

REDUCED END-EFFECTOR MOTION AND DISCONTINUITY IN SINGULARITY HANDLING TECHNIQUES WITH WORKSPACE DIVISION

Denny Oetomo

*Singapore Institute of
Manufacturing Technology*

Marcelo Ang Jr.

*Dept. of Mech. Engineering
National University of Singapore*

Abstract

There have been many singularity handling techniques proposed in the past. Many of these techniques divide the manipulator workspace into singular and non-singular regions. A singular region is defined in the vicinity of a singular configuration, within which, a different control algorithm is applied to handle the singularity. Generally, it involves removing the degenerate component(s) of motion. This results in discontinuity in the motion of the end-effector as it moves in and out of the singular region. This paper discusses the motion of the end-effector inside the singular region and the disturbances in motion as it crosses the boundary of the singular region. Types of disturbances, the causes, proposed solutions and implementation result on PUMA 560 are presented in the paper.

1 Introduction

Many approaches have been suggested in the past to handle the singular configurations of a manipulator. In the interest of this paper, there are mainly two types of singularity handling methods: ones that introduce workspace division and ones that do not.

Techniques without workspace division generally introduce a continuous modification either to the Jacobian matrix or to the trajectory of the end-effector. This continuous function is close to zero when the end-effector is further away from a singular configuration as not to affect the tracking performance. In the vicinity of singularity, it introduces a slight modification to either the Jacobian matrix, the trajectory, or to the workspace to avoid the ill-conditioned inverse to the Jacobian matrix. The methods found in [1, 2, 3, 4], among others are designed within this concept.

The second category of singularity handling techniques introduces a region in the vicinity of the singular configuration in which a different set of control algorithm is applied to the manipulator. The methods in this category include [5, 6, 7, 8, 9], among many others.

The techniques in the second category share the main concept of removing the element that causes singularity from the control algorithm when the end-effector is within the singular region. The method we adopted in this paper is briefly explained in Section 2.

In this paper, we look into practical issues of the end-effector motion which result from the removal of the degenerate element from the control task. This motion is removed not just at the exact point of singularity, but throughout the period that the end-effector spends inside the singular region. As the result, error would occur and accumulate in the singular direction while tracking the desired trajectory. The causes of motion discontinuity were explored and solutions proposed and implemented real-time on the PUMA 560 manipulator.

2 Handling Singularity by Removing Degenerate Components

Firstly, unique sets of singular configurations are identified. A singular region is then defined in the vicinity of the singular configurations. When the end-effector enters one of the singular regions, corresponding components of task in the degenerate direction are removed from the task (which means removing them from the Jacobian matrix and the operational forces). Upon exit from the singular region, the components are replaced back into the task. The detail of the algorithm can be found in [8]. This temporary removal causes discontinuity issues that are investigated in this paper.

2.1 The Singular Region

The idea we adopt is to identify and remove the degenerate components as motivated by Chiaverini's [6] and Khatib's [10] earlier work. In the vicinity of a singularity, a region is defined as the singular region. This is done by specifying a threshold to the value of determinant of the Jacobian matrix. The determinant can be found symbolically, and each factor $s(\mathbf{q})$ represents one singularity in the system. A region \mathcal{D} is defined around each of these factors [10]:

$$\mathcal{D} = \{\mathbf{q} \mid |s(\mathbf{q})| \leq s_0\} \quad (1)$$

where $s(\mathbf{q})$ is the factor in determinant of the Jacobian matrix which is zero at specific singularity and s_0 is the threshold value that defines the singular region.

This divides the entire workspace into two subspaces: one with full Jacobian and the other with reduced Jacobian matrix (having had the degenerate components removed). Switching occurs at the boundary of the two subspaces.

2.2 Removing Degenerate Components

The unique sets of singular configuration can be found by symbolically solving $\det(\mathbf{J}) = 0$ for a square \mathbf{J} matrix or $\det(\mathbf{J}\mathbf{J}^T) = 0$ for redundant manipulators.

