Kinodynamic path switching for robotic manipulators

McGovern, R., & Athanasopoulos, N. (2023). Kinodynamic path switching for robotic manipulators. In H. Ishii, Y. Ebihara, J. Imura, & M. Yamakita (Eds.), *Proceedings of the 22nd IFAC World Congress 2023* (pp. 1597-1602). (IFAC-PapersOnline; Vol. 56, No. 2). Elsevier. https://doi.org/10.1016/j.ifacol.2023.10.1860

Published in:
Proceedings of the 22nd IFAC World Congress 2023

Document Version:
Publisher's PDF, also known as Version of record

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 1597–1602

Kinodynamic path switching for robotic manipulators ⋆

Ryan McGovern and Nikolaos Athanasopoulos

School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Northern Ireland (e-mail: rmcgovern03@qub.ac.uk, n.athanasopoulos@qub.ac.uk).

Abstract: We extend trajectory planning methods for robotic manipulators to the more general and complex path planning problem in both the space of joints and their velocities. We consider a set of prespecified feasible paths, together with their associated, precomputed safe sets in the projected space of the dynamics restricted to the path. We ask a natural question, namely, whether we can state geometric conditions, together with their algebraic equivalents, that allow switching between paths, providing thus greater flexibility, enabling path planning capabilities. We answer affirmatively, and pose conditions whose consistency can be checked by solving a set of algebraic conditions and by computing intersections between safe sets, precisely, reach-avoid sets. Controllers can also be extracted for closed-loop switched paths. Our proof-of-concept analysis suggests that under the developed framework, composition between paths is possible, thus, potentially significantly extending the applicability of the approach and complementing existing planning methods.

Keywords: planning, robotics, safety, reach avoid sets, switching, control.

1. INTRODUCTION

Planning in robotics is particularly important and timely due to the ever growing demand for safer robots interacting with humans in various settings Villani et al. (2018), Robot motion planning falls in the category of planning of systems under differential constraints LaValle (2006). This type of problem is particularly complex and has been studied extensively. In the context of robotic manipulators, a variety of methods from control have been utilised to address planning e.g., Gasparetto and Zanotto (2007), Baressi Šegota et al. (2020) , Perez et al. (2011). In recent years, there has been an increased focus on the application of kinodynamic constraint inclusion in motion planning with a wide range of approaches taken from control to machine learning based methods Caron et al. (2017), Wang et al. (2021). Typically, there are two distinct phases of motion planning, namely, path and trajectory planning. Trajectory planning decides the velocity profile of for a given path, while path planning decides a safe, feasible path in the workspace and joint space of the robot.

Our work is inspired by compositional approaches, where a set of prespecified paths or trajectories can be combined and reused, to produce complex motions, see for example works related to motion primitives Hauser et al. (2008), Cohen et al. (2011) and identification of symmetries Sibai and Mitra (2020). Moreover, in the literature there has been increased focus on reachability analysis for cyber-physical systems Chen and Sankaranarayanan (2022). This is a challenging task due to the nonlinear/hybrid nature of such systems. One method of tackling the reachability is to perform an abstraction of the non linear system, into lower dimensional, less complex systems that describe some behaviours of such a system Alur et al. (2000a).

We use as a starting point of approach our previous work McGovern and Athanasopoulos (2022), McGovern et al. (2022), which explored safety of velocity profiles of prespecified paths. We utilise a two dimensional model induced by projecting the complex Lagrangian dynamics of general N-link robotic manipulators, a method used traditionally in optimal control problems Kang Shin and McKay (1985). In this projected space, an appropriate parameterisation of the family of admissible state feedback controllers allows characterisation of the set of admissible joint velocities and the corresponding control actions that can be applied so a desired path can be traversed without any constraint violation. Roughly, the parameterisation of the controller is a convex function of the state-dependent constraints for each point on the path.

By having a set of such paths and their admissible sets in the associated projected spaces, called reach avoid sets, we formulate conditions on when it is possible to switch between them. The resulting conditions (Proposition 1, Theorem 1) can be grouped in (a) geometric ones, corresponding to existence of intersections between the paths in the joint space, and alignment of the joint velocities on these points, and (b) set-based ones, that relate to the existence of feasible controllers driving the manipulator so that the switched path can be traversed in its entirety. The equivalent algebraic conditions to (a) are easily solved numerically or analytically for simple curves, producing admissible switching intervals, whereas the set conditions (b) can be verified if nonempty intersections between sets that are partially or fully precomputed, exist. This is advantageous as it allows computations to be reused, and hence speed up the robot's ability to perform more complex tasks by refinement of paths.

⋆ R.M. gratefuly acknowledges support from the UK DfE and N.A. from EU 2020-1-UK01-KA203-079283, EPSRC EP/T021942/1 and the UKRI Belfast Maritime Consortium 107138.

10.1016/j.ifacol.2023.10.1860

Ultimately, path switching enables combinations of provably safe motions, allowing for new motions to be achieved without having to form reach avoid sets for each specific path. We report preliminary, however encouraging results, that concern the switching between two generic paths. In the future, we wish to apply our ability to switch between paths to form graphs that may be used as a basis for formal abstractions Alur et al. (2000b), Hussien and Tabuada (2018), Egidio et al. (2022), allowing for complex objectives. Section 2 introduces preliminaries and definitions to discuss our framework. Section 3 introduces the concept of path switching and provides an exact definition of an admissible switched path, before presenting the corresponding conditions. Moreover, we discuss special cases for path switching for some simple paths in the joint space, and also the form of the state feedback controller. Section 4 illustrates our findings via a numerical example, while conclusions are drawn in Section 5.

