2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Unifying Behavior-Based Control Design and Hybrid Stability Theory

Vladimir Djapic[1,2,3], Jay Farrell[3], and Wenjie Dong[3]
[1] NATO Undersea Research Centre (NURC), La Spezia, Italy.
[2]Work performed while at Unmanned Maritime Vehicle (UMV) Lab, SSC-SD, San Diego, CA 92152.
[3] Department of Electrical Engineering, University of California at Riverside, Riverside, CA 92521.

Abstract— **This article describes the design and simulation implementation of a behavior-based control system. The controllers are designed using a command filtered, vector backstepping (CFBS) approach. We use the results from hybrid system control where multiple Lyapunov functions are used in order to prove overall controller stability. We describe our approach using a second order system as a simple example. The article includes design of the control law and simulation based analysis of the performance. The simulation confirms the theoretical result. This method can be applied in designing a control system that can be used for maneuvering of nonholonomic robotic vehicles. As opposed to a single controller approach, we apply a behavior-based control approach with switching among different controllers which perform different tasks. All the behaviors would be used together to complete complex mission goals.**

I. Introduction

Advances in robotic sensor technology, navigation, and control enable advances in vehicle maneuverability and planning strategies to allow scientific and military operators to efficiently test and utilize new sensors in missions of interest. For some searching applications it is a requirement for a vehicle to accurately follow a specific trajectory, make accurate turns and continue to follow the next specified trajectory. Many of these missions require the vehicle to function in complex, cluttered environments, to react to changing environmental parameters, and find a collision-free path through a workspace containing a significant number of obstacles. For successful accomplishment of such a mission, at least two approaches are possible: a single analytic controller or a number of controllers each accomplishing a specific behavior.

An alternative approach to the single controller approach is behavior-based control, wherein each behavior has a single well-defined task. In [1], the behaviors are coordinated in a subsumption architecture. An example is presented in [6]. A set of behaviors to achieve a task and a switching logic coordinating the behaviors can be utilized. These algorithms are criticized due to the lack of rigorous stability analysis [8], [11]. However, when each behavior is implemented as a nonlinear controller with a rigorous stability analysis the main remaining issue is the design of behavior switching. This issue is addressed herein from a hybrid systems perspective.

This article describes a Behavior-Based Command Filtered Backstepping (CFBS) control design using a second order system as a simple example. The model of the second order system is described by eqns. (1–2). The reason that we use the second order system is the analogy of its equations with the kinematic and dynamic equations of motion of robotic vehicles. The variable x_1 is analogous to position while the x_2 variable is analogous to the velocity of a mobile robot. In this article we consider two behaviors:

1) Behavior 1: a controller which controls both states, x_1 and x_2, that defines x_{2c}^o and u to achieve tracking of x_{1c}; and,
2) Behavior 2: a controller which controls the second system state, x_2, through u to achieve tracking of x_{2c}.

Making the comparison with the mobile robot control, the Behavior 1 uses the velocity to control position while the Behavior 2 controls the velocity without regard to the position.

Throughout, this article refers to filtering of a signal x_c^o to produce a bandwidth limited signal x_c and its derivative $\dot{x}_c$. This is referred to herein as command filtering. As outputs of command filter, x_c is continuous, bounded, and differentiable as long as x_c^o is bounded. The design of the command filters used throughout this article is explained in Appendix I. It is important that the commanded trajectories be within the maneuverability and actuation limits of the robots. Actuator and control input saturation can be an issue; however, our command filter can be designed to accommodate magnitude, rate, and bandwidth constraints on the robot states and the actuator signals. This design is beyond the scope of this paper but the can be found in [5].

In this article, we derive and provide proofs for the two theorems that ensure the stability of each behavior, as well as, the stability of the overall hybrid Behavior-Based system. The purpose of this article is to clearly demonstrate our methodology on a simple system. Once demonstrated, this methodology will be applied to a nonholonomic land robot, Autonomous Underwater Vehicle (AUV), and Autonomous Surface Vehicle (ASV) in our future work.

The article is organized as follows. Section II and Section III show the second order system dynamics and derive the control law signals for Behavior 1 and Behavior 2, respectively. Section IV follows the method of [5], [3], [4]. It presents and proves the theorems that guarantee the stability of each behavior designed. Section V describes our method for stable switching among behaviors. Finally, the performance of the control system proposed is illustrated in simulation in Section VI.

978-1-4244-4524-0/09/$25.00 ©2009 AACC

II. Control Signal Derivation: Behavior 1

The objective of Behavior 1 is to force x_1 to track x_{1c} where x_{1c} and x_{1c}^o are known exogenous signals, specified by an operator or Mission Planner. The x_1 state is controlled by using x_2 as an auxiliary control signal. The state x_2 is controlled by the control signal u.

