UNIVERSITY OF SOUTHAMPTON

Identification of Electrically Stimulated Muscle after Stroke

by

Fengmin Le

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering and Applied Science
Department of Electronics and Computer Science

April 2011

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Fengmin Le

Stroke affects a large percentage of the population in UK and one of the most devastating and common consequences of the stroke is loss of the use of the arm and hand. Currently there is increasing interest in the application of control schemes as part of a rehabilitation programme for survivors of a stroke. Functional Electrical Stimulation is applied, together with the model-based controller in order to ensure that the assistance provided coincides as much as possible with the patient's voluntary intention. The difficulty encountered is lack of a reliable model of electrically stimulated muscle. Motivated by this, this thesis focus on identification of electrically stimulated muscle, especially the impaired arm after stroke.

After studying the muscle behaviors and reviewing the existing muscle models, Hammerstein structure is chosen to model the nonlinear dynamics of the electrically stimulated muscle under isometric conditions. Firstly, batch identification algorithms are considered. A two-stage algorithm is proposed, together with its identification procedure and comparison results on a stimulated muscle system. Due to its simple implementation and good performance, this algorithm has been developed to the later two iterative algorithms. Experimental results are used to demonstrate the superior performance of the algorithms and the model structure when compared with others.

Further more, considering the slowly time-varying properties of the muscle system, recursive identification of Hammerstein structure is investigated later in the thesis. A novel recursive identification algorithm is developed, where the linear and nonlinear parameters are separated and estimated recursively in a parallel manner, with each updating algorithm using the most up-to-date estimation produced by the other algorithm at each time instant. When compared with the leading technique involving over-parametrization together with a Recursive Least Squares algorithm on numerical examples and experimental data, the proposed algorithm exhibits superior performance.

Finally, the identified muscle models have been used in FES control schemes for electrically stimulated muscle under isometric conditions and iterative learning controllers will be used since the repeated nature of the task. Besides the two nonlinear ILC approaches, several trial-dependent and adaptive control schemes has been designed and implemented in the thesis.

Contents

List of Figures

List of Tables

Notations

$G(q)$	transfer function	Page 33
$H(q)$	noise model	Page 33
$f(u)$	nonlinear function	Page 34
θ_n	nonlinear parameter vector	Page 34
$\beta_1 \cdots \beta_{m+2}$	nonlinear parameters	Page 34
θ_l	linear parameter vector	Page 34
$a_1 \cdots a_l \; b_0 \cdots b_n$	linear parameters	Page 34
$u(k)$	input signals	Page 34
$w(k)$	internal signals	Page 34
$v(k)$	noise signals	Page 34
$y(k)$	output signals	Page 34
θ	parameter vector	Page 35
$V_N(\theta)$	cost function	Page 42
$\phi(k)$	regressor	Page 70
λ	forgetting factor	Page 71
$P(k)$	information state in RLS algorithm	Page 71
$\phi_l(k)$	regressor for linear parameters	Page 73
λ_l	forgetting factor for linear parameters	Page 73
$P_l(k)$	information state for linear parameters in ARLS algorithm	Page 73
$\phi_n(k)$	regressor for nonlinear parameters	Page 74
λ_n	forgetting factor for nonlinear parameters	Page 74
$P_n(k)$	information state for nonlinear parameters in ARLS algorithm	Page 74
u_j	control input on the jth trial	Page 93
e_j	tracking error on the jth trial	Page 93
y_d	reference output	Page 93
y_j	output on the jth trial	Page 93

Acronyms

FES	Functional Electrical Stimulation
ILC	Iterative Learning Control
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
PE	parallel (visco-) elastic element
CE	contractile element
SE	series elastic element
FL	force-length
FV	force-velocity
AD	activation dynamics of the stimulation input
AE	active element
IRC	Isometric Recruitment Curve
LNL	Linear-Nonlinear-Linear system
PRBS	Pseudo-Random Binary Sequences
PRMS	Pseudo-Random Multi-level Sequences
SVD	Singular Value Decomposition
SIM	Subspace identification methods
CVA	Canonical Variate Analysis
MOESP	Multivariable Output Error State-space
N4SID	Numerical Subspace State-Space System IDentification
LTI	Linear Time-Invariant
SLS	Separable Least Squares
LS-SVM	Least Squares Supporting Vector Machine
RLS	Recursive Least Squares
ERLS	Extended Recursive Least Squares
PID	Proportional-Integral-Derivative
EMG	electromyographic
ARX	Auto Regressive eXternal
OE	Output-Error
RD	Ramp Deconvolution
LAD	Linear Activation Dynamics
TR	Triangular Ramp

FRN	Filtered Random Noise
SISO	Single Input Single Output
ARLS	Alternative Recursive Least Square
LS	Least Squares
LTV	Linear Time-Variant
NOILC	Norm Optimal Iterative Learning Control
MISO	Multiple Input Single Output

Acknowledgements

Thanks to my husband, who was my boyfriend when I started my PhD, and my family for their continued and invaluable support and encouragement. Sincere gratitude must go to my three supervisors Dr. Ivan Markovsky, Professor Eric Rogers, and Dr. Chris Freeman, without whose help and supervision on my work, it would not have been done. Special thanks to Zhonglun Cai, Daisy Tong and other ISIS people for their friendship and continued help.

Chapter 1

Introduction

1.1 Stroke Rehabilitation

Every year, an estimated 150,000 people in the UK have a stroke, that is, one person every five minutes. Anyone can suffer from a stroke and the percentage increases strongly with age. Similar demographics exist across the EU, and due to an aging population and better acute care, the cost of stroke care, or rehabilitation will continue to increase.

A stroke is a brain injury caused by sudden interruption of blood flow. This will result in a variety of sensory, motor, cognitive and psychological symptoms, such as sensory loss, hemispatial neglect, aphasia, muscle weakness, spasticity, limited movement coordination, attention and memory deficits, depression and behavioral changes. A stroke is the third most common cause of death in the UK and is also the single most common cause of severe disability. More than 250,000 people in the UK live with disabilities caused by a stroke. One of the most devastating and common consequences of the stroke is loss of the use of the arm and hand [Gowland et al., 1992], which causes serious limitations in activities of daily living for the majority of stroke patients and less than 50% have recovered useful upper limb function [Parker et al., 1986; Broeks et al., 1999].

The survivors after a stroke can take part in rehabilitation to overcome or learn to cope with the damage the stroke has caused. Although the dead brain cells cannot start working again, the other parts of the brain can learn to take over from areas that have died, a process known as 'relearning'. A major problem encountered in rehabilitation is so-called learned disuse, which means the patient is unable to practice movements because of impaired motor control. The difficulties to practice and the delay in recovery may lead to a decreased likelihood of recovery occurring [Castro-Alamancos et al., 1992]. Fortunately, technological innovations provide an opportunity to design interventions to stimulate motor relearning. A promising application is the use of rehabilitation robotics to complement conventional therapy. Robotic devices have the possibility to guide movements in a very accurate and reproducible way during specific

parts of a movement and through specific types of guidance, which is hard to accomplish by manual interaction between therapist and patient. Also, there is a growing body of clinical evidence [de Kroon et al., 2002], and theoretical support from neurophysiology [Burridge and Ladouceur, 2001] and motor learning research, to support the use of Functional Electrical Stimulation (FES) to improve motor control. FES is a technique that uses electrical currents to activate nerves innervating extremities affected by paralysis resulting from spinal cord injury, head injury, stroke or other neurological disorders, aiming at restoring function in people with such disabilities. FES can provide the patient with the experience of moving the impaired limb, and when FES is applied coincidently with a patient's voluntary intention whilst performing a task, functional recovery is enhanced [Rushton, 2003].

1.2 Workstation Description

For stroke patients with hemiplegia, one essential function they find very difficult is to reach out from the trunk to, for example, a cup sitting on a table top. When recovering from a stroke, patients go through the same process as an unimpaired person does when learning to play tennis, that is, acquiring the skills necessary by repeated practice. The difficulty is that they can hardly move and consequently do not receive feedback on their progress, and hence get very limited benefit. There is also strong evidence that intervention to help in this situation is best in support of voluntary effort.

Iterative Learning Control (ILC) is a technique for controlling systems operating in a repetitive, or pass-to-pass, mode with the requirement that a reference trajectory $r(t)$ defined over a finite interval $[0, T]$ is followed to a high precision. Examples of such systems include robotic manipulators that are required to repeat a given task to high precision, chemical batch processes or, more generally, the class of tracking systems. Motivated by human learning, the basic idea of ILC [Arimoto et al., 1984] is to use information from previous executions of the task in order to improve performance from pass-to-pass in the sense that the tracking error is sequentially reduced. The objective of ILC schemes is to use their repetitive process structure, that is, information propagation from pass-to-pass and along a pass, to progressively improve the accuracy of the core operation under consideration by updating the control input progressively from pass-to-pass. ILC is the subject of intense research effort, see [Bristow et al., 2006; Ahn et al., 2007] for a starting point on the literature, both in terms of the underlying theory and experimental verification. Recent work ([Freeman et al., 2007a]) has made highly novel use of ILC in robot assisted stroke rehabilitation, including highly promising patient trials.

One of the first research programs to combine FES and robotic therapy together has been undertaken at the University of Southampton. Here a robotic workstation was designed and constructed for use by stroke patients to perform upper limb tracking tasks. This operates in combination with surface FES and their remaining voluntary effort in the hope that they may regain useful voluntary control of their paralysed arm. ILC is used to update the stimulation level in

order to ensure that the assistance provided coincides as much as possible with the patient's voluntary intention.

The robotic workstation consists of a five-link planar robotic arm rigidly connected to an overhead projection system. A subject is strapped to the extreme link and a 6 axis force/torque sensor records the force they apply to the robotic end effector. This also contains an encoder and LEDs to provide visual feedback of the tracking performance. The robotic arm is used to constrain the subject's arm, to impose forces on the end-effector that make the task feel 'natural' to the subject, to apply assistance during the performance of tracking tasks, and to move the patient's arm when necessary. The FES is applied to the triceps muscle, which is one of the muscles primarily affected by stroke, and the patients task is to track a range of reaching trajectories, that are projected onto a target positioned above their hand. Fig. 1.1 shows a stroke participant using the robotic workstation during one of their eighteen treatment sessions, and shows the shoulder strapping used to prevent trunk movement which would reduce the effectiveness of treatment.

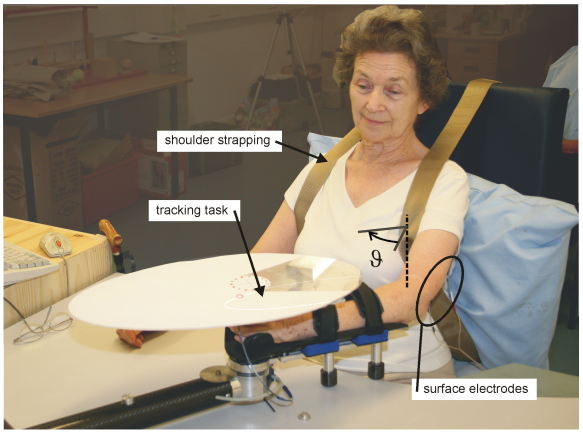


FIGURE 1.1: A Stroke Participant Using the Robotic Workstation

The error between the angle of the forearm in the horizontal plane, $\vartheta(t)$, and the required angle, $\vartheta^*(t)$, is measured during the task, and, at its conclusion, the robot returns the arm to the starting position. Fig. 1.2 shows the control scheme block diagram, which consists of a feed-

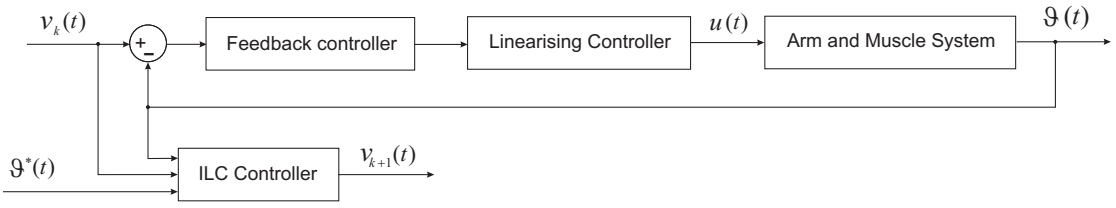


FIGURE 1.2: Block Diagram of ILC Control Scheme

back controller, a linearising controller and an ILC feedforward controller. The former block, taken as a proportional plus derivative controller in the clinical tests, acts as a pre-stabilizer and provides satisfactory tracking during initial trials. During the arm resetting time at the end of trial k, the ILC controller uses a biomechanical model of the arm and muscle system, along with

the previous tracking error, to produce the feedforward update signal $v_{k+1}(t)$ for application in the next trial (full details of the ILC algorithms applied appears in [Freeman et al., 2009a]). The overall performance is clearly dependent on the accuracy of the arm and muscle model, which comprises

- a stimulated muscle structure which accounts for the torque, $y(t)$, acting about the elbow generated in response to the applied electrical stimulation, $u(t)$,
- a kinematic model which gives the component of this torque in the horizontal plane of movement, and
- a two-link system which provides the resulting angular movement, $\vartheta(t)$.

The biomechanical model has been experimentally verified with both unimpaired subjects and stroke patients using a variety of functional parameter forms [Freeman et al., 2009b].

Although the model can predict arm movement resulting from applied FES with reasonable accuracy, experimental data confirms that the model of stimulated muscle adopted is not as accurately identified as the remaining components of the arm. The presence of such modelling inaccuracies necessitated use of relatively low ILC learning gains throughout the clinical trials, but the treatment still resulted in statistically significant improvement for participants across a number of outcome impairment measures [Hughes et al., 2009]. The basic feasibility of the approach was therefore established, but the need for improved modeling of the patient's arm, and the muscle model, in particular, was also highlighted. Hence, identification of the electrically stimulated muscle after stroke is investigated in this thesis.

1.3 Contribution and Thesis Organisation

The main contributions of this thesis can be summarized as follows:

- A specific Hammerstein structure with a cubic spline nonlinearity followed by a transfer function description of linear dynamics is proposed as a suitable model for the electrically stimulated muscle after stroke under isometric conditions, which means the muscle is held at a fixed length.
- A novel identification procedure and corresponding two-stage algorithm is developed. The algorithm is implemented on a simulated muscle system using real experimental data from a stroke patient and outperforms both the ramp deconvolution and separable least squares methods with different noise levels.
- Based on assumed model structures, two iterative algorithms are developed from the two-stage algorithm, where the existing iterative algorithm in [Bai and Li, 2004] has been

extended to encompass an Infinite Impulse Response (IIR) description of the linear component.

- An essential step in any identification algorithm is the selection of appropriate input, or test signals. Here four candidate tests are proposed with particular emphasis on the fact that they will be applied to people, where the staircase test is the first time to be used in this application.

- The iterative algorithms have been applied to experimental data from one impaired subject's arm. Identification, validation and cross-validation results for different tests are analysed.

- Due to the time-varying properties of the muscle system, a novel recursive identification algorithm is developed for the Hammerstein structure. The new algorithm is compared with the over-parametrization approach together with the Recursive Least Squares algorithm in numerical examples and experimental tests.

- The Hammerstein plant model identified by the developed algorithms has been applied to two advanced ILC algorithms to control the isometric muscle system. Following this, two adaptive ILC schemes are proposed in order to better cope with the time-varying muscle dynamics. One is a Linear Adaptor plus Trial-dependent ILC and the other one uses the recursive algorithm developed here to implement online identification, together with an adaptive ILC. Supporting simulation and experimental results are also given.

The following papers are generated based on above contributions:

- F. Le, I. Markovsky, C. Freeman and E. Rogers (2008), "Identification of the Dynamics of Human Arms after Stroke". In: *23rd IAR Workshop on Advanced Control and Diagnosis*, 27-28 November 2008, Coventry University, UK.

- F. Le, I. Markovsky, C. Freeman and E. Rogers (2009), "Identification of Electrically Stimulated Muscle after Stroke". In: *European Control Conference 2009 - ECC'09*, 23-26 August, 2009, Budapest, Hungary.

- F. Le, I. Markovsky, C. Freeman and E. Rogers (2009), "System Identification of Muscle Dynamics for ILC-based Stroke Rehabilitation". In: *UKACC Research Student Event*, 7th May 2009, Institute of Engineering and Technology, Savoy Place, London.

- F. Le, I. Markovsky, C. Freeman and E. Rogers (2010), "Identification of electrically stimulated muscle models of stroke patients". *Control Engineering Practice*,Volume 18. Pages 396-407.

- F. Le, I. Markovsky, C. Freeman and E. Rogers (2010), "Recursive Identification of Hammerstein Systems with application to Electrically Stimulated Muscle". *Control Engineering Practice*. [Submitted]

- F. Le, I. Markovsky, C. Freeman and E. Rogers (2011), "Recursive Identification of Hammerstein Structure", *American Control Conference 2011 - ACC'11*, June 29 - July 1, 2011, San Francisco, California, USA. [Accepted]

- F. Le, I. Markovsky, C. Freeman and E. Rogers (2011), "Online Identification of Electrically Stimulated Muscle Models", *The 18th IFAC World Congress*, August 28 - September 2, 2011, Milan, Italy. [Accepted]

The thesis is organized as follows:

In Chapter 2, a detailed literature review of electrically stimulated muscle models is first given, including a summary of muscle behavior and four categories of existing muscle models. An evaluation of existing identification test procedures is also given. With respect to the identification techniques, seven categories of batch identification algorithm for Hammerstein structures are reviewed, followed by three classes of recursive algorithm. Finally, a brief review of FES control schemes is presented.

