Time Optimal Control along Specified Paths with Acceleration Continuity: A Non-Linear Time Scaling based approach

Arun Kumar Singh[1], Bharath Gopalakrishnan[1] and K.Madhava Krishna[1]

***Abstract*— In this paper we present a *non-linear time scaling* based formulation for computing time optimal motions along specified paths. The primary motivation behind the current work is to introduce acceleration continuity constraints within the time optimal framework. Such constraint necessitates the use of time varying controls instead of commonly used piece-wise constant controls. We propose a novel extension of our previously developed concept of *non-linear time scaling* through which it is possible to parametrize controls as piece-wise product of a exponential and a linear function. We show that such representation leads to a very simple optimization structure with primarily linear constraints. The non-linearity has a quasi-convex structure which we reformulate into a simple *difference of convex form*. A sequential convex programming framework is utilised to solve the optimization as a sequence of sparse quadratic programmes. The proposed optimization is an improvement over the current state of the art frameworks which introduces acceleration continuity constraints in the time optimal framework through highly non-linear and non-convex optimizations.**

I. INTRODUCTION

Computing motions along specified paths forms a key component in many motion planning problems. For example to reduce the complexity of a general motion planning problem, it is often broken down into two hierarchical steps. At the first step smooth kinematically feasible, collision free paths are computed, and then at the second step motion profiles along it are computed subject to velocity, acceleration bounds and higher level constraints such as dynamic collision avoidance. Some such works can be found in [1], [2]. Thus computing time optimal motions along specified paths subject to some generic set of constraints is critical to various motion planning problems.

Time optimal motion planning along specified paths decomposes a general optimal control problem into just two variables, velocity and acceleration along a particular path which can respectively be considered as psuedo state and control input. Existing works like [3], [4], [5], [6], [7] solves this optimal control problem with piece-wise constant parametrization for the control input, which leads to convexity in the resulting optimization structure. Convex optimization are not only computationally efficient but also various open source solvers like [8] exist for rapid prototyping. However, piece-wise constant control leads to bang-bang control trajectory which are difficult to track [9]. Works like [10] introduces additional jerk constraints to smoothen the evolution of the control trajectory. To ensure a simple optimization structure, [10] still assumes piece-wise constant control and thus the jerk constraints smoothens the control trajectory at the cost of time optimality of the trajectory.

The current work goes beyond jerk constraints and introduces acceleration continuity constraints in the time optimal framework. Such constraints would result in a smoother acceleration profile. Moreover for some robotic systems like quadcopter acceleration continuity in the trajectory is imperative [11]. Acceleration continuity requires the use of time varying control input instead of piece-wise constant control. Some works which computes acceleration continuous time optimal motions along given geometric paths can be found in [12] and [13], where B-Spline representation for the time varying control is used. However acceleration continuity in these frameworks come at the cost of loss in convexity of the problem. In particular the optimization frameworks of [12] and [13] has polynomial constraints and thus, are highly non-linear and non-convex. As well known such optimization problems are difficult to solve and sometimes even computing a feasible solution proves to be difficult.

The primary contribution of the current work is that it introduces acceleration continuity in the time optimal framework but at the same time ensures a very simple optimization structure, where most of the constraints are linear. The non-linearity has a special quasi-convex structure which is reformulated into a simple *difference of convex form* [15]. The optimization is solved through *sequential convex programming* as a sequence of sparse, convex quadratic programmes. The simplified structure of the optimization is a direct consequence of our previously developed concept of *non-linear time scaling* which allows us to represent time varying controls as piece-wise product of a exponential and a linear function. As shown later, this exponential representation forms the crux of the proposed framework.

A. Layout of the Paper

The rest of the paper is organized as follows: Section II describes the problem formulation along the lines of [6] and [7] in the form of an optimization problem. Section III introduces the non linear time scaling based reformulation and presents a simplified optimization problem. Section IV explains the solution procedure through *sequential convex programming*. Section V presents some implementation examples and discussions.

II. PROBLEM FORMULATION

Without loss of generalization we consider a path in euclidean space i.e $\mathbf{X}(u) = (x(u), y(u), z(u))^T$, where u is an arbitrary path variable. The objective is to compute

[1]Authors are with Robotics Research Centre,IIIT-Hyderabad,India arunkumar.singh@research.iiit.ac.in

time optimal motion profiles along the path $\mathbf{X}(u)$ subject to actuator constraints. Like [5], [6] we assume that the actuator constraints can be transformed to equivalent bounds on acceleration. This assumption holds for many common robotic systems like manipulators and wheeled mobile robots. Moreover even for robotic systems like quadcopters time optimal planning can be done with acceleration bounds as actuator constraints [11]. Thus building on the philosophy of [4], [5], to compute time optimal motions, at each point along the path, we seek to bring the velocity profile of the robot as close as possible to the maximum velocity limit subject to acceleration bounds. The derivations presented in this paper has been done assuming a free flying robot in 3D space like a quadcopter. Extension to manipulator like systems is trivial.

