2008 IEEE International Symposium on Intelligent Control
Part of 2008 IEEE Multi-conference on Systems and Control
San Antonio, Texas, USA, September 3-5, 2008

Hybrid Control Design Applied in Land Vehicle Behavior-Based Switching Controller

Vladimir Djapic, Jay Farrell, and Wenjie Dong

Abstract— **This article describes the design and simulation implementation of a behavior-based control system that can be used for maneuvering of nonholonomic vehicles. As opposed to a single controller approach which treats tracking a curve with discontinuous yaw, we apply a behavior-based control approach with switching among different controllers for tracking smooth trajectories and yaw control. We use the results from hybrid system control where multiple Lyapunov functions are used in order to prove overall controller stability. The controllers are designed using a command filtered, vector backstepping (CFBS) approach where each controller's objective is to force a specified output to track a desired ideal output. The article includes design of the control law and simulation based analysis of the performance. The simulation confirms the theoretical result.**

I. Introduction

Advances in robotic sensor technology 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.

The single controller approach works well for many applications; however, it might be challenged when multiple control objectives are desired. In that case, a single controller must be optimized to achieve all objectives. There might be a performance trade-off among multiple objectives using a single feedback function. If different controllers are designed and optimized for different objectives, and the switching logic is designed properly, the performance might be better. According to the argument in [7], the performance using a hybrid controller should be no worse than that would be achieved by using any single feedback function without switching. In many cases, the performance using a hybrid controller is substantially better.

V. Djapic is with Unmanned Maritime Vehicle (UMV) Lab, SSC-SD, San Diego, CA 92152, USA and a PhD student at the Department of Electrical Engineering, University of California, Riverside, CA 92521, USA djapic@spawar.navy.mil

J.A. Farrell is a Professor at the Department of Electrical Engineering, University of California, Riverside, CA 92521, USA farrell@ee.ucr.edu

Wenjie Dong a PhD student at the Department of Electrical Engineering, University of California, Riverside, CA 92521, USA dongwj12@yahoo.com

Another solution is behavior-based control, wherein each behavior has a well-defined simple task. In [8], the behaviors are coordinated in a subsumption architecture. An example is presented in [9]. 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 [10], [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.

As a specific example, consider the objective to track a trajectory consisting of intersecting curves, see Figure 1. A control challenge is the requirement to have an accurate transition between curves which intersect at sharp angles. The goal is to track the trajectory γ_1 until the point P_1 is reached. Next the vehicle is supposed to turn to match the direction (tangent) of the next curve, γ_2, at the point P_1. Finally, the vehicle will track the curve γ_2. The process can be repeated for the subsequent curves, for $i = 1, ..., n-1$ to transition from γ_i to γ_{i+1}. This objective can be achieved with two controllers: track curve and turn. This is a very useful process for robotic vehicles in security applications. For interior security, the vehicle may be required to follow a hallway center line while for outdoor security, the vehicle may be required to drive around a fence. In both cases collisions with either a wall or a fence must be avoided, cutting corners are not acceptable since a wall or fence may be there and overshooting the corners may also be unacceptable. We will use the methods from the hybrid systems control literature in order to prove the stability of our behavior-based control approach.

The paper is organized as follows. Section II reviews hybrid control theory, its performance benefits, stability proofs and applies the results from hybrid control to our behavior-based control approach. Section III defines the robot dynamics, describes the formulation of the control problem, and presents discussion of each behavior and switching criteria that was implemented. Our main results, including the control law signals and the stability theorems used are presented in Section IV. Section V presents a specific example and simulation results showing the controller's performance. Section VI is the conclusion. Finally, mission planning needed for implementation of the specific example is explained in Appendix I.

978-1-4244-2224-1/08/$25.00 ©2008 IEEE.

