16th IEEE International Conference on Control Applications
Part of IEEE Multi-conference on Systems and Control
Singapore, 1-3 October 2007



# Alignment Distance in Path Control

Nghia Tran[†], Brandon Rohrer [*], Sean Warnick [†]

[†] Information and Decision Algorithms Laboratories
Department of Computer Science, Brigham Young University, UT
http://idealabs.byu.edu

[*] Intelligent Systems, Robotics and Cybernetics Group
Sandia National Laboratories, NM

*Abstract*— **In this paper we discuss alignment distance for measuring path deviation between curves. We compare properties of the alignment distance to both p-norms and the Hausdorff distance to argue its superiority for use in path following problems. While problems of finding an optimal parameterization of a fixed input curve to a tracking system are not new, typical formulations focus on parameterizations that minimize transversal time while respecting certain system constraints; on-line governors can then be employed that choose a path velocity in real time, trading off computational complexity and time-optimality. By explicitly characterizing the error measure implicit in path control problems, we revisit the off-line, open loop parameterization problem to explore the inherit tradeoffs between command shape, command parameterization, and system dynamics. The utility of the alignment distance as a tool for elucidating these fundamental tradeoffs is demonstrated on a simple PD-controlled mass system.**

## I. Background

Path control problems differ from tracking problems in that performance is measured with respect to the path, or image set of a curve, instead of a particular time-parameterization of the curve. Examples are common, including driving a car along a winding road. Our primary objective is to stay on the road, even when trying to navigate it quickly. This objective is manifest by our common sense reaction to slow down when conditions change, such as in the presence of rain or ice; we know that if we go slow enough we can avoid exciting the undesirable dynamics resulting from poor weather. This ability to slow down, or reparameterize the reference command in time, offers an extra degree of freedom in path control problems that sacrifices travel time for better path following.

Navigation is not the only application of path following problems, however. Production, manufacturing, and assembly processes are all often amenable to a path following formulation since the quality of the resulting product can typically be measured independent of the particular manufacturing schedule or processing time [10]. For example, a high performance vertical cavity surface emitting laser, a well-manufactured car, or a reliable laptop are quality products regardless of how long it took to produce them. Furthermore, applications in robotics and studies of human movement often consider path following objectives, and related problems are beginning to emerge in areas as diverse as web services, drug design, and organizational behavior.

The simplest approach to solving path control problems is to reduced them to tracking problems. A parameterization of the input is chosen, possibly for its simplicity, and the system is designed to track the resulting trajectory. Three approaches are common in the resulting tracking design. The first approach employs asymptotic tracking as a performance objective and appeals to regulator theory for a solution [2]. Internal models in the feedback system ensure that any desired trajectory from a specified class of trajectories is eventually tracked, even in the presence of model uncertainty. This approach yields perfect path following after initial transients die away. As extensions to nonlinear systems emerged, inversion-based methods have offered lower complexity alternatives to regulator designs by focusing on exact output tracking instead of asymptotic convergence [9]. These methods can yield great path following for a particular curve, even during transients. Nevertheless, inversion-based methods can also be very sensitive to model uncertainty, since the system dynamics are essentially inverted to discover the inputs necessary to yield the desired trajectory. The last approach uses loop shaping to determine a feedback design that tracks trajectories with frequency content in a range where tracking performance, usually measured by the energy or peak value of the error, is small [11]. This approach can offer optimal trade-offs between robustness and performance, but small errors in performance can still yield large path errors. Each of the three design approaches may yield poor path following because they are solving a different problem.

More recently, variations of the path following problem have been proposed that explicitly leverage the ability to reparameterize the input to improve performance. In robotics, researchers have typically considered the minimum time necessary to transverse a desired path under torque or force constraints [4], [3], [7]. Off-line techniques pre-compute an optimal parameterization of the command and then use tracking designs to deliver good performance. These methods are typically computationally intensive and can be sensitive

[1]This work is supported by grants from the BYU Office of Research and Creative Activities and Sandia National Labs.

[2]Please direct comments and questions to Nghia Tran at tcnghia@cs.byu.edu or Sean Warnick at sean@cs.byu.edu.

1-4244-0443-6/07/$20.00 ©2007 IEEE.

to model uncertainty. On-line methods first design a tracking controller to yield good performance for a nominal trajectory, then an outer-loop "velocity controller" or "path governor" is employed to change parameterization in real time. These closed-loop methods have the advantage of being responsive to model uncertainty, but the computational complexity of making real-time calculations can force sub-optimal performance compared to the off line approaches. In either case, however, the primary concern is typically reparametrizing the command so that it is time-optimal while satisfying input and state constraints.