2. PRELIMINARIES

For a vector $x_{\text{label}} \in \mathbb{R}^N$ with a label in its subscript, we write its i^{th} element as $(x_{\text{label}})_i$. We denote sets, e.g., $\mathcal{S}, \mathcal{R}$, with capital letters in italics. The dynamics of an N degree of freedom robotic manipulator can be described by

$$M_L(q)\ddot{q} + C_L(q,\dot{q})\dot{q} + g_L(q) = \tau, \tag{1}$$

where $q \in \mathbb{R}^N$ are the joint positions. Matrices $M_L(\cdot), C_L(\cdot) \in \mathbb{R}^{N\times N}$ represent linear and rotational mass components respectively, while $g_L(q) \in \mathbb{R}^N$ represents the effect of gravity. Finally the vector $\tau \in \mathbb{R}^N$ has the torques applied to each joint. We parameterise a path as a function of a state variable x_1 as $q(x_1) : [0,1] \to \mathbb{R}^N$, generating all joint positions the robot will reach on the path to be traversed. Thus, the variation of x_1 from zero to one will continuously vary the robots joint positions from an initial position to a final whilst strictly following the path defined. For this reason we can consider x_1 as a pseudo-displacement. We also define the pseudo-velocity of the system on the specific path as $x_2 = \dot{x_1}$. Following literature Kang Shin and McKay (1985), we define the projected dynamics via a second order dynamical system with states $x = [x_1 \;\, x_2]^\top$. The joint velocities and accelerations are $\dot{q}(x) = \frac{dq(x_1)}{dx_1}x_2$ and $\ddot{q}(x) = \frac{dq(x_1)}{dx_1}\dot{x}_2 + \frac{d^2q}{dx_1^2}(x_2)^2$ respectively. Thus, the dynamics of the N-link robotic manipulator along the prespecified path dictated by $q(x_1)$ are

$$M(x_1)\dot{x}_2 + C(x_1)x_2^2 + g(x_1) = \tau. \tag{2}$$

The vectors $M(\cdot), C(\cdot), g(\cdot) \in \mathbb{R}^N$ represent the robot's physical properties on the path. We can use the path dynamics (2) along with the torque bounds $\tau_{\text{max}}(\cdot), \tau_{\text{min}}(\cdot) \in \mathbb{R}^N$ to find the maximum pseudo-acceleration and pseudo-deceleration of the system $A(x)$ and $D(x)$, by setting $\dot{x}_2 = u$. This allows us to eventually assign constraints on the input above (Lynch and Park, 2017, Chapter 9) as

$$D(x) \le u(x) \le A(x), \tag{3}$$

where $D(\cdot) : \mathbb{R}^2 \to \mathbb{R}, A(\cdot) : \mathbb{R}^2 \to \mathbb{R}$. We denote the *admissible region* by $\mathcal{X}$, representing all states that do not violate a set of state constraints, including the constraint $D(x) \le A(x)$. The admissible set $\mathcal{X}$ may take into account torque limitations along with other state dependent constraints related, e.g., to kinetic energy constraints related to safety considerations in collaborative robotics Rossi et al. (2015).

Assumption 1. The admissible region $\mathcal{X}$ can be written in the form

$$\mathcal{X} = \left\{x \in \mathbb{R}^2 : C^l(x_1) \le x_2 \le C^u(x_1), x_1 \in [0,1]\right\}, \tag{4}$$

where $C^l(\cdot), C^u(\cdot) : \mathbb{R} \to \mathbb{R}$ are bijections.

The resulting system is a double integrator

$$\dot{x} = \Phi x + Eu(x), \tag{5}$$

with $\Phi = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}$, $E = \begin{bmatrix}0\\ 1\end{bmatrix}$. The state feedback $u(\cdot) : \mathbb{R}^2 \to \mathbb{R}$ accounts for the acceleration $\dot{x_2}$, whose admissible range is state-dependent as given by (3). We parameterise $u(x)$ with an *actuation level function* $\lambda(x) : \mathcal{X} \to [0,1]$ as follows

$$u(x, \lambda(x)) = D(x) + \lambda(x)(A(x) - D(x)). \tag{6}$$

By considering a robot traversing a single path, we can derive the set of initial conditions in the projected space that corresponds to the initial joint positions and velocities that can be transferred safely to a target set. This set is termed as *reach-avoid set*, see, e.g. related works in Fisac et al. (2015), Landry et al. (2018), Gleason et al. (2017), Fan et al. (2021). Following convention in our previous works McGovern and Athanasopoulos (2022), McGovern et al. (2022), we denote the maximal reach avoid set by $\mathcal{R}(\mathcal{X}_\text{T})$, where $\mathcal{X}_\text{T}$ represents an interval target set in the state space defined by

$$\mathcal{X}_\text{T} = \{x \in \mathcal{X} : x_1 = c, \;\; x_l^* \le x_2 \le x_u^*\}. \tag{7}$$

Definition 1. Consider the system (5), the constraint set $\mathcal{X}$ (4), a target set $\mathcal{X}_\text{T}$ (7) and the control parameterisation $u(x)$ (6). The *maximal reach-avoid set* is

$$\begin{aligned}\mathcal{R}(\mathcal{X}_\text{T}) = \{x_0 \in \mathcal{X} : (\exists t^* > 0, \exists\lambda(\cdot) : \mathcal{X} \to [0,1] :\\ \phi_f(t^*, x_0, u(x,\lambda(x))) \in \mathcal{X}_\text{T},\\ \phi_f(t, x_0, u(x,\lambda(x))) \in \mathcal{X}, \forall t \in [0,t^*])\},\end{aligned} \tag{8}$$

where

$$\phi(t, x_0, u(x,\lambda(x))) = e^{\Phi t}x_0 + \int_0^t e^{\Phi(t-\tau)}Eu(x,\lambda(x))d\tau \tag{9}$$

is the solution of the system at time t, starting from an initial state $x_0 \in \mathcal{X}$ under feedback control strategy given by $u(x,\lambda(x))$ defined in (6). Roughly, the algorithm for generating $\mathcal{R}(\mathcal{X}_\text{T})$ exploits the Lipschitz continuity and monotonicity of the state dynamics and the ordering between trajectories of the closed loop system with respect to the parameterisation (8).

