BYU MAGICC Lab Technical Report: MAGICC-2003-Nov-No.2

A Control Lyapunov Function Approach to Safeguarded Shared Control

Wei Ren Randal W. Beard*
Electrical and Computer Engineering Department
Brigham Young University
Provo, Utah 84602
{weiren,beard}@ee.byu.edu

***Abstract*— In this paper, control Lyapunov functions are used to define static and dynamic safe regions for a system. Based on a control Lyapunov function, a "feasible input set" is defined as all feasible controls for this CLF according to attractive behaviors and repulsive behaviors. If the system is within a specified safe region, the human input will be used as the control input to the system. If the system is outside the specified safe region, the human input will be snapped to the closest control element in the feasible input set. Behavior based strategies are applied to achieve smooth transition from human input to snapped control input so as to guarantee maximum flexibility for humans as well as system stability and minimum base-line performance. An illustrative example for a mobile robot shows the effectiveness of the approach.**

I. Introduction

Human-machine interaction as an attractive research area has been studied extensively in the literature with both military and civilian applications. Following the taxonomy in [1], human-machine interaction can be categorized as teleoperation (c.f. [2], [3]), shared control (c.f. [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]), traded control (c.f. [14], [15]), and supervisory control (c.f. [16]).

The main purpose of this paper is to propose a safeguarded control scheme with a human in the loop which not only allows maximum flexibility for humans but also guarantees system stability. This paper falls into the category of safeguarded shared control. While most current approaches to shared control are based on heuristic principles that are not easily analyzed, our approach is built on analytically rigorous control theory. This approach rests on three seemingly loosely related theoretical foundations, that is, control Lyapunov functions (CLFs), shared control frameworks, and reactive control techniques.

Control Lyapunov functions proposed in [17], [18] can be applied to generate stabilizing control laws. In [19], robust control Lyapunov functions are used to derive pointwise min-norm control laws, which are shown to be optimal with respect to some meaningful cost functionals. In [20], satisficing control theory parameterizes a set of asymptotically stabilizing control laws that obey an instantaneous cost-benefit inequality. In [21], control Lyapunov functions are used to define a robot formation so that a constrained motion control problem of multiple systems is converted into a stabilization problem for one single system.

Shared control systems involve both human input and time-invariant control input. In [10], a discrete event framework is developed for integrating a human supervisor into an otherwise time-invariant control system, where time-invariant controller, human interactions, and the plant are designed separately and then combined as appropriate based on mathematical constraints. In [5], [6], [9], the operator has full control of the vehicle in benign situations while the vehicle overrides the user commands in hazardous situations.

Reactive control has the advantage of producing timely response in dynamic and unstructured worlds [22]. Potential fields (c.f. [22], [23], [24], [25], [26]) as a reactive approach to motion planning has received significant attention in the robotics community, where a robot is driven by combining potential fields corresponding to different behaviors. The dynamic window approach (c.f. [27], [28], [29]) achieves collision avoidance by considering the kinematic and dynamic constraints of a robot while maximizing an objective function. In [30], an aspiration-based satisficing framework is used to cast the action search problem as an anytime algorithm, which allows directed searches to be used with time limit constraints.

In this paper, we will use control Lyapunov functions to define static and dynamic safe regions for a system. Based on a control Lyapunov function, a "feasible input set" can be defined accordingly, which corresponds to all feasible controls for this CLF. If the system is within a specified safe region, the human input will be used as the control input to the system even if it is outside the feasible input set. If the system is outside the specified safe region, the human input will be snapped to the feasible input set. In this case, the actual control input to the system will be the closest control element in the feasible input set. Behavior based strategies are applied to guarantee smooth transition from human input to snapped control input. Therefore, this approach gives the human maximum flexibility within the constraints that system stability and minimum base-line performance are guaranteed. This paper applies the concept of CLF to find feasible controls according to attractive behaviors and repulsive behaviors. In this sense, it is similar to the

*Corresponding author.

navigation function in potential field approaches (e.g. [25]). However, the distinction between our approach and potential fields is that potential field approaches apply the negative gradient vector field as the control input, which may not have the same dimension as the required dimension for a control input for a general affine nonlinear system. Our approach is valid for any affine nonlinear system where a CLF can be found. The control is chosen from a set of feasible inputs instead of a single specified one. The main contribution of this paper is to introduce a rigorous analytic approach to analyzing safeguarding shared control problems using Lyapunov techniques. Instead of restricting our attention to navigation or obstacle avoidance scenarios, we focus on the stability issue of general nonlinear systems with a human in the loop; for example, vehicles with complicated nonlinear dynamics or unstable modes. In fact, system stability is a critical concern for any control loop with humans involved. This feature also distinguishes our approach from the previous shared control frameworks and reactive control techniques.

The paper is organized as follows. In Section II, we formalize the feasible input set for a system based on a CLF. In Section III, we propose our CLF approach to safeguarded shared control for both static and dynamic safe regions. Section IV shows a simple illustrative example while Section V contains concluding remarks.

II. CLF based Feasible Input Set

In this section, control Lyapunov functions will be used to define feasible input sets for both time-invariant and time-varying affine nonlinear systems.

Consider a time-varying affine nonlinear system

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

where $x \in I\!R^n$, $u \in I\!R^m$, and $f : I\!R^n \times I\!R_+ \rightarrow I\!R^n$ and $g : I\!R^n \times I\!R_+ \rightarrow I\!R^{n\times m}$ are locally Lipschitz in x and piecewise continuous in t.

Similarly, a time-invariant affine nonlinear system is given by

$$\dot{x} = f(x) + g(x)u. \tag{2}$$

We consider two competing behaviors including an attractive behavior and a repulsive behavior.[1] In the case of attractive behaviors, feasible controls correspond to all stabilizing controls with respect to a particular CLF which can drive x to a reference trajectory $x_r(t)$ for system (1) or regulate state x to a certain state x^* for system (2). In the case of repulsive behaviors, feasible controls correspond to all destabilizing controls with respect to a particular CLF which can drive state x away from a reference trajectory $x_r(t)$ for system (1) or from a certain state x^* for system (2). With regard to these two behaviors, we have the following control Lyapunov function definition for system (1).

Definition 2.1: A continuously differentiable function $V : I\!R^n \times I\!R_+ \rightarrow I\!R$ is a control Lyapunov function (CLF) for system (1) if it is positive definite, decrescent, radially unbounded in x, and satisfies

$$\inf_u \left\{ \rho\frac{\partial V}{\partial t} + \rho\frac{\partial V}{\partial x}\left(f(x,t)+g(x,t)u\right)\right\} \leq -W(x), \tag{3}$$

$\forall x \neq 0$ and $\forall t \geq 0$ where $\rho = 1$ or -1 and $W(x)$ is a continuous positive definite function. Function V corresponds to a stabilizing CLF for $\rho = 1$ and a destabilizing CLF for $\rho = -1$.

