Minimum-Time Planar Paths with up to Two Constant Acceleration Inputs and L_2 Velocity and Acceleration Constraints

Victor M. Baez[1], Haoran Zhao[1], Nihal Abdurahiman[2], Nikhil V. Navkar[2], Aaron T. Becker[1]

Abstract— **Given starting and ending positions and velocities, L_2 bounds on the acceleration and velocity, and the restriction to no more than two constant control inputs, this paper provides routines to compute the minimal-time path. Closed form solutions are provided for reaching a position in minimum time with and without a velocity bound, and for stopping at the goal position. A numeric solver is used to reach a goal position and velocity with no more than two constant control inputs. If a cruising phase at the terminal velocity is needed, this requires solving a non-linear equation with a single parameter. Code is provided on GitHub[1], extended paper version at [1].**

I. INTRODUCTION AND RELATED WORK

This paper seeks the minimum-time path for a particle with a restricted set of control inputs: the system can apply no more than two constant thrust inputs, each for a disjoint time. Moreover, this thrust is limited and there is a constraint on the maximum velocity. A representative solution is shown in Fig. 1. We were motivated by a desire for simple optimal control parameterizations of hardware systems with constraints on total maximum acceleration and maximum velocity. Many of these problems are currently approximately solved using iterative numeric solvers. However, when formulated using L_2 bounds, this paper shows there are several problems that provide closed-form solutions, or can be formed as a minimization problem of a single variable. The resulting L_2 problem is interesting mathematically, and the graphical techniques described in this paper enable an intuitive understanding of the solution. This problem could apply to a class of thrusters on a space vehicle such as an astronaut take-me-home system [2], or to other low-friction environments, such as a hovercraft with a single thruster.

While actuator constraints are often expressed using L_∞ norms, payloads often specify acceleration limits in an L_2 sense, such as the 3-G limit on a space shuttle during launch [3], or an acceleration bound in every direction for translating a cup full of water [4]. Similarly, speed limits on highways refer to an L_2 speed and not an L_∞ speed. A given L_2 acceleration constraint a_m generates a corresponding L_∞ constraint $a_m/\sqrt{2}$. This conservative bound can reduce the top acceleration and top speed by almost 30%.

[1] Department of Electrical Engineering, University of Houston, USA, vmontanobaez@gmail.com, atbecker@uh.edu

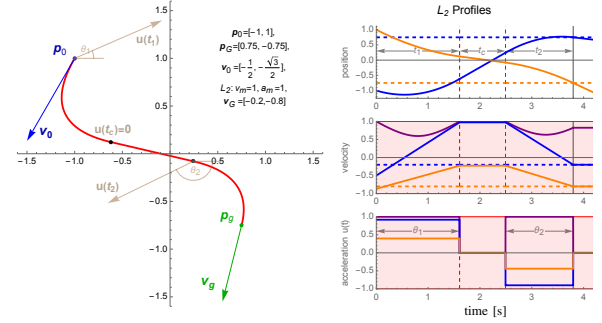[2] Department of Surgery, Hamad Medical Corporation, Doha, Qatar.

[1] https://github.com/RoboticSwarmControl/MinTimeL2pathsConstraints/

Fig. 1. Left: trajectory of a particle starting from $\mathbf{p}_0$ with initial velocity $\mathbf{v}_0$ and ending at $\mathbf{p}_g$ with ending velocity $\mathbf{v}_g$ under two constant acceleration inputs $\mathbf{u}$ applied at directions θ_1 and θ_2 for durations t_1 and t_2; the ● shapes show the switching points at t_1, t_c, and t_2 along the path, where t_c is the duration the particle cruises at its maximum velocity v_m. Right: L_2 position, velocity, and acceleration profiles. x in blue, y in orange, $\sqrt{x^2 + y^2}$ in purple. Bounds on velocity and acceleration are highlighted in pink. The control switch times are shown on the position profile. See video overview at https://youtu.be/2J-p6CDF4FE.

Restricting the number of control inputs can benefit system performance and longevity, for example by avoiding chatter in systems with high amounts of switching [5]–[7]. From a hardware perspective, limiting the number of control switches can improve lifespans. Repeated, alternating stresses are a fundamental concern in mechanical design, particularly when using fatigue-life methods to approximate the lifespan of machine components [8]. Minimizing the number of control switches can facilitate decreasing hardware degradation alongside reducing aggressive actuation by designing jerk profiles to smoothen trajectories [9], [10].

Optimal paths are common in robotics. In 1957, Dubins calculated the shortest planar path for a particle that moved at constant velocity with a constraint on the minimum turning radius [11], which has been widely used for planning for mobile robots. Extending these constraints to acceleration is natural. Carozza, Johnson, and Morgan derive the necessary equations for reaching a goal location (and velocity) in minimum time under an acceleration constraint in their paper [12]. They show that the fastest $C^{1,1}$ path from one point to another in the plane, given initial velocity, final velocity, and a bound on the magnitude of the acceleration a_m, in velocity space is a catenary. They applied it to the "Baserunner's problem" to determine the sequence of acceleration commands that enables a runner with bounded acceleration to run to all four bases on a baseball field. Their numeric process finds a local minimum. They iterate between (a) using prescribed velocities at sequential base positions and optimizing using a multidimensional Newton's method with finite difference boundary value methods to determine

the path and timing at each baseline, and (b) using a gradient descent method to optimize the velocities at each base.

