

Efficient Waypoint Tracking Hybrid Controllers for Double Integrators using Classical Time Optimal Control

Haitham Hindi and Lara Crawford and Rong Zhou and Craig Eldershaw

Abstract—This paper is a response to requests from several respected colleagues in academia for a careful writeup of the classical time-optimal control based hybrid controllers that we have been using for material transport control in our modular reconfigurable manufacturing systems application. Specifically, we show how classical closed form time optimal control, which exploits the special structure of double integrator systems (ie ones with point-mass Newton’s Law dynamics) can be used to design hybrid controllers for waypoint tracking. While double integrator dynamics are very simple, they are extremely prevalent in many domains, beyond manufacturing systems, eg, transportation, disk drives, robotics, and aerospace.

We present two methods, one which will track general way point specs, and another which will track waypoints that lie on convex or concave trajectories. Both approaches are based on appropriate setting (switching) of the state of a reference generator with the same point mass dynamics in a two-degree-of-freedom controller topology. The techniques we present admit very compact implementations, suitable for use in low cost micro controllers and DSP chips used in modular reconfigurable embedded systems applications. Since these controllers were to be integrated with a discrete planner, as part of the control software of a large complex new research platform, every effort was made to keep the controllers as simple as possible. Our controllers can be viewed as examples of basic hybrid controllers that are being used successfully in practice, and can be used as benchmarks by controls researchers for more sophisticated hybrid control design methods.

I. INTRODUCTION

This paper is a response to requests from several respected colleagues in academia for a careful writeup of the classical time-optimal control based hybrid controllers that we have been using for material transport control in our modular reconfigurable manufacturing systems application. Specifically, we show how classical closed form time optimal control [14], [5], [1], [9], which exploits the special structure of double integrator systems (ie ones with point-mass Newton’s Law dynamics) can be used to design hybrid controllers for waypoint tracking. While double integrator dynamics are very simple, they are extremely prevalent in many domains, beyond manufacturing systems, eg, transportation, disk drives, robotics, and aerospace.

We present two methods, one which will track general way point specs, and another which will track waypoints that lie on convex or concave trajectories. Both approaches are based on appropriate setting (switching) of the state of a reference generator with the same point mass dynamics in a two degree of freedom controller topology [14], [5], [1], [16], [9].

The authors are with Palo Alto Research Center (PARC), California, USA. {hhindi, lcrawford, rzhou, celdersh}@parc.com

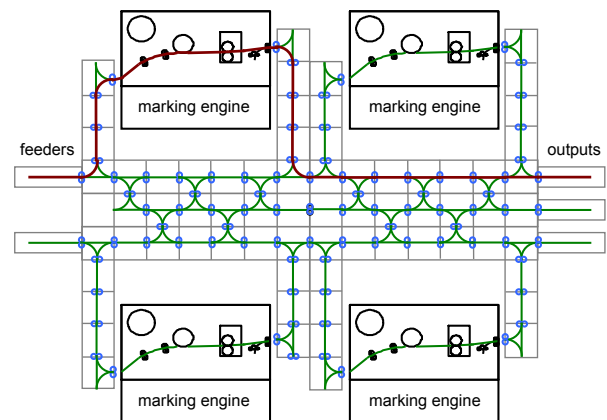


Fig. 1. Diagram of the PARC hypermodular high speed parallel printer fixture. Paper feeders (inputs) are on the left, and paper outputs are on the right. The four large boxes are marking engines, which actually perform the printing. There are three high-speed “highways” in the fixture, with “off-ramps” and “on-ramps” sending paper to and from the marking engines. Many cross-connections link the highways. A typical sheet path is shown in red. Black boxes indicate paper path module boundaries. Typical letterhead paper is about 2 modules in length. Each module contains two or three independently controlled stepper motor actuated rollers to move the paper along, shown as double circles. There are sensors at each module boundary (not shown). The modules communicate via a CAN bus network.

Our approach can be viewed as a special case of the more sophisticated modern methods available today, see for example [3], [6], [2], [4]. However, the techniques we present admit very compact implementations, suitable for use in low cost micro controllers and DSP chips used in modular reconfigurable embedded systems applications. Since these controllers were to be part of a complex system integration effort for a new research platform, every effort was made to keep the controllers as simple as possible. Our controllers can be viewed as examples of basic hybrid controllers that are being used successfully in practice, and can be used as benchmarks by controls researchers for more sophisticated hybrid control design methods [13], [12], [15], [2], [4], [10].

