Graceful Gross Motor Control: A Curvature-based Velocity Control on Precise Path Propagating Agents

Jay Ming Wong [1], Mitchell Hebert [2], Mordechai Rynderman [3], Tiffany Liu [4], Roderic Grupen [5]

Laboratory for Perceptual Robotics, Univeristy of Massachusetts Amherst
School of Computer Science, Amherst, MA 01003, United States of America

[1] jayming@umass.edu, [2] mrhebert@umass.edu, [3] mrynderm@umass.edu,
[4] tliu@cs.umass.edu, [5] grupen.cs.umass.edu

***Abstract*—This paper provides a novel approach and real time implementation for velocity methodologies for graceful velocity planning and path propagation. Our approach yields velocity profiles that achieve the maximum velocity along the path without violating specified path tracking precision and maximum velocity constraints. Along with generating reachable velocity profiles, we introduce a methodology to ensure controlled velocity ramp-down to the goal and curvature-based velocity limitations. In addition, we describe techniques for translational velocity and heading error control. We assume smooth paths generated via harmonic function, which distributes curvature evenly throughout the entire freespace and we complement the smooth heading references with smooth, precise, and graceful motor behavior that complies with acceleration constraints. Furthermore, we introduce and implement a path tracking policy that demonstrates distinct advantages. This paper integrates path planning, path tracking, and longitudinal velocity control for smooth and graceful motor behavior. The methods we describe here are not restricted to agents implementing the harmonic path planner but can be extended and generallized to various other global or local path planners.**

I. Introduction

This research was motivated by the importance of path following precision and the consequences of deviating from the initially planned path, resulting in potential obstacle collisions. A path is generally planned due to some appealing property it possesses, deviation from that path may be dangerous especially in human-robot environments, thus requires replanning, which has a specific associated cost. Therefore, it is cost-effective to precisely track the initially planned path during propagation such that no excessive path deviation is present as well as introducing graceful and smooth controls to limit path deviation resulting from excessive velocity commands. Traditional techniques of robotics locomotion do not stress the ideals of graceful, smooth, and efficient propagation, nor high-performance, and precision; their major purpose is the arrival at the goal and avoidance of static or dynamic obstacles. We define the technical term of grace, in the context of smooth velocity transitions from regions of high and low curvature, and speedy propagation while adhering to minimal path deviations. We say here that grace is thus a combination of speed, smoothness, and precision, a quality of rotational and longitudinal accelerations along the path, resulting in provably precise and achieving high velocities. The curvature based velocity controller is modeled after the tendency for automobiles to decelerate when approaching turns and staying within the bounds of the road; generally, vehicles tend to slow down when approaching turns of high curvature.

II. Related Work

Path planning is a subject that has received tremendous attention since the mid-1970s and thus there has been a large amount of work done in path planning, although few introduce the notion of *graceful* control [7]. One such approach, the *Dynamic Window Approach*, yields to local obstacle avoidance and produce dynamically feasible paths, however, such approaches may be trapped in local minima, and hence left to themselves, they violate an important precision requirement for *grace* [3]. Other approaches take advantage of agents traveling along arcs and assume path planning techniques to implement obstacle avoidence and velocity control using various control parameters in an objective function while working in velocity space rather than configuration space. Furthermore, agents that implement this, the curvature-velocity method, may too be trapped in local minima, provided that all solutions to the objective function are equally bad; the agent may become indecisive between projected solutions, in which case, the agent implements a rotate away method to discover more promising objective function solutions [4]. To avoid local minima in path planning, artificial potentials can be used to project energy-referenced control constraints using the gradients of the potentials to evaluate reference accelerations and compute wheel torques using the *Hamiltonian framework* in conjunction with the harmonic function path planner [1], an approach simular to the *Artificial Potential Field Method* using attractive and repulsive forces for obstacles and goals to provide collison-free control [9]. However, since such reference control use in the Hamiltonian framwork describes the kinetic energy of the agent with respect to the distance away from the goal, the agent may attempt to accelerate towards the goal without a proper ramp-down policy causing either abrupt velocity changes or discontinuous velocity profiles near the goal, and therefore, against the proposed terms of *graceful* control. Furthermore the harmonic function path planner, the solution to the Laplace's Equation, ensure that curvature is distributed evenly along the path providing an optimally smooth, collision free path which is beneficial towards our definition of *graceful* control [8]; they are continuous and are local minima free, therefore making any rotate away method unnecessary [4]. However if an agent solely depends on the harmonic function for path following purposes, any deviation from the original planned path will result in the agent assuming a new path with gradients computed from the planner, following this new path rather than attempting to return to the initially planned path. And for this reason, propagation along the initial, optimally smooth, collision free path can not be achieved by the harmonic function alone as it violates an important aspect of path following precision [2].