Another approach to the path following problem has been to convert the existing tracking controller to a path follower [5]. This is accomplished by computing a function that maps the current system state to a command time, thereby slowing commands as the actual path deviates from the desired maneuver. This approach sheds light on how the dynamics of the tracking system can be easily inherited by a path following system. The objective of this more general framework is to slow down as needed to guarantee transverse stability and hence, ultimate convergence to the desired path.

This study revisits the off-line, open-loop parameterization problem to better understand the relationship between the shape and parameterization of a command with the dynamics of a path-following system. The next section details our motivation for revisiting this problem, and Section III introduces the alignment distance in contrast with alternative metrics. Section IV then formulates our path following parameterization problem, and Section V discusses the solution for a simple example.

## II. Motivation

Our motivation is to better understand how to characterize the relationship between the intrinsic qualities of a command, its extrinsic qualities, and performance that is imposed by the dynamics of a given path following system. The intrinsic qualities of a command include its shape and arc length, and they essentially capture the notion of *complexity* of a given task. Extrinsic qualities include the parameterization of the command and its embedding in the domain of the path following system. These properties also contribute to the inherit difficulty of a system in following the command, essentially entailing *how the command is realized* in time and presented to the dynamic system. Finally, performance is captured by the deviation of the system from the image of the command, and in the total time needed to transverse the command within a specified error. This *dual notion of performance* is important, as it highlights the fact that path following systems inherently trade-off safety with speed, or quality with throughput.

Given a path following system, we would ultimately like to characterize commands a priori by their difficulty with respect to that system. Such a characterization would be useful to path planning systems to help them discern the easiest command that accomplishes a desired mission. For example, there may be many commands that deliver comparable mission results, yet these commands may differ considerably in their difficulty. This is the case when multiple laser designs yield products with comparable characteristics (wavelength, output power, etc.), yet the manufacturability of different designs may vary significantly. Likewise, when an unmanned aerial vehicle must avoid obstacles to reach its destination, some paths may be easier than others. Moreover, knowing how different systems perceive difficulty of a command can strongly impact the coordination process between multiple systems.

A natural way to characterize the difficulty of commands would be to identify a command with the resulting performance function e.g. achievable quality for a specified transversal time. One could then adapt a number of meaningful measures to the particular application, such as the maximal quality for a fixed transversal time, or the minimal transversal time for a fixed quality, etc. Nevertheless, such notions hinge on a precise characterization of quality. In path following problems, such a characterization has essentially been avoided by formulating performance as minimal time, with the desired path taken as a constraint to be satisfied exactly. In this work, the alignment distance is developed as the natural quality metric for path following problems. To make our notions precise, we assume that the closed-loop system, characterized by the transfer function matrix $H(s)$, will asymptotically track steps, ensuring that $\lim_{s\to 0} H(s) = I$, where I is the identity. Although model uncertainty could disrupt these assumptions, is seems reasonable for many applications to suggest that there is a region of "low enough" frequency where the system is well known and designed for good path following.

## III. Alignment Distance

Consider two parameterized plane curves $\alpha(s) = \{\alpha_1(s), \alpha_2(s)\} : [\ 0\ a\ ] \to \mathbb{R}^2$ and $\beta(s) = \{\beta_1(s), \beta_2(s)\} : [\ 0\ b\ ] \to \mathbb{R}^2$, where $[\ 0\ 1\ ]$ denotes the closed unit interval, and $a$ and $b$ are fixed positive numbers. Let $\alpha$ refer to the trace or image of the curve $\alpha(s)$; that is, the *trace*, $\alpha$, is a set of points in $\mathbb{R}^2$, while the *curve*, $\alpha(s)$, is a function. A common question that arises in many applications, and path following in particular, is how to measure the distance between the traces $\alpha$ and $\beta$.

One approach is to identify the curve domains through a monotonically increasing function $z :[\ 0\ a\ ] \to [\ 0\ b\ ]$, and then consider the metric

$$d_p(\alpha, \beta) = \|\alpha(s) - \beta(z(s))\|_p$$

where $\|\cdot\|_p$ denotes the standard *p-norm* for functions defined on the interval $[\ 0\ a\ ]$. A characteristic of these metrics is that they depend on the parameterization $s$ or the map $z$; the distance between the same traces varies depending on how they are parameterized. (See Figure 1.) This property makes them undesirable for path following problems.