Whether in kinematic control or lower level algorithms such as torque control, the concept of removing the singular component(s) of motion involves removing the corresponding elements in the Jacobian matrix and the task space command vector (in kinematic control, the task space command vector will be the joint rate vector, while in torque control the operational space force vector). In this paper, the operational space formulation in torque control was utilised. The singularity handling process is done by first transforming the Jacobian matrix \mathbf{J} and the operational space force \mathbf{F} to the singular frame $\{S\}$. Singular frame $\{S\}$ is a frame defined so that one (or more) of the axes is aligned with the singular direction(s). The row(s) of zeros in the resulting Jacobian matrix ${}^S\mathbf{J}$ (\mathbf{J} matrix expressed in singular frame) - which represents the singular direction - is removed and Jacobian matrix collapsed to one of full rank but of fewer rows. The corresponding component in ${}^S\mathbf{F}$ (Force vector \mathbf{F} in singular frame) is also removed.

2.3 Null Space Control

Within the singular region, with the degenerate component of the task removed, the manipulator is treated as a redundant system with respect to the original task. Task in directions orthogonal to the singular direction will be controlled in the range space while motion in the singular direction can only be generated with the help of the projection of certain potential functions in the null space of the Jacobian matrix. Dynamically consistent force/torque relationship guarantees decoupled operational space and null space behaviour [11]. Motion in degenerate direction was studied in [12, 13].

Null space task can be realized by constructing a potential function, $V_0(\mathbf{q})$, whose minimum corresponds to the goal of the desired null space task. In our operational space formulation which utilises torque control:

$$\begin{aligned}\mathbf{\Gamma} &= \mathbf{J}^T \cdot \mathbf{F} + \mathbf{N}^T \mathbf{\Gamma}_0 \\ \mathbf{N} &= \mathbf{I} - \bar{\mathbf{J}}\mathbf{J}\end{aligned}\tag{2}$$

where $\mathbf{\Gamma}$ is the torque to be sent to the joints and \mathbf{F} is the vector of operational space forces. \mathbf{N} is the null space projection of the Jacobian matrix and $\mathbf{\Gamma}_0$ is the null space torque. The null space torque $\mathbf{\Gamma}_0$ is formulated as the gradient descent of a potential function ($V_0(\mathbf{q})$):

$$\mathbf{\Gamma}_0 = -\mathbf{A}(\mathbf{q})\nabla V_0(\mathbf{q});\tag{3}$$

where $\mathbf{A}(\mathbf{q})$ is the joint space inertia matrix.

Singularities could be categorized into type 1 and type 2, based on the effects of null space torques $\mathbf{\Gamma}_0$ [10]. Type 1 singularity occurs when the null motion creates end-effector motion/forces only in the singular direction(s). When in singular configuration, null space motion in type 1 singularity moves the manipulator out of singularity. To escape a type 1 singularity, the end-effector motion in singular direction can be generated directly through the associated null space torque by

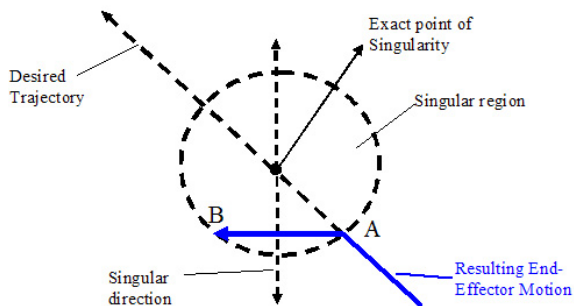


Figure 1: As the end-effector reaches point A at the boundary of singular region, it can no longer track the task in the singular direction. It can only perform motion in the directions perpendicular to the singular direction.

potential function $V_0(\mathbf{q})$ to move it from a singular configuration $s(\mathbf{q}) = 0$ to the boundary of singular region, $s(\mathbf{q}) = s_0$ (as declared in 1). Null space motion in type 2 singularities creates only internal joint motion, and changes the singular directions without affecting the end-effector motion/forces. In type 2 singularity, the null motion does not move the end-effector but shifts the singular direction. Potential function can be constructed such that its minimum occurs when the singular direction is orthogonal to the desired path ($\zeta = \zeta_0$). ζ is the joint configuration vector and ζ_0 is the configuration vector where singular direction is orthogonal to the desired path. This is sometimes termed as *escapable singularities* in some literatures [13], because the singular direction can be shifted away from the desired path.