The total velocity $v(t)$ and acceleration $a(t)$ of a robot operating in 3D space can be written as

$$v(t) = \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2}, a(t) = \sqrt{\ddot{x}(t)^2 + \ddot{y}(t)^2 + \ddot{z}(t)^2} \tag{1}$$

$\dot{x}(t), \ddot{x}(t)$ and similarly others represents individual velocity and acceleration components respectively. They can be derived from path derivatives $x^{'}(u), x^{''}(u)$ by transforming the path variable from u to the time variable t which results in the following expressions.

$$\dot{\mathbf{X}}(t) = \mathbf{X}^{'}(u)\frac{du}{dt}, \ddot{\mathbf{X}}(t) = (\frac{du}{dt})^2\mathbf{X}^{''}(u) + \mathbf{X}^{'}(u)\frac{d^2u}{dt^2} \tag{2}$$

(2) suggests that the velocity components are derived by scaling the path derivatives with the function $\frac{du}{dt}$ which is called the scaling function. $\frac{du}{dt}$ and $\frac{d^2u}{dt^2}$ is equivalent to the concept of path velocity and acceleration in [4], [5] and [7]. In all these works path acceleration, which can be considered as a pseudo control input is piece wise constant. In [6], a piece-wise constant form of $\frac{d^2u}{dt^2}$ is used. But as shown in next section, in the proposed work $\frac{d^2u}{dt^2}$ is piecewise exponential. A careful look into (2) would reveal that irrespective of the smoothness of $\mathbf{X}^{'}$(u), $\dot{\mathbf{X}}(t)$ (i.e individual velocity components) can be made piece-wise arbitrarily smooth by using appropriately smooth enough $\frac{du}{dt}$.

Using the notation $\dot{u} = \frac{du}{dt}$ and substituting (2) in (1) we get

$$v(t(u)) = \dot{u}\sqrt{v(u)}, v(u) = (x^{'}(u))^2 + (y^{'}(u))^2 + (z^{'}(u))^2 \tag{3}$$

As stated above, time optimal motion planning along given specified paths reduces to bringing the velocity profile as close as possible to maximum velocity limit. In other words the following objective function is to be minimized.

$$J = \int_{u_0}^{u_f} (\dot{u}\sqrt{\rho(u)} - v_{max}(u))^2 \tag{4}$$

(4) has to be minimized with respect to the following constraints

$$\dot{u}\sqrt{\rho(u)} \leq v_{max}(u), \dot{u}(u_0) = \dot{s}_0, \dot{u}(u_f) = \dot{s}_f \tag{5}$$

$$\ddot{u}(u_0) = \ddot{s}_0, \ddot{u}(u_f) = \ddot{s}_f, \ddot{u} \in \mathcal{C}^0 \tag{6}$$

$$|x^{''}(u)\dot{u}^2 + x^{'}(u)\ddot{u}| \leq \ddot{x}_{max}(u) \tag{7}$$

$$|y^{''}(u)\dot{u}^2 + y^{'}(u)\ddot{u}| \leq \ddot{y}_{max}(u) \tag{8}$$

$$|z^{''}(u)\dot{u}^2 + z^{'}(u)\ddot{u}| \leq \ddot{z}_{max}(u) \tag{9}$$

The inequality in (5) implies that the total velocity should always be less than the maximum value. The equality constraints in (5) and (6) enforces boundary conditions on the scaling function and it's derivative. Appropriate boundary values for the scaling function and it's derivative can be chosen to satisfy the boundary conditions on velocity and accelerations. The third constraint in (6) ensures that the derivative of scaling function i.e $\ddot{u}$ is continuous throughout. This in turn implies that the velocity profile is continuous and differentiable throughout. It is very easy to verify that continuity in $\ddot{u}$(and consequently $\dot{u}$) would imply continuity in velocity and acceleration components.

Inequalities (7)-(9) are obtained by writing the second equality of (2) component wise and represent the acceleration level constraints. The path derivatives like $x^{'}(u), x^{''}(u)$ and others are determined by the given 3D path and hence the only variable of the optimization problem (4)-(9) is the function $\dot{u}$ (since $\ddot{u}$ is itself determined by $\dot{u}$). Although at the moment (4)-(9) do not reveal any special structure as such, but the reformulation presented in the next section where $\dot{u}$ is represented as paramterized exponential functions allows us to convert the objective function and constraints into quadratic and linear form respectively.

III. Problem Reformulation Through Exponential Form of $\dot{u}$

The reformulation presented in this section is derived by representing $\dot{u}$ as a continuous combination of exponential functions in the following form

$$\dot{u} = pe^{-qu^2 - ru} \tag{10}$$

where p, q, r are constants and u as before is the path variable. For further clarity consider figure 1 which shows the interval $[u_0, u_f]$ discretized into $n+1$ subintervals by grid points $u_1, u_2, u_3....u_n$. At any subinterval i.e between any two grid points a different exponential function determined by parameters $p_1, q_1, r_1, p_2, q_2, r_2...p_n, q_n, r_n, p_{n+1}, q_{n+1}, r_{n+1}$ is defined. To derive the general form of $\dot{u}$ at any subinterval, we start with the first subinterval $[u_0 \;\; u_1]$. Let the initial boundary value of $\dot{u}$ be $\dot{s}_0$. To satisfy the initial boundary condition, we must have

$$\dot{u}(u_0) = p_1e^{-q_1u_0^2 - r_1u_0} = \dot{s}_0 \Rightarrow p_1 = \dot{s}_0e^{q_1u_0^2 + r_1u_0} \tag{11}$$

