Real-time Optimal Trajectory Planning for Robotic Manipulators with Functional Redundancy

Pål Johan From and Jan Tommy Gravdahl
Department of Engineering Cybernetics
Norwegian University of Science and Technology
7491 Trondheim, Norway
E-mail: from@itk.ntnu.no

***Abstract*—In this paper we show that by allowing a small orientation error in the specifications of the end effector orientation for spray paint applications, both speed and performance can be substantially improved. In previous publications, we have shown how a continuous set of orientations can be represented as a positive definiteness test on a given matrix. This allows us to cast the problem into a convex optimisation problem where the optimal orientation for each time step is to be found. The orientation which allows the manipulator to follow a given path as fast as possible and with constant speed is considered the optimal orientation. In this paper we show how to solve this problem for every time step. The solution is computationally fast and can be implemented in real time. Numerical examples are presented to verify the efficiency of the approach.**

I. Introduction

In a wide range of applications the orientation of a rigid body does not need to be restricted to one given frame but can be given as a continuous set of frames. The attitude of a satellite can for example be set so that the transmitter or receiver points approximately in the direction of the earth. Another example is the end effector of a robotic manipulator where an orientation error is allowed. This is motivated by the observation that a small orientation error does not affect the quality of the paint job. The speed at which the paint gun follows the path is far more critical to guarantee uniform paint coating.

In [1] the idea of introducing the paint quality as a constraint and minimise some additional cost function was presented. This opens for the possibility of allowing an orientation error in the specifications of the end-effector orientation in order to improve the performance and speed of the job, reduce torques and so on. It was shown in [2] that by allowing an orientation error in the end-effector configuration of a robotic manipulator, the speed and the quality of the job was improved. However, the orientation error was chosen intuitively, and the approach presented was not suitable for implementation in an optimisation algorithm.

In [3] the orientation error constraints were transformed into a test of positive definiteness of a matrix. For different types of orientation errors a suitable matrix was found and it was shown that positive definiteness of these matrices is equivalent to an orientation that satisfies the given restrictions on the orientation. Further it was shown how to cast the restrictions on the orientation into Linear Matrix Inequalities (LMIs).

In this paper we show how to solve the optimisation problem when the constraints are written on the form of barrier functions. We show that by applying the gradient method the optimal solution is found in just a few iterations. When the optimal solution is given by a cost function representing two conflicting orientations of the end effector, 10-15 iterations are needed. This requires less than 1ms of computation time on a standard personal computer. The optimal orientation can be computed on-line with only the current position, the centre point of the surface to be painted and the allowed orientation error as inputs. Several examples where an orientation error is allowed for spray paint applications are shown to verify the efficiency of the approach and we show that the time required to paint a surface can be decreased substantially.

II. Representing Rotations

Most of the fundamental principles of rotation were presented in two papers by Leonhard Euler in 1775 [5]. The first paper shows that any rotation can be accomplished by a sequence of three rotations about the coordinate axes. In the second paper, Euler states that any orientation can be represented by a rotation of some angle ϕ about a fixed axis $\boldsymbol{n}$ and that the composition of two rotations is itself a rotation.

A. The Unit Quaternion

The unit quaternion representation closely relates to the results presented in Euler's second paper. A good introduction to quaternions is found in [6]. Any positive rotation ϕ about a fixed unit vector $\boldsymbol{n}$ can be represented by the four-tuple

$$Q = \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix}, \tag{1}$$

where $q_0 \in \mathbb{R}$ is known as the scalar part and $\boldsymbol{q} \in \mathbb{R}^3$ as the vector part. $Q(\phi, \boldsymbol{n})$ is written in terms of ϕ and $\boldsymbol{n}$ by

$$q_0 = \cos\left(\frac{\phi}{2}\right), \qquad \boldsymbol{q} = \sin\left(\frac{\phi}{2}\right)\boldsymbol{n}. \tag{2}$$

Q is a quaternion of unit length and denoted a *unit quaternion*. Henceforth, all quaternions have unit length if not other is stated. Let $Q_P = \begin{bmatrix} p_0 & \boldsymbol{p}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$. A multiplication of two quaternions is given by a quaternion product and is written in vector algebra notations as

$$Q_P * Q = \begin{bmatrix} p_0 q_0 - \boldsymbol{p} \cdot \boldsymbol{q} \\ p_0 \boldsymbol{q} + q_0 \boldsymbol{p} + \boldsymbol{p} \times \boldsymbol{q} \end{bmatrix}. \tag{3}$$

The cross product implies that quaternion multiplication is not commutative. Let $Q_P = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix}^\mathsf{T}$ and $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$. Then the quaternion product is written as

$$Q_P * Q = \begin{bmatrix} p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3 \\ p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2 \\ p_0q_2 + p_2q_0 + p_3q_1 - p_1q_3 \\ p_0q_3 + p_3q_0 + p_1q_2 - p_2q_1 \end{bmatrix}. \tag{4}$$

The quaternion product of two unit quaternions is a unit quaternion. By the definition of the quaternion the quaternions Q and $-Q$ produce the same rotation. This is referred to as the dual covering. The quaternion identity is given by $Q_I = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$.

