \Delta\tilde{S}', \Delta S', \Delta U']'$ and $u - u_r(t) = [0, 0, 0, \Delta H', \Delta D']'$, then $x^+ = f(x, u, 0, t)$ satisfies*

$$x^+ - x_r(t+1) = [0, 0, \Delta\tilde{S}', (\Delta S - \Delta H - \Delta D)', (\Delta U - \Delta H)']'$$

We also require an additional assumption for inventory and backlog costs.

Assumption 8 (Inventory and backlog penalty)**.** All final products incur positive inventory and backlog cost, i.e., there exists $c_1 > 0$ such that $\pi_k^U, \pi_k^S \geq c_1$ for all $k \in \mathbf{K}^P$. All intermediate and feedstock materials do not incur inventory cost. Inventories for feedstock and intermediate materials are bounded, i.e., $\tilde{\psi}_k < \infty$ for all $k \in \mathbf{K}^I$.

With this reference trajectory, we may define the terminal control law as

$$\kappa_f(x,t) := \begin{bmatrix} W_r(t) \\ B_r(t) \\ V_r(t) \\ H_r(t) + \min\{\Delta U, \sigma\} \\ D_r(t) + \min\{\Delta S, \frac{\mu}{2}\} - \min\{\Delta U, \sigma\} \end{bmatrix} \tag{16}$$

in which $\Delta S := S - S_r(t)$ and $\Delta U := U - U_r(t)$. With these assumptions, we can guarantee that $\kappa_f(x,t)$ is a feasible input for all $x \in \mathbb{X}_f(t)$ and ensure that $x^+ \in \mathbb{X}_f(t+1)$.

Lemma 3. *Let Assumptions 1, 6, and 7 hold. If $x \in \mathbb{X}_f(t)$, then $\kappa_f(x,t) \in \mathbb{U}$ and $f(x, \kappa_f(x,t), 0, t) \in \mathbb{X}_f(t+1)$.*

Proof. By Assumption 1, $\kappa_f(x,t)$ satisfies the required input (and mixed) constraints for W, B, and V for all $x \in \mathbb{X}_f(t)$. For $x \in \mathbb{X}_f(t)$, we know that $H = H_r(t) + \min\{U - U_r(t), \sigma\} \leq H_r(t) + \sigma \leq \eta$. Furthermore, we have

$$D = D_r(t) + \min\{S - S_r(t), \mu/2\} - \min\{U - U_r(t), \sigma\} \leq \mu$$

and since $D_r(t) \geq \sigma$, we know that $D \geq 0$ as well. Thus, $(x, \kappa_f(x,t)) \in \mathbb{Z}$ for all $x \in \mathbb{X}_f$ and $t \in \mathbb{T}$.

Next, we consider $x^+ = f(x, \kappa_f(x,t), 0, t)$. From Assumption 7, we have that if $x \in \mathbb{X}_f(t)$,

$$x^+ - x_r(t+1) = [0, 0, \Delta\tilde{S}', (\Delta S - \Delta H - \Delta D)', (\Delta U - \Delta H)']'$$

in which $\Delta S \geq 0$ and $\Delta U \geq 0$. We have that $\Delta H = \min\{\Delta U, \sigma\}$ and therefore, $\Delta U - \Delta H \geq 0$. We also have that $\Delta S - \Delta H - \Delta D = \Delta S - \min\{\Delta S, \mu/2\}$ and therefore, $\Delta S - \Delta H - \Delta D \geq 0$. Thus, the values of inventory and backlog for the successor state are no less than the reference trajectory. We also have that $\Delta S \geq \Delta S - \Delta H - \Delta D$ because $\min\{\Delta S, \mu/2\} \geq 0$. Since $x \in \mathbb{X}_f(t)$, we know that $0 \leq \Delta S \leq \omega$ and $0 \leq \Delta\tilde{S} \leq \tilde{\omega}$. Therefore, the values of inventory satisfy the subsequent terminal constraint and we have that $x^+ \in \mathbb{X}_f(t+1)$. □

Using this terminal control law we construct an analytic terminal cost from the infinite horizon cost relative to the reference trajectory.

$$V_\infty^{\kappa_f}(x,t) := \sum_{k=t}^{\infty} \bar{\ell}(x(k), \kappa_f(x(k),k), k) \tag{17}$$

in which $x(k+1) = f(x(k), \kappa_f(x(k),k), 0, k)$ and $x(t) = x$. Note that $V_\infty^{\kappa_f}(\cdot)$ satisfies Assumption 3.

We begin by redefining the system of interest on the terminal set $\mathbb{X}_f(t)$ and subject to the terminal control law $\kappa_f(x,t)$. Since $\bar{W} = \bar{W}_r(t)$, $\bar{B} = \bar{B}_r(t)$, $W = W_r(t)$, $B = B_r(t)$, and $V = V_r(t)$ for all $x \in \mathbb{X}_f(t)$, $t \in \mathbb{T}$, and $u = \kappa_f(x,t)$, we can reduce the system to a lower dimension without loss of information. Furthermore, we note that by Assumption 8 the inventory of intermediate and feedstock materials does not affect the stage cost or terminal control action. We define the reduced variables

$$z := R_x(x - x_r(t)) = [(S - S_r(t))', (U - U_r(t))']'$$
$$v := R_x B(u - u_r(t))$$

for all $x \in \mathbb{X}_f(t)$ and $t \in \mathbb{T}$ using an appropriate transformation matrix R_x. Using Assumption 7 and letting $u = \kappa_f(x,t)$, we also define the reduced system dynamics,

$$z^+ = z - \min\{z, m\}$$

in which $m = [(\mu/2)', \sigma']'$ and stage cost,

$$\bar{\ell}(x,u,t) = \bar{q}'z + \bar{r}'\min\{z, m\}$$

in which $\bar{q}' = [(\pi^S)', (\pi^U)']$ and $\bar{r}' = [(\pi^D)', -(\pi^D)']$. The states and inputs are nonnegative, $z, v \geq 0$. The state variable z_i at future time $k \geq t$ is therefore

$$z_i(k) = \min\{z_i(t) - (t-k)m_i, 0\}$$

Next, we calculate the infinite horizon cost generated by a single element i of z and v, i.e., $V_{\infty,i}^{\kappa_f}(z_i(t), t)$.

$$\begin{aligned} V_{\infty,i}^{\kappa_f}(z_i(t), t) &= \sum_{k=t}^{\infty} \bar{q}_i z_i(k) + \bar{r}_i v_i(k) \\ &= \sum_{k=0}^{N(z_i(t))} \bar{q}_i(z_i(t) - km_i) + \bar{r}_i z_i(t) \\ &= \bar{q}_i(N(z_i(t)) + 1)(z_i(t) - m_i N(z_i(t))/2) + \bar{r}_i z_i(t) \end{aligned}$$

in which $N(z_i(t)) = \lfloor z_i(t)/m_i \rfloor$ and $\lfloor\cdot\rfloor$ denotes rounding down to the nearest integer. Therefore, the infinite horizon cost may be represented as a sum of this element-wise infinite horizon cost and is an analytic equation,

$$V_\infty^{\kappa_f}(x,t) = \sum_{i=1}^{\bar{n}} V_{\infty,i}^{\kappa_f}(z_i, t)$$

in which $\bar{n}$ is the number of elements in z.

