19AIE201 "Introduction to Robotics"

Project Report

NAO Humanoid Robot

Bachelor of Technology
in
Artificial Intelligence and Engineering

SUBMITTED BY GROUP NO : 2A

Roll No.	Names Of Students
AM.EN.U4AIE20051	NADEEM N
AM.EN.U4AIE20053	NEHITH SAI VEMULAPALLI
AM.EN.U4AIE20054	PALADUGULA PRUTHVI
AM.EN.U4AIE20069	SRUNGARAPU JATIN ABIT SAI
AM.EN.U4AIE20077	Y DHANVANTH REDDY

Subject Teacher
Pritam Bhattacharjee

School of Computer Science and Engineering
Kollam, Kerala, India – 690 525.

3rd Semester 2021

ABSTRACT

Humanoid robots are not only a man alike machines that resembles human's physical appearance or attributes; like a head, a chest, arms, and legs but they also have the senses and capability to communicate and interact with humans or robots through interpreted information to perform a specified task or activity according to the given input. These robots are characteristically pre-programmed for resolute specific actions.

Humanoid robots are robots that resemble the shape and characteristics of humans such as the abilities of humans like walking , recognizing things , talking and others. However, the real intention of a humanoid robot was to build a better and proficient orthosis and prosthesis system for human beings. The examples of it can be seen as neuro-muscularly impaired mechanical leg prosthesis, biological ankle feet orthosis, and forearm prosthesis.

As the humanoid tech got matured over time, characteristics of the humanoid robots got improved now. With the emergence of Artificial Intelligence in robotics, human alike robots now have the capability of autonomous learning and safe interaction with the surrounding environment of humans and other smart machines. Humanoid robots are now employed in various platforms.

One of the advanced humanoid robot which we are going to talk about is the NAO robot manufactured by SoftBank Robotics (previously Aldebaran Robotics)

Contents

List of Figures

Chapter 1

Introduction

The main goal of the NAO robot has been devised with the concern of cost reduction without sacrificing quality and performance. It is completely a custom designed robot as the whole process of manufacturing and designing is mastered (mechanics, electronics, software). The costs which are allowed to be reduced at every stage of design. The company employs subcontractors to produce plastic parts or electronic circuits on a large scale. One way to achieve cost reduction was the reuse of the same actuator modules for several joints. Another way consisted of reducing the number of motors without sacrificing mobility. The robot will cost about 10K euros for laboratories. Due to the mass production and reduction of functionalities a version will be publicly available for approximately 4K euros.

Figure 1.1: NAO HUMANOID ROBOT

A biped robot must show good motion performance for its height to weight ratio or body mass index (BMI). NAO has a BMI of about 13.5[kg/m2], which means that it is very light compared to other existing robots of the same height. Table I gives the BMI for different functional humanoid robots. The quest for performance of NAO robot is different from the objectives followed by teams of robotics laboratories who focus on a particular research subject. One of the subject concerns the development of walking gaits on biped or humanoid prototypes, like the ETL-humanoid , the BIP biped , the 2D Rabbit biped , or the LOLA humanoid and Johnnie robots . Other studies aim at designing robots with artificial limbs for disabled people, like the Robian biped , or at exploring natural dynamics of flexible actuators, like the series of robots designed at MIT .

First, the performance targets for NAO are smooth walking gaits even when changing speed , and a rich panel of movements that the robot must execute with smoothness and precision and direction. The speed and walking must be similar to the walking speed of children of the same size as of children, that is about 0.6[km/h]. These objectives of performance involve building strong and reliable hardware together with precise joint control. The robot was built with high precision magnetic sensor devices and high quality brush DC motors . The gear reduction for each motor mechanism was optimized carefully . The robot in further should be equipped with cognition and artificial intelligence capabilities, but above all, interactive autonomy. It tells us that the robot is able to recognize features of human and human faces in the environment, to self localize and to operate in this environment.

One of the big challenges for a legged robot consists of self-localizing and recognizing moving objects while walking. Research teams have already built humanoid robots dedicated to studying autonomous interaction with humans and the environment, such as the Japanese JSK-H7 humanoid , or the recent REEMA and REEMB humanoid robots . For NAO to interact with its environment, the head was equipped with interactive sensors (see table II). The cognitive capabilities of the robot will depend on the quality of the embedded software. Modularity is the 3rd guideline followed by the French designers of NAO. 1stly, modularity refers to actuator modules that could be used for different joints. This was also the concern of designers of the LOLA bipeds or the QRIO humanoid .

TABLE I
CHARACTERISTICS OF FUNCTIONAL HUMANOIDS.

	Height (h) (m)	Weight (w) (kg)	BMI (kg/m^2)	Price
KHR-2HV	0.34	1.3	10.9	1K US $
HOAP	0.50	7.0	28.0	50K US $
NAO	**0.57**	**4.5**	**13.5**	**10K euros**
QRIO	0.58	6.5	19.0	NA
ASIMO	1.30	54.0	32.0	NA
REEM-A	1.40	40.0	20.4	NA
HRP-2	1.54	58.0	24.5	400K US $ (5 year lease)
Human	1.5-2	50-100	18-25	NA

BMI: body mass index = w/h^2, NA: not available

TABLE II
SENSORS THAT EQUIP NAO.

