Phase Lag Bounded Velocity Planning for High Performance Path Tracking

Mitchell Hebert, Jay Ming Wong, Roderic A. Grupen
Laboratory for Perceptual Robotics
College of Information and Computer Sciences, University of Massachusetts Amherst
{mhebert, jayming, grupen}@cs.umass.edu

***Abstract*— We propose an algorithm for planning the velocity of a vehicle on a pre-planned path applicable to differentially steered, zero turn radius, mobile robots with symmetric mass distribution about the turn axis. This approach uses estimates of path curvature to maintain tracking precision in the vehicle's heading controller. The longitudinal speed of the vehicle is restricted to limit the bandwidth of the input forcing function (the path). As a result, we guarantee bounds on the driving frequency in the heading controller so that the robot stays close to the intended path. The result allows the robot to use the full performance envelope of the drive motors and provides a principled means of regulating precision and time performance during path traversal. Evaluation of the technique is conducted in simulation and in real experiments on the UMass uBot. Results indicate that the proposed velocity planner can make full use of motor performance and reduces path tracking error and traversal times relative to constant velocity plans.**

I. Introduction

In general, nonholonomic constraints couple path planning, control, and motor constraints and make mobility control an interesting and difficult computational problem. This problem is exacerbated by complex under-actuated kinodynamic systems and sophisticated approximate techniques that exist for computing motor plans (see Section II). In this paper, our focus is *path tracking* for an important class of nonholonomic platforms. In particular, we address wheeled platforms with a symmetric mass distribution around the turn axis and differential steering capable of zero radius turns . We call this class of nonholonomic vehicle *kinodynamically transparent* because the system can track arbitrary paths—even those with path discontinuities—if longitudinal velocity is controlled. For example, large path curvature requires the vehicle to slow in order to keep current vehicle heading aligned to the local path tangent and paths with low (or zero) curvature can exploit the full performance envelope of the vehicle's drive motors.

For these types of systems, path planning and path tracking can be completely decoupled. As a result, the control of longitudinal velocity along a reference path completely determines the driving frequency of inputs to the heading controller. Consequently, lag in the heading controller subjected to this time varying reference input is a nearly exact proxy for tracking precision. Therefore, path planners can attend exclusively to collision avoiding motion plans without concern for kinematic motion constraints and trackability and path trackers can consider tracking precision exclusively for a given path and regulate it by controlling longitudinal velocity.

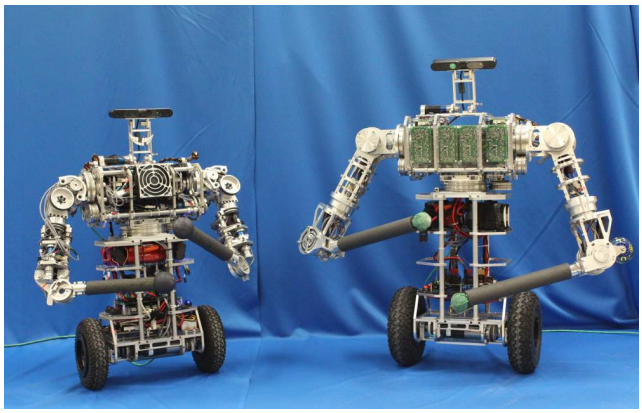


Fig. 1: **The uBot platforms:** uBot5 (left) [1] and uBot6 (right) [2] developed at the UMass Laboratory for Perceptual Robotics (LPR).

In this paper, we consider tracking precision over a given path plan in such systems. Given knowledge of heading control dynamics, path curvature can be used to limit longitudinal velocity so that the heading controller can keep up with the planned path. The approach uses the exact dynamics of the heading controller and the published motor torque constants for the differential drive to create a high performance longitudinal velocity plan.

II. Related Work

Path planning and tracking has been an important area of research which has received an enormous amount of attention in robotics. Practical approaches to path planning include potential field methods [3], [4], [5], and contemporary probabilistic techniques such as PRM and RRT [6], [7], [8], [9]. These techniques (and others) could conceivably be the source of paths for the proposed tracker. In this work, we make no recommendations about which is best in this regard although there are interesting research questions concerning the coupled behavior.

