2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

Online Active Safety for Robotic Manipulators

Andrew Singletary, Petter Nilsson, Thomas Gurriet, and Aaron D. Ames

***Abstract*— Future manufacturing environments will see an increased need for cooperation between humans and machines. In this paper we propose a method that allows industrial manipulators to safely operate around humans. This approach guarantees that the manipulator will never collide with human operators while performing its normal tasks. This is done in an near-optimal way by considering how forward reachable sets of human operators grow with time, and by continuously updating these reachable sets based on current position estimates of the operators near the robot. An implicit active set invariance filter is then used to constrain the system—in a minimally invasive way—to stay in the complement of that forward reachable set. We demonstrate this approach in simulation on an industrial robotic arm: the ABB IRB 6640.**

I. INTRODUCTION

The global industrial robot market has more than doubled in the past five years, and the International Federation of Robotics expects almost two million new robot installations in factories by 2020 [1]. However, concern for the safety of their human counterparts grows along with the density of robots in factories. As a result, in heavy manufacturing machines and humans are mostly separated. This makes the process rigid: it becomes spatially constrained and manual intervention in the vicinity of a robot may require halting the process altogether. To reduce downtime and allow for more human-robot interaction, large strides must be made in ensuring the safety of these robots in dynamic environments.

Safety in control can be generalized to the concept of constraining a system to a set of safe states. Thus, if the system remains in that safe set for all time, it is considered safe. More formally, a system is considered safe if its defined safe set is forward invariant. The most straightforward way to prove forward invariance is through reachability analysis, but for high-dimensional systems current methods are either intractable, or overly conservative, even for closed-loop systems [2]. Barrier certificates offered an alternative to expensive reachability calculations for invariant set verification of closed-loop dynamical systems [3]. These barrier certificates were then extended to control systems in the form of control barrier functions [4], which provide a tractable approach for ensuring forward invariance of a system under any control law, in the absence of input constraints. For more results on control barrier functions, see [5] for information on robustness and [6] [7] for applications. However, barrier functions are still challenging to find in the presence of input constraints. To ensure feasibility of the optimization problem, one must ensure that the set described by the barrier function is *viable*, [8]. This has been done on robotic systems, [9]; however, finding these viable sets is often intractable for high dimensional systems [10], or overly conservative [11]. The issue of finding viable sets is particularly apparent in the context of time-varying barrier functions. While there are results on time-varying barrier functions [12], they don't take the case of bounded inputs into account. Importantly, current methods would require recomputation of viable sets in real-time, which is infeasible even for the conservative methods.

A recent approach was introduced for implementing barrier functions on input-constrained systems without the need for explicit computation of viable sets [13]. This approach, which will be explored further in this paper, is unique in that it allows us to guarantee safety in complex systems for which finding viable sets is not realistic. In this paper, we will show that this approach is suitable for high-dimensional, complex systems, whose dynamics may not have simple, closed-form expressions, and furthermore, that it readily allows for time-varying safe sets. We also provide an important proof for the that reinforces the validity of this method in practice.

In particular, we demonstrate the method on a robotic manipulator in a factory environment with human workers. This type of system is of particular interest as it has several characteristics that make it difficult for these types of safety guarantees: high dimensionality, complex dynamics, and a dynamic environment. The constraint of avoiding humans is particularly difficult, as the safety constraint is time-varying and in Cartesian space, while the dynamics evolve in the joint space. However, recent advances in rigid-body dynamics algorithms [14] allow for rapid computation of the dynamics and control inputs and their derivatives for such systems. By utilizing these algorithms, we are able to guarantee safety for robotic manipulators in real-time, with no prior computations.

The most noteable alternative to our approach is the potential field approach [15]. However, it has certain limitations: in cluttered environments the method can be conservative by rendering some collision-free paths infeasible, and there are limitations in terms of the complexity of the nominal tasks.

The outline of the paper is as follows: Section II details the preliminary mathematics and notation used moving forward and Section III introduces the theory of safety for control systems. Section IV then details the online safety approach used to enforce set invariance under any control law. Section V introduces the robotic arm and setup that will be used to demonstrate the method, while Section VI showcases the results from simulation. Lastly, Section VII concludes the results and discusses future work.

