Singularity Robust Manipulator Control using Virtual Joints

Denny Oetomo, Marcelo H. Ang Jr., Tao Ming Lim
Department of Mechanical Engineering
National University of Singapore
Singapore 119260

Abstract

A singularity handling method is proposed in this paper. It is done by introducing virtual redundant joints into the Jacobian matrix to maintain the rank of the Jacobian matrix when singularity occurs. These additional joints do not exist physically. Therefore, although mathematically stable, the manipulator still can not perform tasks in the degenerate direction(s). This method is comparatively straight forward to implement and it does not have a singular subspace defined within which a special and different control algorithm is performed, thus it avoids the problem associated with discontinuous control or switching of control. The method was tested on simulation and implemented in real-time on the PUMA 560 robot.

1 Introduction

Singularity is a well-known problem in which the manipulator loses one or more of its degrees-of-freedom. This is reflected in the Jacobian matrix of the manipulator, which becomes rank deficient at the singular configurations. In many control algorithms, a rank deficient Jacobian matrix introduces unbounded joint velocities. Many methods have been presented in the past, to enable robots to go through singular configuration in a stable manner.

There are methods formulated based on the idea of removing the degenerate degree(s) of freedom from the rows of the Jacobian matrix. Aboaf and Paul [1] handled spherical wrist singularity by eliminating the singular direction and the contribution of the roll joint of the wrist. The velocity of the eliminated joint is then bounded at some maximum joint rate to avoid excessive joint velocity. The effect is compensated in task space. This approach results in a reduced (5 by 5) Jacobian which is of full rank. Chiaverini and Egeland [2] identified and removed the degenerate components of motion, and applied pseudo inverse with the collapsed Jacobian in their kinematics-based approach. Cheng [3] performed an analysis and handling method on PUMA 560 also by releasing the exactness in the singular direction and providing the extra redundancy to the achievable direction. Oetomo, Ang and Lim [4] removed the degenerate components from the Jacobian matrix and performed the dynamically consistent inverse [5] to obtain the inverse of Jacobian matrix. The method was implemented in the operational space formulation [6], which enabled singularity handling while performing motion and force control.

These methods divide the whole workspace into two sub-spaces: the singular region (in the vicinity of singularity) and the non-singular region. An issue with this approach is the switching of control algorithms causing discontinuities as the manipulator goes in and out of the singular regions.

There are other methods whereby no region is defined in the vicinity of singularities. In Nakamura [7], damped least-squares method was used to obtain a modified Jacobian that is not singular. Kircanski [8] utilized the Singular Value Decomposition (SVD) and replaced the zero singular value at the diagonal matrix with a continuous function of non-zero value.

Other efforts incorporate the handling of singularities into the design of their manipulators, for example by designing a redundant spherical wrist (4 joints) [9], or by designing a manipulator with isotropic Jacobian matrix, so its inverse is equal to its transpose [10].

The method proposed in this paper is to supply additional rank to the Jacobian matrix to maintain its rank, even in singular configurations. This can also be thought of as introducing virtual joint(s) to handle the lost degree(s)-of-freedom in the vicinity of singularities. These joints only exist computationally, not physically. Therefore, at singular configuration, the manipulator is still not able to perform any tasks in the degenerate directions although the inverse of the Jacobian matrix exists. This would prevent the manipulator from going into excessive joint rates (or unstable regions). The advantage of this method is that there is no division of workspace or switching of control algorithms, therefore resulting in smoother performance and simpler method.

An immediate issue with this method is that we do not want the robot to rely on the virtual joints to complete its tasks, a case that would adversely affect the tracking performance, since the virtual joints do not actually exist. Dynamics-based null space projection is then used to force the torque command assigned to the virtual joints to zero, therefore fully utilizing the physically existent joints.