Using (11) $\dot{u}$ in the interval $[u_0 \;\; u_1]$ can be represented in the following form

$$\dot{u}(u) = \dot{s}_0e^{q_1(u_0^2 - u^2) + r_1(u_0 - u)}, \forall u \in [u_0 \;\; u_1) \tag{12}$$

Similarly from figure 1 , it can be seen that in the interval $[u_1 \;\; u_2]$ $\dot{u}$ is defined as $p_2e^{-q_2u^2 - r_2u}$. To ensure continuity between the scaling functions at the adjacent interval we must have

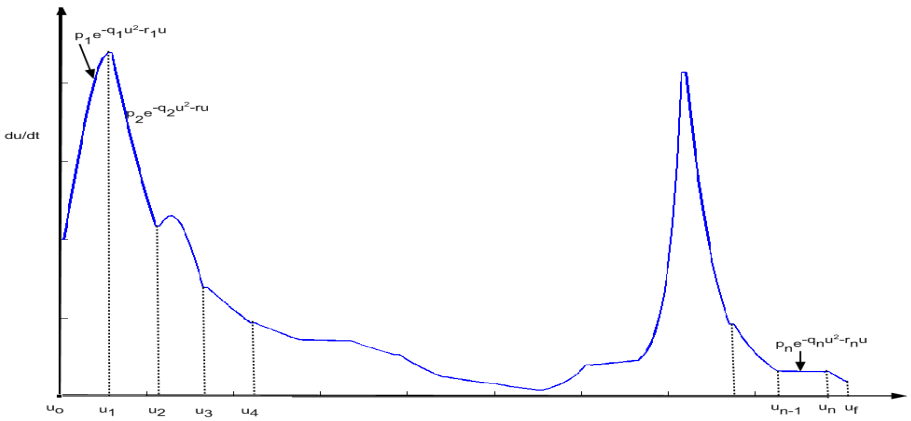


Fig. 1. (a): General form of $\dot{u}$ as a differentiable combination of parametrized exponential function. $\ddot{u}$ can also be derived from $\dot{u}$

$$\begin{aligned} \dot{u}(u_1) &= p_2 e^{-q_2 u_1^2 - r_2 u_1} \\ \Rightarrow \dot{s}_0 e^{q_1(u_0^2-u_1^2)+r_1(u_0-u_1)} &= p_2 e^{-q_2 u_1^2 - r_1 u_1} \\ \Rightarrow p_2 = \dot{s}_0 e^{q_1(u_0^2-u_1^2)+r_1(u_0-u_1)+q_2 u_1^2 + r_2 u_1} & \end{aligned} \tag{13}$$

Using (13), $\dot{u}$ in the interval $[u_1 \;\; u_2]$ can be defined as

$$\dot{u}(u) = \dot{s}_0 e^{q_1(u_0^2-u_1^2)+r_1(u_0-u_1)+q_2(u_1^2-u^2)+r_2(u_1-u)}, \forall u \in [u_1 \;\; u_2) \tag{14}$$

Following the same procedure as above $\dot{u}$ at the n^{th} subinterval i.e between grid points u_{n-1} and u_n can be represented as

$$\dot{u}(u) = \dot{s}_0 e^{q_1(u_0^2-u_1^2)+r_1(u_0-u_1)\ldots+q_n(u_{n-1}^2-u^2)+r_n(u_{n-1}-u)}, \forall u \in [u_{n-1} \;\; u_n) \tag{15}$$

$\dot{u}$ at any arbitrary grid point can be represented using (16) as

$$\dot{u}(u_n) = \dot{s}_0 e^{q_1(u_0^2-u_1^2)+r_1(u_0-u_1)\ldots+q_n(u_{n-1}^2-u_n^2)+r_n(u_{n-1}-u_n)} \tag{16}$$

Using (16) it is easy to verify that $\ddot{u}$ at n^{th} subinterval can be expressed as

$$\ddot{u}(u) = -(2q_n u + r)\dot{u}^2(u), \forall u \in [u_{n-1} \;\; u_n) \tag{17}$$

Hence $\ddot{u}$ at any arbitrary grid point can be written as

$$\ddot{u}(u_n) = -(2q_n u + r_n)\dot{u}^2(u_n) \tag{18}$$

We are now in a position to utilise the form of $\dot{u}$ and $\ddot{u}$ to reformulate the velocity and acceleration level constraints (5)-(9) and the objective function (4).