We define intervals $\mathcal{I} = [(x_a)_1, (x_b)_1]$, $(x_a)_1 \le (x_b)_1 \le 1$. The call the *slice* of a set $\mathcal{X}$ on an interval $\mathcal{I}$ the set

$$\mathcal{W}(\mathcal{X}, \mathcal{I}) = \mathcal{X} \cap \{x : x_1 \in \mathcal{I}\}. \tag{10}$$

The mapping (10) remains valid when $\mathcal{I} = \{x_a\}$ is a singleton, in this case we write for simplicity $\mathcal{W}(\mathcal{X}, x_a)$.

3. PATH SWITCHING

In our setting, the reach-avoid set $\mathcal{R}(\mathcal{X}_T)$ provides the set of admissible and safe initial conditions that can be transferred to a target set, or equivalently, traverse a prespecified path. This allows to choose between different velocity trajectories. Moreover, it immediately poses the question whether such reach-avoid sets can be combined and used as building blocks in defining a set of paths the robot can traverse and potentially switch over. We address this question by establishing conditions on combining two paths, and their associated reach avoid sets. We consider two separate paths, and allow the path being followed to be switched before the target set is reached, whilst maintaining the safety guarantees. Specifically, for a given

N link robotic manipulator with dynamics (1), we consider two feasible paths A and B. We consider two parameterised functions $q_{\mathrm{A}}(\cdot) : [0,1] \to \mathbb{R}^N$, and $q_{\mathrm{B}}(\cdot) : [0,1] \to \mathbb{R}^N$ which characterize the paths. The above parameterizations induce two state space representations with state vectors $x \in \mathcal{X}_{\mathrm{A}}$ and $y \in \mathcal{X}_{\mathrm{B}}$ that account for the projected dynamics of the system (2). Sets $\mathcal{X}_{\mathrm{A}}, \mathcal{X}_{\mathrm{B}}$ represent the constraint sets, induced by the input torque limitations and additional constraints for each state space respectively. We define target sets for each path (7), denoted by $\mathcal{X}_{\mathrm{T}}^{\mathrm{A}} \in \mathcal{X}_{\mathrm{A}}$ and $\mathcal{X}_{\mathrm{T}}^{\mathrm{B}} \in \mathcal{X}_{\mathrm{B}}$. Following McGovern and Athanasopoulos (2022), McGovern et al. (2022), we can compute the corresponding reach avoid sets which we denote by $\mathcal{R}_{\mathrm{A}} = \mathcal{R}_{\mathrm{A}}(\mathcal{X}_{\mathrm{T}}^{\mathrm{A}})$ and $\mathcal{R}_{\mathrm{B}} = \mathcal{R}_{\mathrm{B}}(\mathcal{X}_{\mathrm{T}}^{\mathrm{B}})$ respectively.

Assumption 2. We assume for both paths A and B it holds

$$\mathcal{W}(\mathcal{R}_{\mathrm{A}}, 0) \neq \emptyset, \qquad \mathcal{W}(\mathcal{R}_{\mathrm{B}}, 0) \neq \emptyset.$$

Assumption 2 ensures that it is feasible to individually complete in their entirety both paths.

Definition 2. A *switched path* from A to B is any path in the joint space starting at $q_{\mathrm{A}}(0)$ and ending at $q_{\mathrm{B}}(1)$, and is created by considering only segments of paths of either A or B.

Definition 3. An *admissible switched path* for paths A, B, with corresponding parameterisations $q_{\mathrm{A}}(\cdot)$, $q_{\mathrm{B}}(\cdot)$, is a switched path, for which, there exists a control law that drives the system (1) from state $q_{\mathrm{A}}(0)$ to $q_{\mathrm{B}}(1)$ or $q_{\mathrm{A}}(1)$, by following segments of paths A or B.

The conditions for existence of admissible switched paths are below.

Proposition 1. Consider system (1), paths A, B, with corresponding parameterisations $q_{\mathrm{A}}(\cdot), q_{\mathrm{B}}(\cdot) : [0,1] \to \mathbb{R}^N$, and reach avoid sets $\mathcal{R}_{\mathrm{A}}$, $\mathcal{R}_{\mathrm{B}}$. There exists an admissible switched path between A and B if there are vectors $x^* \in \mathcal{R}_{\mathrm{A}}$, $y^* \in \mathcal{R}_{\mathrm{B}}$ such that

(i) $q_{\mathrm{A}}(x_1^*) = q_{\mathrm{B}}(y_1^*)$,
(ii) $\dot{q}_{\mathrm{A}}(x^*) = \dot{q}_{\mathrm{B}}(y^*)$, i.e., the joints velocities are the same,
(iii) There is at least one initial state x_0 in path A, with $(x_0)_1 = 0$, from which both $x^\star$ and $\mathcal{X}_{\mathrm{T}}^{\mathrm{A}}$ can be reached.

Proof Conditions (i) and (ii) ensure there is an admissible mapping between the projected spaces, i.e., $x^* \in \mathcal{X}_{\mathrm{A}}$ and $y^* \in \mathcal{X}_B$. Condition (iii) states that at the beginning of path A, it is feasible to drive the system to both target sets $\mathcal{X}_{\mathrm{T}}^{\mathrm{A}}$ and x^*. Moreover, since $y^* \in \mathcal{R}_{\mathrm{B}}$, the target set $\mathcal{X}_{\mathrm{T}}^{\mathrm{B}}$ can be reached from y^*. Thus, there is at least one initial condition x_0 and an admissible control law so that an admissible switched path can be followed. ■

It is worth noticing that condition (i) is the only necessary condition required for the existence of a geometric *switched path* to be formed. On the other hand, conditions (ii), (iii) are the stricter conditions that guarantee the existence of an *admissible switched path*, i.e., a geometric switched path that is a feasible trajectory for the system, meaning that there exists a state feedback so that the system can follow the path with the differential constraints LaValle (2006) satisfied.