Type	nb
30 FPS CMOS videocamera	1
Gyrometer	2
Accelerometer	3
Magnetic rotary encoder (MRE)	34
FSR	8
Infrared sensor (emitter/receiver)	2
Ultrasonic sensor	2
Loudspeaker	2
Microphone	4

For NAO robot , there are 4 kinds of actuators based on 2 types of brush DC motors and 2 types of gear reduction for each motor. 2ndly, the modular design of the robot's limbs is also very useful to promote further evolution. Modular design was used by SONY for the ERS-2xx versions of AIBO quadrupeds used in RoboCup . In case of trouble , legs and head could be changed quickly. The legs of humans could also be replaced by wheels or other limbs. Coming to the case of NAO the head can be easily unplugged and replaced by a specialized one. Hands and forearms can also be changed. 3rdly the maintenance problem is not negligible, especially in the case of human-size robots such as HRP-2 , or small-size sophisticated humanoids such as SDR-4x/QRIO .

TABLE III
MAIN CHARACTERISTICS OF THE NAO HUMANOID.

Body	
Height (m)	0.57
Weight (kg)	4.5
Battery	
Type	Lithium-ion
Capacity	55 Wh
Degrees of freedom (DOF): 25	
Head	2 DOF
Arms	5 DOF X 2
Pelvis	1 DOF
Leg	5 DOF X 2
Hands	1 DOF X 2

Masses [g]	
Chest	1217.1
Head	401
Upper Arm	163
Lower Arm	87
Thigh	533
Tibia	423
Foot	158
Total	4346.1

TABLE IV
JOINTS TYPE, RANGE, AND ACTUATOR TYPE.

Part	Motion	Range (°)	Actuator type
Leg (left)	hip twist (45°)	-68 to 44	M1R11
	hip roll	-25 to 45	M1R11
	hip pitch	-100 to 25	M1R12
	knee pitch	0 to 130	M1R12
	ankle pitch	-75 to 45	M1R12
	ankle roll	-45 to 45	M1R11
Arm (left)	shoulder roll	0 to 95	M2R22
	shoulder pitch	-120 to 120	M2R21
	elbow roll	-120 to 120	M2R22
	elbow yaw	0 to 90	M2R21
Head	yaw	-90 to 120	M2R21
	pitch	-37 to 31	M2R22

Since NAO would be for sale on a large scale, its maintenance must be optimized so that spare parts can be changed quickly and easily. The modular design should help maintainability and should increase the robot's reliability. The final guideline refers to the open architecture. Open architecture means that the robot should be easy to get started with and to handle. This involves ergonomic software for a maximum of people to access and understand the programming functions, even for people who are not experts at programming. Few of the existing humanoid robots suffer from the lack of ergonomy .

The software proposed with NAO is user-friendly and relies on a distributed architecture that allows interfacing with embedded and remote applications that are useful for debug and development. The majority of the embedded software including the operating system can be changed by the user if desired which is also meant as Open architecture. The access to the Low level hardware is also open to allow users to change joint control laws. Some companies that have built humanoid robots available for purchase do not propose those features, and possibilities of changing the embedded software and limit the access . This paper focuses on the design of the NAO humanoid biped, that was designed to be affordable and performant. The guidelines like cost, performance, modularity, and open architecture were followed by the French designers of the Aldebaran Robotics company. The outer body shell was specially designed for the robot to make it look friendly. The first section of this paper is dedicated to the mechanical design and describes kinematics and actuation system. The 2nd section deals with the computer architecture and electronics. The third section is devoted to the software architecture specially developed for NAO.

Chapter 2

Background Study

NAO H25 (v4.0) is a 58cm, 5kg humanoid robot (Fig. 1) manufactured by Aldebaran Robotics in Paris, France. The NAO robot carries a fully capable computer on-board with an x86 ATOM Z530 processor at 1.6 GHz, 1.0 GB of RAM, 2.0 GB of flash disk, and up to 8 GB of user flash memory running an Embedded Linux distribution. It is powered by a 6-cell Lithium-Ion battery which provides about 60 minutes of continuous operation and communicates with remote computers via an IEEE 802.11g wireless or a wired Ethernet link. NAO H25 has 25 degrees of freedom; 2 in the head, 6 in each arm, 5 in each leg and 1 in the pelvis (there are two pelvis joints which are coupled on one servo and cannot move independently). All joints are position-controlled, using closed-loop PID controllers and encoders. It also features a variety of sensors: an Inertial Measurement Unit (IMU) in the torso, Force Sensitive Resistors (FSR) on each foot, ultrasonic range sensors on the chest, and two 960p@30fps cameras on the head.