A. Reformulating Velocity Level Constraints

Evaluating the velocity level constraint (5) at the grid points, substituting corresponding expression of $\dot{u}$ and taking logarithmic transformation, results in following n inequalities and one equality constraint.

$$C_{vi} = \log(\dot{u}(u_i)) - \log(s(u_i)) \leq 0, \quad \forall i = 1, 2, 3....n \tag{19}$$

where

$$s(u) = \frac{v_{max}(u)}{\sqrt{(x'(u))^2 + (y'(u))^2 + (z'(u))^2}} \tag{20}$$

$$C_{veq} = \log(\dot{u}(u_f)) - \log(\dot{s}_f) = 0 \tag{21}$$

The first term in n velocity level inequality constraints C_{vi} and one equality constraint C_{veq} is a logarithm of an exponential function $\dot{u}$. Hence it is linear in terms of variables $q_1, r_1, q_2, r_2, q_3, r_3..q_n, r_n, q_{n+1}, r_{n+1}$. Moreover the form of $\dot{u}$ ensures that not all variables appear in all the inequalities resulting in a sparse structure of the constraint matrix. In fact the inequality constraint matrix has a lower diagonal like structure. The remaining terms in C_{vi} and C_{veq} are just constants and hence these constraints are linear in terms of variable q_i and r_i.

B. Reformulating Acceleration Level Constraints

The boundary constraints on $\ddot{u}$ can be simplified in the following manner

$$C_{aeq} = \begin{cases} \ddot{u}(u_0) = \ddot{s}_0 \Rightarrow -(2q_1 u_0 + r_1)\dot{u}^2(u_0) = \ddot{s}_0 \\ \qquad \Rightarrow -(2q_1 u_0 + r_1) - \frac{\ddot{s}_0}{\dot{s}_0^2} = 0 \\ \ddot{u}(u_f) = \ddot{s}_f \Rightarrow -(2q_{n+1} u_f + r_{n+1})\dot{u}^2(u_f) = \ddot{s}_f \\ \qquad \Rightarrow -(2q_{n+1} u_f + r_{n+1}) - \frac{\ddot{s}_f}{\dot{s}_f^2} = 0 \end{cases}$$

The continuity requirement between $\ddot{u}$ of the adjacent intervals can be represented in terms of following equalities.

$$C_{cont} = \begin{cases} -(2q_i u_i + r_i)\dot{u}(u_i)^2 = -(2q_{i+1} u_i + r_{i+1})\dot{u}(u_i)^2 \\ \qquad \Rightarrow -(2q_i u_i + r_i) + (2q_{i+1} u_i + r_{i+1}) = 0 \\ \qquad \forall i = 1, 2, 3...n \end{cases} \tag{22}$$

It can be seen that the boundary constraints on $\ddot{u}$ i.e C_{aeq} and the continuity constraints C_{cont} are all linear in terms of variable q_i and r_i.

Similar to velocity level constraints, evaluating acceleration bound constraint (7) at the grid points and taking logarithmic transformation results in the following inequalities

$$\log(\dot{u}(u_i)) + 0.5 f_{xi}(q_i, r_i) - 0.5 \log(\ddot{x}_{max}(u_i)) \leq 0, \quad \forall i = 1, 2, ..n \tag{23}$$

$$f_{xi}(q_i, r_i) = \log(|x^{''}(u_i) - (2q_i u_i + r_i)x^{'}(u_i)|) \quad (24)$$

The first term in each acceleration level constraints are same as the velocity level constraints. Hence the first term is linear. The second term in the i^{th} constraint is a logarithmic function in terms of the i^{th} optimization variable q_i, r_i. This logarithmic non-linearity has a special structure. To understand this further consider the surface of the i^{th} constraint shown in figure 2(a). As can be seen from the figure, it consists of two concave surfaces on either side of the line passing through the set of points q^*, r^* satisfying the following equation. q^*, r^* as given by 25 are the singular points of the logarithmic non-linearity.

$$x^{''}(u_i) - (2q^* u_i + r^*)x^{'}(u_i) = 0 \quad (25)$$

As shown in the Appendix the logarithmic non-linearity in the i^{th} constraint is actually quasi-convex in nature which allows us to make the following final modification to the acceleration bound constraints (23)

In (26), the slack variables $\mathcal{J}_{xi}$ are used to split each i^{th} acceleration constraint into three separate inequalities. The first two inequalities in each constraint is linear. The third inequality has a linear term followed by a purely concave non-linearity. Concave non-linearity can be represented as a difference of two convex functions and hence the constraint has a simple *difference of convex* (**DC**) form. Similar expressions can be derived for y and z component as C_{ayi} and C_{azi}.

$$C_{axi} = \begin{cases} f_{xi}(q_i, r_i) \leq \log(\mathcal{J}_{xi}) \\ \Rightarrow x^{''}(u_i) - (2q_i u_i + r_i)x^{'}(u_i) - \mathcal{J}_{xi} \leq 0 \\ \Rightarrow -x^{''}(u_i) + (2q_i u_i + r_i)x^{'}(u_i) - \mathcal{J}_{xi} \leq 0 \\ \log(\dot{u}(u_i)) + 0.5\log(\mathcal{J}_{xi}) - 0.5\log(\ddot{x}_{max}(u_i)) \leq 0 \\ \forall i = 1, 2, 3...n \end{cases} \quad (26)$$