Another approach is to consider a set-based metric like the *Hausdorff distance*, given by

$$d_H(\alpha, \beta) \max\{\sup_{p\in\alpha} \inf_{q\in\beta} \|p - q\|, \sup_{q\in\beta} \inf_{p\in\alpha} \|p - q\|\}$$

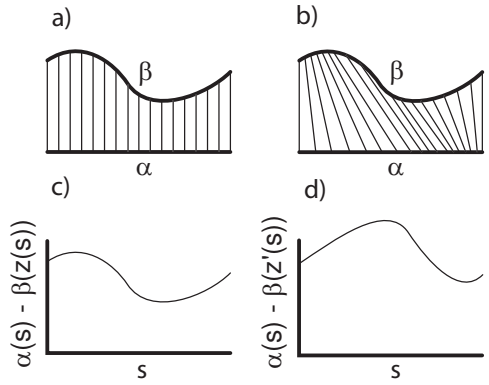


Fig. 1. An illustration of the effect of recrystallization on the p-norms. a) $\alpha$ and $\beta$ are two curves in $\mathbb{R}^2$. $z(s)$ is chosen such that $\alpha(s)$ maps to $\beta(z(s))$ as shown by the connecting vertical lines. b) This plot is similar to panel a) with the exception that a different parameterization, $z\prime(s)$ is chosen. c) The quantity $\alpha(s) - \beta(z(s))$, used in the calculation of the p-norms, is shown. d) This plot is similar to panel c) except that $z\prime(s)$ is used. Note that this recapitalization would dramatically effect the resulting p-norms.

where $\|\cdot\|$ here is the Euclidean norm. This metric operates on the traces directly, as sets in $\mathbb{R}^2$, and thus has the desirable property of being independent of parameterization. Nevertheless, unlike an arbitrary set, traces of curves are one-dimensional objects that can be ordered naturally; the failure of the Hausdorff distance to necessarily respect this orderability property makes it undesirable for path following problems since it can be a misleading measure of how far one curve deviates from another. (See Figure 2.)

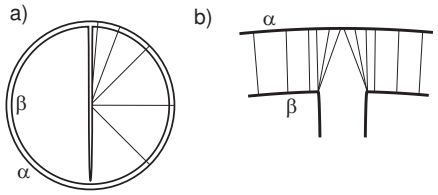


Fig. 2. An illustration of the inability of the Hausdorff distance to accurately reflect the tracking error between two paths. a) The test case is a circular path, $\alpha$, and a nearly circular path $\beta$. $\beta$ deviates from a circle near the 12 o'clock position, takes an excursion to near the 6 o'clock position, and then returns. Fine red lines show instances of $\inf_{q\in\beta}\|p-q\|$, that is, the minimum distance to any point on $\alpha$ for every point on $\beta$. b) This is a magnified portion of panel a) near the 12 o'clock position. Fine red lines show instances of $\inf_{p\in\alpha}\|p-q\|$, that is, the minimum distance to any point on $\beta$ for every point on $\alpha$. Note that neither side of the Hausdorff distance produces a value that reflects the true excursion of $\beta$ from $\alpha$ in a path tracking sense: approximately the diameter of $\alpha$.

A metric that would be useful for path following problems needs to not only be independent of parameterization, but also preserve the orderability property intrinsic to the traces of curves. It should capture a notion of how far one curve deviates from the other as they are each transversed by some parameterization. One possibility is the *alignment distance*, and given by

$$d(\alpha,\beta) = \inf_{z(t)} \sup_{t} \|\alpha(t) - \beta(z(t))\|$$

where here $\|\cdot\|$ is the Euclidean norm and $z$ is a monotonically increasing function as defined earlier. This metric describes the radius of the smallest tube centered on one curve and containing the other as it is continuously transversed from beginning to end.

The alignment distance was originally proposed by Maurice Frechet [6]. Later, it was shown that for two polygonal curves with $m$ and $n$ points respectively, the alignment distance can be computed in $O(mn\log(mn))$ time [1]. Recently, an $O(mn\log(mn))$ time algorithm has been found to compute the alignment distance between two piecewise smooth curves having $n$ well-behaved pieces [8]. This paper introduces the use of the Frechet metric to formulate the path following reparameterization problem.

## IV. Reparameterization problem

Equipped with the alignment norm, the path following reparameterization problem can be posed in alternate forms. The first form resembles the problem that has been most studied in the literature, as it emphasizes a minimum time transversal of the command with path error presented as a constraint. This problem differs from the more common formulation, however, in that the path is not expected to be followed exactly, but may exhibit an error $\gamma$ as measured by the alignment norm.

Problem 1: *Minimum-Time Fixed-Error* Given a square MIMO system, H, with n inputs and outputs, a fixed error tolerance $\gamma > 0$, and a rectifiable curve $u(s) : \mathbb{R} \to \mathbb{R}^n$ of length S and arc length parameter s, find $t(s) : \mathbb{R} \to \mathbb{R}$ such that

$$\begin{array}{ll} \inf_{t(s)} & t(S) \\ \text{subject to :} & \\ & d(u(t), Hu(t)) \le \gamma \\ & \frac{dt}{ds} > 0 \\ & \text{t(0) = 0} \end{array}$$

Let us call this problem $MTFE(\gamma)$.