Chapter 3 discusses the batch identification of a Hammerstein structure. After a formal statement of the identification problem, a two-stage algorithm is first developed, together with its identification procedure and comparison results on a stimulated muscle system. Then two iterative algorithms are derived for two different linear component representations. In the experimental results section, some important issues in designing tests are discussed and four candidate tests are developed. Based on the identification, validation and cross-validation results from experiments, choice of the best model structure, identification algorithm and candidate test is discussed.

Chapter 4 focuses on the recursive identification of a Hammerstein structure. Firstly, the leading technique involving over-parametrization together with a Recursive Least Squares algorithm is reviewed. Then a novel recursive identification algorithm is developed for a Hammerstein structure. Two algorithms are compared using numerical examples and then applied to the electrically stimulated muscle system.

The identified muscle model is then used to design FES control schemes for the electrically stimulated muscle under isometric conditions in Chapter 5. After description of the Hammerstein plant, two existing advanced model-based ILC algorithms are first applied, followed by the proposal of two adaptive control schemes. Finally, simulation and experimental results are given to show the efficacy of the designs.

The thesis concludes in Chapter 6 with a summary of the progress made and recommendations for further work.

Chapter 2

Literature Review

From birth, humans begin getting to know their world through interaction with the environment and thereby receiving information and learning how to control actions by predicting their effects. These predictions are based on "*Mental*" or "*Intuitive*" models, which are summarized from past experience of the object's behavior. Starting from very simple actions, humans become adept at much more complicated challenges, such as playing tennis and cooking. These same mechanisms can be used to tackle more complex processes such as tidal variation, or the dynamics of airplanes. In this case, an explicit mathematical model is required but the procedure of using experience to inform the model remains the same.

There are basically three steps in the construction of a mathematical model of a given system.

1. Decide on model structure

   In many cases, it is possible to proceed by directly examining the mechanisms that generate signals and variables inside the system. Based on physical or biological or other laws the relationships that the govern the systems behaviour can be constructed. Incomplete knowledge of the system could, however, mean that the parameters of the model are unknown and also there can be uncertainty associated with the final model obtained.

2. Design tests

   To be effective, test signals have to be designed together with deciding which signals to measure and where these measurements should take place. Also which signals to use as inputs needs to be decided. The aim is to make the data obtained maximally informative.

3. Parameter estimation

   This stage constructs a mathematical model from the measured output data produced in response to a known input signal. The parameters in the identified model are estimated using an appropriate criterion or cost function.

The model is then validated by checking its predictive abilities using a new data set which resembles that of the required application. If the model cannot satisfy a desired margin of error, the procedure outlined must be repeated..

In the recent stroke rehabilitation research undertaken at Southampton, model-based controllers are used to update the stimulation level to assist the stroke patients' completion of planar reaching tasks, meaning that muscle models with reasonable accuracy are required. Thus in this application, the objective system is the electrically stimulated muscle after stroke. The remainder of this chapter reviews existing literature related to the modelling of electrically stimulated muscle, and is organized to correspond with the 'three steps' of the modelling process.

2.1 Muscle Models

One potential solution is to use a black box model. However, pure black box models are free of any assumed model structure, and the structure selection will most often require a long testing time which is not acceptable in the application area considered. Consequently a model with a well defined structure is required.

Obviously, no muscle model will be perfect with the variations from the ideal representation due to the assumptions invoked in each case. Some existing models are biophysically based and can capture a variety of muscle behaviour. Others are relatively simple but still are highly accurate in particular cases. Before deciding on the model structure, some prior knowledge of muscle behaviour is necessary in order to capture the features required. Consequently the next section gives a brief description of muscle behaviour under stimulation.

2.1.1 Muscle Behavior

In the 'unimpaired', physiologically intact body, the signal to initiate muscle contraction is generated in the central nervous system. This signal is propagated to and along the peripheral nerve and, via the synapsis, transferred to the muscle, where it induces a contraction. If this natural muscle activation process is interrupted by a lesion such as stroke, the activation signals from the central nervous system cannot reach the muscles, and hence they are paralyzed. FES is a technique to artificially generate an activation signal in the peripheral nerve as follows: when the stimulator sends a signal to the electrodes placed on the muscle body, an electrical field is generated between the two electrodes. This electrical field changes the external potential in the tissue surrounding the nerve. If the depolarization is strong enough, an action potential is induced in the nerve and propagated along the nerve fiber. This action potential is then chemically transferred to the muscle fibers via the synapsis and induces muscle contraction and consequently the tendon force.

- Fiber contraction: Cross-bridge theory

From a microscopic point of view, a set of muscle fibers make up a muscle and, in turn, each fiber is constructed from a very large number of sarcomeres, the basic contractile unit, separated by elastic zones and themselves sheltering filaments. These filaments are made up of "thick" (myosin molecule) filaments surrounded by "thin" (actin molecules) filaments and myosin heads stick out from the myosin filament along the length of the myosin filament except for a region in the middle. When stimulated by an action potential, the muscles generate a force because of the contraction of the every single sarcomere. Next, the cross-bridge theory of the microscopic mechanisms of fiber contraction is discussed.

Huxley [Huxley, 1957] speculated that unbonded myosin heads are in a cocked position, due to a molecule of adenosine diphosphate attached to the myosin head. Meanwhile, an activation potential causes calcium ions to be released by the sarcoplasmic reticulum, and these ions bond to parts of the actin filament. In so doing, the actin filaments conformation has been altered and bonding sites are exposed. The myosin heads attach to these exposed sites to form cross-bridges, which have a different preferred structure that requires the myosin head to rotate while releasing the molecule of adenosine diphosphate. As the myosin head rotates, the overlap between the actin and myosin filaments increases and results in the overall length of the sarcomere shortening. This is called the cross-bridge power stroke. To release the bonds between the myosin and actin filaments at the end of the power stroke, adenosine triphosphate is required. It binds to the myosin head, which then detaches from the actin filament and the muscle fiber relaxes. Subsequently, the adenosine triphosphate is dephosphorylated (releases a phosphate molecule) to become adenosine diphosphate, which once again cocks the myosin head for future attachment to another actin binding site [Kandel et al., 2000].

- Force summation

  Within the muscle, a single action potential causes a twitch contraction, which will result in relatively little force. When action potentials occur more often, cross-bridge formation will increase, and individual twitches will overlap and begin to add together. This is called unfused tetanus. Eventually, when the calcium ion release rate is greater than the rate at which the ions are re-uptaken, a fused tetanus will occur in which a constant force is sustained over a period of time. Overlapping twitch responses is not the only way to generate forces greater than that of a single twitch. It is also possible to generate higher forces by contracting multiple muscles. This is done by muscle recruitment.

- Muscle recruitment

  Muscles can be recruited either by spatial summation, which means the area of stimulation is increased, or temporal summation, i.e., stimulation is sustained in order to increase the concentration of calcium ions required to form cross-bridges.

- Force-length and force-velocity relationships

Under non-isometric conditions, the generated force also depends on the length and velocity of the muscles. Muscles operate with greatest active force when close to their resting length. When stretched or shortened beyond this, the maximum active force generated decreases. This decrease is minimal for small deviations, but the force drops off rapidly as the length deviates further from the ideal. Moreover, the speed at which a muscle changes length also affects the force it can generate. Force declines in a hyperbolic fashion relative to the isometric force as the shortening velocity increases, eventually reaching zero at some maximum velocity. The reverse holds true when the muscle is stretched - force increases above the isometric maximum, until finally reaching an absolute maximum.

Modelling of electrically stimulated muscle has been a widely investigated area, and there exist a large number of models developed for different aspects of muscle contraction under both isometric and non-isometric conditions, considering not only the microscopic-scale mechanics but also the macroscopic-scale relationships. They can be roughly divided into four categories: Hill-type models, biophysical models, mathematical models and Hammerstein-Wiener or Wiener-Hammerstein Models. The better the model, the closer the response to a stimulation input matches that observed experimentally. At the same time, however, their degree of computational efficiency and their feasibility for implementation in a practical controller, is an important issue. Thus, the four types of model, including their performance and feasibility for application, will be summarised next.

2.1.2 Physical Models

Hill-type model is the most popular physical model, which is described by a mass, spring, and damper-like system, and by far the most widely used muscle representatives due to their relative simplicity and their ability to be analyzed by classical mechanical methods. The original Hill model suggested in [Hill, 1938] consists of masses, springs, dampers, and black-box contractile elements, and produce a representation which includes the force-length and force-velocity relationship, see Fig. 2.1. The parallel (visco-) elastic element (PE) models the force generated by the fascicles under passive conditions. In Fig. 2.1, it consists of a damper and a stiffness. This damper is not always present. The contractile element (CE) models the active contractile machinery force of the muscle. This force is transmitted through the series elastic element (SE) to the point of attachment. The SE models the series elasticity of the tendon and aponeurosis. The combination of the contractile and (visco-)elastic elements is referred to as the muscle-tendon-unit.

There is another widely used muscle model, where the PE is parallel to both the CE and SE. Because the SE is much stiffer than the PE over the primary operating range, there is little difference between the two forms. The precise nature of the series and parallel elastic elements differs from model to model. But the general implementation is a nonlinear spring with a limited operating range. The implementation of the contractile element is a more complex part of the

model. It accounts for the generation of active force, which is the product of three independent experimentally measured factors. These factors describe the force-length (FL) property, the force-velocity (FV) property and the activation dynamics (AD) of the stimulation input.

$$F_{CE} = AD \cdot FL \cdot FV \tag{2.1}$$

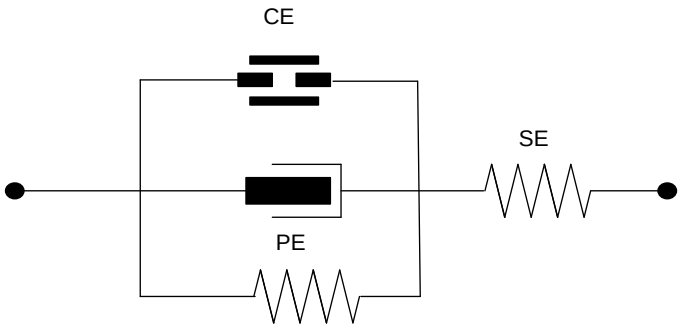


FIGURE 2.1: Hill-type Model Structure

A well-known Hill-type based model was introduced in [Durfee and Palmer, 1994], and is shown in Fig. 2.2. Force is generated through the parallel combination of a PE with force F_{PE} and an active element (AE) with force F_{AE} such that

$$F_{MT} = F_{PE} + F_{AE} \tag{2.2}$$

The PE force is produced by a parallel combination of a nonlinear spring that defines the passive force-length relation f_{PEL} and a nonlinear dashpot that defines the passive force-velocity relation f_{PEV} of the muscle

$$F_{PE} = f_{PFL}(x(t)) + f_{PFV}(\dot{x}(t)) \tag{2.3}$$

The AE force generator consists of a CE in series with a SE. To simplify the identification procedures, SE was neglected by assuming it to be infinitely stiff. Force is generated in the CE through the product of four factors: the normalized stimulation force $f_{STIM}(u,t)$, where u is the stimulation activation strength, the normalized active FL relation $f_{AFL}(x(t))$ where $x(t)$ is the length of the CE, the normalized active FV relation $f_{AFV}(\dot{x}(t))$, and a scaling factor f_{SCALE} that recovers absolute muscle force

$$F_{AE} = F_{CE} = f_{STIM}(u,t) \times f_{AFL}(x(t)) \times f_{AFV}(\dot{x}(t)) \times f_{SCALE} \tag{2.4}$$

The stimulation force is modeled as a Hammerstein structure with a static nonlinearity representing the Isometric Recruitment Curve (IRC), cascaded with linear muscle activation dynamics to produce $f_{STIM}(u,t)$.

In subsequent research, these three factors (FL, FV and AD) were found to be mutually coupled through comparison of the uncoupled model and coupled models in [Shue et al., 1995]. Following this, a coupled model was used by [Chizeck et al., 1999] to identify the electrically

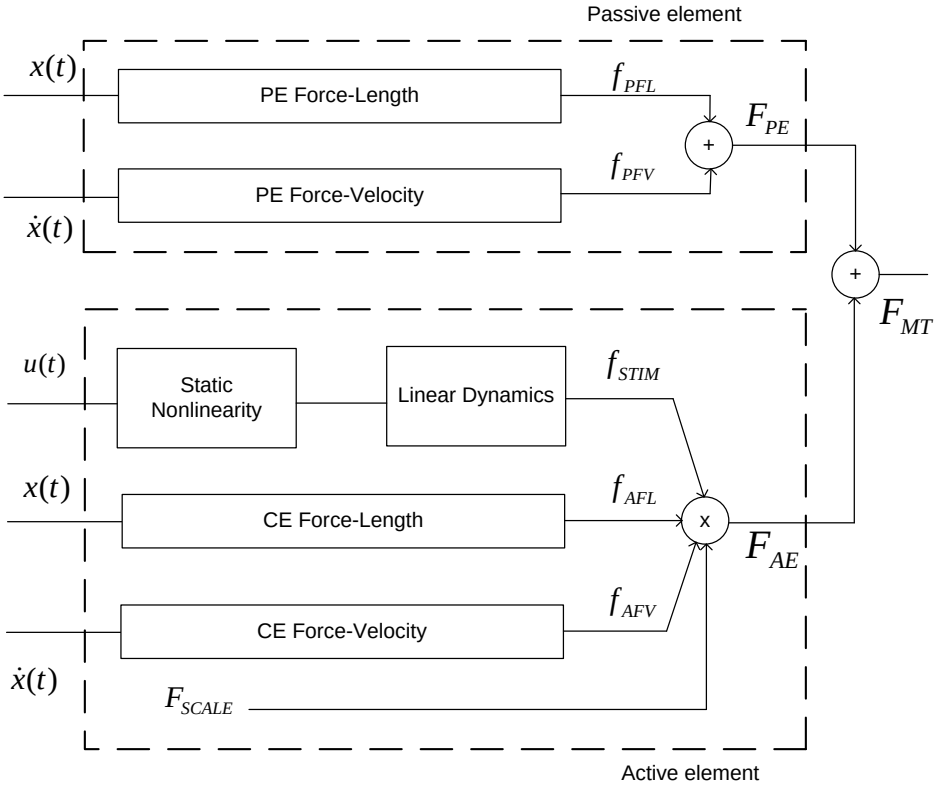


FIGURE 2.2: *Durfee* Model

stimulated quadriceps muscle in paraplegic subjects. This coupled model involves the following components.

Torque-Angle factor (or FL):

$$T_a(k) = 1 + d\Phi(k) \tag{2.5}$$

Torque-Angular velocity factor (or FV):

$$T_v(k) = 1 - cV(k) \tag{2.6}$$

AD factor:

$$A(k) = T_v(k)[a_1 A(k-1) + a_2 A(k-2)] + bu(k-h) \tag{2.7}$$

$$y(k) = A(k)T_a(k) \tag{2.8}$$

Here $\Phi(k)$ and $V(k)$ are the measured angle and angular velocity respectively, $u(k)$ is the stimulation input and a_1, a_2, b, c, d are the parameters to be estimated. The output torque $y(k)$ is the product of two factors: AD $A(k)$ and Torque-Angle $T_a(k)$, given by (2.8), where the AD depends on the Torque-Angular velocity relationship (2.7).

In either coupled or uncoupled form, the Hill-type model is the most commonly used over the past four decades. It has been successfully used in many fields, such as the design of neural prostheses and the design of controllers to adjust the electrical stimulation applied.

2.1.3 Biophysical Models

Hill-type models are useful for gaining an insight into the muscle as described by a mass, spring, and damper-like system, but they are purely phenomenological; that is, they are based only on output behavior and make no reference to the relatively well-understood underlying cross-bridge mechanics causing the behavior. Some researchers have developed models based on cross-bridge kinetics as formulated by Huxley [Huxley, 1957].

Zahalak et al.[Zahalak and Ma, 1990] simplified the classic Huxley two-state contraction model to form a so-called Distribution Moment model, which is fourth-order state variable model for contractile tissue. The four state variables are the first three moments of the actinmyosin bond-distribution function (representing stiffness, force, and elastic energy, respectively) and the free calcium concentration. In spite of the simplification, the bond distribution moment model has a large number of parameters that must be solved nonlinearly.

Another form comprised a combination of Hill-type and Huxley-type elements, such as in [Dorgan and O'Malley, 1997] where a detailed muscle model was to represent many aspects of muscle contraction. This is divided into five consecutive components: Reverse-Order Recruitment Dynamics, the Fiber Active State, Fiber Contraction Dynamics, Derivation of an Excitation Function and Activation Dynamics. Although it was demonstrated that this model was capable of capturing a variety of nonlinear behavior observed in electrically stimulated muscle through simulation results, it is too complicated for control applications.

