Torque Minimization for Redundant Manipulators

Amr Aly (1848399)

December 22, 2019

Abstract

We study torque minimization as a criterion for resolving kinematic redundancy in manipulators. The minimization is studied, locally, at different time instants (current and/or future) in order to address the torque stability problem.

Introduction

Solving the inverse kinematics problem for non-redundant robots always has a unique solution (assuming no singularities). A robot is said to be redundant if the number of actuated degrees of freedom $n > m$, where the desired cartesian task to be realized is m-dimensional.

For redundant robots, the inverse kinematics problem yields infinite solutions. To solve redundancy, joint motions are selected from infinite possibilities to achieve a secondary objective besides the desired end-effector motion. In this report, we define the secondary objective to be the minimum requirement of joint torques constrained by the defined trajectory.

Methods for resolving the kinematic redundancy are classified into two kinds – kinematic resolution or dynamic resolution – according to whether the manipulator dynamics are considered or not. Methods studied here fall under dynamic resolution since the manipulator dynamics affect the joint torques directly through the manipulator's dynamic model.

We study four methods to minimize the torque output. Some of the methods considered develop a global optimization scheme for the whole trajectory, however, this proved to be a very complex computation. Instead, local schemes were developed that require local information of the end-effector motion.

The first method by Hollerbach et al [1] minimizes the instantaneous motor torques by resolving the redundancy at the acceleration level. The solution was unstable for longer movements. It produced motion oscillations, growing joint velocities, and sudden whipping torque effects. The second method is a variation of the first method, which takes into account the torque bounds.

The third method by Li et al [2] formulates the problem as using the torque at the current time instant τ_k to drive the joint velocities to an optimal value $\dot{q}_{k+1}^{opt}$ that minimizes the torque at the next time instant τ_{k+1} subject to the task and torque bounds constraints.

The fourth method by Al Khudir et al [3] introduces an extension to the first method, that addresses the torque instability issue. Here, the joint torque norm is minimized over two successive discrete-time steps using a preview window. The dynamics are approximated in the future proximal state to keep the linear-quadratic property of the problem and to obtain a closed-form solution.

Dynamic Resolution of Redundancy

Kinematic resolution methods presume that, since the sum of squares of joint velocities is minimized by the pseudoinverse of the task jacobian, then the kinetic energy is approximately minimized. However, to incorporate the generalized inverse into dynamics, the pseudoinverse must be formulated in terms of joint accelerations.

We define some basic terminology for describing the manipulator. Consider a robot manipulator with generalized coordinates $\boldsymbol{q} \in \mathbb{R}^n$ with the Lagrangian dynamic model

$$\tau = M(q)\ddot{q} + c(q, \dot{q}) + g(q) \tag{1}$$

with inertia matrix M, coriolis and centrifugal terms c, gravity term g and commanded joint torque τ. We also use the shorthand notation

$$n(q, \dot{q}) = c(q, \dot{q}) + g(q) \tag{2}$$

The desired task described by the variables $\boldsymbol{x} \in \mathbb{R}^m$, with $m < n$, is related to $\boldsymbol{q}$ by the task forward kinematics $\boldsymbol{x} = f(\boldsymbol{q})$.

The joint velocities $\dot{\boldsymbol{q}}$ is related to the end-effector velocity through

$$\dot{\boldsymbol{x}} = J\dot{\boldsymbol{q}} \tag{3}$$

where $J \subset \mathbb{R}^{m \times n}$ is the task jacobian.