C. Reformulating Minimum Time Optimization Problem

Using (19)-(26), the minimum time optimization framework can be reformulated as

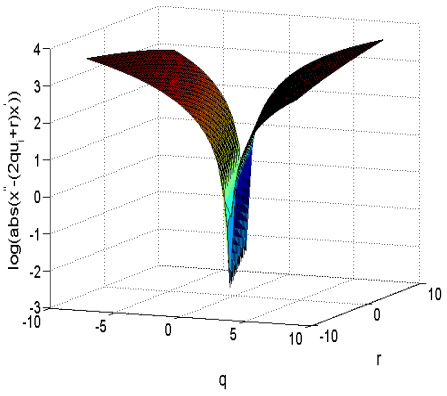
$$\min J_{new} \quad (27)$$
$$s.t, C_{vi} \leq 0, C_{axi} \leq 0, C_{ayi} \leq 0, C_{azi} \leq 0, \forall i = 1, 2, 3..n$$
$$C_{veq} = 0, C_{aeq} = 0, C_{cont} = 0$$

where

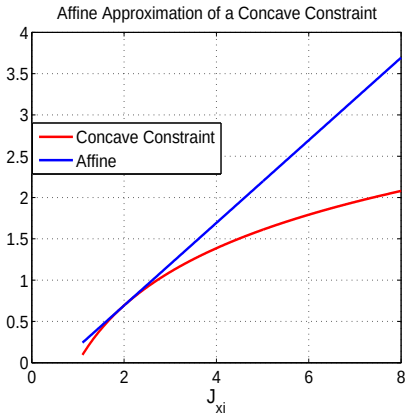
$$J_{new} = \sum_{i=1}^{i=n} [\log(\dot{u}(u_i)) - \log(s(u_i))]^2 \quad (28)$$

J_{new} as given in (28) has been obtained from (4) by approximating the integral through summation and by noting that minimizing an objective $(ax-b)^2$ is similar to minimizing $(\log(x) - log(\frac{b}{a}))^2$.

We next described the solution procedure for the optimization problem (27).



(a)



(b)

Fig. 2. (a): Plot showing the surface of logarithmic non-linearity $f_{xi}(.)$. As it can be seen, the surface consists of two concave parts around a continuum of singular points. It is easy to verify that the non-linearity is actually quasi-convex in nature. (b): Affine approximation of a concave constraints. Affine approximation acts as a global upper bound for the original concave non-linearity. It implies that feasible region of the optimization problem with approximate affine constraints is contained in the feasible region of the optimization with the original non-linear concave constraint.

IV. Solving the Optimization Problem

As shown in the previous section, the reformulated minimum time optimization problem has convex quadratic objective function with primarily linear constraints. The non-linearity has a purely concave form and thus the optimization problem is an instance of *difference of convex programming problem* **DCP** [15]. *Sequential convex programming* (**SCP**) [15] routine is an efficient choice for solving such problems since the notion of trust region usually associated with **SCP** is not relevant for **DCP** problems.

In **SCP** procedure, at each iteration, the convex part of the problem is preserved while the non-convex part is replaced with a convex approximation around the solution obtained at the previous iteration. The resulting convex problem can be solved very efficiently and quickly. The **SCP** procedure

is continued till some stopping criteria is met.

A. Convex Approximation

The non-convexity in optimization problem (27) arises out of purely concave logarithm functions in the acceleration constraints C_{axi}, C_{ayi} and C_{azi}. Let $\mathcal{J}_{xi}^k$, $\mathcal{J}_{yi}^k$ and $\mathcal{J}_{zi}^k$ represent the slack variables at the end of k^{th} iteration. At $(k+1)^{th}$ iteration, the concave part is replaced with following affine approximation around the solution obtained at the k^{th} iteration

$$\log(\mathcal{J}_{xi}) \simeq \log(\mathcal{J}_{xi}^k) + (\frac{1}{\mathcal{J}_{xi}^k}(\mathcal{J}_{xi} - \mathcal{J}_{xi}^k)) \tag{29}$$

Using (29), the non-linearity in the acceleration constraint at $k+1^{th}$ iteration is approximated by the following affine inequality.

$$\log(\dot{u}(u_i))+0.5(\log(\mathcal{J}_{xi}^k)+(\frac{1}{\mathcal{J}_{xi}^k}(\mathcal{J}_{xi}-\mathcal{J}_{xi}^k)))-0.5\log\ddot{x}_{max}(u_i) \leq 0 \tag{30}$$

Similar expressions can be written for y and z component as well. The affine approximation is shown in figure 2(b). As it can be seen from the figure, affine approximation to a concave non-linearity acts as a global upper bound. This implies that the feasible region of relaxed problem with approximate affine constraints is contained in that of the original optimization problem.

B. Approximate Sparse Quadratic Programme

The affine approximation presented above results in following sparse quadratic programme with only linear constraints at the $(k+1)^{th}$ iteration of **SCP**.

$$\min J_{new} \tag{31}$$
$$s.t, C_{vi} \leq 0, C_{axi}^{k+1} \leq 0, C_{ayi}^{k+1} \leq 0, C_{azi}^{k+1} \leq 0, \forall i = 1, 2, 3..n$$
$$C_{veq} = 0, C_{aeq} = 0, C_{cont} = 0$$

Based on above discussions the **SCP** procedure for solving optimization problem (27) can be summarized in following steps.