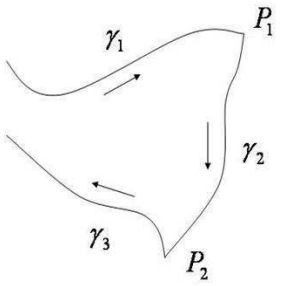

Fig. 1. Objective - Mission Scenario

II. Hybrid Systems (HS), Stability analysis of HS, and Behavior-based control

Some researchers argue that the hybrid control approach using a set of controllers stored in a bank and switching logic between them in order to acquire the desired performance is preferable to a single controller design. Malmborg [5], [6] and McClamroch [7] point out that hybrid control systems can outperform single controller systems and that they can solve problems that can not be dealt with by conventional control, such as when there are multiple design goals that cannot be met by a single controller and can be achieved by using several controllers. There is a set of two or more controllers to choose from in a hybrid control system. Examples in which this strategy is effective includes flight control, air traffic control, missile guidance, process control, robotics etc. Malmborg [5], [6] argues that, for instance, helicopters present dynamical systems with several behaviors of operation. One controller is needed for hovering, one for slow motions and one for fast motions due to major differences between the models appropriate for these two regimes.

There is a large published literature on advantages and stability analysis for Hybrid Systems [7], [13], [14], [15], [16]. Stability of hybrid systems was analyzed using multiple Lyapunov function approach in [12], [14], [15], [16]. The stability of the overall system is ensured if the Lyapunov function is decreasing for each subsystem and if the stability is maintained during switching instances. Fierro [12], Davrazos [13], and Branicky [14], [15] use multiple Lyapunov functions to formulate a theorem for proving asymptotic stability. According to this theorem, stability is guaranteed if a Lyapunov function for each subsystem is non-increasing (meaning that each subsystem is stable) and by requiring 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.

An objective of this article is to present a behavior-based control strategy as means to control an autonomous vehicle. Both the behavior definitions and the behavior switching scheme are discussed. Behaviors and behavioral parameters are selected by a Mission Planner. Each mission behavior includes an algorithm to define a vehicle trajectory that is useful for accomplishing a specific mission task. The behaviors are implemented via nonlinear control algorithms. Similar to Secchi et. al [11] our control architecture combines the advantages of behavior-based and classic control, but at the same time allows for a theoretical study of the stability conditions for the system as used in hybrid control.

We use definitions similar to those in [12], [14], [15] to formulate our model, controllers, and the requirements for stable switching. The proof of last item in the following objective is presented in [12].

Objective:

1) Given b models

$$z_m = h_m(x_m) \quad (1)$$
$$\dot{x}_m = F_m(x_m, u), \quad (2)$$

where z_m are outputs and x_m are states, and b controllers described by

$$u = g_m(x_m, z_m^d), \quad (3)$$

where $m = 1, ..., b$, and z_m^d are the desired outputs,

2) each of which is asymptotically stable with

$$\alpha \|x_m\|^2 \leq V_m(x_m) \leq \beta \|x_m\|^2 \quad (4)$$

and

$$\dot{V}_m(x_m) \leq -c \|x_m\|^2, \quad (5)$$

where V_m is the system's Lyapunov function,

3) we ensure that the Lyapunov function of the system prior to the next switching instant decreases with respect to the Lyapunov function of the system just prior to the previous switching instant

$$V_m(x_m(t_{m_{i+1}})) - V_{mi}(x_m(t_{m_i})) \leq -\phi(x_m), \quad (6)$$

where i is switching time instant and $\phi(x_m)$ is a positive definite function.

If these items are satisfied then the state of the system is globally stable. This can be summarized with the following Theorem.

Theorem 1: If the Lyapunov function is decreasing when an behavior is active and if the Lyapunov function at the time when one behavior is switched in is equal or less than the Lyapunov function at the time when the same behavior is previously switched in then the system is stable in sense of Lyapunov. △

III. Problem Statement and Control Design

The vehicle model, the specific aspects of each behavior that we implemented and the logic criteria for switching between the behaviors are described in this section.