The joint acceleration is related to the end-effector acceleration by differentiating (3) to obtain

$$\ddot{\boldsymbol{x}} = J\ddot{\boldsymbol{q}} + \dot{J}\dot{\boldsymbol{q}} = J\ddot{\boldsymbol{q}} + h(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{4}$$

Instantaneous torque minimization

The method proposed by Hollerbach, denoted Minimum Torque Norm (MTN), minimizes the instant squared torque norm

$$\begin{aligned} \min_{\ddot{q}_k} \quad & H_1 = \frac{1}{2}\|\tau_k\|^2 \\ \text{s.t.} \quad & \tau_k = M_k\ddot{q}_k + n_k \\ & \ddot{x}_k = J_k\ddot{q}_k + h_k \end{aligned} \tag{5}$$

The unique solution to (5) can be obtained, using the method of Lagrangian multipliers, as

$$\ddot{q}^* = J^{\#}_{M^{-2}}(\ddot{x}_k - h_k) - (I - J^{\#}_{M^{-2}}J_k)M_k^{-1}n_k \tag{6}$$

where $J^{\#}_{M^{-2}}$ is the weighted pseudoinverse

$$J^{\#}_{M^{-2}} = M_k^{-2}J_k^T(J_kM_k^{-2}J_k^T)^{-1} \tag{7}$$

where the following shorthands are used: $M_k = M(q_k), n_k = n(q_k, \dot{q}_k), J_k = J(q_k), h_k = h(q_k, \dot{q}_k)$.

From (6) we notice that $M_k^{-1}n_k$ is the null space action that minimizes the torque norm. Also, the direct use of $\ddot{x}_k = \ddot{x}_{d,k}$ (the desired cartesian acceleration) results in an open loop solution. Therefore, a stabilizing PD feedback term is introduced to reduce the trajectory tracking error

$$\ddot{x}_k = \ddot{x}_{d,k} + K_D(\dot{x}_{d,k} - \dot{x}_k) + K_P(x_{d,k} - x_k) \tag{8}$$

with $m \times m$ gain matrices $K_D > 0$ and $K_P > 0$.

An extension to this method which considers the torque bounds, denoted Minimum Torque Norm with Bounds (MTNB), is also reviewed. Assume that the upper and lower joint torque limits are τ^+ and τ^-, respectively, and that the limits are motion-independent. The goal is to place the joint torques closest to the midpoints of the limits $\frac{1}{2}(\tau^+ + \tau^-)$. Thus, the minimization problem becomes

$$\begin{aligned} \min_{\ddot{q}_k} \quad & H_2 = \frac{1}{2}\left(\tau - \frac{\tau^+ + \tau^-}{2}\right)^T W \left(\tau - \frac{\tau^+ + \tau^-}{2}\right) \\ \text{s.t.} \quad & \ddot{x}_k = J_k\ddot{q}_k + h_k \end{aligned} \tag{9}$$

where W is a symmetric positive-definite weighting matrix

$$W = diag\left[\frac{1}{(\tau_i^+ + \tau_i^-)^2}\right] \tag{10}$$

and τ_i^+, τ_i^- are the upper and lower torque limits for joint i. H_2 is considered as a function of τ, which in turn is a function of $\ddot{q}$. Therefore, using the Lagrangian multipliers, it is possible to obtain the optimal $\ddot{q}^*$

$$\ddot{q}^* = A_k(\ddot{x}_k - h_k) + B_k \tag{11}$$

where

$$A_k = (M_k^TWM_k)^{-1}J_K^T[J_k(M_k^TWM_k)^{-1}J_k^T]^{-1} \tag{12}$$

$$B_k = (A_kJ_k - I)M_k^{-1}\left(n_k - \frac{\tau^+ + \tau^-}{2}\right) \tag{13}$$

Similar to the previous method, PD feedback is applied to $\ddot{x}_k$. It is worth to mention that both q and $\dot{q}$ are obtain using the euler integration of $\ddot{q}$ with $T_s = t_{k+1} - t_k$ as the timestep

$$\dot{q}_{k+1} = \dot{q}(t_{k+1}) = \dot{q}_k + \ddot{q}_kT_s \tag{14}$$

$$q_{k+1} = q(t_{k+1}) = q_k + \dot{q}_kT_s + \frac{1}{2}\ddot{q}_kT_s^2 \tag{15}$$

Both MTN and MTNB have the following disadvantages:

- they minimize the joint torques locally without global consideration
- they minimize the joint torques regardless of whether they are within torque limits or not

Peak Torque Reduction

The method proposed by Li, denoted Peak Torque Reduction (PTR), follows a different approach. Instead of minimizing the joint torques at all times, PTR minimizes joint torques only when they exceed the limits. It also considers the contribution of both $\dot{q}$ and $\ddot{q}$, unlike previous methods.