Our work is motivated by tracking control problems encountered, while working on the PARC hypermodular high speed parallel printer fixture, shown in Fig.1. The fixture is designed to be highly modular and reconfigurable, both in hardware and in software, and to serve as a testbed to explore the integration of control and planning. The goal of the control is drive paper along the paper paths shown, while meeting waypoint tracking requirements. The waypoints for

each sheet of paper, along with the routing and scheduling, are determined by a separate combinatorial search planner which will not be discussed here, see [11], [7]. In this paper, our focus will be on the controllers used to achieve the desired waypoint tracking requirements.

II. PROBLEM STATEMENT

We will be concerned mainly with the double integrator system:

$$\begin{aligned}\dot{s} &= v \\ \dot{v} &= u\end{aligned}\quad (1)$$

where s and v are the position (m) and speed (m/s), respectively; u is the acceleration control signal (m/s/s). If we define the plant state $x_P = (s, v)$, then (1) can be written in state space form as

$$\dot{x}_P(t) = Ax_P(t) + Bu(t); \quad x_P(t_0) = x_{P0} \quad (2)$$

where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Note that $[A, B]$ is controllable, and the system has an invertible time evolution matrix

$$\Phi(t_0, t_f) = e^{A(t_f - t_0)} = \begin{bmatrix} 1 & (t_f - t_0) \\ 0 & 1 \end{bmatrix}.$$

In many modern control systems, such as the one in Fig.1, an abstract planner handles task ordering and scheduling, and then passes a set of *waypoints* to the physical controller. These are a set of specific times, locations, and speeds, $\{(t_0, s_0, v_0), \dots, (t_N, s_N, v_N)\}$ for the system to achieve at certain times. Suppose we view the way points as samples at instants $\{t_0, \dots, t_N\}$ of a fictitious reference state trajectory $x_R(t)$, whose values at those times are $x_R(t_0) = (s_0, v_0), \dots, x_R(t_N) = (s_N, v_N)$. Then we may state the *waypoint tracking problem* in its most skeletal form as

$$\begin{aligned}\min_u \quad & \sum_{t_i \in \{t_0, \dots, t_N\}} \|x_P(t_i) - x_R(t_i)\|_p \\ \text{s.t.} \quad & \dot{x}_P(t) = Ax_P(t) + Bu(t); \quad x_P(t_0) = (s(t_0), v(t_0)) \\ & \|u\|_\infty \leq a_{\max}\end{aligned}$$

This captures our primary objective, namely that of hitting the waypoints without exceeding our motor peak acceleration constraints.

As systems become more modular, and as the modules themselves become more autonomous, eventually, one arrives at the following single module waypoint tracking problem, which will be the focus of this paper. The multiple waypoint tracking problem above can be decomposed into a sequence of single module problems, possibly by using tricks such as fictitious modules, to handle nonuniform spacing, etc.

Problem Statement: [*single module waypoint tracking*] Given a module with double integrator dynamics (2), find a control $u: \mathbf{R}^2 \rightarrow \mathbf{R}$ that will take the state from $x_{P0} = (s_0, v_0)$ at t_0 to $x_{Pf} = (s_f, v_f)$ at t_f , while not exceeding the peak acceleration limit a_{\max} . If such a control u can be found from a certain class of control laws, then we say that the

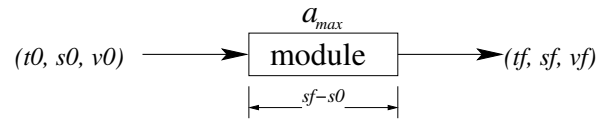


Fig. 2. Single module waypoint tracking

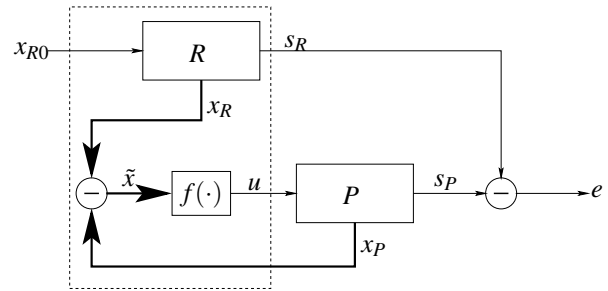


Fig. 3. Two degree of freedom controller topology. The controller consists of R and $f(\cdot)$; takes 2 inputs: initial condition x_{R0} and plant state x_P ; and produces the control output $u = f(\tilde{x})$.

pair $\{(t_0, x_{P0}), (t_f, x_{Pf})\}$ is a_{\max} -feasible under that class of control laws. See Fig.2.

