DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DESIGN AND VISION BASED CONTROL OF A MOBILE MANIPULATOR

by

Levent ÇETİN

February, 2008

İZMİR

DESIGN AND VISION BASED CONTROL OF A MOBILE MANIPULATOR

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Mechanical Engineering, Machine Theory and Dynamics Program

by
Levent ÇETİN

February, 2008
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"DESIGN AND VISION BASED CONTROL OF A MOBILE MANIPULATOR"** completed by **LEVENT ÇETİN** under supervision of **PROF. DR. EROL UYAR** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Erol UYAR

Supervisor

Assoc. Prof. Dr. Yalçın ÇEBİ

Thesis Committee Member

Assist. Prof. Dr. Zeki KIRAL

Thesis Committee Member

Prof. Dr. İbrahim YÜKSEL

Examining Committee Member

Prof. Dr. Mustafa SABUNCU

Examining Committee Member

Prof.Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge the supervision of Prof.Dr. Erol Uyar, during this work. I thank Assoc. Prof. Dr. Yalçın Çebi and Assist. Prof. Dr. Zeki Kıral for the discussions on periodical meetings of this research. I would like to acknowledge the help of Assist.Prof. Dr. Mutlu Boztepe with experimental setup and his numerous suggestions on the electronics design to improve the performance of implementation setup and general advice.

I am grateful to all my friends from Dokuz Eylül University for their moral support. I thank especially to my colleagues Aytaç Gören, Özgün Başer Onur Keskin and Emre Akçay for numerous stimulating discussions, help with experimental setup and I would like to acknowledge also the help of group of students: Sercan Karakoç, Ersin Arslan, Sevban Köse Arıkan Doğan and Osman Korkut for their assistance with all types of mechanical problems.

Finally, I am forever indebted to my family for their understanding, endless patience and encouragement when it was most required.

Levent Çetin

DESIGN AND VISION BASED CONTROL OF A MOBILE MANIPULATOR

ABSTRACT

An autonomous mobile manipulator, a wheeled mobile platform carrying a manipulator arm, with a vision based target acquisition system can be able not only to find and maintain fixation on a moving target but also to manipulate or handle it while the system itself is in motion.

In this context, a mobile manipulator is designed and constructed with its subsystems: electromechanical, electronics and computer parts. Resultant robotic system consists of differentially driven mobile platform and a planar manipulator with servo controllers and measurement circuits. A calibrated stereo rig is mounted on mobile platform. This vision system is to obtain task space feedback for mobile manipulator control. A computer is integrated to overall setup as a controller which exploits vision system data.

Using vision based control methodology, two different control types for mobile platform and manipulator are applied. Mobile platform is controlled with a hybrid visual servo controller. To control instantaneous linear velocity of mobile platform, a position based visual servo algorithm is developed and angular velocity, it pivots, is controlled with image based visual servo control law. Manipulator motion is controlled via dynamic look and move control strategy, in which actuator references updated instantaneously according to acquired target position.

Control of the mobile manipulator, first, is done by task sequencing. A task is separated in to two parts: Mobile platform moves till target is placed in manipulator workspace and in proper orientation that gripper (tracking) can hold, afterwards manipulator moves to hold target (handling).The second control strategy developed is focused on redundancy of mobile manipulator and its motion controlled via moving manipulator and mobile platform simultaneously.

BİR MOBİL ROBOT KOLUN TASARIMI VE GÖRÜNTÜ DESTEKLİ KONTROLÜ

ÖZ

Hareketli bir platform ve üzerine monte edilmiş bir robot koldan oluşan otonom mobil robot bir görüntü sistemi kullanarak bir hedefi bulup takip edebildiği gibi ona kol yardımı ile müdahele etme yeteneğine de sahiptir.

Bu fikri takip ederek, çalışmanın ilk aşamasında, bir mobil robot kol tüm alt sistemleri ile birlikte tasarlanıp imal edilmiştir. Bu alt sistemler, mekanik kol ve çift tekerlekli araç düzeneği ve bunların hareketliliğini sağlayacak motorlardan oluşan elektromekanik sistem, bu sistemin kontrolüne olanak sağlayacak gömülü işlemcili kontrolcülerden ve ölçme devrelerinden oluşan elektronik sistem, robotun çalışma alanını gözlemlemeye yarayan çift kameradan oluşan yapay görüş sistemi ve görüntü verilerinin alınıp işlendiği ve kontrol büyüklüklerinin hesaplandığı bilgisayardır.

Çalışmanın ikinci aşamasında, görüntü tabanlı kontrol metodları yardımı ile iki farklı özellikte ki mekanik sistem için iki ayrı kontrol yöntemi geliştirilmiştir. Hareketli platformun doğrusal hızı görüntü verileri işlenerek tahmin edilen hedef uzaklığı kullanılarak, kendi ekseni etrafında dönme hızı ise görüntü üzerinde hesaplanan hata tahmini kulalnılarak kontrol edilmiştir. Robot kol ise dinamik bak ve hareket et yöntemi olarak adlandırılabilecek, görüntü verileri kullanılarak hedefin bulunduğu noktanın tahmin edilmesi ve bu tahmine göre mafsal hareket referanslarının anlık olarak değiştirilmesi yöntemi ile kontrol edilmiştir.

Çalışmanın son aşamasında, bu iki sistem hareketi önce ardışık olarak tekrar ettirilerek mobil robot kontrolü için kullanılmış, daha sonra ise sistemin kinematik özellikleri göz önüne alınarak mobil robot kol hareketinin iki alt sistemin beraber çalıştırılarak kontolünü sağlayan bir yöntem geliştirilmiştir.

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

1.1 Motivation

The human capacity to control movements based on visual perceptions is called hand eye coordination. By analogy robotics systems build from cameras and manipulators are also assumed to have ability of hand eye coordination. Technology referring to this machine vision ability is visual servoing. Visual servoing implements perception action cycles as a close loop robot control, where feedback and set points are extracted from vision system data.

Technical hand eye coordination like its human counter part should ideally be flexible and effective while retaining the power of the machines. These dual goals point to the fact that such systems are at the crossroad of artificial intelligence, machine vision, robotics, control theory, mechanics and geometry in engineering science.

In very general terms, the overall aim is to produce a desired 3D motion of the mobile manipulator relative to a target by observing its position and motion in the images and exploiting these observations to calculate suitable controls for the actuators. For exploiting the interaction between perceptions and actions, it is necessary to model system and calibrates different systems to act in cascaded operations. Scientific foundation of this modeling is Euclidian Geometry because it provides ready to use concepts as interpretations for rigid motion, kinematics and camera projections. The modeling of robot and camera systems is a straight forward procedure if tools of Euclidian Geometry are implemented.

Robot control tasks are typically defined in the world coordinate frame of the task space. This environment can include the robot and the objects to be manipulated. The control strategy is formulated to map this world frame task definition into control sub goals for each actuator, if the described visual control scheme is considered in

particular. The task space is monitored via cameras. This leads representation of object pose in image plane and camera coordinate frame. Absolute world coordinate frame representation is omitted because the references for motions and even task space positioning errors can be represented using vision system output. However, it must be noted implementing camera coordinate frame based task space recovery does not change the fact that robotics system operates in world reference frame. The procedure to compromise robot motion, sensory inputs is calibration.

Using imaging models developed for cameras and priori known patterns, it is possible to fully parameterize the image point and world point relation. For binocular stereo applications which fully imitates human eye configuration, it is possible to find 3D coordinates of a monitored point with calibrated setup.

Implementation of binocular stereo rig applications in robotic systems, sensor data can be converted to task frame data and can be exploited in standard kinematics models. This setup reserves certain amount of autonomy to systems, in which addition of mobility to manipulator configuration can be justified.

Using a wheeled mobile platform and manipulator arm together, a mobile manipulator, robotic system can operate in environments that are either partially or completely unknown. Vision data enable the robot to navigate in these environments is well motivated because vision sensors can have wide field-of-view, can have millisecond sampling rates, and can be easily used for trajectory planning. Using two cameras provides 3D position recovery, which is a major advantage on single camera systems.

1.2 Scope

Prime interest of this thesis is mobile manipulators, in structural context. The first step of research focuses on structure of the mobile manipulators and deals with structural decomposition. After functions of each subsystem are defined, mechatronical design procedure is initiated. Modules in subsystems are designed

concerning overall operation of the robotic manipulator. Once experimental rig is constructed, focus of the thesis oriented to find a vision based control method for mobile manipulator. In this stage of study, system concerned as if it consists of two independent subsystems. Controllers for these two systems are designed separately. Resultant control subsystems are gathered via task sequencing.

A detailed formulation of the problems investigated is as follows:

Given a task of exploration and remote manipulation in unknown environment, what structural properties should a robotic system have? The answer to this question is evident for most robotics engineer, besides concerning the answer from cost efficiency point of view, leads more detailed synthesis of mobile manipulator systems. Mobile manipulator is concerned as a combination of three subsystems: electromechanical, electronics and computer subsystems. Then, these systems are considered independently but their functions are defined concerning to their contribution to overall systems.

Given a robot manipulator, what kind of a control law should be applied to fulfill mobile manipulator positioning tasks? Concerning two mechanical structures mobile platform and manipulator, what are the efficient vision based methodologies to control two systems? As it is demanded that they operate together, what is the correct task planning to coordinate their motion? And can the tracking performance of mobile manipulator be improved?

1.3 Organization of Manuscript

This thesis is organized in three sections, enclosed by an introduction and conclusion.

In the second chapter of the thesis, design and construction of mobile manipulator are explained. Three subsystems of the robotics system are handled one by one and design constraints and developed solutions are given.

The third, fourth and fifth chapters form the second section, in which theoretical fundamentals of the thesis is recapitulated and restated using the terminology and formalism adopted also in succeeding sections. Basically these fundamental concepts are dynamic analysis of robotic mechanism, mathematical modeling of vision system and notion of vision based control.

The third section of thesis starts with sixth chapter and it focuses development of control system. First, low level actuators and their controllers are concerned, calibration of variables and experimental tuning of controllers are represented. In chapter seven, vision based control of robot manipulator is explained. After explaining capabilities of mobile manipulator, vision based controller for mobile platform is designed. Considering poor motion characteristics of mobile platform, a prediction module is designed for delayed vision data. In the next step visually guided control of manipulator is succeeded with a modification on the manipulator actuator controllers. On the last stage of vision based control design, study seeks a method to control two subsystems together. As a first method sequential control of two systems are experimentally tested. Afterwards a novel methodology for cooperating action of the mobile manipulators are developed and experimentally verified.

1.4 State of Art

The following synopsis on the state of art aims to situate this work with respect to main approaches and the key papers in the related field of research. It moreover relates it to the previously published research which has inspired or influenced the research.

Three different functional areas are identified and addressed in this work. The first area is about autonomous robots in general, the second is referred to the vision based control strategies taking into account primarily and in the last one is detailed as the nonholonomic mobile manipulator control.

A mobile manipulator consists of an articulated arm mounted on a mobile platform. Since this mechanical arrangement combines the dexterity of the former with the workspace extension of the latter, it is clearly appealing for many applications (Khatib, Yokoi, Chang, Ruspini, Holmberg & Casal 1996). The locomotion of the platform is typically obtained through wheels. Therefore analysis and control of the mobile robots are parallel to developments in wheeled mobile robot research.

Wheeled Mobile Robot is defined as a wheeled vehicle which is capable of autonomous motion (without an external human driver) because it is equipped, for its motion, with motors that are driven by an embarked computer (Campion, Bastin & Andrea-Novel, 1996). We should add to this definition that the vehicle is also equipped with sensors to perceive about the environment (Jörg, 1995).

The mobile part of mobile manipulator studied in this research is a kind of a simple nonholonomic mechanical system. Many studies in nonholonomic control systems have been carried on in past decades (Bloch, Baillieul, Crouch & Marsden, 2003). But universal control methodology has not been developed even for a simple differentially driven mobile platform. These limitations can be overcome by either relaxing the constraints on desired pose, i.e. stabilizing to a point without a guarantee on orientation (Lambrinos, Moller, Labhart, Pfiefer & Wehner,.2000), using discontinuous control techniques, or by using time-varying control (Walsh, Tilbury, Sastry, Murray, & Laumond, 1994).

Nonholonomic property is seen in many mechanical and robotic systems, particularly those using velocity inputs. Smaller control space compared with configuration space causes conditional controllability of these systems (Laumond, 1998). So the feasible trajectory is limited. Kinematic model of parallel wheeled mobile robot fails to meet Brockett's necessary condition for feedback stabilization (Brockett 1983). This implies that no static feedback law for stabilization is available for such systems. This constraint cannot be eliminated from model equations, then the standard planning and control algorithms developed for under-constrained robotic

manipulators are difficult to apply. For wheeled vehicles, dead reckoning of internal sensors is not enough to achieve precise motion control because of slippage. Therefore, any usage of external sensors is indispensable.

Extending capabilities of robot systems through artificial perceptual actions is one of the major goals of the scientist and engineers in field of robotics. One major clue to follow to achieve this goal is designing vision systems. Vision systems are, in most general sense, imitates human eye. (Coombs & Brown, 1993) By analogy it provides very important means both for recognition tasks (Ekvall, Kragic & Hoffmann, 2005) and environment interaction for a robotic system. (Kragic & Christensen, 2001) Beside advanced image processing systems for recognition and data modeling for environment mapping, there exist researches which exploit visual data, extracted from images, to control actuators. Those systems are generalized in use for robotics systems where visual functionality improves the area of use of such systems.

Vision based control is a technique where the appearance of a target in the image or an estimation of the observed target position is used to control the position of the robot and to move it to a desired position in the 3D scene. More generally, vision based control is an appealing technique which enables the loop to be closed between sensing and action (Malis,Chaumette & Boudet, 1999).Details about vision based control systems can be found in (Hutchinson, Hager & Corke,1996), (Hashimoto, 2003).

Vision based systems are exploited in various schemes for mobile platform control (Ma, Koseck'a & Sastry, 1999),(Das, Fierro, Kumar, Ostrowski, Spletzer & Taylor, 2002) Common point in reported studies is that vision data is used to interpret pose of the manipulator in world reference frames. In this study, error dynamics (in SO(2)) of the mobile platform are measured through vision system. This leads direct interpretation of error posture, whereas in mentioned studies, current and reference postures are calculated first and then error posture is calculated.

Manipulator part of mobile manipulator does not subject to any kinematical constraint. Therefore design methodologies referring to inverse and forward representation of kinetics are exploited in control strategies. Detailed explanation of various control schemes can be found in standard robotics text books (Murray, Li & Sastry, 1994), (Fu, Gonzalez & Lee, 1987). In this thesis, applied control strategies are proportional integral and proportional with gravity compensation schemes. Developed gravity compensation law is distinguished from the standard application since it is defined for task specific redefinition of the workspace and implemented via embedded controller.

Control of the mobile manipulators is generally based on combined kinematical analysis of the both subsystems. (Hootsmans & Dubowsky, 1991). Developed methodologies for computing actuator commands for such systems allow them to follow desired end-effector and platform trajectories without violating the nonholonomic constraints. Based on mathematical manipulations on the system dynamics, model-based controllers are designed to eliminate tracking errors (Papadopoulos & Poulakakis, 2000) (De Luca, Oriolo & Robuffo Giordano, 2006).

Like any other configuration, vision based control systems are exploited to improve performance of mobile manipulators in unstructured environments. These controllers primarily focus on deriving jacobian matrix which relates perceived task space features to mobile manipulator control inputs (De Luca, Oriolo & Robuffo Giordano, 2007). In this study control of non holonomic mobile manipulator is done by task sequencing. Any positioning task is realized in two stages: Mobile platform moves till target is placed in manipulator workspace and in proper orientation that gripper (tracking) can hold, afterwards manipulator moves to hold target (handling).This distribution reduces the control problem in to two sub problems: control of the mobile platform and control of manipulator and controllers described above can be implemented independently.

CHAPTER TWO
ROBOTIC SYSTEM

2.1 Scope of Mechatronics Design

Although robots are familiar to many of people, the knowledge of its subsystems and internal dynamics are only of interest of technical community. Robots, technically, are end products of mechatronical design procedure. Therefore, structural decomposition of a robot is done by taking functionalities of subsystems into account.

McKerrow, in 1991 stated that a robot is a machine which can be programmed to do a variety of tasks in the same way a computer is an electronic circuit which can be programmed to do a variety of tasks. In context of this study, this definition highly appreciated by considering a robot as a stand-alone electromechanical system that performs physical and computational activities. These activities vary in a wide range in relation with application field of designed robot. In any case, reconsidering the adopted definition, a robotic system is composed of three synergic subsystems: Computer system, electronics system and mechanical system.

This thesis, when considered from robotic design point of view, focuses on design and production of an electromechanical system that has capability of performing controlled motions in existence of visual clues, autonomously. By term autonomy, it is mentioned that robot dynamically interacts with perceived environment. Such a configuration is intended to be commanded by simple supervisory inputs (Taylor& Kleeman, 2006).

In the following sections of this chapter, mechatronical design procedure is explained and sub procedures for three systems are defined. Implementation schemes for developed methodologies are represented.

2.2 Mechanical Subsystem

There exists a phenomenal confusion concerning form of the robots, how they look like. General belief is highly affected with science fictive image that they are human like. Nevertheless, humanoid projects are very complicated designs and result of a vast development in mechatronics and fusion of researches in areas from psychology to material science besides robotics.

In context of industrial robotics three mechanical structures are prominent: Serial manipulator arms, parallel manipulators and mobile robots. These systems offer different solutions to manipulation tasks in different physical ranges in force requirements and positioning units. First two systems are rigidly mounted on a base and work in predefined work cells and require precisely programmed actions. (McKerrow, 1991) On the other hand mobile robots are mostly equipped with wheels and can move in a workspace. Mobile robots are available for both indoor and outdoor applications. However due to kinematic contacts between wheel and floor they require external measurement system for motion log and control. Mobile robots generally perform investigation tasks (Chatila, Lacroix, Simeon & Herrb, 1995) and manipulation tasks by acting contact forces like pushing a box (Mason, Pai, Rus, Taylor, & Erdmann, 1999).

A more recent structure of robots is a result of demand on combining versatile manipulation and investigation task. A solution to this problem is mobile manipulators. Mobile manipulators are, structurally, mobile robots that carry robot manipulators. Mobile manipulation offers advantage of mobility provided by the platform and dexterity provided by the manipulator (Yamamoto & Yun, 1994). However, degrees of freedom of the platform also add to the redundancy of the system. (Bayle, Fourquet & Renaud, 2001). This robotic configuration is extensively used where robotic assistance is required. Integrating sensors with capability to perceive environmental changes, mobile manipulators become also widely available in applications other than factories or work cells.

By reinterpreting proceeding facts in robotics systems, this study focuses on a mobile manipulator as experimental setup. Proposed system is a differentially driven mobile platform carrying a two degrees of freedom planar manipulator and a gripper with roll and pitch motions (Figure 2.1).

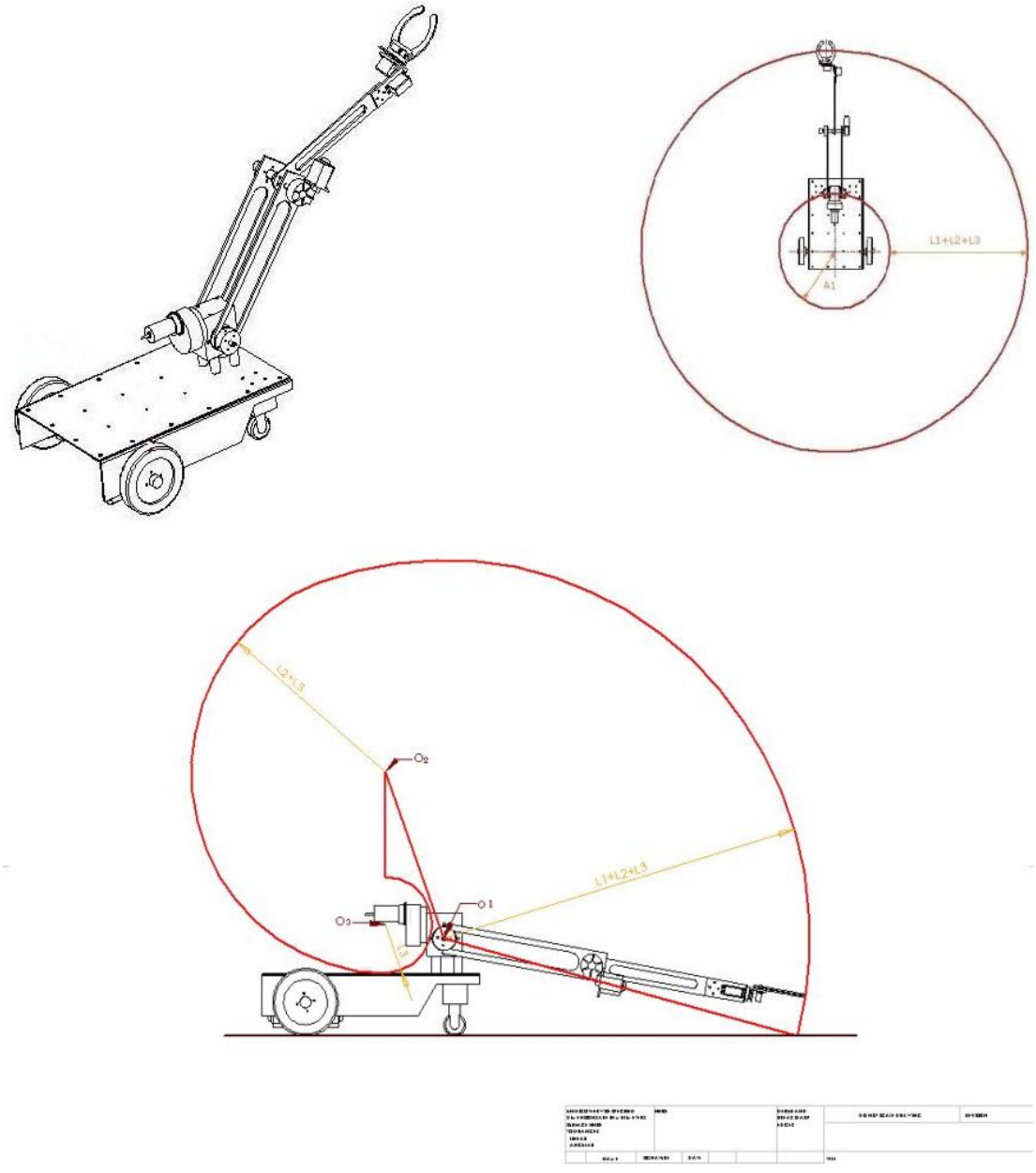


Figure 2.1 Mobile manipulator design with side and perspective views.

The main characteristic of the differential platform is that two wheels mounted on a single axis are independently powered and controlled, thus providing both drive

and maneuvering functions. A differentially driven system is simple and reliable and this wheel-based drive system is commonly used in mobile robots. To obtain a mobile platform, a sheet metal chassis is designed. Two driven and two free wheels are mounted on this chassis.

A robot manipulator is created from a sequence of links and joint combinations. The links are the rigid members connecting the joints (or axes). The axes are the movable components of the robot that enables relative motion between adjoining links. The mechanical joints used to construct the manipulator define type of relative motion: If two of the joints are linear, the relative motion is reciprocal translation and if they are rotary, the relative motion involves rotation.

The robot manipulator has two functional sub sections: Arm (or body) and wrist. The arm is used to position parts or tools within a workspace. The wrist is used to orient the parts or tools at the work location.

Arm part of robot manipulator is constructed with aluminum supporters. Joint axes are designed so that they can also be used for sensor mounting. Gripper (wrist in general context) is constructed with rigid flexi glass material and equipped only with motor mounting flanges. Manipulator links are also connected with helical springs so that they can tolerate some extra gravitational load on motion axes and compensate mechanical defects on joint motions.

2.3 Electromechanical Subsystem

Electronics by no means is a negligible part of robotics. It is always utilized in control schemes and drive units. Following this evident fact, an integrated circuit application is developed to fulfill tasks in actuator level.

Electromechanical subsystem is simply integration of actuator to mechanism and electronics implementation of low level control tasks. Task at actuator level is mainly to apply a control law to manipulate actuator output. Natural requirements of the

feedback control and motor drive technology defines design specifications, which are: Measurement of critical physical variables, implementation of control law and controlled motor motion.

Actuators for mobile platform and arm part of manipulator are permanent magnet DC motors. PMDC Motor is composed of two cascaded physical systems: Electrical part which generates current to generate magnetic coupling forces and mechanical part which generates motion on output axe. Electrical system is the armature circuit of the PMDC motor. It is a simple electrical system with an inductance and resistor and generates a current flow which is directly proportional to torque generated on motor axe. Motor axe is also under the affect of torque which is caused by load side of the system. This torques affect mechanical system which is a first order with effect of dynamic friction (damping effect) and inertia of rotary parts. Output is angular velocity in motor output shaft. Two significant parameters for PMDC (or any DC) Motor is armature current and angular velocity, which are to be monitored to predict motion dynamics (Figure 2.2).

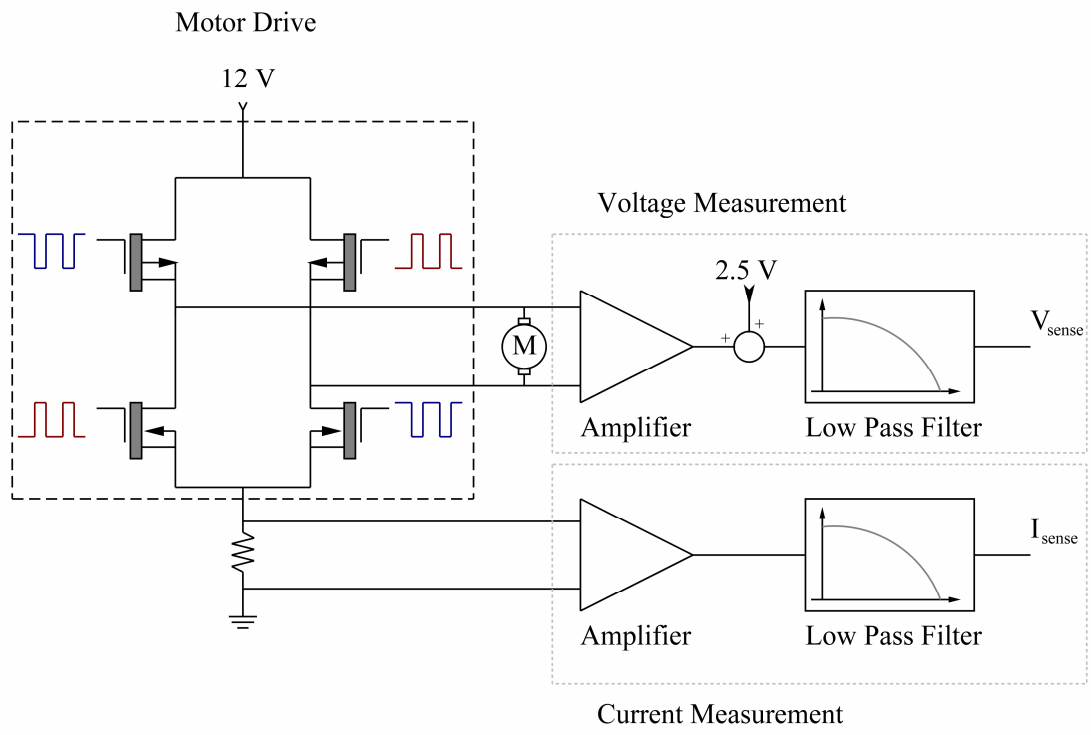


Figure 2.2 Schematic presentation of back electromotor force and armature current measurements

Motor current can easily be measured by measuring the voltage drop on a low value resistance in series to motor. Measured value is relatively small values since resistance is 0.01 ohm. This small measurement is first amplified by a gain 48 and then filtered with a low pass filter with edge frequency 162 Hz. Resultant voltage value is conveyed to PIC via its analogue input pins.

Methods for angular velocity measurement vary from encoders to tachometers. But in our case due to robot gearbox coupling, it is not possible to connect neither encoder nor tachometer implementation. Hence back electromotor force is used to monitor motor velocity. Typically a motor takes power in the form of voltage and current and converts the energy into mechanical energy in the form of rotation. A generator simply reverses this process. It takes mechanical energy and converts it into both electrical energy with a voltage and current. Most motors can be generators by just spinning the motor axe and looking for a voltage/current on the motor windings. When doing Back-EMF measurements for motion control, the fact that a motor can also be a generator is exploited. When current is passed through the armature of a DC motor, a torque is generated by magnetic reaction, and the armature revolves. The turning of the armature induces a voltage in the armature windings. This induced voltage is opposite in direction to the outside voltage applied to the armature, and hence is called back voltage or counter electromotive force. As the motor rotates at constant speed, the back voltage rises until it is almost equal to the applied voltage. The current is then small, and the speed of the motor will remain constant as long as the motor is not under load and is performing no mechanical work except that required to turn the armature.

The voltage drop on motor terminals is given with following mathematical expression:

$$V_a = -V_{emf} + R.i(t) + L\frac{di(t)}{dt} \tag{2.1}$$

In steady state:

$$V_{emf} = -(V_a - R.i(t)) \tag{2.2}$$

If we can measure armature voltage (voltage on motor terminals), back e.m.f. can be estimated by following given derivation. Armature voltage measurement setup is similar to current measurement setup but it differs in two practical issues. Amplifier gain is 1/6 to decrease voltage range from actuator values (12 Volts) to 5 Volts for PIC integration. A bias of 2.5 Volts is added to measured value to make it unipolar.

Control of the arm joint and wheel actuators is realized with an embedded control system. Two measurement circuits are implemented with operational amplifier ICs. A PIC is used as processing unit. Outputs of current and angular velocity circuits are connected to analogue inputs of PIC. Using 10 bit analogue controllers, current and b.e.m.f estimates are converted to an integer value in range from 0 to 1024. These values are placed in equation 2.2 that gives an estimate of motor angular velocity. In arm joint control circuits, there is also position feedback. Relative position of the link is measured by a potentiometer that is coupled to joint axes. That results another analogue voltage input for embedded control circuit.

PIC circuits have two different operating modes for implementing two different control modes: Angular velocity control mode and angular position control mode. Both modes are PI controllers. Proportional control part reduces the rise time, increases the overshoot of the system and reduces the steady-state error but do not exterminate the error. On the other hand, an integral control decreases the rise time, increases both the overshoot and the settling time, and but eliminates the steady-state error. Control algorithm for PI controller is given in continuous and discrete form as follows. Although discrete form is not direct conversion, it simplifies calculations for constant sample rate implementations.

$$u(t) = K_p e(t) + K_i \int_0^{\tau} e(\tau) d\tau \tag{2.3}$$

$$u(k) = K_p e(k) + K_i \sum_{i=1}^{k} e(i) \tag{2.4}$$

This algorithm is applied as angular velocity control law for cart motors and position control law for manipulator controllers in given schemes.

When computing above equations PIC controller uses decimal numbers whereas parameters of controllers are to be more precise than integer increment. To overcome this difficulty and to manage to use fractions of one in implementation, fixed-point number representation is adopted. Fixed point representation of a real is a number that has a fixed number of digits after the radix point. Conversion is by multiplying or dividing the real number with fixed value like 2^n. Fixed-point arithmetic provides improved performance for the application.

Result of the calculations is written to PWM register of PIC. Embedded PWM circuit works in units of cycles, each of which consists of 1024 time slots. Output values defines the duty cycle of the PWM signal where full cycle is 1024 corresponding to +12 volts and 0 corresponds to -12 volts. PWM output of the circuit is input for two cascaded circuits. The first part of the circuits does logic computations for the succeeding circuits which is a transistor bridge for driving motor. Transistor bridges are conventional configurations for driving motors with digital commutation.

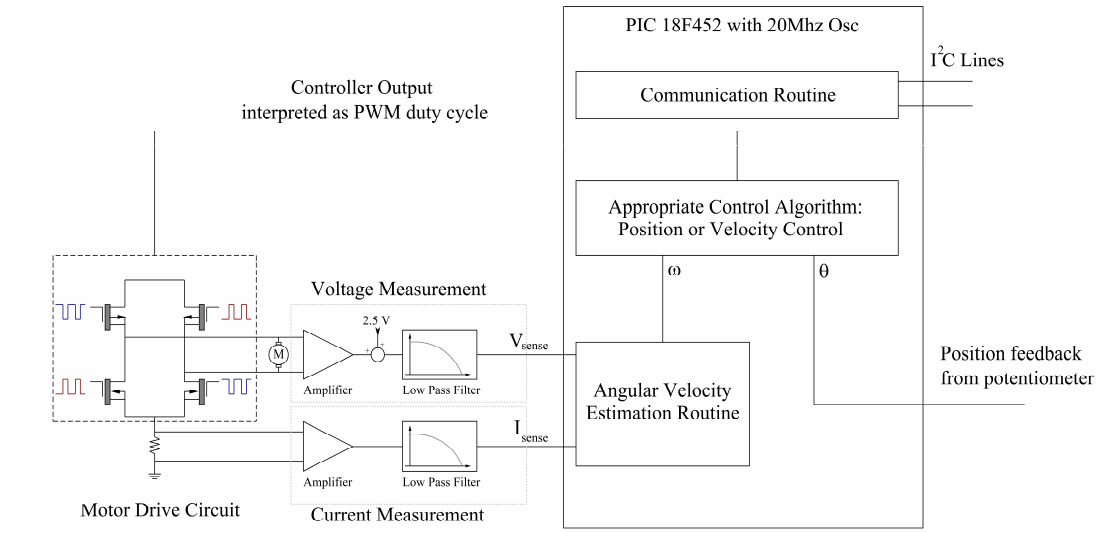


Figure 2.3 Schematic presentation of node controller

The controller circuit described so far is a servo controller node because it operates independently to control one joint actuator. To control mobile platform and manipulator arm motion, four controllers operate simultaneously. It can be easily deduced that this simultaneous operation is meaningful if synchronization exist between controllers. This commanding task for controlling operation of nodes is provided with a main controller (Figure 2.3).

Main controller is another embedded controller circuit with a PIC. It acts as a status logger for controllers and transfers commanding inputs from supervisory controller (computer) to servo control nodes. It may be seen as a communication buffer that synchronizes computer subsystem and electromechanical subsystem. PIC circuits communicate with each other through I^2C protocol. I^2C protocol provides good support for communication with peripheral devices that are accessed intermittently, while being extremely modest in its hardware resource needs. It is a simple, low-bandwidth, short-distance protocol. I^2C is easy to use to link multiple devices together since it has a built-in addressing scheme. Details on operation of I^2C can be found in Kalinsky D. 2001. Main controller communicates with computer through serial communication protocol at 38400 Baud rate.

Another variation of the embedded controller circuit is position servo control circuits for control of gripper. Gripper operates in an open loop manner which means that system provides only commanding input for motor and does not take any feedback because two gripper degrees of freedom actuated with hobby servo motors which have inset positional controllers. Position commands for servo motors are pulses with different widths form 1 to 2 ms and followed by a pause of 40 ms.

Implementation of the circuits on mechanical system completes actuator level requirements of overall robotic system. But it must be noted that system is still open for inputs in actuator space not in task space. Any robotics mechanism needs an interaction model for its task space so that it can autonomously generate its way points to follow to reach the target. Therefore a supervisory computer subsystem is also integrated into mechatronics design procedure.

2.4 Computer Subsystem

A computer is with standard peripheral devices can be exploited in control tasks with standard control equipment if a serial or parallel interface is available in control devices. A bunch of applications requires some complementary devices to extend processing capabilities of the computer. On the other hand computational power and wide compatibility of operating systems make computers important parts of advanced control algorithms.