The torque is not minimized upto a certain time instant t^P, at which the torque is expected to exceed the limits. In fact, the choice of $\ddot{q}(t)$ is arbitrary as long as it satisfies (4) and

$$\dot{q}^P_{opt} = \int_0^{t^P} \ddot{q}(t)dt \tag{16}$$

subject to $\tau_i^- \leq \tau_i(t) \leq \tau_i^+$. The commanded torque can be rewritten by substituting (6) into (1) to obtain

$$\tau = M(q)\left(A(q)[\ddot{x}(t) - h(q,\dot{q})] + B(q,\dot{q})\right) + n(q,\dot{q}) \tag{17}$$

Finally, the minimization problem is formulated as a nonlinear programming problem by substituting (17) into (9), obtaining a fourth order objective function of $\dot{q}$

$$\begin{aligned} \min_{\dot{q}} \quad & H_3 = \frac{1}{2}\left(\tau^P(q^P,\dot{q}^P,\ddot{x}^P) - \frac{\tau^+ + \tau^-}{2}\right)^T \\ & \times W\left(\tau^P(q^P,\dot{q}^P,\ddot{x}^P) - \frac{\tau^+ + \tau^-}{2}\right) \\ \text{s.t.} \quad & \dot{x}^P = J(q^P)\dot{q}^P \end{aligned} \tag{18}$$

$\dot{q}^P_{opt}$ is then obtained by minimizing H_3 and $\ddot{q}^P_{opt}$ is available from (11).

However this approach offers a global optimization scheme and is very difficult to compute since it requires the information of the entire trajectory. It is also difficult to determine the time t^P at which the torque exceeds the limit, and the choice of $\ddot{q}(t)$ for $t < t^P$ that satisfies (16). It is, therefore, required to develop a local optimization scheme that requires local information about the desired motion.

Consider the interval $t_k \leq t \leq t_{k+1}$, where t_k, t_{k+1} are two immediate time instants. The goal is to use the future state $k+1$ as an objective and use current torque τ_k to accelerate/decelerate the joint velocity from $\dot{q}_k$ to $\dot{q}_{k+1}^{opt}$ (thus, minimizing τ_{k+1}), while keeping τ_k within limits.

The minimization problem in (18) can then be rewritten as

$$\begin{aligned}\min_{\dot{q}} \quad & H_3 = \frac{1}{2}\left(\tau_{k+1}(q_{k+1}, \dot{q}_{k+1}) - \frac{\tau^+ + \tau^-}{2}\right)^T \\ & \times W \left(\tau_{k+1}(q_{k+1}, \dot{q}_{k+1}) - \frac{\tau^+ + \tau^-}{2}\right) \qquad (19)\\ \text{s.t.} \quad & \dot{x}_{k+1} = J(q_{k+1})\dot{q}_{k+1}, \\ & \tau_k^- \le \tau_k^i \le \tau_k^+\end{aligned}$$

In other words, τ_k is used to minimize τ_{k+1}. Hence, τ_k is based on the preview information of the next state. When a large torque is needed at the next state due to the large $\dot{x}_{d,k+1}$ and $\ddot{x}_{d,k+1}$, a large current torque τ_k^{opt} may be used (subject to the torque limits) to drive the manipulator to the optimum velocity $\dot{q}_{k+1}^{opt}$ at the next state, which along with $\ddot{q}_{k+1}^{opt}$, will reduce the joint torque τ_{k+1}.