Note that the above only captures the most critical constraints. In practice, one would like to have all the following:

- 1) pass through the waypoints (or as close as possible),
- 2) don't exceed acceleration limits
- 3) easy to check that $\{(t_0, x_{P0}), (t_f, x_{Pf})\}$ is a_{\max} -feasible
- 4) feedback implementation to reject disturbances
- 5) low memory and computation costs for DSP implementation
- 6) compact parametrization to fit network bandwidth limits
- 7) minimal transit time through overall path
- 8) smooth machine-friendly position and velocity trajectories
- 9) speed should always be positive (no moving backwards)

III. 2DOF CONTROL TOPOLOGY

We now introduce the basic idea that we use to solve the waypoint tracking problem. Specifically, we will make use of the two degree of freedom controller (2DOF) topology, Fig.3, which uses a reference generator to provide *feedforward* control, and a state *feedback* for stabilization and disturbance rejection. This is a classical architecture, discussed in many standard references [14], [5], [1], [16], [9]. The equations for the overall system are now

$$\begin{aligned}\dot{x}_R &= Ax_R & ; x_R(t_0) &= x_{R0} \\ \dot{x}_P &= Ax_P + Bu & ; x_P(t_0) &= x_{P0}\end{aligned}\quad (3)$$

In this paper, we assume the plant state is directly available to the controller; if not, an observer can be used as part of the controller to estimate x_P . In the hypermodular printer fixture in Fig.1, each stepper motor is controlled independently by its own DSP processor, which runs such a 2DOF controller.

Proposition 1: Consider the problem of tracking the state of a linear reference generator, as in (3), whose state is fully

observable, with the same dynamics A -matrix as the plant. Let $u: \mathbf{R}^n \rightarrow \mathbf{R}$ be a stabilizing (possibly nonlinear) state feedback for the plant. Let $\tilde{x} := x_P - x_R$. Then the control $u(\tilde{x})$ will stabilize the tracking error \tilde{x} and drive it to zero with the same closed loop dynamics as it has on the plant by itself.

Proof: Trivially we observe that subtracting the two state equations in (3) gives

$$\dot{\tilde{x}} = A\tilde{x} + Bu(\tilde{x})$$

which has exactly the same dynamics as the original system in closed loop with the stabilizing state feedback u . ■

Proposition 2: Assume that $\Phi(t_0, t_f) = e^{A(t_f - t_0)}$ is invertible and that $[A, B]$ is controllable. Set $x_R(t_0) = e^{A(t_0 - t_f)} x_{Pf}$. Then if u is a control law which takes the state $\tilde{x} := x_P - x_R$ to zero in finite time $t^* \in [t_0, t_f]$, then $u(\tilde{x})$ solves the single module waypoint trajectory control problem.

Proof: Since Φ is invertible, we can propagate x_R back in time from t_f to t_0 . Now from time t^* onwards, we have $x_P(t^*) \equiv x_R(t^*)$. By controllability, this means that $u \equiv 0, \forall t \geq t^*$. Thus x_P and x_R will both evolve homogeneously according to $x_P(t) = x_R(t) = e^{A(t - t^*)} x_R(t^*), \forall t \geq t^*$. Since the reference generator was evolving homogeneously all along, we have $x_R(t^*) = e^{A(t^* - t_0)} x_R(t_0) = e^{A(t^* - t_0)} e^{A(t_0 - t_f)} x_{Pf} = e^{A(t^* - t_f)} x_{Pf}$. Substituting into the expression for $x_P(t)$ and evaluating at $t = t_f$ gives x_{Pf} . ■

Thus given any finite time stabilizing control law u for the double integrator system, we have a paradigm for solving the single module waypoint tracking problem as follows:

- 1) check points are a_{\max} -feasible
- 2) set $x_R(t_0) = e^{A(t_0 - t_f)} x_{Pf}$
- 3) apply control law $u(\tilde{x})$ from t_0 to t_f

IV. TIME OPTIMAL CONTROL BASED TRACKING

We will now review some classical results on time optimal control for double integrator systems, which will supply us with a finite time stabilizing control law, that can be used in the propositions above to design waypoint tracking controllers, see [14], [5], [1], [9].

The *time optimal control* problem is

$$\begin{aligned} \min_u \quad & T = \int_{t_0}^{t_0+T} 1 \, dt \\ \text{s.t.} \quad & \dot{x}(t) = Ax(t) + Bu(t); \quad x(0) = x_o \\ & x(T) = 0 \\ & \|u\|_{\infty} \leq a_{\max} \end{aligned}$$

The solution to the time optimal control problem for the double integrator system is a *nonlinear* state-dependent control law

$$\begin{aligned} u(t) &= f_{\text{TOC}}(x(t)) \\ &= -a_{\max} \text{sgn} \left(\text{sgn}(s(t)) \sqrt{2a_{\max}|s(t)|} + v(t) \right) \end{aligned} \quad (4)$$

where recall that the state $x(t) = (s(t), v(t))$.

