Resolved Rate and Acceleration Control in the Presence of Actuator Constraints

Manja Kircanski and Nenad Kircanski

Experimental evaluation of various task-space control algorithms shows that significant deviations from desired trajectories may occur, especially at higher speeds, or in the vicinity of singularities. The experiments on a modular reconfigurable robotic setup confirmed this conclusion; the lateral deviations may range up to several centimeters from the closest point on the desired trajectory. The main reason is that the control output often reaches the saturation value of an actuator. The lateral error is also found very dangerous in obstacle-cluttered environments due to possible damage to the manipulator arm while hitting an obstacle at high speed. This article is devoted to solving this problem. Widely used task space control algorithms such as the resolved-rate and the resolved-acceleration algorithms are modified by introducing monitoring on the control output and, accordingly, modifying the commanded control signal so that the end-effector is prevented from leaving the commanded path. As a consequence, the manipulator tip may slow down along the path due to limited actuator power, but it will be accelerated after that to reach the desired position as soon as possible. The algorithms also take into account the presence of singularities to avoid uncontrollable excursions of the manipulator tip. Hundreds of experiments have been carried out to confirm the concept. Some of those with a desired trajectory reaching both the regular and the singular region of the workspace are provided to illustrate the method.

Introduction

Robot control systems are typically hierarchical so that the desired task space trajectory is generated on higher control levels, while the trajectory tracking is performed at a lower level, often called the servo-control level. The desired trajectory can be known in advance, or it can be generated online while the robot is in motion. This depends on the knowledge of the environment at the time of execution. It is often the case that the environment changes and has to be identified during the task execution. Visual, ultrasonic, and other sensors are used to facilitate online trajectory planning. Based on sensor information, the robot can be commanded to move along a straight line segment in a given time interval (or at a given speed). Since there exists a nonlinear relationship between the task and the joint coordinates, it may easily occur that some of the corresponding joint trajectories may not be achieved given the limited power of actuators. Then, the end-effector will most probably leave the commanded path and possibly hit an obstacle.

To avoid such prohibited behavior, the inability of an actuator to generate enough power can be signaled to the higher control level, or it can be resolved at the servo level. Obviously, the second approach is preferable due to modularity of the control structure. In other words, a problem encountered at a certain control level should be resolved at this level if at all possible. Thus, the problem of limited actuator power should be resolved at the lowest control level. Ideally, the manipulator should automatically slow down to prevent the control output from reaching the saturation, and then accelerate as soon as possible to compensate for the time loss.

Saturation may occur in the vicinity of singularities where the required joint speeds become large, but also in any other region of the work space due to nonlinear transformation between the task and the joint space coordinates. In the case of off-line trajectory planning, this would be easily recognized, and the trajectory would be regenerated. However, with online motion planning, it cannot be guaranteed that actuator limits will not be reached. Within the resolved-rate [1] and the resolved-acceleration control [2] strategies, the higher control level provides desired speed (acceleration) and position in task space in each sampling interval. The kinematic transformations (inverse Jacobian, etc.) are integrated with the servo-loops into a nonlinear control law from which it is difficult to predict whether the actuator’s limits will be reached during the motion, since it depends both on commanded and real position and velocity of the robot arm. Failing to resolve such a situation may result in considerable deviation from the desired straight line trajectory.

Not much research has been devoted to elimination of large position errors in task space introduced by actuators’ constraints. In [3], the problem of geometric limits in joint coordinates was resolved. The problem of power limits is not considered. Some industrial robot controllers instantly enlarge time-to-travel (e.g., 30%) when a joint reaches its speed limit, resulting in an abrupt drop in joint velocities. The controller does not compensate for the time lost even when it is possible after leaving the critical zone. In [7] an idea of introducing a braking factor at the kinematic control level was proposed. It provided that limits on joint
velocities were not exceeded, while planning joint trajectories for given straight-line paths.