The 25 joints of NAO are organized into five kinematic chains as follows:

- Head: HeadYaw, HeadPitch
- Left Arm: LShoulderPitch, LShoulderRoll, LElbowYaw, LElbowRoll, LWristYaw, LHand
- Right Arm: RShoulderPitch, RShoulderRoll, RElbowYaw, RElbowRoll, RWristYaw, RHand
- Left Leg: LHipYawPitch, LHipRoll, LHipPitch, LKneePitch, LAnklePitch, LAnkleRoll
- Right Leg: RHipYawPitch, RHipRoll, RHipPitch, RKneePitch, RAnklePitch, RAnkleRoll

LeftHipYawPitch and RightHipYawPitch are just different names for the shared (common) joint (HipYawPitch) between the two legs. Clearly, the effective operation of the robot in the RoboCup soccer field requires precise control of all five manipulators. In particular, the head must be precisely controlled for effective camera operation, while the legs and arms must be controlled to generate walk motions and to perform manipulation of the environment (kicking, dribbling, etc.). Likewise, in any other robot application with the NAO, any kind of motion design for effective behavior will require precise control of one or more manipulators of the robot.

Chapter 3

Mechanical Architecture

3.1 Kinematics

NAO has a total of 25 degrees of freedom, 11 degrees of freedom (DOF) for the lower part that includes legs and pelvis, and 14 DOF for the upper part that includes trunk, arms and head.

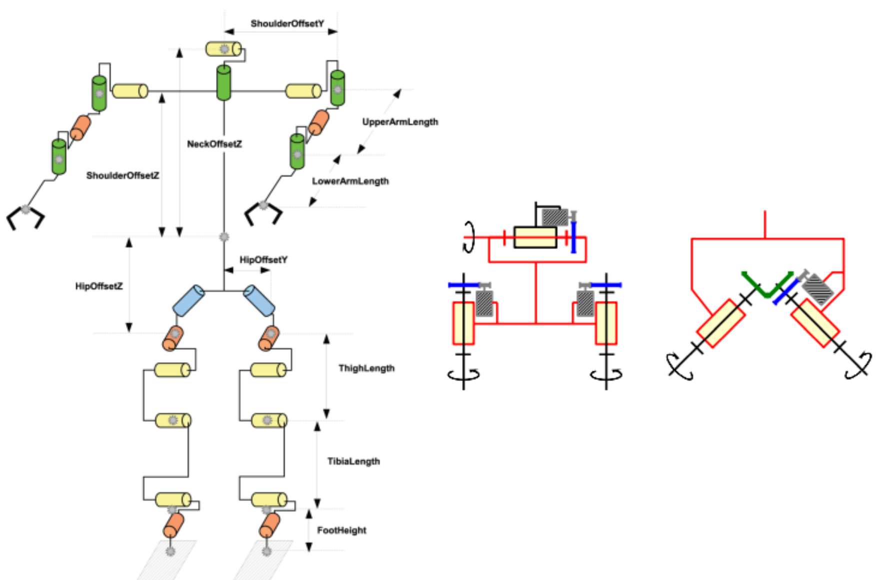

Figure 3.1: 1) Detailed Kinematics Of NAO Robot 2)Left-hand side: classical set of three rotary joints, one horizontal axis rotary joint at the waist and two vertical axis rotary joints for the legs. Right-hand side: coupled inclined axis rotary joints for the NAO pelvis.

Each leg has 2 Degrees of Freedom(DOF), at the ankle, 1 DOF(degree of freedom) at knee, and 2 DOF at the hip. A special mechanism that is composed of two coupled joints at each hip equips the pelvis. The rotation axis of these two joints is inclined at

$45\circ$ towards the body. This mechanism replaces the classical set of three active rotary joints encountered in most of the humanoid robots. This classical set includes the horizontal axis rotary joint at the waist and the rotary joints of the vertical axis for each leg hip. Only one motor is required to drive the pelvis mechanism of NAO. This allows saving one motor at a hip level without reducing the overall mobility. The mechanism permits NAO to bend the body forward while spreading the legs simultaneously. It is therefore useful for sitting down and for bending to grasp or lift something on the ground. This mechanical design was registered for patenting. Unlike prototypes such as H7, Wabian-2R or LOLA, the foot sole of the present version of NAO doesn't feature any passive or active joint that would enhance higher speed gait performances.

In addition, each arm features 2 DOF(Degrees of Freedom) at the shoulder, 2 DOF at the elbow, 1 DOF at the wrist, and 1 additional DOF for the hand's grasping. The head of the robot can rotate about yaw and pitch axes.