3 Effects of Removal of Singular Direction

This section discusses the effect of the removal of the singular direction in the singular region on both type 1 and type 2 singularities [10].

Projection into null motion is utilised in our experiment to minimise errors and disturbance by creating some 'component of motion' in the originally lost degree-of-freedom. The following subsections discuss the motion of the end-effector around the singular configuration.

3.1 Motion in singular region

Singularity happens only at an exact point in the workspace. However, behaviour of the manipulator control in the vicinity of the singularity is not stable. Therefore singularity is handled within a region declared around the exact singular configuration. This introduces some approximation and therefore some degree of discrepancy as the end-effector will not be able to perform the task in the singular direction while inside the singular region. Motion is only available in the direction perpendicular to the singular direction (see Figure 1). Tracking error in the singular direction builds up within the singular region.

3.2 Exiting the Singular Region

Upon exiting the singular region, the control algorithm switches back to its full degrees-of-freedom. This is where the error that has been accumulated inside

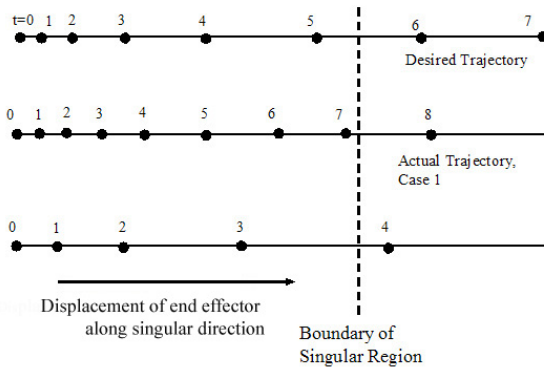


Figure 2: Case 1 is when the end-effector exits the singular region slower than the desired trajectory. As it exits, it has to catch up with the desired trajectory. Case 2 is when the end-effector exits the singular region ahead of the desired trajectory.

the singular region (in the degenerate direction), if not handled, manifests into disturbance or jerkiness in the motion.

3.2.1 Type 1 singularity

If the end-effector exits the singular region at the same point (in $(x, \dot{x}, \ddot{x},$ and $t)$) as the desired trajectory $(x_d, \dot{x}_d, \ddot{x}_d,$ and $t_d)$, then the motion will be continuous. However, as mentioned earlier, following a time-dependent trajectory near singularity is near to impossible. This is a problem for type 1 singularity because the end-effector exits the singular region through the singular direction, and hence the exit motion is not controllable according to the desired trajectory. It moves to the correct position, but not at the desired time. It moves to the correct position because only the motion control in the singular direction is released, while all other DOFs are still in motion control.

The elbow singularity of PUMA 560 is a good example. When the elbow is straightened, the singular direction is the motion of retracting the elbow from the straight posture to either an elbow up or elbow down configuration. This motion is not controllable to the time-dependent trajectory. As the result, the following cases could happen (see Figure 2).

The top diagram is the desired trajectory in task space. In case 1 (Figure 2 middle), the null motion drove the end-effector to move slower than the desired trajectory while inside the singular region. As it exits the singular region (at $t = 8$ in the example in Figure 2), the desired trajectory is far ahead. As the full degrees of freedom is returned to the control algorithm, the large error between the desired and actual end-effector position causes a jump in the torque being sent to the manipulator.

In case 2, the end-effector exits the singular region ahead of the desired trajectory. As the control algorithm returns to its full degrees of freedom (at $t=4$ in the example in Figure 2), the desired end-effector position is still inside the singular region. The motion control then pushes the end-effector back into the singular region, where the null motion would try to push it out again. This would cause a rattling at the boundary of singular region.

