

Control of a Gantry Crane: A Reach Control Approach

Marijan Vukosavljev and Mireille E. Broucke

Abstract— We investigate synthesis of piecewise affine feedback controllers to achieve an aggressive maneuver for a gantry crane operating in a cluttered environment. The methodology is based on formulating the problem as a reach control problem on a polytopic state space. The boundaries of this polytope arise from the safety constraints imposed by the environment. We show that these theoretical tools can be merged with an iterative control synthesis method to obtain an aggressive, albeit safe and robust, maneuver without the need for a predefined open-loop trajectory.

I. INTRODUCTION

This paper proposes a novel framework for heuristic-based iterative synthesis of feedback controllers on a reduced dimensional system for aggressive maneuvering of mechanical systems. It is demonstrated via a benchmark example, a gantry crane system, that by using basic intuition about the physics of the system, a feedback controller can be iteratively constructed to achieve an aggressive collision avoidance maneuver. The significance of the method is the ability to produce a feedback controller, rather than an open-loop control, with the concomitant benefits of robustness and operation over a wider range of initial conditions. In contrast to most literature, we do not compute or learn an open-loop reference trajectory [4], [8], [10], [16], [17]. Our controller has as its foremost objective to avoid safety violations. That the resulting maneuver is “aggressive” is not because of the choice of the open-loop trajectory, but because the safety requirements push the system to react decisively to obstacles in its environment.

A significant effort in recent years has been devoted to aggressive maneuvering, particularly of aerial vehicles. The authors of [8], [12], [13], [16] attempt precise trajectory tracking, flips, and balancing inverted pendulums using quadcopters. In [8], [16] a predefined trajectory is tracked, achieved via a learning strategy employed over multiple experimental runs to modify the input to non-causally compensate for disturbances, whereas in [12], [13] a parameterized open-loop control input is determined. Similarities can be drawn to our approach in the sense that intuition is used to subdivide the maneuver into phases, but in our approach we use event-based switching criteria rather than time-based. Furthermore, in our approach we completely avoid using predefined trajectories and instead use an iterative learning method to determine only the feedback controller parameters.

The authors are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: mario.vukosavljev@mail.utoronto.ca, broucke@control.utoronto.ca). Supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

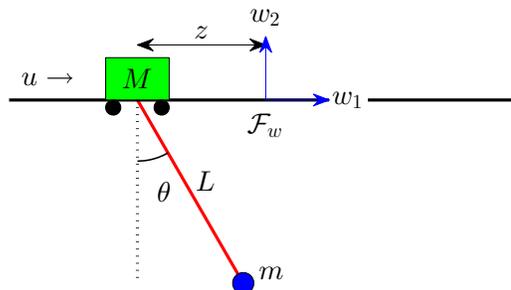


Fig. 1. The crane model.

In [10] the focus is on a two stage process to first compute a geometrically valid trajectory, and then a refined dynamically feasible trajectory for maneuvering UCAV’s through complex 3D city-like environments. Our example imposes tighter geometrical constraints on the states compared to city navigation in order to highlight our approach better. A related example is found in [14], where a rotary pendulum must perform a rotational motion to bypass an obstacle. However, in [10], [14] it was necessary to determine a predefined safe path to track, which is often difficult to compute, whereas in our approach we do not require a predefined path.

To solve the crane-obstacle problem, we formulate it as a *Reach Control Problem* (RCP) on a polytope [1]–[3], [6], [7], [15]. The reach control problem is for the trajectories of a dynamical system to reach a predefined facet of a polytope in finite time without first exiting the polytope. The boundaries of this polytope are determined by safety constraints imposed on the state variables in the transient mode. The control of transient behavior and the guidance of the system states to a desirable range of values are naturally accommodated in the RCP framework. However, in this paper we relax the requirement to solve RCP exactly, for reasons to be explained below. This relaxation of the RCP problem statement is new and evocative for the further development of the theory, as our future work will show.

Our method is comparable to the approaches in [5] and [9] in the sense that the problem is formulated as a reach control problem. In [9], given a polytope, a set of predicates, and a linear temporal logic (LTL) formula, a fully automated framework is provided to construct affine feedbacks that lead trajectories from initial states to target states in order to satisfy the LTL formula. In [5] a nonlinear system is approximated by an affine system on each simplex of a triangulation of the polytope. Similar to [9], an algorithm is given to construct affine feedbacks following the method of [6], resulting in a sequence of simplices that corresponds

to trajectories executing the desired behaviour.