III. Ensuring Velocity Reachability

Due to the appealing optimally smooth and collision free path that the harmonic function planner provides, velocity planning along this path can be achieved by evaluating areas along the path at which constraints have to be met. Provided that velocities are planned along the path, all velocities should be reachable given the maximum acceleration and deceleration of the agent similar to maximum constraints proposed in [5].

The reachability of the velocity profiles can be ensured implicitly by the satisfaction of the following expression. As long as this expression is satisfied we are guaranteed that all velocities are reachable; any continuous policy function that satisfies the expression will produce reachable velocity profiles at all time intervals. The Ensured Velocity Reachability (EVR) expression is thus

$$v_{min} \leq v_i - a_{max}dt \leq v_{i+dt} \leq v_i + a_{max}dt \leq v_{max} \quad (1)$$

v_i is defined to be the velocity at time $t = i$, a_{max} is the maximum translational acceleration, and v_{i+dt} is the velocity at dt time interval after the moment $t = i$. This expression must hold for all times i during the path following in order to have reachable velocity profiles as well as being within the bounds of maximum and minimum velocity assuming that dt is the time derivative according to the frequency at which we are planning velocities. Demonstrating this explicitly, the following algorithm is produced

```
ALGORITHM EVR ($vels, max\_a, max\_v, min\_v$)
  ▷ Ensure velocities are within maximum acceleration
  for $i \leftarrow 0$, $vels\_size$
      if $vels[i] + max\_a(dt) < vels[i+1]$ then
          ▷ Assume $dt$ is time from $vels[i]$ to $vels[i+1]$
          if $vels[i] + max\_a(dt) > max\_v$ then
              $vels \leftarrow max\_v$
          else
              $vels[i+1] \leftarrow vels[i] + max\_a(dt)$
  ▷ Ensure velocities are within maximum decceleration
  for $i \leftarrow vels\_size - 1$, $0$
      if $vels[i] < min\_v$ then
          $vels[i] \leftarrow min\_v$
      if $vels[i-1] > vels[i] + max\_a(dt)$ then
          $vels[i-1] \leftarrow vels[i] + max\_a(dt)$
  return $vels$
```

Fig. 1. Ensuring Velocity Reachability (EVR) algorithm

In two passes or linear time through the input size, we can ensure that all velocities in the profile are within the physical bounds of the agent in terms of maximum velocity and acceleration. We will refer to the ensuring maximum acceleration reachability as the *forward iteration* and the maximum deceleration reachability as *backward iteration*. The complete, provably reachable, velocity profile is returned via the EVR algorithm in the array $vels$. After satisfying velocity constraints, it is important that all preliminary velocity profile assumptions are reachable given the conditions the agent is currently in and will be in in the future.

A. Proof of Correctness

This section will prove the correctness of the proposed EVR algorithm; in this case, correctness refers to the reachability of all velocities in the profile. First, let the term *unreachability* refer to the point in the velocity profile such that the velocity at i cannot be reached from velocity at $i-1$ given maximum acceleration or deceleration, in short max_a, within the defined time derivative dt. A velocity profile may have an arbitrary amount of these points; a profile with n unreachabilities will be called an *n-unreachable* velocity profile. We will use induction on the number of unreachabilies n.