The dynamic equation for the second order system is described as

$$\begin{aligned} \dot{x}_1 &= f_1(x_1) + x_2 && (1)\\ \dot{x}_2 &= f_2(x_1, x_2) + u && (2) \end{aligned}$$

We define tracking and integral errors as

$$\begin{aligned} \bar{x}_1 &= x_1 - x_{1c} && (3)\\ \bar{x}_2 &= x_2 - x_{2c} && (4)\\ \bar{e} &= \int \bar{x}_2 dt. && (5) \end{aligned}$$

The signal x_{2c} and its derivatives $\dot{x}_{2c}$ is generated from the signal x_{2c}^o using the command filters as explained in Appendix I. Define x_{2c}^o as

$$x_{2c}^o = -f_1(x_1) - K_1\bar{x}_1 + \dot{x}_{1c} \quad (6)$$

Differentiating the tracking error $\bar{x}_1$ defined in eqn. (3) we obtain the $\bar{x}_1$ tracking error dynamics

$$\begin{aligned} \dot{\bar{x}}_1 &= \dot{x}_1 - \dot{x}_{1c}\\ &= f_1 + x_2 - \dot{x}_{1c} = f_1 + x_{2c}^o + (x_2 - x_{2c}^o) - \dot{x}_{1c}\\ \dot{\bar{x}}_1 &= -K_1\bar{x}_1 + (x_{2c} - x_{2c}^o) + \bar{x}_2. && (7) \end{aligned}$$

We define the compensated tracking errors as

$$\begin{aligned} \nu_1 &= \bar{x}_1 - \xi_1 && (8)\\ \nu_2 &= \bar{x}_2 - \xi_2 && (9) \end{aligned}$$

where

$$\begin{aligned} \dot{\xi}_1 &= -K_1\xi_1 + (x_{2c} - x_{2c}^o) + \xi_2 && (10)\\ \dot{\xi}_2 &= 0 && (11) \end{aligned}$$

with $\xi_1(0) = 0$ and $\xi_2(0) = 0$. If we then subtract eqn. (10) from eqn. (7) we get the dynamics of the x_1 compensated tracking error

$$\dot{\nu}_1 = -K_1\nu_1 + \nu_2. \quad (12)$$

Differentiating tracking error $\bar{x}_2$ defined in eqn. (4) and following a similar procedure, now using u defined as

$$u = -f_2(x_1, x_2) - K_2\bar{x}_2 - K_i\bar{e} + \dot{x}_{2c} - \bar{x}_2^{bs}, \quad (13)$$

where the backstepping term, $\bar{x}_2^{bs}$, will be defined in eqn. (23) of the stability analysis of Section IV, we obtain the $\bar{x}_2$ tracking error dynamics

$$\begin{aligned} \dot{\bar{x}}_2 &= \dot{x}_2 - \dot{x}_{2c}\\ &= f_2(x_1, x_2) + u - \dot{x}_{2c}\\ \dot{\bar{x}}_2 &= -K_2\bar{x}_2 - K_i\bar{e} - \bar{x}_2^{bs}. && (14) \end{aligned}$$

Subtracting eqn. (11) from eqn. (14) the dynamics of compensated tracking errors for x_2 are

$$\dot{\nu}_2 = -K_2\nu_2 - K_i\bar{e} - \bar{x}_2^{bs}. \quad (15)$$

The stability analysis in Section IV will use eqns. (12) and (15).

III. Control Signal Derivation: Behavior 2

The objective of Behavior 2 is to control x_2. This is implemented by design of the control signal u. While Behavior 2 is active, the state x_1 is not controlled.

Consider only the second state dynamics

$$\dot{x}_2 = f_2(x_2) + u. \quad (16)$$

The control signal u can be defined as

$$u = -f_2(x_2) - K_2\bar{x}_2 - K_i\bar{e} + \dot{x}_{2c}, \quad (17)$$

where $\bar{x}_2$ is the tracking error defined in eqn. (4). Since, the signal $\xi_2 = 0$, thus $\bar{x}_2 = \nu_2$, by substitution of eqn. (17) into the eqn. (16), the compensated tracking error dynamics for this controller is

$$\dot{\nu}_2 = -K_2\nu_2 - K_i\bar{e}. \quad (18)$$

The stability analysis in Section IV will use eqn. (18).

IV. Stability Analysis of Behavior-Based Control Design

This purpose of this section is to show that each of the behaviors implements a stable controller. Two theorems stated and proved in [3] hold. They can be applied to our control design to ensure that it creates closed loop behavior implementations in sense of Lyapunov. For clarity, we will state the theorem for each of the behaviors separately.