The authors of [5] and [9] acknowledge that their methods are conservative, but clever subdivisions of the polytope were not a focus of either [5] or [9], nor our approach. Moreover, these methods require designs on the full dimensional space; however, both only illustrated convincing results on 2D examples. In contrast, we aim to solve a 4D problem with an aggressive control specification. While the design methods in [5] and [9] are applicable, solving our problem with those methods would result in a very complex design, with possibly tens of thousands of different affine feedbacks as suggested in [9]. As such, we explore a novel approach where we do not perform the design on the full dimensional space, as required by the existing theory of RCP, but instead work in a more wieldy 2D *output space* to reduce the controller complexity.

While the immediate contribution of the paper is to apply the reach control methodology in output space to achieve an aggressive maneuver, the ultimate contribution is to stimulate future research on RCP in two directions. First, the introduction of more formal iterative learning methods in RCP, as related to the literature cited above. Second, to formulate RCP on lower dimensional spaces, such as the output space. *The fact that we used heuristics and iterative learning is a consequence of the lack of theory to support a rigorous design on a lower dimensional space.* For simplicity of exposition of these ideas, we restricted our iterative learning method to more-or-less trial and error. Despite this somewhat crude and unchartered approach, we succeed in illustrating the utility of the reach control concept in that we can perform interesting aggressive maneuvers without the need for tracking a predefined trajectory, as in the majority of the literature.

II. MODEL OF GANTRY CRANE

The model for the gantry crane consists of a trolley on a horizontal track carrying a load by a suspension chord. Let z be the lateral displacement of the trolley with respect to a fixed world frame (w_1, w_2) along the track. Let θ be the angle of the load from the downward vertical position in the counterclockwise sense, and let the control input u be the force applied to the trolley. The trolley and load have mass M and m , respectively. The suspension chord has a fixed length L and is assumed to be massless and rigid. See Figure 1. Frictional losses are neglected for simplicity of modeling. Note that the suspension chord is restricted to a fixed length to make the control problem of obstacle avoidance more challenging. Let $x_1 = z$, $x_2 = \dot{z}$, $x_3 = \theta$, and $x_4 = \dot{\theta}$ and define the state vector $x = (x_1, x_2, x_3, x_4)$. Then the state model is:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{mLx_4^2 \sin(x_3) + \frac{1}{2}mg \sin(2x_3) + u}{M + m \sin^2(x_3)} \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -\frac{(M + m)g \sin(x_3) + \frac{1}{2}mLx_4^2 \sin(2x_3) + u \cos(x_3)}{L(M + m \sin^2(x_3))}. \end{aligned} \quad (1)$$

Next we linearize the state model about the equilibrium $x = 0$, $u = 0$, corresponding to the load hanging down vertically.

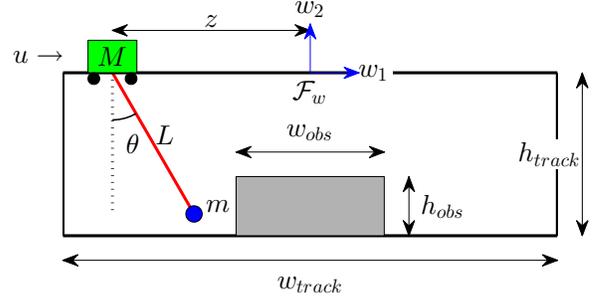


Fig. 2. The crane-obstacle problem

This yields the linearized state space model

$$\dot{x} = Ax + Bu = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{gm}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{g(M+m)}{LM} & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{LM} \end{bmatrix} u. \quad (2)$$

The output is $y = (x_1, x_3)$. For the purposes of simulation, the model parameters are fixed at $m = 0.23$ kg, $M = 1.07$ kg, $L = 0.33$ m, and $g = 9.8$ m/s². These parameters are representative of what could potentially be tested in a laboratory using a cart and pendulum to simulate a gantry crane trolley and load.