The feasible input set for system (1) is defined as

$$F(x) \triangleq \left\{ u : \rho\frac{\partial V}{\partial t} + \rho\frac{\partial V}{\partial x}\left(f(x,t)+g(x,t)u\right) \leq 0 \right\}, \tag{4}$$

where $\rho = 1$ or -1. The set $F(x)$ corresponds to a stabilizing set for $\rho = 1$, denoted as $S(x)$, and a destabilizing set for $\rho = -1$, denoted as $\bar{S}(x)$. Similarly, the asymptotically feasible input set for system (1) is defined as

$$F_a(x) \triangleq \left\{ u : \rho\frac{\partial V}{\partial t} + \rho\frac{\partial V}{\partial x}\left(f(x,t)+g(x,t)u\right) \leq -W(x) \right\}. \tag{5}$$

Note that both $F(x)$ and $F_a(x)$ are guaranteed to be nonempty by Definition 2.1.

For system (2), CLF and feasible input set are defined similarly except that CLF $V(x)$ is not an explicit function of time.

The value of ρ is specified according to design purposes. For the particular example of vehicle navigation, we assume that $V = 0$ at a certain goal and V qualitatively represents how far a vehicle is from its goal. If we want the vehicle to move to this goal, ρ is specified as 1, that is, controls corresponding to negative $\dot{V}$ are feasible. If we want the vehicle to move away from this goal, ρ is then specified as -1, that is, controls corresponding to positive $\dot{V}$ are feasible.

Remark 2.2: Note that Definition 2.1 can be extended to a weak control Lyapunov function (wCLF) definition by allowing V to be positive semi-definite, which may be appropriate for certain applications where only part of the states are of interest.

In the case of $u \in I\!R^2$, at each time t, the line defined by $\dot{V} = 0$ separates the control space (a plane in this case) into two halves, where one half plane represents the feasible input set and the other half represents the unfeasible input set. Of course, if $\frac{\partial V}{\partial x}g = [0,0]$ at a certain time t_1, the feasible input set will be the whole plane given the existence of the CLF.

We have the following lemma regarding the connection between the stabilizing set $S(x)$ and the destabilizing set $\bar{S}(x)$.

[1] Although terms "attractive behavior" and "repulsive behavior" are used exclusively in robot navigation and motion planning literature for goal seeking and obstacle avoidance, here we employ the same terms but refer to general constraints for the system state. In this paper, attractive behavior corresponds to the case when some nonnegative function of the state is constrained to be below a certain upper bound while repulsive behavior corresponds to the case when some nonnegative function of the state is constrained to be above a certain lower bound.

Lemma 2.1: Given a certain CLF without input constraints, if $\frac{\partial V}{\partial x} g(x) \neq 0$ for all $x \neq 0$, then a function V is a stabilizing CLF (destabilizing CLF) if and only if it is also a destabilizing CLF (stabilizing CLF). That is, both $S(x)$ and $\bar{S}(x)$ are nonempty for all $x \neq 0$. Otherwise, if a function V is a stabilizing CLF (destabilizing CLF), it cannot be a destabilizing CLF (stabilizing CLF).

Proof: If $\frac{\partial V}{\partial x} g(x) \neq 0$ for all $x \neq 0$, there always exists a control $u \in \mathbb{R}^m$ to make $\dot{V}$ negative or positive for all $x \neq 0$, that is, both $S(x)$ and $\bar{S}(x)$ are nonempty for all $x \neq 0$.

If $\frac{\partial V}{\partial x} g(x) = 0$ for some $x \neq 0$, a stabilizing CLF implies

$$\frac{\partial V}{\partial x} g = 0 \Longrightarrow \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f < 0.$$

Thus this CLF cannot be a destabilizing CLF. Similarly, a repulsive CLF cannot be a stabilizing CLF in this case. ■

To illustrate Lemma 2.1, consider the following system

$$\begin{aligned} \dot{x}_1 &= -x_1 + u \\ \dot{x}_2 &= -x_2 - u. \end{aligned} \tag{6}$$

It is straightforward to show that $V = \frac{1}{2}(x_1^2 + x_2^2)$ is an stabilizing CLF for system (6). The derivative of V is given by

$$\dot{V} = -x_1^2 - x_2^2 + (x_1 - x_2)u.$$

In the case of $x_1 = x_2 \neq 0$, the stabilizing set $S(x) = \mathbb{R}$ while the destabilizing set $\bar{S}(x) = \emptyset$. That is, in general nonempty $S(x)$ does not imply that $\bar{S}(x)$ is nonempty, and vice versa.

Note that both the feasible input set and asymptotically feasible input set can be extended easily to the case when there are input constraints for the system if a corresponding constrained CLF is known.

III. CLF Approach to Safeguarded Shared Control

In this section, we propose a CLF approach to safeguarded shared control. The basic idea behind this approach is that both system stability and minimum levels of performance are guaranteed in the presence of human inputs.

A. Safeguarded Control for Static Safe Region

Motivated by the properties of CLF, we use it to define safe regions for the shared control problem.

Definition 3.1: A static safe region corresponds to a region of the state space where the system has guaranteed minimum base-line performance.

A simple example of a static safe region for an airplane may correspond to a region of its states (e.g. range of roll angles, velocity, and so forth) where the airplane can fly safely.

Safe regions for system (2) are defined for two different behaviors. In the case of attractive behaviors, a safe region is defined as $R = \{x : V(x) \leq c\}$, where c is a positive constant and can be specified based on the design requirement, and $V(x)$ is a stabilizing CLF (wCLF) for system (2). In the case of repulsive behaviors, a safe region is defined as $\bar{R} = \{x : V(x) \geq c\}$, where c is defined similarly and $V(x)$ is a destabilizing CLF (wCLF) for system (2). In both cases, we assume that the CLF (wCLF) $V(x) = 0$ at $x = x^*$, where x^* is the reference state as mentioned in section II. For example, in a vehicle navigation scenario, x^* can be designed as the goal for the goal seeking behavior or as the center of the hazardous region for the obstacle avoidance behavior. For general cases, x^* can be designed as the state reference to which a system needs to be stabilized or as the state reference of a certain unstable mode for systems with unstable modes to guarantee system stability. If we associate different CLFs (wCLFs) to a system for different design purposes, the overall safe region for the system can be designed correspondingly as an intersection of a group of the two kinds of subregions defined above. We have the following definition for the construction of the overall safe region.

Definition 3.2: Given known CLFs V_i and positive scalars c_i, $i = 1, \cdots, K$, the overall static safe region for system (2) can be constructed as

$$R_s(c) = \left(\bigcap_{i=1}^{J} R_i(c_i) \right) \bigcap \left(\bigcap_{i=J+1}^{K} \bar{R}_i(c_i) \right),$$

where $c = [c_1, \cdots, c_K]^T, R_i$, $i = 1, \cdots, J$, represents the attractive subregion $\{x : V_i(x) \leq c_i\}$, and $\bar{R}_i$, $i = J + 1, \cdots, K$, represents the repulsive subregion $\{x : V_i(x) \geq c_i\}$.