The authors are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena CA 91125, U.S.A. Email addresses: {asinglet, pettni, tgurriet, ames}@caltech.edu.

II. PRELIMINARIES

In this paper, we consider continuous-time affine control systems of the form:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x) + g(x)u, \tag{1}$$

with f and g continuous functions defined on $\mathbb{R}^n$, and with $u \in U$ a compact subset of $\mathbb{R}^m$. Existence and uniqueness of solutions to (1) is required for most of the discussed results to hold. For example, Lipschitz continuity of f, g and u provides a sufficient condition in that regard.

For a feedback controller $u(x)$ the flow operator $\phi_t^u(x_0)$ is defined as the solution of the initial value problem

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= f(x) + g(x)u(x), \\ x(0) &= x_0. \end{aligned} \tag{2}$$

The flow satisfies the semi-group property

$$\phi_{t+s}^u(x_0) = \phi_t^u \circ \phi_s^u(x_0). \tag{3}$$

Furthermore, when $f + gu$ is smooth, the sensitivity of the flow operator with respect to the initial time and state satisfies

$$D\phi_t^u(x_0) = Q(t), \tag{4}$$

where $Q(t)$ is a solution to the initial value problem [16]

$$\begin{aligned} \frac{\mathrm{d}Q}{\mathrm{d}t} &= D\left[(f+gu)\circ\phi_t^u(x)\right]Q(t), \\ Q(0) &= I. \end{aligned} \tag{5}$$

III. SAFETY, INVARIANCE, AND BARRIERS

We utilize a notion of safety that is formalized as invariance of a set $S \subset \mathbb{R}^n$.

Definition 1: A set S is **forward invariant** for system (1) if $x(0) \in S$ implies that $x(t) \in S$ for all $x \geq 0$.

The main tool at our disposal for set invariance is Nagumo's theorem [17] that states that the forward invariance of S for system (1) is equivalent to the *sub-tangentiality condition*:

$$f(x) + g(x)u(x) \in \mathcal{T}_S(x), \tag{6}$$

being satisfied for all $x \in S$, where $\mathcal{T}_S(x)$ is the contingent cone to S at x [8], [17].

In the following we restrict attention to *practical sets* [17] that are defined as the zero super level set of a collection of r continuously differentiable functions $h_j : \mathbb{R}^n \to \mathbb{R}$:

$$S = \left\{ x \in \mathbb{R}^n \mid \bigwedge_{j=1}^{r} (h_j(x) \geq 0) \right\}. \tag{7}$$

For such sets, the contingent cone can be expressed as

$$\mathcal{T}_S(x) = \left\{ v \in \mathbb{R}^n \mid \bigwedge_{j \in Act(x)} \langle \nabla_x h_j(x), v \rangle \geq 0 \right\}, \tag{8}$$

$$Act(x) \triangleq \{ j \in \{1,\dots,r\} \mid h_j(x) = 0 \}.$$

In that case, the sub-tangentiality condition (6) can be written as

$$TC_j(x,u) \triangleq L_f h_j(x) + L_g h_j(x) u \geq 0, \tag{9}$$

for all $x \in \partial S$, and $j \in Act(x)$, where $L_f h$ and $L_g h$ denote the Lie derivatives of h along f and g, respectively. Therefore, condition (9) defines for any $x \in S$ a set $U_S(x)$ of admissible inputs that guarantee forward invariance of S:

$$U_S(x) \triangleq \begin{cases} \bigcap_{j=1}^{r} \{u \in \mathbb{R}^m : TC_j(x,u) \geq 0\}, & \text{if } x \in \partial S, \\ \mathbb{R}^m, & \text{otherwise.} \end{cases} \tag{10}$$