The kinematic and dynamic equations for a land vehicle are described as

$$\dot{x} = u\cos(\psi) \quad (7)$$
$$\dot{y} = u\sin(\psi) \quad (8)$$
$$\dot{\psi} = r \quad (9)$$
$$\dot{u} = g(u,r) + F \quad (10)$$
$$\dot{r} = f(u,r) + \tau \quad (11)$$

where x and y are the earth relative position, ψ is the yaw angle relative to north, u is the forward component of the velocity in body frame, r is the yaw rate in body frame, F is the body-frame control force, τ is the body-frame control moment, $g(u,r)$ and $f(u,r)$ are forces and moments acting on the robot. We also assume that the position and velocity are measured at the center of the horizontal axle. The lateral component of the body frame velocity v is zero, which imposes a nonholonomic constraint on the vehicle motion.

In this paper we consider two behaviors:

1) position tracking along smooth trajectories satisfying Assumptions 1 and 2, called the track curve behavior; and,
2) yaw and speed tracking, called the turn behavior.

The first behavior accomplishes position trajectory tracking. The second behavior enables the robot to make accurate turns before it starts the next path segment by controlling robot's speed and yaw.

A. Control Design

The objective of this section is to explain each control behavior that we implemented.

1) Track Curve behavior: In this behavior, we are interested in the position tracking problem where the objective is to force the system output $z(t) = [x(t), y(t)]^\top \in \Re^2$ to track a desired ideal output $z_d(t) = [x_c(t), y_c(t)]^\top \in \Re^2$. The state variables for this controller are $x_m = [x, y, \psi, u, r]$. For this behavior we make two assumptions

Assumption 1: The desired trajectory is smooth:

$$z_d \in C'.$$

Assumption 2: The desired velocity vector is nonzero:

$$\left\| \begin{array}{c} \dot{x}_c \\ \dot{y}_c \end{array} \right\| \geq \epsilon > 0.$$

Since lateral velocity, v, is not controllable, the track curve nonlinear controller is a trajectory tracking controller that generates horizontal speed, u_c^o, and yaw, ψ_c^o commands. The objective of those controllers is to stabilize the dynamic system of eqns. (7, 8). Following this, backstepping is used to control eqns. (7–11).

2) Turn behavior: In the second behavior, we are interested in the yaw and speed tracking problem where the objective is to force the system output $k(t) = [\psi(t), u(t)]^\top \in \Re^2$ to track a desired ideal output $k_d(t) = [\psi_c(t), u_c(t)]^\top \in \Re^2$. The state variables for this controller are $x_n = [\psi, u, r]$. The reason why this controller is desired is explained by an example in Section I where it is stated that the curves intersect at sharp angles which would violate Assumption 1.

B. Switching Logic

The objective of this section is to explain behavior switching criteria. The stability of the overall system is guaranteed if the Lyapunov function meets the criteria specified in Section II. In this section we address a Zeno phenomenon.

Switching is a logical or decision-making process. Since we have decreasing Lyapunov functions for each continuous subsystem, to guarantee stability we need to put restrictions on switching and to preclude the so called Zeno phenomenon, a situation where the solution of the system makes an infinite number of discrete transitions in a finite amount of time. This means that we must only allow a finite number of switches of the behaviors in a specified amount of time. This is accomplished at the mission planning level. For example, the specified curve γ_2, from point P_1 to point P_2, is being tracked at some desired speed, that can be selected by the Mission Planner to ensure a sufficiently long time duration; therefore, the Zeno phenomenon can be prevented. Similarly, the time for turning process can be managed.

IV. Control Signal Implementation

This section summarizes the control law and the stability properties of the closed loop system. The control law is derived in [1] and [2]. In this section we state Theorems which prove that each sub-controller is stable.