The algorithm is evaluated in simulation and implemented on the PUMA 560 manipulator. The results of the real-time experiments are presented in this paper. We use the *operational space formulation* to achieve unified force and motion control [6]. The robot dynamics in operational space is derived using the modified Jacobian (with virtual joints). Force and motion control without interference from the virtual joints are realized using null-space projection of the virtual joint motions, thus enabling smooth motion across singularities in both motion and force control tasks. The method could easily be implemented in other control algorithms such as the resolved motion rate control.

2 Virtual Joints

When singularity happens, the manipulator loses its degree(s)-of-freedom, and the Jacobian matrix loses its rank(s) accordingly. In this condition, the inverse of the Jacobian matrix no longer exists. As the inverse of Jacobian matrix is essential in transforming control commands from Cartesian space (in which our tasks are specified) to joint space (in which the robot joints are controlled). It is desired to have a Jacobian matrix whose inverse always exists.

The method proposed is to supply the Jacobian matrix with extra column(s), that would guarantee that the Jacobian always has full rank even when in singular configuration. This is done by placing 'virtual joints' in the Jacobian matrix to replace the lost DOF when singularity occurs. This expands the Jacobian matrix from its original size of $m \times n$ matrix to $m \times (n + v)$ matrix. m is the dimension of the tasks specified, which is usually 6, representing 3 DOFs for translation (position) and 3 DOFs for orientation. n is the number of joints the manipulator possesses. v is the number of extra (virtual) joints that is to be added into the system.

Figure 1 shows a two-link planar robot in singular configuration (left). This is a boundary singularity. The degree of freedom lost in this case is the ability to translate along the line described by the straight arm. However, if more degrees of freedom were added to the system, for example: two prismatic joints (Figure 1 center), the robot can now translate anywhere even when the arm is straightened (assuming the prismatic joint does not reach its limit). A more efficient way would be to identify the lost degree of freedom at such singular configuration and to supply an extra DOF in this direction with a virtual joint (Figure 1 right), which is a prismatic joint to allow the end-effector to translate along the lost DOF.

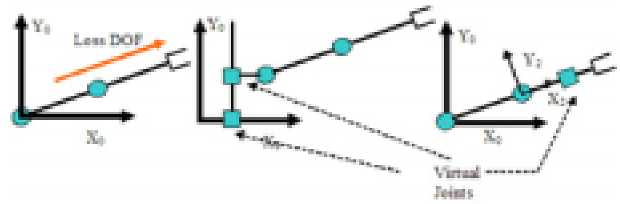

Figure 1: Example of a two-link planar manipulator in singular configuration and its lost DOF (left), and two ways of supplementing virtual joints into the system, where circles represent the revolute joints and squares represent the prismatic (virtual) joints

As the modification is done permanently to the Jacobian matrix, the matrix would never be singular, the control algorithm is uniform throughout the workspace. This eliminates the switching of control algorithm as found in some other methods, as mentioned in the introduction.

It should be noted that although the Jacobian matrix is made invertible at singular configurations, the virtual joints still do not exist physically, therefore the manipulator still can not perform tasks along the directions of the lost degrees-of-freedom.

The control algorithm would produce a command vector of size $(n+v)$ by 1, containing commands to be sent to individual joints (motors). In our case of operational space formulation, it would be a torque vector. The elements corresponding to the virtual joints are therefore ignored and only those corresponding to physically existing joints would be sent accordingly.

3 Avoiding Assignment of Command to Virtual Joints

The Jacobian matrix is a mapping from the joint space to the task space velocities. When there are more joints that the number degrees-of-freedom required by the task, the Jacobian is now considered 'redundant'. In a redundant system, there are many solutions in the joint space that would map onto the desired path of the end-effector in task space. We can choose one set of the solution by having the desired behavior of the extra joints projected into the null space of the Jacobian.

In our case, the joints added to the system are virtual, and they only exist computationally and not physically. As the manipulator is now described as having real and virtual joints, there is an issue whereby the manipulator might rely on the non-existent joints to complete the specified task. Some non-zero torque values may be assigned to the virtual joints in its attempt to track the given trajectory. This would have an adverse effect on the tracking performance of the robot.