The sub-tangentiality condition is however not very desirable to enforce in practice as it only restricts the set of admissible inputs when the system is on the boundary of the safety set, which is a measure zero surface in the state space. The idea introduced in [4] is to consider a strengthening term in (9) and to impose this new *barrier condition*:

$$BC_j(x,u) \triangleq L_f h_j(x) + L_g h_j(x) u \geq -\alpha_j(h_j(x)), \tag{11}$$

for all $x \in S$, $j \in \{1,\dots,r\}$ and with the *strengthening* extended class $\mathcal{K}$ functions $\alpha_j : \mathbb{R} \to \mathbb{R}$. This barrier condition defines a set $\widetilde{U_S}(x)$ of admissible inputs:

$$\widetilde{U_S}(x) \triangleq \{ u \in \mathbb{R}^m \mid \forall j \in \{1,\dots,r\}, BC_j(x,u) \geq 0 \} \tag{12}$$

and because for all $x \in S$, $\widetilde{U_S}(x) \subseteq U_S(x)$, this new condition also implies forward invariance of S.

This is however only half of the story as nothing guarantees that $\widetilde{U_S}(x) \cap U$ is non empty for all $x \in S$. This is because for a given control system, arbitrary sets cannot *a priori* be rendered forward invariant. A set that can be rendered forward invariant is commonly referred to as **viable** set [8].

Definition 2: A closed set S is **viable** for system (1) if for all $x(0) \in S$, there exists a control law $u : \mathbb{R}^n \to U$ such that $\forall t \geq 0$, $x(t) \in S$ under that policy.

Equivalently, a viable set can be defined as a set with the property that $\widetilde{U_S}(x) \neq \emptyset$ for all (t,x) inside the set. Note that in most cases $\widetilde{U_S}(t,x) \subsetneq U_S(t,x)$, so finding a viable set is not sufficient to ensure that $\widetilde{U_S}(t,x) \cap U$ is always non-empty. One has to be careful and choose the strengthening functions α_j such that for all $x \in S$, $\widetilde{U_S}(t,x) \cap U \neq \emptyset$.

IV. ONLINE ACTIVE SAFETY

Ideally, one would want to find the largest viable subset of S—the *viability kernel*—to maximize the operational freedom of the system. This is however notoriously hard—just as hard as finding an optimal control law. But as in optimal control, there is a dual approach: continuously solving for the optimal control action at the current state. Unfortunately, solving viability this way requires finding a trajectory over an *infinite* time horizon, which is not possible in practice. In this section we show how a viable subset can be implicitly characterized via a small viable *backup set* $S_0 \subset S$ (which is easy to compute) and a *backup controller* that renders S_0 invariant.

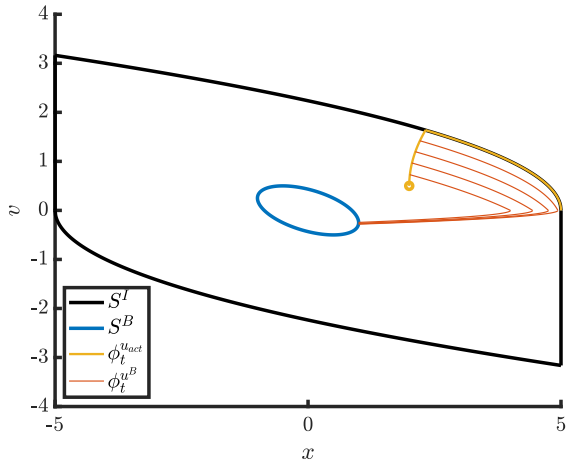

Fig. 1. Illustration of the Backup Set and Invariant Set.

A. Safety via Backup Reachability

Consider a backup set $S^B = \{x \mid h^B(x) \geq 0\} \subset S$ and a backup controller $u^B : \mathbb{R}^n \to U$ with the property that for all $x \in \partial S^B$,

$$L_f h^B(x) + L_g h^B(x) u^B(x) \geq 0, \tag{13}$$

which implies that the set S^B is invariant under the closed-loop dynamics resulting from u^B. For an appropriately designed u^B and a time horizon T a larger implicit invariant set S^I is defined as the domain of attraction within time T of S^B under the closed-loop dynamics. This set S^I is defined as follows:

$$S^I = \left\{ x \mid \bigwedge_{\tau \in [0,T]} \left(\phi_\tau^{u^B}(x) \in S \right) \wedge \left(\phi_T^{u^B}(x) \in S^B \right) \right\}. \tag{14}$$

That is, the implicit set S^I consists of all initial conditions that are steered to S^B within time T without exiting S along the way.