A. Track Curve Behavior

1) Control Law: The following equations describe the control signals for the track curve behavior

$$u_c^o = \gamma \parallel [-K_{xy}\bar{x} + \dot{x}_c, -K_{xy}\bar{y} + \dot{y}_c] \parallel \quad (12)$$
$$\psi_c^o = atan2\left[\gamma(-K_{xy}\bar{y} + \dot{y}_c), \gamma(-K_{xy}\bar{x} + \dot{x}_c)\right] \quad (13)$$
$$r_c^o = -K_\psi\bar{\psi} + \dot{\psi}_c - \bar{\psi}_{bs} \quad (14)$$
$$F = -g(u,r) - K_u\bar{u} - K_u^i\bar{e}_u + \dot{u}_c - \bar{u}_{bs} \quad (15)$$
$$\tau = -f(u,r) - K_r\bar{r} - K_r^i\bar{e}_r + \dot{r}_c - \bar{r}_{bs}, \quad (16)$$

The backstepping terms for track curve behavior are defined as

$$\bar{\psi}_{bs} = g_{\psi_c}^\top B_{\psi_c}^\top \left[\begin{array}{c} v_x \\ v_x \end{array} \right], \bar{u}_{bs} = \frac{A_{\psi_c}^\top \left[\begin{array}{c} v_x \\ v_y \end{array} \right]}{p_{2u}}, \text{ and } \bar{r}_{bs} = \frac{v_\psi}{p_{2r}}.$$

The symbols K_ψ, K_u, K_r, K_u^i K_r^i, p_{2u}, and p_{2r} represent positive design parameters. The symbol K_{xy} is a positive design parameter, which can be time-varying and a function of the state as explained in [1] and [2]. The parameter $\gamma = \pm 1$ is selected by the user. The sign of γ will determine the sign of the commanded speed in body frame; therefore, for $\gamma = 1$ the vehicle drives forward while for $\gamma = -1$ the

vehicle drives backward. The eqns. (12–16) contain certain subscript and superscript notation. For example, in addition to the variable u, we introduce the variables u_c^o and u_c. The symbol u_c^o represents the ideal desired value for u. The symbol u_c represents a filtered version of u_c^o. This notation will also be used similarly to define x_c^o, x_c, y_c^o, y_c, ψ_c^o, ψ_c, r_c^o, and r_c. Given this notation, the tracking error variables are defined as

$$\bar{x} = x - x_c \quad \bar{y} = y - y_c \quad \bar{\psi} = \psi - \psi_c \quad \bar{u} = u - u_c$$
$$\bar{r} = r - r_c \quad \bar{e}_u = \int \bar{u} dt \quad \bar{e}_r = \int \bar{r} dt,$$

and the compensated tracking error variables as

$$v_x = \bar{x} - \zeta_x \quad v_y = \bar{y} - \zeta_y \quad v_\psi = \bar{\psi} - \zeta_\psi$$
$$v_u = \bar{u} - \zeta_u \quad v_r = \bar{r} - \zeta_r.$$

The integral errors are introduced to compensate for plant model uncertainty.

In addition, the control law implements the signals ζ_x, ζ_y, and ζ_ψ using eqns. (17) and (18)

$$\begin{bmatrix} \dot{\zeta}_x \\ \dot{\zeta}_y \end{bmatrix} = \begin{bmatrix} -K_{xy}\zeta_x \\ -K_{xy}\zeta_y \end{bmatrix} + \begin{bmatrix} A_{\psi_c}^\top & B_{\psi_c}^\top g_{\psi_c} \end{bmatrix} \begin{bmatrix} \zeta_u \\ \zeta_\psi \end{bmatrix} + \begin{bmatrix} u_{x_c} - u_{x_c}^o \\ u_{y_c} - u_{y_c}^o \end{bmatrix} \quad (17)$$

$$\dot{\zeta}_\psi = -K_\psi \zeta_\psi + (r_c - r_c^o) + \zeta_r \quad (18)$$

$$\dot{\zeta}_u = 0 \quad (19)$$

$$\dot{\zeta}_r = 0 \quad (20)$$

with $\zeta_x(0) = 0$, $\zeta_y(0) = 0$, $\zeta_\psi(0) = 0$, $\zeta_u(0) = 0$, and $\zeta_r(0) = 0$.