Since the Jacobian is now redundant, the problem mentioned above can be prevented by choosing a set of solutions where the desired null space behavior is to have the virtual joints stationary at zero position. This keeps the virtual joints unused in completing the specified task and assigns the roles of completing the desired trajectory to the existing joints.

This is different from simply setting the torque or velocity command values to the virtual joints to zero. If a solution in the joint space (joint command) has been obtained by including the virtual joints as real joints, setting command values to these joints to zero would produce incomplete actuation for the end-effector to follow the task space trajectory. Using null space projection, however, the velocity of the virtual joints can be made zero, while maintaining the resulting forces/velocity at the end-effector.

The null space behavior is specified by defining a potential function as follows:

$$V_0(\mathbf{q}) = \frac{1}{2}\sum_i k_i[f(\mathbf{q}_i) - f(\mathbf{q}_{i(desired)})]^2 \qquad (1)$$

where $V_0(\mathbf{q})$ is the potential function, k_i is a constant gain. To obtain the desired null space behavior, which is to keep the virtual joints stationary at zero, the potential functions are defined as:

$$V_0(\mathbf{q}) = \frac{1}{2}\sum_i k_i[q_i - q_{i(desired)}]^2 \qquad (2)$$

where q_i are the joints that we would like to control in null space, which in this case, are the virtual joints. $q_{i(desired)}$ are the desired values of these joints, which are set to zero.

The gradient descent of the potential functions are then used as null space torque or velocity to be projected into the null space of the Jacobian. The details are given in section 4.

4 Implementation Example on PUMA 560

The algorithm above was implemented on the PUMA 560 manipulator. The frame assignments are shown in Figure 2 using the modified Denavit-Hartenberg (DH) convention and the parameters can be found in [11]. To simplify the identification of singularities, the control reference point of the manipulator is transformed to the center of the spherical wrist, resulting in a decoupled Jacobian [12]:

$$J_d = \begin{bmatrix} J_{11} & 0_{3x3} \\ J_{21} & J_{22} \end{bmatrix} \qquad (3)$$

Singularities are identified by evaluating the determinants of J_{11} and J_{22} representing position and orientation singularities respectively [12]. The PUMA 560 has the following singularities:

wrist: $S_5 = 0$	orientational
elbow: $d_4C_3 - a_3S_3 = 0$	positional
head: $d_4S_23 + a_2C_2 + a_3C_23 = 0$	positional

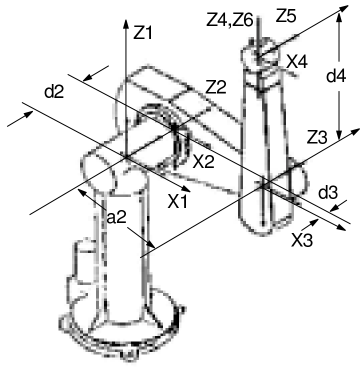

Figure 2: PUMA and the frame assignment used in the experiment, where a_2=0.43m, a_3=-0.0203m, d_2=0.2435m, d_3=-0.0934m, and d_4=0.4331m

The following example shows the handling of wrist singularity in detail. Equation 4 shows the lower right quadrant of the Jacobian matrix (${}^4J_{22}$), expressed in Frame{4}. When wrist singularity occurs, $S_5 = 0$, and the first row of this matrix is all zeros, therefore the determinant of the Jacobian is zero.

$${}^4J_{22} = \begin{pmatrix} 0 & 0 & S_5 \\ 0 & 1 & 0 \\ 1 & 0 & C_5 \end{pmatrix} \qquad (4)$$

This row of zeros corresponds to the rotation around the X-axis of $Frame\{4\}$ (see Figure 3 (a)), which is the lost degree of freedom, i.e, the end-effector can not rotate around X-axis of Frame{4} at wrist singularity.