Theorem 1: For the system described by eqns. (1–2):

B1. The feedback control law defined in eqns. (6), (10–11), and (13), and the x_{2c} command filter (using the design in Appendix I) provides asymptotic stability for ν_1, ν_2 and boundedness of ξ_1, ξ_2, and $\bar{e}$.

B2. The feedback control law defined by eqn. (17), with $\xi_2 = 0$ provides asymptotic stability for ν_2 and boundedness of $\bar{e}$.

△

This theorem will be proved in this section.

It is possible to derive standard backstepping (BS) controllers for Behavior 1 and Behavior 2. Such approach would require analytic computation of the derivative of each pseudo control signal: x_{1c}^o, x_{2c}^o. This would be straight forward in this simple example, but can be quite complicated for systems with higher state order. A motivation for the CFBS approach is that analytic computation of pseudo-command derivatives is not required. Theorem 2 in [3] shows that the difference between the BS tracking errors denoted by $\tilde{x}$ and CFBS tracking errors denoted by $\bar{x}$ (i.e. $|\bar{x}_1 - \tilde{x}_1|$ and $|\bar{x}_2 - \tilde{x}_2|$) are $\mathcal{O}\left(\frac{1}{\omega_n}\right)$, which shows that the solution of the BS and CFBS implementations can be made arbitrarily close by choice of the command filter natural frequency ω_n.

***Proof:** B1*

Choose the Lyapunov function for ν_1 as

$$V_{x_1} = \frac{1}{2}\nu_1^2.$$

Its derivative along solutions of eqn. (12) is

$$\dot{V}_{x_1} = -K_1\nu_1^2 + \nu_1\nu_2. \tag{19}$$

Combining eqns. (5) and (14) yields

$$\ddot{\bar{e}} + K_2\dot{\bar{e}} + K_i\bar{e} \;\; = \;\; \bar{x}_2^{bs}. \tag{20}$$

We define the tracking error vector as $\bar{\mathbf{q}} = [\bar{e}, \nu_2]^\top$. The tracking error vector dynamic equation is

$$\dot{\bar{\mathbf{q}}} \;\; = \;\; \begin{bmatrix} 0 & 1 \\ -K_i & -K_2 \end{bmatrix}\bar{\mathbf{q}} - \begin{bmatrix} 0 \\ 1 \end{bmatrix}\bar{x}_2^{bs}, \tag{21}$$

where $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -K_i & -K_2 \end{bmatrix}$, and $\mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.
We choose the Lyapunov function for x_2 as

$$V_{x_2} = \frac{1}{2}(\bar{\mathbf{q}}^\top\mathbf{P}\bar{\mathbf{q}}),$$

where $\mathbf{P} = diag([p_1, p_2])$ is a positive definite diagonal matrix. By the analysis in Appendix II

$$\bar{\mathbf{q}}^\top(\mathbf{A}^\top\mathbf{P} + \mathbf{P}\mathbf{A})\bar{\mathbf{q}} = -K_2p_2 < 0.$$

The derivative of Lyapunov function along solutions of eqn. (21) is

$$\begin{aligned} \dot{V}_{x_2} &= \frac{1}{2}(\dot{\bar{\mathbf{q}}}^\top\mathbf{P}\bar{\mathbf{q}} + \bar{\mathbf{q}}^\top\mathbf{P}\dot{\bar{\mathbf{q}}}) \\ &= \frac{1}{2}[\bar{\mathbf{q}}^\top(\mathbf{A}^\top\mathbf{P} + \mathbf{P}\mathbf{A})\bar{\mathbf{q}}] - \bar{\mathbf{q}}^\top\mathbf{P}\mathbf{B}\bar{x}_2^{bs} \\ &= -K_2p_2\bar{x}_2^2 - \bar{x}_2^{bs}p_2\bar{x}_2 \\ &= -K_2p_2\nu_2^2 - \bar{x}_2^{bs}p_2\nu_2. \end{aligned} \tag{22}$$

Define the overall Lyapunov function for Behavior 1 as

$$V_{b_1} = V_{x_1} + V_{x_2}.$$

The time derivative of V_{b_1} along solutions of eqns. (12) and (21) is

$$\dot{V}_{b_1} = -K_1\nu_1^2 + \nu_1\nu_2 - K_2p_2\nu_2^2 - \bar{x}_2^{bs}p_2\nu_2.$$

which is the sum of eqns. (19) and (22). To remove the sign indefinite terms of the above result, we define the backstepping term as

$$\bar{x}_2^{bs} = \frac{\nu_1}{p_2}. \tag{23}$$

With these definitions, the derivative of $V_{b_1}(t)$ satisfies

$$\dot{V}_{b_1} \leq -K_1\nu_1^2 - K_2p_2\nu_2^2. \tag{24}$$

Since the error state is $[\nu_1, \nu_2, \bar{e}]$, the derivative of $V_{b_1}(t)$ is negative semi-definite. This fact proves that the error state is stable and, in particular, $\bar{e}$ is bounded for all $t \geq 0$. LaSalle's invariance theorem (page 128 in [7]), proves that the error state subvector $[\nu_1, \nu_2]$ converges to zero asymptotically.