A pure quaternion is a quaternion with zero scalar part. A vector, $\bar{\boldsymbol{v}} = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T}$ is represented by a pure quaternion

$$\boldsymbol{v} = \begin{bmatrix} 0 \\ \bar{\boldsymbol{v}} \end{bmatrix}. \tag{5}$$

The conjugate of a quaternion is defined as

$$Q^* = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{bmatrix}^\mathsf{T}. \tag{6}$$

B. *Quaternions and Rotations*

Let a vector $\bar{\boldsymbol{v}}_1$ be represented by the pure quaternion $\boldsymbol{v}_1$. This vector can be rotated ϕ radians around the axis $\boldsymbol{n}$ by

$$\boldsymbol{v}_2 = Q * \boldsymbol{v}_1 * Q^*. \tag{7}$$

Every vector $\bar{\boldsymbol{v}} \in \mathbb{R}^3$ can be represented by a pure quaternion, hence $\boldsymbol{v}$ is not necessarily a unit quaternion. The quaternion, $Q(\phi, \boldsymbol{n})$, however, is unitary. This represents the angle and the axis that the vector $\bar{\boldsymbol{v}}_1$ rotates about. The resulting vector, $\bar{\boldsymbol{v}}_2$, is then of the same length as $\bar{\boldsymbol{v}}_1$ if and only if Q is a unit quaternion. The quaternion representation also leads to a useful formula for finding the shortest rotation from one orientation to another. Let Q_P and Q be two orientations. Then, by taking

$$E = Q_P^* * Q, \tag{8}$$

E will rotate Q_P into Q by the shortest rotation.

Note that Equation (8) rotates one frame into another frame. By a *frame* it is meant a coordinate system in $\mathbb{R}^3$ using Cartesian coordinates. One frame with respect to another frame represents three degrees of freedom and is referred to as *orientation*. The inertial frame is denoted $\mathcal{F}_I$ and the frame that corresponds to the inertial frame by a rotation Q from the inertial frame is denoted $\mathcal{F}_Q$. Equation (7) rotates one vector into another vector and has two degrees of freedom (e.g. longitude and latitude) [7]. A unit vector with respect to a unit reference vector is referred to as *direction*. Henceforth, the main concern is with the direction of the central axis, which is assumed to be the body frame z-axis of the end effector.

C. *Rotation Sequences*

In this paper, the orientation is represented by a rotation sequence of three rotations about the unitary axes. The ZYZ-sequence is given by first a rotation α about the z-axis followed by a rotation β about the new y-axis. This describes the direction of the z-axis. The last degree of freedom is given by the rotation γ about the z-axis. When the sequence is given, a one-to-one mapping[1] between (α, β, γ) and the quaternion $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ can be found whenever $\beta \neq 0$.

Given a quaternion Q. Then α, β and γ from the ZYZ-sequence are found by [9]

$$\alpha = \arctan2\left(\frac{q_2q_3 - q_0q_1}{q_0q_2 + q_1q_3}\right), \tag{9}$$

$$\beta = 2\arcsin\sqrt{q_1^2 + q_2^2}, \tag{10}$$

$$\gamma = \arctan2\left(\frac{q_2q_3 + q_0q_1}{q_0q_2 - q_1q_3}\right). \tag{11}$$

The following relations are also used in the following:

$$\alpha = \arctan(\frac{q_3}{q_0}) - \arctan(\frac{q_1}{q_2}), \tag{12}$$

$$\gamma = \arctan(\frac{q_3}{q_0}) + \arctan(\frac{q_1}{q_2}), \tag{13}$$

and hence

$$\alpha + \gamma = 2\arctan(\frac{q_3}{q_0}). \tag{14}$$

III. Orientation Error Constraints as Barrier Functions

In [3] the orientation error constraints were transformed into LMIs. In this section we use the same basic idea to transform the orientation error constraints into barrier functions.

A. *Cone*

Assume that one would like to restrict the z-axis of $\mathcal{F}_Q$ to point in approximately the same direction of the z-axis of the inertial frame $\mathcal{F}_I$. This can be visualised by a cone of directions and restricted by $|\beta| \leq \beta_{lim}$ where $0 \leq \beta_{lim} \leq \pi$. The orientation error β can be found from q_1 and q_2 from Equation (10). Due to this observation, a test to verify if the z-axis of $\mathcal{F}_Q$ does not deviate from the z-axis of $\mathcal{F}_I$ by more than β_{lim} is given in the following.

Proposition 3.1: Given a restriction in the orientation error, β_{lim}. Then the z-axis of $\mathcal{F}_Q$ rotated by $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ from the inertial frame $\mathcal{F}_I$ lies within the restrictions given by β_{lim} if and only if

$$\eta^2 - q_1^2 - q_2^2 \geq 0 \tag{15}$$

where $\eta = \sin(\frac{\beta_{lim}}{2})$ and $0 \leq \beta_{lim} \leq \pi$.

Proof:

$$\begin{aligned} \eta^2 - q_1^2 - q_2^2 &\geq 0 \\ \eta^2 &\geq q_1^2 + q_2^2 \\ \eta &\geq \sqrt{q_1^2 + q_2^2} \\ \sin(\frac{\beta_{lim}}{2}) &\geq \sqrt{q_1^2 + q_2^2} \end{aligned} \tag{16}$$

[1]If the dual covering of the quaternion is taken into account, a one-to-two mapping can be found.