Several techniques have been proposed that consider the fully coupled planning and tracking problems that incorporate dynamics and kinodynamic constraints into integrated, dynamically feasible paths [10], [11]. These techniques can be considered heuristic generalizations of the proposed approach. However, the proposed approach has the added virtue that precise models of heading control dynamics are available for this simple drive configuration and exact motor

dynamics as published by motor manufacturers can be used (Section III-B).

Other curvature-based velocity constraints have been proposed and implemented [12], [13], [14], [15] and a framework based on the probability of collision was demonstrated by [16]. However, these approaches use extremely conservative velocity constraints and hence do not exploit the full performance envelope of the platform. Promising results were shown by Ferguson et al. in the DARPA Urban Challenge with "Boss," Carnegie Mellon University's autonomous SUV, in which curvature was used to match to elements of a set of velocity profiles consisting of constant, linear, linear ramp, and trapezoidal profiles [17]. A second-order spline is generated using the best match velocity profile. However, these approaches do not aim to maximize the vehicle velocity when conditions permit. The approach discussed in this paper provides a control theoretic and dynamical systems account that results in a continuous velocity controller, exploits the full range of the actuator performance in the mobile platform, and yields velocity profiles consistent with the dynamics of DC motors.

The experiments in this paper employ the uBot platform (shown in Fig. 1), a perfect example of a kinodynamically transparent mobile platform. However, there are many other examples of platforms in the literature that can use this approach directly as well, including Herb [18], the Segway RMP[19], and Golem Krang [20] among others. With some extensions, the approach may also be generalized to other zero turn radius platforms as well [21], [22], [23], [24] especially as the capacity for longitudinal velocity on these platforms increases [25].

III. Velocity Constraints

In the longitudinal velocity controller proposed, a "path" is considered to be a function defining the heading of the vehicle $\theta(s)$ that varies with distance along the path s. Such a path *drives* the heading controller with non-stationary reference heading captured by the path curvature $d\theta/ds$.

A. Heading Control Dynamics - Performance Limitations

The heading is assumed to be controlled using a PD feedback controller implemented using a pair of gains, which is a 1-DOF second-order system subject to changing reference headings. Heading control for differential steer, zero turn radius vehicle yields a single degree-of-freedom dynamical system represented as a second-order differential equation,

$$\ddot{\theta} + \frac{B}{I}\dot{\theta} + \frac{K}{I}\theta = 0 \tag{1}$$

where K and B represent the proportional and derivative gains in the heading controller and I is the scalar moment of inertia around the rotation axis. It is assumed that K, B, and I are constant. As a result of this assumption, $\omega_n = \sqrt{K/I}$, or natural frequency of the heading controller, is constant. Equation 1 describes the dynamics of the second-order harmonic oscillator using the homogenous form of the characteristic equation for the controlled plant. For this kind of system, the phase response as a function of ω/ω_n is fully determined by the well-known phase lag diagram illustrated in Figure 2 [26], where $\zeta = B/(2\sqrt{KI})$.

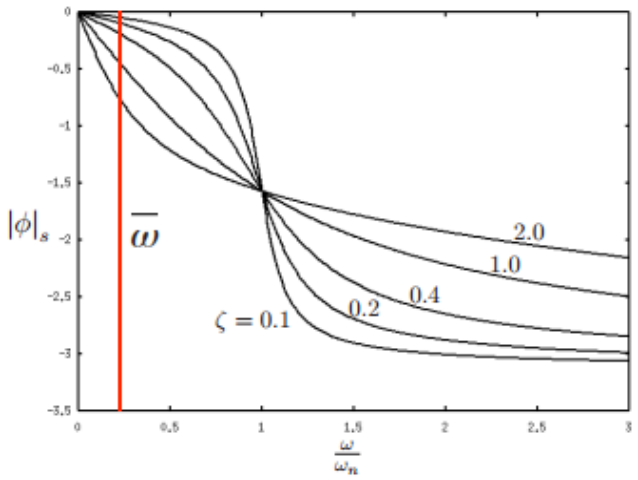


Fig. 2: Phase lag of the second-order heading controller with respect to the dimensionless forcing frequency (ω/ω_n), where ω_n is the natural frequency. The corresponding phase lag ϕ is bounded by limiting $\bar{\omega}$ (red line) given the damping ratio ζ. In this figure, $\bar{\omega} = 0.25$, therefore, for a critically damped system with $\zeta = 1.0$, $\phi \approx -0.4$ rad.