If $\psi_c = \psi_c^o$, $\dot{\psi}_c = \dot{\psi}_c^o$, $u_c = u_c^o$, $\dot{u}_c = \dot{u}_c^o$, $r_c = r_c^o$, and $\dot{r}_c = \dot{r}_c^o$, then eqns. (12–16) would implement a conventional backstepping control law. However, the conventional backstepping approach would require analytic expressions for $\dot{x}_c^o$, $\dot{y}_c^o$, $\dot{\psi}_c^o$, $\dot{u}_c^o$, and $\dot{r}_c^o$, which can be quite complicated, especially when the designer chooses K_{xy} as a function of the state. The command filtered backstepping approach avoids the analytic derivation of these expressions by the use of filters. The filter, with bandwidth determined by a parameter ω_n, the derivation of the CFBS terms ($\bar{\psi}_{bs}$, $\bar{u}_{bs}$, and $\bar{r}_{bs}$), and the terms that are used to define them (A_{ψ_c}, B_{ψ_c}, and g_{ψ_c}) are explained in detail in [1] and [2]. The approach is designed to maintain the exponential stability properties of the backstepping approach for a set of the compensated tracking errors denoted by v_x, v_y, v_ψ, v_u, and v_r, and to ensure that the control signals ψ_c, u_c, and r_c are the same as those of the conventional backstepping approach within an error proportional to $\frac{1}{\omega_n}$.

Including the three filters defined by eqns. (17) and (18), which are designed to compensate for the difference between the desired (ψ_c^o, u_c^o, and r_c^o) and filtered commands (ψ_c, u_c, and r_c), the controller has eleven states: ζ_x, ζ_y, ζ_ψ, ψ_c, $\dot{\psi}_c$, u_c, $\dot{u}_c$, r_c, $\dot{r}_c$, $\bar{e}_u$, and $\bar{e}_r$.

Mission planing of track curve behavior needed for a specific example that we simulated is explained in the Appendix I-A.

B. Turn Behavior

1) Control Law: The turn controller is a subset of the track curve controller. The control law eqns. (14–16) describe the turn controller. The following are the definitions of the backstepping terms for turn behavior: $\bar{\psi}_{bs} = 0$, $\bar{u}_{bs} = 0$, and $\bar{r}_{bs} = \frac{v_\psi}{p_{2r}}$.

Mission planing of turn behavior needed for a specific example that we simulated is explained in the Appendix I-B.

C. Stability

The closed loop system has the stability properties stated in the following theorem and in Theorem 2 in [3], [4].

Theorem 2: For the system described by eqns. (7–11) with the feedback control law given by eqns. (12–16), the following stability properties hold:

1) during the track-curve behavior the error state $v_x, v_y, v_\psi, v_u, v_r$ is exponentially stable,
2) during the turn behavior the error state v_ψ, v_u, v_r is exponentially stable,
3) mission planning switching logic can be designed such that the switched system is stable and the Zeno phenomenon is avoided.

△

Theorem 2 shows that the compensated tracking errors of the 2D command filtered backstepping approach have the same properties as the tracking errors of the standard backstepping approach. LaSalle's theorem proves that that the compensated error state subvector $[v_x, v_y, v_\psi, v_u, v_r]$ converges to zero asymptotically and e_u and e_r are bounded. The detailed proof of this stability result can be found in [1], [2].

Article [3], [4] analyzes the relative performance between the command filtered and conventional backstepping approaches as a function of the bandwidth parameter ω_n of the command filters. The track curve and turn controllers also have the properties stated in Theorem 2 in [3], [4]. This Theorem states that by increasing w_n, the solution to the command filtered backstepping closed-loop system can be made arbitrarily close to the backstepping solution that relies on analytic derivatives. This Theorem is applicable to the behavior based controllers, but not applicable at the switching instances.

V. Simulation Results

In this section we give a specific example where the two control behaviors explained in Sections III-A.1 and III-A.2 are used. The vehicle is required to track a square box. The implementation is done by designing a trajectory tracking controller which enables the vehicle to follow the edge of the box, by designing a controller which will enable the vehicle to turn at the corner of the box, and by designing an appropriate logic which switches between the two controllers.

