Introduction to Robotics

Lecture 5

Lasse Einig, Jianwei Zhang
[einig, zhang]@informatik.uni-hamburg.de

University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
Technical Aspects of Multimodal Systems

May 03, 2018

Outline

Outline (cont.)

Jacobian of a Manipulator

Definition

- A Jacobian-matrix is a multidimensional representation of partial derivatives.
- The Jacobian of a manipulator links the joint velocities with the cartesian velocity of the TCP.
- The Jacobian matrix depends on the current state of the robot joints.

Jacobian of a Manipulator (cont.)

- Consider an n-link manipulator with joint variables $q_1, q_2, ..q_n$.
- Define $q = [q_1, q_2, ..q_n]^T$
- Let the transformation from base to end-effector frame be:

$$T = \begin{bmatrix} R_n^0(q) & o(q) \\ 0 & 1 \end{bmatrix} \tag{52}$$

- We define ω_n^0 to be the angular velocity of the end-effector
- The linear velocity of the end-effector is v_n^0
- The **Jacobian** matrix consists of two components, that solve the following equations:

$$v_n^0 = J_v \dot{q} \qquad \text{and} \qquad \omega_n^0 = J_w \dot{q}$$

Jacobian of a Manipulator (cont.)

The manipulator Jacobian

$$J := \begin{bmatrix} J_v \\ J_w \end{bmatrix}$$

We define the body velocity of the endeffector:

$$\xi := \begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} := \begin{bmatrix} {}^T d_x \\ {}^T d_y \\ {}^T d_z \\ {}^T \delta_x \\ {}^T \delta_y \\ {}^T \delta_z \end{bmatrix} \qquad \xi = J\dot{q}$$

Angular Velocity Jacobian

Revolute joints

If the i^{th} joint is revolute, the axis of rotation is given by z_{i-1}.

Let $\omega_{i-1,i}^{i-1}$ represent the angular velocity of the link i w.r.t. the frame $i-1$.

Then, we have: $\omega_{i-1,i}^{i-1} = \dot{q}_i z_{i-1}^{i-1}$

Prismatic joints

If the i^{th} joint is prismatic, the motion of frame i relative to frame $i-1$ is a translation.

Then, we have: $\omega_{i-1,i}^{i-1} = 0$

Angular Velocity Jacobian (cont.)

Overall angular velocity:

$$\omega_{0,n}^{0} = \omega_{0,1}^{0} + R_{1}^{0}\omega_{1,2}^{1} + \ldots + R_{n-1}^{0}\omega_{n-1,n}^{n-1} \tag{53}$$

We get:

$$\omega_{0,n}^{0} = p_1\dot{q}_1 z_0^0 + p_2\dot{q}_2 R_1^0 z_1^1 + \ldots + p_n\dot{q}_n R_{n-1}^0 z_{n-1}^{n-1} \tag{54}$$

$$= p_1\dot{q}_1 z_0^0 + p_2\dot{q}_2 z_1^0 + \ldots + p_n\dot{q}_n z_{n-1}^0 \tag{55}$$

where:

$$p_i = \begin{cases} 0 & \text{if i is prismatic} \\ 1 & \text{if i is revolute} \end{cases} \tag{56}$$

Angular Velocity Jacobian (cont.)

The complete Jacobian

$$\begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} = \begin{bmatrix} J_v \\ J_w \end{bmatrix} \dot{q} \tag{57}$$

The Angular Velocity Jacobian

$$J_w = [p_1 z_0^0 \quad p_2 z_1^0 \quad \dots \quad p_n z_{n-1}^0] \tag{58}$$

(Hint: J_w is a $3xn$ matrix; due to matrix multiplication rules the representation is equal to those on the last slide.)

Linear Velocity Jacobian

The linear velocity of the end effector is: $\dot{o}_n^0$

By the chain rule of differentiation:

$$\dot{o}_n^0 = \frac{\delta o_n^0}{\delta q_1}\dot{q}_1 + \frac{\delta o_n^0}{\delta q_2}\dot{q}_2 + \dots + \frac{\delta o_n^0}{\delta q_n}\dot{q}_n \tag{59}$$

therefore the linear part of the Jacobian is:

$$J_v = \begin{matrix} \frac{\delta o_n^0}{\delta q_1} & \frac{\delta o_n^0}{\delta q_2} & \dots & \frac{\delta o_n^0}{\delta q_n} \end{matrix} \tag{60}$$

Linear Velocity Jacobian – Prismatic

Every prismatic joint influences the velocity of the endeffector depending on:

- the current linear velocity of the joint ($\dot{d}_i$)
- the current orientation of the z-axis of the joint (z_{i-1})
  - depending on q

$$\dot{o}_n^0 = \dot{d}_i z_{i-1} \tag{61}$$

Therefore:

$$J_{v_i} = \frac{\delta o_n^0}{\delta q_n} = z_{i-1} \tag{62}$$

Linear Velocity Jacobian – Revolute

Every revolute joint influences the velocity of the end-effector depending on:

- the current angular velocity of the joint ($\dot{q}_i$)
- the current orientation of the z-axis of the joint (z_{i-1})
- the current vector from the joint origin o_{i-1} to the end-effector
  - the two latter depending on q