Most of the advanced control algorithms require extrinsic perceptual inputs to develop control laws or planning motions in dynamically changing environments. Vision systems are widely used because of their ability to provide wide range of spatial data in one sampling instance (Berry, Martinet & Gallice, 1997). In context of this study a stereo vision system is also integrated to track a target to provide a positional feedback in task space of the robotic system.

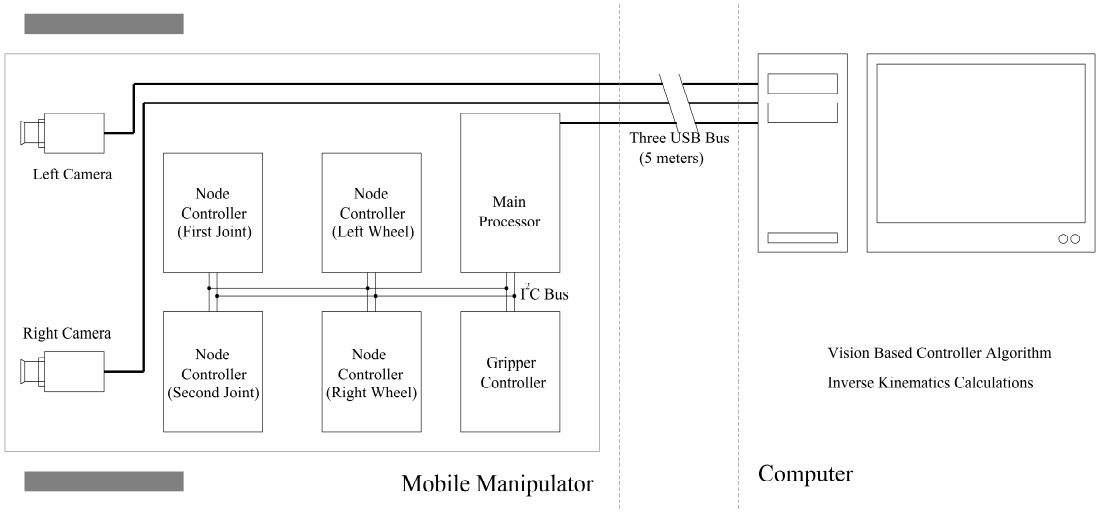


Figure 2.4 Schematic description of controller hierarchy: Two cameras, embedded controllers, computer and communication

In this study, a standard PC is utilized as a supervisory controller. Its task is defined as supervisory because it performs high level tasks for robot operation, image processing to commute robot task planner and manipulates the low level controller through manipulating gain matrices or actuator references instantaneously.

As mentioned in previous section, robot mechanism is connected to computer via serial bus operating at Baud Rate 38400 whereas electronics on robot operates at with I^2C protocol. As vision inputs, two cameras, mounted on mobile platform, are connected to computer via USB 2.0 ports (Figure 2.4).

Data from whole subsystem is processed in a VC++ program, namely Mobile Manipulator Control Program (MOMAC). This program executes two primal tasks: commanding and monitoring the electromechanical system and exploiting visual data to obtain necessary information from environment. Operation of the program is not sequential as expected in many systems. Multithread capability of operating system is exploited to enhance real time performance as a servo system.

Vision algorithm is exploited to obtain target position relative to mobile platform and generate an error representation for control of mobile platform. Constructive fundamental of the image processing system is Open CV libraries and implementation routine is image acquisition following by processing of each image frame and extraction of set of features from previous step to reconstruct an estimation of 3D point.

Image acquisition process is simply initializing data structures to store instant data and generating virtual handles to control cameras. Operating system automatically sets up memory locations and member functions to manipulate virtual objects with a set of functions, defined in OpenCV binary files, once it is initiated.

Start point of the processing is that there exists a planar shape in every image related to target object via camera projection (Trucco & Verri, 1998). A wide range of algorithms are available for finding meaningful data from statistical procedures to geometric representations or color properties. Methods also have different application areas according to their inherent data representations. Statistical methods enlighten urban sites in satellite images, geometric methods enables precise 3D recovery in augmented reality applications and image level properties are used in fast

routines where statistically rough representation of data cause relative tolerable errors (Figure 2.5).

Image processing algorithms developed affect the real time performance of the system directly. Hence, implemented methods are intended to be simple because using complex methodology would require computational optimization problems, which are out of the scope of this study. On the other hand developed system can be used as a test bed for more detailed vision algorithms concerning unconstrained visual inputs, which may be developed further.

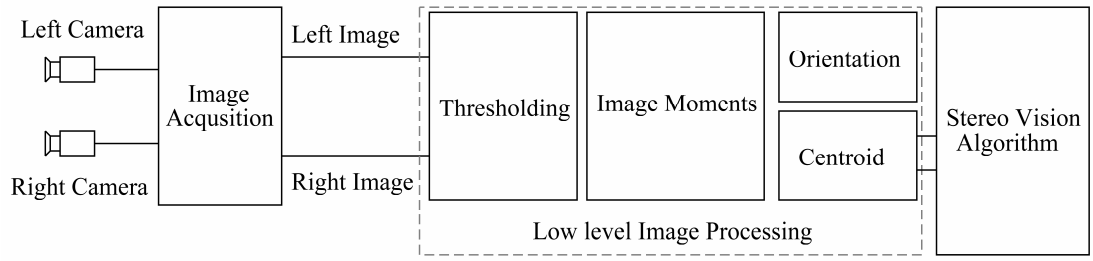


Figure 2.5 Image processing procedure layout

Acquired images are compared with a predefined threshold value pixel by pixel. Highly illuminated pixels are tagged as object points since LEDs are used as markers of target point. Segmentation process retrieves a projection of target on images from left and right cameras. Image moments up to second order are calculated for two segmented images. Calculated image moments are used to find centroid of the region of interests in both images.

Centroids of left and right images are used to calculate 3D position of the target point in the left camera coordinate system. It is known that for a calibrated stereo rig, it is possible to calculate 3D position of a point by using its projection on left and right images. But it should be noted that result of the stereo vision calculations is in the left camera plane not in task frame of the robotic system. Object orientation found in left image is also utilized for roll motion control of gripper.

Result of the stereo vision algorithms are converted to robot coordinates via homogenous transformation. This transformation simply defines stereo setup configuration on robotic system mathematically.

Once sensory system generates output, controller algorithms are initiated. Because there are three different electromechanical systems exist, three different control algorithms are developed for mobile platform, manipulator arm and gripper.

Control of mobile platform depends up on two parameters. One of these parameters is distance to target because the target must be placed in workspace of the planar manipulator. This distance can be calculated using Euclidian norm of planar components of output vector from stereo vision algorithm represented in robot coordinate frame. This value used to define control action concerning linear velocity of the cart. The other parameter for control of mobile platform is its orientation. To be able to manipulate target with planar manipulator, which it carries, target and gripper of the manipulator must be aligned. Proper gripper pose corresponds to a line in image plane. So the task can be simplified as driving projection of target point on to a horizontal line in an image. Calculations in image frame generate a control action proportional to angular velocity of the mobile platform

Manipulator arm references are calculated with the help of x and z coordinates of the extracted 3D points represented in robot coordinate frame if target is placed in its workspace. Implementing inverse kinematics algorithm generates angular position references. Gripper references are directly calculated from image features. Roll motion reference is found by image orientation. Pitch motion reference is used to hold target parallel or vertical to floor. In both cases reference is calculated by using manipulator arm angles.

The supervisory controller algorithm updates references to deliver robotics system. Controller thread in MOMAC receives results of the three controller algorithms, which are in configuration space, and convert them to actuator inputs using provided calibration data. Afterwards, all updated references are sent to related

controllers. Control thread sets references and reads time dependent control parameters of controllers periodically.

In this chapter, design of the electromechanical system is explained in a constructive point of view. The components of the overall system are described with explanations about the reasons why they are designed and constructed as they are.

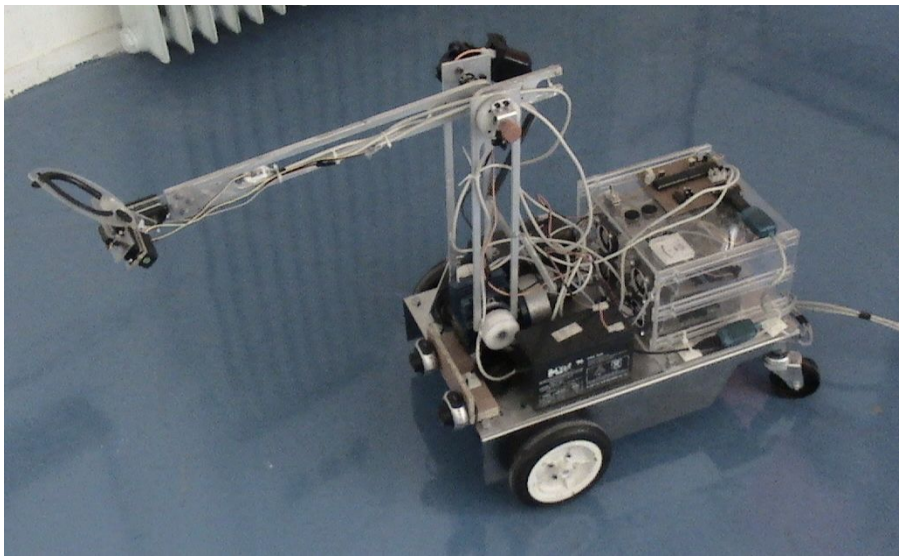


Figure 2.6 Mobile manipulator

The final product of design procedure is shown in figure 2.6. Constructed mechanical system consists of a mobile platform with two driven wheels and two casters moving on horizontal plane and a manipulator and gripper combination working on vertical plane. Manipulator is mounted on the front side of the mobile platform. Stereo setup is also placed mobile platform and beside manipulator base. Fields of view of these cameras are forward motion space of mobile platform and workspace of manipulator. Electronic circuits are placed on the back side of the card between accumulators. Desktop computer is connected to card with five meters USB cables (Figure 2.6).

More detailed explanations about dynamical models of the mechanism and control process are given in following chapters.

CHAPTER THREE
MATHEMATICAL MODEL OF MOBILE MANIPULATOR

3.1 Structural Decomposition of Mobile Manipulator

The experimental robotic system consists of three parts: Differentially driven mobile platform, manipulator arm and gripper. An important taxonomy methodology for mechanisms is according to constraints on their motion. Mobile platform posses non integrable constraints and referred as nonholonomic system whereas manipulator and gripper have no constraints on their motions. (Bloch, Baillieul, Crouch & Marsden, 2003).

In following sections, three systems are modeled concerning their kinematical category. Both kinematical and kinetic analysis are carried out. Using dynamical equations with time dependent variables, differential equations of the joint motions are found. Different dynamical characteristics of subsystems are taken into account to design appropriate controllers.

3.2 Coordinate Frames

Coordinate frames and vectors are primal tools for dynamical analysis. Therefore different coordinate frames in robotic system and their internal relations are clarified to avoid any ambiguity.

World coordinate frame (F_0) is the basis of the robot manipulator in absolute coordinates. Its origin is assumed to be projection of the mid point of the drive axis of mobile platform on to floor. Exact coordinates and pose of the system can be calculated by recalculating forward dynamic equations (Tsai, 1999) with measured joint and wheel motions.

Mobile platform motion is represented with a coordinate frame (F_v), which is attached to mid point of the axis of driven wheels. Manipulator coordinate frame

(F_m) is used both for task space interpretation for manipulator motion and relative motion reference for mobile platform. Manipulator coordinate frame origin coincides with the first joint axis. Wrist coordinate frame (F_w) defines gripper orientation and it is placed at the distal joint of the second link of robot arm (Figure 3.1).

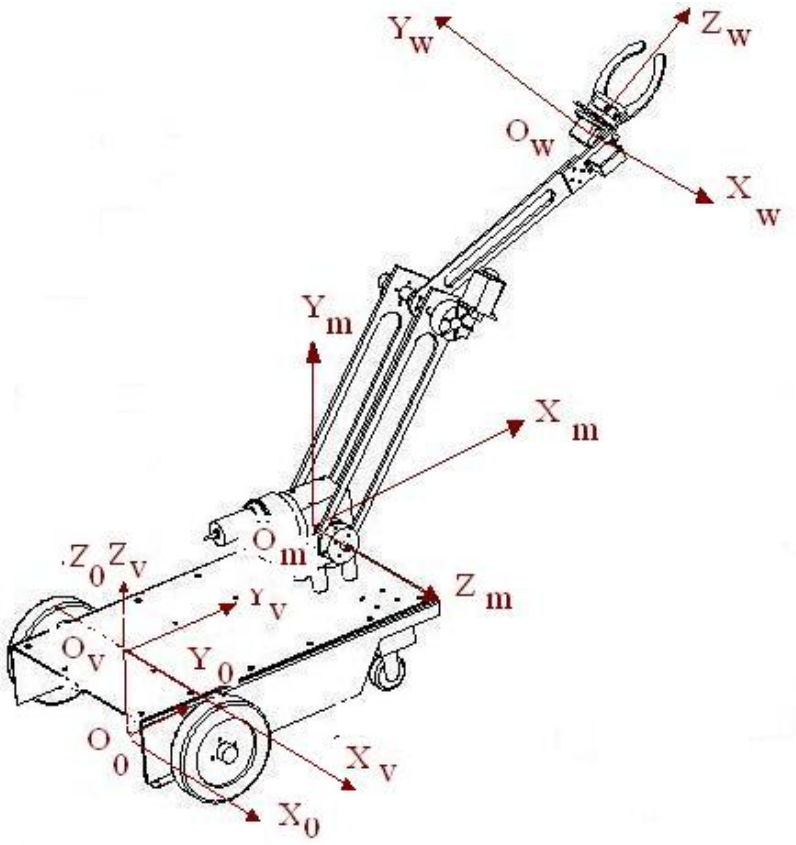


Figure 3.1 Mobile manipulator and assigned coordinate frames

Vectors are transformed from one coordinate system to other with homogenous transformation matrices. Matrix $({}^{m}H_{w})_{4x4}$ defines relative pose of wrist reference frame in manipulator coordinate frame. It also converts a vector $({}^{w}r)_{4x1}$ in wrist coordinate frame to a vector $({}^{m}r)_{4x1}$ in manipulator reference frame.

3.3 Mobile Platform Motion

Main characteristic of the cart is that two wheels mounted on a single axis are independently powered and controlled, thus providing both drive and maneuvering

functions. A differentially steered drive is simple and reliable and this wheel-based drive system is commonly used in robots (Balkcom & Mason, 2004).

Similarity between the differentially steered drive system and a wheelchair provides us an intuitive grasp of the way it behaves. If both drive wheels turn in tandem, the robot moves in a straight line. If one wheel rotates faster than the other, the robot follows a curved path inward toward the slower wheel. If the wheels turn at equal speed, but in opposite directions, the robot pivots. Thus, steering the robot is just a matter of varying the speeds of the drive wheels.

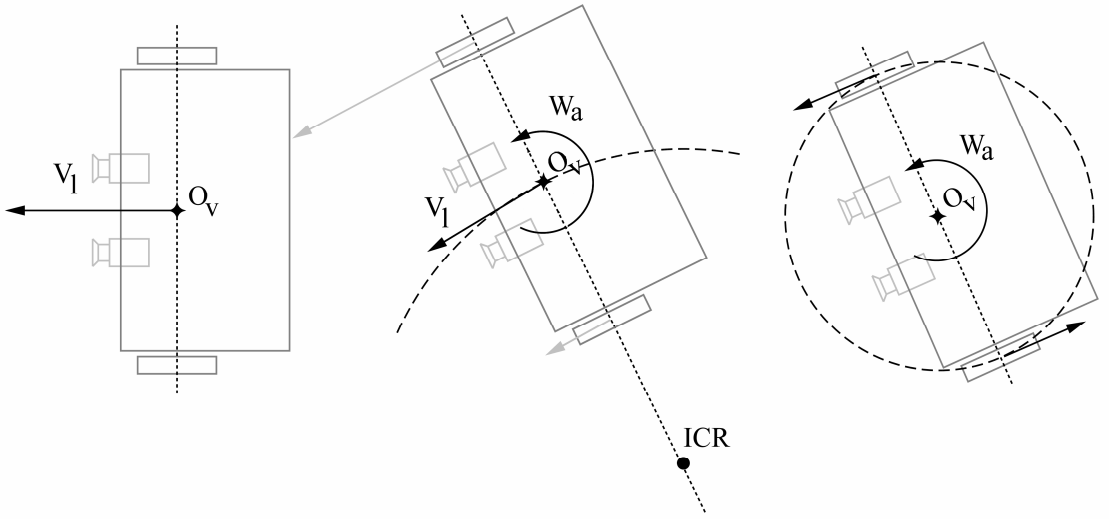


Figure 3.2 Schematic representation of differentially driven mobile platform motion drive motion linear, rotation around center of drive axis and following a path

At any instant, the x and y coordinates of the robot's center point (mid point of the driven wheel axis) are changing based on its speed and orientation. It is treated that orientation as an angle θ measured in radians, counter-clockwise from the x-axis. Recalling Cartesian decomposition of position vector clarifies that the vector giving the direction of forward motion for the robot will be simply inform of a multiple of $\cos\theta$ and $\sin\theta$. The x and y coordinates for the robot's center point will change depending on the speed of its motion along that vector (Figure 3.2).

To continue kinematical modeling a distinction must be enlightened that is the difference between global and local coordinate systems. The desired pose of the mobile platform is expressed in an inertial coordinate system (F_0) and another coordinate frame is attached (F_v) to middle of the wheel axis of mobile platform. This coordinate frame is exploited for defining actual pose of the mobile platform.

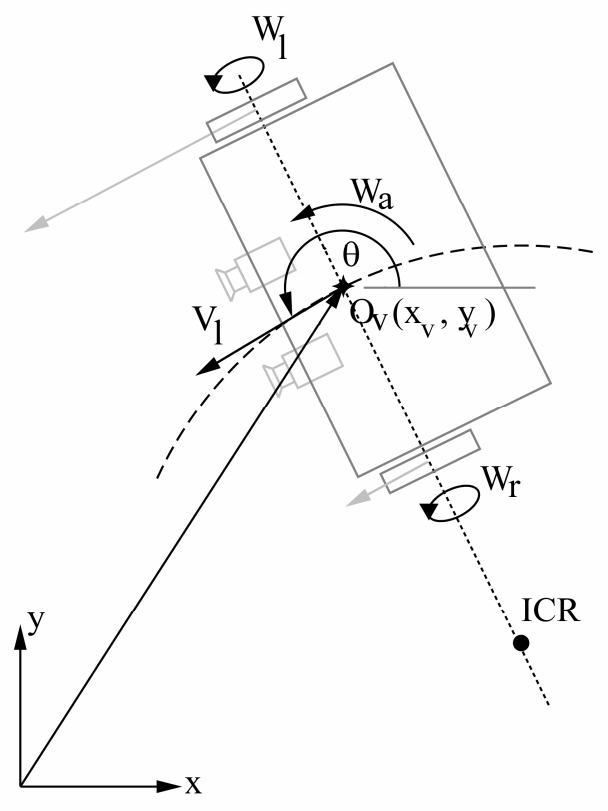


Figure 3.3 Parameters related to mobile platform motion

Location of any point in F_v, with respect to a world coordinate frame F_0, can be described by a matrix, denoted here as ${}^{0}H_{v}$. This type of matrix is called a homogeneous transformation matrix and matrices of this type are formed in a specific form so that they include rotation around Z_0 and translation $[x_v\ y_v]^T$ in F_w necessary to describe F_v in F_w (Figure 3.3).

$$ {}^{0}H_{V} = \begin{bmatrix} \cos\theta & -\sin\theta & x_v \\ \sin\theta & \cos\theta & y_v \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1} $$

Intuitively, columns of the matrix can be represented as three vectors:

$$ {}^{0}H_{V} = \begin{bmatrix} a_x & n_x & d_x \\ a_y & n_y & d_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2} $$

where

$\vec{a}$ represents the *approach* vector

$\vec{n}$ represents the *normal* vector

$\vec{d}$ represents the *distance* vector

It is evident that matrix ${}^{0}H_{v}$ is nonsingular and has inverse in form:

$$ {}^{v}H_{0} = \begin{bmatrix} cos\,\theta & sin\,\theta & -x_v\,cos\,\theta - y_v\,sin\,\theta \\ -sin\,\theta & cos\,\theta & -y_v\,cos\,\theta + x_v\,sin\,\theta \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3} $$

3.3.1 Velocity Kinematics of Mobile Platform

The kinematic equations governing motion of the differential drive are given in F_v as:

$$ -x'\sin\theta + y'\cos\theta = 0 \tag{3.4} $$

$$ x'\,cos\,\theta + y'\,sin\,\theta + L\theta' = r_w w_R \tag{3.5} $$

$$ x'\,cos\,\theta + y'\,sin\,\theta - L\theta' = r_w w_L \tag{3.6} $$

$$ \begin{bmatrix} -sin\,\theta & cos\,\theta & 0 \\ \frac{1}{r_w}cos\,\theta & \frac{1}{r_w}sin\,\theta & \frac{L}{r_w} \\ \frac{1}{r_w}cos\,\theta & \frac{1}{r_w}sin\,\theta & \frac{L}{r_w} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} 0 \\ w_R \\ w_L \end{bmatrix} \tag{3.7} $$

Matrix equation in given form represents inverse velocity kinematics for cart. Forward kinematics is solution of the given system and in this particular case (nonsingular matrix) it can be obtained using matrix inversion.

$$\begin{bmatrix} -\sin\theta & \frac{r_w}{2}\cos\theta & \frac{r_w}{2}\cos\theta \\ \cos\theta & \frac{r_w}{2}\sin\theta & \frac{r_w}{2}\sin\theta \\ 0 & \frac{r_w}{2L} & \frac{r_w}{2L} \end{bmatrix} \begin{bmatrix} 0 \\ w_R \\ w_L \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} \tag{3.8}$$

An important observation is that any point in the $[x,y,\theta]^T$ space can be reached by controlling the inputs $[w_1\ w_2]^T$. However, trajectories, in the space requiring the robot to move in a direction along the wheel axis, can not be followed by the vehicle due to kinematic contact, referred as rolling without slipping. The constraint, specifying the no slip condition, is written in the form:

$$x'\sin\theta - y'\cos\theta = 0 \tag{3.9}$$

This explicit equation of velocities is not integrable and systems that have a kinematic constraint which is an explicit function of velocity are called nonholonomic systems. Another characteristic of nonholonomic system is that the number control variables are less than the number of output variables.

3.3.2 Dead Reckoning Algorithm to Estimate Position

To develop a forward kinematic equation for the motion of a mobile platform, the robot is considered as rigid body. The middle of the wheel axis is select as central point and it is reference for the motion. All motion in frame F_v is treated relative to the central point O_v. Because the wheels are mounted perpendicular to the axle, its motion within the frame of reference follows a circular arc with a diameter corresponding to the length of the axle 2L.

Based on these observations, a differential equation describing the change in orientation as with respect to time of time can be derived. The definition of an angle given in radians is the length of a circular arc divided by the radius of that circle. The relative velocity of the right wheel gives us that length of arc per unit time. The

length from the wheel to the center point gives us the radius. Combining these facts, it is found that:

$$\theta' = \frac{(v_r - v_l)}{L} \tag{3.10}$$

Integrating equation (3.10) and taking the initial orientation of the robot as θ_0 , we find a function for calculating the robot's orientation as a function of wheel velocity and time.

$$\theta = \frac{(v_r - v_l)t}{L} + \theta_0 \tag{3.11}$$

As noted above, this change in orientation also applies to the absolute frame of reference. Shifting our attention back to the absolute frame, we recall from (3.7) above that the robot's overall motion depends on the velocity of its center point. That velocity is simply the average of that for the two wheels. If this fact is combined with what is obtained about orientation as a function of time, and following differential equations is found:

$$x' = \frac{(v_r + v_l)}{2}\cos\theta \tag{3.12}$$

$$y' = \frac{(v_r + v_l)}{2}\sin\theta \tag{3.13}$$

Integrating and applying the initial position of the robot $x(0)=x_0$ and $y(0)=y_0$ gives explicit position equations in coordinates of mobile robot.

$$x = x_0 + \frac{L(v_r + v_l)}{2(v_r - v_l)}(\sin(\frac{(v_r - v_l)t}{L} + \theta_0) - \sin\theta_0) \tag{3.14}$$

$$y = y_0 + \frac{L(v_r + v_l)}{2(v_r - v_l)}(\cos(\frac{(v_r - v_l)t}{L} + \theta_0) - \cos\theta_0) \tag{3.15}$$

Once above equations are found, dead reckoning is performable. Term "dead reckon" means to estimate position without external references. Position is dead reckoned using knowledge about a vehicle's course and speed over a period of time. In robotics, the necessary information is often obtained by measuring wheel revolutions. Encoders generally provide an estimate of displacements for the right and left wheels, respectively. In this study we adopt visual feedback instead of dead reckoning odometry algorithm. Since it is advantageous in two ways:It provides a position estimate without referring kinematics and real time position recovery is easier and does not carry integral error.

3.3.3 Mobile Platform Dynamics

A large class of mechanical nonholonomic systems is described by the following form of dynamic equations based on Euler Lagrange formulation:

$$M(q)q'' + C(q,q')q' + G(q) = B(q)\tau + J^T(q)\lambda \tag{3.16}$$

where

$J^T(q)$ is nonholonomic constraint,

$M(q)$ is a symmetric positive definite mass-inertia nxn matrix,

$C(q,q')$ presents the n vector of centripetal and Coriolis torques,

$G(q)$ is the n vector of gravitational torques,

$B(q)$ is the rxn input transformation matrix (r<n)

and q is the n dimensional vector of configuration variables and τ is r dimensional vector of inputs and λ the Lagrange multipliers of constrained forces. (Murray, Li& Sastry, 1994).

A simple structure of differential drive mobile robot is shown in figure 3.1. Two independent analogous DC motors are the actuators of left and right wheels, while two free wheel casters are used to keep the platform stable.

Pose vector of robot in the surface is defined as $=[x\ y\ \theta]^T$, x and y are the coordinates of point O_v; center of axis of wheels, and θ is the orientation angle of robot in the inertial frame. The dynamic equations of mobile robot can be written according to Equation (3.16), considering the fact that centripetal and Coriolis torques and gravitational torques are zero.

$$\begin{bmatrix} m_v & 0 & 0 \\ 0 & m_v & 0 \\ 0 & 0 & I_v \end{bmatrix} \begin{bmatrix} x'' \\ y'' \\ \theta'' \end{bmatrix} = \frac{1}{r_w} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & -L \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} + \begin{bmatrix} \sin\theta \\ -\cos\theta \\ 0 \end{bmatrix} \lambda \tag{3.17}$$

Where τ_l and τ_r are the torques of the left and right motors, m_v and I_v presents mass and inertia of mobile platform, r_w is radius of wheels and 2L is the distance of rear wheels

Assuming

$$\begin{bmatrix} \tau_l \\ \tau_a \end{bmatrix} = \frac{1}{r_w} \begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{3.18}$$

And rearranging equation 3.17:

$$x'' = \frac{\tau_l}{m_v}\cos\theta + \frac{\lambda}{m_v}\sin\theta \tag{3.19}$$

$$y'' = \frac{\tau_l}{m_v}\sin\theta - \frac{\lambda}{m_v}\cos\theta \tag{3.20}$$

$$\theta'' = \frac{\tau_a}{I_v} \tag{3.21}$$

Where τ_l and τ_a are linear and angular torques respectively. They are related to linear and angular accelerations of the mobile platform with direct interpretation of Newton's second law.

$$v' = \frac{\tau_l}{m_v} \tag{3.22}$$

$$w' = \frac{\tau_a}{I_v} \tag{3.23}$$

Where v and w are:

$$x' = v\cos\theta \tag{3.24}$$

$$y' = v\sin\theta \tag{3.25}$$

$$\theta' = w \tag{3.26}$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{3.27}$$

The last equation relates changes in the tasks space of the robot with linear and angular velocities of the mobile platform.

3.4 Planar Manipulator

In this section, we proceed with derivations for manipulator arm and assign a fixed coordinate frame to each link of the manipulator. Forward kinematics is used to describe the static position and orientation of the manipulator linkages and is the transformation from joint space to Cartesian space. This transformation depends on the configuration of the robot (i.e., link lengths, joint positions, type of each joint, etc.). Therefore we intend to pass a description of positions and orientations through each link of the kinematic chain, and to obtain a computational representation, depending on the joint position, to find a description in the base frame F_m (attached to mobile platform). Denavit Hartenberg Convention introduces a technique how to set up transformations between frames of neighboring links.

3.4.1 Denavit-Hartenberg Parameters

Robot manipulators are sequences of links articulated at joints. To analyze the motion of robot manipulators, reference frames are attached to each link starting at frame F_o, attached to the fixed link, all the way to frame F_n, attached to the robot end-effector (assuming the robot has n joints). The process followed in assigning frames to links is based on the Denavit-Hartenberg parameters.

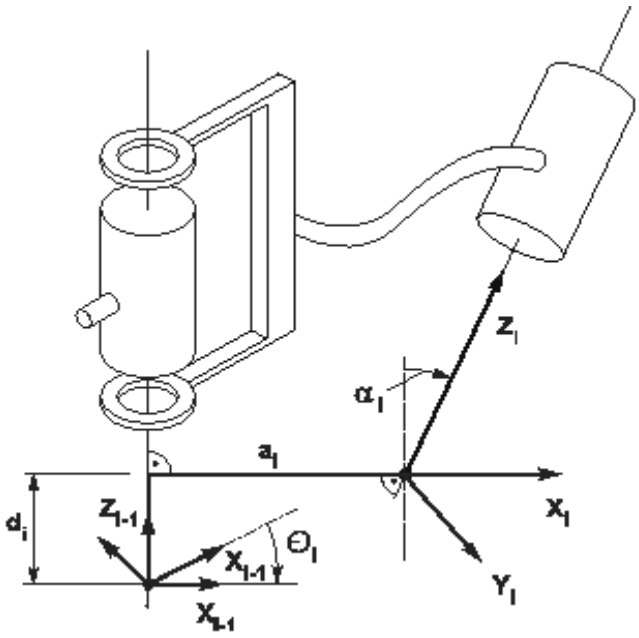


Figure 3.4. General robot link and Denavit Hartenberg parameters

Given two consecutive link frames on a robot manipulator, frames F_{i-1} and F_i, frame F_i will be uniquely determined from frame F_{i-1} by use of the parameters d_i, a_i, α_i, and θ_i illustrated in Figure 3.4.

- The z vector of any link frame is always on a joint axis. The only exception to this rule is the end-effector (tool) of the robot which does not have a joint axis.
- d_i is the algebraic distance along axis z_{i-1} to the point where the common perpendicular intersects axis z_{i-1}.
- a_i is the length of the common perpendicular to axes z_{i-1} and z_i.
- θ_i is the angle about z_{i-1} that the common perpendicular makes with vector x_{i-1}.

- α_i is the angle, about x_i, that vector z_i makes with vector z_{i-1}.

The rules given here simplify the kinematic modelling and analysis of robot manipulators:

- The parameter d_i is algebraic and may be negative. It is constant if joint i is revolute and variable when joint i is translational.
- The parameter a_i is always constant and positive.
- α_i is always chosen positive with the smallest possible magnitude.
- θ_i is variable when joint i is revolute, and constant when joint i is translational. When joint i is translational, θ_i is constant and determined by the structure of the robot.
- d_i is variable when joint i is translational, and constant when joint i is revolute.

Starting from one frame to the next can be found by following the four steps. From the origin of frame i-1, move a distance d on the z_A axis. Note that d can be positive or negative. Determine the direction of x_B by rotating vector x_A by an angle θ about z_A. Move a distance a in the direction of vector x_B. The position reached is the origin of Frame B. At this point vector x_B is determined as well. Rotate the vector z_A about x_B by an angle α_i to determine the unit vector z_B. Those refer to transformations (3.28) that constructs general transformation matrix for each link (3.29).

$$A_i = Rot(z,\theta)Trans(z,d)Trans(x,a)Rot(x,\alpha) \tag{3.28}$$

$$\begin{Bmatrix} \cos\theta & -\cos\alpha\sin\theta & \sin\alpha\sin\theta & a\cos\theta \\ \sin\theta & \cos\alpha\cos\theta & -\sin\alpha\cos\theta & a\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{3.29}$$

Through procedural implementation of the rules, a mathematical description for robot manipulator is obtained as a table of DH-parameters.

On our design, second, third and fourth joint axes are parallel to one another. The first joint axis points up vertically and the fifth joint axis intersects the fourth perpendicularly. The Denavit-Hartenberg parameters for our robot are shown in Table 3.1. The parameters for the links are constants with the exception of θ's,

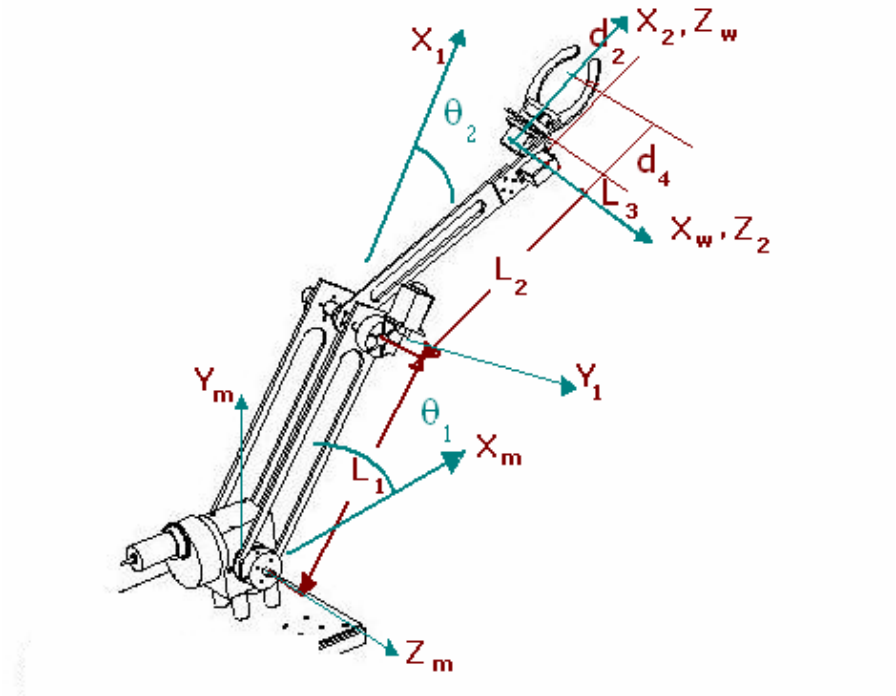


Figure 3.5 Robot manipulator and reference frame assignments

Table 3.1 Symbolic DH parameters for the robot

Link Angle	Link Offset	Link Length	Link Twist
θ_1	0	l_1	0
θ_2	0	l_2	d_2
θ_3	$\pi/2$	l_3	0
θ_4	0	0	d_4

Transformation describing overall robot manipulator is found via sequential multiplication of link transformation matrices:

$$A_0^5 = A_1 A_2 A_3 A_4 \tag{3.30}$$

3.4.2 Inverse Kinematics

Inverse kinematics solves for the joint angles given the desired position and orientation in Cartesian space. This is a more difficult problem than forward kinematics. The complexity of inverse kinematics can be described as follows, given a 4x4 homogeneous transformation, which gives the required position and orientation. The homogeneous transformation matrix results in 12 nonlinear equations.

$$A = \begin{Bmatrix} R_{3x3} & d_{3x1} \\ 0 & 1 \end{Bmatrix} \tag{3.31}$$

For example, to find the corresponding joint variables (θ_1, θ_2, θ_3, θ_4) for our manipulator shown in figure 3.4, loop closure matrix, which corresponds to recommendations about end effector position and orientation, is used as right hand side of the equation. That provides ease to solution in a sense that the pose of the manipulator is preliminary generated via sensor system or man machine interface.

$$A = \begin{bmatrix} u_x & v_x & w_x & q_x \\ u_y & v_y & w_y & q_y \\ u_z & v_z & w_z & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.32}$$

The homogeneous transformation matrix eliminates the possibility of finding the solution by solving those 12 simultaneous set of nonlinear trigonometric equations. These equations are much too difficult to solve directly in closed form and therefore we need to develop efficient techniques that solves for the particular kinematics structure of the manipulator. To solve the inverse kinematics problem, closed form solution of the equations or a numerical solution can be used (Manocha & Canny, 1994).Closed form solution is preferable because in many applications where the manipulator supports or is to be supported by a sensory system, the results need to be supplied rapidly (in real-time). Since inverse kinematics can result in a range of

solutions rather than a unique one, finding a closed form solution will make it easy to implement the fastest possible sensory tracking algorithm (Buttazzo, Allotta & Fanizza, 1994).