III. COMPLEX CONTROL SPECIFICATIONS

In the present work the primary control objective is to transport a payload from one end of a room to another. The environment is cluttered by a large obstacle that stands in the way of the start and end points. The height of the obstacle is such that the payload would collide with it when hanging vertically, but pass over it when rotated to a sufficiently large angle. While the obstacle may be clearable by hoisting the payload, we attempt this without hoisting. Doing so demands for a rather aggressive crane maneuver and we aim to show that safety, interpreted strictly as avoiding all collisions, can be maintained using our control framework. The desired behaviour of the crane can be broken down into three conceptual stages:

- 1) Move the crane to a starting position and begin motion to start clearing the obstacle.
- 2) Fully clear the obstacle while remaining within operational limits.
- 3) Bring the crane to a full stop at the desired end point while remaining within operational limits.

Figure 2 shows the crane model embedded within the workspace. For purposes of simulation, we selected $w_{track} = 2.2$ m, $w_{obs} = 0.3$ m, $h_{track} = L = 0.33$ m, and $h_{obs} = \frac{1}{2}h_{track} = 0.165$ m. The crane arm may not hit the wall boundaries of the room, the ceiling or railing of the trolley, and the obstacle.

The control specifications can be formalized in terms of the states of the system. Let d and θ^* have the meaning shown in Figure 3. By the geometry it is seen that $d = \sqrt{L^2 - (h_{track} - h_{obs})^2}$ and $\theta^* = \cos^{-1} \left(\frac{h_{track} - h_{obs}}{L} \right)$. The



Fig. 3. Possible collisions of crane with obstacle

complex control specifications for the crane-obstacle problem are:

Safety Specifications:

- (S1) $|x_1| \leq \frac{1}{2}w_{track}$
- (S2) $|x_3| \leq \frac{1}{2}\pi$
- (S3) $|\frac{\pi}{2L}x_1 + x_3| \leq \frac{\pi}{4L}w_{track}$
- (S4) If $|x_1| \leq (\frac{1}{2}w_{obs} + d)$ then $|x_3| \geq \theta^*$ or $|x_1 + \frac{d}{\theta^*}x_3| \geq \frac{1}{2}w_{obs}$

Desired Temporal Sequence: The load is transported from the left wall to the right wall.

The desired temporal sequence is a qualitative description of the sequence of events which must be enacted by the closed-loop system, and it can be further formalized using discrete event system theory. (S1) characterizes that the trolley position may not exceed the limits of the track. (S2) imposes that the pendulum arm may not swing too high to hit the track. (S3) describes how far the pendulum arm may swing as the trolley approaches the limits of the track on either side. (S4) imposes that as the trolley position gets within a vicinity of the obstacle, the pendulum arm must be within a particular range of angles to avoid collision with the obstacle on either side. The safety constraints to avoid collisions are, strictly speaking, nonlinear constraints in (x_1, x_3) . Here the safe region of operation is approximated by linear constraints, yielding (S3)-(S4); details are omitted.

IV. METHODOLOGY

The safety specifications impose that the pair (x_1, x_3) remain inside a non-convex, doughnut-shaped polytopic region \mathcal{P} , as shown in Figure 4. The hole in \mathcal{P} corresponds to safety constraints induced by the obstacle, whereas the outer boundaries correspond to safety constraints induced by the walls. Note that the obstacle is of a significant size so as to push the limitations of the system and thereby illustrate the capabilities of the proposed control method.

The set \mathcal{P} determines the output space of the problem. We partition \mathcal{P} into a set of regions where controllers can be specified. Ideally these regions would be topologically identical so that a standardized control synthesis method could be employed. Since \mathcal{P} is a non-convex polytope we employ a triangulation of \mathcal{P} into simplices.

An n -dimensional *simplex* $\mathcal{S} := \text{co}\{v_0, \dots, v_n\}$ is the convex hull of $n + 1$ affinely independent points in \mathbb{R}^n . Informally, it is the higher-dimensional generalization of a triangle. A *facet* of a simplex is a boundary face of dimension $n - 1$. A *triangulation* is a partition of a set $\mathcal{P} \subset \mathbb{R}^n$ into p simplices and is denoted as $\mathbb{T} = \{\mathcal{S}_1, \dots, \mathcal{S}_p\}$ [11]. Then \mathbb{T} satisfies the properties:

- (i) $\mathbb{T} = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_p$ and

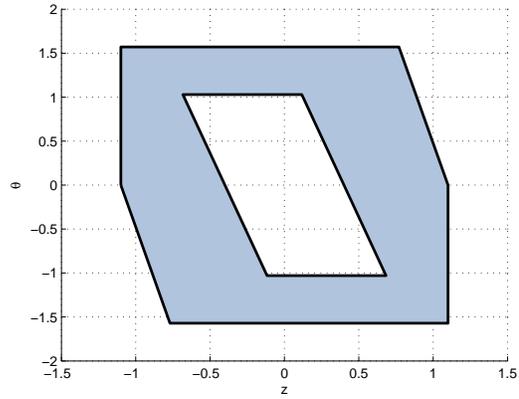


Fig. 4. The output space determined by the safety specifications.