***Proof:** B2*

Similarly, choosing the Lyapunov function for Behavior 2 as

$$V_{b_2} = V_{x_2} = \frac{1}{2}(\bar{\mathbf{q}}^\top\mathbf{P}\bar{\mathbf{q}}),$$

by analysis similar to that above, the time derivative of V_{b_2} along solutions of eqn. (21), with the backstepping terms $\bar{x}_2^{bs}$ set to zero, satisfies

$$\dot{V}_{b_2} \leq -K_2p_2\nu_2^2. \tag{25}$$

Therefore, each element of $\bar{\mathbf{q}}$ is bounded and ν_2 approaches zero asymptotically.

Remark 1: The purpose of the integrator is to compensate for model error in the dynamics of the x_2 state. If, for example, the actual system is $\dot{x}_2 = f_2 + A + u$ where eqn. (16) has been modified by a model error term A, then a straight forward analysis shows that $\bar{e}(t)$ converges to (A/K_i). We do not discuss model error further herein. Of course, in case of land robot, AUV, or ASV there will be some model error.

V. Switching Analysis of Behavior-Based Control Design

The purpose of this Section is to show that the switching among behaviors does not lead to instability. The main idea is that for overall stability of our Behavior-Based Control design we much ensure three things:

1) Maintain stability during the time each behavior is active.
2) Prevent Zeno Phenomenon.
3) Maintain stability at switching times.

We will show that the CFBS approach allows us to ensure that the Lyapunov function, defined in terms of compensated tracking errors, of the overall switched system is bounded at all times by appropriate choice of the command filter's initial conditions made by a Mission Planner.

A. Behavior Stability

In Section IV we stated the stability properties for each behavior. Using CFBS control design we ensure that during each time interval for which a behavior is active, its compensated tracking errors converge to zero asymptotically. We use a simple notation useful for two behaviors, but the idea generalizes to a larger number of behaviors. Let $i = 0, 1, 2, ...$ denote switching time. Note that i can be expressed as a function of $j = 0, 1, 2, ...$ We assume that Behavior 1 is active initially at $i = 0$, i.e. $t = t_0$. Behavior 1 is active on $t \in [t_{2j}, t_{2j+1}]$ while Behavior 2 is active on $t \in [t_{2j+1}, t_{2j+2}]$. We have already shown that

1. $\dot{V}_{b_1}(t) \leq 0$ and discussed stability properties for the time interval $t \in [t_{2j}, t_{2j+1}]$, and
2. $\dot{V}_{b_2}(t) \leq 0$ and discussed stability properties for the time interval $t \in [t_{2j+1}, t_{2j+2}]$.

B. Zeno Phenomenon

We ensure that the Zeno effect, a situation where the solution of the system makes an infinite number of discrete transitions between behaviors in a finite amount of time, is precluded. This means that we must only allow a finite

number of switches of the behaviors in any finite amount of time. This is accomplished at the mission planning level by ensuring sufficiently long duration of each behavior, for instance, by limiting the rate of change of our command signals. This is discussed in [5] and will not be discussed further herein.

C. Switching Stability

We use the results from hybrid systems theory [9], [10], [2] in order to prove the stability during the time instances of switching among behaviors. As explained in [9], [10], [2], we can ensure stability of the switched system if we can show that the Lyapunov functions for each behavior do not increase at subsequent switching instants.

Given a second order system described by eqns. (1) and (2). We define the following two behaviors:

B1. *The control law for the first behavior is*

$$\begin{aligned} x_{2c}^o &= -f_1(x_1) - K_1\bar{x}_1 + \dot{x}_{1c}, \\ \ddot{x}_{2c} &= \zeta_2\omega_{n_2}\dot{x}_{2c} - \omega_{n_2}(x_{2c} - x_{2c}^o), \\ \dot{\xi}_1 &= -K_1\xi_1 + (x_{2c} - x_{2c}^o), \\ u &= -f_2(x_1, x_2) - K_2\bar{x}_2 - K_i\bar{e} + \dot{x}_{2c} - \bar{x}_2^{bs}. \end{aligned}$$

Mission Planner or operator defines x_{1c}^o and x_{1c}.