The next result translates the conditions of Proposition 1 in verifiable relations, namely, algebraic ones or set inclusions.

Theorem 1. Consider system (1), paths A, B with corresponding parameterisations $q_{\mathrm{A}}(\cdot), q_{\mathrm{B}}(\cdot) : [0,1] \to \mathbb{R}^N$, and reach avoid sets $\mathcal{R}_{\mathrm{A}} = \mathcal{R}_{\mathrm{A}}(\mathcal{X}_{\mathrm{T}}^{\mathrm{A}})$, and $\mathcal{R}_{\mathrm{B}} = \mathcal{R}_{\mathrm{B}}(\mathcal{X}_{\mathrm{T}}^{\mathrm{B}})$ where $\mathcal{X}_{\mathrm{T}}^{\mathrm{A}}$, where $\mathcal{X}_{\mathrm{T}}^{\mathrm{B}}$ are the target sets for paths A and B respectively. Then, an *admissible switched path* exists if there is at least one pair of vectors $(x^*, y^*) \in \mathcal{X}_A \times \mathcal{X}_B$ so that

(i) $q_{\mathrm{A}}(x_1^*) = q_{\mathrm{B}}(y_1^*)$,
(ii) If $x_2^* > 0$, and $y_2^* > 0$, for any $(i,j) \in \{1,N\} \times \{1,N\}$, $i \neq j$, it holds

$$\left.\frac{d(q_{\mathrm{A}}(x_1))_i}{d(q_{\mathrm{A}}(x_1))_j}\right|_{x_1 = x_1^*} = \left.\frac{d(q_{\mathrm{B}}(y_1))_i}{d(q_{\mathrm{B}}(y_1))_j}\right|_{y_1 = y_1^*}, \tag{11}$$

(iii) $y^* \in \mathcal{R}_{\mathrm{B}}$,
(iv) Given y^*, let $\mathcal{X}^\star$ denote the set of states x^* satisfying (i)–(iii). Then, $\mathcal{W}((\mathcal{R}(\mathcal{X}^\star) \cap \mathcal{R}_{\mathrm{A}}), 0) \neq \emptyset$.

Proof Condition (i) coincides with condition (i) of Proposition 1 and is necessary for the existence of a switched path. Let us consider a pair $(x_1^*, y_1^*) \in [0,1]^2$ satisfying conditions (i) and (ii) of Proposition 1. Then, $\dot{q}_{\mathrm{A}}(x_1^*) = \dot{q}_{\mathrm{B}}(y_1^*)$, and it follows

$$\left.\frac{dq_{\mathrm{A}}(x_1)}{dx_1}\right|_{x_1 = x_1^*} x_2^* = \left.\frac{dq_{\mathrm{B}}(y_1)}{dy_1}\right|_{y_1 = y_1^*} y_2^*, \tag{12}$$

where $\frac{dq_{\mathrm{A}}(x_1)}{dx_1}, \frac{dq_{\mathrm{A}}(x_1)}{dx_1} \in \mathbb{R}^N$. Consequently, from (12) we can find the pairs (x_2^*, y_2^*) satisfying

$$\frac{y_2^*}{x_2^*}\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix} = \left.\begin{bmatrix}\frac{d(q_{\mathrm{A}}(x_1))_1}{dx_1}\left(\frac{d(q_{\mathrm{B}}(y_1))_1}{dy_1}\right)^{-1} \\ \vdots \\ \frac{d(q_{\mathrm{A}}(x_1))_N}{dx_1}\left(\frac{d(q_{\mathrm{B}}(y_1))_N}{dy_1}\right)^{-1}\end{bmatrix}\right|_{\substack{x_1 = x_1^*,\\ y_1 = y_1^*}}. \tag{13}$$

The vector equation (13) must hold elementwise, which is true if (11) holds for any $i \in [1,N]$ and $j \in [1,N]$, $i \neq j$. Therefore, when $x_2^* > 0$, $y_2^* > 0$, the condition $\dot{q}_{\mathrm{A}}(x^*) = \dot{q}_{\mathrm{B}}(y^*)$ holds if and only if the gradients of the two paths are the same in the joint space. We note when $x_2^* = 0$, $y_2^* = 0$, the joint velocity of each joint is zero thus (11) is not required. Condition (iii) coincides with the statement in Proposition 1. By definition of the reach-avoid set, condition (iv) guarantees existence of an initial condition $x_0 \in \mathcal{W}((\mathcal{R}(\mathcal{X}^\star) \cap \mathcal{R}_{\mathrm{A}}), 0)$ such that the path A can be traversed, and additionally there is a time t^* such that $\phi(t^\star, x_0, u(x, \lambda(x))) \in \mathcal{R}(\mathcal{X}^\star)$, effectively implying condition (iii) of Proposition 1. ■

A few remarks are in order: Condition (i) of Theorem 1 induces a system of N equations with two unknowns, and x_1^* and y_1^* are attainable from solving a set of algebraic equations regardless of the parameterisation. It should also be noted that multiple solutions may exist. The equality constraints (13) form an admissible constraint set of pairs (x_2^*, y_2^*), effectively inducing two intervals for a fixed pair (x_1^*, y_1^*). Conditions (iii), (iv) can be verified by utilising the framework in McGovern et al. (2022). To showcase this, particular cases of the shapes of the involved paths and solution sets of conditions (i)-(ii) of Theorem 1 are outlined in the following subsection.

3.1 Special cases of path switching

We note that paths are defined in the joint space, leading to possibly complex corresponding configurations in the workspace, as it is shown in numerical example in Section 4.

