Distributed Economic Model Predictive Control for Operational Safety of Nonlinear Processes

Fahad Albalawi
Department of Electrical Engineering, University of California, Los Angeles, CA 90095

Helen Durand
Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095

Panagiotis D. Christofides
Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095

Department of Electrical Engineering, University of California, Los Angeles, CA 90095

DOI 10.1002/aic.15710
Published online March 24, 2017 in Wiley Online Library (wileyonlinelibrary.com)

Achieving operational safety of chemical processes while operating them in an economically-optimal manner is a matter of great importance. Our recent work integrated process safety with process control by incorporating safety-based constraints within model predictive control (MPC) design; however, the safety-based MPC was developed with a centralized architecture, with the result that computation time limitations within a sampling period may reduce the effectiveness of such a controller design for promoting process safety. To address this potential practical limitation of the safety-based control design, in this work, we propose the integration of a distributed model predictive control architecture with Lyapunov-based economic model predictive control (LEMPC) formulated with safety-based constraints. We consider both iterative and sequential distributed control architectures, and the partitioning of inputs between the various optimization problems in the distributed structure based on their impact on process operational safety. Moreover, sufficient conditions that ensure feasibility and closed-loop stability of the iterative and sequential safety distributed LEMPC designs are given. A comparison between the proposed safety distributed EMPC controllers and the safety centralized EMPC is demonstrated via a chemical process example. © 2017 American Institute of Chemical Engineers *AIChE J*, 63: 3404–3418, 2017

Keywords: distributed economic model predictive control, process safety, computation time

Introduction

The chemical process industries have experienced staggering profit losses due to accidents; for example, it was reported that the 20 accidents that caused the largest property damage losses in the hydrocarbon industry from 1974 to 2015 cost over $15 billion, with the total accumulated value of the 100 largest losses at more than $33 billion (estimates in 2015 dollars[16]). It is clear from these numbers that it is necessary to improve process safety from both the ethical perspective of saving lives and property, and also from an economics standpoint. Chemical process safety can be addressed through process design decisions; for instance, designing the process to be inherently safe in terms of its chemistry and physics.[19,20] Inherently safer designs are achieved through four primary principles: minimize (reduce the quantity of hazardous substances used and stored by a process), substitute (utilize less hazardous process chemistries), moderate (dilute chemicals or change operating conditions), and simplify (choose designs with less complexity and less potential to create hazardous conditions when faults or errors occur).[2] In addition, control and safety system design decisions (e.g., adding sensors for critical process variables that trigger an alarm when a measurement outside of the desired range is obtained[12]) are used to promote process safety. The traditional approach to preventing unsafe situations assumes that accidents have a root cause, rather than viewing them as a property of a system.[5] Recent works have called for a systems approach to process safety where past catastrophic incidents are studied from a systems engineering perspective to better design and control such systems in the future.[17,18]

Motivated by a systems-based, control-inspired approach to thinking about safety where a relationship exists between safety and model-based control,[17] our prior work proposed an EMPC design that includes explicit constraints on process safety.[10] The proposed safety-based controller shrinks the region of process operation to a smaller level set of operation termed a safety level set when a safety logic unit determines that certain regions of state-space might introduce process safety issues. The proposed control technique was designed in a centralized fashion where all the decision variables are solved together in one optimization problem. For a relatively

Correspondence concerning this article should be addressed to P. D. Christofides at pdc@seas.ucla.edu.

small process (e.g., one unit), the centralized safety-based EMPC formulation in Ref. 10 may be capable of computing an optimal solution that meets the safety-based constraints within a reasonable time frame. However, for large-scale nonlinear process systems, which are the common case in industry, the computational burden of solving a centralized EMPC design with potentially tens or hundreds of optimization variables increases. Hence, the ability of the centralized EMPC to enhance process safety for such high-dimensional nonlinear processes may decrease due to the computation time limitations within a sampling period. Alternatively, a model predictive control (MPC) scheme that overcomes this computational burden of solving a centralized EMPC design is a distributed model predictive control architecture.[11,13,15,26] Several research works[7–9] have shown the computation time benefits of the distributed MPC architecture over the centralized MPC while maintaining closed-loop stability and recursive feasibility. Distributed designs also can be beneficial from the perspective of fault-tolerance,[11] which is another safety consideration. A recent research work developed two different distributed economic model predictive controller (DEMPC) schemes that reduce the computation time of a centralized EMPC scheme while maintaining similar closed-loop performance.[14] However, these two schemes lack the ability to drive the state of the closed-loop system to a safe region of operation because their formulations do not include safety-based constraints. To date, no work on incorporating safety-based constraints within a DEMPC has been completed.

In this work, we design two different DEMPC architectures, namely, a sequential DEMPC architecture and an iterative DEMPC architecture, for nonlinear systems via Lyapunov-based techniques that incorporate safety-based constraints. A discussion on how to group the inputs into different distributed EMPC controllers based on their impact on process safety and process economics is presented. Sufficient conditions under which the state of the closed-loop system can be driven to the safety region and remain there for all subsequent times are derived. A catalytic reactor example is utilized to demonstrate the computational time improvement of the proposed control architectures over the centralized one while achieving similar closed-loop performance and process safety performance.

Preliminaries

Notation

The operator $|\cdot|$ denotes the Euclidean norm of a vector. x^T represents the transpose of a vector x. The symbol Ω_ρ is used to represent a level set of a sufficiently smooth, positive definite scalar-valued function $V(x)$ and is defined by $\Omega_\rho := \{x \in R^n : V(x) \leq \rho\}$. The operator "/" denotes set subtraction, that is, $A/B := \{x \in R^n : x \in A, x \notin B\}$. The symbol $S(\Delta)$ denotes the family of piecewise constant, right-continuous functions with a fixed time interval $\Delta \geq 0$. A diagonal matrix which has the components of a vector v as its diagonal elements is denoted by the symbol $\mathrm{diag}(v)$. A function $\alpha(\cdot) : [0, a) \rightarrow [0, \infty)$ belongs to class $\mathcal{K}$ (i.e., $\alpha \in \mathcal{K}$) if it is strictly increasing and continuous, and $\alpha(0)=0$.

Class of Nonlinear Process Systems

In this work, we consider a nonlinear process system with the following state-space description:

$$\dot{x}=f(x)+\sum_{i=1}^{m} g_i(x)\bar{u}_i+b(x)w \tag{1}$$

where $x \in R^n$ and $w \in R^{n_w}$ are the state and disturbance vectors, respectively. Due to the implementation strategy of the proposed safety-based DEMPC, the full input vector is divided into m input vectors where the ith manipulated input vector is denoted by $\bar{u}_i \in R^{m_i}$ for $i=1,\dots,m$, and each of these input vectors is bounded in a convex set U_i (i.e., $U_i := \{\bar{u}_i \in R^{m_i} : |\bar{u}_i| \leq \bar{u}_i^{\max}\}$, $i=1,\dots,m$, where the $\bar{u}_i^{\max}$, $i=1,\dots,m$, represent the magnitudes of the input constraints). The vector functions f, g_i, $i=1,\dots,m$, and b are assumed to be locally Lipschitz vector functions of their arguments. Furthermore, it is assumed that the state of the system of Eq. 1 is synchronously sampled at time instances $t_k=t_0+k\Delta$, $k=0,1,\dots$, where t_0 is the initial time. The vector w is bounded within the set $W := \{w \in R^{n_w} : |w| \leq \theta,\ \theta > 0\}$ (i.e., $w \in W$). We assume that the origin is an equilibrium point of the unforced nominal system (i.e., $f(0)=0$, $g_i(0)=0$, $i=1,\dots,m$, and $b(0)=0$).

Remark 1. *The systems of equations describing the behavior of many chemical process systems are of the form of Eq. 1. For those that are not, the distributed safety-based controller formulations developed in this work can still be utilized, but the closed-loop stability and feasibility results presented may not hold.*

Stabilizability Assumption

We consider systems of the form of Eq. 1 for which Assumption 1 (stabilizability assumption) holds.

Assumption 1. *There exists a locally Lipschitz feedback control law* $\bar{h}^T(x)=[\bar{h}_1(x)\ \dots\ \bar{h}_m(x)]$ *with* $\bar{h}(0)=0$ *for the nominal closed-loop system of Eq.* 1 *(i.e.,* $w(t) \equiv 0$*) that renders the origin of the nominal system of Eq.* 1 *under* $\bar{u}_i=\bar{h}_i(x)$, $i=1,\dots,m$, *asymptotically stable for all* $x \in D \subseteq R^n$*, where D is an open neighborhood of the origin, when applied continuously in the sense that there exists a continuously differentiable Lyapunov function V(x)*[22,23] *for the nominal closed-loop system and class* $\mathcal{K}$ *functions* $\alpha_i(\cdot)$, $i=1,2,3,4$, *such that the following inequalities hold*:

$$\alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|) \tag{2a}$$

$$\frac{\partial V(x)}{\partial x}\left(f(x)+\sum_{i=1}^{m} g_i(x)\bar{h}_i(x)\right) \leq -\alpha_3(|x|) \tag{2b}$$

$$\left|\frac{\partial V(x)}{\partial x}\right| \leq \alpha_4(|x|) \tag{2c}$$

$$\bar{h}_i(x) \in U_i, i=1,\dots,m \tag{2d}$$

The stability region of the closed-loop system under the feedback control law that meets Assumption 1 is defined as a level set of the Lyapunov function within D where Eq. 2 holds, and it is denoted by Ω_ρ.

By continuity, the local Lipschitz property assumed for the vector fields *f*, g_i, $i=1,\dots,m$, and *b*, the continuous differentiability property of the Lyapunov function *V(x)*, and taking into account that the manipulated inputs $\bar{u}_i$, $i=1,\dots,m$, and the disturbances w are bounded in convex sets, there exist positive constants L_w, L_x, $L_{\bar{u}_i}$, $i=1,\dots,m$, and M such that

$$\left| f(x)+\sum_{i=1}^{m} g_i(x)\bar{u}_i+b(x)w \right| \leq M \tag{3}$$

$$\left| \frac{\partial V}{\partial x} f(x)-\frac{\partial V}{\partial x} f(x') \right| \leq L_x|x-x'| \tag{4}$$

$$\left| \frac{\partial V}{\partial x} g_i(x)-\frac{\partial V}{\partial x} g_i(x') \right| \leq L_{\bar{u}_i}|x-x'|, \;\; i=1,\ldots,m \tag{5}$$

$$\left| \frac{\partial V}{\partial x} b(x) \right| \leq L_w \tag{6}$$

for all $x, x' \in \Omega_\rho$, $\bar{u}_i \in U_i$, $i=1,\ldots,m$, and $w \in W$.

Centralized Safety-Based Lyapunov-Based Economic Model Predictive Control

In a centralized Safety-Lyapunov-based economic model predictive controller (LEMPC), the control actions for all m input vectors are computed together in one optimization problem.[10] The centralized Safety-LEMPC design for the nonlinear system of Eq. 1 is formulated as follows:

$$\max_{\bar{u}_1,\ldots,\bar{u}_m,K_c \in S(\Delta)} \int_{t_k}^{t_{k+N}} L_e(\tilde{x}(\tau),\bar{u}_1(\tau),\ldots,\bar{u}_m(\tau))- \tag{7a}$$

$$\phi(\rho_{sp}-\tilde{\rho}(\tau))d\tau$$

$$\text{s.t. } \dot{\tilde{x}}(t)=f(\tilde{x}(t))+\sum_{i=1}^{m} g_i(\tilde{x}(t))\bar{u}_i \tag{7b}$$

$$\bar{u}_i(t) \in U_i, \;\; i=1,\ldots,m, \; \forall \quad t \in [t_k, t_{k+N}) \tag{7c}$$

$$\tilde{x}(t_k)=x(t_k) \tag{7d}$$

$$K_c(t) \geq 0, \; \forall \quad t \in [t_k, t_{k+N}) \tag{7e}$$

$$V(\tilde{x}(t)) \leq \tilde{\rho}(t), \; \forall \quad t \in [t_k, t_{k+N}) \tag{7f}$$

$$\frac{d\tilde{\rho}}{dt}=K_c(t)(\rho_{sp}-\tilde{\rho}(t)) \tag{7g}$$

$$\tilde{\rho}(t_k)=V(x(t_k)), \quad \text{if } \; x(t_k) \notin \Omega_{\rho_{sp}}$$

$$\tilde{\rho}(t_k)=\rho_{sp}, \quad \text{if } \; x(t_k) \in \Omega_{\rho_{sp}}$$

$$\frac{\partial V(x(t_k))}{\partial x}\left(\sum_{i=1}^{m} g_i(x(t_k))\bar{u}_i(t_k)\right) \tag{7h}$$

$$\leq \frac{\partial V(x(t_k))}{\partial x}\left(\sum_{i=1}^{m} g_i(x(t_k))\bar{h}_i(x(t_k))\right),$$

$$\text{if } \; x(t_k) \in \Omega_\rho/\Omega_{\tilde{\rho}_{sp}} \;\; \text{or} \; t_k > t_s$$

where the optimization variables are the piecewise-constant input trajectories $\bar{u}_1(t),\ldots,\bar{u}_m(t)$, over the prediction horizon $N\Delta$, as well as the piecewise-constant auxiliary variable $K_c(t)$ that plays a role in the safety-based constraints. L_e is a cost function that is determined based on economic considerations and is not required to have its minimum at a steady-state. The Safety-LEMPC formulation is a variation on the LEMPC formulation developed in Ref. 21 that has been augmented with safety-based constraints, and as a result it contains many of the standard constraints utilized in EMPC (e.g., a nominal process model for the predicted state $\tilde{x}$ (Eq. 7b), input constraints (Eq. 7c), and state feedback (Eq. 7d)). The time t_s represents a time after which the constraint of Eq. 7h is active for all subsequent times.