- (ii) $\mathcal{S}_i \cap \mathcal{S}_j, i \neq j$, is a lower dimensional simplex of both \mathcal{S}_i and \mathcal{S}_j or the empty set $\forall i, j \in \{1, \dots, p\}$.

Once a triangulation of \mathcal{P} has been specified, the next step of the design is to identify a sequence of simplices to be visited by trajectories in order to "play out" the desired temporal sequence. Since the left and right vertical boundaries of \mathcal{P} correspond to the left and right walls, this sequence of simplices should start with a simplex containing the point $(x_1^0, x_3^0) = (-\frac{w_{room}}{2}, 0)$ and finish with a simplex containing the point $(x_1^f, x_3^f) = (\frac{w_{room}}{2}, 0)$.

Using the sequence of simplices, an *exit facet* for each simplex is designated. The trajectories starting in the given simplex may only exit the simplex through the exit facet. Finally, controllers are designed for each simplex based on the *reach control problem* (RCP). The reach control problem formulation, its theoretical developments, and conditions for solvability are discussed in the literature [2], [3], [6], [7], [15]. The main purpose of RCP is to guide trajectories of a dynamical system through a specific region of the state space defined by safety and performance requirements. Ultimately, the trajectories should reach a target set of states in finite time without violating the boundaries of the specified region.

The following algorithm summarizes the basic design steps assuming no iteration for learning the controllers. Each step of the algorithm will be further elaborated in Section V, where also the iterative learning method associated with steps 4-5 of the algorithm will be explained.

Algorithm 1: Consider system (1) and a control specification consisting of safety specifications and a desired temporal sequence.

- 1) Express the safety specifications as affine inequalities in terms of the state variables of (1). Denote the region of the state space defined by these inequalities by \mathcal{P} .
- 2) Triangulate \mathcal{P} using the vertices of \mathcal{P} . (A refinement of the initial triangulation using more vertices other than those of \mathcal{P} may be required).
- 3) Determine a sequence of simplices $\mathcal{S}_1 - \mathcal{S}_p$ such that the desired temporal sequence can be satisfied by the closed-loop trajectory. (This may require iterating on the triangulation to ensure solvability of step 4 of the

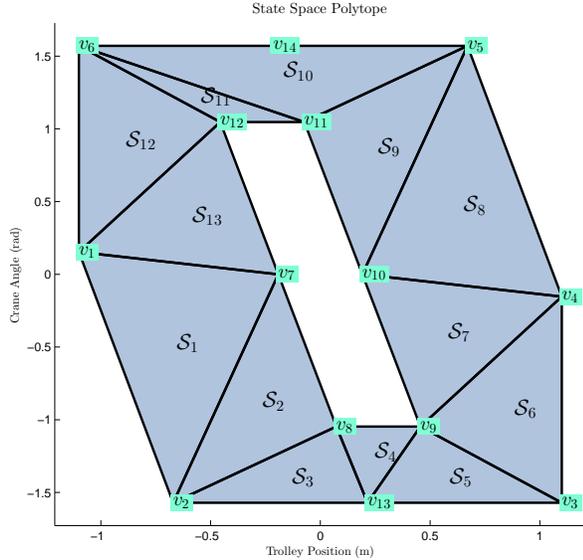


Fig. 5. Triangulation of \mathcal{P} .

algorithm). Designate an exit facet for each simplex in the sequence.

- 4) For each simplex \mathcal{S}_i , $i = 1, \dots, p - 1$, compute an affine feedback $u = Ky + g$ by formulating a reach control problem on \mathcal{S}_i in the output space. This involves selecting control values u_0, u_1, u_2 at the vertices v_0, v_1, v_2 , respectively, and solving for K and g using the formula

$$\begin{bmatrix} -v_0^T & -1 \\ -v_1^T & -1 \\ -v_2^T & -1 \end{bmatrix} \begin{bmatrix} K^T \\ g^T \end{bmatrix} = \begin{bmatrix} u_0^T \\ u_1^T \\ u_2^T \end{bmatrix}. \quad (3)$$

- 5) Design a pole placement controller $u = K^{pp}x + g^{pp}$ for the linearized model (2) to be activated upon reaching a target simplex \mathcal{S}_q in order to stabilize to a chosen rest position x^f .