Linear-to-linear paths. Linear paths in the joint space are linear functions of x_1 with start and end positions $q_A(0)$ and $q_A(1)$ respectively. A suitable parameterization is a function of $x_1 \in [0,1]$ as $q(x_1) = q(0) + x_1(q(1) - q(0))$. Let path A begin at $A_s = q_A(0)$ and end at $A_e = q_A(1)$, and path B begin at $B_s = q_B(0)$ and end at $B_e = q_B(1)$ be defined by

$$q_{\mathrm{A}}(x_1) = A_s + x_1(A_e - A_s), \tag{14}$$
$$q_{\mathrm{B}}(y_1) = B_s + y_1(B_e - B_s), \tag{15}$$

with $y_1 \in [0, 1]$. At the intersection point we can calculate the input parameters x_1^* and y_1^* by equating equations (14) and (15) as $q_{\mathrm{B}}(y_1^*) = q_{\mathrm{A}}(x_1^*)$, or,

$$y_1^*(B_e - B_s) = x_1^*(A_e - A_s) + (A_s - B_s). \tag{16}$$

Let a solution pair be (x_1^*, y_1^*). To retrieve the corresponding admissible pairs (x_2^*, y_2^*), at the point of intersection, Theorem 1 (ii) becomes $\dot{q}_B(y_1^*) = \dot{q}_A(x_1^*)$, or $\frac{dq_B(y_1^*)}{dy_1} y_2^* = \frac{dq_A(x_1^*)}{dx_1} x_2^*$, or

$$(B_e - B_s)y_2 = (A_e - A_s)x_2. \tag{17}$$

Rearranging each row of equation (17) leads to

$$\begin{bmatrix} y_2^* \\ \vdots \\ y_2^* \end{bmatrix} = \begin{bmatrix} \frac{(A_e)_1 - (A_s)_1}{(B_e)_1 - (B_s)_1} \\ \vdots \\ \frac{(A_e)_N - (A_s)_N}{(B_e)_N - (B_s)_N} \end{bmatrix} \begin{bmatrix} x_2^* \\ \vdots \\ x_2^* \end{bmatrix}. \tag{18}$$

From equation (18) it can be seen that all elements in the the middle vector must be equal, thus, by equating any two elements i, j we obtain

$$\frac{(B_e)_j - (B_s)_j}{(B_e)_i - (B_s)_i} = \frac{(A_e)_j - (A_s)_j}{(A_e)_i - (A_s)_i}. \tag{19}$$

From (19), it follows that in the case $x_2^* > 0$ or $y_2^* > 0$, the only admissible solution would concern paths that are parallel to one another. If $x_2^* = y_2^* = 0$ then the lines need not be parallel. In agreement with intuition, in this case the robotic manipulator will have to stop its motion in order to switch the path.

Reversing a path. We can consider this case as switching between two paths A and B covering the same trajectory in the joint space however in opposite directions. Consequently, path A begins at $q_A(0)$ and ends at $q_A(1)$ and path B begins at $q_A(1)$ and ends at $q_A(0)$. We can define both these paths via parameterisations $q_A(x_1)$ and $q_B(y_1)$ respectively. It is possible to couple the input parameters as $y_1 = 1 - x_1$, and $x_1 = 1 - y_1$. This allows us to associate the parameterisations as $q_A(x_1) = q_B(1 - y_1)$, $q_B(y_1) = q_A(1 - x_1)$. From Theorem 1 (ii), for any pair $(x_1^*, 1 - x_1^*)$, it can be seen that the only feasible solution for the pair (x_2^*, y_2^*) is $(0, 0)$. This is in agreement with intuition, as the only way for a robotic manipulator to go in reverse direction would be to instantaneously halt.

Linear-to-Quadratic paths. We consider a linear path A parameterised by (14) and a quadratic path B parameterised by $q_B(y_1)$ as follows

$$q_{\mathrm{B}}(y_1) = \begin{bmatrix} (B_s)_1 + y_1((B_e)_1 - (B_s)_1) \\ \vdots \\ (B_s)_{N-1} + y_1((B_e)_{N-1} - (B_s)_{N-1}) \\ a y_1^2 + b y_1 + c \end{bmatrix}. \tag{20}$$

At a path intersection, we solve for x_1^* and y_1^* using $q_{\mathrm{B}}(y_1^*) = q_{\mathrm{A}}(x_1^*)$. The difference with the linear-to-linear switching case is in the N^{th} element. One can first solve for finding the range of candidate pairs (x_1, y_1) satisfying the first $N - 1$ equations, and next identify the roots of $a(y_1^*)^2 + b(y_1^*) + c = (A_s)_N + x_1^*((A_e)_N - (A_s)_N)$. Solving the first $N - 1$ equations of (13) generate admissible intervals of pairs (x_2^*, y_2^*) for any pair (x_1^*, y_1^*), while the N-th equation further restricting the range to $(2ay_1^* + b)y_2^* = ((A_e)_N - (A_s)_N)x_2^*$. We note more general quadratic parameterisations can be utilised by setting $q'_{\mathrm{B}}(y_1) = Tq_{\mathrm{B}}(y_1) + c$, where $T \in \mathbb{R}^{N \times N}$ is a nonsingular transformation matrix and $c \in \mathbb{R}^N$ a constant vector.