Base Case, $n = 0$:

The velocity profile will be returned unchanged due to being originally correct.

Inductive Hypothesis , Given EVR(n), prove EVR(n+1):

When executing $EVR(n+1)$, we see that when we approach the index i at which the first unreachability is present in $EVR(n+1)$, one of two cases can occur:

1) The iteration eliminates the unreachability
2) The iteration eliminates the unreachability but creates another unreachability at index $i+1$ *(or* $i-1$ *during the backward iteration)*

If 1) occurs, eliminating the unreachability results in $EVR(n)$ which was the premise. If 2) occurs, we continue to iterate for the next unreachability at index $i+1$ (or $i-1$) and so on; eventually, a unreachability is eliminated and a new unreachability is not introduced because the next index has a velocity that is reachable or we have exhausted iterating through the entirety of the profile. Either way, this results in $EVR(n)$ which was the premise.

IV. Velocity Constraints

We introduce velocity constraints to ensure that the agent yield ideal behaviors in terms of curvature-base velocity restrictions, ramping down velocity in proximity to the goal, and minimization of path deviation due to excessive velocity commands and heading error.

A. Curvature-bound Constraints

Modeled after vehicles decelerating at regions of curvature, a curvature-bound constraint is introduced for the agent during path progation. The policy for curvature-bound constraints is derived from evaluating curvature estimated using the gradients of the aritifical potentials; the change in the gradient g_{i-1} and g_i, corresponding to the previous and current gradients along the initially planned path, are computed resulting in Δg. These gradients are represented as an angle relative to the positive x axis in polar coordinates. The magnitude of Δg is then evaluated and velocity limitation is determined through the velocity reduction factor δ_{vr} where

$$\delta_{vr} = K_{vr}(\Delta g)a_{max}dt \quad (2)$$

We assume a linear system in which larger curvature results in a greater difference in the gradients g_{i-1} and g_i, resulting in a Δg with larger magnitude, which corresponds to a larger velocity reduction. We define K_{vr} as the multiplicative delta parameter, a constant used to proportionalize Δg, dt is defined as the time derivative. In our current implementation $K_{vr} = 500$.

B. Velocity Ramp-down Policy

Due to being dangerous for the agent to propagate at maximum velocity into the goal region, an form of velocity ramp down must be introduced dependent on current position to goal proximity. In order to ensure that the agent ramps down its velocity when in proximity to the goal, we can use simple equations of motion [6] to determine the point at which the velocity ramp-down should occur. By the following equation, which is generally used to calculate vehicle skid distances or pre-breaking velocities, we assume that $v_f = 0$ since the agent is to come to a complete stop at the goal, and hence derive that ${v_f}^2 = {v_i}^2 + 2a\Delta x$ results in $v_i = \sqrt{|2a\Delta x|}$. Using v_i as the velocity required for proper ramp-down at Δx distance away from the goal and introducing a ramp-down control parameter K_{rd} that controls the distance at which ramp-down should occur. The K_{rd} used in our current implementation is 0.85. We specify a_{max} here since we are using the maximum acceleration in our evaluations. Assuming v_{cm} is the current commanded velocity after all other constraints, we only ramp down if v_{cm} exceeds v_{rd}, where

$$v_{rd} = \sqrt{|2K_{rd}(a_{max})\Delta x|} \tag{3}$$

C. Other Constraints

We introduce other constraints deriving from the idea of exceeding heading error; the agent's heading error must be within a defined deadband to be able to accelerate at maximum acceleration, otherwise, the agent accelerates at a fraction of the maximum acceleration. Cautious acceleration limits the potential growing path deviation by allowing the heading controller to keep up with the translational controls. Furthermore, exceeding heading error may result in deceleration if the error is too large. These additional constraints help regulate exceeding heading error and potential path deviation as a result of heading error and excessive velocity.