As $0 \leq \sqrt{q_1^2+q_2^2} \leq 1 \Rightarrow 0 \leq \arcsin\sqrt{q_1^2+q_2^2}$, the following holds

$$0 \leq 2\arcsin\sqrt{q_1^2+q_2^2} \leq \beta_{lim}. \tag{17}$$

Then Equation (10) concludes the proof as

$$0 \leq \beta \leq \beta_{lim}. \tag{18}$$

■

We will see that (15) can easily be transformed into a barrier function. By writing the constraint on this form the optimisation problems presented later can be solved very efficiently. Note that the restrictions in Proposition 3.1 are on the directions of the z-axis only and that rotations about the z-axis itself are not restricted (the pointing task). This is an important property that will be used in the following.

B. Restriction on the Rotation about the Central Axis

In the following a constraint on the orientation about the central axis is given. Assume that the x-axis points in the direction of the velocity and that it is desired that the body frame x-axis points in approximately the direction of the x-axis of the reference orientation. Again consider the ZYZ-sequence. In the case when no orientation error is allowed for the direction of the central axis, the constraint on the rotation about the central axis is given trivially by $|\gamma| < c_{lim}$, where c_{lim} is the maximum allowed orientation error of the x-axis. For the ZYZ-sequence the direction of the x-axis is given by both α, β and γ. Assume that the orientation error of the direction of the z-axis is restricted as in the previous section. When this is restricted to be relatively small, the error in the direction of the x-axis can be approximated by the error in the orientation about the central axis. This error is written as

$$\epsilon = \alpha + \gamma. \tag{19}$$

This leads to the following result.

Proposition 3.2: Assume that the orientation error of the direction of the z-axis is small and the orientation error about the central axis is restricted to ϵ_{lim}. Then the x-axis of $\mathcal{F}_Q$ rotated by $Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\mathsf{T}$ from the inertial frame $\mathcal{F}_I$ lies within the restrictions given by ϵ_{lim} if and only if

$$\kappa^2 - \frac{q_3^2}{q_0^2} \geq 0 \tag{20}$$

where $\kappa = \tan\left(\frac{\epsilon_{lim}}{2}\right)$.

Proof:

$$\kappa^2 - \frac{q_3^2}{q_0^2} \geq 0$$

$$\kappa^2 \geq \frac{q_3^2}{q_0^2}$$

$$\kappa \geq \left|\frac{q_3}{q_0}\right|$$

$$\tan\left(\frac{\epsilon_{lim}}{2}\right) \geq \left|\frac{q_3}{q_0}\right|$$

$$\epsilon_{lim} \geq \left|2\arctan\left(\frac{q_3}{q_0}\right)\right| \tag{21}$$

Then Equation (14) concludes that

$$\epsilon_{lim} \geq |\epsilon| \tag{22}$$

where ϵ is given by Equations (14) and (19). ■

C. Restriction on the Direction of the x-axis

Alternatively, we can restrict the direction of the x-axis directly. Similarly to Equation (15), the requirement that the body frame x-axis is to point in the direction of the inertial frame x-axis is given by

$$\xi^2 - q_2^2 - q_3^2 \geq 0 \tag{23}$$

where $\xi = \sin\left(\frac{\beta_{lim}}{2}\right)$. Thus, if Equation (23) is satisfied, this will guarantee that the x-axis of $\mathcal{F}_Q$ does not deviate from the x-axis of $\mathcal{F}_I$ by more than β_{lim} radians.

Also note that the results presented are not restricted to the global reference frame $\mathcal{F}_I$. Assume that the direction of the body frame x-axis is to point in an arbitrary direction given by the direction of the x-axis of $Q_d = \begin{bmatrix} d_0 & d_1 & d_2 & d_3 \end{bmatrix}^\mathsf{T}$. In order to apply the restriction given by (23), but to the direction of the x-axis of $\mathcal{F}_{Q_d}$ and not that of $\mathcal{F}_I$, Q is transformed back into the inertial frame and the test is performed on the transformed quaternion

$$Q_t = \begin{bmatrix} t_0 \\ t_1 \\ t_2 \\ t_3 \end{bmatrix} = Q_d^* * Q = \begin{bmatrix} * \\ * \\ -d_2q_0 + d_0q_2 - d_3q_1 + d_1q_3 \\ -d_3q_0 + d_0q_3 - d_1q_2 + d_2q_1 \end{bmatrix}. \tag{24}$$

D. Barrier Functions

The constraints presented in this section can be augmented to the objective function by letting the barrier function increase to infinity as the orientation error approaches the orientation error limit. The constraint in Proposition 3.1 can be written as a logarithmic barrier function and augmented to the objective function so the optimisation problem becomes

$$\text{minimise} \quad \phi(x) = F(Q) - \log(\eta^2 - q_1^2 - q_2^2). \tag{25}$$

where $F(Q)$ is the objective function to be minimised.