Theorem 1: If (13) holds (i.e., S^B is invariant under u^B), then S^I is a viable set contained in S.

Proof: Consider $x \in S^I$, then $\phi_\tau^{u^B}(x) \in S$ for all $\tau \in [0, T]$. Containment of S^I in S follows from the special case $\tau = 0$. From the semi-group property of the flow and invariance of S^B under u^B it follows that for any $\tau, s > 0$

$$\begin{aligned} x \in S^I \implies \phi_s^{u^B} \circ \phi_\tau^{u^B}(x) \in S \wedge \phi_T^{u^B} \circ \phi_\tau^{u^B}(x) \in S^B \\ \implies \phi_\tau^{u^B}(x) \in S^I, \end{aligned}$$

which implies that also S^I is invariant under u^B. Hence S^I is viable. ∎

Figure 1 illustrates the implicit viable set for a double integrator under an optimal backup controller with walls at $x = \pm 5$.

B. Implicit Level Set Functions

With a properly defined u^B the implicit set S^I can be significantly larger than S^B, and thus be less conservative. However, enforcement of the barrier condition (11) requires knowledge of a collection of level set functions h_j^I that together define S^I, as well as their derivatives. In the following we construct such functions and then discuss how they can be computed on-the-fly via numerical integration.

Two types of constraints are required to define S^I: one for ensuring that S^B is reached within time T, and a family of constraints that ensure that S is kept invariant along the trajectory. Such constraints can be defined in terms of the functions h^B and h_j that define the sets S^B and S, and the flow of the backup controller:

$$h_T^I(x) = h^B \circ \phi_T^{u^B}(x), \tag{15a}$$

$$h_{j,\tau}^I(x) = h_j \circ \phi_\tau^{u^B}(x). \tag{15b}$$

Equation (15b) represents an infinite collection of functions, which poses an issue that we will address later. The validity of the following proposition is clear from the definition of S^B.

Proposition 1: S^I is the super 0 level set of the functions defined in (15), i.e.

$$S^I = \left\{ x : h_T^I(x) \geq 0 \wedge \bigwedge_{\tau \in [0,T]} \bigwedge_{j=1}^{r} \left(h_{j,\tau}^I(x) \geq 0 \right) \right\}. \tag{16}$$

Thus, we can enforce invariance of S^I either via the traditional condition (9) or the strengthened barrier condition (11). However, both of these conditions require knowledge of the gradient of the barrier functions (15). From the chain rule of differentiation the gradients can be written as follows:

$$\nabla h_T^I(x) = D\left[h^B\right]_{\phi_T^{u^B}(x)} D\left[\phi_T^{u^B}\right]_x \left(f(x) + g(x)u^b(x)\right), \tag{17a}$$

$$\nabla h_{j,\tau}^I(x) = D\left[h_j\right]_{\phi_\tau^{u^B}(x)} D\left[\phi_\tau^{u^B}\right]_x \left(f(x) + g(x)u^b(x)\right). \tag{17b}$$

These expressions can both be evaluated if the flow $\phi_\tau^{u^B}(x)$ and the flow sensitivity $D[\phi_\tau^{u^B}]_x$ are known, which can be found via numerical integration of the closed-loop dynamics under the backup controller.

C. Online Safety Filter

We now turn to the issue of having an infinite number of functions defining the set. In practice we can only enforce positivity of a finite number of the functions in (15) defining S^I, and therefore propose a safety filter that enforces positivity of a subset of ϵ-tightened constraints evenly spaced in time:

$$L_f h_T^I(x) + L_g h_T^I(x) u \geq -\alpha_T(h_T^I(x)), \tag{18a}$$

$$L_f h_{j,k\eta}^I(x) + L_g h_{j,k\eta}^I(x) u \geq -\alpha_k(h_{j,k\eta}^I(x) - \epsilon), \tag{18b}$$

for $k = 0, 1, \ldots, T/\eta$.