V. Precision in Propagation

Due to the tendency for new path selection when deviating from the initial path using solely the harmonic function path planner for path propagation, a precise propagation policy must be implemented to minimize path deviation errors and introduce a desire to return to the initial path. The policy is described using the following diagram Given the initial path λ and a straight-line distance deviation of θ at the current, actual position pos_{act}, and the closest point along the path pos_{clo}, the gradient at the point pos_{clo} can be computed resulting in g_{clo}. A parallel gradient at the actual position pos_{act} can be drawn relative to g_{clo} resulting in g_{par}.

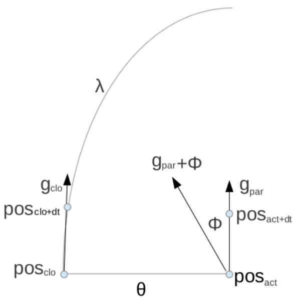


Fig. 2. Path tracking procedure using parallel gradients and offset controls.

However, if the agent is to solely follow the parallel gradients g_{par}, the agent is incapable of returning to the path λ, rather the agent will simply follow the path always either maintaining constant deviation or increasing deviation. Therefore, the introduction of the offset ϕ allows for the agent to progress towards the initial path λ. The policy for evaluating this offset is dependent on the distance of deviation θ. The system can be described as $k_p\theta + k_d\dot{\theta} = \phi$. The result is an offset controller, a PD-controller in which θ corresponds to the deviation distance and $\dot{\theta}$ corresponds to the change in the deviation distance. The gains used in our current implementation are $k_p = 4.15$ and $k_d = 0.88$. This approach is simular to the *pure pursuit* approach, however, we found that an undamped system may exhibit oscillating deviation behaviors when given a higher proportional gain the basis of this tracking strategy is derived from the incremental goal points of the Control Theory tracker presented in [10].

VI. Path Deviation Interpretation

In order to evaluate path following precision, the path deviation along with integrated error is evaluated to formally and numerically express the advantages of the proposed methodologies. The integrated error between the initial path $f(\phi_o)$ and the path the agent physically followed $f(\phi_f)$ can be evaluated to described the path deviation for that particular trajectory. The integrated error can be expressed as the area between the two curves represented as $f(\phi_o)$ and $f(\phi_f)$ where ϕ is defined as any global path; in our implementation we assumed a harmonic function path planner, therefore defining $f(\cdot)$ as an artificial potential function that the agent follows, hence expressing the initial potential function that was expected to follow as $f(\phi_o)$ and the potential function streamline that the agent is at any time interval afterwards following as $f(\phi_f)$. Failure in precision in path tracking will result in a $f(\phi_f)$ that is far away from $f(\phi_o)$, whereas using precise tracking mechanisms will result in all $f(\phi_f)$ along the path to be close or identical to $f(\phi_o)$. We can formally express this as

$$\int |f(\phi_o) - f(\phi_f)| dt \tag{4}$$

At any time $t = n$ during the propagation along the trajectory, the agent can compute the integrated error by integrating its current deviation since the last time interval and summing with the historical value e_{n-1} resulting in $e_n = e_{n-1} + |f(\phi_o) - f(\phi_f)|\,\Delta t$, in which we can express as the sum

$$e_n = \sum_{i=0}^{n} |f(\phi_o) - f(\phi_f)|\,\Delta t \tag{5}$$

VII. Experimentation

A. Experimental Design

The experiments will contrast and evaluate the potential advantages by exploring propagation time, generated velocity profiles, path following precision, smooth and graceful transitions as well as ramp-down. The experimentations will contrast propagation along gradients generated by the harmonic function with the introduced path tracking procedures as well as evaluating the proposed velocity control policy with the current implementation. Smoothness and grace are very subjective, but we hope to define them distinctively by minimal path deviation and adherence to the path in propagation.