For the constraint given in Equation (23), the logarithmic barrier function is augmented to the objective function by

$$\text{minimise} \quad \phi(x) = F(Q) - \log(\xi^2 - q_2^2 - q_3^2). \tag{26}$$

If the orientation that best satisfies the restriction on the z-axis and the (rotated) x-axis is desired, the two constraints can be combined and the solution is found by

$$\text{minimise} \quad \phi(x) = -k_z\log(\eta^2 - q_1^2 - q_2^2) - k_x\log(\xi^2 - t_2^2 - t_3^2), \tag{27}$$

where $t_{2,3}$ are taken from (24) and $k_{z,x}$ weigh the importance of the direction of the z- and the x-axes.

Note that in this case, the constraints are treated as objective functions. The orientation that best satisfies two (in general conflicting) objectives is chosen as the optimal orientation. When two or more objectives can be written as logarithmic barrier functions, the optimal solution is in general found very efficiently.

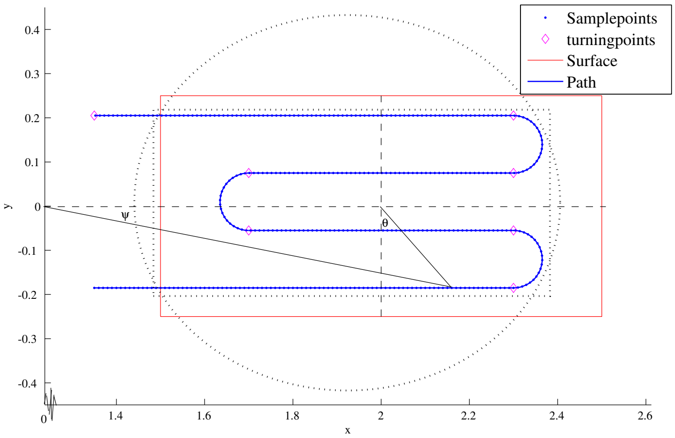

Fig. 1. The path of the tool centre point (TCP) in the xy-plane.

IV. Spray Painting

A. Conflicting Objective Functions

We now show an example where the direction of the z-axis is determined by two cone-shaped sets of orientations. The direction given by the two sets at each time step is in general conflicting and the solution is given by the minimum of a cost function of the sum of the two orientation errors.

Assume a manipulator that is to paint a surface in the xy-plane by following the path in Figure 1. There are two main criteria that will guarantee uniform paint coating, the orientation of the spray gun with respect to the surface and the velocity of the paint gun. The first restriction is ensured by the constraint

$$\eta^2 - q_1^2 - q_2^2 > 0 \tag{28}$$

where $\eta = \sin\left(\frac{\beta_{lim}}{2}\right)$, and β_{lim} is the maximum allowed orientation error for which the quality of the paint job is satisfying. In general, an orientation error of $5^\circ - 20^\circ$ guarantees uniform paint coating. For a manipulator which is to paint a surface in the xy-plane, this restriction can be visualised by a cone. The cross section of this cone is given by the circle in Figure 1.

The second important factor that determines the quality of the paint job is constant velocity. In general, the velocity of the end effector is very critical and want the manipulator to follow the path with constant velocity, also when high accelerations are needed, as in turns. Hence we want to choose an orientation error which allows the manipulator to follow the path as fast as possible with constant velocity. For a manipulator with a structure similar to the one in Figure 2 to follow the path in Figure 1, the work load of the main axes is far higher than the wrist axes. Hence, we assume that by decreasing the torques of the main axes, the manipulator can follow the path with a higher velocity. We decrease the displacement of the main axes by forcing the centre of the wrist towards the centre of the surface, which will make the main axes move less.

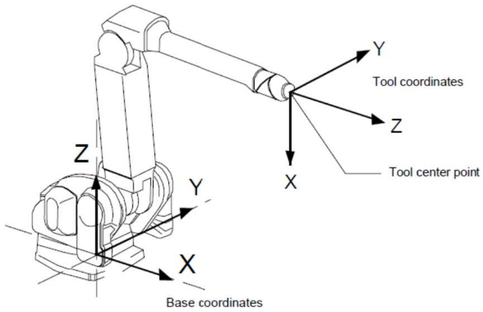

Fig. 2. General structure of a robotic manipulator.

Assume we want to paint the surface in the xy-plane with a constant distance z_{des} between the tool and the surface. Let c be the vector from the centre of the surface, at height z_{des}, denoted p_{cent}, to the current position p_{tcp} on the surface

$$c = p_{tcp} - p_{cent}. \tag{29}$$

This is the direction of the end effector for which the main axes don't need to move at all, i.e. pure rotation of the wrist. This is chosen as the desired direction of the paint gun when the orientation error is not considered.

Let the quaternion describing the desired direction of the z-axis be given by Q_d. Then the set of orientations for which the z-axis points in approximately the direction of the z-axis of Q_d is found by writing

$$Q_p = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = Q_d^* * Q = \begin{bmatrix} * \\ -d_1q_0 + d_0q_1 - d_2q_3 + d_3q_2 \\ -d_2q_0 + d_0q_2 - d_3q_1 + d_1q_3 \\ * \end{bmatrix} \tag{30}$$

The constraint that forces the end effector to point in the direction of Q_d with a maximum orientation error α_{lim} is given by Proposition 3.1 as

$$\xi^2 - p_1^2 - p_2^2 > 0 \tag{31}$$

where $\xi = \sin\left(\frac{\alpha_{lim}}{2}\right)$. Note that there are infinitely many quaternions for which the end effector points in the desired direction. What orientation that is chosen affects the solution. This freedom is treated as the pointing task problem and is dealt with separatetly to improve performance further.

