Presenting haptic feedback from laser range sensor data to an operator of a humanoid robot

Bachelor thesis (*Afstudeerscriptie*)

Author **Patrick de Kok**
`pkok@science.uva.nl`

Supervisor **Arnoud Visser**
`arnoud@science.uva.nl`

Submitted in partial fulfillment for a Bachelor's degree in **Kunstmatige Intelligentie** at the **Universiteit van Amsterdam**

UNIVERSITEIT VAN AMSTERDAM

Abstract

In this thesis I present a method to keep a point on a robotic non-prismatic limb in a plane relative to the coordinate system of another non-prismatic limb, while both limbs can move. The position of the limbs is tracked solely by its shaft encoders. **Zijn dat shaft encoders?** As an example of practical use of this method, I have implemented a method of generating force feedback to an operator when controling a robot arm. It generates the force feedback when a laser range finder on the robot's head detects an obstacle in front of the robot hand.

Acknowledgement

Of course I have people to thank.

Universiteit van Amsterdam
Instituut voor Informatica
Science Park 904
1098 XG Amsterdam

UNIVERSITEIT VAN AMSTERDAM

Contents

Chapter 1

Introduction

Within the field of Artificial Intelligence (AI), the focus lies at designing and building rational agents, where an agent differs from most programs in at least the fact that they operate autonomously [RN03, p.4]. This science hasn't developed enough to design autonomous applications that can be used in all situations. Even within the AI community there is debate whether "AI-complete" problems [JM00, p.738], like the above problem class very well might be, would ever be solved. To make optimal use of the currently developed techniques, researchers are looking at the best way of putting a human on the spot of an autonomous agent. One of these fields of research is telepresence. Telepresence focuses on converting sensor data to corresponding human experiences and letting the human act upon this remote environment. Examples of telepresence systems include telephones and video conference systems. One could also think of remotely operated robots. This is called remote robot presence, and lies at the intersection of the fields of telepresence and human robot interaction (HRI).

There are many cases in which robots can be used. They are regularly sumarrised by "the 3 D's: jobs that are dirty, dull or dangerous" [Mur00, p.6]. A fourth class of tasks in which robotics is often applied, are those involving areas which are otherwise inaccessible. This last class is one of the classical areas in which telepresence is applied. But it is also interesting to look at designing telepresence systems for dirty and dangerous jobs, as it could increase the comfort of the human operator.

Throughout the years, the role of the human has moved from robot operator, having full control over the robot, to someone who monitors the status of the robot and gives high-level instructions [vEDJ+06]. It is shown that fully automating tasks does not always ensure better system performance [vE00], which is an argument for keeping the human in the operating position.

Telepresence lets the user experience a remote environment and act upon it. One way of reaching this, is raising the situation awareness of the human operator [End88, vEDJ+06]. Even though it has been demonstrated that the sense of touch, or haptic sense, is accurate and fast [KLM85], it is often not used in telepresence systems.

Most human senses consist of multimodal input. For example, vision is composed of properties describing color and light intensity. The haptic sense has been broken down into seven properties: texture, hardness, temperature, weight, volume, global shape and exact shape. Humans investigate each of these properties with stereotypical hand movements, called “exploratory procedures” [LK87]. In a remote robot presence system, one could implement all of these, but it might be preferable to shield off the user of some of these properties, like temperature. Other properties, like hardness, volume and shape, are very useful when represented with little to no modification to the human operator.

For this to be useful to the operator, it is important that the sensors that gather the haptic data should interfere as little as possible with the work of the operator. When the operator has to aim its sensor at his tools manually, the correct usage of the system will be tedious. One could choose for several designs to overcome this problem. In the first place, the haptic feedback can be left out, with the obvious drawback of not getting any additional information conveyed to the operator. As another option, the system could be designed with only short ranged sensors, such as touch sensors, which are mounted on or near the tool. But sometimes one needs to generate feedback before the inspected object is touched. Other solutions, like using gyroscopes or accelerometers, can be used only in very specific cases.

Laser range sensors offer a good heuristic solution. We assume that most objects that are detected by laser range sensors, generate some form of haptic feedback, and that most objects that we can touch, can be recognized by the sensor. Because of the size and fragility of current laser range scanners, it is uncommon that they are mounted on a robot’s tool. For this, there should be movement limits be imposed, so the tools, when used, should be in the plane that is scanned by the laser range scanner.