In general, this is a problem of optimal control with a rich history [13]. Numerous approximations have been used. In [14], a path between two points with prescribed states is found by generating a maximum allowed velocity profile for a curve described by a spline between the points. They find the extremes of their velocity profile using the extremes in the curvature along the spline and hardware limitations on acceleration. Afterwards, the spline's control points are optimized for time using parametric programming and a lookup table containing pre-calculated paths. Similar work on optimal drone trajectories also used an iterative numeric solver [15]. Since we directly plan using a small set of acceleration commands, our solution eliminates the need for this iterative procedure.

A. Problem statement

Given scalar maximum velocity v_m and acceleration a_m, initial position $\mathbf{p}_0 = \mathbf{p}(0)$ and velocity $\mathbf{v}_0 = \dot{\mathbf{p}}(0)$ both in $\mathbb{R}^2$, and acceleration equal to the control input $\ddot{\mathbf{p}}(t) = \mathbf{u}(t)$, under the constraints that $\|\mathbf{u}(t)\|_2 \leq a_m$ and $\|\dot{\mathbf{p}}(t)\|_2 \leq v_m$, design a $\mathbf{u}(t)$ with a restricted number of changes that brings the system to $\mathbf{p}_G = \mathbf{p}(T)$ and $\mathbf{v}_G = \dot{\mathbf{p}}(T)$, both in $\mathbb{R}^2$, in minimum time T. For notational convenience, distances are in m, velocities in m/s, and acceleration in m/s^2.

B. The solution in 1D (L_∞ and L_2 solutions are equivalent)

Expressions for time-optimal trajectories for joints of a robot manipulator with velocity and acceleration constraints are provided in [16]. This section uses results from [16] to solve the problem in 1D and give context for the remaining sections.

The trajectories vary depending on the initial and final desired states, so velocity profiles are used in [16] to classify them as either critical, under-critical, or over-critical, depending on whether the distance $|\mathbf{p}_G - \mathbf{p}_0|$ allows for the bound v_m to be reached. The critical profile is defined by a critical displacement, Δp_c that results in a linear velocity profile from $\mathbf{v}_0$ to $\mathbf{v}_G$.

$$\Delta p_c = s_v \frac{\mathbf{v}_G^2 - \mathbf{v}_0^2}{2a_m}, \quad \text{where } s_v = \text{Sign}(\mathbf{v}_G - \mathbf{v}_0). \quad (1)$$

Without a constraint on velocity, the peak velocity is

$$v_p = \sqrt{s_p(p_G - p_0)a_m + \frac{v_G^2 + v_0^2}{2}}, \quad (2)$$

where $s_p = \text{Sign}(p_G - p_0 - \Delta p_c)$, $s_p \in \{-1, 0, 1\}$, accounts for the direction of the initial acceleration.

The acceleration control input for the triangular and trapezoidal velocity profiles are

$$u(t) = \begin{cases} s_p a_m & 0 \leq t < t_1 \\ 0 & t_1 \leq t < t_1 + t_c \\ -s_p a_m & t_1 + t_c \leq t < t_1 + t_c + t_2 \\ 0 & \text{otherwise} \end{cases}. \quad (3)$$

If $v_p \leq v_m$, then the velocity profile will be triangular and

$$t_1 = \frac{v_p - s_p v_0}{a_m}, \quad t_c = 0, \quad t_2 = \frac{v_p - s_p v_G}{a_m}. \quad (4)$$

If $v_p > v_m$, then the velocity profile will be trapezoidal and

$$t_1 = \frac{v_m - s_p v_0}{a_m}, \quad (5)$$

$$t_c = \frac{2 s_p a_m (p_G - p_0) + v_0^2 + v_G^2 - 2v_m^2}{2 a_m v_m}, \quad (6)$$

$$t_2 = \frac{v_m - s_p v_G}{a_m}. \quad (7)$$

The total time required is $T = t_1 + t_c + t_2$. This procedure must be modified with more than 1DOF, which is more complicated because each DOF must reach the desired position and velocity at the same time. Kroger and Wahl give a L_∞ solution in [17] that first attempts to make each DOF reach the goal velocity and position at the same time as the slowest DOF. This is done by solving for a free intermediate cruising speed, which is the root of a sextic polynomial. Sometimes a solution in this time is impossible, and a search is conducted to find the next smallest candidate synchronization time.

Similar processes are used for L_∞ controllers with limits on higher derivatives, see [18]–[22]. We used a related procedure in [22] and provided open-source code to generate smooth multi-DOF L_∞ trajectories with sinusoidal jerk profiles under jerk, acceleration, and velocity constraints.

II. SOLVING FOR UNCONSTRAINED FINAL VELOCITY

What is the fastest way to reach a goal position with one, constant, bounded acceleration input given a starting position and velocity? This problem has two variations. Either the goal position is far enough away that the robot reaches maximum velocity and coasts to the goal, or the robot does not reach maximum velocity and accelerates the entire time.

It is easy to show that all positions are reachable under a constant acceleration input by examining the reachable set. At time t, the locus of positions reachable by the particle is a circle centered at $[p_{0x} + v_{0x}t, p_{0y} + v_{0y}t]$ with radius $\frac{1}{2} a_m t^2$. The location of the particle on the circle is determined by the angle of acceleration θ_1. The gray circles in Fig. 2 show these circular loci at different times, and the initial velocity is shown by a blue arrow.