Wexler et al. developed a model involving three coupled nonlinear differential equations [Wexler et al., 1997]. The first described calcium release and uptake by the sarcoplasmic reticulum. The second equation modeled calcium and troponin binding and release, and the third concerned itself with the force mechanics of cross-bridges. The first two differential equations were with respect to calcium concentration. The third relation was based on a Hill-type model with a spring, damper, and motor in series and modelled force mechanics. This model has been successfully used to predict the force of human skeletal muscles by [Ding et al., 2002]. Moveover, when compared with six other models in terms of its ability to fit forces generated by stimulated ankle dorsiflexors in [Bobet et al., 2005] and with two other mathematical models in terms of fitting experimental data from paralyzed muscles in [Law and Shields, 2005] and [Law and Shields, 2007], it is the most advanced and accurate one.

Models such as those above have the advantage that their parameters can often be directly related to characteristics present in the muscle. However, they tend to be complex and thus computationally expensive. Their parameters are often difficult to identify, and their structure is rarely controller orientated. Therefore, although biophysical models can be accurate descriptors of muscle contractile dynamics, their numerical complexity makes them slow and difficult to realistically implement.

2.1.4 Black-box Models

This category comprises all black-box based models. The simplest choice is a linear model, which can fit experimental data fairly well if the experimental conditions are suitably constrained. For example, a critically damped second-order model has been shown to satisfactorily model the dynamical response of nine different skeletal muscles in [Baratta and Solomonow, 1990]. The muscles are modelled under isometric conditions and the stimulation is applied with fixed pulse width and frequency. Although the double real pole locations and the value of the pure time delay are different from muscle to muscle, a simple second-order linear model can perform very well. Another case is given in [Kirsch et al., 1994], where the muscle stiffness, the dynamic relationship between muscle displacement and force, is modelled as linear during transient movement tests undertaken at given operating points.

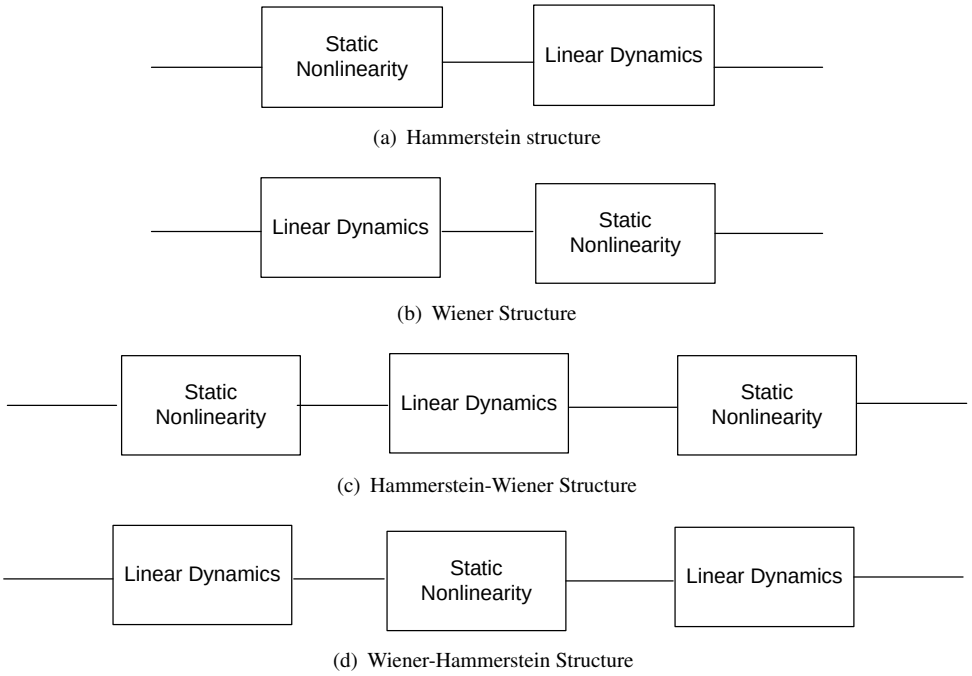
The linear model can be too simple due to the underlying nonlinear behaviour of the muscle. Thus, the parameters of the linear model have been assumed to vary under different experimental environments. In [Bobet et al., 1993], a variety of experimental conditions are tested whilst assuming a critically damped second-order linear model: different stimulation frequency, slow or fast muscle types, rested or fatigued muscles, and maximal or submaximal stimulation voltage. In order to adjust to these variations, the model parameters are allowed to vary between interstimulus intervals. However, this model is still found to be insufficiently flexible because the parameters are constrained to be constant within an interstimulus. In order to react to the more rapidly changing stimulation, a linear time-varying model is employed and a recursive least squares method is implemented using the MATLAB block RARX to estimate the parameters online in [Ponikvar and Munih, 2001].

Another successful example can be found in [Gollee et al., 2001], which in essence closely resembles approaches described above. Second-order linear models are constructed but are only valid for certain operating regions. By blending them together and forming a scheduler to select the model closest to the current operating point, the eventual local model network can capture muscle behavior under a wide range of operation conditions. Despite accurate results in capturing dynamics behavior, a disadvantage of these approaches is that local linearized models are less tractable for developing controllers for practical implementation.

2.1.5 Hammerstein and Wiener Models

The Hammerstein and Wiener models belong to a block-oriented modeling technique, which captures nonlinear dynamic systems using one or two static nonlinear blocks in series with one or two linear ones, see Fig. 2.3. The benefit of breaking the system model into discrete, independent blocks is that the individual blocks may correspond to different natural phenomena. In [Hunter and Korenberg, 1986], the three structures: Hammerstein, Wiener and Wiener-Hammerstein, also known as LNL ('L' represents a dynamic linear subsystem and 'N' represents a static nonlinear subsystem) or "sandwich" systems, have been considered as block structured

approaches for nonlinear system identification, together with identification schemes with special concentration on application to biological systems. This paper provides successive researchers with much inspiration and rich possibilities for models of biological systems. These structures will be reviewed in turn:



(a) Hammerstein structure

(b) Wiener Structure

(c) Hammerstein-Wiener Structure

(d) Wiener-Hammerstein Structure

FIGURE 2.3: Hammerstein-Wiener and Wiener-Hammerstein Structures

- Hammerstein structure

  This model is a system consisting of a static nonlinearity followed by a dynamic linear subsystem, shown in Fig. 2.3(a), has been used so often that it has almost become standard. The reason for using the Hammerstein structure is based mainly on empirical evidence and without any rigid structured association. There is only occasional suggestion that there may be a correspondence to the biophysics: the static nonlinearity $f(u)$ represents the IRC, which is defined as the static gain relation between stimulus activation level and output torque when the muscle is held at a fixed length and the linear dynamics $G(q)$ represents the dynamic response of electrically stimulated muscle.

  The history of using a Hammerstein structure in muscle modeling can be traced back to 1967. In [Vodovnik et al., 1967], a Hammerstein model is used to describe muscle behavior and then a closed-loop controller is developed for an elbow prosthesis.

  In 1986 [Bernotas et al., 1986] used a discrete-time Hammerstein model to describe the input-output properties of electrically stimulated isometric muscle. The static block representing the recruitment characteristics as mentioned above is tested beforehand and separated from the muscle system. A second-order linear model for the muscle dynamics is then estimated by employing exponentially weighted recursive least squares methods. The efficiency of the model is confirmed using tests on cat soleus and plantaris muscles with

varying muscle length and stimulus periods, and also comparison between the closed-loop control of stimulated muscle and simulations using the discrete-time model. Two years later, this model was used successfully to design a feedback controller to accurately and robustly regulate the properties of the electrically stimulated muscle in [Chizeck et al., 1988].

In [Durfee and MacLean, 1989], four methods are developed to estimate the IRC within the Hammerstein model. Of these, the ramp deconvolution method was introduced for the first time and gave superior performance compared to the others, as well as having a shorter computation time and higher accuracy. This was employed in the stroke rehabilitation project at Southampton to estimate the IRC by [Freeman et al., 2009b], and is used to provide baseline performance comparison with approaches developed in Chapter 3 of this thesis. The estimation of the IRC also improves the design of the controller, where an inverse recruitment map is included in the forward path to partially cancel the effects of the muscle recruitment curve.

Five years later, this Hammerstein model with a static nonlinearity, IRC, followed by the linear dynamics, describing the isometric nonlinear muscle dynamics as a whole, was used as the CE in a Hill-type structure to model the non-isometric muscles in [Durfee and Palmer, 1994], see Fig. 2.2. An identification method for estimating the parameters is given and tested experimentally. Experimental force data fitting and prediction results demonstrated that the model was able to predict the behavior of stimulated muscle with reasonable accuracy for a wide range of length, velocity, and activation inputs. The major weakness of the model is the absence of time-variation. This led [Durfee and Palmer, 1994] to suggest an approach combining off-line and online identification, where one or two of the most important parameters (such as a scaling factor accounting for the presence of fatigue) could be tracked online.

After this work the popularity of the Hammerstein structure in this application area declined. However, subsequent improvements in identification theory and experimental techniques led to further work on improving the deficiencies in this model. [Hunt et al., 1998] identified a Hammerstein model using experimental data from the plantarflexors of an intact human, and when compared with local linear models valid at different activation levels, the Hammerstein model was found to be less accurate due to the fact that the dynamics are not independent of activation levels. Subsequently, [Munih et al., 2000] further investigated variation within the IRC and the dynamic response of the Hammerstein model, based on tests from six groups of people. It was found that the IRC depends on many factors, including electrode placement and location, physiological properties of the muscle, its innervation and electrode properties and displays variation with time due to fatigue. Besides these similar factors, the dynamics also depend on stimulation intensity, which can be explained by Heinneman's principle [Heinneman and Olson, 1965].

Some of these imperfections are quite easily overcome by performing an identification tests before the treatment so that the dependencies due to the subject or sample tested and the experimental environments can be excluded. However, some are more difficult to

eliminate, such as fatigue or intensity-dependent dynamics. With case, the Hammerstein model still can satisfy the requirements. In order to further improve the accuracy, a slowly time-varying Hammerstein model may be assumed. Fatigue takes effect slowly and if the input stimulation could be allowed to gradually vary, the dynamics would also be slowly time-varying. Thus, the optimal solution would seem to be an online updating scheme for the parameters in the Hammerstein model, which is one of the major contributions of this thesis and is the subject of Chapter 4.

- Wiener structure

  A dynamic linear subsystem followed only by a static nonlinearity is commonly referred to as a Wiener system, see Fig. 2.3(b). The Wiener structure has also been proposed for biological nonlinear system modeling by [Hunter and Korenberg, 1986]. [Hunter, 1985] and [Hunter, 1986] found that for active frog tibialis anterior muscle fibers, the dynamic relation between muscle length and tension can be more accurately represented by a Wiener model rather than a Hammerstein model.

  In these experiments, the stimulation was fixed at the maximal level and the length of the muscle was varied using stochastic perturbations. Thus, it is to be expected that the first nonlinear block, IRC, would be unnecessary and another output nonlinearity, accounting for the force-length relationship, would appear. This explains why a Wiener model outperforms a Hammerstein model in this application and does not affect the priority of the Hammerstein model when the stimulation intensity is modulated and the muscle is under isometric conditions.

- Hammerstein-Wiener structure

  When the input and output nonlinearities are both present and a linear block is situated in the middle, the structure is called a Hammerstein-Wiener structure, see Fig. 2.3(c). In [Farahat and Herr, 2005] and [Schauer et al., 2005a], a Hammerstein-Wiener structure is used to model the dynamics of electrically stimulated muscle under non-isometric conditions. Another output nonlinearity $h(w,x,\dot{x})$ is added to represent the relationship between the force produced and the mechanical state of the muscle in order to incorporate the force-length and force-velocity relationship, see Fig. 2.4. Thus, the model can characterize the muscle dynamics under more general conditions. In [Farahat and Herr, 2005], a two-stage identification scheme is presented to initially estimate the first two blocks, which themselves comprise a Hammerstein structure, and then to estimate the output nonlinear block separately. Only simulation results are provided to demonstrate the efficiency of the proposed structure. In [Schauer et al., 2005a], the model is estimated online using an extended Kalman filter and tested in both simulation and with real experimental data.

  This model is actually complementary to the Hammerstein structure because, after identifying the isometric muscle dynamics as a Hammerstein model, another output nonlinearity can be easily estimated by applying separate non-isometric tests, as in [Farahat and Herr, 2005]. However, this model structure is too simple compared with the Hill-type model in Fig. 2.2 to adequately model muscles under non-isometric conditions, since

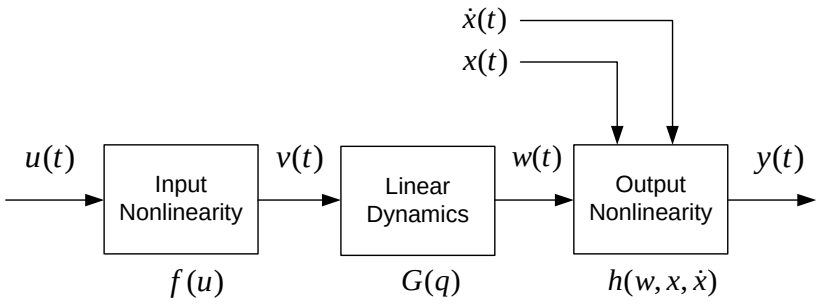


FIGURE 2.4: Hammerstein-Wiener structure under non-isometric conditions

experimental results show some discrepancies exist between the model output and the measured output in [Schauer et al., 2005a].

- Wiener-Hammerstein structure

  A Wiener-Hammerstein structure consists of a dynamic linear subsystem followed by a static nonlinearity, which is then followed by another dynamic linear subsystem, see Fig. 2.3(d). In [Bobet and Stein, 1998], a time-varying Wiener-Hammerstein model was proposed. The model consists of two first-order linear systems separated by a static nonlinearity and, in order to compensate for exponential drops in the output in the absence of a change in the input, a nonlinear time-varying equation with three unknown parameters was used in the last block. The model was validated using three cat soleus and plantaris muscles and can reproduce these responses accurately over a wide range of stimulation patterns.

  A large number of unknown parameters result in identification complexity becoming an issue for this model structure. An improvement over this model has been proposed recently by [Bai et al., 2009], which takes the form of a modified Wiener-Hammerstein structure, see Fig. 2.5.

  It only has four unknown parameters, a_1, a_2, b_1 and b_2, and the difference compared to a standard Wiener-Hammerstein structure is that a_1 is a time-varying switching function. Experimental data from the soleus muscles of individuals with spinal cord injury shows that the identified model performs comparably to the model in [Ding et al., 2002], which has been established as the most advanced and accurate one through comparison undertaken in [Bobet et al., 2005], but with less complexity.

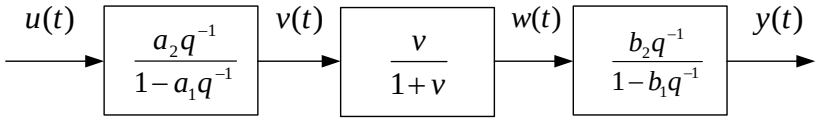


FIGURE 2.5: Modified Wiener-Hammerstein structure [Bai et al., 2009]

These two papers model the isometric muscle force in response to a variation in interpulse intervals. The output force can be modulated by varying either the number of active muscle fibers (recruitment) or the frequency of activation (temporal summation). The

modulation by temporal summation (stimulus period modulation, or, inversely, pulse frequency modulation) is achieved by varying the time interval between the start of successive pulses, in, for example, [Bobet and Stein, 1998], [Bobet et al., 2005] and [Ding et al., 2002] where the range of frequency is from $12.5Hz$ to $100Hz$. However, it is shown that the higher the stimulation frequency, the more quickly muscle fatigue becomes evident, especially over $50Hz$ in [Baker et al., 1993]. [Carroll et al., 1989] also found frequency modulation alone may not provide the ranges of torque required to achieve a variety of functional tasks. Due to the fact that the subjects in this work are stroke patients who are more easily fatigued than intact subjects, and the aim is to help them achieve functional tasks and not to investigate the muscle behaviors per se; frequency modulation is not a good choice.

Recruitment modulation involves varying the number of muscle fibers activated, by varying the amplitude (current or voltage amplitude) or the duration (width) of stimulus pulses. Pulse width modulation is preferred because it is easier to quantify and control than the stimulation pulse amplitude, provides a more consistent response across subjects, requires a smaller charge per stimulus pulse, and allows for greater selectivity of recruitment than amplitude modulation [Crago et al., 1980]. Therefore, pulse width modulation is employed here and it is open to debate whether the *Bai et al.* model will work at all under this modulation. In Chapter 3, the *Bai et al.* model and its identification algorithm will be implemented and compared with a Hammerstein model and the corresponding identification algorithm using the experimental data from isometric human muscles under pulse width modulated stimulation. The *Bai et al.* model shows very poor performance and it is speculated that the muscles exhibit different behaviors due to the type of stimulation used.