Figures 2–3 present the results of a simulated mission, during which the vehicle navigates around a [15 x 15] meter box. Only the first 70 seconds is shown since the main

points have been demonstrated by that time. The vehicle is initially at $(0,0)$ with yaw of zero degrees. The turn behavior commands zero speed and turns the vehicle to within a 5° threshold to the desired yaw of the first leg (90°), then the commanded trajectory traverses the box in the clockwise direction with desired speed 0.5 m/s until the vehicle is within 0.1 m of the next corner, end point of the line segment. The control law parameters are as follows: $\omega_n^x = \omega_n^y = 10\frac{rad}{s}$, $\omega_n^\psi = \omega_n^u = \omega_n^r = 40\frac{rad}{s}$, $\zeta = 0.9$, $K_\psi = 2$, $K_u = 10$, $K_r = 10$, $K_u^i = 1$, and $K_r^i = 1$. K_{xy} is set to 2 but can be time-varying as explained in [1].

To show the robustness of the controller to modeling uncertainty, the nonlinear forces and moments are generated for the simulated vehicle as: $g(u,r) = X_u u + X_{uu}|u|u$ and $f(u,r) = X_r r + X_{rr}|r|r$ with $X_u = X_r = 0.1$ and $X_{uu} = X_{rr} = 0.01$. In the controller implementation $g(u,r) = f(u,r) = 0$; however, the results show that the integral control compensates for this modeling uncertainty.

The 2D position plot is shown in Figure 2. The Figure contains three indistinguishable curves: actual trajectory, x, the commanded trajectory, x_c^o, and the filtered commanded trajectory, x_c. Convergence to and maintenance of the trajectory is exhibited in Figure 2. The horizontal speed and yaw angle are shown Figure 3. The tracking performance is accomplished with tracking error after each behavior switch going to zero exponentially as proved in Theorem 2. The Simulink simulation is downloadable from the second authors web site (http://www.ee.ucr.edu/ farrell/).

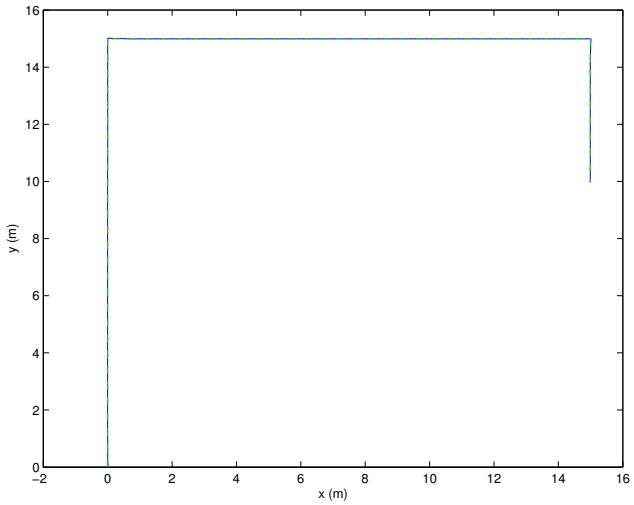

Fig. 2. 2 D Position. The actual vehicle trajectory (solid), the commanded trajectory (dashed), and the filtered command (dotted) start at at $(0,0)$.

VI. 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

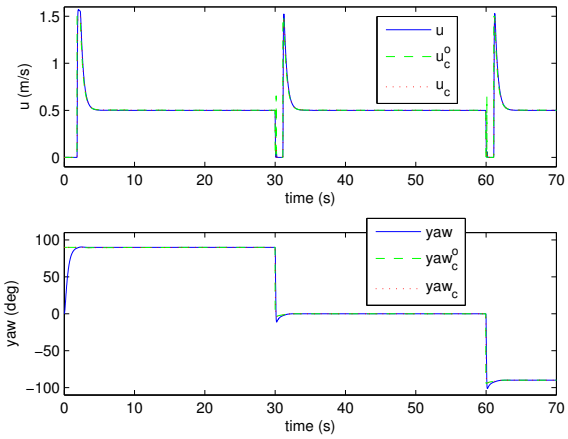

Fig. 3. Speed and Yaw Rate vs. Time: Solid line is the actual vehicle speed/yaw, dashed line is the command, and the dotted line is filtered command.