B. Spray painting

We now turn to the problem of spray painting the surface in the xy-plane in Figure 1, also addressed in [2]. The surface is to be painted from above, so the set representing the orientation error needs to be rotated 180° so that it points downwards. This can be done similar to Equation (30) with $Q_d = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^\mathsf{T}$ or, the approach that we will take here, instead of the restriction

$$\eta^2 \geq q_1^2 + q_2^2, \tag{32}$$

which we used in Section III-A, we write

$$\eta^2 \leq q_1^2 + q_2^2, \tag{33}$$

and replace $\beta_{lim} \leftarrow \pi - \beta_{lim}$ in $\eta = \sin(\frac{\beta_{lim}}{2})$. This will guarantee that the set of orientations points in exactly the opposite direction of the set of Equation (31). Equation (27) then becomes

$$\begin{aligned}\phi &= k_{err}\phi_{err} + k_{tcp}\phi_{tcp} \\ &= -k_{err}\log(q_1^2 + q_2^2 - \eta^2) - k_{tcp}\log(\xi^2 - p_1^2 - p_2^2).\end{aligned} \tag{34}$$

V. The Gradient Method and Implementation

In this section we transform the constraints into barrier functions and show how to solve it by the gradient method. The cost function to be minimised is on the form

$$\begin{aligned}\phi &= k_{err}\phi_{err} + k_{tcp}\phi_{tcp} \\ &= -k_{err}\log(q_1^2 + q_2^2 - \eta^2) - k_{tcp}\log(\xi^2 - p_1^2 - p_2^2).\end{aligned} \tag{35}$$

Before we find the explicit expressions of the cost function and its gradient we find p_1^2 and p_2^2 in terms of Q from (30)

$$\begin{aligned}p_1^2 =&(-d_1q_0 + d_0q_1 - d_2q_3 + d_3q_2)^2 \\ =&d_1^2q_0^2 + d_0^2q_1^2 + d_2^2q_3^2 + d_3^2q_2^2 - 2d_0d_1q_0q_1 + 2d_1d_2q_0q_3 \\ &- 2d_1d_3q_0q_2 - 2d_0d_2q_1q_3 + 2d_0d_3q_1q_3 - 2d_2d_3q_2q_3\end{aligned} \tag{36}$$

$$\begin{aligned}p_2^2 =&(-d_2q_0 + d_0q_2 - d_3q_1 + d_1q_3)^2 \\ =&d_2^2q_0^2 + d_0^2q_2^2 + d_3^2q_1^2 + d_1^2q_3^2 - 2d_0d_2q_0q_2 + 2d_2d_3q_0q_1 \\ &- 2d_1d_2q_0q_3 - 2d_0d_3q_1q_2 + 2d_0d_1q_2q_3 - 2d_1d_3q_1q_3\end{aligned} \tag{37}$$

and

$$\begin{aligned}p_1^2 + p_2^2 =&(d_1^2 + d_2^2)q_0^2 + (d_0^2 + d_3^2)q_1^2 \\ &+ (d_0^2 + d_3^2)q_2^2 + (d_1^2 + d_2^2)q_3^2 \\ &- 2(d_0d_1 - d_2d_2)q_0q_1 - 2(d_1d_3 + d_0d_2)q_0q_2 \\ &- 2(d_0d_2 + d_1d_3)q_1q_3 - 2(d_0d_1 - d_2d_3)q_2q_3.\end{aligned} \tag{38}$$

The partial derivatives are found to be

$$\frac{\partial\phi_{err}}{\partial q_0} = 0, \quad \frac{\partial\phi_{err}}{\partial q_1} = -\frac{2q_1}{q_1^2 + q_2^2 - \eta^2},$$
$$\frac{\partial\phi_{err}}{\partial q_3} = 0, \quad \frac{\partial\phi_{err}}{\partial q_2} = -\frac{2q_2}{q_1^2 + q_2^2 - \eta^2},$$

and

$$\frac{\partial\phi_{tcp}}{\partial q_0} = -\frac{2(d_1^2 + d_2^2)q_0 - 2(d_0d_1 - d_2d_3)q_1 - 2(d_1d_3 + d_0d_2)q_2}{\xi^2 - p_1^2 - p_2^2},$$

$$\frac{\partial\phi_{tcp}}{\partial q_1} = -\frac{2(d_0^2 + d_3^2)q_1 - 2(d_0d_1 - d_2d_3)q_0 - 2(d_0d_2 + d_1d_3)q_3}{\xi^2 - p_1^2 - p_2^2},$$

$$\frac{\partial\phi_{tcp}}{\partial q_2} = -\frac{2(d_0^2 + d_3^2)q_2 - 2(d_1d_3 + d_0d_2)q_0 - 2(d_2d_3 - d_0d_1)q_3}{\xi^2 - p_1^2 - p_2^2},$$

$$\frac{\partial\phi_{tcp}}{\partial q_3} = -\frac{2(d_1^2 + d_2^2)q_3 - 2(d_0d_2 + d_1d_3)q_1 - 2(d_2d_3 - d_0d_1)q_2}{\xi^2 - p_1^2 - p_2^2}.$$