The resulting cost function however is unwieldy to use for optimization. If we use the approximation $N(z_i(t)) = z_i/m_i$, then we have

$$V_{\infty,i}^{\kappa_f}(z_i, t) = \frac{\bar{q}_i}{2m_i} z_i^2 + \left(\frac{\bar{q}_i}{2} + \bar{r}_i\right) z_i$$

Note the linear-quadratic form of this terminal cost. This approximate infinite horizon cost, unfortunately, does not satisfy Assumption 3.[4] However, this analysis provides us significant insight. We require only a minor modification to the parameters elucidated by this analysis to construct the following valid terminal cost.

$$V_f(x,t) := \sum_{i=1}^{\bar{n}} \left(\frac{\bar{q}_i}{2m_i} z_i^2 + (\bar{q}_i + \bar{r}_i) z_i \right) \tag{18}$$

Proposition 4. *Let Assumptions 1, 6, 7, and 8 hold. Then Assumption 3 is satisfied for the terminal constraint (15) and terminal cost (18) defined for production scheduling.*

Proof. The sets $\mathbb{X}_f(t)$ are closed and $V_f(\cdot)$ is continuous. By the structure of $V_f(\cdot)$, we know that $V_f(x_r(t), t) = 0$ for all $t \in \mathbb{T}$. We proceed by establishing that for all $x \in \mathbb{X}_f(t)$ and $t \in \mathbb{T}$, the terminal control law defined in (16) satisfies $\kappa_f(x,t) \in \mathbb{U}_f(x,t)$ and

[4] For values of $z_i < m_i$, the infinite horizon cost is strictly linear and the approximation fails to satisfy Assumption 3 in this region.

therefore the set $\mathbb{U}_f(x,t)$ is nonempty. From Lemma 3, we know that if $x \in \mathbb{X}_f(t)$, then $\kappa_f(x,t) \in \mathbb{U}$ and $x^+ := f(x, \kappa_f(x,t), 0, t) \in \mathbb{X}_f(t+1)$. We also know from Assumption 8 that $\bar{q}_i/(2m_i)$ is strictly positive and therefore $V_f(x,t)$ is uniformly bounded from below for all $x \in \mathbb{X}_f$ and $t \in \mathbb{T}$.

Next, we establish that $\kappa_f(x,t)$ satisfies the cost descent condition required to be in the set $\mathbb{U}_f(x,t)$, i.e.,

$$V_f(x^+, t+1) - V_f(x,t) + \bar{\ell}(x, \kappa_f(x,t), t) \le 0 \tag{19}$$

for all $x \in \mathbb{X}_f(t)$ and $t \in \mathbb{T}$. We note (from the previous discussion) that we may equivalently write (19) as a sum of scalar terms.

$$\sum_{i=1}^{\bar{n}} \left(V_{f,i}(z_i^+, t+1) - V_{f,i}(z_i, t) + \bar{q}_i z_i + \bar{r} v_i \right) \le 0 \tag{20}$$

We now establish that each term in the summation is nonpositive. We begin with the terminal cost at $z_i^+ = z_i(t+1)$.

$$\begin{aligned} V_{f,i}(z_i^+, t+1) &= \frac{\bar{q}_i}{2m_i}(z_i - v_i)^2 + (\bar{q}_i + \bar{r}_i)(z_i - v_i) \\ &= \frac{\bar{q}_i}{2m_i} z_i^2 - \frac{\bar{q}_i}{m_i} z_i v_i + \frac{\bar{q}_i}{2m_i} v_i^2 + (\bar{q}_i + \bar{r}_i) z_i - (\bar{q}_i + \bar{r}_i) v_i \\ &= V_{f,i}(z_i, t) - \frac{\bar{q}_i}{m_i} z_i v_i + \frac{\bar{q}_i}{2m_i} v_i^2 - (\bar{q}_i + \bar{r}_i) v_i \end{aligned}$$

We rearrange this equation as follows and note from Assumption 8 that $\bar{q}_i \ge 0$.

$$\begin{aligned} &V_{f,i}(z_i^+, t+1) - V_{f,i}(z_i, t) + \bar{q}_i z_i + \bar{r}_i v_i \\ &\quad \le \bar{q}_i \left(\frac{1}{2m_i} v_i^2 - \frac{1}{m_i} z_i v_i + z_i - v_i \right) \end{aligned}$$

Now we use the terminal control law, $v_i = \min\{z_i, m_i\}$. If $z_i \le m_i$, we know that $v_i = z_i$ and

$$\frac{1}{2m_i} v_i^2 - \frac{1}{m_i} z_i v_i + z_i - v_i = -\frac{1}{2m_i} z_i^2 \le 0$$

If instead $z_i > m_i$, we know that $v_i = m_i$ and

$$\frac{1}{2m_i} v_i^2 - \frac{1}{m_i} z_i v_i + z_i - v_i = -\frac{1}{2} m_i \le 0$$

Thus, each term of the summation is nonpositive and $\kappa_f(x,t)$ satisfies (19). The set $\mathbb{U}_f(x,t)$ is therefore nonempty for all $x \in \mathbb{X}_f(t)$ and $t \in \mathbb{T}$ and Assumption 3 is satisfied. □

Thus, we define the linear-quadratic terminal cost for production scheduling as

$$V_f(x,t) := (x - x_r(t))' P (x - x_r(t)) + p'(x - x_r(t)) \tag{21}$$

in which $P \in \mathbb{R}^{n \times n}$ and $p \in \mathbb{R}^n$. By defining

$$P := \mathrm{diag}([0, 0, 0, (\pi^S/\mu)', (\pi^U/(2\sigma))'])$$

$$p := [0, 0, 0, (\pi^S + \pi^D)', (\pi^U - \pi^D)']'$$

in which '/' is element-wise division, we observe that (21) is equivalent to (18) and therefore satisfies Assumption 3 by Proposition 4.[5]

Remark 4. The terminal cost requires quadratic terms and we therefore must solve a mixed-integer quadratic program (MIQP). Although such an increase in complexity may appear undesirable, we note that (1) most mixed integer solvers can handle MIQPs as well and (2) the increase is computational complexity is minimal since the majority of the problem is linear and the terminal cost is convex (P is positive semi-definite). If, despite all these features, quadratic terms are still a nonstarter, we note that $P = 0$ and

$$p := [0, 0, 0, (b\pi^S/(\mu/2) + \pi^D)', (b\pi^U/(\sigma) - \pi^D)']$$

in which $b \ge 0$, is a valid terminal cost if $z_i \le b$ for all $i \in \mathbb{I}_{[1,\bar{n}]}$. Thus, a large linear penalty may be sufficient in practice. Risbeck et al. (2019) also achieve satisfactory results by applying a large linear penalty.