In this article, a new control strategy based on (i) the resolved-rate control (RRC), and (ii) the resolved-acceleration control (RAC) is introduced. The original control algorithms are modified by introducing a braking coefficient that affects the commanded velocity and acceleration in RRC and RAC, respectively. The commanded velocity/acceleration is effectively decreased in the case of saturation of one or more actuators. This creates lagging along the desired trajectory, but in the lateral direction no error is introduced. It also speeds up the motion once the actuators enter the lidar region again, and the lagging is minimal. The braking coefficient is obtained as an output from the Monitor Control Block (MCB) which takes into account actuators' limits (saturation levels), current status of the arm (position and speed), and desired status as dictated by the higher levels. The main purpose of the MCB is to prevent the arm from leaving a desired path. The proposed control strategy is applicable to the robots that have all degrees of freedom (DOF) actuated since the braking may cause uncontrollable excursions of passive joints. Such robots are rarely used in industry, though.

The results are illustrated by providing simulation and experimental results. Simulation results include resolved-rate control with braking for a PUMA robot. The RAC-based braking algorithm is illustrated by experimental results obtained by using an advanced setup based on modular and reconfigurable robotic arms—Robotwin—developed at the Robotics and Automation Laboratory of the University of Toronto [9]. The effectiveness is proved on both high-speed trajectories and those entering singular region of the workspace.

Damped Resolved-Rate Control with Braking

The well-known resolved-rate control scheme [1] can be presented in the form

$$\dot{q} = J^{-1}(q)(\ddot{x}_d + K_p(x_d - x))$$  \quad (1)$$

where $\dot{q}$ is the vector of commanded joint velocities, $n$ is the number of degrees of freedom of the manipulator, $J$ is the Jacobian matrix, $\ddot{x}_d$ is the vector of desired task space velocities, $K_p$ is the position gain matrix, $x_d$ is the desired task space position vector, and $x$ is the real position/orientation vector that depends on measured joint coordinates. The control signals are actually the commanded joint velocities, properly scaled prior to being sent to the actuators. Therefore, actuators constraints are actually constraints on commanded joint velocities$$|\dot{q}_i| \leq V_i, \quad i = 1, ..., n$$  \quad (2)$$

where $V_i > 0$ are given constants. Due to singularity of the Jacobian matrix, some components in $\dot{q}_i$ tend to infinity in the vicinity close to a singular configuration. At the same time, the task space velocities may have very modest values. Therefore, we shall apply the damped resolved-rate control scheme, where the inverse Jacobian is replaced by the damped least-squares Jacobian matrix $J^s$ [4-8]

$$\dot{q} = J^s(q)(\ddot{x}_d + K_p(x_d - x))$$  \quad (3)$$

Outside robot singularities, matrix $J^s$ coincides with the inverse/pseudoinverse Jacobian matrix (for nonredundant/redundant robots), while in the vicinity of singularities it yields the damped least-squares solution. The damped Jacobian matrix reduces the commanded joint velocities in a zone around a singular configuration, but there is no guarantee that the limits on joint velocities will not be reached. Besides, reducing the joint speed by introducing too-high damping factors could result in increased position errors [6, 7]. Thus, it is recommended to apply the damped least-squares solution in a rather narrow zone around singularities. Therefore, upon calculating commanded joint velocities from (3), it should be checked whether the limits on joint velocities have been reached. If the constraints have been violated, the braking factor should be determined and the originally desired trajectory should be modified. This is the function of the Monitor Control Block that incorporates two braking algorithms: the regular and the total-signal braking algorithms.

Regular braking: This algorithm is applied if the only reason for constraints violation is high desired velocity $\dot{x}_d$, while the position error is not significant. In this case $\dot{x}_d$ is reduced by a factor $k$, $k \in [0,1]$, and an altered desired trajectory is called commanded trajectory $x_d(t)$ is introduced. The commanded trajectory is identical to the desired trajectory $x_d(t)$ provided that there is no violation of actuators' limits. Initially, when the robot starts moving ($t = 0$), the commanded position is set to be equal to the desired one: $x_c(0) = x_d(0)$. The commanded veloc-
ity is computed from the difference between the originally desired position at the time instant \( t + T \) and the commanded position at time \( t \) as \( \mathbf{x}_c(i) = (\mathbf{x}_c(i + T) - \mathbf{x}_c(i)) / T \) where \( T \) is the sample time. The commanded joint velocities are computed from

\[
\mathbf{\dot{q}}_c = \mathbf{J}^T(\mathbf{q})(\mathbf{\dot{x}}_c + \mathbf{K}_p(\mathbf{x}_c - \mathbf{x})),
\]