3.2 Controller design

We let the pair (x^*, y^*) define the switching between two paths A, B. For the path dynamics (2) we have for path A and B

$$M^A(x_1)u_A + C^A(x)x_2^2 + g^A(x_1) = \tau^A, \tag{21}$$
$$M^B(y_1)u_B + C^B(y)y_2^2 + g^B(y_1) = \tau^B. \tag{22}$$

Letting $x = x^*$, $y = y^*$, $u_A = u_A^*$, $u_B = u_B^*$ and $\tau^A = \tau^B = \tau$ at the point of intersection, and since $g_A(x_1^*) = g_B(y_1^*)$, by rearranging equations (21) and (22) we obtain

$$M^B(y_1^*)u_B^* = M^A(x_1^*)u_A^* + C^A(x^*)(x_2^*)^2 - C^B(y^*)(y_2^*)^2.$$

Solving for u_B^* leads to

$$u_{\mathrm{B}}^* = \frac{M_i^A(x_1^*)u_{\mathrm{A}}^* + C_i^A(x^*)(x_2^*)^2 - C_i^B(y^*)(y_2^*)^2}{M_i^B(y_1^*)}, \tag{23}$$

for any $i \in [1, N]$, when $M_i^B \neq 0$. The constraints on u_A^* and u_B^* are different depending on the path. We can denote them using equation (3) as $D_{\mathrm{A}}(x^*) \leq u_{\mathrm{A}} \leq A_{\mathrm{A}}(x^*)$, $D_{\mathrm{B}}(y^*) \leq u_{\mathrm{B}} \leq A_{\mathrm{B}}(y^*)$ For an admissible switched path we have by Theorem 1 that $x^* \in \mathcal{R}_{\mathrm{A}} \subseteq \mathcal{X}_{\mathrm{A}}$ and $y^* \in \mathcal{R}_{\mathrm{B}} \subseteq \mathcal{X}_{\mathrm{B}}$. Therefore we can guarantee that $D_{\mathrm{A}}(x^*) \leq A_{\mathrm{A}}(x^*)$ and $D_{\mathrm{B}}(y^*) \leq A_{\mathrm{B}}(y^*)$. To ensure feasibility, we can enforce

$$D_{\mathrm{B}}(y^*) \leq u_{\mathrm{B}}^* \leq A_{\mathrm{B}}(y^*). \tag{24}$$

By taking into account the state feedback controller parameterisation for each path from (6) we obtain

$$u_{\mathrm{A}}(x, \lambda_{\mathrm{A}}(x)) = D_{\mathrm{A}}(x) + \lambda_{\mathrm{A}}(x)(A_{\mathrm{B}}(x) - D_{\mathrm{B}}(x)), \tag{25}$$
$$u_{\mathrm{B}}(y, \lambda_{\mathrm{A}}(y)) = D_{\mathrm{B}}(y) + \lambda_{\mathrm{B}}(y)(A_{\mathrm{B}}(y) - D_{\mathrm{B}}(y)), \tag{26}$$

Since $D_{\mathrm{A}}(x), D_{\mathrm{B}}(y), A_{\mathrm{A}}(x), A_{\mathrm{B}}(y)$, are under mild assumptions Lipschitz continuous McGovern et al. (2022), it holds that $u_A(x, \lambda_{\mathrm{A}}(x))$ and $u_A(x, \lambda_{\mathrm{A}}(x))$ will share the property if $\lambda_{\mathrm{A}}(x)$ and $\lambda_{\mathrm{B}}(y)$ are Lipschitz continuous as well. We can use equations (23)-(26), to define the relationship between the outputs of $\lambda_{\mathrm{A}}(x^*)$ and $\lambda_{\mathrm{B}}(y^*)$, and at the switch point to maintain Lipschitz continuity around the switch. A possible algorithmic procedure to retrieve a solution is summarised in the following steps.

1. Calculate $u_{\mathrm{A}}^* = u_{\mathrm{A}}(x^*, \lambda_{\mathrm{A}}(x^*))$ from (25).
2. Substitute u_{A}^* into (23) and obtain u_{B}^*.
3. Verify whether u_{B}^* satisfies (24). If not, a path switch is possible but the state feedback controller will not be Lipschitz continuous on the switching point for the choice (x^*, y^*).
4. If (24) is met, solve (26) to obtain $\lambda_{\mathrm{B}}(y)$.
5. Form $\lambda_{\mathrm{B}}(y)$, for $y_1 \in [y^*, 1]$ so that a Lipschitz continuous controller is constructed for the remaining path on path B.

Possible choices for designing $\lambda_{\mathrm{A}}(x)$ and $\lambda_{\mathrm{B}}(y)$, to maintain Lipschitz continuity properties is discussed in McGovern et al. (2022).

4. ILLUSTRATIVE EXAMPLE

We consider a planar robotic manipulator with two degrees of freedom, consisting of two joints of 0.2m length, with the center of mass of each link lying at the end, i.e. all the mass of the robot focused on the second joint, and end effector with the mass of each link set as 0.25kg. We define the two paths in the joint space. The parameterisation $q_A(x_1)$ transfers the robot from $A_1 = [-1.57\ 0]^\top$ to $A_2 = [1.75\ -0.87]^\top$, while $q_B(y_1)$ transfers the robot from $B_1 = [-2\ 2.5711]^T$ to $B_2 = [2.18\ 1.4764]^T$. In particular, we define

$$q_A(x_1) = \begin{bmatrix} -1.57 + 3.32x_1 \\ -0.00505 + 8.984x_1 - 9.854x_1^2 \end{bmatrix},$$
$$q_B(y_1) = \begin{bmatrix} -2 + 4.18y_1 \\ 2.5711 - 1.0947y_1 \end{bmatrix}.$$

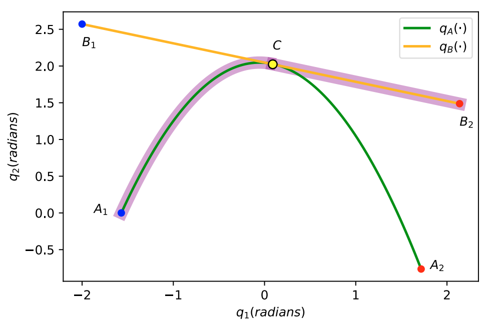


Fig. 1. Paths A and B in the joint space, the purple path represents the switched path from path A to path B, the switch occurs at point C.