The suggested solution to the problem is to plan a new trajectory. In this

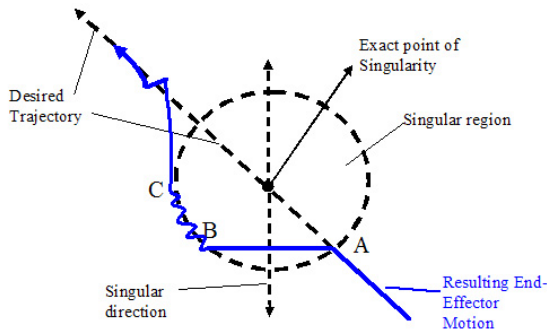


Figure 3: As motion is disabled in the singular direction, the end-effector will move only perpendicular to singular direction. Error accumulated inside the singular region in the singular direction causes the end-effector to exit the singular region not according to the desired trajectory.

method, a new trajectory is planned when the end-effector exits the singular region. This method assumes negligible volume for the singular region and takes the point immediately outside of the singular boundary as the starting of the new task space trajectory. The current position and velocity of the end-effector at that point are taken as the initial conditions. The final desired goal is kept the same as the original one. The desired length of time for the newly planned trajectory is the desired length of time of the original trajectory minus the time spent inside the singular region. This method produces the smoothest result with minimum effort so far.

3.2.2 Type2 singularity

While type 1 singularity faces problems with the time that the end-effector reaches the boundary of the singular region upon exit, type 2 has a problem whereby the end-effector exits at the wrong position from the desired trajectory (if not handled) (see Figure 3. This is because motion control in the degenerate direction is released, hence end-effector does not move in accordance to the desired trajectory in this direction. The solution is to utilise the null motion to change the configuration of the manipulator so that the singular direction is shifted orthogonal to the desired motion vector. When this is implemented, the end-effector will exit the singular region not in singular direction, but in the component of motion that is under motion control. Motion is therefore continuous.

4 Implementation on PUMA560

The singularities of a 6DOF PUMA manipulator arm have been identified in many literatures:

Wrist lock when the wrist is straightened, $q_5 = 0$. The singular direction is the rotation around X axis of Frame 4.

Elbow lock when the elbow is straightened, $q_3 = 92.69^\circ$.

Head lock when the wrist point is immediately above the Z axis of joint 1, $(d_4 S_2^3 + a_2 C_2 + a_3 C_2^3) = 0$.

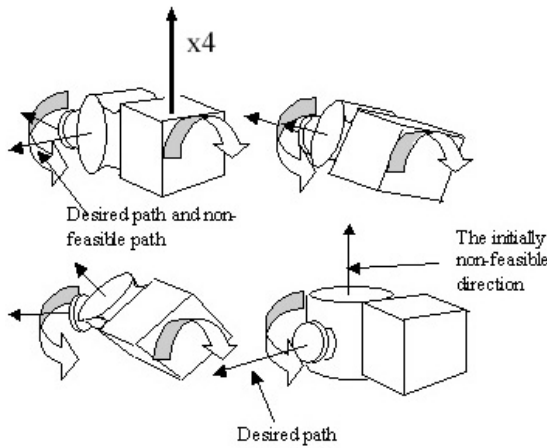


Figure 4: The singular direction is the rotation around axis X_4 . To be able to move in that direction requires a null space motion that rotates joint 4 so that the singular direction is shifted away from the desired path. This is a type 2 singularity.

4.1 Wrist Singularity

This is an example of a type 2 singularity (see Figure 4). As end-effector enters singular region, the Jacobian matrix and the operational space force vector are transformed to the singular frame. The singular row in \mathbf{J} and force/moment in singular direction are removed. All control calculations are performed with the reduced Jacobian matrix, which no longer considers the singular direction as part of the task. The null motion is implemented to shift the singular direction to be perpendicular to the desired trajectory. The end-effector exits the region perpendicular to the singular direction, therefore its exit trajectory is in motion control according to the desired trajectory. A simple damping technique can be implemented to eliminate any small disturbances caused by small numerical error in computation.

When implemented on PUMA, exiting the singular region without the assistance of the null motion to shift the singular direction would create jerkiness due to accumulated tracking error in the singular direction while inside the singular region. The result of the experimental run is shown in Figure 5. In this experiment, the PUMA robot was made to move from point ${}^B(x, y, z) = (0.681, 0.149, 0.013)$ to ${}^B(x, y, z) = (0.681, 0.150, 0.50)$ with respect to the Base Frame. Z axis is vertical with respect to absolute ground. This defines a trajectory that moves vertically upwards (in positive Z direction), with a very slight motion in Y direction while passing through the singular region. The singular direction is the rotation around X axis of Frame 4 which, in this specific case, is coupled with the translation along Y axis of Frame 4.