The resulting control law is a piecewise affine feedback with switching between controllers occurring at the boundaries between contiguous simplices.

V. ITERATIVE LEARNING OF AFFINE CONTROLLERS

We now elaborate each step of the algorithm and also discuss an iterative method for learning the affine controllers for each simplex.

The first step of the algorithm to express the safety specifications as affine inequalities in the states has been completed in Section III. For the second step, in the crane problem \mathcal{P} is defined in the output space (x_1, x_3) ; rather than working with 4D simplices we only require 2D simplices. Figure 5 shows a triangulation consisting of 13 2D simplices. This triangulation was naively generated using existing polytope vertices. After experimentation in simulation it was found that adding an additional vertex at the bottom of \mathcal{S}_4 improved the ability to design the controllers. This will be further described below.

For the third step of the algorithm we must determine a sequence of simplices of the triangulation to be visited by closed-loop trajectories based on the desired temporal sequence. Based on our triangulation, the sequence must start with \mathcal{S}_1 and end with \mathcal{S}_8 . It is clear from Figure 4 there are two ways to get around the hole: either $\mathcal{S}_1 \rightarrow \mathcal{S}_2 \rightarrow \mathcal{S}_3 \rightarrow \mathcal{S}_4 \dots$ or $\mathcal{S}_1 \rightarrow \mathcal{S}_{13} \rightarrow \mathcal{S}_{12} \rightarrow \mathcal{S}_{11} \dots$. According to the physics of the system, when the trolley moves from left to right, it is natural for the pendulum arm to swing backwards, attaining negative angles corresponding to simplices $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$, etc. Therefore, we choose the sequence $\mathcal{S}_1 \rightarrow \mathcal{S}_2 \rightarrow \mathcal{S}_3 \rightarrow \mathcal{S}_4 \dots$. This choice also fixes the exit facet for each simplex in the sequence.

In the fourth step of the algorithm it is required to choose an RCP controller for each simplex. In the present problem we do not follow the usual approach to select control values at the vertices of each simplex to achieve the invariance conditions (for details, see the RCP literature mentioned earlier). This is because the simplices are defined in the output space and satisfying the associated two dimensional invariance conditions would not guarantee that closed-loop trajectories exit only through the designated exit facets. Instead we use an iterative procedure to choose the three vertex control values u_0, u_1 , and $u_2 \in \mathbb{R}$ for each simplex using only basic intuition about the physics. This method is now elaborated.

The main concept of the affine control design is to work sequentially from the beginning of the temporal sequence, ensuring that over each simplex the closed-loop trajectory exits through the desired exit facet. If a collision occurs in the simulation, we redesign the control values at the latest simplex and resimulate to verify the change. Typically only one vertex control value should be modified at a time in small increments. If difficulty is encountered clearing a particular simplex, earlier simplex control values must be reselected in such a way to give better leverage in later simplices. Related approaches using iterative learning have been based on actual experimental results [16] in order to correct for disturbances; in contrast, we iterate based on simulation results using the full nonlinear model (1) to find the gains in our controllers.

While it is possible to automate this iterative approach, there is a simple intuitive rule that can be employed when selecting control values manually for the present problem. First we recall that control input u is the lateral force applied to the cart. The control value at each vertex is the force applied at that (x_1, x_3) coordinate. Using the feedback (3), the control value at an arbitrary point in each simplex is an affine combination of the control values at the vertices. Hence positive values of u push the cart forward, making the pendulum arm swing backwards and vice versa for negative values. Also large values of u result in a strong push, causing quick gains in speed and changes in state, while the opposite occurs for small values. Therefore, a feedback on each simplex is generated by characterizing physically how much force is required in that configuration to reach the next configuration.