After reviewing most of the existing models in the literature, a Hill-type model is chosen by us to describe the whole non-isometric muscle behavior. Following the identification schemes introduced in [Durfee and Palmer, 1994], the passive and active force-length and force velocity relationships are not difficult to recover by simply fitting piecewise polynomials as has been implemented in [Freeman et al., 2009b]. The most burdensome part is to identify the Hammerstein structure in the Hill-type model because of the interactions existing between the linear and nonlinear block. The difficulties encountered in this task, together with the existing techniques available will be reviewed in Section 2.3.

Moreover, the Hammerstein structure, as a model of the nonlinear muscle dynamics under isometric conditions, is likely to be important in the design of systems for restoring force or motion using FES. This is because the activities that FES is intended to replace (reach to grasp tasks, tracking trajectories) are typically slow, controlled motions. For these activities, the effects of inertia, velocity, and series elasticity are likely to be small and the isometric behavior of muscle is likely to dominate.

Therefore, identification of isometric muscle dynamics will be investigated in this thesis and the Hammerstein structure is chosen as the underlying model. The existing tests used in the literature to identify the linear dynamics and IRC, the two blocks in the Hammerstein structure, will be reviewed in the next section.

2.2 Identification Tests

In an early study [Bernotas et al., 1986], Pseudo-Random Binary Sequences (PRBS) were used in identification tests and at each stimulus instant, the pulse width switched between two activation levels. The PRBS has widely been used in linear system identification and can ensure persistent excitation of the muscle dynamics so that subsequently many researchers made fair use of it to identify the linear dynamics of muscles, [Shue et al., 1995], [Chizeck et al., 1999].

A pilot study of tests and methods for estimating the IRC of electrically stimulated muscles was undertaken in [Durfee and MacLean, 1989]. This paper described three methods for estimating the IRC and made a brief mention of a fourth. They are now summarized.

- Steady-state step response method: This method uses a step input to activate the muscle at a fixed activation level for a period, n, and then averages the force over a further period, m, at the end of the step input. The IRC was drawn by cross-plotting the averaged forces against corresponding activation levels.
- Peak impulse response method (or twitch response method): A sequence of stimulus pulses at different stimulation levels were applied to the muscle in a random order and the peak of the muscle twitch responses were plotted against the stimulation level to obtain the IRC.
- Triangular ramp method: A triangular ramp test was applied to the muscle and the muscle ramp response was deconvolved by the linear dynamics of the muscle system. Then the IRC was estimated by cross-plotting the deconvolved signal against the input ramp signal.
- Stochastic iteration method: The method considered used a stochastic input and performed an iterative algorithm involving the muscle's response to derive an estimate of the IRC and the linear dynamics.

Later research into identification tests for electrically stimulated muscle is to a greater or lesser extent based on these methods. The Steady-State Step Response Method is the simplest and most common method in the literature, however, it is known to fatigue muscle so that its popularity has since diminished. The Peak Impulse Response Method is fast and non-fatiguing. In [Hunt et al., 1998], it was termed the Twitch Response Method, and was used to estimate the IRC and was also extended to estimate the linear dynamics as well. When compared to the PRBS test, the linear model identified from the twitch response was shown to be inferior

to the PRBS based model on a given set of validation data, primarily due to the low dynamic information content of the twitch response. Therefore, the PRBS is a better signal to identify the linear dynamics but it cannot excite the whole nonlinearity. A similar and superior choice are Pseudo-Random Multi-level Sequences (PRMS), used in [Schauer et al., 2005a]. The PRMS is a periodic, deterministic signal having an autocorrelation function similar to white noise. Meanwhile, the Triangular Ramp Method was demonstrated to be a promising alternative by [Durfee and MacLean, 1989]. This method has also been used in frequency modulation [Bobet et al., 2005] and has already been implemented in the stroke rehabilitation project at Southampton in [Freeman et al., 2009b]. The last approach was not implemented by the authors [Durfee and MacLean, 1989] but inspired a lot of later researchers. One of the methods developed in this thesis is also an iterative algorithm based on a single identification test to estimate both the IRC and the linear dynamics at the same time.

Another important paper which compares different identification tests is [Munih et al., 2000]. Three tests were applied: the Twitch response test, PRBS test and varied-frequency sinusoidal signals (which vary the frequency of sinusoidal signals at $[0.3, 0.5, 0.8, 1.2, 2, 3.2]Hz$). The IRC was estimated using a twitch response test and the linear dynamics were estimated using a twitch response test and a PRBS test. The authors also calculated prediction errors on the validation data for the models identified from the twitch response test and the PRBS test to demonstrate that the PRBS is a more information-rich test than the twitch response test. Also it was stated that use of varied-frequency sinusoidal signals led to large prediction errors but no tables or figures were presented to support the statement.

Having reviewed the literature, there is a distinct lack of discussion related to the design of tests for the identification of electrically stimulated muscle. Most identification algorithms give little attention to the choice of input signal or the criteria used to assess the resultant accuracy, instead picking an existing test input from the literature. A few of them compared several identification tests but almost no one has conducted a comprehensive validation analysis or even a cross-validation analysis. Therefore, a 'test design' section will be presented in Section 3.4.1 and some important issues arising in the design of tests for identifying electrically stimulated muscles will be discussed. Moreover, four candidate tests will be given and implemented experimentally and identification, validation and cross-validation results will be given to evaluate the performance of the tests.

2.3 Hammerstein Structure Identification

Finally, the last stage in building a model has been reached. The identification of Hammerstein structures will be reviewed here. The Hammerstein structure is depicted in Fig. 2.6. It consists of a memoryless nonlinear block followed by a linear dynamic system, and the difficulty is that the inner signal $w(t)$ is not measurable.

This structure was introduced in 1966 by [Narendra and Gallman, 1966]. A new technique for

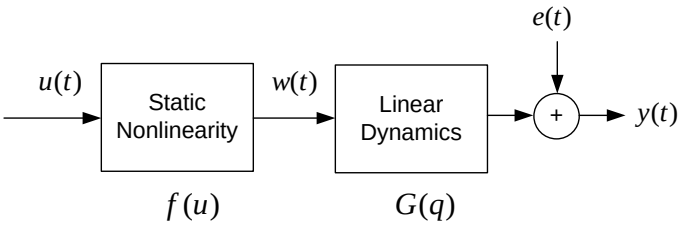


FIGURE 2.6: Hammerstein structure

the identification of nonlinear systems was suggested, based on the underlying model structure:

$$y(t) = \int_{-\infty}^{t} k(\tau) f(u(t-\tau)) d\tau$$

where $u(t)$ and $y(t)$ are the system input and output respectively, $k(\tau)$ represents the impulse response of the linear dynamics and $f(\cdot)$ is the static nonlinear function.

As an extension to the class of linear systems, the Hammerstein structure could be interpreted as a linear-in-parameter model, e.g., by the over-parametrization technique [Bai, 1998], so that many techniques from the well-researched area of linear system identification are possible. Meanwhile, the presence of the static nonlinearity allows a much wider range of dynamics to be described compared to those of purely linear models. Moreover, the structure has advantages over more general nonlinear models in terms of practical issues such as computational time and initial parameter selection, assuming that the real process fits into this particular form. Therefore, Hammerstein structures have received considerable attention and have been used in various areas to, for example, model chemical [Sung, 2002; Park et al., 2004], biological [Westwick and Kearney, 2001] and electrical [Wang et al., 2009] processes.

The term "identification" refers as the parameter estimation process which uses measured input-output data to forge the link between the mathematical model and the real world system and a comprehensive introduction of the literature and methods of system identification can be found in [Ljung, 1999; Soderstrom and Stoica, 1989]. In more detail, let $z(t)$ denote the item of data received at time t. This is in general a vector, composed of several different measurements, such as input and output signals $z(t) = [u(t) \;\; y(t)]^T$. Assuming that the data acquisition takes place in discrete time, as is normally the case, at time t, a sequence of measurements $z(1), z(2), \ldots, z(t)$ is available. Let us use a superscript to denote the whole data set:

$$z^t = \{z(1), z(2), \ldots, z(t)\}.$$

The purpose of identification is to estimate the model parameters θ based on the data z^t recorded during designed identification tests:

$$z^t \rightarrow \hat{\theta}.$$

Identification should be concerned with not only theoretical and computational difficulties, but with practical issues as well. Generally, it can be implemented in one of two procedures:

- Off-line identification or batch identification

  A batch of data is collected from the system and subsequently, is a separate procedure, this batch of data is used to construct a model. After collecting the data up to some time instant N, then a mapping from the data set z^N to the parameter space

  $$\hat{\theta} = F(N, z^N) \tag{2.9}$$

  is computed, where the function F may be implicitly defined (e.g., by the minimizing argument of some function).

  For batch identification, there is no need to predetermine the model structure beforehand, since identification can consider different model structures and model orders.

- Recursive identification

  However, in many cases it is necessary, or useful to have a model available on-line so that it is available for making decisions during the operation of the system, e.g., adaptive control. In such a situation, the model is updated at each time instant once the new data becomes available. In principle, $\hat{\theta}$ could still be a general function of previous data as in (2.9). However, in practice it is important that memory space and computation time do not increase with t. Thus, an "information state" $x(t)$ could be introduced, which has fixed dimensions and is a result of condensing the past data:

  $$x(t) = H\left(x(t-1), \hat{\theta}(t-1), z(t)\right). \tag{2.10}$$

  Then the updated estimate $\hat{\theta}$ is formed using current data, the previous estimate and the current state:

  $$\hat{\theta}(t) = F\left(\hat{\theta}(t-1), x(t), z(t)\right) \tag{2.11}$$

  where $F(\cdot)$ and $H(\cdot)$ are given functions.

  The major reason for the interest in such a procedure is as a useful tool in adaptive control, adaptive filtering, adaptive prediction and adaptive signal-processing problems. Furthermore, it could be used to track variations in systems which are time-varying. However, this procedure, due to its condensing of data into an informative state of fixed dimension, does not yield the same accuracy as models identified using the batch procedure, with a few exceptions.

There exist a large number of research papers on the topic of Hammerstein model identification and a literature review for both batch algorithms and recursive algorithms will be carried out in the following sections.

2.3.1 Batch Identification

Batch Identification methods can be roughly divided into seven categories:

- Over-parametrization

  By expanding the mathematical representation of the Hammerstein structure, one will end up with an expression involving products of both linear and nonlinear parameters. This results in a non-convex optimization problem. To overcome this difficulty, an over-parametrization method was proposed by [Chang and Luus, 1971], where every cross-product of unknowns is replaced by a new independent parameter. Then all the new parameters are estimated by least squares methods and the nonlinear parameters are separated by minimizing the root mean-square error of the model output. In that method, a transfer function is used to represent the linear dynamics, but no filtered noise is considered, which is then included in [Hsia, 1976]. In this paper, a multi-stage least squares method was presented to estimate the nonlinear parameters, the linear transfer function parameters and the noise parameters, using simple least squares solutions applied step by step. In [Bai, 1998], the algorithm was expanded to identify systems containing an output nonlinearity, also called a Hammerstein-Wiener system, and a Singular Value Decomposition (SVD) was used to separate the parameters. Because the separated parameters were obtained by searching over the entire parameter space, whilst [Chang and Luus, 1971] and [Hsia, 1976] only searched a small subset, the author claims that global optimality is achieved with no noise or white noise while [Chang and Luus, 1971] and [Hsia, 1976] are only locally optimal.

- Subspace

  Subspace identification methods (SIM) estimate the state-space model or the extended observability matrix directly from the input and output data. There are several influential methods: Canonical Variate Analysis (CVA), Multivariable Output Error State-space (MOESP) and Numerical Subspace State-Space System IDentification (N4SID). A unified statistical framework is given in [Shi and MacGregor, 2001], which consists of three steps: estimating the predictable subspace for multiple future steps, then extracting state variables from this subspace and finally fitting the estimated states to a state-space model. Since it requires a modest computational load without the need of iterative optimization procedures, this method has received lots of attention.

  Previously, SIM only applied to Linear Time-Invariant (LTI) systems and it was for perhaps the first time that Verhaegen and Westwick in 1996 extended it to Hammerstein systems in [Verhaegen and Westwick, 1996]. Here the MOESP method was applied to two types of Hammerstein identification problems: polynomial parametric nonlinearity and only limited a prior knowledge of the nonlinearity. Later on, this method was expanded to all the families of SIM including the CVA, the MOESP and the N4SID by [Gomez and Bayens, 2005]. Meanwhile, [Goethals et al., 2005] used the least squares support vector machines regression to extend the linear N4SID subspace method to the Hammerstein structure. This method usually consists of two steps: the first step is to apply a certain subspace-based identification algorithm to the over-parametric state-space representation of the Hammerstein structure, where a similar approach is exploited to that

described above, replacing each cross-product term by a new parameter; and then SVD is used to recover the original parameters. The differences from the former category are the state-space model and the corresponding subspace method.

The key scheme in both of these two methods is the over-parametrization approach, which treats every cross-product term as a new parameter, resulting in linear-in-parameter difference equations or state-space models. Thus, these two methods suffer from the same problems: The first one is the so-called dimension problem. When the dimensions of the parameter vectors of the nonlinearity and linear subsystem increase, the number of extra unknown parameters to be estimated in the over-parametric linear system increases very quickly. As a result, the performance may decrease sharply. The second problem relates to an implicit rank constraint. In the above two approaches, the least squares method or the SIM is applied directly to a generalized difference equation or state-space model. However, the model has a specialty that is often ignored, that is, the newly defined parameter vector or matrix should have a rank constraint. Consequently, the performance is not quite satisfactory, for example, in [Bai, 2006], the over-parametrization method shows sensitivity to the presence of noise, when compared with the iterative and numerical methods at a high noise level.

Besides, due to its linear-in-parameter property, these two approaches both have recursive counterparts [Boutayeb and Darouach, 1995; Boutayeb et al., 1996] and [Bako et al., 2009], which will be reviewed in the next section.

- Stochastic

  Stochastic, relay feedback and blind methods are quite similar in the way they deal with the difficulty of handling coupled linear and nonlinear parameters. All of them separate the linear dynamics from the whole system by employing particular inputs with certain properties. For the stochastic method, the inputs are assumed to consist of white noise so that the internal signals $w(t)$ and the outputs $y(t)$ are also white. Thus, the linear dynamics can be easily estimated by the correlation method. Once the linear system is known, estimation of the static nonlinearity is straightforward. A kernel regression estimation is presented in [Greblicki and Pawlak, 1986] and the orthogonal series estimation is employed in [Pawlak, 1991]. Although the latter one improves the rate of convergence, they are still sensitive to irregularities in the input probability density. In [Greblicki, 1996], algorithms with convergence rates independent of the shape of the input signal density have been proposed using ordered observations in the cost of computational effort caused by ordering. Later on, the stochastic method has been implemented recursively by Greblicki [Greblicki, 2002] and Chen [Chen, 2004], which will be discussed in the next section.

- Relay feedback

  The essence of this method is also to remove the effect of the nonlinear component. [Sung, 2002] suggested a special test signal, consisting of a binary signal and a multi-step signal. The first part is used to exclude the effects of the nonlinearity so that all techniques from

linear system identification are valid to subsequently recover the linear subsystem. Then the purpose of the multi-level signal is to excite the whole nonlinearity, and the input-output data together with the estimated linear dynamics are used to identify the static nonlinear component. [Bai, 2004] shared the same idea but separated it into two steps, the first of which uses a pseudo-random binary sequence. In the same year, [Park et al., 2004] also considered the two-step approach but the difference is that a biased relay feedback test is applied, which plays the same role in the whole approach.

- Blind

  This approach exploits piece-wise constant input signals in order to decouple the linear system. In [Sun et al., 1999], this blind approach is borrowed from identification of linear systems for use in Hammerstein structure identification for the first time. An input-holding scheme is used, which means the test input signal is held during a multiple of the output sampling interval and after the estimation of the numerator and denominator of the linear dynamics, a deconvolution method is considered to recover the intermediate signal without restriction to minimum phase systems. In [Bai and Fu, 2002], the approach is applied to a wide range of inputs with the employment of a fast sampling technique, and the inner signal is estimated by taking a direct inverse or by exploiting the Bezout identity for minimum and non-minimum phase linear dynamics. [Wang et al., 2007] considers the noise-corrupted case and the estimation of the process orders and the time delay. An even more important improvement over the previous two is made in [Wang et al., 2007] where a different method is used to estimate the inner signal, leading to reduced noise effects. However, the output noise is propagated into the estimates in [Bai and Fu, 2002], and the process in [Sun et al., 1999] is unnecessary complicated. Furthermore, in order to remove the error propagation, [Wang et al., 2009] omits the step of estimating the numerator, and only identifies the denominator of the linear dynamics and uses a subspace direct equalization method to estimate the unmeasurable inner signal. The results in [Wang et al., 2009] show significant improvement compared to the former, especially under noisy situations.

These three approaches do not require prior knowledge or an explicit parameterization of the nonlinearity so that they significantly relax restrictions imposed on the identified system. This is important since the nonlinearity could have many possible structures or could be hard to represent by parametric models.