Remark 5. For final products $k \in \mathbf{K}^P$ that we can sell for profit but do not have demand (i.e., $\nu_k^S > 0$ and $\xi_k(t) = 0$ for all $t \in \mathbb{T}$), we can set $\sigma_k = 0$ in the reference trajectory. The terminal cost for exceeding the backlog of the reference trajectory is therefore infinite and is equivalent to a hard constraint, i.e., we require that $U_k(t+N) = U_{r,k}(t+N)$ in the terminal constraint. However, since we have no demand for this product, $U_k(t+N) = U_{r,k}(t+N) = 0$ for all $t \in \mathbb{T}$ and $N \in \mathbb{I}_{\ge 0}$, this cost/constraint is irrelevant to the optimization problem and does not affect feasibility. Similarly, if we set $\mu_k = 0$ for this same product, we effectively have a terminal equality constraint for inventory of product k. For products that can always be sold in excess of demand, this terminal equality constraint may not affect the feasibility of the optimization problem.

4.3. Verifying Assumption 5

Thus far, we have constructed a terminal constraint and cost to satisfy both Assumptions 3 and 4. To establish robust performance for the closed-loop scheduling formulation, we also must ensure that Assumption 5 is satisfied. We note the following properties of the system and stage cost. The proofs of these lemmata are available in the appendix.

Lemma 5. *Let Assumption 2 hold. For the nominal system $x^+ = Ax + Bu + c(t)$ and fixed $N \in \mathbb{I}_{\ge 0}$, there exists $e_1, e_2 > 0$ satisfying*

$$|\hat{\phi}(k; x_1, \mathbf{u}_1, t) - \hat{\phi}(k; x_2, \mathbf{u}_2, t)| \le e_1 |x_1 - x_2| + e_2 \tag{22}$$

for all $x_1, x_2 \in \mathcal{X}_N(t)$, $\mathbf{u}_1, \mathbf{u}_2 \in \mathcal{U}_N(x,t)$, $t \in \mathbb{T}$, and $k \in \mathbb{I}_{[t,t+N]}$.

Lemma 6. *Let Assumptions 2 and 8 hold. For the stage cost $\ell(x,u,t) = q'x + r'u$, there exists $c_1, c_2 > 0$ and $d_1, d_2 \ge 0$ satisfying*

$$c_1|x_1 - x_2| - d_1 \le |\ell(x_1, u_1, t) - \ell(x_2, u_2, t)| \le c_2|x_1 - x_2| + d_2 \tag{23}$$

for all $x_1, x_2 \in \mathbb{X}$, $u_1, u_2 \in \mathbb{U}$, and $t \in \mathbb{T}$.

In addition, we require an assumption concerning the perturbed system evolution.

Assumption 9 (Bounded perturbation)**.** There exists $e_3 \ge 0$ such that

$$|f(x,u,w,t) - f(x,u,0,t)| \le e_3$$

for all $x \in \mathbb{X}$, $u \in \mathbb{U}$, $t \in \mathbb{T}$, and $w \in \mathbb{W}$.

We note that Assumption 9 is satisfied for the production scheduling model and disturbances considered. Now we use these properties to prove that the production scheduling model satisfies Assumption 5.

Proposition 7. *Consider the system (1), constraints $\mathbb{X}$, $\mathbb{U}$, $\mathbb{Z}$, $\mathbb{W}$, and stage cost (6) defined for production scheduling. Let the terminal constraint and terminal cost be defined by (15) and (21). Let Assumptions 1–4, 8, and 9 hold. Then Assumption 5 holds.*

Proof. We choose $x \in \mathcal{X}_N(t-1)$, $t-1 \in \mathbb{T}$ and define the perturbed and nominal evolution from x as $x_1 := f_c(x, w, t-1)$, $x_2 := f_c(x, 0, t-1)$ for some $w \in \mathbb{W}$. From these initial conditions, we define the optimal control trajectories $\mathbf{u}_1 := \mathbf{u}^0(x_1, t)$, $\mathbf{u}_2 := \mathbf{u}^0(x_2, t)$

[5] If $\pi_k^D > \pi_k^U$ for some $k \in \mathbf{K}^P$, we can set the last term in p as $\max\{\pi^U - \pi^D, 0\}$ and Proposition 4 still holds.

and $t \in \mathbb{T}$. Furthermore, we denote $x_1(k) := \hat{\phi}(k; x_1, \mathbf{u}_1, t)$, $x_2(k) := \hat{\phi}(k; x_2, \mathbf{u}_2, t)$, and $x_{1,f} := x_1(t+N)$, $x_{2,f} := x_2(t+N)$. Note that Lemma 5 and 6 hold. We combine the bound in Assumption 9 with (22) to give

$$|x_1(k) - x_2(k)| \le e_1 e_3 + e_2 =: e_4 \tag{24}$$

We write the optimal cost difference as follows.

$$\begin{aligned}
&\bar{V}_N^0(x_1, t) - \bar{V}_N^0(x_2, t) \\
&\quad = \sum_{k=t}^{t+N-1} (\ell(x_1(k), u_1(k), k) - \ell(x_2(k), u_2(k), k)) \\
&\qquad + V_f(x_{1,f}, t+N) - V_f(x_{2,f}, t+N) \\
&\quad \le \sum_{k=t}^{t+N-1} (c_2 |x_1(k) - x_2(k)| + d_2) \\
&\qquad + V_f(x_{1,f}, t+N) - V_f(x_{2,f}, t+N) \\
&\quad \le \tilde{b}_1 + V_f(x_{1,f}, t+N) - V_f(x_{2,f}, t+N)
\end{aligned}$$

in which $\tilde{b}_1 := N(c_2 e_4 + d_2)$. Next, we bound the terminal cost difference.

$$\begin{aligned}
&V_f(x_{1,f}, t+N) \\
&\quad = |x_{1,f} - x_r(t+N)|_P^2 + p'(x_{1,f} - x_r(t+N)) \\
&\quad \le |x_{2,f} - x_r(t+N)|_P^2 + p'(x_{2,f} - x_r(t+N)) \\
&\qquad + 2(x_{1,f} - x_{2,f})' P (x_{2,f} - x_r(t+N)) \\
&\qquad + |x_{1,f} - x_{2,f}|_P^2 + p'(x_{1,f} - x_{2,f}) \\
&\quad \le V_f(x_{2,f}, t+N) + 2e_4 |P| |x_{2,f} - x_r(t+N)| + \tilde{b}_2
\end{aligned}$$

in which $\tilde{b}_2 := |P| e_4^2 + |p| e_4$. Now we use (22) with $x_1 = x_r(t)$ to bound the distance of $x_{2,f}$ from the trajectory.

$$|x_{2,f} - x_r(t+N)| \le e_1 |x_2 - x_r(t)| + e_2$$

Next, we consider x, $x_r(t-1)$, and $k = t$ in (22) and (23) to give

$$\begin{aligned}
|x_2 - x_r(t)| &\le e_1 |x - x_r(t-1)| + e_2 \\
&\le \frac{e_1}{c_1} |\bar{\ell}(x, u, t-1)| + \frac{e_1 d_1}{c_1} + e_2
\end{aligned}$$

and thus

$$2e_4 |P| |x_{2,f} - x_r(t+N)| \le b_1 |\bar{\ell}(x, u, t-1)| + \tilde{b}_3$$

in which $b_1 := 2e_4 |P| e_1^2 / c_1$ and $\tilde{b}_3 := 2e_4 |P| (b_1 d_1 + e_1 e_2 + e_2)$. Therefore, the optimal cost difference has the following upper bound.