3.1.1 Dimensionning of leg actuators

Some humanoid designers advanced an iterative procedure of mechanical layout and dynamic simulations to get parameters of joint torques and velocities that have been used for motor and tools choice. The layout procedure of NAO isn't always iterative and is based on a dimensioning methodology. The robot's version is simplified and simulated dynamically in the sagittal aircraft and the frontal aircraft. The software program used for simulation became Working-Model related with MATLAB-Simulink for the manage part. A set of primary moves in the sagittal aircraft for the only part, and with inside the frontal aircraft for the alternative part, have been defined and simulated. Movements in sagittal aircraft helped measurement the knee actuators and the pitch joint actuators of hips and ankles. Movements in the frontal aircraft helped in completing the dimensioning of hip and ankle roll joint actuators. The period of each movement became set for the robot to acquire energetic motion. Motor and tools choice must ensure that the robot can be able to attaining those moves. The robot's model in the sagittal aircraft consists of a square trunk and an unmarried leg with rectangular femur, tibia and foot. The version with inside the frontal aircraft functions each leg. Hip, knee and ankle joints are rotary joints. In the frontal aircraft knee joints are blocked.. The set of basic movements in the sagittal plane are the following:

1. knee flexion on the spot, 1[sec],
2. standing up from flexed knee position, 1[sec],
3. leg transfer during walking step, body motionless,1/3[sec],
4. body translation in the direction of motion during leg stance, 1/3[sec],
5. body sinusoidal movement in the direction of motion during leg stance, 1/3[sec],
6. simultaneous knee flexion and body forward bending for pick up, 1.2[sec].

The set of basic movements in the frontal plane are the following:

1. lopsided move on both legs, 0.5[sec] ,
2. leg lift off in lopsided position , 0.5[sec],

All motors are controlled using a PID law. Velocity and output torque are recorded

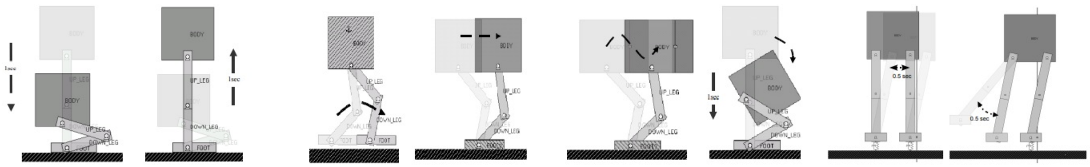

Figure 3.2: 1)Sagittal plane simulation experiments of knee flexion (1) and standup (2). 2) Sagittal plane simulation experiments of one step leg transfer (3) and body translation (4) 3) Sagittal plane simulation experiments of body sine move (5) and pick up (6). 4) Frontal plane simulation experiments of sideways move (7) and leg lift-off in lopsided position (8).

for all the movements. Actuator power is also recorded. For each joint the data relative to all the simulated movements are grouped to get the velocity variations as a function of torque over all the movements. Then the convex envelop is calculated. Figure 8 presents the curves of speed vs torque relative to the knee joint for experiments 1 to 6 in the sagittal plane. The convex envelope is also represented.

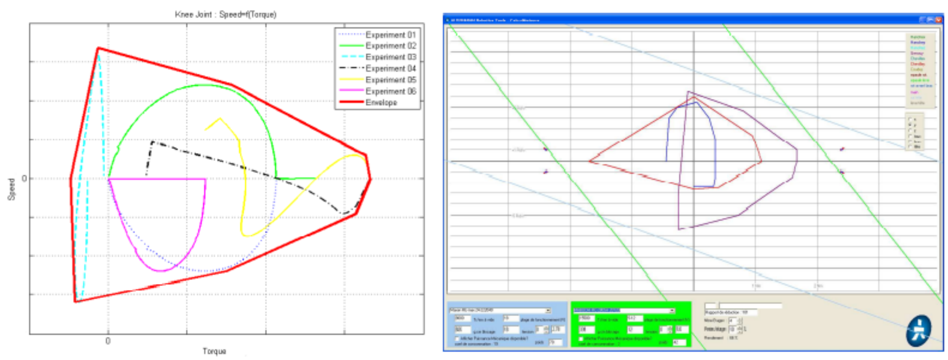

Figure 3.3: 1)Speed versus torque of knee actuator relative to experiments 1 to 6 in the sagittal plane. 2)Software used to choose motor and reduction ratio for the hip, knee and ankle pitch actuators.

The rpm versus torque specifications of off-the-shelf motors were compared with the desired variations to select the best suited motors. For this purpose a special graphical interface was developed to display the convex envelopes of the speed versus torque requirements drawn from the simulated experiments, and the speed versus torque specifications of existing motors. It is possible to vary the power voltage, the reduction ration, the number of reduction stages, and to see the influence of these changes on the different parameters such as yield and on the speed vs torque curve. Figure 9 shows the speed vs torque specifications of two motors, one from Maxon and the other from Mabuchi. The convex envelopes are relative to the knee, hip, and ankle pitch joint data drawn from the family of experiments simulated in the sagittal plane. This study is very helpful to select the best motor that can be used for several joints. After dimensioning it is interesting to check the load to motor inertia ratio k.

