Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

ThB05.6

Set-Point Navigation of a Redundant Robot in Uncertain Environments using Finite Range Sensors

Apoorva Kapadia, Enver Tatlicioglu, and Darren M. Dawson

Abstract: In this work, control of redundant robot manipulators in an uncertain environment is considered. The manipulator is equipped with finite range sensors to detect obstacles in its workspace. A navigation function-based kinematic controller is proposed to ensure the regulation of the end-effector to a desired set-point while the entire manipulator simultaneously avoids the obstacle points detected by the sensors. A joint-space controller is then utilized to ensure asymptotic tracking of the desired joint-space trajectory.

I. INTRODUCTION

While there are multiple definitions available to explain redundancy in robots [1], the most widely used form in the literature is kinematic redundancy, where the number of joints of a robot manipulator is greater than the degrees of freedom in its task-space. This property of redundant robot manipulators is essential for various applications, especially those requiring the manipulator to perform complex tasks. As explained in [2], [3], [4] and [5], there are an infinite number of solutions possible for the inverse kinematics since redundant robot manipulators can have multiple joint configurations for the same end-effector position. As a result, there exists joint motion which can be propagated in the null-space of the manipulator Jacobian matrix without affecting the end-effector position, a phenomenon commonly referred to as self-motion.

Some of the previous research has focused on utilizing the kinematic redundancy of such manipulators to achieve task-space tracking while also meeting various secondary objectives. Seraji [6] explained configuration control methods by achieving task-space tracking while using the redundancy of the robot manipulator to satisfy kinematic and dynamic constraints. Hsu *et al.* [7] outlined a control law that guaranteed tracking of a desired end-effector trajectory while controlling the joint velocities of the redundant manipulator. Kircanski and Vukobratovic [8] discussed the use of kinematic redundancy to simplify the control problem in obstructed workspaces. Zergeroglu *et al.* [9] developed an

adaptive controller for redundant manipulators to compensate for uncertainties in the dynamic model by proposing a non-linear controller dependent on the system model to perform task-space tracking while meeting sub-task objectives. This issue was further addressed by Tatlicioglu *et al.* [10] by implementing the controller from [11] while integrating the sub-task controller in the stability analysis.

Past research has also analyzed utilizing the self-motion property of redundant manipulators to avoid obstacles in its workspace. Maciejewski *et al.* [12], and Lozano-Perez [13] proposed control schemes to avoid obstacles present in the workspace of the redundant robot manipulator. Baillieul [14] presented a method realizing constrained motions for the joint variables of a redundant robot manipulator to avoid disc-like obstacles. Jou *et al.* [15] developed one of the first methods for a 3-link revolute planar redundant robot manipulator to navigate through a known environment with pre-defined disc-like obstacles which could be extended to more general shapes. In order to avoid collisions between the manipulator and the obstacles, the authors defined two regions; the obstructed region and the collision-free region in which manipulator operated, with the algorithm also ensuring singularity avoidance and minimum joint rate. Khatib [16] presented a real-time obstacle avoidance approach for general manipulators using the artificial potential field method first outlined by Koditschek and Rimon [17] which could be adapted for redundant robot manipulators. Nemec and Zlajpah [18] devised a force control method for redundant manipulators operating in an unstructured static environment. To compensate for the unknown workspace, Nemec and Zlajpah allowed the manipulator to bump into the obstacle while ensuring that the resulting forces are minimized to prevent damage to the manipulator and the obstacle. Charifa and Masoud [19] described a potential-field based motion planner for a mobile robot in a complex, static unknown environment using information acquired from finite range sensors. The aforementioned papers were susceptible to the local minima problem inherent in potential-field approaches as well as potentially damaging contact between the manipulator joints and the obstacles, issues that are still open research problems.

In this paper, a kinematically redundant robot manipulator, equipped with finite range sensors, is considered to be placed in an uncertain environment. These sensors detect obstacles as points that must be avoided by the entire manipulator. The control objective is to regulate the robot manipulator's end-effector to a desired set-point while simultaneously avoiding the unknown obstacles in the workspace as they are detected. To meet the control objectives, a kinematic controller is

Contact Author: Apoorva Kapadia is with the Department of Electrical & Computer Engineering, Clemson University, Clemson, SC 29634-0915 (akapadi@clemson.edu).
Enver Tatlicioglu is with the Department of Electrical & Electronics Engineering, Izmir Institute of Technology, Urla, Izmir 35430 Turkey (enver@envertatlicioglu.com).
Darren M. Dawson is with the Department of Electrical & Computer Engineering, Clemson University, Clemson, SC 29634-0915 (darren.dawson@ces.clemson.edu).