The linear velocity of the end-effector is of form:

$$\omega \times r$$

with $\omega = \dot{q}_i z_{i-1}$ and $r = o_n^0 - o_{i-1}^0$

Therefore:

$$J_{v_i} = \frac{\delta o_n^0}{\delta q_n} = z_{i-1} \times (o_n^0 - o_{i-1}^0) \qquad (63)$$

Computing the final Jacobian

$$J := \begin{bmatrix} J_v \\ J_w \end{bmatrix}$$

$$J_v = \begin{bmatrix} J_{v_1} & J_{v_2} & J_{v_n} \end{bmatrix} \quad \text{with} \tag{64}$$

$$J_{v_i} = \begin{cases} z_{i-1} & \text{if i is prismatic} \\ z_{i-1} \times (o_n^0 - o_{i-1}^0) & \text{if i is revolute} \end{cases} \tag{65}$$

and

$$J_w = \begin{bmatrix} J_{w_1} & J_{w_2} & J_{w_n} \end{bmatrix} \quad \text{with} \tag{66}$$

$$J_{w_i} = \begin{cases} 0 & \text{if i is prismatic} \\ z_{i-1} & \text{if i is revolute} \end{cases} \tag{67}$$

Computing the final Jacobian (cont.)

Target

Compute z_i and o_i.

- z_i is equal to the first three elements of the 3rd column of matrix ${}^0T_i$
- o_i is equal to the first three elements of the 4th column of matrix ${}^0T_i$

${}^0T_i$ has to be computed for every joint.

Jacobian of a Manipulator – DOF

Consider a Manipulator with 6 DOFs:

$$T_6 = A_1 A_2 A_3 A_4 A_5 A_6$$

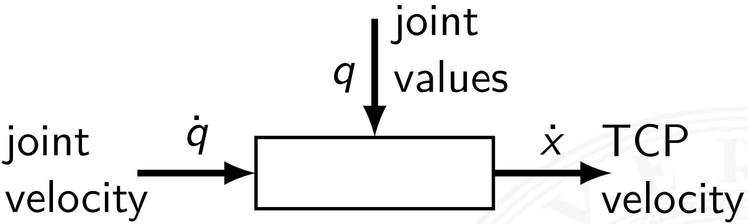
the Jacobian is:

$$\begin{bmatrix} {}^{T_6}d_x \\ {}^{T_6}d_y \\ {}^{T_6}d_z \\ {}^{T_6}\delta_x \\ {}^{T_6}\delta_y \\ {}^{T_6}\delta_z \end{bmatrix} = J_{6\times 6} \begin{bmatrix} dq_1 \\ dq_2 \\ dq_3 \\ dq_4 \\ dq_5 \\ dq_6 \end{bmatrix}$$

$$\dot{\mathbf{x}} = J(\mathbf{q}) \quad \dot{\mathbf{q}}$$

In case of a 6-DOF manipulator, we get a 6×6 matrix.

Inverse Jacobian



Question

Is the Jacobian invertible?

If it is, then: $\dot{\mathbf{q}} = J^{-1}(\mathbf{q})\dot{\mathbf{x}}$

$\Longrightarrow$ to move the the end-effector of the robot in Cartesian Space with a certain velocity.

Singular Configurations

For most manipulators there exist values of $\mathbf{q}$ where the Jacobian gets singular.

Singularity

$$\det J = 0 \Longrightarrow J \text{ is not invertible}$$

Such configurations are called singularities of the manipulator.
Two Main types of Singularities:

- Workspace boundary singularities
- Workspace internal singularities

Singular Configurations – Workarounds

- generally only for 6-DOF manipulators the Jacobian is invertible
- there are workarounds for other types of manipulators

$n < 6$ manually restrict the DOF of the end-effector
$\Longrightarrow$ square Jacobian matrix.
Example:

$$\begin{bmatrix} {}^{T_6}d_x \\ {}^{T_6}d_y \end{bmatrix} = J_{2\times 2} \begin{bmatrix} dq_1 \\ dq_2 \end{bmatrix}$$

for a 2-joint planar manipulator

$n > 6$ use the `pseudoinverse of J`

$$A^+ = (A^T \cdot A)^{-1} \cdot A^T, \text{ linear independent colums} \tag{68}$$

$$A^+ = A^T \cdot (A^T \cdot A)^{-1}, \text{ linear independent rows} \tag{69}$$

Bibliography

[1] K. Fu, R. González, and C. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*.
McGraw-Hill series in CAD/CAM robotics and computer vision, McGraw-Hill, 1987.

[2] R. Paul, *Robot Manipulators: Mathematics, Programming, and Control: the Computer Control of Robot Manipulators*.
Artificial Intelligence Series, MIT Press, 1981.

[3] J. Craig, *Introduction to Robotics: Pearson New International Edition: Mechanics and Control*.
Always learning, Pearson Education, Limited, 2013.

[4] J. F. Engelberger, *Robotics in service*.
MIT Press, 1989.