1.1 Research question

We have investigated how to characterize the movement restraints of end effectors, so that they move in a plane relative to another end effector. We have solved this problem for the Nao robot[Rob] using geometric algebra. Even though we have solved this for this specific platform, it should be a general solution, which should be applicable to all robots with a pure revolute kinematic chain between the end effector that should be moved, and the laser range scanner.

1.2 Document structure

The document is set up as follows. In Chapter 2 we build the theoretical knowledge to come to the solution. Section 2.1 gives an introduction to the terms used in robotics. In Section 2.2 we introduce the reader to forward and inverse kinematics, the framework which describes movement for a broad class of robots. In Chapter 2.3 we describe conformal geometric algebra, and its advantage over homogeneous models of Euclidean space. Chapter 3 describes the algorithm we have used and extended. In Chapter 4 we discuss the quality of our solution, and give a reflection on our approach and a discussion of future work.

Chapter 2

Background Theory

In this chapter, we will introduce a mathematical framework for robot movement. To do this, we will first introduce some common concepts in robotics. We assume that the reader has a basic understanding of linear algebra.

2.1 Robotics terminology

Definiëren wat een robot is? Met onderstaande ben ik het niet eens, denk ik

Robots are physical agents that performs tasks by manipulating the physical world [RN03, p.901]. A robot's body consists of *links* which are connected with each other through *joints*. For the discussion in this document we assume all links to be *rigid bodies*; physical objects that cannot be deformed. A link's position can change with respect to another link through change in its joint. It can either revolve around a point, or it can be moved some distance. Joints that enable the first type of change are called *revolute*, the second class is called a *prismatic joint*. We count 1 *degree of freedom* (DOF) for each transformation a joint can exercise on a link. Joints with a DOF greater than 1 are said to be composite, and can be represented by multiple joints that are connected to each other by links with no length.

Links are thus connected to other links by joints. When considering links and joints that connect to each other as a whole, we speak of a *kinematic chain*. Kinematic chains can be *closed*; it can contain a loop, where a link is connected to another link that is already in the chain. When there are no loops, the kinematic chain is *open*.

Because of its links and joints, the robot can affect the world. The parts of its body that do so, are called *effectors*. The parts at the very end of an (open) kinematic chain are called *end effectors*.

Links hebben een coördinatenstelsel, niet joints toch?

It often is more convenient to know where an object is relative to a link, than knowing where the link and the object are in the world coordinate system. Because of this we define a *frame*, or basis, $\mathfrak{B}_i$ for each link l_i. The basis is usually orthogonal. The origin of the basis is defined relative to its corresponding link. We will use $[\mathbf{p}]_{\mathfrak{B}}$ to denote a vector that represents a point p in the basis $\mathfrak{B}$. As a shorthand for $[\mathbf{p}]_{\mathfrak{W}}$, a vector representing a point p in the basis of the world, we write $\mathbf{p}$.

As rigid bodies cannot be deformed, they can only undergo operations that are *isometric*. A matrix A is said to be isometric when the length of a vector is preserved after applying A to it. Each isometry in a Euclidean space can be decomposed by a rotation (or a special orthogonal transformation), a translation and a reflection. As a physical object cannot reflect its body in the real world, we will not further discuss those isometries that reflect points in a certain plane.

TODO: Decompositiestelling kort behandelen

TODO: Stelling "2 spiegelingen maken 1 rotatie" bespreken

TODO: Denavit-Hartenberg bespreken!

Verhaal verduidelijken aan de hand van een plaatje?

2.2 Forward and inverse kinematics

Forward and inverse kinematics form a framework for working with robot joint configurations and their positions. In the following discussion we have assumed all kinematic chains to be open.

Forward kinematics is used to compute the position and orientation of a link with respect to some basis $\mathfrak{B}$, given the configuration of the joints in the kinematic chain.