B2. *The control law for the second behavior is*

$$\begin{aligned} \ddot{x}_{2c} &= \zeta_2\omega_{n_2}\dot{x}_{2c} - \omega_{n_2}(x_{2c} - x_{2c}^o), \\ u &= -f_2(x_2) - K_2\bar{x}_2 - K_i\bar{e} + \dot{x}_{2c}. \end{aligned}$$

Mission Planner or operator defines x_{2c}^o and x_{2c}.

The variables $\bar{x}_1 = x_1 - x_{1c}$ and $\bar{x}_2 = x_2 - x_{2c}$ are the state tracking errors, while ζ_2, ω_{n_2}, K_1, K_2, K_i are positive design parameters.

The concern for proving stability of the switched system is to show that $V_{b_1}(t_{2j}) \geq V_{b_1}(t_{2j+2})$ and $V_{b_2}(t_{2j+1}) \geq V_{b_2}(t_{2j+3})$. We have now set the notation framework for stating and proving Theorem 2 which ensures that the stability properties are maintained at behavior switching time instances.

Theorem 2:

B1. If, at each start of Behavior 1 ($t = t_{2j}$), the Mission Planner selects the initial values of the filtered command signals to be equal to the current state values, $x_{1c}(t_{2j}) = x_1(t_{2j})$ and $x_{2c}(t_{2j}) = x_2(t_{2j})$, the signals $\xi_1(t_{2j}) = \xi_2(t_{2j}) = 0$, and the integral error to maintain its value (i.e. $\bar{e}(t_{2j}^+) = \bar{e}(t_{2j})$), then the Lyapunov function, V_{b_1}, at each start of each Behavior 1 is a decreasing sequence, $V_{b_1}(t_{2j+2}) \leq V_{b_1}(t_{2j})$.

B2. If, at each start of Behavior 2 ($t = t_{2j+1}$), the Mission Planner selects the initial values of the filtered command signal to be equal to the current state value, $x_{2c}(t_{2j+1}) = x_2(t_{2j+1})$, and the integral error to maintain its value (i.e. $\bar{e}(t_{t_{2j+1}}^+) = \bar{e}(t_{t_{2j+1}})$), then the Lyapunov function, V_{b_2}, at each start of each Behavior 2 is a decreasing sequence, $V_{b_2}(t_{2j+3}) \leq V_{b_2}(t_{2j+1})$.

△

Proof: *Behavior Switching*

The Lyapunov functions for each behavior are defined as

$$V_{b_1} = \frac{1}{2}\nu_1^2 + \frac{1}{2}(\bar{\mathbf{q}}^\top \mathbf{P}\bar{\mathbf{q}})$$

and

$$V_{b_2} = \frac{1}{2}(\bar{\mathbf{q}}^\top \mathbf{P}\bar{\mathbf{q}}),$$

where $\nu_1 = x_1 - x_{1c} - \xi_1$ and $\nu_2 = x_2 - x_{2c} - \xi_2$ are the compensated tracking errors, and the tracking error vector $\bar{\mathbf{q}} = [\bar{e}, \bar{x}_2]^\top$, with each element defined as $\bar{e} = \int \bar{x}_2$ and $\bar{x}_2 = \nu_2$.

B1. If Mission Planner selects the initial values of the filters at each start of Behavior 1 ($t = t_{2j}$) as stated in Theorem 2, we have $\nu_1(t_{2j}) = \nu_2(t_{2j}) = \xi_1(t_{2j}) = \xi_2(t_{2j}) = 0$. Thus, the Lyapunov function, $V_{b_1}(t_{2j}) = \bar{e}^2(t_{2j})$. Since $\dot{V}_{b_1}(t) \leq 0$ for $t \in [t_{2j}, t_{2j+1}]$, we have that $\bar{e}^2(t_{2j}) \geq \bar{e}^2(t_{2j+1})$.

B2. If Mission Planner selects the initial values of filters at each start of Behavior 2 ($t = t_{2j+1}$) as stated in Theorem 2, we have $\nu_2 = \xi_2 = 0$. Thus, the Lyapunov function, $V_{b_2}(t_{2j+1}) = \bar{e}^2(t_{2j+1})$. Since $\dot{V}_{b_2}(t) \leq 0$ for $t \in [t_{2j+1}, t_{2j+2}]$, we have that $\bar{e}^2(t_{2j+1}) \geq \bar{e}^2(t_{2j+2})$.