Let $\mathcal{I} = \{1, \cdots, J\}$ and $\bar{\mathcal{I}} = \{J + 1, \cdots, K\}$. Accordingly, define the ith feasible input set corresponding to the ith safe subregion as

$$\begin{aligned} U_i(x) &= \begin{cases} \{u : u \in \mathbb{R}^m\}, & x \in R_i(c_i - \epsilon_i) \\ \{u : \dot{V}_i(x) \leq 0\}, & x \notin R_i(c_i - \epsilon_i) \end{cases} \quad i \in \mathcal{I} \\ \bar{U}_i(x) &= \begin{cases} \{u : u \in \mathbb{R}^m\}, & x \in \bar{R}_i(c_i + \epsilon_i) \\ \{u : \dot{V}_i(x) \geq 0\}, & x \notin R_i(c_i + \epsilon_i) \end{cases} \quad i \in \bar{\mathcal{I}}, \end{aligned} \tag{7}$$

where ϵ_i represents the tolerance for safeguarding with $0 < \epsilon_i < c_i$, $i \in \mathcal{I}$, and $\epsilon_i > 0$, $i \in \bar{\mathcal{I}}$.

We have the following definition for the construction of the overall feasible input set.

Definition 3.3: Given the overall safe region defined in Definition 3.2, the overall feasible input set for system (2) can be constructed as

$$U = (\bigcap_{i=1}^{J} U_i(x)) \bigcap (\bigcap_{i=J+1}^{K} \bar{U}_i(x)),$$

where $U_i(x)$ and $\bar{U}_i(x)$ are given by (7).

Note that in the above Lemma for the overall feasible input set, we did not consider the possible input constraints for the system, which can be incorporated conveniently in Eq. (7)

by constraining $u \in \mathcal{U}$, where $\mathcal{U}$ is the constraint set for the input. Also, the above overall feasible input set can be easily extended to the overall asymptotically feasible input set, denoted as U_a, following Eq. (5).

To illustrate the above safe region and feasible input set concepts, we consider a navigation scenario when a human drives a vehicle in a room. We want to give the human authority and flexibility to drive the vehicle but modify or override his or her control under hazardous situations (e.g. collision). For illustrative purpose, three CLF level curves are plotted in Figure 1. We use a wedge to represent the vehicle. Let $V_i([x, y]),\ i = 1, 2, 3$, where (x, y) is the Cartesian position of the vehicle, be the CLF corresponding to goal (x_i^*, y_i^*) respectively. We assume that $V_i = 0$ at each goal point. We also assume that the vehicle should be constrained inside the level curve $V_1 = c_1$ (e.g. wall avoidance) as well as outside level curves $V_2 = c_2$ and $V_3 = c_3$ (e.g. obstacle avoidance). In this case, the safe region is defined as $R = R_1 \bigcap \bar{R}_2 \bigcap \bar{R}_3$, where $R_1 = \{x : V_1(x) \leq c_1\}$, $\bar{R}_2 = \{x : V_2(x) \geq c_2\}$, and $\bar{R}_3 = \{x : V_3(x) \geq c_3\}$. Accordingly, the overall feasible input set is defined as $U = U_1(x) \bigcap \bar{U}_2(x) \bigcap \bar{U}_3(x)$, where

$$U_1(x) = \begin{cases} \{u : u \in I\!R^2\}, & x \in R_1 \\ \{u : \dot{V}_1(x) \leq 0\}, & x \notin R_1 \end{cases},$$

$$\bar{U}_2(x) = \begin{cases} \{u : u \in I\!R^2\}, & x \in \bar{R}_2 \\ \{u : \dot{V}_2(x) \geq 0\}, & x \notin \bar{R}_2 \end{cases},$$

$$\bar{U}_3(x) = \begin{cases} \{u : u \in I\!R^2\}, & x \in \bar{R}_3 \\ \{u : \dot{V}_3(x) \geq 0\}, & x \notin \bar{R}_3 \end{cases}.$$

Consider another scenario where a pilot controls an airplane as shown in Figure 2. Assume we desire to constrain the flight altitude as $h \in [h_{min}, h_{max}]$ and the bank angle as $|\phi| \leq \phi_{max}$ for system safety. Let $V_h = h^2$ and $V_\phi = \phi^2$ be the weak CLF for altitude control and bank angle control respectively. The safe region in this case is defined as $R = \bar{R}_1 \bigcap R_2 \bigcap R_3$, where $\bar{R}_1 = \{h : V_h \geq h_{min}^2\}$, $R_2 = \{h : V_h \leq h_{max}^2\}$, and $R_3 = \{\phi : V_\phi \leq \phi_{max}^2\}$. Accordingly, the overall feasible input set is defined as $U = \bar{U}_1(x) \bigcap U_2(x) \bigcap U_3(x)$, where

$$\bar{U}_1(x) = \begin{cases} \{u : u \in I\!R^m\}, & x \in \bar{R}_1 \\ \{u : \dot{V}_h(x) \geq 0\}, & x \notin \bar{R}_1 \end{cases},$$

$$U_2(x) = \begin{cases} \{u : u \in I\!R^m\}, & x \in R_2 \\ \{u : \dot{V}_h(x) \leq 0\}, & x \notin R_2 \end{cases},$$

$$U_3(x) = \begin{cases} \{u : u \in I\!R^m\}, & x \in R_3 \\ \{u : \dot{V}_\phi(x) \leq 0\}, & x \notin R_3 \end{cases}.$$

Note that in this case the same CLF V_h corresponds to different feasible input sets for different safe regions.

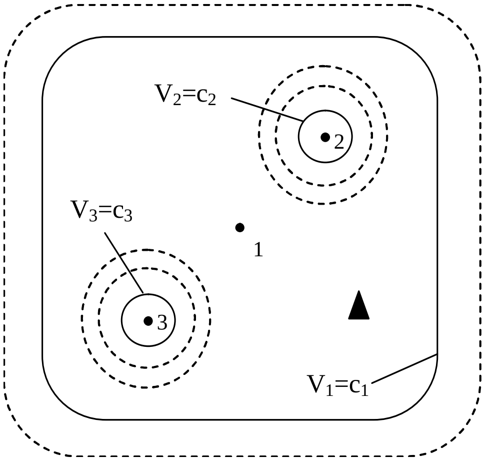

Fig. 1. CLF level curves.

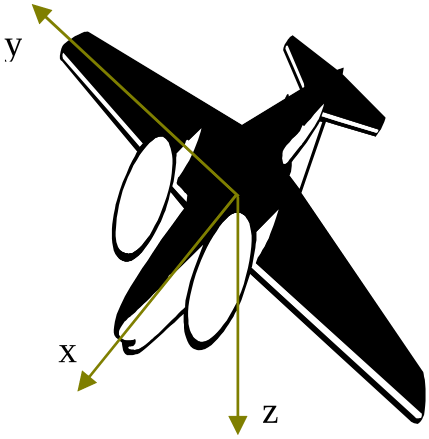

Fig. 2. An airplane with body frame {x,y,z}.