We remark that the above procedure is more intuitive

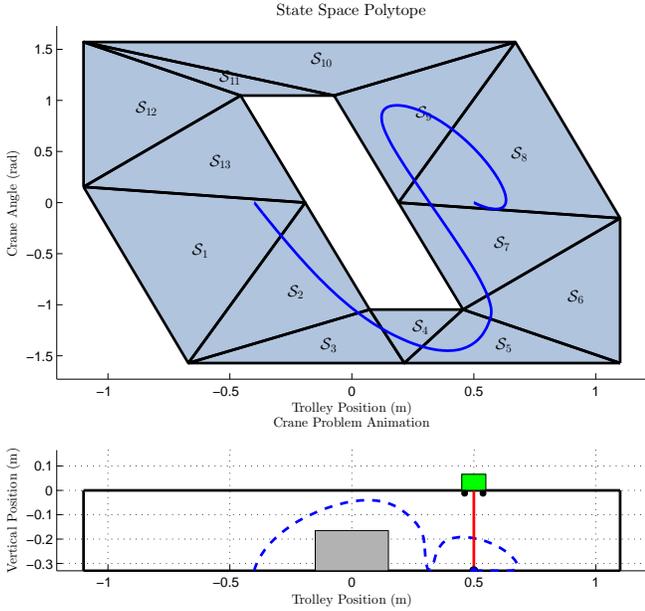


Fig. 6. Nonlinear simulation result for a high, moderately wide obstacle.

than selecting closed-loop poles to generate linear state feedbacks. In any case, the poles are related to settling time and overshoot, which are irrelevant to meeting our complex specifications. In fact, to achieve this maneuver we are making the system temporarily unstable to push it over the obstacle, which is why the maneuver is inherently aggressive. Heuristics for the control values are described next.

We keep in mind the three conceptual stages of the crane behavior. For the first few simplices ($\mathcal{S}_1 - \mathcal{S}_2$) a good start is to push forward to gain a sufficiently large angle to be able to clear the obstacle. In the next simplices ($\mathcal{S}_3 - \mathcal{S}_5$) a good start is to keep the force constant to maintain the angle. However, it is possibly better to “juggle” the force applied forward and backward in each sequential simplex to start to reduce forward speed. This is the reason why this region is triangulated in more simplices in order to provide more opportunities to exercise the rapid changes in force. In the remaining simplices ($\mathcal{S}_6 - \mathcal{S}_8$) the obstacle has been cleared and so a good start is to push backward very hard to slow down the system. Doing so causes the closed-loop trajectory to brush closely against the right side of the obstacle region in \mathcal{P} , which then gives more forward room for the pendulum arm to swing back safely when the stabilizing pole placement controller is activated. From these starting points the values can be iterated gradually in the manner described above.

A useful simplification for the initial guess of vertex control values is to use the same control value at all the vertices of a given simplex. From (3) it can be seen that this results in $K = 0$ and $g = u_0 = u_1 = u_2$. If this does not work, then the control values can be modified one at a time.

Finally, for step 5 of the algorithm a pole placement controller based on the linearized model is design (despite the fact that the nonlinear model operates far from equilib-

rium). This controller is activated permanently after the first instance that the state is detected in \mathcal{S}_9 . Thus, it is used to terminate the design because RCP controllers cannot stabilize the system to rest. Upon reaching \mathcal{S}_9 and just prior to activating the pole placement controller, the RCP controllers have already performed their duty in transporting the crane over the obstacle. If the heuristics described earlier were followed over the conceptual stages of the maneuver, then the pole placement controller should have little trouble keeping the closed-loop trajectory within \mathcal{P} as the crane comes to rest. As the stopping occurs, the states backtrack through the simplices to reach the final state, x^f , which is situated on the line of equilibrium states, $x_3 = 0$. The decision to initiate the pole placement controller upon reaching \mathcal{S}_9 was based on obtaining a feasible closed-loop trajectory; in fact we could have tried any of the simplices $\mathcal{S}_6 - \mathcal{S}_{10}$ as the switching criterion.

It can be verified that the linearized system (A, B) in (2) is controllable. Hence the closed-loop poles can be set arbitrarily, which gives K . The affine term can be selected as $g = -Kx^f$. Poles that are too large can cause an aggressive response, resulting in the pendulum arm to hit the track. Poles that are too small do not stop the cart early enough and the pendulum hits the right wall.