One aim of this work is to try to find closed solutions for a prototype robot, which is a general 2 dof robot having an arbitrary kinematic configuration connected to wrist. These closed form solutions could be attained by different approaches. One possible approach is to decouple the inverse kinematics problem into two simpler problems, known respectively, as inverse position kinematics, and inverse orientation kinematics. To put it in another way, for a 2-dof manipulator with a deficient wrist, the inverse kinematics problem may be separated into two simpler problems, by first finding the position of the intersection of the wrist axes, the gripping point and then finding the orientation of the wrist.

$$ {}^{0}A_1 = \begin{bmatrix} cos\,\theta_1 & -sin\,\theta_1 & 0 & l_1 cos\,\theta_1 \\ sin\,\theta_1 & cos\,\theta_1 & 0 & l_1 sin\,\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.33} $$

$$ {}^{1}A_2 = \begin{bmatrix} cos\,\theta_2 & -sin\,\theta_2 & 0 & l_2 cos\,\theta_2 \\ sin\,\theta_2 & cos\,\theta_2 & 0 & l_2 sin\,\theta_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.34} $$

$$ {}^{2}A_3 = \begin{bmatrix} \cos\theta_3 & 0 & -\sin\theta_3 & 0 \\ \sin\theta_3 & 0 & \cos\theta_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.35} $$

$$ {}^{3}A_4 = \begin{bmatrix} cos\,\theta_4 & -sin\,\theta_4 & 0 & 0 \\ sin\,\theta_4 & cos\,\theta_4 & 0 & 0 \\ 0 & 0 & 1 & l_3 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.36} $$

$$ {}^{0}A_{2} = \begin{bmatrix} \cos\theta_{12} & -\sin\theta_{12} & 0 & l_1\cos\theta_1 + l_2\cos\theta_{12} \\ \sin\theta_{12} & \cos\theta_{12} & 0 & l_1\sin\theta_1 + l_2\sin\theta_{12} \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.37} $$

$$ {}^{2}A_{4} = \begin{bmatrix} \cos\theta_3\cos\theta_4 & -\cos\theta_3\sin\theta_4 & -\sin\theta_3 & -(l_3+d_4)\sin\theta_3 \\ \sin\theta_3\cos\theta_4 & \sin\theta_3\sin\theta_4 & \cos\theta_3 & (l_3+d_4)\cos\theta_3 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.38} $$

3.4.3 Inverse Kinematics Solution Scheme for Manipulator

Inverse kinematics solution is decomposed in to three stages .The first two joint axes actuated in relation with positional references. The third link compensates z axis orientation error. The last link sets gripper orientation parallel to target object orientation.

It is seen from the related transformation matrix that the two link manipulator is highly deficient and can only track points in XY plane which are within the maximum reach pose.

Then task space can be formulated as following

$$ \mathbf{T}: [x\ y\ z]^{\mathrm{T}} \in \Re^3 \mid (z = d_2)\ \&\ (\sqrt{x^2+y^2} \le l_1 + l_2) \tag{3.39} $$

Inverse kinematics task is to find joint variables to position end point of two link manipulator (which is referred as wrist center).Task is easy to solve via well known trigonometric approach.

$$ x^2 + y^2 = l_1^2 + l_2^2 - l_1 l_2\ cos(\pi - \theta_2) \tag{3.40} $$

$$ \theta_2 = \mp a\cos\left(\frac{x^2+y^2-l_1^2-l_2^2}{l_1 l_2}\right) \tag{3.41} $$

$$ \theta_1 = a\tan\left(\frac{y}{x}\right) - a\tan\left(\frac{l_2\ sin\theta_2}{l_1 + l_2\ cos\theta_2}\right) \tag{3.42} $$

Once location of wrist center is found, the next steps are related to orientation of the gripper and based on visual features of target. The third joint angle is found according to shape of the target. If the shape is compact, z axis of image plane is aligned with y axis of end effector coordinate frame (which is also parallel to x axis of robot base coordinate frame figure 3.6). If it is like rectangular in shape then z axis of image plane is aligned with z axis of end effector coordinate frame.

Solution for θ_3 is done using right angled trapezoid formed by robot manipulator. Firstly exploiting the fact that sum of inner angles of closed polygon with for corners is equal to 2π, constraint for θ_3 is found. Afterwards explicit equations are given respectively for the situations corresponding to given conditions:

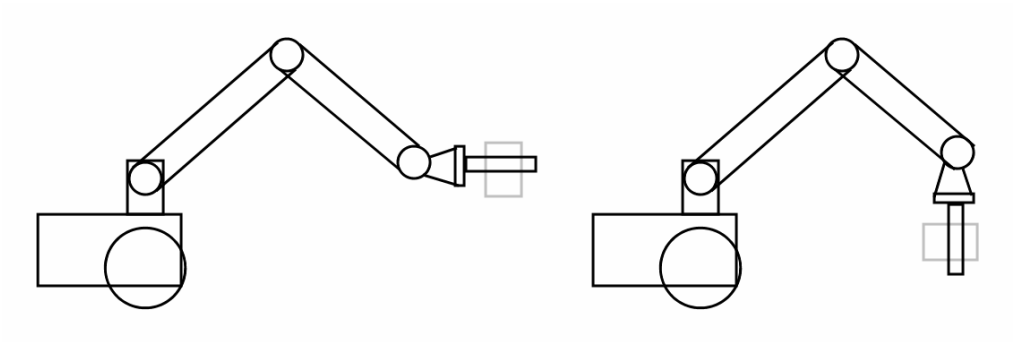


Figure 3.6 Two gripping configurations corresponding to shape of target

$$\theta_1 + \pi - \theta_2 + \pi - \theta_3 + \frac{\pi}{2} = 2\pi \tag{3.43}$$

$$\theta_3 = \frac{\pi}{2} + \theta_1 - \theta_2 \tag{3.44}$$

$$\theta_3 = \theta_1 - \theta_2 \tag{3.45}$$

The first step to derive information about orientation is to construct a covariance matrix using the second order central moments. Hence, it follows:

$$\mathrm{cov}[I(i,j)] = \begin{bmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{bmatrix} \tag{3.46}$$

The eigenvectors of this matrix correspond to the major and minor axes of the image intensity. Thus, the orientation can be extracted from the angle of the eigenvector associated with the largest eigenvalue. The angle, θ_4 is given by the following formula:

$$\theta_4 = \frac{1}{2}\tan^{-1}\left(\frac{2\mu_{11}^{'}}{\mu_{20}^{'} - \mu_{02}^{'}}\right) \tag{3.47}$$

Eventually, all of the necessary variables are solved throughout calculations either through trigonometric equations, iterative calculations or tools for image level calculations.

Distal links of manipulator are equipped with position servos. Therefore only positional equations are solved. In order to maintain reliability of calculations, steady state orientation of mobile platform (angular velocities must converge to zero) must be provided.

3.4.4 Velocity Kinematics of Manipulator

Differential changes in the position and orientation of the end effector are caued by differential changes in the joint coordinates. To obtain a velocity relationship, first forward kinematic map is reconsidered:

$$f(\bar{\theta}) = \bar{x} \tag{3.48}$$

To obtain velocity relationship, equation is differentiated with respect to time

$$\frac{df(\bar{\theta})}{dt} = \frac{d\bar{x}}{dt} \tag{3.49}$$

$$\frac{df(\bar{\theta})}{d\theta}\frac{d\theta}{dt} = \frac{d\bar{x}}{dt} \tag{3.50}$$

$$\frac{df(\bar{\theta})}{d\bar{\theta}}\bar{\theta}' = \bar{x}' \tag{3.51}$$

$$J = \frac{df(\bar{\theta})}{d\bar{\theta}} \tag{3.52}$$

J is a nxm matrix (n is degree of freedom of task space and m is degree of freedom of configuration space) and known as manipulator jacobian matrix. It is formed by taking all partial derivatives of **f** with respect to θ_i. It can be deduced from this derivation that velocity map from task space to configuration space is linear mapping whereas position kinematics are highly nonlinear. But in this study a geometric approach which relates joint motions to wrist center motion through vector operations is adopted because it is more general in application sense (Figure 3.7).

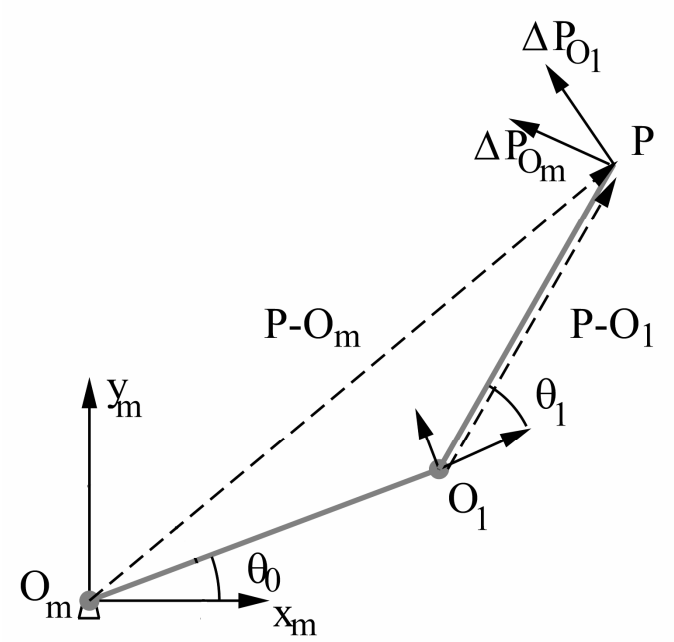


Figure 3.7 Planar projection of manipulator for demonstrating effects of joint motions to end effector displacement.

The incremental movement of the wrist center due to incremental changes in $\Delta\theta_0$ and $\Delta\theta_1$ are given with following equations.

$$\Delta\overline{w}_0 = \Delta\theta_1 \begin{bmatrix} [z_0 x | WO_0|]^T \\ z_{0_0}^T \end{bmatrix} \tag{3.53}$$

$$\Delta\overline{w}_1 = \Delta\theta_2 \begin{bmatrix} [z_1 x | WO_1|]^T \\ z_{1_0}^T \end{bmatrix} \tag{3.54}$$

Expressions in parenthesis define perpendicular vector to rotation axis that connects rotation point and wrist center Jacobian for two degrees of freedom manipulator can be found by placing these equations in order.

$$\overline{w}' = \begin{bmatrix} [z_0 x | WO_0|]^T & [z_1 x | WO_1|]^T \\ z_0^T & z_1^T \end{bmatrix} \overline{\theta}' \tag{3.55}$$

By doing necessary calculations with given parameters, we obtain manipulator jacobian in form (3.57):

$$|O_0 W| = \begin{bmatrix} l_1 \cos\theta_1 + l_2 \cos\theta_{12} \\ l_1 \sin\theta_1 + l_2 \sin\theta_{12} \\ d_2 \end{bmatrix}, |O_1 W| = \begin{bmatrix} l_2 \cos\theta_{12} \\ l_2 \sin\theta_{12} \\ d_2 \end{bmatrix}, z_0 = z_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.56}$$

$$J = \begin{bmatrix} -l_1 \sin\theta_1 - l_2 \sin\theta_{12} & -l_2 \sin\theta_{12} \\ l_1 \cos\theta_1 + l_2 \cos\theta_{12} & l_2 \cos\theta_{12} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \tag{3.57}$$

Manipulator jacobian also relates static forces and joint torques. Inversion of the J is not possible in this situation. But if first two rows are taken into account then joint velocities can be easily calculated:

$$\overline{\theta}' = J^{-1} \overline{x}' \tag{3.58}$$

3.4.5 Euler Lagrange Formulation of Dynamic Equations of Manipulator

In this section, basic concepts of Lagrangian mechanics are presented and dynamic equations of manipulator are derived as a Euler-Lagrange system. Following derivations are done according to didactic input from book Robot Modeling and Control (Spong, Hutchinson & Vidyasagar, 2005)

Euler Langrangian approach formulates equations of motions by using a set of generalized coordinates (joint variables in this case). Therefore it does not contain constrained forces of adjacent links.

The Lagrangian of a system is defined as the difference between its kinetic and potential energy.

$$L = K - P \tag{3.59}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial q_k'}\right) - \frac{\partial L}{\partial q_k} = \tau_k \text{ for k=1,2…n} \tag{3.60}$$

K : Total kinetic energy of robot

P : Total potential energy of robot

q_k, q'_k, :Generalized variable of k-th joint and its first time derivative

τ_k: Generalized force (torque) at k-th joint

Kinetic energy is a quadratic function of velocities of generalized coordinates:

$$K = \frac{1}{2}\bar{q}'^{T} M(\bar{q})\bar{q}' \tag{3.61}$$

Here M(q) is nxn inertia matrix, which is symmetric and positive definite and can be found using equation:

$$M(\bar{q}) = \left[\sum_{i=1}^{n}\left\{m_i J_{vi}(\bar{q})^T J_{vi}(\bar{q}) + J_{wi}(\bar{q})^T R_i(\bar{q}) I_i R_i(\bar{q})^T J_{wi}(\bar{q})\right\}\right] \tag{3.62}$$

m_i : mass of i-th link (3.63)

I_i : mass moment of inertia of i-th link according to its mass center (3.64)

$$v_i = J_{vi}(\bar{q})\bar{q}' \tag{3.65}$$

$$w_i = J_{wi}(\bar{q})\bar{q}' \tag{3.66}$$

It is important to remark here that potential energy is independent from joint velocities. Implementing the given fact, partial derivatives of Lagrangian occurs as:

$$\frac{\partial L}{\partial q_k'} = \sum_j m_{kj}\bar{q}_j' \tag{3.67}$$

$$\frac{d}{dt}\frac{\partial L}{\partial q_k'} = \sum_j m_{kj}q_j'' + \sum_j \frac{\partial m_{kj}}{\partial q_i}q_i'q_j' \tag{3.68}$$

Partial derivatives of the Langrangian w.r.t. k-th joint position is given by:

$$\frac{\partial L}{\partial q_k} = \frac{1}{2}\sum_j \frac{\partial m_{ij}}{\partial q_k}q_i'q_j' - \frac{\partial P}{\partial q_k} \tag{3.69}$$

Thus, Euler Lagrange equation can be written for link k:

$$\sum_j m_{kj}q_j'' + \sum_j \left\{\frac{\partial m_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial m_{ij}}{\partial q_k}\right\}q_i'q_j' - \frac{\partial P}{\partial q_k} = \tau_k \tag{3.70}$$

Although this equation directly defines dynamic behavior of robot, formulation follows using derivations on velocity terms to ease calculations.

$$\sum_j \left\{\frac{\partial m_{kj}}{\partial q_i}\right\}q_i'q_j' = \frac{1}{2}\sum_j \left\{\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j'}\right\}q_i'q_j' \tag{3.71}$$

$$\sum_j \left\{\frac{\partial m_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial m_{ij}}{\partial q_k}\right\}q_i'q_j' = \frac{1}{2}\sum_j \left\{\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j'} - \frac{\partial m_{ij}}{\partial q_k}\right\}q_i'q_j' \tag{3.72}$$

Resultant equation enables us to define coefficients which are referred as Christoffel symbols:

$$c_{ijk} = \frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q'_j} - \frac{\partial m_{ij}}{\partial q_k} \tag{3.73}$$

and by making one more definition

$$g_k = \frac{\partial P}{\partial q_k} \tag{3.74}$$

Euler Lagrange equations for link k can be rewritten:

$$\sum_j m_{kj} q''_j + \sum_i \sum_j c_{ikj}(q) q'_i q'_j - \frac{\partial P}{\partial q_k} = \tau_k \text{ where i,j=1,2,…n.} \tag{3.75}$$

Final equation can be decomposed in to three parts:

- First part involves the second derivative of the generalized coordinatesThe second involves multiples of the first derivatives of generalized coordinates.
- Terms with q_i^2 are called centrifugal and with $q_i q_j$ are Coriolis terms.
- The third term only involves derivatives of generalized coordinates. These terms are arisen from differentiation of potential energy.

This equation is written in common form:

$$M(\bar{q})\bar{q}'' + C(\bar{q},\bar{q}')\bar{q}' + G(\bar{q}) = \tau \tag{3.76}$$

$$C(\bar{q},\bar{q}') = \left[\sum_i c_{ijk} q'_i \right] \tag{3.77}$$

It is important to note that this equation is very general and covers systems with given kinetic energy form and potential energy independent from rates of generalized coordinates.

Now following derivations are due to planar manipulator:

$$J_{vc1} = \begin{bmatrix} -l_{c1} \sin\theta_1 & 0 \\ -l_{c1} \cos\theta_1 & 0 \\ 0 & 0_i \end{bmatrix} \tag{3.78}$$

$$J_{vc1} = \begin{bmatrix} -l_1 \sin\theta_1 - l_{c2} \sin\theta_{12} & -l_{c2} \sin\theta_{12} \\ l_1 \cos\theta_1 + l_{c2} \cos\theta_{12} & l_{c2} \cos\theta_{12} \\ 0 & 0 \end{bmatrix}_i \tag{3.79}$$

$$J_{wc1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \tag{3.80}$$

$$J_{wc2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \tag{3.81}$$

Translational part of kinetic energy

$$\frac{1}{2}\dot{\bar{q}}^{T}\left(m_1 J_{vc1}^{T} J_{vc1} + m_2 J_{vc2}^{T} J_{vc2}\right)\dot{\bar{q}} \tag{3.82}$$

and the rotational kinetic energy

$$\frac{1}{2}\dot{\bar{q}}^{T}\left(I_1\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + I_2\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\right)\dot{\bar{q}} \tag{3.83}$$

form Lagrangian of the system and then inertia matrix and Christoffel symbols are found:

$$M(\bar{q}) = m_1 J_{vc1}^{T} J_{vc1} + m_2 J_{vc2}^{T} J_{vc2} + I_1\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + I_2\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{3.84}$$

$$M(\bar{q}) = \begin{bmatrix} m_1 l_{c1}^{2} + m_2(l_1^{2} + l_{c2}^{2} + l_1 l_{c2}\cos\theta_2) + I_1 + I_2 & m_2(l_{c2}^{2} + l_1 l_{c2}\cos\theta_2) + I_2 \\ m_2(l_{c2}^{2} + l_1 l_{c2}\cos\theta_2) + I_2 & m_2 l_{c2}^{2} + I_2 \end{bmatrix}$$

$$c_{111} = \frac{\partial m_{11}}{\partial q_1} = o \tag{3.85}$$

$$c_{121} = c_{211} = \frac{\partial m_{11}}{\partial q_2} = m_2 l_1 l_{c2} \sin\theta_2 \tag{3.86}$$

$$c_{221} = \frac{\partial m_{12}}{\partial q_2} - \frac{\partial m_{22}}{\partial q_1} = m_2 l_1 l_{c2} \sin\theta_2 \tag{3.87}$$

$$c_{112} = \frac{\partial m_{21}}{\partial q_1} - \frac{\partial m_{11}}{\partial q_2} = -m_2 l_1 l_{c2} \sin\theta_2 \tag{3.88}$$

$$c_{122} = c_{212} = \frac{\partial m_{22}}{\partial q_1} = 0 \tag{3.89}$$

$$c_{222} = \frac{\partial m_{22}}{\partial q_2} = 0 \tag{3.90}$$

$$C(\overline{q}, \overline{q}') = \begin{bmatrix} m_2 l_1 l_{c2} \sin\theta_2 \theta_2^{'} & m_2 l_1 l_{c2} \sin\theta_2 \theta_2^{'} + m_2 l_1 l_{c2} \sin\theta_2 \theta_1^{'} \\ -m_2 l_1 l_{c2} \sin\theta_2 \theta_1^{'} & 0 \end{bmatrix} \tag{3.91}$$

For each link potential energy is its mass times gravitational acceleration times height of its mass center. Differentiating potential energy w.r.t. joint coordinates gives gravitational vector:

$$G(\overline{q}) = g\begin{bmatrix} m_1 l_{c1} + m_2 l_1 \cos\theta_1 + m_2 l_{c2} \cos\theta_{12} \\ m_2 l_2 \cos\theta_{12} \end{bmatrix} \tag{3.92}$$

With calculated matrices, equation 3.76 is differential equation of the overall robot arm.

Mobile platform is nonholonomic system and only velocity kinematical derivations are useful when real time control of the system. Derived dead reckoning formulas are only applicable when offline data is available.

Manipulator arm is basic serial kinematical chain. Kinematics derivations are applicable in both cases velocity kinematics and position based analysis. Gripper is only used to achieve proper approach angle and griping orientation. Therefore, only position based methodology is applied.

CHAPTER FOUR
MATHEMATICAL MODEL OF VISION SYSTEM

4.1 Introduction

This chapter introduces use of vision as a robotic perceptual action. After main considerations about robotic vision are given, a mathematical frame work represented via modeling a camera and a stereo setup.

4.2 Vision as Robotic Ability

Seeing in the third dimension offers solution to task: robot guidance such as positioning and manipulation of environment. Solutions vary from 2D to 3D including hybrid systems where both structures combined specifically (Hartley&Zisserman, 2000).

The robot vision systems are used in various production steps of all industries in world wide and as integrated systems in ongoing processes. The further spread of these systems is mainly due to expanding capability of vision systems when they are cooperated with flexible (dynamically reconfigurable) machines.

Vision systems can be thought of in two ways, according to methodology they use: Low level and high level processes. Low level processes generally deals with image plane and extract knowledge from pixels to construct features (Gonzalez&Woods, 2002). High level processes engage in features from images in a cognitive way to take decisions on situation represented in image (Berry, Martinet & Gallice, 1997). In either case cameras have key functionality on proper operation. This naturally requires a complete understanding of imaging process.

4.3 Camera as a Mapping from 3D to 2D

A camera basically defines a mapping from 3D world (object space) to 2D image. The transformation from 3D world coordinate to 2D image coordinate system is a four step process. First world point is transformed to camera coordinates, and then image plane coordinates are found via perspective projection with pinhole camera geometry. Radial lens distortion is added to approximate pinhole model to real camera systems. Finally, distorted image plane coordinates are transformed into pixel coordinate system (Figure 4.1).

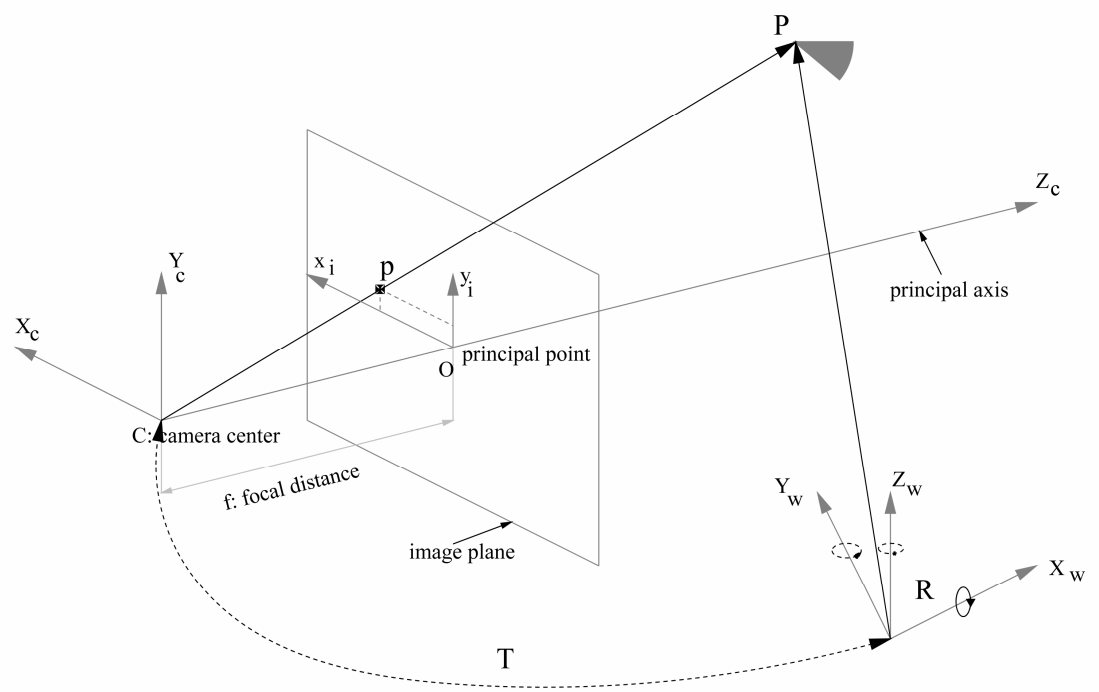


Figure 4.1 Pinhole camera model imaging geometry

In general points are described in different coordinate frames according to tasks they are related. This discrepancy must be overcome with proper rotation and translation (rigid body motion) of coordinate frames. So with well known equations, world point is expressed in camera coordinate frame.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{4.1}$$

Camera performs a mapping from 3D camera to 2D image coordinates. This mapping of three dimensions onto two is called a perspective projection. An ideal model which represents perspective projection is pinhole camera model which has an infinitesimally small hole through which light enters before forming an inverted image on the camera surface facing the hole. To simplify modeling, a pinhole camera is modeled by placing the image plane between the focal point of the camera and the object, so that the image is not inverted (Forsyth & Ponce, 2002).

Under the assumption of pinhole camera model, the undistorted image coordinates (x_i,y_i) are related to the object coordinates (x_c,y_c,z_c) by a set of nonlinear equations.

$$x_i = f\frac{x_c}{z_c} \tag{4.2}$$

$$y_i = f\frac{y_c}{z_c} \tag{4.3}$$

These nonlinear equations turn into linear mapping between homogenous coordinates of corresponding points.

$$\begin{bmatrix} fx_c \\ fy_c \\ z_c \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{4.4}$$

$$p = MP \tag{4.5}$$

Real optical systems suffer from a number of inevitable geometric distortions. In optical systems made of spherical surfaces, with centers along the optical axis, a geometric distortion occurs in the radial direction. A point is imaged at a distance

from the principle point that is larger (pin-cushion distortion) or smaller (barrel distortion) than predicted by the perspective projection equations; the displacement increasing with distance from the center (Heikkila & Silven, 1997). It is small for directions that are nearly parallel to the optical axis and grows as power series of the distance from the image center:

$$\begin{aligned} x_i &= x_u + D_x \\ y_i &= y_u + D_y \\ D_x &= x_d(\kappa_1 r^2 + \kappa_2 r^4 + ...) \\ D_y &= y_d(\kappa_1 r^2 + \kappa_2 r^4 + ...) \text{ where } r = \sqrt{x_d^2 + y_d^2} \end{aligned} \tag{4.6}$$

In CCD cameras, the initially discrete sensor signal is conditioned to produce a smooth video output signal in standard form that hides the transitions between cells of the sensor. This waveform is then digitized in the frame grabber. The sampling in the horizontal direction in the frame grabber is typically not equal to the spacing of sensor cells, and is not known accurately. The horizontal spacing between pixels in the sampled image do not in general correspond to the horizontal spacing between cells in the image sensor. Therefore to convert distorted image coordinates to computer image coordinates, three parameters are introduced two for vertical and horizontal scaling (to from meters to pixels) and a skew parameter to indicate deviation of pixels from being rectangular (s_x,s_y,α).

$$\begin{bmatrix} \frac{f}{s_x}x_c + \alpha\frac{f}{s_x}y_c + z_c o_x \\ \frac{f}{s_y}y_c + z_c o_y \\ z_c \end{bmatrix} = \begin{bmatrix} \frac{f}{s_x} & \alpha\frac{f}{s_x} & o_x & 0 \\ 0 & \frac{f}{s_y} & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{4.7}$$

4.4 3D Reconstruction and Stereo Vision

An important research topic in vision, referred as stereo vision has the objective to compute a 3-Dimensional representation of some scenery from two 2-Dimensional digital images.

Constructing a 3-dimensional representation involves finding pairs of pixels from the two images which correspond to the same point in space. Pixel based matching algorithms require lots of processing power and memory space. However, stereo vision applications are often subject to strict real-time requirements. In order to meet timing constraints, a simplified approach is adopted using primitive image processing techniques. However improved matching algorithms are available in existence of equipments, capable of parallel processing (Sun&Peleg, 2004). Given the disparity map and the geometry of the stereo setup, triangulation is the task of computing the 3D position of points in the images (Trucco&Verri, 1998).

Consider now a pair of pinhole cameras mounted onto a stereo rig, briefly called a “stereo camera". The basic properties of such a system are recapitulated in this section. It is presented as a triangulation device which reconstruction of the three-dimensional workspace in the respective ambient space. This section concludes with the “rigid stereo assumption", which perception algorithm of this thesis is based on. Briefly, it states that a stereo camera with a fixed geometry results a well defined world point.

4.4.1 Geometry of Stereo Camera

Consider a pair of Euclidean reference frames, F_L and F_R, on the left and right camera. Then, the extrinsic geometry of the rig is described by the pose of the left camera with respect to the right one. This is expressed as a rotation ${}_LR^R$ followed by the translation ${}_LT^R$, which would move the left frame onto the right one, or analogously, which transforms the coordinates in the left ambient space to the right ones (Figure 4.2).

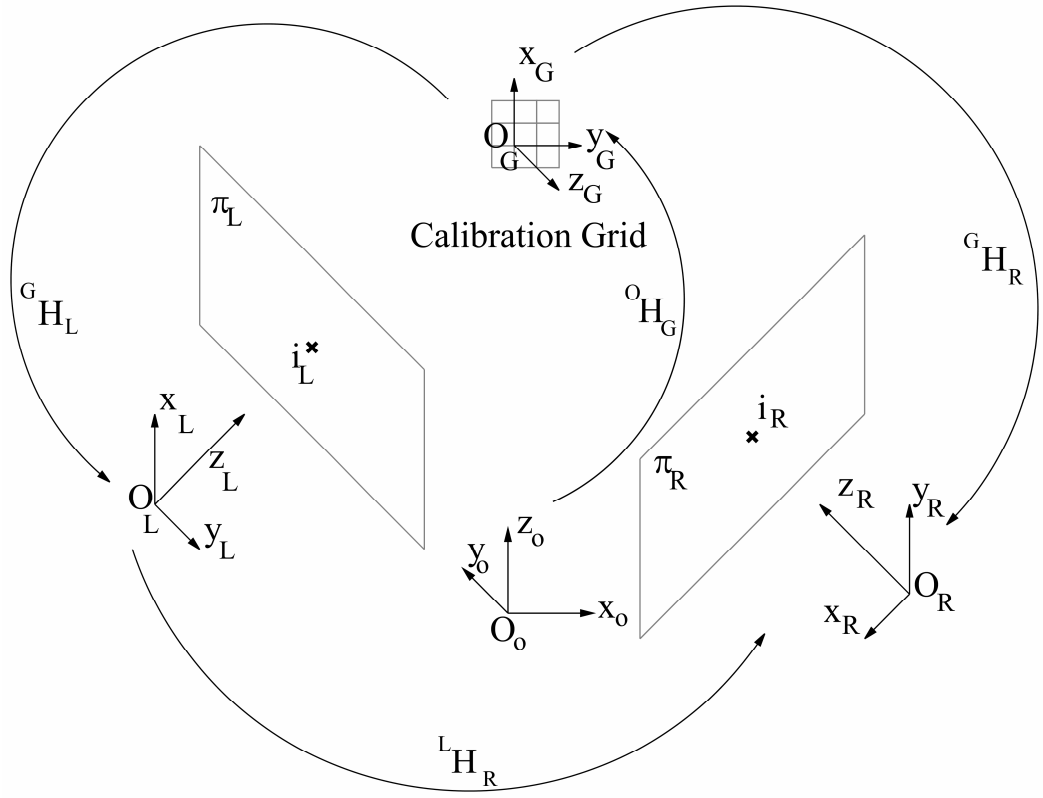


Figure 4.2 Stereo camera setup

$$P_L = {}_L R^R P_R + {}_L T^R \tag{4.8}$$

The formal extension to a homogeneous fourth coordinate for a rigorous treatment which allows this transformation, which is non-linear in the summed form above, to be represented linearly as a single multiplication with a 4 x4 homogenous transformation matrix:

$$\begin{bmatrix} P_L \\ 1 \end{bmatrix} = \begin{bmatrix} {}_L R^R & {}_L T^R \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} P_R \\ 1 \end{bmatrix} \tag{4.9}$$

Implemented methods for 3 dimensional recoveries mainly correspond to three situations. If only the intrinsic parameters of the cameras are known then 3D point can be found only up to a scale factor. In the absence of intrinsic and extrinsic

parameters, only a projective transformation can be obtained and the triangulation is used if fully calibrated fixed cameras are available (Trucco&Verri, 1998).

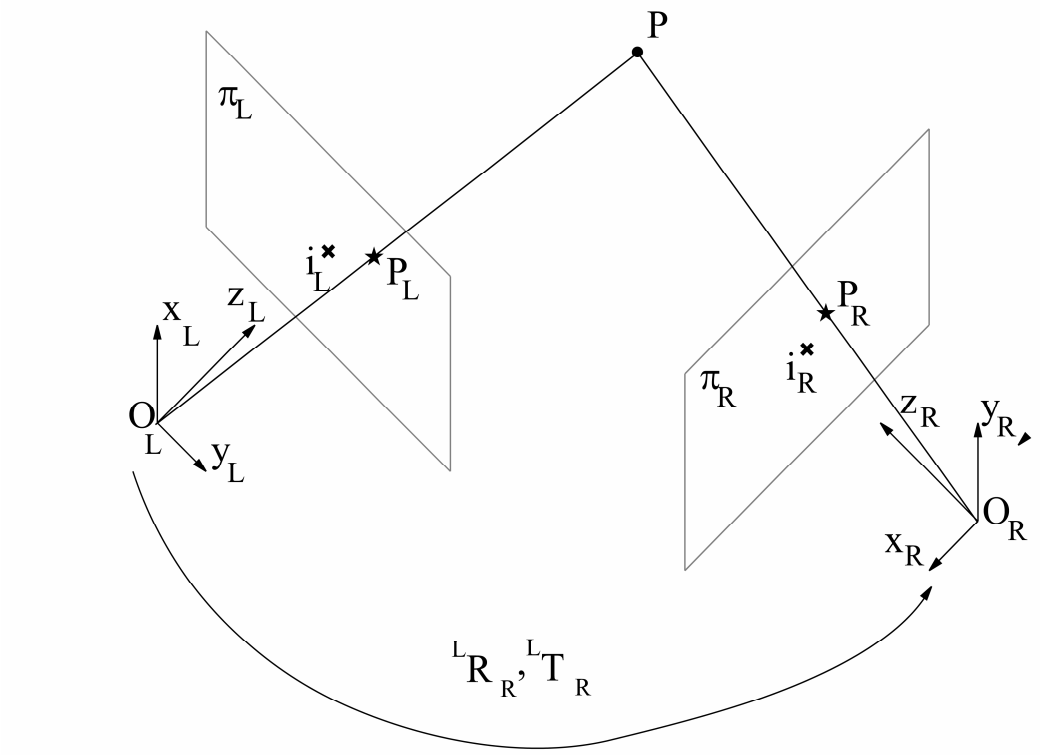


Figure 4.3 Stereo triangulation

4.4.2 Stereo Triangulation

Two corresponding position vectors of same point in camera reference frames, P_L and $P_{R,}$ and image coordinates of the point P, p_L and p_R, evidently provides a solution for triangulation equations. The 3D world reference system is chosen coincident with the left camera reference system. The right camera is translated and rotated with respect from the left one (Figure 4.3).