The particle first achieves the goal position at time T. The locus of positions the particle could be at time T (under all constant accelerations) are drawn in dark red. The optimal trajectory is in red, the optimal constant acceleration input in light brown, and the final velocity on the optimal trajectory is shown with a purple arrow.

The solution has two forms, depending on if the system reaches terminal velocity v_m or not. If it does not, the time t_1 can be directly solved and used to solve for θ_1. If the system reaches terminal velocity, finding t_1 requires solving for the roots of a sextic equation. The next two sections explain these approaches.
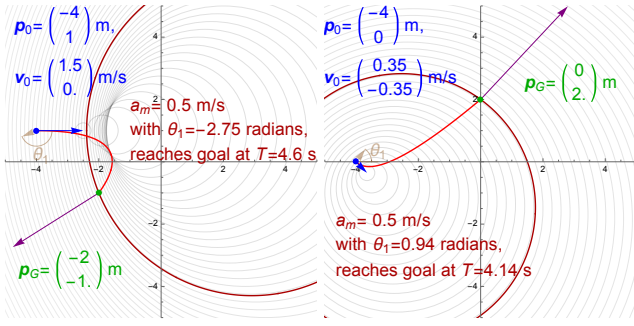
Fig. 2. Two examples of accelerating a particle from a starting position and velocity to a goal position as fast as possible with a bounded input. The locus of reachable positions is a circle whose center moves with $\mathbf{v}_0$. 100 isochrones (gray circles ◎) are evenly spaced in squared time: $t = \sqrt{k}$ for $k \in [0, 100]$ to show these loci.
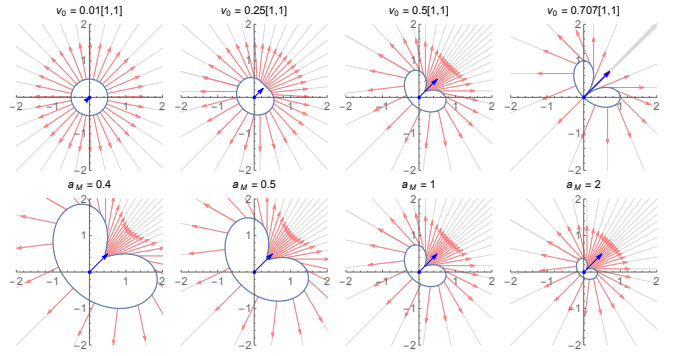

Fig. 3. Locus of positions where the particle reaches $v_m = 1$ in light blue ■. Top row shows four different starting velocities (blue). The velocity of the particle due to thrust $a_m = 1$ in directions $\theta \in k\pi/16$, $k \in [0, 31]$ is shown with pink arrows, all of length v_m. These arrows point in every direction. The bottom row shows $\mathbf{v}_0 = [1/2, 1/2]$ for four values of a_m.

A. The system does not reach terminal velocity v_m

Without loss of generality, we transform coordinates so that $\mathbf{p}_G$ is the origin. If we know the time t_1 that the particle reaches $\mathbf{p}_G$, we can solve for the angle of the acceleration:

$$\theta_1(t) = \arctan\left(-\frac{2(p_{0x} + v_{0x}t_1)}{t_1^2}, \frac{2(p_{0y} + v_{0y}t_1)}{t_1^2}\right). \quad (8)$$

Since $t_1 > 0$, we can simplify this expression to $\theta_1(t_1) = \arctan\left(-(p_{0x} + t_1 v_{0x}), p_{0y} + t_1 v_{0y}\right)$. The time t_1 is when the distance from $(\mathbf{p}_0 + \mathbf{v}_0 t_1)$ to $\mathbf{p}_G$ is $\frac{1}{2}a_m t_1^2$. Since we translated $\mathbf{p}_G$ to the origin, this results in

$$\left(\frac{1}{2}a_m t_1^2\right)^2 = (p_{0x} - v_{0x}t_1)^2 + (p_{0y} - v_{0y}t_1)^2, \quad (9)$$

This is illustrated by the dark red isochrone in Fig. 2. If we rotate the coordinate frame so $v_{0y} = 0$, and scale velocity and positions, $\tilde{\mathbf{p}}_0 = \mathbf{p}_0/a_m$, $\tilde{\mathbf{v}}_0 = \mathbf{v}_0/a_m$, we remove two constants. The smallest non-negative, real root t_1 is optimal:

The speed of the system at time t_1 is $\sqrt{(v_{0x} + a_m \cos(\theta_1)t_1)^2 + (v_{0y} + a_m \sin(\theta_1)t_1)^2}$. If this speed is greater than v_m, the system must enter a coasting phase at terminal velocity. The solution approach is described in the following section.

B. The system reaches terminal velocity

If the ending configuration is sufficiently far from the initial configuration, the goal is reachable in minimum time by a two-phase input which consists of a maximum acceleration input in direction θ_1 for t_1 seconds, followed by a coasting phase for t_c seconds.

