Institute of Parallel and Distributed Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit Nr. 2469

Efficient Tele-Operation of a Robot Manipulator by Means of a Motion Capture Interface

Stefan Gutowsky

Course of Study:	Engineering cybernetics
Examiner:	Prof. Dr. Marc Toussaint
Supervisor:	Andrea Baisero
Commenced:	March 19,2015
Completed:	August 18, 2015
CR-Classification:	I.1.2, I.2.9

Abstract

One of the core challenges of developing an autonomous robotic system is that of establishing appropriate channels for imparting commands to the robotic system. A successful communication infrastructure is an important component not only for a robotic system at the final deployment state, where it would allow the users to specify tasks, but also during the development stages where the robotic system still has to learn desired behaviours and appropriate responses to the environment's events and other external stimuli.

In the framework of Learning by Demonstration, positive (and sometimes negative) examples are provided to the system by means of direct execution. Kinesthetic teaching consists in providing such demonstrations by physically moving the robot's body – alas, it is known to be a cumbersome and time-consuming procedure. Another approach is that of controlling the robotic system more indirectly through tele-operation.

We would like to build the appropriate infrastructure to control our PR2 robot using tele-operation, such that a human executer can perform a variety of tasks with minimal effort. We aim at achieving this using our high-precision Polhemus G4 motiontracking system to keep track of human poses and gestures, and interpret these as appropriate commands for the PR2. The ability of switching between various modalities of control is highly desired (e.g in one modality the user controls the position of the robot in the room; in another he controls the arms to perform manipulation; etc).

This project touches upon the subjects of task-space control, collision avoidance and compliance, and gesture recognition.

Kurzfassung

Das zuverlässige Senden von Befehlen zu einem Robotersystem ist eine kern Herausforderung für die Entwicklung solcher Systeme. Nicht nur im finalen Entwicklungszustand zum Steuern, sondern auch schon während der Entwicklung werden gut funktionierende Kommunikationsinfrastrukturen benötigt. Besonders während der Entwicklung, bei dem das System Bewegungsabläufe und Reaktionen auf die Umgebung noch zu erlernen hat.

Aus dem Kontext von "Leaning by Demonstration", bei dem durch physisches bewegen des Roboters, posistive sowohl als auch negative Beispiele generiert werden. Solche physische Demonstrationen sind bekannt dafür eine zeitaufwendige Prozedur zu sein. Ein weiterer Ansatz wäre ein Robotersystem indirekt über Teleoperation zu steuern.

Das Ziel dieser Arbeit ist es eine Infrastruktur zu errichten um den PR2 Roboter mittels Teleoperation zu steuern. Der Mensch soll die Möglichkeit erlangen eine Vielzahl von Aufgaben mit möglichst geringem Aufwand durchzuführen. Dafür benutzen wir unser hochpräzises G4 Motiontracking-System um Bewegungen des Menschen in Steuerbefehle für unseren Roboter zu übersetzen. Zusätzlich soll der Benutzer zwischen verschieden Moden wechseln können (z.B von Positioniersteuerung des Greifers zu Positioniersteuerung des Roboters im Raum).

Dieses Projekt umfasst die Themen "Task-space control", Kollisionsvermeidung, "Compliance" sowie Gestenerkennung.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Distantly human operated machines is a concept with a long histroy which led in the past decades to more complex devices. They enable users to fulfill tasks in hazardous situations, such as bomb disposals or explorations in deep sea and space. Teleoperated robots have proven to be helpful for physically disabled persons to achieve more independence and a higher quality of life by performing basic tasks for them.

The variety of areas in which teleopreated systems are used leads to the assumption that the success of a teleoperation scheme is determined by the systems design, including the human-machine-interface, hardware and controlling scheme.

It is in general possible to develop robotic systems that autonomously complete complex tasks which obsoletes procedures of direct human teleoperation.

In autonomous robotics an unknown environment pose is an unsolved problem and subject of current research in terms of reliability and execution time of complex tasks. In the framework of learning by demonstration it requires the access of example data.

In context of teleoperation and autonomy of a robot, a setup is needed allowing the robotic system to learn through the acquisition of data by teleoperation. The setup needs to feature high level of generality by covering a large amount of possible scenarios in which data gathering may occur. Another feature is intuitive controlling, but with precision, so that the user is able to safely manipulate a common household environment.

The objective of this bachelor thesis is to describe a possible scheme of a PR2 robots teleoperation via magnetic position and orientation sensors fixed on the users hand and foot. This framework helps to control the PR2 easily in an laboratory environment and it allows to define flexibly desired movements of the robot by mapping movements of the human operator. Gesture recognition is used for switching operational mode which leads to the opportunity to use the right hand as a steering device for driving the robot.

The teleoperation program used is based on a robotics framework provided by MLR. This robotics framework includes the low level controller executed on the robot with high level task space control and various math operation methods implemented in C++. This framework is crucial for this project because of its developing time. Teleoperation in general covers a wide array of topics, such as the design of the robots controller or its reception of sensor data. Moreover, this work accounts for hypothetically other robot controllers and shows possible effects.

The structure of this work has three main components: gesture recognition, G4 sensor mapping and pose execution. Each component is designed separately but tested on the whole system in the context of a modular structure for possible alternatives of the components.

1.1 Outline

Chapter 2 – Related Work At first, current work for each components are provided separately in a general context of teleoperation and robotics. Furthermore, it explains the particular choices for a method for developing the components in comparison with current research.

Chapter 3 – Background This chapter shows and describes the used hardware and theoretical background. Teleoperation is defined in a robotic context and the background for our high and low level controller.

Chapter 4 – Implementation This chapter provides the description of the components implementations, including the teleoperation system in the MLR robotic framework.

Chapter 5 – Evaluations and Conclusions In this chapter the evaluation is provided by comparing the results with related work. In addition conducted experiments with our teleoperation system are described.

Chapter 6 – Summery At last we make a short summery for this.

2 Related Work

Teleoperation or telerobotic systems are designed to meet specific requirements. Revisiting related work in this field is needed to show possible solutions and limitations of existing strategies. Furthermore, the objectives are decribed in detail and compared it with the revisited work.