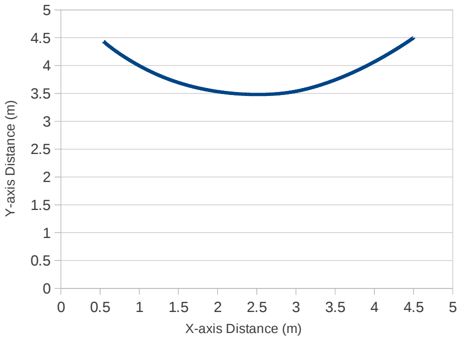
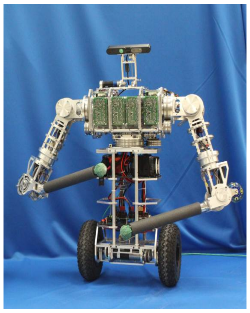


Fig. 3. Initial slightly curved path used in the experimental design

Figure 3 demonstrates the trajectory of the initial path used for our experimentations; odometry values, denoted as the triple $<x,\ y,\ \theta>$ (the $<x,\ y>$ pair denote the agent's coordinates in Cartesian space and θ refers to the agent's heading relative to the positive x-axis), are initially set at $<4.5,\ 4.5,\ 5\pi/4>$ and the agent's goal is thus $<0.5,\ 4.5,\ 3\pi/4>$.

B. Experimental Platform

The experiments were conducted on uBot6, a small, toddler-sized, twelve degree of freedom, dynamically balancing humanoid robotics platform for mobile manipulation at the University of Massachusetts Amherst. The platform, which implements Robotics Operating Systems (ROS) software, has four degrees of freedom in each arm, a twisting torso, a tilting head, and two wheels. uBot6 is capable of achieving a maximum translational velocity of $1.2m/s$, although most of our experiments will be reaching a fraction of the maximum velocity, generally around $0.6m/s$.

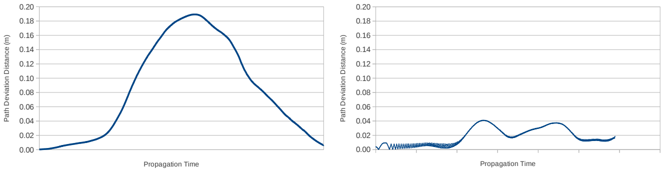


VIII. Experimental Results

A. Comparative Precision During Path Propagation

This section demonstrates the advantages between using the presented path tracking procedure and propagation generalized by following the harmonic function generated potentials. The experiements comparing the harmonic function and path tracking procedure were executed under the same gains and conditions, using the only difference of path tracking technique. The gains that were used were $v_{max} = 0.4$ and $a_{max} = 0.08$.

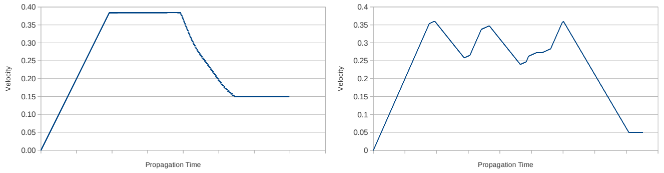
The trajectory of the path generated by implementing the path tracking precision procedure is labeled λ_{prs} whereas the trajectory of the path generated via harmonic function propagation is denoted as λ_{hrm}. The path deviation graphs shown are a measure of the deviation (m) with time (s). The velocity profiles demonstrated shows the velocity (m/s) as time (s) progresses.



(a) Path Deviation of λ_{hrm}

(b) Path Deviation of λ_{prs}

Fig. 4. There is a significant difference in path deviations between following gradients along the harmonic function and intentionally selecting the initial planned path and recomputing deviations derived from path deviation distance.



(a) Velocity Profile of λ_{hrm}

(b) Velocity Profile of λ_{prs}

Fig. 5. The velocity profiles of the two policies are seen above. The deceleration areas shown in λ_{prs} corresponds to the path deviations presented as an attempt to regulate excessive path deviation.