presented in the form of two sub-controllers namely, the task-space and null-space controllers. The end-effector controller (task-space controller) drives the end-effector to a desired set-point by utilizing a navigation function-based control scheme. Taking advantage of self-motion in kinematically redundant robots, the null-space controller repels the body of the manipulator away from the detected obstacle points while the end-effector is driven to the goal point. A desired joint level trajectory is developed by filtering the kinematic controller [20] and the control algorithm developed by Xian *et al.* [21] is then utilized as the jont-space controller. This controller provides asymptotic tracking of a desired trajectory for a class of non-linear systems with uncertain dynamic models.

The paper is organized as follows, Section II defines and develops the robot models as well as the necessary functions for controller development. Section III describes the development of the primary controller, *i.e.* the end-effector controller, along with the associated stability analysis. Section IV describes the null-space controller and the accompanying stability analysis. Section V details the robot manipulator trajectory generator and the joint-space controller. We end with concluding remarks in Section VI and some of the important properties of the Moore-Penrose pseudo-inverse are provided in Appendix I.

II. MODEL DEVELOPMENT

An n-joint, revolute, direct drive redundant robot manipulator ($n \geq 6$) is equipped with $nd \in \mathbb{R}^+$ finite-range sensors mounted on its surface. These sensors are used to identify obstacles that might obstruct the path of the manipulator as it traverses the workspace.

A. Kinematic Model

Using the Denavit-Hartenberg method described in [22], the kinematic model for the n-joint robot manipulator is described as

$$x_i = f_i(q), \tag{1}$$

where $i \in \{1, nd\}$ represents the detection points with nd being the number of sensors and $x_i(t) \in \mathbb{R}^m$ are the coordinates of the ith detection point. Without loss of generality, the sensors are assumed to be equidistant from each other and uniformly distributed over the links of the robot manipulator. The final sensor is assumed to be placed on the end-effector, resulting in $x_{nd} = x_e$ where $x_e(t) \in \mathbb{R}^m$ representing the task-space position of the end-effector. In (1), $f_i(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is the forward kinematics of the manipulator, while $q(t) \in \mathbb{R}^n$ refers to the manipulator joint position. The velocity kinematics for the manipulator can be obtained by taking the time derivative of the kinematic model in (1)

$$\dot{x}_i = J_i(q)\dot{q}, \tag{2}$$

where $J_i(\cdot) \in \mathbb{R}^{m \times n}$ is the Jacobian matrix defined as follows

$$J_i \triangleq \frac{\partial f_i(\cdot)}{\partial q}, \tag{3}$$

where $\dot{x}_i(t)$ represents the task-space velocity and $\dot{q}(t)$ is the joint velocity.

B. Dynamic Model

The dynamic model of the robot manipulator is assumed to be of the following form

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau(t), \tag{4}$$

where $\ddot{q}(t)$ refers to the joint acceleration, $M(\cdot) \in \mathbb{R}^{n \times n}$ is the unknown inertia matrix, $N(\cdot) \in \mathbb{R}^n$ represents the unknown dynamic effects due to Centripetal and Coriolis forces, gravity and dynamic friction, and $\tau(t) \in \mathbb{R}^n$ is the control input torque vector. The subsequent development requires the standard assumption that $M(q)$ is symmetric and positive-definite, and satisfies the following inequalities [23]

$$m_1\|\xi\|^2 \leq \xi^T M(q)\xi \leq m_2\|\xi\|^2 \quad \forall \xi \in \mathbb{R}^n,$$

where $m_1, m_2 \in \mathbb{R}$ are positive constants and $\| \cdot \|$ denotes the standard Euclidean norm. In the subsequent development, it will be assumed that $M(q), N(q, \dot{q}) \in \mathcal{C}^2$, and $M(\cdot), \dot{M}(\cdot), \ddot{M}(\cdot), N(\cdot), \dot{N}(\cdot), \ddot{N}(\cdot) \in \mathcal{L}_\infty$ provided that $q(t), \dot{q}(t), \ddot{q}(t), \dddot{q}(t) \in \mathcal{L}_\infty$. It is also assumed that $q(t), \dot{q}(t) \in \mathbb{R}$ are measurable.