At time t_1 the system reaches velocity v_m under a constant acceleration $a_m[\cos(\theta_1), \sin(\theta_1)]^\top$:

$$\sqrt{(v_{0x} + a_m\cos(\theta_1)t_1)^2 + (v_{0y} + a_m\sin(\theta_1)t_1)^2} = v_m. \quad (10)$$

This is a quadratic equation with two solutions for t_1, but only the positive value is relevant since we are planning forward in time. We express t_1 as a function of angle θ_1:

$$t_1(\theta_1) = \frac{\sqrt{v_m^2 - v_{0x}^2 - v_{0y}^2 + (v_{0x}\cos(\theta_1) + v_{0y}\sin(\theta_1))^2}}{a_m}$$
$$- \frac{(v_{0x}\cos(\theta_1) + v_{0y}\sin(\theta_1))}{a_m}. \quad (11)$$

The position of the particle at time t_1 is

$$p_x(t_1) = p_{0x} + v_{0x}t_1 + \frac{a_m}{2}\cos(\theta_1)t_1^2$$
$$p_y(t_1) = p_{0y} + v_{0y}t_1 + \frac{a_m}{2}\sin(\theta_1)t_1^2, \quad (12)$$

and the velocity of the particle at time t_1 is

$$v_x(t_1) = v_{0x} + a_m\cos(\theta_1)t_1$$
$$v_y(t_1) = v_{0y} + a_m\sin(\theta_1)t_1. \quad (13)$$

Figure 3 shows eight variations of the locus of positions at the terminal velocity from (12) in light blue, along with arrows showing the velocities along this set from (13) in pink. We want solutions for θ_1 that result in the velocity pointing toward to the goal at time t_1. We could check directly that

$$\arctan(v_x(t_1), v_y(t_1)) \equiv \arctan(-p_x(t_1), -p_y(t_1)), \quad (14)$$

but this involves solving for inverse trigonometric functions. Instead, we compare the slope of the velocity to the slope of the position error:

$$\frac{v_y(t_1)}{v_x(t_1)} \equiv \frac{-p_y(t_1)}{-p_x(t_1)}. \quad (15)$$

This results in two candidate solutions, but we can check both using (14) and save the correct solution. We will also have to check for zeros of the equation in the same way. The resulting equation is

$$v_x(t_1)p_y(t_1) - v_y(t_1)p_x(t_1) \equiv 0. \quad (16)$$

Solving for $\sin(\theta_1)$ results in a sextic equation. This equation is long, so it is shared in the Appendix [23]. We solve for the six roots of a sextic equation in $\sin(\theta_1) = s$, and discard the complex roots. Each remaining root is a solution for $\sin(\theta_1)$ and provides two possible θ_1 solutions since $\theta_1 = \arctan(\pm\sqrt{1 - s^2}, s)$. We substitute each possible θ_1 solution into (11) to get at most 12 candidate t_1 solutions. The smallest, real, non-negative t_1 that satisfies (14) is used.

Finding the root of a sextic equation can be efficiently computed in many programming languages [24].
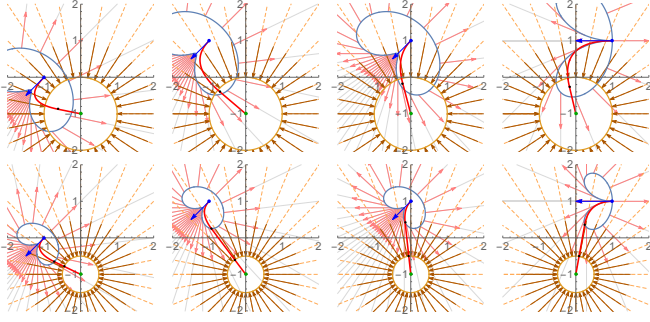
Fig. 4. Solution with nonzero starting and zero ending velocity. The locus of positions where the particle reaches $v_m = 1$ from the initial position are shown with a light blue set for $\mathbf{p}(t_1)$, ■. The corresponding set backwards from $\mathbf{p}_G$ is a golden-colored circle with radius (20a), ■. The solution trajectory is drawn in red. The positions where thrust 1 stops and thrust 2 starts are indicated by black dots. Top row shows four different initial positions with $\mathbf{v}_0 = [-0.5, -0.5]$ or $[-1, 0]$ and $a_m = 1/2$. The system never exceeds terminal velocity. Bottom row shows the same positions and velocities, but with $a_m = 1$, so each requires a coasting phase.

III. SOLVING FOR ZERO FINAL VELOCITY

This section provides solutions to problems that stop at the goal position $\mathbf{p}_G$. There are two cases depending on if the solution requires coasting at the maximum velocity or not. Both cases require finding the roots of a sixth order polynomial. Sample solutions are shown in Fig. 4, which also shows the set $\mathbf{p}(t_1)$ that is calculated with (12), as well as the corresponding set centered at $\mathbf{p}_G$. Because $\mathbf{v}_G = [0, 0]$, this set is a circle. For any goal state with zero velocity, we can transform the coordinates so the goal position is at $[0, 0]^\top$. We can then rotate the coordinate frame such that $v_{0y} = 0$.

A. The solution is bang-bang

If the starting and ending position are sufficiently close such that the velocity never exceeds v_m, then the goal is reachable in minimum time by a two-phase input which consists of a maximum acceleration input in direction θ_1 for t_1 seconds, followed by a maximum acceleration input opposing the current velocity to bring the system to rest in $t_2 = \|\mathbf{v}(t_1)\| / a_m$ seconds ($t_c = 0$).