$$\bar{V}_N^0(x_1, t) - \bar{V}_N^0(x_2, t) \le b_1 |\bar{\ell}(x, u, t-1)| + \tilde{b}_1 + \tilde{b}_2 + \tilde{b}_3 \tag{25}$$

From Assumption 2, there exists $m \in \mathbb{R}$ such that $\bar{\ell}(x, u, t) \ge m$ for all $(x, u, t) \in \mathbb{Z} \times \mathbb{T}$. We now use the nominal cost decrease ensured by Assumption 3 to write

$$\bar{V}_N^0(x_2, t) - \bar{V}_N^0(x, t-1) \le -\bar{\ell}(x, u, t-1) \le -m$$

We combine this equation with (25) to give

$$\bar{V}_N^0(x_1, t) - \bar{V}_N^0(x, t-1) \le b_1 |\bar{\ell}(x, u, t-1)| + b_2 \tag{26}$$

in which $b_2 \ge \tilde{b}_1 + \tilde{b}_2 + \tilde{b}_3 - m$. Finally, we note that $x_1 = f_c(x, w, t-1)$, redefine $t - 1 := t$, and the proof is complete. □

With Proposition 7, we have verified that Assumption 5 holds for the production scheduling algorithm proposed.

Remark 6. The parameter b_1, as noted in Remark 2, is particularly significant for the robustness to large, infrequent disturbances. In the proof of Proposition 7, we define b_1 as proportional to $|P|$. In addition, the terminal cost is defined such that $|P|$ is inversely related to the minimum element of σ. Thus, as the minimum value of σ approaches zero, $|P| \to \infty$, $b_1 \to \infty$, and $\delta \to 0$. Conversely, if we allow all elements of σ and μ to approach infinity, then $P \to 0$ and $\delta \to 1$. This observation indicates a potential design consideration: if we increase the "robustness" of the reference trajectory by increasing the value of σ, i.e., overproduction, we may thereby increase the robustness of the closed-loop scheduling algorithm as well (at the expense of the nominal performance guarantee). In the future, we may be able to leverage results in robust and stochastic optimization to produce superior reference trajectories (offline) and thereby improve the robustness of the deterministic closed-loop scheduling algorithm (online).

4.4. Summary

Through specific terminal constraints, terminal costs, and reasonable assumptions, we have established that the defined closed-loop scheduling formulation satisfies Assumptions 1–5. We summarize these results through the following theorem.

Theorem 8. *Consider the system (1), constraints $\mathbb{X}$, $\mathbb{U}$, $\mathbb{Z}$, $\mathbb{W}$, and stage cost (6) defined for production scheduling. Let the terminal constraint and terminal cost be defined by (15) and (21). Let Assumptions 1, 2, 4, and 6–9 hold. Then the closed-loop system $x^+ = f_c(x, w, t)$, $w \in \mathbb{W}$ is economically robust to large, infrequent disturbances, i.e., the closed-loop scheduling algorithm is inherently robust.*

Assumptions 7 and 8 hold due to the structure of the nominal production scheduling problem and appropriate costs. Assumption 6 requires a suitable overproduction margin σ is used in the reference trajectory and there is the option to dispose of inventory for products. We note that Assumptions 4 and 9 are the only assumptions that address the behavior of the perturbed system. If we want to add any additional disturbances that may be relevant to closed-loop scheduling problems, we need to check only that Assumptions 4 and 9 hold as the remaining assumptions are unaffected.

In practice, the proposed methods to construct a terminal constraint and cost may be inconvenient. We therefore suggest a more practical approximation of this algorithm as follows. First, find a suitable reference trajectory for the nominal system. Use this reference trajectory as a terminal equality constraint for all state variables aside from inventory and backlog. Allow the terminal region to include any values of inventory and backlog that exceed the reference trajectory and apply a large linear penalty to these deviations. While this algorithm does not exactly satisfy all the assumptions required of Theorem 8, it is a well-motivated approximation.

5. Example

To demonstrate the implications of this analysis, we study the following example problem. We consider a manufacturing facility with two units. The first unit (U1) may run task 1 (T1) to consume raw material (assumed to be in abundant supply) and produce the intermediate material M1. Either task 2 or 3 (T2 or T3) may be run on unit 2 (U2) that consumes the intermediate material M1 to produce either M2 or M3, respectively. The min/max batch sizes for units 1 and 2 are 5/20 kg and 10/20 kg, respectively. The processing times required to complete tasks 1 and 2 are 2 hours. The processing time to complete task 3 is 3 hours.

Next, we describe the economics of the facility. There is demand for 45 kg of M2 every 6 hours of operation. If we are unable to meet this demand, we incur backlog. The penalty for maintaining backlog of M2 is \$10/kg/hr. There is no explicit demand for M3, but we may sell up to 5 kg of M3 each hour at a sales price of \$10/kg. The penalty for maintaining inventory of M2 or M3 is \$1/kg. Each hour we may dispose of up to 1 kg of M2 or M3 at a cost of \$12/kg. We can store up to 40 kg of M1, 100 kg of M2,

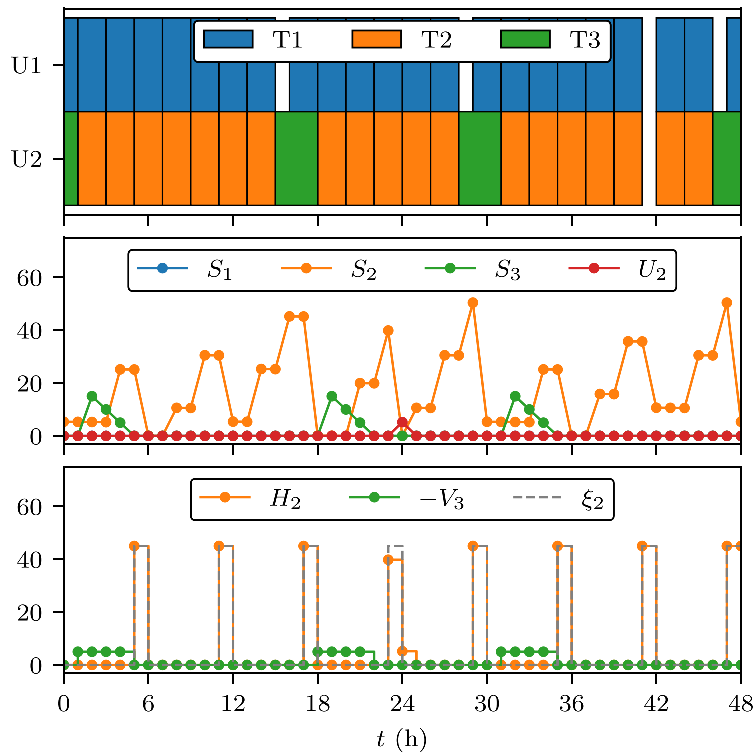

Fig. 3. Optimal periodic schedule for an horizon length of 48 hours with $\sigma_2 = 0.05$. The top plot is a Gantt chart for both units. The middle plot shows the inventory (S) for each material and the backlog for M2 (U_2). The bottom plot shows the shipments of M2 to meet demand (H_2), sale of excess M3 (V_3), and demand for M2 (ξ_2).

and 20 kg of M3. Thus, an optimal schedule is one that meets demand for M2 while maximizing the production and sale of M3 in the remaining time.