The gradient is then given by

$$\nabla\phi = \begin{bmatrix} k_{err}\frac{\partial\phi_{err}}{\partial q_0} + k_{tcp}\frac{\partial\phi_{tcp}}{\partial q_0} \\ k_{err}\frac{\partial\phi_{err}}{\partial q_1} + k_{tcp}\frac{\partial\phi_{tcp}}{\partial q_1} \\ k_{err}\frac{\partial\phi_{err}}{\partial q_2} + k_{tcp}\frac{\partial\phi_{tcp}}{\partial q_2} \\ k_{err}\frac{\partial\phi_{err}}{\partial q_3} + k_{tcp}\frac{\partial\phi_{tcp}}{\partial q_3} \end{bmatrix}. \tag{39}$$

The problem is solved by the gradient method

$$\phi^{k+1} = \phi^k - a\nabla\phi. \tag{40}$$

For a feasible initial condition and for a relatively small and constant step size a the stability and convergence of the method is good. Due to the low computational burden of this approach, a constant step is used instead of a search. This requires that a is chosen conservatively which may lead to slower convergence. When a proof of convergence is desired, a search algorithm should be applied for each iteration. A simple solution is to reduce a until the objective function is decreasing.

A. Pointing Task

By the approach described in the previous section, the orientation about the central axis (z-axis) is not determined. As only the direction of the central action is taken into account, the last degree of freedom can be chosen freely and used to improve performance further. Three approaches are described in the following.

1) From and Gravdahl (2007) [2]: The first approach presented is the intuitive approach given in [2]. The orientation about the central axis at point i is set as

$$\psi_i = k_\psi \operatorname{arctan2}\left(\frac{y_i}{x_i - x_{cent}}\right) \tag{41}$$

for $k_\psi \in (0, 1]$ and where x_i and y_i give the position of the end effector in the xy-plane and x_{cent} is the centre of the surface in the x-direction (see Figure 1). This will guarantee that the main axes move less.

2) x-axis Direction: A similar approach is to force the end effector x-axis to point in the direction of the base of the manipulator. By projecting the end-effector x-axis into the globally defined xy-plane and force this to point in the direction of the base will have approximately the same effect as the approach in the previous section, but this constraint can easily be written on the form of (24) as

$$Q_r = \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = Q_e^* * Q = \begin{bmatrix} * \\ * \\ -e_2q_0 + e_0q_2 - e_3q_1 + e_1q_3 \\ -e_3q_0 + e_0q_3 - e_1q_2 + e_2q_1 \end{bmatrix}, \tag{42}$$

where Q_e is time varying and takes the end-effector x-axis into the desired direction. Further, we want the end-effector x-axis to point in the opposite direction of the global x-axis, so we let $\gamma_{lim} \leftarrow \pi - \gamma_{lim}$ and write the corresponding cost function as

$$\phi_x = -\log(r_2^2 + r_3^2 - \nu^2), \tag{43}$$

where $\nu = \sin(\frac{\gamma_{lim}}{2})$ and γ_{lim} is the maximum error allowed in the direction of the x-axis. The partial derivatives are given by

$$\frac{\partial \phi_x}{\partial q_0} = -\frac{2(e_2^2+e_3^2)q_0 - 2(e_0e_2 - e_1e_3)q_2 - 2(e_1e_2+e_0e_3)q_3}{r_2^2+r_3^2-\nu^2},$$

$$\frac{\partial \phi_x}{\partial q_1} = -\frac{2(e_2^2+e_3^2)q_1 - 2(e_0e_3 + e_1e_2)q_2 - 2(e_1e_3-e_0e_2)q_3}{r_2^2+r_3^2-\nu^2},$$

$$\frac{\partial \phi_x}{\partial q_2} = -\frac{2(e_0^2+e_1^2)q_2 - 2(e_0e_2 - e_1e_3)q_0 - 2(e_0e_3+e_1e_2)q_1}{r_2^2+r_3^2-\nu^2},$$

$$\frac{\partial \phi_x}{\partial q_3} = -\frac{2(e_0^2+e_1^2)q_3 - 2(e_1e_2 + e_0e_3)q_0 - 2(e_1e_3-e_0e_2)q_1}{r_2^2+r_3^2-\nu^2}.$$

Thus, the search direction for every time step is given by

$$\nabla\phi = \begin{bmatrix} k_{err}\frac{\partial\phi_{err}}{\partial q_0} + k_{tcp}\frac{\partial\phi_{tcp}}{\partial q_0} + k_x\frac{\partial\phi_x}{\partial q_0} \\ k_{err}\frac{\partial\phi_{err}}{\partial q_1} + k_{tcp}\frac{\partial\phi_{tcp}}{\partial q_1} + k_x\frac{\partial\phi_x}{\partial q_1} \\ k_{err}\frac{\partial\phi_{err}}{\partial q_2} + k_{tcp}\frac{\partial\phi_{tcp}}{\partial q_2} + k_x\frac{\partial\phi_x}{\partial q_2} \\ k_{err}\frac{\partial\phi_{err}}{\partial q_3} + k_{tcp}\frac{\partial\phi_{tcp}}{\partial q_3} + k_x\frac{\partial\phi_x}{\partial q_3} \end{bmatrix}. \quad (44)$$