2.1 Gesture recognition

The works objective of gesture recognition is to develop a method for switching the control modularity. For instance, indicated by a foot tap the controller switches the position of the effector to driving mode and vice-versa. The work of [SPHB08] suggests good results with a machine learning approach, even with low amounts of sample data. However, interpreting a gesture as a time series and recognizing another leads to a more simple suitable solution comparable with a matched filter. In the work [AH] a matched filter is used for locating blood vessels in retinal images, we use an adapted version of this method of detection on live data because of its less time-consuming features for supplement of a machine learning approach. The time series of a recorded double tap Fig. 2.1 on the following page shows the short time duration and distinguishable course of the recorded tap, as well as how the method enables the recording of different gestures.

2.2 Sensor Mapping

This thesis defines sensor mapping as the interface between the human and the robot, regarding the hardware and the task to achieve. Furthermore, clarifying which motion of the human correspond to which motion of the robot, is important and it suggest dependence of the hardware used. This becomes more clear regarding [MB15], the authors present a brain control teleoperation scheme. The paper [JB13] investigate a teleoperation scheme for high latency networks with a Cartesian impedance velocity controller. A more related project [BA] investigates a similar approach as our, using teleoperation for learning by demonstration purposes. They compare teleoperation with kinesthetic teaching as a way to acquire learning data with the result, teleoperation in general is more complicated then kinesthetic. Therefore a

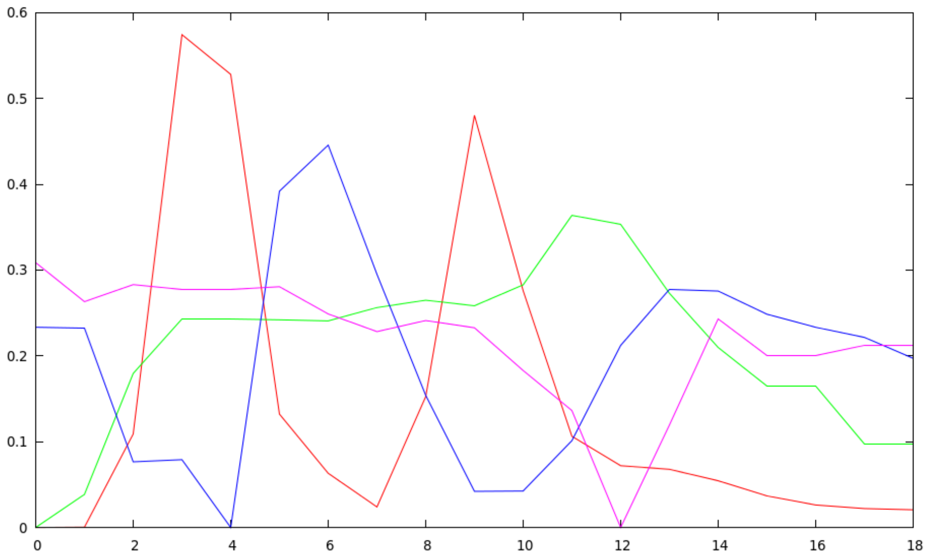

Figure 2.1: Time series of a double tap. Red(z-axis),blue(first quaternion), green and pink(live data)

from of hybrid teaching used which yield good results. The criteria for a successful teleoperation system is simplicity and user friendliness.

2.3 Controller

The controller determine the robot properties, for that reason we want so compare which controller strategy would yield the best results for a teleoperation system. Further for achieving compliance and collision avoidance, this work uses hybrid control which enables the user to control the position and the maximum exerted force by the endeffector. The aim for collision avoidance is to prevent the endeffector for destroying it self and the environment, this is achieved using our hybrid control strategy. The work [WFM91] predict unstable behavior in the case of hybrid control. By using selector matrices with the Jacobian, the hybrid control scheme loses the unstable properties. In addition the authors shows the cause of the unstable behavior, which is contributed by motion controller. This result is suggestive because force and position cant be always controlled perfectly at the same time. Force get controlled in the expense of position. The authors of [RJA88] show a hybrid impedance control approach which is similar to ours. However we did not model the environment and we formulate our force controlling as an inequality problem. The inequality term just get active in the case of force violation of the endeffector.

3 Background

This chapter present the methods and background used for implementation for our teleoperation system. First we show a definition of teleoperation presented by the authors of [SK08]. Followed by the description of the hardware and their properties used for this work. For the robotics background we use the lecture scripts of Marc Toussaint [Tou14].

3.1 Teleoperation/Telerobotics

The basic concept of teleoperation or telerobotic is to control a machine from a distance. This concept makes use of the cognitive capabilities of a human to decide what the machine shall do next. In Handbook of Robotics [SK08], different concept of telerobotics are presented. The architectures are explained with Fig. 3.1, the autors distinguish between direct control, shared control and supervisory control with increasing autonomy in that order. It is possible for teleoperation systems to have properties of out of every category.

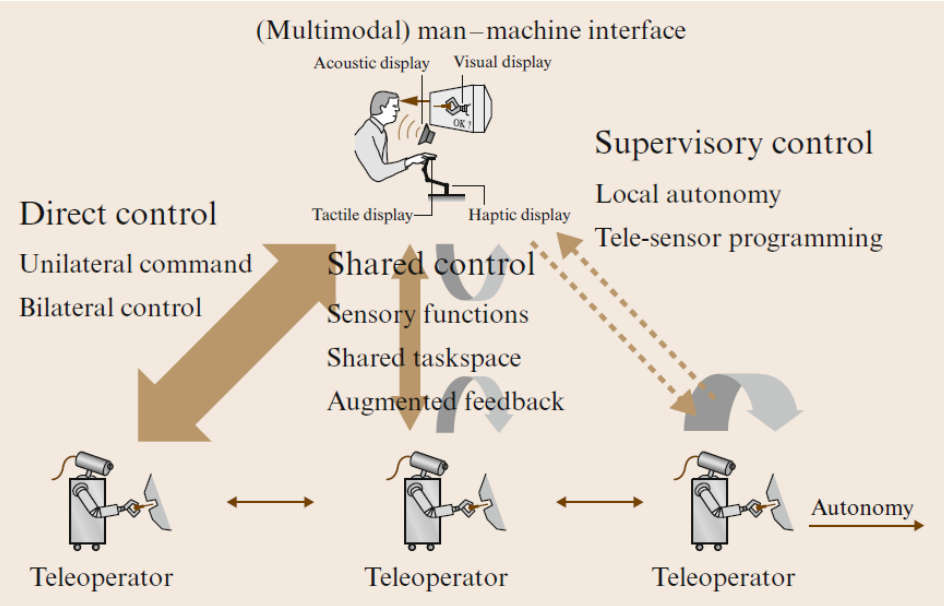

Figure 3.1: Different concept for telerobotics control architecture
Source: [SK08]