Figure 5 (top) shows the end-effector position tracking error. The translation in Y direction and the singular direction of rotation around X axis of Frame 4 are the dependant rows in the Jacobian matrix (in this particular example) and are therefore coupled. This shows as the drift in the motion control of Y axis with respect to Base Frame while inside the singular region. As it exits the region, it snaps back into the desired trajectory, causing a sudden disturbance to the

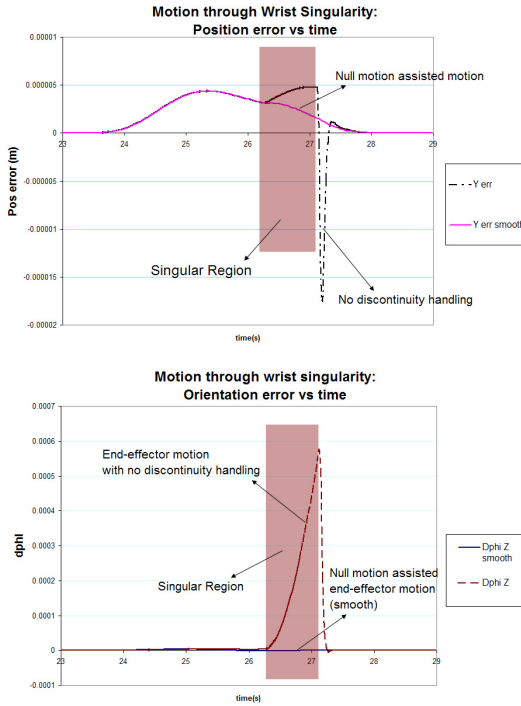


Figure 5: Top figure shows the end-effector position error with respect to time - with and without assistance of null motion. It is shown that motion in Y direction loses its control inside the singular region because it is coupled with the singular direction. Upon exit, it jerks back into its desired position. Bottom graph shows the end-effector orientation error, which also shows the drift from desired orientation in the singular direction, which snaps back in place upon exit.

motion. The smoother graph is the result of null motion in joint 4 to keep the desired trajectory away from the singular direction.

The result shows that it does not take a large motion in the singular direction to create such disturbance. In this experiment, motion in Y direction was only for a total displacement of 1 mm. This illustrates the significance of handling the motion inside singular region properly.

The bottom graph shows the end-effector orientation tracking error (in the singular direction $d\phi_Z$), shown with respect to the Base Frame. The singular direction is therefore reflected as the rotation around Z axis. It is shown that the motion in this direction drifts from the desired trajectory and only snaps back into the desired trajectory after full 6 DOF motion control is returned to the control algorithm, which is when the end-effector exits the singular region.

Both graphs also show the effect of null motion in shifting the singular direction to be orthogonal to the desired direction of motion. When this is done properly, the discontinuity is reduced significantly and the resulting tracking motion is smooth.

4.2 Elbow Singularity

In this type 1 singularity, null motion is implemented to create motion in the singular direction. The end-effector exits the singular region in the singular direction. Discontinuity was handled to obtain a smooth motion.

The desired trajectory in this experiment was to retract the PUMA manipulator arm, from a stretched-out (with a straight elbow) position to an elbow up or down configurations. The singular direction is the motion of retracting the straightened

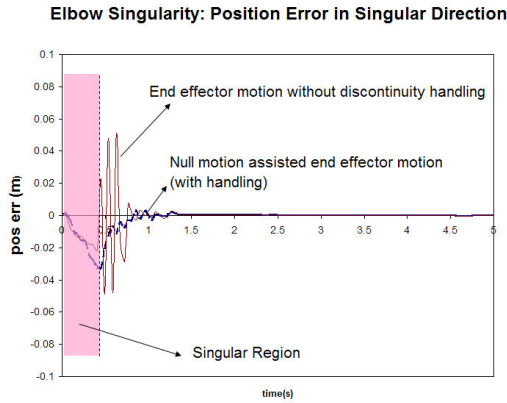


Figure 6: Motion of the end-effector through the elbow singularity. The singular direction: translation along X axis of Frame B, which is the line connecting wrist point to the origin of the Base Frame.

arm (elbow). Null motion is applied either joint 2 or joint 3 to assist the motion in the singular direction. The experimental result is only shown for Case 2 (Figure 6), as problem is Case 1 is easily solved by a damping effect as explained in Section 3.2.1.