The ability of the heading controller to track a reference path $\theta(s)$ depends on the temporal frequency of the driving function, that depends, in turn, on the vehicle's longitudinal velocity:

$$\omega = \frac{d\theta}{dt} = \frac{d\theta}{ds} * \frac{ds}{dt}. \tag{2}$$

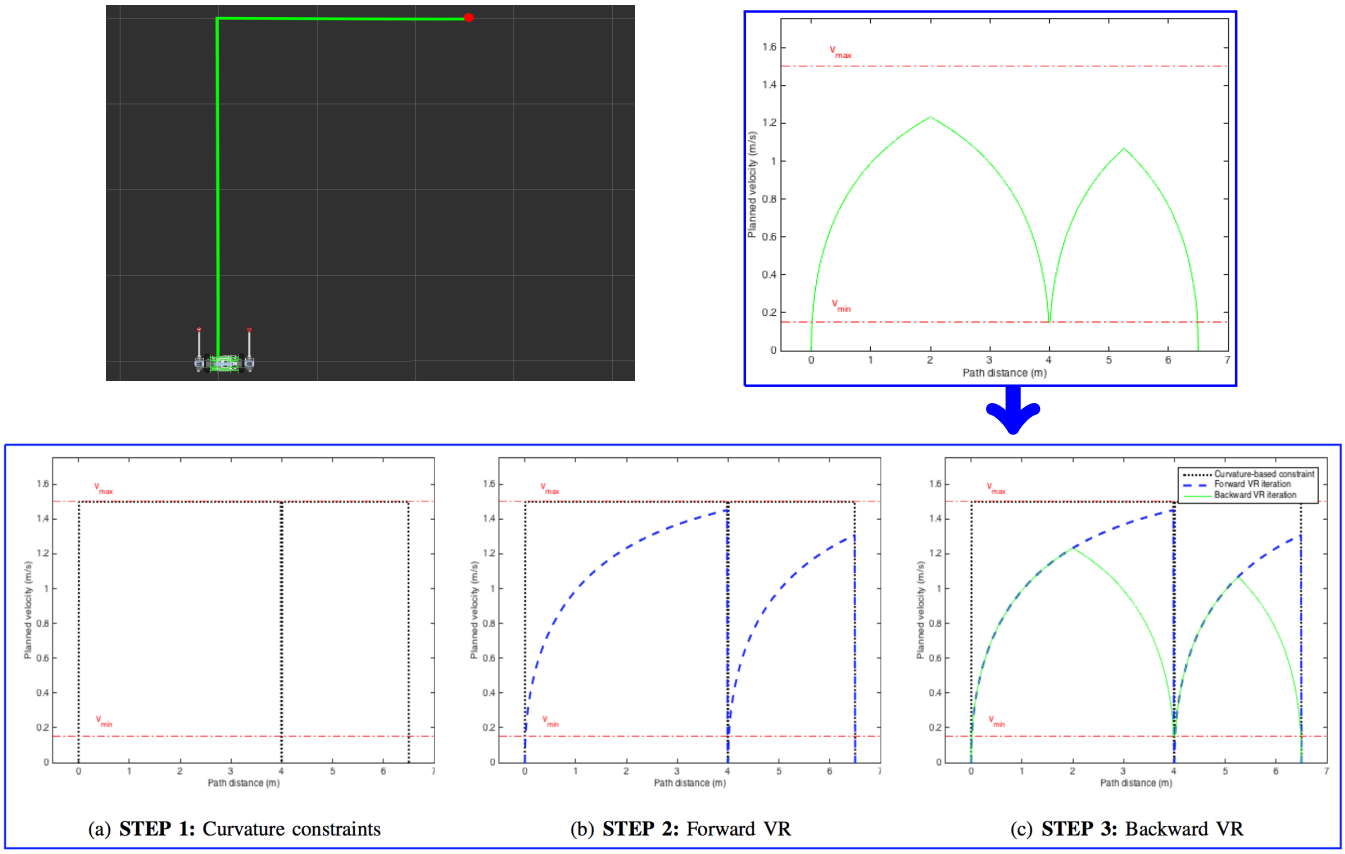
The chain rule in Equation 2 relates the product of the path curvature $d\theta/ds$ prescribed by the path planner and the longitudinal velocity of the platform ds/dt to the driving frequency $\omega = d\theta/dt$ of the forcing function for the heading controller.