Although this just enforces positivity of a finite number of the functions $h_{j,\tau}^I$, under some regularity conditions and appropriate margin ϵ we expect that this should be sufficient

to guarantee positivity of the whole family of functions. We make this more precise below via the following lemma.

Lemma 1: Let L_h be the Lipschitz constant of h with respect to the Euclidean norm and let

$$L_\phi = \sup_{x \in S} \|f(x) + g(x)u^B(x)\|_2 \tag{19}$$

be the maximal velocity of the closed-loop vector field. Then

$$\left| h \circ \phi_t^{u^B}(x) - h \circ \phi_s^{u^B}(x) \right| \leq L_h L_\phi |t - s| \tag{20}$$

Proof: Assume WLOG that $t \geq s$ and let $y = \phi_s^{u^B}(x)$

$$\begin{aligned} \left| h \circ \phi_t^{u^B}(x) - h \circ \phi_s^{u^B}(x) \right| &\leq L_h \left\| \phi_t^{u^B}(x) - \phi_s^{u^B}(x) \right\|_2 \\ &= L_h \left\| \phi_{t-s}^{u^B}(y) - y \right\|_2 \leq L_h L_\phi |t - s|, \end{aligned}$$

since L_ϕ is the maximal velocity of the vector field. ■

It follows that enforcing invariance of S via a finite subset of constraints as in

Theorem 2: For $\epsilon \geq L_h L_\phi \frac{\eta}{2}$ the safety filter in (18) enforces invariance of S^I.

Proof: The filter implies that $h^I_{j,\tau}(x) = h_j \circ \phi_\tau^{u^B}(x) \geq \epsilon$ for all $\tau = k\eta$, so by Lemma 1 we can for each τ find a k^* such that,

$$\left| h^I_{j,\tau}(x) - h^I_{j,k^*\eta}(x) \right| \leq L_h L_\phi \frac{\eta}{2}, \tag{21}$$

meaning that

$$h^I_{j,\tau}(x) \geq \epsilon - L_h L_\phi \frac{\eta}{2} \geq 0. \tag{22}$$

Thus all the functions defining S^I are positive, and hence S^I is invariant. ■

V. APPLICATION: ROBOTIC ARM OBSTACLE AVOIDANCE

We now apply the method to the problem of collision avoidance in an environment with a robotic arm and a human. An advantage of the implicit approach is that the implicit safe set can be time-varying even when the backup set and backup controller are not. The dynamics of the robotic arm are described by the usual manipulator equations

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau, \tag{23}$$

where q describe the joint angles and τ is a vector of applied torques.

For manipulators with many degrees of freedom the explicit expressions for $M(q)$, $C(q, \dot{q})$ and $G(q)$ are very complicated. As an alternative, they can be evaluated at given points via the Articulated Body Algorithm (ABA) that steps over links of the manipulator [18]. Only having "black-box" access to the equations of motion would pose a problem for most methods for finding invariant sets, but the implicit method proposed in this paper only requires access to the numerical values of the dynamics and its derivatives. We rewrite the dynamics on state-space form and also add a time variable so that we can enforce safety of time-varying sets: $X = [q, \dot{q}, t]^T$.

We now consider the 6-link IRB 6640 manipulator from ABB, depicted in Figure 2. This robot has six degrees of freedom, making the overall system in 13-dimensional.

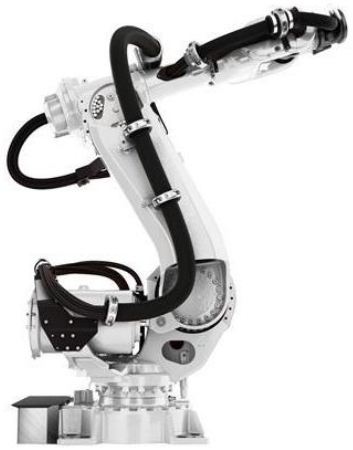

Fig. 2. The IRB 6640 industrial manipulator.