In a design by [9], a four jointed spherical wrist was designed to handle the problem of singularity. In our method, a virtual revolute joint is added to compensate for the lost DOF. The diagram of the wrist with the virtual joint added is shown in Figure 3 (b).

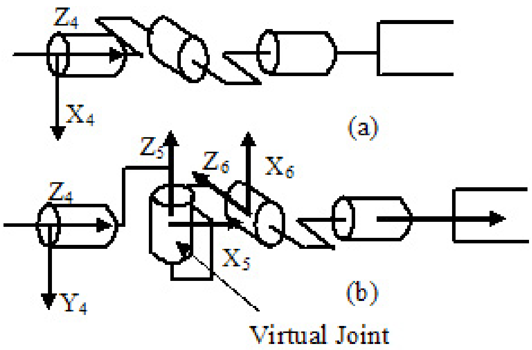

Figure 3: The diagram of the PUMA spherical wrist (a), and the wrist with added virtual joint (b)

Since the modified DH parameter requires choosing axis x_{i-1} along the common normal of axes z_{i-1} and z_i, (with direction from joint $i-1$ to joint i) the addition of virtual joint requires reassignment of DH frames. The new frame assignment for the wrist is shown in Figure 3 (b).

The additional joint is reflected on the lower right quadrant of the Jacobian with respect to Frame{4}:

$$ {}^{4}J_{22v} = \begin{pmatrix} 0 & 0 & -S_5 & -S_5C_6 \\ 0 & -1 & 0 & S_6 \\ 1 & 0 & C_5 & C_5C_6 \end{pmatrix} \quad (5) $$

where ${}^{4}J_{22v}$ is the virtual-joint-supplemented version of ${}^{4}J_{22}$, with the system assumed to be a 7-jointed mechanism.

The operational space formulation [6] was implemented in our system. The general equation relating the forces and torques to be sent to the robot joints is:

$$ \begin{aligned} \mathbf{\Gamma} &= J^T\mathbf{F} + \mathcal{N}^T\mathbf{\Gamma}_0 \\ \mathcal{N} &= [I - \bar{J}J] \end{aligned} \quad (6) $$

where $\mathbf{F}$ is the operational space force vector that incorporates robot dynamics and the control law for motion and force tracking , $\mathbf{\Gamma}$ is the torque vector to the sent to the joints, and $\mathbf{\Gamma}_0$ is the null space torque, which is the gradient descent of the potential functions to control the null space behavior of the system:

$$ \mathbf{\Gamma}_0 = -\nabla V_0(q) = k_i(q_i - q_{i(desired)})Z \quad (7) $$

Z is a vector of size $n+v$: $Z = [z_1 z_2 ... z_{n+v}]^T$, where $z_i = 0$ if q_i is a real joint and $z_i = 1$ if q_i is a virtual joint. For the inverse of the Jacobian, we use the the dynamically consistent inverse $\bar{J}$ to ensure null-space dynamics does not interfere with the end-effector dynamics. [5].

When implemented in velocity control, the equation would be defined as:

$$ \delta\mathbf{q} = \bar{J}\delta\mathbf{x} + \mathcal{N}\delta\mathbf{q}_0 \quad (8) $$

where δq is the joint velocity vector, δx is the cartesian velocity vector, and

$$ \delta\mathbf{q}_0 = -\nabla V_0(q) \quad (9) $$

The other two singularities (head and elbow) are position singularities (causes one or more of the top three rows of the Jacobian matrix to be zero). The same method as that in handling wrist singularity can be applied. However, it was shown in [4] that the lost DOF for elbow singularity (when elbow is straightened) is of very complex expression. This is because the singular direction is not aligned with any axes of the frames defined by the DH convention. Transforming the Jacobian into this frame can result in a very complicated matrix.