The motivation for adding safety-based constraints to this formulation is that situations may arise in which parts of Ω_ρ become unsafe to operate within due to, for example, prolonged closed-loop operation in a high-temperature region of state-space or expected effects from process disturbances. In such cases, a safety logic unit that determines the safest level set of V for the process to operate within may find that the closed-loop state should enter and remain within the set $\Omega_{\rho_{sp}}$, $\rho_{sp} < \rho$, to avoid unsafe scenarios. The safety level set $\Omega_{\rho_{sp}}$ is determined based on data on the probability of potential failures of process equipment, control system failures and measurement sampling time of the process state.[10] To drive the closed-loop state rapidly into $\Omega_{\rho_{sp}}$ while maintaining feasibility of the optimization problem, safety-based constraints (Eqs. 7e–7h) are added to the LEMPC, in addition to adding a penalty term $\phi(\rho_{sp}-\tilde{\rho}(\tau))$ to the objective function of Eq. 7a, that penalizes the difference between the the upper bound of the Lyapunov function $\tilde{\rho}(\tau)$ and ρ_{sp}. The function $\phi(\cdot)$ is selected based on the need to drive the process state into the safety region; for example, $\phi(\cdot)=|\cdot|^2$ is a potential function since its minimum occurs with $\rho_{sp}=\tilde{\rho}_{sp}$. When the penalty term is significant, the Safety-LEMPC will seek to find trajectories for $\bar{u}_i(t)$, $i=1,\ldots,m$, and $K_c(t)$ that drive the predicted closed-loop state into $\Omega_{\rho_{sp}}$ more quickly than without the penalty and dynamic constraints of Eqs. 7e–7h. Specifically, to decrease $\tilde{\rho}(t)$ from Eq. 7g toward ρ_{sp} to minimize the objective function including ϕ, a positive value of $K_c(t)$ (Eq. 7e) is computed for which inputs $\bar{u}_i(t)$, $i=1,\ldots,m$, are found to decrease $V(\tilde{x}(t))$ at a rate that allows Eq. 7f to be satisfied at all times given the rate of decrease of $\tilde{\rho}$ from Eq. 7g. The constraint of Eq. 7h (contractive constraint) forces the time derivative of the Lyapunov function under the Safety-LEMPC to be less than or equal to the time derivative of the Lyapunov function under the explicit stabilizing controller $\bar{h}(x)$. A subset of the safety level set $\Omega_{\tilde{\rho}_{sp}}$ activates the contractive constraint of Eq. 7h and should be chosen to make $\Omega_{\rho_{sp}}$ an invariant set.[10]

Safety-Distributed-LEMPC

For large-scale industrial nonlinear process systems, the time required to solve the centralized Safety-LEMPC design of Eq. 7 with the full process model and potentially tens or hundreds of optimization variables may be large. Therefore, a large sampling period in the LEMPC may be required. However, the closed-loop stability, feasibility, and safety-related proofs in Ref. 10 hold only for a sufficiently small sampling period and sufficiently small disturbances. Furthermore, even if the sampling period is sufficiently small to ensure that closed-loop stability within Ω_ρ is guaranteed, the length of the sampling period affects the minimum size of the level set of the stability region into which the closed-loop state is driven under repeated application of the contractive constraint.[21] This minimum size level set corresponds to the minimum size of a safe level set of operation that can be chosen within the stability region. To improve process safety, it is desirable to be able to make the safety region as small as possible (i.e., to be able to decrease the sampling period to a small value) to provide great flexibility in handling unsafe scenarios. When the time required to solve the centralized Safety-LEMPC is high, the computation time issue cannot be handled with decentralized control designs (i.e., multiple controllers utilize the same process model to compute subsets of the entire set of available control actions without communication between the controllers), because such designs may pose safety concerns since the controllers do not coordinate their actions.[6] However, a distributed Safety-LEMPC design (i.e., multiple controllers utilize the same process model to compute subsets of the entire set of available control actions but the controllers communicate) can be used to address the computation time concerns. Therefore, both sequential and iterative distributed Safety-LEMPC designs are proposed in this work.

Remark 2. *In this work, we assume that the upper bound on the disturbance is known, and thus we appeal to the conditions guaranteeing closed-loop stability and feasibility from Ref.* 10 *to motivate the use of distributed Safety-LEMPC. However, in industry, it is more common that the upper bound on the disturbance is estimated but not known, and in that case reducing the computation time of Safety-LEMPC using a distributed architecture has the safety benefit of allowing more frequent feedback to reduce the likelihood that the closed-loop state will exit the safety level set during a sampling period if a large disturbance potentially greater than the expected/typical bound affects the process. However, further discussion of this point is outside the scope of this work.*

Safety-Sequential-DLEMPC

A sequential design for a distributed Safety-LEMPC (Safety-S-DLEMPC) involves a hierarchy of m controllers, each of which solves the optimization problem in Eq. 7 but optimizes only $\bar{u}_i$ for a given $i \in \{1,\ldots,m\}$ and assumes a value of the other inputs. The designation "sequential" arises because the controllers are connected in series. The ith controller in the hierarchy (which we will refer to as Safety-S-DLEMPC i) assumes the values of $\bar{u}_p$, $p{=}1,\ldots,i{-}1$, throughout the prediction horizon calculated by the controllers higher up in the hierarchy, and the values $\bar{u}_p(t){=}\bar{h}_p(\tilde{x}(t_q))$, $p{=}i{+}1,\ldots,m$, $\forall\ t \in [t_q, t_{q+1})$, $q{=}k,\ldots,k{+}N{-}1$, for the rest of the control inputs when calculating $\bar{u}_i$. The optimal input trajectory for $\bar{u}_i$ determined for Safety-S-DLEMPC i at t_k is denoted by $\bar{u}_i^*(t|t_k)$, $t \in [t_k, t_{k+N})$, $i{=}1,\ldots,m$. Two considerations with respect to the distributed control design are: (1) whether it is necessary to solve for K_c in all m Safety-S-DLEMPC's, and (2) how to decide which inputs should be placed within $\bar{u}_1$, which within $\bar{u}_2$, and so on. To address these points, the main results of the proof of feasibility and closed-loop stability for the Safety-S-DLEMPC will be utilized (which will be rigorously presented in the Appendix).

To determine the number of distributed controllers that must solve for K_c, consider first the case that all m distributed controllers solve for K_c. First, Safety-S-DLEMPC 1 solves Eq. 7 for the piecewise-constant trajectories for $\bar{u}_1$ and K_c throughout the prediction horizon and sets $[\bar{u}_2(t),\ldots,\bar{u}_m(t)]$ to the corresponding $[\bar{h}_2(\tilde{x}(t_q)),\ldots,\bar{h}_m(\tilde{x}(t_q))]$, $\forall\ t \in [t_q, t_{q+1})$, $q{=}k,\ldots,k{+}N{-}1$. The input trajectory $\bar{u}_1(t){=}\bar{h}_1(\tilde{x}(t_q))$, $\forall\ t \in [t_q, t_{q+1})$, $q{=}k,\ldots,k{+}N{-}1$, and the gain $K_c=0$, $\forall\ t \in [t_k, t_{k+N})$, is a feasible solution to the resulting optimization problem because it satisfies all constraints. Therefore, there is always a feasible solution to Safety-S-DLEMPC 1. Now, consider that Safety-S-DLEMPC 2 receives the optimal trajectory of $\bar{u}_1$ throughout the prediction horizon from Safety-S-DLEMPC 1, sets $[\bar{u}_3(t),\ldots,\bar{u}_m(t)]{=}[\bar{h}_3\ (\tilde{x}(t_q)),\ldots,\bar{h}_m(\tilde{x}(t_q))]$, $\forall\ t \in [t_q, t_{q+1})$, $q{=}k,\ldots,k{+}N{-}1$, and solves for both the trajectory of $\bar{u}_2$ and of K_c. When $\bar{u}_2(t){=}\bar{h}_2(\tilde{x}(t_q))$, $\forall\ t \in [t_q, t_{q+1})$, $q{=}k,\ldots,k{+}N{-}1$, all inputs $\bar{u}_i$, $i{=}1,\ldots,m$, take the same values as they did for the optimal solution of Safety-S-DLEMPC 1 and the problem is feasible, assuming that K_c also takes the same trajectory as for that optimal solution. Therefore, a feasible solution to safety-S-DLEMPC 2 exists, which is the same as the feasible solution to Safety-S-DLEMPC 1. Recursively applying such arguments to Safety-S-DLEMPC 3 through Safety-S-DLEMPC m shows that each optimization problem in the Safety-S-DLEMPC structure has a feasible solution, and that the final solution satisfies Eqs. 7f and 7h with $\bar{u}_1^*(t|t_k),\ldots,\bar{u}_m^*(t|t_k)$, $\forall\ t \in [t_k, t_{k+N})$. When Eq. 7h is satisfied throughout a sampling period, then given a sufficiently small Δ and a sufficiently small θ, and due to Eq. 2b, the distributed Safety-S-DLEMPC architecture will cause the Lyapunov function value to decrease between two sampling periods when $x(t_k) \in \Omega_\rho/\Omega_{\bar{\rho}_{sp}}$ until it reaches the safety region.[21] Due to the safety penalty term in the objective function and safety-based constraints, there is a possibility that the rate at which $V(x)$ decreases along the closed-loop state trajectories under the Safety-S-DLEMPC paradigm may be faster than under a distributed LEMPC paradigm without safety-based constraints; however, in general, no guarantee can be made regarding this, and no proof can even be made regarding whether the rate of approach is the fastest rate that was obtained in any one of the m Safety-S-DLEMPC optimization problems.

The above discussion shows that if K_c is solved in all m Safety-S-DLEMPC's of the distributed architecture, then the Safety-S-DLEMPC is guaranteed to cause the closed-loop state to enter the safety region in finite time and to remain there. In the above discussion, it was noted that $K_c=0$ allowed a feasible solution in each Safety-S-DLEMPC, but potentially a less restrictive solution than if the value of K_c was allowed to be positive. Therefore, it is possible to set $K_c=0$ (i.e., remove K_c as an optimization variable) for some subset of the m Safety-S-DLEMPC's to reduce the number of optimization variables in some of these controllers when that provides a computation time benefit. The resulting control actions may not decrease the Lyapunov function as quickly as if K_c was optimized; however, if the inputs in the vector $\bar{u}_i$, for some $i \in \{1,\ldots,m\}$, have very little impact on the value of $V(\tilde{x})$ throughout the prediction horizon, the result of solving Safety-S-DLEMPC both including K_c as an optimization variable and the result with $K_c \equiv 0$ may produce similar results because the vector $\bar{u}_i$ is not able to affect the safety penalty term in the objective function highly. This implies that grouping inputs with regard to their impact on process safety may be beneficial for helping to reduce the number of optimization variables in some of the m Safety-S-DLEMPC problems. However, the full effects of the input partitioning and of setting $K_c=0$ in some optimization problems should be evaluated through closed-loop simulations.

A schematic of the sequential distributed safety-based LEMPC architecture with m controllers is shown in Figure 1. Safety-S-DLEMPC j calculates an input vector $\bar{u}_j$ where $\bar{u}_j^*(\tau|t_k)$, $\tau \in [t_k, t_{k+N})$ denotes the optimal solution of Safety-S-DLEMPC j at time t_k. Safety-S-DLEMPC j may calculate the gain K_c as well throughout the prediction horizon (the trajectory of the optimal gain throughout the prediction horizon calculated by Safety-S-DLEMPC j at time t_k is denoted by $K_c^*(\tau|t_k)$, $\tau \in [t_k, t_{k+N})$; it is not shown in Figure 1 because it is

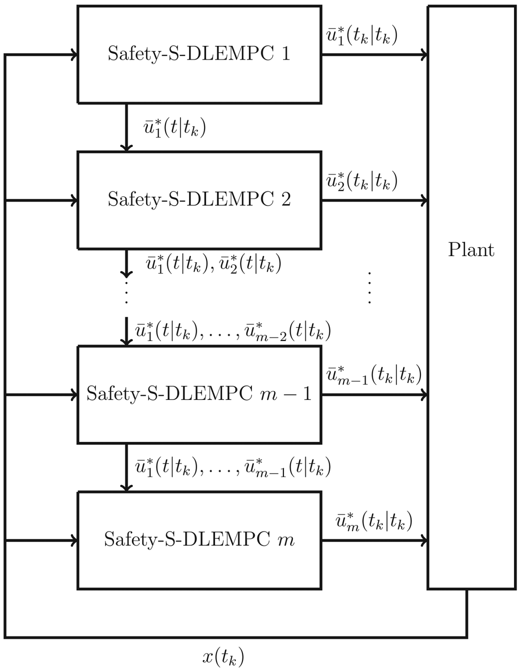

Figure 1. Block diagram of the Safety-S-DLEMPC scheme.

not communicated to the other Safety-S-DLEMPC controllers). The implementation strategy for the Safety-S-DLEMPC design is summarized as follows:

1. At t_k, all Safety-S-DLEMPC controllers receive a measurement of the current state $x(t_k)$ from the sensors. Go to step 2.
2. For $j=1$ to m:
   a. Safety-S-DLEMPC j receives the set of input trajectories $\bar{u}_p^*(\tau|t_k),\ \tau \in [t_k, t_{k+N}),\ p{=}1,\dots,j{-}1,$ from Safety-S-DLEMPC $j-1$ and assumes the input trajectories $\bar{u}_r(t){=}\bar{h}_r(\tilde{x}(t_q)),\ t \in [t_q, t_{q+1}),\ q{=}k,\dots,k+N{-}1$, for $r{=}j{+}1,\dots,m$. Based on these input trajectories and $x(t_k)$, Safety-S-DLEMPC j evaluates the input trajectory of $\bar{u}_j$ and, when $K_c \not\equiv 0$, the trajectory of the gain K_c. If $j \neq m$, go to step 2b else, go to step 2c.
   b. Safety-S-DLEMPC j sends $\bar{u}_p^*(\tau|t_k),\ \tau \in [t_k, t_{k+N}),$ $p{=}1,\dots,j$, to Safety-S-DLEMPC $j+1$. Go to step 2a.
   c. Go to step 3.
3. Each Safety-S-DLEMPC sends its optimal solution for the first sampling period of the prediction horizon to its actuator (i.e., all $u_i^*(t_k|t_k),\ i{=}1,\dots,m$, are implemented on the process). Go to step 4.
4. When a new state measurement is received at t_{k+1}, go to step 1 ($k \leftarrow k{+}1$).