Figure 1 illustrates the trajectories in the joint space. Solving the condition $q_A(x_1^*) = q_B(x_1^*)$ from Theorem 1 (i), we find a single solution pair $x_1^* = y_1^* = 0.5$. Condition of Theorem 1 (ii) is met by verifying the derivative of path A evaluated at $x_1^* = 0.5$ equals the gradient of the path B.

Figure 2 illustrates the paths A and B in the workspace. The objective is to verify whether there exists an admissible switched path that will allow a motion that begins at A_1, to reach B_2, whilst following either path A or path B. In situations such as pick-and-place applications or manufacturing, where a robot is working in a limited space and has limited pathways to complete an objective without collision, it can be advantageous to allow for adaption to changing objectives during motion. The third condition of Theorem 1 requires at least one state $y^* \in \mathcal{R}_B$. In order to switch from path A to path B we must find a corresponding state $x^* \in \mathcal{R}_A$ such that $\dot{q}_A(x^*) = \dot{q}_B(y^*)$. Substituting values, we obtain

$$y_2^* = 0.7943x_2^*. \tag{27}$$

To satisfy all conditions of Theorem 1 we find first the set of states x^* and y^* such that $x^* \in \mathcal{I}_A = \mathcal{W}(\mathcal{R}_A, x_1^*)$, $y^* \in \mathcal{I}_B = \mathcal{W}(\mathcal{R}_B, y_1^*)$. We obtain

$$\mathcal{I}_A = [0, 5.75], \qquad \mathcal{I}_B = [0.95, 6.6].$$

The set $\mathcal{I}_B$ is shown in Figure 3b. From $\mathcal{I}_A$ and (27), the candidate interval $\mathcal{I}_B^A$ for path switching from A to B is $\mathcal{I}_B^A = [0.7546, 5.29]$. This lets us directly define the interval of x_2^* states that enable path switching as $\mathcal{I}^* = \mathcal{I}_A \cap \mathcal{I}_B^A$, shown in Figure 3a. Finally, for the path to be an admissible switched path, we must also satisfy condition (iv) of Theorem 1, namely, it is possible to reach a state $x^* \in \mathcal{W}(\mathcal{R}_A, \mathcal{I}^*)$, from an initial condition x_0 with $(x_0)_1 = 0$. In this example, the entire slice $\mathcal{W}(\mathcal{R}_A, 0)$ is included in the reach avoid set $\mathcal{R}_A$, thus, from any choice of $x_0 \in \mathcal{W}(\mathcal{R}_A, 0)$ i.e. the system begins at A_1, it is possible to reach B_2. Figures 3a, and 3b show the state spaces of both paths A and B. The possible admissible switched path is shown in Figure 2c.

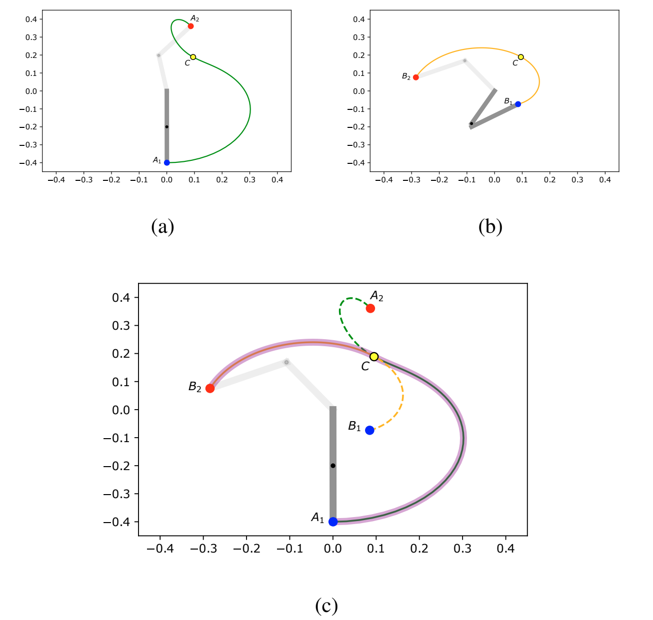


Fig. 2. (a) Path A in the workspace (b) path B int the workspace (c) Representation of both paths A and B in the workspace with the admissible switched path to be followed highlighted in purple.

Figure 4 illustrates via a directed graph the possible paths that can be taken in this example. In the graph, the nodes represent the start and end points in the paths A and B, together with their intersection which is denoted by C. The directed edges suggest an admissible path (together with a corresponding controller) exists between these points.

5. CONCLUSIONS

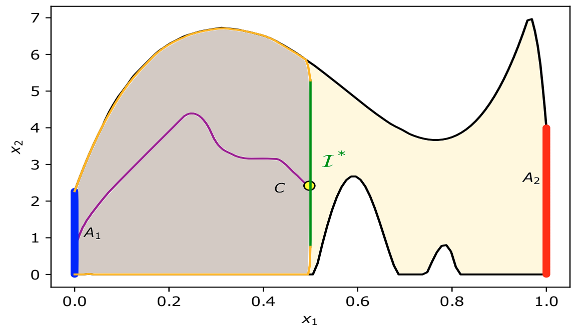
In this paper we explored combining predefined paths for robotic manipulators for which a set of admissible velocity profiles and corresponding state feedback controllers exist. Based on previous work that allows an efficient computation of reach avoid sets that characterize all safe velocity profiles for any given path, we established switching conditions that are verifiable by solving a set of algebraic relations or checking for nonempty intersections between largely precomputed sets. These results can be potentially used for for exploring the establishment of a formal framework leading to path planning formulations by composition and switching of sample paths.