(4)

The scalar form of this equation is

\[
\mathbf{\dot{q}}_{\text{sc}} = \mathbf{J}^T(\mathbf{x}_c + \mathbf{K}_p(\mathbf{x}_c - \mathbf{x})),
\]

(5)

where \( \mathbf{\dot{q}}_{\text{sc}} \) is the \( i \)th commanded joint velocity, while \( \mathbf{J}^T \) is the \( i \)th row of the Jacobian matrix. If one or more constraints (2) are violated, we modify the corresponding scalar equations (5) by introducing a coefficient \( k_i \) which satisfies the condition

\[
\mathbf{V}_i \text{sgn}(\mathbf{\dot{q}}_{\text{sc}}) = \mathbf{J}^T(k_i \mathbf{\dot{x}}_c + \mathbf{K}_p(\mathbf{x}_c - \mathbf{x}))
\]

(6)

that is,

\[
\mathbf{V}_i \text{sgn}(\mathbf{\dot{q}}_{\text{sc}}) = \mathbf{J}^T(k_i \mathbf{\dot{x}}_c + \mathbf{K}_p(\mathbf{x}_c - \mathbf{x}))
\]

(7)

for \( i \in L \), while for all those degrees of freedom for which \( \mathbf{\dot{q}}_{\text{sc}} \leq V_i \), we assume \( k_i = 1 \). If the solutions to the equations (7) yield coefficients \( k_i \) in the range \([0, 1] \) for each \( i \in L \), we determine which of the joints requires the most intensive braking. This is obtained by determining the smallest coefficient \( k_i \)

\[
k = \min_{i \in L} k_i
\]

(8)

The commanded joint velocity vector \( \mathbf{\dot{q}}_c \) is now computed as

\[
\mathbf{\dot{q}}_c = \mathbf{J}^T(k \mathbf{\dot{x}}_c + \mathbf{K}_p(\mathbf{x}_c - \mathbf{x}))
\]

(9)

while the commanded position for the next control interval becomes \( \mathbf{x}_c(i + T) = \mathbf{x}_c(i) + k \mathbf{\dot{x}}_c T \). Note that the commanded velocity in task space is equal to \( k \mathbf{\dot{x}}_c \). Since all components of the vector \( \mathbf{\dot{x}}_c \) are multiplied with the same coefficient, it is clear that the manipulator tip will slow down, but remain on the path. Note that with the original Resolved-Rate Control (1) this is not the case: the speed-limits encountered by some actuators will cause manipulator tip to leave the desired path.

If any of the solutions to Equation (7) yields the coefficient \( k_i \) out of the range \([0, 1] \), it follows that the velocity braking is not sufficient to reduce the control output \( \mathbf{\dot{q}}_c \) to the linear operation range (2). This may happen because \( \mathbf{\dot{x}}_c \) is close to zero, or because \( \mathbf{K}_p(\mathbf{x}_c - \mathbf{x}) \) is large as compared to \( \mathbf{\dot{x}}_c \). Then, the following algorithm is applied:

**Braking by the total signal:** If at least one of the coefficients \( k_i \) is out of the range \([0, 1] \), braking by the total signal \( \mathbf{\dot{q}}_{\text{sc}} = \mathbf{K}_p(\mathbf{x}_c - \mathbf{x}) \) should be applied. The braking coefficients are then obtained as

\[
k_i = \begin{cases} 
1, & |\mathbf{\dot{q}}_{\text{sc}}| \leq V_i \\
\frac{V_i}{|\mathbf{\dot{q}}_{\text{sc}}|}, & |\mathbf{\dot{q}}_{\text{sc}}| > V_i
\end{cases}
\]

(10)

This yields \( k = \min_{i \in L} k_i \). Reduced commanded joint velocities are now computed as

\[
\mathbf{\dot{q}}_c = \mathbf{J}^T(k \mathbf{\dot{x}}_c + k \mathbf{K}_p(\mathbf{x}_c - \mathbf{x}))
\]

(11)

Again, the commanded velocity is \( k \mathbf{\dot{x}}_c \), so that the commanded position for the next control interval becomes \( \mathbf{x}_c(i + T) = \mathbf{x}_c(i) + k \mathbf{\dot{x}}_c T \).