After applying the constant input $a_m[\cos(\theta_1), \sin(\theta_1)]^\top$ for t_1 seconds, the position and velocity are

$$\mathbf{p}(t_1) = \begin{bmatrix} p_{0x} + v_{0x}t_1 + \frac{a_m}{2}\cos(\theta_1)t_1^2 \\ p_{0y} + \frac{a_m}{2}\sin(\theta_1)t_1^2 \end{bmatrix}$$
$$\mathbf{v}(t_1) = \begin{bmatrix} v_{0x} + a_m\cos(\theta_1)t_1 \\ a_m\sin(\theta_1)t_1 \end{bmatrix} \quad (17)$$

The deceleration command is in the opposite direction of $\mathbf{v}(t_1)$ so that $\theta_2 = \arctan(-v_x(t_1), -v_y(t_1))$, and lasts for $t_2 = \|\mathbf{v}(t_1)\| / a_m$ seconds. At time $t_1 + t_2$, we want the x and y positions to be zero and the final velocity to be zero. The final position is entirely controlled by the initial conditions and the selected θ_1 and θ_2:

$$t_2 = \frac{\|\mathbf{v}(t_1)\|}{a_m} = \sqrt{\left(\frac{v_{0x}}{a_m} + \cos(\theta_1)t_1\right)^2 + (\sin(\theta_1)t_1)^2}$$

$$0 = p_x(t_1) + v_x(t_1)\frac{t_2}{2}, \qquad 0 = p_y(t_1) + v_y(t_1)\frac{t_2}{2}. \quad (18)$$

We then scale the starting position and velocity by dividing each by a_m and remove the term a_m from the calculation: $\tilde{\mathbf{p}}_0 = \mathbf{p}_0/a_m$, $\tilde{\mathbf{v}}_0 = \mathbf{v}_0/a_m$. We apply a change of variables to eliminate the two trigonometric functions: $\cos(\theta_1) = c$, and $\sin(\theta_1) = \pm\sqrt{1 - c^2}$. The resulting position constraints simplify to:

$$0 = 2\tilde{p}_{0x} + 2\tilde{v}_{0x}t_1 + ct_1^2 + (\tilde{v}_{0x} + ct_1)\sqrt{\tilde{v}_{0x}^2 + 2c\tilde{v}_{0x}t_1 + t_1^2}$$

$$0 = 2\tilde{p}_{0y} + \sqrt{1 - c^2}t_1\left(\sqrt{\tilde{v}_{0x}^2 + 2c\tilde{v}_{0x}t_1 + t_1^2} + t_1\right). \quad (19)$$

This set of equations can be solved for c as a function of t_1. The calculations are long, and are shared in Appendix [23] (see also the code implementation).

We solve for the roots of this sextic equation in t_1 to get six candidate t_1 values. We substitute each non-negative, real value into a closed-form equation to compute candidate $c = \cos(\theta_1)$ from t_1. Since $\theta_1 = \arctan(c, \pm\sqrt{1 - c^2})$, this provides at most 12 candidate θ_1 values. We test each θ_1 value in (18) and select the solution with zero position error that minimizes the total time.

Figure 4, top row shows four solutions. In each, the solution trajectory is entirely contained within the union of the $\mathbf{p}(t_1)$ locus and a maximum braking radius circle centered at $\mathbf{p}_G$.

B. The solution requires a cruising phase

If the solution from Section III-A requires a velocity larger than v_m, we must have a cruising phase at the maximum velocity. The goal is reachable by a three-phase input. This control consists of a maximum acceleration input in direction θ_1 for t_1 seconds, followed by a cruising phase for t_c seconds, followed by a maximum acceleration input opposing the current velocity to bring the system to rest in t_2 seconds.

We first find the two-phase solution from Sec. II-B to reach the goal position $\mathbf{p}_G$ by accelerating in direction θ_1 for t_1 seconds. Rather than cruising from time t_1 to the goal, we need to start braking t_2 seconds away from the goal.

$$r = \frac{v_m^2}{2a_m} \quad \text{braking radius.} \quad (20\text{a})$$

distance from $\mathbf{p}(t_1)$ to goal:

$$d = \left\| \mathbf{p}_G - \left(\mathbf{p}_0 + \mathbf{v}_0 t_1 + \frac{a_m}{2}\begin{bmatrix}\cos(\theta_1)\\\sin(\theta_1)\end{bmatrix}t_1^2\right) \right\| \quad (20\text{b})$$

$$t_c = \frac{d - r}{v_m} \quad \text{cruising time, } t_2 = \frac{v_m}{a_m} \quad \text{braking time.} \quad (20\text{c})$$

braking direction: $\quad (20\text{d})$

$$\theta_2 = -\arctan\left(v_{0x} + a_m\cos(\theta_1)t_1, v_{0y} + a_m\sin(\theta_1)t_1\right).$$

The thrust time t_1 is a function of θ_1 as given in (11), and so (20) are all functions of only θ_1. The bottom row of Fig. 4 shows four representative solutions.