3.2 Hardware setup

Fig. 3.2 shows our hardware and sensor setup up for the right hand. In addition to the G4 sensors we use a gamepad Fig. 3.2 (G) for providing the teleoperation program with instructions.

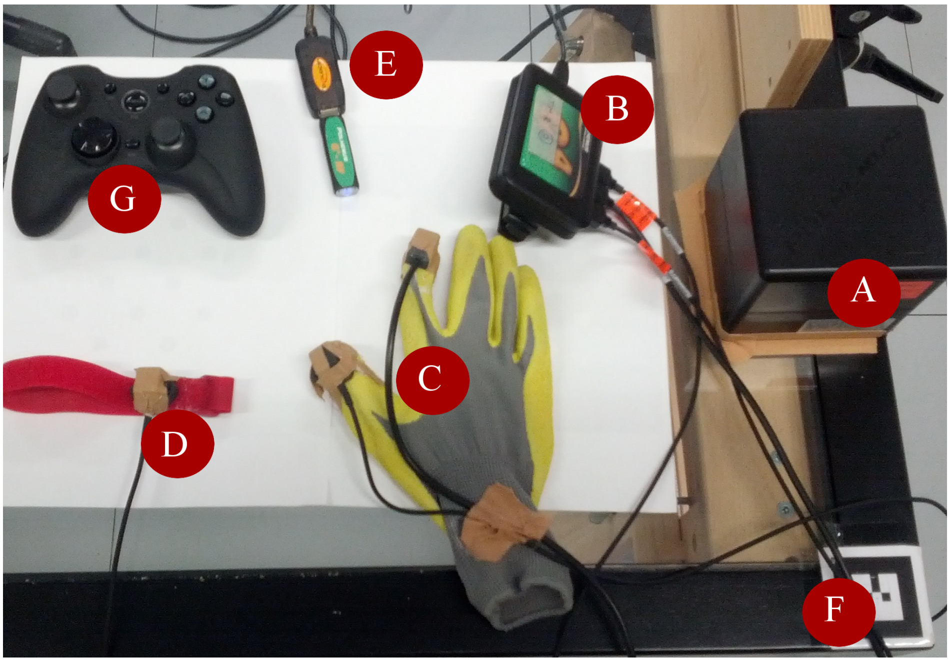

Figure 3.2: Our hardware setup with indicators

3.2.1 The PR2

The PR2 (Personal Robot 2) from Willow Garage is a humanoid alike robot, which is well suited for research applications. Equipped with two high-performance on board servers, numerous actuator and the option to attach additional hardware. Fig. 3.3 on the facing page shows the PR2 basic structure with A) the arm (4DOF), B) the wrist (3DOF) and C) the gripper (1DOF). Our PR2 is equipped with force/torque sensors mounted between the gripper and the forearm for measuring the exerted force by the gripper. In addition we use the work of [DK08] to calibrate the f/t sensor which compensate the offset and the gravity load. The f/t sensors provide the force readings for our hybrid controller.

The PR2 runs ROS (Robot Operation System) as an operating system. ROS is a open source project and enables the user a more easy developing of application. Furthermore, ROS provides a communication structure via nodes and topics. This infrastructure is used by our high and low level controller as a commutation channel for sending status and references. In the data

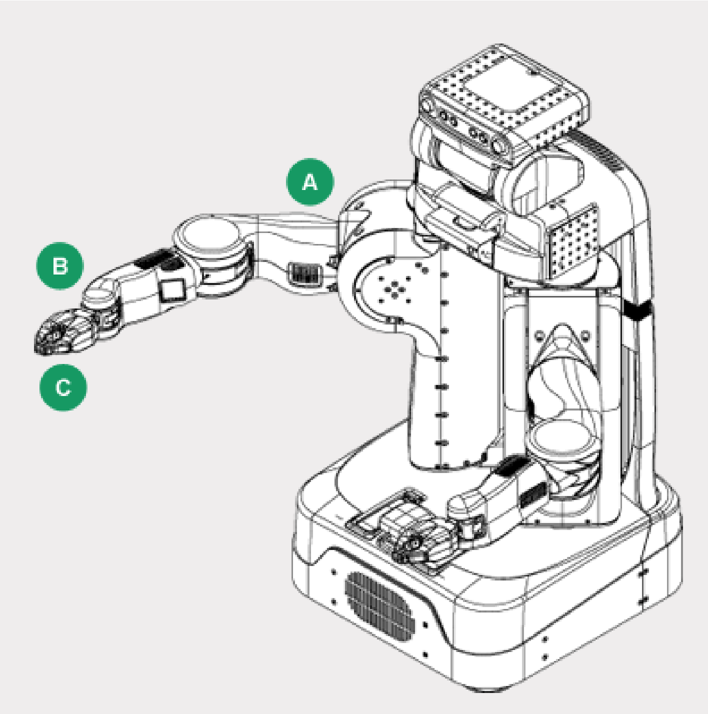

Figure 3.3: Schematic of a PR2 Source: [Wil]

acquisition context of Learning by Demonstration we want to use the published topics by the controller. This neglect the need of designing a data output port in our teleoperation system. If the need of extra output arises, the program can be modified with low amount of effort.