This problem finds a reparameterization of the command $u$ that minimizes transversal time while keeping the alignment distance between the commanded input and the actual output less than a fixed number $\gamma$. Note that the constraint on $dt/ds$ being positive simply keeps the path parameter from reversing direction or stopping along the curve. A variation of the problem is to minimize the path error while keeping the transversal time fixed. This problem becomes:

Problem 2: *Minimum-Error Fixed-Time* Given a square MIMO system, H, with n inputs and outputs, a fixed time $T > 0$, and a rectifiable curve $u(s) : \mathbb{R} \to \mathbb{R}^n$ of length S and arc length parameter s, find $t(s) : \mathbb{R} \to \mathbb{R}$ such that

$$\begin{array}{ll} \inf_{t(s)} & d(u(t), Hu(t)) \\ \text{subject to :} & \\ & t(S) \le T \\ & \frac{dt}{ds} > 0 \\ & \text{t(0) = 0} \end{array}$$

Let us call this problem $MEFT(T)$. Ideally, one would like the parameterization $t$ that simultaneously minimizes both of these problems. One can realize this parameterization by iteratively solving the problems, where the optimal time

given an error from Problem 1 is substituted in the constraint in Problem 2 to yield a new optimal error for Problem 1, etc. This parameterization is important because it reveals the desired tradeoff between shape and performance for a given path following system and suggests that the problems are in some senses, equivalent.

Finally, note that these problems consider the infimization of two distinct parameterizations, $t$ and $z$ ($z$ is understood in the definition of the alignment distance). Since the concatenation of two proper reparameterizations is itself just another reparameterization, one may wonder whether formulations considering both are necessary. It turns out that, in fact, they are both critical to the problem. The function $t$ reparameterizes the input command $u$, and hence changes the shape of the output curve $Hu$. The function $z$, on the other hand, does not affect the shape of either curve, but is used to reparameterize one of the curves after they have been generated in order to capture the alignment distance between them.

## V. Second Order Example

To illustrate how the path following reparameterization problem is affected by use of the alignment norm, we select a dynamic system and construct an example. The system is given by:

$$\begin{bmatrix} \dot{v_x} \\ \dot{x} \\ \dot{v_y} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -\frac{b}{m} & -\frac{k}{m} & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -\frac{b}{m} & -\frac{k}{m} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_x \\ x \\ v_y \\ y \end{bmatrix} + \begin{bmatrix} \frac{k}{m} & 0 \\ 0 & 0 \\ 0 & \frac{k}{m} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \quad (1)$$

This is a second-order system, a representation of an isotropic mass-spring-damper system constrained to move in a plane. $k$ is the spring constant, $b$ is the damping coefficient, and $m$ is the mass of the system. This system differs slightly from a PD-controlled mass in that the damper is to ground, rather than to the commanded position.

Now, consider a curve $\alpha(s)$, where $s$ is the arc length parameterization. For the second-order system we selected, we can determine the input necessary to generate $\alpha(s(t))$ for any proper reparameterization (i.e. $\frac{ds}{dt} > 0$) of $s(t)$ by substitution: For $\alpha(s(t)) = \{\alpha_x(s(t)), \alpha_y(s(t))\}$,

$$x(t) \leftarrow \alpha_x(s(t)) \quad (2)$$

$$y(t) \leftarrow \alpha_y(s(t)) \quad (3)$$

$$v_x(t) \leftarrow \frac{\partial \alpha_x}{\partial s}\frac{\partial s}{\partial t} \quad (4)$$

$$v_y(t) \leftarrow \frac{\partial \alpha_y}{\partial s}\frac{\partial s}{\partial t} \quad (5)$$

$$\dot{v}_x(t) \leftarrow \frac{\partial \alpha_x}{\partial s}\frac{\partial^2 s}{\partial t^2} + \frac{\partial^2 \alpha_x}{\partial s^2}\left(\frac{\partial s}{\partial t}\right)^2 \quad (6)$$

$$\dot{v}_y(t) \leftarrow \frac{\partial \alpha_y}{\partial s}\frac{\partial^2 s}{\partial t^2} + \frac{\partial^2 \alpha_y}{\partial s^2}\left(\frac{\partial s}{\partial t}\right)^2 \quad (7)$$

This yields the following second-order differential equations of motion:

$$\frac{\partial \alpha_x}{\partial s}\frac{\partial^2 s}{\partial t^2} + \frac{\partial^2 \alpha_x}{\partial s^2}\left(\frac{\partial s}{\partial t}\right)^2 + \frac{b}{m}\frac{\partial \alpha_x}{\partial s}\frac{\partial s}{\partial t} + \frac{k}{m}\alpha_x(s(t)) = \frac{k}{m}u_x \quad (8)$$

$$\frac{\partial \alpha_y}{\partial s}\frac{\partial^2 s}{\partial t^2} + \frac{\partial^2 \alpha_y}{\partial s^2}\left(\frac{\partial s}{\partial t}\right)^2 + \frac{b}{m}\frac{\partial \alpha_y}{\partial s}\frac{\partial s}{\partial t} + \frac{k}{m}\alpha_y(s(t)) = \frac{k}{m}u_y \quad (9)$$