REFERENCES

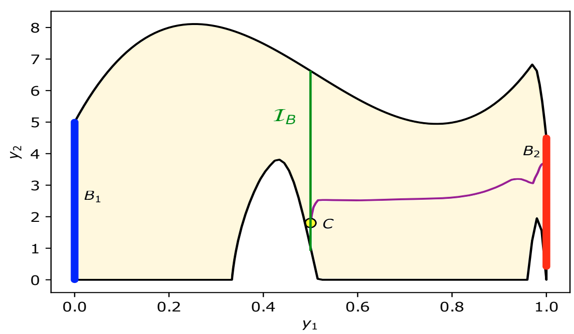
Alur, R., Henzinger, T., Lafferriere, G., and Pappas, G. (2000a). Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7), 971–984. doi:10.1109/5.871304.

Alur, R., Henzinger, T.A., Lafferriere, G., and Pappas, G.J. (2000b). Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7), 971–984.

Baressi Šegota, S., Anđelić, N., Lorencin, I., Saga, M., and Car, Z. (2020). Path planning optimization of six-degree-of-freedom robotic manipulators using evolutionary algorithms.



(a)



(b)

Fig. 3. (a)The state space of path A: The reach avoid set boundaries $\mathcal{R}_\mathrm{A}$ are represented in black. The reach avoid set $\mathcal{R}_\mathrm{A}(\mathcal{I}^*)$ is represented by the gray shaded region with orange boundaries.A trajectory to the interval $\mathcal{I}^*$ is given from which we switch to the path B state space. (b)The state space of path B: The switched path starts from point C within the interval $\mathcal{I}_\mathrm{B}$ and reaches the target set, depicted in red, relating to point B_2 in the joint space.

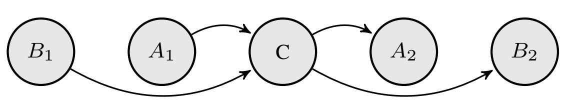


Fig. 4. Directed graph representing the possible paths that can be composed from the two paths A and B.

International Journal of Advanced Robotic Systems, 17(2), 1729881420908076.

Caron, S., Pham, Q.C., and Nakamura, Y. (2017). Completeness of randomized kinodynamic planners with state-based steering. *Robotics and Autonomous Systems*, 89, 85–94.

Chen, X. and Sankaranarayanan, S. (2022). Reachability analysis for cyber-physical systems: Are we there yet? In *NASA Formal Methods Symposium*, 109–130. Springer.

Cohen, B.J., Subramania, G., Chitta, S., and Likhachev, M. (2011). Planning for manipulation with adaptive motion primitives. In *2011 IEEE International Conference on Robotics and Automation*, 5478–5485. IEEE.

Egidio, L.N., Lima, T.A., and Jungers, R.M. (2022). State-feedback abstractions for optimal control of piecewise-affine systems. *arXiv preprint arXiv:2204.00315*.

Fan, C., Qin, Z., Mathur, U., Ning, Q., Mitra, S., and Viswanathan, M. (2021). Controller synthesis for linear system with reach-avoid specifications. *IEEE Transactions on Automatic Control*, 67(4), 1713–1727.

Fisac, J.F., Chen, M., Tomlin, C.J., and Sastry, S.S. (2015). Reach-avoid problems with time-varying dynamics, targets and constraints. In *Proceedings of the 18th international conference on hybrid systems: computation and control*, 11–20.

Gasparetto, A. and Zanotto, V. (2007). A new method for smooth trajectory planning of robot manipulators. *Mechanism and machine theory*, 42(4), 455–471.

Gleason, J.D., Vinod, A.P., and Oishi, M.M.K. (2017). Under-approximation of reach-avoid sets for discrete-time stochastic systems via lagrangian methods. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 4283–4290. doi:10.1109/CDC.2017.8264291.

Hauser, K., Bretl, T., Harada, K., and Latombe, J.C. (2008). Using motion primitives in probabilistic sample-based planning for humanoid robots. In *Algorithmic foundation of robotics VII*, 507–522. Springer.

Hussien, O. and Tabuada, P. (2018). Lazy controller synthesis using three-valued abstractions for safety and reachability specifications. In *2018 IEEE Conference on Decision and Control (CDC)*, 3567–3572. IEEE.

Kang Shin and McKay, N. (1985). Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, 30(6), 531–541. doi: 10.1109/TAC.1985.1104009.

Landry, B., Chen, M., Hemley, S., and Pavone, M. (2018). Reach-avoid problems via sum-or-squares optimization and dynamic programming. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4325–4332. IEEE.

LaValle, S.M. (2006). *Planning algorithms*. Cambridge university press.

Lynch, K.M. and Park, F.C. (2017). *Modern Robotics*.

McGovern, R., Athanasopolous, N., and McLoone, S. (2022). Set based velocity shaping for robotic manipulators. doi:10.48550/ARXIV.2211.05342. URL `https://arxiv.org/abs/2211.05342`.

McGovern, R. and Athanasopoulos, N. (2022). Kinodynamic planning for robotic manipulators using set-based methods. In *2022 European Control Conference (ECC)*, 1309–1314. IEEE.

Perez, A., Karaman, S., Shkolnik, A., Frazzoli, E., Teller, S., and Walter, M.R. (2011). Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms. In *2011 IEEE/RSJ international conference on intelligent robots and systems*, 4307–4313. IEEE.

Rossi, R., Polverini, M.P., Zanchettin, A.M., and Rocco, P. (2015). A pre-collision control strategy for human-robot interaction based on dissipated energy in potential inelastic impacts. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 26–31. IEEE.

Sibai, H. and Mitra, S. (2020). Symmetry abstractions for hybrid systems and their applications. *arXiv preprint arXiv:2006.09485*.

Villani, V., Pini, F., Leali, F., and Secchi, C. (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55, 248–266.

Wang, Z., Garrett, C.R., Kaelbling, L.P., and Lozano-Pérez, T. (2021). Learning compositional models of robot skills for task and motion planning. *The International Journal of Robotics Research*, 40(6-7), 866–894.