In order to make the problem simpler and the dependencies more clear, we use the following substitutions. At current state k, q_k and $\dot{q}_k$ are already known from the previous timestep. The joint position and velocity for the next state are computed using euler integration (like in (14) and (15))

$$q_{k+1} \approx q_k + \dot{q}_k T_s \qquad (20)$$

$$\dot{q}_{k+1} \approx \dot{q}_k + \ddot{q}_k T_s \qquad (21)$$

$\ddot{q}$ can be obtained by rewritting (1) to be

$$\ddot{q}_k = M^{-1}(q_k)\left[\tau_k - n(q_k, \dot{q}_k)\right] \qquad (22)$$

equation (21) then becomes

$$\dot{q}_{k+1} = \dot{q}_k + M^{-1}(q_k)\left[\tau_k - n(q_k, \dot{q}_k)\right] T_s \qquad (23)$$

It is clear that q_{k+1} can be directly computed and that the only unknown in $\dot{q}_{k+1}$ is τ_k. Thus the only unknown variable in (19), in both objective and constraint, is τ_k. So the optimization problem results in τ_k^{opt}, and from (22), (23) we obtain $\ddot{q}_k^{opt}$, $\dot{q}_{k+1}^{opt}$, respectively.

Torque Minimization Using Short Preview

The method proposed by Al Khudir, denoted Model Based Preview (MBP), augments the idea of MTN with the idea of PTR. Here, we use two successive discrete-time samples to minimize the joint torque norm. The two samples are the current time instant and a future instant (possibly, the next sampling instant). To keep the linear-quadratic formulation of the problem, suitable approximations are applied to the dynamics of the future state.

MBP tries to address two shortcomings of the previous methods, real-time simplicity and stable torque behaviour. Computational efficiency is achieved thanks to the closed-form solution of a linear-quadratic (LQ) problem. Stability is realized by using a single preview state, which may be close to or further away from the current one.

To prevent the unstable behaviour, consider a preview window $T = pT_s$ from some integer $p \ge 1$. In the following, we consider the case when $p = 1$, but the same formulas can be applied for $p > 1$.

Similarly to previous methods, the goal is to minimize the following objective

$$\begin{aligned}\min_{\ddot{q}_k, \ddot{q}_{k+1}} \quad & H_4 = \frac{1}{2}\left(\omega_k \|\tau_k\|^2 + \omega_{k+1}\|\tau_{k+1}\|^2\right) \\ \text{s.t.} \quad & \tau_k = M_k \ddot{q}_k + n_k, \\ & \ddot{x}_k = J_k \ddot{q}_k + h_k, \qquad (24)\\ & \tau_{k+1} = M_{k+1}\ddot{q}_{k+1} + n_{k+1}, \\ & \ddot{x}_{k+1} = J_{k+1}\ddot{q}_{k+1} + h_{k+1}\end{aligned}$$

where $\omega_k > 0$, $\omega_{k+1} > 0$ weigh the torque norms at current and future instants.

When using (14) and (15) in (24), the problem loses the LQ structure in the unknown joint accelerations. In fact, the objective function will no longer be quadratic in $\ddot{q}_k$ and the constraints become nonlinear in $\ddot{q}_k$. Therefore, we introduce dynamic approximations to the future state in order to preserve the forward coupling between current acceleration and the command at the future state allowing a linear dependence of the constraints and a quadratic dependence of the objective on $\ddot{q}_k$.

Consider the approximated task constraint at $t = t_{k+1}$

$$\ddot{x}_{k+1} \approx J_{k^+}\ddot{q}_{k+1} + (J_{k^+} - J_k)\ddot{q}_k + h_{k^+} \qquad (25)$$

where the following shorthands are used $J_{k^+} = J(q_k + \dot{q}_k T)$, $h_{k^+} = \frac{J_{k^+} - J_k}{T}\dot{q}_k$.