The primary control objective is to ensure that the end-effector is regulated to a desired set-point, $x^* \in \mathbb{R}^m$. The self-motion of the redundant robot manipulator is utilized to introduce a secondary control objective to avoid the obstacles while keeping the entire manipulator inside a restricted workspace. To facilitate the design of the self-motion controller, an obstacle detection function and a workspace boundary function are introduced. To assist in the development of these functions, a smooth bump function $\rho_h(\cdot) : \mathbb{R}_+ \rightarrow [0, 1]$, where $\rho_h(x)$ is defined as

$$\rho_h(x) \triangleq \begin{cases} 1 & \text{for} \quad 0 \leq x \in [0, h] \\ \frac{1}{2}\left\{1 + cos\left(\pi\frac{x-h}{1-h}\right)\right\} & \text{for} \quad x \in (h, 1) \\ 0 & otherwise. \end{cases} \tag{5}$$

where $h \in (0, 1)$ is a positive constant. For different values of h, the smooth bump function is given in Figure 1. An important property of the smooth bump function is that it allows for an analytically smooth transition over a unit interval, with the derivatives of the function being equal to zero at the end-points. Given that the sensors have a finite range, the smooth bump function provides an analytical switch from zero to one of the repulsion due to the obstacles present in the workspace and the workspace boundary.

C. Workspace Boundary Function

As seen in [24], the workspace boundary function for the robot manipulator denoted by $\beta_w(\cdot) \in \mathbb{R}$ is defined as follows

$$\beta_w \triangleq \prod_{i=1}^{nd} \rho_{h_w}\left(\frac{1}{r_w - r_{s,min}}\|x_i - x_w\|\right), \tag{6}$$
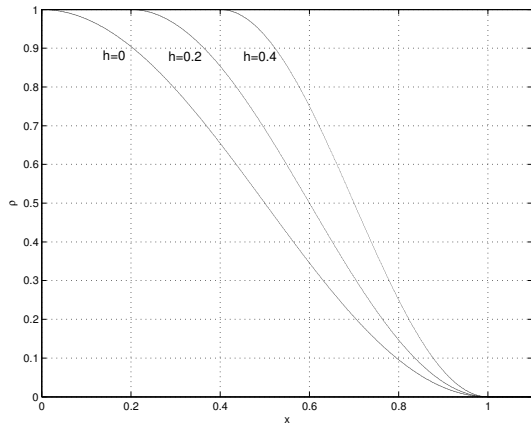
Fig. 1.   Smooth Bump Function Plots for Different Values of h.

where $r_w \in \mathbb{R}$ is the radius of the workspace, $x_w \in \mathbb{R}^3$ is the coordinates of the workspace center, $r_{s,max} \in \mathbb{R}$ and $r_{s,min} \in \mathbb{R}$ are the sensor's maximum and minimum range, respectively, and $h_w \triangleq \frac{r_w - r_{s,max}}{r_w - r_{s,min}}$. The application of the smooth bump function in (5) on the workspace boundary is denoted by ρ_{h_w}. The boundary function establishes the relationship between the workspace boundary and the position of the joints of the manipulator and its objective is to prevent any manipulator link from coming into contact with any part of the workspace edge. From (6), it can be seen that $\beta_w(\cdot) \in \mathbb{R}$ satisfies

$$\beta_w = \begin{cases} 0 & \text{At least one manipulator link makes} \\ & \text{contact with the workspace boundary.} \\ \in (0,\ 1] & \text{Otherwise.} \end{cases}$$

D. Obstacle Detection Function

The obstacles are detected as points that must be avoided by the manipulator as the end-effector of the manipulator moves towards its goal position. If any of the manipulator links approaches the ith obstacle ($i \in \mathbb{N}$), it is detected using the following function

$$\beta_i \triangleq 1 - \rho_{h_o}\left(\frac{1}{r_{s,max}}\|x_i - x_o\|\right), \qquad (7)$$

where $h_o \triangleq \frac{r_{s,min}}{r_{s,max}}$ and $x_o \in \mathbb{R}^m$ is the coordinates of the detected obstacle point that is to be avoided and its use in the smooth bump function is given by ρ_{h_o}. It can be seen that obstacle function $\beta_i(\cdot) = 0$ when the manipulator touches an obstacle while $\beta_i(\cdot) > 0$ when no contact is made.

Remark 1: The sensors can also detect manipulator links as obstacle points, allowing our algorithm to also be used to assist the manipulator avoid its joint limits.