A. Backup Set and Safe Set of the Robot

For the ARB 6640, the backup set is considered to be a vertical tube around the robot. In practice, this would be a small closed-off area that is inaccessible to the human. For this implementation, it is described by the following set of angle constraints:

$$S^B = \left\{ X \in \mathbb{R}^{13} \mid q_2 = \left[-\frac{\pi}{12}, \frac{\pi}{12} \right] \quad q_3 = \left[-\frac{7\pi}{12}, -\frac{5\pi}{12} \right] \right\}$$

The safe set is then simply the union of the backup set and complement of the reachable set of the human in space-time over the duration of the backup control maneuver.

For the purpose of this demonstration, the human is modeled as a single integrator with a maximum velocity, meaning that the size of its reachable set grows linearly in time. By adding time as a state, we prevent the filter from being overly conservative, which would be the result if we only used the reachable set of the human over the time horizon of the backup controller.

If (x_0, y_0) is the current position of the human, the reachable set of the human, or the complement of S, can be simply expressed as an ellipsoid (or an n-cylinder [15]) centered at $(x_0, y_0, H/2)$, where H is the height of the human. We can then write this set as the superlevel set of a time-dependent differentiable function $h : \mathbb{R}^n \to \mathbb{R}$

$$\begin{aligned} h(t, x, y, z) = (x - x_0)^2 + (y - y_0)^2 + \frac{(z - \sqrt{H})^2}{H/(r_0 + v_{\max} t)} \\ - (r_0 + v_{\max} t)^2 \end{aligned}$$

Thus, when $h(t, x, y, z) > 0$, for all points along the robot, the robot is not contacting the human. Similarly to sampling along the backup trajectory, we would theoretically need to check an infinite number of points. Again, however, we can pick a finite number of samples along the robot to enforce this condition. This sampling does not affect the guarantee on safety, as one can simply increase the radius of the human r_0 and the height H_h by the spacing between the points. It does, however, add conservativeness to the problem, so the

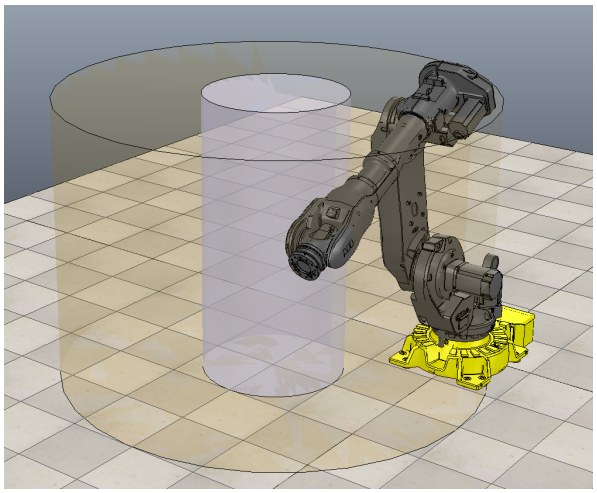

Fig. 3. The set describing the human at t_0 and reachable set after one second.

choice is sampling becomes a tradeoff between computational performance and system performance.

As the dynamics of the robot are defined in joint space, and the safety set is defined in Cartesian space, one must be careful when implementing the barrier condition. Let us define our forward kinematics function, that takes us from joint space to Cartesian space as $K(q,t) : \mathbb{R}^{n+1} \to \mathbb{R}^4$, augmenting it with the identity map for time.

Consider $E = [x, y, z, t]^T$ and $X = [q, \dot{q}, t]^T$. The gradient of h with respect to the states is

$$\frac{\partial h(E)}{\partial X} = \frac{\partial h(K(q))}{\partial X} = \left(\frac{\partial h(E)}{\partial E} \circ K(q)\right) \frac{\partial K}{\partial X}$$

where

$$\frac{\partial K}{\partial X} = \begin{bmatrix} \frac{\partial K}{\partial q} & \frac{\partial K}{\partial \dot{q}} & \frac{\partial K}{\partial t} \end{bmatrix} = \begin{bmatrix} J & \vec{0} & J\dot{q} \\ \vec{0} & \vec{0} & 1 \end{bmatrix}$$

where the Jacobian J is calculated numerically.

B. Backup Controller

For the backup controller, we will leverage the power of the recursive Newton-Euler algorithm (RNEA) [19], which provides the necessary joint torques to generate desired joint accelerations. The flexibility of this method is again showcased by the fact that we do not need an analytic expression for the backup controller, as long as we know its gradient.