Next, consider the approximated squared norm of the torque at $t = t_{k+1}$

$$\begin{aligned}\|\tau_{k+1}\|^2 \approx\ & \ddot{q}_{k+1}^T M_{k^+}^2 \ddot{q}_{k+1} \\ & + 2\left(S_{k^+}(\dot{q}_k + \ddot{q}_k T) + g_{k^+}\right)^T M_{k^+}\ddot{q}_{k+1} \\ & + (\dot{q}_k + \ddot{q}_k T)^T S_{k^+}^T S_{k^+}(\dot{q}_k + \ddot{q}_k T) \qquad (26)\\ & + 2 g_{k^+}^T S_{k^+}(\dot{q}_k + \ddot{q}_k T) + g_{k^+}^T g_{k^+}\end{aligned}$$

where the following shorthands are used $M_{k^+} = M(q_k + \dot{q}_k T)$, $g_{k^+} = g(q_k + \dot{q}_k T)$, $S_{k^+} = S(q_k + \dot{q}_k T, \dot{q})$. And S is obtained through a convenient factorization of $c(q, \dot{q})$ given by $c(q, \dot{q}) = S(q, \dot{q})\dot{q}$ in which matrix S yields skew-symmetry for $\dot{M} - 2S$.

A full derivation of the approximations can be found in [3].

Now, using (25) and (26), the nonlinear optimization problem (24) has the following LQ approximation

$$\begin{aligned}\min_{\ddot{q}_k,\ddot{q}_{k+1}} \quad & H_4 = \frac{1}{2}\left(\omega_k\|\tau_k\|^2 + \omega_{k+1}\|\tau_{k+1}\|^2\right) \\ \text{s.t.} \quad & \tau_k = M_k\ddot{q}_k + n_k, \\ & \ddot{x}_k = J_k\ddot{q}_k + h_k, \\ & \tau_{k+1} = M_{k+}\ddot{q}_{k+1} + S_{k+}(\dot{q}_k + \ddot{q}_k T) + g_{k+}, \\ & \ddot{x}_{k+1} = J_{k+}\ddot{q}_{k+1} + (J_{k+} - J_k)\ddot{q}_k + h_{k+}\end{aligned} \tag{27}$$

Again, using Lagrangian multipliers, the solution can be written in the following form

$$\ddot{q}^* = A_Q^{\#}b - (I - A_Q^{\#}A)Q^{-1}r \tag{28}$$

with the weighted pseudoinverse

$$A_Q^{\#} = Q^{-1}A^T(AQ^{-1}A^T)^{-1}$$

with the following substitutions

$$Q = \begin{pmatrix}\omega_k M_k^2 + \omega_{k+1}T^2 S_{k+}^T S_{k+} & \omega_{k+1}TS_{k+}^T M_{k+} \\ symm & \omega_{k+1}M_{k+}^2\end{pmatrix}$$

$$r = \begin{pmatrix}\omega_k M_k(S_k\dot{q}_k + g_k) + \omega_{k+1}TS_{k+}^T(S_{k+}\dot{q}_k + g_{k+}) \\ \omega_{k+1}M_{k+}(S_{k+}\dot{q}_k + g_{k+})\end{pmatrix}$$

$$A = \begin{pmatrix}J_k & 0 \\ J_{k+} - J_k & J_{k+}\end{pmatrix}$$

$$b = \begin{pmatrix}\ddot{x}_k - h_k \\ \ddot{x}_{k+1} - h_{k+}\end{pmatrix} \tag{29}$$

Similarly, the task based constraint in (27) can be written in the matrix form as

$$A\begin{pmatrix}\ddot{q}_k \\ \ddot{q}_{k+1}\end{pmatrix} = b$$

The closed form solution is the concatentaion of the two states

$$\ddot{q}^* = \left(\ddot{q}_k^T \; \ddot{q}_{k+1}^T\right)^T \tag{30}$$

In the implementation, only $\ddot{q}_k$ is used at the time instant $t = t_k$ while $\ddot{q}_{k+1}$ is discarded. The position and velocity at the next time instant are obtained by applying euler integration, using (14) and (15). Similar to the instantaneous torque minimization methods, PD feedback is applied to the term $\ddot{x}_k$ (*only*) in (29) using (8).

