WRITTEN BY
PROF. KISUNG SEO
EDITED BY
ALDEBARAN ROBOTICS
& NT RESEARCH, INC

USING NAO: INTRODUCTION TO INTERACTIVE HUMANOID ROBOTS

ALDEBARAN Robotics

WORDS FROM THE AUTHOR

Robots are having a profound impact in various fields including manufacturing, extreme jobs, and service sectors. Robots will have a wider range of application in the near future. Humanoid robots are attracting the most attention compared to other robots because 1) they look similar to people so they seem friendlier and are recognized as being a better fit for helping (or replacing) humans for certain tasks, 2) much like humans, biped walking is possible and jobs can be performed using both hands, and 3) they mimic the most evolutionally outstanding human form and function.

Furthermore, humanoid robots are getting a lot of attention from educators and researchers because they are surrounded by challenging issues including difficulties in walking and general motion control, effectiveness issue with processing the recognition sensors, and implementation of intelligence. It has been very tough because there aren't a lot of commercialized robots we can use to develop new controls and intelligence algorithms and for fully utilizing the advanced features of these humanoid robots in actual educational and research sites. Fortunately, more humanoid robots are being released, and out of all of them, NAO from Aldebaran Robotics is the world's most widely known humanoid robot being used for education and research. In August of 2007, it was designated as the official platform for RoboCup (Robot Soccer World Cup) instead of Sony's Aibo (puppy robot), and has been adopted in Suzhou, China starting from the 2008 competition.

Humanoid NAO consists of 25 joints that make walking and general motion control possible. Diverse interactions are possible through wireless/cable network enabled communication, cameras, infrared sensors, microphone, speakers and LEDs. The software structure is based on open source embedded Linux and supports programming languages like C, C++, URBI, Python, and. Net Framework. It also provides a graphic-based programming called Choregraphe.

This book will try to focus on using Aldebaran's humanoid NAO robot to explain the environment and tools, programming techniques, and basic theory and applications for educational and research purposes of vocational high schools, universities, and the general public.

This book is largely divided into two parts: **Chapters 1-3** for beginners and **Chapters 4-6** for advanced users. **Chapters 1-3** introduce Choregraphe and Python necessary for basic NAO robot usage. **Chapters 4-6** handle information for professional use. I would like to advise anyone just learning about the NAO robot and people who are unfamiliar with C and Python to become familiar with the information in **Chapters 1-3**. **Chapters 2** and **4-6** are recommended for anyone with previous experience in robot programming or anyone who wants to perform specialized algorithms and control commands.

Chapter 1 introduces the NAO robot and the Monitor program that can be used to verify NAO's internal memory and image processing. It will also explain how to do the initial setup for the system. Because this chapter discusses NAO's special features, it would be good for readers who are not quite familiar with NAO.

Chapter 2 will teach you how to use Choregraphe, a graphic-based programming tool, to operate NAO. Choregraphe uses a program module called Diagram to explain how to program and how to set NAO's movements in Timeline. Additionally, it will provide a description of how to use box libraries and FTP in Choregraphe.

Chapter 3 will have a short introduction to Choregraphe scripts and Python for NAOqi. There is a basic description of Python syntax and a discussion about creating and editing Choregraphe script boxes. This would be a good chapter if you are already familiar with Python.

Chapter 4 explains the NAOqi framework which forms the foundation of the NAO robot and the DCM used for controlling all the devices. Special characteristics including the NAOqi framework structure, file structure, and Broker as well as the NAOqi framework are used to control NAO. It also explores how to load modules into NAO using Linux, C++, and cross-compiling as well as what to do when several commands are received in Time Command. There will also be an introduction to the structures of DCM controlled devices and how to synchronize using DCM's synchronization method.

Robot kinematics in **Chapter 5** explains NAO's joint structure and provides information for each joint. The Denavit-Hartenberg (DH) method is used to explain the calculation for forward kinematics. In addition, Python will be used to create an actual forward kinematics calculation program. This chapter will also describe inverse kinematics calculations and use Python to implement the inverse kinematics calculation program for NAO's right arm. You will need quite a bit of mathematical and robotics knowledge to understand the contents in **Chapter 5.**

Comprehensive Exercises in **Chapter 6** use the information thus far to look at different methods and examples for implementing NAO's applications. Advanced Choregraphe features and expansion methods will be used here and you will be able to practice using Timeline Editor. In addition, landmark recognition will be used to create a path finding program, and the multiplication example will help you learn some of the techniques for Python and NAOqi API. Last, but not least, image recognition will be used to classify objects and inverse kinematics and NAOqi usage will be explained.