Due to the nature of the harmonic function path planner, there is no desire to return to the initial path, which results in much larger deviation compared to the path tracking procedure that was implemented that actually recomputes gradients to stubbornly return to the path that was originally planned. Furthermore, the agent does not exceed more than $4cm$ of deviation when following the curved path in λ_{prs}. The velocity profile yields decelerations corresponding to the areas of deviation due to the path precision procedure and the offset controls introducing additional curvature to return to the original path.

B. Significance of Velocity Control Policy

This section presents various experiments that demonstrates the significance of the velocity control policy in terms of path deviation using various velocity gains schemes. λ_{cur} refers to the current implementation in the uBot-6 platform without curvature-based velocity control and the various described methodologies and constraints, λ_{con} refers to the implementation with the controls and constraints; both implement path tracking. Three sets of experiments were conducted; drive gains are presented by the pair $<v_{max},\ a_{max}>$ denoting max

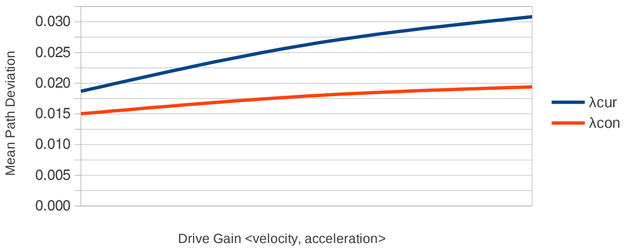


Fig. 6. Mean path deviations with increasing drive gains from experimentation

velocity and max acceleration. We ran five trials per set using the drive gains of <0.4, 0.08>, <0.6, 0.11>, and <0.8, 0.15>.

In Figure 6, λ_{con} demonstrates better scalability in terms of mean path deviation as the agent propagates with increasing drive gains. The mean path deviation for each of the drive gains that were experimented on are plotted and fit to a curve shown in Figure 6. It is computed as the ratio between the sum of the individual deviations at each time derivative dt, expressed through our proposed methodology by integration, and the total propagation time, resulting in a value that expresses the mean, most probable, path deviation at any instantaneous moment during path propagation. Therefore, it is demonstrated that using the velocity controls and constraints in λ_{con} produces less path deviation, which is a desirable trait in path propagation.

C. Further Experimentation

This section provides extensions to our system that promote better path tracking precision as well as curvature-based velocity controls. Furthermore, we test our controls on a sinusodial path which possesses numerous areas of high curvature, therefore, the agent must be able to accelerate and decelerate accordingly.

Extensions to our experiments included removing a conservative deadband in the offset controller and as a result, we found that using the gains $k_p = 1.1$ and $k_d = 0.088$ provided better performance. Additional changes included estimating curvature solely based on the initial planned plan as opposed to the trajectory that the agent will be following; we found that due to deviation, the offset controller will provide gradients for the agent to approach the initial path, this gradient may introduce curvature to the path that had not existed before. In [10], this intentionally added curvature is necessary for returning to the path, however, we do not want to introduce deviation-yielded curvature in the computation for curvature-based velocity control since this curvature does not exist in the original path; the curvature restrictions should only be dependent on the original path, not the tracking trajectories that is introduced by deviation. Using these techniques we were able to minimize the oscillating deviation behavior when the agent entered and exited the deadband as well as reduced further velocity changes due to taking into account additional curvature derived from the offset controller in places of deviation.

Although the offset controller provides a path tracking tendency for the agent, due to the curvature of the sinusoidal curve there exists deviations in which the heading controller is not able to complete the tight turn, resulting in deviation which is later corrected by the gradient offset controls. Furthermore, we define the look-ahead procedure, appreviated as (LA) as estimating the curvature and gradients along the path by looking ahead in the path by a specific distance interval rather than using the next position to compute parallel gradients. We provide two methods to compensate for such deviation due to low-gained heading controllers: 1) increase the look-ahead of the offset controllerand 2) increasing the proportional gain of the heading PD-controller.