However, there are assumptions on the input properties, which make implementation infeasible. The stochastic approach requires white noise inputs, the relay feedback approach needs the input to be composed of both binary and multi-step signals, and the blind method uses a piecewise constant input. By imposing these assumptions on the input, the linear dynamics can be identified in a decoupled manner, and then the nonlinearity can be identified in a nonparametric way. Unfortunately because the identification has been separated into two stages, the error generated in the first approximation will propagate to the second, where it is usually ignored, which will lead to a significant discrepancy from the true model, especially under noisy situations. Thus, it

is neither wise nor practical to implement these approaches alone.

- Separable Least Squares

  The idea behind the Separable Least Squares (SLS) approach is to write one set of variables as a function of the other set based on the first-order necessary and sufficient conditions. Thus, the dimension of the optimization space is reduced. In [Westwick and Kearney, 2001], a technique based on SLS optimization was developed for application to the Hammerstein structure. Firstly, the output is represented by a convolution sum of the impulse response of the linear dynamics and a polynomial of the corresponding input with unknown polynomial coefficients. Since the output is linear in its impulse response, named linear parameters θ_l, the optimal value of θ_l corresponding to any choice of polynomial coefficients, θ_n, can be found in closed form by solving a linear regression. Thus, θ_n is a function of θ_l and also the modeled output and the minimization criterion. Therefore, an iterative algorithm is used to optimize θ_n, involving employing Levenberg-Marquardt algorithm to compute the updating step, and θ_l being updated by linear regression. Later on, [Dempsey and Westwick, 2004] considered the use of cubic splines instead of a polynomial to represent the static nonlinearity. Both of them have been applied to a biological system (stretch reflex electromyogram) and the model with the cubic spline nonlinearity was seen to provide more accurate predictions than the polynomial based model for the experimental data. Also this method is found to be particularly useful for a class of non-smooth nonlinearities [Bai, 2002b].

- Iterative

  The idea behind the iterative method, introduced by Narendra and Gallman [Narendra and Gallman, 1966], is the alternative estimation of parameters. Although there are some variations, the parameter set is usually divided into two subsets: a linear and nonlinear part. One finds the optimal values for the first set while the second set is fixed. Then, the two sets are switched in order to find the optimal value for the second set whilst the first one is fixed. If convergent, the iterative algorithm converges rapidly and is very efficient. However, guaranteeing convergence is a problem. It was shown in [Stoica, 1981] by means of a well-constructed Finite Impulse Response (FIR) example that in general the iterative algorithm does not converge and in fact, the parameter estimates may diverge and become unbounded. Since then, the convergence of this iterative algorithm has been an open problem. [Bai and Li, 2004] carried out a detailed study on the convergence properties of the iterative algorithm and derived some global convergence properties for the normalized iterative algorithm applied to the identification of Hammerstein systems with smooth nonlinearity and a FIR linear component. The results are extended to Hammerstein systems with an Infinite Impulse Response (IIR) linear part and nonsmooth nonlinearities, together with a FIR linear part in [Liu and Bai, 2007].

These two iterative methods do not require a particular form of input, which would restrict their practical use, and are simple and efficient in practice. For this reason, they are the only two cat-

egories amongst the seven considered to be applied to real biological systems, e.g., [Westwick and Kearney, 2001] and [Hunter and Korenberg, 1986]. Although SLS and the iterative technique suffer from the problem of global convergence, in applications, they generally converge very fast and the converged values, even when they can not be proved to constitute a global optimum, are quite accurate. Therefore, these two techniques are ripe candidates for the application considered in this thesis.

With regards to the convergence problem, one possible solution is to choose a good initial estimate. For SLS, starting from initial estimates of the nonlinear parameters, [Dempsey and Westwick, 2004] suggested that if the initial estimates are close enough to the global optimum, the model does not converge to a suboptimal local minimum and the over-parameterization method was recommended to complete this task. Iterative algorithms can begin with either the initial values for the linear part or the nonlinear part. [Bai and Li, 2004] proposed a certain set of conditions for the initial values so as to guarantee convergence and, moreover, guarantee rapid convergence. Therefore, generating a good initial value deserves special consideration in this thesis in order to remedy the defects of these two approaches.

2.3.2 Recursive Identification

However, as stated above, the Hammerstein structure is not sufficiently complex to model muscle dynamics, especially in the application considered. When applied to stroke rehabilitation, stimulation must be applied during intensive, goal orientated practice tasks in order to maximize improvement in motor control [Schmidt and Lee, 1998]. In clinical trials this translates to sustained application of stimulation during each treatment session of between 30 minutes and 1 hour duration [de Kroon et al., 2005]. In this case, slowly time-varying properties of the muscle system arise due to fatigue, changing physiological conditions or spasticity [Graham et al., 2006]. Therefore, a slowly time-varying Hammerstein system is assumed and online identification, also termed recursive identification will be considered. Only a few of the existing identification methods are recursive, and can be divided into three categories:

- The first category is the recently developed recursive subspace identification method by [Bako et al., 2009], where the nonlinear function is first recursively estimated by over-parameterization and component-wise Least Squares Supporting Vector Machine (LS-SVM). This is followed by estimation of the Markov parameters by recursive least squares, and then a propagator-based method is used to recursively estimate system state-space model matrices from these parameters. This procedure does not have sparsity due to the LS-SVM model, and the resulting computational load makes it unsuitable for real-time implementation.
- The second category comprises stochastic approximation [Greblicki, 2002; Chen, 2004] where a stochastic approximation algorithm with expanding truncations is developed for recursive identification of Hammerstein systems. Two major issues with this method are

the rather slow rates of convergence, and the lack of information on how to select the optional parameters in the algorithm when applied to problems from different areas.

- The third category is recursive least squares or extended recursive least squares. The Recursive Least Squares (RLS) algorithm is a well known method for recursive identification of linear-in-parameter models and if the data is generated by correlated noise, the parameters describing the model of the correlation can be estimated by Extended Recursive Least Squares (ERLS). Here, a typical way to use these two algorithms is to treat each of the cross-product terms in the Hammerstein system equations as an unknown parameter. This procedure, which results in an increased number of unknowns, is usually referred to as the over-parametrization method [Bai, 1998] and [Chang and Luus, 1971]. Following this step, the RLS or ERLS method can be applied [Boutayeb and Darouach, 1995; Boutayeb et al., 1996; Zhao and Chen, 2009].

The limitations of current algorithms are stated next and are used to justify some of the critical choices necessary for this work to progress

- The first two categories have only been applied in simulation and the stochastic approximation has not considered time-varying linear dynamics. This, together with the drawbacks described above, is the reason for not considering them further for the intended application. The third category is the most promising as it has already been applied to electrically stimulated muscle in [Chia et al., 1991] and [Ponikvar and Munih, 2001].

- Most of the test signals used comprise random noise in order to guarantee persistent excitation, even when applied to the human muscle. For example, [Ponikvar and Munih, 2001] employs pseudorandom binary sequences. However, this type of signal, which excites the motor units abruptly, will cause patient discomfort and may elicit an involuntary response, as reported in [Baker et al., 1993]. In [Chia et al., 1991] a test consisting of 25 pulses is used, each of which is of 1 second duration in the form of a noisy triangular wave. This test meets our requirements but is too short to exhibit time-varying properties.

- The most relevant previous work is [Chia et al., 1991] where the system considered had linear constraints and a RLS technique was developed for constrained systems. However, the results given do not establish that the constraints are achieved. Moreover, when showing the prediction error, the posteriori estimated output without constraints was observed to outperform the one with constraints. Thus, there is clearly more work to be done in this area, and hence the idea of adding constraints to RLS, leading to increased computational load, is well worth consideration..

Overall, RLS is the most promising technique for application to electrically stimulated muscle, but the problem of consistent estimation must be resolved [Chen, 2004; Chia et al., 1991]. This algorithm is implemented in Section 4.2.1 but due to its unsatisfactory performance, especially for noisy measurements, another recursive algorithm for Hammerstein systems is developed in

Section 4.2.2. Moreover, a long-period test signal needs to be designed for our application, which is persistently exciting and also gradually recruits the motor units. This problem is addressed in Section 4.4.

2.4 FES Control Schemes

There exists a wide variety of control schemes for electrically stimulated muscle. Many address the case of paralysed muscle and are applied to subjects with spinal injury. Others are intended for use in rehabilitation where there is evidence [Rushton, 2003] that functional recovery is enhanced when stimulation is applied coincidentally with a patient's voluntary intention whilst performing a task. In the latter case, a control scheme is required to precisely provide stimulation which allows the desired movement to be realised. A brief review and evaluation of existing control schemes will be given next.

Open-loop methods have found favor in clinical use due to their simplicity. For example, [Davoodi et al., 2002] used an open-loop method to control lower limb movement of paraplegic subjects in a rowing exercise. The user voluntarily performed the upper body movement, while manual and automatic control schemes were investigated. In the former category, the user pressed the control button, and in the latter, the controller automatically applied the stimulation depending on a set of rules governed by the instantaneous position of the set and the handle. However, open-loop methods have not been able to provide the high level of performance which is necessary to fully promote the required association between the subjects intended movement, and the action of the applied electrical stimulation in realizing it.

In a laboratory, as opposed to clinical setting, a wide variety of model-based controllers for electrical stimulation have been implemented which may be able to produce such accuracy. These include the use of multichannel Proportional-Integral-Derivative (PID) control of the wrist [Watanabe et al., 2003], and H_∞ ([Hunt et al., 2001]), optimal control ([Hunt et al., 1997]) and fuzzy logic control ([Davoodi and Andrews, 1998]) for paraplegic standing, and sliding mode control of shank movement [Jezernik et al., 2004], and data-driven control ([Previdi et al., 2005]) for control of the paraplegic knee joint. Few such model-based schemes have been applied to upper limb movement, exceptions principally comprising the use of neural networks for paraplegic arm movements (see for example [Lan et al., 1994]; [Tresadern et al., 2006]). [Lan et al., 1994] used an Artificial Neural Network (ANN) to control single joint human arm movements in paralysed subjects. The ANN is trained over a range of movements to learn and store the optimal patterns of muscle stimulation and can reproduce range of scaled optimal movements well. Feedforward, recurrent feedback and time delay topologies of ANN are considered but the ultimate structure is decided by the training process in order to provide good predictions for novel movements. Thus, ANN schemes are often not suited to rehabilitation purposes because retraining the network for each subject is slow, and online adaptation is not possible. Stability is another issue due to their black-box structure.

Another factor is that any suitable control method selected for the stimulation must also operate in the presence of voluntary effort supplied by the patient. A simple method of achieving this is for the controller to directly use electromyographic (EMG) or myoelectric activity of the muscle being stimulated (see for example [Thorsen et al., 2001]). However, model-based control methods have not yet incorporated this information since it does not directly relate to the force or torque generated by the muscle, and because the signal is often either weak and unreliable or that the artefact produced by the stimulation signal corrupts the natural EMG signal (although in this case blanking techniques may be applied).

Iterative Learning Control (ILC) is one of the very few model-based approaches that has been employed clinically and moreover, the repetitive nature of rehabilitation exercises makes ILC particularly suitable for this application. ILC is specifically developed for the systems needed to repeat a finite duration task over and over again. The novel feature is to use information from previous trials to update the control signal to be used on the current trial. The most basic control problem here is to design the control input in such a way that the system learns, by iteration from trial-to-trial, to produce the required output or reference signal whilst ensuring that the control signal does not exceed the limits imposed by the actuators used. Comprehensive reviews of the history and categorization of ILC can be found in [Ahn et al., 2007] and [Bristow et al., 2006]. This technique has previously been applied to control the electrically stimulated human upper limb required to repeatedly perform a given task in [Dou et al., 1999]. The proposed control strategy consists of a PD feedback controller and a high-order feedforward ILC controller. The simulation and experimental results are presented, although a high level of performance has not been achieved. In the current project, in order to control the FES applied to the upper extremity for rehabilitation of stroke patients, [Freeman et al., 2009a] considered two ILC schemes in the ILC feedforward controller: Phase-lead ILC and Gradient descent ILC. With the addition of a linearizing controller and a PD feedback controller, these two ILC schemes showed superior tracking performance in the study of 18 unimpaired subjects compared with the alternative control methods in the literature. Furthermore, it possesses the advantages of reduced identification experiments and simplicity of tuning. Later on, a more advanced ILC algorithm is applied in the same application [Davies et al., 2008]: Newton method based ILC, which is a nonlinear ILC approach which inherits the fast convergence rate of the Newton method. The algorithm exhibits robustness together with a high level of performance. The results from these two papers confirm the feasibility and efficacy of this approach and the following study, involving the treatment of stroke patients, provides statistical support using a number of outcome impairment measures [Hughes et al., 2009]. However, the experimental data confirms that the model of stimulated muscle adopted is not as accurately identified as the remaining components of the arm, which results in relatively low ILC learning gains used throughout the clinical trials. Such difficulties in obtaining reliable and accurate muscle models are also the reason for the lack of model-based methods in clinical application. Thus, the need for improved modeling of the muscle system is particularly important, providing a strong motivation of the research in this thesis.

Moreover, the underlying musculoskeletal system is highly sensitive to physiological conditions (such as skin impedance, temperature and moisture) and electrode placement, as well as time-varying effects such as spasticity and fatigue [Baker et al., 1993], which means the plant to be controlled is highly nonlinear and also time-varying. Thus, the need for online identification and associated adaptive control schemes is highlighted. Adaptive control has been applied to several existing control schemes for FES systems. For example, [Ferrarin et al., 2001] applied an adaptive control scheme for FES-induced single joint movements, which resulted in additional improvement by accounting for the time-variant effects of the system. Unfortunately, only one researcher has implemented adaptive ILC for electrically stimulated muscle systems [Wu et al., 2000]. Wu *et al* used a PID feedback controller together with a feedforward adaptive ILC controller, where the ILC gains adapt in response to the previous performance of the controller. The controller is not model-based and does not require prior knowledge of the system dynamics, and the problem addressed is the stability of the PID controller. By adding the adaptive ILC scheme, the sharp oscillation caused by the PID controller alone is overcome, and the overall scheme is demonstrated by clinical experiments involving motion trajectory tracking of the elbow joint and wrist joint. Unfortunately, exactly how the leaning gains are calculated is not given in detail.

From the above discussion, there do not exist ILC schemes in which the model is updated using control input and corresponding muscle output data, in order to allow full adaptation of the ILC scheme in response to changing dynamics. In Chapter 5, the implementation of a trial-dependent adaptive gradient descent ILC and adaptive gradient descent ILC will be developed in detail, together with the non-adaptive gradient descent ILC and Newton method based ILC.

Chapter 3

Identification

For the reasons provided in Chapter 2, a Hammerstein structure has been selected to model the electrically stimulated muscle under isometric conditions. Therefore, in this Chapter, identification of the parameters of the Hammerstein structure is investigated.

3.1 Problem Statement

A discrete-time Hammerstein model structure will be considered and is shown in Fig. 3.1. The stimulation input u is first scaled by the static nonlinear function f and then passed to a linear time-invariant system described by a transfer function $G(q)$. The noise v is zero mean and white and $H(q)$ is the noise model. The internal signal w is not measurable, which is the reason that the identification of such a model structure presents difficulties.

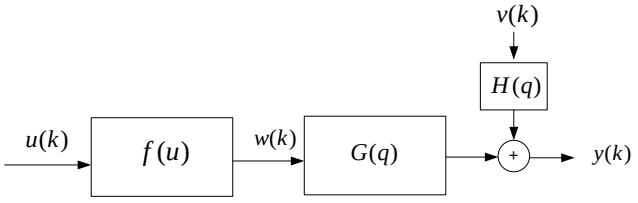


FIGURE 3.1: Discrete-time Hammerstein Model Structure

The linear system is represented by the transfer function

$$G(q) = \frac{B(q)}{A(q)} = \frac{b_0 q^{-d} + b_1 q^{-(d+1)} + \cdots + b_n q^{-(n+d)}}{1 + a_1 q^{-1} + \cdots + a_l q^{-l}} \tag{3.1}$$

where q^{-1} is the delay operator and n, l and d are the number of zeros, poles and the time delay, respectively. The parameters n, l and d are assumed to be known.

The nonlinear function $f(u)$ is represented by a cubic spline function, defined as

$$f(u) = \sum_{i=1}^{m-2} \beta_i |u - u_{i+1}|^3 + \beta_{m-1} + \beta_m u + \beta_{m+1} u^2 + \beta_{m+2} u^3 \tag{3.2}$$

where $u_{\min} = u_1 < u_2 < u_3 < \cdots < u_m = u_{\max}$ are the spline knots. Although this is a non-standard implementation of cubic splines, it has been used by several authors, see, for example, [Zhu, 2000]. It is easy to verify that the first and second derivatives of the function are continuous and hence the function possesses similar properties to the standard cubic spline representation, whilst benefitting from increased simplicity.

$$\theta_n = \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_{m+2} \end{bmatrix}^T \tag{3.3}$$

are the parameters of the nonlinear block and

$$\theta_l = \begin{bmatrix} \theta_a \\ \theta_b \end{bmatrix} = \begin{bmatrix} a_1 & \cdots & a_l & b_0 & b_1 & \cdots & b_n \end{bmatrix}^T \tag{3.4}$$

are the parameters of the linear block.