At this point one might be tempted to apply $u(t) = f_{\text{TOC}}(\tilde{x}(t))$ right away to the plant and reference system to solve the problem. But first we need to be able to check

a_{\max} -feasibility of a given pair of waypoints under the control law $f_{\text{TOC}}(\tilde{x}(t))$. Fortunately, once again thanks to the special structure of the double integrator system, there is a closed form expression for the time optimal transfer time: For the double integrator system (2), the time taken to bring the state from x_0 to the origin under the time optimal control law (4) is given by

$$t_{\text{TOC}}(x(t_0)) = \begin{cases} v(t_0) + 2\sqrt{s(t_0) + \frac{v(t_0)^2}{2a_{\max}}} & ; s(t_0) > -\frac{v(t_0)|v(t_0)|}{2a_{\max}} \\ -v(t_0) + 2\sqrt{-s(t_0) + \frac{v(t_0)^2}{2a_{\max}}} & ; s(t_0) < -\frac{v(t_0)|v(t_0)|}{2a_{\max}} \\ |v(t_0)| & ; s(t_0) = -\frac{v(t_0)|v(t_0)|}{2a_{\max}} \end{cases} \quad (5)$$

Using these facts together with Prop1 and Prop2, we can propose the following controller for the single module waypoint tracking problem:

Proposition 3: Consider the double integrator system, with acceleration limit a_{\max} , together with the reference generator system as in (3). Let $\tilde{x} := x_P - x_R$, and suppose we want to move from $x_{P0} = (s_0, v_0)$ at t_0 to $x_{Pf} = (s_f, v_f)$ at t_f . Then by taking $x_R(t_0) = e^{A(t_0 - t_f)} x_{Pf}$, the control

$$u(\tilde{x}) = f_{\text{TOC}}(\tilde{x}) \quad (6)$$

will take the state from (t_0, x_{P0}) to (t_f, x_{Pf}) while not exceeding the peak acceleration limit a_{\max} , if and only if

$$t_{\text{TOC}}(\tilde{x}(t_0)) \leq t_f - t_0. \quad (7)$$

where $\tilde{x}(t_0) = x_P(t_0) - x_R(t_0)$.

Remark: Note that here evaluating $x_R(t_0) = e^{A(t_0 - t_f)} x_{Pf}$ is merely setting

$$x_R(t_0) = \begin{bmatrix} 1 & (t_0 - t_f) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_f \\ v_f \end{bmatrix} = \begin{bmatrix} s_f - v_f(t_f - t_0) \\ v_f \end{bmatrix}.$$

In other words, we simply extrapolate back the desired output trajectory, start the reference generator on that desired trajectory, and then apply the control that will move the system from its input trajectory to the desired one in minimal time, see Fig.5(top). ■

Thus if we use (6) for $f(\cdot)$ in the 2DOF architecture of Fig.3, the resulting controller satisfies almost all of our desired practical criteria above, except for the last two, which we will address later. Note, in particular, the exact end-to-end state transfer, the compactness, low memory and computational burden, and the ease with which a_{\max} -feasibility under the above control law of a given point to point state transfer can be determined using the t_{TOC} formula. Furthermore, the messaging bandwidth requirements are also minimal: once feasibility is checked, the only information that must be sent to the controllers to execute the end-to-end transfer is $\tilde{x}(t_0) = x_P(t_0) - x_R(t_0)$.

Fig.4 shows the feasible input and output speeds for a module of fixed length and a given a_{\max} . Pairs that are outside the boundaries are infeasible. Again, the formulas (5) and (7) are easy to check in a low cost processor.

We now refer back to Fig.5, and confirm that the last two specs were indeed not met. First, the control (acceleration)

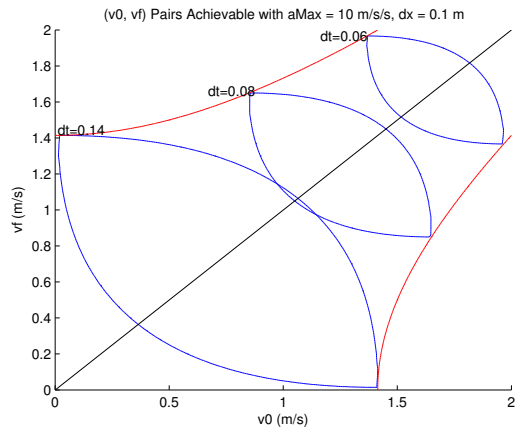


Fig. 4. As we sweep Δt from 0 to 0.15, each value produces a “lemon” shape region, corresponding to the set of (v_0, v_f) pairs that are a_{\max} feasible with the control law (6) for that choice of Δt and the fixed Δx and a_{\max} .

profile does have some large swings as it switches from $-a_{\max}$ to a_{\max} at various points in time. This can cause significant wear and tear on mechanical components as well as induce undesirable vibrations in the system. Second, it also can require moving the paper backwards which could interfere with previous modules’ tasks that have been handed off to downstream modules. This motivates us to explore the method below.