It was considerably difficult to write this book because there was a disadvantage of dealing with such a specific model of humanoid robot.

There wasn't much material about it, and the ones that were available were quite disorganized. I was also conflicted about how to handle the variety of readership because of the content and general difficulty of the subject matter. I am sincerely hoping that this book will serve as a good introduction to humanoid robots.

I would like to express my sincere gratitude to the people at NT Research Inc. who gave me both material and emotional support.

I would especially like to thank Jae-young Jang, Byung-yong Hyun, Su-hwan Hyun, Oh-sung Kwon, Jae-min Lee, and Young-kyun Kim in Intelligent Systems Laboratory for conducting the series of experiments with the NAO robot to help me verify the information in this book. Although every effort has been put into gathering information for this book, I am sure that there is still room for improvement, and I acknowledge that this is wholly due to the fact that I still have a lot to learn about this vast and amazing field.

Ki-sung Suh

HOW TO USE THIS CURRICULUM

YOU ARE ALLOWED TO REPRODUCE THE CONTENT OF THIS BOOK AND TO SHARE IT WITH YOUR CLASSROOM ONLY.

This curriculum has been done with the 1.8.16 version of Choregraphe, our programming software. However, most of the features are compatible with newer versions. The screenshot of the software included in this curriculum may be different depending of the version of Choregraphe you have.

TABLE OF CONTENTS

1 INTRODUCTION

LEARNING

Chapter 1 introduces the NAO robot and the Monitor program that can be used to verify NAO's internal memory It will also explain how to do the initial setup for the system.

Because this chapter discusses NAO's special features, **it would be good for readers who are not yet familiar with NAO.**

CONTENT

2 CHOREGRAPHE

LEARNING

Chapter 2 will teach you how to use Choregraphe, a graphic-based programming tool, to operate NAO. Choregraphe uses a program module called Diagram to explain how to program and how to set NAO's movements in Timeline.

Additionally, it will provide a description of how to use box libraries and FTP in Choregraphe.

CONTENT

3 PYTHON

LEARNING

Chapter 3 will have a short introduction to Choregraphe scripts and Python for NAOqi. There is a basic description of Python syntax and a discussion about creating and editing Choregraphe script boxes.

This would be a good chapter if you are already familiar with Python

CONTENT

4 NAOqi & DCM

LEARNING

Chapter 4 explains the NAOqi framework which forms the foundation of the NAO robot and the DCM used for controlling all the devices. Special characteristics including the NAOqi framework structure, file structure, and Broker as well as the NAOqi framework are used to control NAO.

It also explores how to load modules into NAO using Linux, C++, and cross-compiling as well as what to do when several commands are received in Time Command. There will also be an introduction to the structures of DCM controlled devices and how to synchronize using DCM's synchronization method.

CONTENT

5 NAO KINEMATICS

LEARNING

Robot kinematics in Chapter 5 explains NAO's joint structure and provides information for each joint. The Denavit-Hartenberg (DH) method is used to explain the calculation for forward kinematics. In addition, Python will be used to create an actual forward kinematics calculation program.

This chapter will also describe inverse kinematics calculations and use Python to implement the inverse kinematics calculation program for NAO's right arm.

PREREQUISITE

You will need quite a bit of mathematical and robotics knowledge to understand the contents in **Chapter 5**

CONTENT

6 COMPREHENSIVE EXAMPLES

LEARNING

Comprehensive Exercises in **Chapter 6** use the information thus far to look at different methods and examples for implementing NAO's applications. Advanced Choregraphe features and expansion methods will be used here and you will be able to practice using Timeline Editor.

In addition, landmark recognition will be used to create a path-finding program, and the multiplication example will help you learn some of the techniques for Python and NAOqi API. Last, but not least, image recognition will be used to classify objects and inverse kinematics and NAOqi usage will be explained.

CONTENT

ABOUT THE AUTHOR

KI-SUNG SUH

1986	Yonsei University School of Electrical Engineering, BS.
1988	Yonsei University School of Electrical Engineering, Master of Engineering.
1993	Yonsei University School of Electrical Engineering, Ph.D.
1993-1998	Seokeyeong University, Department of Industrial Engineering, Department of Electronic Engineering, Assistant Professor
1999-2003	Michigan State University, Genetic Algorithms Research and Applications Group, Research Associate.
2002-2003	Michigan State University, Electrical & Computer Engineering, Visiting Assistant Professor.
2004-Present	Seokeyeong University, Department of Electronic Engineering, Associate Professor

Areas of interest include Intelligent Robot, Evolutionary Computation, Genetic Programming, Evolutionary Neural Networks, and Evolutionary Design.