In Figure 6, it can be observed that the initial position error in singular direction is negative, because the end-effector lead the desired trajectory. As it leaves the singular region, the full 6 DOF motion control pushes the end-effector back into the singular region to close the tracking error, as the desired trajectory lags the actual. The end-effector rattled back and forth across the boundary of the singular region several times before the desired trajectory caught up and lead ahead. Defining a new intermediate desired trajectory with current position and velocity as initial condition ensure the smooth transition. However, the new (intermediate) desired trajectory will be slightly off the original desired trajectory, so the control law will not attempt to close the tracking error immediately. The effect will be more significant if the portion inside the singular region constitute a large portion of the desired trajectory.

5 Conclusion

When a division in workspace is defined to handle singular configurations, discontinuity issues arise at the boundary between the sub-spaces. Error was observed especially when exiting the singular region, as the end-effector regains its full set of DOF in motion control and attempts to close the tracking error accumulated in the singular direction while the end-effector was inside the singular region. Solutions were suggested and implemented on simulation and on PUMA 560 manipulator. Results show that although it is possible to obtain smooth trajectory while moving across singular regions, in the vicinity of singularity, certain tracking performance still has to be sacrificed to achieve motion in singular direction. This is unavoidable, as singularity is a robot physical phenomenon, which can only be truly removed through the mechanical design of the robot.

References

- [1] Y. Nakamura and H. Hanafusa, "Inverse kinematics solutions with singularity robustness for robot manipulator control," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 108, pp. 163–171, 1986.
- [2] Manja V. Kircanski, "Symbolical singular value decomposition for a 7-dof manipulator and its application to robot control," *IEEE Conf. on Robotics and Automation*, vol. 3, pp. 895–900, 1993.
- [3] Jon Kieffer, "Differential analysis of bifurcations and isolated singularities for robots and mechanisms," *IEEE Transactions on Robotics and Automation*, vol. RA-10, pp. 1–10, Feb 1994.
- [4] D.N. Oetomo, M. H. Ang, and T. M. Lim, "Singularity robust manipulator control using virtual joints," *Proc. IEEE Conf. of Robotics and Automation*, vol. 3, pp. 2418–2423, May 2002.
- [5] E. W. Aboaf and R. P. Paul, "Living with the singularity of robot wrists," *IEEE Intl. Conf. for Robotics and Automation*, pp. 1713–1717, 1987.
- [6] S. Chiaverini and O. Egeland, "A solution to the singularity problem for six-joint manipulators," *Proc. IEEE for Robotics and Automation*, vol. 1, pp. 644–649, 1990.
- [7] Fan-Tien Cheng, Tzung Liang Hour, York-Yin Sun, and Tsing-Hua Chen, "Study and resolution of singularities for a 6-dof puma manipulator," *IEEE Trans. Systems, Man and Cybernetics*, vol. 27 2, pp. 332–343, 1997, Part B.
- [8] D. Oetomo, M. H. Ang, and S. Y. Lim, "Singularity handling on puma in operational space formulation," in *Lecture Notes in Control and Information Sciences*, Daniela Rus and Sanjiv Singh, Eds., vol. 271, pp. 491–501. Springer Verlag, 2001.
- [9] D.N. Nenchev and M. Uchiyama, "Singularity consistent path tracking: a null space based approach," *Proc. IEEE Robotics and Automation*, pp. 2482–2489, 1995.
- [10] Oussama Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robotics and Automation*, vol. RA-3, no. 1, pp. 43–53, 1987.
- [11] K. Chang and O. Khatib, "Manipulator control at kinematic singularities: A dynamically consistent strategy," *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, vol. 3, pp. 84–88, 1995.
- [12] C. Chevallereau, "Feasible trajectories for a non redundant robot at a singularity," *Proc. Intl. Conf. on Robotics and Automation*, vol. 2, pp. 1871 – 1876, April 1996.
- [13] N.S. Bedrossian, "Classification of singular configurations for redundant manipulators," *Proc. of IEEE Intl. Conf. Robotics and Automation*, vol. 2, pp. 818–823–17, 1990.