The hybrid system stability requirement states that the Lyapunov function at the initial occurrence of one subsystem is equal or less than the Lyapunov function at the initial time of the last occurrence of the same subsystem [9], [10], [2]. Following our notation for Behavior-Based hybrid system, the Lyapunov function at the time when one behavior is switched in is equal or greater than the Lyapunov function at the time when the same behavior is switched in next. Since we showed that

$$V_{b_1}(t_{2j}) = \bar{e}^2(t_{2j}) \geq \bar{e}^2(t_{2j+2}) = V_{b_1}(t_{2j+2})$$

for Behavior 1 and

$$V_{b_2}(t_{2j+1}) = \bar{e}^2(t_{2j+1}) \geq \bar{e}^2(t_{2j+3}) = V_{b_2}(t_{2j+3})$$

for Behavior 2 the hybrid stability requirement is met.

D. Summary

The analysis in Section IV showed that the Lyapunov function defined for each behavior is nonincreasing during the time when each of the behaviors is active.The analysis in this section showed that the Lyapunov function defined for each behavior is nonincreasing at behavior switching time instances. Therefore, we can conclude that when perfect modeling of the plant is assumed, the Lyapunov function of each behavior remains zero at all times.

VI. Simulation Results

The purpose of this section is to show the simulation of our controller for the second order plant. We simulated both behaviors and switching between them. Behavior 1 has the goal to cause x_1 to track a signal defined by the Mission Planner to be $x_{1c}^o = sin(t)$. During Behavior 2, the goal is to cause x_2 to track a signal defined by the Mission Planner

to be $x_{2c}^o = 5$. In Behavior 1, control of the x_1 state generates a command for the x_2 state by eqn. (6), while the second state generates control signal u to achieve the x_{2c}^o command according to eqn. (13).

Figures 1–3 present the results of an 60 second simulated mission, during which five switching instances between behaviors occurred. The control law parameters are as follows: $f_1 = f_2 = 1$, $K_1 = 1$, $K_2 = 1$, $K_i = 0.1$, $\zeta_1 = \zeta_2 = 0.9$, $p_2 = 1$ for Behavior 1 and $f_2 = 1$, $K_2 = 0.1$, $K_i = 0.1$, $\zeta_2 = 0.9$, $p_2 = 1$ for Behavior 2.

The plot showing x_1 and x_2 versus time is shown in Figure 1. Each of these two plots contains three curves, for example x, x_c^o, and x_c. Note that during the time period that Behavior 2 is active, x_1 state plot in Figure 1 is not shown, actually zeroed out, since during that behavior x_1 state is not controlled and x_{1c}^o and x_{1c} are undefined. The same was done for the remaining plots in Figures 2–3. Note that for each state, x_c converges to x_c^o at the rate determined by $\omega_{n_1} = 10\frac{rad}{s}$ and $\omega_{n_2} = 100\frac{rad}{s}$ for Behavior 1, and and $\omega_{n_2} = 1\frac{rad}{s}$ for Behavior 2. The signal x_2 converges to and tracks x_{2c} throughout the simulation. The top graph of Figure 2 shows $\bar{x}_1$ and ν_1. Due to the selection of the CF initial condition, as discussed in Section V, $\nu_1(t)$ is zero. During the time interval following behavior switching while x_{2c} converges to x_{2c}^o, $\bar{x}_1$ increases and then converges back toward zero as predicted by the theory. During such time intervals a bounded transient is clearly evident in ξ_1 as seen in the bottom graph of Figure 2.

Figure 3 plots $V_{b_1}(t)$ versus time for Behavior 1 and $\left\|[\bar{x}_1, \bar{x}_2]^\top\right\|$. Since the tracking error for x_1 state is not zero until the x_2 state converges to its desired value, x_{2c}^o, the $\left\|[\bar{x}_1, \bar{x}_2]^\top\right\|$ will increase during that convergence time, as expected. The Lyapunov function defined in terms of the compensated tracking error is decreasing at all times. This result confirms our theoretical conclusion showing that the Lyapunov function of the CFBS approach starts at the value which is a function of the integral error ($\bar{e}$) at the beginning of each behavior, decreases during the duration of each behavior, and maintains its value during the instances of switching between behaviors.