This provides the critical joint is controlled at its maximum speed, while the other joints are proportionally scaled down in speed. Whenever the regular braking can keep joint velocities within their limits, it should be applied, since it provides for the elimination of the position error without changing the gains of the system. Namely, the "regular braking" only affects the desired speed along a given path. Thus, the only implication of introducing the braking factor is in reducing the manipulator end-effector speed while one or more actuators are in saturation.

With "total signal braking," the implication is not only in reducing the speed as dictated by the term \( k \mathbf{\dot{x}}_c \) in (11), but also in reducing the gains by a factor of \( k \). This is the consequence of the term \( k \mathbf{K}_p(\mathbf{x}_c - \mathbf{x}) \) in (11). Thus, the fundamental difference between the "regular" and "total signal" braking is in overall system behavior: "regular" braking changes the speed, while the "total signal" braking changes both the speed and all gains proportionally by the factor of \( k \).

A simulation of the Damped Resolved-Rate Control scheme for a PUMA 560 robot, and the Damped Resolved-Rate control enhanced by the algorithm (4-11), is illustrated in Fig. 1. The robot is moving from the point \( \mathbf{x}_0 = (0.35, -0.35, 0.04, 0.785, 3.14, 0) \) to the point \( \mathbf{x}_f = (0.35, 0.35, 0.04, 0.785, 3.14, 0) \) along the straight line. Here, the first three coordinates show Cartesian position of the end-effector in the base frame in meters, while the second three correspond to yaw, pitch, and roll angles of the end-effector in radians. Zero initial position and orientation errors were adopted, to enable a possible error due to the braking algorithm to show up. The maximum velocity of the manipulator tip is 1.2 m/s. Constraints on joint velocities are 3 rad/s for the first three joints, and 6 rad/s for the last three. The constraint is violated for the first joint along this movement. The lateral position error is...
measured as the shortest distance between the straight-line path and the real manipulator position $x(t)$, $y(t)$, $z(t)$. The longitudinal error is the distance between the desired position at some time instant and the point that lies in the path and is nearest to the actual position at the same time instant. The diagrams on the left-hand side in Fig. 1 show the robot behavior controlled by the original Damped Resolved-Rate Control scheme (with limited joint velocities), while the plots on the right-hand side show the robot behavior when the braking algorithm (4-11) was implemented. Note that the maximum lateral error with the Damped-RRC is 9 mm, while it is only 0.35 mm with the modified Damped-RRC. The limited speed of the joint 1 caused lagging along the given path (up to 27 mm), but not the deviation from the desired trajectory.

**Damped Resolved-Acceleration Control with Braking**

The classical resolved-acceleration control scheme [2] is given by

$$ u = H(q)J^T(q)[\ddot{x}_r - J\dot{q} + K_v(x_r - x) + K_p(x_r - x)] + h(q, q) $$

where $H$ is the identified inertial matrix of the system (the model of the mechanism and the actuators), $h$ is the vector of centrifugal, gravitational forces, viscous and Coulomb friction, $K_v$ and $K_p$ are diagonal matrices of position and velocity gains, and $u$ is the vector of actuator input signals. In [5,8], this control scheme was modified to cover the vicinity of robot singularities

$$ u = H(q)J^T(q)[\ddot{x}_r - J\dot{q} + K_v(x_r - x) + K_p(x_r - x)] + h(q, q) $$

(12)

Experimental verification of this control law [8] showed that in the vicinity of singularities actuator constraints

$$ |u_i| \leq U_i $$

(13)

($U_i > 0; i = 1, ..., n$) might be violated, although the damped least-squares Jacobian matrix reduces joint velocities considerably and eliminates numerical problems. In this case, the real manipulator tip trajectory significantly deviates from the desired trajectory in the task space. A good example is the experiment carried with PUMA 560 robot controlled by Damped-RAC scheme described in [8]. The thicker line in Fig. 2 shows the executed trajectory, while the desired one consists of four straight

segments ABCD. Due to saturation of one of the actuators, a large deviation from the desired trajectory occurs even in the zones that are far from the singular region (the arc BC).

This is very undesirable in most applications. Thus, the recomputation of the desired accelerations is necessary to enable the manipulator end-effector to follow the straight-line path.