V. INFLECTION FREE CONTROL BASED TRACKING

In this section we will describe another waypoint tracking control method which has smoother profiles, at the expense of some restriction on the feasible set of state transfers. Similar ideas can be found in standard references, eg: [8]. However, we have not seen elsewhere the closed loop implementation via time optimal control, presented here. The resulting controller contains switching not only due to the time optimal control law, but also due to deliberate switching of the state of the reference generator. It can be viewed as a practical example of a simple hybrid systems switching controller.

Specifically, in some applications, it is known ahead of time, or by design, that the overall function of a certain module is either to slow down or to speed up. In this case, one can design *inflection free* trajectories. These are trajectories where the acceleration does not flip sign and hence are either convex or concave. We will design controllers for a specific subclass of inflection free trajectories.

Assumption: For the rest of this section, we will assume that it is required to keep the controlled objects moving in the forward direction, ie, we will not allow the velocity to reverse.

Definition 1: A *constant acceleration inflection free* (CAIF) trajectory is one which is generated by applying a constant acceleration a_{\max} for some subinterval of time $[t'_0, t'_f] \subseteq [t_0, t_f]$, see Fig.6.

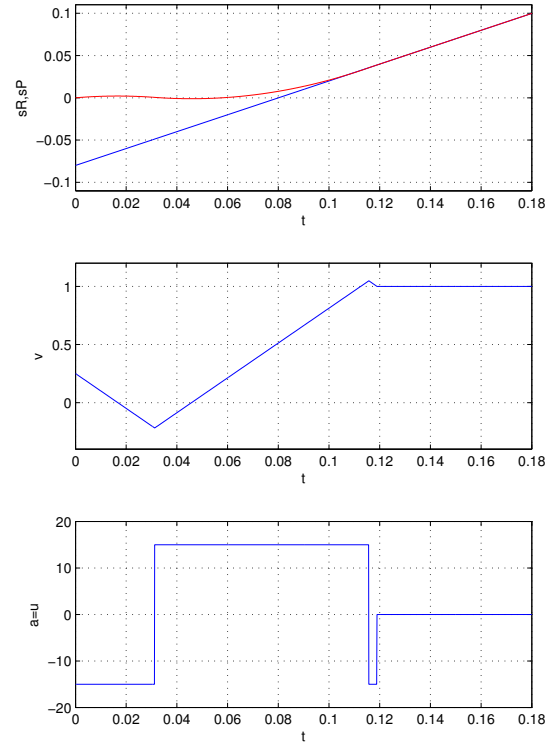


Fig. 5. Reference tracking using control law (6), for waypoints $((0,0,0,0,0,0.25), (0.18,0.10,1.0))$. (Top) position plant (red) and reference (blue). (Mid) plant speed. (Bot) acceleration control.

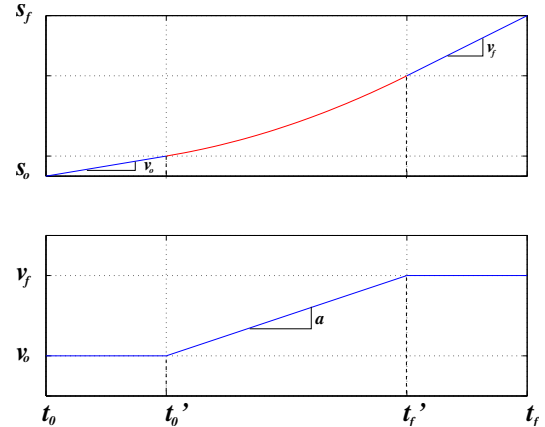


Fig. 6. Constant acceleration inflection free trajectory.