•1 Using the slack variable obtained at the k^{th} step $\mathcal{J}_{xi}^k$, $\mathcal{J}_{yi}^k$ and $\mathcal{J}_{zi}^k$, concave part of acceleration constraints are replaced with an affine approximation. It is to be noted that at $k = 0$, an initial guess for the slack variables needs to be provided.

•2 The **SCP** iteration continues till the velocity and acceleration constraints are satisfied and $|J_{new}^{k+1} - J_{new}^k| < \lambda$. Here λ is user defined small constant.

The **SCP** procedure converges within two to three iterations for the optimization problem (27).

We next present some implementation results to validate the proposed minimum time optimization framework.

V. Examples and Discussions

The proposed minimum time optimization framework was validated on various random paths. Results on one of those paths are shown in figures 3(a)-3(c). To highlight the effect of acceleration continuity constraints, we compare the proposed framework with our earlier work [16] which also computes time optimal motions along specified paths but without acceleration continuity constraints. In [16] the following form of scaling function was used which resulted in almost identical optimization structure as that of current work.

$$\dot{u} = pe^{-qu} \tag{32}$$

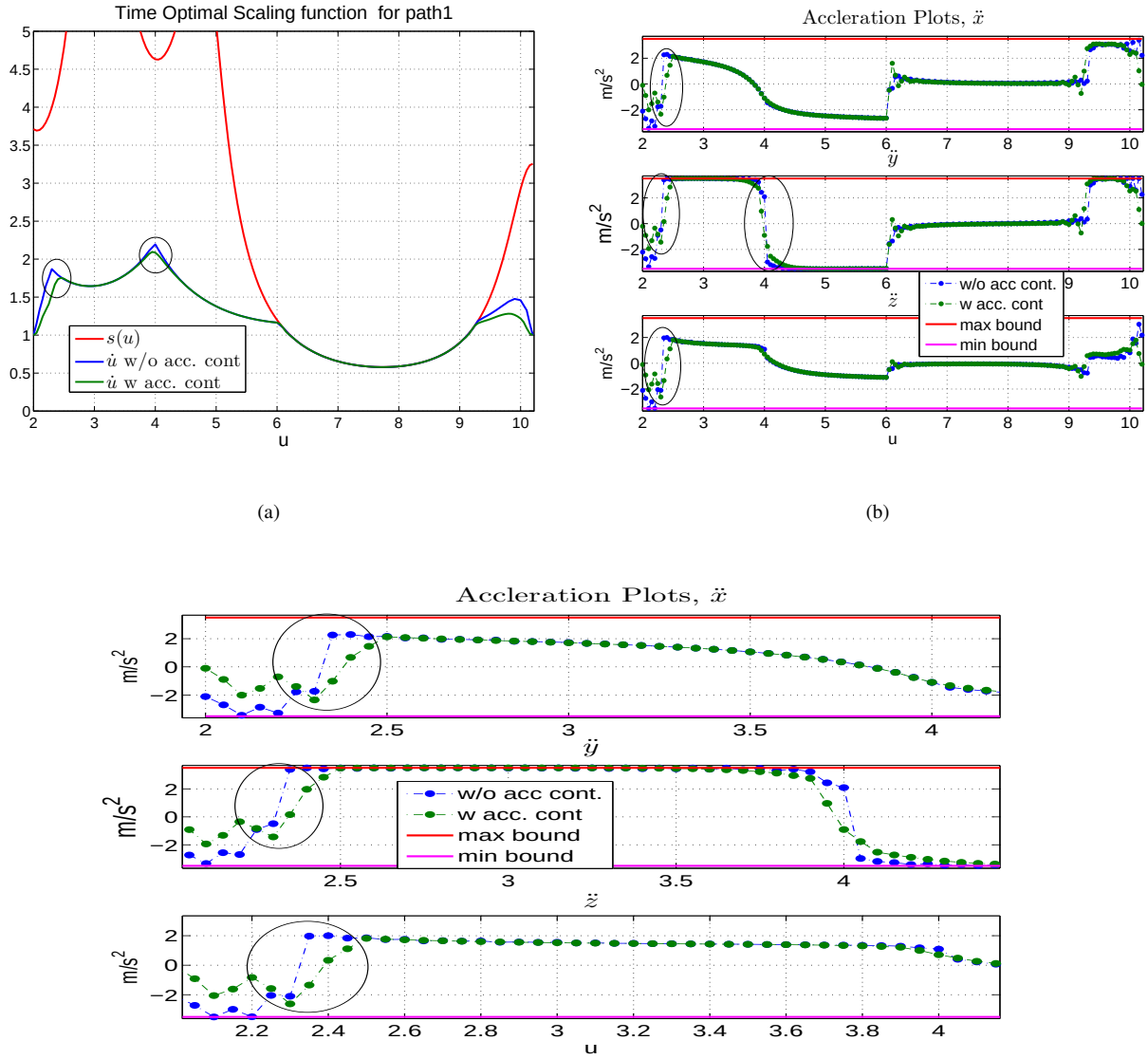
It can be seen that as compared to (10), (32) has one parameter less and hence the equality constraints pertaining to the acceleration continuity could not be enforced through (32). It is also straightforward to conjecture that the optimality (objective value) achieved through scaling function (32) without acceleration continuity constraints would be similar to that obtained with the proposed scaling function (10) with acceleration continuity constraints. This is because through (10) and as compared to (32) [16] , we are adding one parameter per discretized grid point in the optimization along with simultaneously increasing one affine equality constraints per grid. This conjecture was empirically verified in our simulations; can also be inferred from the figure 3(a), where the $\dot{u}$ profile achieved through (10) and (32) is almost same. Similar $\dot{u}$ profile implies similar objective value (28).