Define the continuous scalar function $k : I\!R^n \to [0, 1]$ as follows:

$$k(x) = \begin{cases} 1, & x \in R_s(c - \epsilon) \\ l(x), & x \in R_s(c) \setminus R_s(c - \epsilon) \text{ and } x \notin B(R_s(c)) \\ 0, & x \notin R_s(c) \text{ or } x \in B(R_s(c)) \end{cases}, \tag{8}$$

where $R_s(\cdot)$ and c are defined in Definition 3.2, $B(\cdot)$ represents the boundary of a set, $\epsilon = [\epsilon_1, \cdots, \epsilon_J, -\epsilon_{J+1}, \cdots, -\epsilon_K]^T$ with $0 < \epsilon_i < c_i$, $i \in \mathcal{I}$ and $\epsilon_i > 0$, $i \in \bar{\mathcal{I}}$, and $l(x) \in [0, 1]$ can be any continuous function in x. The introduction of function $l(x)$ defines a smooth transition from human input to snapped control input when the system starts moving out of the safe region.

Note that $l(x)$ in Eq. (8) may be defined in various ways. One option of defining $l(x)$ is based on the following two sets of functions.

Define two sets of continuous scalar functions $\mathcal{F}_a$ and $\mathcal{F}_r$ as follows:

$$\begin{aligned}\mathcal{F}_a(\epsilon_a, c_a) &= \{f : \mathbb{R}^+ \rightarrow [0,1] | f(r) \\ &= \begin{cases} 1, & 0 \leq r \leq c_a - \epsilon_a \\ p(r), & c_a - \epsilon_a < r < c_a \} \\ 0, & r \geq c_a \end{cases} \qquad (9)\\ \mathcal{F}_r(\epsilon_r, c_r) &= \{f : \mathbb{R}^+ \rightarrow [0,1] | f(r) \\ &= \begin{cases} 0, & 0 \leq r \leq c_r \\ q(r), & c_r < r < c_r + \epsilon_r \}, \\ 1, & r \geq c_r + \epsilon_r \end{cases} \qquad (10)\end{aligned}$$

where c_a and c_r are positive constants, $0 < \epsilon_a < c_a$, $\epsilon_r > 0$, and $p(r), q(r) \in [0,1]$ are monotonically continuous with $p(c_a - \epsilon_a) = 1$, $p(c_a) = 0$, $q(c_r) = 0$, and $q(c_r + \epsilon_r) = 1$.

Accordingly, $l(x)$ can be defined as

$$l(x) = \prod_{i=1}^{K} k_{hi}(V_i(x)), \qquad (11)$$

where $V_i(x)$ is the ith CLF given in Definition 3.2, $k_{hi} \in \mathcal{F}_a(\epsilon_i, c_i)$ for $i \in \mathcal{I}$, and $k_{hi} \in \mathcal{F}_r(\epsilon_i, c_i)$ for $i \in \bar{\mathcal{I}}$. Obviously $p(r)$ and $q(r)$ are easy to define in this case, for example, a linear strictly decreasing function for $p(r)$ and a linear increasing function for $q(r)$.

Define the control input to system (2) as

$$u = (1 - k(x))d(x) + k(x)h(t), \qquad (12)$$

where $k(x)$ is given by Eq. (8), $h(t)$ is the human input, and $d(x)$ is the feasible control snapped from the human input to the feasible input set U given by

$$d(x) = \arg\ \min_{u \in U} \|h - u\|. \qquad (13)$$

With the asymptotically feasible input set U_a, $d(x)$ can also be the asymptotically feasible control given by

$$d(x) = \arg\ \min_{u \in U_a} \|h - u\|. \qquad (14)$$

From Eq. (12), we can see that the control input to the system is the human input when the system state stays in the safe region and is switched to the feasible control when the system state leaves the safe region. When the human input is overridden in unsafe regions, the control input u to the system approximates the human input as closely as possible from the feasible input set. The function $k(x)$ in Eq. (8) can be constructed based on design requirements. The tolerance vector ϵ in Eq. (8) guarantees transition from human input to the feasible control starts when the system state is close to the boundary of the safe region.

Definition 3.4: We say that the safeguarded shared control problem is solved if the following two conditions are satisfied.

- If the system state is initially within the safe region, then the state will stay within the region for all the time and for any human input.
- If the system state is initially outside the safe region, the state will be driven into the safe region within finite time for any human input.

We have the following theorem for safeguarded shared control for static safe regions.

Theorem 3.5: If the control input to system (2) is given by Eq. (12) and $d(x)$ is given by Eq. (14), the shared control problem is solved. Also, if $d(x)$ is given by Eq. (13), the first condition in Definition 3.4 is satisfied. Furthermore, if the human input $h(t)$ is continuous (piecewise continuous), so is the control input u.

Proof:

Condition 1.

Assume that the system state x is initially within R_s and reaches the boundary of R_s at time $t = t_1$.

From Eq. (8), we know that $k(x(t_1)) = 0$ for $x(t_1) \in B(R_s)$. Thus $u = d(x(t_1))$ at $t = t_1$ from Eq. (12).

Given Eq. (13), it is easy to see that $u \in U$ at $t = t_1$. Given Eq. (14), it is easy to see that $u \in U_a$ at $t = t_1$. From the property of the feasible input set and asymptotically feasible input set, we know that $\dot{V}_i(x) \leq 0$ for all $x \in B_i(R_i(c_i))$, $i \in \mathcal{I}$, and $\dot{V}_i(x) \geq 0$ for all $x \in B_i(R_i(c_i))$, $i \in \bar{\mathcal{I}}$, where $B_i(\cdot)$ represents the boundary of the ith safe subregion, at $t = t_1$.

Therefore, the state x cannot leave the safe region R_s for $t \geq t_1$.

Condition 2.

Assume that the system state x is outside the safe region R_s at $t = t_0$. Similar to condition 1, we know that $u = d(x)$ from Eqs. (8) and (12).

Given Eq. (14), it is easy to see that $u \in U_a, \forall x \notin R_s$, which implies that $\dot{V}_i(x) < 0$ for all $x \notin R_i(c_i)$, $i \in \mathcal{I}$, and $\dot{V}_i(x) > 0$ for all $x \notin R_i(c_i)$, $i \in \bar{\mathcal{I}}$.

Therefore, from the standard Lyapunov theory (see e.g. [31]), the state x will be driven to R_s within finite time.

If $h(t)$ is continuous (piecewise continuous), we know that $d(x)$ is also continuous (piecewise continuous) following a similar argument for the continuity of the min-norm control laws in [19].

Therefore, u is also continuous (piecewise continuous) since $k(x)$ is always continuous.

■

B. Safeguarded Control for Dynamic Safe Region

Definition 3.6: A dynamic safe region corresponds to a time-varying region of state space where the system has guaranteed minimum base-line performance.