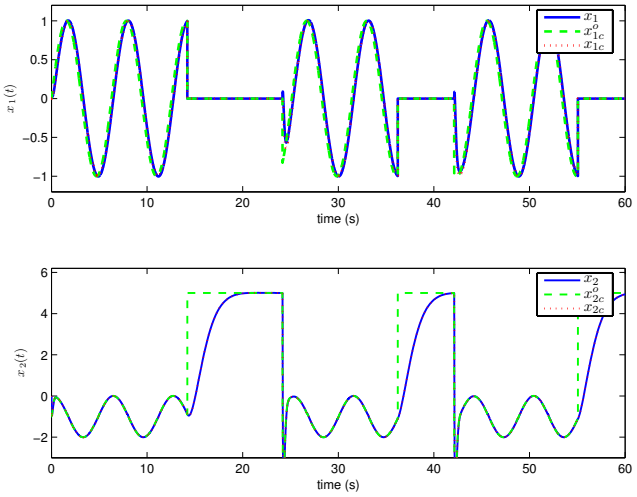

Fig. 1. States x_1 and x_2 vs. time: Blue (solid) line is the actual state, green (dashed) line is the command, and the red (dotted) line is the filtered command.

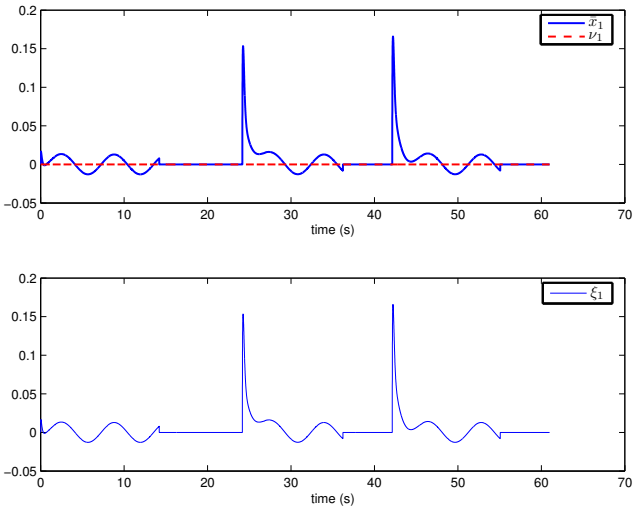

Fig. 2. Top - Error signals $\bar{x}_1$ and ν_1 vs. time. Bottom - Signal ξ_1 vs. time.

VII. Conclusion

In this article we argue and provide simulation based confirmation that some missions can be successfully solved by decomposing the mobile robot control problem in terms of different control behaviors while assuring that the asymptotic stability is maintained. We used results from hybrid control research to design and rigorously show that our behavior-based control approach is provably stable in the sense of Lyapunov. The control design is defined by control behaviors and a logic for switching between the behaviors. We have explicitly proven stability of the switching phenomenon by selecting the behavior thresholds, at mission planning and command filter design level, consistently based on the multiple Lyapunov function. We created two behaviors and the logic of switching between them for the second order system. Our future plans include the similar control design for land robot, AUV, and ASV in which we would extend our design and formulate multiple behaviors for solving some complex missions.

Appendix I
Command Filter

The purpose of this appendix is to provide an example and discussion of a command filter. Advanced control approaches often assume the availability of a continuous and bounded desired trajectory $x_c(t)$ and its first r derivatives $x_c^{(r)}(t)$. The first time that this assumption is encountered it may seen unreasonable, since a user will often only specify a command signal $x_c^o(t)$. However, this assumption can always be satisfied by passing the commanded signal $x_c^o(t)$ through a single-input, multi-output prefilter.

The motivation of command filtering is therefore to de-

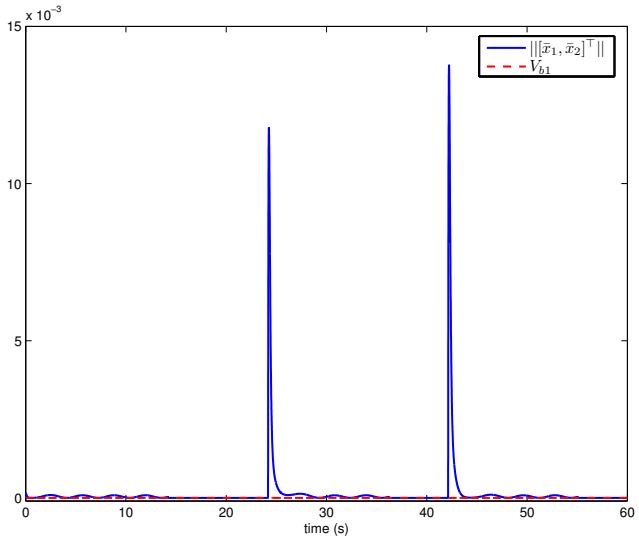

Fig. 3. Lyapunov Function and $||[\bar{x}_1, \bar{x}_2]^\top||$ vs. time.

termine the signals $x_c(t)$ and $\dot{x}_c(t)$ with $|x_c^o(t) - x_c(t)|$ being small, without having to analytically or numerically differentiate x_c^o. The effects of command filtering on the backstepping stability analysis are analyzed in [5], [3], [4]. The summary of that analysis is that for a properly designed command filter (unity DC gain to the first output which is the integral of the second output) the closed-loop command filtered implementation of the backstepping controller will be stable and the tracking error will be $\mathcal{O}\left(\frac{1}{\omega_n}\right)$ where ω_n is the bandwidth of the command filter. Therefore, the effect of command filtering on tracking error can be made arbitrarily small by increasing the parameter ω_n. The choice of ω_n is not dependent on the actuator bandwidth.