$$k = J_{\ L}/N^2.1/J_{\ M}$$

where N is the reduction ration, JM is the motor inertia, and JL is the load inertia. A too high value of load to motor inertia can cause instabilities along with oscillations due to resonance. The more compliant the system is more subject to oscillation it will be. A low value of the load to motor inertia leads to easier control and better dynamic response(fast acceleration/deceleration), but reduces the bandwidth of the system. Therefore the load to motor inertia results from a trade-off. Whatever the value of k, the most important parameter for actuators is the stiffness of the gear mechanism. If the actuators are stiff enough, high values for the load to motor ratio are acceptable, as the control module can handle close loop corrections. Taking into consideration that the drive mechanism is made of flat plane gears, the load to motor inertia ratio should be comprised between 1 to 10. Since the load varies when the robot walks, the ratio will vary inside this range. We have to check the load to motor inertia ratio at the knee joint in the case of simple support and in the case of double support. The mass above the knee is the mass of the robot's upper part and the masses of the thighs, that is approximately M = 3.2[kg]. For the robot's lower part, the actuators are based on motor M1 whose inertia is Jm = 4.17/107[kg.m2], and reduction ratio N is 130.85. The distance d of the knee joint to the center of mass is around 0.15[m] in the standard upright position. In the case of double support, we consider that each knee supports half of the mass M,

$$k_{\ d} = M/2.d^2/N^2.1/J_{\ M} = 5$$

In the simple support phase, the load inertia must take into account the mass of the robot's upper part, the thigh of the supporting leg and the mass of the leg in the air. We assume that the position of COG does not vary.

$$k_{\ s} = (M - m_{\ tibia} - m_{\ foot}).d^2/N^2.1/J_{\ M} = 8.3$$

The values of load to motor ratio are within the acceptable range.

3.1.2 Modular actuator unit design

The use of off-the-shelf RC servomotor modules was discarded from the beginning because they limit the performance of biped robots for a number of reasons: packaging not suited and generally bulky, gear reduction mechanism and joint control fixed. Considering that actuators represent the major cost it is necessary to conduct a careful study of how to choose and assemble motor, driving mechanism and sensor in the same module. The actuator must be compact, lightweight and highly back-drivable, efficient and precise and also reliable. Backdrivability was studied by, it defines the facility of movement transmission from output to input axis. The performance criteria of the actuator are power overweight ratio, temporary high torque generation, bandwidth, and response time. Among the specifications for NAO, the response time of the actuator tau must be maximized by:

$$\gamma <= \delta\theta_{\ max}/\omega$$

where omega stands for the maximal output angular velocity. The value of omega = 6[rad/s] is set according to the desired times needed to achieve planned motion trajectories in the Cartesian space. thetamax is the maximal angular delay that can be authorized as a consequence to a slope command assigned at the ankle joint when the robot is moving while having one leg in support. Taking into account a maximal deviation of the COG ground projection from the foot center, the maximal angular delay is set to:

$$\delta\theta_{\ max} = arcsin(\eta.L_{\ f}/L_{\ G})$$

where LG is the distance between ankle and COG, it's the foot width, and tau is the margin of foot width accepted for the maximal deviation related to command delay. The command delay should leave the possibility for the robot to react before it tips over. With tau = 1/8, Lf = 0.125[m], and LG = 0.27[m], the response time should be less than 6[ms]. In addition to this, the angular joint backlash must remain between negative 3 and +3 These values were set according to the maximal distance the COG can move from the upright position when tilting about the ankle with knee relaxed. Usually, this distance is set to be one-quarter of the foot length in the case of static equilibrium. Off-the-shelf motors for the legged - robots don't even exist. Designers of humanoids usually use existing brush DC motor brushless DC motor, or proprietary motors. Non-Brush DC motors present a good power density, and higher torque and speed bandwidths compared to brush DC motors. However, electronics is more complex and therefore more expensive. The motors used for the NAO actuators are Maxon coreless brush DC motors, that are known for precision and reliability. Even though harmonic drives are widely used for human-sized humanoids, they were not selected because they remain expensive, and there were not many providers. In addition off-the-shelf harmonic drives do not present enough back drivability and the reduction ratio is very high. In the case of the HRP-2, harmonic drives were used in conjunction with the timing belt and pulley. Taking this into account the designers of NAO decided to use spur and planetary gears in order to have fairly good back drivability. This technique was also adopted for the development of the ISA actuators. These kinds of gear offer very small technical difficulties. In addition, special plastics loaded with PTFE (Polytetrafluoroethylene) and carbon fiber were used to meet torque and longevity requirements. Investigating velocity as a function of required output torque yields to choose motor and gear reduction ratio for each joint. There are 2 kinds of motor, M1 and M2, and 2 types of reduction gear for each motor, R11 or R12 associated with M1, and R21 or R22 associated with M2.

Chapter 4

Computer - Electronics Architecture

4.1 Computer Architecture

NAO's head is equipped with an x86 AMD GEODE 500 MHz CPU motherboard with 256 Mb SDRAM. In additional we are available with 1Gb Flash memory . Communication with the robot is possible only through WiFi 802 , through Ethernet port and 11g protocol. The CPU of the robot can manage audio, video, and WiFi and other advanced modules. One ARM7- 60MHz microcontroller which is located in the torso distributes information to all the AMM(actuator module microcontrollers (Microchip 16 bit dsPICS)) through a RS485 bus (throughput of 460[Kbits/s]). It has 2 RS485 buses, one that connects the ARM7 microcontroller to the dsPICS modules of the top part of the robot body, and the other that connects the ARM7 to the dsPICS modules of the bottom part of the robot . The partition of the bus permits to increase the data throughput. The microcontroller(ARM-7) communicates with the CPU board through a USB-2 bus with a theoretical throughput of 11[M bits/s]. The Micro Controller can be used to control the robot's stability using the inertial unit. The operating system is based on Linux, but the whole system can be modified.

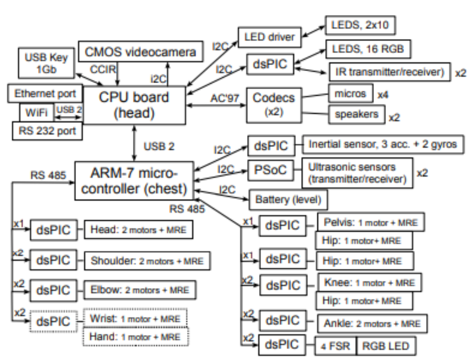

Figure 4.1: Electronic architecture of NAO. MRE stands for magnetic rotary encoder.

4.2 Electronic system

To control the actuators Custom designed integrated circuits based on Microchip 16 bit dsPICs microcontrollers were designed . These circuits are responsible for servo-control, bus control, sensor management, and power converters. The circuit can drive up to two actuators. The actuator is equipped with magnetic rotary encoders (MRE) that yield absolute outputs. Figure shows the overall electronic architecture of the system. One dsPIC based circuit, connected to the ARM7 board through I2C TM bus, is devoted to the signal acquisition from two gyrometers and three accelerometers. Signals issued from accelerometers and gyrometers can be combined to get an acceptable feedback of the robot's trunk orientation [12], [30]. Another dsPIC based circuit manages an infrared transmitter/receiver and a series of LEDS. For vision purposes the head of NAO houses a CMOS video camera. It is a 30 FPS camera with a 640x480 resolution. It can be controlled through the I2C TM bus. The robot is also equipped with two ultrasonic sensors, four FSRs per foot (force sensitive resistors), 4 microphones, 2 loudspeakers, Ethernet, serial and USB ports, 1 Gb USB key, WiFi interface through USB-2 bus, and a series of 20 LEDS. Low level control is updated every millisecond. The high level decision loop can be executed every 20[ms]. Sensor data is refreshed every cycle of 20[ms].

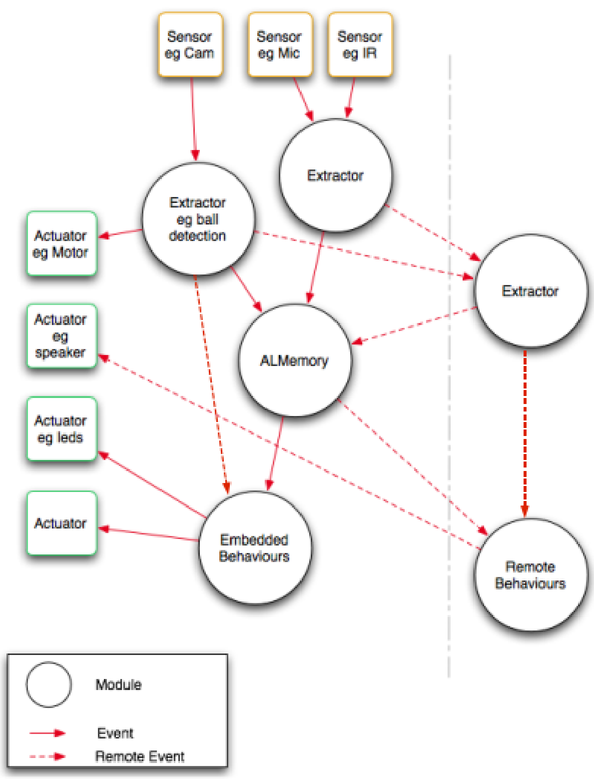

Figure 4.2: NaoQi architecture concept.

Chapter 5

Software Architecture

5.1 NaoQi distributed architecture

NaoQi is the Aldebaran Robotics software framework, that is registered for patenting. It is a modular and distributed environment that can deal with a variable number of executable binaries, depending on the user's architectural choices. The architecture is event-driven. A lot of effort was required to make the architecture support both parallel and sequential calling methods. The NaoQi architecture appears to be more user friendly than Corba-based architectures such as open-hrp used for HRP-2. The advantages of a distributed environment are manyfold. It allows the user to run behaviours locally or remotely. Robot functionalities such as motion, vision, text-to-speech, etc., can be run on a robot in the same executable or in a standalone executable that interacts with other modules on other computers. Development is made easier in a distributed environment since the same code can be compiled on different platforms and cross-compiled for embedded execution. A distributed environment also allows the user to look at variables and running methods on any real or simulated robot from the programming interfaces.

The functionalities of NaoQi are listed below.

- Programming in many languages. Users can control the real robot or simulate it with C++, Ruby, or Urbi . Programming in C, python, MATLAB,and Java will be possible in a future version.
- Executing methods through parallel, sequential or event driven calls.
- Process management. It includes finding a process and running a method in the process tree.
- Modularity. The user can choose whether to compile an application as a dynamic library or as an executable without changing source code.
- Multi-platform framework that supports Linux, Specific linux OS on Geode (AMD processor inside Nao), Windows XP, Windows Vista, and Mac OS X.
- Encapsulation of communication. The user can choose the process or the method to be executed without knowing SOAP or CORBA message passing for example.
- API management to show or hide methods to other applications.
- Shared memory management. Read, write and subscription procedures are available

There are three main object types in the NaoQi architecture:

- BROKER: its role is to expose modules to the rest of the architecture. A module must be linked to a broker to be accessible. Brokers manage network communication, and a broker can be defined as a child of another broker, this yields a distributed broker tree.
- MODULE: this contains user defined methods. It can expose methods called ”bound methods” to the rest of the system
- PROXY: this is designed to call a module wherever it is. The proxy explores the broker tree to discover the module location in the network, then it chooses the most optimised way to communicate with it.

When creating modules, the user instantiates a broker, then links his modules to the broker instance. The module will automatically declare the bound methods. At any time, the user can instantiate a proxy by specifying the desired module name. As soon as the proxy is ready the user can call bound methods without having to consider whether the module is local or remote, or if it is written in ruby, C++ or in any other compatible language. Three modules form the core of the NaoQi system, they are ALMemory, ALLogger, and ALPreferences. They are automatically loaded at boot-up. AL Memory enables intra-process or inter-process way to share memory. Aldebaran modules as well as usermodules can add and inspect variables from AL Memory. The ALPreferences module manages all preferences and initialization XML files. Each module can use ALPreferences to read or write attributes, or store them in the ALMemory module. It is important to note that the whole system is thread safe, and that the NaoQi architecture allows dynamic introspection.

5.2 Device Control Manager (D.C.M.)

The D.C.M. is a NAO real time module that is part of the NaoQi system, and is linked as a library. It is in charge of the communication with the electronic devices of the robot except sound sensors and video-camera. It can be seen as the link between the upper level software composed of other modules, and the lower level software that is embedded inside the electronic boards. The D.C.M. gets information from the electronic devices through the chest board and also accesses devices located in the head and connected though the I2C TM bus. For example, modules like Motion and LED’s can send commands directly to actuators using the D.C.M., while extractors and other modules use sensor results delivered by the D.C.M. to ALMemory. In case of actuators, modules need to send an update request to the D.C.M. using a timed command.

A timed command is an order containing a float number to be sent to a sub-device actuator and the time at which the order should be executed. More than one timed command can be included in the same request. The time is an absolute value in milliseconds based on the system time, and coded as a 4 byte integer. The D.C.M. stores all timed command for each actuator, then at each 20[ms] cycle it analyzes the previous and next orders based on the current time and computes the appropriate command to send using a linear interpolation. The main interests of timed command are:

- There is no need for the upper level to know the D.C.M. update time precisely, and no need of any synchronization mechanism, because precise command times are automatically used by the D.C.M. to send a good evolution of the command to actuators. The only requirement is to send commands in advance.
- As the D.C.M. knows actuators commands in advance, it can send them previously using its own thread, so that there is no delay between two commands from the actuator point of view, even if the module itself is delayed by the system or by the network latency.
- Other modules do not need to be real time. This relaxes contraints on programming.
- A whole choregraphy can be sent to many actuators at the same time. Whatever the communication delay or lags, actuators commands will be sent correctly.
- Synchronizing many actuators from different modules is possible, just by sharing the time.

5.3 Control using ALMotion module

The motion module, named ALMotion, was designed to facilitate the control of NAO, going further than simple joint-space commands to allow direct control of end effectors, or to directly manipulate the center of mass and request high level motions such as walk 10[cm] straight. The motion module offers the following options:

- resolve the kinematic model
- control the robot in joint space for direct control of joint angles
- control an end effector in the Cartesian space
- control the torso orientation
- control the COG (center of gravity) position relative to the support foot
- create and control walk primitives.
- control hardware parameters such as joint stiffness
- open-loop stabilizer

Before using inertial and FSR sensor feedback it is necessary to generate robust moves and walking motion in open loop. The control of the COG position relative to the support foot is inspired by the work developed by Sugihara et al, who defined the COG Jacobian to generate stable real-timed humanoid motion. Each joint, Cartesian or COG command is called a task. All tasks are filtered by priority, with the highest priority given to the balancing task (i.e. COG Jacobian) and the lowest priority to joint space commands. Once a task is given control and assigned the resources it needs, it can only be interrupted and killed by a higher priority task. For Example, if NAO is in double support mode (i.e. two feet on the ground), all the Cartesian and joint space commands that involve both feet will be ignored by balancing tasks that are active in this case. Before the balancing task, an internal kinematic model of NAO is updated relatively to all other joint and Cartesian tasks.

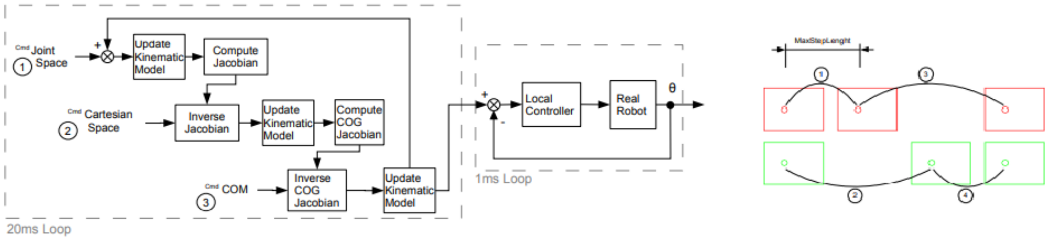

Figure 5.1: 1)Algorithm for ALMotion.2)Step primitives for straight forward walking.

So the COG Jacobian is computed taking into a count of all the perturbations on the robot COG involved by these tasks. Regarding the high level walking task, a foot step planner generates a body motion trajectory, the foot swing trajectory, based on a cycloid function and the joint space arm motions. The planner enables and walks to be composed using four walking primitives (straight, turn-in-place, side-step and arc). The COG trajectory is generated through a simplified dynamic 3D inverted pendulum model. Due to this simplification and because there is no feedback of real COG, some instability might occur in the robot. Therefore some adjustable parameters were defined. The ZMP (Zero Moment Point) offset in the x-direction describes a positive length starting under the heel position of a footstep, which causes the COG trajectory to linger longer on the supporting foot. The ZMP offset in the y-direction can be used to reduce the width of the COG sideways move. The robot is capable of walking forward and following arc of circle trajectories using the open-loop stabilizer based on the COG Jacobian. The robot doesn't fall as long as the ground remains flat enough. Future developments aim at incorporating a closed-loop stabilizer to equip the robot with better capabilities of resisting the disturbances while walking, that may arise from ground irregularities or collisions. This requires reliable feedback of the real COG and real ZMP.

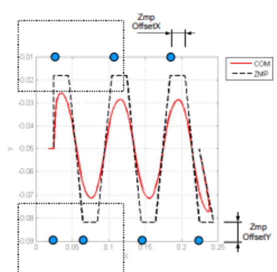

Figure 5.2: COG trajectory generated from the adapted ZMP trajectory.

5.4 Sensors

Interaction with environment, recognizing and understanding surroundings is an important fundamental skill for humans. Depending on the domain they are used, robots need to be designed to respond accordingly. As NAO is a humanoid robot, which looks like a person, it should also "function" like a human. To enable it to do so, several sensors and actuators are built into NAO.

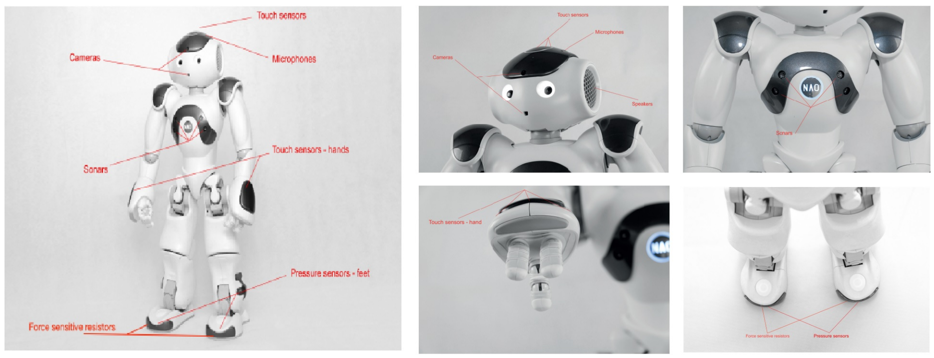

Figure 5.3: All sensing capabilities of NAO.

5.4.1 NAO's Head

NAO has three tactile or touch sensors on the top of its head, two cameras (the top camera is on the forehead and the bottom one is in the mouth) and four microphones. There are also two speakers. The tactile sensors can be used through a box.

5.4.2 NAO's Chest

On NAO's chest are four ultrasonic or sonar sensors. They can measure distances from walls, people, or objects.

5.4.3 NAO's Hands

There are three tactile sensors installed in each hand, which can also be controlled using a box.

5.4.4 NAO's Feet

NAO has one pressure sensor on the front of each foot which are called bumpers. There are also Force Sensitive Resistors (FSR) that measure the resistance change according to the pressure applied. They are used to detect if NAO is lifted for instance.

Chapter 6

References

1. NAO The Humanoid And Programmable Robot, SoftBank Robotics
2. NAO The Humanoid , A Combination Of Performance And Affordability, David Gouaillier, Vincent Hugel, Pierre Blazevic , Aldebaran-Robotics
3. Basics Of NAO
4. Project Ideas Of NAO
5. Choregraphe