Remark 2: It should be noted that the control methodology presented in this paper could also be implemented using calibrated cameras instead of the finite range sensors. The camera could detect the edges of the obstacles in its field of view as feature points which can be tracked as obstacle points to be avoided.

E. Navigation Function Development

While navigating from its initial position to the goal point, the manipulator has to remain inside the workspace while simultaneously avoiding obstacles. Thus the free configuration space of the manipulator $\mathcal{D} \subset \mathbb{R}^m$, is defined as a subset of the entire workspace where all the manipulator configurations involving collisions with the detected obstacle points are removed. It is assumed that both the initial and final positions are in $\mathcal{D}$. The navigation function $\varphi(x_e) \in \mathbb{R}$ that will be used to regulate the end-effector to the desired set-point is based on the definition in [17]

$$\varphi(x_e) \triangleq \frac{\|x_e - x^*\|^2}{\left(\|x_e - x^*\|^{2\kappa} + \beta\right)^{\frac{1}{\kappa}}}, \qquad (8)$$

where $\kappa \in \mathbb{R}$ is a positive constant and $\beta(t) \in \mathbb{R}$ is defined as follows

$$\beta \triangleq \beta_w \prod_{i=1}^{nd} \beta_i. \qquad (9)$$

It should be noted that $\varphi(x_e)$ is designed such that when $\varphi(x_e) \in \mathcal{L}_\infty$, then $x_e(t) \in \mathcal{L}_\infty$.

Remark 3: It should be noted that Koditschek and Rimon [25] formally showed that it was not mathematically possible to construct a navigation function $\varphi(x_e)$, that satisfies $\frac{\partial}{\partial x_e}\varphi(x_e) = 0$ only at $x_e(t) = x_e^*$ thus having strict global navigation capabilities. This is because the appearance of unstable equilibrium points (also called saddle points) as local minima in the free configuration space of the manipulator is inevitable, as shown in [17]. However, this does not cause problems in practice because the domain of attraction of these saddle points is infinitely thin. For more information on navigation functions, see [24] and [25].

III. Control Development

A. Kinematic Controller Design

For the design of the kinematic controller, it is assumed that the manipulator joint velocities will serve as the control input such that

$$\dot{q} = u, \qquad (10)$$

where $u(t) \in \mathbb{R}^n$ is of the following form

$$u \triangleq J_e^+ u_e + \left(I_n - J_e^+ J_e\right) u_m, \qquad (11)$$

where $I_n \in \mathbb{R}^{n \times n}$ is the standard identity matrix, $J_e = J_{nd}$, $J_e^+(q) \in \mathbb{R}^{n \times m}$ is the pseudo-inverse[1] of $J_e(t)$ defined as $J_e^+ \triangleq J_e^T \left(J_e J_e^T\right)^{-1}$, $u_e(t) \in \mathbb{R}^m$ is the end-effector controller and $u_m(t) \in \mathbb{R}^n$ is the null-space controller.

Remark 4: It is assumed that the minimum singular value of the manipulator Jacobian matrix, denoted by ϕ_m, is lower bounded by a known constant $\delta \in \mathbb{R}^+$. This ensures $\max\{\|J_n^+(q)\|\}$ is always known a priori allowing for all kinematic singularities to be avoided.

---

[1]The properties of the pseudo-inverse used in the development of our controller are provided in Appendix I.

B. Path Planner Design

Based on the subsequent stability analysis, the end-effector controller is designed as follows

$$u_e \triangleq -K_e \left(\frac{\partial \varphi (x_e)}{\partial x_e} \right)^T, \qquad (12)$$

where $K_e \in \mathbb{R}^{m \times m}$ is a positive definite, diagonal gain matrix.

C. Stability Analysis

Theorem 1: Contingent upon $x_e(0) \in \mathcal{D}$ the end-effector controller in (12) ensures $x_e(t) \in \mathcal{D} \, \forall t$ and the end-effector is driven to the desired goal point such that

$$x_e(t) \to x^* \text{ as } t \to \infty. \qquad (13)$$

Proof: Refer to [24] for a similar result. ∎

IV. NULL SPACE CONTROLLER

To harness the property of self-motion for redundant robot manipulators, a null-space controller is designed to ensure the end-effector as well as the body of the manipulator are repelled from any detected obstacle, while the end-effector controller simultaneously navigates the end-effector to the desired goal point. An auxiliary positive function $y_a(t) \in \mathbb{R}$ is defined as