Here, again, regular braking and braking by the total signal can be recognized.

**Regular braking:** The aim of the regular braking is to alter the desired acceleration $\ddot{x}_r$ in order to bring the control signals into their limits. Initially, when the robot starts moving ($t=0$), the commanded position in time instants $t$ and $t+T$ coincide with the desired ones, i.e., $x_r(0) = x_d(0)$, $x_r(T) = x_d(T)$. The commanded velocity is equal to $\dot{x}_r(t) = \dot{x}_r(t+T) - \dot{x}_r(t))/T$, while the velocity at $t + T$ is computed from the expression $\dot{x}_r(t+T) = (x_r(t+2T) - x_r(t+T))/T$. The acceleration is computed as $\ddot{x}_r(t) = (\dot{x}_r(t+T) - \dot{x}_r(t))/T$. The control output is

$$ u = H(q)J^T(q)\left[\ddot{x}_r - J\dot{q} + K_v(x_r - x) + K_p(x_r - x)\right] + h(q, q) $$

(14)

If the control signals violate the constraint (13), the commanded acceleration $\ddot{x}_r(t)$ should be modified. However, it is not possible just to scale the commanded acceleration by a factor of $k$. Namely, at time instant $t$, the following variables are already known and fixed: commanded positions at time instant $t$ and $t+T$, i.e., commanded velocity $\dot{x}_r$ at time instant $t$. The only variable that can be varied is the position at time instant $t+T$ or, equivalently, commanded velocity at time instant $t+T$. Therefore, if the $i$th control signal in time instant $t$ does not satisfy the condition (13), the braking factor for this degree of freedom $k_i$ will be determined from the condition

$$ U_i \|u_i\| = H_iJ^T_i\left[(k_i\ddot{x}_r(t+T) - \ddot{x}_r(t))/T - J\dot{q} + K_v(x_r - x) + K_p(x_r - x)\right] + h_i $$

(15)

where $H_i$ is the $i$th row of matrix $H$, $h_i$ is the $i$th element of vector $h$, and $u_i$ is the control determined from (14). The solution of Equation (15) for $k_i$ is

$$ k_i = \frac{1}{H_iJ^T_i}U_i \|u_i\| - H_iJ^T_i\left[-\ddot{x}_r(t)/T - J\dot{q} + K_v(x_r - x) + K_p(x_r - x)\right] + h_i $$

(16)

assuming that $\|H_iJ^T_i\ddot{x}_r(t)/T\| > \varepsilon$. Otherwise, the total signal braking algorithm is applied. If all $k_i$ are in the range [0,1], ($i = 1, ..., n$) the final braking factor is determined from (8) assuming that $k_i = 1$ for all joints satisfying $|u_i| \leq U_i$.

Modification of accelerations, velocities, and positions should be done according to

$$ \ddot{x}_r(t) = (k_\ddot{x}_r(t+T) - \ddot{x}_r(t))/T $$

$$ \dot{x}_r(t+2T) = \dot{x}_r(t+T) + k_\ddot{x}_r(t+T)T $$

$$ \dot{x}_r(t+T) = k_\ddot{x}_r(t+T) $$

(17)

New control signals are now recomputed from (14).
Braking by the total signal: If at least for one degree of freedom braking factor determined from (16) lies outside the range [0,1], the total signal braking should be applied. The condition from which the braking factor for the ith degree of freedom is evaluated, now becomes

\[ U_i \text{sgn}(u_i) = H_i \int_0^T \left\{ k_i (\dot{x}_i(t+T) - \dot{x}_i(t)) + \frac{1}{T} \right\} - \ddot{q}_i \]

\[ + h_i K_i (\dot{q}_i(t) - \dot{q}_i) + k_i (\alpha_i - \alpha_i) \]

Now, we get \( k_i = \begin{cases} 1 & \text{for } |u_i| \leq U, \\ \frac{U_i \text{sgn}(u_i)}{H_i \int_0^T \left\{ k_i (\dot{x}_i(t+T) - \dot{x}_i(t)) + \frac{1}{T} \right\} - \ddot{q}_i + h_i K_i (\dot{q}_i(t) - \dot{q}_i) + k_i (\alpha_i - \alpha_i)}{U} & \text{otherwise} \end{cases} \]

The total braking factor \( k \) is evaluated from (8). Modified accelerations, velocities, and positions are still computed according to Equation (17), while control signals are recomputed with new commanded accelerations (17) and position and velocity gains \( K_p \) and \( K_v \) reduced by the factor of \( k \).