3.2.2 G4 Polhemus

The G4 sensors from the company Polhemus provide the hardware to capture the motion of the human operator. The sensor system is capable of a sampling rate of 120Hz and is accurate by 2 mm in position and and 0.5 degree in orientation(in steady state operation and no radio noise, Source: [Pol]). Our setup consists out of 2 stations Fig. 3.2 on page 12 (A) with two hubs (B), containing two sensors (C) for the left/right hand each and one sensor (D) for the foot. The sensors are attached to a glove via medical tape. The sensors detect the magnetic field generated by the stations to determine the sensors position and orientation. The G4 setup transmit a vector with the position in meters and the orientation in quaternions in a common reference frame, (F) in this case the lower right corner of the table. The transmission is wireless therefore (E) is the receiving dongle.

3.3 Robotics

Our PR2 platform use two controllers, a high level controller running at 50Hz sample rate and a low level controller running at 1kHz sample rate. The high level controller generate the reference for the low level controller. The reference is computed by a task-space scheme using the motion capture data provided by the G4 sensors. The low level controller runs on the PR2 which executes the reference in a real time environment. Furthermore the low level controller regulates directly the force reference send by the high level controller. The basic controller scheme is shown by Fig. 3.4. The human operator provide the feedback in this control loop. The "human-in-the-loop" corrects via visual feedback possible controller error, therefore the need of controlling the steady state error by the controller is neglected.

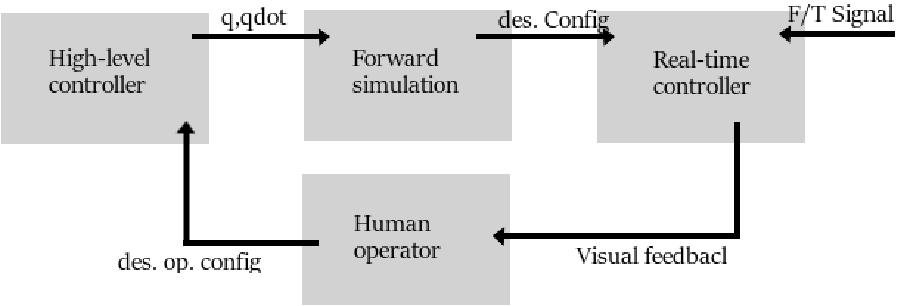

Figure 3.4: Controller strategy

Firstly we define the notation which is used for further description in this work. We define $\vec{q}$, $\vec{\dot{q}} \in \mathbb{R}^n$ as a vector containing the angle and angle velocity of each joint. $\vec{u} \in \mathbb{R}^n$ as the control signal,$\vec{f} \in \mathbb{R}^d$ as a force vector and $J \in \mathbb{R}^{d \times n}$ as the Jacobian matrix. Furthermore $\phi(\vec{q}) = \vec{y}$ with $\vec{y} \in \mathbb{R}^d$ denoted the kinematic map of the robot and y the cartesian position of a body (e.g endeffector).Further M, C, G denotes the inertia, coriolis/centripetal and gravity matrices. The assumption of ideal properties of the robot structure is made, so that there are no deformations under force.

3.3.1 Kinematics

The kinematics of the robot describe the structural properties of the the robot in means of $\vec{q} \rightarrow \vec{y}$. For known $\vec{q}$ the position $\vec{y}$ is certain. This relation can be described by kinematic tree, therefore by a sequence of coordinate transformation matrices.

$$\phi(\vec{q}) = T_{base \rightarrow body} = \prod_{i=0}^{body} R_i * T_i \tag{3.1}$$

R_i,T_i is a rotation and translation matrix, chaining these matrices from the origin for each body describes the kinematic map $\phi(\vec{q})$.

(3.2) $\phi(\vec{q}) = \vec{y}$

This relation state the equation 3.2. In order to find a relation $\vec{y} \rightarrow \vec{q}$ we formulate the inverse.

(3.3) $\phi(\vec{y})^{-1} = \vec{q}$

Since ϕ is often overdetermined due the robotic kinematic or there is no solution, ϕ gets singular and therefore it exist no inverse. To find a approximation, we formulate this problem as an optimization problem.

(3.4) $\vec{q} = argmin_q \, \|\phi(\vec{q}) - \vec{y}\|_C^2 + \|\vec{q} - \vec{q}_0\|_W^2$

This formulation uses W and C as regulation matrices to ensure numeric stability, further the terms can be interpreted as potentials. Solving this equation state:

(3.5) $\vec{q} = \vec{q}_0 + J^{\#}(\vec{y} - \vec{y}_0)$

with the Woodburry identity.

(3.6) $J^{\#} = (J^T C J + W)^{-1} J^T C = W^{-1} J^T (J W^{-1} J^T + C^{-1})^{-1}$

Forming the derivative of the task map w.r.t each joint gives us the jacobian, which can be interpreted as an infinitesimal offset in the joints correspond to infinitesimal offset in the position.

(3.7) $J(\vec{q}) = \dfrac{\partial}{\partial \vec{q}} \phi(\vec{q}) = \begin{pmatrix} \frac{\partial \phi(\vec{q})_1}{\partial \vec{q}_1} & \dots & \frac{\partial \phi(\vec{q})_1}{\partial \vec{q}_n} \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ \frac{\partial \phi(\vec{q})_d}{\partial \vec{q}_1} & \dots & \frac{\partial \phi(\vec{q})_d}{\partial \vec{q}_n} \end{pmatrix}$

This approach of inverse kinematics is the linearized solution of an non linear system, therefore the solution is only valid in near surrounding of the linearisation point. However, using 3.4 we are able to define more tasks. Combining this task-space approach with imprinting motion profiles to each task, we are able to generate a PD-like behavior for each task. The table 3.1 shows the task used in this work, further the PD behavior is set at initialization and the parameters are heuristic determined.