$$y_a \triangleq \sum_{i=1}^{nd} h_i (x_i), \qquad (14)$$

where $h_i (x_i) \triangleq k_{hi} exp \left(-\alpha_i \beta_i^2 (x_i) \right)$, where $k_{hi}, \alpha_i \in \mathbb{R}$ are positive constants, $\beta_i (x_i) \in \mathbb{R}$ is the obstacle detection function defined in (7) and $exp (\cdot)$ is the standard logarithmic exponential function. The usefulness of this auxiliary positive function is due to the fact that it relates the geometric information of the surface of an obstacle to the manipulator joint positions while attempting to keep the manipulator body away from the obstacles, utilizing the self-motion property of redundant manipulators. For a detailed explanation of this auxiliary function, refer to [10].

The following expression can be obtained from the time derivative of (14)

$$\dot{y}_a = \frac{\partial y_a (x_1, \dots, x_e)}{\partial \left[x_1^T \dots x_e^T \right]} \begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_e \end{bmatrix}. \qquad (15)$$

From (2) and (10), we obtain

$$\dot{x}_i = J_i u \, \forall i. \qquad (16)$$

After substituting (16) into (15), we get

$$\dot{y}_a = J_s u, \qquad (17)$$

where $J_s(t) \in \mathbb{R}^{1 \times n}$ is a Jacobian-type vector

$$J_s \triangleq \frac{\partial y_a (x_1, \dots, x_e)}{\partial \left[x_1^T \dots x_e^T \right]} \begin{bmatrix} J_1 \\ \vdots \\ J_e \end{bmatrix}. \qquad (18)$$

After substituting (11) into (17), the following expression is obtained

$$\dot{y}_a = J_s J_e^+ u_e + J_s \left(I_n - J_e^+ J_e \right) u_m. \qquad (19)$$

Based on the subsequent stability analysis, the null-space controller is designed as

$$u_m \triangleq -k_m \left[J_s \left(I_n - J_e^+ J_e \right) \right]^T y_a, \qquad (20)$$

where $k_m \in \mathbb{R}$ is a positive constant. After substituting (20) into (19) the following expression is obtained

$$\dot{y}_a = J_s J_e^+ u_e - k_m \left| \left| J_s \left(I_n - J_e^+ J_e \right) \right| \right|^2 y_a. \qquad (21)$$

A. Stability Analysis

Theorem 2: The null-space controller described in (20) guarantees that $y_a(t)$ is globally uniformly ultimately bounded such that

$$|y_a| \leq \sqrt{y_a^2 (t_0) \, exp(-2\mu t) + \frac{\omega}{\mu}}, \qquad (22)$$

provided the following sufficient condition is satisfied

$$k_m > \frac{1}{\delta_1 \delta_2}, \qquad (23)$$

where ω, μ, δ_1 and $\delta_2 \in \mathbb{R}$ are positive constants.

Proof: See [10] for a similar proof. ∎

V. DYNAMIC CONTROLLER

In this section, a desired joint trajectory is designed. The structure of the desired trajectory generator is influenced by the subsequently designed joint-space controller [21] which requires that the desired trajectory be bounded up to its fourth order derivative.

A. Desired Trajectory Generator

The desired joint trajectory $q_d(t) \in \mathbb{R}^n$ is generated using the following expression [20]

$$q_d(s) \triangleq \frac{1}{\left(\frac{s}{\epsilon} + 1 \right) \left(\frac{s}{\kappa} + 1 \right)^3} Sat (u), \qquad (24)$$

where $s \in \mathbb{C}$ is the Laplace variable, $\epsilon \in \mathbb{R}$ is a small positive constant and $\kappa \in \mathbb{R}$ is a large positive constant and $u(t)$ is the kinematic controller defined in (11). In (24), $Sat(\cdot) \in \mathbb{R}^n$ is the vector saturation function defined as follows

$$Sat(\eta) \triangleq \begin{bmatrix} sat (\eta_1) & sat (\eta_2) & \dots & sat (\eta_n) \end{bmatrix}^T, \quad (25)$$

where $\eta = [\eta_1 \, \eta_2 \, \dots \, \eta_n]^T \in \mathbb{R}^n$ and $sat (\eta_i) \in \mathbb{R} \, \forall i = 1, \dots, n$ is defined as

$$sat (\eta_i) \triangleq \begin{cases} -\eta_{min} & \text{if} \quad \eta_i \leq -\eta_{min} \\ \eta_i & \text{if} \quad \eta_i > -\eta_{min} \text{ or } \eta_i < \eta_{max} \\ \eta_{max} & \text{if} \quad \eta_i \geq \eta_{max}. \end{cases}$$

where η_{min} and $\eta_{max} \in \mathbb{R}$ are the upper and lower saturation values, respectively. The trajectory generator in (24) implies that $q_d(t), \dot{q}_d(t), \ddot{q}_d(t), \dddot{q}_d (t), \ddddot{q}_d (t) \in \mathcal{L}_\infty$.