Similar to a static safe region, a dynamic safe region for attractive behaviors is defined as $R = \{x : V(x,t) \leq c\}$, where $V(x,t) = 0$ at $x = x_r(t)$ and $x_r(t)$ is the reference state trajectory to follow. A dynamic safe region for repulsive behaviors is defined accordingly as $R = \{x : V(x,t) \geq c\}$, where $V(x,t) = 0$ at $x = x_r(t)$ and $x_r(t)$ is the reference state trajectory to avoid.

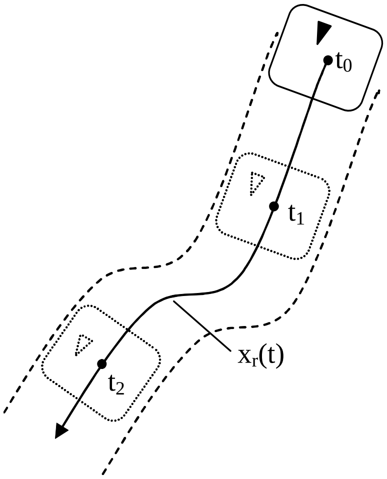

Fig. 3. trajectory following.

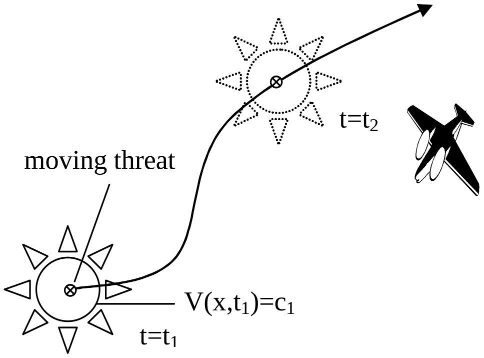

Fig. 4. Dynamic threat avoidance.

Two illustrative examples are shown in Figure 3 and 4 respectively. In Figure 3, a human controls a vehicle to follow a given trajectory $x_r(t)$. As long as the tracking error is below a certain upper bound, the human has the full authority and flexibility to maneuver the vehicle. Once the tracking error is above the upper bound, controls from the feasible input set take over the human input and guarantee minimum tracking performance. We can see that the vehicle (denoted by a wedge) is required to stay within the rectangle area centered at the current trajectory $x_r(t_*)$ at each time t_*. In Figure 4, a human maneuvers an airplane but needs to avoid a moving threat with trajectory $x_r(t)$. Here we assume that the vehicle can measure the current position and velocity of the moving threat. We can see that the airplane is required to stay outside the circle area centered at the current trajectory $x_r(t_*)$ at each time t_*.

For the case of dynamic safe regions, safeguarded control u is defined the same as equation (12) except that $V_i(x)$ is replaced with $V_i(x,t)$. We will not repeat the similar procedures as in Section III-A but consider safeguarded control for the special case of the trajectory following problems.

Given a time-invariant system, a common way to tackle a tracking problem is to perform a state substitution, i.e. define $\tilde{x}(t) = x(t) - x_r(t)$, where $x_r(t)$ is the reference state trajectory to follow. As a result, we have a time-varying system for $\tilde{x}$. Then we need to find a CLF $V(\tilde{x},t)$ for this time-varying system in order to apply the safeguarded control, which may be hard to find. Also, even if such a CLF can be found, a safeguarded control may not be implemented effectively due to the possible input constraints of the vehicle. That is, the trajectory may evolve too fast for the vehicle to track even with the maximum control effort the vehicle can have. Therefore, we introduce the following pointwise CLF and parameterized trajectory ideas motivated by [21], [32] to add feedback from the vehicle to the reference trajectory.

Definition 3.7: A continuously differentiable function $V : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$ is a pointwise control Lyapunov function (pCLF) for system (2) if it is positive definite, radially unbounded, and satisfies

$$\inf_u \left\{ \frac{\partial V(x,s)}{\partial x} \left(f(x,s) + g(x,s)u \right) \right\} < 0, \tag{15}$$

$\forall x \neq 0$ and each constant $s \in [s_1, s_2]$.

Hereafter we assume that a pointwise CLF can be found for a smooth parameterized reference trajectory $x_r(s)$, that is, $V(x,s) = 0$ at $x = x_r(s)$ for each $s \in [s_1, s_2]$. We also define the safe region for a parameterized trajectory following as

$$R_s(c,s) = \{x : V(x,s) \le c\}, \quad s \in [s_1, s_2]. \tag{16}$$

Unlike the static safe region in section III-A, this safe region is time-varying with respect to the parameter s.

Define $\underline{b}(x,s)$ as

$$\underline{b}(x,s) \triangleq \begin{cases} -\sigma(V(x,s)) / \left(\frac{\partial V(x,s)}{\partial x} f \right), & \frac{\partial V(x,s)}{\partial x} g = 0 \\ \mu, & \frac{\partial V(x,s)}{\partial x} g \neq 0 \end{cases},$$

where $\sigma(\cdot)$ is a class $\mathcal{K}$ function and μ can be any given positive number.

The pointwise stabilizing set in this case is defined as

$$S_b(x,s) = \left\{ u : \frac{\partial V(x,s)}{\partial x}(f + gu) \le -\frac{1}{b}\sigma(V(x,s)), b \ge \underline{b} \right\}, \tag{17}$$

where $s \in [s_1, s_2]$.

Lemma 3.1: For each $s \in [s_1, s_2]$, $\underline{b}(x)$ is positive and $S_b(x,s)$ is nonempty for each $x \neq x_r(s)$.

Proof: We follow a similar argument used in [20].

If $\frac{\partial V(x,s)}{\partial x} g = 0$, we know that $\frac{\partial V(x,s)}{\partial x} f < 0$ for each $x \neq x_r(s)$ since $V(x,s)$ is a pointwise CLF. Thus $\underline{b}(x) = -\sigma(V(x,s)) / \left(\frac{\partial V(x,s)}{\partial x} f \right) > 0$ for each $x \neq x_r(s)$.

Also, we know that

$$\begin{aligned} & b \ge \underline{b}(x) > 0 \\ \Longrightarrow & -\frac{1}{b}\sigma(V(x,s)) \ge -\frac{1}{\underline{b}(x)}\sigma(V(x,s)) = \frac{\partial V(x,s)}{\partial x} f, \end{aligned}$$

which means that $S_b(x,s)$ is nonempty when $\frac{\partial V(x,s)}{\partial x}g = 0$.

If $\frac{\partial V(x,s)}{\partial x}g \neq 0$, we know that $\underline{b}(x) = \mu > 0$ and a control u can always be found to satisfy the inequality represented in $S_b(x,s)$. ■

Following [21], let $\dot{s}$ be given by

$$\dot{s} = \begin{cases} \min\left\{\frac{v_0}{\delta + \|\frac{\partial x_r(s)}{\partial s}\|}, \frac{-(\frac{\partial V}{\partial x})^T\dot{x}}{\delta + |\frac{\partial V}{\partial s}|}\left(\frac{\sigma(c)}{\sigma(V(x,s))}\right)\right\}, & s_1 \leq s < s_2 \\ 0, & s = s_2 \end{cases}, \quad (18)$$

where c is the upper bound for the pointwise CLF, $\delta > 0$ is a small positive constant, and v_0 is the nominal velocity for the reference trajectory.