As the ratio of forcing frequency ω to ω_n approaches 1, the phase lag in the system, ϕ, approaches $-\pi/2$ (illustrated in Figure 2). Given a specified value for $\zeta = B/2\sqrt{KI}$ in the heading controller, the limiting value of the ratio $\bar{\omega} = \omega/\omega_n$ corresponds to amount of phase lag, ϕ, that is tolerated by the control design. From Equation 2, the maximum command velocity for a specified maximum phase lag and a discrete point in the path s_i becomes:

$$v_i = \left.\frac{ds}{dt}\right|_{s_i} = \frac{\bar{\omega}\,\omega_n}{(d\theta/ds)|_{s_i}} \tag{3}$$

B. Motor Performance Limitations

Heading control and longitudinal acceleration is ultimately constrained by motor performance, in particular, by the no load speed ω_0 [rad / sec] and the stall torque τ_s [Nm] of the wheel motors. These parameters are reported in most DC motor catalogs. Given values for these two parameters, the torque available, for a given motor speed, ω_c is calculated by using Equation (4).



(a) **STEP 1:** Curvature constraints (b) **STEP 2:** Forward VR (c) **STEP 3:** Backward VR

Fig. 3: **The construction of a velocity profile:** the geometry of a hypothetical step function path shown in a top-down view above. Also shown is the final velocity profile which is executable by the robot and the construction of such profile using CBMV (Algorithm 1, Step 1) and VR (Algorithm 2, Steps 2-3).

$$\tau = \tau_s - \omega_c \frac{\tau_s}{\omega_0} \tag{4}$$

The maximum available torque at each velocity is used to compute the maximum available platform acceleration. In addition, we can use ω_0 and the wheel radius, r_{wheel} to obtain the maximum velocity of the platform, $v_{max} = \omega_0 r_{wheel}$ [m/s]. Note that in real systems v_{min} is often necessary to overcome the motor and floor contact stiction. v_{max} and v_{min} can be seen as the two horizontal lines in Figure 3.

Additionally, experiments in Sections IV and V are executed on a balancing robot, which imposes additional restrictions on the application of motor torque. These platforms can not accelerate instantaneously. For example, in order to decelerate, the drive wheels must first accelerate to get in front of the center of mass before deceleration torques can be applied. Our experiments compensate for this restriction by reserving torque by changing the motor models, the details of which are beyond the scope of this paper.

C. Curvature-Based Maximum Velocity

Given a path $\theta(s)$ and a sampling interval Δs , the Curvature-Based Maximum Velocity (CBMV) algorithm (Algorithm 1) computes the maximum velocity the robot can consider at any point along the path given the $\bar{\omega}$ specification. The heading function $\theta(s)$ is computed at the sample interval along the path.

Algorithm 1 Curvature-Based Maximum Velocities (CBMV)

1: **procedure** CBMV($\theta(s), \Delta s$)
2: $\{\theta_0, \theta_1, ..., \theta_n\} \leftarrow Sample(\theta(s), \Delta s)$
3: **for each** θ_i *where* $i \neq \{0, n\}$ **do**
4: $d\theta/ds_i \leftarrow (\theta_{i+1} - \theta_{i-1})/(2\Delta s)$
5: $v_i \leftarrow \begin{cases} v_{max} & \text{if } d\theta/ds = 0, \\ \bar{\omega}\,\omega_n/(d\theta/ds)_i & \text{otherwise} \end{cases}$
6: $V \leftarrow \{v_0, v_1, ..., v_n\}$
7: **return** V

Algorithm 1 estimates the curvature using finite differences and assigns the curvature-limited longitudinal velocity at each sample point in the path.

Consider a test path (pictured in Figure 3) sampled at $\Delta s = 0.01\ m$ for the infinite curvature right angled turn where two linear path segments of length $4.0\ m$ and $2.5\ m$, respectively, are separated by an infinite curvature, 90 degree right turn. The straight line segments have zero curvature and the point of intersection between these two segments is a

unit step heading change reference of magnitude $-\pi/2$ and therefore, introduces infinite curvature $d\theta/ds$ at this point. All other positions on the path have zero curvature. The fine dotted line in the velocity profiles show the result of Algorithm 1 for the unit step heading path. It defines a maximum velocity at each point along the path. Because $d\theta/ds = 0$ for the two straight sections of the path, Algorithm 1 sets the maximum velocity to v_{max} and the point with infinite curvature to ≈ 0.