The formulation of Safety-S-DLEMPC j is as follows:

$$\max_{\bar{u}_j, K_c \in S(\Delta)} \int_{t_k}^{t_{k+N}} L_e(\tilde{x}^j(\tau), \bar{u}_1(\tau), \dots, \bar{u}_m(\tau)) - \phi(\rho_{sp} - \tilde{\rho}(\tau))d\tau \tag{8a}$$

$$\text{s.t. } \dot{\tilde{x}}^j(t) = f(\tilde{x}^j(t)) + \sum_{i=1}^{m} g_i(\tilde{x}^j(t))\bar{u}_i(t) \tag{8b}$$

$$\bar{u}_j(t) \in U_j,\ \forall \quad t \in [t_k, t_{k+N}) \tag{8c}$$

$$\bar{u}_r(t) = \bar{h}_r(\tilde{x}^j(t_{k+q})),\ r{=}j{+}1,\dots,m,\ \forall \quad t \in [t_{k+q}, t_{k+q+1}),\quad q{=}0,\dots,N{-}1 \tag{8d}$$

$$\bar{u}_p(t) = \bar{u}_p^*(t|t_k),\ p{=}1,\dots,j{-}1,\ \forall\ t \in [t_k, t_{k+N}) \tag{8e}$$

$$\tilde{x}^j(t_k) = x(t_k) \tag{8f}$$

$$K_c(t) \geq 0,\ \forall \quad t \in [t_k, t_{k+N}) \tag{8g}$$

$$V(\tilde{x}^j(t)) \leq \tilde{\rho}(t),\ \forall \quad t \in [t_k, t_{k+N}) \tag{8h}$$

$$\frac{d\tilde{\rho}}{dt} = K_c(t)(\rho_{sp} - \tilde{\rho}(t)) \tag{8i}$$

$$\tilde{\rho}(t_k) = V(x(t_k)), \quad \text{if } x(t_k) \notin \Omega_{\rho_{sp}}$$

$$\tilde{\rho}(t_k) = \rho_{sp}, \quad \text{if } x(t_k) \in \Omega_{\rho_{sp}}$$

$$\frac{\partial V(x(t_k))}{\partial x}\left(\sum_{i=1}^{m} g_i(x(t_k))\bar{u}_i(t_k)\right) \tag{8j}$$

$$\leq \frac{\partial V(x(t_k))}{\partial x}\left(\sum_{i=1}^{m} g_i(x(t_k))\bar{h}_i(x(t_k))\right),$$

$$\text{if } x(t_k) \in \Omega_\rho / \Omega_{\tilde{\rho}_{sp}} \text{ or } t_k > t_s$$

where $\tilde{x}^j(t)$ denotes the predicted state trajectory under Safety-S-DLEMPC j. The values of the inputs $\bar{u}_r,\ r{=}j{+}1,\dots,m$, that have not yet been computed by a Safety-S-DLEMPC are set to the corresponding elements of $\bar{h}(x)$ applied in a sample-and-hold fashion by the constraint of Eq. 8d. The trajectories of $\bar{u}_p, p{=}1,\dots,j{-}1$, are set to the optimal trajectories $\bar{u}_p^*(t|t_k),\ t \in [t_k, t_{k+N})$, calculated by the Safety-S-DLEMPC's $p{=}1,\dots,j{-}1$, by the constraint of Eq. 8e. The other constraints of the optimization problem of Eq. 8 follow those in Eq. 7.

If K_c is set to zero in Safety-S-DLEMPC j, the controller will only solve for the input vector $\bar{u}_j$. As a result, the objective function of Eq. 8a will only include the economic cost $L_e(\tilde{x}^j(\tau), \bar{u}_1(\tau), \dots, \bar{u}_m(\tau))$. When $K_c(t) \equiv 0,\ \forall\ t \in [t_k, t_{k+N})$, the constraints of Eqs. 8h and 8i reduce to:

$$V(\tilde{x}^j(t)) \leq \tilde{\rho},\ \forall \quad t \in [t_k, t_{k+N})$$

$$\tilde{\rho} = V(x(t_k)), \quad \text{if } x(t_k) \notin \Omega_{\rho_{sp}}$$

$$\tilde{\rho} = \rho_{sp}, \quad \text{if } x(t_k) \in \Omega_{\rho_{sp}}$$

The contractive constraint of Eq. 8j will also be imposed in the Safety-S-DLEMPC that only solves for the input vector $\bar{u}_j$. This constraint guarantees that regardless of the value of K_c, the closed-loop state can be driven to the safety level set $\Omega_{\rho_{sp}}$ and maintained within that set thereafter (as will be shown in the proof of Theorem 1 in the Appendix).

We will now prove recursive feasibility and closed-loop stability of the Safety-S-DLEMPC implementation strategy, with the design of Safety-S-DLEMPC j following Eq. 8, and allowing for $K_c \equiv 0$ in any of the m Safety-S-DLEMPC's as desired. To proceed with this analysis, we first state a proposition that describes the closed-loop stability properties of the Lyapunov-based controller utilized in defining constraints of the Safety-S-DLEMPC design of Eq. 8.

Proposition 1. (c.f. Ref. 24). *Consider the trajectory* $\hat{x}(t)$ *of the system of Eq.* 1 *in closed-loop for a controller* $\bar{h}(x)$, *which satisfies the condition of Eq.* 2, *obtained by solving recursively*:

$$\dot{\hat{x}}(t)=f(\hat{x}(t))+\sum_{i=1}^{m} g_i(\hat{x}(t))\bar{h}_i(\hat{x}(t_k))+b(\hat{x}(t))w(t) \tag{9}$$

where $t\in[t_k,t_{k+1})$ *with* $k=0,1,\ldots$. *Let* $\Delta,\epsilon_w>0$ *and* $\rho>\rho_s>0$ *satisfy*:

$$-\alpha_3(\alpha_2^{-1}(\rho_s))+(L_x+\sum_{i=1}^{m} L_{\bar{u}_i}\bar{u}_i^{\max})M\Delta+L_w\theta\le-\epsilon_w/\Delta. \tag{10}$$

Then, if $\hat{x}(t_0)\in\Omega_\rho$ *and* $\rho_{\min}<\rho$ *where*

$$\rho_{\min}=\max\{V(x(t+\Delta)) : V(x(t))\le\rho_s\}, \tag{11}$$

the following inequality holds:

$$V(\hat{x}(t_k))\le\max\{V(\hat{x}(t_0))-k\epsilon_w,\rho_{\min}\}. \tag{12}$$

Proposition 1 guarantees several points regarding operation of the closed-loop system under $\bar{h}(x)$ implemented in sample-and-hold, namely that with a sufficiently small sampling period and bound on the disturbance such that Eq. 10 is satisfied: (1) If $\hat{x}(t_k)\in\Omega_\rho$, then $\hat{x}(t_{k+1})\in\Omega_\rho$, (2) if $\hat{x}(t_k)\in\Omega_\rho/\Omega_{\rho_s}$, then $V(\hat{x}(t_{k+1}))<V(\hat{x}(t_k))$, and (3) if $\hat{x}(t_k)$ enters Ω_{ρ_s}, $\hat{x}(t)$ obtained from recursively solving Eq. 9 remains within $\Omega_{\rho_{\min}}$ (ultimate boundedness of the closed-loop state of Eq. 9 within $\Omega_{\rho_{\min}}$). We note that $\rho_{\min}$ is defined in Eq. 11 with respect to the state $x(t)$ in Eq. 1, rather than with respect to the state under $\bar{h}(x)$ as in Eq. 9 (i.e., $\rho_{\min}$ is defined with respect to the worst-case deviation of $V(x)$ from ρ_s throughout a sampling period given Δ, θ, and $\bar{u}_i^{\max}$, $i=1,\ldots,m$, and does not assume any specific feedback control law in its definition).

The following theorem provides sufficient conditions under which the Safety-S-DLEMPC design of Eq. 8 guarantees recursive feasibility and closed-loop stability of the system of Eq. 1.

Theorem 1. *Consider the system of Eq.* 1 *in closed-loop under the sequential distributed safety-based LEMPC design of Eq.* 8 *based on a controller* $\bar{h}(x)$ *that satisfies the conditions of Eq.* 2. *Let* $\epsilon_w>0$, $\Delta>0$, $\rho>\rho_{sp}>\bar{\rho}_{sp}>\rho_s>0$ *satisfy*

$$-\alpha_3(\alpha_2^{-1}(\rho_s))+\left(L_x+\sum_{i=1}^{m} L_{\bar{u}_i}\bar{u}_i^{\max}\right)M\Delta+L_w\theta\le-\epsilon_w/\Delta. \tag{13}$$

and let $\bar{\rho}_{sp}$ *be defined such that if* $x(t_k)\in\Omega_{\bar{\rho}_{sp}}$, *then* $x(t_{k+1})\in\Omega_{\rho_{sp}}$. *If* $x(t_0)\in\Omega_\rho$, $\rho_{\min}<\rho$ *and* $N\ge1$, *then the state x(t) of the closed-loop system can be driven in a finite time to* $\Omega_{\rho_{sp}}$ *and then be bounded there, and after* t_s *the state x(t) of the closed-loop system is ultimately bounded in* $\Omega_{\rho_{\min}}$ *with* $\Omega_{\rho_{\min}}$ *defined as in Proposition 1.*

The proof of Theorem 1 can be found in the Appendix.

Remark 3. *The definition of* $\Omega_{\bar{\rho}_{sp}}$ *in Theorem 1 removes the direct correspondence between a constraint of the form in Eq.* 8h *and the proof of closed-loop stability that is made in other works on LEMPC (e.g., Ref.* 21*). To determine* $\bar{\rho}_{sp}$, *closed-loop simulations could be performed utilizing worst-case scenarios for the process model of Eq.* 1 *based on bounds on the disturbances and inputs in calculating the value of* $\Omega_{\bar{\rho}_{sp}}$. *In such a case, the constraint of Eq.* 8h *would not play a role in the closed-loop stability proof. An alternative implementation of the Safety-S-DLEMPC strategy would, however, allow a bound on* $\bar{\rho}_{sp}$ *to be determined based on satisfaction of a constraint of the form of Eq.* 8h. *Specifically, because the primary purpose of the constraint of Eq.* 8h *is in driving the closed-loop state to the safety region, once the closed-loop state enters the safety region, it is no longer necessary to utilize the safety-based constraints. Therefore,* Eqs. 8g–8j *can be replaced by the standard Mode 1 and Mode 2 constraints of Ref.* 21 *once the closed-loop state enters* $\Omega_{\rho_{sp}}$ *(and the penalty term in the objective function could be removed). The Mode 1 constraint would be the constraint of Eq.* 8h *but with the upper bound on the Lyapunov function fixed to* $\bar{\rho}_{sp}$, *and the activation condition being that* $x(t_k)\in\Omega_{\bar{\rho}_{sp}}$. *The Mode 2 constraint would be the constraint of Eq.* 8j. *With this modification, an explicit bound can be utilized on* $\bar{\rho}_{sp}$ *to prove that the closed-loop state is maintained within* $\Omega_{\rho_{sp}}$ *for all times after this region is entered, where the bound is based on satisfaction of the Mode 1 constraint requiring* $V(\tilde{x}^j)\le\bar{\rho}_{sp}$ *when* $x(t_k)\in\Omega_{\bar{\rho}_{sp}}$.

Remark 4. *The focus of this work is on distributed safety-based LEMPC designs; however, a safety-based tracking Lyapunov-based model predictive control (LMPC) design was proposed in Ref.* 1 *which takes the form of the centralized safety-based LEMPC design in Eq.* 7 *but with the contractive constraint of Eq.* 7h *enforced for all times, regardless of the location in state-space of the measurement of the state at* t_k. *Due to the similarity of this design to the centralized safety-based LEMPC design considered in this work, the results in this work, including the closed-loop stability and feasibility results, can be readily extended to the LMPC design considered in that work. For the sequential design, the same architecture and implementation strategy can be employed, with a similar formulation for the* $j-th$ *distributed controller as in Eq.* 8 *but with the contractive constraint always activated,* K_c *can be set to zero in some of the distributed controllers and inputs can be grouped based on their effect on* $V(\tilde{x})$, *and the results of Theorem 1 would hold for the resulting formulation, effectively with* $t_s=t_0$ *due to the repeated application of the contractive constraint.*

Safety-Iterative-DLEMPC

An alternative to the Safety-S-DLEMPC that may in some cases demonstrate improved performance compared to the Safety-S-DLEMPC (i.e., the implemented control actions may minimize the objective function more significantly) is a Safety-Iterative-DLEMPC (Safety-I-DLEMPC). As for the Safety-S-DLEMPC, there are *m* controllers, but unlike for the Safety-S-DLEMPC, all *m* controllers are solved simultaneously. In addition, the constraint of Eq. 8j in the *j*th Safety-S-DLEMPC, $j=1,\ldots,m$, is reformulated. The first time that the *m* controllers are solved, the *j*th controller (Safety-I-DLEMPC *j*) solves for $\bar{u}_j$ and K_c and assumes that $\bar{u}_z(t)$, $z\in\{1,\ldots,m\}$ but $z\ne j$, are equal to $\bar{h}_z(\tilde{x}^j(t_q))$, $\forall\ t\in[t_q,t_{q+1})$, $q=k,\ldots,k+N-1$. After the solutions of all *m* controllers have been obtained, the Safety-I-DLEMPC can cause the solutions of these *m* controllers to be implemented, or they can be exchanged. If the solutions are exchanged, each of the Safety-I-DLEMPC's is re-solved for $\bar{u}_j$ and K_c assuming that $\bar{u}_z$, $z\in\{1,\ldots,m\}$ but $z\ne j$, are equal to the trajectories of $\bar{u}_z$ returned by each of the *m* controllers at the prior iteration. In general, the number of iterations is an integer $c\in[1,\infty)$. When it is necessary to clearly specify the iteration number associated with the solution of the Safety-I-DLEMPC's below, we will refer to the solution to Safety-I-DLEMPC j at time t_k at iteration c as $\bar{u}^*_{j,c}(t|t_k)$ and $K_{c,c}(t|t_k)$, $\forall\ t \in [t_k, t_{k+N})$. Termination of the exchange of solutions (i.e., preventing further iterations at a given time t_k) can be triggered by various conditions. Examples of considerations that could be used are a fixed number of iterations or terminating when the value of the objective function evaluated for the predicted state of the nominal process under the inputs calculated by the m Safety-I-DLEMPC's at iteration c is no better than the cost function at iteration $c-1$ or is better by no more than a termination condition ϵ.