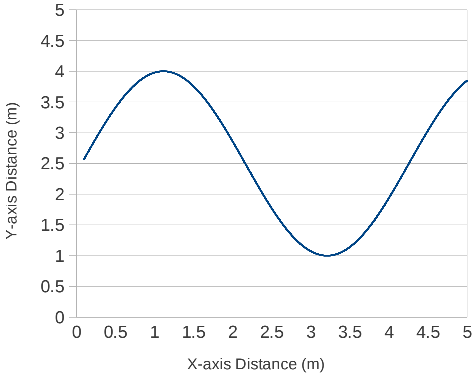


Fig. 7. The initial sinusodial path used for further experimentation.

We observe a sinusodial trajectory under several conditions; the initial path used for further evaluation of our proposed methodologies is a sinusoidal curve that has an amplitude of 1.5 and exists in the domain of $[0.1, 5.0]$ using the function $1.5sin(1.5x)$ presented in Figure 7. In Figure 8, the velocity profile for the sinusodial path is presented. Due to the fact that curvature estimates are solely based on the initially planned path, the trajectories present the same velocity profile is demonstrated. Notice that decelerations in the profile corresponds to the areas of curvature in the sinusodial function and accelerations correspond to close-to-straight areas. Additionally, the controlled velocity ramp down when approaching the goal is shown.

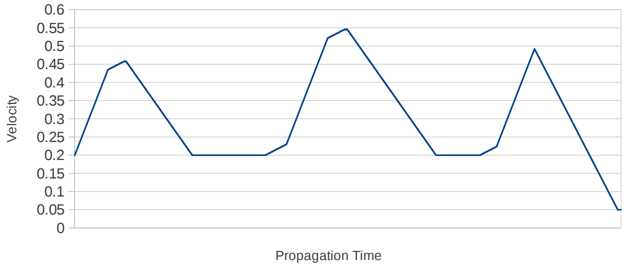
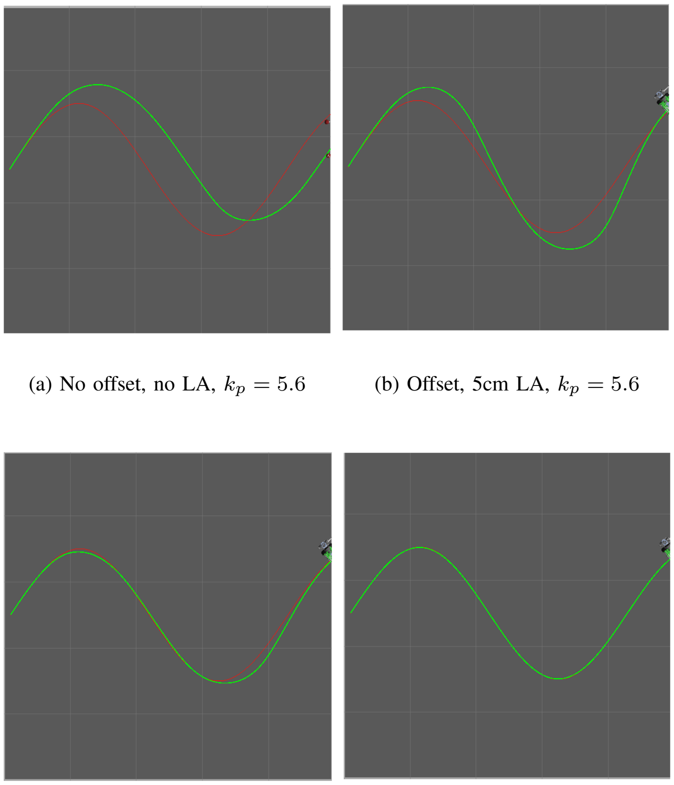


Fig. 8. Velocity profile for the propagation of the sinusodial path.