IV. SOLVING FOR NON-ZERO FINAL VELOCITY

Solving for a non-zero final velocity $\mathbf{v}_G$ is harder, but is necessary for smoothly traversing through waypoints [25].
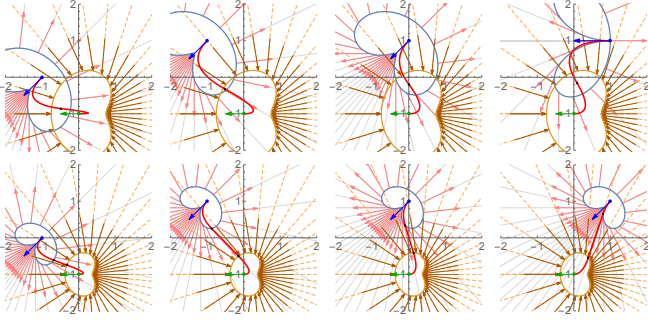
Fig. 5. Solution with nonzero starting and ending velocity. The locus of positions where the particle reaches $v_m = 1$ from the initial position are shown with a light blue set for $\mathbf{p}(t_1)$. A similar set constructed starting from the goal position using $-\mathbf{v}_G$ is shown in orange. The solution trajectory is drawn in red. The positions where thrust 1 stops and thrust 2 starts are indicated by black dots. Top row shows four different initial positions with $\mathbf{v}_0 = [-0.5, -0.5]$, $\mathbf{v}_G = [-0.5, 0]$ and $a_m = 1/2$ (the right differs to ensure v_m is not reached). The system never reaches terminal velocity. Bottom row shows the same positions and velocities, but with $a_m = 1$. The system reaches terminal velocity v_m in each case.

A final velocity adds two parameters to the equations. Currently, we use a numeric solver to find a bang-bang solution to get to the goal with no velocity limit. If this solution results in a maximum velocity greater than v_m at the switching point of the bang-bang controller, we call a second function to solve for a cruising phase.

Sample solutions are shown in Fig. 5, which also shows the set $\mathbf{p}(t_1)$ that is calculated with (12). The corresponding set centered at $\mathbf{p}_G$ is generated using the same process, but with $-\mathbf{v}_G$ as input.

A. No cruising phase, non-zero final velocity

The bang-bang controller is described by the following inputs, which apply maximum thrust in direction θ_1 for t_1 seconds, and then maximum thrust in direction θ_2 for t_2 seconds. There is no cruising phase, so $t_c = 0$. Here, $\mathbf{p}(t_1)$ and $\mathbf{v}(t_1)$ are the same as in (12) and (13). Define the position and velocity generated by starting at $\mathbf{p}_G$ with velocity $\mathbf{v}_G$ and applying acceleration in direction θ_2 *backwards in time* for t_2 seconds as $[\mathbf{p}_{\text{-}t_2}, \mathbf{v}_{\text{-}t_2}]$.

$$p_x(t_1) \equiv p_{Gx} - v_{Gx} t_2 + \frac{a_m}{2} \cos(\theta_2) t_2^2 = p_{\text{-}t_2,x} \quad (21\text{a})$$

$$p_y(t_1) \equiv p_{Gy} - v_{Gy} t_2 + \frac{a_m}{2} \sin(\theta_2) t_2^2 = p_{\text{-}t_2,y} \quad (21\text{b})$$

$$v_x(t_1) \equiv v_{Gx} - a_m \cos(\theta_2) t_2 \qquad = v_{\text{-}t_2,x} \quad (21\text{c})$$

$$v_y(t_1) \equiv v_{Gy} - a_m \sin(\theta_2) t_2 \qquad = v_{\text{-}t_2,y} \quad (21\text{d})$$

Solving this nonlinear set of constraints requires a good starting estimate. We run the solver multiple times, using a set of candidate guesses for the unknowns $[\theta_1, t_1, \theta_2, t_2]$. To generate good candidate guesses, we first wrap the procedure in Section III-A into the function

$$(\theta_1, t_1) = \text{stopAtGoalNoCoast}[\mathbf{p}_0, \mathbf{v}_0, \mathbf{p}_G, a_m, v_m], \quad (22)$$

that returns the necessary thrust direction and time for a given initial position and velocity, a desired stopping position $\mathbf{p}_G$,

and acceleration and velocity constraints. Next, we generate a set of positions pts:

$$pts = \{\mathbf{p}_0,\ \mathbf{p}_1,\ \frac{1}{2}(\mathbf{p}_1 + \mathbf{p}_4),\ \mathbf{p}_4,\ \mathbf{p}_G\}, \text{ where} \quad (23)$$

$$\mathbf{p}_1 = \mathbf{p}_0 + \frac{\|\mathbf{v}_0\|}{2a_m}\mathbf{v}_0, \quad \mathbf{p}_4 = \mathbf{p}_G - \frac{\|\mathbf{v}_G\|}{2a_m}. \quad (24)$$

Here $\mathbf{p}_1$ is reached by maximum braking starting from $\mathbf{p}_0$ with velocity $\mathbf{v}_0$, and $\mathbf{p}_4$ is reached by maximum braking from $\mathbf{p}_G$ and initial velocity $-\mathbf{v}_G$. We use candidate starting values

$$\text{stopAtGoalNoCoast}[\mathbf{p}_0, \mathbf{v}_0, pts[k], a_m, v_m],\ \ k \in \{3, 4, 5\}$$

$$\text{stopAtGoalNoCoast}[\mathbf{p}_G, -\mathbf{v}_G, pts[j], a_m, v_m],\ \ j \in \{1, 2, 3\}$$

Of all solutions that reach the goal, we select the solution that minimizes the total time. The fastest velocity occurs at the switching time. If $\|\mathbf{v}_{t_1}\| > v_m$ a cruising phase is required.