B. Control Development

The control objective is to ensure that the manipulator asymptotically tracks the desired trajectory $q_d(t)$ such that $q(t) \rightarrow q_d(t)$ as $t \rightarrow \infty$. To ensure asymptotic tracking, we define an error signal $e_1(t) \in \mathbb{R}^n$ such that

$$e_1 \triangleq q_d - q. \qquad (26)$$

To facilitate in the control development, a filtered tracking error signal denoted by $e_2(t) \in \mathbb{R}^n$ is defined as

$$e_2 \triangleq \dot{e}_1 + \Upsilon_1 e_1, \qquad (27)$$

where $\Upsilon_1 \in \mathbb{R}^+$ is a positive control gain.

Since the dynamic model of the robot manipulator in (4) is assumed to be uncertain, to achieve the tracking objective detailed above, the control law outlined in [21] is applied as the joint-space controller such that

$$\tau = (K_c + I_n)\left(e_2(t) - e_2(t_0) + \Upsilon_2 \int_{t_0}^t e_2(\Omega)\, d\Omega\right)$$

$$+ \Lambda \int_{t_0}^t Sgn(e_2(\Omega))\, d\Omega, \qquad (28)$$

where $\Upsilon_2 \in \mathbb{R}^+$ is a positive constant while K_c and $\Lambda \in \mathbb{R}^{n \times n}$ are positive definite diagonal control gain matrices. The vector signum function $Sgn(\cdot) \in \mathbb{R}^n$ is defined as

$$Sgn(\eta) \triangleq \left[\begin{array}{cccc} sgn(\eta_1) & sgn(\eta_2) & \dots & sgn(\eta_n) \end{array}\right]^T, \qquad (29)$$

$\forall \eta = [\eta_1\ \eta_2\ \dots\ \eta_n]^T \in \mathbb{R}^n$. The use of this controller was facilitated by its ability to compensate for the uncertainties present in the dynamic model.

VI. Conclusion

In this paper, a navigation function-based controller for a kinematically redundant robot manipulator, equipped with finite-range sensors, in an uncertain environment was developed. A kinematic controller was presented that navigates the end-effector of the robot manipulator to a desired setpoint while simultaneously repelling the entire manipulator away from the obstacles all along forcing it to remain in the workspace. The desired joint-space trajectory was generated by filtering the kinematic controller and the control developed in [21] was applied to ensure asymptotic tracking of the desired trajectory while compensating for the uncertainties in the manipulator dynamic model.

Appendix I
Pseudo-inverse Properties

For the development of the kinematic controller, the pseudo-inverse of $J_e(q)$, denoted by $J_e^+(q) \in \mathbb{R}^{n \times m}$, which is defined as

$$J_e^+ \triangleq J_e^T \left(J_e J_e^T\right)^{-1}, \qquad (30)$$

where $J_e^+(t)$ is given by

$$J_e J_e^+ = I_m, \qquad (31)$$

and $I_m \in \mathbb{R}^{m \times m}$ is the standard identity matrix. As described in [3], the pseudo-inverse defined in (30) satisfies the Moore-Penrose conditions given by

$$\begin{array}{cc} J_e J_e^+ J_e = J_e & J_e^+ J_e J_e^+ = J_e^+ \\ (J_e^+ J_e)^T = J_e^+ J_e & (J_e J_e^+)^T = J_e J_e^+ \end{array} \qquad (32)$$

In addition the the properties listed above, the matrix $(I_n - J_e^+ J_e)$ also satisfies these useful properties,

$$\begin{array}{c} (I_n - J_e^+ J_e)(I_n - J_e^+ J_e) = I_n - J_e^+ J_e \\ (I_n - J_e^+ J_e)^T = (I_n - J_e^+ J_e) \\ J_e(I_n - J_e^+ J_e) = 0_{m \times n} \\ (I_n - J_e^+ J_e) J_e^+ = 0_{n \times m}. \end{array} \qquad (33)$$