By integrating the area under the curves, one can see that the trajectory will have the following form:

$$s(t) = \begin{cases} s_0 + v_0(t - t_0) & ; t_0 \leq t < t'_0 \\ s_0 + v_0(t - t_0) + \frac{1}{2}a_{\max}(t - t'_0)^2 & ; t'_0 \leq t < t'_f \\ s_0 + v_0(t'_f - t_0) + \frac{1}{2}a_{\max}(t'_f - t'_0)^2 + v_f(t - t'_f) & ; t'_f \leq t < t_f \end{cases} \quad (8)$$

The next question is: what pairs of waypoints are CAIF a_{\max} -feasible? Since (8) gives us the explicit expression for CAIF a_{\max} -feasible points, then we can check the feasibility of a given waypoint pair $((t_0, s_0, v_0), (t_f, s_f, v_f))$ by checking

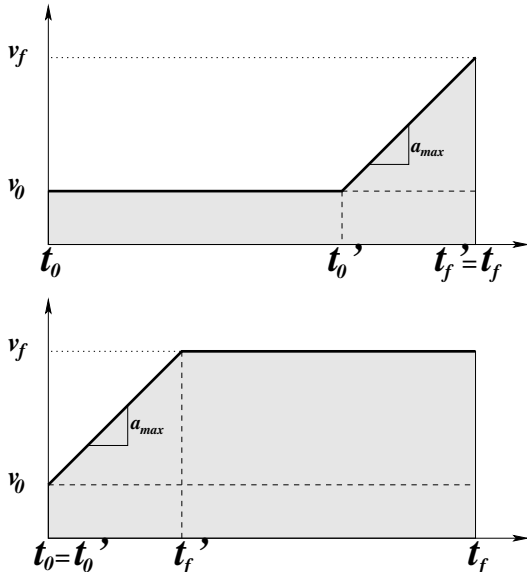


Fig. 7. Extreme cases: given a fixed a_{\max} , the top figure shows the minimal possible change in distance (smallest possible area under the velocity curve for a_{\max}); the bottom figure shows the maximal possible distance change (maximum possible area).

if they are consistent with (8). Specifically, if we evaluate (8) at $t = t_f$, we obtain the expression

$$s_f = s_0 + v_0(t'_f - t_0) + \frac{1}{2}a_{\max}(t'_f - t'_0)^2 + v_f(t_f - t'_f).$$

Using this together with the observation (see Fig.6)

$$a_{\max} = \frac{v_f - v_0}{t'_f - t'_0} \Rightarrow t'_f = t'_0 + \frac{v_f - v_0}{a_{\max}} \quad (9)$$

we can solve for the switching times as functions of $((t_0, s_0, v_0), (t_f, s_f, v_f))$:

$$\begin{aligned} t'_0 &= \left(\frac{1}{v_f - v_0} \right) \left\{ v_f t_f - v_0 t_0 - (s_f - s_0) - \frac{1}{2a_{\max}}(v_f - v_0)^2 (t_f - t_0) \right\} \\ t'_f &= \left(\frac{1}{v_f - v_0} \right) \left\{ v_f t_f - v_0 t_0 - (s_f - s_0) + \frac{1}{2a_{\max}}(v_f - v_0)^2 (t_f - t_0) \right\}. \end{aligned}$$

Therefore, the waypoint pair $((t_0, s_0, v_0), (t_f, s_f, v_f))$ being feasible and consistent with (8), is equivalent to plugging it into (10) and obtaining valid switching times which satisfy:

$$t_0 \leq t'_0 \quad \text{and} \quad t'_f \leq t_f. \quad (11)$$

Equivalent consistency conditions can be obtained by inspecting the extreme cases in Fig.7. Noting that the distance covered is the area under those curves, one obtains the following conditions, which turn out to be exactly equivalent to (10) and (11):

$$\begin{aligned} v_0 \Delta t + \delta &\leq \Delta s \leq v_f \Delta t - \delta & ; v_0 \leq v_f \\ v_f \Delta t + \delta &\leq \Delta s \leq v_0 \Delta t - \delta & ; v_0 > v_f \end{aligned} \quad (12)$$

where $\Delta v = v_f - v_0$, $\Delta t = t_f - t_0$, $\Delta s = s_f - s_0$, and $\delta = \frac{\Delta v^2}{2a_{\max}}$.

Fig.8 shows the feasible input and output speeds for a module of fixed length and a given a_{\max} . Pairs that are outside the boundaries are infeasible. Again, the formulas (11) and (12) are easy to check in a low cost processor.