There are only two joints that require actuation to reach the backup set. A simple PD controller is used to obtain desired joint accelerations for these joints, which is fed into the RNEA that generates the control inputs, as well as their gradient. The controller is of the form,

$$a_{\text{des}}(q, \dot{q}) = -k_p(q - q_d) - k_d(\dot{q})$$
$$u_{\text{b}}(q, \dot{q}) = \text{RNEA}(q, \dot{q}, a_{\text{des}}(q, \dot{q}))$$

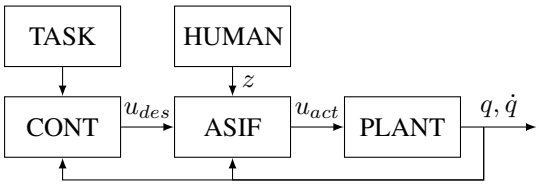

Fig. 4. Block diagram of the ROS nodes used in the simulations.

The gradient of this backup controller, which is required to evaluate (17) online, is described by

$$\frac{\partial u_b}{\partial q} = \frac{\partial \text{RNEA}}{\partial q} + \frac{\partial \text{RNEA}}{\partial a_{\text{des}}}\frac{\partial a_{\text{des}}}{\partial q} = \frac{\partial \text{RNEA}}{\partial q} - k_p\frac{\partial \text{RNEA}}{\partial a_{\text{des}}},$$
$$\frac{\partial u_b}{\partial \dot{q}} = \frac{\partial \text{RNEA}}{\partial \dot{q}} + \frac{\partial \text{RNEA}}{\partial a_{\text{des}}}\frac{\partial a_{\text{des}}}{\partial \dot{q}} = \frac{\partial \text{RNEA}}{\partial \dot{q}} - k_d\frac{\partial \text{RNEA}}{\partial a_{\text{des}}},$$
$$\frac{\partial u_b}{dt} = 0.$$

Since the RNEA provides the exact torques needed to achieve desired joint accelerations, the forward invariance of the backup controller is almost trivially guaranteed under the proper choice of desired joint accelerations.

VI. RESULTS

The rigid body algorithm library used for this simulation is Pinocchio [20]. This C++ library has been shown to be the fastest of its kind, with the Table I illustrating the average computation times of each necessary expression for our robot.

TABLE I
COMPUTATION TIME OF IRB 6640 IN PINOCCHIO

Expression	Time (μs)
Affine forward dynamics ($f(x)$ and $g(x)$)	4
Gradient of closed-loop forward dynamics	42
Backup controller	5
Gradient of backup controller	31

A ROS environment was created to simulate the system, with V-REP used as a visualizer. The ROS package consisted of five nodes: the robotic arm (PLANT), the task giver (TASK), a nominal controller (CONT), the human (HUMAN), and the safety filter (ASIF), connected as shown in Figure 4. Each component of the system ran at 200 Hz on a desktop PC with an Intel 8700k processor. The dynamics were integrated in the plant node via the Boost C++ library, with the `runge_kutta_dopri5` scheme over the timestep of 5 ms.

The controller node tracked a sequence of desired end-effector positions, given to it by the task giver node. Once the system reached the desired position, the task giver would send a new desired location to the system. The RNEA is also used for this tracking controller.

The human node allowed the user to joystick a human, modeled as a single integrator, around the factory floor.

Lastly, the safety filter node handled safety for the system. It takes in the state from the plant and the desired inputs from

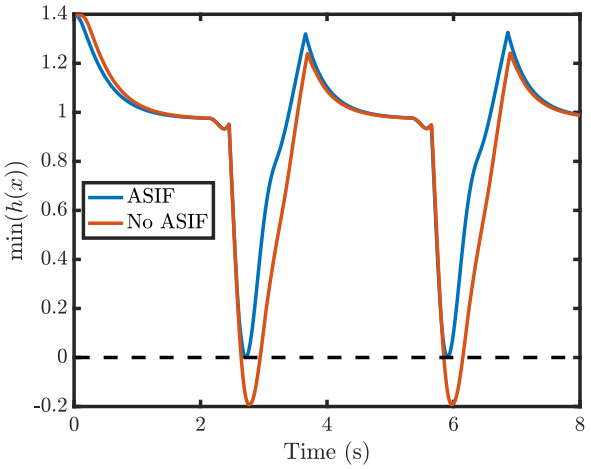

Fig. 5. Value of the Barrier Function with and without ASIF engaged.

the controller, and outputs the actual inputs that are used for integration by the plant.