Time optimal motions are characterized by the saturation of either velocity or acceleration level constraints at any given instant. This is followed quite closely in figures 3(a)-(3(b). For example, in figure (3(b)), in the interval $u \in [2.2 \;\; 4.0]$ the y component of the acceleration constraint is saturated at the maximum limit, while in the interval $u \in [4.0 \;\; 6.0]$, it is saturated at the minimum limit. Similarly from figure 3(a), it can be seen that in the interval $u \in [6.0 \;\; 9.0]$, the $\dot{u}$ profile lies on the $s(u)$ curve, suggesting the saturation of velocity level constraints (19). Finally, the y component of the acceleration component is again saturated in the interval $u \in [9.1 \;\; 10.10]$.

The above described characteristic of constraint saturation, forces the acceleration profile to keep switching between the maximum and minimum bound. As can be seen from the figures 3(b) and 3(c), that without the acceleration continuity constraints, this switching is abrupt and discontinuous. For example, consider figure 3(b) and 3(c) where the x component of the acceleration jumps from $-1.729m/s^2$ to $2.263m/s^2$, y component jumps from $-0.491m/s^2$ to $3.387m/s^2$ and z component jumps from $-2.098m/s^2$ to $1.968m/s^2$ between the interval $u = 2.3$ and $u = 2.35$, if the acceleration continuity is not enforced. However, in contrast the continuous acceleration profile varies between $-2.34m/s^2$ and $-1.016m/s^2$ for the x component, between $-1.438m/s^2$ and $0.517m/s^2$ for the y component and between $-2.615m/s^2$ and $-1.3417m/s^2$, for the z component. As stated in [9], a smooth acceleration switching is essential for high fidelity trajectory tracking. Another important thing to note from figures 3(a) is that continuity in $\ddot{u}$ also results in a smoother $\dot{u}$ profile which in turn translates to smoother velocity profiles.

A. Robustness of *SCP* procedure

Figure 4(a) and 4(b) show the objective value at various iterations of **SCP** for various initial guesses of the slack



(a)

(b)

(c)

Fig. 3. (a) and (b): Implementation of the proposed optimization framework on two example paths. For comparative purposes, we implement the minimum time optimization framework proposed in [16] which has an identical optimization structure as the current work but without the acceleration continuity constraints. Figures show that acceleration continuity constraints ensure a smooth switching of acceleration profile between the maximum and minimum bound. Acceleration profile without continuity constraints have abrupt and discontinuous jumps in the magnitude.

variable. It was observed that even with large variations in initial guesses, the **SCP** always converged to almost the same objective value in same number of iterations. It can be seen that although there is a large variation in the objective value after the first iteration, the objective value converges to almost the same value after second or third iterations for all the initial guesses.

B. Computational Aspects of the Proposed Minimum Time Optimization Framework

The proposed optimization framework was implemented in MATLAB using the open source tool box OPTI [17] on a standard PC with an Intel Core 2 Duo CPU 2.93 Ghz with 4.0 GB of RAM. On an average an optimization with 100 grid points, 600 variables and 2000 inequality constraints could be solved at 600 ms. Much of the speed is attributed to the fact the constraint matrix has an extremely sparse structure. The run time of the optimization can be significantly improved by prototyping it in Python using solvers like CVXOpt [8].

VI. Conclusions and Future Work

In this paper, we solved the problem of time optimal control along specified paths with acceleration continuity constraints. We used the concept of *non-linear time scaling* to represent time varying controls as parametrized exponential functions. We showed that such representation leads to

a very simple optimization structure with primarily linear constraints. The non-linearity have a quasi-convex form which can be easily reformulated to a simple *difference of convex* form. This highly efficient optimization structure is an improvement over the current state of the art frameworks like [12] and [13] which introduces acceleration continuity constraints through highly non-linear and non-convex optimization frameworks.

We showed how the proposed optimization structure can be efficiently solved through *sequential convex programming* technique where at each iteration a sparse quadratic programme is solved. The optimization converged in two to three iterations in most cases. The effect of acceleration continuity was clearly highlighted in the optimization structure. Moreover it was also explained how the proposed scaling function allows us to introduce acceleration continuity with little reduction in optimality.

Future work concerns with extending the framework to include force and torque constraints. An optimization framework with almost identical structure as the current proposed one has already been developed. We are currently looking to apply the time optimal framework to various differentially constrained motion planning problems.