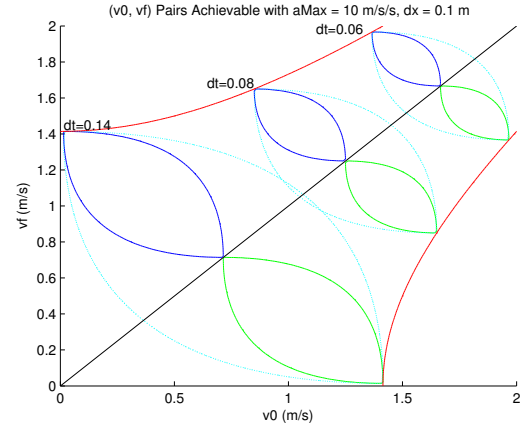


Fig. 8. As we sweep Δt from 0 to 0.15, each value produces a pair of "lemon" shaped regions, corresponding to the set of (v_0, v_f) pairs that are a_{\max} CAIF feasible for that choice of Δt and the fixed Δs and a_{\max} . Also shown for comparison in the cyan dotted lines are the regions of the control law in the previous section.

Furthermore, if we ignore the waypoint timing constraint and simply ask: what pairs $((s_0, v_0), (s_f, v_f))$ are CAIF a_{\max} -feasible for some $\Delta t = t_f - t_0$, one obtains the condition:

$$\frac{1}{2a_{\max}}(v_f^2 - v_0^2) \leq (x_f - x_0)$$

which is shown as the solid boundary lines in Fig.8. This follows from (9) and Fig.7 & Fig.6, which show that any CAIF curve must contain at least the trapezoidal area (distance) between t'_0 and t'_f , which is $\frac{1}{2a_{\max}}(v_f^2 - v_0^2)$.

Proposition 4: Given a_{\max} , the waypoint pair $((t_0, s_0, v_0), (t_f, s_f, v_f))$ is CAIF a_{\max} -feasible if and only if (11) or (12) hold. Then the double integrator system can be made to track the CAIF trajectory using the 2DOF architecture with the time optimal control law by setting $x_R(t_0) = x_P(t_0) = (s_0, v_0)$ and applying the control law $u(\tilde{x}) = f_{\text{TOC}}(\tilde{x})$; then at time $t = t'_0$ given by (10), the reference generator state is switched to track the desired output trajectory, $x_R(t'_0) = e^{A(t'_0 - t_0)} x_f = (s_f - v_f(t_f - t'_0), v_f)$, while continuing to apply $u(\tilde{x}) = f_{\text{TOC}}(\tilde{x})$ all along.

Proof: By construction, the a_{\max} CAIF curve is the result of applying a_{\max} to the double integrator on the interval $[t'_0, t'_f]$. Since the TOC control will attempt to drive \tilde{x} to zero in minimum time, it must also apply a_{\max} for that same duration - it cannot do anything else to reduce the error any faster. ■

Thus if we use (6) for $f(\cdot)$ in the 2DOF architecture of Fig.3, with the reference generator switching as described above, the resulting controller now satisfies all of our desired practical criteria above, see Fig.9. Of course the satisfaction of all of the constraints comes at a cost: the inflection free property of the trajectories reduces the set feasible end-to-end transfers, see Fig.8; the reference switching adds (minimal) complexity to the controller; and there is a small additional messaging bandwidth cost: once feasibility has been established, instead of sending \tilde{x} at t_0 , we must now send $x_R(t'_0)$ and also include t'_0 . Nevertheless, we have found these tradeoffs quite acceptable, in order to achieve the specs

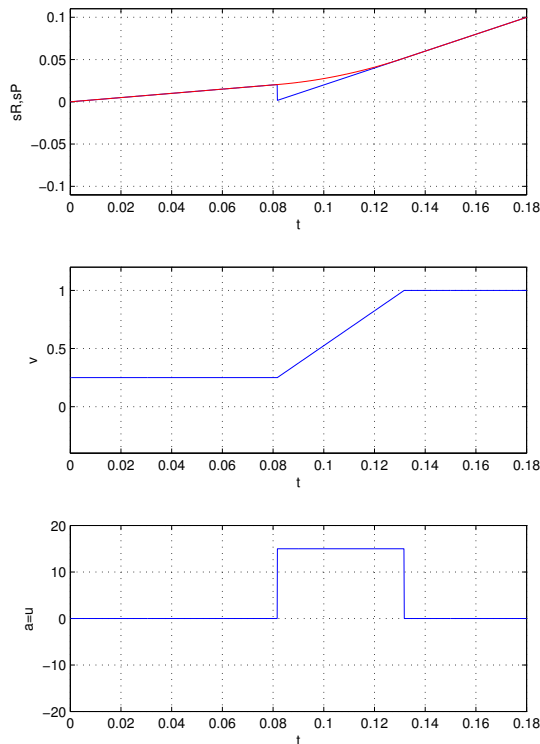


Fig. 9. Tracking using the CAIF control of the same waypoints as in Fig.5. (Top) position plant (red) and reference (blue). (Mid) plant speed. (Bot) acceleration control.

of smoother trajectories and no backwards motion.