Task	Description
homing	always active task, preventing to robot for strange configurations
endeffector position L/R	for achieving a desired position in euclidean space
endeffector orientation L/R	for achieving a desired orientation
endeffector gripper joint L/R	for controlling the configuration of the gripper
robots driving	for driving and rotating the robot

Table 3.1: The tasks used in this work

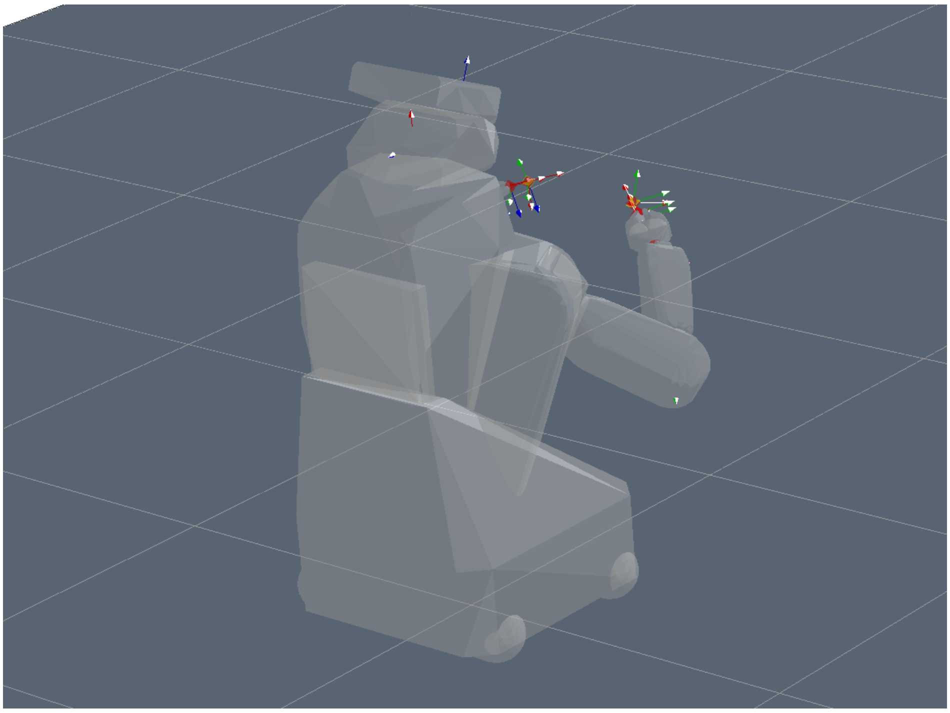

Figure 3.5: This image shows the homing pose of the PR2

3.3.2 Dynamics

Considering controlling torques $\vec{u}$ of the robot joins, there is the need for regarding the masses. By deriving the robots dynamics with the Euler-Lagrange equation, we formulate:

(3.8) $\vec{u} = M(\vec{q})\ddot{\vec{q}} + C(\vec{q}, \dot{\vec{q}})\dot{\vec{q}} + G(\vec{q})$

Formulating this problem as an optimization problem like in the task space approach, we are able to find suitable control torques.

(3.9) $\vec{u} = argmin_u \left\| \ddot{\phi}(\vec{q}) + \ddot{\vec{y}} \right\|_C^2 + \|\vec{u}\|_H^2$

A rearrangement of this equation sets the opportunity to tune the controller for a PD-like behaviour.

Hybrid Control

For controlling the endeffectors force, we add to the equation 3.10 an integral term.

(3.10) $\vec{e} = \gamma\vec{e} + (\vec{f^*} - J_{ft}\vec{u})$

J_{ft} is computed by the high-level controller and is send to the low-level controller. With this addition we are now able to control the endeffectors position and force. This enables the user to write on a surface with constant force. However due to possible instability we set the activation of the force integral in certain contains in means of force boundaries. If the measured force is to high, the force integral starts to contribute to $\vec{u}$. γ is a decay rate values and contributes to the stability of the controller.

4 Implementation

The following chapter we presents the implementation of the three parts of our teleoperation system. 4.1 shows the structure of our teleoperation program. Each box represent one part of the program and the arrows indicate which information are transmitted and received. In the following sections each part is elucidated in more detail. This work uses the MLR robotics framework and therefore gesture recognition, G4 mapping and controller are in principle timed threads with a step-function which get called in fix time intervals. Therefore each section start to show the step-function routine.

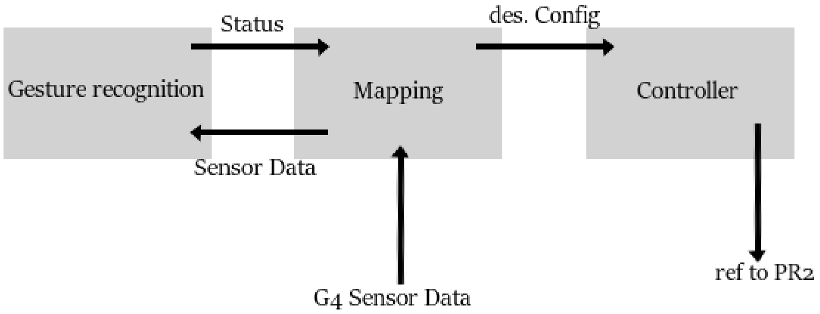

Figure 4.1: Basic teleoperation structure

4.1 Gesture recognition

As mentioned in related work, the requirement of our gesture recognition is to recognize one tap with a suitable processing speed and accuracy in detection. The sensor is attached to the foot by a Velcro fastener. Our gesture routine can further be divided in three subroutines: recording, testing and operational. The recording routine records the tap, the testing routine firstly test the recording with the operator taping again. If the recording and testing are successful the operational routine starts and indicates tap.

4.1.1 Implementation

The step routine implementation is show at 4.1. Firstly the input sensor data is tested by comparing if the data is not zero. The recording is started by pressing and holding the corresponding button on the joy pad. By holding the button for testing, the recorded sample is tested with the live data. If the testing is successful the routine start to listen to the input data continuously and indicate the occurrence of a tap.