Following the example described in Figure 1, a more straight forward method can be done by adding three prismatic (virtual) joints with respect to the base frame, as opposed to supplying only two in the direction of lost degrees of freedom (for head and elbow singularities).

The resulting frame assignment is shown in Figure 4, where $Z_{1v}, X_{1v}, Z_{2v}, X_{2v}, Z_{3v}, X_{3v}$ denotes the frames of the virtual prismatic joints. The diagram continues to the rest of the PUMA robot, with

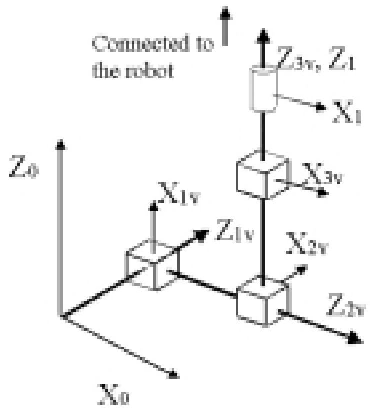

Figure 4: Three prismatic virtual joints added to the base to handle position singularities.

Frame{1} (denoted by Z_1, X_1) is the Frame{1} of PUMA robot as shown in Figure 2.

To incorporate the virtual joints to handle all the singularities (wrist, elbow, and head), the PUMA is now regarded as a 10-jointed mechanism, with the virtual joints labelled as joints 1, 2, 3, and 8. This results in the following 6×10 Jacobian matrix:

$$J = \left(\begin{array}{ccc|c|c} 0 & 1 & 0 & & \\ 1 & 0 & 0 & J_{11} & 0_{3\times4} \\ 0 & 0 & 1 & & \\ -- & -- & -- & ---- & ---- \\ & 0_{3\times3} & & J_{21} & {}^{0}J_{22v} \end{array} \right) \tag{10}$$

where J_{11} and J_{22} is as defined in (3).

The potential function projected into the null space to prevent the virtual joints from being relied upon to complete the task can be designed for the prismatic joints as:

$$V_0(\mathbf{q}) = \frac{1}{2} \sum_i k_i (q_i - q_{i(desired)})^2 \tag{11}$$

where $V_0(\mathbf{q})$ is the potential function for the virtual joints d_1, d_2, d_3 and q_8, d_i denotes prismatic joint and q_i revolute. The desired values are all set to zero.

5 Implementation Results

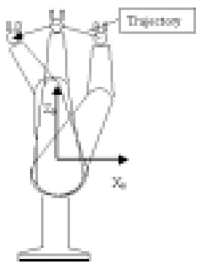

Figure 5: The trajectory of the PUMA in going through the combined wrist, elbow, and head singularities

Two sets of experimental results are presented in this paper. Figure 6 shows the performance of PUMA as it goes through a wrist singularity, and Figure 7 as it goes into a combined wrist, elbow and head singularity (when it points straight up vertically) and out of it (see Figure 5).

The PUMA is now regarded as a 10-joint mechanism, with original PUMA joints labelled as joint 4,5,6,7,9, and 10, and virtual joints as joint 1,2,3, and 8.

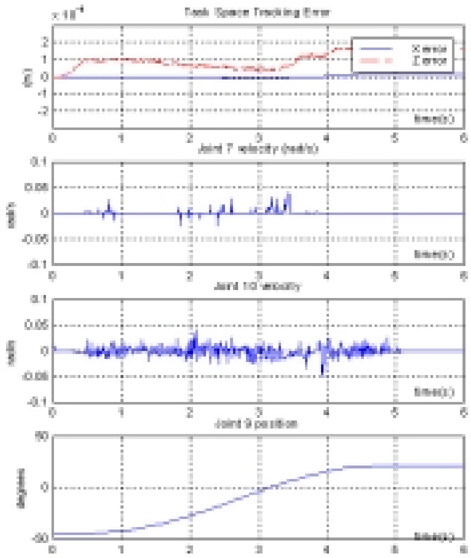

Figure 6: The result of the experiment, on tracking a trajectory through wrist singularity.