Simulation Results

We consider a 3R planar arm ($n = 3$) moving on a horizontal plane (no influence of gravity). The robot has 3 uniform links of length $l = 1$ [m], mass $m_l = 10$ [kg], and moment of inertia $I_l = m_l l^2/12$. The simulated end-effector motion is a rest-to-rest straight-line cartesian trajectory, with bang-bang acceleration profile. The end-effector orientation is unconstrained, giving one degree of redundancy ($m = 2$). The upper and lower torque limits for joints 1-3 are $\pm 54, \pm 24$ *and* $\pm 6\,N.m.$ and the initial configuration

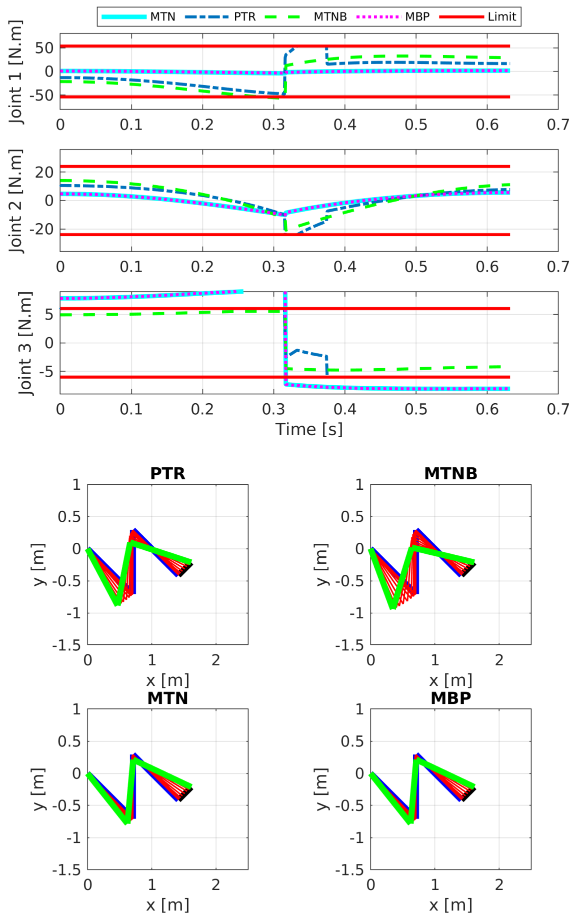

Figure 1: Short movement

is $q_0 = (-45° \; 135° \; -135°)^T$. The integration step was fixed to $T_s = 0.001$ [s]. The MBP method uses equal weights $\omega_k = \omega_{k+1} = 1$, and a preview window of $T = 100T_s = 0.1$ [s]. In all methods, the feedback gains on the cartesian task error were $K_P = 10I$ and $K_D = I$.

First, the case of the short movement is considered where the end-effector makes a movement of 0.2 [m] in both x and y directions ($\|L_1\| = 0.2828$ [m]), with an acceleration/deceleration of 2 [m/s^2] in both x and y directions ($\|A_1\| = 2.8284$ [m/s^2]).

Torque profiles and continual motion configurations are shown in Fig. 1. We compare the 4 methods discussed previously (MTN, MTNB, PTR, MBP). We notice that PTR and MTNB make almost full or complete usage of the torque for joint 3 in order to reduce peaks during the whole trajectory, while MTN and MBP use minimal torques for the first two joints, but exceed the limit for the last one.