In Figure 9, the red curve represents the planned trajectory and the green curve represents the propagating agent's actual trajectory (in a) there exists tremendous deviation due to the lack of an offset control). It is shown that an increase in the look-ahead (LA) allows for a better path tracking when the agent's heading control has a low proportional gain, which we denote k_p. Of course, when we crank the k_p gain up (as shown in experiment d) in Figure 9), the agent is able to almost perfectly track the path; therefore, we can conclude that the



(a) No offset, no LA, $k_p = 5.6$

(b) Offset, 5cm LA, $k_p = 5.6$

(c) Offset, 45cm LA, $k_p = 5.6$

(d) Offset, 5cm LA, $k_p = 55.6$

Fig. 9. Experiments on different look-ahead policies and heading controller.

deviations presented in experiments b) and c) are mainly due to the heading controller gains rather than excessive velocity.

IX. Discussion and Conclusion

We presented methodologies for path tracking precision that demonstrated significantly less path deviation compared to propagation along gradients generated by the harmonic function path planner alone. We presented ideas for velocity control and limitations which demonstrated scalability with higher velocity gains, however, there exists a trade-off between path tracking precision and propagation time that is still an ongoing problem. Furthermore, improvement can be made by tuning the PD-controller gains k_p and k_d for both the heading controller and the gradient offset controller to increase performance, to allow for larger curvature turns, and to further decrease path deviations.

X. Acknowledgements

The author Jay Ming Wong was funded by the Research Experience for Undergraduates Program of the National Science Foundation under NSF award number CNS-1062749, hence, acknowledging the Principal Investigator, Rick Adrion and the Program Manager, Wendy Cooper. Jay would like to acknowledge and thank his graduate mentor Tiffany Liu and faculty advisor Roderic Grupen for their continued assistance and feedback, for their ideas and suggestions, and for their general help during the duration of this project. Furthermore, feedback and assistance from graduate students, Dirk Ruiken and Michael Lanighan were greatly appreciated and largely influenced this project. Additionally, feedback from the Laboratory for Perceptual Robotics members, post-doctoral and graduate students, Shiraj Sen, Hee-Tae Jung, and Takeshi Takahashi during lab meetings and in general helped progress this project as well.

References

[1] Connolly, C.I.; Souccar, K.; Grupen, R.A., "A Hamiltonian framework for kinodynamic planning and control," Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on , vol.3, no., pp.2746,2751 vol.3, 21-27 May 1995

[2] Christopher Connolly and Roderic Grupen. "Nonholonomic Path Planning Using Harmonic Functions. University of Massachusetts Amherst. 1994.

[3] Dieter Fox, W. Burgard, and Sebastian Thrun, "The Dynamic Window Approach to Collision Avoidance". IEEE Robotics and Automation, Vol. 4, No. 1, 1997

[4] Simmons, R., "The curvature-velocity method for local obstacle avoidance," Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on , vol.4, no., pp.3375,3382 vol.4, 22-28 Apr 1996

[5] Gil Manor, Elon Rimon: "VC-method: high-speed navigation of a uniformly braking mobile robot using position-velocity configuration space". Auton. Robots 34(4): 295-309 (2013)

[6] Serway, Raymond A. Physics for Scientists and Engineers. 3rd ed. Philadelphia: Saunder's College, 1990. Print.

[7] Nadia Adnan Shiltagh, Lana Dalawr Jalal. "Path Planning of Intelligent Mobile Robot Using Modified Genetic Algorithm". International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Vol-3, Issue-2, May 2013.

[8] C. I. Connolly , J. B. Burns , R. Weiss. "Path Planning Using Laplace's Equation." *In Proceedings of the 1990 IEEE International Conference on Robotics and Automation*. 1990.

[9] P. K. Mohanty, D. R. Parhi, "Controlling the Motion of an Autonomous Mobile Robot Using Various Techniques: a Review", Journal of Advance Mechanical Engineering, 2013, 1: 24-39.

[10] Amidi O. (1990) Integrated Mobile Robot Control. Carnegie Mellon University. Robotics Institute. Techinical Report CMU-RI-TR-90-1.