However, the output of Algorithm 1 does not yield executable velocities—the acceleration limits of the motors preclude the aggressive accelerations resulting from this test path. To make the plan executable, the ability of the wheel motors to generate accelerations must be considered. This is the role of the motor models and the Velocity Reachability Algorithm.

D. Velocity Reachability Algorithm

The Velocity Reachability (VR) algorithm (Algorithm 2) is used to guarantee that all planned velocities are reachable given motor constraints. A curvature-based velocity planned at position $i+1$ may not be reachable from the state of vehicle motion at position i given the acceleration limits of the robot. We assume that the robot has an initial velocity and goal velocity of $v_0, v_n = 0$, though VR can be modified to accept arbitrary v_0, v_n.

Algorithm 2 Velocity Reachability (VR)

procedure VR(V, Δs, v_{min}, v_{max})
  $V \leftarrow \{v_0, v_1, v_2, ..., v_n\}$
  $v_0, v_n \leftarrow 0$
  for $i \leftarrow 0$ *to* $n-1$ **do**
    $a \leftarrow a_{max}|v_i$
    $v' \leftarrow \sqrt{v_i^2 + 2a\Delta s}$
    $v_{i+1} \leftarrow \min(v_{i+1}, v', v_{max})$
  for $i \leftarrow n$ *to* 1 **do**
    $a \leftarrow a_{max}|v_i$
    $v' \leftarrow \sqrt{v_i^2 - 2a\Delta s}$
    $v^*_{i-1} \leftarrow \max(\min(v_{i-1}, v'), v_{min})$
  $V^* \leftarrow \{v^*_0, v^*_1, ..., v^*_n\}$
  return V^*

To calculate the velocity at $i+1$ given i, we use the standard equations of motion:

$$v_{i+1}^2 = v_i^2 + 2a\Delta s \tag{5}$$

where a is the acceleration limit computed from the motor model. At each sample position i along the path, VR guarantees that,

$$v_{min} \leq \sqrt{v_i^2 - 2a\Delta s} \leq v_{i+1} \leq \sqrt{v_i^2 + 2a\Delta s} \leq v_{max}. \tag{6}$$

Velocities are planned, subject to (6) using the two-pass VR algorithm (Algorithm 2), which takes the curvature-based velocity limits V, the sampling interval Δs, and the maximum and minimum velocity, v_{max} and v_{min} as input and generates the maximum achievable (reachable) velocity plan V^*. The results of the forward and backwards iterations of the VR algorithm are shown in Figure 3. Since the accelerations are based on the motor models of the system, the resulting velocity plan is reachable and can be used as velocity commands for the robot during execution.

The final Phase Lag Bounded Velocity (PLBV) planner, used to generate and plan velocities given a path function $\theta(s)$ and $\bar{\omega}$, is done by executing Algorithm 1 and 2 (CBMV and VR) in sequence.