In going through the wrist singularity, the trajectory is a vertical path in X-Z plane with respect to the base frame (constant X, constant Y, increasing Z). (Figures 2 and 4 show the definition of base frame.)

The robot is shown to go through the wrist singularity, as in the bottom plot of Figure 6, where the wrist joint goes through $\theta_9 = 0$. The top plot shows the tracking error of the end-effector from the desired trajectory. Since the path is in increasing Z direction, while maintaining constant X and Y, little error was observed in X and Y direction. Maximum error of 0.2 mm in the Z-axis is comparable to the performance of the robot while tracking a non-singular path. Velocities for joints 7 and 10 of the PUMA robot (see Figure 2 for frame assignment) were shown to be stable. No sudden change or excessive velocity was observed (second and third plots in Figure 6).

Similar result are observed (Figure 7) as the robot follows a trajectory in X-Z plane to go into wrist, elbow and head singularity and out again (see Figure 5). The task space tracking performance (Figure 7 top) is comparable to that in non-singular path. The second and third plots of Figure 7 show that there is no sudden jerks or excessive joint rates while the robot goes through singularity. The plot of the determinant of the Jacobian matrix (Figure 7 confirms that the robot went through the singularity.

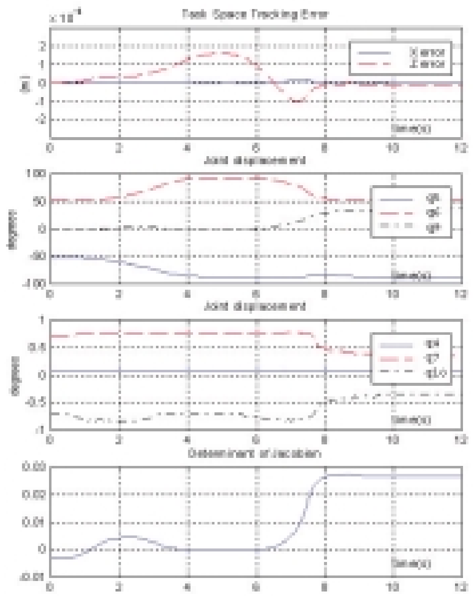

Figure 7: The result of the experiment, on tracking a trajectory through combined wrist, elbow, and head singularity.

6 Conclusion

The singularity handling method proposed in this paper is to supply 'extra joints' to the system in the lost degrees-of-freedom. This prevents the Jacobian from going rank-deficient as the manipulator enters singularity. Null space projection together with proper choice of potential functions to control the virtual joints ensure that the tasks are not compromised. The method has been shown to work well, and is able to go through singular configuration in a stable manner. The advantage is shown in the result as having a smooth continuous motion through the singular region, as there is no switching in control algorithm in the vicinity of singularity. Task in the lost degree-of-freedom during singular configuration is still not feasible.

References

[1] 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.

[2] 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.

[3] 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.

[4] Denny Oetomo, Marcelo Ang, and Ser Yong 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] Eric Schwartz and Keith Doty, "Derivation of redundant wrist manipulators to avoid interior workspace singularities," *IEEE Conf. Proc. of Southeastcon'88*, pp. 403–407, 1988.

[10] Etemadi Zangareh and Jorge Angeles, "On the isotropic design of general six-degree-of-freedom parallel manipulators," *Proc. Computational Kinematics Workshop*, pp. 213–220, Sept 1995.

[11] J. Burdick B. Armstrong, O. Khatib, "The explicit dynamic model and inertial parameters of the puma 560 arm," *IEEE Intl. Conf. Robotics and Automation*, pp. 510–518, 1986.

[12] V.D. Tourassis and M.H. Ang Jr., "Identification and analysis of robot manipulator singularities," *International Journal of Robotics Research*, vol. 11, no. 3, pp. 248–259, 1991.