VI. SIMULATION RESULTS

In this section we present the results for the crane problem. After applying the method described above, the resulting piecewise affine feedback is

$$u = \begin{cases} 25, & x \in \mathcal{S}_1 \\ 5, & x \in \mathcal{S}_2 \\ [158.15 & 0.00 & -167.26 & 0.00]x - 196.70, & x \in \mathcal{S}_3 \\ -10, & x \in \mathcal{S}_4 \\ -11, & x \in \mathcal{S}_5 \\ -80, & x \in \mathcal{S}_6 \\ -100, & x \in \mathcal{S}_7 \\ 150, & x \in \mathcal{S}_8 \\ [-60.71 & -38.52 & 56.08 & -3.50]x + 30.35, & x \in \mathcal{S}_9. \end{cases}$$

The last controller on \mathcal{S}_9 is the pole placement controller, and the control is not switched again after reaching \mathcal{S}_9 .

We simulate the closed-loop response using the nonlinear model (1) and the results are shown in Figures 6 and 7. The top subplot of Figure 6 shows the triangulation of \mathcal{P} and the closed-loop trajectory in the output space (x_1, x_3) : by remaining inside \mathcal{P} the crane has avoided all possible collisions. The bottom subplot of Figure 6 shows a snapshot of an animation where the dotted line represents the motion of the load. Finally, the four subplots of Figure 7 show the states as a function of time. We remark that the control inputs to achieve this maneuver are very large, on the order of 100 N in some instances. If we had imposed actuator constraints, it may have been that the desired maneuver would inherently be too aggressive to perform.

An important consideration in evaluating the usefulness of the proposed method is the robustness of the controller performance to perturbations in the payload mass, m . The effect of varying m is shown in Figure 8. Our simulation results showed at least a $\pm 30\%$ insensitivity to variations

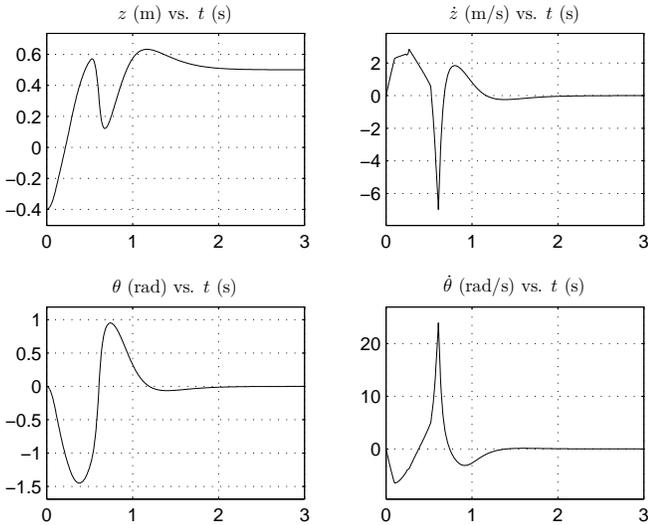


Fig. 7. State trajectories as a function of time for a high, moderately wide obstacle.

in m under the same control law, while greater perturbations would result in collisions. In particular, decreasing m further would result in a collision around \mathcal{S}_5 because of excessive input force, and increasing m further would result in a collision in $\mathcal{S}_8 - \mathcal{S}_{10}$ because of insufficient input force to slow down the crane.

A second consideration is robustness to choice of initial condition. The proposed controller cannot be used for all initial conditions in the state space \mathcal{P} as one would expect of a feedback controller. This is because we have not strictly solved RCP since we work on 2D simplices, not 4D simplices. However, in practical terms the proposed controller can be used for all initial conditions corresponding to the cart being to the left of the obstacle. One can employ a pole placement controller with weak poles to bring the cart to our designed initial condition within the starting simplex.

Acknowledgements. The authors thank Krishnaá Mehta, Hecheng Wang, and Sharaf Mohamed who provided foundations to solve the crane-obstacle problem.

REFERENCES

- [1] G. Ashford and M.E. Broucke. Design of reach controllers on simplices. *IEEE Conference on Decision and Control*. Florence, Italy, Dec. 2013.
- [2] M.E. Broucke. Reach control on simplices by continuous state feedback. *SIAM Journal on Control and Optimization*. vol. 48, issue 5, pp. 3482-3500, February 2010.
- [3] M.E. Broucke and M. Ganness. Reach control on simplices by piecewise affine feedback. *SIAM J. Control and Optimization*. Accepted, August 2014. arXiv:1212.6450 [math.OC]
- [4] V. Gavrillets, E. Frazzoli, B. Mettler, M. Piedmonte, and E. Feron. Aggressive Maneuvering of Small Autonomous Helicopters: A Human-Centered Approach. *International Journal of Robotics Research*. Vol. 20, no. 10, pp. 795-807, October 2001.
- [5] A. Girard and S. Martin. Synthesis for Constrained Nonlinear Systems using Hybridization and Robust Controllers on Simplices. *IEEE Transactions on Automatic Control*. Vol. 57, no. 4, pp. 1046-1051, April 2012.
- [6] L.C.G.J.M. Habets and J.H. van Schuppen. A control problem for affine dynamical systems on a full-dimensional polytope. *Automatica*. no. 40, pp. 21-35, 2004.