Lyapunov. The control design is defined by control behaviors and a logic for switching between the behaviors. We have not explicitly proven stability of the switching phenomenon, but expect this to be achievable by selecting the behavior thresholds, at mission planning and command filter design level, consistently based on the multiple Lyapunov function. This will be considered in our future work. We created two behaviors and the logic of switching between them for the simple mission. We argue that this approach can be built on; more mission specific control behaviors can be programmed in order to improve control performance for more complex missions. In situations in which additional behaviors are required, our design enables easy methods for creating new control behaviors. This paper also shows that the CFBS tracking algorithm can be extended to applications with non-holonomic constraints. Our future plans include the similar control design for Autonomous Underwater Vehicle (AUV), in which we would extend our design to 3D and formulate multiple behaviors for solving some complex missions.

Appendix I
Mission Planning Implementation

In this section we explain what is needed to implement the desired mission specified by the example shown in simulation Section V from the mission planning perspective. The Mission Planner must provide the desired trajectory (eg. a square box) and provide the parameters which will determine each behavior and the completion of the behaviors.

A. Mission Planning - Behavior Track Curve

A very commonly used control behavior for the autonomous vehicle is to track a desired trajectory specified by the user. This can be a complex smooth trajectory or piecewise smooth intersecting trajectories. Often, the piecewise smooth trajectories are straight lines. For instance, many search techniques require the vehicle to implement a lawn mower pattern during which an accurate line tracking as well

as sharp vehicle turns are desired. Track Curve can be used to track any smooth curve satisfying Assumptions 1 and 2. In Section V we use it to track a line. Because we track a line from a start to an end point in our example, we design a specific command filter.

The inputs to the track curve controller from the Mission Planner is a start point (x_s, y_s) and the end point (x_e, y_e) of the line. In addition the desired speed of the vehicle is sent to the controller. The speed command, start and end point coordinates are used to generate x_c, y_c, $\dot{x}_c$, and $\dot{y}_c$ signals which generate speed and yaw control signals according to eqns. (12, 13).

The 2D command filter is used to create the command signals x_c, y_c, and their derivatives, $\dot{x}_c$, $\dot{y}_c$ with the desired speed u_c^o, start line point (x_s, y_s), and end line point (x_e, y_e) inputs. The filter output signals are used to control $[x, y]$ by specification of desired values for $[u_c^o, \psi_c^o]$.

Line Command Filter Design:

1) Based on the vector from the start to the end point, $d_x = x_e - x_s$ and $d_y = y_e - y_s$, the trajectory direction is calculated as

$$\psi_T = atan2(d_y, d_x) \tag{21}$$

2) Then, the commanded position and its derivative are defined by integration of

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix} = \begin{bmatrix} u_T \cos\psi_T \\ u_T \sin\psi_T \end{bmatrix}$$

where u_T is the desired vehicle speed specified to avoid the Zeno effect and achieve a mission profile.

B. Mission Planning - Behavior Turn

This behavior can be used to turn the vehicle to a desired yaw at the desired location when the specified trajectory to be followed contains sharp turns. The Mission Planner passes a desired horizontal speed command u_T and calculates desired yaw command ψ_T for the start of the next segment as in eqn. (21).

The inputs to the turn behavior controller are the desired speed and yaw angle. Typically, we choose the desired speed to be zero (other values of u_c^o are also valid). The yaw is calculated using the current vehicle position and the yaw of the next desired line segment. The goal is to align the vehicle yaw with the yaw of the next desired line segment (tangent) as. The main difference between this behavior and the track behavior is the fact that the speed and yaw are commanded, not calculated, and the fact that the position is not controlled.