VII. Appnedix

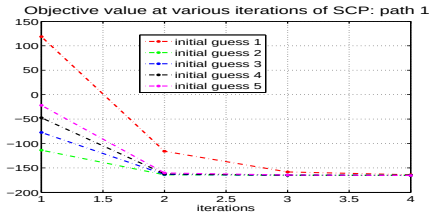
A. Proof of Quasi-Convexity of $f_{xi}(.)$, $f_{yi}(.)$ and $f_{zi}(.)$

Lemma 7.1: $f_{xi}(q,r) = \log(|x^{''}(u_i) - (2q_iu_i + r_i)x^{'}(u_i)|)$ is quasi-convex.

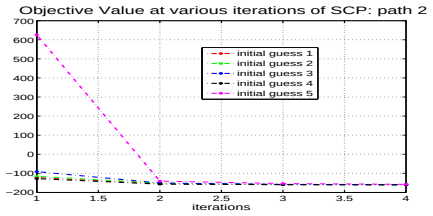
Proof: A multivariate function f defined over a convex set S is quasi-convex if $Q^{\tau} = \{q, r \in S : f \leq \tau\}$ is convex for every τ. Thus to show quasi-convexity, we compute the solution set of $f(q,r) \leq \tau$ and show that it is convex. The solution set is defined by the following inequalities.

$$f(q,r) \leq \tau = \begin{cases} x^{''}(u_i) - (2q_iu_i + r_i)x^{'}(u_i) - e^{\tau} \leq 0 \\ -x^{''}(u_i) + (2q_iu_i + r_i)x^{'}(u_i) - e^{\tau} \leq 0 \end{cases} \tag{33}$$

For a given τ, (33), is a set of linear inequalities and hence it's solution set is convex and consequently $f_{xi}(q,r)$ is quasi-convex. Proof for $f_{yi}(.)$ and $f_{zi}(.)$ is similar. ■



(a)



(b)

Fig. 4. (a) and (b): Figures show that even with large variations in the initial guesses, **SCP** converged to the same objective value in almost the same number of iterations.

References

[1] Kant, Kamal, and Steven W. Zucker. "Toward efficient trajectory planning: The path-velocity decomposition." The International Journal of Robotics Research 5.3 (1986): 72-89.

[2] Singh, Arun Kumar, and K. Madhava Krishna. "Reactive Collision Avoidance for Multiple Robots by Non Linear Time Scaling."

[3] Bobrow, James E., Steven Dubowsky, and J. S. Gibson. "Time-optimal control of robotic manipulators along specified paths." The International Journal of Robotics Research 4.3 (1985): 3-17.

[4] Shiller, Zvi. "On singular time-optimal control along specified paths." Robotics and Automation, IEEE Transactions on 10.4 (1994): 561-566.

[5] Kunz, Tobias, and Mike Stilman. "Time-Optimal Trajectory Generation for Path Following with Bounded Acceleration and Velocity." Proceedings of the 2012 Robotics: Science and Systems Conference. Vol. 8. 2012.

[6] Hauser, Kris. "Fast interpolation and time-optimization on implicit contact submanifolds." Robotics: Science and Systems. 2013.

[7] Verscheure, Diederik, et al. "Time-optimal path tracking for robots: A convex optimization approach." Automatic Control, IEEE Transactions on 54.10 (2009): 2318-2327.

[8] Dahl, Joachin, and L. Vandenberghe. "Cvxopt: A python package for convex optimization." Proc. eur. conf. op. res. 2006.

[9] Guarino Lo Bianco, Corrado, and Oscar Gerelli. "Online trajectory scaling for manipulators subject to high-order kinematic and dynamic constraints." Robotics, IEEE Transactions on 27.6 (2011): 1144-1152.

[10] Diehl, M., Debrouwere, F., De Schutter, J., Dinh, Q. T., Swevers, J., Pipeleers, G., and Van Loock, W. (2013). Time-Optimal Path Following for Robots With Convex-Concave Constraints Using Sequential Convex Programming. Ieee Transactions On Robotics, 29(EPFL-ARTICLE-196131), 1485-1495.

[11] Hehn, Markus, and Raffaello DAndrea. "Quadrocopter trajectory generation and control." IFAC World Congress. Vol. 18. No. 1. 2011.

[12] Costantinescu, D., and E. A. Croft. "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths." Journal of robotic systems 17.5 (2000): 233-249.

[13] Bouktir, Y., M. Haddad, and T. Chettibi. "Trajectory planning for a quadrotor helicopter." Control and Automation, 2008 16th Mediterranean Conference on. IEEE, 2008.

[14] Singh, Arun Kumar, K. Madhava Krishna, and Srikanth Saripalli. "Planning trajectories on uneven terrain using optimization and non-linear time scaling techniques." Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.

[15] Boyd, Stephen. "Sequential convex programming." Lecture Notes, Stanford University (2008).

[16] Bharath Gopalakrishnan, Arun Kumar Singh and K Madhava Krishna " Time Scaled Collision Cone based Trajectory Optimization Approach for Reactive Planning in Dynamic Environments" to appear in *IROS 2014*.

[17] Currie, Jonathan, and David I. Wilson. "OPTI: lowering the barrier between open source optimizers and the industrial MATLAB user." Foundations of Computer-Aided Process Operations, Savannah, Georgia, USA (2012): 8-11.