We need the following lemma to show our main theorem.

Lemma 3.2: If $\dot{s}$ is given by (18), then $V(x(t_0), s(t_0)) \leq c$ implies that $V(x(t), s(t)) \leq c$, $\forall t \geq t_0$. Furthermore, if $s \in [s_1, s_2]$ and there is a control $u(x,s)$ and a constant $L > 0$ such that $\frac{-\frac{\partial V}{\partial x}^T\dot{x}}{\sigma(V(x,s))} \geq L$, s will reach s_2 within finite time and $\left\|\frac{dx_r}{dt}\right\| \approx v_0$ for $V(x,s) \ll c$ and $\delta v_0 \ll \left\|\frac{\partial x_r(s)}{\partial s}\right\|$.

Proof: see [21]. ■

Lemma 3.3: Given Eq. (18), for each $s \in [s_1, s_2]$ and $x \neq x_r(s)$, controls chosen from $S_b(x,s)$ guarantee that there exists bounded $b \geq \underline{b}$ such that $\frac{-\frac{\partial V}{\partial x}\dot{x}}{\sigma(V(x,s))}$ has a positive lower bound. Furthermore, controls chosen from $S_b(x,s)$ for bounded b guarantee that $V(x(t), s(t)) \leq c$ for arbitrary $V(x(t_0), s(t_0) > c$ within finite time.

Proof: If $\frac{\partial V(x,s)}{\partial x}g = 0$, we know that $\underline{b}(x,s) = -\sigma(V(x,s))/(\frac{\partial V(x,s)}{\partial x}f)$ is continuous for each $x \neq x_r(s)$ from the property of the pCLF. From the first argument in Lemma 3.2, $\|x\|$ is bounded given Eq. (18) for each $x \neq x_r(s)$. When $x \to x_r(s)$, function $\sigma(\cdot)$ can be chosen so that $\sigma(V(x,s))$ has a higher order than $\frac{\partial V(x,s)}{\partial x}f$, that is

$$\lim_{x \to x_r(s)} -\sigma(V(x,s))/(\frac{\partial V(x,s)}{\partial x}f) = 0.$$

Thus $\underline{b}(x,s)$ is bounded for each $x \neq x_r(s)$.

If $\frac{\partial V(x,s)}{\partial x}g \neq 0$, $\underline{b}(x,s) = \mu$, which is obviously bounded.

Controls chosen from $S_b(x,s)$ satisfies

$$\frac{\partial V}{\partial x}\dot{x} \leq -\frac{1}{b}\sigma(V(x,s))$$
$$\Longleftrightarrow \frac{-\frac{\partial V}{\partial x}^T\dot{x}}{\sigma(V(x,s))} \geq \frac{1}{b},$$

which has a positive lower bound since an upper bounded b can be chosen to be above bounded $\underline{b}(x,s)$.

The last part of the lemma follows a similar argument for Theorem 4.4 in [33]. ■

To choose stabilizing controls from (17), it is desirable to have an explicit expression for a locally Lipschitz continuous b.

Define b as

$$b(x,s) \triangleq \begin{cases} -\sigma(V)/\frac{\partial V}{\partial x}f, & \frac{\partial V}{\partial x}g = 0 \\ \frac{2\frac{\partial V}{\partial x}f + 2\sqrt{(\frac{\partial V}{\partial x}f)^2 + \sigma(V)|\frac{\partial V}{\partial x}g|}}{|\frac{\partial V}{\partial x}g|}, & \frac{\partial V}{\partial x}g \neq 0 \end{cases}, \quad (19)$$

which is locally Lipschitz for each $x \neq x_r(s)$ following a similar argument in [34] and [20].

For safeguarded shared control for parameterized trajectory following, we define the control input as

$$u(x,s) = (1 - k(x,s))d(x,s) + k(x,s)h(t), \quad (20)$$

where $h(t)$ is the human input, $k(x,s)$ is defined similar to Eq. (8) with $R_s(\cdot)$ given by Eq. (16), and $d(x,s)$ is the feasible control snapping the human input to the feasible input set $S_b(x,s)$ with bounded b given by

$$d(x,s) = \arg\min_{u \in S_b(x,s)} \|h - u\|.$$

We have the following theorem for safeguarded shared control for parameterized trajectory following.

Theorem 3.8: If $s \in [s_1, s_2]$, $\dot{s}$ is given by Eq. (18), $u(x,s)$ is given by Eq. (20), the safeguarded shared control problem for parameterized trajectory following is solved. Furthermore, the last two arguments in Lemma 3.2 are also satisfied, that is, finite completion time and given reference trajectory velocity are guaranteed.

Proof: Condition 1 for shared control problem is guaranteed by the definition of $\dot{s}$ following Lemma 3.2. Condition 2 for shared control problem is guaranteed by the last part of Lemma 3.3. The proof for the last part of this theorem follows a combination of Lemma 3.3 and 3.2. ■

C. Discussion

One limitation of our approach is that CLFs may not be easy to find for general affine nonlinear systems although CLFs can be constructed for systems with certain structures [35], for example, feedback linearizable systems and systems with a cascade structure. This limitation can be somewhat mitigated by using weak CLFs if proper for some applications since weak CLFs are relatively easy to find. A concern with any approach for shared control is the computation cost. Although the safe region defined in our approach is an intersection of a group of subregions, we only need to consider the relevant subregions where the system state is close to its boundaries, which can significantly drop the computation cost.

Compared to potential field approaches, our approach generates the control input to the system as a combination of the human input and the feasible control chosen from a feasible input set, where the feasible control from the feasible input set reflects the human's intention best in the sense of minimizing the norm of their difference. For the feedback linearizable models of the mobile robots used in the potential field approaches, the artificial potential function or navigation function itself can be a valid CLF such that a set of controls can be chosen correspondingly. From this perspective, the negative gradient of the potential function or navigation function used as the control is only one element of the feasible input set, which may not reflect the human's intention best in the shared control scenario. For general nonlinear systems, the negative gradient may not have the same dimension as the control input.

As shown in the illustrative example in section III-A, our approach can only represent obstacles with certain regular shapes (e.g. circles and rectangles) for navigation and obstacle avoidance applications. It is possible to transform the level curves of a given CLF to a star shaped level curves and apply the generalized sphere world idea in [25] so that more complicated obstacle shapes can be modelled and a set of feasible controls can be chosen corresponding to various shapes of obstacles.

The main theme of our approach is to preserve human intention as much as possible while guaranteeing system stability. Even with the system state outside the safe region, if the human input is within the feasible set, it will be the input to the system. If the human input is outside the feasible input set, the closest element in the set will be chosen as the input to the system. Therefore, in the case of human neglect, minimum performance (e.g. system stability) is still guaranteed.

IV. Illustrative Example

In this section, we apply the CLF based safeguarded shared control approach to a simple scenario where a mobile robot wanders around a room while needing to avoid walls and obstacles.