The proposed formulation of Safety-I-DLEMPC j, j=1, $\ldots, m$, at iteration c is as follows:

$$\max_{\bar{u}_j, K_c \in S(\Delta)} \int_{t_k}^{t_{k+N}} L_e(\tilde{x}^j(\tau), \bar{u}_1(\tau), \ldots, \bar{u}_m(\tau)) - \phi(\rho_{sp} - \tilde{\rho}(\tau)) d\tau \tag{14a}$$

$$\text{s.t. } \dot{\tilde{x}}^j(t) = f(\tilde{x}^j(t)) + \sum_{i=1}^{m} g_i(\tilde{x}^j(t)) \bar{u}_i(t) \tag{14b}$$

$$\bar{u}_j(t) \in U_j, \ \forall \quad t \in [t_k, t_{k+N}) \tag{14c}$$

$$\begin{aligned} \bar{u}_z(t) &= \bar{u}^*_{z,c-1}(t|t_k), \ z \in \{1, \ldots, m\}, \\ & z \neq j, \ \forall \quad t \in [t_{k+r}, t_{k+r+1}), \\ & r = 0, \ldots, N-1, \ c \geq 2 \end{aligned} \tag{14d}$$

$$\begin{aligned} \bar{u}_z(t) &= \bar{h}_z(\tilde{x}^j(t_{k+r})), \ z \in \{1, \ldots, m\}, \\ & z \neq j, \ \forall \quad t \in [t_{k+r}, t_{k+r+1}), \\ & r = 0, \ldots, N-1, \ c = 1 \end{aligned} \tag{14e}$$

$$\tilde{x}^j(t_k) = x(t_k) \tag{14f}$$

$$K_c(t) \geq 0, \ \forall \quad t \in [t_k, t_{k+N}) \tag{14g}$$

$$V(\tilde{x}^j(t)) \leq \tilde{\rho}(t), \ \forall \quad t \in [t_k, t_{k+N}) \tag{14h}$$

$$\begin{aligned} \frac{d\tilde{\rho}}{dt} &= K_c(t)(\rho_{sp} - \tilde{\rho}(t)) \\ \tilde{\rho}(t_k) &= V(x(t_k)), \quad \text{if } x(t_k) \notin \Omega_{\rho_{sp}} \\ \tilde{\rho}(t_k) &= \rho_{sp}, \quad \text{if } x(t_k) \in \Omega_{\rho_{sp}} \end{aligned} \tag{14i}$$

$$\begin{aligned} & \frac{\partial V(x(t_k))}{\partial x} g_j(x(t_k)) \bar{u}_j(t_k) \\ & \leq \frac{\partial V(x(t_k))}{\partial x} g_j(x(t_k)) \bar{h}_j(x(t_k)), \\ & \text{if } x(t_k) \in \Omega_\rho / \Omega_{\bar{\rho}_{sp}} \text{ or } t_k > t_s \end{aligned} \tag{14j}$$

where as for the Safety-S-DLEMPC, K_c may be set to zero in any of the m Safety-I-DLEMPC's as desired. The constraint of Eq. 14d sets the input trajectories $\bar{u}_z(t)$, $z \in \{1, \ldots, m\}$ where $z \neq j$, to their optimal solution in the previous iteration assuming $c > 1$, whereas the constraint of Eq. 14e sets the input trajectories to the corresponding Lyapunov-based control laws implemented in sample-and-hold when there is no prior iteration (i.e., $c = 1$). The notation of the other constraints follows that in Eq. 8.

The implementation strategy for the Safety-I-DLEMPC architecture is as follows:

1. At t_k, all m Safety-I-DLEMPC's receive a measurement of the current state $x(t_k)$ from the sensors. Go to step 2.
2. At iteration c ($c \geq 1$):
    a. If $c = 1$, Safety-I-DLEMPC j assumes $\bar{u}_z(t) = \bar{h}_z(\tilde{x}^j(t_{k+q}))$, $\forall\ t \in [t_{k+q}, t_{k+q+1})$, $z \in \{1, \ldots, m\}$ but $z \neq j, q$=0, $\ldots$, $N-1$. If $c > 1$, Safety-I-DLEMPC j assumes $\bar{u}_z(t) = \bar{u}^*_{z,c-1}(t|t_k)$, $\forall\ t \in [t_{k+r}, t_{k+r+1})$, r=0, $\ldots$, $N-1, z \in \{1, \ldots, m\}$ but $z \neq j$. Using these values, Safety-I-DLEMPC j evaluates both the optimal input trajectory $\bar{u}^*_{j,c}(\tau|t_k)$, and the optimal gain $K^*_{c,c}(\tau|t_k)$, $\tau \in [t_k, t_{k+N})$, or only the optimal input trajectory $\bar{u}^*_{j,c}(\tau|t_k)$, $\forall\ \tau \in [t_k, t_{k+N})$, when Safety-I-DLEMPC j sets the value of the gain K_c to zero. Go to step 2b.
    b. Both the constraint of Eq. 14h under $\bar{u}^*_{j,c}(\tau|t_k)$, $\forall\ \tau \in [t_k, t_{k+N})$, where j=1, $\ldots, m$ (i.e., $V(\tilde{x}^{tot}) \leq V(x(t_k))$, $\forall\ t \in [t_k, t_{k+N})$, if $x(t_k) \notin \Omega_{\rho_{sp}}$, or $V(\tilde{x}^{tot}(t)) \leq \rho_{sp}$, $\forall\ t \in [t_k, t_{k+N})$, if $x(t_k) \in \Omega_{\rho_{sp}}$, where $\tilde{x}^{tot}$ is the predicted state trajectory of the nominal system of Eq. 1 under $\bar{u}^*_{j,c}(\tau|t_k)$, $\forall\ \tau \in [t_k, t_{k+N})$, j=1, $\ldots, m$) and the iteration termination condition are evaluated. If Eq. 14h is not met or the iteration termination condition is met, go to step 2c. Else, go to step 2d.
    c. If $c > 1$, implement $[\bar{u}^*_1(t_k|t_k) \ \ldots \ \bar{u}^*_m(t_k|t_k)] = [\bar{u}^*_{1,c-1}(t_k|t_k) \ \ldots \ \bar{u}^*_{m,c-1}(t_k|t_k)]$. Else, implement $[\bar{u}^*_1(t_k|t_k) \ \ldots \ \bar{u}^*_m(t_k|t_k)] = [\bar{h}_1(x(t_k)) \ \ldots \ \bar{h}_m(x(t_k))]$. Go to step 3.
    d. The optimal input trajectories are exchanged between the Safety-I-DLEMPC controllers. The controller stores any required values related to the iteration termination condition (e.g., the calculated value of the objective function used in evaluating the iteration termination condition). Go to step 2a ($c \leftarrow c+1$).
3. When a new state measurement is received at t_{k+1}, go to step 1 ($k \leftarrow k+1$).

A schematic of the Safety-I-DLEMPC scheme is shown in Figure 2. At iteration c, Safety-I-DLEMPC j calculates the optimal solution $\bar{u}^*_{j,c}(t|t_k)$, $\forall\ t \in [t_k, t_{k+N})$, with the piecewise-constant gain $K^*_{c,c}(\tau|t_k)$, $\tau \in [t_k, t_{k+N})$, corresponding to that iteration. The values of $\bar{u}_1, \ldots, \bar{u}_m$ that are implemented on the process throughout the sampling period from t_k to t_{k+1} as a result of the above implementation strategy for the Safety-I-DLEMPC architecture are denoted by $u^*_1(t_k|t_k), \ldots, u^*_m(t_k|t_k)$.

As for the Safety-S-DLEMPC architecture, the number of controllers in which to solve for K_c and the method of partitioning the inputs into vectors $\bar{u}_1$, $\bar{u}_2$, and so on are important considerations, which rely on the above implementation strategy for the Safety-I-DLEMPC. It is noted that because the m Safety-I-DLEMPC's are solved independently, assuming in each controller different values of $\bar{u}_z$, $z \in \{1, \ldots, m\}$ but $z \neq j$, than are used by the other controllers, there is no guarantee that the constraint of Eq. 14h is satisfied for the nominal system of Eq. 1 under the set of trajectories $\bar{u}^*_{1,c}, (t|t_k) \ldots, \bar{u}^*_{m,c}(t|t_k)$, $t \in [t_k, t_{k+N})$, returned by the set of Safety-I-DLEMPC's at iteration c, even if $K_c = 0$ in Eq. 14h. However, satisfaction of Eq. 14h by this trajectory would be required for proving feasibility of the next iteration for the Safety-I-DLEMPC design. Therefore, it is necessary to check whether Eq. 14h is satisfied by the optimal control actions at the end of every iteration (i.e., compute the solution $\tilde{x}^{tot}$ to the

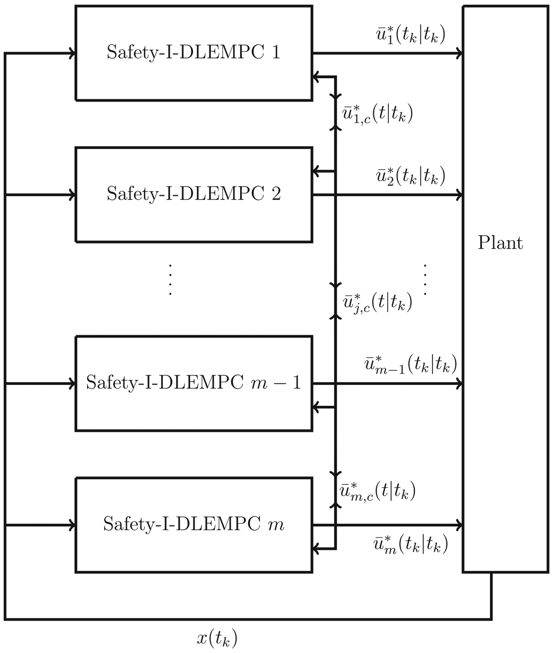

Figure 2. Block diagram of the Safety-I-DLEMPC scheme.

nominal system of Eq. 1 under $\bar{u}_{1,c}^*(t|t_k),\ldots,\bar{u}_{m,c}^*(t|t_k),\ \forall\ t\in[t_k,t_{k+N})$, and check whether $V(\tilde{x}^{tot})\leq V(x(t_k))$ if $x(t_k)\in\Omega_\rho/\Omega_{\rho_{sp}}$ or $V(\tilde{x}^{tot})\leq\rho_{sp}$ if $x(t_k)\in\Omega_{\rho_{sp}}$ throughout the prediction horizon). If this condition is satisfied, then the solution $\bar{u}_{1,c}^*(t|t_k),\ldots\ \bar{u}_{m,c}^*(t|t_k),\ \forall\ t\in[t_k,t_{k+N})$, at iteration c can be implemented or exchanged between the controllers and another iteration can begin. If Eq. 14h is not satisfied, then either the solution from iteration $c-1$ that met the condition should be implemented when $c>1$, or $\bar{h}(x(t_k))$ should be implemented if $c=1$. This strategy, which keeps the optimization problem of Eq. 14 feasible at each sampling time t_k, has been included in the above implementation strategy.

To determine whether K_c can be set to zero in some of the Safety-I-DLEMPC's given this implementation strategy, without negatively impacting closed-loop stability, to decrease the number of optimization variables in some of the Safety-I-DLEMPC's, we appeal to feasibility and closed-loop stability arguments (which will be presented in greater detail in the Appendix). First, consider iteration $c=1$. In this case, the jth Safety-I-DLEMPC assumes that $\bar{u}_z(t),\ z\in\{1,\ldots,m\}$ but $z\neq j$, is equal to $\bar{h}_z(\tilde{x}(t_q)),\ q{=}k,\ldots,k{+}N{-}1,\ \forall\ t\in[t_k,t_{k+N})$, and solves for $\bar{u}_j$ and K_c. The solution $\bar{u}_j(t){=}\bar{h}_i(\tilde{x}(t_q)),\ q{=}k,\ldots,k{+}N{-}1,\ \forall\ t\in[t_k,t_{k+N})$, with $K_c=0,\forall\ t\in[t_k,t_{k+N})$, is a feasible solution for the jth Safety-I-DLEMPC; therefore, there is always a feasible solution to all Safety-I-DLEMPC's for $c=1$. To ensure feasibility of subsequent iterations, there must be a feasible solution to the constraint of Eq. 14h at the next iteration. This is ensured, regardless of whether $K_c(t)\equiv0,\ \forall\ t\in[t_k,t_{k+N})$, if $V(\tilde{x}^{tot})$ is below a required bound throughout the prediction horizon at the prior iteration. The LEMPC implementation strategy ensures that no subsequent iterations are performed if this iteration condition is not met; therefore, all attempted iterations will have a feasible solution, regardless of whether $K_c(t)\equiv0,\ \forall\ t\in[t_k,t_{k+N})$, under the Safety-I-DLEMPC implementation strategy. It is important to ensure that a control action implemented by the Safety-I-DLEMPC implementation strategy will be stabilizing (i.e., $x(t)\in\Omega_\rho$ for all times, and $x(t)$ enters $\Omega_{\rho_{sp}}$ in finite time and remains in $\Omega_{\rho_{sp}}$ thereafter). If $\bar{u}_{1,c-1}^*(t_k|t_k),\ldots,\bar{u}_{m,c-1}^*(t|t_k),\ \forall\ t\in[t_k,t_{k+N})$, is implemented, a summation of the constraints of Eq. 14j for all m Safety-I-DLEMPC's reveals that $V(x(t_{k+1}))<V(x(t_k))$, as will be demonstrated in the Appendix. If instead $\bar{h}(x)$ is implemented in sample-and-hold, $V(x(t_{k+1}))<V(x(t_k))$ from Proposition 1. If $x(t_k)\in\Omega_{\bar{\rho}_{sp}}$, then under either $u_{1,c-1}^*(t_k|t_k)$, $\ldots,u_{m,c-1}^*(t_k|t_k)$, or $\bar{h}(x)$ implemented in sample-and-hold, $x(t_{k+1})\in\Omega_{\rho_{sp}}$ from the definition of $\Omega_{\bar{\rho}_{sp}}$. This establishes that closed-loop stability is maintained under the Safety-I-DLEMPC implementation strategy because this implementation strategy ensures that the implemented control actions satisfy both Eqs. 14h and 14j. Furthermore, this stability proof does not depend on the value of K_c in each controller, and $K_c=0,\ \forall\ t\in[t_k,t_{k+N})$, is guaranteed to provide a feasible solution to the Safety-I-DLEMPC at $c=1$ and all subsequent attempted iterations. Therefore, it is possible to set K_c to zero in some of the Safety-I-DLEMPC optimization problems to reduce the number of decision variables in these problems. It may be helpful to partition the inputs with a large effect on $V(\tilde{x})$ into some $\bar{u}_j$ vectors and those with more minimal effect into others, so that the Safety-I-DLEMPC's for which solving for $\bar{u}_i$ may have less effect on the safety penalty term can be selected to have $K_c\equiv0$. However, the effects of partitioning and of setting $K_c\equiv0$ in some controllers should be assessed with closed-loop simulations.