The noise model $H(q)$ is not specified here. Two kinds of noise models are plotted in Fig. 3.2.

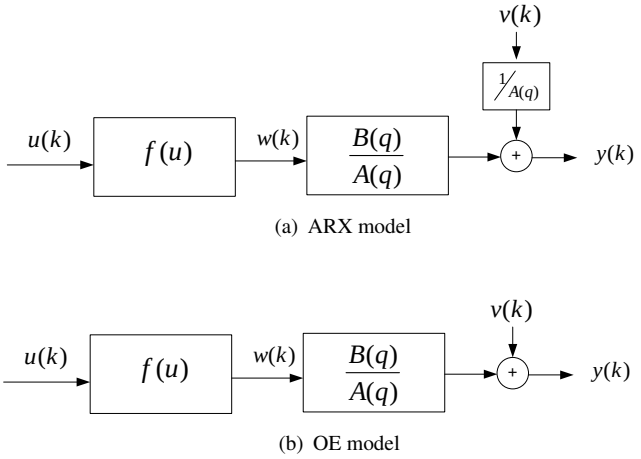


FIGURE 3.2: Two Discrete-time Hammerstein Model Structures with Different Noise Models

The first choice is an Auto Regressive eXternal (ARX) model [Ljung, 1999], in Fig. 3.2(a), where the noise filter, $H = 1/A(q)$, is coupled to the linear component of the plant model. In Section 3.2 and 3.3, Algorithms 1 and 2 are based on this model structure, where

$$y(k) = \frac{B(q)}{A(q)} f(u(k)) + \frac{1}{A(q)} v(k) \tag{3.5}$$

However, from a physical point of view, it is perhaps not the most natural form because the white noise, e.g. measurement noise, is assumed to pass through the denominator dynamics of the linear block before being added to the output. Therefore, Algorithms 2 is extended to identify the

Hammerstein structure with another linear model, an Output-Error (OE) model [Ljung, 1999], in Fig. 3.2(b). In this case, the noise model is $H = 1$ and

$$y(k) = \frac{B(q)}{A(q)} f(u(k)) + v(k) \tag{3.6}$$

The details are given in Section 3.3 and also the resulting Algorithm 3.

Therefore, the identification task is to estimate the parameter vector

$$\theta = \begin{bmatrix} \theta_l \\ \theta_n \end{bmatrix} \tag{3.7}$$

that minimizes the cost function

$$||v||_2^2 = \sum_{k=1}^{N} v^2(k) \tag{3.8}$$

from collected input output data

$$[u(1), y(1), u(2), y(2), \ldots\ldots, u(N), y(N)] \tag{3.9}$$

3.2 Two-stage Algorithm

The two-stage algorithm establishes the groundwork for the research that follows. The idea of this algorithm is to separate the linear and nonlinear parameters and alternately identify or optimize them. Whilst the idea is not new, the proposal scheme uses a novel projection approach to update the parameters. This is adopted continuously in a similar way to that of the later algorithms.

In order to obtain accurate estimates for each patient during treatment, the identification tests need to be performed just before each treatment session commences. A triangular ramp test is applied to the muscle and a two-stage algorithm is then performed on the data. This shows superior performance when compared with the Ramp Deconvolution (RD) method and Separable Least Squares (SLS) Optimization Algorithm on a stimulated muscle system. A preliminary step is also designed, whose purpose is to obtain an initial estimate of the linear parameters, which is then used in the later identification procedure.

In order to concisely introduce the two-stage algorithm, (l, n, d) is assumed to be $(2, 1, 1)$, which has been assumed in a related problem by [Hunt et al., 1998]. Thus, the model formula (3.5) becomes

$$y(k) = \frac{b_0 q^{-1} + b_1 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} f(u(k)) + \frac{1}{1 + a_1 q^{-1} + a_2 q^{-2}} v(k) \tag{3.10}$$

and the parameter vector (3.7) is given by a more simple expression

$$\theta = [a_1, a_2, b_0, b_1, \beta_1, \beta_2, \ldots, \beta_{m+2}]^T. \tag{3.11}$$

3.2.1 Preliminary Step: Pseudo-Random Binary Sequences Test

The purpose of the preliminary step is to identify the linear parameters and this step is performed when the stroke patients first arrive for their treatment session. Because the nonlinear muscle contraction mechanism of a stroke patient is impaired and varies, see [Mccrea et al., 2003], and may also change between trials, the identified linear parameters only can be used as an initial estimate. However, this step is also very useful. The reason is that in order to avoid patient fatigue, the before-treatment test should be as short as possible, which means the data may not be very rich. Under such circumstances, the preliminary step can provide a fairly good initial guess for a given patient, even when not performed on the same day. This improves the identification results with benefit to the computational time consumed.

In the preliminary step, a Pseudo-Random Binary Signal (PRBS) test is used. This signal changes between two levels so that it can exempt the nonlinearity from the isometric muscle system and the linear dynamics alone can be identified as expected. Now, the identification procedure will be explained. First, the two levels of PRBS inputs are assumed to be 0 and $+c$ (where $c > 0$) because the input pulse-width can not be negative. Also it is known that $f(0) = 0$ and hence c is chosen such that $f(+c) = r$ (where $r \neq 0$) with the value of r large enough compared to the possible noise level. Hence for all k

$$w(k) = f(u(k)) = \alpha u(k) \tag{3.12}$$

where $\alpha = r/c$, and the output of the Hammerstein structure with PRBS inputs is

$$\begin{aligned} y(k) &= \frac{\alpha b_0 q^{-1} + \alpha b_1 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} f(u(k)) + \frac{1}{1 + a_1 q^{-1} + a_2 q^{-2}} v(k) \\ &= \frac{\bar{b}_0 q^{-1} + \bar{b}_1 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} f(u(k)) + \frac{1}{1 + a_1 q^{-1} + a_2 q^{-2}} v(k) \end{aligned} \tag{3.13}$$

or, with $\bar{b}_j = \alpha b_j \quad j = 0, 1$

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) + \bar{b}_0 u(k-1) + \bar{b}_1 u(k-2) + v(k) \tag{3.14}$$

and in matrix form

$$Y = \Phi \theta_l + V \tag{3.15}$$

where

$$Y = [y(3)\ y(4)\ \cdots\ y(N)]^T \tag{3.16}$$

$$V = [v(3)\ v(4)\ \cdots\ v(N)]^T \tag{3.17}$$

and

$$\Phi = \begin{bmatrix} -y(2) & -y(1) & u(2) & u(1) \\ -y(3) & -y(2) & u(3) & u(2) \\ \vdots & \vdots & \vdots & \vdots \\ -y(N-1) & -y(N-2) & u(N-1) & u(N-2) \end{bmatrix} \tag{3.18}$$

Hence the initial estimate, $\hat{\theta}_l$ can be obtained using the least squares method as

$$\hat{\theta}_l = [\hat{a}_1, \hat{a}_2, \hat{\hat{b}}_0, \hat{\hat{b}}_1]^T == (\Phi^T \Phi)^{-1} \Phi^T Y \tag{3.19}$$

Using (3.10), the gains of $f(u)$ and $B(q)$ are not unique. In order for these to be uniquely identifiable, the gains are normalized, e.g., set $\hat{\hat{b}}_0 = 1$.

3.2.2 Two-stage Algorithm: Triangular Ramp Test

In order to identify the nonlinear part, a triangular ramp test is applied to the muscle, which is rich enough to excite the whole nonlinearity.

This algorithm consists of two stages: firstly, identify the nonlinear part by using the linear parameters estimated from the preliminary step; Secondly, optimize the linear parameters using the nonlinear parameters estimated in stage one.

1. Identify nonlinear parameters:

   In stage one, the estimated values of the linear parameter vector $[\hat{a}_1, \hat{a}_2, \hat{b}_0, \hat{b}_1]$ from the preliminary step are used to substitute $[a_1, a_2, b_0, b_1]$ in (3.10) and extend the equation.

   $$y(k) = -\hat{a}_1 y(k-1) - \hat{a}_2 y(k-2) + \hat{b}_0 f(u(k-1), \theta_n) + \hat{b}_1 f(u(k-2), \theta_n) + v(k) \tag{3.20}$$

   Sequences of input and output signals are known and can be moved to the right hand side and the unknown items to the left hand side to produce

   $$y(k) + \hat{a}_1 y(k-1) + \hat{a}_2 y(k-2) = \hat{b}_0 f(u(k-1), \theta_n) + \hat{b}_1 f(u(k-2), \theta_n) + v(k) \tag{3.21}$$

   From (3.2), it is known that $f(u)$ is just linear combination of β_i if u is known so that after

substituting (3.2) into (3.21)

$$\begin{aligned} y(k)+\hat{a}_1 y(k-1)+\hat{a}_2 y(k-2) = \sum_{i=1}^{m-2} \beta_i \underbrace{(\hat{b}_0|u(k-1)-u_{i+1}|^3+\hat{b}_1|u(k-2)-u_{i+1}|^3)}_{f_i(u(k),\hat{\theta}_l)} \\ +\beta_{m-1}\underbrace{(\hat{b}_0+\hat{b}_1)}_{f_{m-1}(u(k),\hat{\theta}_l)} \\ +\beta_m \underbrace{(\hat{b}_0 u(k-1)+\hat{b}_1 u(k-2))}_{f_m(u(k),\hat{\theta}_l)} \\ +\beta_{m+1}\underbrace{(\hat{b}_0 u(k-1)^2+\hat{b}_1 u(k-2)^2)}_{f_{m+1}(u(k),\hat{\theta}_l)} \\ +\beta_{m+2}\underbrace{(\hat{b}_0 u(k-1)^3+\hat{b}_1 u(k-2)^3)}_{f_{m+2}(u(k),\hat{\theta}_l)} \end{aligned} \tag{3.22}$$

and using the definition of the nonlinear parameter vector

$$\theta_n = [\beta_1, \beta_2 \ldots, \beta_{m+2}]^T \tag{3.23}$$

it becomes

$$\underbrace{\begin{bmatrix} y(3)+\hat{a}_1 y(2)+\hat{a}_2 y(1) \\ y(4)+\hat{a}_1 y(3)+\hat{a}_2 y(2) \\ \vdots \\ y(N)+\hat{a}_1 y(N-1)+\hat{a}_2 y(N-2) \end{bmatrix}}_{Y_n(y,\hat{\theta}_l)} = \underbrace{\begin{bmatrix} f_1(u(3),\hat{\theta}_l) & \cdots & f_{m+2}(u(3),\hat{\theta}_l) \\ f_1(u(4),\hat{\theta}_l) & \cdots & f_{m+2}(u(4),\hat{\theta}_l) \\ \vdots & & \vdots \\ f_1(u(N),\hat{\theta}_l) & \cdots & f_{m+2}(u(N),\hat{\theta}_l) \end{bmatrix}}_{\Phi_n(u,\hat{\theta}_l)} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{m+2} \end{bmatrix} \tag{3.24}$$

Thus, (3.24) can be solved by the least square method

$$\hat{\theta}_n = [\hat{\beta}_1, \ldots, \hat{\beta}_{m+2}]^T = (\Phi_n(u,\hat{\theta}_l)^T \Phi_n(u,\hat{\theta}_l))^{-1} \Phi_n(u,\hat{\theta}_l)^T Y_n(y,\hat{\theta}_l) \tag{3.25}$$

2. Optimize linear parameters:

After estimating nonlinear parameter vector $\hat{\theta}_n$, the linear parameters can be optimized by linear regression in stage two. The model formula is rewritten as

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) + b_0 f(u(k-1),\hat{\theta}_n) + b_1 f(u(k-2),\hat{\theta}_n) + v(k) \tag{3.26}$$

which can be written in matrix form as

$$\underbrace{\begin{bmatrix} y(3) \\ y(4) \\ \vdots \\ y(N) \end{bmatrix}}_{Y_l(y)} = \underbrace{\begin{bmatrix} -y(2) & -y(1) & f(u(2),\hat{\theta}_n) & f(u(1),\hat{\theta}_n) \\ -y(3) & -y(2) & f(u(3),\hat{\theta}_n) & f(u(2),\hat{\theta}_n) \\ \vdots & \vdots & \vdots & \vdots \\ -y(N-1) & -y(N-2) & f(u(N-1),\hat{\theta}_n) & f(u(N-2),\hat{\theta}_n) \end{bmatrix}}_{\Phi_l(u,y,\hat{\theta}_n)} \begin{bmatrix} a_1 \\ a_2 \\ b_0 \\ b_1 \end{bmatrix} \tag{3.27}$$

The solution for linear parameter is

$$\hat{\theta}_l = [\hat{a}_1, \hat{a}_2, \hat{b}_0, \hat{b}_1]^T = (\Phi_l(u,y,\hat{\theta}_n)^T \Phi_l(u,y,\hat{\theta}_n))^{-1} \Phi_l(u,y,\hat{\theta}_n)^T Y_l(y) \tag{3.28}$$

The two-stage algorithm can be summarized as

Algorithm 1 Two-stage algorithm

Inputs: an initial value of the linear parameters, $\hat{\theta}_l$, an input/output data set $u(k)$, $y(k)$, $k = 1,2,\ldots,N$.
$\hat{\theta}_n = (\Phi_n(u,\hat{\theta}_l)^T \Phi_n(u,\hat{\theta}_l))^{-1} \Phi_n(u,\hat{\theta}_l)^T Y_n(y,\hat{\theta}_l)$
$\hat{\theta}_l = (\Phi_l(u,y,\hat{\theta}_n)^T \Phi_l(u,y,\hat{\theta}_n))^{-1} \Phi_l(u,y,\hat{\theta}_n)^T Y_l(y)$
Output: $\hat{\theta} = \begin{bmatrix} \hat{\theta}_n \\ \hat{\theta}_l \end{bmatrix}$

3.2.3 Simulation Study

Use of the Ramp Deconvolution (RD) method has led to high performance tracking within the Southampton ILC rehabilitation project. However, in order to motivate use of alternative identification schemes, in this section the two-stage and Separable Least Squares (SLS) Optimization Algorithm will be compared against the RD method using a simulated muscle system. Although this muscle system adopts the underlying form assumed by the RD method, it will be shown that the alternative approaches are more accurate and show greater robustness properties. The simulated muscle system will first be introduced and a triangular ramp input will be applied to the system. The resulting stimulated data will be used to identify the Hammerstein structure by three methods: Two-stage algorithm, Ramp Deconvolution method and SLS. The latter two approaches will be introduced briefly in the next section.

1. Simulated muscle system

   The simulated muscle system is designed as follows:

   (a) The nonlinearity $f(\cdot)$, shown in Fig. 3.3, is a sigmoid function:

$$f(u) = x_1 \cdot \frac{e^{x_2 u} - 1}{e^{x_2 u} + x_3} \tag{3.29}$$

where $x_1 = 6.8994$, $x_2 = 0.0410$ and $x_3 = 2.3897 \times 10^3$

(b) The linear dynamics $G(q)$ is an underdamped second order system with $T_\omega = 0.5284$, $\zeta = 0.6369$ and gain is 1.

(c) The noise $e(t)$ will be designed to different noise levels:

- $e(t) = 0$ ideal system without any noise.
- $e(t)$ is normally distributed random noise with zero mean and standard deviation $\lambda = 0.02, 0.04, 0.06, 0.08, 0.10$, respectively.

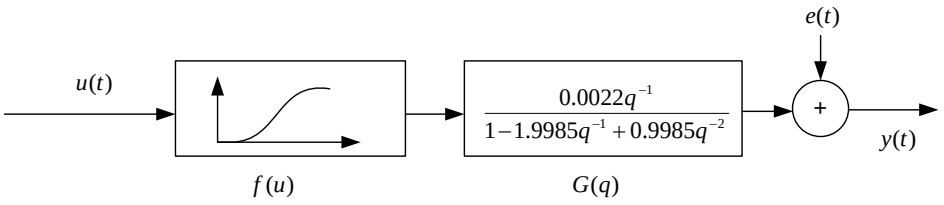


FIGURE 3.3: Simulated Muscle System

The parameter values used here come from real experimental data from a stroke patient. After a long period of testing on this patient, we have estimated a model which is representative and reliable. This model has been used in clinical tests and has produced a high level of accuracy when applied with an simple ILC scheme [Freeman et al., 2009b]. For example, when stimulated with the same triangular input signals, the responses of the simulated muscle system and real muscle of the stroke patient are very close to each other, see Fig. 3.4.