In Fig. 3, we plot the optimal periodic schedule for a 48 hour horizon assuming we overproduce M2 by 0.05 kg/hr, i.e., $\sigma_2 = 0.05$. For all subsequent simulations and analysis, we use this periodic schedule as the reference trajectory. Note that the periodic solution and therefore the reference trajectory includes nonzero backlog at $t = 24$. We intentionally consider a problem with this feature to emphasize that (1) the schedule for even simple scheduling problems may be nonobvious and (2) we may include nonzero backlog in the reference trajectory without disrupting any of the results in this work. Note that the reference trajectory disposes of 0.05 kg of M2 every hour, but we omit D_2 from the plot since the trajectories of D_2 are impossible to distinguish from zero on the y-axis scales used.

5.1. Case 1: Unit 1 breakdown

For the first case study, we consider a single disturbance type. We assume that Unit 1 experiences a breakdown (for one hour) with probability ε. We also assume that there are no other disturbances affecting the system and therefore $\Pr(|w| > 0) = \varepsilon$. The open-loop scheduling algorithm has a horizon of 12 hours, and we use terminal constraints and costs constructed from the reference trajectory as detailed in Section 4. The initial state of the facility is in phase with the reference trajectory. We observe the disturbance every hour and update the schedule accordingly.

In Fig. 4, we plot an example trajectory for this system with $\varepsilon = 0.2$. The closed-loop (executed) schedule is drawn in solid colors and the open-loop (current) schedule is drawn in faded colors. The black boxes shown in the Gantt chart represent breakdowns.

We note a few important features in this example. First, the terminal constraint used in the open-loop optimization requires the open-loop schedule to include a run of T3 at the end of the open-loop horizon ($t = 30$) despite the fact that a run of T2 may be more economically favorable. However, the backlog is not required to terminate along the reference trajectory (i.e., $U_2(t+N) \geq 0$) and therefore the optimization problem remains feasible despite previous breakdowns that disrupted production of M2.

We also note that there is no inventory disposal at any point in the closed-loop schedule, despite including this option in the reference trajectory. The inventory disposal option is assessed a large cost per kg and is therefore avoided whenever possible. The point of the inventory disposal option is to provide a worst-case recourse option if we overproduce a product and thereby allows us to construct a periodic reference trajectory with overproduction. Allowing this recourse is essential to guarantee robustness for a general scheduling problem, but this recourse is not necessarily used for a given example.

To characterize the economic performance of the closed-loop schedule, we define $\Delta(t)$ as follows.

$$\Delta(t) := \frac{1}{t+1}\sum_{k=0}^{T} \bar{\ell}(x(k), u(k), k)$$

If $\Delta(t) > 0$ the average cost of the closed-loop schedule is worse than the reference trajectory. We simulate the closed-loop schedule over 2 weeks for 30 different realizations of the disturbance trajectory. In Fig. 5, we plot the value of $\Delta(t)$ for each of theses simulations as a function of time t. The sample average (expected value) of these trajectories is plotted in black. Note that the expected value of $\Delta(t)$ increases from zero as breakdowns cause the closed-loop facility to perform worse than the nominal reference trajectory. However, the expected value holds steady around \$40/hr after 2 weeks. Thus, the perturbed system is, on average, costing \$40/hr more than the nominal reference trajectory due to the disturbance.

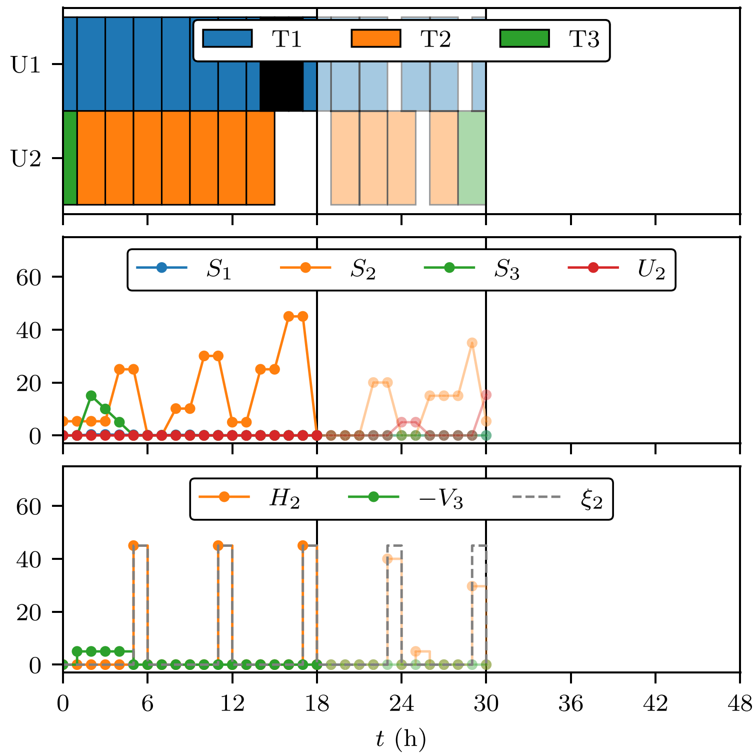

Fig. 4. An example trajectory with $\varepsilon = 0.2$. The closed-loop (executed) schedule is drawn in solid colors and the open-loop (current) schedule is drawn in faded colors. The top plot is a Gantt chart for both units with black boxes used to represent breakdowns. The middle plot shows the inventory (S) for each material and the backlog for M2 (U_2). The bottom plot shows the shipments of M2 to meet demand (H_2), sale of excess M3 (V_3), and demand for M2 (ξ_2).

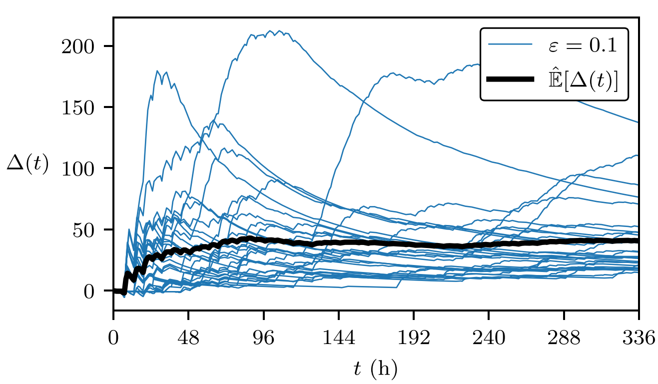

Fig. 5. The trajectory of $\Delta(t)$ for 30 simulations of the closed-loop system with a 0.2 chance of a breakdown for U1. The sample average is plotted in black.

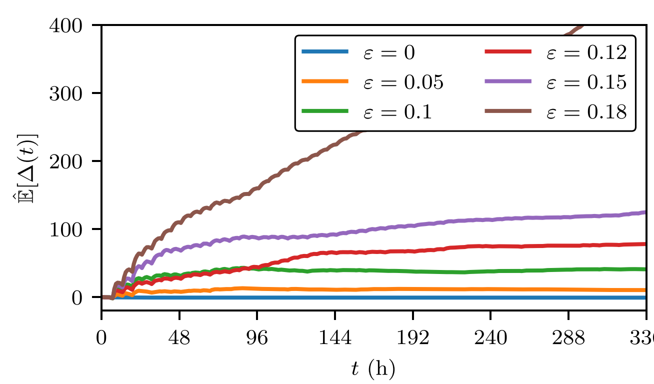

Fig. 6. The samples average of $\Delta(t)$ for 30 simulations for multiple values of ε.