B. Cruising phase, non-zero final velocity

If the solution from Sec. IV-A requires a velocity greater than v_m, we must have a cruising phase at the maximum velocity. Solving with a cruising phase is conceptually easier. We know that the solution trajectory reaches terminal velocity, and thus t_1 is determined by θ_1 and t_2 by θ_2, both by using (11). We merely need to find a θ_1 and a θ_2 that solve the problem. The velocities while cruising and the scaled difference in position are all equal, along vector ϕ:

$$\mathbf{v}_{t_1} \equiv \mathbf{v}_{\text{-}t_2} \equiv v_m \frac{\mathbf{p}_{\text{-}t_2} - \mathbf{p}_{t_1}}{\|\mathbf{p}_{\text{-}t_2} - \mathbf{p}_{t_1}\|} \equiv v_m \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix}. \quad (25)$$

Given a ϕ, we can solve for θ_1 and θ_2:

$$\theta_1 = \arctan(v_m \cos(\phi) - v_{0x}, v_m \sin(\phi) - v_{0y}) \quad (26)$$

$$\theta_2 = \arctan(v_{Gx} - v_m \cos(\phi), v_{Gy} - v_m \sin(\phi)) \quad (27)$$

Then we must solve for the parameter ϕ:

$$\phi \equiv \arctan(\mathbf{p}_{\text{-}t_2 x} - \mathbf{p}_{t_1 x}, \mathbf{p}_{\text{-}t_2 y} - \mathbf{p}_{t_1 y}), \text{ let}$$

$$r_0 = \sqrt{(v_{0x} - v_m \cos(\phi))^2 + (v_{0y} - v_m \sin(\phi))^2},$$

$$r_G = \sqrt{(v_{Gx} - v_m \cos(\phi))^2 + (v_{Gy} - v_m \sin(\phi))^2}, \text{ then}$$

$$\phi \equiv \arctan(2a_m(p_{0x}+p_{Gx})-v_m(r_0+r_G)\cos(\phi)-r_0 v_{0x}-r_G v_{Gx},$$

$$-2a_m(p_{0y}-p_{Gy})-v_m(r_0+r_G)\sin(\phi)-r_0 v_{0y}-r_G v_{Gy}). \quad (28)$$

This equation is nonlinear, but ϕ is the only variable. We use a Van der Corput sequence with base 2 in the range $[-\pi, \pi]$ to sample ϕ evenly with increasing refinement. We perform root finding on ϕ, initializing our guess with an element from the Van der Corput sequence, and iterate until $\|\mathbf{p}_G - \mathbf{p}(T)\| + \|\mathbf{v}_G - \mathbf{v}(T)\|$ is less than $e_{\min}$.

To test this algorithm, we randomly generated initial and final positions $(\mathbf{p}_0, \mathbf{p}_G)$ within a circle of radius $r = 2$, and velocities $(\mathbf{v}_0, \mathbf{v}_G)$ in a radius $r = 1$ until we had 10,000 initial conditions that required a cruising phase for $a_m = 1, v_m = 1$. We used $e_{\min} = 10^{-12}$. Of these, 9562 converged within $e_{\min}$ using the first sequence value, 9712 in the first two values, and 9911 in the first 5 values. The average number of values needed was 1.17, and the largest
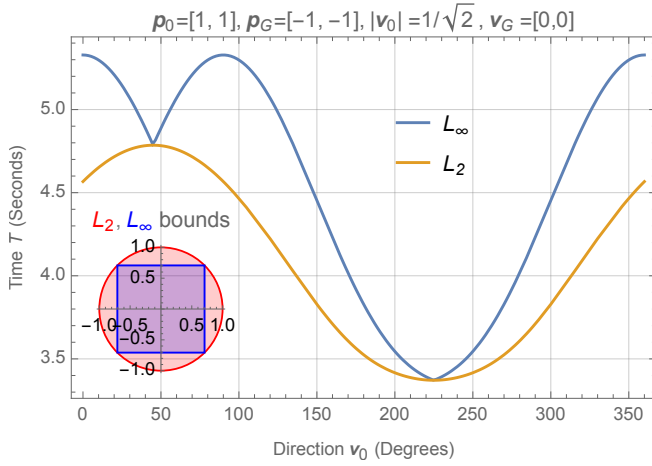
Fig. 6. Comparing the finishing times for L_2 and L_∞ solutions with bounds $L_2 : a_m = 1, v_m = 1$ and $L_\infty : a_m = 1/\sqrt{2}, v_m = 1/\sqrt{2}$, which are shown in the inset graphic in the lower left. Because of these bounds the L_∞ solution is in general longer than the L_2 solution.

was 87. Solving 10,000 queries required 26.5 seconds on a 3.3 GHz i7 laptop.

V. TRAJECTORY EXAMPLES AND ANALYSIS

This section showcases several examples and contrasts our minimum-time trajectory solutions modeled using L_2 bounds to those modeled with L_∞ bounds. The time-optimal trajectories modeled with L_∞ bounds were calculated using methods in [16], and the two DOF's were synchronized using Kroger and Wahl's search technique from [17].

Using a controller designed using L_2 bounds results in a controller that is never slower than a control that uses the L_∞ bounds that obey the L_2 bounds. The times are only the same if the solution trajectory lies entirely along a slope of ± 1. This is illustrated in Fig. 6 for $\mathbf{p}_0 = [1, 1]$, $\mathbf{p}_G = [-1, -1]$; finishing times for L_2 and L_∞ are only equal for $\mathbf{v}_0$ angular directions of $45°$ and $225°$. At these $\mathbf{v}_0$ directions, the problem is effectively 1D (identical x and y velocities).