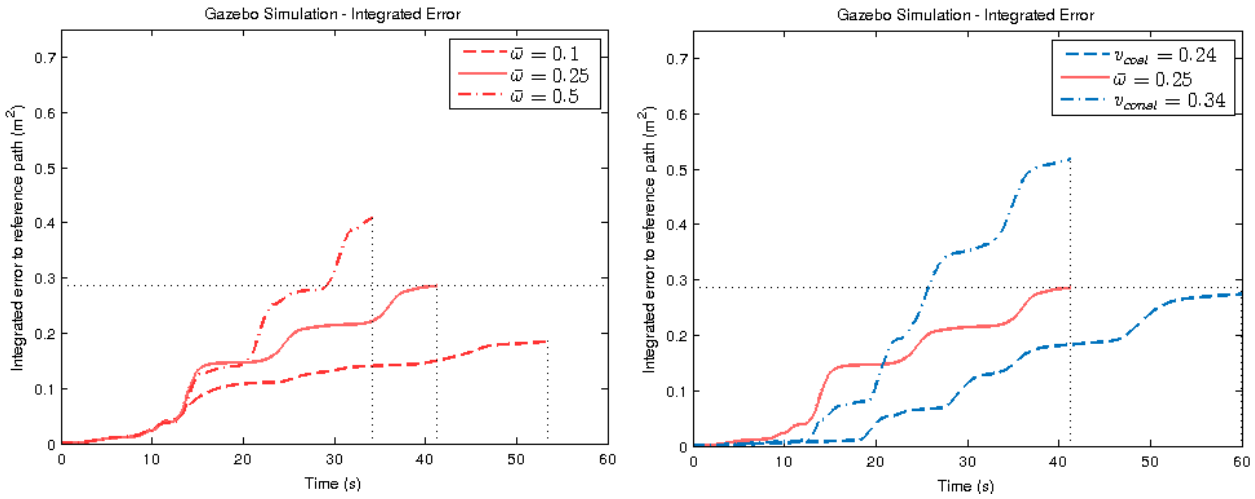
IV. Simulator Results

Our proposed approach PLBV, was tested on a dynamic simulation of the uBot platform in Gazebo. For both the simulation and real robot results, we measure the performance of PLBV in total traversal time and integrated error. Integrated is defined as an approximation of the area between reference path function $\theta_{ref}(s)$ and the actual path taken by the robot in the cartesian space.

The path used to compare the two velocity controllers is shown in Figure 5(a). A zero-radius, unit step heading change of $-\pi/2$ is followed by one and a half cycles of a sinusoidal path that incorporates a range of changing path curvatures. The end of each plot in Figure 4(a) indicates the completion of this test path. The solid red plot in the middle of Figures 4(a) and 4(b) presents the same configuration of the PLBV regulated system with $\bar{\omega} = 0.25$. Figure 4(a) compares this result to other values of $\bar{\omega}$ and Figure 4(b) compares its performance to the commonly used constant velocity controller. As expected, different values for $\bar{\omega}$ in Figure 4(a) produce different integrated errors and overall traversal times as well. As $\bar{\omega}$ is increased, higher velocities are permitted for curved paths at the expense of increased phase lag in the heading controller, and consequently, more cumulative error occurs as traversal times decrease. Figure 4(b), shows that the performance of the PLBV controller is superior to the constant velocity controller. In order for a constant velocity controller to traverse the path in the same time as the PLBV controller ($\bar{\omega} = 0.25$), it must accept $\approx 0.2\ m^2$ more cumulative error over the trajectory. Similarly, in order for the constant velocity control to achieve the same cumulative error, it must go much slower, taking over 19 seconds longer to traverse the path.

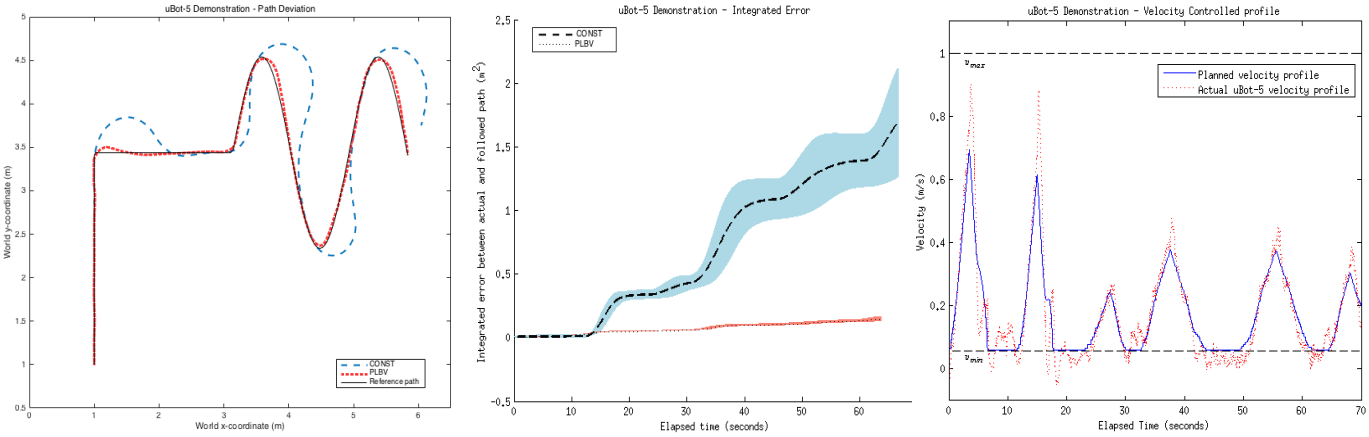
V. Results on the uBot-5 Personal Robot

We implemented the PLBV algorithm on the uBot-5 robot (shown in Figure 1). The robot is a 13 DOF balancing mobile manipulator [1]. Precisely the same version of the algorithm that was used in simulation was also used in the experiments conducted on the real robot. The wheel motors for the uBot-5 platform have no-load speeds $\omega_0 = 596.90\ rad/s$ and stall torques $\tau_s = 531\ mNm$. The natural frequency of the integrated robot was measured empirically by driving the heading controller using a sinusoidal input whose frequency is increased until the phase lag of the response is observed to be $-\pi/2$. In this operating condition, the driving frequency is approximately equal to the natural frequency ω_n. The result