Let (x, y), θ, and (v, ω) denote the Cartesian position, orientation, and linear and angular velocity of the robot respectively. The kinematic equations for the robot are

$$\begin{aligned} \dot{x} &= v\cos(\theta) \\ \dot{y} &= v\sin(\theta) \\ \dot{\theta} &= \omega. \end{aligned}$$

We consider the simplified dynamics via feedback linearization [36] for a fixed point off the center of the wheel axis denoted as (x_h, y_h), where $x_h = x + L\cos(\theta)$ and $y_h = y + L\cos(\theta)$. Let $R(\alpha)$ be the rotation matrix given by

$$R(\alpha) = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ \sin(-\alpha) & \cos(\alpha) \end{bmatrix}.$$

The simplified kinematic equations can be represented as

$$\begin{pmatrix} \dot{x}_h \\ \dot{y}_h \end{pmatrix} = R(-\theta) \begin{pmatrix} v \\ L\omega \end{pmatrix}.$$

Setting v and $L\omega$ in the control to

$$\begin{pmatrix} v \\ L\omega \end{pmatrix} = R(\theta) \begin{pmatrix} u_x \\ u_y \end{pmatrix},$$

we obtain

$$\begin{pmatrix} \dot{x}_h \\ \dot{y}_h \end{pmatrix} = \begin{pmatrix} u_x \\ u_y \end{pmatrix}.$$

Similar to Fig. 1, we define CLFs to represent safe regions. For simplicity, it is assumed that the center of the room is the origin. Let $V_1 = \frac{x_h^2}{\lambda^2 - x_h^2} + y_h^2$, where λ is a positive constant, and $V_i = (x_h - x_i^*)^2 + (y_h - y_i^*)^2$, where $i = 2, 3, 4$ and (x_i^*, y_i^*) denotes the center of the obstacle. The level curves of each CLF are shown in Fig. 5, where the level curve of V_1 denotes the boundary of the walls and the level curves of V_2, V_3, and V_4 denote obstacles. In Fig. 5, we choose $V_1 = 36$, where $\lambda = 4$, and $V_i = 0.25$, where $i = 2, 3, 4$, $(x_2^*, y_2^*) = (2, 2)$, $(x_3^*, y_3^*) = (-2, -2)$, and $(x_4^*, y_4^*) = (-2, 4)$.

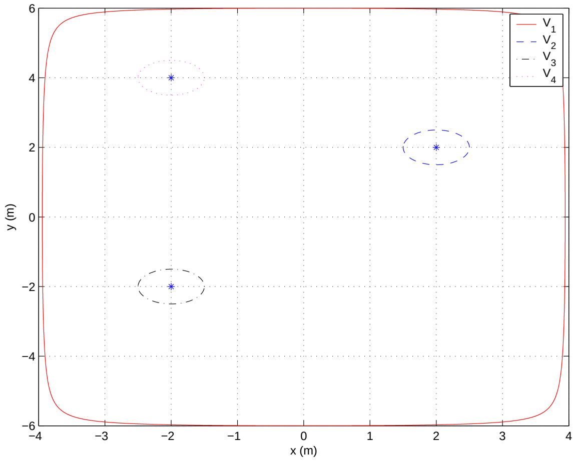

Fig. 5. Safe region defined by CLF level curves for a mobile robot.

To repeatably and systematically test the effectiveness of our approach, we replace the human input with the control described in [37] that produces chaotic robotic trajectories. Fig. 6 shows the trajectory of the robot without applying the safeguarding approach. Note that the trajectory is outside the region defined by the level curve of V_1 sometime. As a comparison, Fig. 7 shows the trajectory of the robot when safeguarding is applied for wall avoidance. We can see that the robot is constrained within the region defined by the level curve of V_1. When the robot approaches the boundary of the wall, its control input is overridden so that a new control input that is closest to the original input in the feasible input set is enforced. As a result, the robot can keep moving forward by following the wall rather than stopping to avoid the wall. Fig. 8 shows the trajectory of the robot when safeguarding for both wall avoidance and obstacle avoidance is applied. It can be seen that the robot never enters the region defined by level curves of V_2, V_3, and V_4.

V. Conclusion

This paper has presented a preliminary result for safeguarded shared control using rigorous Lyapunov stability perspective. We applied control Lyapunov functions to define static and dynamic safe regions and also feasible input sets. The control input was defined as a combination of the human input and the feasible control element that is closest to the human input in the feasible set. We showed that this approach not only gives humans maximum flexibility but guarantees system stability and minimum base-line performance. The

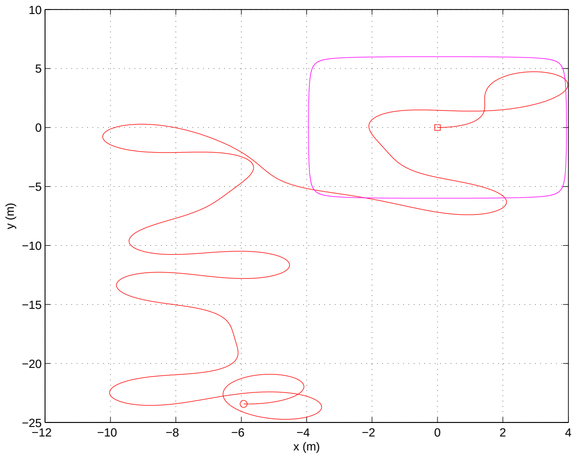

Fig. 6. Trajectory of the robot without safeguarding.

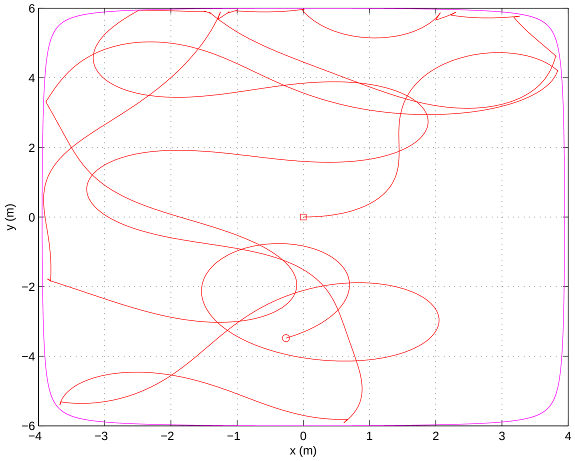

Fig. 7. Trajectory of the robot with safeguarding for wall avoidance.

approach was applied to a simulation study where a mobile robot effectively achieved wall avoidance and obstacle avoidance.

VI. Acknowledgments

This work was funded by AFOSR grants F49620-01-1-0091 and F49620-02-C-0094, and by DARPA grant NBCH1020013.

VII. References

[1] L. Conway, R. Volz, and M. Walker, “Teleautonomous systems: projecting and coordinating intelligent action at a distance,” *IEEE Transactions on Robotics and Automation*, vol. 6, pp. 146–158, April 1990.

[2] T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control*. Cambridge, MA: MIT Press, 1992.