The desired speed, u_c^o, and yaw, ψ_c^o, commands are send to the command filter which outputs the command signals u_c, ψ_c, and their derivatives, $\dot{u}_c$, $\dot{\psi}_c$. The command filter design, which is used to create these command signals and their derivatives, is explained in detail in [1] and the discussion will not be repeated herein. The maximum yaw rate allowed by the command filter can be adjusted to ensure the Zeno effect is precluded.

ACKNOWLEDGMENT

The authors gratefully acknowledge the Office of Naval Research (ONR) and Space and Naval Warfare Systems Center San Diego's (SSC-SD) In-house Independent Research (ILIR) Program, and the National Science Foundation (NSF) funding under Grant No. ECCS-0701791. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ONR, SSC-SD, or NSF.

REFERENCES

[1] V. Djapic, J. A. Farrell, W. Dong, "Land Vehicle Control Using Command Filtered Backstepping Approach," Submitted to IEEE Transactions on Control Systems Technology Journal, Nov 2007.

[2] V. Djapic, J. A. Farrell, W. Dong, "Land Vehicle Control Using Command Filtered Backstepping Approach," In press for the 2008 American Control Conference, Seattle, WA. (available at http://www.ee.ucr.edu/ farrell/)

[3] J. A. Farrell, M. Polycarpou, M. Sharma, W. Dong, "Command Filtered Backstepping," In press for publication in IEEE Trans. on Automatic Control, March 2007.

[4] J. A. Farrell, M. Polycarpou, M. Sharma, W. Dong, "Command Filtered Backstepping," In press for the 2008 American Control Conference, Seattle, WA.

[5] J. Malmborg, "Analysis and Design of Hybrid Control Systems," PhD thesis, ISRN LUTFD2/TFRT-1050-SE, Department of Automatic Control, Lund Institute of Technology, 1998.

[6] J. Eker, J. Malmborg: "Design and Implementation of a Hybrid Control Strategy." IEEE Control Systems Magazine, vol. 19, number 4, August 1999

[7] McClamroch, N. Harris, Kolmanovski I., "Performance benefits of hybrid control design for linear and nonlinear systems," Proceedings of the IEEE, Vol. 88, No. 7, July 2000.

[8] R. A. Brooks, "A robust layered control system for a mobile robot," IEEE J. Robot. Automation RA-2, pp. 14-23, 1986.

[9] S. Pang, "Reactive Planning and On-line Mapping for Chemical Plume Tracing," Ph.D. thesis, Department of Electrical Engineering, UC Riverside, 2004.

[10] M. Mataric, "A Distributed Model for Mobile Robot Environment-Learning and Navigation," MIT EECS Master's Thesis, Jan 1990, MIT AI Lab Tech Report AITR-1228, May 1990.

[11] H. Secchi, V. Mut, R. Carelli, H. Schneebeli, M. Sarcinelli, T. Freire Bastos, " A Hybrid Control Architecture For Mobile Robots. Classic Control, Behavior Based Control, And Petri Nets (1999)," The Pennsylvania State University CiteSeer Archives, http://citeseer.ist.psu.edu/243751.html

[12] Fierro R., Lewis F.L. "A framework for hybrid control design," IEEE Tr. on Syst., Man, Cyb. Part A, Vol. 26, Nov. 1997.

[13] Davrazos G., Koussoulas N.T., "A review of Stability results for Switched and Hybrid Systems," Proc. 9th Mediterranean Conference on Control and Automation (MED2001).

[14] Branicky M.S. "Multiple Lyapunov functions and other analysis tools for switched and hybrid systems," IEEE Trans. on Automatic Control, Vol. 43, No. 4, April 1998.

[15] Branicky M.S. "Stability of switched and hybrid systems," in Proc. IEEE Conf. Decision and Control, Lake Buena Vista, FL, Dec. 1994.

[16] R. A. DeCarlo, M. S Branicky, S. Pettersson, and B. Lennartson, "Perspectives and results on the stability and stabilizability of hybrid systems," Proc. IEEE, vol. 88, July 2000.