Each link l_i in the kinematic chain has its own coordinate system $\mathfrak{B}_i$. One can express a point $\mathbf{p} = [p_x, p_y, p_z]^T$ in one coordinate system $\mathfrak{B}_i$ in a different coordinate system $\mathfrak{B}_j$ when the rotation θ_i and translation $\mathbf{d}_i = [t_x, t_y, t_z]^T$ is given. As known, a rotation over an angle θ_i on $\mathbf{p}$ is a linear operation and can be represented by the matrix:

2D $R(\theta)$ kan niet op 3D $\mathbf{p}$ toegepast worden!

$$R(\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}$$

This is just a simple rotation in 2 dimensional space. We need a rotation in 3 dimensional space. For this, one can use either R_x, R_y or R_z for a rotation over one of three axes: **Waarom zou ik eigenlijk een 2D-rotatie behandelen...**

$$
\begin{aligned}
R_x(\theta_i) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_i & -\sin\theta_i \\ 0 & \sin\theta_i & \cos\theta_i \end{bmatrix} \\
R_y(\theta_i) &= \begin{bmatrix} \cos\theta_i & 0 & \sin\theta_i \\ 0 & 1 & 0 \\ -\sin\theta_i & 0 & \cos\theta_i \end{bmatrix} \\
R_z(\theta_i) &= \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

As a comprehensive way of denoting a rotation over all three directions $\theta_{i,x}$, $\theta_{i,y}$ and $\theta_{i,z}$, we will write $R(\theta_\mathbf{i}) = R_z(\theta_{i,z})R_y(\theta_{i,y})R_x(\theta_{i,x})$.

Translation over $\mathbf{d}_i$ is not a linear operation on $\mathbf{p}$. Translation over $\mathbf{d}_i$ is, however, a linear operation on $\mathbf{p}' = [p_x, p_y, p_z, 1]$. This can be represented by the matrix:

$$
D(\mathbf{d}_i) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

such that

$$
D(\mathbf{d}_i)\mathbf{p} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} p_x + t_x \\ p_y + t_y \\ p_z + t_z \\ 1 \end{bmatrix}
$$

This extension on our normal coordinate system is called a *homogeneous coordinate system*. When extending the rotation matrix R to a homogeneous coordinate system, you get:

$$
R^H(\theta_i) = \begin{bmatrix} R(\theta_i) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}
$$

A more general form of this matrix, in which both operations are combined can be given by:

$$
{}^iT_j = R^H(\theta_i)D(\mathbf{d}_i) = D(\mathbf{d}_i)R^H(\theta_i) = \begin{bmatrix} R(\theta_i) & \mathbf{d}_i \\ \mathbf{0}^T & 1 \end{bmatrix}
$$

Here ${}^iT_j$ denotes the homogeneous transformation from coordinate system $\mathfrak{B}_i$ to $\mathfrak{B}_j$.

Alles in deze sectie hoort eigenlijk ook nog in de vorige...

2.3 Geometric algebra

Chapter 3

Results

Chapter 4

Conclusion

4.1 Conclusion

4.2 Discussion

4.3 Future work

Bibliography

[End88] M. R. Endsley. Design and evaluation for situation awereness enhancement. In *Proc Human Factors Society 32nd Annual Meeting*, Santa Monica, CA, 1988.

[JM00] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Prentice Hall series in Artificial Intelligence. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[KLM85] Roberta L. Klatzky, Susan J. Lederman, and V. Metzger. Identifying objects by touch: An 'expert system'. *Perception & Psychophysics*, 37:299–302, 1985.

[LK87] Susan J. Lederman and Roberta L. Klatzky. Hand movements: A window into haptic object recognition. *Cognitive Psychology*, 19:342 – 368, 1987.

[Mur00] Robin R. Murphy. *Introduction to AI robotics*. Intelligent Robots and Autonomous Agents. The MIT Press, 2000.

[RN03] Stuart J. Russel and Peter Norvig. *Artificial Intelligence - a modern approach*. Prentice Hall series in Artificial Intelligence. Prentice Hall, Upper Saddle River, New Jersey, second edition, 2003.

[Rob] Aldebaran Robotics. Nao academics edition datasheet. http://www.aldebaran-robotics.com/Files/DSV3_En.pdf. Visited at April 6, 2010.

[vE00] Jan B. F. van Erp. Controlling unmanned vehicles: The human solution. In *RTO Meeting Proceedings*, 4, pages B81 – 8/12, 2000.

[vEDJ+06] Jan B.F. van Erp, Maaike Duistermaat, Chris Jansen, Eric Groen, and Marieka Hoedemaeker. Tele-presence: Bringing the operator back in the loop. In *Virtual Media for Military Applications*, pages 9–1 – 9–18. Meeting Proceedings RTO-MP-HFM, 2006.