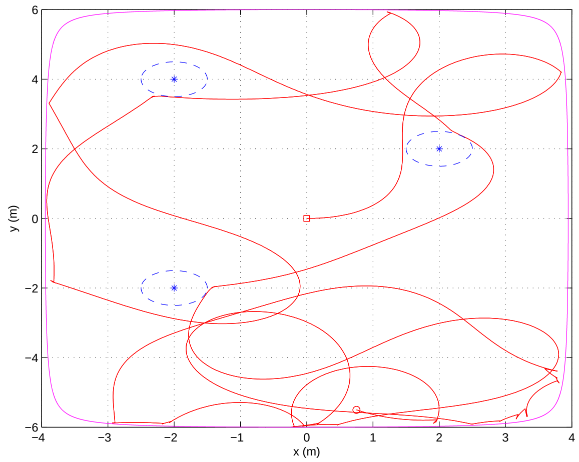

Fig. 8. Trajectory of the robot with safeguarding for both wall avoidance and obstacle avoidance.

[3] I. Elhajj, N. Xi, W. K. Fung, Y.-H. Liu, Y. Hasegawa, and T. Fukuda, “Supermedia-enhanced internet-based telerobotics,” *Proceedings of the IEEE*, vol. 91, pp. 396–421, March 2003.

[4] R. Arkin, “Reactive control as a substrate for telerobotic systems,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 6, pp. 24–31, June 1991.

[5] E. Krotkov, R. Simmons, F. Cozman, and S. Koenig, “Safeguarded teleoperation for lunar rovers: From human factors to field trials,” in *IEEE Planetary Rover Technology and Systems Workshop*, (Minneapolis, MN), 1996.

[6] T. Fong, C. Thorpe, and C. Baur, “A safeguarded teleoperation controller,” in *IEEE International Conference on Advanced Robotics*, (Budapest, Hungary), August 2001.

[7] T. Rofer and A. Lankenau, “Ensuring safe obstacle avoidance in a shared-control system,” in *IEEE International Conference on Emerging Technologies and Factory Automation*, (Minneapolis, MN), pp. 1405–1414, October 1999.

[8] A. Douglas and Y. Xu, “Real-time shared control system for space telerobotics,” *Journal of Intelligent and Robotic Systems*, no. 13, pp. 247–262, 1995.

[9] D. Bell, S. Levine, Y. Koren, L. Jaros, and J. Borenstein, “Design criteria for obstacle avoidance in a shared control system,” in *RESNA Annual Conference*, (Nashville), June 1994.

[10] P. Aigner and B. McCarragher, “Modeling and constraining human interactions in shared control utilizing a discrete event framework,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 30, pp. 369–379, May 2000.

[11] P. Aigner and B. McCarragher, “Shared control framework applied to a robotic aid for the blind,” *IEEE Control Systems Magazine*, vol. 19, pp. 40–46, April

1999.

[12] J. Crandall and M. Goodrich, “Experiments in adjustable autonomy,” in *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1624–1629, 2001.

[13] P. Rybski, S. Stoeter, N. Papanikolopoulos, I. Burt, T. Dahlin, M. Gini, D. Hougen, D. Krantz, and F. Nageotte, “Sharing control [multiple miniature robots],” *IEEE Robotics and Automation Magazine*, vol. 9, pp. 41–48, December 2002.

[14] M. Goodrich and E. Boer, “Designing human-centered automation: trade-offs in collision avoidance system design,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, pp. 40–54, March 2000.

[15] M. Goodrich, E. Boer, and H. Inoue, “A model of human brake initiation behavior with implications for acc design,” in *IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, pp. 86–91, 1999.

[16] D. Pomerleau, “Ralph: rapidly adapting lateral position handler,” in *Proceedings of the Intelligent Vehicles ’95 Symposium*, pp. 506–511, 1995.

[17] Z. Artstein, “Stabilization with relaxed controls,” *Nonlinear Analysis, Theory, Methods, and Applications*, vol. 7, no. 11, pp. 1163–1173, 1983.

[18] E. D. Sontag, “A Lyapunov-like characterization of asymptotic controllability,” *SIAM Journal on Control and Optimization*, vol. 21, pp. 462–471, May 1983.

[19] R. A. Freeman and P. V. Kokotovic, *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques*. Systems & Control: Foundations & Applications, Birkhauser, 1996.

[20] J. W. Curtis and R. W. Beard, “Satisficing: A new approach to constructive nonlinear control,” *IEEE Transactions on Automatic Control*, (to appear).

[21] P. Ogren, M. Egerstedt, and X. Hu, “A control Lyapunov function approach to multiagent coordination,” *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 847–851, October 2002.

[22] R. C. Arkin, *Behavior-Based Robotics*. The MIT Press, 1998.

[23] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *The International Journal of Robotics Research*, vol. 5, pp. 90–98, Spring 1986.

[24] J. Borenstein and Y. Koren, “The vector field histogram - fast obstacle avoidance for mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 278–288, June 1991.

[25] E. Rimon and D. Koditschek, “Exact robot navigation using artificial potential functions,” *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 501–518, October 1992.

[26] J.-C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1992.

[27] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics and Automation Magazine*, vol. 4, pp. 23–33, March 1997.

[28] O. Brock and O. Khatib, “High-speed navigation using the global dynamic window approach,” in *IEEE International Conference on Robotics and Automation*, pp. 341–346, 1999.

[29] R. Simmons, “The curvature-velocity method for local obstacle avoidance,” in *IEEE International Conference on Robotics and Automation*, pp. 3375–3382, 1996.

[30] T. Palmer and M. Goodrich, “Satisficing anytime action search for behavior-based voting,” in *IEEE International Conference on Robotics and Automation*, pp. 1013–1018, 2002.

[31] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ: Prentice Hall, 2nd ed., 1996.

[32] M. Egerstedt and X. Hu, “Formation constrained multiagent control,” *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 947–951, December 2001.

[33] W. Ren, R. W. Beard, and J. W. Curtis, “Satisficing control for multi-agent formation maneuvers,” in *Proceedings of the IEEE Conference on Decision and Control*, (Las Vegas, NV), pp. 2433–2438, December 2002.

[34] M. Krstić and H. Deng, *Stabilization of Nonlinear Uncertain Systems*. Communication and Control Engineering, London: Springer, 1998.

[35] Krstić, Kanellakopaulis, and Kokotovic, *Nonlinear and Adaptive Control Design*. Wiley, 1995.

[36] J. Pomet, B. Thuilot, G. Bastin, and G. Campion, “A hybrid strategy for the feedback stabilization of nonholonomic mobile robots,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Nice, France), pp. 129–134, May 1992.

[37] Y. Nakamura and A. Sekiguchi, “The chaotic mobile robot,” *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 898–904, December 2001.

[38] M. A. Goodrich, R. W. Beard, D. R. Olsen, and T. W. McLain, “Adjustable interaction: Neglect tolerant interaction for multi-robot teams,” Tech. Rep. MARS Robotic Vision 2020, Department of Computer Science, Brigham Young University, Provo, UT 84602, 2001.