(a) (b)

Fig. 4: **Simulated Gazebo Experiments:** (a) Integrated error for three different values of $\bar{\omega}$= $\{0.1$, 0.25, $0.5\}$ (b) Integrated error for 2 constant velocities where $v_{const} = 0.34$ m/s and a particular parameter for velocity control where $\bar{\omega} = 0.25$ have similar traversal time and $v_{const} = 0.24$ m/s has similar integrated error with the velocity control parameter $\bar{\omega} = 0.25$



(a) Path deviations of both controllers (b) Integrated error along path (c) Velocity-control profile example

Fig. 5: **uBot-5 Experiments:** (a) illustrates the reference path and each controller's most-deviated path (b) shows the integrated error in the five trials for each of the controllers. The shaded region around the lines represent the mean and one standard deviation. (c) illustrates a single run of the velocity controller with $\bar{\omega} = 0.25$ and associated velocity profile.

of this empirical process indicates that the natural frequency of the as-built heading controller is approximately $\omega_n = 1.05$ [rad/sec].

Controller	**Time (s)**	**Integrated Error** (m^2)
PLBV ($\bar{\omega}$ = 0.25)	63.64	0.177
CONST (v = 0.33 m/s)	66.73	2.053
CONST (v = 0.15 m/s)	267.93	0.183

TABLE I: Proposed PLBV vs. CONST on uBot

Five trials were conducted on the test path (Figure 5(a)) for the proposed PLBV algorithm and the commonly used constant velocity controller CONST. Figure 5 and Table I report the experimentally measured difference in cumulative error and traversal time between the PLBV and CONST. In order for the constant velocity controller to yield the same traversal time as the PLBV plan it must make large deviations from the path, which increases the risk of colliding with obstacles in the environment. These deviations are depicted in Figure 5(a) where we can see large deviations from the path where there is high curvature content. Note that the PLBV plan has relatively low deviation at these points. The difference of path tracking accuracy is shown in Figure 5(b) where we see that the constant controller accumulates a much larger integrated error. Additionally, due to the instability introduced by the high driving frequency, there was a large variation in tracking error over the 5 trials.

The constant velocity controller with $v = 0.15\ m/s$ is able to traverse the path with the same cumulative error as the PLBV plan with $\bar{\omega} = 0.25$. However, this resulted a traversal time over four times that of the proposed PLBV

plan. Conversely, for equivalent traversal times, the constant velocity control had to run at $v = 0.33\ m/s$ and at that speed, it incurred approximately ten times the cumulative error. In contrast, Figure 5(c) the PLBV plan executed velocities from $-0.05\ m/s$ to $+0.9\ m/s$ without compromising precision more than permitted by the $\bar{\omega} = 0.25$ specification. Figure 5(c) also illustrates the effects of balancing on the platform's ability to accelerate. At local maxima in robot velocity, it must accelerate before it is able to decelerate in order to advance the base ahead of the platform's center of mass. It is evident from Figure 5 that constant velocity control is incapable of tracking a path plan with high curvature accurately unless a very low constant velocity is used and that the PLBV velocity plan can do significantly better. This robot data on the test path agrees with the data gathered from our simulations, however, the differences in performance are much higher in the real robot experiments, which suggests that our method is robust enough to handle the additional uncertainties present in a real system.

VI. Conclusion and Future Work

The control theoretic motivations underlying the PLBV approach have proven to provide significant insight into the relationship between heading control and longitudinal velocity and inform relatively simple methods for exploiting the platform's dynamics which is also more graceful than constant velocity controller. The result is a simple and more fundamental form of dynamic guidance for velocity control than already exists in the literature.