In Fig. 6, we plot the value of $\hat{\mathbb{E}}[\Delta(t)]$ (the sample average of $\Delta(t)$) for multiple values of ε. For $\varepsilon \leq 0.12$, as t increases, $\hat{\mathbb{E}}[\Delta(t)]$ decays towards some nonnegative constant specific to each value of ε. As ε is increased, this constant increases and, conversely, as $\varepsilon \to 0$, $\hat{\mathbb{E}}[\Delta(t)] \to 0$. Note that for the nominal system ($\varepsilon = 0$), $\Delta(t) \leq 0$. For $\varepsilon \leq 0.12$, the breakdowns increase the cost of operation, but on average the closed-loop system can recover. However, if ε exceeds 0.12, the sample average diverges with increasing t. In this region, the breakdowns occur too frequently and on average the system is unable to keep up with demand. These results are consistent with Theorem 2 and we presume for this system and disturbance $0.12 \leq \delta < 0.15$.

We treat the value of $\hat{\mathbb{E}}[\Delta(t)]$ for $t = 336$ as an approximation of the infinite limit in (14) and denote this value as $\hat{\gamma}(\varepsilon)$ for each $\varepsilon \leq 0.12$. We plot the value of $\hat{\gamma}(\varepsilon)$ in Fig. 7. In addition to the terminal constraints and costs proposed in this work (denoted the *LQ* algorithm), we consider the performance of other algorithms with different terminal costs and constraints. We compare with the online scheduling algorithm detailed in Gupta and Maravelias (2019), which omits the terminal constraint ($\mathbb{X}_f(t) = \mathbb{X}$) and sets the terminal cost based on the stage cost ($V_f = \ell(x, 0, t)$). We refer to this algorithm as the *NTC* (no terminal constraints) algorithm. The *linear* algorithm uses the terminal constraint proposed in this work, but approximates the linear-quadratic cost with a large linear cost (detailed in Remark 4) for computational convenience. We note that for values of $\varepsilon > 0.12$, the trajectories of $\hat{\mathbb{E}}[\Delta(t)]$ diverge for all algorithms. Since this divergence is primarily the result of inherent limitations of the facility and not the algorithm of choice, this result is reasonable.

For each algorithm, the most obvious trend in Fig. 7 is consistent: The value of $\hat{\gamma}(\varepsilon)$ increases with increasing ε. This trend is

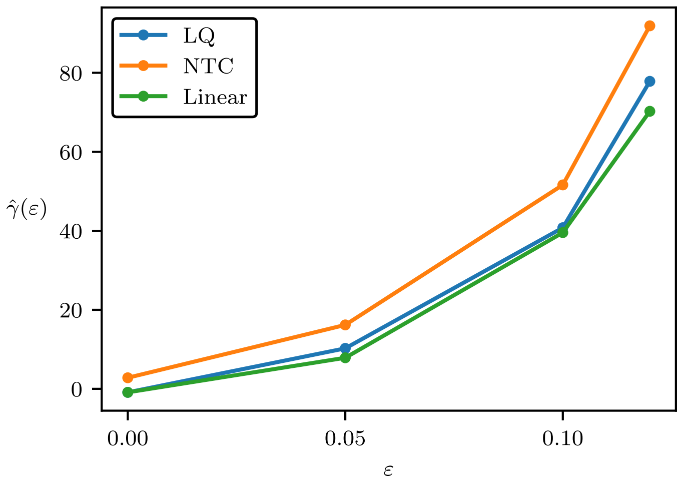

Fig. 7. The value of $\hat{\gamma}(\varepsilon)$ for multiple values of ε and three production scheduling algorithms with different terminal costs and constraints.

Table 1
The values of $\hat{\gamma}(\varepsilon)$ ($/hr) for closed-loop scheduling algorithms that use the LQ, standard, or linear terminal conditions for Case 1 (Unit 1 Breakdown) and Case 2 (All Disturbances).

Case	Alg.	ε					
		0	0.05	0.1	0.12	0.15	0.18
	LQ	−0.9	10.2	40.8	77.8	-	-
1	NTC	2.8	16.2	51.6	91.9	-	-
	Linear	−0.9	7.9	39.5	70.2	-	-
	LQ	−0.9	4.4	13.5	14.9	28.7	59.2
2	NTC	2.8	11.5	23.3	27.3	41.1	73.8
	Linear	−0.9	5.1	15.6	20.1	38.5	71.7

consistent with the conclusions in Gupta and Maravelias (2020). But, we also note key differences in performance. The NTC algorithm does not ensure that Theorems 1 and 2 hold. Consequently, as $\varepsilon \to 0$ and indeed for $\varepsilon = 0$, the performance of the NTC algorithm is worse than the LQ algorithm and the reference trajectory. In addition, this performance gap persists for all values of ε. We note that the performance of the linear algorithm is similar to the LQ algorithm and both achieve equivalent nominal performance.

We also observe that the nominal performance of the LQ and linear approximation algorithms are superior to the reference trajectory used to construct their terminal constraint. In fact, their nominal performance is nearly identical to 48-hr periodic optimal solution with $\sigma = 0$.[6] Thus, using a overproduction reference trajectory ($\sigma > 0$), as opposed to a standard optimal periodic reference trajectory ($\sigma = 0$), does not impair the nominal closed-loop performance for this example. Numerical values are provided in Table 1.

5.2. Case 2: All disturbances

In the second case study, we consider all potential disturbance types for all units. We consider the potential for breakdowns, 1 h delays, and 20% yield losses for both units at each hour. We denote the probability that any disturbance occurs as ε, i.e., $\Pr(|w| > 0) = \varepsilon$, and split the probability of a disturbance occurring equally between each disturbance type. Thus, we define $e := 1 - (1-\varepsilon)^{1/6}$ and consider $\Pr\big(Y_j = 1\big) = \Pr\big(Z_j = 1\big) = \Pr\big(L_j = 0.2\big) = e$ for both $j = 1, 2$. We run 30 simulations of the facility for each value of ε and perform the same analysis used in the previous case study.

The values of $\hat{\gamma}(\varepsilon)$ are shown in Table 1. Note that all observed trends are consistent with those discussed for Case 1. In particular, the LQ and linear terminal conditions proposed in this work outperform the standard scheduling algorithm at all values of ε. In fact, the LQ algorithm also performs better than linear algorithm. The values of $\hat{\gamma}(\varepsilon)$, however, are lower in Case 2 for all algorithms. In addition, the value of δ is larger for Case 2 ($0.18 \leq \delta < 0.2$) than Case 1 ($0.12 \leq \delta < 0.15$).