We will now provide the conditions that guarantee closed-loop stability of a nonlinear process under the Safety-I-DLEMPC implementation strategy, as well as conditions that guarantee feasibility of the Safety-I-DLEMPC optimization problem of Eq. 14 at a given iteration.

Theorem 2. *Consider the system of Eq.* 1 *in closed-loop under the implementation strategy (steps 1–3) of the iterative distributed safety-based LEMPC design of Eq.* 14 *based on a controller h(x) that satisfies the conditions of Eq.* 2. *Let* $\epsilon_w>0,\ \Delta>0,\ \rho>\rho_{sp}>\bar{\rho}_{sp}>\rho_s>0$ *satisfy the constraint of Eq.* 13, *with* $\bar{\rho}_{sp}$ *defined such that if* $x(t_k)\in\Omega_{\bar{\rho}_{sp}}$, *then* $x(t_{k+1})\in\Omega_{\rho_{sp}}$. *For any* $N\geq1$ *and* $c\geq1$, *if* $x(t_0)\in\Omega_\rho$, $\rho_{\min}<\rho$, *then the state x(t) of the closed-loop system can be driven in a finite time to* $\Omega_{\rho_{sp}}$ *and then be bounded there, and after* t_s *the state x(t) of the closed-loop system is ultimately bounded in* $\Omega_{\rho_{\min}}$ *with* $\Omega_{\rho_{\min}}$ *defined as in Proposition 1.*

The proof of Theorem 2 can be found in the Appendix.

Remark 5. *For the proof of closed-loop stability and feasibility of the Safety-I-DLEMPC design, similar comments as in Remark 3 can be made. First, the constraint of Eq.* 14h *is not utilized in the proof of closed-loop stability. Also, once the closed-loop state enters the safety region, the Safety-I-DLEMPC can be modified to no longer include the penalty term in the objective function or safety-based constraints, but can instead by formulated like an iterative distributed LEMPC with the constraints of Eqs.* 14g–14i *replaced by the constraint of Eq.* 14h *but with a static upper bound of* $\bar{\rho}_{sp}$ *on the Lyapunov function, and the constraint activated whenever the closed-loop state is within* $\Omega_{\bar{\rho}_{sp}}$. *The same implementation strategy could continue to be used after this modification (e.g., checking the value of* $V(\tilde{x}^{tot}(t))$ *between iterations). This discussion brings up two important points regarding the Safety-I-DLEMPC closed-loop stability and feasibility proof*:

1. *Although satisfaction of the condition on* $V(\tilde{x}^{tot}(t))$ *is not directly utilized for proving closed-loop stability,*

checking the condition on $V(\tilde{x}^{tot}(t))$ was shown through the proof of feasibility to be important in ensuring that there was a feasible solution to Safety-I-DLEMPC j, $j=1,\ldots,m$, at each iteration attempted.

2. *Because only the slight modifications discussed in this remark to Eqs.* 14a and 14g–14i *are required to transform the Safety-I-DLEMPC into an iterative distributed LEMPC (i.e., not including safety-based constraints), the implementation strategy proposed above with the resulting guarantees on closed-loop stability within $\Omega_{\rho_{sp}}$ and feasibility of the optimization problem at every sampling time for $c=1$ and at subsequent sampling times when the condition on $V(\tilde{x}^{tot}(t))$ is met would also hold. This is significant because it is the first closed-loop stability result for iterative distributed LEMPC in general.*

Remark 6. *Due to the similarity between the centralized safety-LEMPC and safety-LMPC formulations as mentioned in Remark 4, an iterative distributed design for the safety-LMPC formulation, with the implementation strategy and associated closed-loop stability and feasibility proofs, would follow that of this section, with $t_s=t_0$.*

Remark 7. *An assumption throughout this work is that the time to calculate the solutions to the distributed safety-LEMPC problems is much less than the sampling time such that the calculations can be considered instantaneous. When such short time scales are assumed for the computations, an alternative to terminating the iterations as soon as the condition on $V(\tilde{x}^{tot}(t))$ is not met would be to re-perform optimization iteration c with different initial guesses to try to meet the condition on $V(\tilde{x}^{tot}(t))$ at the iteration and potentially improve the optimality of the implemented solutions from a safety and economics perspective.*

Application to a Chemical Process Example

In this section, we demonstrate the advantages of the proposed Safety-DLEMPC schemes over the centralized Safety-LEMPC of Eq. 7 by applying them to a benchmark catalytic reactor example. The closed-loop economic performance and the on-line computation time needed to solve the three Safety-LEMPC optimization problems are the key performance metrics. A chemical process example (catalytic reactor) is considered in which the oxidation of ethylene to ethylene oxide takes place in a non-isothermal continuous stirred tank reactor (CSTR) according to the following reactions:

$$C_2H_4+\frac{1}{2}O_2\xrightarrow{r_1}C_2H_4O \tag{R1}$$

$$C_2H_4+3O_2\xrightarrow{r_2}2CO_2+2H_2O \tag{R2}$$

$$C_2H_4O+\frac{5}{2}O_2\xrightarrow{r_3}2CO_2+2H_2O \tag{R3}$$

To remove the heat generated by the exothermic reactions, a cooling jacket is used. The dimensionless material and energy balances for the catalytic reactor are developed in Ref. 3 where the rate laws for the reactions use the nonlinear Arrhenius reaction in Ref. 4. The dimensionless mass and energy balances for this process are described by the following equations[3]:

$$\frac{dx_1(t)}{dt}=u_1(1-x_1x_4) \tag{15a}$$

$$\frac{dx_2(t)}{dt}=u_1(u_2-x_2x_4)-A_1e^{\frac{\gamma_1}{x_4}}(x_2x_4)^{0.5} -A_2e^{\frac{\gamma_2}{x_4}}(x_2x_4)^{0.25} \tag{15b}$$

$$\frac{dx_3(t)}{dt}=-u_1x_3x_4+A_1e^{\frac{\gamma_1}{x_4}}(x_2x_4)^{0.5}-A_3e^{\frac{\gamma_3}{x_4}}(x_3x_4)^{0.5} \tag{15c}$$

$$\frac{dx_4(t)}{dt}=\frac{u_1}{x_1}(1-x_4)+\frac{B_1}{x_1}e^{\frac{\gamma_1}{x_4}}(x_2x_4)^{0.5} +\frac{B_2}{x_1}e^{\frac{\gamma_2}{x_4}}(x_2x_4)^{0.25}+\frac{B_3}{x_1}e^{\frac{\gamma_3}{x_4}}(x_3x_4)^{0.5}-\frac{B_4}{x_1}(x_4-u_3) \tag{15d}$$

The resulting dimensionless dynamic model of this reactor has four states x_1, x_2, x_3, and x_4 and three manipulated inputs u_1, u_2, and u_3. The four dimensionless states represent the reactor gas mixture density, ethylene concentration, ethylene oxide concentration, and temperature in the reactor, respectively. The three dimensionless inputs u_1, u_2, and u_3 of the reactor are the feed volumetric flow rate, the concentration of ethylene in the feed, and the coolant temperature, respectively. The values of the parameters of this model are presented in Table 1. Due to the physical constraints on the control actuators, the manipulated inputs are bounded (i.e., $u_1\in[0.0704,\ 0.7042]$, $u_2\in[0.2465,\ 2.4648]$, $u_3\in[0.6,\ 1.1]$). The economic performance index of the catalytic reactor is the average yield of ethylene oxide where the yield is defined by:

$$Y(t_f)=\frac{\int_{t_0}^{t_f}u_1(\tau)x_3(\tau)x_4(\tau)\ d\tau}{\int_{t_0}^{t_f}u_1(\tau)u_2(\tau)\ d\tau} \tag{16}$$

where t_f is the operating period. A limitation on the amount of reactant material that may be fed to the reactor is fixed by the following integral material constraint:

$$\frac{1}{t_f}\int_{t_0}^{t_f}u_1(\tau)u_2(\tau)\ d\tau=0.175. \tag{17}$$

Since the denominator of Eq. 16 is fixed over the length of operation, the various Safety-LEMPC schemes considered in this work will maximize the following stage cost:

$$L_e(x,u)=u_1x_3x_4. \tag{18}$$

The dynamic model of the catalytic reactor has an open-loop asymptotically stable steady-state that satisfies the integral material constraint of Eq. 17 with $x_s^T=[x_{1s}\ x_{2s}\ x_{3s}\ x_{4s}]=[0.998\ \ 0.424\ \ 0.032\ \ 1.002]$ which corresponds to the steady-state input $u_s^T=[0.35\ \ 0.5\ \ 1.0]$. The contractive constraint of Eqs. 7h, 8j, and 14j was not imposed in all the simulations below since closed-loop stability under the various Safety-LEMPC schemes is not an issue for the region of operation considered for the dynamic model of this reactor. To determine the safety level set, a characterization of the closed-loop stability region Ω_ρ of the dynamic model of the reactor is

Table 1. Values of the Dimensionless Parameters of the Ethylene Oxidation CSTR

$A_1=92.8$	$B_2=10.39$	$\gamma_2=-7.12$
$A_2=12.66$	$B_3=2170.57$	$\gamma_3=-11.07$
$A_3=2412.71$	$B_4=7.02$	
$B_1=7.32$	$\gamma_1=-8.13$	

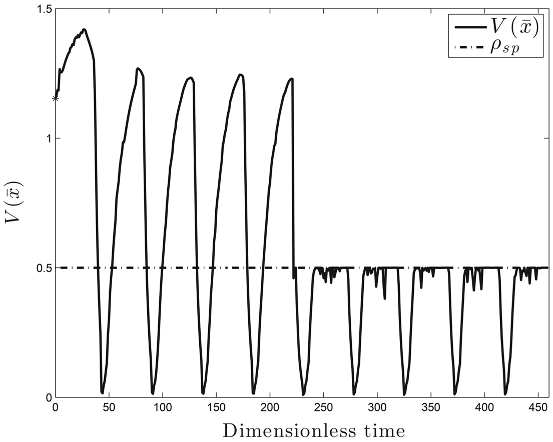

Figure 3. Evolution of the Lyapunov function value of the closed-loop state under the centralized Safety-LEMPC.

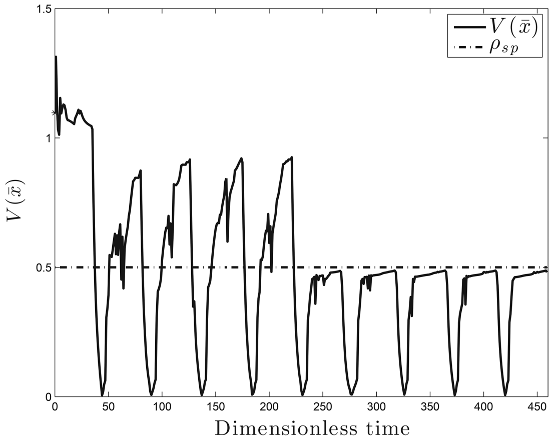

Figure 4. Evolution of the Lyapunov function value of the closed-loop state under the Safety-I-DLEMPC.

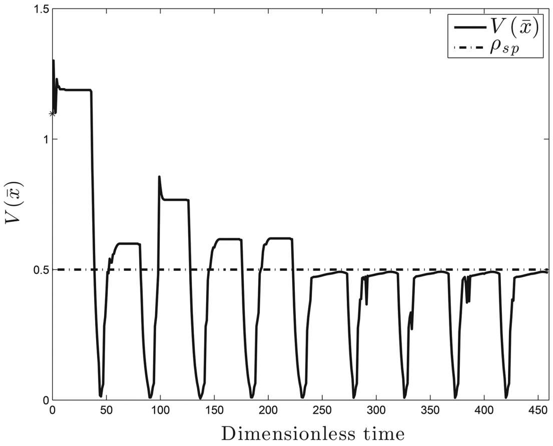

Figure 5. Evolution of the Lyapunov function value of the closed-loop state under the Safety-S-DLEMPC.