Vectors P_L and P_R are related to each other with rigid body motion equations if rotation matrix R and translation between cameras coordinate frames are known.

$$P_L = {}_L R^R P_R + {}_L T^R \tag{4.10}$$

Recalling perspective projection equations in pinhole camera models, image points p_L and p_R can be written in form:

$$\begin{aligned} P_L &= z_L p_L \\ P_R &= z_R p_R \end{aligned} \tag{4.11}$$

Placing these equations in (4.10) leads following relation:

$$z_L p_L = {}_L R^R z_R p_R + {}_L T^R \tag{4.12}$$

These relations describes the first relation (equation 4.10) in a manner that it is possible to find z coordinate, namely depth, of real world point in camera coordinate frames if two corresponding image points for a stereo pair are given.

$$\begin{bmatrix} -{}_L R^R p_R & p_L \end{bmatrix} \begin{bmatrix} z_R \\ z_L \end{bmatrix} = {}_L T^R \tag{4.13}$$

$$A = \begin{bmatrix} -{}_L R^R p_R & p_L \end{bmatrix} \tag{4.14}$$

Matrix A is rectangular in form 3x2 and matrix equation gives over determined system with three equations for two unknowns which leads least square solution for $[z_R\ z_L]^T$.

$$\begin{bmatrix} z_R \\ z_L \end{bmatrix} = (A^T A)^{-1} A^T {}_L T^R \tag{4.15}$$

Right camera frame z coordinate for given point is explicitly found via following equation

$$z_R = \frac{\|p_L\|^2 \langle -{}_L R^R p_R, T \rangle - \langle -{}_L R^R p_R, p_L \rangle \langle p_L, T \rangle}{\|p_L\|^2 \| -{}_L R^R p_R \|^2 - \langle -{}_L R^R p_R, p_L \rangle^2} \tag{4.16}$$

where ||.|| is Euclidian norm operator and <.,.> is scalar product operator. 3D point coordinates in right camera frame are calculated by recalling equation 4.16.

4.5 Camera Calibration

Camera calibration is defined as the process of relating the ideal model of the camera to the actual physical device and of determining the position and orientation of the camera with respect to a world reference system (Hartley&Zisserman, 2000). Above developments show that 3D reconstruction of a world point strictly depends on the amount of a priori knowledge available on the parameters imaging system. It means that camera matrix and extrinsic parameters must be known.

Well defined camera calibration procedure (Sturm&Maybank,1999), (Zhang, 1999) based on imaging landmarks with known positions, is utilized to calculate necessary parameters.

Method utilized in this study is adopted from implementation scheme given in Tsai (1987) and implemented through camera calibration toolbox for Matlab (Bouget, 2007).

4.5.1 Camera Calibration Problem

Camera calibration in the context of three-dimensional machine vision is the process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and/or the 3D position and orientation of the camera frame relative to a certain world coordinate system (extrinsic parameters). In many cases, the overall performance of the machine vision system strongly depends on the accuracy of the camera calibration. The result of camera calibration is an explicit transformation that maps a 3D world point $P_w=(X_w,Y_w,Z_w,1)^T$ into a 2D pixel $P_i=(x_i,y_i,1)^T$. This mapping can be represented by a 3 x 4 projection matrix, M, that encompasses physical parameters: rotation angles, translations, the coordinates of the principal point (x_0,y_0), the skew alpha_c between the image axes and the radial distortion.

4.5.2 Camera Parameters

Camera coordinate frame position must be described according to a known world coordinate frame. S extrinsic parameter of camera (rigid body motion in 3D space) has obviously six degrees of freedom represented by transformation. The unknowns to be recovered in the problem are the translation vector T and the rotation matrix R. In cameras, the initially discrete sensor signal is conditioned to produce a smooth video output signal in standard form that hides the transitions between cells of the sensor. Due to structure of imaging sensor and sampling during acquisition, the ratio of picture cell size in the horizontal and in the vertical direction is not known a priori from the dimensions of the sensor cells and needs to be determined with using frequency domain methods. Alternatively, the extra scaling parameter "s" can be recovered as part of the camera calibration process. In this case we use a rectified equation for x_i.

$$\frac{x_i - o_x}{f} = s\frac{x_c}{z_c} \tag{4.17}$$

$$\frac{y_i - o_y}{f} = \frac{y_c}{z_c} \tag{4.18}$$

If we combine the equations for interior and exterior orientation we obtain:

$$\frac{x_i - o_x}{f} = s\frac{r_{11}x_w + r_{12}y_w + r_{13}z_w + t_x}{r_{31}x_w + r_{32}y_w + r_{33}z_w + t_z} \tag{4.19}$$

$$\frac{y_i - o_y}{f} = \frac{r_{21}x_w + r_{22}y_w + r_{23}z_w + t_y}{r_{31}x_w + r_{32}y_w + r_{33}z_w + t_z} \tag{4.20}$$

Last equations ignore radial distortions described in camera model a post processing step is proceeded to find radial distortion coefficients. Process implements least square optimization: Redundant points are used to optimize solution and exterminate the distortion in image.

4.5.3 Solution Strategy

In calibration process, a target of known geometry is imaged. Correspondences between target points and their images are obtained. These form the basic data on which the calibration is based.

Tsai's method, first tries to obtain estimates of as many parameters as possible using linear least-squares fitting methods. In this initial step, constraints between parameters (such as the orthonormality of a rotation matrix) are not enforced, and what is minimized is not the error in the image plane, but a quantity that simplifies the analysis and leads to linear equations. This does not affect the final result, however, since these estimated parameter values are used only as starting values for the final optimization (Tsai, 1987).

In a subsequent step, the rest of the parameters are obtained using a non-linear optimization method that finds the best fit between the observed image points and those predicted from the target model. Parameters estimated in the first step are refined in the process.

4.5.4 Estimating the Rotation and Part of the Translation

Initially we assume that we have a reasonable estimate of the position of the principle point (o_x ,o_y). This point is usually near the middle of the sensor. We refer coordinates to this point using equations:

$$x_i^{'} = x_i - o_x \tag{4.21}$$

$$y_i^{'} = y_i - o_y \tag{4.22}$$

So that;

$$\frac{x_i^{'}}{f} = s\frac{x_c}{z_c} \tag{4.23}$$

$$\frac{y_i^{'}}{f} = \frac{y_c}{z_c} \tag{4.24}$$

Next, we consider only the direction of the point in the image as measured from the principle point. This yields a result that is independent of the unknown principle distance f. It is also independent from radial distortion.

$$\frac{x_i^{'}}{y_i^{'}} = s\frac{x_c}{y_c} \tag{4.25}$$

Using the expansion in terms of components of the rotation matrix *R* we obtain:

$$\frac{x_i^{'}}{y_i^{'}} = s\frac{r_{11}x_w + r_{12}y_w + r_{13}z_w + t_x}{r_{21}x_w + r_{22}y_w + r_{23}z_w + t_y} \tag{4.26}$$

$$sy_i^{'}\left(r_{11}x_w + r_{12}y_w + r_{13}z_w + t_x\right) - x_i^{'}\left(r_{21}x_w + r_{22}y_w + r_{23}z_w + t_y\right) = 0 \tag{4.27}$$

Equation (14) is linear homogeneous equation with the eight unknowns which are: $sr_{11}, sr_{12}, sr_{13}\ r_{21}, r_{22}, r_{23}, t_y, st_x$ The coefficients in the equation are products of components of corresponding scene and image coordinates.

This equation can be obtained for every correspondence between a calibration target point (x_w ,y_w ,z_w) and an image point (x_i ,y_i). Its is important to take into account that there is an unknown scale factor since these equations are homogeneous. Therefore if there is a solution for the eight unknowns, then any multiple of that solution is also a solution. In order to obtain one solution, the homogeneous equation is converted into inhomogeneous equation by arbitrarily setting one unknown.

$$t_y{'} = 1 \tag{4.28}$$

There are seven unknowns for which a solution can be found if there are seven correspondences between target coordinates and image coordinates. If there is more

than seven correspondences, solution set can be tuned by minimizing the sum of squares of errors. Supposing that the best fit solution of the set of equations is:

$$s(r_{11})', s(r_{12})', s(r_{13})', (r_{21})', (r_{22})', (r_{23})', s(t_x)' \text{ and } t_y' = 1 \tag{4.29}$$

Correct scale factor "c" for solution, which converts t_y' to t_y, must be found. The correct scale factor can be estimated by noting that the rows of the rotation matrix are supposed to be unit vectors.

$$c = \frac{1}{\sqrt{(r_{21})'^2 + (r_{22})'^2 + (r_{23})'^2}} \tag{4.30}$$

The ratio s of the horizontal pixel spacing to the vertical pixel spacing can be calculated via normality of first row.

$$\frac{c}{s} = \frac{1}{\sqrt{s(r_{11})'^2 + (sr_{12})'^2 + (r_{13})'^2}} \tag{4.31}$$

Since the rotation matrix is orthonormal so third row of matrix can be obtained simply by taking the cross-product of the first two rows.

4.5.5 Focal Distance and Distance to the Scene

Up to the point, the rotation matrix R and the first two components of the translation (t_x, t_y) are estimated. The third component t_z of the translation vector and the principle distance f can be estimated by deriving equations:

$$\frac{x_i'}{f} = s\frac{r_{11}x_w + r_{12}y_w + r_{13}z_w + t_x}{r_{31}x_w + r_{32}y_w + r_{33}z_w + t_z} \tag{4.32}$$

$$\frac{y_i'}{f} = \frac{r_{21}x_w + r_{22}y_w + r_{23}z_w + t_y}{r_{31}x_w + r_{32}y_w + r_{33}z_w + t_z} \tag{4.33}$$

$$x_i'(r_{31}x_w + r_{32}y_w + r_{33}z_w) = s(r_{11}x_w + r_{12}y_w + r_{13}z_w + t_x)f - x_i't_z \quad (4.34)$$

$$y_i'(r_{31}x_w + r_{32}y_w + r_{33}z_w) = (r_{21}x_w + r_{22}y_w + r_{23}z_w + t_y)f - y_i't_z \quad (4.35)$$

Given the estimates for *R*, these equations are linear equations with two unknowns *f* and t_z. We can solve these equations for *f* and t_z using one or more correspondence between target and image. If there are many correspondences to use, the resulting over-determined system can be solved using least-squares methods.

Only parameters, which remained undefined, are image center coordinates. Paradigm that covers the methodology to calculate these points is by using vanishing points that occurs in perspective projection. Vanishing points are meeting points of lines in the image, real world correspondences of which are parallel. Image center can be calculated with projective geometry postula: If three of the vanishing points in image are known image center stands on the orthocenter of the triangle formed by these points.

Once all estimations according to parameters are done, a verification process is consistently repeated to minimize the image errors which is the difference between the observed image positions and the positions predicted based on the known target coordinates. Results of the process are evaluated and parameters are statistically determined. Error in the calibration parameters is directly proportional to error in image measurements. The proportionality factor depends on the imaging geometry and on the design of the target.

In this chapter, mathematical models for robotic vision system are derived from 3D recovery point of view. Pinhole camera model is analyzed with basic notions of projective geometry and homogenous transformations. Afterwards fundamental concepts of stereo systems are recapitulated and mathematics of stereo triangulation is presented. The experimental procedure for calibration of cameras is also analyzed according to Tsai (1987).

Concepts introduced in this part of work are strictly followed when designing perceptual system for robotic motion generation. Implementation constraints and schemes are expressed where they are used in control algorithms. Therefore results of these applications are given in chapters concerning control.

CHAPTER FIVE
VISION BASED CONTROL METHODS

5.1 Introduction

Traditional robot control uses world or joint coordinate representations to describe tasks composed of planning and execution of the moves. In a static environment, this works well if the environment, the robot model and the task is known for any sequence of operation. This known architecture of a-priori models is exploited to transform the task into a sequence of robot motions.

5.2 Vision Based Control of Robot Manipulators

From the beginning of this text, it is implied that vision as a perceptual action is an important phenomenon for control systems because it provides more information than conventional encoders and tachogenerators which are only joint based sensors. In contrast vision sensors (cameras) provide a feedback which enables autonomous programming of motion from task specifications to joint actuation (Shirai & Inoue, 1973), (Weiss, Sanderson & Neuman, 1987), (Hutchinson, Hager & Corke,1996), (Hashimoto, 2003).

Vision based control is a technique where the appearance of a target in the image or an estimation of the observed target position is used to control the position of the robot and to move it to a desired position in the 3D scene. More generally, vision based control is an appealing technique which enables the loop to be closed between sensing and action (Malis, Chaumette & Boudet, 1999).

An important application of visual control ability is control of robotic systems. The approaches from disciplines such as robotics, high-speed image processing, and real-time control had addressed the vision based control from different views. The richness of the data that can be derived from vision, as well as the inherent need of

endures to observe visual images, motivates the use of vision in controlling robot systems (Corke, 1994), (Namiki, Hashimoto & Ishikawa, 2003).

5.3 Taxonomy of Vision Based Control Systems

All subsequent visual servo systems can be categorized according to answers to two questions (Hutchinson, Hager & Corke,1996):

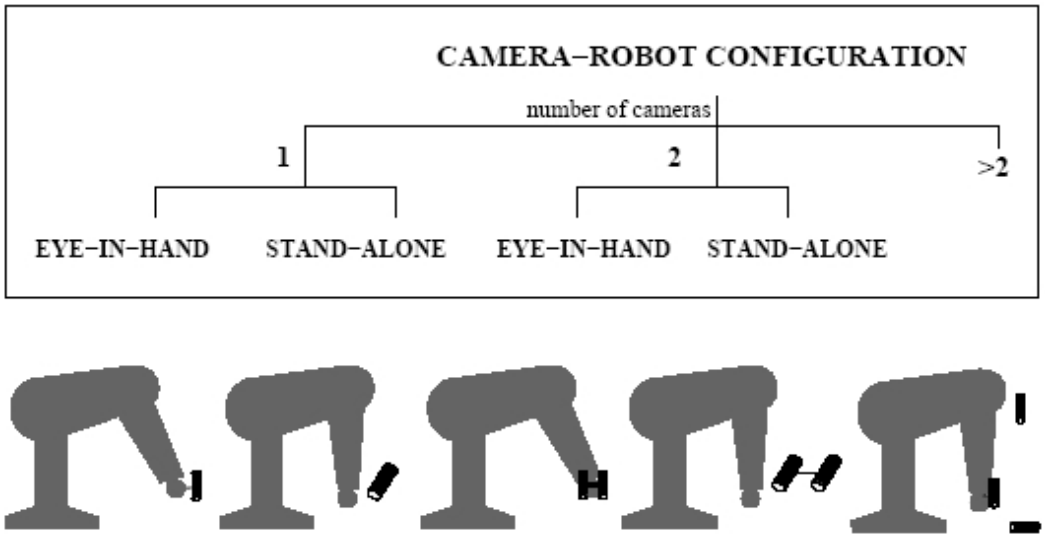


Figure 5.1 Integration of vision system and an industrial robot arm

1. Is the control structure hierarchical, with the vision system providing set-points as input to the robot's joint-level controller, or does the visual controller directly compute the joint-level inputs?

2. Is the error signal defined in 3D (task space) coordinates, or directly in terms of image features?

5.3.1 Models of Execution (direct or indirect)

The first question characterizes the execution mode of the robotic system. If the control architecture is hierarchical and uses the vision system to provide set-point inputs to the joint-level controller, thus making use of joint feedback to internally

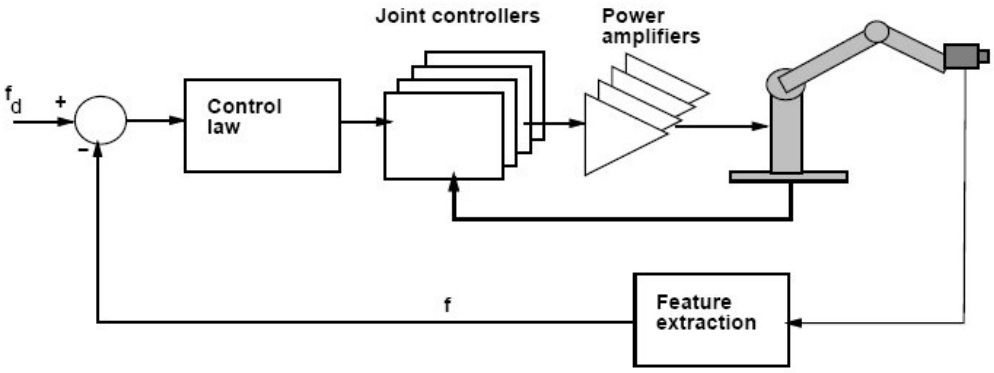
stabilize the robot, it is referred to as a dynamic look-and-move system or indirect visual servo system (Figure 5.1).

In contrast, direct visual servo eliminates the robot controller entirely replacing it with a visual servo controller that directly computes joint inputs, thus using vision alone to stabilize the mechanism.

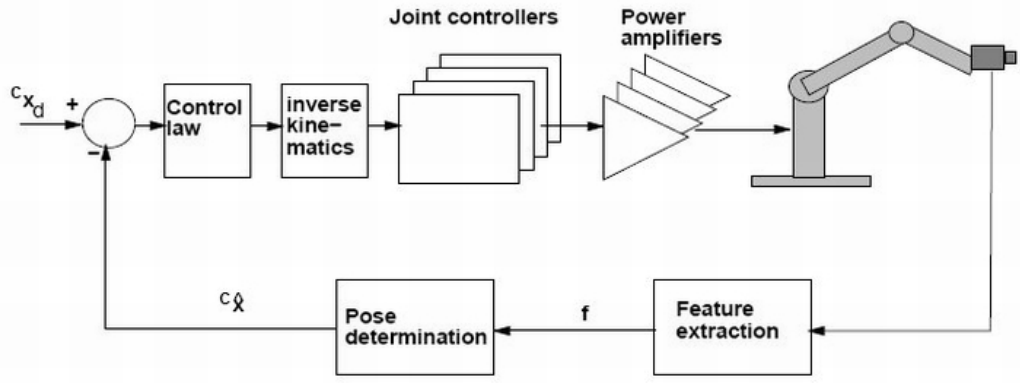
5.3.2 Control Law Computation (position based or vision based)

The second major classification of systems distinguishes position-based control from image-based control. In position-based control, features are extracted from the image and used in conjunction with a geometric model of the target and the known camera model to estimate the pose of the target with respect to the camera. Feedback is computed by reducing errors in estimated pose space. In image-based servoing, control values are computed on the basis of image features directly. The image-based approach may reduce computational delay, eliminate the necessity for image interpretation and eliminate errors due to sensor modeling and camera calibration. However it does present a significant challenge to controller design since the plant is nonlinear and highly coupled.

With given algorithms, the position of the end-point relative to the object is determined only indirectly by its known kinematic relationship with the camera. Errors in this kinematic relationship lead to positioning errors which cannot be observed by the system. Observing the end-effector directly makes it possible to sense and correct for such errors. In general, there is no guarantee on the positioning accuracy of the system but to emphasize this distinction, systems that only observe the target object is referred as endpoint open-loop (EOL) systems, and systems that observe both the target object and the robot end-effector as endpoint closed-loop (ECL) systems.



Dynamic image-based look-and-move structure.



Position-based visual servo (PBVS) structure as per Weiss.

Figure 5.2 Presentation of image and position based visual servo control of articulated arm

5.4. Visual Servo Control

As described in the previous section, in position-based visual servoing (Martinet & Gallice, 1999), features are extracted from the image and used to estimate the pose of the target with respect to the camera. Using these values, an error between the current and the desired pose of the robot is defined in the task space. In this way, position-based control neatly separates the control issues, namely the computation of the feedback signal, from the estimation problems involved in computing position or pose from visual data (Figure 5.2).

The notion of a positioning task is formalized as follows: “A positioning task is represented by a function. This function is referred to as the kinematic error function. A positioning task is fulfilled with the end-point is in desired pose x_e, if E(.) = 0”.

Once a suitable kinematic error function has been defined and the parameters of the functions are instantiated from visual data, a regulator is defined that reduces the estimated value of the kinematic error function to zero. This regulator produces at every time instant a desired end-effector velocity screw that is sent to the robot control subsystem.

The process is to first determine the relative motion that would fulfill the task, and then to write a control law that would produce that motion. A point to point positioning task is considered in details to enlighten operational scheme of servoing architecture.

Let ${}^{E}r$ is end effector point to position and S is target position which is in view of field of vision system. A proper kinematic error function for this task is defined as:

$$E(H_E : S, {}^{E}P) = S - H_E \, {}^{E}r \tag{5.1}$$

H_E is homogenous transformation matrix representing en effector pose

Because point to point positioning is considered, only 3 of the 6 degrees of freedom in space are constrained in this kinematic error function. Hence only translational velocity of the end effector can be controlled.

Assuming a calibrated vision system; camera pose is known. Then measured stationing point position is found through following equation

$$\hat{S} = H_C \, {}^{C}\hat{S} \tag{5.2}$$

H_C : homogenous transformation matrix representing camera pose

Placing estimation in equation 5.2 to kinematic error function and applying proportional control yields a control value as:

$$u_T = k\left(H_C{}^C \hat{S} - H_E{}^E r\right) \tag{5.3}$$

This control value drives system to an equilibrium state, where visually measured position and end effector point are coincident.

From resultant equation, it can be deduced that positioning accuracy is strongly related to camera calibration, visual observation of target point and robot transformation matrix. It is also important that Cartesian control developed above must be converted to actuator velocities.

On the other aspect of visual control similar operations are done in image plane. (Espiau, Chaumette & Rives, 1992) In this case; kinematic error function is replaced by an image error function. This can be done by directly using projection models derived in chapter three. The issue to mark in image based control is that a specific configuration on image is the definition of the stationing point. Control task is to find proper feedback law to relate changes in image plane with robot control inputs.

It is an obvious requirement for image based control that differential changes in image plane cause differential changes in robot motion. The image jacobian is defined as a linear mapping providing this coupling.

$$f' = J_{image} r' \tag{5.4}$$

J_{image} : image jacobian (or interaction matrix [references])

f' : vector representing changes in image plane

r' : velocity vector in R^3 representing robot motion in world coordinates

Computing image jacobian is procedural work and is widely used in applications. General procedure to find image jacobian can be summarized as following (Hutchinson, Hager & Corke,1996):

1. Representation of robot motion in camera reference frame is realized by using homogenous transformation matrix from world coordinates to camera coordinates.

$${}^{C}r' = {}^{C}\Omega \times {}^{C}r + {}^{C}T$$

2. Use of camera projection model to relate point r=[x y z]^{T} with image point f=[u v] (or feature). If perspective projection is considered, Set of equations are:

$$\pi(x,y,z) = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{z}\begin{bmatrix} x \\ y \end{bmatrix}$$

3. Rearranging the explicit equations to form matrix equation in n.

Process is highly related to selections on the image. Features to be selected must have some crucial properties concerning motion of the target point so that proper tracking of the target becomes possible.

When point to point positioning considered in image plane motion, a necessary and sufficient condition to for ${}^{E}r$ and S is to coincide in each image.(From given facts in chapter about vision systems, it is known that 3D problems can be overcome using two or more images).

Assuming two cameras, a stereo setup, is observing the scene, [u^{l},v^{l}] and [u^{r},v^{r}] are projections of point ${}^{E}r$ in left and right image plane respectively. Feature vector is then

$$f = \begin{bmatrix} u^{l} \\ v^{l} \\ u^{r} \\ v^{r} \end{bmatrix} \tag{5.5}$$

and desired feature vector in a similar manner is

$$f_d = \begin{bmatrix} u_S^l \\ v_S^l \\ u_S^r \\ v_S^r \end{bmatrix}. \tag{5.6}$$

This yields an error function in simplest for as

$$E(f) = f - f_d \tag{5.7}$$

Image jacobian for this task can be constructed by following instructive process described above. But it is important to mark that velocity vector must be in either one of the camera frames so it is necessary to implement transformations relating two camera frames.

One of the chief advantages to image based control over position based control is that the positioning accuracy of the system is less sensitive camera calibration. It is important to note however that most of the image based control methods appearing in the literature still rely on an estimate of point position or target pose to parameterize the Jacobian In practice the unknown parameter for Jacobian calculation is distance from the camera One disadvantage of image based methods over position based methods is the presence of singularities in the feature mapping function which reflect themselves as unstable points in the inverse Jacobian control law These instabilities are often less prevalent in the equivalent position based scheme (Martinet, 1999).

As described in this chapter visual servoing is the fusion of results from many elemental areas including high-speed image processing, kinematics, dynamics, control theory, and real-time computing. It has much in common with research into active vision and structure from motion, but is quite different from the often described use of vision in hierarchical task-level robot control systems (Namiki, Hashimoto & Ishikawa, 2003).

5.5 Implemented Visual Control Methodology

Many of the control and vision problems are similar to those encountered by active vision researchers who are building “robotic heads”. However the task in visual servoing is to control a robot to manipulate its environment using vision as opposed to just observing the environment. In proposed system, robot ability is not limited with tracking or manipulation ability as it is separated in active vision and visual servoing applications but a fusion of the capabilities from two research field. Concerning structure of implementation model, three problems are occurred to be solved (Figure 5.3):

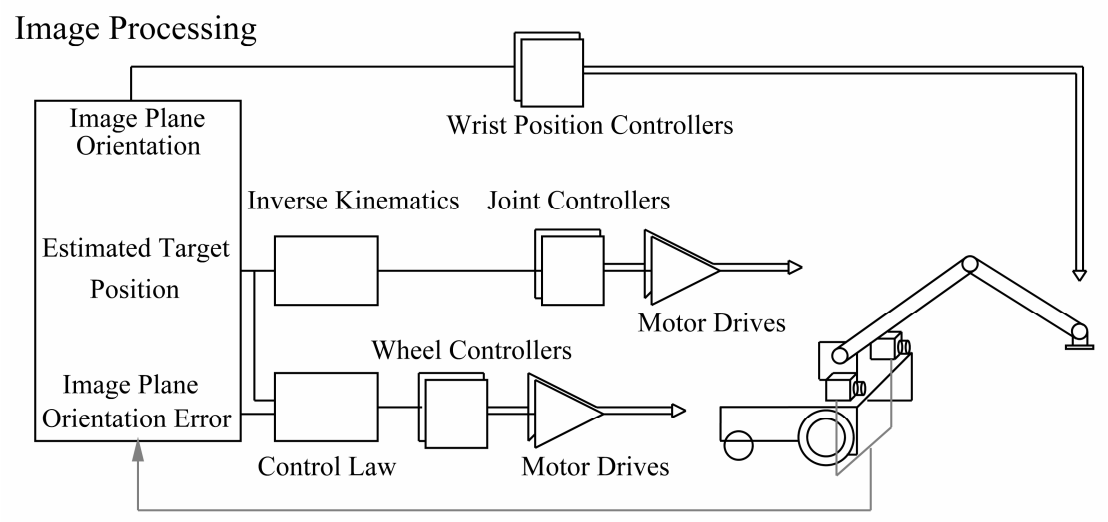


Figure 5.3 Vision system integration to mobile robot

1. Control of mobile platform: It is necessary to track targets in the field of vision and in the reach limitations of the robot manipulator. This problem, by its nature, is a visual servoing problem where pose error cannot be directly related to sensory and actuators signals but related to integral of them. Cameras mounted on a mobile platform are not only track objects to keep them in the field of view of cameras but also keep the object within dexterous workspace of the planar manipulator. Tracking ability of active vision system, namely cameras on mobile platform, is constrained with operation plane of the planar manipulator.

2. Control of robot manipulator: Planar manipulator is to track position of the target in a perpendicular plane which forms 3D world when combined with plane that mobile platform moves. Control task is to dynamically handle target, viewed by cameras. Dynamic look and move paradigm is suitable for the general case in this study, because the system equipped with servo controllers.

3. Control of gripper: When tracking algorithm reaches a steady state, gripper commanded to operate and interact with target. Gripper actuators are position servos and they are also commanded through vision process by referring key features due to target orientation.

CHAPTER SIX
ACTUATOR MODELS AND CONTROL STRATEGY

6.1 Introduction

In chapter two, it is mentioned that actuator for mechanical systems are D.C. motors. It is also mentioned that control methods for three different parts of mechanical system are also different. This chapter will discuss characteristics and tuning of actuator controllers.

Mobile platform is equipped with two D.C. Motors. Because mobile platform dynamics are mainly depended upon velocity constraints, these D.C. motors are controlled with PI algorithm. The cart motion requires lower speed and high torques when compared with direct motor axis ratings so two motors with worm gear box are settled as wheel drives. Final output axis speed is 60 rpm.

There are two different types of D.C. motors implemented on manipulator arm. One, which drives the first link, has a reduction ratio about 250. The output speed of motor gearbox configuration is approximately 30 rpm. The second link of the robot has a same D.C. Motor setup as mobile platform. Main difference between mobile platform drive setup and manipulator arm setup is that they have integrable constraints and thus they need position feedback. Two precise potentiometers are used for converting relative angular motions to voltage values. The gripper motion is controlled with two hobby servo motors. These motors are also D.C. motors but they have integrated in a specific feedback control system. (Sawicsz, 2002)

This chapter will demonstrate control system for low level actuator from practical point of view. The mathematical model of permanent magnet D.C. Motor is given as a starting point for analysis. Then joint dynamics is inserted to resulting model. Final system is characterized as nonlinear. An experimental tuning methodology is represented for PI controller which is linear in its nature. Related calibration routine

for voltage motion conversion is explained where it is implemented in steps of algorithm.

6.2 Actuators

In previous chapters, mathematical models are developed for perceptual and mechanical system. These models are exploited for relating robot configuration space and task space. When system is taken from experimental point of view, modeling has to be extended one level deeper to actuators.

There are some points to be taken into account before starting modeling of actuators. Stand point of these aspects is the results of vision based control systems. According to controller to be either a manipulator or differential drive platform controller, interpretation of commands on actuator space change. In a dynamic look and move system, manipulator references are positions and actuator controllers are position PI controllers. Whereas for differential drive platform controllers, where position feedback is not available and error is generated in image plane, system inputs are generally angular velocities. This evidently leads velocity servos in differential drive platform.

Since all drives are permanent magnet DC motors, the control problem is simplified as angular velocity and position control according to commanded reference movements. The motion variables are the voltage and current to the actuators that excite the system.

PMDC Motor is composed of two cascaded physical systems: Electrical part which generates current to obtain magnetic coupling forces and mechanical part which generates motion on output axis. Electrical system is the armature circuit of the PMDC motor. It is a simple electrical system with an inductance and resistor and generates a current flow which is directly proportional to torque generated on motor axe. Motor axe is also under the affect of torque which is caused by load side of the system. The torqueses affect mechanical system which is a first order with effect of

dynamic friction (damping effect) and inertia of rotary parts. Differential equations for the electrical and mechanical parts of PMDC motor are obtained Kirrchoff Voltage law on armature circuit and torque balance in motor axis:

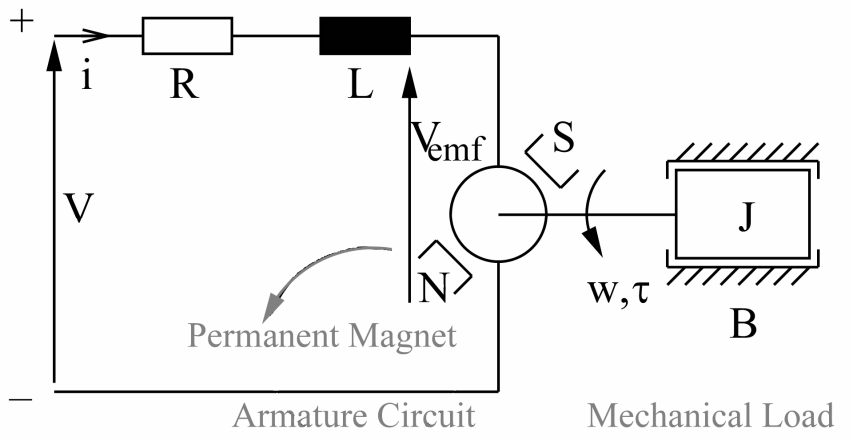


Figure 6.1 Permanent Magnet Dc Motor Block diagram

$$V(t) - V_{emf}(t) = L\frac{di(t)}{dt} + Ri(t) \tag{6.1}$$

$$T(t) = K_t i(t) \tag{6.2}$$

$$T(t) = J\frac{dw(t)}{dt} + Bw(t) + T_L(t) \tag{6.3}$$

$$V_{emf}(t) = K\ w(t) \tag{6.4}$$

where, $i, R, L, K_v, \omega, V, K_t, J, B, T$ are the armature current, armature resistance, inductance of armature coil, back electromotor force constant, rotational velocity of the armature, armature voltage, torque constant, inertia of the rotor, damping coefficient of mechanical system and torque of mechanical load respectively. Exploiting Laplace Transform and re organizing equations, system models are transformed to transfer functions form,

$$G_E(s) = \frac{T(s)}{V_a(s)} = \frac{K_t}{Ls + R} \tag{6.5}$$

$$G_M(s) = \frac{\theta}{T(s) - T_d} = \frac{1}{Js + B} \tag{6.6}$$

As it is seen in equations relation between applied voltage and motor speed is a second order differential equation consisting two cascaded systems. A simplifying assumption is that motor inductance taken as zero because it has a slight effect on system response when compared with pole of mechanical system.

$$\frac{w(s)}{V(s)} = \frac{K_t}{RJs + RB + K_t K_m} \tag{6.7}$$

6.3 Controllers

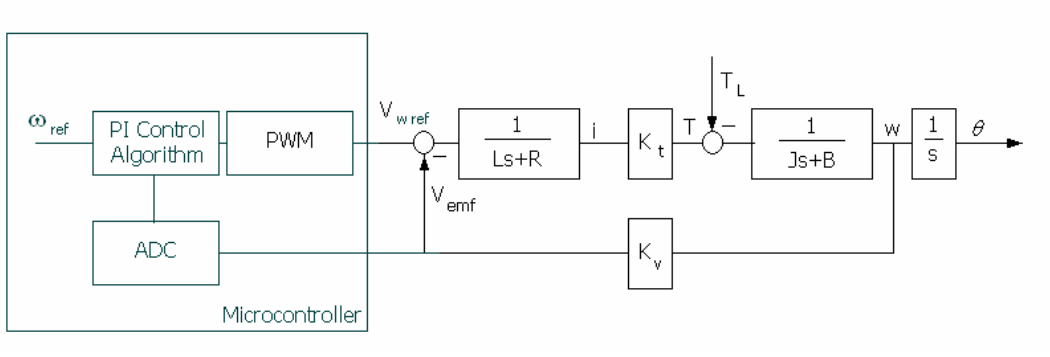
A feedback control scheme for mobile manipulator wheels are velocity control servo systems. The compensator has the transfer function:

$$G_c(s) = K_p + \frac{K_I}{s} \tag{6.8}$$

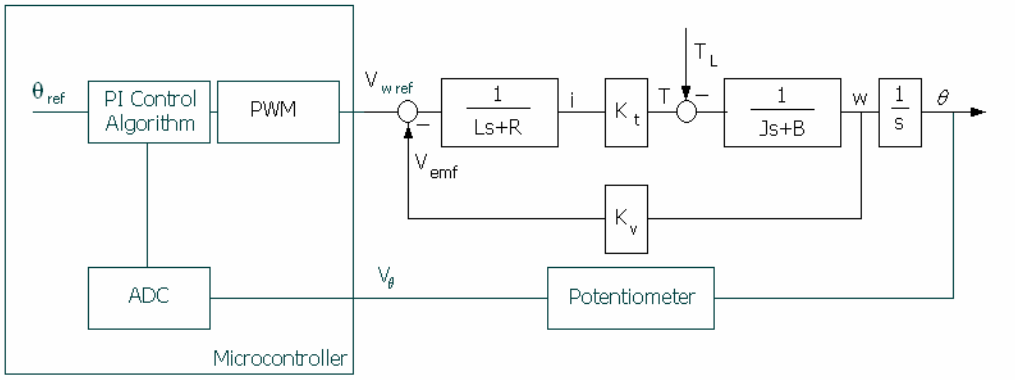
And it is a proportional integral (PI) compensator. Microcontroller implementation of this algorithm is done via back electro motor voltage estimation using measured values described in chapter two (equation 2.2) as velocity feedback and implementing numerical integration.

$$u(k) = K_p e(k) + K_i \sum_{i=1}^{k} e(i) \tag{6.9}$$

This computation strategy is applied as angular velocity control law for cart motors and position control law for manipulator controllers.