The ASIF uses an adaptive-step RK4 scheme for integration under the backup controller, and the resulting quadratic program is solved by the OSQP library [21].

Figure 5 shows the value of the ASIF when a human attempts to pass through the arm. This image well illustrates the minimally invasive property of the ASIF, as the filter keeps the value of $h(x)$ just barely above zero. For a more clear demonstration of the filter's capabilities, please see [22]

VII. CONCLUSION

In this paper, we demonstrated how safety can be guaranteed for robotic manipulators in dynamic environments through the use of an implicit active set invariance filter. To do this, we first rigorously showed how this theory can be adapted to practice where only finitely many constraints can be satisfied. Critically, we demonstrated how the filter can handle dynamic environments and time-varying safety sets while requiring no offline computations. We demonstrated the minimally invasive aspect of this filter, while also proving the validity of our numerical scheme, by applying the method to a high-dimensional, complex system whose dynamics were not known analytically. To further improve the method, ways to handle uncertainty in the dynamics and sensing will be explored. To further improve the application to robotic arms, it would be possible to also implement barriers in the joints to prevent self-collision as well as improve the overall safety of the robot and its environment.

REFERENCES

[1] IFR, *International federation of robotics*. [Online]. Available: https://ifr.org/downloads/press2018/Executive_Summary_WR_2018_Industrial_Robots.pdf.

[2] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," in *2008 47th IEEE Conference on Decision and Control*, IEEE, 2008, pp. 4042–4048.

[3] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *International Workshop on Hybrid Systems: Computation and Control*, Springer, 2004, pp. 477–492.

[4] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017. DOI: 10.1109/TAC.2016.2638961.

[5] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[6] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[7] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.

[8] J.-P. Aubin, *Viability Theory*. Birkhäuser Boston, 2009, vol. 1542. DOI: 10.1007/978-0-8176-4910-4.

[9] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron, and A. Ames, "Towards a framework for realizable safety critical control through active set invariance," in *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*, IEEE Press, 2018, pp. 98–106.

[10] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

[11] A. A. Ahmadi and A. Majumdar, "Dsos and sdsos optimization: Lp and socp-based alternatives to sum of squares optimization," in *2014 48th annual conference on information sciences and systems (CISS)*, IEEE, 2014, pp. 1–5.

[12] G. Wu and K. Sreenath, "Safety-critical geometric control for systems on manifolds subject to time-varying constraints,"," *IEEE Transactions on Automatic Control (TAC), in review*, 2016.

[13] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, "An online approach to set invariance," in *Proc. IEEE CDC*, 2018.

[14] J. Carpentier and N. Mansard, "Analytical derivatives of rigid body dynamics algorithms," in *Robotics: Science and Systems (RSS 2018)*, 2018.

[15] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, IEEE, vol. 2, 1985, pp. 500–505.

[16] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential Equations, Dynamical Systems, and an Introduction to Chaos*, 3rd ed. Academic Press, 2012.

[17] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Boston, MA: Birkhäuser Boston, 2008. DOI: 10.1007/978-0-8176-4606-6.

[18] R. Featherstone, "A divide-and-conquer articulated-body algorithm for parallel o (log (n)) calculation of rigid-body dynamics. part 1: Basic algorithm," *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 867–875, 1999.

[19] W. Khalil, "Dynamic modeling of robots using recursive newton-euler techniques," in *ICINCO2010*, 2010.

[20] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library,"

[21] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "Osqp: An operator splitting solver for quadratic programs," in *2018 UKACC 12th International Conference on Control (CONTROL)*, IEEE, 2018, pp. 339–339.

[22] *Video of the simulation.* https://vimeo.com/320906655.