2. Ramp deconvolution method

In this method, the Isometric Recruitment Curve (IRC) is estimated by deconvolving the response of a muscle to a ramp input [Durfee and MacLean, 1989] and the Linear Activation Dynamics (LAD) is represented by a critically damped second-order system [Baratta and Solomonow, 1990]. This method has already been used to identify models of stroke patients in this project and it is implemented as follows, also see Fig. 3.5 for visual aid. First, a triangular input is applied in which the up and down segments are each of 5 second duration and the elbow torque is recorded. Second, the elbow torque is deconvolved using the Linear Activation Dynamics (LAD). When plotted against the applied pulsewidth, this provides two isometric recruitment curves, corresponding to the increasing and decreasing ramps respectively. Then the expression

$$f(u) = x_1 \cdot \frac{e^{x_2 u} - 1}{e^{x_2 u} + x_3} \tag{3.30}$$

is selected to fit the data by a nonlinear Matlab function '*lsquarefit*'. Third, $h(\tau)$ is convolved with $f(u)$ to produce the whole nonlinear dynamical model. The transfer function of Linear Activation Dynamics (LAD) is

$$h(s) = \frac{1}{T_\omega^2 s^2 + 2T_\omega s + 1} \tag{3.31}$$

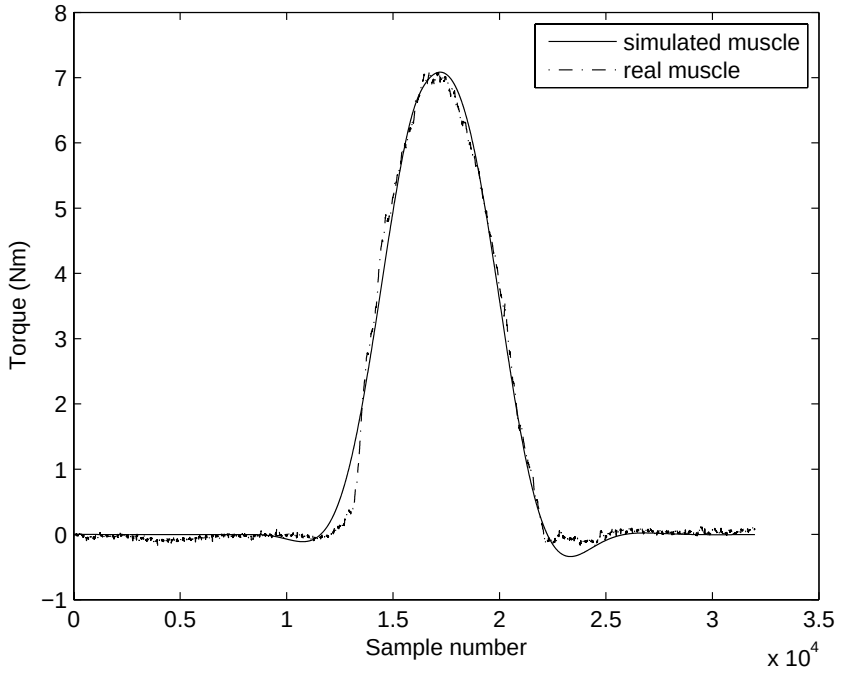


FIGURE 3.4: Comparison of the responses of the simulated muscle system and the real muscle of a stroke patient to a triangular ramp input signal

The natural frequency $f_\omega = \frac{1}{T_\omega}$ of the system is chosen as 0.85π which has been shown to be accurate for most people.

3. Separable Least Squares optimization algorithm

The Separable Least Squares (SLS) Optimization Algorithm was proposed for Hammerstein Structure identification and applied to a biological system (stretch reflex electromyogram) successfully by [Westwick and Kearney, 2001]. For this reason, SLS has been implemented here and compared with two-stage method in a simulation study. In this method, the linear dynamics is described by its impulse response, $h(\tau)$, which is assumed to be of finite length $T = 41$. The static nonlinearity $f(u)$ is expressed by (3.30), including three nonlinear parameters a_1, a_2 and a_3. Thus, the model output is

$$\hat{z}(t) = \sum_{\tau=0}^{T-1} h(\tau) f(u(t-\tau)). \tag{3.32}$$

It is obvious that the output is a linear function of the filter weights $h(\tau)$ and is nonlinear in the parameters a_1, a_2 and a_3. Thus, a parameter vector θ, including filter weights and nonlinear parameters is used to represent model output.

$$\theta = [h(0),\ \cdots,\ h(T-1),\ a_1,\ a_2,\ a_3]^T = [\theta_l^T | \theta_n^T]^T \tag{3.33}$$

The SLS method performs the iterative search only for the nonlinear parameters to find

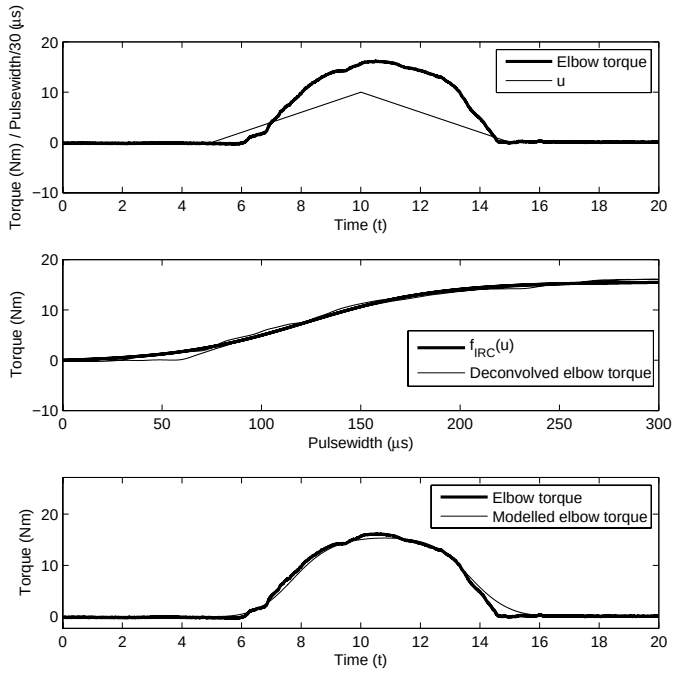


FIGURE 3.5: (1) Stimulation ramp input and recorded elbow torque, (2) deconvolved elbow torque plotted against pulsewidth with fitted fuction, (3) measured elbow torque and modelled elbow torque

the parameter vector θ, that minimizes the cost function

$$V_N(\theta) = \frac{1}{2N}\sum_{t=1}^{N}\varepsilon^2(t,\theta). \tag{3.34}$$

where $\varepsilon(t,\theta) = z(t) - \hat{z}(t,\theta)$ which is the error between the model output and the measured output. N is the number of samples used as the input and output signals.

4. Results

Results are in terms of the Best Fit rate, defined as the percentage,

$$Best\ Fit = \left(1 - \frac{\|y-\hat{y}\|_2}{\|y-\bar{y}\|_2}\right) \times 100 \tag{3.35}$$

where y is the measured output, $\hat{y}$ is the simulated model output and $\bar{y}$ is the mean of y. For each noise level, 100 independent trials are performed and the mean values and standard deviations of the Best Fit rates are calculated and shown in Table 3.1. Under noise-free conditions, the two-stage method can almost reconstruct the simulated system. Although the variance of the two-stage method is larger than the other two, it is still comparably small enough to not affect the performance at all. More intuitively, from Fig. 3.6, it can be seen that under different levels of normally distributed random noise, the three methods all degrade to some extent but the two-stage method is still superior to the other two.

TABLE 3.1: Best Fit (%) for different noise levels

	Two-Stage	Ramp Deconv	SLS
noise-free	99.98	91.24	92.06
$\lambda = 0.02$	99.07 ± 0.07	91.19 ± 0.00	92.01 ± 0.01
$\lambda = 0.04$	98.14 ± 0.02	91.07 ± 0.00	91.87 ± 0.00
$\lambda = 0.06$	97.27 ± 0.03	90.86 ± 0.00	91.64 ± 0.00
$\lambda = 0.08$	96.31 ± 0.15	90.58 ± 0.00	91.32 ± 0.01
$\lambda = 0.10$	95.33 ± 0.22	90.24 ± 0.00	90.94 ± 0.00

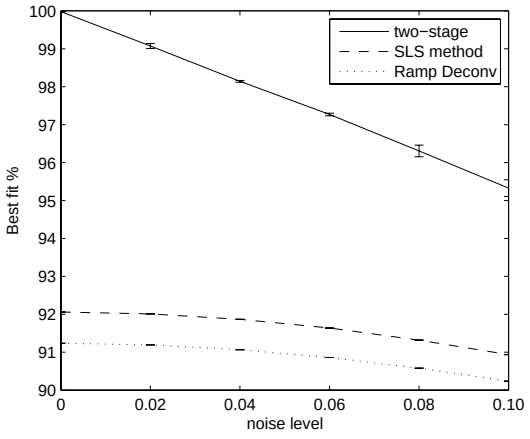


FIGURE 3.6: Simulation results for different noise levels

3.3 Two Iterative Algorithms

From the results above, the two-stage algorithm developed is shown to outperform the Ramp Deconvolution method and Separable Least Square method on a simulated muscle system with different noise levels. However, only one iteration of optimization of both the linear and nonlinear parameters is executed in the two-stage algorithm. In order to further improve the accuracy, it is proposed to repeatedly execute the two stages until convergence is achieved, which leads to the following algorithms.

These two iterative algorithms differ in terms of the assumed noise models:

one is for the ARX model with the system equation in (3.5) and

$$\frac{1}{\hat{A}(q)}v = y - G(q,\hat{\theta}_l)f(u,\hat{\theta}_n) = y - \frac{\hat{B}(q)}{\hat{A}(q)}f(u,\hat{\theta}_n) \tag{3.36}$$

and the other one is for the OE model, where

$$v = y - G(q,\hat{\theta}_l)f(u,\hat{\theta}_n) = y - \frac{\hat{B}(q)}{\hat{A}(q)}f(u,\hat{\theta}_n) \tag{3.37}$$

is transformed from (3.6).

3.3.1 Nonlinear Parameter Identification

Assume that an initial estimate of the linear parameter vector, $\hat{\theta}_l$, is available. The nonlinear parameters can be identified using the initial estimate of the linear parameter as follows.

- ARX model:

  Multiplying (3.36) by $\hat{A}(q)$ and substituting the resulting expression for v in (3.8) yields

$$\hat{\theta}_n = \arg\min_{\theta_n} \left\| \hat{A}(q)y - \hat{B}(q)f(u,\theta_n) \right\| \tag{3.38}$$

  From (3.2), it can be seen that $f(u,\theta_n)$ is linear in θ_n, so that

$$\begin{aligned}
\left(\hat{B}(q)f(u,\theta_n)\right)(k) = \sum_{i=1}^{m-2} \beta_i \underbrace{\left(\hat{b}_0|u(k-d)-u_{i+1}|^3 + \cdots + \hat{b}_n|u(k-d-n)-u_{i+1}|^3\right)}_{f_i(u(k),\hat{\theta}_b)} \\
+ \beta_{m-1} \underbrace{\left(\hat{b}_0 + \cdots + \hat{b}_n\right)}_{f_{m-1}(u(k),\hat{\theta}_b)} \\
+ \beta_m \underbrace{\left(\hat{b}_0 u(k-d) + \cdots + \hat{b}_n u(k-d-n)\right)}_{f_m(u(k),\hat{\theta}_b)} \\
+ \beta_{m+1} \underbrace{\left(\hat{b}_0 u(k-d)^2 + \cdots + \hat{b}_n u(k-d-n)^2\right)}_{f_{m+1}(u(k),\hat{\theta}_b)} \\
+ \beta_{m+2} \underbrace{\left(\hat{b}_0 u(k-d)^3 + \cdots + \hat{b}_n u(k-d-n)^3\right)}_{f_{m+2}(u(k),\hat{\theta}_b)}
\end{aligned} \tag{3.39}$$

  Therefore, (3.38) can be rewritten as an ordinary least squares problem

$$\arg\min_{\theta_n} \left\| Y_n(y,\hat{\theta}_a) - \Phi_n(u,\hat{\theta}_b)\theta_n \right\|_2 \tag{3.40}$$

  where assuming that $l > n+d$,

$$Y_n(y,\hat{\theta}_a) = \begin{bmatrix} y(l+1) + \hat{a}_1 y(l) + \cdots + \hat{a}_l y(1) \\ y(l+2) + \hat{a}_1 y(l+1) + \cdots + \hat{a}_l y(2) \\ \vdots \\ y(N) + \hat{a}_1 y(N-1) + \cdots + \hat{a}_l y(N-l) \end{bmatrix}$$

  and

$$\Phi_n(u,\hat{\theta}_b) = \begin{bmatrix} f_1(u(l+1),\hat{\theta}_b) & \cdots & f_{m+2}(u(l+1),\hat{\theta}_b) \\ f_1(u(l+2),\hat{\theta}_b) & \cdots & f_{m+2}(u(l+2),\hat{\theta}_b) \\ \vdots & & \vdots \\ f_1(u(N),\hat{\theta}_b) & \cdots & f_{m+2}(u(N),\hat{\theta}_b) \end{bmatrix}$$

Therefore, the solution of (3.38) is

$$\hat{\theta}_n = \left(\Phi_n(u,\hat{\theta}_b)^T \Phi_n(u,\hat{\theta}_b)\right)^{-1} \Phi_n(u,\hat{\theta}_b)^T Y_n(y,\hat{\theta}_a) \tag{3.41}$$

- OE model

Let $\hat{y}$ be the output of $\hat{G}$ when the input is $f(u,\theta_n)$, i.e.,

$$\hat{y}(k) = \frac{\hat{B}(q)}{\hat{A}(q)} f(u,\theta_n) \tag{3.42}$$

Multiplying both sides of (3.42) by $\hat{A}(q)$, gives

$$\hat{A}(q)\hat{y}(k) = \hat{B}(q) f(u,\theta_n) \tag{3.43}$$

and expanding $\hat{B}(q) f(u(k),\theta_n)$ as in (3.39) yields the matrix equation

$$T(\hat{\theta}_a)\hat{Y} = \Phi_n(u,\hat{\theta}_b)\theta_n \tag{3.44}$$

where

$$T(\hat{\theta}_a) = \begin{bmatrix} \hat{a}_l & \cdots & \hat{a}_1 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & \hat{a}_l & \cdots & \hat{a}_1 & 1 & \cdots & \cdots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & \cdots & \cdots & 0 & \hat{a}_l & \cdots & \hat{a}_1 & 1 \end{bmatrix} \quad \text{and} \quad \hat{Y} = \begin{bmatrix} \hat{y}(1) \\ \hat{y}(2) \\ \vdots \\ \hat{y}(N) \end{bmatrix} \tag{3.45}$$

However, $T(\hat{\theta}_a)$ is of dimension $(N-l)\times N$, which implies that the solution for $\hat{Y}$ is not unique. The system theoretic interpretation of this linear algebra fact is that: the output cannot be uniquely determined by the given model and input. Indeed, there are additional degrees of freedom in the choice of the *initial conditions*. In order to make the solution of problem (3.44) unique, let zero initial conditions be assumed. This choice is justifiable in the context of the muscle identification problem because the experiment starts with the muscle "at rest". The choice of zero initial conditions amounts to extending the data by zeros in the past, which in turn means that the matrices $T(\hat{\theta}_a)$ and $\Phi_n(u,\hat{\theta}_b)$ are extended to comprise N columns. Then (3.44) becomes

$$T_{ext}(\hat{\theta}_a)\hat{Y} = \Phi_n(u_{ext},\hat{\theta}_b)\theta_n \tag{3.46}$$

with

$$T_{ext}(\hat{\theta}_a) = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ \hat{a}_1 & 1 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & & & & & & \vdots \\ \hat{a}_l & \cdots & \hat{a}_1 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & \hat{a}_l & \cdots & \hat{a}_1 & 1 & \cdots & \cdots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & \cdots & \cdots & 0 & \hat{a}_l & \cdots & \hat{a}_1 & 1 \end{bmatrix} \tag{3.47}$$

and

$$\Phi_n(u_{ext}, \hat{\theta}_b) = \begin{bmatrix} f_1(u(1), \hat{\theta}_b) & \cdots & f_{m+2}(u(1), \hat{\theta}_b) \\ f_1(u(2), \hat{\theta}_b) & \cdots & f_{m+2}(u(2), \hat{\theta}_b) \\ \vdots & & \vdots \\ f_1(u(N), \hat{\theta}_b) & \cdots & f_{m+2}(u(N), \hat{\theta}_b) \end{bmatrix} \tag{3.48}$$

Consequently, $\hat{Y}$ can be solved from (3.46) as

$$\hat{Y} = T_{ext}^{-1}(\hat{\theta}_a)\Phi_n(u_{ext}, \hat{\theta}_b)\theta_n \tag{3.49}$$

Substituting $\hat{Y}$ in (3.37), the cost function (3.8) becomes

$$\hat{\theta}_n = \arg\min_{\theta_n} ||Y - T_{ext}^{-1}(\hat{\theta}_a)\Phi_n(u_{ext}, \hat{\theta}_b)\theta_n||_2 \tag{3.50}$$

which can be solved approximately in the least squares sense to obtain the estimate of the nonlinear parameter vector, $\hat{\theta}_n$

$$\hat{\theta}_n = \left(\left(T_{ext}^{-1}(\hat{\theta}_a)\Phi_n(u_{ext}, \hat{\theta}_b) \right)^T T_{ext}^{-1}(\hat{\theta}_a)\Phi_n(u_{ext}, \hat{\theta}_b) \right)^{-1} \left(T_{ext}^{-1}(\hat{\theta}_a)\Phi_n(u_{ext}, \hat{\theta}_b) \right)^T Y \tag{3.51}$$

3.3.2 Linear Parameter Identification

Given an estimate $\hat{\theta}_n$ for the nonlinear parameter vector θ_n, the cost function (3.8) can be minimized over the linear parameter vector θ_l. This subproblem is a linear least squares minimization in the ARX case but it is a difficult nonlinear least squares problem in the OE case.