(a)



(b)

Figure 6.2 Permanent Magnet DC Motor Block diagram (a) Speed control loop (b) Position control loop

Table 6.1 Measured link angles and voltage values

The first link		The second link	
Angle θ_1 (radians)	Voltage (Bits)	Angle θ_2 (radians)	Voltage (Bits)
$\frac{\pi}{2}$	492	$-\frac{3\pi}{4}$	220
$\frac{\pi}{3}$	544	$-\frac{2\pi}{3}$	280
$\frac{\pi}{4}$	662	$-\frac{\pi}{2}$	400

In manipulator joint servo control routines, error driving compensator is computed via position feedback potentiometers. Standard 5 Volts are connected to the full resistance terminal of potentiometers. Between ground terminal and moving

tip of the potentiometer, a voltage value proportional to relative angular position of the links is acquired through analogue to digital converter of the controller. Angle to bits in 1024 bit ADC is obtained experimentally. Measured link angles and voltage values are used to find explicit expression which can be implemented in algorithms.

$$\theta_1 = 0.0087 \cdot V - 2.6331 \tag{6.10}$$

$$\theta_2 = 0.0044 \cdot V - 3.3175 \tag{6.11}$$

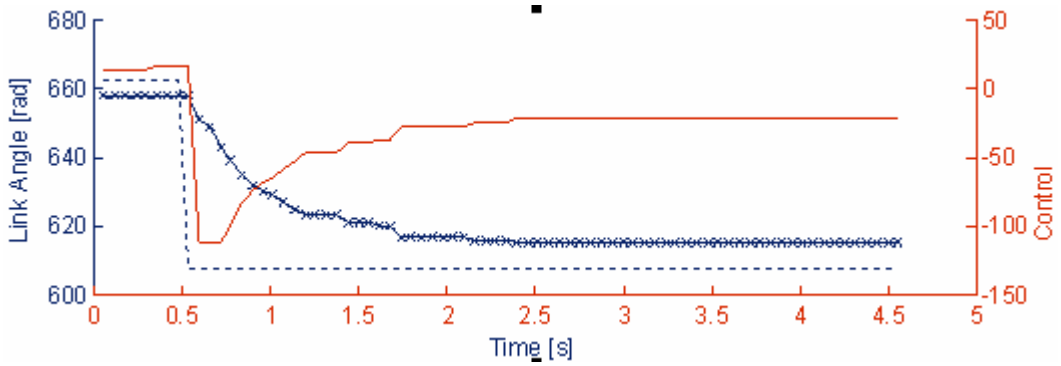
Resulting equations are used both in computer program and microcontroller program.

6.4 Effects of Mechanical System

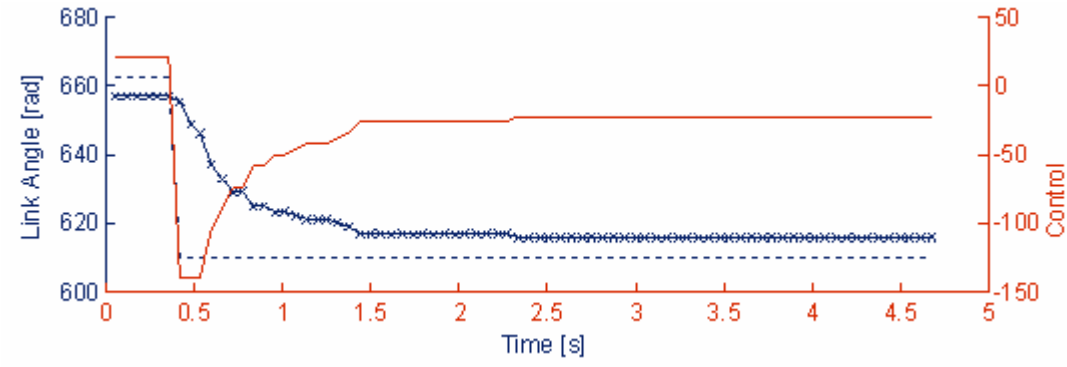
Parameters of mechanical system, inertia and damping ratio, affect system response directly. As the derivations given in dynamical modeling chapter are reconsidered, it becomes clear that robot arm joint actuator systems carry nonlinearities, owing to the fact that each link affected with centrifugal, Coriolis and gravitational torques which changes depending upon robot pose. On the other hand, wheel actuators are subjected to pose dependent inertia and time varying frictional forces according to floor type.

Besides, poor mechanical coupling between motor axes and joints and gear boxes causes nonlinear effects like dead zone and backlash. In existence of all mentioned effects and absence of a priori knowledge of the motor parameters. System can only be modeled approximately using time response curves.

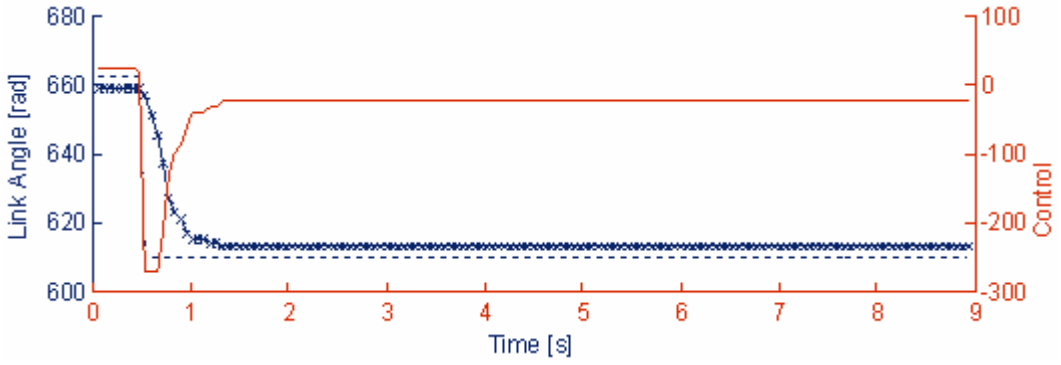
As time responses of the actuators are considered in context this study, overall performances of the controller are tuned experimentally. Experimental tuning procedure is defined as changing gain parameters of PI controller and observing step response of the actuator. In graphics, three variables are presented: joint reference with dashed line, measured value with crosses and control signal with solid line. Control signal is a signed PWM duty cycle ratio value, changing in range [-512,512] of bits corresponding to motor voltage in range [-12 , 12] of Volts.



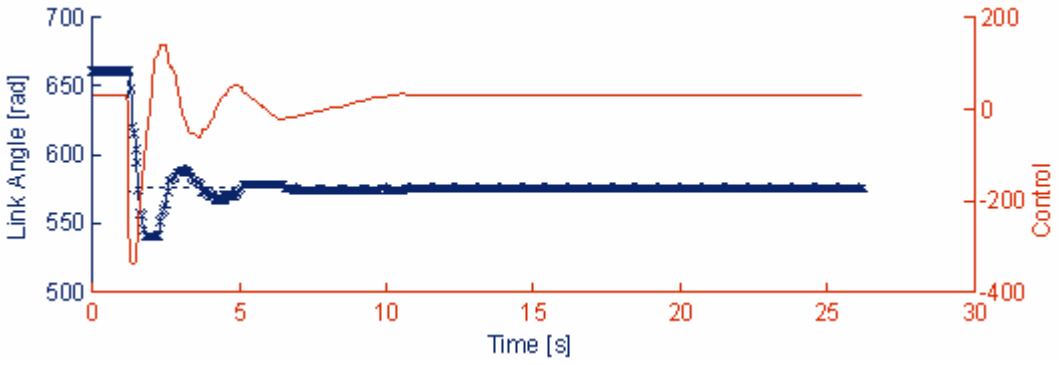
(a) K_p=800 (-- reference, x actual) and control signal (-)



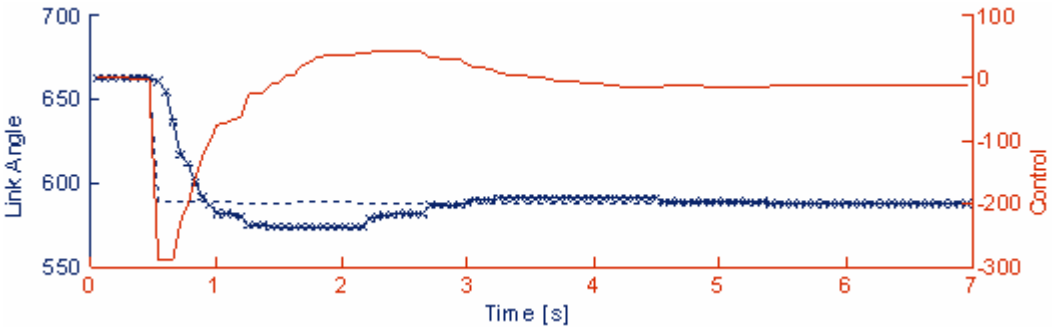
(b) K_p=1000 (-- reference, x actual) and control signal (-)



(c) K_p=2000 (-- reference, x actual) and control signal (-)

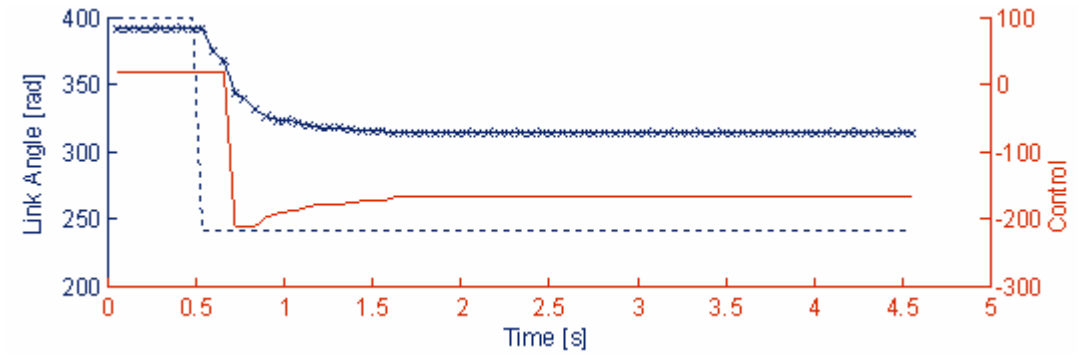


(d) K_p=1200 K_I=10 (-- reference, x actual) and control signal (-)

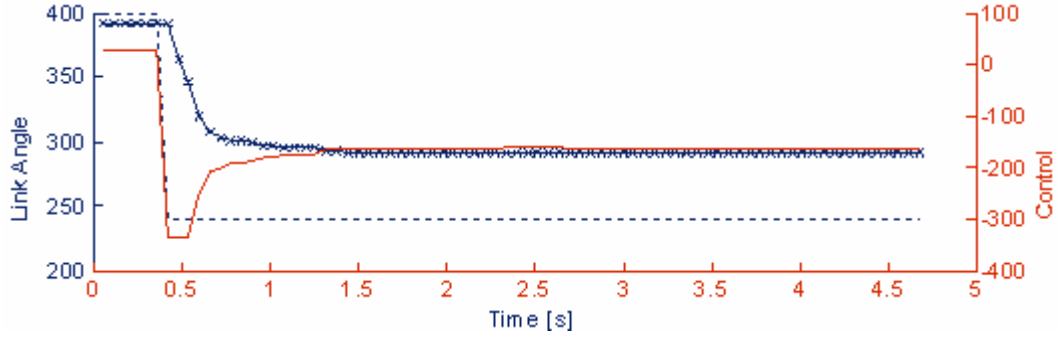


(e) K_p=1200 K_I=4 (-- reference, x actual) and control signal (-)

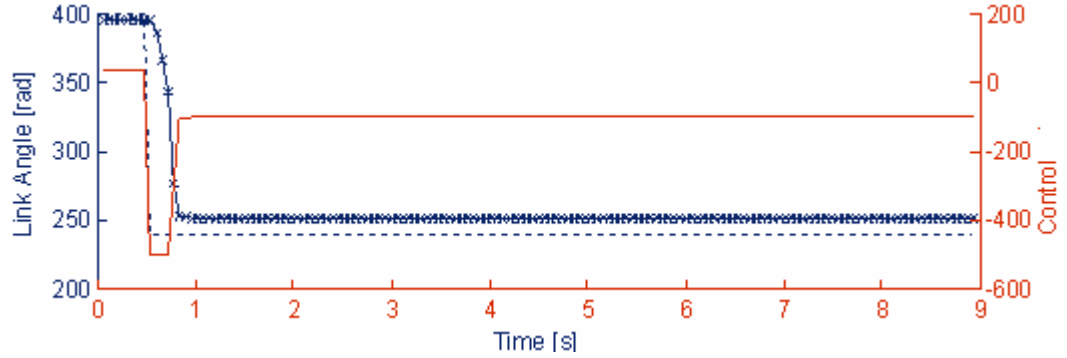
Figure 6.3 Steps of experimental controller parameter tuning (1st Link)



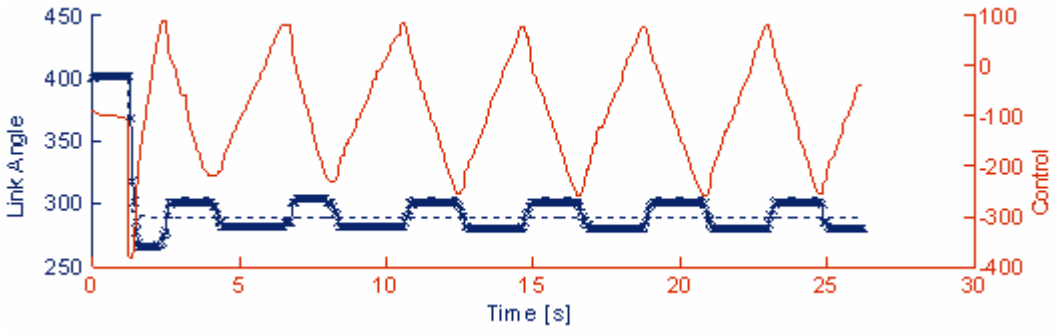
(a) K_p=800 (-- reference, x actual) and control signal (-)



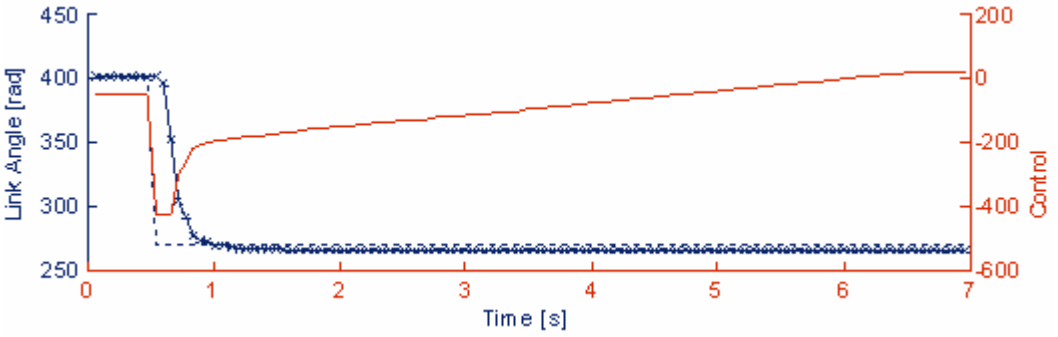
(b) K_p=1000 (-- reference, x actual) and control signal (-)



(c) K_p=2000 (-- reference, x actual) and control signal (-)

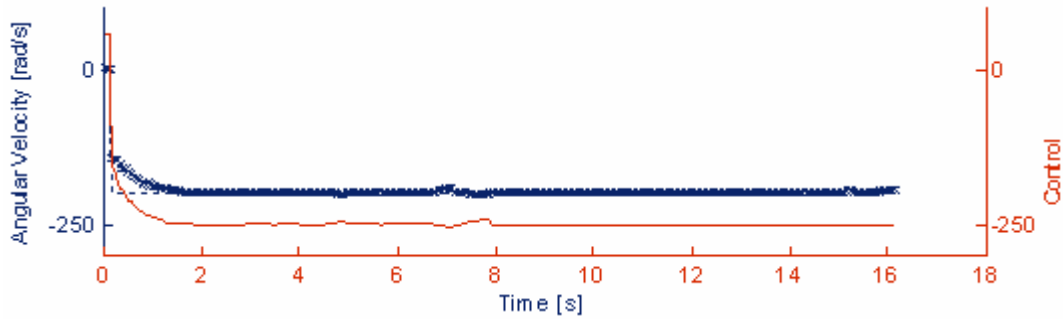


(d) K_p=1400 K_I=10 (-- reference, x actual) and control signal (-)

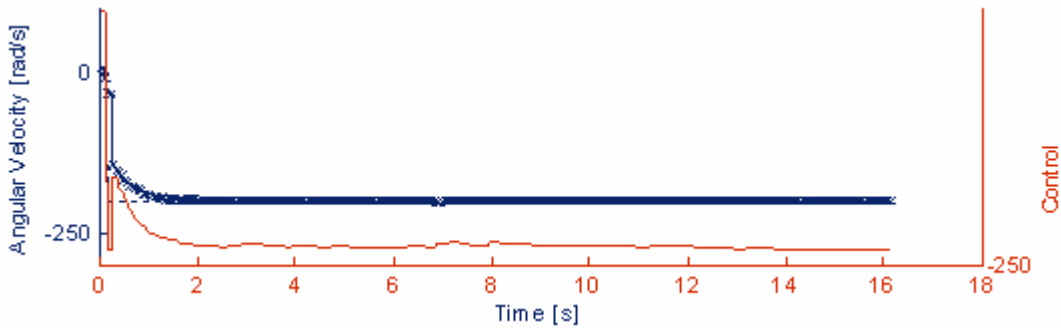


(e) K_p=1400 K_I=6 (-- reference, x actual) and control signal (-)

Figure 6.4 Steps of experimental controller parameter tuning (2nd Link)



(a)Left wheel K_p=400 K_I=10 (-- reference, x actual) and control signal (-)



(b) Right wheel K_p=400 K_I=10 (-- reference, x actual) and control signal (-)

Figure 6.5 Time response of the mobile platform actuators.

Parameters of controllers are defined in two steps. The first step is defining proportional control parameter by increasing it till system converges to a reasonable error value and control effect. As it is known that only proportional control does not eliminate steady state error, integral component is added in the second step of tuning process. In the next chapter developed algorithms are tested by exploiting found control parameters.

Gripper servos are hobby servo motors and they track position input and keep the given position even in the presence of external resistance through internal closed loop control. Input signal for hobby servos are PWM signals with pre specified timing.

Three different type of position servos are used in the system. Two relatively high torque motor used for roll and pitch motion. The third servo used for moving fingers to grip target.

This part of thesis is focused on low level actuator models and tuning of their controllers. Results of the experiments yields, position servo controllers for manipulator and velocity servo controllers for mobile platform. Dynamic effects of these controllers on mobile manipulator are concerned in the next chapter.

CHAPTER SEVEN
VISION BASED CONTROL OF MOBILE MANIPULATOR

7.1 Introduction

Extending capabilities of robot systems through artificial perceptual actions is one of the major goals of the scientist and engineers in field of robotics. One major clue to follow to achieve this goal is designing vision systems. Vision systems are, in most general sense, imitates human eye. (Coombs & Brown, 1993) By analogy it provides very important means both for recognition tasks (Ekvall, Kragic & Hoffmann, 2005) and environment interaction for a robotic system (Kragic & Christensen, 2001). Beside advanced image processing systems for recognition and data modeling for environment mapping, there exist researches which exploit visual data, extracted from images, to control actuators. Those systems are generalized in use for robotics systems where visual functionality improves the area of use of such systems.

A mobile manipulator consists of an articulated arm mounted on a mobile platform. Since this mechanical arrangement combines the dexterity of the former with the workspace extension of the latter, it is clearly appealing for many applications. The locomotion of the platform is typically obtained through wheels. Most wheeled vehicles are subject to nonholonomic constraints arising from the rolling without slipping of the wheels on the ground (related derivations are given in chapter three). These constraints restrict the admissible generalized velocities and thus the instantaneous mobility of the platform, but do not affect the global accessibility of its configuration space, which is still guaranteed through suitable maneuvering with a reduced set of independent velocity commands (Laumond, 1998). On the other hand, manipulators are kinematically unconstrained systems; arbitrary joint velocities can be imposed at each configuration.

Control of the mobile manipulators is generally based on combined kinematical analysis of the both subsystems (Hootsmans & Dubowsky, 1991). Developed

methodologies for computing actuator commands for such systems allow them to follow desired end-effector and platform trajectories without violating the nonholonomic constraints. Based on mathematical manipulations on the system dynamics, model-based controllers are designed to eliminate tracking errors (Papadopoulos & Poulakakis, 2000) (De Luca, Oriolo & Robuffo Giordano, 2006).

Like any other configuration, vision based control systems are exploited to improve performance of mobile manipulators in unstructured environments. These controllers primarily focus on deriving jacobian matrix which relates perceived task space features to mobile manipulator control inputs (De Luca, Oriolo & Robuffo Giordano, 2007). In this study control of non holonomic mobile manipulator is done by task sequencing. Any positioning task is realized in two stages: Mobile platform moves till target is placed in manipulator workspace and in proper orientation that gripper (tracking) can hold, afterwards manipulator moves to hold target (handling).This discrimination reduces the control problem two sub problems: control of the mobile platform and control of manipulator (A preliminary implementation scheme is given at the end of this section).

As it is described in previous chapters, design of a visually controlled robotic system involves set of choices both in hardware means and use of models to achieve desired tasks. Proposed vision based control algorithm is a hybrid approach to control and perception relation. Gripper and mobile platform motion is controlled with visual servoing; manipulator arm motion is controlled with a dynamic look and move methodology.

Robotics system includes a stereo setup which estimates 3D point in manipulator coordinate frame and calculates image features for motion control. This setup is fully calibrated and implements stereo triangulation. The each camera of the stereo system is placed both sides of the manipulator and monitoring forward motion space of the mobile platform and workspace of the manipulator arm. The advantage of this setup is that when a target point is extracted, the result corresponds to task space error for mobile platform and desired task space position for wrist point on manipulator arm.

Mobile platform controller is another hybrid controller. Input to this controller is Euclidian distance to target and an image plane deviation corresponding to angular displacement between gripping point and car direction. Manipulator controller is a standard setup for two degrees of freedom manipulator but its references are updated instantaneously according to vision data. Gripper motion is also directly controlled via image feature vectors and resultant manipulator configuration.

Task defined for this study is picking up a target object via visual feedback in task space. Target object used in experiments is a LED torch. Expected motion of the robotic system is execution of following tasks, sequentially.

- Low level image processing on images from left and right camera and get centroid of detected projection of the target.
- Stereo triangulation for estimating position of the target in camera coordinate frame.
- Calculate wheel velocities for differential drive platform by using kinematic model of the system
- Check condition if cart is oriented properly according to target so that gripper of the planar manipulator can pick it. If so, flag orientation check.
- Check condition if cart is placed properly according to target so that it is in the workspace of the planar manipulator. If so, flag reached check.
- Perform inverse kinematics of robot manipulator from instantaneous data from vision system if the cart is oriented and target is reachable.
- Update all actuator references and grip the target.

In the following sections, implementations of described sub tasks are explained and experimental results are presented.

7.2 Stereo Tracking System

Context of this section is a subsystem of a mobile manipulator control algorithm. This system solves stereo triangulation for calculating distance to target in left

camera frame and performs a transformation to convert found coordinates to robot motion reference frame and eventually solves inverse kinematics equations to obtain references for joint controllers.

The increase in rate of microprocessors is triggered use of visual perception systems in wide range of applications from graphic animations (Kaufman, Edmunds & Pai, 2005) to robotic systems (Kriegman, Triendl & Binford, 1989).

Vision based control loops have been introduced in order to increase the flexibility and the accuracy of robotic systems. The aim of such systems is to control an electromechanical system exploiting the information provided by a vision system. (Corke, 1994) A general classification method for vision systems is on the number of cameras (Hartley & Zisserman, 2000): Single camera vision systems and multi-camera vision systems.

Single camera is often used in applications when an a priori known pose feature set is available. Camera generates an output which is instantaneous feature vector pose and system acts according to error vector in feature space (Ohya, Kosaka & Kak, 1998).Single camera systems are deficient in generating feature measurements in 3D space if they are static. If motion of the camera generates two or more image of same feature pose, it is also possible to make 3D estimations about position of the features. But this is the case where relative motion of the target and camera must be fast enough to take same pose in two different view angles. Another fallback using a mono camera setup in 3D reconstruction is system highly dependent on odometry system which also carries a certain amount of error. If the problem to solve with vision system is 3D reconstruction, stereo pairs or systems with more cameras are preferable since they are avoiding the fallbacks of single camera system. On the other hand, using two cameras in a stereo configuration (two cameras with a common field of view) increases cost of the system not only in monetary sense but also in computational time cost in single operation step.

3D reconstruction problem is a two stage task. The first task is to find projections of same features in each image plane: Correspondence. The second task is to calculate an estimation 3D position with correspondent feature vectors. Correspondence problem is a very complicated since its nature is statistical and dynamical. A large group of researchers are appealed this problem and some standard algorithms are also available (Hartley & Zisserman, 2000). But in this study a simple scene construction is preferred since the problem attacked is to generate motion references for mobile manipulators. Calculating 3D position is relatively simple problem and three cases as to available information about hardware configuration.

This study develops an alternative calculation method for stereo triangulation method. Proposed method uses planar projection of points to calculate cross section of rays from cameras whereas standard algorithm uses and 3D vector representations and least mean square calculation (Trucco & Verri, 1998)

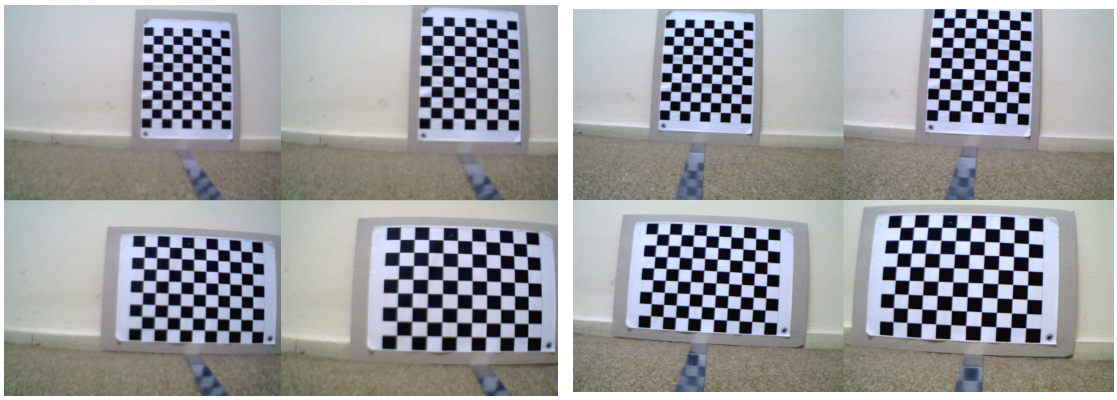
In the following, conventional stereo triangulation for calibrated cameras is described first; afterwards new solution modification on this algorithm is presented. Results are given at the end of stereo vision section of this study.

Following calculations are related with the 3D reconstruction part of stereo vision algorithm where a feature point is detected in both images. A generic approach on 3D reconstruction is initiated with covering knowledge of camera setup. As it is known from pinhole camera model representation of imaging process, a single camera is subjected to effect of two matrices. One of these matrices is formed with intrinsic parameters of the camera and the other is a representation of pose of the camera as extrinsic parameters. (Pose is represented well with a homogenous transformation matrix.). If both matrices are available for calculations, it is possible to find a 3D estimate of feature point with an ambiguity only form quantizing and spatial sampling effects of overall setup (Trucco & Verri, 1998). Process to gain information on intrinsic and extrinsic parameters is camera calibration.

7.2.1 Camera Calibration

Camera calibration is a routine to find out the quantities that affect the imaging process: Position of image center in the image, focal length, different scaling factors for row pixels and column pixels, skew factor, lens distortion and translation and rotation of the feature point. From a mathematical point of view, camera calibration is solving inverse of the imaging equations which are nonlinear. As an general approach, all calibration methods stand on imaging of a calibration pattern with known geometry Although various methodologies are developed as a solution procedure, most used method is developed by Tsai in 1987. Tsai method uses a planar calibration target with a checker board pattern. Camera parameters are estimated using extracted corners from calibration grid which is imaged from different view points. This standard algorithm is implemented by using Camera Calibration Toolbox for Matlab (Bouguet, 2007).

To calibrate cameras a checker board pattern with known square dimensions are imaged from different views. Since it is intended to calibrate a stereo setup, two cameras are mounted on mobile platform and the calibration sets of images are obtained.



(a) (b)

Figure 7.1 Sample images from left (a) and right camera (b) for camera calibration

Using two sets of images for calibration of right and left cameras, matlab code solves calibration parameters given in chapter four. If the resulting camera

calibration matrices are once more used for calibration, concerning they are placed at a constant pose relative to each other, Parameters for left and right camera images and stereo setup is found. (Parameters are described in chapter four)

Table 7.1Camera calibration results

parameter	Left Camera		Right Camera
Focal Distance (pixels)	[726.11070 725.97108]		[749.31536 751.04768]
Central Point (pixels)	[327.31297 264.42672]		[308.20238 251.60095]
Skew (radian)	0		0
Radial distortion	-0.43031 0.45677 0.00322 -0.00026		-0.35503 0.00000 0.00531 0.00145
	Stereo Setup		
Translation (mm)	-110.37198	-3.80882	4.79475
Rotation (radians)	0.01138	-0.00432	0.02229

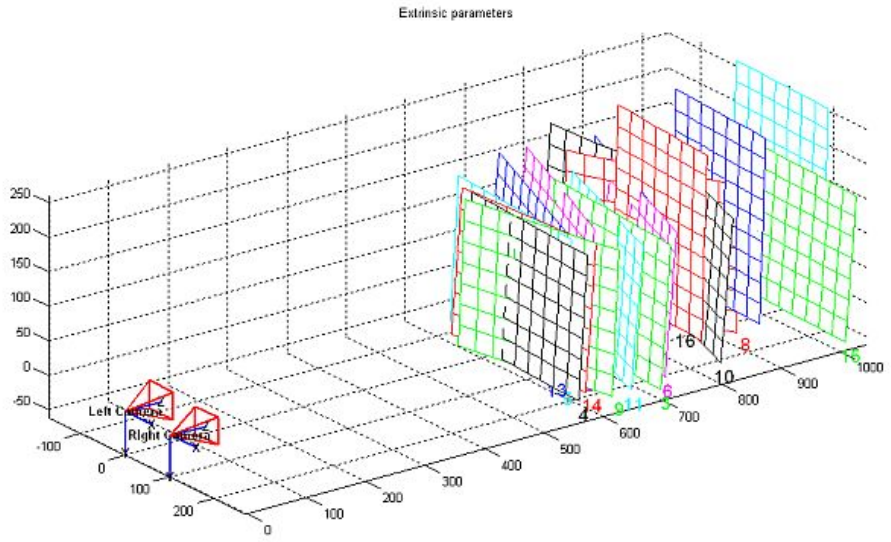


Figure 7.2 Extrinsic parameters of stereo setup

Using calibration procedure and robot kinematics, geometrical relationships among fundamental frames and camera parameters are obtained. Standard stereo vision 3D reconstruction algorithm exploits this information in a following manner.

7.2.2 Triangulation Algorithm for Calibrated Binocular Stereo Rig

As a standard algorithm, it is meant that stereo triangulation algorithm in OpenCV. Two corresponding position vectors of same point in camera reference frames, P_L and P_R, and image coordinates of the point P, p_L and p_R, evidently provides a solution for triangulation equations. The 3D world reference system is chosen coincident with the left camera reference system. The right camera is translated and rotated with respect from the left one.

Vectors P_L and P_R are related to each other with rigid body motion equations if rotation matrix R and translation between cameras coordinate frames are known.

$$P_L = {}_L R^R P_R + {}_L T^R \tag{7.1}$$

Recalling perspective projection equations in pinhole camera models, image points p_L and p_R can be written in form:

$$\begin{aligned} P_L &= z_L p_L \\ P_R &= z_R p_R \end{aligned} \tag{7.2}$$

Placing these equations in (7.1) leads following relation:

$$z_L p_L = {}_L R^R z_R p_R + {}_L T^R \tag{7.3}$$

This relation describes the first relation (equation 7.1) in a manner that it is possible to find z coordinate, namely depth, of real world point in camera coordinate frames if two corresponding image points for a stereo pair are given.

$$\begin{bmatrix} -{}_L R^R p_R & p_L \end{bmatrix} \begin{bmatrix} z_R \\ z_L \end{bmatrix} = {}_L T^R \tag{7.3}$$

$$A = \left[-{}_L R^R p_R \quad p_L \right] \tag{7.4}$$

Matrix A is rectangular in form 3x2 and matrix equation gives over determined system with three equations for two unknowns which leads least square solution for $[z_R \; z_L]^T$. this is a natural issue arising from the fact that intersection of the 3D lines are highly sensitive to extracted position of line points. On the other hand calibration procedure gives an estimation of all parameters not exact parameters, so it is obvious that search of intersection point is converted to search of a closest distance between two lines as stated in Bouguet, 2007.

$$\begin{bmatrix} z_R \\ z_L \end{bmatrix} = (A^T A)^{-1} A^T {}_L T^R \tag{7.5}$$

Right camera frame z coordinate for given point is explicitly found via following equation

$$z_R = \frac{\|p_L\|^2 \langle -{}_L R^R p_R, T \rangle - \langle -{}_L R^R p_R, p_L \rangle \langle p_L, T \rangle}{\|p_L\|^2 \|-{}_L R^R p_R\|^2 - \langle -{}_L R^R p_R, p_L \rangle^2} \tag{7.6}$$

where ||.|| is Euclidian norm operator and <.,.> is scalar product operator. If the derivations are concerned in a wider aspect standard triangulation solution is implementation of following layout:

- Normalization of extracted point coordinates (p_L, p_R) by subtracting them from image center (o_x, o_y) and dividing by focal distance.
- Elimination of radial disturbances using Oulu University model(Heikkila & Silven, 1997)
- Calculation of equation 7.6
- 3D point coordinates in right camera frame are calculated by recalling equation 7.2.