Remark: Note that in the absence of noise, the optimal control will be zero on the initial interval $[t_0, t'_0]$ since both the reference and the plant have the same dynamics and initial conditions, and hence the tracking state error \tilde{x} will be zero. The control will be nonzero on the interval $[t'_0, t'_f]$, during which the tracking error will be reduced to zero. Then again after t'_f , the control will again be zero. See Fig.9. Of course, the benefit of having the closed loop control implementation is that, in practice, the system will always be subject to noise and disturbances, and thus there is the opportunity for the control to correct for these perturbations.

Remark: Our presentation has been in continuous time; however, in practice, we have implemented our systems in discrete time. There are various subtleties in approximating continuous time optimal control with discrete time-optimal control. However there are solutions too. We refer the reader to [9] for a discussion of these issues. We have found the *proximate time optimal servo* (PTOS) described there to be quite satisfactory in addressing the main issues in discrete time implementation.

Remark: Note that we have derived the conditions for switching in terms of times t'_0 and t'_f . One could just as well compute corresponding transition conditions in terms of the state x_P . It would be interesting to compare the hybrid robustness properties of both approaches, along the lines of [15].

VI. CONCLUSION

In conclusion we have presented two classical time optimal control based methods for waypoint tracking for double integrator systems that give rise to hybrid switching dynamics. Both are based on using a finite time stabilizing control law in a two-degree-of-freedom topology with a reference generator in conjunction with the plant model. We have found the constant acceleration inflection free method to be quite effective in our practical applications, especially when implemented using the switched time optimal scheme. Other finite time controllers, such as sliding mode controllers, can also be used to construct similar hybrid dynamics.

Simplicity was one of our main design objectives and challenges, to keep costs to a minimum, and to minimize complexity of system integration with the discrete planner. In addition to tracking the waypoints from the planner, our controllers had nine other practical criteria to meet, including fast and exact end-to-end state transfer, actuator saturation, disturbance rejection, low memory and cpu burden, trajectory smoothness and monotonicity. While there are some very sophisticated control design methods available that could handle such constraints, the two classical time optimal control based methods presented here are able to meet most, if not all, the desired constraints, with quite minimal hardware, software and communications burden.

REFERENCES

- [1] M. Athans and P. Falb. *Optimal Control*. McGraw Hill, 1966.
- [2] A. Bemporad, F. Borrelli, and M. Morari. Model predictive control based on linear programming - the explicit solution. *IEEE Trans. Aut. Contr.*, 2002.
- [3] J. Bobrow, S. Dubowsky, and J. Gibson. Time-optimal control of robotic manipulators along specified paths. *Int. J. Robotic Research*, 1985.
- [4] F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*. Springer, 2004.
- [5] A. Bryson and Y.C. Ho. *Applied Optimal Control: Optimization, Estimation and Control*. Taylor & Francis, 1975.
- [6] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun. *Principles of Robot Motion*. MIT Press, 2005.
- [7] R. Cogill and H. Hindi. Optimal routing and scheduling in flexible manufacturing systems using integer programming. In *IEEE Conf. on Dec. & Contr.*, December 2007.
- [8] J. Craig. *Introduction to Robotics: Mechanics and Control*. Prentice Hall, 2004.
- [9] G. Franklin, D. Powell, and M. Workman. *Digital Control of Dynamic Systems*. Addison Wesley, 1998.
- [10] A. Hassibi, S. Boyd, and J. How. A class of Lyapunov functionals for analyzing hybrid dynamical systems. In *Proc. American Control Conference*, 1999.
- [11] H. Hindi and W. Ruml. Network flow modeling for flexible manufacturing systems with re-entrant lines. In *IEEE Conf. on Dec. & Contr.*, December 2006.
- [12] M. Johansson. *Piecewise Linear Control Systems*. Springer, 2003.
- [13] M. Johansson and A. Rantzer. Computation of piecewise quadratic Lyapunov functions for hybrid systems. In *Proc. European Control Conference*, 1997.
- [14] F. Lewis and V. Syrmos. *Optimal Control*. Wiley, 1995.
- [15] R. Sanfelice, R. Goebel, and A. Teel. Invariance principles for hybrid systems with connections to detectability and asymptotic stability. *IEEE Trans. Aut. Contr.*, 2007.
- [16] W.M. Wonham. *Linear Multivariable Control: A geometric Approach*. Springer-Verlag, 1985.