Algorithmus 4.1 Gesture step function

```
procedure STEP($button$,$s_f$)
    if sensor data or gamepad is not okay then return
    end if
    if ready then
        CORRELATEWITHSAMPLE($s_f$)
        if Tap recognized then
            taped ← true
        end if
    else if button for recording pressed and not ready then
        RECORDSAMPLE($s_f$)
    else if button for testing pressed then
        CORRELATEWITHSAMPLE($s_f$)
        if Correlation okay then
            ready ← true
        end if
    end if
end procedure
```

The recorded sample determine the length of the array and therefore the length of the live data array. To achieve position independent recognition each array get transformed in respect to the first entry, so the first entry states the origin of each array. The sample and the live data get normalized and shifted such that the lowest value is zero. The correlation value is bounded between zero and one. In this case identical signals the correlation value becomes one. Therefore a arbitrary choice of the degree of similarity can be set. Since the provided data by the G4 sensor contains the position and orientation we choose to use just the entry for z-axis and the first entry of the orientation for minimizing computation time.

$$(r \star s)[n] = \sum_{m=-\infty}^{\infty} f[m]g[m+n] \tag{4.1}$$

Using the definition of the discrete cross-correlation 4.1 and the live data input stream, we don't have shift the samples because the position get shifted by the live data input stream. Therefore the cross correlation becomes a scalar product analyzed just once at a step-tick.

4.2 Sensor Mapping

In order to generate control signals from our G4 sensors we need to find a suitable mapping for the sensors. This requires finding a transformation from the sensors frame, to the robots frame of reference. We implement separate routines for manipulating the arms/gripper and for driving mode. In order to manipulate the arms/grippers, we implement a transformation for each task defined by the task-space-controller. For driving mode we need a nice way how the user can send driving commands. The algorithmus 4.2 shows the step-function, each mapping is substituted in the according place in the algorithm.

Algorithmus 4.2 Mapping step function

```
procedure STEP($button$,$s_d$)
    if sensor data or gamepad is not okay then return
    end if
    TRANSFORMTOROBO($fixpoint$,$s_d$)
    if drive init is ready then
        INITDRIVING($button$,$s_d$) return
    end if
    WAITFORINPUT($button$)
    if Tap recognized and init finish then
        DRIVING($s_d$)
    end if
    PROCESSDATA($s_d$)
end procedure
```

Firstly the step-function checks if the G4 sensors and the game pad are ready and connected. The function TransformtoRobo performs the first transformation by rotating the sensor frame to the robots frame such that x,y,z are aligned. Initdriving just get called using the stencil mapping. It performs the calibration for the stencil in use. WaitforInput wait until the operator is ready for operation, this is indicated by a button on the game pad. If a tap get recognized the function Driving get called and the driving mapping get executed. ProcessData performs the transformation according to the active mapping.

4.2.1 Arm/gripper mapping

In order to find a suitable mapping we show two different strategies, human fixed and a reference frame fixed coordinate system approach. The main difference is that the human fixed strategy need calibration prior for transforming the sensor data properly.

Human fixed frame

For the calibration procedure we denote a optimization problem.

$$f = min_{s,r} \sum \left\| \left\| \vec{s} - \vec{d_i} \right\|^2 - r^2 \right\|^2 \quad (4.2)$$

$\vec{s}$, $\vec{d_i}$ and r denote the shoulder position, the live data and the operator arm radius. All sensor data are expressed in the frame of a human fixed sensor, which is attach to the chest or neck. By minimizing $\vec{s}$ and r we are able to determine the shoulder position and arm radius of the user. This requires the operator to do circular motion with his hands and with straight arms for a predefined number of samples. The shoulder position is than used as the reference frame for the right and left arm. The human fixed sensor is the reference for orientation. The position signal is than generated by sending the sensor signals which are expressed in the shoulder frame to the robots shoulder frame.

Reference frame fixed

This approach transform a fixed point in the sensor frame of reference, to fixed point in the robot frame of reference. The fixed point is used as the new reference point, this method requires no calibration for initialization.

The orientation of the gripper is for both transformations the same. The index finger and thumb is used to construct a new coordinate system in between the sensors. The same is valid for the gripper which is a direct mapping of the distance of the finger sensors to the gripper.

4.2.2 Driving mapping

The driving commands for the PR2 consists of forward/backward, left/right, turn left/right. We present two different methods how the user can generate these commands. The thumbs-up mapping generate the commands by tilting and rotating a thumb-up gesture and the stencil mapping by pointing at the commands on a piece of paper.

Thumbs-up mapping

As mentioned this mapping use a thumb-up gesture by using the sensors fixed to the index finger and thumb. Subtracting and normalizing the position vector of the index and thumb, the relative projection on the x-y plane is determined. The projection in x direction is interpreted as forward/backward and in y as left/right. By turning the gesture in the z-axis the turning command is generated.

Stencil mapping

This mapping use a printed stencil with commands like on a control board therefore it requires calibration prior to work, fig:stencil shows a stencil layout. The calibration is done by pointing on each outer bound for forward/backward and left/right. This generates a linear function for the two directions. The turning command is calibrated by pointing the foot on predefined spots. The gray area on the stencil indicate a movement just in one direction, the white area indicate a composition and the black area is for no operation.

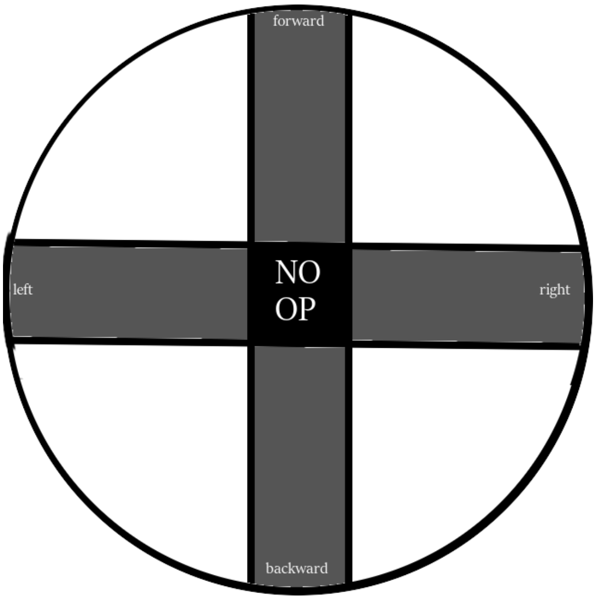

Figure 4.2: A possible layout for a stencil

4.3 Control

The algorithm 4.3 shows the high-level controller step-function. Firstly the consistency of the model with the low-level model is checked by counting the number of joins. Then the status of the mapper is checked. If everything is okay the tasks is set accordingly of a occurrence of a tap and the status of the mapper. Then the tasks are set to the reference of the mapper. Followed by the task-space controller and forward simulation via Euler-solver. The force control parameters are calculated with the real state of the robot, resulting in $J_f t$ and a K_i gain. SetRosMsg send all necessary massages via ROS topic to the low-level controller.