As it is seen equation d explicit expression of z_R is complicated and requires computation of some transforms and scalar products. As another approach to triangulation derivations, a methodology based on planar projection of 3D points are developed.

7.2.3 A Planar Projection Modification on Stereo Triangulation Algorithm

Applying first two steps of the given layout, undistorted normalized points are found. The developed algorithm recalculates undistorted image points by scaling normalized points with focal distance. Then undistorted image point in camera coordinate frames are:

$$p_{L,u} = \begin{bmatrix} f_L x_{L'u} \\ f_L x_{L,u} \\ f_L \end{bmatrix} \tag{7.7}$$

$$p_{R,u} = \begin{bmatrix} f_R x_{R'u} \\ f_R x_{R,u} \\ f_R \end{bmatrix} \tag{7.8}$$

These two points are lies on projectiles from left and right camera center, respectively. To calculate an intersection point for these two projectiles, all points must be expressed in same coordinate frame. The left camera coordinate frame is selected as a reference coordinate frame and two camera center origins and undistorted image points are found using extrinsic parameters from camera calibration.

$${}^{L}O_L = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.9}$$

$${}^{L}O_R = {}^{L}T_R = {}^{L}R_G({}^{G}T_R - {}^{G}T_L) \tag{7.10}$$

$$ {}^{L}p_{L,u} = \begin{bmatrix} f_L x_{L'u} \\ f_L y_{L,u} \\ f_L \end{bmatrix} \tag{7.11} $$

$$ {}^{L}p_{R,u} = {}^{L}R_R \begin{bmatrix} f_R x_{R'u} \\ f_R y_{R,u} \\ f_R \end{bmatrix} \tag{7.12} $$

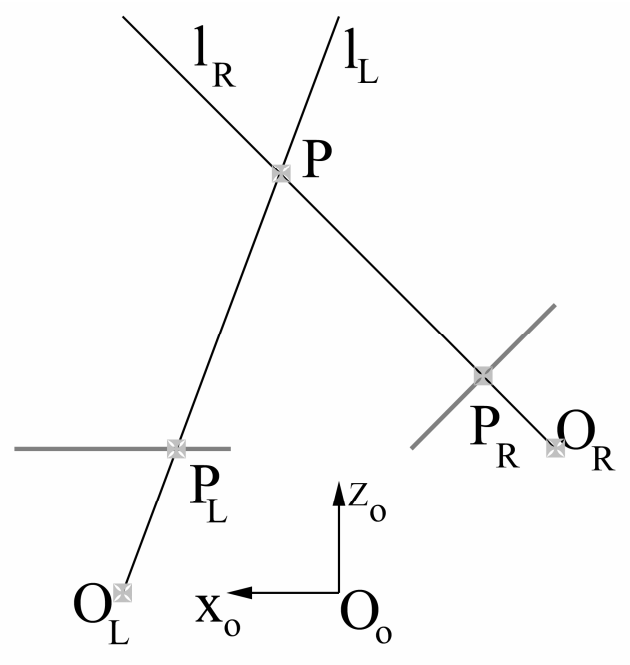


Figure 7.3 Extrinsic parameters of stereo setup

Formation of the points is similar to standard stereo triangulation algorithm but they are not normalized. In this algorithm to calculate intersection point of two projectiles, XZ plane projections of 3D rays are used. These planar points are converted to homogenous coordinates by adding unity as a third component.

$$ O_L{}' = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{7.13} $$

$$ O_R{}' = \begin{bmatrix} \{ {}^{L}O_R \}_{XZ} \\ 1 \end{bmatrix} \tag{7.14} $$

$$p_L' = \begin{bmatrix} f_L x_{L'u} \\ f_L \\ 1 \end{bmatrix} \tag{7.15}$$

$$p_R' = \begin{bmatrix} \{{}^L p_R\}_{XZ} \\ 1 \end{bmatrix} \tag{7.16}$$

Using advantages of homogenous coordinates, line equations from left and right points are calculated from cross product of homogenous points.

$$l_R = o_R \times p_R \tag{7.17}$$

$$l_L = o_L \times p_L \tag{7.18}$$

Cross section of these two lines gives coordinates of 3d world point corresponding to two image points from left and right camera.

$$\begin{bmatrix} \{P\}_{xz} \\ 1 \end{bmatrix} = l_L \times l_R \tag{7.19}$$

Using the second component of resulting vector as z_L and recalculating perspective projection, coordinates of the point P is found in left camera frame. Test of the algorithms are done in Matlab. Images from calibration sets are used to demonstrate relative errors in two algorithms.

At this point of calculations stereo 3D reconstruction problem seems to be solved but from implementation point of view one more issue should be taken into account. The origin of the left camera coordinate frame is only known with respect to a grid coordinate system and grid is not a part of the system any other positioning operations due to robotic manipulation can not be resolved neither in left camera frame nor in grid reference frame. A reasonable choice to place origin of the free reference frame described in the initial part of the study is the origin of task space of the robot. This enables forward and inverse kinematics mappings to use stereo vision results directly. But this implementation requires definition of the transformation

from left camera frame to robot base coordinate system, which is the coordinate reference system of task space.

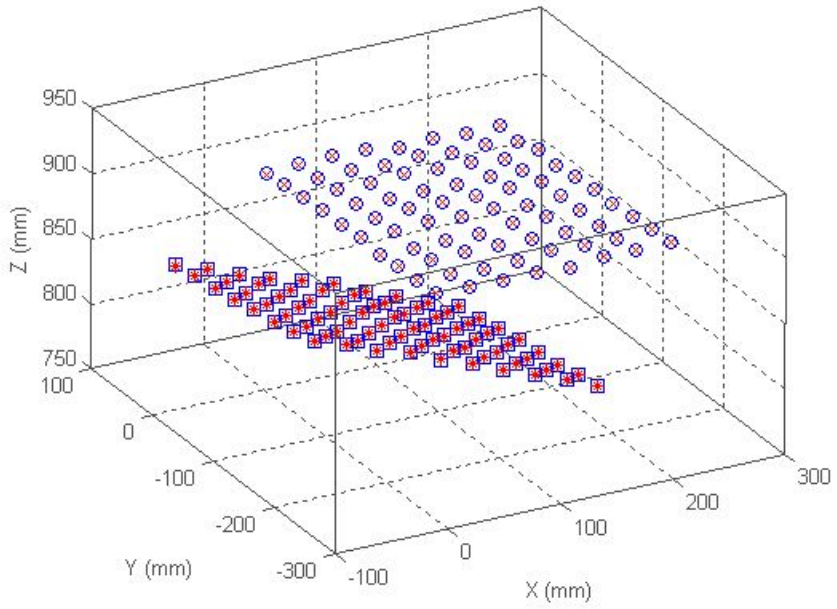


Figure 7.4 Verification of stereo triangulation results (o,□ points to standard algorithm results whereas x,* shows results of developed algorithm)

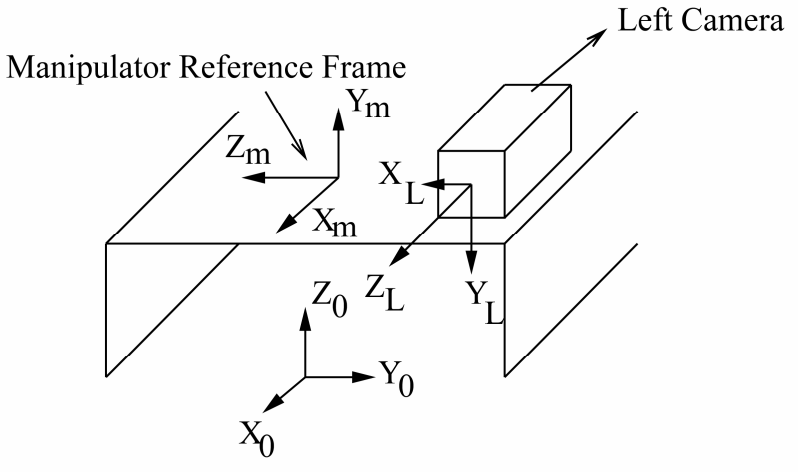


Figure 7.5 Left Camera and robot coordinate frame relation

As mentioned previously placing calibration grid at a known pose with respect to robot base frame gives grid to robot transformation matrix. This matrix can be obtained from measurements either through link angles and forward dynamics

solutions or task space measurements. Exploiting achieved result leads desired transformation.

$$ {}^{m}H_{L} = {}^{m}H_{G}\,{}^{G}H_{L} \tag{7.20} $$

$$ {}^{m}P = {}^{m}H_{L}\,{}^{L}P \tag{7.21} $$

Implementation of stereo triangulation with the extension given in above equations expresses an image feature point in task space of the robot

$$ {}^{m}H_{L} = \begin{bmatrix} 0 & 0 & 1 & 82 \\ 0 & -1 & 0 & -70 \\ 1 & 0 & 0 & -107 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.22} $$

It is previously mentioned before that motion of the cart and gripper does not only depend on the world coordinates of the target point. They are also related to features on image plane. The first feature is centroid of target image and the second feature is image plane orientation.

In this section, the derivation and implementation scheme of vision based task space feedback is represented. In the following how this feedback is exploited to control mobile manipulator motion will be in focus.

7.3 Control of Mobile Platform

A central issue in mobile manipulator control search is control of non holonomic differential platform. Presence of rolling without slip condition forces system trajectories to be controlled suitable manner to be achievable (Laumond, 1998).

A mobile platform, which is located on 2D plane, possesses 3 degrees of freedom in the world coordinates

$$p_c = \begin{bmatrix} x_c \\ y_c \\ \theta_c \end{bmatrix} \tag{7.23}$$

This posture represents planar position and robot orientation in a right handed coordinate frame. For a given path in a plane, posture p is a function of time and follows entire locus of point $[x_c(t)\ y_c(t)]^T$. If time derivatives of trajectory points exist, then $\theta_c(t)$ is not an independent variable.

$$\theta_c(t) = tan\left(\frac{\dot{x}_c}{\dot{y}_c}\right) \tag{7.24}$$

Motion of the differential drive platform is governed via linear velocity and rotational velocity, which are also functions of the time. The vehicle kinematics is defined with an equation (derivations are given in chapter three):

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix} = \dot{p}_c = J_c \dot{q}_c = \begin{bmatrix} cos\,\theta & 0 \\ sin\,\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{lc} \\ w_{lc} \end{bmatrix} \tag{7.25}$$

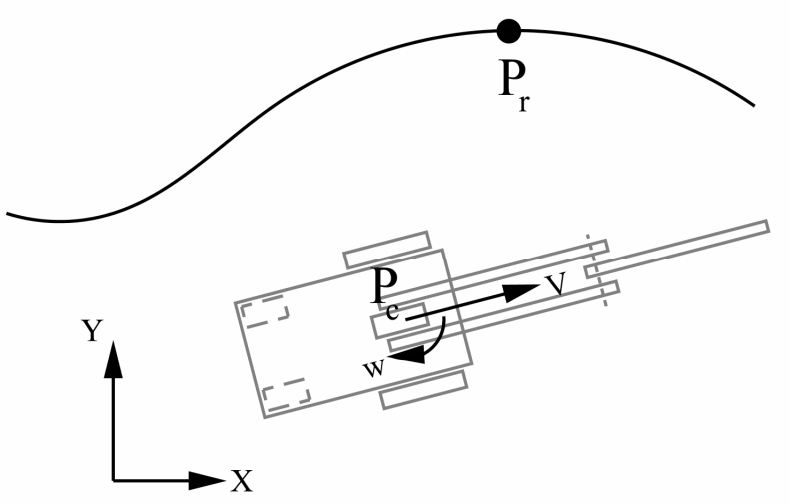


Figure 7.6 Cart control parameters and reference and current postures

Vector q defines the variables to control in order to cause change in pose of differential drive mobile platform. Defining a control law for this system, like in any system in sense of feedback control, needs representation of error dynamics so that control problem becomes regulation problem of error variables in kinematical model. To obtain an error posture (Kanayama, Kimura, Miyazaki & Noguchi, 1990) reference posture and current posture is to be defined. Reference posture p_r is goal posture of the differential drive platform and current posture p_{ci} is instantaneous posture of the cart. Error posture is a transformation of the reference posture in a local coordinate system with an origin $[x_{ci}\ y_{ci}]^T$ and x- axis in direction of θ_{ci}.

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta_c & \sin\theta_c & 0 \\ -\sin\theta_c & \cos\theta_c & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix} - \begin{bmatrix} x_{ci} \\ y_{ci} \\ \theta_{ci} \end{bmatrix} \right) \tag{7.26}$$

Last equations enlighten important ability in sense of overall robotic system. Cameras are mounted on differential drive platform and their posture is the same as the posture of differential drive platform. On the other hand reference posture of the cart is defined with respect to target position which is also acquired through cameras. It is important to note that orientation of the cart is remained uncontrolled since target reference is defined as a point on the plane.

Orientation of the differential drive platform is also important in guidance problem of mobile manipulators because it is also necessary to orient itself to target in a proper manner to permit manipulator to execute a pick up task.

Orientation error is defined in left image plane directly because prime focus on control is to align manipulator w.r.t target. Reference position of the gripper with respect to left camera stays on a static line in image plane. Using a learning methodology, gripper holding the target is monitored through vision system and reference position is defined.

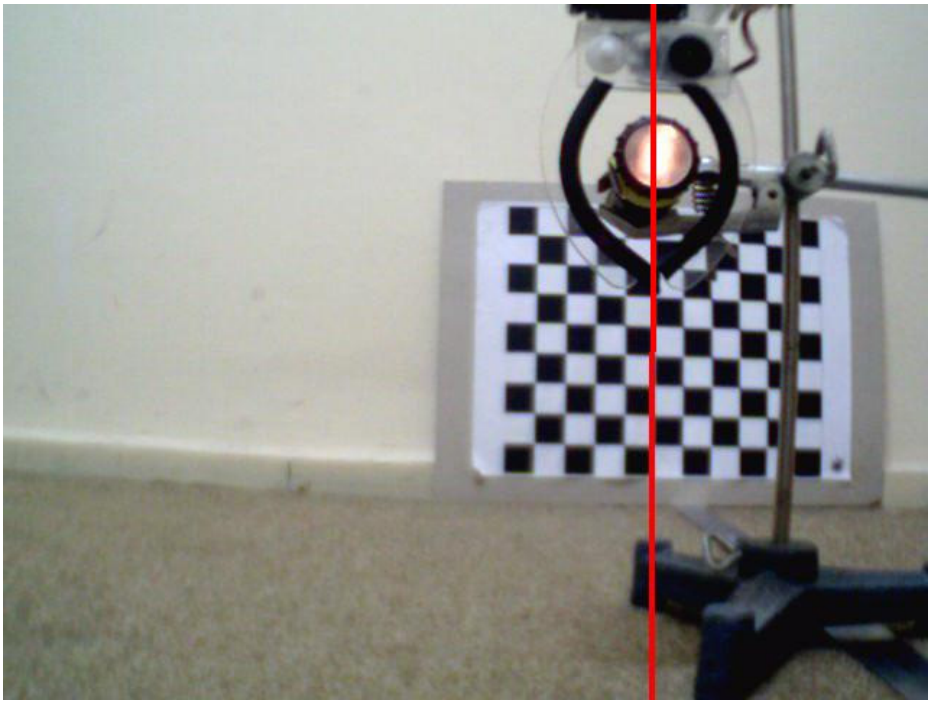


Figure 7.7 Gripper reference position

Using result of image processing data and reference position an estimate of the orientation error is found.

$$\hat{\theta}_e = \begin{bmatrix} 1 & 0 \end{bmatrix} \left(\begin{bmatrix} x_i \\ y_i \end{bmatrix}_{inleftimage} - \begin{bmatrix} x_r \\ y_r \end{bmatrix} \right) \tag{7.27}$$

Mentioning that this is a very rough estimate, it is not convenient to use orientation error in Kanayama transformation. Hence it is necessary to find a control law including error dynamics of orientation and Kanayama transformation resulting posture.

If kinematics of the cart is recalled, it is seen that two velocities affect the system. If linear velocity defined as a function of Euclidian distance to target and rotational velocity is through orientation estimate on image plane. Resulting system is a combination of error posture equation and image based orientation error representation. Simple proportional controller is defined as following:

$$w_c = K_{\theta e}\hat{\theta}_e \tag{7.28}$$

$$v_c = K_{ae} l_e = K_{ae}\left(\left\|\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x_e\\ y_e\\ \theta_e\end{bmatrix}\right\| - d_r\right) \tag{7.29}$$

Where d_r is distance defined according to dexterous workspace of the planar manipulator. One more transformation is necessary to convert these velocities to wheel angular velocities.

$$\begin{bmatrix}w_l\\ w_r\end{bmatrix} = \begin{bmatrix}\frac{1}{r_w} & -\frac{L}{2r_w}\\ \frac{1}{r_w} & \frac{L}{2r_w}\end{bmatrix}\begin{bmatrix}v_c\\ w_c\end{bmatrix} \tag{7.30}$$

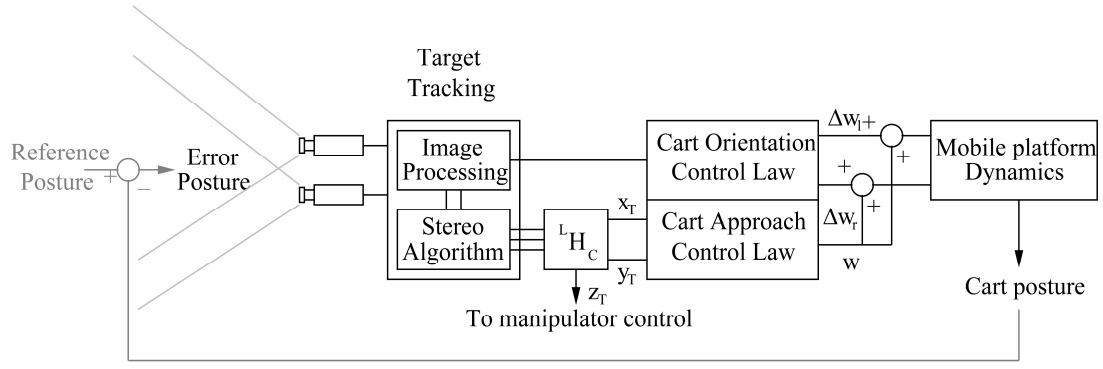


Figure 7.8 Mobile platform control algorithm

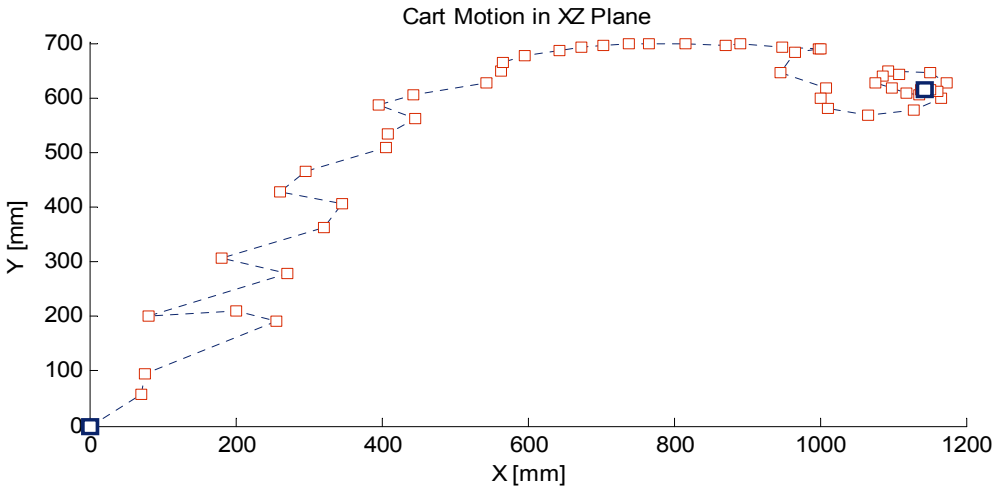
Following the given decomposition, an image plane control law is developed for cart orientation control. Centroid of the target projection (C_T) on left image is used as feedback value corresponding to angular misalignment. Image plane is divided into two parts. The first part is a neighborhood of value "width" in pixels. If the feedback value is in this region a proportional control law is calculated. The second part of the image left is remained in saturation zone of controller.

$$\Delta w = \begin{cases} \Delta w_{max} & \dfrac{C_T - C_r}{width} < -1 \\ \dfrac{C_T - C_r}{width} * \Delta w_{max} & 1 \geq \dfrac{C_T - C_r}{width} \geq -1 \\ \Delta w_{max} & \dfrac{C_T - C_r}{width} > 1 \end{cases} \tag{7.31}$$

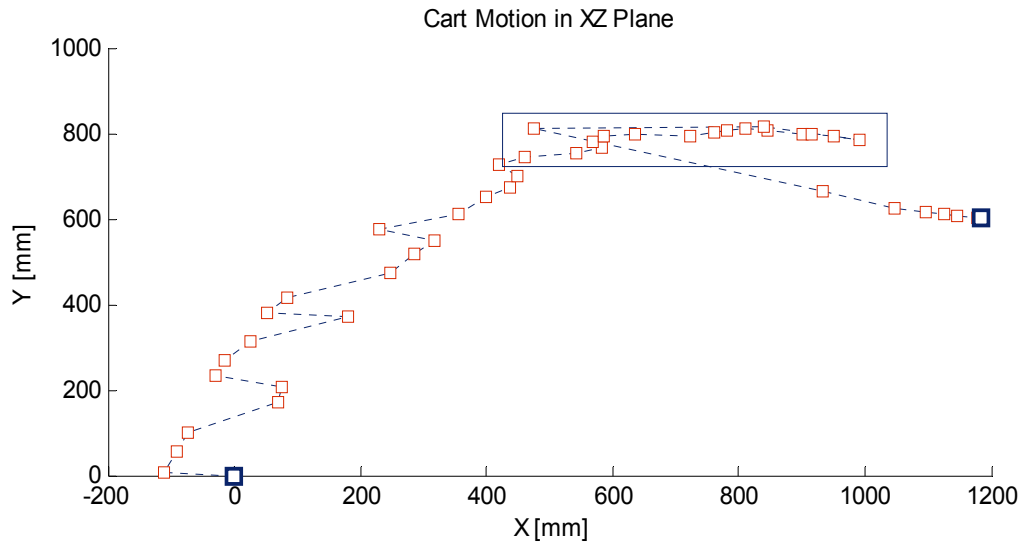
This control law turns the developed control strategy to an image level abstraction and tuning of the controller is done via design parameters width and Δw_{max}.

In a very similar manner, linear velocity of the cart is found by applying equation a in a neighborhood of reference value d_r. Here the reference value is a distance to target which is a smaller than maximum reach of the manipulator. Linear velocity controller works only for positive error values because cancellation condition of the controller is not place mobile platform on a certain position but placing target in the workspace of manipulator. Width and w_{max} are the parameters to determine controller performance.

$$w = \begin{cases} \dfrac{d_T - d_r}{width} * w_{max} & \dfrac{d_T - d_r}{width} \leq 1 \\ w_{max} & \dfrac{d_T - d_r}{width} > 1 \end{cases} \tag{7.32}$$



(a) Mobile Platform trajectory for low level actuator gains (K_p=200,K_I=10) (rectangles: measured positions bold rectangles: references)



(b) Mobile Platform trajectory for low level actuator gains (K_p=400,K_I=10) (rectangles: measured positions bold rectangles: references)

Figure 7.9 Trajectories of mobile platform controlled with developed algorithm. In (b) non smooth motion of the mobile platform becomes evident.

When motion of the mobile platform is analyzed carefully, effect of the latency in the vision system becomes evident. When mobile platform moves relatively fast, observed motion demonstrates a strong deviation from target and angular velocity effect in motion becomes evident.

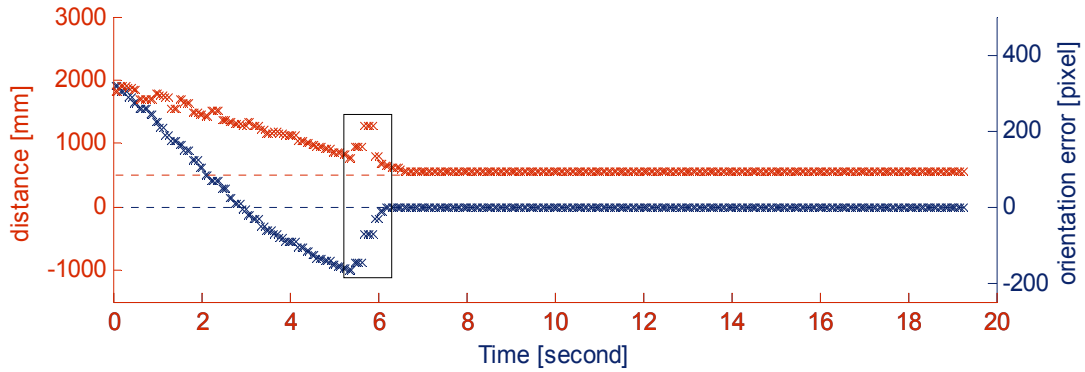


Figure 7.10 Error dynamics of the second experiment. Marked region shows unexpected error deviation.

Latency problems in vision based control systems are reported from the early researches. Several procedures to overcome are developed like Kalman Filters and

steady state Kalman Filters (Hashimoto, 2003), but all proposed setups have extra hardware possibilities to compensate additional computing time.

In this study, a linear prediction methodology is adopted. Latency time is measured first. This time gives the time between two consequent outputs of the stereo vision system. Using this two value and measured time, spatial velocity estimation is obtained. This velocity value is used to update position estimates from stereo vision.

To design an appropriate prediction routine, timing of two processes in Momac program should be considered. Image processing routine updates in every 180 ms task space feedback and controller routine updates actuator references in every 60 ms.

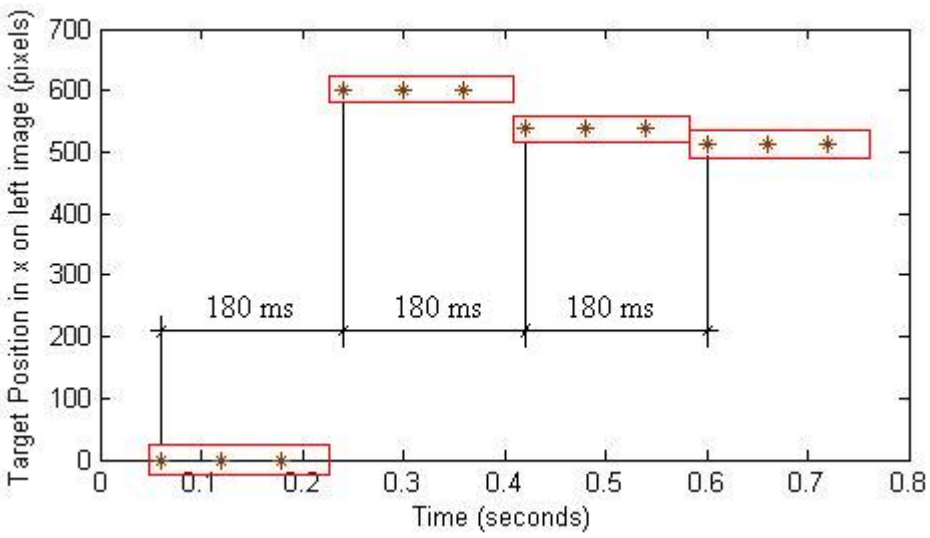


Figure 7.11 Measurement of latency in vision system

Using experimental results and velocity derivation given above, algorithm for predicting next step position is through steps given below:

For every time span corresponding 3τ calculate spatial velocity:

$$\hat{v}(t) = \frac{p(t-3\tau) - p(t-6\tau)}{3\tau} \tag{7.33}$$

Then update position estimation for the same time instant

$$\hat{p}(t) = p(t-3\tau) + p(t-3\tau) - p(t-6\tau) \tag{7.34}$$

For the time instants between next vision system update calculate:

$$\begin{aligned} \hat{p}(t+\tau) &= p(t) + \hat{v}(t)\tau \\ \hat{p}(t+2\tau) &= p(t+\tau) + \hat{v}(t)\tau \end{aligned} \tag{7.35}$$

These calculations are made in left camera coordinate space and result is predicted target position. But it must be noted that cart orientation control is also dependent upon image feature point. Therefore, one more equation is derived to estimate C_T

$$\hat{C}_T = f_L \frac{\hat{x}}{\hat{z}} \tag{7.36}$$

where $\hat{p} = \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \end{bmatrix}^T$.

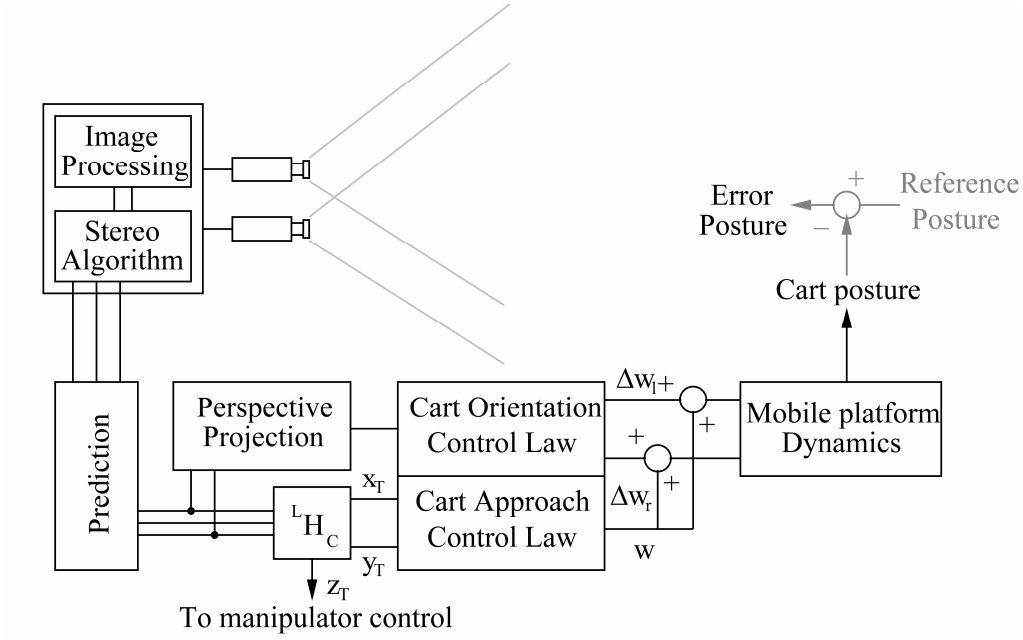
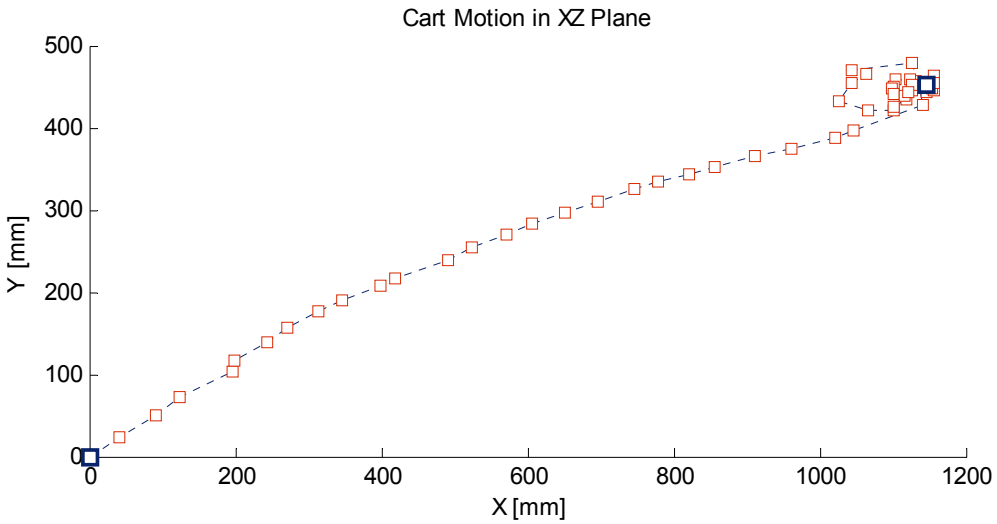
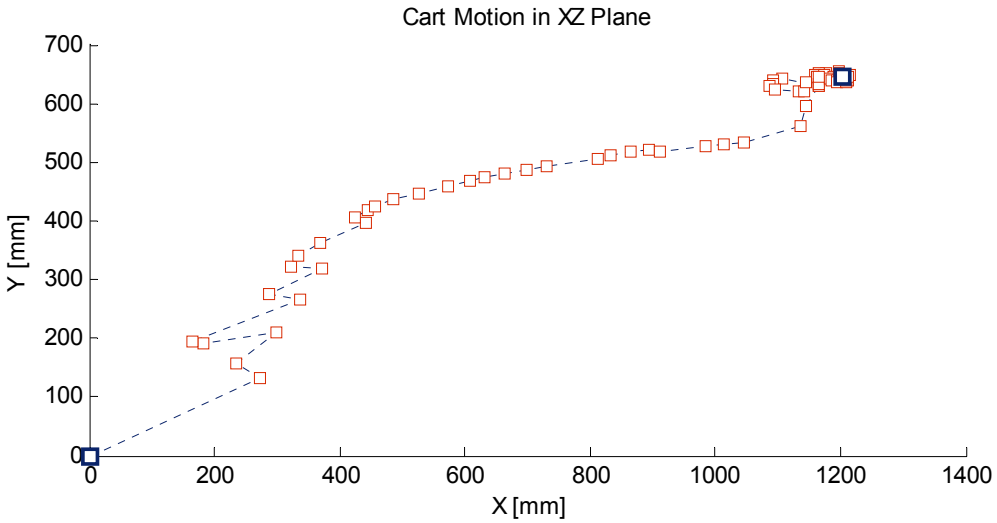


Figure 7.12 Mobile platform control algorithm with prediction



(a) Mobile Platform trajectory for low level actuator gains (K_p=400,K_I=10) (rectangles: measured positions bold rectangles: references)



(b) Mobile Platform trajectory for low level actuator gains (K_p=500,K_I=10) (rectangles: measured positions bold rectangles: references)

Figure 7.13 Trajectories of mobile platform controlled with prediction extension. Albeit high controller gains, mobile platform moves smoothly.

As it can be seen in resulting motion graphics, prediction procedure improved the performance of the mobile platform. But it must be noted that prediction by using derivatives, increases the sensitivity of control system to deviations.

7.4 Control of Manipulator

The second subsystem of mobile manipulator to control is manipulator arm. Manipulator control becomes active when cart orientation and approach reach a steady state. Resultant position vector is transferred to inverse kinematic solver which generates angular position references for planar manipulator by solving following equations:

$$x^2 + y^2 = l_1^2 + l_2^2 - l_1 l_2 \cos(\pi - \theta_2) \tag{7.37}$$

$$\theta_2 = \mp a\cos\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{l_1 l_2}\right) \tag{7.38}$$

$$\theta_1 = a\tan\left(\frac{y}{x}\right) - a\tan\left(\frac{l_2 \sin\theta_2}{l_1 + l_2 \cos\theta_2}\right) \tag{7.39}$$

These references are transferred to low level controllers where PI algorithms for position inputs are exploited.