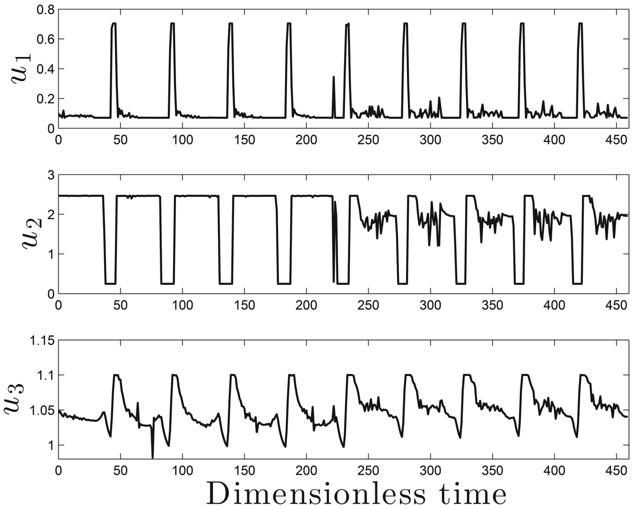

Figure 6. Input trajectories computed by the centralized Safety-LEMPC.

required. To estimate the stability region Ω_ρ, a PI controller $h^T(x)=[h_1(x)\ h_2(x)\ h_3(x)]$ is implemented in a sample-and-hold fashion for the three manipulated inputs (i.e., $h_a(x)=K_{P_a}(x_a-x_{as})+\frac{1}{\tau_a}\int_0^t(x_a-x_{as})dt$, $a=1,2,3$, where $K_{P_1}=3.0$, $K_{P_2}=0.105$, $K_{P_3}=0.1$, $\tau_1=0.00001$, $\tau_2=0.0002081$, and $\tau_3=0.005$). The centralized and distributed Safety-LEMPC schemes are implemented with a shrinking prediction horizon that covers the entire operating window $t_p=47$; specifically, at the beginning of the lth operating window, the prediction horizon was set to t_p/Δ and the horizon was decreased by one at each sampling period where $\Delta=1$. At the beginning of the $(l+1)^{th}$ operating window where $l=0,\ldots,9$, the prediction horizon is reinitialized to t_p/Δ. To satisfy the material constraint of Eq. 17, this constraint is imposed over the ten operating windows (i.e., the average molar flow rate of ethylene must be equal to 0.175 at the end of each operating interval of length t_p). The dynamic model of the catalytic reactor is simulated numerically by using the explicit Euler method with a step size of 10^{-5}, while the step size used for the model within the Safety-LEMPC optimization problems is 0.0005. All the optimization problems were solved using the interior-point solver Ipopt.[25]

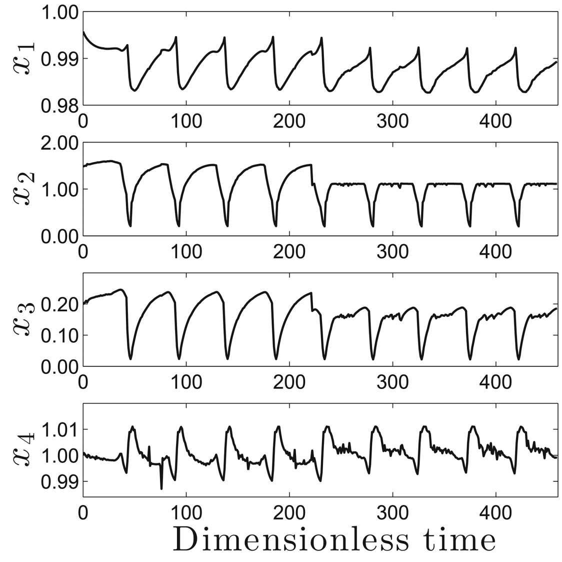

Figure 7. Process state trajectories under the centralized Safety-LEMPC.

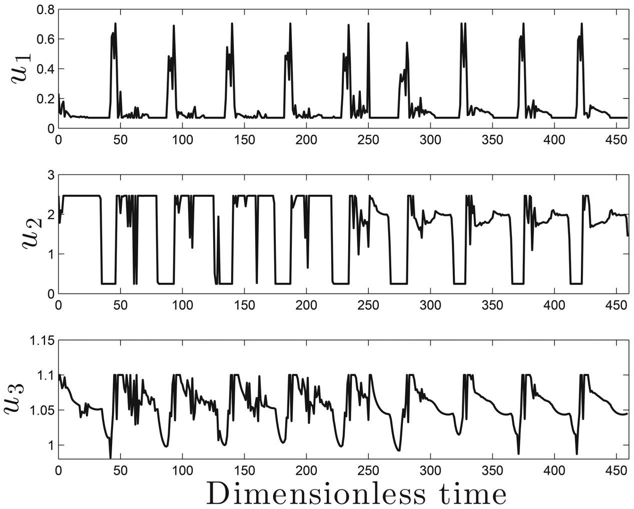

Figure 8. Input trajectories computed by the Safety-I-DLEMPC.

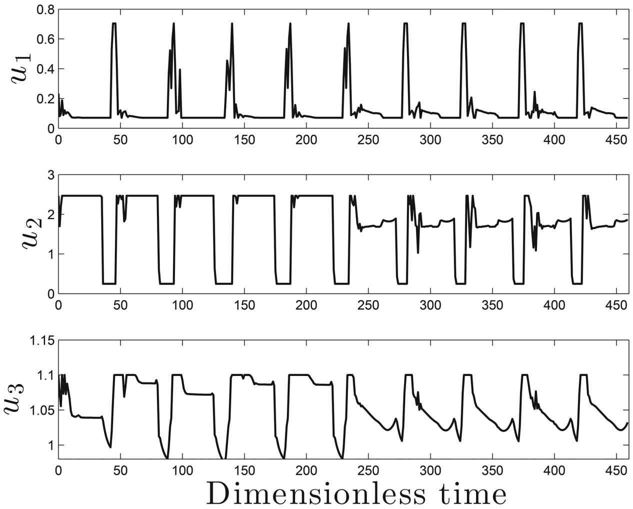

Figure 10. Input trajectories computed by the Safety-S-DLEMPC.

We use a quadratic Lyapunov function of the form $V(\bar{x})=\bar{x}^TP\bar{x}$ to estimate the stability region of the closed-loop system under $h(x)$ where $P=\text{diag}([1\ \ 1\ \ 1\ \ 1])$. The notation $\bar{x}$ denotes the process state vector in deviation form (i.e., $\bar{x}=x-x_s$). The safety level set $\Omega_{\rho_{sp}}$ is chosen to operate the closed-loop process in a relatively small region around the steady-state to avoid the boundary of the stability region. Following this technique and using the Lyapunov function $V(\bar{x})$, the values of ρ and ρ_{sp} were chosen to be 2.1 and 0.5, respectively. As a result of the integral material constraint of Eq. 17, the inputs u_1 and u_2 are optimized by one Safety-DLEMPC (i.e., $\bar{u}_1^T=[u_1\ \ u_2]$), as well as K_c, while only u_3 is computed by another (i.e., $\bar{u}_3=u_3$ with $K_c\equiv 0$) for both iterative and distributed Safety-DLEMPC's. The termination condition for the Safety-I-DLEMPC algorithm was to stop iterating the optimization problem when the cost function at the current iteration is less than or equal to the cost function at the previous iteration. In this example, the condition on the value of $V(\tilde{x}^{tot})$ along the closed-loop state trajectories of the nominal system under the control actions calculated by the two iterative distributed controllers was not checked between iterations, but no issues with feasibility occurred during the iterations performed. Ipopt was forced to stop optimizing the problem after 100 iterations to take real-time computation considerations into account. The computation time for the Safety-S-DLEMPC is evaluated as the sum of the computation times of Safety-S-DLEMPC 1 and Safety-S-DLEMPC 2 at each sampling time because the distributed controllers are evaluated in sequence which means that the minimal time to obtain a solution is the sum of the evaluated times of all controllers. However, the computation time for one iteration of the Safety-I-DLEMPC is computed as the maximum computation time of the two optimization problems because the distributed controllers are evaluated in parallel which implies that the minimal time to obtain a solution is the largest computation time among all the Safety-I-DLEMPC controllers.

In these simulations, the catalytic reactor was initiated far from $\Omega_{\rho_{sp}}$ with $x^T(t_0)=[0.9818\ \ 1.4566\ \ 0.1987\ \ 1.0523]$

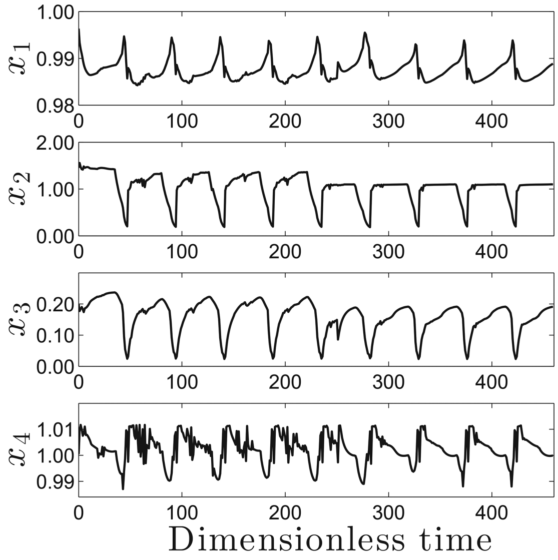

Figure 9. Process state trajectories under the Safety-I-DLEMPC.

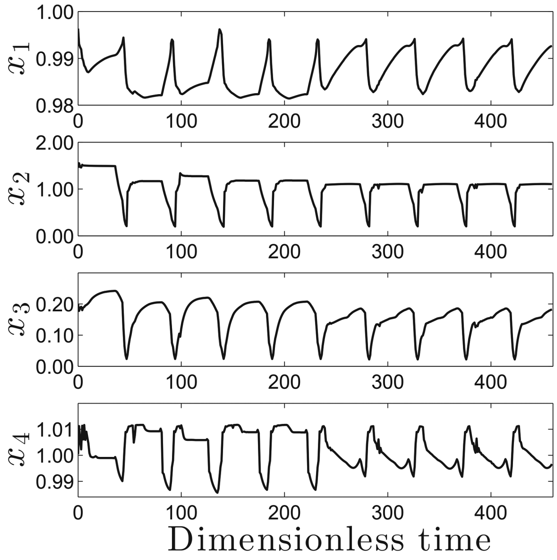

Figure 11. Process state trajectories under the Safety-S-DLEMPC.

Table 2. The Average Yield and Computation Time Under the Safety-LEMPC Strategies

Strategy	Yield (%)	Computation time (s)
Safety-S-DLEMPC	9.85	6.64
Safety-I-DLEMPC	9.94	5.59
Centralized Safety-LEMPC	10.15	16.87

(i.e., $V(x(t_0))=1.09 > \rho_{sp}=0.5$). Starting at $t_k=222$, the safety logic unit requests the closed-loop state to move toward the safety level set under the centralized and distributed Safety-LEMPC schemes. Figures 3–5 show the Lyapunov function value of the closed-loop states under the centralized Safety-LEMPC and iterative and sequential Safety-DLEMPC controllers, respectively. From these figures, the closed-loop states under the three controllers successfully entered the safety level set after one sampling time (i.e., $V(t_k) < \rho_{sp}$ where $t_k=223$). Figures 6–11 represent the closed-loop state trajectories and the manipulated input trajectories of the centralized Safety-LEMPC and iterative and sequential Safety-DLEMPC controllers, respectively. As in Ref. 14, the centralized Safety-LEMPC, the Safety-S-DLEMPC, and the Safety-I-DLEMPC dictate periodic operation (i.e., the ethylene is distributed in a non-uniform fashion with respect to time) to maximize the yield of ethylene oxide. The input trajectories u_1 and u_2 satisfied the material constraint of Eq. 17 under all Safety-LEMPC schemes. Figures 7 and 9, and 11 show that the closed-loop trajectories under all the Safety-LEMPC schemes changed after the safety-based constraints are activated at $t_k=222$ while periodic operation is still maintained. Due to the nonlinearity and non-convexity of the optimization problem, the Safety-I-DLEMPC under the termination condition described above terminates most of the time after the second iteration so that the $c=1$ solution is applied (i.e., the cost function at the first iteration is generally greater than or equal to the cost function at the second iteration). Table 2 shows the average yield and average computation time required to solve each of the three optimization problems over the ten operating windows. From Table 2, the average yield of ethylene oxide under the centralized safety-LEMPC and distributed (iterative and sequential) safety-LEMPC's is similar. Both the iterative and the sequential Safety-DLEMPC's require over 60% less computation time than that required to solve the centralized safety-LEMPC of Eq. 7. Additionally, the average yield of ethylene oxide over ten operating periods under the PI controllers is 5.34%; the average yield under the centralized safety-LEMPC is 70% better than that under the PI controllers.

Remark 8. *Even though the dynamic model of the reactor of Eq.* 15 *does not explicitly follow the class of systems of Eq.* 1 *due to the bilinear term in the right hand side of the second differential equation (i.e.,* $u_1(u_2-x_2x_4)$*), the system can be reformulated to be in the class of systems of Eq.* 1*. Since the manipulated input* u_2 *only appears in that term and the Safety-DLEMPC 1 solves for the inputs* u_1 *and* u_2 *together in one optimization problem due to the material constraint of Eq.* 17*, a new variable* $u_4=u_1u_2$ *can be introduced to make the process model be in the class of systems of Eq.* 1 *(input affine with inputs* u_1*,* u_3*, and* u_4*). Furthermore, as is demonstrated above, the distributed control methodology of this work performed well for this example.*

Conclusion

In this work, sequential and iterative Safety-DLEMPC schemes were proposed as alternatives to centralized Safety-LEMPC that may have less on-line computation time while achieving similar closed-loop performance and safety constraints satisfaction. An implementation strategy and mathematical formulation for the Safety-Sequential-DLEMPC design and the Safety-Iterative-DLEMPC design were developed. The main objective of the two distributed Safety-LEMPC schemes is to improve the computation time with respect to the centralized Safety-LEMPC while maintaining similar closed-loop performance. For a sufficiently small sampling period, proofs of recursive feasibility and closed-loop stability of a class of nonlinear systems under the Safety-S-DLEMPC and Safety-I-DLEMPC formulations in the presence of uncertainty were given. Using a catalytic reactor example, the proposed iterative and sequential Safety-DLEMPC strategies were able to yield comparable closed-loop performance while significantly decreasing the on-line computation time compared to that required to solve the centralized Safety-LEMPC. This illustrates that distributed implementation may allow Safety-LEMPC to be implemented on processes where the computation time of the centralized implementation strategy exceeds the controller sampling time.

Acknowledgment

Financial support from the National Science Foundation and the Department of Energy is gratefully acknowledged.

Literature Cited

1. Albalawi F, Durand H, Alanqar A, Christofides PD. Achieving operational process safety via model predictive control. *J Loss Prevent Process Ind*. In press.
2. Khan FI, Amyotte PR. How to make inherent safety practice a reality. *Canad J Chem Eng*. 2003;81:2–16.
3. Özgülşen F, Adomaitis RA, Çinar A. A numerical method for determining optimal parameter values in forced periodic operation. *Chem Eng Sci*. 1992;47:605–613.
4. Alfani F, Carberry JJ. An exploratory kinetic study of ethylene oxidation over an unmoderated supported silver catalyst. *La Chimica e L'Industria*. 1970;52:1192–1196.
5. Leveson NG. *Safeware: System Safety and Computers*. Reading, MA: Addison-Wesley Publishing Company, 1995.
6. Leveson N. A new accident model for engineering safer systems. *Safe Sci*. 2004;42:237–270.
7. Chen X, Heidarinejad M, Liu J, Christofides PD. Distributed economic MPC: application to a nonlinear chemical process network. *J Process Control*. 2012;22:689–699.
8. Christofides PD, Scattolini R, Muñoz de la Peña D, Liu J. Distributed model predictive control: a tutorial review and future research directions. *Comput Chem Eng*. 2013;51:21–41.
9. Chilin D, Liu J, Muñoz de la Peña D, Christofides PD, Davis JF. Detection, isolation and handling of actuator faults in distributed model predictive control systems. *J Process Control*. 2010;20:1059–1075.
10. Albalawi F, Alanqar A, Durand H, Christofides PD. A feedback control framework for safe and economically-optimal operation of nonlinear processes. *AIChE J*. 2016;62:2391–2409.
11. Christofides PD, Liu J, Muñoz de la Peña D. *Networked and Distributed Predictive Control: Methods and Nonlinear Process Network Applications. Advances in Industrial Control Series*. London, UK: Springer-Verlag, 2011.
12. Marlin T. *Operability in Process Design: Achieving Safe, Profitable, and Robust Process Operations*. Ontario, Canada: McMaster University, 2012.
13. Scattolini R. Architectures for distributed and hierarchical model predictive control—a review. *J Process Control*. 2009;19:723–731.
14. Anderson TL, Ellis M, Christofides PD. Distributed economic model predictive control of a catalytic reactor: evaluation of sequential and

iterative architectures. In: *Proceedings of IFAC International Symposium on Advanced Control of Chemical Processes*, Whistler, Canada. 2015:26–31.