Second, the case of the long movement is considered where the end-effector makes a movement of 0.83 [m] in both x and y directions ($\|L_2\| = 1.1738$ [m]), with an acceleration/deceleration of 1 [m/s^2] in both x and y directions ($\|A_2\| = 1.1412$ [m/s^2]).

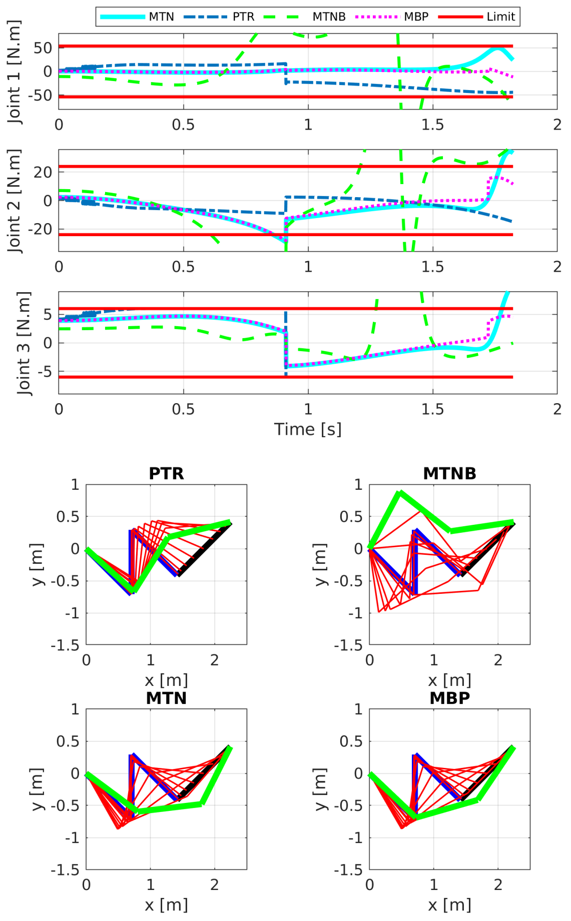

Figure 2: Long movement

Torque profiles and continual motion configurations are shown in Fig. 2. We notice the extremely large torques produced by MTNB and the whipping effect by MTN. On the other hand, since PTR and MBP use some information from a future state to compensate the locality of the algorithm, their performance is much more robust for longer movements (with the exception of MBP slightly exceeding the limit for joint 2). We cannot neglect the perturbation in torques realized by PTR in the beginning of the movement which shows indeed that MBP is better at addressing the torque instability issue. Also, for most of the motion MBP did not make full utilization of the torques for any joint.

In all simulations, the cartesian tracking error was in the order of 10^{-4} [m]. Attempting to increase the preview window of MBP ($100T_s \leq T \leq 900T_s$) lead to similarly good behaviour by MBP.

Conclusion

Different methods have been reviewed for kinematically redundant robots, with the aim of reducing joint torques consumption. Methods use local information (at current and/or future instants) about end-effector motion to minimize the joint torques. MTN and MTND use only local information at current time instant leading to an unstable behaviour for longer movements. On the other hand, PTR and MBP attempt to *foresee* or *anticipate* future torque requirements and thus produce more stable results.

Future work will address the combination of MTNB and MBP methods to produce stable torque profiles for both short and long movements.

References

[1] Hollerbach, J. O. H. N. M., and Ki Suh. "Redundancy resolution of manipulators through torque optimization." IEEE Journal on Robotics and Automation 3.4 (1987): 308-316.

[2] Li, Degao, Andrew A. Goldenberg, and Jean W. Zu. "A new method of peak torque reduction with redundant manipulators." IEEE Transactions on Robotics and Automation 13.6 (1997): 845-853.

[3] Al Khudir, Khaled, et al. "Stable Torque Optimization for Redundant Robots using a Short Preview." IEEE Robotics and Automation Letters 4.2 (2019): 2046-2053.