For all these solutions there are several advantages of the L_2 solution. The path requires less time is often a shorter distance. The path also requires less control changes. Additionally, the L_2 acceleration and velocity profiles spend more time at their bounds.

VI. CONCLUSION

This work found control expressions for position and velocity control in 2D with L_2 constraints on acceleration and velocity. Future work should extend this to 3D, which is promising for cases with high symmetry such as when the final velocity is zero. Extending the work of [12] to quickly derive the optimum solution and to incorporate velocity constraints is another exciting direction for future work.

REFERENCES

[1] V. M. Baez, H. Zhao, N. Abdurahiman, N. V. Navkar, and A. T. Becker, "Minimum-time planar paths with up to two constant acceleration inputs and l_2 velocity and acceleration constraints," 2024.

[2] K. R. Duda, R. W. Loffi, and P. M. Handley, "System and method for assisted extravehicular activity self-return," May 2018, US Patent 9,977,427.

[3] V. Evans, "Newton's laws, g-forces and the impact on the brain," *The Australasian journal of neuroscience*, vol. 30, no. 1, pp. 24–29, 2020.

[4] M. Laux and A. Zell, "Robot arm motion planning based on geodesics," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7585–7591.

[5] A. De Marchi, "On the mixed-integer linear-quadratic optimal control with switching cost," *IEEE Control Syst. Lett.*, vol. 3, no. 4, pp. 990–995, 2019.

[6] F. Bestehorn, C. Hansknecht, C. Kirches, and P. Manns, "Mixed-integer optimal control problems with switching costs: a shortest path approach," *Math. Program.*, vol. 188, no. 2, pp. 621–652, 2021.

[7] C. Kirches, E. Kostina, A. Meyer, M. Schlöder, and S. PN, "Numerical solution of optimal control problems with switches, switching costs and jumps," *Optimization Online*, vol. 6888, 2018.

[8] R. G. Budynas, *Shigley's mechanical engineering design*, 10th ed., ser. McGraw-Hill series in mechanical engineering. New York, NY: McGraw-Hill Education, 2015.

[9] S. Lu, J. Zhao, L. Jiang, and H. Liu, "Solving the time-jerk optimal trajectory planning problem of a robot using augmented lagrange constrained particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2017, 2017.

[10] R. Zhao and D. Sidobre, "Trajectory smoothing using jerk bounded shortcuts for service manipulator robots," in *IEEE/RSJ IROS*. IEEE, 2015, pp. 4929–4934.

[11] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Am. J. Math.*, vol. 79, no. 3, pp. 497–516, 1957.

[12] D. Carozza, S. Johnson, and F. Morgan, "Baserunner's optimal path," *The Mathematical Intelligencer*, vol. 32, no. 1, pp. 10–15, 2010.

[13] H. J. Sussmann and J. C. Willems, "300 years of optimal control: from the brachystochrone to the maximum principle," *IEEE Control Systems Magazine*, vol. 17, no. 3, pp. 32–44, 1997.

[14] M. Lepetič, G. Klančar, I. Škrjanc, D. Matko, and B. Potočnik, "Time optimal path planning considering acceleration limits," *Robotics and Autonomous Systems*, vol. 45, no. 3, pp. 199–210, 2003.

[15] M. Hehn, R. Ritz, and R. D'Andrea, "Performance benchmarking of quadrotor systems using time-optimal control," *Autonomous Robots*, vol. 33, no. 1-2, pp. 69–88, 2012.

[16] F. Ramos, M. Gajamohan, N. Huebel, and R. D'Andrea, "Time-optimal online trajectory generator for robotic manipulators," Eidgenössische Technische Hochschule Zürich, Institute for Dynamic Systems, Tech. Rep., 2013.

[17] T. Kröger and F. M. Wahl, "Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 94–111, 2010.

[18] H. Liu, X. Lai, and W. Wu, "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints," *Robot. Comput.-Integr. Manuf.*, vol. 29, no. 2, pp. 309–317, 2013.

[19] A. A. Ata, "Optimal trajectory planning of manipulators: a review," *Int. J. Eng. Sci. Technol.*, vol. 2, no. 1, pp. 32–54, 2007.

[20] S. A. Bazaz and B. Tondu, "Minimum time on-line joint trajectory generator based on low order spline method for industrial manipulators," *Robot. Auton. Syst.*, vol. 29, no. 4, pp. 257–268, 1999.

[21] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Trans. Autom. Control*, vol. 54, no. 10, pp. 2318–2327, 2009.

[22] H. Zhao, N. Abdurahiman, N. Navkar, J. Leclerc, and A. T. Becker, "Jerk-continuous online trajectory generation for robot manipulator with arbitrary initial state and kinematic constraints," in *IEEE/RSJ IROS*. IEEE, 2022, pp. 5730–5736.

[23] V. M. Baez and A. T. Becker, "Appendix: Equations for minimum-time planar paths with l2 velocity and acceleration constraints and a limited number of constant acceleration inputs," University of Houston, Tech. Rep., 2024. [Online]. Available: https://github.com/RoboticSwarmControl/MinTimeL2pathsConstraints

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[25] J. Lin, N. Somani, B. Hu, M. Rickert, and A. Knoll, "An efficient and time-optimal trajectory generation approach for waypoints under kinematic constraints and error bounds," in *IEEE/RSJ IROS*. IEEE, 2018, pp. 5869–5876.