15. Venkat AN, Hiskens IA, Rawlings JB, Wright SJ. Distributed MPC strategies with application to power system automatic generation control. *IEEE Trans Control Syst Technol.* 2008;16:1192–1206.
16. The 100 Largest Losses 1974-2015: Large Property Damage Losses in the Hydrocarbon Industry. Technical report, Marsh & McLennan Companies Inc., 2016.
17. Leveson NG, Stephanopoulos G. A system-theoretic, control-inspired view and approach to process safety. *AIChE J.* 2014;60:2–14.
18. Venkatasubramanian V. Systemic failures: challenges and opportunities in risk management in complex systems. *AIChE J.* 2011;57:2–9.
19. Heikkilä A-M, Hurme M, Järveläinen M. Safety considerations in process synthesis. *Comput Chem Eng.* 1996;20:S115–S120.
20. Gentile M, Rogers WJ, Mannan MS. Development of an inherent safety index based on fuzzy logic. *AIChE J.* 2003;49:959–968.
21. Heidarinejad M, Liu J, Christofides PD. Economic model predictive control of nonlinear process systems using Lyapunov techniques. *AIChE J.* 2012;58:855–870.
22. Khalil HK. *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.
23. Massera JL. Contributions to stability theory. *Annals of Mathematics.* 1956;64:182–206.
24. Muñoz de la Peña D, Christofides PD. Lyapunov-based model predictive control of nonlinear systems subject to data losses. *IEEE Trans Automat Control.* 2008;53:2076–2089.
25. Wächter A, Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math Program.* 2006;106:25–57.
26. Liu J, Chen X, Muñoz de la Peña D, Christofides PD. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. *AIChE J.* 2010;56:2137–2149.

Appendix

Proof of Theorem 1. The proof consists of three parts. We first prove that the optimization problem of Eq. 8 is recursively feasible for all $x(t_0) \in \Omega_\rho$. Subsequently, we prove that under the Safety-S-DLEMPC design of Eq. 8, the closed-loop state of the system of Eq. 1 is maintained within Ω_ρ at all times (i.e., Ω_ρ is a forward invariant set), and is driven in finite time into $\Omega_{\rho_{sp}}$ and thereafter bounded there. Finally, we prove that after t_s, the closed-loop state under the Safety-S-DLEMPC of Eq. 8 is ultimately bounded in $\Omega_{\rho_{\min}}$.

Part 1: The feasibility of the optimization problem for Safety-S-DLEMPC j (for $j=1,\ldots,m$) when $x(t_0) \in \Omega_\rho$ follows because the solution $K_c(t)=0,\ \forall\ \ t \in [t_k, t_{k+N})$, $\bar{u}_j(t)=\bar{h}_j(\tilde{x}^j(t_n)),\ \forall\ \ t \in [t_n, t_{n+1})$, with $n=k,\ldots,N+k-1$, is a feasible solution both when K_c is pre-set to zero throughout the prediction horizon and when it is not. The gain $K_c(t)=0,\ \forall\ \ t \in [t_k, t_{k+N})$, is feasible since it satisfies Eq. 8g over the prediction horizon. When $K_c(t)=0$, then by Eq. 8i, $\tilde{\rho}(t)$ will be equal to its initial value throughout the prediction horizon, and thus the upper bound on the Lyapunov function in Eq. 8h will be fixed (i.e., either $\tilde{\rho}(t_k)=V(x(t_k)) \Rightarrow V(\tilde{x}^j(t)) \le V(x(t_k)),\ \forall\ \ t \in [t_k, t_{k+N}),$ if $x(t_k) \notin \Omega_{\rho_{sp}}$ or $\tilde{\rho}(t_k)=\rho_{sp} \Rightarrow V(\tilde{x}^j(t)) \le \rho_{sp},\ \forall\ \ t \in [t_k, t_{k+N}),$ if $x(t_k) \in \Omega_{\rho_{sp}}$). In such a case, $\bar{u}_j(t)=\bar{h}_j(\tilde{x}^j(t_n)),\ \forall\ \ t \in [t_n, t_{n+1})$, with $n=k,\ldots,N+k-1$, satisfies the input constraint of Eq. 8c. To prove that $\bar{u}_j(t)=\bar{h}_j(\tilde{x}^j(t_n)),\ \forall\ \ t \in [t_n, t_{n+1}), n=k,\ldots, N+k-1$, satisfies Eqs. 8h and 8j and is thus a feasible solution to Safety-S-DLEMPC j when $\bar{u}_r(t)= \bar{h}_r(\tilde{x}^j(t_{k+q}))$, $r=j+1,\ldots,m$, $\forall\ t \in [t_{k+q}, t_{k+q+1})$, $q=0,\ldots,N-1$, and $\bar{u}_p(t) = \bar{u}_p^*(t|t_k)$, $p=1,\ldots,j-1$, $\forall\ t \in [t_k, t_{k+N})$, as required by Eqs. 8d and 8e, the sequence of distributed controllers must be evaluated. We will proceed by induction. When $j=1$, $\bar{u}_j(t)=\bar{h}_j(\tilde{x}^j(t_n)),\ \forall\ t \in [t_n, t_{n+1}), n=k,\ldots,N+k-1$, satisfies Eq. 8h in Safety-S-DLEMPC 1 by Proposition 1 when $x(t) \in \Omega_\rho$, and trivially satisfies the constraint of Eq. 8j since $\bar{u}_r(t)$, $r=2,\ldots,m$ are set to $\bar{h}_r(\tilde{x}^j)$ implemented in sample-and-hold through Eq. 8e. Thus, $K_c(t)=0,\ \forall\ \ t \in [t_k, t_{k+N})$, $\bar{u}_j(t)=\bar{h}_j(\tilde{x}^j(t_n)),\ \forall\ \ t \in [t_n, t_{n+1}), n=k,\ldots,N+k-1$, is a feasible solution for Safety-S-DLEMPC 1.

Now, assume that there exists a feasible solution to Safety-S-DLEMPC j – 1 (i.e., $\bar{u}_p^*(t|t_k), p=1,\ldots,j-1, \forall\ t \in [t_k, t_{k+N})$) and that feasibility of $\bar{u}_j(t)=\bar{h}_j(\tilde{x}^j(t_n)),\ \forall\ \ t \in [t_n, t_{n+1})$, $n=k,\ldots,N+k-1$, is being considered for Safety-S-DLEMPC j. Because Safety-S-DLEMPC j – 1 was feasible (i.e., Eqs. 8h and 8j were satisfied) when $\bar{u}_p(t)=\bar{u}_p^*(t|t_k), p=1,\ldots,j-1$, $\forall\ t \in [t_k, t_{k+N})$, with all other inputs set to the corresponding components of $\bar{h}(x)$ implemented in sample-and-hold, the same input trajectory (i.e., $\bar{u}_j(t)=\bar{h}_j(\tilde{x}^j(t_n)),\ \forall\ \ t \in [t_n, t_{n+1})$, $n=k,\ldots,N+k-1$, and the other inputs defined according to Eqs. 8d and 8e) will be feasible for Safety-S-DLEMPC j because it will again satisfy Eqs. 8h and 8j; the feasibility of this solution is independent of the value of K_c in Safety-S-DLEMPC j – 1 or Safety-S-DLEMPC j. Therefore, $K_c(t)=0,\ \forall\ t \in [t_k, t_{k+N}),\ \bar{u}_j(t)=\bar{h}_j(\tilde{x}^j(t_n)),\ \forall\ \ t \in [t_n, t_{n+1}),\ n=k,\ldots, N+k-1$, is a feasible solution for Safety-S-DLEMPC 1 and also for Safety-S-DLEMPC j when Safety-S-DLEMPC j – 1 is feasible; by induction, $K_c(t)=0, \forall\ \ t \in [t_k, t_{k+N}),\ \bar{u}_j(t)= \bar{h}_j\ (\tilde{x}^j(t_n)),\ \forall\ \ t \in [t_n, t_{n+1}), n=k,\ldots,N+k-1$, is therefore a feasible control action for each Safety-S-DLEMPC j, $j=1,\ldots,m$. Recursive feasibility of the Safety-S-DLEMPC follows if the closed-loop state trajectory is maintained within Ω_ρ (which will be proven in Part 2 to hold for all times if $x(t_0) \in \Omega_\rho$).

Part 2: We now prove that if $x(t_k)$ is initialized outside the safety level set (i.e., $x(t_k) \in \Omega_\rho/\Omega_{\rho_{sp}}$ and $t_k \le t_s$), then the closed-loop state remains bounded within Ω_ρ (i.e., $x(t) \in \Omega_\rho$ when $x(t_0) \in \Omega_\rho$) and within finite time, the closed-loop state will be driven to $\Omega_{\rho_{sp}}$ and remain there for all subsequent times under the Safety-S-DLEMPC design of Eq. 8.

Due to the sequential solution strategy of the Safety-S-DLEMPC architecture, the set of control actions $u_j^*(t_k|t_k)$, $j=1,\ldots,m$, that are implemented on the process (and thus affect closed-loop stability) satisfy the constraints of the Safety-S-DLEMPC of Eq. 8 when $j=m$. When $x(t_k) \in \Omega_\rho/\Omega_{\bar{\rho}_{sp}}$, from the constraint of Eq. 8j of the Safety-S-DLEMPC m of Eq. 8 and from Eq. 2b, we obtain:

$$\begin{aligned}\frac{\partial V(x(t_k))}{\partial x}&\Big(f(x(t_k))+\sum_{i=1}^{m} g_i(x(t_k))\bar{u}_i^*(t_k|t_k)\Big)\\ &\le \frac{\partial V(x(t_k))}{\partial x}\Big(f(x(t_k))+\sum_{i=1}^{m} g_i(x(t_k))\bar{h}_i(x(t_k))\Big) \qquad \text{(A1a)}\\ &\le -\alpha_3(|x(t_k)|) \qquad \text{(A1b)}\end{aligned}$$

The time derivative of the Lyapunov function along the actual system state trajectory $x(t)$ for $t \in [t_k, t_{k+1})$ can be written as follows:

$$\dot{V}(x(t))=\frac{\partial V(x(t))}{\partial x}\Big(f(x(t))+\sum_{i=1}^{m} g_i(x(t))\bar{u}_i^*(t_k|t_k)+b(x(t))w(t)\Big) \qquad \text{(A2)}$$

Adding and subtracting $\frac{\partial V(x(t_k))}{\partial x}(f(x(t_k))+\sum_{i=1}^{m} g_i(x(t_k))\bar{u}_i^*(t_k|t_k))$ to/from the above equation and accounting for Eq. A1, the bound on the disturbance ($|w| \le \theta$), and the Lipschitz properties of Eqs. 4–6, we can write:

$$\dot{V}(x(t)) \le -\alpha_3(|x(t_k)|)+\left(L_x+\sum_{i=1}^{m} L_{\bar{u}_i}\bar{u}_i^*(t_k|t_k)\right)|x(t)-x(t_k)|+L_w\theta \tag{A3}$$

From Eq. 3 and the continuity of $x(t)$, the following bound can be written for all $t \in [t_k, t_{k+1})$:

$$|x(t)-x(t_k)| \le M\Delta \tag{A4}$$

Since $x(t_k) \in \Omega_\rho/\Omega_{\bar{\rho}_{sp}}$, it can be concluded that $x(t_k) \in \Omega_\rho/\Omega_{\rho_s}$. Using this, as well as Eqs. A3 and A4 and the bounds on the inputs $\bar{u}_i$, $i=1,\ldots,m$, we obtain the following bound on $\dot{V}(x(t))$ for $t \in [t_k, t_{k+1})$:

$$\dot{V}(x(t)) \le -\alpha_3(\alpha_2^{-1}(\rho_s))+\left(L_x+\sum_{i=1}^{m} L_{\bar{u}_i}\bar{u}_i^{\max}\right)M\Delta+L_w\theta \tag{A5}$$

If the condition of Eq. 13 is satisfied, then there exists $\epsilon_w > 0$ such that the following inequality holds for $x(t_k) \in \Omega_\rho/\Omega_{\bar{\rho}_{sp}}$:

$$\dot{V}(x(t)) \le -\epsilon_w/\Delta \qquad \forall\ t \in [t_k, t_{k+1}) \tag{A6}$$

Integrating the bound of Eq. A6 on $t \in [t_k, t_{k+1})$ we obtain that:

$$V(x(t_{k+1})) \le V(x(t_k))-\epsilon_w \tag{A7a}$$

$$V(x(t)) \le V(x(t_k)), \quad \forall\ t \in [t_k, t_{k+1}) \tag{A7b}$$

for all $x(t_k) \in \Omega_\rho/\Omega_{\bar{\rho}_{sp}}$. Using Eq. A7 recursively, it is proved that, if $x(t_k) \in \Omega_\rho/\Omega_{\bar{\rho}_{sp}}$, the state converges to $\Omega_{\bar{\rho}_{sp}}$ in a finite number of sampling times while remaining within Ω_ρ throughout the transition since $V(x)$ does not increase. Once the state converges to $\Omega_{\bar{\rho}_{sp}} \subseteq \Omega_{\rho_{sp}}$, it remains inside $\Omega_{\rho_{sp}}$ for all times from the definition of $\Omega_{\bar{\rho}_{sp}}$ in Theorem 1 (i.e., if $x(t_k) \in \Omega_{\bar{\rho}_{sp}}$, then $x(t_{k+1}) \in \Omega_{\rho_{sp}}$) and re-activation of the contractive constraint of Eq. 8j to decrease the Lyapunov function value until $x(t_k) \in \Omega_{\bar{\rho}_{sp}}$ whenever $x(t_k) \in \Omega_{\rho_{sp}}/\Omega_{\bar{\rho}_{sp}}$. Since $\Omega_{\rho_{sp}} \subseteq \Omega_\rho$, the state of the closed-loop system is always maintained within Ω_ρ making it a forward invariant set.