Solving for for the $x$ and $y$ components of $u(s(t))$:

$$u_x(s(t)) = \alpha_x(s(t)) + \left[\frac{m}{k}\frac{\partial \alpha_x}{\partial s}\frac{\partial^2 s}{\partial t^2} + \frac{m}{k}\frac{\partial^2 \alpha_x}{\partial s^2}\left(\frac{\partial s}{\partial t}\right)^2 + \frac{b}{k}\frac{\partial \alpha_x}{\partial s}\frac{\partial s}{\partial t}\right] \quad (10)$$

$$u_y(s(t)) = \alpha_y(s(t)) + \left[\frac{m}{k}\frac{\partial \alpha_y}{\partial s}\frac{\partial^2 s}{\partial t^2} + \frac{m}{k}\frac{\partial^2 \alpha_y}{\partial s^2}\left(\frac{\partial s}{\partial t}\right)^2 + \frac{b}{k}\frac{\partial \alpha_y}{\partial s}\frac{\partial s}{\partial t}\right] \quad (11)$$

Fixing $s(t)$ for the moment, consider the $\infty$-norm of the error:

$$\sup_t \quad \|u(s(t)) - \alpha(s(t))\|_2^2 = \quad (12)$$

$$\sup_t \quad \left\| \begin{bmatrix} u_x(s(t)) \\ u_y(s(t)) \end{bmatrix} - \begin{bmatrix} \alpha_x(s(t)) \\ \alpha_y(s(t)) \end{bmatrix} \right\|_2^2 = \quad (13)$$

$$\sup_t \quad \left( \left[\frac{m}{k}\frac{\partial \alpha_x}{\partial s}\frac{\partial^2 s}{\partial t^2} + \frac{m}{k}\frac{\partial^2 \alpha_x}{\partial s^2}\left(\frac{\partial s}{\partial t}\right)^2 + \frac{b}{k}\frac{\partial \alpha_x}{\partial s}\frac{\partial s}{\partial t}\right]^2 + \left[\frac{m}{k}\frac{\partial \alpha_y}{\partial s}\frac{\partial^2 s}{\partial t^2} + \frac{m}{k}\frac{\partial^2 \alpha_y}{\partial s^2}\left(\frac{\partial s}{\partial t}\right)^2 + \frac{b}{k}\frac{\partial \alpha_y}{\partial s}\frac{\partial s}{\partial t}\right]^2 \right) \quad (14)$$

Now consider the "alignment norm", that is, allow for reparameterization of $\alpha$:

$$\begin{aligned} \sup_t \inf_{z(t)} \|u(s(t)) - \alpha(z(t))\|_2^2 = \\ \sup_t \inf_{z(t)} \left([\alpha_x(s(t)) - \alpha_x(z(t))] + \right. \\ \left[\frac{m}{k}\frac{\partial \alpha_x}{\partial s}\frac{\partial^2 s}{\partial t^2} + \frac{m}{k}\frac{\partial^2 \alpha_x}{\partial s^2}\left(\frac{\partial s}{\partial t}\right)^2 + \right. \\ \left.\left. \frac{b}{k}\frac{\partial \alpha_x}{\partial s}\frac{\partial s}{\partial t}\right]\right)^2 \\ + \left([\alpha_y(s(t)) - \alpha_y(z(t))] + \right. \\ \left[\frac{m}{k}\frac{\partial \alpha_y}{\partial s}\frac{\partial^2 s}{\partial t^2} + \frac{m}{k}\frac{\partial^2 \alpha_y}{\partial s^2}\left(\frac{\partial s}{\partial t}\right)^2 \right. \\ \left.\left. + \frac{b}{k}\frac{\partial \alpha_y}{\partial s}\frac{\partial s}{\partial t}\right]\right)^2 \end{aligned} \quad (15)$$