[5] W. Böhm, G. Farin, and J. Kahmann, "A Survey of Curve and Surface Methods in CAGD," *Comput. Aided Geom. Des.*, vol. 1, pp. 1–60, July 1984.

Bibliography (cont.)

[6] J. Zhang and A. Knoll, "Constructing Fuzzy Controllers with B-spline Models - Principles and Applications," *International Journal of Intelligent Systems*, vol. 13, no. 2-3, pp. 257–285, 1998.

[7] M. Eck and H. Hoppe, "Automatic Reconstruction of B-spline Surfaces of Arbitrary Topological Type," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, (New York, NY, USA), pp. 325–334, ACM, 1996.

[8] M. C. Ferch, *Lernen von Montagestrategien in einer verteilten Multiroboterumgebung*.
PhD thesis, Bielefeld University, 2001.

[9] J. H. Reif, "Complexity of the Mover's Problem and Generalizations - Extended Abstract," *Proceedings of the 20th Annual IEEE Conference on Foundations of Computer Science*, pp. 421–427, 1979.

Bibliography (cont.)

[10] J. T. Schwartz and M. Sharir, "A Survey of Motion Planning and Related Geometric Algorithms," *Artificial Intelligence*, vol. 37, no. 1, pp. 157–169, 1988.

[11] J. Canny, *The Complexity of Robot Motion Planning*. MIT press, 1988.

[12] T. Lozano-Pérez, J. L. Jones, P. A. O'Donnell, and E. Mazer, *Handey: A Robot Task Planner*. Cambridge, MA, USA: MIT Press, 1992.

[13] O. Khatib, "The Potential Field Approach and Operational Space Formulation in Robot Control," in *Adaptive and Learning Systems*, pp. 367–377, Springer, 1986.

[14] J. Barraquand, L. Kavraki, R. Motwani, J.-C. Latombe, T.-Y. Li, and P. Raghavan, "A Random Sampling Scheme for Path Planning," in *Robotics Research* (G. Giralt and G. Hirzinger, eds.), pp. 249–264, Springer London, 1996.

[15] R. Geraerts and M. H. Overmars, “A Comparative Study of Probabilistic Roadmap Planners,” in *Algorithmic Foundations of Robotics V*, pp. 43–57, Springer, 2004.

[16] K. Nishiwaki, J. Kuffner, S. Kagami, M. Inaba, and H. Inoue, “The Experimental Humanoid Robot H7: A Research Platform for Autonomous Behaviour,” *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 79–107, 2007.

[17] R. Brooks, “A robust layered control system for a mobile robot,” *Robotics and Automation, IEEE Journal of*, vol. 2, pp. 14–23, Mar 1986.

[18] M. J. Mataric, “Interaction and intelligent behavior.,” tech. rep., DTIC Document, 1994.

[19] M. P. Georgeff and A. L. Lansky, “Reactive reasoning and planning.,” in *AAAI*, vol. 87, pp. 677–682, 1987.

[20] J. Zhang and A. Knoll, *Integrating Deliberative and Reactive Strategies via Fuzzy Modular Control*, pp. 367–385. Heidelberg: Physica-Verlag HD, 2001.

[21] J. S. Albus, "The nist real-time control system (rcs): an approach to intelligent systems research," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 157–174, 1997.

[22] A. Meystel, "Nested hierarchical control," 1993.

[23] G. Saridis, "Machine-intelligent robots: A hierarchical control approach," in *Machine Intelligence and Knowledge Engineering for Robotic Applications* (A. Wong and A. Pugh, eds.), vol. 33 of *NATO ASI Series*, pp. 221–234, Springer Berlin Heidelberg, 1987.

[24] T. Fukuda and T. Shibata, "Hierarchical intelligent control for robotic motion by using fuzzy, artificial intelligence, and neural network," in *Neural Networks, 1992. IJCNN., International Joint Conference on*, vol. 1, pp. 269–274 vol.1, Jun 1992.

Bibliography (cont.)

[25] R. C. Arkin and T. Balch, “Aura: principles and practice in review,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 175–189, 1997.

[26] E. Gat, “Integrating reaction and planning in a heterogeneous asynchronous architecture for mobile robot navigation,” *ACM SIGART Bulletin*, vol. 2, no. 4, pp. 70–74, 1991.

[27] L. Einig, *Hierarchical Plan Generation and Selection for Shortest Plans based on Experienced Execution Duration.*
Master thesis, Universität Hamburg, 2015.

[28] J. Craig, *Introduction to Robotics: Mechanics & Control. Solutions Manual.*
Addison-Wesley Pub. Co., 1986.

[29] H. Siegert and S. Bocionek, *Robotik: Programmierung intelligenter Roboter: Programmierung intelligenter Roboter.*
Springer-Lehrbuch, Springer Berlin Heidelberg, 2013.

Bibliography (cont.)

[30] R. Schilling, *Fundamentals of robotics: analysis and control*. Prentice Hall, 1990.

[31] T. Yoshikawa, *Foundations of Robotics: Analysis and Control*. Cambridge, MA, USA: MIT Press, 1990.

[32] M. Spong, *Robot Dynamics And Control*. Wiley India Pvt. Limited, 2008.