6. Conclusions

We defined the term robustness for closed-loop scheduling and illustrated the implications of this property through a simple example. By leveraging the inventory disposal decision included in our production scheduling formulation, we constructed a terminal constraint and cost from a suitable reference trajectory that satisfy Assumption 3 while ensuring the optimization problem is robustly recursively feasible (Assumption 4). Furthermore, we established under reasonable assumptions that the proposed formulation is inherently robust to large and infrequent disturbances such as breakdowns, delays, and yield losses.

We note that these robustness properties extend to any additional disturbances that may be relevant to closed-loop scheduling applications provided they satisfy Assumptions 4 and 9. The specific disturbances considered in this paper (e.g. breakdowns, delays, and yield losses) were chosen to illustrate the range of disturbances covered by this analysis and do not represent the full extent of these results. In addition, we also expect that these results can be extended to more complex scheduling models (see Gupta and Maravelias (2017)). In most cases, we can use terminal equality constraints for additional state variables without affecting Assumption 4. Thus, the remaining assumptions hold if the stage cost and nominal model remain linear.

We emphasize that the proposed LQ algorithm may not always outperform the closed-loop scheduling algorithm without terminal conditions. The proposed algorithm, however, offers nominal and robust performance guarantees that prevent poor performance of the closed-loop trajectory. Without these terminal conditions, there are no guarantees. We construct these algorithms and establish these theoretical results to guide the modification and development of current and future closed-loop scheduling algorithms.

We also note that the analysis herein establishes an interesting, previously not recognized, connection between methods and results in the areas of scheduling and control. Scheduling problems are often classified as short-term or cyclic/period problems. The former arise in systems with high demand variability, where one has to continuously generate different solutions to react to changing demand, whereas the latter, which appear in systems with regular demand, aim to generate a single schedule that can be repeatedly implemented. However, in the presence of disturbances, a periodic schedule can (almost) never be implemented. In fact, in this paper we showed that one should always be solving an open-loop short-term problem (even in the presence of regular demand), but, importantly, subject to terminal constraints and costs that are obtained by solving a period scheduling problem. In other words, periodic schedules become necessary as terminal constraints rather than directly implementable schedules.

There are multiple avenues for future research. To implement optimization algorithms online for industrial scale problems, solving until the optimality gap reaches zero is typically unrealistic. Thus, algorithms that provide performance guarantees with suboptimal solutions are desirable. If there is significant uncertainty in the predicted demand pattern, the scheduling algorithm may ben-

[6] The difference between the optimal 48hr periodic reference trajectory with and without overproduction is -$0.9/hr.

efit from incorporating stochastic information in the optimization problem (e.g., stochastic MPC) or in the design of the reference trajectory and terminal conditions.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

James B. Rawlings reports a relationship with Johnson Controls Inc that includes: consulting or advisory. Christos T. Maravelias reports a relationship with Johnson Controls Inc that includes: consulting or advisory.

CRediT authorship contribution statement

Robert D. McAllister: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Visualization. **James B. Rawlings:** Conceptualization, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Christos T. Maravelias:** Conceptualization, Writing – review & editing, Supervision, Funding acquisition.

Acknowledgment

The authors gratefully acknowledge the financial support of the NSF through grant #2027091.

Appendix A. Additional Proofs

Proof of Lemma 5. We denote $x_1(k) := \hat{\phi}(k; x_1, \mathbf{u}_1, t)$, $x_2(k) := \hat{\phi}(k; x_2, \mathbf{u}_2, t)$ and write

$$\begin{aligned}&|x_1(t+k) - x_2(t+k)|\\ &= \left|A^k(x_1 - x_2) + \sum_{i=0}^{k-1} A^{k-1-i}B(u_1(i+t) - u_2(i+t))\right|\\ &\leq |A|^k|x_1 - x_2| + \sum_{i=0}^{k-1} |A|^{k-1-i}|B||u_1(i+t) - u_2(i+t)|\end{aligned}$$

Since $\mathbb{U}$ is compact, there exits $c \geq 0$ such that $|u_1(i+t) - u_2(i+t)| \leq c$. Thus, we define $e_1 := \max_{k\in\{0,1,\ldots,N\}} |A|^k$, and have

$$|x_1(t+k) - x_2(t+k)| \leq e_1|x_1 - x_2| + N(e_1|B|c) \tag{A.1}$$

We define $e_2 := N(e_1|B|c)$ and the proof is complete. □

Proof of Lemma 6. We note that $\mathbb{U}$ is compact and therefore $|u_1 - u_2| \leq c$ for some $c > 0$. Thus,

$$|\ell(x_1, u_1, t) - \ell(x_2, u_2, t)| \leq |q||x_1 - x_2| + |r|c$$

and $c_2 = |q|$, $d_2 = |r|c$. We also have

$$|\ell(x_1, u_1, t) - \ell(x_2, u_2, t)| \geq |q'(x_1 - x_2)| - |r|c$$

We note that the variables $\bar{W}$, $\bar{B}$, $\tilde{S}$, are bounded (Assumption 8 ensures bounded $\tilde{S}$). Thus, there exists $d \geq 0$ such that $c_1|x_1 - x_2| \leq d$ for all $x_1, x_2 \in \mathbb{X}$ in which $[U', S']_1 = [U', S']_2 = 0$. Therefore,

$$\begin{aligned}|q'(x_1 - x_2)| &\geq |\pi^U||[U', S']_1' - [U', S']_2'|\\ &\geq c_1|[U', S']_1' - [U', S']_2'|\\ &\geq c_1|x_1 - x_2| - d\end{aligned}$$

We define $d_1 := d + |r|c$ and the proof is complete. □

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.compchemeng.2022.107678.