To evaluate the approach, we chose to compare results only for implementations that pre-plan the paths and velocities. However, our laboratory implementations re-plan when errors occur and, therefore, introduce interesting issues that we wish to explore regarding the impact of various methods for velocity control on the total computational loads they represent in this context. This becomes even more important in applications in dynamic and only partially modeled environments.

Acknowledgments

This research was partially supported by NASA-GCT-NNX12AR16A. 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 National Aeronautics and Space Administration. We would like to thank the members of the LPR especially Dirk Ruiken and Tiffany Liu.

References

[1] S. R. Kuindersma, E. Hannigan, D. Ruiken, and R. A. Grupen, "Dexterous mobility with the ubot-5 mobile manipulator," *In International Conference on Advanced Robotics*, 2009.

[2] D. Ruiken, M. W. Lanighan, and R. A. Grupen, "Postural modes and control for dexterous mobile manipulation: the umass ubot concept," in *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*. IEEE, 2013, pp. 721–726.

[3] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots*, vol. 13, no. 3, pp. 207–222, 2002.

[4] Y. Wang and G. S. Chirikjian, "A new potential field method for robot path planning," vol. 2, pp. 977–982, 2000.

[5] K. O., "Real-time obstacle avoidance for manipulators and mobile robots," in *ICRA*. IEEE, March 1985, pp. 500–505.

[6] L. E. Kavraki and J.-C. Latombe, "Probabilistic roadmaps for robot path planning," 1998.

[7] E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 1, pp. 166–171, 1998.

[8] S. M. LaValle, "Rapidly-exploring random trees a new tool for path planning," 1998.

[9] S. Rodriguez, X. Tang, J.-M. Lien, and N. M. Amato, "An obstacle-based rapidly-exploring random tree," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 895–900.

[10] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[11] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 341–346.

[12] O. Amidi, "Integrated mobile robot control," Carnegie Mellon University Robotics Institute, Tech. Rep. CMU-RI-TR-90-1, 1990.

[13] R. Simmons, "The curvature-velocity method for local obstacle avoidance," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 4. IEEE, 1996, pp. 3375–3382.

[14] G. Lie, G. Ping-shu, Y. Xiao-li, and L. Bing, "Intelligent vehicle trajectory tracking based on neural networks sliding mode control," in *Informative and Cybernetics for Computational Social Systems (ICCSS), 2014 International Conference on*, Oct 2014, pp. 57–62.

[15] C. Chen, Y. He, C. Bu, J. Han, and X. Zhang, "Quartic bezier curve based trajectory generation for autonomous vehicles with curvature and velocity constraints," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 6108–6113.

[16] S. Coenen, J. Lunenburg, M. van de Molengraft, and M. Steinbuch, "A representation method based on the probability of collision for safe robot navigation in domestic environments," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sept 2014, pp. 4177–4183.

[17] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments," *Journal of Field Robotics*, vol. 25, no. 11-12, pp. 939–960, 2008.

[18] S. Srinivasa, D. Ferguson, C. Helfrich, D. Berenson, A. Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M. Weghe, "Herb: a home exploring robotic butler," *Autonomous Robots*, vol. 28, no. 1, pp. 5–20, 2010.

[19] H. G. Nguyen, J. Morrell, K. D. Mullens, A. B. Burmeister, S. Miles, N. Farrington, K. M. Thomas, and D. W. Gage, "Segway robotic mobility platform," in *Optics East*. International Society for Optics and Photonics, 2004, pp. 207–220.

[20] M. Stilman, J. Olson, and W. Gloss, "Golem krang: Dynamically stable humanoid robot for mobile manipulation," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3304–3309.

[21] J. Dietsch, "Darpa entices roboticists to take the next step [news and views]," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 3, pp. 9–10, 2012.

[22] S. Brown, "Nasas new iron man robot valkyrie," *SCIENCE*, 2013.

[23] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent asimo: System overview and integration," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3. IEEE, 2002, pp. 2478–2483.

[24] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep planning for the honda asimo humanoid," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 629–634.

[25] Y. Huang, B. Vanderborght, R. Van Ham, Q. Wang, M. Van Damme, G. Xie, and D. Lefeber, "Step length and velocity control of a dynamic bipedal walking robot with adaptable compliant joints," *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 2, pp. 598–611, April 2013.

[26] J. Craig, *Introduction to Robotics: Mechanics and Control - Second Edition*. Reading, MA: Addison Wesley, 1986.