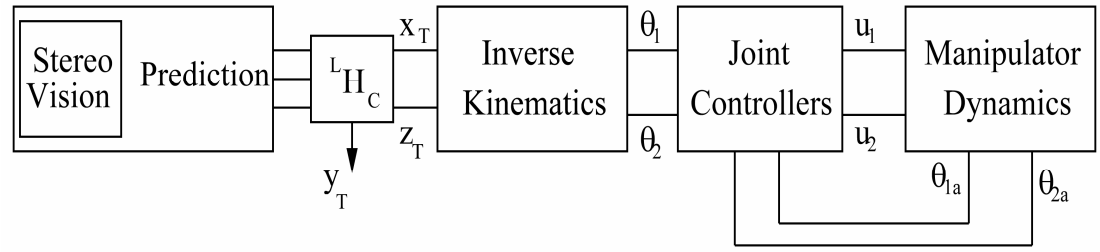
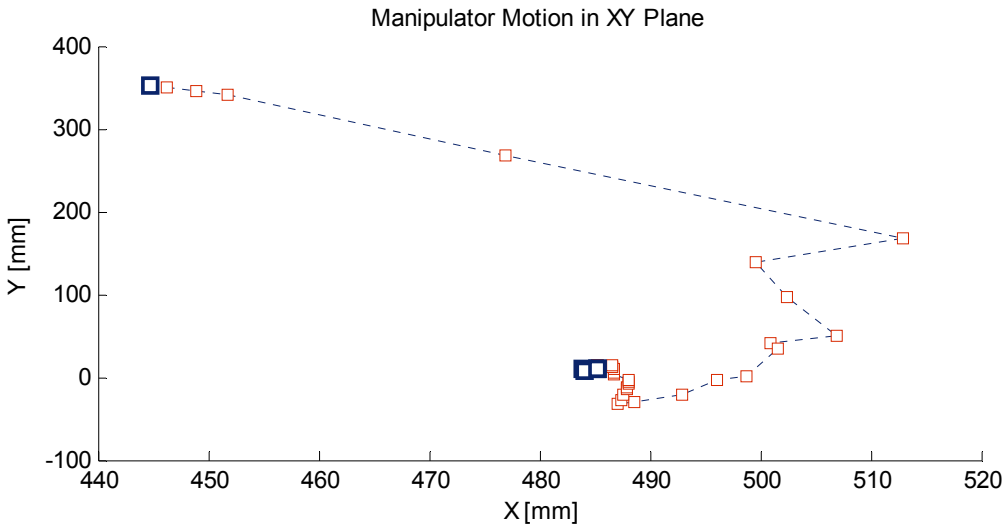
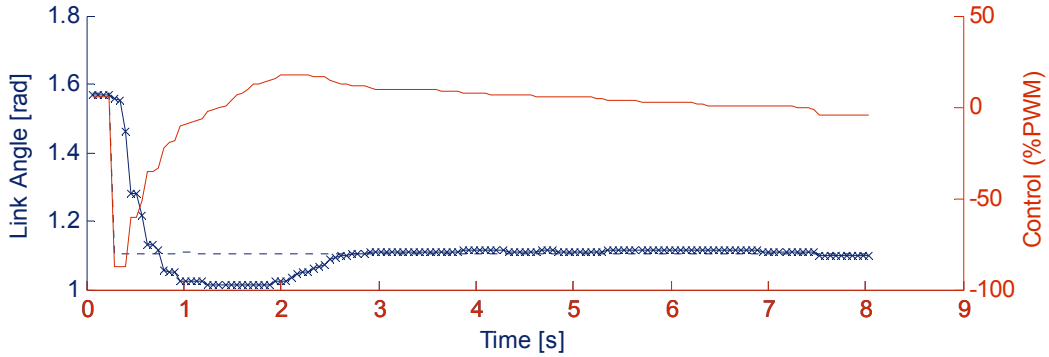


Figure 7.14 Manipulator arm control procedure

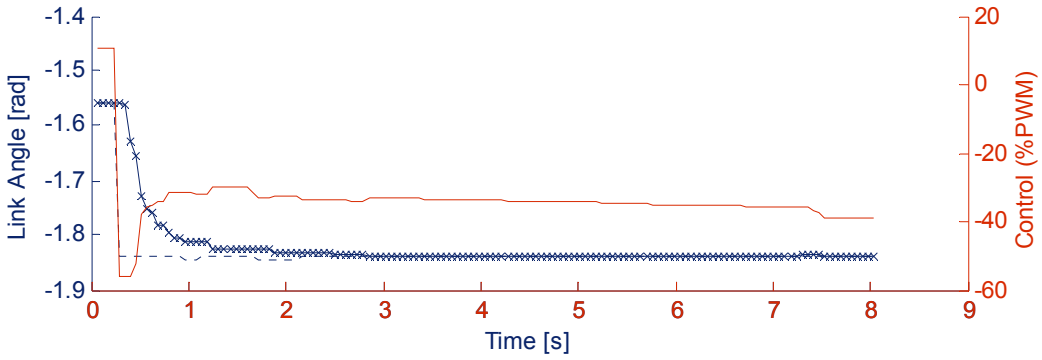
(In figures for spatial data presentation, rectangles represent measured positions and bold rectangles stand for references from vision system. In time domain graphics reference values are marked with dashed line, actual values with crosses and solid line shows control effort. All values are represented with 1024 bit resolution integer conversion.)



(a) End effector trajectory (rectangles: measured positions bold rectangles: references)



(b)Time response of 1st :link (-- reference, x actual) and control signal (-)



(c)Time response of 2nd :link (-- reference, x actual) and control signal (-)

Figure 7.15 Controlled manipulator motion (1st : Kp=1200,K_I=4; 2nd : Kp=1400,K_I=6)

Time domain characteristics of the manipulator outputs represent two problems. The first is oscillating motions due to integral part of the controller and the second is dead zone like effects at the beginning of the motion. When dynamics of the motion is analyzed, it is seen that for a certain error value proportional controller generates a

PWM signal with constant duty cycle ratio. According to instantaneous pose of manipulator and amplitude of error signal, this value becomes insufficient for moving the link. As time passes effect of the integral part of controller becomes evident and control signal becomes large enough to move the link. This phenomenon leads poor control signal dynamics.

The described problems lead a question about controller design: “Is it possible to implement a proportional control law which overcomes the problem due to nonlinear dynamics of the manipulator arm?”

To find a solution to this question, first the efficient work space of the manipulator is described. The positioning task due to acquired vision data can only be executed if and only if the target is placed in intersection of field of view of each camera and the workspace of the manipulator

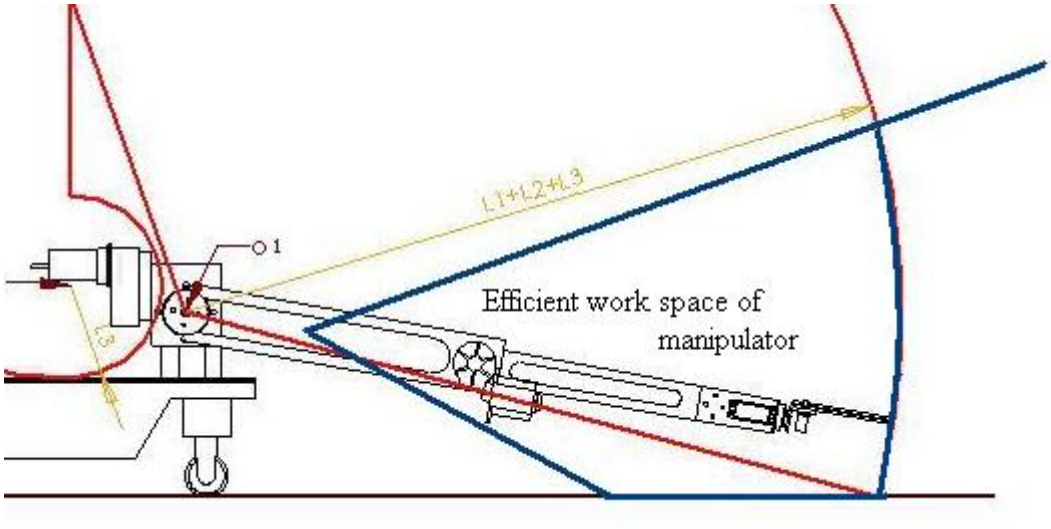


Figure 7.16 Efficient workspace of manipulator arm(Intersection of common filed of view of cameras and manipulator workspace)

Experimental definition of the efficient work space is done by moving the target to various extreme positions and feedback values from potentiometers are logged (This values are bit values corresponding to angular position of the link and they are related to real radian values with equations 6.10, 6.11). In the previous section it is noted that manipulator links are connected to each other via helical springs. Main

function of the springs is compensating the mechanical gap in actuator link connections. Another effect of them is they generate a torque opposite to torque due to gravity. Sum of this two torques effect the motion of the manipulator by generating torque bias according to pose of the manipulator

By analogy, the effect of this torques on actuator motion is a duty cycle bias. An experimental procedure is developed to determine this effect analytically. Starting from arbitrary initial angular position, a computer program apply duty cycle incrementally from zero to value motion is detected. This duty cycle value is logged for that specific position of the link as a bias to overcome static moments. Applying this procedure for sample points in efficient workspace for each link and logging the results form a look up table for this compensation in a limited region.(It is known from kinematical analysis that effect of weight of the second link on the first link changes with $\theta_2(t)$ but an approximation is done by making zero which is the maximal interaction moment between two links).

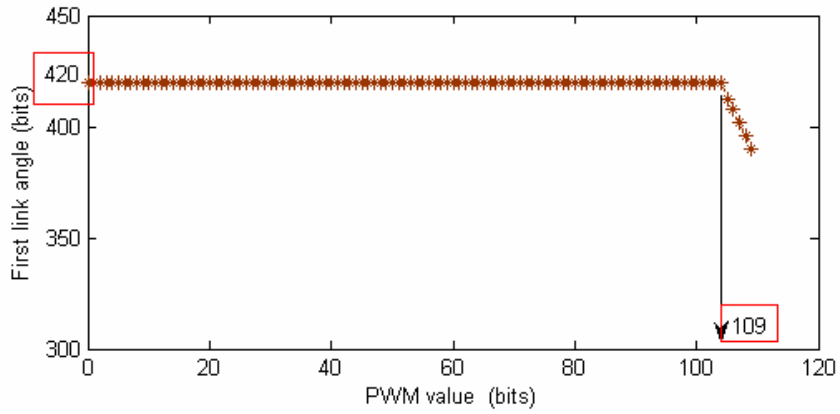


Figure 7.17 Sample output of experimental system. When first link is at stationery position on 420 bit value, duty cycle increase incrementally till position changes.

$$u^{-}_{1,bias} = -0.4374\,\theta_1 + 285.1043 \tag{7.40}$$

$$u^{+}_{1,bias} = 0.6933\theta_1 - 453.9456 \tag{7.41}$$

$$u^{-}_{2,bias} = -0.6089\,\theta_2 + 383.6040 \tag{7.42}$$

$$u^{-}_{2,bias} = -0.7962\theta_2 + 148.7899 \tag{7.43}$$

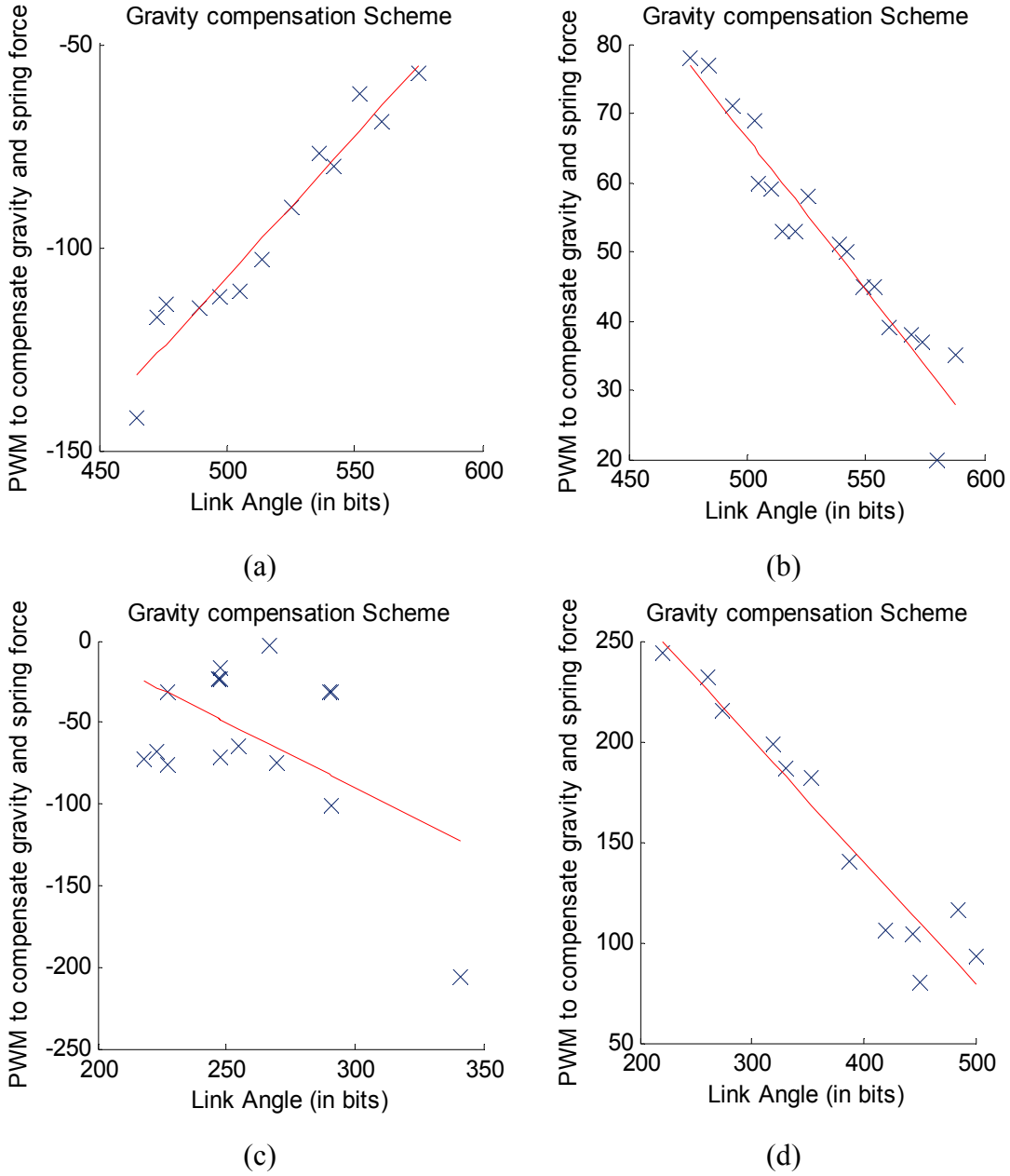


Figure 7.18 Gravity compensation experimental results

Resultant equations are linear relationships between link angle values and pwm outputs. A major advantage of using analog to digital converter values is equations can be implemented in embedded controllers.

Recalling differential equation of the manipulator and placing the effect of springs and gravity compensation torque resultant system is quasi linear:

$$M(\bar{q})\bar{q}'' + C(\bar{q},\bar{q}')\bar{q}' + G(\bar{q}) = \tau \tag{7.44}$$

$$M(\bar{q})\bar{q}'' + (C(\bar{q},\bar{q}')\bar{q}' + G(\bar{q}) - F_s(\bar{q}) - u_{bias}) = \tau \tag{7.45}$$

Term in parentheses is assumed that it is zero.(Reduced computed torque control scheme shows that effects of Coriolis and Centrifugal terms are negligible in low velocities, especially for position controller algorithms.(Khosla & Kanade, 1988).

$$M(\bar{q})\bar{q}'' = \tau \tag{7.46}$$

A model based proportional control law for each link is developed instead of PI controllers:

$$u_1 = \begin{cases} K_p(\theta_{1,r} - \theta_{1,c}) + u^-_{1,bias} & (\theta_{1,r} - \theta_{1,c}) < 0 \\ K_p(\theta_{1,r} - \theta_{1,c}) + u^+_{1,bias} & (\theta_{1,r} - \theta_{1,c}) > 0 \end{cases} \tag{7.47}$$

$$u_2 = \begin{cases} K_p(\theta_{2,r} - \theta_{2,c}) + u^-_{2,bias} & (\theta_{2,r} - \theta_{2,c}) < 0 \\ K_p(\theta_{2,r} - \theta_{2,c}) + u^+_{2,bias} & (\theta_{2,r} - \theta_{2,c}) > 0 \end{cases} \tag{7.48}$$

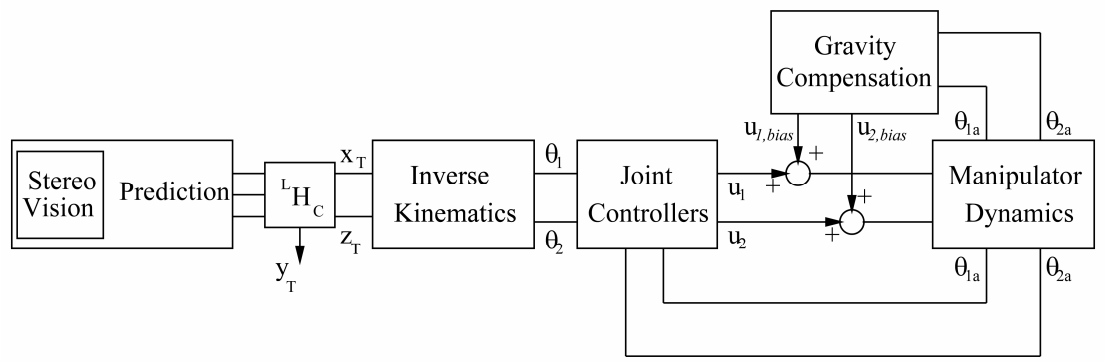
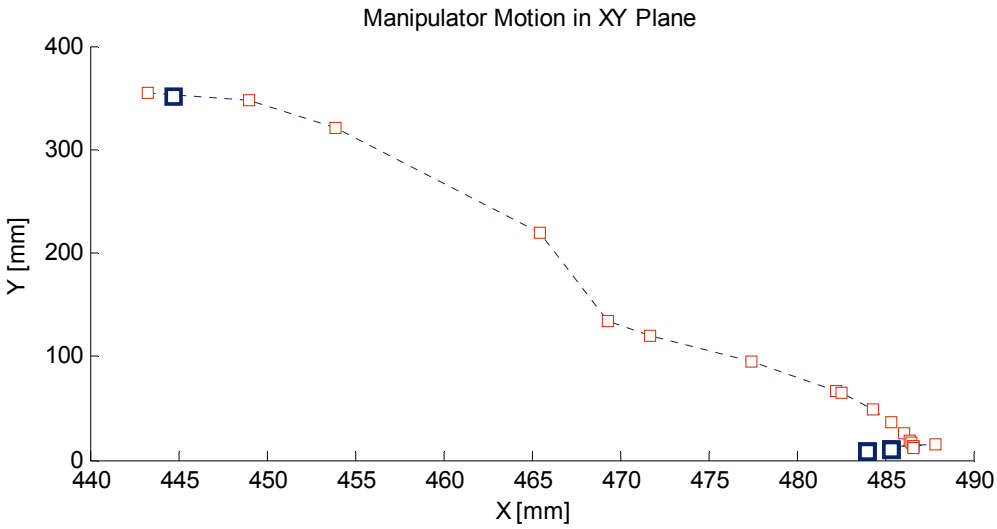
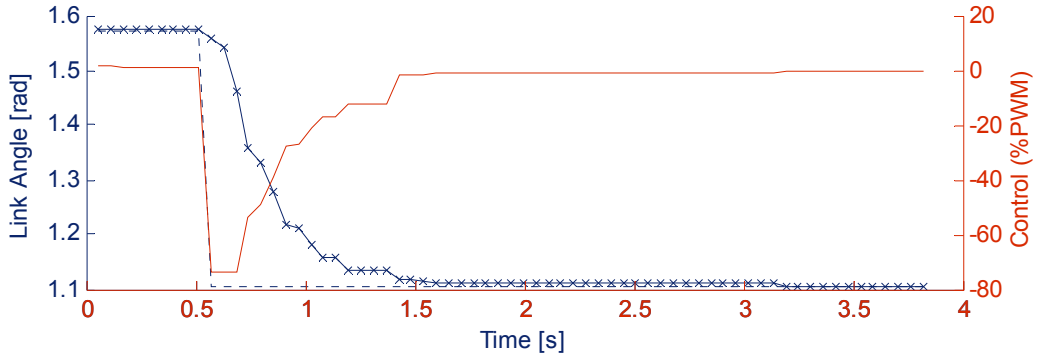


Figure 7.19 Modified control algorithm for manipulator

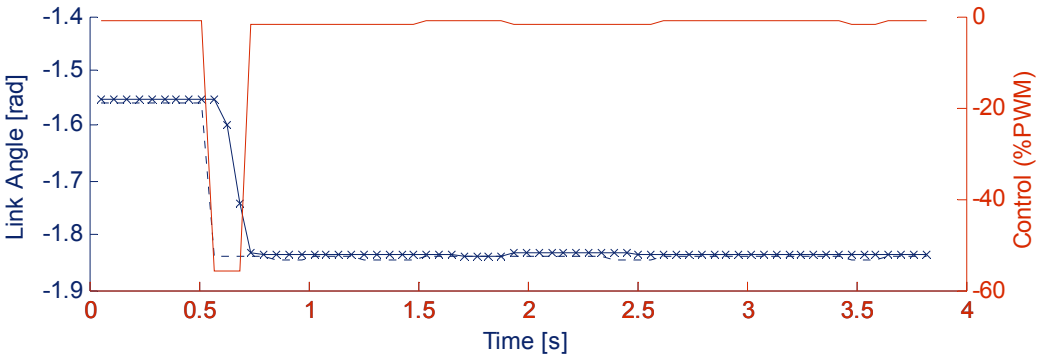
When low level controllers with gravity compensators implemented, it is seen that not only controller efforts but also spatial motion characteristics is improved.



(a) End effector trajectory (rectangles: measured positions bold rectangles: references)

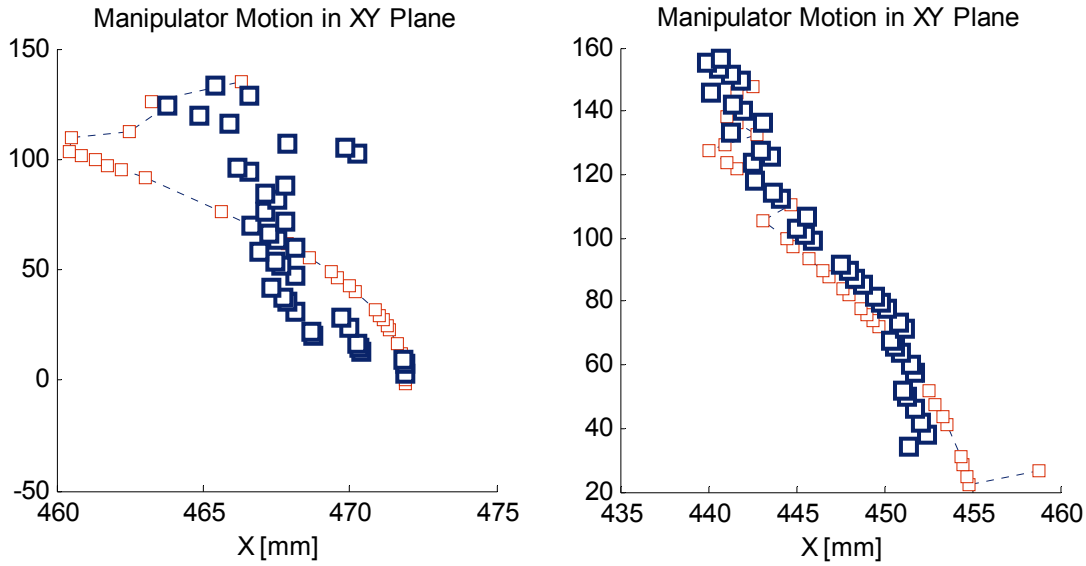


(b)Time response of 1st :link (-- reference, x actual) and control signal (-)

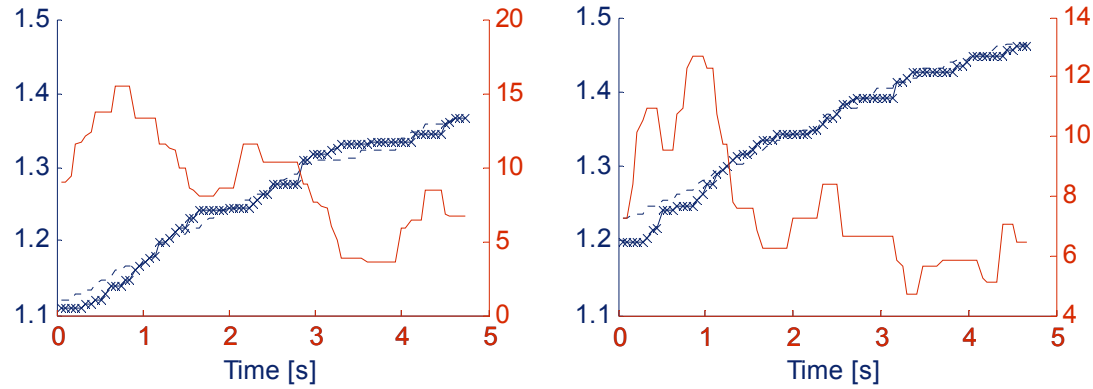


(b)Time response of 2nd :link (-- reference, x actual) and control signal (-)

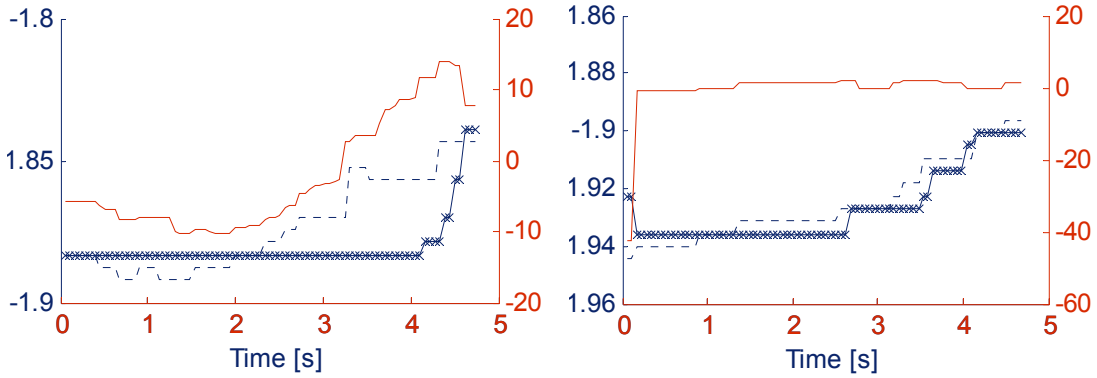
Figure 7.20 Controlled manipulator motion proportional controllers with gravity compensation (1st : Kp=1200; 2nd : Kp=1400)



(a) End effector trajectory (rectangles: measured positions bold rectangles: references)



(b)Time response of 1st :link (-- reference, x actual) and control signal (-)



(b)Time response of 2nd :link (-- reference, x actual) and control signal (-)

Figure 7.21 Comparison of manipulator control algorithms for horizontal tracing task

Another good example to represent effect of the gravity compensation is to compare performance of two control algorithms for a horizontal (on z axis) tracking

task. On the left column PI controller performance is shown on the right column, results of proportional controller with gravity compensation is presented.

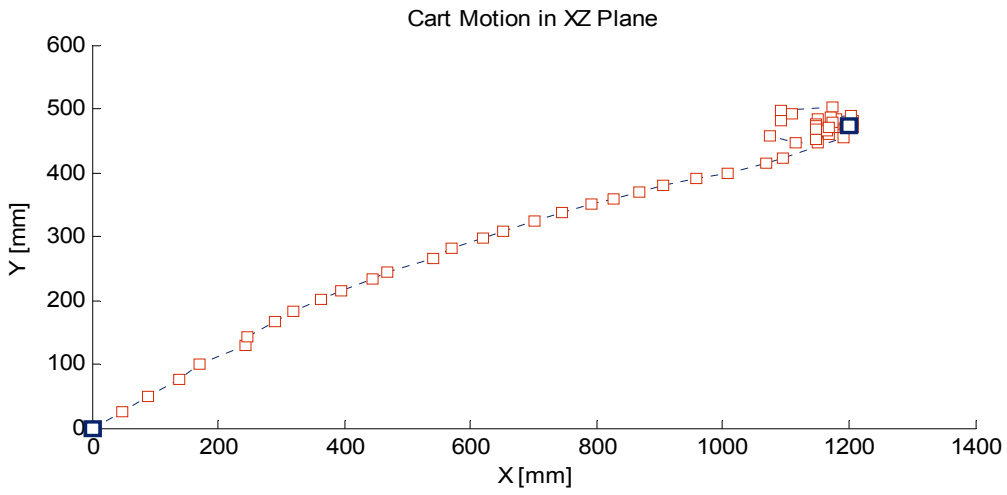
In the previous two sections, two different control algorithms for two different subsystems are developed. In the next section, control of mobile manipulator is concerned.

7.5 Control of a Mobile Manipulator

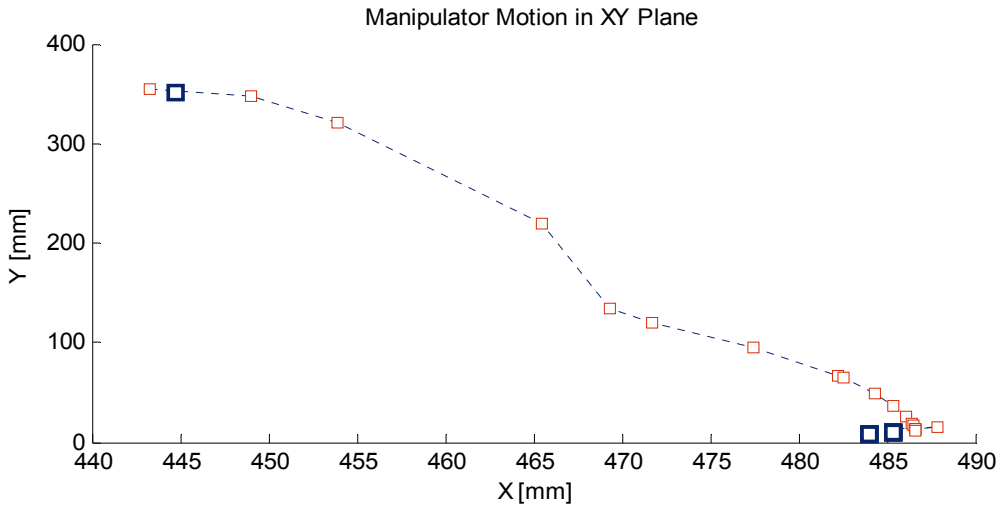
Controlled motion of the overall system is obtained by sequentially combining two tasks. The mobile platform performs the major task, which is placing the target in the workspace of manipulator then manipulator controller performs gripping the target.

Table 7.2 Control of the mobile manipulator motion (two sub controllers acts sequentially)

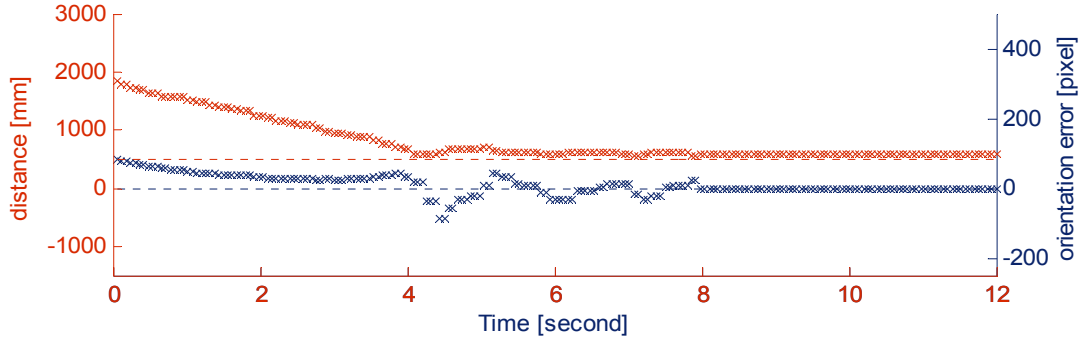
Mobile Platform Controller		Low level controllers	
r	500	1st Link (K_p ,K_I)	1200+grav. com.
w_{max}	100	2nd Link (K_p ,K_I)	1400+grav. com.
Width	160	Left Wheel (K_p ,K_I)	400,10
Δw_{max}	80	Right Wheel (K_p ,K_I)	400,10



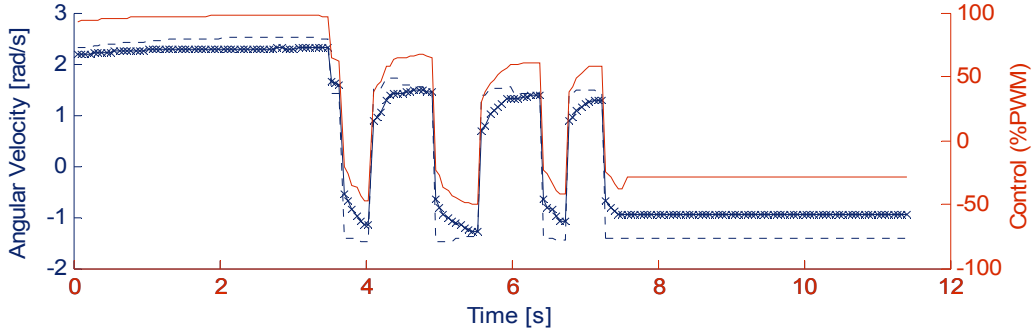
(a) Mobile Platform trajectory (rectangles: measured positions bold rectangles: references)



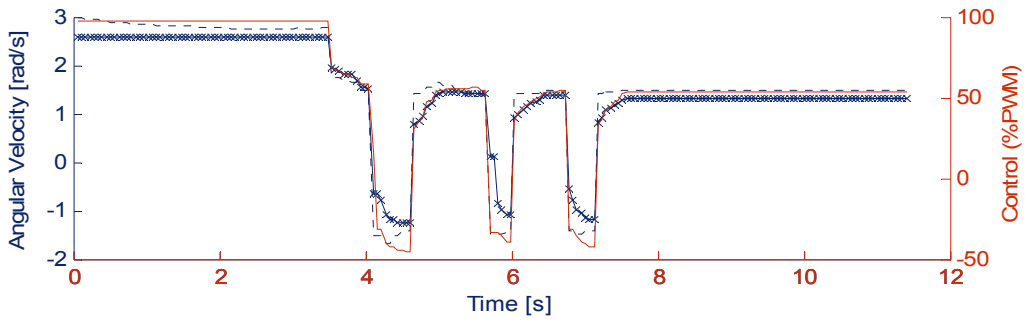
(b) Manipulator end effector trajectory (rectangles: measured positions bold rectangles: references)



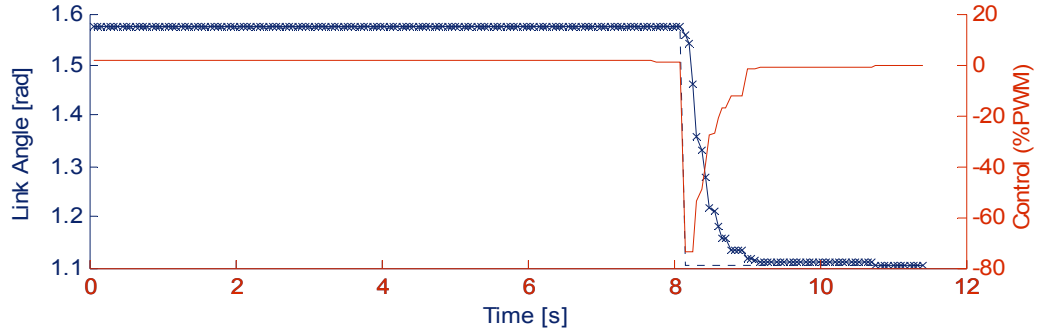
(c) Mobile Platform Error Dynamics (-- reference, x actual)



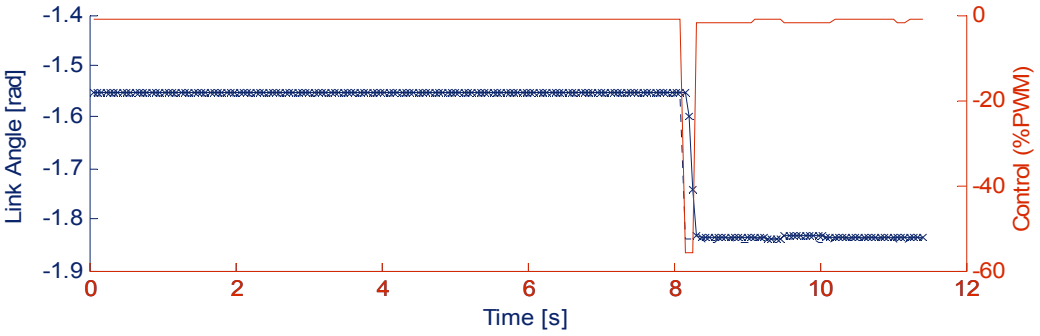
(d) Left wheel actuator time response (-- reference, x actual) and control signal (-)



(e) Right wheel actuator time response (-- reference, x actual) and control signal (-)



(f) First link actuator time response (-- reference, x actual) and control signal (-)



(g) Second link actuator time response (-- reference, x actual) and control signal (-)

Figure 7.22 Results of mobile manipulator control algorithm

Given results prove the success of proposed control algorithm. Mobile platform control law stays active till target is in the manipulator workspace. Afterwards manipulator control routine, performance of which is improved via gravity compensation scheme, is activated and manipulator moves to target coordinates.

One of the advantages of mobile manipulators is that they add extra degrees of freedoms to robot manipulators. Kinematical redundancy is an accepted methodology to avoid singularities in configuration space (De Luca,Oriolo & Robuffo Giordano, 2006) and it is also advantageous if there is a force control demand in tool frame of the robot manipulator (Baeten & Schutter, 2004).

Control algorithms concerning redundancies in mobile manipulators are based on combined kinematical representation of the system. Resultant system contains more control parameters than degrees of the system. Presented techniques mainly focus on rectified methods of linear algebra like pseudo inverse or projected kinematics methods (Bishop & Spong, 1998).

The key issue, laying under this redundancy resolution methods, is that in existence of multiple actuators affecting same degree of freedom how the control effort should be shared between them?

In this study, a novel approach is offered as a solution to this problem. In mobile manipulators, causes of the redundancy are coincident degrees of freedoms of cart and manipulator. On the other hand, the distribution of the demanded change in corresponding task dimension is not an unconstrained task because of the limits of the manipulator actuators. Therefore, prime issue to consider is to carefully adopt a methodology, which concerns limits of the actuators.

Proposed methodology primarily considers the portion of the demanded change which will be tolerated via limited actuator configuration (in our case it is manipulator joint actuators.). If we suppose a task space displacement x is conveyed to mobile manipulator to tolerate, this value can be tolerated via controlling linear velocity of the mobile platform or end point extension mobile manipulator (if the target is in the workspace of manipulator arm).

$$x = x_v + x_m \tag{7.49}$$

$$x = (1-\alpha)x + \alpha x \tag{7.50}$$

It is evident that final displacement will be distributed as given in 7.50 if α is a value between 0 and 1. It can be deduced from derived motion characteristics. Amount of displacement, which is tolerated with manipulator, increases when it is close to the base frame of the manipulator and goes to zero out of the boundaries of work space.

$$\begin{array}{ll} 0 < \alpha < 1 & 0 < x \le r \\ \alpha \approx 0 & x > r \end{array} \tag{7.51}$$

where parameter r corresponds to workspace limit of manipulator

A workspace function is defined to satisfy these conditions by using analogy with step input response.

$$\alpha = e^{-\frac{x}{r/5}} \tag{7.52}$$

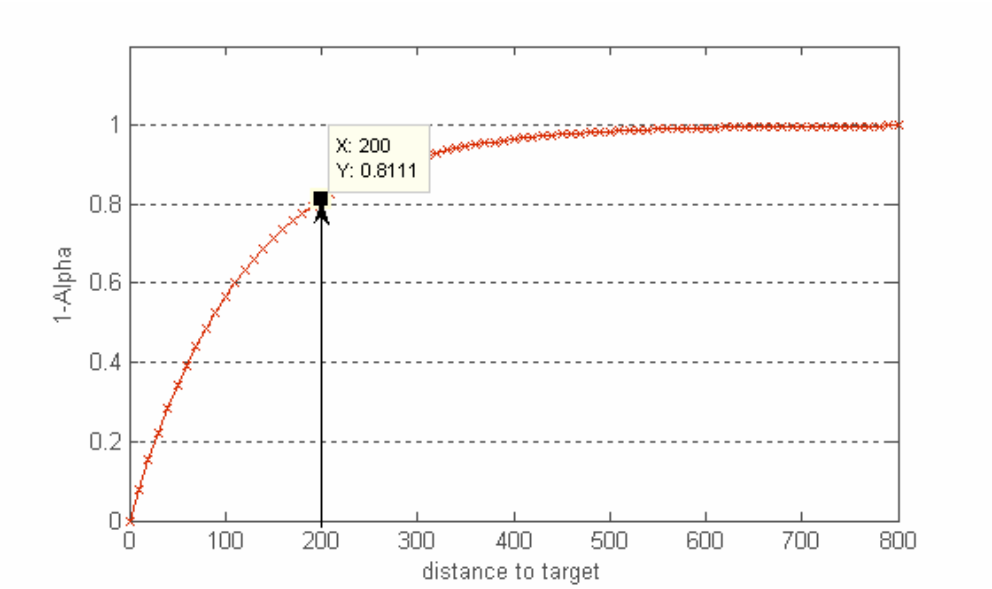


Figure 7.23 Workspace function (representation of equation 7.52 where r=600. If the target is formerly placed in x=200mm and moves x mm then cart will move 0.8111x and manipulator will move 0.1989x)

This function is implemented as following,

$$x_i = \sqrt{x_T^2 + y_T^2} \tag{7.53}$$

$$\alpha = e^{-\frac{x}{r/5}} \tag{7.54}$$

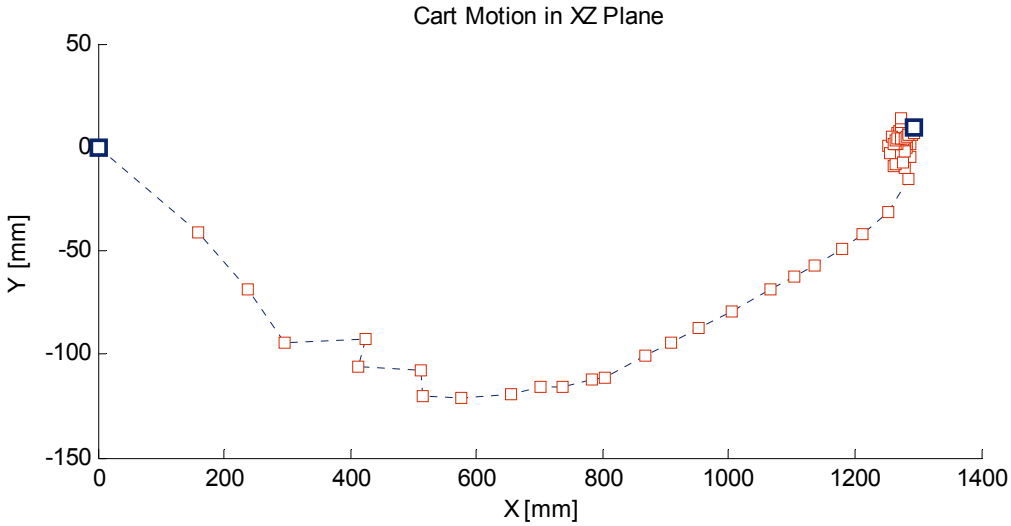
Calculated values are used when finding references for motion

$$x_v = x_{v,i} + (1-\alpha)(x_i - x_{i-1}) \tag{7.55}$$

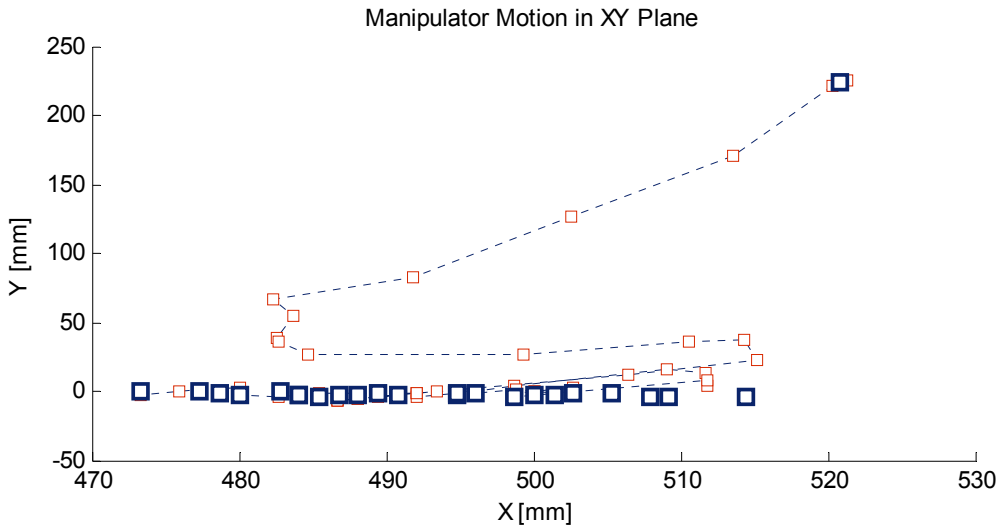
$$x_m = x_{m,i} + \alpha(x_i - x_{i-1}) \tag{7.56}$$

Table 7.3 Control of the mobile manipulator motion

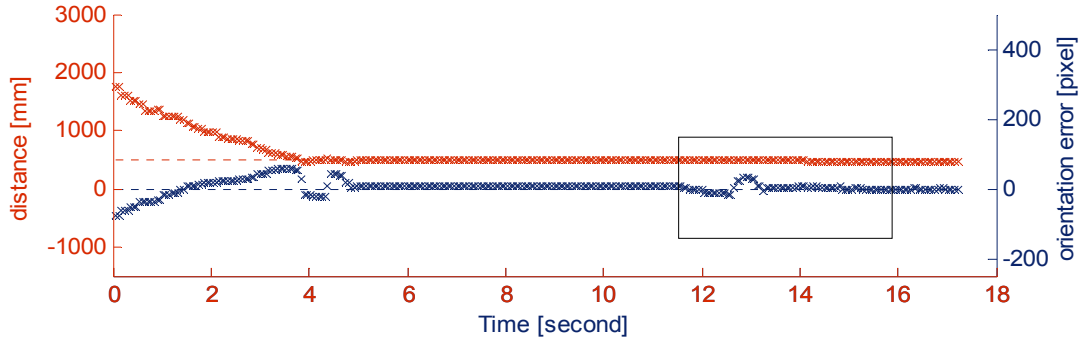
Mobile Platform Controller		Low level controllers	
r	600	1st Link (K_p ,K_I)	1200+grav. com.
w_{max}	100	2nd Link (K_p ,K_I)	1400+grav. com.
Width	160	Left Wheel (K_p ,K_I)	400,10
Δw_{max}	80	Right Wheel (K_p ,K_I)	400,10



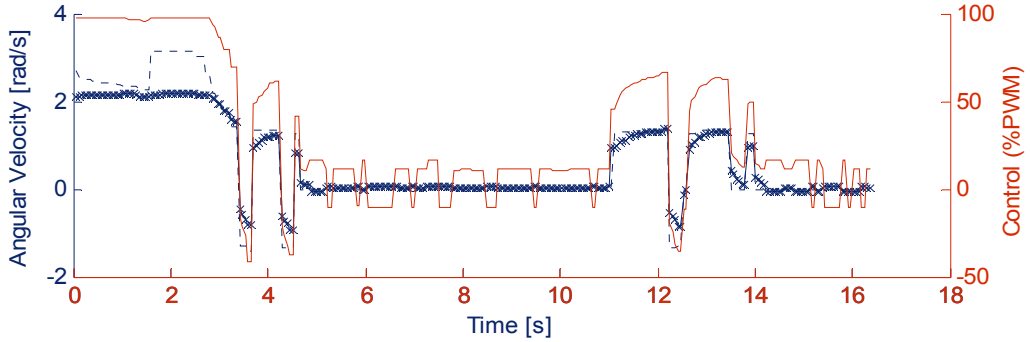
(a) Mobile Platform trajectory



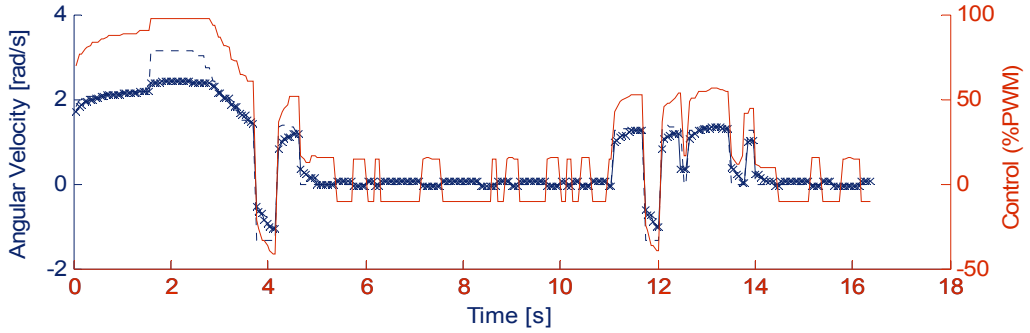
(b) Manipulator end effector trajectory (rectangles: measured positions bold rectangles: references)



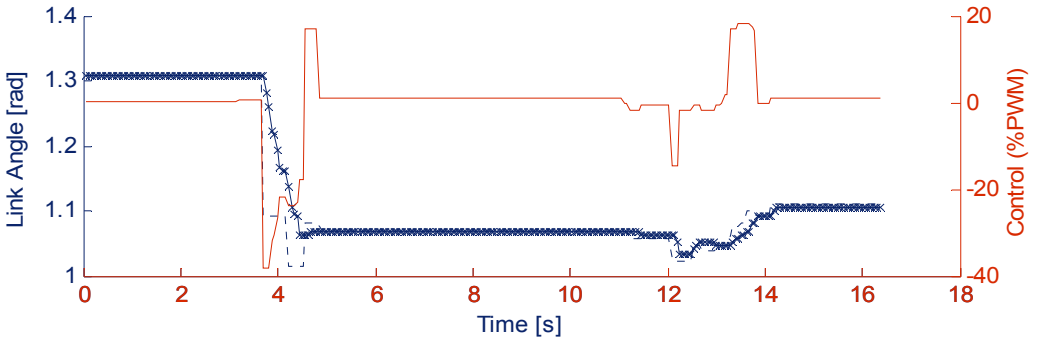
(c) Mobile Platform Error Dynamics (-- reference, x actual)



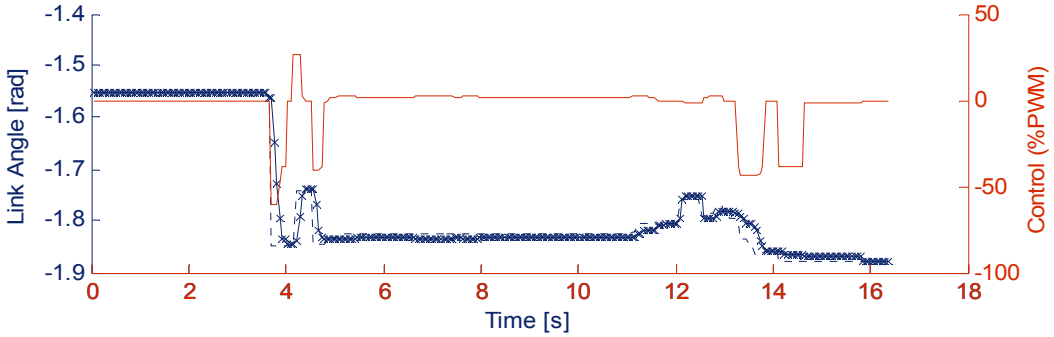
(d) Left wheel actuator time response (-- reference, x actual) and control signal (-)



(e) Right wheel actuator time response (-- reference, x actual) and control signal (-)



(f) First link actuator time response (-- reference, x actual) and control signal (-)



(g) Second link actuator time response (-- reference, x actual) and control signal (-)

Figure 7.24 Mobile manipulator control via exploiting workspace function.

As it is seen in time response of the actuators, regions marked with rectangular, demonstrates that a change in distance target tolerated not by each system separately but cooperative work of two systems. This cooperation is provided via implementation of developed workspace function.

The first stage of the experiments mainly focuses on mobile platform and manipulator motion separately. Afterwards motion of the overall system is concerned.

The first experiments are done to enlighten dynamical tracking capability of the mobile platform. Developed control algorithms (equations 7.31 and 7.32) are implemented directly to observe real-time performance. While increasing low level actuator gains and respectively increasing rate of response of mobile platform, spatial observations showed non smooth motion in figure 7.9 (b) (by non smooth motion it is referred that mobile platform does not moves translational, it only pivots to track target).

Concerning characteristics of vision system, reason for this motion phenomenon is found as intermittent delay in hardware. It is observed that at time instant k a delayed vision system feedback is introduced. A backward difference methodology is adopted to calculate a rectification on vision system feedback. Implementation of modified control scheme in figure 7.12 improved the motion characteristics of

mobile platform (Figure 7.13) and this scheme is adopted as final control methodology.

The second stages of experiments are due to manipulator motion. In chapter 6, it is stated that low level actuator are tuned experimentally. In this chapter, dynamic look and move controller scheme is defined as control methodology for manipulator motion. Observations of the time response curves of actuators represent poor motion characteristics (figure 7.15) (References could not be tracked with actuators continuously). It is evident that applying a linear PI control law directly on inherently nonlinear system causes this performance failure. Recalling reduced computed torque technique, an implementation scheme concerning efficient workspace of the manipulator is developed using experimental modeling of static torque requirements of manipulator links (described in figure 7.19). Application of new methodology improves both spatial and temporal characteristics of manipulator controllers. System posses lower spatial overshoot and no attenuation in control signals (figure 7.20). This improvement also leads better tracking capability (figure 7.21).

The last stage of the experimental study focuses on mobile manipulator motion. The first experiments combine motion of two subsystems sequentially. That is tracking and gripping phases. Performance of the system can be observed in figure 7.22. In the last stage of the mobile manipulator control experiments, a policy that cooperates motion of two subsystems is developed. A workspace function, basically an exponential function is introduced to compromise approach of the mobile platform and the reach of manipulator (equations 7.54-7.56). Implementation of this new methodology causes cooperative motion which overcomes redundancy problem in mobile manipulator kinematics.

CHAPTER EIGHT
CONCLUSION

8.1 Summary

In this thesis, the object of study is mobile manipulator system consisting stereo cameras and controlled with vision based algorithms. It investigates how to operate such systems, where operation refers especially to servoing of a tool with respect to an object to be manipulated.

In the first stage of this study a mobile manipulator system is designed and produced. Experimental rig for this thesis is composed of three subsystems: electromechanical, electronics and computer subsystems. Mechatronical design procedure is applied concerning subsystem functions and cooperative operation performance.

In the second stage, Mechanical and vision system characteristics are focused to gain a sense on dynamics which concerns controller design procedure. Models for two different kinematical mechanisms are presented. Results of analysis are exploited to enlighten effect of actuators on motion dynamics.

In the third stage, vision based control is reviewed and application methodology is demonstrated. Mathematical models for integration of vision data into feedback loops are concerned and controller design procedure is presented as a theoretical fundamental to adopted control procedure.

Control of the system is organized in a hierarchic way: Low level servo controllers (embedded controllers) are developed to track input from high level controller (computer) where vision based control laws are implemented.

Control related part of this thesis starts with low level controller design methodology. Dynamic characteristics of actuators and mechanism are analyzed and experimental tuning of controllers for servo controllers is demonstrated.

In last section of this study, computer and vision systems are integrated as a part of control process. Using developed kinematical decomposition in modeling chapter, design procedure starts with mobile platform controller. Using image and position based control methodologies; control of the mobile platform is succeeded to fulfill the task keeping the manipulator gripper in the workspace of the manipulator and target in the same vertical frame with gripper. Manipulator controller is operated in a dynamic look and move procedure where joint references instantaneously updated.

As a solution to sequential motion of the overall system, a novel method is proposed. This method develops a workspace function which relates motion of the cart and the manipulator. Implementation of this controller leads cooperative motion of the mobile manipulator.

8.2 Contributions

Primal contribution of this thesis, from technological point of view, is a design and production of a mobile manipulator equipped with stereo vision system. Mechatronical design procedure is justified by designing mechanical, electronics and computer subsystem to build physical experimental model. In context of living in developing country, system components are chosen from the standard components. All electromechanical equipment from motor joint couplings to servo controllers are designed and produced. A stereo rig is formed with two USB cameras and a multithread computer program is developed for both visual perception task and supervisory control.

This thesis introduces also novel computational algorithm for solving stereo triangulation for 3D recovery of a point on the image pair. This new method duplicated the results from standard algorithm whereas it has fewer implementation

steps. This new algorithm uses planar projection of the 3D rays instead of solving shortest distance between lines problem.

An implementation scheme for supervisory vision based control with task sequencing is proposed. System decomposed into two kinematical structures and each subsystem is controlled with independent controller. Mobile platform motion is controlled with a hybrid visual servoing methodology by using position based control law for linear velocity component and image based regulator for angular velocity component.

Manipulator motion is controlled with conventional control methodology with gravity compensation. Although gravity compensation or computed torque control techniques are standards of robot control technology, a novel implementation methodology is developed for embedded controllers. To define this control methodology, firstly, efficient workspace of the manipulator is defined. This new workspace is found by intersecting robot workspace and common field of view of cameras. An experiment is designed to measure necessary torque bias for various robot manipulator poses. Linear approximation to experiment results is exploited in embedded controller.

Concerning kinematic redundancy in mobile manipulator configuration, a novel model based approach for distributing control effort to two subsystems is introduced. An exponential workspace function is derived for this purpose. By exploiting this new function, cooperating operation of two subsystems are obtained.

In conclusion, final robotic system operates sequentially till the target is in the manipulator workspace then system switches control mode into cooperating scheme, in which manipulator and mobile platform motion are simultaneous.

8.2 Future Work

However many things remain to be done, and immediate perspectives of the presented methodology are manifold. First are the control theoretical questions need to be studied, such as the robustness, adaptability, and convergence properties of the proposed control laws in more abstract level. Optimal motion planning concerning energy efficiency is also in the future perspective this study. One major point for research is that this system offers a platform which has a open structural programming strategy. That means in further studies it can be integrated to exploration researches where not only manipulation task but also environment modeling tasks are concerned.

One aspect of the study which deserves further attention is that of control of the mobile manipulator using workspace function. While our formal approach proved successful for our particular application, it was created with further implementation in mind. Applications in mobile manipulators which have redundancy on more than one axis will be also kept in concern.

Perspectives beyond the field of robotics are numerous, and should generally be expected in domains in which actions and visual perceptions, and especially their geometry and dynamics, have to be coupled at high rates. Mobile manipulator mechanisms are already has wide use in field applications. One extreme example is planetary exploration tasks. But main issue lying under developed designs is that they are very expensive and not found wide in market. Use of standard technologies in this application can be revolutionary (As an example USB cameras are about 20 $ but a frame grabber and camera setup costs 500 $ minimal). Therefore any approach on software modifications or supplementary electronic designs are also proved to be useful.

REFERENCES

Baeten, J. & De Schutter, J. (2004). *Integrated visual servoing and force control*. Berlin: Springer

Balkcom, D. & Mason, M. (2004). Time optimal trajectories for bounded velocity differential drive robots, *In Proceedings of IEEE International Conference on Robotics and Automation,* 3, 2499 - 2504.New Orleans, USA.

Bayle, B., Fourquet, J.-Y., & Renaud, M. (2001). Manipulability analysis for mobile manipulators. *In Proceedings of IEEE International Conference on Robotics & Automation*, 1251-1256. Seoul, Korea

Berry, E., Martinet, P., & Gallice, J. (1997). Trajectory generation by visual servoing. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2, 1066–1072. Grenoble, France

Bishop, B.E. & Spong, M.W. (1998). Control of redundant manipulators using logic based switching. *IEEE Conference on Decision and Control*, 553-559. Tampa, Fl.

Bloch, A., Baillieul, J., Crouch, P.E. & Marsden, J.E. (2003). *Nonholonomic mechanics and control*. Berlin: Springer.

Bouguet, J.Y. (April 13, 2007). Camera Calibration Toolbox for Matlab. Retrieved July 13, 2007 from http://www.vision.caltech.edu/bouguetj/calib_doc/

Brockett, R.(1983). Asymptotic stability and feedback stabilization. *Differential Geometric Control Theory*, pages 181-191.

Buttazzo, G.C., Allotta, B. & Fanizza, F.P. (1994). Mousebuster: a robot for real-time catching. *Control Systems Magazine*. 14(1), 49-56.

Campion, G., Bastin, G. & Andrea-Novel, B.(1996). The vector field histogram fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1):47-62.

Chatila, R., Lacroix, S., Simeon, T. & Herrb, M. (1995). Planetary exploration by a mobile robot: Mission teleprogramming and autonomous navigation. *Autonomous Robots, 2*(4), 333-344.

Coombs, D. & Brown, C. (1993) Real-time binocular smooth pursuit. *International Journal of Computer Vision*, 11(2):147-164.

Corke, P.I. (1994). Visual control of robot manipulator: A review. In Hashimoto, K. (Ed), *Visual Servoing*. (1-32) Singapore:World Scientific.

Das, A.K., Fierro, R., Kumar, V., Ostrowski, J.P., Spletzer, J. & Taylor, C.J. (2002). A vision-based formation control framework. *IEEE Transactions Robotics and Automation*.18(5). 813--825.

De Luca, A., Oriolo, G. & Robuffo Giordano P. (2006). Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators. *In Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 1867-1873 Orlando, Florida.

De Luca, A., Oriolo, G. & Robuffo Giordano P. (2007). Image-based visual. servoing schemes for nonholonomic mobile manipulators. *Robotica* 25, 131–145.

Ekvall, S., Kragic, D. & Hoffmann, F. (2005). Object Recognition and Pose Estimation using Color Cooccurrence Histograms and Geometric Modeling. *Image and Vision Computing*, 23(11), 943-955.

Espiau, B., Chaumette, F. & Rives, F. (1992). A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313-326.

Forsyth, D.A. & Ponce, J. (2002). *Computer Vision: A Modern Approach*. NY: Prentice Hall.

Fu, K.S., Gonzalez, J. & Lee, S. (1987). *Robotics: Control, sensing, vision, and intelligence*. New York: McGraw-Hill.

Gonzalez, R. C. & Woods, R.E. (2002). *Digital Image Processing*. NY: Prentice-Hall.

Hashimoto, K. (2003). A review on vision-based control of robot manipulators. *Advanced Robotics* 17(10), 969-991.

Hartley, R. I & Zisserman, A. (2000). Multiple View Geometry in Computer Vision. Cambridge: Cambridge University Press.

Heikkila, J. & Silven, O. (1997). A four-step camera calibration procedure with implicit image correction. *In Proceedings of Conference on Computer Vision and Pattern Recognition*, 1106-1113.

Hootsmans, N.A.M., & Dubowsky, S. (1991). The control of mobile manipulators including vehicle dynamic characteristics, *In Proceedings of. IV Topical Meeting on Robotics and Remote Systems*, 24-28, NM Albuquerque.

Hutchinson, S.A., Hager, G.D. & Corke P.I. (1996). A tutorial on visual servo control. *IEEE Transactions on robotics and. automation*, 12(5), 651—670.

Jörg, K.W.(1995). World modeling for an autonomous mobile robot using heterogenoussensor information. *Robotics and Autonomous. Systems*., 14, 159-170.

Kalinsky, D. (July 17, 2001). *Introduction to I^2C.* Retrieved December 12, 2007, from http://www.embedded.com/story/OEG20010718S0073.

Kanayama, Y., Kimura, Y., Miyazaki, F. & Noguchi, T.(1990). A Stable Tracking Control Method for an Autonomous Mobile Robot. *In Proceedings of IEEE Conference on Robotics and Automation*. 384-389 Ohio.

Kaufman, D.M., Edmunds, T. & Pai, D.K. (2005). Fast frictional dynamics for rigid bodies. *ACM Transactions on Graphics (SIGGRAPH 2005)*, 24(3), 946-956 August 2005.

Khatib, O., Yokoi, K. Chang, K., Ruspini, D., Holmberg R. & Casal,A. (1996). Coordination and Decentralized Cooperation of Multiple Mobile Manipulator. *Journal of Robotic Systems* 13(11) 755-764.

Khosla, P. & Kanade, T. (1988). Experimental evaluation of nonlinear feedback and feed-forward control schemes for manipulators. *International Journal of Robotics Research* 7, pages 18--28.

Kragic, D. & Christensen, H.I. (2001). Cue Integration for Visual Servoing. *IEEE Transactions on Robotics and Automation*, 17(1), 18-27.

Kriegman, D.J., Triendl, E. & Binford, T.O. (1989). Stereo vision and navigation in buildings for mobile robots. *IEEE Transactions on Robotics and Automation,* 5(6), 792-803.

Laumond J.P.(Ed.) (1998). *Robot motion planning and control*. Berlin:Springer

Lambrinos, D., Moller, R., Labhart, T., Pfiefer, R. & Wehner., R. (2000). A mobile robot employing insect strategies for navigation. *Robotics and Autonomous. Systems*., 30, 39-64.

Ma, Y., Koseck'a, J., & Sastry S. (1999). Vision guided navigation for a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, 15(3), 521-536.

Malis, E., Chaumette, F. & Boudet, S. (1999). 2 1/2 D visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2), 238-250.

Manocha, M.D. & Canny, J.F. (1994). Efficient inverse kinematics for general 6R manipulators. *IEEE Transactions on Robotics and Automation*, 10 (5), 648-657.

Martinet, P. (1999). Comparison of visual servoing techniques: experimental results. *In Proceedings of the European Control Conference*, 1059-1063. Karlsruhe, Germany.

Martinet, P. & Gallice, J. (1999). Position based visual servoing using a non-linear approach. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*,. 531-536, Kyongju, Korea.

Mason, M.T., Pai, D., Rus, D., Taylor, L., & Erdmann, M.A. (1999). A mobile manipulator. *In Proceedings of the IEEE International Conference on Robotics and Automation*, 2322-2327, Detroit MI.

McKerrow, P. (1991). *Introduction to robotics*. Wokingham: Addison-Wesley.

Murray, R., Li, Z. & Sastry, S. (1994). *A Mathematical introduction to robotic manipulation*. NY:CRC Press.

Namiki, A., Hashimoto, K. & Ishikawa, M. (2003). A hierarchical control architecture for high-speed visual servoing. *International Journal of Robotic Research*, 22(10-11): 873-888.

Ohya, A., Kosaka, A. & Kak, A.C. (1998). Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing. *IEEE Transactions on Robotics and Automation*. 14(6), 969-978.

Papadopoulos, E. & Poulakakis, J. (2000). Trajectory planning and control for mobile manipulator systems. In Proceedings of 8th IEEE Mediterranean Conference on Control & Automation , 11-16.

Sawicsz, D. (2002). *Hobby servo fundamentals.* Retrieved January 5, 2008, from http://www.princeton.edu/~mae412/TEXT/NTRAK2002/292-302.pdf.

Shirai, Y. & Inoue.H. (1973). Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5, 99-108.

Spong, M.W., Hutchinson, S. & Vidyasagar, M. (2005). *Robot modeling and control*. NY: Wiley.

Sturm, P.& Maybank S.(1999). On plane-based camera calibration: A general algorithm, singularities, applications. *In proceedings of Conference on Computer Vision and Pattern Recognition*, 432-437. Corfu, Greece.

Sun, C. & Peleg, S. (2004). Fast panoramic stereo matching using cylindrical maximum surfaces. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(1). 760-765.

Taylor, G., & Kleeman, L.(2006). *Visual perception and robotic manipulation*. Berlin: Springer.

Trucco, E. & Verri, A. (1998). *Introductory techniques for 3-d computer vision.* New Jersey: Prentice-Hall.

Tsai, L.W. (1999). *Robot Analysis* New York: Wiley.

Tsai, R. Y. (1987). A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics Automation*, RA-3 (4) 323-344

Walsh, G., Tilbury, D., Sastry, S., Murray, R., & Laumond J.P.(1994). Stabilization of trajectories for systems with nonholonomic constraints. *IEEE Transactions on Automatic Control*, 39(1):216-222.

Weiss, L., Sanderson, A.C., & Neuman, C.P. (1987). Dynamic Sensor-Based Control of Robots with Visual Feedback. *IEEE Journal on Robotics and Automation*, RA-3(5), 404-417.

Yamamoto, Y., Yun, X. (1994). Coordinating locomotion and manipulation of a mobile manipulator in Zheng, Y. (Ed.) *Recent Trends in Mobile Robots*, (16-26). NY: World Scientific.

Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. *In proceedings of International Conference on Computer Vision*, 666-673. Corfu, Greece.

APENDIX 1

NOMENCLATURE

Table A.1 Abbreviations used in thesis

Subscripts	
0	World reference frame
a	Angular
c	Camera
L,l	Left
l	Linear
R,r	Right
m	Manipulator reference frame
w	Wrist reference frame
v	Vehicle reference frame
Terms	
C	Centroid of target
d	Distance
i,j	Pixel indices on image
f	Focal distance
F_0	World Coordinate Frame
F_c	Camera Coordinate Frame
F_m	Manipulator Coordinate Frame
F_w	Wrist Coordinate Frame
F_v	Vehicle Coordinate Frame
${}^{0}H_{v}$	Homogenous transformation matrix from camera c.f. to world c.f
${}^{0}H_{v}$	Homogenous transformation matrix (relates points in F_v to F_0)
${}^{G}H_{L}$	Homogenous transformation matrix from left camera c.f. to grid c.f
${}^{m}H_{v}$	.Homogenous transformation matrix from vehicle c.f. to manipulator c.f.
${}^{m}H_{0}$	Homogenous transformation matrix from world c.f. to manipulator c.f.
J	Jacobian Matrix
O	Origin of coordinate frame

${}^{L}o_{R}$	Origin of right camera frame (o_R) represented in left camera frame (L)
${}^{R}R_{L}$	Rotation matrix from left coordinate frame to Right Coordinate Frame
${}^{E}r$	Position vector in (E) end effector reference frame
${}^{m}r$	Position vector in (m) manipulator reference frame
${}^{w}r$	Position vector in (w) wrist reference frame
S	Stationary points representing target positions in reference frames.
${}^{R}T_{L}$	Translation vector from left coordinate frame to Right Coordinate Frame
τ	Torque
ω,Ω	Angular velocity
ω_a	Angular velocity of mobile platform
V,v	Translational velocity
V_l	Linear velocity of mobile platform