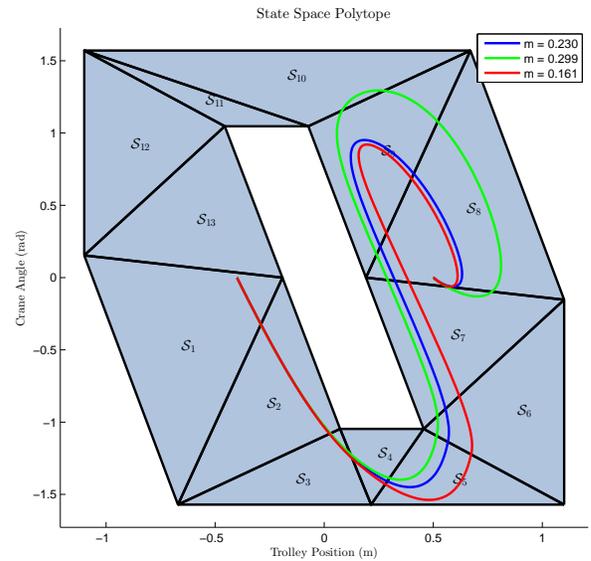


Fig. 8. Robustness of the piecewise affine feedback to $\pm 30\%$ variation of the payload mass.

- [7] L.C.G.J.M. Habets, P.J. Collins, and J.H. van Schuppen. Reachability and control synthesis for piecewise-affine hybrid systems on simplices. *IEEE Trans. Automatic Control*. no. 51, pp. 938-948, 2006.
- [8] M. Hehn and R. D'Andrea. An Iterative Learning Scheme for High Performance, Periodic Quadcopter Trajectories. *European Control Conference*. July 17-19 2013.
- [9] M. Kloetzer and C. Belta. A Fully Automated Framework for Control of Linear Systems from Temporal Logic Specifications. *IEEE Transactions on Automatic Control*. Vol. 53, no. 1, pp. 287-297, February 2008.
- [10] E. Koyuncu, N. K. Ure, and G. Inalhan. Integration of Path/Maneuver Planning in Complex Environments for Agile Maneuvering UCAV's. *Journal of Intelligent and Robotic Systems*. Vol. 57, no. 1, pp. 143-170., January 2010.
- [11] C. W. Lee. Subdivisions and triangulations of polytopes. *Handbook of Discrete and Computational Geometry*. CRC Press Series Discrete Math. Appl., pp. 271-290, 1997.
- [12] S. Lupashin and R. D'Andrea. Adaptive fast open-loop maneuvers for quadcopters. *Autonomous Robots*. Vol. 33, nos. 1-2, pp. 89-102, August 2012.
- [13] S. Lupashin, A. Schoellig, M. Sherback, and R. D'Andrea. A Simple Learning Strategy for High-Speed Quadcopter Multi-Flips. *IEEE International Conference on Robotics and Automation*. May 2010.
- [14] T. Miyoshi, K. Tsuchida, and K. Terashima. Obstacle Avoidance Control for Overhead Crane with Rotary Motion of Load. *SICE Annual Conference in Fukui*, August 2003.
- [15] B. Roszak and M. E. Broucke. Necessary and sufficient conditions for reachability on a simplex. *Automatica*. Vol. 42, no. 11, pp. 1913-1918, November 2006.
- [16] A. Schoellig, F. Mueller, and R. D'Andrea. Optimization-based iterative learning for precise quadcopter trajectory tracking. *Autonomous Robots*. Vol. 33, nos. 1-2, pp. 103-127, August 2012.
- [17] E. A. Wan, A. A. Bogdanov, R. Kiebert, A. Baptista, M. Carlsson, Y. Zhang, and M. Zulauf. Model Predictive Neural Control for Aggressive Helicopter Maneuvers. In *Software-Enabled Control: Information Technology for Dynamical Systems*.