3) Restrictions of the Rotation about the z-axis: By Equation (20), we get that the rotation about the z-axis can be forced to be small by the cost function

$$\phi_x = -\log(\kappa^2 - \frac{q_3^2}{q_0^2}) \quad (45)$$

The partial derivatives are given by

$$\frac{\partial\phi_x}{\partial q_1} = 0, \quad \frac{\partial\phi_x}{\partial q_0} = -\frac{2q_3^2}{q_0(\kappa^2 q_0^2 - q_3^2)},$$
$$\frac{\partial\phi_x}{\partial q_2} = 0, \quad \frac{\partial\phi_x}{\partial q_3} = \frac{2q_3}{\kappa^2 q_0^2 - q_3^2}.$$

We would like the x-axis to point in the direction of the base, which we obtain by a rotation about the z-axis by $Q_e = \begin{bmatrix} e_0 & 0 & 0 & e_3 \end{bmatrix}^{\mathsf{T}}$. Again we use $Q_r = Q_e^* * Q$ and

$$\phi_x = -\log(\kappa^2 - \frac{r_3^2}{r_0^2}) \quad (46)$$

where

$$Q_r = \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = Q_e^* * Q = \begin{bmatrix} e_0q_0 + e_3q_3 \\ * \\ * \\ e_0q_3 - e_3q_0 \end{bmatrix}. \quad (47)$$

and the partial derivatives are given by

$$\frac{\partial\phi_x}{\partial q_1} = 0, \qquad \frac{\partial\phi_x}{\partial q_2} = 0, \quad (48)$$
$$\frac{\partial\phi_x}{\partial q_0} = \frac{2(e_3^4 - e_0^4)q_0q_3^2 - 2(q_3^3 - q_0^2q_3)(e_0^3e_3 + e_0e_3^3)}{r_0^2(\kappa^2r_0^2 - r_3^2)},$$
$$\frac{\partial\phi_x}{\partial q_3} = \frac{2(e_0^4 - e_3^4)q_0^2q_3 - 2(q_0^3 - q_0q_3^2)(e_0e_3^3 + e_0^3e_3)}{r_0^2(\kappa^2r_0^2 - r_3^2)}.$$

Then by choosing Q_e such that the x-axis points in the direction of the base by a rotation about the z-axis, we obtain the desired motion characteristics. Note that in (46) the central axis is assumed to be orthogonal to the surface. Hence, the results are only valid when a small orientation error in the direction of the z-axis is allowed.

B. Normalisation

The optimisation algorithms described optimise freely over all quaternions, and it is thus not guaranteed, nor likely, that the resulting quaternion is of unit length. One simple and very effective, though not very mathematically sound solution is to optimise freely over all quaternions and then normalise the result afterwards. Another option is to add the constraint $|Q| = 1$ in the optimisation algorithm which guarantees that the search space is only the set of quaternions of unit length.

VI. Numerical Examples

A. Convergence

Table I shows the computational efficiency of the algorithms presented. Convergence is in general very good and a solution is found in 10-20 iterations. In some cases a few more iterations are needed, but for all the tests performed, about 50 iterations is sufficient, as a worst-case measure. The simulation were performed on an Intel T7200 2GHz processor. We can see that the time needed for each iteration is very low. Even for the worst case of 50 iterations the time needed to find a solution is less than one millisecond. This makes all the algorithms presented suitable for on-line implementation.

Algoritm	Iteration Time [ms]	Max its needed	Max Time [ms]
z-axis cone	0.00232	50	0.116
z-axis cone & x-axis cone	0.00268	50	0.1608
z-axis cone & restr x-axis	0.00605	50	0.363

TABLE I
SPEED FOR ONE ITERATION, NUMBER OF ITERATIONS NEEDED TO ”GUARANTEE” AN OPTIMAL SOLUTION, AND TIME NEEDED TO OBTAIN OPTIMAL SOLUTION.

B. Path Planning

The three algorithms presented were compared to conventional path planning for spray paint applications. The algorithms tested were i) z-axis cone restrictions as presented in Section IV-A; ii) z-axis cone restrictions as presented in Section IV-A with additional cone restriction on the direction of the x-axis as presented in Section V-A2; iii) z-axis cone restrictions as presented in Section IV-A with additional restriction on the rotation about the z-axis as presented in

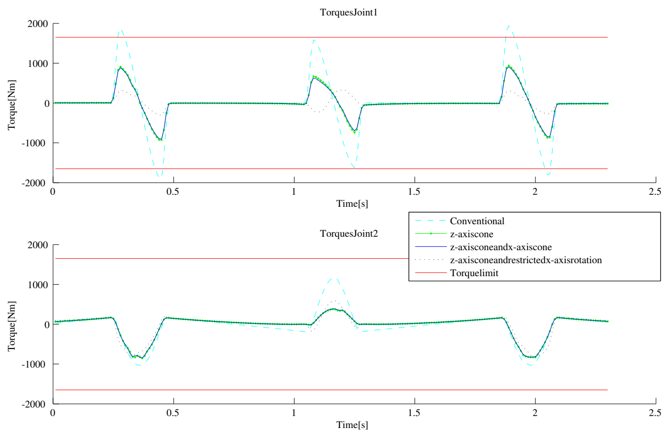

Fig. 3. Torques for joint 1 and 2 for the four different approaches presented.