**Experimental Results**

Experimental verification of the braking algorithm for the Damped Resolved-Acceleration Control (Damped-RAC) scheme was performed on two DOF horizontal planar configuration of the modular robot "Robotwin," Fig. 3, [9]. The robot consists of reconfigurable modules—joints—actuated by brushless DC motors and harmonic drives. The first and third joint in Fig. 3 were used as a planar 2 DOF manipulator. The link lengths were 0.25 m and 0.18 m.

The desired trajectory of the robot is a trapezium ABCD defined by the points: \( A = (0.33, 0.25) \), \( B = (0.417, 0.104) \), \( C = (0.417, -0.104) \), \( D = (0.33, -0.25) \) (all units are meters). The points B and C are on the work-space boundary circle (where the robot is in singularity). Motion execution times for the segments AB, BC, CD were 1.4 s, while for DA was 1.2 s. Manipulator was commanded to wait 0.5 s in point B, 0.3 s in point C, 0.1 s in point D, and 1 s in point A. The diagrams on the left-hand side in Fig. 4 show the robot behavior controlled by the original Damped-RAC algorithm (12), while the plots on the right-hand side show the robot behavior when the braking algorithm (14-18) was implemented. The outputs were not in saturation while the robot was moving along the segments AB, BC, and CD, but the saturation occurred while the robot was moving along the DA segment. This caused significant deviation from the desired path when the robot was controlled by Damped-RAC algorithm.

The deviation \( x(t) - x_d(t) \) is presented in Fig. 4 as a lateral and longitudinal error. The lateral error is the distance between the manipulator tip and the nearest point (P) on the desired path. The longitudinal error is the distance between the desired position and the point P. The points A, B, C, and D in Fig. 4 show the time when the manipulator tip is in the corresponding point, and is about to start moving toward the next one following the path ABCD. Note that the maximum lateral error with the Damped-RAC is 15 mm, while it is only 0.5 mm with the modified Damped-RAC as described in this article. This shows the main feature of the braking algorithm: it does not let manipulator tip leave the programmed path. Fig. 4 shows that the maximal longitudinal errors were 11 cm and 14 cm for the original and the modified algorithms, respectively. The increased lagging with the modified algorithm is natural: the only way to stay on the desired path in presence of saturation is to slow down the motion when the actuators are in saturation, and to speed up after that to compensate for the time delay if possible. Fig. 4 shows that the travel time along the segment DA was 1.4 s with the original algorithm and 1.5 s with the modified one. Clearly, the 1.5 seconds is needed in order to avoid deviation from the straight line.

The bottom two plots in Fig. 4 show the braking factors. Since the original Damped-RAC is equivalent to the modified one when the braking coefficient is 1, the left bottom plot shows \( k = 1 \). The right bottom plot shows the time evolution of \( k \) with the modified Damped-RAC. Clearly, the braking was active (\( k \) small) when the robot was moving along the path DA. The braking factor was only about 0.05 in this region due to the fact that very intensive braking was necessary to prevent the manipulator from leaving the desired path (DA). Clearly, this is the consequence of the fact that the manipulator was "forced" to accomplish the motion DA (0.5 m segment) in only 1.2 seconds, while for the other segments which are shorter (only 0.2 m) the travel time was 1.4 seconds. In the same experiment, but with the DA time set to about 2 seconds, the coefficient \( k \) was always above 0.5. In conclusion, the larger the discrepancy between the desired performance (speed, accelerations, etc.), and actual performance that a system can deliver, the smaller the coefficient \( k \).
Conclusion
In this article, the idea of braking along straight-line Cartesian paths, when actuator constraints are reached (previously developed at the kinematic control level [7]), is broadened to cover the Cartesian space resolved-rate and resolved-acceleration control. The lack of the actuator power is transformed into the minimal lagging along desired straight line segments. Experiments show that the stability of the system was preserved, and that position deviations less than 0.5 mm were achieved. This method eliminates the need for very precise real-time planning of motion execution times. It is applicable both for and close to robot singularities.

References