The state space implementation of such a filter is

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -2\zeta w_n x_2 - w_n^2 (x_1 - x_c^o) \end{aligned}$$

where $x_c = x_1$ and $\dot{x}_c = x_2$. Note that if x_c^o is bounded, then x_c and $\dot{x}_c$ are bounded and continuous. The transfer function from x_c^o to x_c is

$$\frac{X_c(s)}{X_c^o(s)} = H(s) = \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2} \tag{26}$$

which has a unity gain at low frequencies, damping ratio ζ and undamped natural frequency ω_n. The error $|x_c^o(t) - x_c(t)|$ is small if the bandwidth of $x_c^o(s)$ is less than the bandwidth of $H(s)$. If the bandwidth of x_c^o is known and the goal of the filter is to generate x_c and its derivative with $|x_c^o - x_c|$ small, then the designer simply chooses ω_n sufficiently large.

Note that the signal $\dot{x}_c$ is computed by integration, not differentiation. This helps to decrease the effects of measurement noise; nonetheless, noise will impose a tradeoff in how large of a value can be selected for ω_n.

APPENDIX II
LYAPUNOV EQUATION

The purpose of this appendix is to show that by choosing a specific symmetric and positive definite matrix $\mathbf{P}$ we get a symmetric and positive semidefinite matrix $\mathbf{Q}$ that is useful in the proof of eqn. (21).

We can write the Lyapunov Equation as

$$\mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} = -\mathbf{Q}.$$

Since

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -K_2 & -K_1 \end{bmatrix},$$

and

$$\mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

if we choose

$$\mathbf{P} = \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix},$$

where $K_1 = K_u$ or K_r, $K_2 = K_u^i$ or K_r^i, p_1, and p_2 are positive constants, we have

$$-\mathbf{Q} = \begin{bmatrix} 0 & p_1 - K_2 p_2 \\ p_1 - K_2 p_2 & -2K_1 p_2 \end{bmatrix}.$$

We can choose $p_1 = K_2 p_2$, then $p_1 - K_2 p_2 = 0$. Therefore,

$$\mathbf{P} = \begin{bmatrix} K_2 p_2 & 0 \\ 0 & p_2 \end{bmatrix},$$

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 \\ 0 & 2K_1 p_2 \end{bmatrix},$$

and

$$\mathbf{P}\mathbf{B} = \begin{bmatrix} 0 \\ p_2 \end{bmatrix},$$

which will be used in the derivation of eqn. (22).

REFERENCES

[1] R. A. Brooks. A robust layered control system for a mobile robot. *Journal of Robotics and Automation*, 1986.

[2] R. A. DeCarlo, M. S Branicky, S. Pettersson, and B. Lennartson. Perspectives and results on the stability and stabilizability of hybrid systems. In *Proceedings of the IEEE*, 2000.

[3] J. A. Farrell, M. Polycarpou, M. Sharma, and W. Dong. Command filtered backstepping. *IEEE Trans. on Automatic Control*, 2007.

[4] J. A. Farrell, M. Polycarpou, M. Sharma, and W. Dong. Command filtered backstepping. In *Proc. of the IEEE ACC*, 2008.

[5] J. A. Farrell and M. M. Polycarpou. *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches*. John Wiley, 2006.

[6] Jay A. Farrell, Shuo Pang, and Wei Li. Chemical plume tracing via an autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, 2005.

[7] H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, 3 edition, 2002.

[8] M. Mataric. A distributed model for mobile robot environment-learning and navigation. Master's thesis, MIT, 1990.

[9] Branicky M.S. Stability of switched and hybrid systems. In *Proc. of the IEEE CDC*, 1994.

[10] Branicky M.S. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. on Automatic Control*, 1998.

[11] H. Secchi, V. Mut, R. Carelli, H. Schneebeli, M. Sarcinelli, and T. Freire Bastos. A hybrid control architecture for mobile robots. classic control, behavior based control, and petri nets. Technical report, The Pennsylvania State University CiteSeer Archives, 1999.