Expanding $\|u(s(t)) - \alpha(z(t))\|_2^2$ and regrouping terms, several observations allow simplification. In particular, note that since $s$ is the arc length parameter, we have

$$\left(\frac{\partial \alpha_x}{\partial s}\right)^2 + \left(\frac{\partial \alpha_y}{\partial s}\right)^2 = 1 \tag{16}$$

$$\left(\frac{\partial^2 \alpha_x}{\partial s^2}\right)^2 + \left(\frac{\partial^2 \alpha_y}{\partial s^2}\right)^2 = \kappa^2 \tag{17}$$

where $\kappa$ is the curvature of the path, and

$$\left(\frac{\partial^2 \alpha_x}{\partial s^2}\frac{\partial \alpha_x}{\partial s}\right) + \left(\frac{\partial^2 \alpha_y}{\partial s^2}\frac{\partial \alpha_x}{\partial s}\right)$$
$$= \begin{bmatrix} \frac{\partial^2 \alpha_x}{\partial s^2} & \frac{\partial^2 \alpha_y}{\partial s^2} \end{bmatrix} \begin{bmatrix} \frac{\partial \alpha_x}{\partial s} \\ \frac{\partial \alpha_y}{\partial s} \end{bmatrix} \tag{18}$$
$$= \kappa \hat{n} \cdot \hat{t} \tag{19}$$

where $\hat{n}$ and $\hat{t}$ are the unit normal and tangent vectors, respectively. Since they are orthogonal, $\hat{n} \cdot \hat{t} = 0$, we find

$$\left(\frac{\partial^2 \alpha_x}{\partial s^2}\frac{\partial \alpha_x}{\partial s}\right) + \left(\frac{\partial^2 \alpha_y}{\partial s^2}\frac{\partial \alpha_x}{\partial s}\right) = 0 \tag{20}$$

As a result, after some manipulations the argument of Equation 15 simplifies to three main terms:

$$\|u(s(t)) - \alpha(z(t))\|_2^2 = \mathcal{A} + \mathcal{B} + \mathcal{C} \tag{21}$$

where

$$\mathcal{A} = \left[\kappa^2 \left[\frac{m}{k}\left(\frac{\partial s}{\partial t}\right)^2\right]^2 + \left[\frac{m}{k}\frac{\partial^2 s}{\partial t^2} + \frac{b}{k}\frac{\partial s}{\partial t}\right]^2\right] \tag{22}$$

$$\mathcal{B} = [(\alpha_x(s(t)) - \alpha_x(z(t)))^2 + (\alpha_y(s(t)) - \alpha_y(z(t)))^2] \tag{23}$$

$$\begin{aligned}\mathcal{C} = {} & 2[\alpha_x(s(t)) - \alpha_x(z(t))] * \\ & \left[\left(\frac{m}{k}\frac{\partial \alpha_x}{\partial s}\frac{\partial^2 s}{\partial t^2}\right) + \left(\frac{m}{k}\frac{\partial^2 \alpha_x}{\partial s^2}\left(\frac{\partial s}{\partial t}\right)^2\right) + \left(\frac{b}{k}\frac{\partial \alpha_x}{\partial s}\frac{\partial s}{\partial t}\right)\right] + \\ & 2[\alpha_y(s(t)) - \alpha_y(z(t))] * \\ & \left[\left(\frac{m}{k}\frac{\partial \alpha_y}{\partial s}\frac{\partial^2 s}{\partial t^2}\right) + \left(\frac{m}{k}\frac{\partial^2 \alpha_y}{\partial s^2}\left(\frac{\partial s}{\partial t}\right)^2\right) + \left(\frac{b}{k}\frac{\partial \alpha_y}{\partial s}\frac{\partial s}{\partial t}\right)\right]\end{aligned} \tag{24}$$

It is interesting to note that in the case of $z(s(t)) = s(t)$, terms $\mathcal{B}$ and $\mathcal{C}$ both become zero, and $\mathcal{A} = (u(t) - \alpha(t))^2$. This is simply the distance between $u(s(t))$ and $\alpha(s(t))$ without considering alignment. Given our system, this can be verified by looking at the forces that act on the mass and solving for $F = ma$. Tangential and radial directions can be considered decoupled due to the isotropic nature of our system. First in the tangential direction:

$$\begin{aligned} F_t &= k(u_t(s(t)) - \alpha_t(s(t))) - b\left(\frac{\partial s}{\partial t}\right) \\ &= m\frac{\partial^2 s}{\partial t^2} \end{aligned} \tag{25}$$

$$\Rightarrow \quad u_t(s(t)) - \alpha_t(s(t)) = \frac{m}{k}\frac{\partial^2 s}{\partial t^2} + \frac{b}{k}\frac{\partial s}{\partial t} \tag{26}$$

where $k$ is the spring constant, $b$ is the damping coefficient,and $m$ is the mass of the system. Both the spring stiffness and the damping exert forces in the tangential direction, resulting in a given tangential acceleration. Then in the radial direction:

$$\begin{aligned} Fr &= k(u_r(s(t)) - \alpha_r(s(t))) \\ &= \kappa\left(\frac{\partial s}{\partial t}\right) \end{aligned} \tag{27}$$

$$u_r(s(t)) - \alpha_r(s(t)) = \kappa\frac{m}{k}\left(\frac{\partial s}{\partial t}\right)^2 \tag{28}$$

where $\kappa(s)$ is the curvature at a given point a distance $s$ along the path. It is then straightforward to show that:

$$\begin{aligned} & (u(s(t)) - \alpha(s(t)))^2 \\ & \quad = [u_t(s(t)) - \alpha_t(s(t))]^2 + \\ & \quad [u_r(s(t)) - \alpha_r(s(t))]^2 \\ & \quad = \mathcal{A} \end{aligned} \tag{29}$$