Algorithmus 4.3 Control step function

```
procedure Step
    if use ROS and number of joins are NOT okay then return
    end if
    if Mapper is NOT ready then return
    end if
    SetTask($Tap$,$Mapper$)
    if Mapper ready then
        SetAllRef
        OperationalSpaceControll
        EulerForwardSim
        CalculateForceParameter
        SetRosMsg
    end if
end procedure
```

5 Evaluations and Conclusions

To evaluate our teleoperation system we want to test how well the user can interact with the robot and the environment. Since finding measurable quantities which indicate the success of this teleoperation scheme for acquiring learning data is not a trivial problem, we test the system on different tasks like driving around while manipulating the environment and the stability of the hybrid control plus the users experience while using this scheme.

5.1 Evaluations

For the testing we use the stencil mapping plus the the reference fixed mapping. The human fixed mapping turned out to be problematic while calibrating and in use. If the operator is not calibrating the system with straight arms, the result are not suitable for operation. Further, because of breathing or head movement the chest or neck fixed sensor is moving and therefore it introduced jitter to the transformation plus the calibration is time intensive regarding conducting a set of experiments. The thumbs-up drive mapping turned out to be problematic as well because it can happen while concentrating on opening a door that the operator gives unintended driving instructions. The stencil mapping needs calibration but just for 7 points which is acceptable for setting this system up.

5.1.1 Experiments

First we want to conduct a experiment to show witch force our hybrid control will exert. This setup use a digital scale for measuring weight. A operator control the arm to push on the surface of the scale. The maximal force is set to 6NM.

Further experiment are aiming to drive and manipulate objects in the environment, e.g driving to a door and open it, grasping and releasing a bottle and placing a bottle in a paper bag.

5.1.2 Results

While conducting the experiments we experienced from time to time connection losses to the G4 sensors, which resulted in constant values of the input data. In general this would not state a problem but further the foot fixed sensor swapped sometimes the sign of the z-axis which render the gesture recognition incapable to recognize a tap. Since this swapping occurs not regularly a well reasoned validation for the gesture recognition is not possible.

The hybrid control experiment shows the force is limited and its value is in proximity of the reference (6NM). Further the hybrid control can be unstable in some configurations.

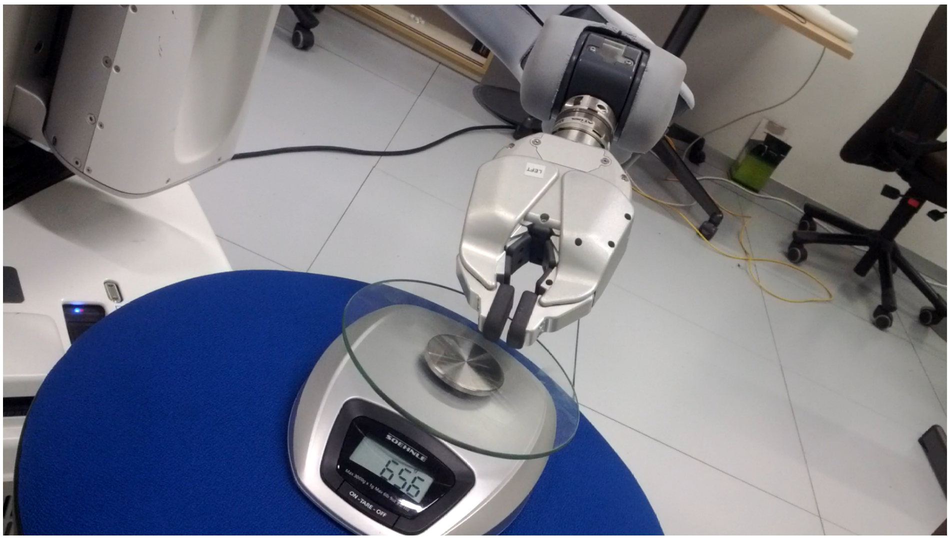

Figure 5.1: This image shows the PR2 pushing on a scale

The operator was able to perform each test shown at Fig. 5.2 on the next page, Fig. 5.3 on the facing page, Fig. 5.4 on page 28 successfully.

However this teleoperation system enable the user to open doors, to exert a constant force while position controlled and to manipulate the environment while driving. Figures shows the first try with the teleoperation system. The clumsy behavior is rather due the unpracticed operator who struggle with the shift in perspective. The force instability could be minimized by further tuning of the force gains.

5.2 Conclusions

Regarding the results, this teleoperation system is capable to perform a descent large set of possible tasks. The setup time is fast enough to perform quick experiments and it is safe enough to operate among humans. However, [WFM91] predicted the possible instability of this scheme,