Part 3: Finally, we prove ultimate boundedness of the closed-loop state within $\Omega_{\rho_{\min}}$ when $t_k > t_s$. If $t_k > t_s$, then Eq. 8j is active at all subsequent sampling times. Since Eq. A6 holds whenever $x(t_k) \in \Omega_\rho/\Omega_{\rho_s}$, Eq. A7a also holds and thus for $x(t_k) \in \Omega_\rho/\Omega_{\rho_s}$, $V(x(t_{k+1})) < V(x(t_k))$ and the closed-loop state moves to lower level sets until $x(t_k) \in \Omega_{\rho_s}$. From the definition of $\Omega_{\rho_{\min}}$ in Proposition 1, once the state converges to $\Omega_{\rho_s} \subseteq \Omega_{\rho_{\min}}$, it remains inside $\Omega_{\rho_{\min}}$ for all times. □

Proof of Theorem 2. Like the proof of Theorem 1, the proof of Theorem 2 consists of three parts. We first prove that under steps 1–3 of the safety-I-DLEMPC implementation strategy, the optimization problem of Eq. 14 is feasible for each iteration c that is executed when $x(t_0) \in \Omega_\rho$, and that the control actions implemented on the process under this implementation strategy have characterizable properties. Then we prove that the closed-loop state of the system of Eq. 1 can be driven in finite time into $\Omega_{\rho_{sp}}$ under the control actions from the Safety-I-DLEMPC implementation strategy, and then be bounded there. We also prove that under the Safety-I-DLEMPC implementation strategy, the closed-loop state is always maintained in Ω_ρ if $x(t_0) \in \Omega_\rho$ (i.e., Ω_ρ is a forward invariant set). Finally, we prove that after t_s, the closed-loop state under the Safety-I-DLEMPC implementation strategy is ultimately bounded in $\Omega_{\rho_{\min}}$.

Part 1: At the initial iteration (i.e., $c=1$) and for all $x(t_0) \in \Omega_\rho$, the solution $K_{c,1}(t)=0$, $\forall\ t \in [t_k, t_{k+N})$, $\bar{u}_{j,1}(t)=\bar{h}_j(\tilde{x}^j(t_n))$, $\forall\ t \in [t_n, t_{n+1})$, with $n=k,\ldots,N+k-1$, is a feasible solution to each Safety-I-DLEMPC j of Eq. 14, $j=1,\ldots,m$, both when K_c is fixed at zero and when it is not. Feasibility of $K_{c,1}(t)=0$, $\forall\ t \in [t_k, t_{k+N})$, at $c=1$ follows because $K_{c,1}(t)=0$, $\forall\ t \in [t_k, t_{k+N})$, satisfies Eq. 14g throughout the prediction horizon. When $K_{c,1}(t)=0$, then as described in the proof of Theorem 1, the upper bound on the Lyapunov function in Eq. 14h is fixed to either $V(x(t_k))$ or ρ_{sp}. In such a case, $\bar{u}_{j,1}(t)=\bar{h}_j(\tilde{x}^j(t_n))$, $\forall\ t \in [t_n, t_{n+1})$, $n=k,\ldots,N+k-1$, satisfies the input constraint of Eq. 14c. Because $\bar{u}_z(t)=\bar{h}_z\ (\tilde{x}^j(t_{k+r}))$, $z \in \{1,\ldots,m\}$, $z \ne j$, $\forall\ t \in\ [t_{k+r}, t_{k+r+1})$, $r=0,\ldots,N-1$, from Eq. 14e, the constraint of Eq. 14h is satisfied by Proposition 1,[25] as is the constraint of Eq. 14j (trivially). For the subsequent iterations (i.e., $c>1$), the solution $K_{c,c}(t)=0$, $\forall\ t \in [t_k, t_{k+N})$, $\bar{u}_{j,c}(t)=\bar{u}_{j,c-1}^*(t|t_k)$, $\forall\ t \in [t_n, t_{n+1})$, with $n=k,\ldots,N+k-1$, is a feasible solution to Safety-I-DLEMPC j, $j=1,\ldots,m$ (regardless of whether K_c is fixed to zero in the optimization problem or not) when the condition of Eq. 14h is satisfied by the solutions $\bar{u}_{j,c-1}^*(t|t_k)$, $\forall\ t \in [t_n, t_{n+1})$, $n=k,\ldots,N+k-1$, $j=1,\ldots,m$, from the prior iteration, i.e., when $V(\tilde{x}^{tot}(t)) \le V(x(t_k))$, $\forall\ t \in [t_k, t_{k+N})$, if $x(t_k) \notin \Omega_{\rho_{sp}}$, or when $V(\tilde{x}^{tot}(t)) \le \rho_{sp}$, $\forall\ t \in [t_k, t_{k+N})$, if $x(t_k) \in \Omega_{\rho_{sp}}$, where $\tilde{x}^{tot}(t)$, $\forall\ t \in [t_k, t_{k+N})$, is defined as the solution obtained by recursively solving:

$$\dot{\tilde{x}}^{tot}=f(\tilde{x}^{tot})+\sum_{i=1}^{m} g_i(\tilde{x}^{tot})\bar{u}_{i,c-1}^*(t|t_k) \tag{A8}$$

given $\tilde{x}^{tot}(t_k)=x(t_k)$. Feasibility of this solution follows because since it was feasible at the prior iteration, it satisfied the input constraint of Eq. 14c and will also satisfy the constraints of Eqs. 14h and 14j. Because the upper bound on the Lyapunov function in Eq. 14h is the same between two iterations since it is based only on a state measurement at t_k and thus will be the same for all iterations at t_k, when the condition on $V(\tilde{x}^{tot}(t))$ is checked under $\bar{u}_{z,c-1}^*(t|t_k)$, $z=1,\ldots,m$, $\forall\ t \in [t_k, t_{k+N})$, at the end of the prior iteration and now $\bar{u}_z(t)=\bar{u}_{z,c-1}^*(t|t_k)$, $z \in \{1,\ldots,m\}$, but $z \ne j$, $\forall\ t \in [t_k, t_{k+N})$, within Safety-I-DLEMPC j from the constraint of Eq. 14d, it is already known from the check at the prior iteration that with those trajectories for all $\bar{u}_z(t)$ for $z \ne j$ that $\bar{u}_{j,c-1}^*(t|t_k)$, $\forall\ t \in [t_k, t_{k+N})$, will meet the constraint of Eq. 14h. Finally, unlike the constraint of Eq. 14h, the contractive constraint of Eq. 14j does not depend on control actions $\bar{u}_z(t)$, $z \ne j$; therefore, the solution $\bar{u}_{j,c-1}^*(t)$ will satisfy the contractive constraint of Safety-I-DLEMPC j, where $j=1,\ldots,m$, at iteration c if it is satisfied at the prior iteration. If the termination condition is met or the condition on $V(\tilde{x}^{tot}(t))$ under $\bar{u}_{j,c}^*(t|t_k)$, $\forall\ t \in [t_k, t_{k+N})$, $j=1,\ldots,m$, is not satisfied and $c>1$, then a new iteration is not performed. When the new iteration is not performed, a solution that was feasible at the prior iteration (i.e., $\bar{u}_{z,c-1}^*(t|t_k)$, $t \in [t_k, t_{k+N})$, $z=1,\ldots,m$) is implemented. Because this solution was feasible for all j Safety-I-DLEMPC's, $j=1,\ldots,m$, at the prior iteration, it is known to have satisfied the constraint of each Safety-I-DLEMPC and therefore has characterizable properties. If $c=1$ and the condition on $V(\tilde{x}^{tot}(t))$ is not satisfied, $\bar{h}(x)$ is implemented in sample-and-hold, which also has characterizable properties (e.g., Proposition 1). Therefore, feasibility of the Safety-I-DLEMPC is ensured at each iteration that is attempted due to checking of the condition on $V(\tilde{x}^{tot}(t))$ before attempting a new iteration. However, there is no guarantee that this condition will be met at the end of any iteration. When it is not met and iterating stops, however, the solution applied under the implementation strategy (i.e., either $\bar{u}_{j,c-1}^*(t_k|t_k)$, $j=1,\ldots,m$, or $\bar{h}(x(t_k))$) has characterizable properties.

Part 2: We now utilize the known properties of the implemented control actions under the Safety-I-DLEMPC implementation strategy to prove closed-loop stability of a nonlinear process under this implementation strategy in the sense of boundedness of the closed-loop state. First, we prove that if $x(t_k) \in \Omega_\rho/\Omega_{\bar{\rho}_{sp}}$ then $V(x(t_{k+1})) < V(x(t_k))$ and in finite steps, the closed-loop state converges to $\Omega_{\bar{\rho}_{sp}}$ (i.e., $x(t_{k+p}) \in \Omega_{\bar{\rho}_{sp}}$ where p is a finite positive integer) in a manner that maintains the closed-loop state within Ω_ρ. We then demonstrate that once the closed-loop state enters $\Omega_{\rho_{sp}}$, it is bounded there for all subsequent times.

When $x(t_k) \in \Omega_\rho/\Omega_{\bar{\rho}_{sp}}$ and $\bar{u}^*_{j,c-1}(t_k|t_k)$, $j=1,\ldots,m$, is applied to the plant, Eq. 14j holds for each implemented control action. By summing the constraints of Eq. 14j for all j Safety-I-DLEMPC's, $j=1,\ldots,m$, and utilizing Eq. 2b, we obtain:

$$\sum_{j=1}^{m} \frac{\partial V(x(t_k))}{\partial x} g_j(x(t_k))\bar{u}^*_{j,c-1}(t_k|t_k) \leq \sum_{j=1}^{m} \frac{\partial V(x(t_k))}{\partial x} g_j(x(t_k))\bar{h}_j(x(t_k)) \tag{A9a}$$

$$= \frac{\partial V(x(t_k))}{\partial x}\left(f(x(t_k)) + \sum_{j=1}^{m} g_j(x(t_k))\bar{u}^*_{j,c-1}(t_k|t_k)\right) \leq \frac{\partial V(x(t_k))}{\partial x}\left(f(x(t_k)) + \sum_{j=1}^{m} g_j(x(t_k))\bar{h}_j(x(t_k))\right) \tag{A9b}$$

$$\leq -\alpha_3(|x(t_k)|) \tag{A9c}$$

Following the same approach as in the proof of Theorem 1, if the condition of Eq. 13 is satisfied, then $V(x(t_{k+1})) < V(x(t_k))$ under the implemented control action. If $x(t_k) \in \Omega_\rho/\Omega_{\bar{\rho}_{sp}}$ but $\bar{h}(x(t_k))$ is applied to the plant, then by Proposition 1, $V(x(t_{k+1})) < V(x(t_k))$. Therefore, at any given sampling time when $x(t_k) \in \Omega_\rho/\Omega_{\bar{\rho}_{sp}}$, regardless of whether $\bar{u}^*_{j,c-1}(t_k|t_k)$, $i=1,\ldots,m$, or $\bar{h}(x(t_k))$ is implemented according to the implementation strategy of the Safety-I-DLEMPC, $V(x(t_{k+1})) < V(x(t_k))$ and this will cause the closed-loop state to be driven into $\Omega_{\bar{\rho}_{sp}}$ in finite time in a manner that cannot exit Ω_ρ. When $\Omega_{\bar{\rho}_{sp}}$ is defined as in Theorem 2 such that if $x(t_k) \in \Omega_{\bar{\rho}_{sp}}$, then $x(t_{k+1}) \in \Omega_{\rho_{sp}}$, the result is that $\Omega_{\rho_{sp}}$ is a forward invariant set. This is because if $x(t_k) \in \Omega_{\rho_{sp}}/\Omega_{\bar{\rho}_{sp}}$, the constraint of Eq. 14j is active when computing the $\bar{u}^*_{j,c-1}(t_k|t_k)$, $j=1,\ldots,m$, that are applied to the plant, and thus either a solution that meets that constraint or $\bar{h}(x(t_k))$ will be applied to the plant. The result will be that $V(x(t_{k+1})) < V(x(t_k))$, so if $x(t_k) \in \Omega_{\rho_{sp}}/\Omega_{\bar{\rho}_{sp}}$, then $x(t_{k+1}) \in \Omega_{\rho_{sp}}$. If $x(t_k) \in \Omega_{\bar{\rho}_{sp}}$, then $x(t_{k+1}) \in \Omega_{\rho_{sp}}$ from the definition of $\Omega_{\bar{\rho}_{sp}}$. Therefore, once the closed-loop state enters $\Omega_{\rho_{sp}}$ under this implementation strategy, it cannot leave it. Furthermore, since $\Omega_{\rho_{sp}} \subseteq \Omega_\rho$, the closed-loop state under this implementation strategy is always bounded in Ω_ρ.

Part 3: When $t_k > t_s$, either inputs $\bar{u}^*_{j,c-1}(t_k|t_k)$, $j=1,\ldots,m$, that cause Eq. A9 to hold are applied to the plant, or the Lyapunov-based controller implemented in sample-and-hold is applied, for which the results of Proposition 1 hold. Following similar steps as in the proof of Part 3 of Theorem 1, this causes $V(x(t_{k+1})) \leq V(x(t_k))$ while $x(t_k) \in \Omega_\rho/\Omega_{\rho_s}$, driving the closed-loop state into Ω_{ρ_s} in finite time. Subsequently, from the definition of $\Omega_{\rho_{\min}}$, the system state is ultimately bounded in an invariant set $\Omega_{\rho_{\min}}$ under the implementation strategy of the Safety-I-DLEMPC. □

Manuscript received Dec. 12, 2016, and revision received Jan. 31, 2017.