Section V-A3. The manipulator was to follow the path given in Figure 1 with a constant speed of 1m/s. The torques of joints 1 and 2 for each case is shown in Figure 3 together with the torque limits of each joint. We can see that all approaches improve performance substantially. The approach with restricted z-axis only performs very well and is very easy to implement. For large allowed orientation errors of the z-axis the x-axis cone will reduce the orientation error not only of the x-axis but also the z-axis. This may be considered a side-effect of this cone constraint as the main motivation behind this restriction is to change the direction of the x-axis and not the z-axis. This side-effect is not present for the last approach which determines the direction of the x-axis by restricting the rotation around the end-effector z-axis. This approach will thus perform better in some cases as the orientation error of the z-axis, which is our main concern, is not reduced. This approach does, however, have a numerical singularity when q_0 approaches zero. In this case, the performance is drastically reduced and this must be handled in the implementation.

Table II shows the maximum speed for which the manipulator can follow the path for each algorithm. The speed increases for all the approaches presented. Table II also shows the maximum orientation error of the z-axis in each case. The maximum allowed orientation error is set to 20° for all approaches. We see that the maximum orientation error when both the z- and x-axes are restricted by a cone is lower than for the two other cases. This is because, as described above, the restriction on the x-axis cone will also affect direction of the z-axis. As the direction of the z-axis is our main tool to improve performance, this approach does not perform as well as the other two when large orientation errors are allowed.

Algorithm	Max Speed [m/s]	Max orientation error
Conventional	0.91	0
z-axis cone	1.35	20
z-axis cone & x-axis cone	1.28	12
z-axis cone & restr x-axis	1.37	20

TABLE II
MAXIMUM SPEED FOR WHICH THE MANIPULATOR CAN FOLLOW THE PATH FOR THE FOUR DIFFERENT APPROACHES PRESENTED.

VII. FUTURE WORK

In this paper the step size was chosen as constant to keep the computational burden for each iteration low. A search algorithm which searches for the optimal step size may improve performance and guarantee convergence. The approach presented requires a feasible initial condition. An improvement would probably be to follow the ideas of [11] and turn the problem into a primal-dual problem which can be solved efficiently without the need of a feasible initial condition.

VIII. CONCLUSIONS

This paper casts constraints on the orientation error into logarithmic barrier functions. The search direction of each time step is found and the problem is solved very efficiently by the gradient method. It is shown that the optimal orientation error can be found very fast and how this can be implemented in real-time path planning to improve performance. An example of a spray paint robot is used to verify the results numerically.

ACKNOWLEDGMENT

The authors wish to acknowledge the support of the Norwegian Research Council and the TAIL IO project for their continued funding and support for this research. The TAIL IO project is an international cooperative research project led by StatoilHydro and an R&D consortium consisting of ABB, IBM, Aker Kvaerner and SKF. During the work with this paper the first author was with the Hong Kong University of Science and Technology and University of California at Berkeley.

REFERENCES

[1] V. Potkonjak, G. Dordevic, D. Kostic and M. Rasic, *Dynamics of anthropomorphic painting robot: Quality analysis and cost reduction* Robotics and Autonomous Systems, Vol. 32, No 1, 2000.

[2] P. J. From and J. T. Gravdahl, *General solutions to kinematic and functional redundancy* Proc. 46th IEEE Conference on Decision and Control, 2007.

[3] P. J. From and J. T. Gravdahl, *On the Equivalence of Orientation Error and Positive Definiteness of Matrices* Proc. 10th International Conference on Control, Automation and Vision, 2008.

[4] M. Buss, H. Hashimoto and J. B. Moore, *Dextrous Hand Grasping Force Optimization* IEEE Trans. on robotics and automation Vol. 12 no. 3, 1996.

[5] B. Alpern, L. Carter and M. Grayson and C. Pelkie, *Orientation Maps: Techniques for Visualizing Rotations (A Consumers Guide)* IEEE Conference on Visualization, 1993.

[6] J. B. Kuipers, *Quaternions and Rotation Sequences* Princeton University Press, 2002.

[7] J. M. Ahuactzin and K. K. Gupka, *The Kinematic Roadmap: A Motion Planning Based Global Approach for Inverse Kinematics of Redundant Robots* IEEE Trans. on Robotics and Automation, 1999.

[8] L. Vandenberghe, S Boyd and S. P. Wu, *Determinant maximization with linear matrix inequality constraints* http://stanford.edu/∼boyd/papers/maxdet.html, 1996.

[9] P. J. From and J. T. Gravdahl, *Representing Attitudes as Sets of Frames* Proc. American Control Conference, 2007.

[10] L. Han, J. Trinkle and Z. Li, *Grasp analysis as linear matrix inequalitiy problem* IEEE Trans. on Robotics and Automation Vol. 16, no 6, 2000.

[11] S. Boyd and B. Wegbreit, *Fast Computation of optimal contact forces* IEEE Transactions on Robotics Vol. 23, no 6, 2007.