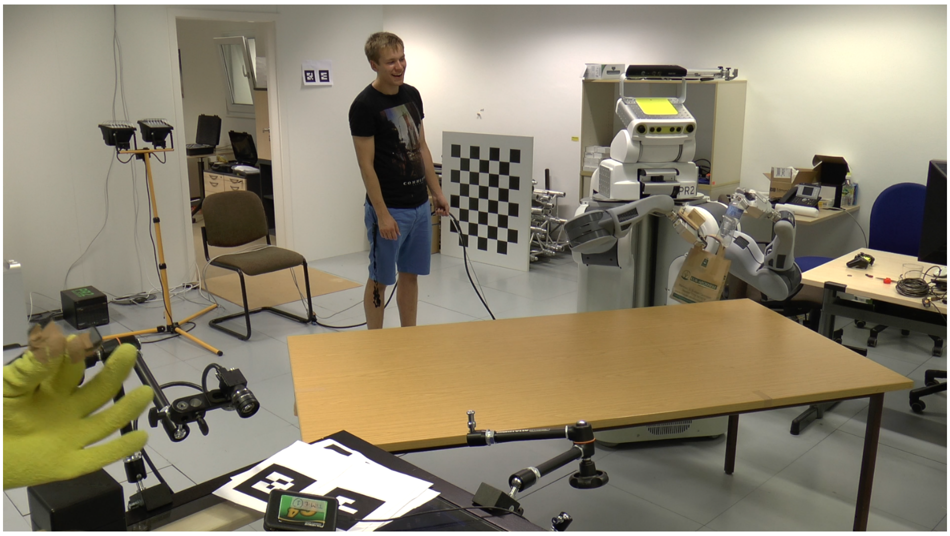

Figure 5.2: Performing: bottle into bag

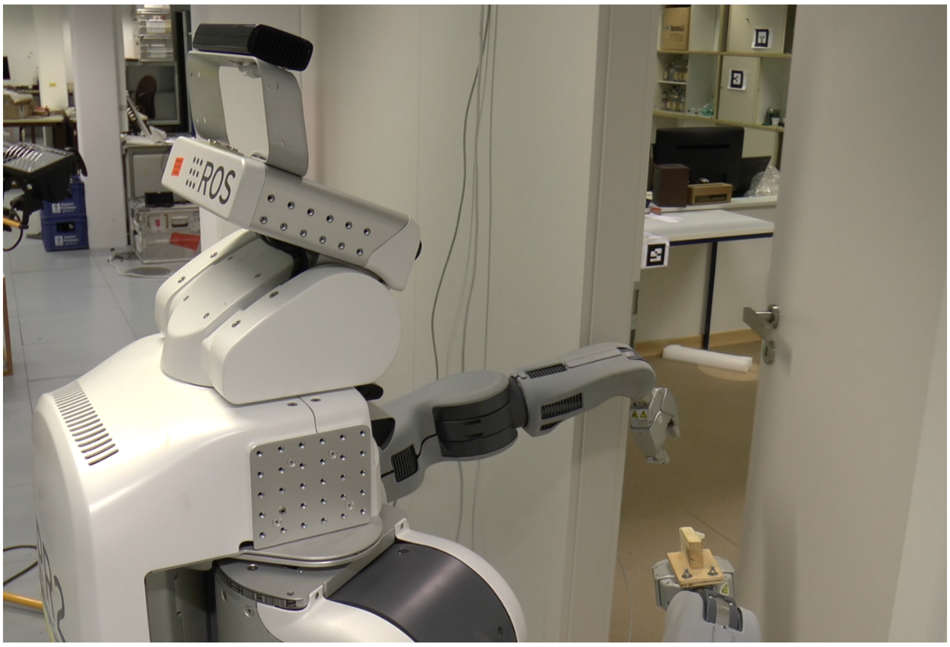

Figure 5.3: Opening a door

therefore implementation of other controller strategies could improve the teleoperation scheme shown in this work. Real live scenarios will show if the result of [BA] are correct since we tried to minimize setup time and user friendliness.

Since the G4 sensors cant provide force feedback it would be nice to use hardware which enables the user to feel a force. A gripper force feedback device could provide useful information about a grasping process. Equipping a PR2 model figure with a sensor could state suitable alternative

Figure 5.4: Preforming bottle hand over

for a new mapping strategy. The PR2 robot would be moved like a chess figure along a plane. In addition a GUI for our teleoperation program would improve user friendliness further with a head mounted display to use the PR2s head cameras.

6 Summery

This thesis investigates and present a teleoperation setup with G4 sensors and a PR2 to enable a human operator to record learning data for machine learning purposes. In the course of this work we consider related work in the field of teleoperation, gesture recognition and controlling to find a suitable solutions and possible restriction of a method. Further we compare the related work with ours to perform a validation.

Prior to the implementation we explain our setup on hardware and theoretical background used in this work. We explain basic robotics and the general controller scheme for our PR2. Provided with the necessary background we show the implementation of our gesture recognition, sensor mapping and the controller. The teleoperation system is evaluated by experiments conducted, to rate the use of this teleoperation scheme to manipulate objects. The result are discussed along with related work to state possible points of improvements. With the method shown in this work a suitable manipulation of the environment is possible.

Bibliography

[AH] M. G. Adam Hoover, Valentina Kouznetsova. Locating Blood Vessels in Retinal Images by Piece-wise Threshold Probing of a Matched Filter Response. (Cited on page 9)

[BA] A. L. T. Baris Akgun, Kaushik Subramanian. Novel Interaction Strategies for Learning from Teleoperation. (Cited on pages 9 and 27)

[DK08] F. W. D. Kubus, T. Kroger. On-line Rigid Object Recognition and Pose Estimation Based on Inertial Parameters, 2008. (Cited on page 12)

[JB13] D. B. K. K. S. P. S. H. W. L. S. G. D. H. L. L. W. Jonathan Bohren, Chavdar Papazov. A Pilot Study in Vision-Based Augmented Telemanipulation for Remote Assembly over High-Latency Networks, 2013. (Cited on page 9)

[MB15] G. N. L. C. J. R. S. R. P. N. R. Matthew Bryan, Vibinash Thomas. What You Think is What You Get: Brain-ControlledInterfacing for the PR2, 2015. (Cited on page 9)

[Pol] Polhemus, http://polhemus.com/motion-tracking/all-trackers/g4. *G4 Specs*. (Cited on page 13)

[RJA88] M. W. S. Robert J. Anderson. Hybrid Impedance Control of Robotic Manipulators, 1988. (Cited on page 10)

[SK08] P. B. Siciliano, P. O. Khatib. *Handbook of Robotics*. Springer-Verlag Berlin Heidelberg, 2008. (Cited on page 11)

[SPHB08] T. Schlomer, B. Poppinga, N. Henze, S. Boll. Gesture Recognition with a Wii Controller, 2008. (Cited on page 9)

[Tou14] M. Toussaint. Introduction to Robotics, 2014. (Cited on page 11)

[WFM91] D. William, D. Fisher, D. M. S. Mujtaba. Hybrid Position/Force Control: A Correct Formulation, 1991. (Cited on pages 10 and 26)

[Wil] Willow Garage, http://www.willowgarage.com/pages/pr2/specs. *PR2 Specs*. (Cited on page 13)

All links were last followed on August 16, 2015.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature