AUTONOMOUS JOINT CALIBRATION USING ADAPTIVE CONTROL

Oliver Richard Astley

Department of Electrical Engineering
McGill University, Montréal

February 1996

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements of the degree of
Master of Engineering

ISBN 0-612-12108-9

Abstract

The autonomous calibration of a manipulator is considered with respect to both dynamic and joint sensor properties. Using methods based on adaptive control, a new formulation is introduced such that bench calibration of the robot joint sensors and actuators is no longer necessary. When adaptive control is used in identification, inaccuracies caused by various sources of noise are averaged out because the identification takes place on-line, this is in contrast to static methods which rely on a limited number of input data. This method is unique because the joint calibration is done with respect to invariant forces due to gravity loading. The method also guarantees convergence to the true values from arbitrary initial estimates. Experimental results are presented which were performed on two links of a a six degree of freedom hand-controller. Results show that angles can be recovered to an accuracy of $\pm 1.5^o$ in the absence of initial estimates. From both the theoretical derivations and experiments, the properties and performance of the algorithm are discussed. Conclusions and topics for future research are presented.

Résumé

On considère la calibration autonome des paramètres des capteurs articulaires et des propriétés inertielles pour un robot manipulateur. On introduit une nouvelle formulation dérivée de la commande adaptative qui éliminine la nécessité d'une calibration sur banc d'essais. La commande adaptative est utilisée comme outil d'identification des paramètres "en ligne", ce qui fait que le bruit est filtré sur de grandes quantités de données qui ne sont pas enregistrées. Ceci s'oppose aux méthodes statiques qui se basent sur des quantités limitées de données enregistrées à l'avance. L'originalité de la méthode tient au fait que la calibration se base sur les forces invariantes de la gravité. De plus, la convergence de l'estimation vers les vraies valeurs est guarantie en l'absence d'estimées initiales. On présente aussi des résulats expérimentaux effectués sur deux des articulations d'un palonnier à six degrés de liberté. Les angles peuvent être obtenus avec une précision de $\pm 1.5°$ sans avoir aucune estimation initiale. On discute les dérivations théoriques, les expériences, les propriétés et la performance de l'algorithme, ainsi que des conclusion et des questions de recherches futures.

Acknowledgements

I would like to take this opportunity to thank some people who have given me guidance and encouragement throughout my Master's degree.

First I must thank my supervisor, Dr. Vincent Hayward, for constant encouragement, ideas, and interesting conversation throughout my Master's Degree work. This thesis would not have been possible without him. I also thank him for showing me, by example, that understanding of the fundamental concepts in engineering promotes better understanding of complex problems and original research.

I would also like to thank everybody at the McGill Centre for Intelligent Machines; everybody at the centre contributes to a very interesting and friendly working environment. In particular I must thank Michel Doyon and Danny Grant for their assistance in maintaining and improving the hardware and software in which the experimental work was performed; their help has been invaluable. My office partners Manuel Cruz, Bin Mu, and Soumen Sarkar have made the office a fun and productive place throughout my research. Thanks also to Michael Glaum for proof reading the thesis.

Dr. R.V. Patel at Concordia University has given me many opportunities over the last three years. I am greatly indebted to him for challenging me in robotics during my undergraduate degree, writing countless scholarship recommendations, and giving me sound guidance.

During the writing of this thesis, I could always rely on Sam Kaseir, Kosta Vitoratos, and Paul McAdams, for an escape and good conversation Friday nights at Thompson House. Thanks!

A constant throughout my life has been the support of my parents, Rick and Anne Astley and my younger brother Thomas Astley. Their unconditional love, guidance, and support, is a priceless gift which I have received from them. I must also thank my wife's family, Vimlesh, Ashwani, and Sunil Peetush for their love and support.

Most of all, I thank my beautiful, loving, and caring wife Anita. She is supporting and considerate in all that I do, and makes even the toughest day enjoyable. Her unrelenting faith in me means so much. Anita, I dedicate this to you.

Claim of Originality

The proceeding components of this thesis, to the author's knowledge, are original contributions to the field of robotics.

- Calibration of robot joint position sensors using gravity alone.
- Demonstration of polynomial approximations to formulate robot dynamics in the regressor form.
- Formulation of adaptive control for autonomous estimation of joint sensor parameters.
- Experimental evidence of the strengths and limitations of the adaptive control approach.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Humans and animals possess the ability to autonomously calibrate muscles and sensors by continuous interaction with the environment around them. Although sensors and actuators used in robotics are very different, the precedent set by humans and animals presents an interesting challenge, Can a robot's joint sensors be calibrated solely through interaction with the environment?

Joint calibration schemes have, in general, relied on either constraining the robot to a known position in space, required extra measurement devices, or needed a preset reference within the joint. These constraints prevent the robot from being truly autonomous. The importance of autonomy is evident when a robot is operated in a hazardous and unknown environment. During operation, a hardware or software failure may cause the robot to loose calibration. In the event that the robot cannot autonomously regain its calibration, it will be rendered inoperable. Therefore the calibration task requires a minimum of a priori information and only ordinary environmental interaction.

The closest robot control has come to autonomous calibration is in the field of adaptive control. In order to optimize a robot's performance, adaptive controllers are used to adjust gains and system model parameters by using feedback from past and present information. In the process, the dynamic parameters of the system can be estimated. In the mid 1980's Slotine and Li proposed an adaptive controller which, using a robot's dynamic model, was globally convergent with respect to both trajectory tracking and dynamic parameters [46]. By using the full dynamic model (which is nonlinear) within the control, information directly related to the parameters of the system could be found. What is more, a priori knowledge of the dynamic parameters is not required, making the controller autonomous. The importance of their adaptive controller lies also in its practicality for actual robotic systems. Neither measurement of joint accelerations nor inversion of (possibly singular) matrices is required. However, the algorithm presented by Slotine and Li, requires the joints to be fully calibrated.

The joint calibration problem involves finding a relationship between the joint sensor output and the joint angle or displacement. For most sensors the relationship is linear, such that a joint gain (α), and a joint offset (β) must be found for each joint. With the adaptive control method of Slotine and Li in mind, the ideal scenario would be to estimate the joint calibration parameters within the adaptive control framework. Unfortunately this is not possible because all unknown parameters must occur linearly with respect to joint positions and velocities. In the case of rotational joint sensors, the sensor model parameters generally occur within transcendental functions such as $\cos(\alpha q + \beta)$; this prevents the parameters being written in the linear fashion.

This thesis presents a new method to enable joint calibration through use of the Slotine and Li Composite Adaptive Control Algorithm. In essence, the algorithm replaces transcendental terms, which occur within the gravity vector of the robot dynamics, with polynomial approximations. It is well known that a polynomial can be used to approximate a region of a nonlinear function with arbitrary accuracy; what is more, the coefficients which define the shape of the polynomial occur linearly with respect to the known parameters of the system. For example, $\cos(\theta)$ can be approximated by $a + b\theta + c\theta^2$. The parameters, a, b, and c can be estimated within the Slotine and Li adaptive control framework and then used to obtain joint calibration information.

The complete algorithm estimates both joint calibration and dynamic parameters, requiring no human intervention, special equipment, or physical constraints on the robot. Experimental results show that joint gains and offsets can be calibrated to an accuracy of approximately $\pm 1.5°$. From this experimental work several important properties are evident, most notably the global convergence of joint calibration parameters. In general, methods which rely on static input output data experience problems in numerical stability and convergence, requiring good initial estimates of the unknown parameters. The adaptive control based calibration did not experience such problems and achieved convergence from arbitrary initial conditions.

The thesis is organized as follows. Chapter 2 presents a review of the work done in both joint calibration and adaptive control schemes. Chapter 3 introduces the adaptive control methods introduced by Slotine and Li, paying particular attention to the innovations of the algorithm which make it suitable for calibration and practical application. In Chapter 4 the new scheme for joint calibration is presented. Within the theoretical development, attention is given to the overall properties of the calibration scheme which are evident from the theory alone, including practicality, weaknesses, and strengths. Chapter 5 presents the results of the algorithm when applied to a real robotic system – a force feedback hand-controller. Problems and solutions, not evident from the original theory, which arose during implementation are discussed. Also, weaknesses and strengths of the algorithm are presented. Finally, Chapter 6 presents conclusions, and suggests improvements and considerations for future work.

CHAPTER 2

Literature Review

The union of adaptive control and joint sensor calibration is novel to this thesis. Nevertheless, active research in the respective topics has been performed over several decades. Adaptive control methods have evolved to the point that unknown parameters in nonlinear systems can be accurately estimated; this makes the adaptive control parameter estimation schemes applicable to calibration.

1. Evolution of Adaptive Control in Robotics

The need for nonlinear control methods in robotics originated because linear control, such as PID feedback, did not satisfactorily address the handling of nonlinearities and couplings which exist in robot manipulator systems. For these reasons, linear feedback, although reliable and simple to implement, gives conservative performance. It was realized that nonlinear controllers based on the dynamic properties of a manipulator could, if properly implemented, vastly increase performance with respect to both speed and accuracy.

Unfortunately the dynamic parameters are not easy to measure. What is more, when a manipulator picks up a payload with unknown inertial properties, the dynamic parameters of the system will change. It was realized that nonlinear control methods, which are based on a plant with constant parameters, were not robust to parameter uncertainty while the robot was in operation. This not only decreased the precision of the robot, but could also cause instability. To counter these problems adaptive control were sought.

The motivating concept in adaptive control is that the system parameters, be they gains or plant parameters, could be adjusted on-line from empirical data to optimise performance. For example, when a manipulator picks up a payload, the controller would automatically update the affected parameters to reflect the change in the system and hence optimise performance.

In the early days of adaptive control, research was directed toward linear systems. It was not until the 1980's that attention was directed to the multi-variable, coupled, nonlinear systems found in robotics. Proposed solutions can be split into two categories: Model Reference Adaptive Controllers (MRAC) and Self-Tuning Adaptive Controllers (STAC) [47].

1.1. Model Reference Adaptive Control A typical MRAC system is shown in Figure 2.1 [47]. The plant is considered to have a known structure, however, some or all of the parameters of the system are unknown. The reference model is a representation of how the system should ideally perform. The choice of model is dependent on control engineering trade offs such as sensitivity, settling time, and complexity. The error measure between the model and the actual plant is used to adjust the parameters of controller using a derived parameter update law. The combination of the adaptation law and controller should ensure both stability and convergence. In nonlinear systems, this has generally been assured using Lyapunov Stability analysis [22], hyperstability [33], or passivity theory [13] [40].

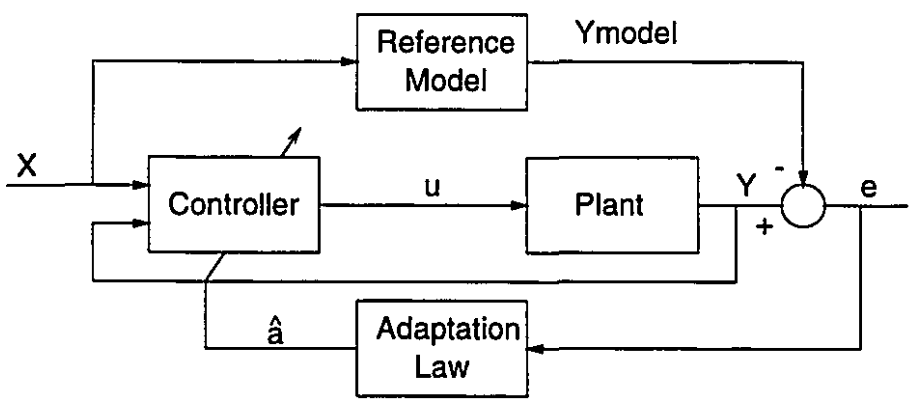


Figure 2.1: Block Diagram of a Model Reference Adaptive Control System

Existing MRAC systems generally do not take advantage of the full robot dynamic model which can normally derived for serial manipulators. Therefore, from the perspective of calibration, MRAC methods yield little useful information.

1.2. Self Tuning Adaptive Controllers Self-tuning adaptive controllers attempt to identify system parameters on-line by minimising the input-output error between the actual plant and the plant model. A block diagram of a typical STAC is shown in Figure 2.2.

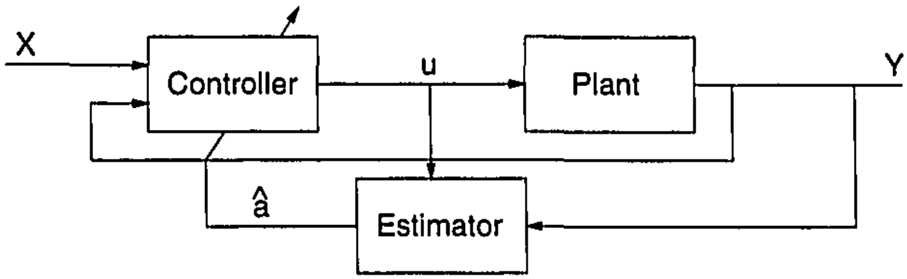


Figure 2.2: Block Diagram of a Self Tuning Adaptive Control System

The estimator updates unknown parameters of the plant based on differences between the dynamic response of the plant and the predicted response of the plant dynamic model. The parameter estimates are then used within the nonlinear controller.

Methods prior to 1985 relied on some sort of restriction on the controller or system; for example, linearization of dynamics, decoupling of dynamics equations, or a slow rate of change in the inertia matrix. By imposing restrictions of this nature on the system, the adaptive control problem becomes easier to manage [3] [22] [26] [50].

After this period, globally convergent adaptive controllers were proposed which did not require such restrictions. A catalyst for this generation of adaptive controllers was the regressor model of the Lagrangian robot dynamics.

Robot dynamics equations are most commonly expressed in the form,

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \tag{2.1}$$

where τ is the force or torque input, $M(\theta)$ is the inertia matrix, $C(\theta, \dot{\theta})$ is the centrifugal and Coriolis terms, $G(\theta)$ is the contribution due to gravity, and θ is a vector of joint angles. In Khosla and Kanade, the dynamic equation (2.1) was expressed linearly in terms of the physical parameters ($\bar{a}$) of the system [24]

$$\tau = Y(\theta, \dot{\theta}, \ddot{\theta})\bar{a}. \tag{2.2}$$

Equation (2.2) is known as the regressor form of the dynamics. The vector $\bar{a}$ comprises the unknown parameters of the system. Generally this includes masses, link dimensions, and even friction constants. The regressor form is *not* an approximation of the robot dynamics, but rather an alternative format for the dynamics equations which isolates the physical parameters. Using this parameterisation method, globally convergent adaptive control algorithms were developed.

1.3. The Algorithm of Craig, Hsu, and Sastry The adaptive control algorithm of Craig *et al.* implements adaptive feedback linearization of a nonlinear system [11] [12]. Feedback linearization utilises both a nonlinear inner loop and a linear outer feedback loop. The nonlinear inner control loop is used to cancel the nonlinearities of the plant [10]. The nonlinearities, however, will remain if the model of the robot does not exactly match the physical system. This may lead to unpredictable performance and instability. For this reason, adaptive methods are required to continuously update the model to ensure the best possible cancellation of nonlinearities. Figure 2.3 shows the block diagram of the Craig *et al.* controller [11].

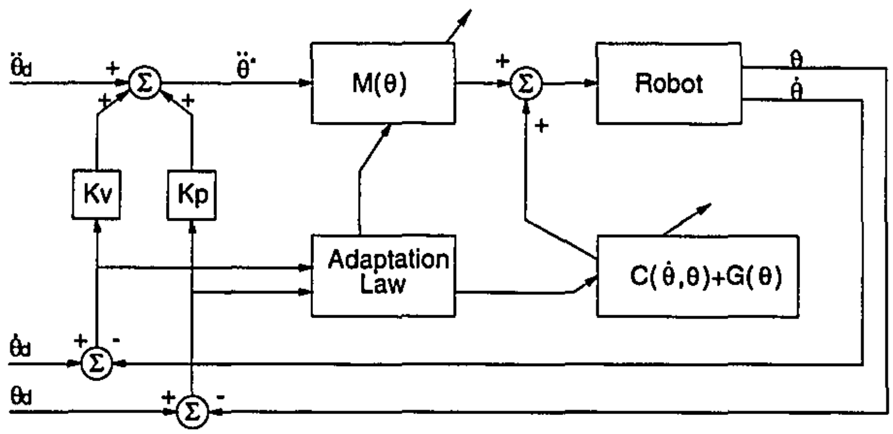


Figure 2.3: Block Diagram of the Craig *et al.* Feedback Linearization Adaptive Control System

The control law for the nonlinear controller is:

$$\tau = \hat{M}(\theta)\ddot{\theta}^{*} + \hat{C}(\theta,\dot{\theta}) + \hat{G}(\theta) \tag{2.3}$$

where $\hat{M}$, $\hat{C}$, and $\hat{G}$ are the estimates of the real physical matrices M, C, and G of the robot. Defining E as the joint position error $\theta_d - \theta$, and $\dot{E}$ as the joint velocity error $\dot{\theta}_d - \dot{\theta}$, θ^{*} is described by:

$$\begin{aligned} \ddot{\theta}^{*} &= \ddot{\theta}_d + K_v(\dot{\theta}_d - \dot{\theta}) + K_p(\theta_d - \theta) \\ &= \ddot{\theta}_d + K_v\dot{E} + K_pE \end{aligned} \tag{2.4}$$

The linearization of the system becomes apparent when (2.1) is equated with (2.3).

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) = \hat{M}(\theta)\ddot{\theta}^{*} + \hat{C}(\theta,\dot{\theta}) + \hat{G}(\theta) \tag{2.5}$$

expanding and re-arranging gives,

$$\begin{aligned} \ddot{E} + K_v\dot{E} + K_pE &= \hat{M}^{-1}(\theta)[M(\theta)\ddot{\theta} - \hat{M}(\theta)\ddot{\theta} + C(\theta,\dot{\theta}) - \hat{C}(\theta,\dot{\theta}) + G(\theta) - \hat{G}(\theta)] \\ &= \hat{M}^{-1}(\theta)[\tilde{M}(\theta)\ddot{\theta} + \tilde{C}(\theta,\dot{\theta}) + \tilde{G}(\theta)] \\ &= \hat{M}^{-1}(\theta)Y(\theta,\dot{\theta},\ddot{\theta})\tilde{a} \end{aligned} \tag{2.6}$$

where Y is the regressor form of the dynamics and $\tilde{a}$ is the error between the physical parameters ($\bar{a}$) and the estimated parameters ($\hat{a}$). Using (2.6) and Lyapunov stability criteria the adaptation law is derived as:

$$\dot{\hat{a}} = \Gamma Y^T \hat{M}^{-1} E_1 \tag{2.7}$$

where Γ is a positive definite gain matrix, and E_1 is a filtered joint error measure.

The Craig *et al.* algorithm was pivotal to adaptive control in robotics because it was the first algorithm which did not add constraints to the nonlinear dynamics and, most importantly, had a global trajectory convergence proof.

Unfortunately, from an applied perspective, the Craig *et al.* algorithm has two major limitations, both evident in (2.6) and (2.7).

- Inversion of the inertia matrix (M) is required. Although physically the inertia matrix is always positive definite, there is not a guarantee that the estimated inertia matrix $\hat{M}$ will also be positive definite. This is especially true at startup when parameters are assumed unknown. Therefore, $\hat{M}$ must be monitored for positive definiteness.
- Measurement or estimation of joint acceleration is required. Acceleration measurements are notoriously noisy; hence their use in control can cause loss of performance and even instability.

Simulation results nevertheless, showed that under ideal conditions the adaptive controller was superior to that of a classical PD controller for a two link robot.

With respect to potential application in calibration techniques, the Craig *et al.* algorithm is not satisfactory. Despite the use of the robot dynamic model, there is no guarantee that the dynamic parameters of the system ($\hat{a}$) will converge to the physical values. It can be shown in simulation that for a two degree of freedom robot, the parameters of the system rarely converge to their simulated values. Therefore, the lack of guarantee of parameter convergence excludes the Craig algorithm for calibration methods.

1.4. The Adaptive Control Methods of Slotine and Li The implementation problems associated with the preceding algorithm motivated the work of Slotine and Li. In [41], [42], and [44] a globally convergent adaptive controller is developed which does not require inversion of the inertia matrix or measurement of joint accelerations. The theoretical background to the Slotine and Li adaptive control techniques is given in Chapter 3.

The original Slotine and Li algorithm in [42] and [44] is similar to the Craig *et al.* algorithm because they both use the full robotic dynamic structure in the regressor form, and the parameter update algorithm is driven by joint tracking errors. However, the Slotine and Li algorithm does not attempt to feedback linearise the system. Instead, the algorithm is centred about an important property of the robot dynamics.

It was shown by Koditschek [25] that the matrices M and C in the robot dynamics (2.1) are not independent. Due to conservation of energy, the derivative of the system kinetic energy, equals the power input to the system. This implies:

$$\frac{1}{2}\frac{d}{dt}\dot{\theta}^T M(\theta)\dot{\theta} = \dot{\theta}^T[\tau - G(\theta)]. \tag{2.8}$$

The gravity term is subtracted from the torque input since $G(\theta)$ is a potential energy term. The implication of (2.8) is:

$$\dot{\theta}^T \left(\frac{1}{2}M(\theta) - C(\theta,\dot{\theta})\right)\dot{\theta} = 0 \tag{2.9}$$

The expression $\frac{1}{2}M(\theta) - C(\theta,\dot{\theta})$ in (2.9) is therefore skew symmetric. (A proof is given in [48] page 143.) Using the property of (2.9), the control law is derived using Lyapunov stability theory without the requirement of inverting the inertia matrix. Measured joint acceleration is eliminated from control by defining the joint errors on a sliding surface. This property of the control law is explained in detail in Chapter 3.

The Slotine and Li method, as with the Craig *et al.* method, falls into the category of *direct* adaptive control. The term "direct" is used because the adaptation is driven by errors in joint tracking.

The Slotine and Li algorithm, although globally convergent, did not ensure parameter convergence. Therefore this algorithm alone is not useful for calibration. However, research in *indirect* adaptive control proved that parameter convergence could be guaranteed.

1.5. Indirect Adaptive Control of Middleton and Goodwin Although still a STAC algorithm, the algorithm of Middleton and Goodwin differs from the previous algorithms because the adaptive controller is driven not by joint tracking error, but by torque error [29]. This is achieved by comparing the torque applied to the system with the estimated torque based on the robot model. The result is an adaptive controller focussed on minimising the error between the model parameters and system parameters rather than trajectory tracking errors. Global tracking ability is proved in [29], but inversion of the inertia matrix is required. By using a first order stable filter, it is shown that use of measured joint acceleration is not required. This idea was later used by Hsu *et al.* [23] in a continuation of the Craig *et al.* controller featured in Section 1.3.

1.6. Composite Adaptive Control of Slotine and Li Composite Adaptive Control is the amalgamation of indirect controllers [27] [28] [29] and the direct adaptive control methods [41] [42] [44]. This amalgamation, known as composite adaptive control, appeared in [43] [45] [46]. Composite Adaptive control methods were shown to guarantee both tracking and exponential parameter convergence under excitatory conditions. The algorithm also maintained the properties of requiring neither measurement of acceleration nor inversion of the mass matrix.

An important improvement of this generation of adaptive controllers is that the speed of convergence was no longer dictated by a constant gain matrix. Originally, the rate of parameter convergence was set by a positive definite gain matrix P. Theoretically, the larger the magnitude of P, the faster the convergence. (In practice however, an upper bound on P is necessary due to noise.) By making P time-varying, the speed of convergence could be adjusted depending on the current conditions in the system. For example, if the parameter estimates are oscillatory, then P must be lowered; conversely, P should be increased when parameter movement over time is small. A mathematical treatment is given for these concepts in Chapter 3.

Composite adaptive control gives both convergence in joint tracking and system parameters. This makes it useful for calibration methods. What is more, without the necessity of inverting the inertia matrix, or measuring joint accelerations, the algorithm is suitable for implementation on a real robot.

1.7. Other Issues in Adaptive Control There has been much research into stability analysis and robustness of adaptive control; so much so, the term robust adaptive control has been coined as a distinct research field. The bulk of this research is concentrated on finding excitatory trajectories and eliminating instability due to unmodeled dynamics, noise, and time varying system parameters.

1.7.1. *Trajectory Issues* One of the major concerns associated with adaptive controllers is that they require the system to be reasonably active. In non-excitatory conditions, inaccuracies in the model can result in poor parameter convergence or parameter drift, ultimately leading to instability.[1] This can be explained by noting that the adaptation schemes are based on dynamic properties of the system. Therefore if the dynamic content of the system is predominantly due to noise, then the adaptation will lock onto the dynamics of the noise rather than the joint feedback.

An example of instability due an absence of an excitatory trajectory is given in [47]. They show Rohrs' example in which a linear system, plus an unmodeled dynamic element, is subjected to a slow trajectory [38].

In the Rohrs example, the actual plant is described by:

$$y(p) = \frac{2}{p+1}\frac{229}{p^2+30p+229}U(p) \tag{2.10}$$

However, the adaptation mechanism is based on a first order model, and hence does not consider the higher order dynamics of the plant. The first order system is described by:

$$H_o(p) = \frac{k}{p+a} \tag{2.11}$$

where k and a are the parameters of the system to be estimated.

The unmodeled poles at $-15 \pm 1j$, although very damped and at a relatively high frequency, cause problems in slow trajectories. In simulation, a constant input command is given with additive noise in the form of a sinusoid with a frequency of 16.1 radians a second. Over a period of just 60 seconds the trajectory output goes from stability, to oscillatory, and ultimately becoming unstable.

For this reason there has been much research into minimally acceptable trajectories required for adaptive controllers. This has been done by Armstrong who defined a condition measure for a trajectory, and used it to optimise existing trajectories [1]. However, when this paper was published the algorithm took over one hour to optimise just one trajectory!

[1] An effective fix for parameter drift is to introduce a dead-band to the parameter update law.

Armstrong's results are not explained from an intuitive level. Graphically his results show the original single frequency sinusoid trajectory being optimised into a multiple frequency trajectory. This frequency content issue, largely overlooked by Armstrong, is reported in Boyd and Sastry [6] [7]. They show that for a nth order linear system, there must be at least $2n$ discrete frequencies for parameter convergence. What is more, if there are less than $2n$ discrete frequencies, the parameters will converge to a subspace of the parameter solutions. There is still no proof of such a condition for nonlinear systems, however it is generally accepted that the trajectory should be frequency rich.

It is possible to measure the richness of a trajectory $\theta(t)$. That are several condition measures, the most widely used being [9]:

$$\zeta I \leq \int_{t_o}^{t_o+\delta t} Y^T(\theta,\dot{\theta},\ddot{\theta})Y(\theta,\dot{\theta},\ddot{\theta})dt \leq \eta I \quad \forall t_o \tag{2.12}$$

where Y represents the system dynamics, and ζ, η, and δt are positive. In the case of regressor techniques, Y would be the regressor matrix.

The necessity for trajectory richness has been one of the main retardants in preventing adaptive control being widely used in real systems. With respect to robot calibration, this is not a critical problem. This is because autonomous calibration is a separate stage of robot operation; therefore the calibration algorithm can chose an arbitrary trajectory. For this reason, one of the major setbacks in adaptive control is effectively bypassed.

1.7.2. *Instability Issues* Most of the adaptive controllers, including the ones studied in this and the proceeding sections, assume that the parameters of the system are time invariant. This is a critical assumption necessary to perform the Lyapunov and passivity analysis. For this reason there has been much research in the effect of the breakdown of this assumption and the possible solutions to it. Reed and Ioannou raise the following question, What happens to a nonlinear adaptive controller when the model does not match the plant or when the parameters are time varying [35] [36]? They show that a bounded disturbance produces a bounded output disturbance. Based on this finding, they derive a controller designed to be more robust to such disturbances. (No experimental results are given.)

1.8. Summary The evolution of adaptive control has matured from conservative linear systems in the late seventies to multi-input multi-output time varying nonlinear systems in the mid eighties. The convergence of both system parameters and trajectories makes such methods applicable to calibration schemes. Fortunately, calibration techniques can specify arbitrary trajectories; thus by-passing one of the major stumbling blocks in the practical application of adaptive control.

2. Joint Calibration Techniques

The importance of joint calibration in robotics cannot be understated.[2] An uncalibrated robot renders most control schemes inoperable. For this reason, there has been much research directed toward calibration of various robot components including: position sensors, force sensors, dynamic properties, and kinematic properties. In general, calibration schemes aim to find either kinematic and joint sensor properties, or dynamic properties. The algorithm presented in this thesis breaks this mold somewhat by finding joint sensor and dynamic properties together. The kinematic and joint sensor calibration algorithms discussed in this section are listed below.

- Pre-set position (Open loop) calibration.
- Constrained calibration.
- Metrology based calibration.

These topics serve to illustrate the issues and problems associated with joint and kinematic calibration.

2.1. Open Loop Calibration Open loop calibration schemes require the manipulator to be set in several known configurations. Using the known end-effector position and joint sensor data, the forward kinematics problem is solved such that:

$$\bar{x} = f(a, \alpha, d, \theta) \tag{2.13}$$

where $\bar{x}$ is a 6 component vector made up of three translations (x, y, and z) and three rotations (roll, pitch, and yaw). a, α, d, and θ represent Denavit Hartenberg parameters which define the position and orientation of each joint of the robot in space. Other constraints can also be added to the problem, in particular, non geometric constraints such as backlash, sensor gains, and elasticity [18].

The disadvantage of the open loop method is that a measuring system is required to perform the calibration. For this reason, closed loop methods have evolved.

2.2. Closed Loop Calibration Methods Closed loop calibration methods (for serial link robots) constrain the robot, normally at the end-effector, to the environment [5]

[2] Hollerbach and Hunter in [20] state: "We should expect to spend most of our experimental effort in calibration, relatively less in actually running the experiments in robot control."

[18]. Therefore the robot and the environment form a closed loop. The calibration is then performed by moving the robot joints while the end effector remains fixed.
Due to the closure of the robot loop through the environment (2.13) now becomes.

$$0 = f(a, \alpha, d, \theta) \tag{2.14}$$

The parameters of the system are estimated based on the consistency of (2.14).

The advantage of this method is that external measurement is not required. However it is still necessary to physically constrain the robot, making any closed loop algorithm not truly autonomous. Unless an additional passive linkage is used, the robot must also have some redundancy to allow the joints to move while the end effector is at the fixed position.

From the perspective of joint sensor calibration, closed loop methods were shown to give poor estimates of the gain parameter α, which relates joint sensor output q and joint displacement or angle θ in the equation [31]:

$$\theta = \alpha q + \beta \tag{2.15}$$

The requirement of fixing the end-effector is impractical for non-redundant robots. For this reason, researchers have developed methods which constrain some of the degrees of freedom at the end-effector while others are allowed to move in free space. This is possible because the forward kinematics equations in (2.13) represent six equations. By eliminating some of the forward kinematics equations, less degrees of end-effector freedom need to be considered. For example, Newman and Osborn use a laser beam on which the robot end-effector is position servoed [32]. By tracking the straight light beam, the task is reduced to just two dimensions. Collected data, from the straight line tracking, is then fit to the kinematic model. The advantage of such a system is its simplicity and low cost. Also no external measuring is required.

An interesting use of closed loop methods has been shown in parallel manipulators, specifically parallel mechanisms which have at least one degree of actuator-sensor redundancy. An example of an actuator and sensor redundant mechanism is the Hayward hydraulic robot shoulder which has three degrees of freedom, but four hydraulic actuators and four position sensors [14]. Due to the redundancy and closed kinematic loops inherent within the mechanism, there is no longer a need to clamp the device, allowing closed loop calibration procedures to be used in a truly autonomous fashion. Examples of using closed loop calibration techniques on this type of mechanism can be found in [21] and [31].

2.3. Metrology Based Calibration A significant branch of robot calibration is based on measuring some or all of the position and rotation components of the end effector. This is done by continuously sensing the position and orientation of the end-effector while the robot is moved in free space. Using these external measurements (X), the forward kinematic equations in (2.13) become:

$$0 = f(a, \alpha, d, \theta) - X \tag{2.16}$$

There are six closed loop equations in (2.16), therefore it is possible to eliminate up to five of the these equations for calibration. This is important because the complexity and cost of the measuring system generally increases with the number of elements in X. Nevertheless, kinematic calibration schemes have been adopted which measure all six of the position and orientation components of X [34]. This method, as with the majority of the metrology based methods, uses laser light in conjunction with interferometry to measure position and orientation. In contrast, Tang and Liu present a metrology based method which measures just one degree of freedom [49]. Renders *et al.* address the problem of measurement equipment complexity by focusing on straight line motion only [37].

2.4. Other Calibration Techniques The bulk of research in calibration has been in the preceding areas, however there has been work in other techniques which do not fall into the categories already discussed. Most notably screw axis techniques [8] [30] and Jacobian based calibration techniques [4] [19]. The Jacobian techniques are interesting because they do not use the forward kinematic relationship common to methodologies in the preceding section. Instead, they focus on the relationship between joint velocity ($\dot{\theta}$) and Cartesian and angular velocity $\dot{V}$, or joint torque τ and Cartesian force and angular torque F. These relationships are specified through the Jacobian matrix J,

$$\dot{X} = J\dot{\theta} \tag{2.17}$$

$$\tau = J^T F \tag{2.18}$$

By using the input output relationships in (2.18), the elements of J are estimated using a minimization strategy.

2.5. Summary The calibration ideas presented in this section are by no means exhaustive. What is common to most of these algorithms is the requirement of equipment which is not "standard" with the robot; for example, lasers, torque sensors, and clamping points. Also, all of the algorithms do not estimate the parameters on-line. Rather, data points are collected and then processed using minimization strategies such as least squares minimization. Although the algorithm presented in this thesis does not attempt to calibrate kinematic parameters, it differs from the majority of these calibration algorithms in that calibration is done continuously on-line, and uses only position sensors for calibration.

3. Summary of Adaptive Control and Calibration Techniques

The adaptive control techniques presented in this chapter show that global model parameter convergence and trajectory convergence can be achieved in an on-line nonlinear controller. What is more, this can be done without putting physical restrictions on the plant or simplifying the robot model. The global convergence property implies that a priori parameter information is not required. Also, the adaptive controllers do not require human intervention or use of non standard measuring equipment. These properties make the adaptive controller autonomous. These adaptive controllers however, all assume that the relationship between joint sensor output and joint angle or displacement is known; i.e. the robot must be calibrated before using adaptive control.

The review of calibration methods presented in this chapter showed that calibration generally required non standard equipment or constraining the robot. This prevents the robot from autonomously calibrating itself. This presents the challenge, Can a robot be calibrated using standard joint sensors and unconstrained environmental interaction? This thesis attempts to answer this question.

Although some adaptive control methods can calibrate robot dynamic parameters, adaptive control techniques have not, so far, entered the field of joint calibration. In Roth *et al.* the line is firmly drawn differentiating these two fields as discrete and continuous events [**39**]. This thesis challenges this notion by using adaptive control to estimate both dynamic parameters and calibrate joint sensors.

There would appear several advantages to using continuous methods to estimate unknown system parameters. Specifically, continuous techniques avoid the use of discrete least squares estimation. Least squares solutions cannot, in general, guarantee global convergence, while Lyapunov based methods can. Also, continuous calibration schemes have the potential to constantly monitor the robot for any deviations in robot parameters which may signal structural, sensor, or actuator failure. This thesis presents a novel method to implement on-line joint calibration within an adaptive control framework. The result is an algorithm which can identify dynamic parameters and joint calibration parameters without the use of non standard equipment, physically constraining the robot, or requiring human intervention.

CHAPTER 3

Adaptive Control Algorithms of Slotine and Li

In this chapter the adaptive control methods developed by Slotine and Li are discussed. They form the basis of the autonomous joint calibration introduced in Chapter 4.

The basic Slotine and Li algorithm introduces an adaptive controller which is globally convergent and does not require measurement of joint acceleration or inversion of the inertia matrix. This method drives parameter adaptation by errors in the trajectory tracking of manipulator; this is known as direct adaptive control. Indirect adaptive control methods however, drive adaptation from prediction errors in the manipulator model. The focus on model errors enables superior parameter convergence when compared to joint error driven direct controllers. However, this comes at the expense of trajectory tracking ability. For this reason, "Composite Adaptive Control" was conceived; it combines both approaches and refines the update mechanisms.

1. The Dynamics of a Robot Manipulator

For many robot manipulators, the structure of the dynamic equations, which characterise the evolution of the mechanical system subject to holonomic constraints, can be derived using techniques such as Euler-Lagrange or Newton-Euler formulations [10] [48]. The dynamics equations are most commonly written in the form:

$$\tau = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) \tag{3.1}$$

where $M(\theta)$ is the inertia matrix, $C(\theta,\dot{\theta})$ is the centrifugal and Coriolis terms, $G(\theta)$ is the contribution due to gravity, and τ is the torque at the joint. The variables θ, $\dot{\theta}$, and $\ddot{\theta}$ are vectors of joint angles or displacements, joint velocities, and joint accelerations respectively. For example, a two degree of freedom planar manipulator shown in Figure

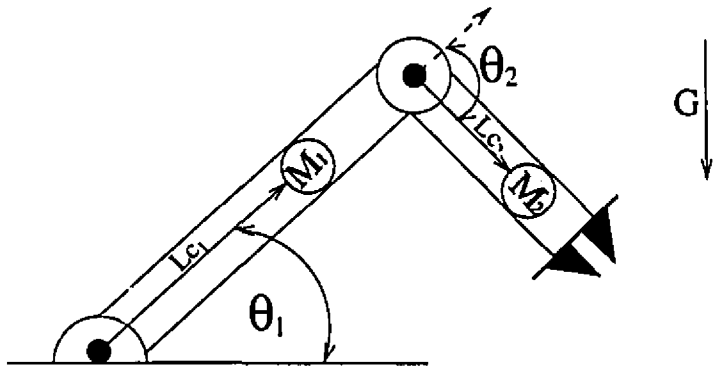


Figure 3.1: A Two Degree of Freedom Planar Robot

3.1 has the dynamics[1]:

$$\begin{bmatrix} m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2 l_1 l_{c2} c_2) & m_2(l_{c2}^2 + l_1 l_{c2} c_2) \\ m_2(l_{c2}^2 + l_1 l_{c2} c_2) & m_2 l_{c2}^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$
$$+ \begin{bmatrix} -m_2 l_1 l_{c2} s_2 \dot{\theta}_2 & -m_2 l_1 l_{c2} s_2 \dot{\theta}_2 - m_2 l_1 l_{c2} s_2 \dot{\theta}_1 \\ m_2 l_1 l_{c2} s_2 \dot{\theta}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$
$$+ \begin{bmatrix} (m_1 l_{c1} + m_2 l_1) g c_1 + m_2 l_{c2} g c_{12} \\ m_2 l_{c2} g c_{12} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{3.2}$$

Where l_{ci} denotes the distance from the link base to the centre of mass of link i, l_i is the link length of link i, m represents mass, and c_1 and c_{12} are defined as $\cos(\theta_1)$ and $\cos(\theta_1 + \theta_2)$ respectively.

It was shown in Chapter 2 that, in general, the mass and link length parameters of the system must be known with reasonable accuracy to ensure stability in control systems which utilise the robot dynamics. Unfortunately, these parameters are generally not known and, in the case of a robot picking up a load, variable. Therefore the task of the adaptive controller is to find these parameters and track them. In the form of (3.2) it is clear that these unknown parameters are not easy to isolate for estimation purposes. This motivates the use of the regressor form of the robot dynamics which express the physical parameters of the system linearly with respect to the known components of the dynamics. The regressor form is written as:

$$\tau = Y(\theta, \dot{\theta}, \ddot{\theta})\bar{a}, \tag{3.3}$$

[1]The model assumes that masses are lumped into point masses

where the vector $\bar{a}$ represents the physical parameters. The dynamics of the planar two link robot written in the regressor form are:

(3.4)

$$\begin{bmatrix} gc_1 & gc_{12} & \ddot{\theta}_1 & \ddot{\theta}_1 + \ddot{\theta}_2 & c_2\ddot{\theta}_1 + c_2\ddot{\theta}_2 - s_2\dot{\theta}_2\dot{\theta}_1 - s_2\dot{\theta}_2^{\,2} - s_2\dot{\theta}_1\dot{\theta}_2 \\ 0 & gc_{12} & 0 & \ddot{\theta}_1 + \ddot{\theta}_2 & c_2\ddot{\theta}_1 + s_2\dot{\theta}_1^{\,2} \end{bmatrix} \begin{bmatrix} m_1 l_{c1} + m_2 l_1 \\ m_2 l_{c2} \\ m_1 l_{c1} + m_2 l_1^2 \\ m_2 l_{c2}^2 \\ m_2 l_{c2}^2 \\ m_2 l_1 l_{c2} \end{bmatrix}$$

Using the regressor form in (3.4) the unknown parameters appear concisely within the $\bar{a}$ vector. In terms of estimation, the problem is reduced to estimating $\bar{a}$.

2. The Basic Adaptive Control Algorithm of Slotine and Li

The most important theoretical property of an adaptive control algorithm is its ability to guarantee convergence and system stability. These conditions are addressed by Slotine and Li through Lyapunov stability theory.

Lyapunov stability theory adopts a positive definite function V as a measure of system energy. Although V is arbitrary, it is most often motivated by functions which describe the total energy within the system. If it can be shown that the rate of change of the energy function ($\dot{V}$), is less than or equal to zero, then the system energy is not increasing and therefore the system is stable. What is more, if $\dot{V}$ is strictly less than zero, or zero only when V is zero, then the system energy will go to zero and convergence is also guaranteed. Mathematically, this can be summarised as:

Given the System: $\dot{x} = f(x,t)$
$V(x,t)$ is a Lyapunov function for f on $G \subset \Re^n$ if $\dot{V}(x,t) \leq 0 \quad \forall x \in G \quad \forall t \geq 0$

When deriving the adaptation mechanism using the Lyapunov technique, the controller must ensure that the Lyapunov stability condition is met. Consequently, the choice of controller is motivated predominantly by the Lyapunov function itself. Also, because the choice of Lyapunov energy function (V) is arbitrary, the controller can, in theory, take as many forms as there are viable energy functions. Therefore an appropriate choice of Lyapunov function for the controller should address both the stability constraints and implementation issues.

Before the introduction of the original Slotine and Li algorithm there were several adaptive controllers which used the Lyapunov technique to meet stability criteria; however the controllers required both measurement of acceleration, and inversion of the inertia matrix.

2.1. The Interdependence of the Robot Dynamics Removing the requirement of inversion of the inertia matrix is achieved by suitable choice of the Lyapunov energy function. In the Slotine and Li algorithm, the choice is motivated predominantly by a property of the robot dynamics.

It has been shown that the inertia matrix M, and velocity matrix C, are not independent, they are related by:

$$\frac{1}{2}\frac{d}{dt}\dot{\theta}^T M(\theta)\dot{\theta} = \dot{\theta}^T[\tau - G(\theta)] \tag{3.5}$$

Physically (3.5) can be interpreted as:

$$\frac{d}{dt}\text{Kinetic Energy} = \text{Power Input} \tag{3.6}$$

Substituting (3.1) into (3.5) and expanding gives:

$$\frac{1}{2}\dot{\theta}^T\dot{M}\dot{\theta} + \dot{\theta}^T M\ddot{\theta} = \dot{\theta}^T[M\ddot{\theta} + C\dot{\theta}] \tag{3.7}$$

$$\dot{\theta}^T[\frac{1}{2}\dot{M} - C]\dot{\theta} = 0 \tag{3.8}$$

which implies that $\frac{1}{2}\dot{M} - C$ is always skew symmetric. This property is applied within the Lyapunov derivation.

2.2. Using a Sliding Surface to Guarantee Trajectory Convergence It is common, when using Lyapunov stability theory, that the strongest stability condition on the derivative of the energy function $\dot{V}(t)$ is:

$$\dot{V}(t) \leq 0 \tag{3.9}$$

Although (3.9) *does* guarantee stability, it *does not* guarantee the elimination of steady state errors. For nonlinear systems there are several techniques used to prove that steady state error can be eliminated. Most notably LaSalle's theorem which can be used to show that:

$$\dot{V}(x,t) = 0 \quad \text{iff} \quad x = 0 \tag{3.10}$$

Since $V(x,t) \geq 0$ and is a positive definite decrescent function, (3.10) implies that $\dot{V}(x,t)$ equals zero *only* when $V(x,t)$ equals zero.

Unfortunately the LaSalle condition in (3.10) is hard to apply, especially for multi-input multi-output (MIMO) non linear systems. Instead Slotine and Li restrict the joint errors to the sliding surface:

$$s = \dot{\tilde{\theta}} + K_P\tilde{\theta} \tag{3.11}$$

where $\dot{\tilde{\theta}}$ is the joint velocity error, $\tilde{\theta}$ is the joint position error, Λ is a constant gain matrix which has all eigenvalues on the right half plane, and s represents the sliding surface. By defining a virtual trajectory, $\dot{\tilde{\theta}}_r = \dot{\tilde{\theta}} + K_P\tilde{\theta}$, it is possible to define θ_r and its derivatives in terms of desired and measured variables.

$$\theta_r = \theta_d - K_P \int_0^t \tilde{\theta}\, dt \tag{3.12}$$

$$\dot{\theta}_r = \dot{\theta}_d - K_P\tilde{\theta} \tag{3.13}$$

$$\ddot{\theta}_r = \ddot{\theta}_d - K_P\dot{\tilde{\theta}} \tag{3.14}$$

Through the derivation of the controller in Section 2.3 it will be shown that (3.12) is not required; hence calculation of the integral term is not necessary.

If it can be shown that the sliding surface $s \to 0$ by proving $\dot{V} \leq 0$, then this will imply that both position error $\tilde{\theta}$, and velocity error $\dot{\tilde{\theta}}$, both go to zero due to the relationship in (3.11).

2.3. The Adaptive Control Derivation In this section the adaptive controller is derived using both the sliding surface and the skew symmetric properties discussed in the preceding sections.

Before introducing the adaptive controller derivation, it is necessary to define several parameters. Joint error is defined as $\tilde{\theta} = \theta - \theta_d$. Parameter estimate error is defined as $\tilde{a} = \hat{a} - \bar{a}$, where $\hat{a}$ is a vector of the adaptive control estimates of the physical parameters $\bar{a}$. Estimates of dynamics matrices are shown as, $\hat{M}$, $\hat{C}$, and $\hat{G}$. The error between the real and the estimated dynamic matrices are defined as: $\tilde{M} = \hat{M} - M$, $\tilde{C} = \hat{C} - C$ and $\tilde{G} = \hat{G} - G$. There are also several positive definite gain matrices used in the formulation: proportional feedback gain K_P, velocity feedback gain K_D, and parameter gain matrix P. The properties and affect of these gain matrices on the system are important to successful implementation; this is discussed in Section 2.6.

The Lyapunov function candidate proposed by Slotine and Li in [42] is:

$$V(t) = \frac{1}{2}s^T M(\theta)s + \frac{1}{2}\tilde{a}^T P\tilde{a} \tag{3.15}$$

The function $V(t)$ is positive definite, for all time t. Differentiating $V(t)$, gives:

$$\dot{V}(t) = s^T \left[M(\theta)\ddot{\theta} - M(\theta)\ddot{\theta}_r\right] + \frac{1}{2}s^T\dot{M}(\theta)s + \tilde{a}^T P\dot{\tilde{a}} \quad (3.16)$$

Substituting $M\ddot{\theta}$ from the system dynamics in (3.1),

$$M\ddot{\theta} = \tau - C(\theta,\dot{\theta})\dot{\theta} - G(\theta) \quad (3.17)$$

$$= \tau - C(\theta,\dot{\theta})(s+\dot{\theta}_r) - G(\theta) \quad (3.18)$$

Substituting (3.18) into (3.16) and using the skew symmetric property of (3.8), $\dot{V}(t)$ becomes:

$$\dot{V}(t) = s^T \left[\tau - M(\theta)\ddot{\theta}_r - C(\theta,\dot{\theta})\dot{\theta}_r - G(\theta)\right] + \tilde{a}^T P\dot{\tilde{a}} \quad (3.19)$$

The input to the system τ is defined as a combination of the feed-forward nonlinear robot dynamics and linear feedback:

$$\tau = \hat{M}\ddot{\theta}_r + \hat{C}(\theta,\dot{\theta})\dot{q}_r + \hat{G}(\theta) - K_D s. \quad (3.20)$$

Note that in (3.20) the linear feedback term comprises both derivative and proportional feedback because:

$$K_D s = K_D\left(\dot{\tilde{\theta}} + K_P\tilde{\theta}\right)$$

$$= K_D\dot{\tilde{\theta}} + K_P K_D\tilde{\theta}. \quad (3.21)$$

Applying (3.20) to (3.19) gives.

$$\dot{V}(t) = s^T \left[\tilde{M}(\theta)\ddot{\theta}_r + \tilde{C}(\theta,\dot{\theta})\dot{q}_r + \tilde{G}(\theta) - K_D s\right] + \tilde{a}^T P\dot{\tilde{a}} \quad (3.22)$$

Rewriting (3.22) using the regressor form of the dynamics gives:

$$\dot{V}(t) = -s^T K_D s + \tilde{a}^T \left[P\dot{\tilde{a}} + Y^T s\right] \quad (3.23)$$

The Lyapunov stability criteria requires that:

$$\tilde{a}^T \left[P\dot{\tilde{a}} + Y^T(\theta,\dot{\theta},\dot{\theta}_r,\ddot{\theta}_r)s\right] = 0 \quad (3.24)$$

To achieve this condition the adaptive control must be implemented such that:

$$P\dot{\tilde{a}} + Y^T(\theta,\dot{\theta},\dot{\theta}_r,\ddot{\theta}_r)s = 0 \quad (3.25)$$

Under the assumption that the system parameters are constant implies that $\dot{a} = 0$. This leads to the parameter update law,

$$\dot{\hat{a}} = -P^{-1}Y^T(\theta,\dot{\theta},\dot{\theta}_r,\ddot{\theta}_r)s \quad (3.26)$$

Applying (3.26) to (3.23) gives:

$$\dot{V}(t) = -s^T K_D s \leq 0 \quad \forall t \geq 0 \tag{3.27}$$

Using (3.27) and the sliding surface defined in (3.11) both stability and trajectory convergence is shown. Since $\dot{V}(t) \leq 0$ and $V(t)$ is lower bounded by 0, this implies that the energy converges to either 0 or a finite positive constant. This convergence shows from (3.27) that $s \to 0$. Since $s \to 0$ both θ and $\dot{\theta}$ converges to 0 as $t \to \infty$. Therefore global joint trajectory convergence is ensured. A detailed proof of the trajectory convergence is given in [42].

Examination of the control input in (3.20) and the parameter update law (3.26) shows that acceleration measurements and inversion of the inertia matrix $M(\theta)$ is not required.

3. Indirect Adaptive Control Methods

Indirect adaptive control methods drive adaptation by monitoring prediction error in the estimated robot model.[2] The consequence of using model prediction errors is that there is a greater emphasis on parameter convergence than trajectory convergence which was the focus in Section 2. In this section the methods used in indirect adaptive control are explained. Specifically, the form of the parameter update law, which differs from direct methods, and the methodology used to avoid measurement of joint acceleration is introduced.

3.1. Filtering to Avoid Measurement of Joint Accelerations Indirect adaptive control methods also adopt the regressor form of the dynamics to linearise the unknown parameters in terms of known measurements, such that:

$$\tau = Y(\theta, \dot{\theta}, \ddot{\theta})\bar{a} \tag{3.28}$$

Measurement of joint acceleration in (3.28) is avoided by filtering both sides of (3.28) with a first order stable filter of the form:

$$F(s) = \frac{b}{s+b} \tag{3.29}$$

where b is a positive constant and s is the Laplace operator. The application of the filter can be viewed from both the frequency domain and the time domain perspective. Applying $F(s)$ to both sides of (3.28) gives:

$$\begin{aligned} F(s)\tau &= F(s)[M(\theta)\ddot{\theta} + C(\dot{\theta}, \theta)\dot{\theta} + G(\theta)] \\ &= F(s)[C(\dot{\theta}, \theta)\dot{\theta} + G(\theta)] + F(s)M(\theta)\ddot{\theta} \end{aligned} \tag{3.30}$$

[2]This differs from the direct adaptive control methods which use joint tracking errors to drive adaptation.

Manipulating the last term in (3.30), Hsu *et al.* show **[23]**:

$$\frac{b}{s+b}M(\theta)\ddot{\theta} = \frac{bs}{s+b}M(\theta)\dot{\theta} - \frac{b}{s+b}\dot{M}(\theta)\dot{\theta} \tag{3.31}$$

Alternatively, the filtering action can be analysed in the time domain by convolution **[47]**. Applying a filter $f(t)$ to the acceleration dependent term in (3.30):

$$\begin{aligned}\int_0^t f(t-r)M(\theta)\ddot{\theta}dr &= f(t-r)M\dot{\theta}\ \Big|_0^t - \int_0^t \frac{d}{dr}[M(\theta)\dot{\theta}]\,dr \\ &= f(0)M(\theta)\dot{\theta} - f(0)M[\theta(0)]\dot{\theta}(0) \\ &\quad - \int_0^t [f(t-r)\dot{M}(\theta)\dot{\theta} - \dot{f}(t-r)M(\theta)M(\theta)\dot{\theta}]\,dr \end{aligned} \tag{3.32}$$

The filtered dynamics can now be defined as:

$$F(s)Y(\theta,\dot{\theta},\ddot{\theta}) \equiv W(\theta,\dot{\theta}) \tag{3.33}$$

The filtering technique relinquishes the need to measure joint acceleration; however, this comes at the cost of loosing higher frequency content. For this reason it is important that the bandwidth of the filter includes the core dynamics and modes of the system.

3.2. Indirect Adaptive Controller Parameter Update A measure of error for the system model can be defined as:

$$e = F(s)Y(\theta,\dot{\theta},\ddot{\theta})\hat{a} - F(s)\tau \tag{3.34}$$

$$= W(\theta,\dot{\theta})\hat{a} - W(\theta,\dot{\theta})a \tag{3.35}$$

$$= W(\theta,\dot{\theta})\tilde{a} \tag{3.36}$$

When implemented, only (3.34) is used. Physically (3.34)-(3.36) define the prediction error between the torque applied to the system and the torque which would be applied, based on the position and velocity of the robot joints in conjunction with the current parameter estimates. From this definition of system error e, indirect adaptive controllers are derived.

The method of parameter update adopted in this research for indirect adaptive control is based on least squares minimization. Parameter update is governed by the minimization of [47]:

$$J = \int_0^t \|\tau(r) - W(r)\hat{a}(t)\|^2\,dr. \tag{3.37}$$

Expanding (3.37) and differentiating, the parameter update formula is derived as:

$$\dot{\hat{a}} = -P(t)W^T e, \tag{3.38}$$

where the time varying gain matrix $P(t)$, is updated based on:

$$\dot{P}(t) = -PW^TWP. \tag{3.39}$$

Slotine and Li show that parameter error $\tilde{a}$, converges to zero under persistently exciting conditions [27].

4. Composite Adaptive Control

The composite adaptive controller combines the parameter estimation abilities of both the direct and indirect adaptive controllers. The controller framework remains the same as the original direct adaptive controller introduced in Section 2, but the parameter update law is now modified to [46]:

$$\dot{\hat{a}} = -P(t)\left(C_1Y^T(\theta,\dot{\theta},\dot{\theta}_r,\ddot{\theta}_r)s + C_2W^T(\theta,\dot{\theta})e\right) \tag{3.40}$$

In (3.40) the parameter update of (3.26) and (3.38) are simply concatenated. Their contributions are weighted by positive definite matrices C_1 and C_2.

Success of the algorithm depends greatly on the update of the gain matrix $P(t)$. Intuitively, large magnitudes of P will enable faster convergence. However, there is a functional upper-bound imposed on P due to sampling and noise. This is because the parameters should not change faster than the bandwidth of the system. Also a large P will cause parameter update to be driven by system disturbances. When there is a lot of movement in the model parameters, P should be low enough to prevent parameter estimate oscillations. Conversely, when system activity is low, P should be increased to stimulate parameters out of the lull. To achieve these properties on-line, without using thresholding heuristics, Slotine and Li modify the basic least squares formulation.

In [27] and [46], Slotine and Li analyse the properties of the least squares update formula with respect to robustness, parameter convergence, and time varying parameters. In particular they found that:

- By minimising the error with respect to $\hat{a}(t)$ over time, spurious errors due to noise are averaged out. This is important for robustness.
- If the system is "persistently exciting" the parameter error $\tilde{a} \to 0$.
- The least squares update responds slowly to time varying parameters because of the memory inherent in the time dependent least squares formulation.

Slotine and Li also found that under persistently exciting conditions the gain matrix $P(t) \to 0$. For this reason, Slotine and Li developed a modified gain update algorithm which introduces a forgetting factor to the least squares formulation and intrinsically puts a bound on the gain matrix $P(t)$. The "bounded-gain-forgetting" (BGF) method is given as [46]:

$$\frac{d}{dt}P^{-1}(t) = -\lambda(t)P^{-1} + W^TW \tag{3.41}$$

where:

$$\lambda(t) = \lambda_0(1 - \frac{\| P \|}{k_0}) \tag{3.42}$$

Equations (3.41) and (3.42) are the result of the limitations found in previous update laws. The product W^TW is the solution to the least squares minimization problem of the actual and computed joint torques. This is also known as the covariance matrix in Kalman filter theory. In equation (3.41), the the term $-\lambda(t)P^{-1}$ is inserted to achieve what is known as a forgetting factor. This forgetting factor is required because of the infinite integral condition:

$$\lim_{t\to\infty}\int_0^t W^TWdt \longrightarrow \infty \tag{3.43}$$

This implies that without a forgetting factor, P will become very small and of little use. By using (3.42), the parameter update gain is gradually reset by the term λ_0P^{-1} and kept from becoming to large, i.e. unbounded, by $(\lambda_0/k_0)\frac{\|P\|}{k_0})P^{-1}$.

It is shown that under excitatory conditions the composite adaptive controller attains exponential trajectory convergence and global parameter convergence. This result is important because it can be used as a basis for calibration algorithms.

CHAPTER 4

Autonomous Joint Calibration Using Adaptive Control

The Composite Adaptive Controller of Slotine and Li accomplishes several important tasks within one package.

- By utilising the manipulator model, useful insights can be drawn from the control of the robot.
- The algorithm is not a computationally expensive, it is sympathetic to practical limitations (acceleration measurements and matrix inversion), and is robust.
- It achieves *global* parameter convergence. This makes the algorithm an estimator, as well as an adaptive controller.

For these reasons, the composite adaptive controller is attractive for use on real robot systems.

The global convergence property is a powerful result when one considers the non-linearity of robotic systems. However, in its present form, the controller requires that the relationship between joint sensor output and joint angle to be known a priori. It is apparent that the global convergence properties of the adaptive algorithm could be used to estimate not only the robot dynamic properties, but also calibrate the joint sensors.

1. Motivation for Autonomous Joint Sensor Calibration

Motivation for autonomous joint calibration can be looked at from several levels.

1.1. Humans and Animals As humans, we take for granted our ability to continuously adjust and re-calibrate our own joint sensors. The first stages of this calibration is exhibited in babies when they wave their arms around in a semi-controlled state. This has also been shown in adults who have been forced to re-learn movement after an accident. Although human actuators and sensors are very different to the ones used in robots, the principle of learning through environmental interaction, used in human and animal calibration, should be applicable in robotics. It seems feasible that a robot could also self calibrate both joint sensors and dynamic parameters through environmental interaction.

1.2. Relinquishing the Necessity for "Non-Standard" Equipment In Chapter 2, several calibration algorithms were introduced which required either non-standard equipment, such as laser tracking systems, or required constraints on the manipulator such as fixing the end-effector of the manipulator. For industrial applications in dangerous environments such as space or nuclear power plants, this is not a sufficient solution. Also, from the commercial perspective, the user should not have to clamp the robot or purchase extra equipment. What is more, in the event of sensor drift these methods do not have potential to automatically sense or re-calibrate on-line without causing inconvenience to the operator.

A robot system ideally should be self contained and not require external intervention. (Exhibited in humans and animals.) This implies that autonomous calibration techniques should take advantage of the intrinsic and predictable forces in nature which are available.

Therefore the goal of this research is to design a control system which, when powered up, will be able to calibrate both joint sensors and dynamic properties without human involvement.

1.3. Combining the Various System Gains and Biases Robotic systems are composed of many components, each one having some degree of inaccuracy. The adaptive controller of Slotine and Li exhibits many advantages in this respect because it effectively lumps these biases and gains together. Therefore during control, these errors are taken into account intrinsically.

Figure 4.1 shows a block diagram of a computer controlled robotic system. For each device in the system there is an error associated with it. For the most part the error is linear with respect to the desired function.[1] For example, the desired voltage output (V_{des}) and real output (V_{out}) of a digital to analog converter (DAC) can be related by.

$$V_{out} = \alpha_d V_{des} + \beta_d \tag{4.1}$$

[1] At the outer limits of the hardware's operational range saturation often occurs.

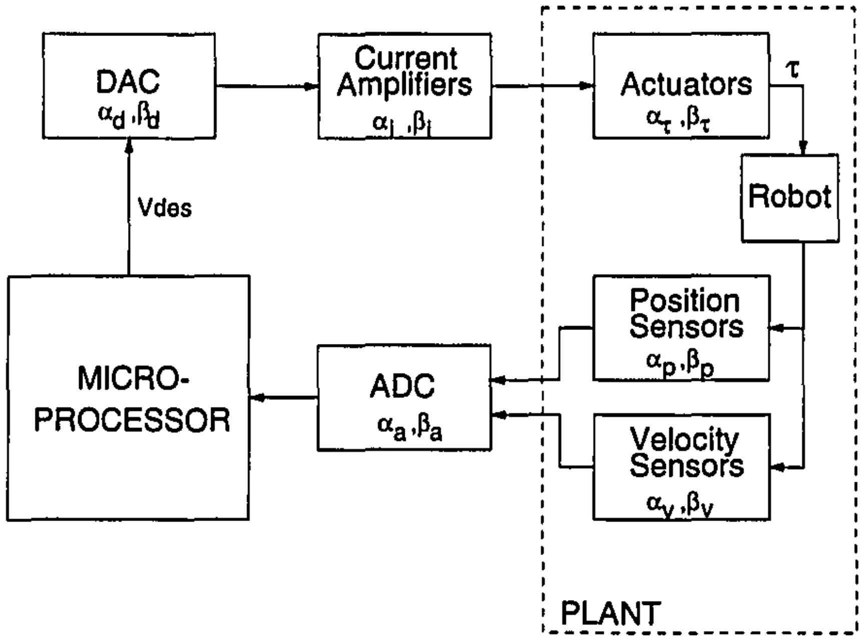


Figure 4.1: Errors Introduced by Hardware in a Robotic System

This assumes a linear relationship between V_{out} and V_{des} which is defined by the gain constant α_d and offset β_d. Using Figure 4.1, the effect of the errors from the microprocessor to the output in the plant become:

$$\tau = \alpha_\tau(\alpha_i(\alpha_d V_{des} + \beta_d) + \beta_i) + \beta_\tau \tag{4.2}$$

$$= \alpha_\tau \alpha_i \alpha_d V_{des} + \alpha_\tau \alpha_i \beta_d + \alpha_\tau \beta_i + \beta_\tau \tag{4.3}$$

$$\equiv \alpha_1 V_{des} + \beta_1 \tag{4.4}$$

Similarly, the path from the robot sensors to the microprocessor can be written as:

$$P_{meas} = \alpha_a \alpha_p P_{os} + \alpha_a \beta_p + \beta_a$$

$$\equiv \alpha_{pos} P_{os} + \beta_{pos} \tag{4.5}$$

and

$$V_{ELmeas} = \alpha_a \alpha_v V_{el} + \alpha_a \beta_v + \beta_a$$

$$\equiv \alpha_{vel} V_{el} + \beta_{vel} \tag{4.6}$$

The adaptive controller enables the various gains in the robotic system to be seen as one lumped and equivalent gain. (As in (4.4), (4.5), and (4.6).) Ideally the DAC, ADC, and motor gains (α's) will have a value of unity, and their offsets (β's) will be zero. The

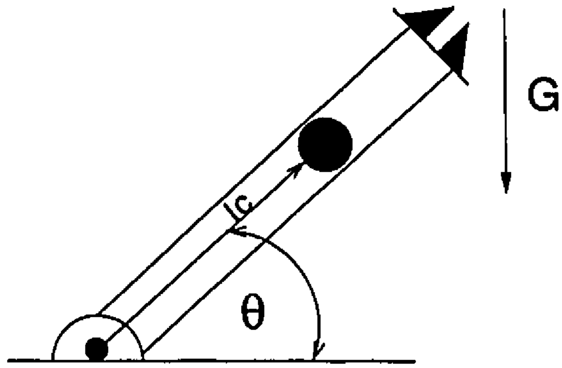


Figure 4.2: A Single Link manipulator

joint sensor gains and offsets however, can take a wide range of values. Combining all these offsets and being able to autonomously re-calibrate is useful because the robot parameters are found as a function of all the system components. Also, the averaging effect in adaptive control allows for the best fit over the entire system range.

2. Using Gravity as a Reference for Calibration

The adaptive control of Slotine and Li uses the robot dynamic model; this enables prediction of nonlinear robot behaviour. Intuitively, this predictable behaviour should also be utilised in joint calibration schemes in an attempt to achieve an autonomous algorithm.

When examining the various forces which are applied to the robot system, *gravity* stands out as being the most predictable and most reliable of these forces. Using gravity as a reference, it should be possible to find the relationship between joint sensor output and joint angle.

To simplify the derivations and issues involved in joint sensor calibration using adaptive control, a single link manipulator is used. The single link manipulator shown in Figure 4.2 has the dynamic model:

$$\tau = ml_c^2\ddot{\theta} + ml_c g\cos(\theta) \tag{4.7}$$

where m represents the point mass of the manipulator, l_c is the distance from the joint to the centre of mass, g is the acceleration constant due to gravity, θ is the joint position, and $\ddot{\theta}$ is joint acceleration. The dynamics in 4.7 can be re-written in the regressor form.

$$[\tau] = \begin{bmatrix} \ddot{\theta} & g\cos(\theta) \end{bmatrix} \begin{bmatrix} ml^2 \\ ml_c \end{bmatrix} \tag{4.8}$$

The model in (4.8) is the regressor model which would be used in the standard Slotine and Li composite adaptive control. Most importantly, (4.8) assumes the relationship between sensor output and joint angle is known.

Assuming a linear relationship between the joint angle θ, and the sensor output q gives,

$$\theta = \alpha q + \beta \tag{4.9}$$

$$\dot{\theta} = \alpha \dot{q} \tag{4.10}$$

$$\ddot{\theta} = \alpha \ddot{q} \tag{4.11}$$

where α represents the unknown joint gain (Degrees/Volt), and β is the unknown joint sensor offset (Degrees). Substituting (4.9)-(4.11) into (4.8) gives:

$$[\tau] = \begin{bmatrix} \ddot{q} & g\cos(\alpha q + \beta) \end{bmatrix} \begin{bmatrix} \alpha m l_c^2 \\ m l_c \end{bmatrix} \tag{4.12}$$

From (4.12) the unknown joint angle parameters of the system (α, β) cannot be extracted using the regressor form.[2] The fundamental problem in this formulation is the transcendental cosine function. The α and β terms cannot be written linearly with respect to the joint sensor values.

An intuitive solution to this problem is to expand the cosine term in (4.12) using well known trigonometric expansion formulas. Unfortunately this approach can only isolate the offset term β. For example,

$$\begin{aligned} \cos(\theta) &= \cos(\alpha q + \beta) \\ &= \cos(\alpha q)\cos(\beta) - \sin(\alpha q)\sin(\beta) \end{aligned} \tag{4.13}$$

Substituting (4.13) into the regressor matrix for the single link robot gives,

$$[\tau] = \begin{bmatrix} \ddot{q} & g\cos(\alpha q) & -g\sin(\alpha q) \end{bmatrix} \begin{bmatrix} \alpha m l_c^2 \\ m l_c \cos(\beta) \\ m l_c \sin(\beta) \end{bmatrix} \tag{4.14}$$

Using the formulation in (4.14) the sensor offset β can be found as:

$$\beta = \arctan\left(\frac{m l_c \sin(\beta)}{m l_c \cos(\beta)}\right) \tag{4.15}$$

However (4.14) requires a priori knowledge of the gain constant α. In the case of discrete incremental encoders, this simple expansion of the manipulator dynamics would

[2] The α term which appears in $\alpha m l_c^2$ in the a vector cannot be extracted because the estimation process will effectively lump all three parameters (m, l_c, and α) as one term.

be adequate (since the joint gain will be always be constant), however in systems which use analog sensors this is not possible. To perform a truly autonomous calibration, α must also be found using the adaptive control process. To accomplish this task, a new approach is required to overcome the problem imposed by the transcendental functions.

3. Introducing Polynomial Approximations

In Astley and Hayward, a method for overcoming the transcendental function problem in autonomous joint calibration is proposed [**2**]. The solution is centred around replacing the cosine functions with polynomials.

It is possible, with arbitrary accuracy, to approximate segments of nonlinear functions such as cosine and sine functions with polynomials. For example, (4.16) shows the expansion of $\cos(\theta)$.[3]

$$\cos(\theta) = \acute{a} + \acute{b}\theta + \acute{c}\theta^2 + (O)^3 \quad \theta_0 \leq \theta \leq \theta_1 \tag{4.16}$$

where $\acute{a}$, $\acute{b}$, and $\acute{c}$ are scalar constants and $(O)^3$ represents the error introduced by the polynomial. The second order approximation is valid over one mode of the cosine function; therefore the range of θ is restricted to region defined by θ_0 and θ_1.

By increasing the order of the polynomial in (4.16), the error between the real and the approximated function can be made arbitrarily small for a given range of input values.

Substituting (4.9) into (4.16) and using terms up to the second order only.

$$\cos(\theta) \approx \acute{a} + \acute{b}\,(\alpha q + \beta) + \acute{c}(\alpha q + \beta)^2 \tag{4.17}$$

Expanding and collecting yields.

$$\cos(\theta) \approx \left(\acute{a} + \acute{b}\beta + \acute{c}\beta^2\right) + \left(2\acute{c}\alpha\beta + \acute{b}\alpha\right) q + \left(\acute{c}\alpha^2\right) q^2 \tag{4.18}$$

From (4.18), it can be seen than the cosine argument can be written in terms of three constants, $\underline{a}$, $\underline{b}$, and $\underline{c}$; where:

$$\underline{a} = \acute{a} + \acute{b}\beta + \acute{c}\beta^2 \tag{4.19}$$

$$\underline{b} = 2\acute{c}\alpha\beta + \acute{b}\alpha \tag{4.20}$$

$$\underline{c} = \acute{c}\alpha^2 \tag{4.21}$$

Therefore (4.16) can be re-written as:

---

[3]This should not be confused with a Taylor approximation which approximates a function around a given point. For example, approximating $f(x+\delta x) = f(x) + \nabla f(x)\delta x + \frac{1}{2}\delta x \nabla^2 f(x)\delta x + (O)^3$. Where $x = [x_1\ x_2\ \ldots\ x_n]^T$.

$$\cos(\alpha q+\beta) = \cos(\theta)$$
$$= \underline{a}+\underline{b}q+\underline{c}q^2+O^3 \tag{4.22}$$

Substituting the cosine approximation in (4.22) to the regressor form of the single link robot dynamics gives,

$$[\tau]=\begin{bmatrix} \ddot{q} & g & gq & gq^2 \end{bmatrix}\begin{bmatrix} \alpha m l_c^2 \\ a \\ b \\ c \end{bmatrix} \tag{4.23}$$

where $a \equiv \underline{a}ml_c$, $b \equiv \underline{b}ml_c$, and $c \equiv \underline{c}ml_c$.

In (4.23), the relationship $\tau = Y\bar{a}$ is now in the correct form; all known parameters and variables are in the matrix Y, and the unknown parameters are in the parameter vector $\bar{a}$. The condition on parameter and trajectory convergence derived in Chapter 3 remains unchanged because the transcendental functions are simply being replaced by an equivalent expression. This is valid as long as two conditions are met.

- The order of the polynomial is of sufficient degree to represent the transcendental function it is replacing.
- The range of operation is not large enough to invoke the periodic properties intrinsic to trigonometric functions.[4]

The restriction on range of operation imposed by using the non-periodic polynomial function is imposed only during calibration. Once the joint sensor parameters are found, the regular manipulator dynamic model can be used.

To find the constants α and β from the polynomial coefficients in (4.23), some simple post processing is needed. After estimating the parameters using composite adaptive control for the single link case, and using the manipulator model given in (4.12), the parameters estimates should approximate:

$$gml_c\cos(\alpha q+\beta) = gml_c\cos(\theta) \tag{4.24}$$
$$\approx g\left(a+bq+cq^2\right) \tag{4.25}$$

The constants α and β can be found by equating the right side of (4.24) with (4.25) such that:

[4]Polynomial approximations can approximate a nonlinear function locally to an arbitrary accuracy, but they cannot replicate its periodicity

(4.26) $$ml_c \cos(\theta) = a + bq + cq^2$$

It is necessary to normalise the cosine term by finding the value of ml_c in (4.26). Differentiating both sides of (4.26) and equating to zero:

(4.27) $$-ml_c \sin(\theta) = \bar{b} + 2\bar{c}q = 0.$$

Assigning q^* to the solution of (4.27) enables ml_c to be found by:

(4.28) $$ml_c \cos(\theta^*) = ml_c = \bar{a} + \bar{b}q^* + \bar{c}q^{*2}$$

Therefore by finding the maximum point on the cosine approximation, the value for ml_c can be found. This allows the fitting of $\cos(\alpha q + \beta)$ to be independent of the mass properties of the manipulator according to:

(4.29) $$\cos(\theta) = \cos(\alpha + \beta) = \frac{\bar{a} + \bar{b}q + \bar{c}q^2}{ml_c}$$

Using (4.29), the value of $\cos(\theta)$ can be found for any value of sensor output q. Rearranging (4.29) to isolate the α and β constants.

(4.30) $$\theta = \alpha q + \beta = \arccos\left(\frac{\bar{a} + \bar{b}q + \bar{c}q^2}{ml_c}\right).$$

To solve for α and β in (4.30) a curve fitting strategy is used; the experimental work presented in Chapter 5 uses a least squares technique. The least squares approach adopts a linear system of equations in the form:

(4.31) $$\begin{bmatrix} q_1 & 1 \\ q_2 & 1 \\ . & . \\ . & . \\ q_n & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ . \\ . \\ \theta_n \end{bmatrix}$$

where θ_i is calculated from (4.30). The number of rows in (4.31) is arbitrary, however the range of q_i is important. The maximum and minimum values of q_i used in (4.31) should not exceed the maximum and minimum values used during the adaptive control estimation stage.

Defining the system in (4.31) as $Ax = B$, the parameters α and β can be found using the least squares pseudo inverse formula.

(4.32) $$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \equiv x = (A^T A)^{-1} A^T B$$

By extracting the joint sensor calibration values α and β, plus the dynamic parameters ml_c and αml_c^2, a full calibration of the single link robot has been achieved. It now remains to extend this result to the multiple link case.

4. Extending the Algorithm for the Multiple Link Case

Unfortunately, extending the algorithm outlined in the previous section is not wholly straight forward. The intuitive extension to the algorithm presented in Section 3 would be to replace all trigonometric functions with polynomial equivalents. However, this solution is not practical which is evident when analysing the dynamics of a two degree of freedom robot.

4.1. Linearly Dependent Columns Within the proof of trajectory and parameter convergence for adaptive control [46], it is not mentioned that parameter convergence also requires full rank with respect to the columns of the regressor matrix. For example, (4.33) shows the approximation of two trigonometric terms.

$$\begin{bmatrix} gc_1 & gs_1 & \dots \\ 0 & 0 & \dots \end{bmatrix} \Longrightarrow \begin{bmatrix} g & gq & gq^2 & g & gq & gq^2 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots \end{bmatrix} \tag{4.33}$$

The new regressor matrix in (4.33) is rank deficient; therefore there exist an infinite number of parameter combinations which will fit the system. For estimation purposes this breaks down the fundamental parameter convergence premise which the calibration scheme requires.

4.2. Frequency Doubling When considering multiple links, components in the regressor matrix often require the cosine or sine of the difference or sum of two angles. This has two adverse effects.

The degree of the polynomial required to model the expression must be quite high. For example, using a polynomial of degree two to represent the function $\cos(\theta_1)$, means that a polynomial of degree 4 is required to model $\cos(\theta_1 + \theta_2)$, i.e.

$$\begin{aligned} \cos(\theta_1) &\approx a + bq_1 + cq_1^2 \\ \cos(\theta_1 + \theta_2) &\approx a + bq_1 + cq_1^2 + dq_2 + eq_2^2 + fq_1q_2 + gq_1^2q_2 + hq_1q_2^2 + iq_1^2q_2^2 \end{aligned} \tag{4.34}$$

However, the major cost of (4.34) is the that the term $\cos(\theta_1 + \theta_2)$, which in the regular dynamic model required just one unknown parameter, now requires 9 parameters. This increases the burden of computation by approximately 9^2 times; with respect to impiementation, this is not practical.

4.3. The Malleability of Polynomials When considering larger systems, which utilise several polynomial approximations within the regressor matrix, an interesting situation arises. Unlike the trigonometric functions which have a fixed response to an input, the polynomial can shape itself to fit many types of input output relationships. Polynomials have the ability to shape themselves from a constant straight line to extreme curvature. This means that without the structure of the cosine and sine terms, the polynomial coefficients could converge to fit the model but in an unpredictable way. This does not mean that the overall dynamic response will be bad, in fact the contrary could be true. The polynomial based unknown parameters may be able to fit themselves to the unknown and unpredictable dynamics of the system; however this is detrimental with respect to parameter estimation. In essence, the regressor matrix will loose all structure, normally imposed by predictable, periodic functions. This would lead to a system having properties closer to a neural network than to a calibration algorithm.

5. A Multiple Link Algorithm

From the discussion in the preceding section, it is clear that it is not possible to simply extend the polynomial approximation idea throughout the multiple link manipulator's dynamics. Instead, a more conservative approach is required with respect to the use of polynomial approximations. It is therefore proposed to split the estimation of gain constants (α) from estimation of offsets (β).

To show the application of the multiple link calibration algorithm, a two degree of freedom planar manipulator is used. Also, to illustrate the algorithm in three dimensional Cartesian space, the robot is is positioned at an angle as shown in Figure 4.3.

In Figure 4.3 the angle γ is fixed, and the joint angles are represented by θ_1 and θ_2. The dynamic model can be derived and put in the regressor form:

$$\begin{bmatrix} gc_1 & gc_{12} & \ddot{\theta}_1 & \ddot{\theta}_1+\ddot{\theta}_2 & \begin{matrix} c_2\ddot{\theta}_1 + c_2\ddot{\theta}_2 - s_2\dot{\theta}_2\dot{\theta}_1 \\ -s_2\dot{\theta}_2^{\,2} - s_2\dot{\theta}_1\dot{\theta}_2 \end{matrix} \\ 0 & gc_{12} & 0 & \ddot{\theta}_1+\ddot{\theta}_2 & c_2\ddot{\theta}_1 + s_2\dot{\theta}_1^{\,2} \end{bmatrix} \begin{bmatrix} (m_1 l_{c1} + m_2 l_1)\sin(\gamma) \\ m_1 l_{c2}\sin(\gamma) \\ m_1 l_{c1} + m_2 l_1^2 \\ m_2 l_{c2}^2 \\ m_2 l_1 l_{c2} \end{bmatrix} \tag{4.35}$$

5.1. Finding the α Gain Constants The use of the polynomial approximation was shown in its application to a single link robot in Section 3. Using the composite adaptive control, *global* parameter convergence enables the gain constant and offset to be found. Since this cannot be expanded to a multiple link case, another approach is needed.

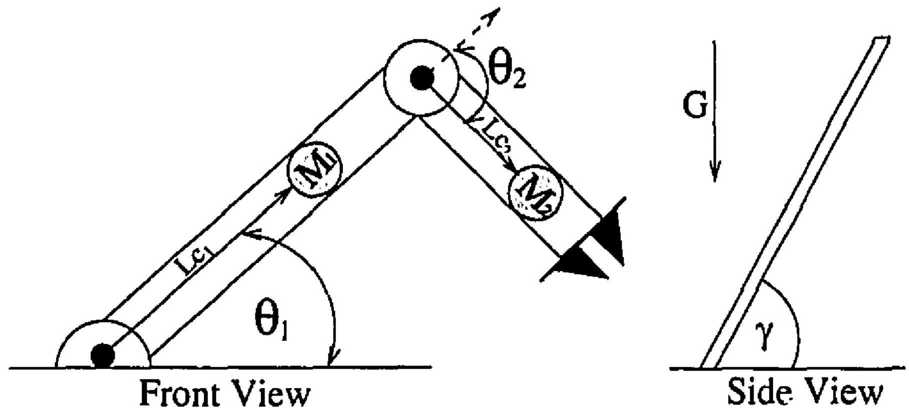


Figure 4.3: Two Degree of Freedom Robot in Three Dimensional Space

A manipulator with n joints, can be constrained to a single degree of freedom by holding $n-1$ joints in a static position. This can be achieved by position servoing the joints at a reference position using PD or PID feedback. A single link model can then be applied to the constrained manipulator and the polynomial approximation technique can be applied as in Section 3.

For example, constraining the second joint of the robot in Figure 4.3 to an arbitrary reference position, the dynamics for the remaining joint become:

$$[\tau_1] = \begin{bmatrix} \ddot{\theta}_1 & g\cos(\theta_1) \end{bmatrix} \begin{bmatrix} m_x l_{cx}^2 \\ m_x l_{cx} \sin(\gamma) \end{bmatrix} \tag{4.36}$$

where m_x and l_{cx} represent the combination of m_1, m_2, l_{c1}, and l_{c2}. This can then be transformed into the polynomial form based on sensor outputs q.

$$[\tau_1] = \begin{bmatrix} \ddot{q} & g & gq_1 & gq_1^2 \end{bmatrix} \begin{bmatrix} \alpha m_x l_{cx}^2 \\ a \\ b \\ c \end{bmatrix} \tag{4.37}$$

From (4.36) and (4.36) it is interesting to notice the effect of the angle γ. The formulation is not affected by the robot not acting directly against gravity because $\sin(\gamma)$ appears only as a constant scaling factor. Evidently, if $\gamma = 0$ then the robot is perpendicular to the force of gravity and the calibration is no longer possible. From a practical perspective, it is favourable to have the contribution due to gravity as large as possible such that it will

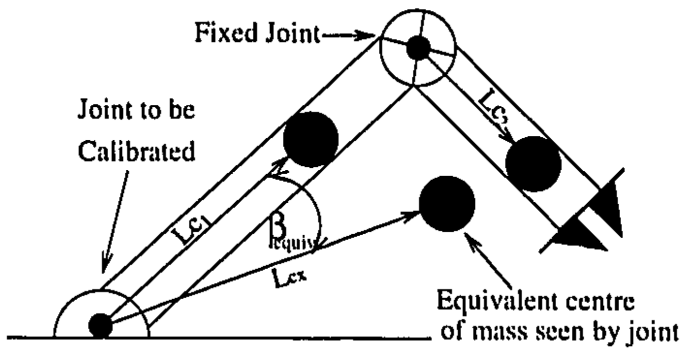


Figure 4.4: Two Degree of Freedom Robot Constrained to One Degree of Freedom

dominate over all "unmodeled" forces. From (4.36), as γ approaches 90^o the proportion of available mass to be considered in the estimation will increase.

Using the polynomial equivalent in (4.37) gives only joint sensor gain information; although joint offset information can be found this value is not useful, Figure 4.4 shows why.

The joint angle offset β is no longer a predictable quantity. The centre of mass, seen by joint 1 in Figure 4.4 is a combination of the position of the centre of mass of link 1 (M_1) and the centre of mass of link 2 (M_2). Since the joints which are being held are in an arbitrary state, the relationship between the desired joint offset and the centre of mass seen by the joint can never be known.

This is not the case for the joint gain α. Although the location of the centre of mass seen by the joint under consideration is arbitrary, the joint gain maintains its integrity. This is because an angular displacement in joint 1, will cause the same angular displacement in the centre of mass of joint 1, joint 2, and also the equivalent centre of mass offset at β_{equiv}.

Using this formulation it is possible to use the composite adaptive control algorithm using the polynomial approximations to find the joint sensor gain constants. This is done by using a single link model in the adaptive control structure on each joint individually. The joints which are not under consideration are held in an arbitrary position, thus making the single link model valid for the multiple link robot.

5.2. Finding the β Joint Angle Offsets For the multiple link robot it is not possible to find the joint offsets separately because link masses will be effectively lumped together when performing the adaptive control. Instead, all the joint offsets are found in one step utilising the method in Section 2. This requires knowledge of the joint gains found using the method presented in the preceding section.

It was shown that with knowledge of joint gains, the joint offset can be found using identities of the type:

$$\begin{aligned} \cos(\theta) &= \cos(\alpha q + \beta) \\ &= \cos(\alpha q)\cos(\beta) - \sin(\alpha q)\sin(\beta) \end{aligned} \tag{4.38}$$

In (4.38) the β constants occur linearly with respect to the known quantities α and q. This can then be written in the regressor form using the full dynamic model. The regressor matrix for the two degree of freedom robot becomes:

$$\begin{bmatrix} gc_{\alpha_1 q_1} & -gs_{\alpha_1 q_1} & g\cos(\alpha_1 q_1 + \alpha_2 q_2) & -g\sin(\alpha_1 q_1 + \alpha_2 q_2) & \alpha_1 \ddot{q}_1 & \alpha_1 \ddot{q}_1 + \alpha_2 \ddot{q}_2 \\ 0 & 0 & g\cos(\alpha_1 q_1 + \alpha_2 q_2) & g\sin(\alpha_1 q_1 + \alpha_2 q_2) & 0 & \alpha_1 \ddot{q}_1 + \alpha_2 \ddot{q}_2 \end{bmatrix.$$

$$\left.\begin{matrix} \cos(\alpha_2 q_2)(\alpha_1 \ddot{q}_1 + \alpha_2 \ddot{q}_2) & -\sin(\alpha_2 q_2)(\alpha_1 \ddot{q}_1 + \alpha_2 \ddot{q}_2) \\ -s_{\alpha_2 q_2}(\alpha_1 \alpha_2 \dot{q}_2 \dot{q}_1 + \alpha_1^2 \dot{q}_1^2 + \alpha_1 \alpha_2 \dot{q}_1 \dot{q}_2) & -c_{\alpha_2 q_2}(\alpha_1 \alpha_2 \dot{q}_2 \dot{q}_1 + \alpha_1^2 \dot{q}_1^2 + \alpha_1 \alpha_2 \dot{q}_1 \dot{q}_2) \\ \cos(\alpha_2 q_2)\alpha_1 \ddot{q}_1 + \sin(\alpha_2 q_2)\alpha_1^2 \dot{q}_1^2 & -\sin(\alpha_2 q_2)\alpha_1 \ddot{q}_1 + \cos(\alpha_2 q_2)\alpha_1^2 \dot{q}_1^2 \end{matrix}\right]$$

$$\begin{bmatrix} (m_1 l_{c1} + m_2 l_1)\sin(\gamma)\cos(\beta_1) \\ (m_1 l_{c1} + m_2 l_1)\sin(\gamma)\sin(\beta_1) \\ m_1 l_{c2}\sin(\gamma)\cos(\beta_1 + \beta_2) \\ m_1 l_{c2}\sin(\gamma)\sin(\beta_1 + \beta_2) \\ m_1 l_{c1} + m_2 l_1^2 \\ m_2 l_{c2}^2 \\ m_2 l_1 l_{c2}\cos(\beta_2) \\ m_2 l_1 l_{c2}\sin(\beta_2) \end{bmatrix} \tag{4.39}$$

Using the expansion in (4.39) the number of unknown parameters has increased by just three from the original regressor matrix. After running the adaptive control, the offset values can be found as:

$$\beta_1 = \arctan\left(\frac{(m_1 l_{c1} + m_2 l_1)\sin(\gamma)\sin(\beta_1)}{(m_1 l_{c1} + m_2 l_1)\sin(\gamma)\cos(\beta_1)}\right) \tag{4.40}$$

$$\beta_1 + \beta_2 = \arctan\left(\frac{m_1 l_{c2}\sin(\gamma)\sin(\beta_1 + \beta_2)}{m_1 l_{c2}\sin(\gamma)\cos(\beta_1 + \beta_2)}\right) \tag{4.41}$$

$$\beta_2 = \arctan\left(\frac{m_2 l_1 l_{c2}\sin(\beta_2)}{m_2 l_1 l_{c2}\cos(\beta_2)}\right) \tag{4.42}$$

It is also theoretically possible to obtain joint sensor information using velocity and acceleration based parameters as in (4.42). (Which was not possible when using a single link.) However, in practice, these terms are less reliable with respect to both modelling and prediction, hence establishing joint parameter information in this way has not been pursued.

Accompanying the sensor offset values are the dynamic parameters of the robot. These can then be reused, along with the joint calibration information for robot control.

6. An Analysis of the Joint Calibration Method

In the preceding sections a new algorithm for joint sensor calibration has been presented. It is principally been motivated by the practical constraints imposed by other joint calibration algorithms; however from a theoretical perspective there are several important theoretical issues which can be addressed.

6.1. A Brief Summary of the Joint Calibration Algorithm The algorithm can be broken into two major components, estimation of the joint sensor gains (α), and estimation of the joint offset parameters (β). In sequence the algorithm for an n link robot is:

(i) Hold, in a servo loop, all joints except the one under consideration. The position at which each joint is held is arbitrary.
(ii) Using the polynomial approximation for gravity terms, operate the joint using a single link equivalent model under the composite adaptive control format.
(iii) From the parameter data found in the adaptive control, a minimization strategy is used to obtain the joint gain value (α).
(iv) Repeat Steps (i)–(iii) for all the joints.
(v) Using the joint gain information and using trigonometric expansion formulas, use the full dynamic model in the composite adaptive control to obtain information on joint offsets.
(vi) Process the estimations found in step (v) to find joint offsets (β).

6.2. Disadvantages of the Joint Calibration Algorithm From a theoretical standpoint there are several components of the calibration structure which could be detrimental to the algorithm's performance.

6.2.1. *Propagation of Error* The algorithm is composed of two main stages, joint sensor gain estimation and joint offset estimation. The joint offset information requires the joint sensor gain information, and implicitly assumes its integrity. In the event of joint sensor gain error (some of which is inevitable), this error will be propagated and used by the second stage. Therefore the second stage will be impeded from the start because it does not receive the correct input data; this in turn will affect the accuracy of the joint sensor offset β.

The extent to which the propagation of error affects the overall estimation is dependent on the magnitude of error being input into the second stage. If the error is not too large (i.e. does not harshly violate the dynamic model) then the joint offset will try to fit itself to the model in the best possible fashion.[5] This implies that the best possible fit will be made using the incorrect joint gain values, this amounts to an intrinsic error compensation within the second stage. Unfortunately this compensation will affect the joint offset estimation.

6.2.2. *Integrity of Dynamic Model* There is an implicit assumption, within both the original Slotine and Li adaptive control format and the joint calibration scheme presented in this chapter, that the dynamic model of the manipulator can be found and is accurate.

Derivation of the dynamic model for serial manipulators is reasonably straight forward. However for parallel manipulators the dynamic model can be difficult to derive. This makes application of adaptive control methods more involved because of the difficulty in obtaining the dynamic model. What is more, the complexity of the dynamics increases exponentially with the increase in the number of joints. This is due to the coupling between joints. Beyond three joints, the complexity and hence the computational requirements on the system become quite stressed. This situation can be somewhat relieved by eliminating components of the dynamics which have relatively small contributions.

More troubling are the forces within the manipulator dynamics which cannot be modelled. Adaptive control techniques can find friction constants within the system using basic friction models; however, complex and nonlinear friction forces, such as stiction, cause voids within manipulator dynamic models. Hysteresis, backlash, and flexibility present similar modelling problems. The influence of these nonlinearities on estimation and control depends on their relative magnitudes within the system. These forces, which cannot be modelled, should be expected to be the principle source of error within the joint calibration.

6.2.3. *Chameleon-like Properties of the Polynomial* When substituting a trigonometric term with a polynomial, two potential detrimental aspects are added to the system.

[5]The definition of large in this context is not intuitively clear because the system is nonlinear.

It has already been mentioned that the polynomial does not have a periodic property. This means that the range of movement of a joint must be restricted such that a second order polynomial, for example, does not approximate more than one peak in the curve it is attempting to estimate.

A more serious consequence of using a polynomial is that it does not have a well defined structure. A second order polynomial can be steeply curved or flat. The ability to change its form so dramatically means that it could also pick up, and try to approximate, unmodeled dynamics of the system. This is undesirable for the purpose of calibration.

6.3. Advantages of the Joint Calibration Algorithm The advantages of the algorithm, which are apparent from a theoretical standpoint, range from adaptive control issues to processor requirements.

6.3.1. *Global Convergence* The most important property of the composite adaptive control algorithm is exponential global parameter convergence. Without this property joint calibration would not be possible.

The global convergence property is the distinguishing characteristic of the algorithm when compared to other automatic joint calibration schemes. Joint calibration schemes, which use a static data retrieval followed by post processing data fitting techniques, have consistently encountered problems in converging to the joint gain parameter, for example [31]. Such methods, which often use least squares minimization techniques, require observability constraints on data and initial estimates to prevent the algorithm converging to the trivial solution. The adaptive control based calibration scheme on the other hand, *does not* require constraints on the inpu[illegible] and will converge to the system parameters *without* any a priori estimate data.

A limitation of static methods is that the least squares fitting strategies do not guarantee global convergence, particularly for nonlinear systems. Dynamic on-line parameter estimation intrinsically averages out noise, erroneous signals, and fluctuations. In general, off-line methods do not filter out these outliers causing deviations in the fit.

Compared to static methods, dynamic on-line methods enable many more data to be considered within the estimation. For example, a 30 second calibration, running at 1kHz will consider 30,000 data points in each of position, velocity, acceleration, and force. Therefore over 100,000 input data points can be considered in the 30 seconds. Static methods in general use in the order of 500 data points.

6.3.2. *Arbitrary Calibration Trajectory* There has been a lot of research in excitatory trajectories in adaptive control and the affect of slow trajectories on stability and convergence. Excitation problems have been one of the major detriments to the more widespread use of adaptive control.

In the calibration framework this is no longer a problem. Calibration is considered a separate procedure from regular robot use, and therefore the optimal trajectory for the system can be chosen for the purposes of calibration.

6.3.3. *Computational Processing Requirements* Although the adaptive control is non-linear, the processing requirements are not excessive, enabling real time implementation on mainstream computers. Matrix multiplication and addition take up the majority of the processing cycle. Inversion of matrices is also not required. Also, unlike static methods, the adaptive controller does not need to store data for all time. This enables substantial savings in microprocessor memory requirements.

Another important component, with respect to processing requirements, is that that the algorithm does not rely on any iterative procedures. This has three important implications.

- The processing time is constant for each sampling period.
- There is never a danger that the algorithm will not converge in any given cycle.
- Integrity checks on matrices and data are not required before data processing. Since no integrity checks are necessary, the computer never has to "bail out" and stop processing.

These factors make the on-line adaptive control calibration algorithm suitable for real time implementation.

6.3.4. *Sufficient Accuracy with Low Order Polynomials* Ideally a transcendental function should be replaced by an infinite series of polynomials. However, using a low order polynomial gives very good accuracy.

Figure 4.5 shows the curve fitting ability of the function $\cos(0.8q + 0.3)$. For just a 2nd order degree polynomial the error is in the order of 2% of the total cosine magnitude and 1% for the cubic polynomial. Therefore it is possible to achieve good accuracy using a low order polynomial approximation. This means fewer unknown variables are required for joint sensor gain determination, making the system less complex and convergence quicker.

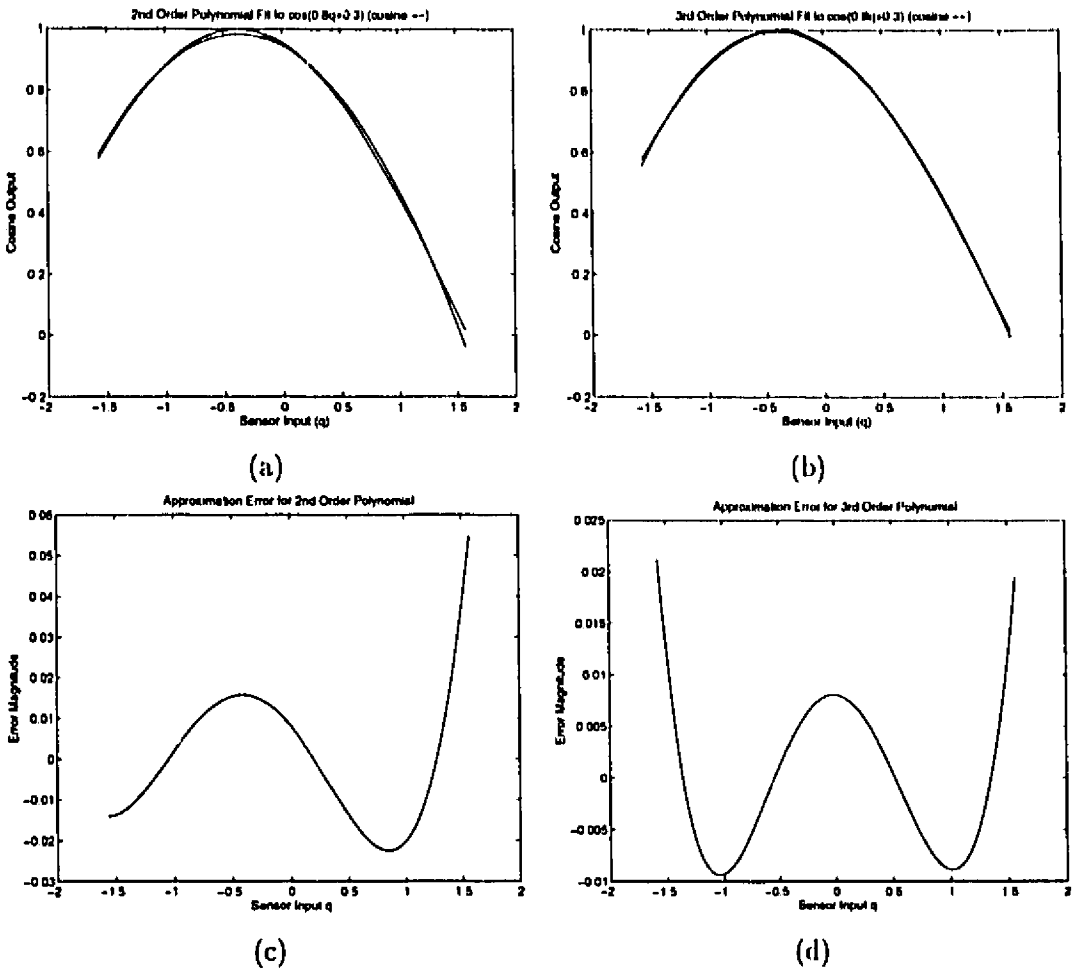


Figure 4.5: Trigonometric Fitting Ability of Low Order Polynomials. (a) 2nd Order Polynomial Approximation of a $\cos(0.8q + 0.3)$ (b) 3rd Order Polynomial Approximation of a $\cos(0.8q+0.3)$ (c) 2nd Order Approximation Error of $\cos(0.8q+0.3)$ (d) 3rd Order Approximation Error of $\cos(0.8q + 0.3)$

6.3.5. ***Estimation of Mass Properties*** A bi-product of joint sensor calibration using adaptive control is that mass properties are also estimated. The mass parameters in the inertia, velocity, and gravity matrices become available for other control tasks. For example, after joint sensor calibration the robot can be gravity compensated or used in further nonlinear control applications.

6.3.6. ***Autonomy of Calibration*** The algorithm is designed such that *no* human intervention is necessary. In essence, the operator has only to power up the robot and computer for calibration of dynamic and joint sensor properties. The ease of use enables easy re-calibration; thus preventing sensor drift affecting accuracy and control.

7. Summary

The algorithm presented in this chapter enables autonomous joint sensor calibration of analog position sensors. As a bi-product of the joint sensor calibration, mass properties are also estimated, which can be used in other control applications.

The algorithm enjoys the property of global convergence relinquishing the need for initial estimates. Also, because the algorithm performs estimation on-line, a large number of data can be considered preventing erroneous signals from adversely influencing estimation.

CHAPTER 5

Experimental Results

In this chapter, the theory presented in the preceding chapters is applied on a real robot system. Originally the theory was tested in simulation with excellent results. The real test however, is the performance of the algorithm under non-ideal conditions.

The hardware used for the experiments is described in Section 1. The problems encountered during the transfer from simulation to practice is described in Section 2. The results of the experiments and a discussion of results follows in the subsequent sections.

1. Hardware

The principal component of the hardware is a 6 degree of freedom force reflecting *haptic* hand controller, shown in Figure 5.1. The design was conceived by Dr. Vincent Hayward of McGill University [15].

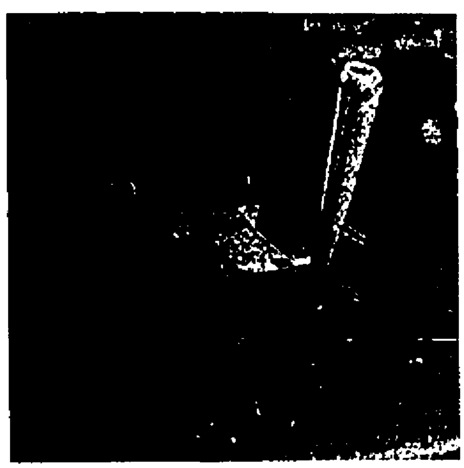


Figure 5.1: The 6 DOF Haptic Hand-Controller

The hand controller is comprised of seven actuator and sensing channels. Three channels are used for translational motion, and four are used for rotational motion. The use of four actuators to achieve the three degrees of rotational freedom (roll, pitch, and yaw) is known as actuator and sensor redundancy and is discussed in [14]. In essence, the extra degree of actuation and sensing enables balanced performance over the entire workspace and the potential for self calibration.

The hand controller is designed to emulate virtual environments in applications such as teleoperation. The actuators enable forces to be transmitted through tendons to the user. The remote placement of the actuators makes the output device both ergonomic and light, critical factors in hand controller design [16].

Figure 5.2 shows the functional representation of the translational stage. Although this has three degrees of freedom, just two are used for the purpose of these experiments.

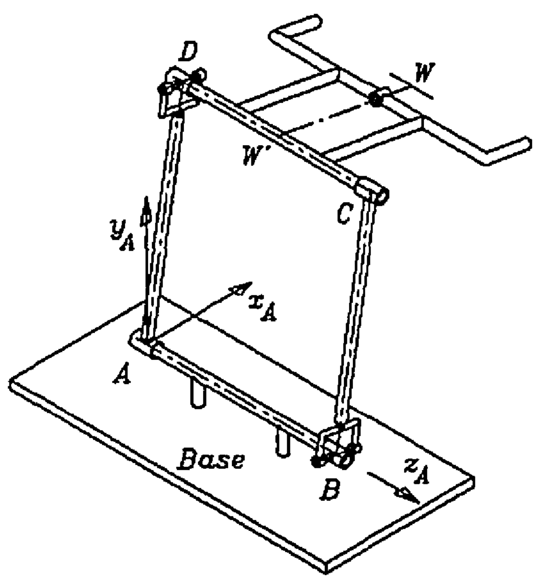


Figure 5.2: Schematic of the Translational Joints of the 6 DOF Haptic Hand-Controller

The degrees of freedom centred at A and W' are used in experiments. The third degree of freedom at B is not used because it introduces a coupling with joint A; the coupling effect is hard to model and results in mechanical instability.[1] This extra degree of freedom is eliminated by mechanically constraining the joint at B; therefore the robot is equivalent

[1] The hand-controller is a version one prototype and is currently being redesigned to eliminate such problems.

to a two degree of freedom planar manipulator working against gravity. The rotational part of the device is not considered because gravity does not directly act upon it.

The force and position sensor outputs are sent to signal conditioning circuitry. Each signal conditioning circuit is part of an analog control board which can implement force feedback, position feedback, reference position setting, or bypass all control to a computer. In all the experiments presented in this chapter, the control is bypassed to the computer.

All processing is executed on a single general purpose Intel 486DX personal computer running at a clock rate of 66MHz. Under the DOS operating system, real time implementation is achieved by running off interrupts triggered by the system clock. Unfortunately multiple processes are not supported under DOS; therefore all calculations, including update of the system dynamics, are executed in the same cycle and at the same rate. The hand-controller is interfaced to the microprocessor through Green Spring 16 bit Analog to Digital Converters (DAC) and 10 bit Digital to Analog Converters (DAC). The torque output to the motors is sent from the DAC and converted to a current by a general purpose Voltage-Current converter.

The position sensor used for each channel has been custom made. Each position sensor consists of two Light Emitting Diodes (LED) and two light receivers, shown in Figure 5.3.

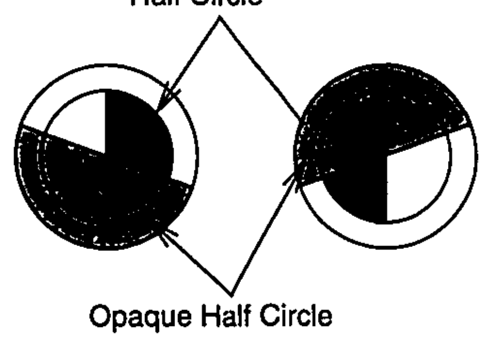


Figure 5.3: The Position Sensor Mechanism

The tendons are wrapped around two pulleys which each have opaque half circles in the middle. Each pulley is mounted on small cylinders which also have opaque half circles. The LED is mounted behind the fixed cylinder, the light is transmitted through the transparent side of the cylinder and pulley. As the pulley is rotated by the rotation of the joint, the proportion of light received by the sensor changes. The amount of light received is proportional to the angular displacement. The position output is the difference between the outputs of each sensor such that:

$$\begin{aligned} \theta &= (\alpha_{P1}q_{P1} + \beta_{P1}) - (\alpha_{P2}q_{P2} + \beta_{P2}) \\ &= (\alpha_{P1}q_{P1} - \alpha_{P2}q_{P2}) + \beta_{P1} - \beta_{P2} \\ &\equiv \bar{\alpha}\bar{q} + \bar{\beta} \end{aligned} \tag{5.1}$$

where q_{P1} and q_{P2} are the outputs of each light receiving transistor, α_{P1}, α_{P2}, are the joint gains, and β_{P1}, β_{P2} are the joint offsets for each sensor unit.

The opaque half circles are mounted to ensure that the transparent area, in which light can be transmitted to the sensors, totals 180^o; i.e. when one sensor has a 30^o transparent arc, the other will have an arc opening of 150^o ($180^o - 30^o$). This facilitates a differential output for the complete position sensor unit expressed in terms of $\bar{\alpha}$, $\bar{q}$, and $\bar{\beta}$. The differential output is used to reduce noise (in particular thermal noise), and to compensate for errors caused by non ideal factors in the tendons such as elasticity. Velocity readings are generated by an Operational Amplifier in a differentiating configuration.

2. Problems Encountered in Implementation

In simulation, a given problem can be tested under optimal conditions. Although non ideal behaviour can be added to the simulation, this, in general, can never fully represent a real system. For this reason, it is often necessary to implement an algorithm on a real system to ensure that the theory is still applicable in the presence of unpredictable and unmodeled behaviour.

The simulation results showed that the theory derived in the preceding chapters was valid. Nevertheless, during the transfer from the simulation environment to the hand-controller system, several factors, which were not apparent from the simulation results, surfaced. These problems had to be addressed for successful algorithm implementation.

2.1. Friction The simulations did not take into consideration friction as an unmodeled force. The original thought was that the hand controller, which was designed to be a low friction device, would not have appreciable friction compared to the forces due to gravity and inertia. However, it was quickly realized that this initial assumption was incorrect, making accurate calibration impossible.

The intuitive answer to the friction problem is to model friction within the adaptive control. Although friction models can be included in the regressor matrix of the robot model, this is not the case for the polynomial equivalent model. It was found that adding a Coulombic friction model to the adaptive control interfered with the convergence of the polynomial. This is due to the chameleon properties of the polynomial which the trigonometric function, which it replaced, does not exhibit.

To eliminate this detrimental property, the friction forces must be dealt with outside of the adaptive controller. This is achieved by using a feed-forward friction signal to the actuators; it is vital however, that this is done autonomously.

To establish the force contribution due to friction, the hand-controller is run in a closed loop trajectory using only a PD feedback for control. The speed of the trajectory is kept as low as possible to reduce dynamic effects, but high enough to ensure that stick-slip (stiction) effects are negligible. The result of the closed loop, PD controlled trajectory, is shown Figure 5.4.

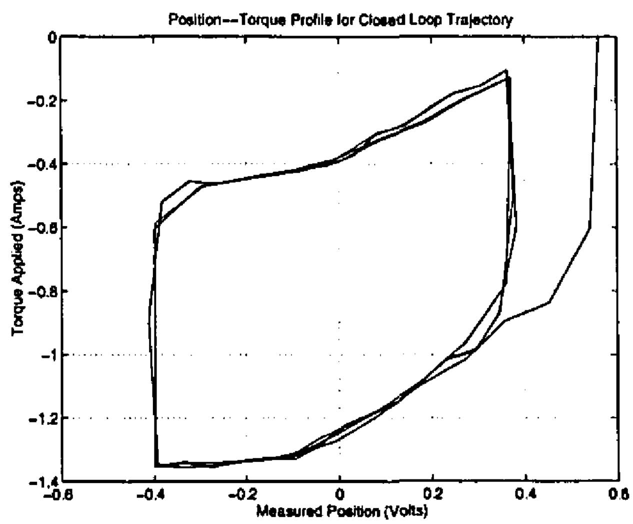


Figure 5.4: Position-Torque Profile of a Hand-Controller Joint

From the position-torque response it is clear that the dominant form of friction is Coulombic in nature. By calculating the average torque difference in the forward and negative directions, an estimate of the Coulombic friction can be made:

(5.2) $$\text{Coloumbic Friction} \approx 0.45\,\text{SIGN}\{V_D\}$$

where V_D is the desired velocity. The Coulombic friction term can be added, in a feed-forward manner, to the motor torque calculated by adaptive control routines. The modified block diagram of the adaptive control system is shown in Figure 5.5.

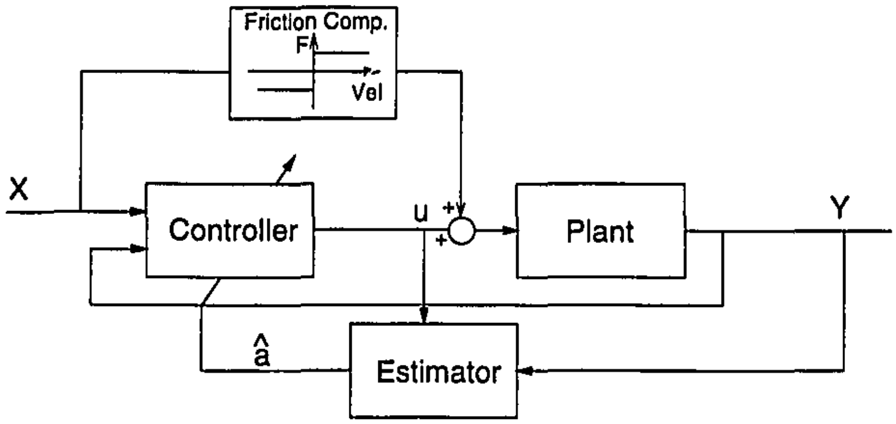


Figure 5.5: Adaptive Control Framework with Friction Compensation

The coefficient of Coulombic friction must be found individually for each joint. This is simply done by moving the joint under test in a closed loop trajectory while the other joints are locked in arbitrary positions. This can be achieved through computer control without human intervention or knowledge of the joint sensor–joint angle relationship. The Coulombic friction model obviously does not model stick–slip friction (stiction) or viscous friction. However these frictional forces do not have such a large relative impact over the overall system dynamics. What is more, friction forces, such as stiction, are hard to model accurately making their application in a real system questionable.

2.2. Calibration Trajectory The adaptive control is used as part of a calibration step; this affords the luxury of being able to choose the best possible trajectory to ensure accurate parameter convergence.

It was stated in Chapter 2 that one of the major hindrances preventing the more widespread use of adaptive control is the need for excitatory trajectory inputs. Consequently, there has been an effort in the research community to understand the influence of trajectories on stability and parameter convergence. One of the key factors which emerged from this research is that accurate parameter convergence requires frequency rich trajectories. This has been proved for linear systems and almost certainly holds true for nonlinear systems[2] [6] [7].

In simulation, single frequency sine wave trajectories proved adequate for convergence; on the real system however, this was not the case. This result is not surprising when one considers the single frequency content of a sinusoidal signal. Ideally, the input trajectory should have an infinite frequency component. This can be achieved theoretically by applying a train of impulses or applying white noise to the system. Unfortunately, these signals cannot be applied in practice. In experiments, it was found that a triangular wave trajectory gave the best parameter convergence results. Although a rigorous explanation is not available to substantiate this finding, it is possible, in hindsight, to hypothesise on the intuitive factors which contribute to this result.

A mechanical system can be approximated to a double integrator, such that a force is applied to the input and the output is a displacement. Since twice differentiating a ramp position trajectory gives an acceleration profile made up of impulses, full frequency content is experienced at the force input to the system. This explanation is easily understood from an intuitive level; however, a more rigorous explanation is beyond the scope of this thesis.

2.3. Sensor Input Scaling The most volatile period of the adaptive control calibration is the initial one or two seconds after start up. This is because the initial parameter estimates, in the manipulator model, are all zero. In some experiments, it was found that during this initial transient stage, parameter overflow would occur causing instability of the adaptive control. There are several reasons for this instability which is not taken into account within the theory.

- Sampling delays caused by the discretization of the system.
- The upper bound of the gain matrix being too high.
- Magnitude of sensor inputs being too large.

Unfortunately sampling delays are unavoidable and compensating for them will complicate the control system further and increase computational requirements.

The upper bound on the magnitude of the gain matrix is an effective cure for the overflow problem; however in experiments, it was found that by lowering the maximum magnitude of the gain matrix P, convergence becomes dramatically slowed during the less

[2] In Boyd and Sastry it was shown that a linear system of order n required $2n$ spectral lines to achieve convergence [6]. This is remarkably similar to the well known Shannon sampling principle in Communications which states that to reproduce a sampled signal requires at least two times the frequency of the input signal.

volatile stages of the calibration. This can be easily remedied in software by changing the upper bound on P on-line, but this is not an elegant solution, and could introduce discontinuities to the control.

A more effective solution is to scale the sensor inputs. The output of the position sensor is a voltage in the approximate range of −6 Volts to +6 Volts. The polynomial approximation causes the sensor output to be raised to the power of two, or three depending on the degree of the polynomial. This relationship causes the components in the regressor model to become quite large causing numerical instability during the volatile stage. Fortunately this problem can be easily fixed by scaling the sensor output such that:

$$\|q_i\| \leq 1 \implies \|q_i^n\| \leq 1 \ \forall n > 0 \tag{5.3}$$

Therefore by scaling the position sensor output to between +1 and −1, the regressor matrix becomes bounded ensuring greater stability, especially during the the initial transient stage. Experiments showed that this was an effective way to stop numerical overflow without reducing the upper bound on the gain matrix P.

2.4. Degree of Polynomial Experiments showed that the higher the degree of the polynomial, the slower the convergence. It was found that the polynomial coefficient associated with the highest degree term, in the regressor model, converged the slowest. (i.e. For a cubic polynomial, the coefficient associated with the q^3 term in the regressor matrix.) Consequently, repeated experiments showed that a quadratic polynomial was favourable over a cubic polynomial with respect to both convergence time and even overall accuracy. For this reason, all experiments using the polynomial approximation use a polynomial of degree two.

2.5. Use of a Dead-Band in the Adaptive Controller The adaptive control parameter estimates have a tendency of creeping from their nominal values. This phenomena is common in adaptive controllers once the estimation system has reached steady state. This is counteracted by inserting a dead-band into the controller causing any small deviations in the parameter update to be ignored.

2.6. Gain Tuning An argument could be made that a truly adaptive controller should be able to tune all its internal gains based on system feedback alone. Unfortunately this is not the case, and several critical gains must be tuned. In particular, the linear feedback gains K_P and K_D, the forgetting factor λ_0, the upper bound on the gain matrix k_0, the relative contributions of the indirect and direct parameter update, and the initial value of the gain matrix P_0. Over the course of algorithm implementation it was found that there is no unique strategy for successful calibration. Overall, the following conditions were found to be best for calibration using adaptive control.

- The values of linear feedback K_P and K_D, should be at least half of the value which would normally be used for PD feedback alone. There is a trade off which occurs when selecting these values. By choosing low values of proportional and derivative feedback, the adaptive part of the controller is forced to do more work. However, this comes at the expense of trajectory tracking ability.
- The forgetting factor λ_0 dictates the amount of memory within the system. If the value is too high, the adaptive controller will forget past experience and will fit only most recent data; consequently parameter estimation evolution will be oscillatory because factors such as random noise will not be averaged out over time. Conversely, a low forgetting factor decreases the adaptive controller's propensity to adapt to dynamically changing parameters. However, since the calibration assumes that the robot parameters are constant, a low forgetting factor is preferable.
- The maximum bound on the magnitude of the gain matrix (k_0) proved to be the least sensitive parameter in the composite adaptive control framework. This is because the magnitude of the gain matrix did not require such a high stimulus to invoke the upper bound. For this reason this was not a critical factor.
- The initial value of P_0 is critical for fast convergence. Even though the gain matrix $P(t)$ is dynamic, there is still a lag time for it to reach an optimal value. If P_0 is too high, numerical instability may ensue after startup. It was found that higher order coefficients required a higher initial value. For a quadratic polynomial, the diagonal elements of P_0 were set at: 8.0 for the coefficient of q^2, 3.0 for the coefficient of q, and 1.0 for the constant term.
- The contribution of the indirect and direct adaptive control components in the composite adaptive control are critical. Recall from (3.40):

$$\dot{\hat{a}} = -P(t)\left(C_1 Y^T(\theta,\dot{\theta},\dot{\theta}_r,\ddot{\theta}_r)s + C_2 W^T(\theta,\dot{\theta})e\right) \tag{5.4}$$

  The constant term C_1 and C_2 dictate how much weight should be put on the trajectory error and the torque error . In experiments it was found that the direct adaptive control constant should be in the order of $C_1 = 0.3$, and the indirect adaptive control constant in the order of $C_2 = 1.0$.

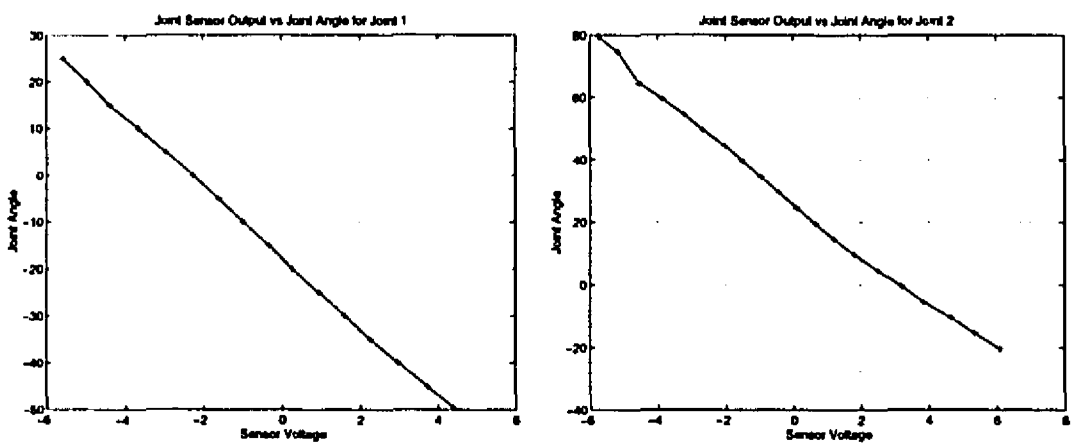
In general, the nonlinearity of the system makes it difficult to find optimal and unique values for these parameters.

2.7. Hardware Inaccuracies Throughout the theoretical derivation there is an assumption that all transducers and actuators respond in a linear manner. In reality this is not the case.

Each channel has two motors which work in a pull-pull configuration which is switched by two diodes. Since the diodes are not ideal, the switching action causes nonlinearities. What is more, the motors are second order electro-mechanical systems and therefore do not have a perfect constant gain response which is assumed in the control.

The transmission, made up of the tendons, is also assumed ideal. However in reality the tendons add both damping and elasticity to the system [17].

The theory developed in Chapter 4 assumes that the relationship between sensor output and joint angle is linear. The position sensors however, which have been made "in house", are not perfectly linear. In Figure 5.6 the measured relationship between sensor output and joint angle is plotted.



(a) (b)

Figure 5.6: Actual Relationship Between Sensor Output and Joint Angle. (a) Joint 1. (b) Joint 2.

The slight nonlinearity exhibited in Figure 5.6 will force the adaptive control to average out the imperfections. The low forgetting factor and the best fit strategy of the dynamic least squares formulation should give the best average value. This will affect the accuracy and the repeatability of the calibration.

Unfortunately the inaccuracies caused by the hardware are inevitable. From the perspective of calibration, the strategy must be to minimise these effects. This is done by not exciting high frequencies causing motor gain attenuation, and preventing fast oscillatory movements making the elasticity of the tendon a factor. Fortunately the calibration stage enables a trajectory to be chosen which can minimise these effects.

3. Implementation Assumptions

The goal of this research is to power up a robot and have it self calibrate with a minimum of constricting assumptions. There are however, several assumptions which are made for the implementation, but they do not restrict its general operation. The assumptions are:

- The manipulator model is known; this a fundamental assumption of STAC adaptive control methods.
- The sensor outputs and torque commands are uniform. In essence, the sign of position and velocity must correspond such that a positive torque will move the manipulator in a positive direction with respect to sensor output. This is necessary to prevent positive feedback in the PD loop of the adaptive control.
- The joint limits are known in terms of sensor output. This prevents the robot from trying to exceed its workspace.

These are the only assumptions made in the adaptive control joint calibration.

4. Experiments

In the proceeding sections the performance of the single link polynomial approximation is studied. The first experiment is performed on the second joint of the robot. The focus of this study will be on polynomial convergence and convergence using the trigonometric expansion. This serves as an applied feasibility study of the algorithm. The second experiment uses a polynomial approximation to estimate the gain parameter of the first joint. This stage tests the assumption that the other joints can be held in an arbitrary position during calibration without causing significant interference through coupling.

4.1. The Polynomial Approximation on a Single Link The distal link of the two degree of freedom robot is stimulated with a ramp trajectory. The adaptive control uses the single link robot model:

$$[\tau] = \begin{bmatrix} \ddot{q} & g & gq & gq^2 \end{bmatrix} \begin{bmatrix} \alpha m l_c^2 \\ a \\ b \\ c \end{bmatrix} \tag{5.5}$$

The results of the parameter estimation are shown in Figure 5.7.

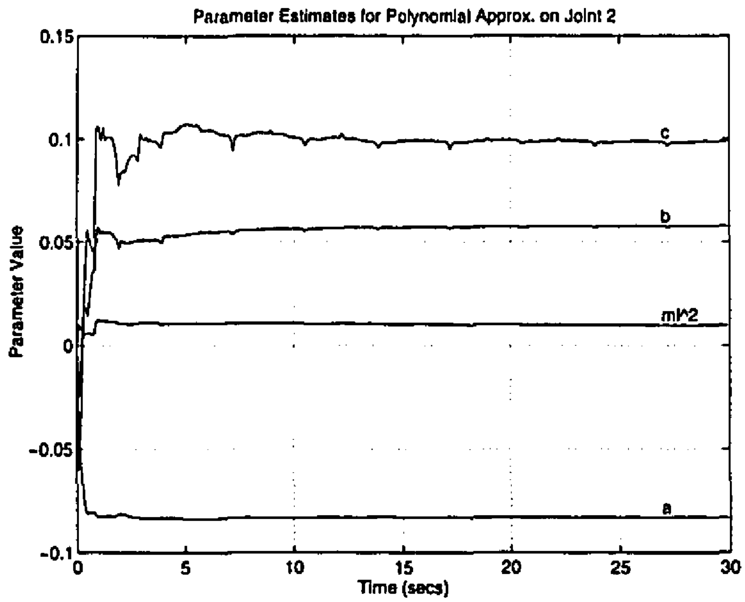


Figure 5.7: Parameter Estimate Evolution for Polynomial Approximation on Joint 2

Figure 5.7 shows that the initial convergence to approximate values is quite fast, settling to an initial estimate within one second. The exponential nature of the convergence after one second is apparent as the parameters slowly converge to nominal values over the next twenty five seconds. The order of polynomial coefficient convergence is a, b, and then c. Therefore the higher the exponent in q, the slower its coefficient converges.

Using the polynomial found in the adaptive control, the values of joint gains and joint offsets can be derived. In Figure 5.8, the polynomial estimated by the adaptive control is compared to the function $ml\cos(\alpha q + \beta)$, where α, β, and ml have been derived from the adaptive control estimates. It can be concluded from Figure 5.8 that the cosine function can be accurately fitted to the polynomial making accurate calibration viable.

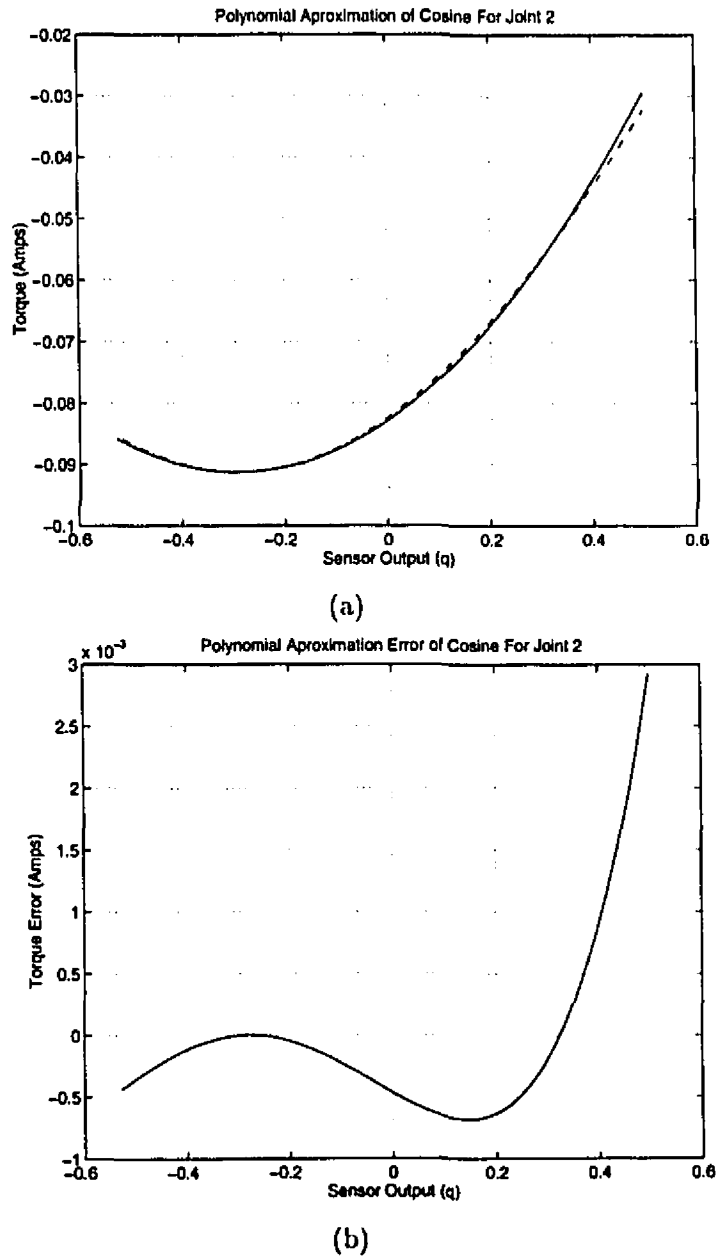


Figure 5.8: Polynomial Approximation of Cosine Function for Joint 2. (a) Polynomial Approximation (Solid Line) Cosine Fitting (.-) (b) Fitting Error

The results of the single link joint calibration are given in Table 5.1. The results show the calibration has been successful to an accuracy of approximately 1 degree[3].

	α $degrees/Volt$	β $degrees$
Manual Calibration	9.052	26.299
A-C Calibration	8.754	25.559
Absolute Error	0.298	0.740

Table 5.1: Accuracy of Adaptive Control Calibration For Joint 2

Using the adaptive control with the friction compensation also improves the overall performance of the manipulator. In Figure 5.9 the position error for the system with and without adaptive control is shown.

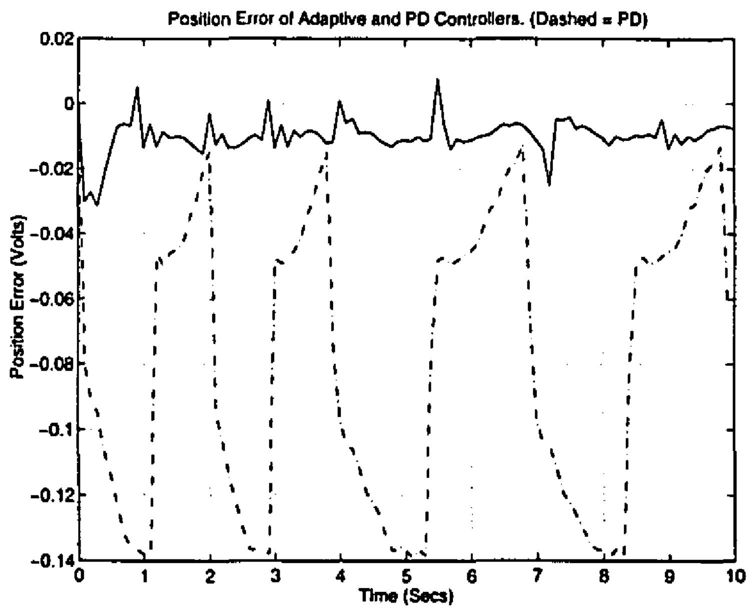


Figure 5.9: Position Error for Adaptive and PD Controllers. Solid=Adaptive Control Position Error, Dashed = PD Control Position Error

[3]These results do not consider the repeatability of the experiments – a critical factor in robot calibration. This topic is addressed in Section 8.

The improvement in position tracking is approximately a factor of 10. This is a good example of the duality of the algorithm as both a joint calibration scheme and an accurate controller. Once the joint calibration adaptive control framework is implemented, other general purpose adaptive controllers can be implemented with very little extra effort.

The results presented in this section show that joint calibration, using the polynomial approximation, achieves good results. The convergence for the single (distal) link is rapid and definite. The parameters estimates do not fluctuate which indicates that the model is an accurate representation of the robot. The distal link is not subject to coupling from other links. Therefore in the next section, the inner link is calibrated with the distal link held in a PD feedback loop.

5. The Polynomial Approximation Using the Inner Joint

The preceding section established the success of the polynomial approximation on a single link. This link however, was not subject to possible coupling caused by holding other joints at arbitrary positions using PD feedback. In this section, joint 1 is calibrated while joint 2 is held in an arbitrary position.

Figure 5.10 shows the parameter evolution for the inner joint. The behaviour of the parameter evolution is quite similar to the the uncoupled single link experiment in the previous section. (Figure 5.7) Good approximations are found within approximately 1 second for all parameters except parameter c, which overshoots its nominal value. The settling time of parameter c requires almost 7 times the settling time of the other parameters. This result is in accordance with the uncoupled link of the preceding section which also exhibited the slowest convergence for the coefficient associated with the highest order term, in this case q^2.

The result of Figure 5.10 also shows that the convergence is definite. The exponential convergence is uniform, and there are no oscillations in the parameter estimates. This indicates that the single link model was accurate and joint coupling caused by holding joint 2 did not adversely affect estimation. Also, the non oscillatory nature of parameter estimates purports that the estimates are accurate.

Figure 5.11 shows the polynomial approximation to the cosine function and the cosine fitting error. This shows that the polynomial successfully approximated the cosine function. The small error between the cosine and the polynomial function implies that the gain

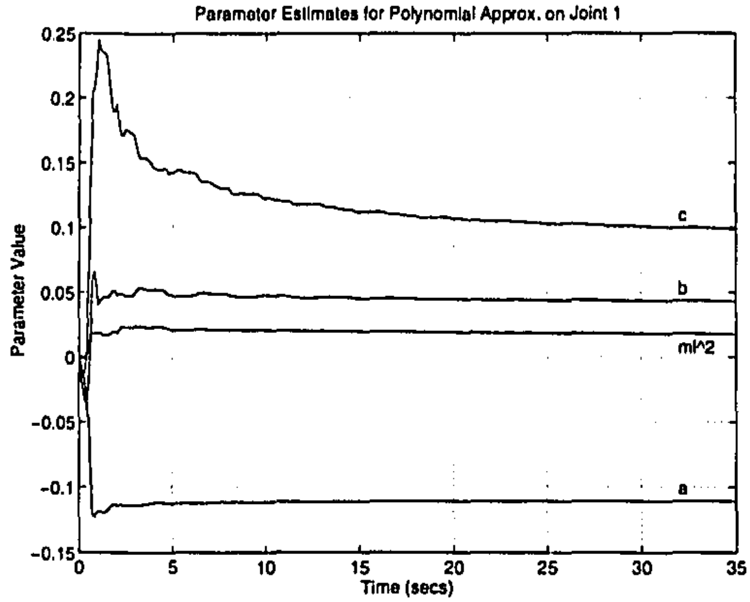


Figure 5.10: Parameter Estimate Evolution for Polynomial Approximation on Joint 1

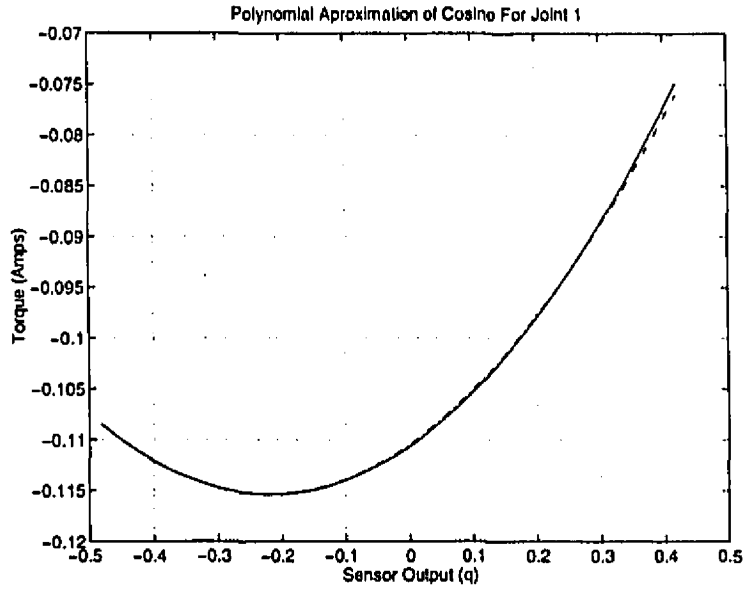
parameter α, will be an accurate representation of the physical joint gain parameter which the experiment attempts to find.

The accuracy is confirmed in Table 5.2 which shows the estimated and the measured values of the gain parameter α.

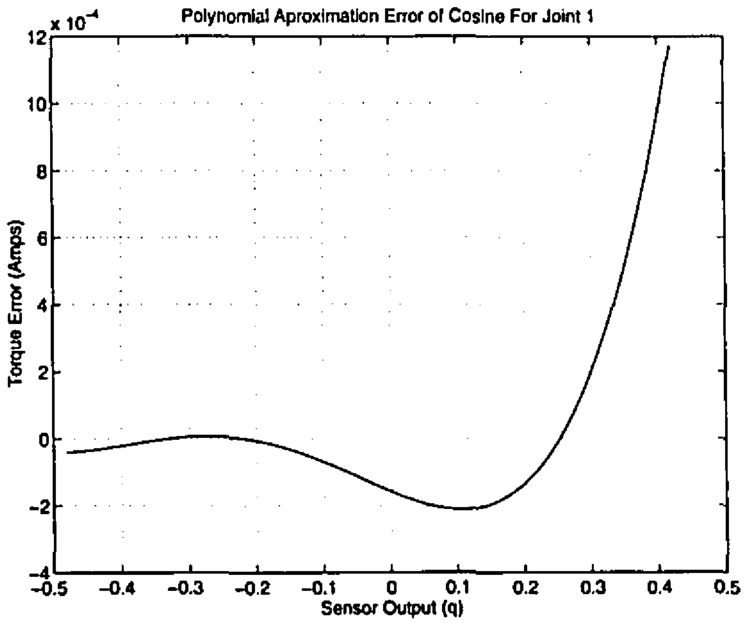
	α *degrees/Volt*	β *degrees*
Manual Calibration	7.638	N/A
A-C Calibration	7.534	16.737
Absolute Error	0.104	N/A

Table 5.2: Accuracy of Adaptive Control Calibration For Joint 1

The accuracy of the gain parameter estimation confirms that locking all but one of the joints in a PD feedback loop, and using the single link model for calibration on remaining joint, is valid.



(a)



(b)

Figure 5.11: Polynomial Approximation of Cosine Function for Joint 2. (a) Polynomial Approximation (Solid Line) Cosine Fitting (.-) (b) Fitting Error

There is some degradation of control when controlling the joint subjected to coupling which is evident in the position tracking plot of Figure 5.12.

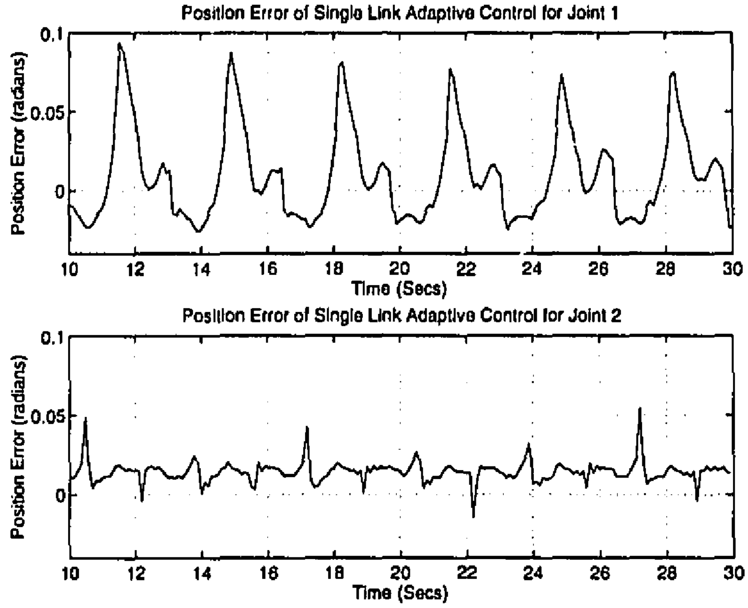


Figure 5.12: Position Error for Adaptive Controllers Using Polynomial Approximation for Joint 1, and Joint 2. Top: Position Error of Joint 1. Bottom: Position Error of Joint 2

The position error of the coupled joint is on average twice as a high as the uncoupled joint 2. This is caused by the movement, and hence coupling, caused by the locked joint 2 affecting joint 1. This coupling effect is inevitable when considering that the PD feedback loop is essentially a mass-spring-damper system and therefore causes unwanted flexibility in the joint.

6. Trigonometric Expansion Convergence for a Single Link

The preceding sections showed that the polynomial approximation, using the single link model, is feasible for calibration of the gain parameters. To complete the joint calibration algorithm, the angle offset values must be found using the technique of trigonometric expansion. As a proof of concept, and to give comparison for the multiple link case, the trigonometric expansion is applied to only joint 2 of the hand-controller.

The model for the single link trigonometric expansion dynamics is:

$$[\tau] = \begin{bmatrix} \ddot{q} & g\cos(\alpha q) & -g\sin(\alpha q) \end{bmatrix} \begin{bmatrix} \alpha m l_c^2 \\ m l_c \cos(\beta) \\ m l_c \sin(\beta) \end{bmatrix} \tag{5.6}$$

Using (5.6), and the joint angle gain information found in Section 4.1, the adaptive control calibration is performed. The evolution of parameters are shown in Figure 5.13

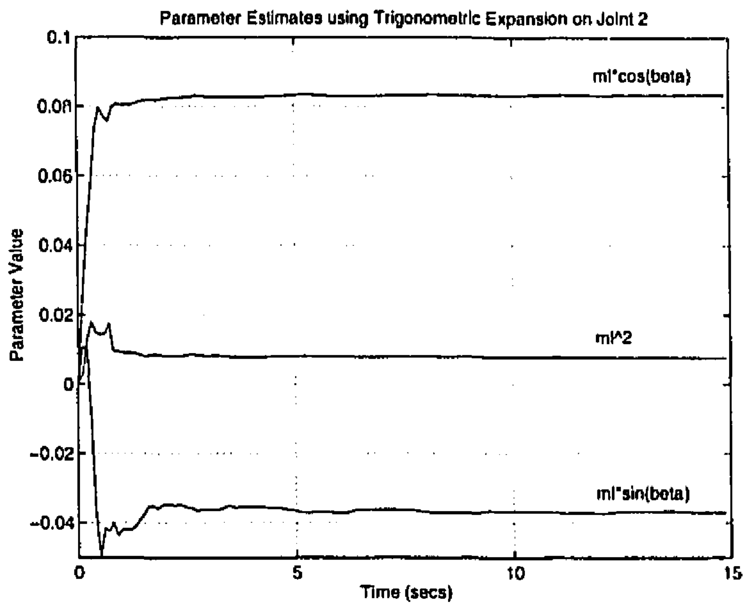


Figure 5.13: Parameter Estimate Evolution for a Single Link Manipulator Using Trigonometric Expansion

The parameter estimates converge quickly; within approximately 5 seconds all parameters converge to nominal values. This quicker convergence, when compared to the polynomial approximations, can be attributed to the greater structure imposed by the trigonometric based model.

The results of the calibration are shown in Table 5.3. The error in the offset value is just over one degree. This is a larger offset error than the polynomial approximation which had a joint offset error of 0.740 degrees. The larger error is a result of the propagation of error caused by using the joint gain found in the polynomial approximation. Since the gain parameter is used within the manipulator model, the model can no longer represent the

	β (*degrees*)
Manual Calibration	26.299
A-C Calibration	24.927
Absolute Error	-1.372

Table 5.3: Accuracy of Adaptive Control Calibration For Joint 2: (Using Trigonometric Expansion)

system so well. The effect of this is shown in slight oscillatory motion in the parameters estimates in Figure 5.13. It can be concluded however, that trigonometric expansion within adaptive control is feasible. It remains to test this concept on the complete manipulator.

7. Joint Offset Estimation Using the 2 DOF Robot

To estimate the joint offset parameters, the robot must be operated under adaptive control using the 2 DOF robot model with trigonometric expansions. The robot model used in experiments is a simplified version of (4.39):

$$\begin{bmatrix} gc_{\alpha_1 q_1} & -gs_{\alpha_1 q_1} & g\cos(\alpha_1 q_1 + \alpha_2 q_2) & -g\sin(\alpha_1 q_1 + \alpha_2 q_2) & \alpha_1 \ddot{q}_1 & \alpha_1 \ddot{q}_1 + \alpha_2 \ddot{q}_2 \\ 0 & 0 & g\cos(\alpha_1 q_1 + \alpha_2 q_2) & g\sin(\alpha_1 q_1 + \alpha_2 q_2) & 0 & \alpha_1 \ddot{q}_1 + \alpha_2 \ddot{q}_2 \end{bmatrix}$$

$$\begin{bmatrix} (m_1 l_{c1} + m_2 l_1)\cos(\beta_1) \\ (m_1 l_{c1} + m_2 l_1)\sin(\beta_1) \\ m_1 l_{c2}\cos(\beta_1 + \beta_2) \\ m_1 l_{c2}\sin(\beta_1 + \beta_2) \\ m_1 l_{c1} + m_2 l_1^2 \\ m_2 l_{c2}^2 \end{bmatrix} \tag{5.7}$$

In (5.7) the contribution due to coupling forces is not required because the actuators are mounted remotely. The robot is purposely operated in a slow trajectory to reduce the centripetal and unmodeled dynamic effects. This enables the contribution due to centripetal forces to be neglected in the manipulator model. By neglecting these parameters, it is possible to keep the size of the regressor matrix within bounds that enable a high sampling rate. Experiments showed that the effect of a reduced sample rate was more detrimental than neglecting centripetal forces.[4] It is important to ensure that the adaptive

[4]The contribution of centripetal forces are approximately an order of magnitude lower than that of gravity terms; this is because the link lengths are small. ($\approx 0.1m$)

control is exposed to as many joint configurations as possible. This is done by running each joint in a ramp trajectory with different periods.

The convergence of the gravity based parameters are shown in Figure 5.14. The convergence of the parameters requires approximately 20 seconds to nominal values and a further 25 seconds for finer approximations.

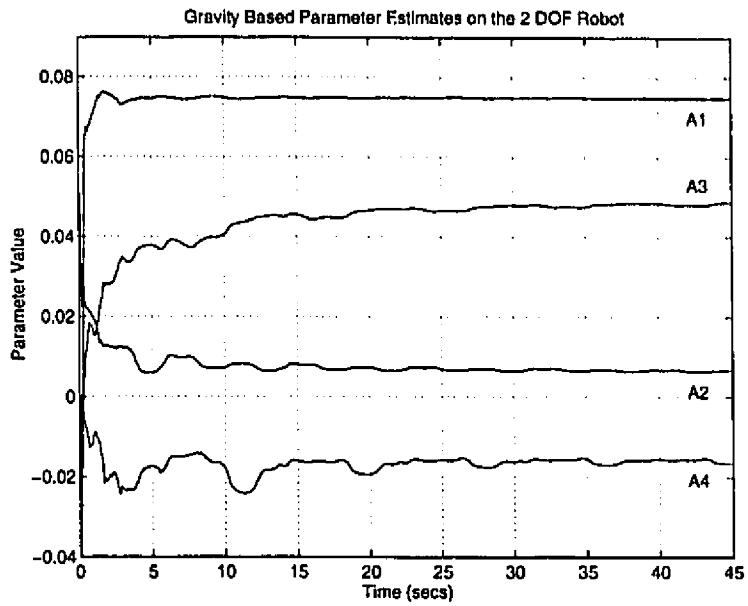


Figure 5.14: Gravity Based Parameter Estimate Evolution for Joint Offset Calibration on 2 DOF Robot. (A1= $(m_1 l_{c1} + m_2 l_1)\cos(\beta_1)$, A2 = $(m_1 l_{c1} + m_2 l_1)\sin(\beta_1)$, A3 = $m_1 l_{c2}\cos(\beta_1 + \beta_2)$, A4 = $m_1 l_{c2}\sin(\beta_1 + \beta_2)$)

It is clear that the convergence for the two link robot is not as good as the single link convergence in Figure 5.7. The parameter estimates exhibit oscillatory behaviour, in particular parameter $A4$. This is an indication that there are dynamics within the system, which are not included in the computer model, affecting estimation accuracy. This is in part due to the absence of the centripetal forces and in part due to dynamics which cannot be modelled. The contribution of the unmodeled parts of the robot dynamics are small with respect to the magnitude of the gravitational components shown in Figure 5.14. For example, the inertial components of the system have magnitudes of almost an order of magnitude smaller than the gravitational terms. (See Figure 5.15.)

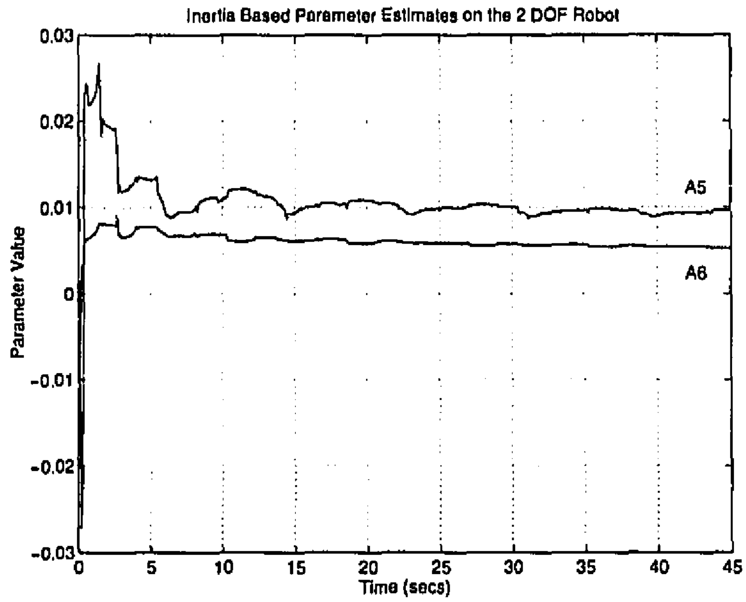


Figure 5.15: Inertial Parameter Estimate Evolution for Joint Offset Calibration. (A5 = $m_1 l_{c1} + m_2 l_1^2$, A6= $m_2 l_{c2}^2$)

Despite the increased problem in accurate robot modelling, the adaptive controller is adept in averaging out error within the system. This enables the calibration of the joints without full knowledge or use of the dynamic model.

It should also be noted that the tuning of the feedback gains, update gains, and forgetting factors was much more difficult for the full two degree of freedom case. In general, the preceding experiments were quite robust to changes in these parameters, whereas the multiple link case had a much narrower margin of error. Analysis showed that the two degree of freedom robot was much more prone to spurious forces, especially during velocity sign changes in the triangular wave trajectory. These transients are impossible to model forcing greater emphasis on tuning the adaptive controller to ensure the transient effects are averaged out; however this must be done without suppressing the predictable response.

The results of the calibration are given in Table 5.4.

The results show that the adaptive control calibration is accurate to within approximately $\pm 1°$. The parameter information found in the adaptive control can also be used for gravity compensation of the hand-controller.

	α_1 *degrees/Volt*	β_1 *degrees*	α_2 *degrees/Volt*	β_2 *degrees*
Manual Calibration	-7.638	-17.544	-9.052	26.299
A-C Calibration	-7.534	-17.988	-8.755	26.595
Absolute Error	0.104	0.444	0.297	0.266

Table 5.4: Accuracy of Adaptive Control for the Complete Robot Calibration

8. Consistency of Results

The results given in the preceding section represent just one complete calibration. What is most important however, is the consistency of results. The algorithm must be relied upon to calibrate the robot within a given accuracy. Table 5.5 gives a summary of results taken over 10 calibration runs.

	α_1 *degrees/Volt*	β_1 *degrees*	α_2 *degrees/Volt*	β_2 *degrees*
Manual Calibration	-7.638	-17.544	-9.052	26.299
Mean A-C Calibration	-7.465	-17.841	-8.537	26.673
Maximum A-C Calibration	-6.963	-16.890	-8.322	27.250
Minimum A-C Calibration	-8.164	-18.115	-8.795	25.273
Standard Deviation	0.465	0.421	0.161	0.615

Table 5.5: Consistency of Adaptive Control Robot Calibration (Over 10 experiments)

The results in Table 5.5 show that the adaptive control calibration gives numerically stable results. The results show that the algorithm has an accuracy of approximately $\pm 1.5^o$ when all factors are considered.

9. Discussion of Experimental Results

From the results of the adaptive calibration, and the experience in implementing the theory on the robot, there are several issues which should be considered.

9.1. Joint Coupling It was mentioned in Section 7 that when the calibration of the joint offsets was performed using two active joints, the control system became less robust to changes in the control system parameters. Convergence to the desired values became more sensitive to changes in linear feedback gains, initial gain matrix (P_0), and forgetting factor. After analysis, it was realized that this was a result of transient torques and forces between joints which could not be predicted within the manipulator model. It is reasonable to assume that as the number of joints are increased, the number of transient forces will also increase; hence the robustness of the algorithm will degenerate with an increase in the number of joints.

9.2. The Need for Accurate Model The calibration algorithm relies on an accurate dynamic and gravity model representation of the robot. If this cannot be achieved, the calibration will be unsuccessful. The hand-controller, for example, has a coupled linkage in its translational stage. This design proved impossible to model accurately forcing the robot to be mechanically constrained to two degrees of freedom for the purpose of the experiments. In general, serial link robots do not pose a difficulty in dynamic modelling, however this may not be the case for parallel mechanisms such as the hand-controller.

The hand-controller tendon transmission is virtually backlash free. Also both motors and sensors do not exhibit measurable hysteresis. (Important design requirements for Hand-Controllers.) Therefore the algorithm has not been exposed to these common and nonlinear elements which would be encountered on geared and hydraulic robots. The effect of these nonlinearities are difficult to model and can be expected to degrade the performance of the calibration in such mechanisms.

For these reasons, the general difficulty in manipulator dynamic modelling is the major weakness of the calibration algorithm.

9.3. The Global Convergence Property The most important property of the adaptive control algorithm is global convergence. All the results in this chapter have been obtained assuming all initial parameter estimates to be zero. In contrast, calibration methods which have relied on static input-output data generally require initial estimates to ensure convergence. It is also interesting to note that the estimation of the joints sensor gain parameter (α). was a more robust and reliable property to estimate than the joint sensor offset (β) when using adaptive control. In comparison, static methods have traditionally had difficulty estimating the joint sensor gain compared to joint sensor offset.

10. Summary

The results of the experiments have shown that the theoretical development derived in Chapter 4 can be successfully implemented on a real robot system. The results showed that the algorithm can be relied upon to calibrate the robot to an accuracy of approximately 1.5^o per joint. It was found, through experimentation, that factors such as excitation trajectory and feed-forward friction elimination are important factors in achieving consistent calibration, factors which were not evident in simulation.

The negative aspect of the algorithm is the requirement of an accurate dynamic and gravity model to achieve consistent calibration. Also, the unmodeled dynamics, largely caused by joint coupling, decreased the calibration robustness and consistency for the multi-link case.

The adaptive control algorithm was able to calibrate the robot autonomously, the operator is only required to switch the system on. The calibration was performed without a priori knowledge of the joint sensor or dynamic properties, without physical constraints on the robot, and without utilising specialised measuring equipment.

CHAPTER 6

Conclusion

The algorithm presented in this thesis enables autonomous joint and dynamic parameter calibration of a multiple link robot. The calibration is performed on-line using adaptive control methods introduced by Slotine and Li. Their adaptive control method is advantageous for calibration because it is globally convergent with respect to both trajectory tracking and parameter convergence. This implies that a priori knowledge of the system parameters is not required to achieve system parameter convergence. The Slotine and Li method however, requires that the manipulator's joints are calibrated before operation.

To achieve autonomous joint calibration using the globally convergent adaptive control method of Slotine and Li, the relationship between the joint sensor output and joint angle or displacement must be estimated within the control. The estimation of system parameters in the adaptive control requires that unknown parameters occur linearly with respect to known quantities such as joint position and velocity. However, rotational joint calibration parameters occur within transcendental functions and therefore cannot be written linearly. This thesis proposed a method to solve this problem by replacing nonlinear trigonometric functions with polynomial approximations. The coefficients of the polynomials occur linearly with respect to the input, and can therefore be used within the adaptive control framework. The resulting polynomial coefficient estimates can then be used to extract joint calibration information.

The resulting algorithm required only knowledge of the manipulator dynamic model and did not require a priori joint sensor information other than the assumption of the linearity of the joint sensor. This means that after switching the power to the robot on, the robot can autonomously find its dynamic and joint sensor properties without human interaction, use of specialised measuring equipment, or physical constraints on the robot.

The algorithm was tested on two links of a six degree of freedom haptic hand-controller. The algorithm was found to be accurate to approximately $\pm 1.5^\circ$. From both the theoretical development and experimental work, several properties, both positive and negative, emerged. These can be summarised as:

- The global convergence property of the algorithm made convergence and numerical stability possible without extra constraints on system observability or initial conditions.
- The algorithm required only matrix addition and multiplication. This reduces stress on the computational requirements of the system and makes real time control possible using ordinary computers.
- The adaptive control algorithm of Slotine and Li does not require inversion of the inertia matrix or measurement of joint accelerations. This makes the algorithm much more robust and applicable to real systems.
- The joint calibration is performed in two stages, first calibrating the joint sensor gains, and then joint offsets. The joint offset estimation requires information found in the previous stage. This makes propagation of error a factor in the final results.
- The algorithm relies heavily on an accurate dynamic, and in particular gravity, model of the robot. If this cannot be achieved, accurate calibration is not possible.
- The reliance on manipulator dynamics limits the overall robustness of the joint calibration algorithm. Real robot systems often exhibit dynamic effects which cannot be modelled. This can cause problems in convergence, especially as the number of links are increased.
- An interesting aspect of the algorithm is that convergence and estimation of joint sensor gains is more accurate and reliable than estimation of joint sensor offsets. Static methods, which generally rely on some sort of least squares fitting, experience the opposite problem, have great difficulty estimating the joint sensor gain and less trouble estimating joint offset.

From these results several conclusions and recommendations can be made for future research.

The experiments were performed on a device which has low backlash and hysteresis. These nonlinearities are difficult to model within the controller. An evaluation of the joint calibration algorithm on hydraulic and geared robots needs to be performed to gauge the affect of these nonlinearities on the algorithm. These nonlinearities will certainly be detrimental. The adaptive controller however, has the ability of averaging out noise and spurious effects which may make calibration still possible under such conditions.

The autonomy of the joint calibration scheme means that in the case of sensor drift, changes in joint calibration parameters can be easily found after each calibration. An interesting application of sensor drift, and even sensor failure monitoring, is to redesign the algorithm so that it can be run in the background during robot operation. This should be possible because the algorithm relies only on ordinary dynamic feedback. Therefore application of a "system watchdog" during operation will allow any unusual changes in system parameters to be monitored.

Finally, as a catalyst for future research, it is intuitive to combine the beneficial properties of both the adaptive control calibration and static calibration schemes. By using the traditional kinematic calibration schemes such as sensor redundancy, or constraining the end-effector, emphasis could be taken off the use of the dynamic model, a negative aspect of the adaptive calibration system. Then adopting an adaptive control approach to benefit from global convergence properties will relinquish the convergence weaknesses of the static approaches. This may come at the expense of a loss of autonomy within the system.

REFERENCES

[1] B. Armstrong. On finding 'exciting' trajectories for identification experiments involving systems with nonlinear dynamics. *IEEE International Conference on Robotics and Automation*, 1987.

[2] O.R. Astley and V. Hayward. Automatic joint calibration using gravity as an invariant. *1995 Robotics and Knowledge Based Systems Workshop*, October 16-18 1995.

[3] K.J. Astrom. Theory and applications of adaptive control – a survey. *Automatica*, 19(5):471–486, 1983.

[4] D.J. Bennet, J.M. Hollerbach, and P.D. Henri. Kinematic calibration by direct estimatin of the jacobian matrix. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 351–357, 1992.

[5] D.J. Bennett and J.M. Hollerbach. Autonomous calibration of single-loop closed kinematic chains formed by manipulators with passive endpoint constraints. *IEEE Transactions on Robotics and Automation*, page 17, 1990.

[6] S. Boyd and S. Sastry. On parameter convergence in adaptive control. *System and Control Letters*, 3:311–319, December 1983.

[7] S. Boyd and S. Sastry. Necessary and sufficient conditions for parameter convergence in adaptive control. *Automatica*, 22(6):629–639, 1986.

[8] G. Canepa, J.M. Hollerbach, and A.J.M.A. Boelen. Kinematic calibration by means of a triaxial accelerometer. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2776–2782, 1994.

[9] J.J. Craig. *Adaptive Control of Mechanical Manipulators.* Adison-Wesley, Reading, 1988.

[10] J.J. Craig. *Introduction to Robotics - Mechanics and Control*. Addison Wesley, 2nd edition, 1989.

[11] J.J. Craig, P. Hsu, and S.S. Sastry. Adaptive control of mechanical manipulators. *Proceedings of the IEEE International Conference on Robotics and Automation*, 1:190-195, 1986.

[12] J.J. Craig, P. Hsu, and S.S. Sastry. Adaptive control of mechanical manipulators. *The International Journal of Robotics Research*, 6(2), 1987.

[13] S. Dubowsky and D.T. DesForges. The application of model reference adaptive control to robot manipulators. *The Journal of Dynamic Systems, Measurement, and Control*, 101(5), September 1979.

[14] V. Hayward. Design of a hydraulic robot shoulder mechanism based on a combinatorial mechanism. *Experimental Robotics 3*, pages 297–310, 1993. Yoshikaswa, T., Myiazaki, F. (Eds.), Lecture Notes in Control and Information Sciences 200, Springer Verlag.

[15] V. Hayward. Toward a seven axis haptic interface. *IROS'95, Int. Workshop on Intelligent Robots and Systems, Pittsbugh, PA,*, 3:134–140, August 1995.

[16] V. Hayward and O. Astley. Performance measures for haptic interfaces. *Preprints of the 7th Int. Symposium on Robotics Research*, Oct. 21–24 1995. Herrshing, Germany.

[17] V. Hayward and M. Cruz-Hernandez. Parameter sensitivity analysis for design and control of tendon transmissions. *Preprints of the ISER-4, International Symposium on Experimental Robotics, Stanford, CA.*, 1995.

[18] J.M. Hollerbach. Advances in robot calibration. *Robotics Research: The Sixth International Symposium*, pages 319–326, 1993.

[19] J.M. Hollerbach, L. Giugovaz, M. Buehler, and Y. Xu. Srew axis measuremnet for kinematic calibration of the sarcos arm. *Proceedings of the 1993 IEEE International Conference on Intelligent robots and systems*, pages 1617–1621, 1993.

[20] J.M. Hollerbach and I.W. Hunter. Robotics: Control without science. *Physics World*, page 17, 1990.

[21] J.M. Hollerbach and D.M Lokhorst. Closed loop kinematic calibration of the rsi 6-dof hand controller. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2:142–148, 1993.

[22] T.C. Hsia. Adaptive control of robotic manipulators. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 183–189, 1986.

[23] P. Hsu, M. Bodson, S. Sastry, and P. Bradley. Adaptive identification and control for manipulators without using joint accelerations. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1210–1215, 1987.

[24] K.K. Khosla and T. Kanade. Parameter identification of robot dynamics. *Proceedings of the 24th Conference on Decision and Control*, pages 1754–1760, 1985.

[25] D. Koditschek. Natural motion of robot arms. *IEEE Conference on Decision Control*, NEED VOLUME:NEED PAGE NUMBERS, 1984.

[26] A. Koivo. Control of robotic manipulator with adaptive controller. *Proceedings IEEE Conference on Decision and Control*, 1:271–276, 1981.

[27] W. Li and J-J.E. Slotine. Parameter estimation strategies for robotic applications. *ASME Winter Annual Meeting*, 5, December 1987.

[28] W. Li and J-J.E. Slotine. Indirect adaptive robot control. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 704–709, 1988.

[29] R.H. Middleton and G.C. Goodwin. Adaptive computed torque control, for rigid link manipulators. *25th I.E.E.E. Conference on Decision and Control*, 1989.

[30] B.W. Mooring, Z.S. Roth, and M.R. Driels. *Fundamentals of manipulator calibration*. John Wiley and Sons, 1991.

[31] A. Nahvi, J.M. Hollerbach, and V. Hayward. Closed-loop kinematic calibration of a parallel drive shoulder joint. *Proceedings of the IEEE International Conference on Robotics and Automation*, 1994.

[32] W.S. Newman and D.W. Osborn. A new method for kinematic parameter calibration via laser line tracking. *IEEE International Conference on Robotics and Automation*, pages 160–165, 1993.

[33] A. Nicosia and P. Tomei. Model reference adaptive control algorithms for industrial robots. *Automatica*, 20(5):635–644, 1984.

[34] J. Prenninger, M. Vincze, and H. Gander. Kinematic calibration and geometrical parameter identification for robots. *IEEE International Conference on Robotics and Automation*, 1:180–185, 1991.

[35] J.S. Reed and P.A. Ioannou. Instability analysis and robust adaptive control of robot manipulators. *Proceedings of the 27th Conference on Decision and Control*, pages 1607–1612, 1988.

[36] J.S. Reed and P.A. Ioannou. Instability analysis and robust adaptive control of robot manipulators. *IEEE Transactions on Robotics and Automation*, pages 381–386, June 1989.

[37] J.M. Renders, E. Rossignol, M. Becquet, and R. Hanus. Kinematic calibration and geometrical parameter identification for robots. *IEEE Transactions on Robotics and Automation*, 7:721–732, December 1991.

[38] C.E. Rohrs, L.S. Valavani, M. Athans, and G. Stein. Robustness of continuous time adaptive control algorithms in the presence of unmodeled dynamics. *IEEE Transactions in Automated Control*, 30:881–889, 1985.

[39] Z.S. Roth, B.W. Mooring, and B. Ravani. An overview of robot calibration. *IEEE Journal of Robotics and Automation*, 3(5):377–385, October 1987.

[40] H Seraji. A new aproach to the adaptive control of manipulators. *ASME Journal of Dynamics Systems, Measurement and Control*, 109:193–202, 1987.

[41] J.J.E. Slotine and W. Li. On the adaptive control of robot manipulators. *A.S.M.E. Winter Annual Meeting, Anaheim, CA*, 1986.

[42] J.J.E. Slotine and W. Li. Adaptive manipulator control: A case study. *IEEE International Conference on Robotics and Automation*, 3:1392–400, 1987.

[43] J.J.E. Slotine and W. Li. Adaptive robot control: A new perspective. *Proceedings IEEE Conference on Decision and Control*, 1:192–198, 1987.

[44] J.J.E. Slotine and W. Li. On the adaptive control of robot manipulators. *The International Journal of Robotics Research*, 6(3), 1987.

[45] J.J.E. Slotine and W. Li. Adaptive robot control: A case study. *IEEE Transactions on Automatic Control*, 33:995–1003, 1988.

[46] J.J.E. Slotine and W. Li. Composite adaptive control of robot manipulators. *Automatica*, 1:509–519, 1989.

[47] J.J.E. Slotine and W. Li. *Applied Nonlinear Control.* Prentice Hall, 1 edition, 1991.

[48] M.W. Spong and M. Vidyasagar. *Robot Dynamics and Control.* John Wiley and Sons, 1 edition, 1989.

[49] G.R. Tang and L.S. Liu. Robot calibration using a single laser displacement meter. *Mechatronics*, 3:503–516, 1993.

[50] R.G. Walters and M.M. Bayoumi. Aplication of self-tuning pole placementregulator to an industrial manipulator. *Proceeding of the IEEE Conference on Decision and Control*, 1:323–329, 1982.