References

[1] E. Conkur and R. Buckingham, "Clarifying the definition of redundancy as used in robotics," *Robotica*, vol. 15, no. 5, pp. 583–586, Aug. 1997.
[2] I. D. Walker and S. I. Marcus, "Subtask performance by redundancy resolution for redundant robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 4, no. 3, pp. 350–354, Jun. 1988.
[3] Y. Nakamura, *Advanced Robot Redundancy and Optimization*. Reading, MA: Addison-Wesley, 1991.
[4] D. Nenchev, "Redundancy resolution through local optimization: A review," *Journal of Robotic Systems*, vol. 6, no. 6, pp. 769–798, 1989.
[5] B. Siciliano, "Kinematic control of redundant robot manipulators," *Jour. of Intelligent and Robotic Systems*, vol. 3, no. 3, pp. 201–212, Mar. 1990.
[6] H. Seraji, "Configuration control of redundant manipulators," *IEEE Trans. Robot. Autom.*, vol. 5, no. 4, pp. 472–490, Aug. 1989.
[7] P. Hsu, J. Hauser, and S. Sastry, "Dynamic control of redundant manipulators," *J. Robot. Sys.*, vol. 6, no. 2, pp. 133–148, 1989.
[8] M. Kircanski and M. Vukobratovic, "Contribution to control of redundant robotic manipulators in an environment with obstacles," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 112–119, 1986.
[9] E. Zergeroglu, D. M. Dawson, I. D. Walker, and P. Setlur, "Nonlinear tracking control of kinematically redundant robot manipulators," *IEEE Trans. Mechatronics*, vol. 9, no. 1, pp. 129–132, Mar. 2004.
[10] E. Tatlicioglu, M. L. McIntyre, D. M. Dawson, and I. D. Walker, "Adaptive nonlinear tracking control of kinematically redundant robot manipulators," *Intl. J. Robotics and Automation*, vol. 23, no. 2, pp. 98–105, 2008.
[11] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Boston, MA: Birkhauser, 2003.
[12] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 109–117, 1985.
[13] T. Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE Trans. Comp.*, vol. C-32, no. 2, pp. 108–120, Feb. 1983.
[14] J. Baillieul, "Avoiding obstacles and resolving kinematic redundancy," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1986, pp. 1698–1704.
[15] C.-C. Jou, C.-J. Lin, and K.-Y. Young, "Collision avoidance analysis of reachable workspaces for redundant manipulators," in *Proc. Int. Conf. Adv. Robot.*, Grenoble, Switzerland, 1991, pp. 1222–1227.
[16] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, St. Louis, MO, 1985, pp. 500–505.
[17] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. & Autom.*, vol. 8, no. 5, pp. 501–518, Oct. 1992.
[18] B. Nemec and L. Zlajpah, "Force control of redundant robots in unstructured environments," *IEEE Trans. Indus. Elec.*, vol. 49, no. 1, pp. 233–240, Feb. 2002.
[19] S. M. Charifa and A. A. Masoud, "Solid mechanics inspired sensor based motion planner," in *Proc. Conf. Control App.*, Toronto, Canada, 2005, pp. 221–226.
[20] D. Braganza, M. L. McIntyre, D. M. Dawson, and I. D. Walker, "Whole arm grasping control for redundant robot manipulators," in *Proc. American Control Conf.*, Minneapolis, MN, 2006, pp. 3194–3199.

[21] B. Xian, D. M. Dawson, M. S. D. Queiroz, and J. Chen, "A continuous asymptotic tracking control strategy for uncertain nonlinear systems," *IEEE Trans. Auto. Cont.*, vol. 49, no. 7, pp. 1206–1211, Jul. 2004.

[22] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York, NY: John Wiley and Sons, 1989.

[23] F. L. Lewis, D. M. Dawson, and C. Abdallah, *Robot Manipulator Control: Theory and Practice*. Boca Raton, FL: CRC, 2003.

[24] J. Chen, D. M. Dawson, M. Salah, and T. Burg, "Multiple uav navigation with finite sensing zone," in *Proc. American Control Conf.*, Minneapolis, MN, 2006, pp. 4933–4938.

[25] D. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Adv. App. Math.*, vol. 11, no. 4, pp. 412–442, Dec. 1990.