Let the quantity $u_t(s(t)) - \alpha_t(s(t))$, the distance between the commanded point and the actual point at a given time, $t$, be called the "tracking error". The tracking error has a clear physical interpretation, but, as was previously discussed, it does not necessarily reflect the quality of the path following performance. The following example will illustrate this.

Figures 3 and 4 provide a simple example highlighting the differences between the tracking error (generated when employing p-norms as a metric for path error), the "nearest-point distance" (generated when employing the Hausdorff distance as a metric for path error), and the alignment distance. In particular, we solve Problem 2 for an elliptical input using each of these error measures in the objective function. We then command each of the resulting reparameterized inputs and observe the system response.

The choice of distance measure strongly affects which parameterizations are judged to be good performers and which are not. Attempting to minimize one distance metric can result in very poor performance in others. As shown in Figure 4 reparameterizing the input to minimize a p-norm results in poor path following performance. The nearest-point distance results in far better tracking performance, yet it fails to capture the most critical piece of information: the worst case deviation from the commanded path. In this simple example, the alignment distance occurs in nearly the same location as the maximum of the nearest-point distance,

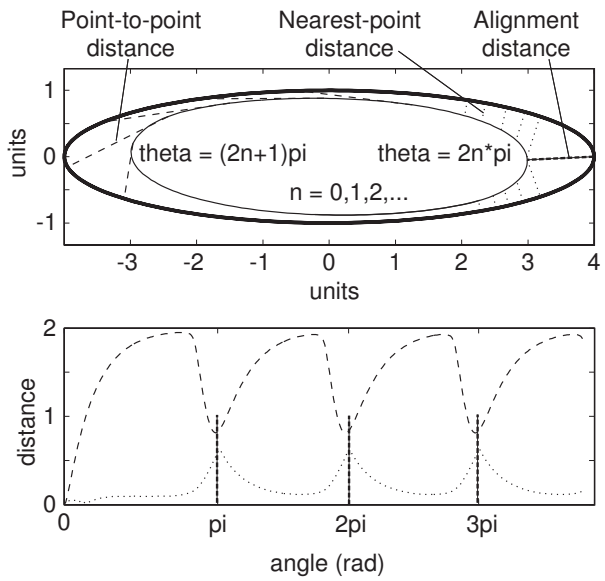


Fig. 3. a) The bold ellipse is a path commanded to a second-order tracking system. The actual path followed by the tracking system is shown by the smaller, solid ellipse. $\theta = 0$ at $x = 4, y = 0$. Peaks in curvature of the commanded ellipse occur at $n\pi$. In this case, the bold ellipse was commanded with the path length parameterization, that is, with constant velocity along the path. Three different distances between the ellipses are represented. The first two, point-to-point distance and nearest-point distance, are defined for every point on the actual path. The $\infty$-norm and the one-sided Hausdorff distance are the maximum of these two distances, respectively. The third measure, alignment distance, is only defined at the one point near each pointed end of the ellipses as shown by the bold dashed line (see the Alignment Distance section for a full definition). If $u(t)$ is the position commanded at time $t$ and $\alpha(t)$ is the output of the system at the time commanded, then $\|u(t) - \alpha(t)\|$ is the point-to-point distance at time $t$. Several instances of $u(t) - \alpha(t)$ are shown by light dashed lines in the figure. For every point on the actual curve $\alpha$, the distance to the nearest point on the commanded curve is the nearest-point distance. Several points on $\alpha$ are shown connected to their corresponding nearest points on $u$ with dotted lines. Note that the nearest-point distance does not capture the alignment distance. There is no point on $\alpha$ which has the far point on $u$ as its nearest point. b) The three distances are shown as a function of $\theta$. They produce markedly differing patterns. The point-to-point distance (light dashed line) is at a maximum on the less-curved portion of the ellipse and at a minimum near the sharply-curved ends of the ellipse. The nearest point distance (dotted line) displays exactly the opposite pattern: it is at a maximum at the ellipse's "points" and at a minimum on the ellipse's "straightaways". Similarly, the alignment distance (bold dashed line), which is a supremum, occurs at the ellipse points as well. If reducing speed is a means of increasing path following performance and decreasing distance between the commanded and actual path, then these different distance measures suggest different speed profiles around the path, that is, reparameterizations. The point-to-point distance suggests slowing down on the straightaways and speeding up through the hairpin turns. The nearest-point distance and the alignment distance suggest the opposite.

however, it is specifically designed to capture the worst case deviation from the commanded path and thus gives the most informative measure of tracking performance.