References

Allan, D.A., Bates, C.N., Risbeck, M.J., Rawlings, J.B., 2017. On the inherent robustness of optimal and suboptimal nonlinear MPC. Sys. Cont. Let. 106, 68–78. doi:10.1016/j.sysconle.2017.03.005.
Amrit, R., Rawlings, J.B., Angeli, D., 2011. Economic optimization using model predictive control with a terminal cost. Annual Rev. Control 35, 178–186.
Angeli, D., Amrit, R., Rawlings, J.B., 2012. On average performance and stability of economic model predictive control. IEEE Trans. Auto. Cont. 57 (7), 1615–1626.
Avadiappan, V., Maravelias, C.T., 2021. State estimation in online batch production scheduling: concepts, definitions, algorithms and optimization models. Comput. Chem. Eng. 146, 107209.
Balasubramanian, J., Grossmann, I.E., 2004. Approximation to multistage stochastic optimization in multiperiod batch plant scheduling under demand uncertainty. Ind. Eng. Chem. Res. 43 (14), 3695–3713.
Bonfill, A., Bagajewicz, M., Espuña, A., Puigjaner, L., 2004. Risk management in the scheduling of batch plants under uncertain market demand. Ind. Eng. Chem. Res. 43 (3), 741–750.
Chu, Y., You, F., 2014. Moving horizon approach of integrating scheduling and control for sequential batch processes. AIChE J. 60 (5), 1654–1671.
Cott, B.J., Macchietto, S., 1989. Minimizing the effects of batch process variability using online schedule modification. Comput. Chem. Eng. 13 (1–2), 105–113.
Cui, J., Engell, S., 2010. Medium-term planning of a multiproduct batch plant under evolving multi-period multi-uncertainty by means of a moving horizon strategy. Comput. Chem. Eng. 34 (5), 598–619.
Dalle Ave, G., Alici, M., Harjunkoski, I., Engell, S., 2019. An explicit online resource–task network scheduling formulation to avoid scheduling nervousness. In: Computer Aided Chemical Engineering, vol. 46. Elsevier, pp. 61–66.
Elkamel, A., Mohindra, A., 1999. A rolling horizon heuristic for reactive scheduling of batch process operations. Eng. Optim. 31 (6), 763–792.
Ellis, M., Durand, H., Christofides, P.D., 2014. A tutorial review of economic model predictive control methods. J. Proc. Cont. 24 (8), 1156–1178.
Ferrer-Nadal, S., Méndez, C.A., Graells, M., Puigjaner, L., 2007. Optimal reactive scheduling of manufacturing plants with flexible batch recipes. Ind. Eng. Chem. Res. 46 (19), 6273–6283.
Grimm, G., Messina, M.J., Tuna, S.E., Teel, A.R., 2004. Examples when nonlinear model predictive control is nonrobust. Automatica 40, 1729–1738.
Gupta, D., Maravelias, C., 2017. A general state-space formulation for online scheduling. Processes 5 (4), 69.
Gupta, D., Maravelias, C.T., 2016. On deterministic online scheduling: major considerations, paradoxes and remedies. Comput. Chem. Eng. 94, 312–330.
Gupta, D., Maravelias, C.T., 2019. On the design of online production scheduling algorithms. Comput. Chem. Eng. 129, 106517.
Gupta, D., Maravelias, C.T., 2020. Framework for studying online production scheduling under endogenous uncertainty. Comput. Chem. Eng. 135, 106670.
Gupta, D., Maravelias, C.T., Wassick, J.M., 2016. From rescheduling to online scheduling. Chem. Eng. Res. Des. 116, 83–97.
Harjunkoski, I., Maravelias, C.T., Bongers, P., Castro, P.M., Engell, S., Grossmann, I.E., Hooker, J., Méndez, C., Sand, G., Wassick, J., 2014. Scope for industrial applications of production scheduling models and solution methods. Comput. Chem. Eng. 62, 161–193. doi:10.1016/j.compchemeng.2013.12.001.
Huercio, A., Espuna, A., Puigjaner, L., 1995. Incorporating on-line scheduling strategies in integrated batch production control. Comput. Chem. Eng. 19, 609–614.
Janak, S.L., Floudas, C.A., Kallrath, J., Vormbrock, N., 2006. Production scheduling of a large-scale industrial batch plant. II. reactive scheduling. Ind. Eng. Chem. Res. 45 (25), 8253–8269.
Kondili, E., Pantelides, C.C., Sargent, R., 1993. A general algorithm for short term scheduling of batch operations–I. MILP formulation. Comput. Chem. Eng. 17, 211–227.
Kopanos, G.M., Pistikopoulos, E.N., 2014. Reactive scheduling by a multiparametric programming rolling horizon framework: a case of a network of combined heat and power units. Ind. Eng. Chem. Res. 53 (11), 4366–4386.
Lappas, N.H., Gounaris, C.E., 2016. Multi-stage adjustable robust optimization for process scheduling under uncertainty. AIChE J. 62 (5), 1646–1667.
Lappas, N.H., Ricardez-Sandoval, L.A., Fukasawa, R., Gounaris, C.E., 2019. Adjustable robust optimization for multi-tasking scheduling with reprocessing due to imperfect tasks. Optim. Eng. 20 (4), 1117–1159.
Lee, H., Gupta, D., Maravelias, C.T., 2020. Systematic generation of alternative production schedules. AIChE J. 66 (5), e16926.
Li, Z., Ierapetritou, M., 2008a. Process scheduling under uncertainty: review and challenges. Comput. Chem. Eng. 32 (4), 715–727.
Li, Z., Ierapetritou, M.G., 2008b. Reactive scheduling using parametric programming. AIChE J. 54 (10), 2610–2623. doi:10.1002/aic.11593.
Li, Z., Ierapetritou, M.G., 2008c. Robust optimization for process scheduling under uncertainty. Ind. Eng. Chem. Res. 47 (12), 4148–4157.
Lin, X., Janak, S.L., Floudas, C.A., 2004. A new robust optimization approach for scheduling under uncertainty: i. bounded uncertainty. Comput. Chem. Eng. 28 (6–7), 1069–1085.
Maravelias, C.T., 2012. General framework and modeling approach classification for chemical production scheduling. AIChE J. 58 (6), 1812–1828.
Mcallister, R.D., Rawlings, J.B., 2021. Inherent stochastic robustness of model predictive control to large and infrequent disturbances. IEEE Trans. Auto. Cont..
McAllister, R.D., Rawlings, J.B., Maravelias, C.T., 2020. Rescheduling penalties for economic model predictive control and closed-loop scheduling. Ind. Eng. Chem. Res. 59 (6), 2214–2228.

Mendez, C.A., Cerdá, J., 2004. An MILP framework for batch reactive scheduling with limited discrete resources. Comput. Chem. Eng. 28 (6–7), 1059–1068.
Novas, J.M., Henning, G.P., 2010. Reactive scheduling framework based on domain knowledge and constraint programming. Comput. Chem. Eng. 34 (12), 2129–2148.
Pannocchia, G., Rawlings, J.B., Wright, S.J., 2011. Conditions under which suboptimal nonlinear MPC is inherently robust. Sys. Cont. Let. 60, 747–755.
Pantelides, C.C., 1994. Unified frameworks for optimal process planning and scheduling. In: Proceedings on the Second Conference on Foundations of Computer Aided Operations. Cache Publications New York, pp. 253–274.
Rawlings, J.B., Risbeck, M.J., 2017. Model predictive control with discrete actuators: theory and application. Automatica 78, 258–265.
Risbeck, M.J., Maravelias, C.T., Rawlings, J.B., 2019. Unification of closed-loop scheduling and control: state-space formulations, terminal constraints, and nominal theoretical properties. Comput. Chem. Eng. 129, 106496.
Risbeck, M.J., Rawlings, J.B., 2019. Economic MPC for time-varying cost and peak demand charge optimization. IEEE Trans. Auto. Cont. 65 (7), 2957–2968. doi:10.1109/TAC.2019.2939633.
Sand, G., Engell, S., 2004. Modeling and solving real-time scheduling problems by stochastic integer programming. Comput. Chem. Eng. 28 (6), 1087–1103.
Shi, H., You, F., 2016. A computational framework and solution algorithms for two-stage adaptive robust scheduling of batch manufacturing processes under uncertainty. AIChE J. 62 (3), 687–703.
Subramanian, K., Maravelias, C.T., Rawlings, J.B., 2012. A state-space model for chemical production scheduling. Comput. Chem. Eng. 47, 97–110.
Vin, J.P., Ierapetritou, M.G., 2000. A new approach for efficient rescheduling of multiproduct batch plants. Ind. Eng. Chem. Res. 39 (11), 4228–4238.
Vin, J.P., Ierapetritou, M.G., 2001. Robust short-term scheduling of multiproduct batch plants under demand uncertainty. Ind. Eng. Chem. Res. 40 (21), 4543–4554.