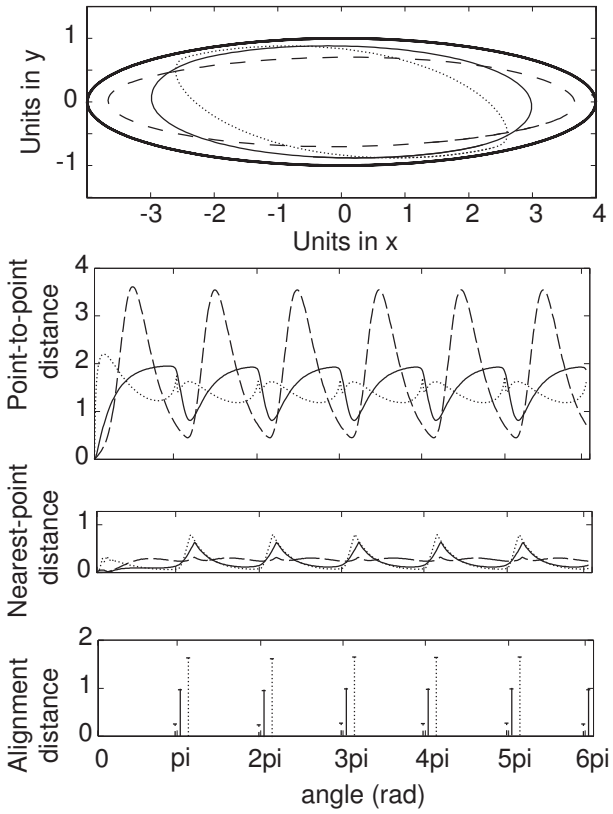


Fig. 4. a) Three different parameterizations of the commanded path (bold ellipse) produced three distinct ellipses. The path length parameterization (solid ellipse) was used as well as a parameterization designed to decrease point-to-point distance (dotted line) and nearest-point distance (dashed ellipse). Note that because the alignment distance occurs near the maximum of the nearest point distance it also suggests the parameterization that produced the dashed ellipse. Qualitatively, it is evident that the dashed ellipse most nearly tracks the commanded ellipse. b) The point-to-point distance for all three parameterizations. The maximum distance from the dotted ellipse to the commanded path (dotted line) is less than the maximum distance from the solid ellipse (solid line) to the commanded path. The dotted ellipse was successful at decreasing the point-to-point distance. On the other hand, the maximum distance from the dashed ellipse (dashed line) to the commanded path is less than the maximum distance from the solid ellipse to the commanded path. The dashed ellipse dramatically increased point-to-point error. c) The nearest-point distance for all three parameterizations. In contrast to panel b), here the dashed ellipse results in the lowest distance, and the dotted ellipse results in the greatest. d) As with the nearest-point distance, the lowest alignment distance occurs with the dashed ellipse and the greatest alignment distance occurs with the dotted line.

## REFERENCES


[1] H. Alt and M. Godau. Computing the frechet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5, 1995.

[2] O.A. Sebakhy B.A. Francis and W.M. Wonham. Synthesis of multivariable regulators; the internal model principle. *J. Appl. Math. Opt.*, 1, 1974.

[3] A. Bemporad, T.J. Tarn, and N. Xi. Predictive path parameterization for constrained robot control. *IEEE Trans. Control Systems Technology*, 7, 1999.

[4] Ola Dahl. *Path constrained robot control*. PhD thesis, Lund Institute of Technology, 1992.

[5] R. Hindman and J. Hauser. Maneuvering modified trajectory tracking. In *Int. Symp. Mathematical Theory Networks and Systems*, June 1996.

[6] Frechet M. Sur quelqes points du calcul fontionel. *Rendiconti del cicolo Mathematico di Palermo*, 22, 1906.

[7] G. Pardo-Castellote and R.H. Cannon Jr. Proximate time-optimal algorithm for on-line path parameterization and modification. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, 1996.

[8] G. Rote. Computing the frechet distance between piecewise smooth curves. *Computational Geometry, Theory and Applications*, May 2005.

[9] D. Chen S. Devasia and B. Paden. Nonlinear inversion-based output tracking. *IEEE Trans Automatic Control*, 41, 1996.

[10] S.C. Warnick and M.A. Dahleh. Feedback synchronization: A relaxation of tracking. In *the Proceedings of American Controls Conference*, 2000.

[11] W.S.Levine. *The Control Handbook*. CRC Press, 1996.