- ARX model

  The minimization problem in the case of an ARX model is

$$\hat{\theta}_l = \arg\min_{\theta_l} \left\| A(q)y - B(q)f(u, \hat{\theta}_n) \right\| \tag{3.52}$$

  or in a matrix form

$$\arg\min_{\theta_l} \left\| Y' - \Phi_l(u, y, \hat{\theta}_n)\theta_l \right\|_2 \tag{3.53}$$

where

$$Y' = \begin{bmatrix} y(l+1) & y(l+2) & \cdots & y(N) \end{bmatrix}^T \tag{3.54}$$

and

$$\Phi_l(u,y,\hat{\theta}_n) = \begin{bmatrix} -y(l) & \cdots & -y(1) & f(u(l+1-d),\hat{\theta}_n) & \cdots & f(u(l+1-d-n),\hat{\theta}_n) \\ -y(l+1) & \cdots & -y(2) & f(u(l+2-d),\hat{\theta}_n) & \cdots & f(u(l+2-d-n),\hat{\theta}_n) \\ \vdots & & \vdots & \vdots & & \vdots \\ -y(N-1) & \cdots & -y(N-l) & f(u(N-d),\hat{\theta}_n) & \cdots & f(u(N-d-n),\hat{\theta}_n) \end{bmatrix} \tag{3.55}$$

Therefore, the solution of (3.53) is

$$\hat{\theta}_l = \left(\Phi_l(u,y,\hat{\theta}_n)^T \Phi_l(u,y,\hat{\theta}_n)\right)^{-1} \Phi_l(u,y,\hat{\theta}_n)^T Y' \tag{3.56}$$

- OE model

Recall the partition (3.4) of the transfer function linear parameter vector θ_l into parameter θ_a of the denominator A and parameter θ_b of the numerator B. The output error can be minimized analytically over θ_b, reducing the number of optimization variables for the minimization problem.

For given θ_a, (3.43) can be rewritten in a matrix form similar to (3.46) as

$$T_{ext}(\hat{\theta}_a)\hat{Y} = \Phi'_l(u_{ext},\hat{\theta}_n)\theta_b \tag{3.57}$$

where

$$\Phi'_l(u_{ext},\hat{\theta}_n) = \begin{bmatrix} f(u(1-d),\hat{\theta}_n) & \cdots & f(u(1-d-n),\hat{\theta}_n) \\ f(u(2-d),\hat{\theta}_n) & \cdots & f(u(2-d-n),\hat{\theta}_n) \\ \vdots & & \vdots \\ f(u(N-d),\hat{\theta}_n) & \cdots & f(u(N-d-n),\hat{\theta}_n) \end{bmatrix} \tag{3.58}$$

so that

$$\hat{Y}(\theta_a,\theta_b) = T_{ext}^{-1}(\hat{\theta}_a)\Phi'_l(u_{ext},\hat{\theta}_n)\theta_b \tag{3.59}$$

Thus, for a given $\hat{\theta}_a$, the solution, $\hat{\theta}_b$, for θ_b is given by

$$\begin{aligned} \hat{\theta}_b &= \arg\min_{\theta_b} ||Y-\hat{Y}||_2 \\ &= \underbrace{\left(\left(T_{ext}^{-1}(\hat{\theta}_a)\Phi'_l(u_{ext},\hat{\theta}_n)\right)^T T_{ext}^{-1}(\hat{\theta}_a)\Phi'_l(u_{ext},\hat{\theta}_n)\right)^{-1}\left(T_{ext}^{-1}(\hat{\theta}_a)\Phi'_l(u_{ext},\hat{\theta}_n)\right)^T Y}_{g(\hat{\theta}_a)} \end{aligned} \tag{3.60}$$

The OE minimization problem is thus reduced to an unconstrained nonlinear least squares

problem

$$\hat{\theta}_a = \arg\min_{\theta_a} ||Y - \hat{Y}(\theta_a, g(\theta_a))||_2 \tag{3.61}$$

with optimization variable θ_a only. Such a problem can be solved by standard local optimization methods, e.g., the Levenberg–Marquardt method.

Imposing stability of the identified model is in general difficult. In the muscle identification context, however, a second order system has been assumed by many authors, and in this case it can be shown that the stability constraint reduces to the following bound constraints on the parameters

$$0 < \hat{a}_2 \leq 1 \qquad \text{and} \qquad -2 \leq \hat{a}_1 \leq 0.$$

3.3.3 Algorithm Summary

For both model structures, the minimization over the θ_n and θ_l parameters can be executed repeatedly, which leads to the Algorithms 2 and 3.

Algorithm 2 Iterative algorithm for Hammerstein system identification with ARX model

Inputs: an initial value of the linear component, $\hat{\theta}_l^0$, an input/output data set $u(k)$, $y(k)$, $k = 1,2,\ldots,N$, and a convergence tolerance ε.
$j = 0$
repeat
$j = j+1$
$\hat{\theta}_n^j = \left(\Phi_n(u,\hat{\theta}_b^{j-1})^T \Phi_n(u,\hat{\theta}_b^{j-1})\right)^{-1} \Phi_n(u,\hat{\theta}_b^{j-1})^T Y_n(y,\hat{\theta}_a^{j-1})$
$\hat{\theta}_l^j = \left(\Phi_l(u,y,\hat{\theta}_n^j)^T \Phi_l(u,y,\hat{\theta}_n^j)\right)^{-1} \Phi_l(u,y,\hat{\theta}_n^j)^T Y'$
until $|V_N(\hat{\theta}_l^j, \hat{\theta}_n^j) - V_N(\hat{\theta}_l^{j-1}, \hat{\theta}_n^{j-1})| < \varepsilon$
Output: $\hat{\theta} = \begin{bmatrix} \hat{\theta}_n^j \\ \hat{\theta}_l^j \end{bmatrix}$

Algorithm 3 Iterative algorithm for Hammerstein system identification with OE model

Inputs: an initial value of the linear component, $\hat{\theta}_l^0$, an input/output data set $u(k)$, $y(k)$, $k = 1,2,\ldots,N$, and a convergence tolerance ε.
$j = 0$
repeat
$j = j+1$
$\hat{\theta}_n^j = \left(\left(T_{ext}^{-1}(\hat{\theta}_a^{j-1})\Phi_n(u_{ext},\hat{\theta}_b^{j-1})\right)^T T_{ext}^{-1}(\hat{\theta}_a^{j-1})\Phi_n(u_{ext},\hat{\theta}_b^{j-1})\right)^{-1} \left(T_{ext}^{-1}(\hat{\theta}_a^{j-1})\Phi_n(u_{ext},\hat{\theta}_b^{j-1})\right)^T Y$
$\hat{\theta}_a^j = \arg\min_{\theta_a} ||Y - \hat{Y}(\theta_a, g(\theta_a))||_2$ where $g(\theta_a)$ is defined in (3.60) and $\hat{\theta}_b^j = g(\hat{\theta}_a^j)$
until $|V_N(\hat{\theta}_l^j, \hat{\theta}_n^j) - V_N(\hat{\theta}_l^{j-1}, \hat{\theta}_n^{j-1})| < \varepsilon$
Output: $\hat{\theta} = \begin{bmatrix} \hat{\theta}_n^j \\ \hat{\theta}_l^j \end{bmatrix}$

3.4 Experimental Results

3.4.1 Test Design

Test Design is a crucial step for a successful identification procedure, which is more, the tests are not applied to a mechanical or physical process, but to a human being, in particular, a stroke patient. This means the test design must be given special attention.

1. Signal amplitude distribution

   The muscle behavior is revealed to be nonlinear so that in order to excite the whole nonlinearity, multi-level signals should be used. Many signals satisfy this requirement, for example, triangular ramp signals, staircase signals, pseudo-random multi-level signals, multiple sinusoids, white noise and filtered white noise. When the number of signal levels is sufficiently high, the next issue is how should the test signal amplitude be distributed over its variation range. The amplitude distribution will influence the accuracy of the model when there is process disturbance and unmodelled dynamics. The general guideline is that the test input amplitude should be similar to, but richer than, the input signal during typical process operations; and/or, use higher density in areas where high model accuracy is desired. A normal distribution of signal amplitude will put more weight on the area around the mean value of input at the cost of other areas; a uniformly distributed signal will treat the whole signal range equally. It is recommended that if there is not enough a priori knowledge about the process, uniformly distributed[1] test signals should be used [Zhu, 2000] which will treat the whole signal range equally.

2. Duration of test

   Common sense dictates that the identification test time should be sufficiently long so that the effects of unmeasured disturbance can be averaged out. A shorter test can be used for processes with lower noise level and/or simple models with smaller number of parameters; a longer test time is necessary when the process noise level is high and/or many parameters need to be determined. The muscle system undoubtedly belongs to the latter. However, a long test time will lead to fatigue in the stroke patients' muscles which will make our time-invariant model unreliable because the fatigue is obviously a time-varying factor. As a result, based on many simulation studies and real experimental experience, a test time of $20-30$ seconds is proposed in most cases.

3. Stimulation pattern

   In frequency modulation, many different stimulation patterns have been used and compared [Bobet et al., 2005]. However, stimulation pattern has not been considered before for the core of pulsewidth modulation. Here the stimulation pattern can be translated into the way the stimulation intensity changes or the way motor units are excited. It has been

[1] uniformly distributed means the occurrence frequencies of all levels are equal.

taken into account because we need to identify the model of the response of the muscle to electrical stimulation but we cannot isolate the muscle from the central nervous system. Therefore, we need to keep the muscle relaxed all the time in order to exclude the effect of an involuntary actuation component from the patient. Furthermore, since experimental tests are intended for application to stroke patients, their comfort is also of particular importance.

One method to ensure the patient is relaxed and comfortable is to recruit motor units gradually rather than abruptly exciting all desired units at once [Baker et al., 1993]. In pulsewidth modulation, this is guaranteed by gradually increasing the duration of each pulse, which will produce a gradual recruitment of nerve fibers after it exceeds the threshold of excitation. Moreover, a gradual reduction of pulse duration is also recommended to provide effectiveness and safety within a treatment program [Baker et al., 1993]. Without any doubt, the Triangular Ramp signal has these qualities. Here we also propose another input signal satisfying this requirement: Staircase signals, which to our knowledge have not previously been considered in identification tests for electrically stimulated muscle.

However, there remain many other tests, which do not fulfill these specifications, such as PRMS, and impulse trains with randomized activation levels [Ding et al., 2007]. In the literature, no discomfort has been reported from human subjects and no unreliability shown in the identified models resulting from the experimental data. Therefore, there is no reason to avoid use of these randomly exciting test signals and we choose two tests of this type. One is from the literature: PRMS and the other one also has not been used before: Filtered Random Noise signals.

4. Candidate tests

Based on the above discussion, the following tests are presented for identification of electrically stimulated muscle, see Fig. 3.7 for an example of each of the four candidate tests.

- Triangular Ramp (TR) test

  The pulse duration rises up from 0 to $300\mu s$ and then back to 0 again and the range of pulse duration is uniformly distributed.

- Staircase test

  The duration of each pulse changes step by step. The number of stairs should be great enough to identify the nonlinearity and the width of stairs should be chosen carefully. Denote $\tau = T_s/4$ (T_s is the 98% settling time). It is recommended to used a mixed stair width: use stair width τ for $1/3$ of the test period, 2τ for another $1/3$ of the test period and 3τ for remaining $1/3$ of the test period and mix them when creating the test signals [Zhu, 2000].

- Filtered Random Noise (FRN) test

  The value of pulse duration is set up by a filtered random noise. The low pass filter is used to make the duration not change too fast which may cause the patients'

discomfort or make their muscles tense and the filter gain can be adjusted so that the duration of pulses spans the whole desired range.

- Pseudo-Random Multi-level Sequences (PRMS) test

  The duration of pulse is decided by a Multi-level Pseudo-Random Sequence. The level is uniformly distributed on the whole range.

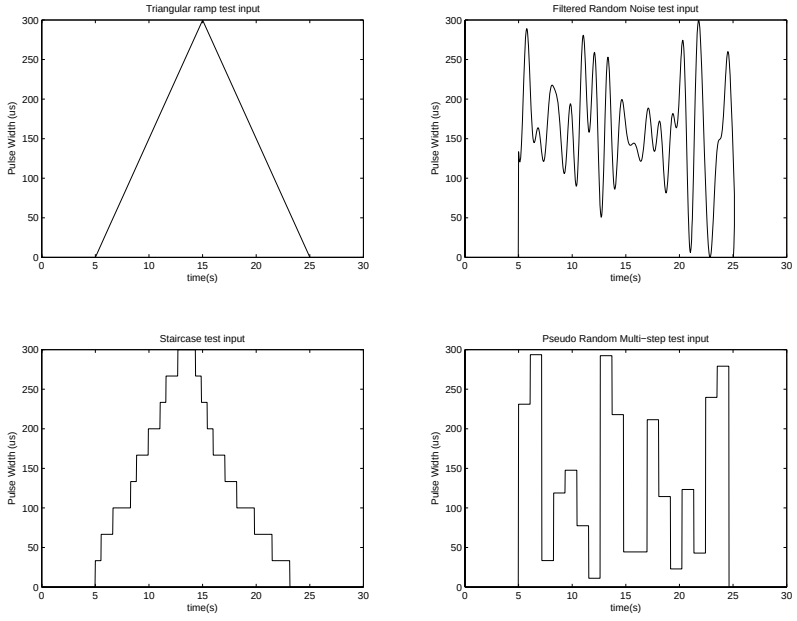


FIGURE 3.7: Test input

3.4.2 Experimental Set-up

University of Southampton Ethical approval has been obtained for conducting muscle identification tests, using the planar robot (S07/04-01). Tests were performed on a single unimpaired subject, and took place during two sessions conducted over consecutive days. Biometric measurements, including the length of the upper arm and forearm, were first made using anatomical landmarks, and then the participant was seated in the workstation. Their right arm was strapped to the extreme link of the five-bar robotic arm which incorporates a six axis force/torque sensor. The robotic arm provides support and constrains the forearm to lie in a horizontal plane. Straps were also applied about the upper torso to prevent shoulder and trunk movement. The subject's upper limb was then moved over as large an area as possible and a kinematic model of the arm was produced in order to calculate the torques applied about the subject's elbow joint. The electrode was then positioned on the lateral head of triceps and adjusted so that the applied FES generated maximum forearm movement. The stimulation consists of a series of bi-phasic pulses at $40Hz$, whose pulsewidth is variable from 0 to $300\mu s$ with a resolution of $1\mu s$. The amplitude, which is fixed throughout all subsequent tests, is determined by setting the pulsewidth

equal to $300\mu s$ and slowly increasing the applied voltage until a maximum comfortable limit is reached. A sample frequency of $40Hz$ is used by the real-time hardware, and all calculations are performed using the Matlab/Simulink environment.

The position of the robotic arm was then fixed using a locking pin, at an elbow extension angle, ϑ, of approximately $\pi/2$ rads. The identification tests that followed were each of 30 sec duration, and used excitation signals in which the first and last 5 sec periods consisted of zero stimulation. Only the middle 20 sec section of input and output data was used for identification, with the adjoining periods used to establish the baseline torque offset (taken as the mean torque value). The identification calculations were carried out immediately following each test in order to establish the efficacy of the data.

For the TR test, Staircase test and FRN test, 10 trials were performed, however, in the case of the PRMS test, only 4 trials were carried out as it was evident that the fit rate was poor. Between every two tests there was a rest period of at least 10 min in order to eliminate fatigue [Graham et al., 2006], and the order of identification tests was also randomized to minimize the effect of subject memory or acclimatization increasing the subject's involuntary response.

3.4.3 Results

The two iterative algorithms: Algorithm 2 and 3 are implemented here and the corresponding Hammerstein structure is specified as follows:

- Linear system

  The parameters (l,n,d) in (3.1) are assumed to be $(2,1,1)$ and hence, the transfer function is

  $$G(q) = \frac{b_0 q^{-1} + b_1 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}}$$

  The delay is chosen to be one since the combined delay resulting from muscle dead-time and communication protocol was anticipated to lie just below the sample time of 25ms, which means that the muscle responds to a change in activation within one sample period. A second order structure was assumed due to its wide use within Hammerstein structure modelling of electrically stimulated muscle [Durfee and MacLean, 1989; Baratta and Solomonow, 1990; Chizeck et al., 1988; Hunt et al., 1998] and its preliminary success in the previous programme of work at Southampton [Freeman et al., 2009b].

- Static nonlinear function

  Only one knot is assumed in the cubic spline function in (3.2), which is in the middle of the full range from 0 to $300\mu s$.

  $$f(u) = \beta_1 |u - 150|^3 + \beta_2 + \beta_3 u + \beta_4 u^2 + \beta_5 u^3$$

Thus, the parameter vectors to be estimated are

$$\theta_l = [a_1, a_2, b_0, b_1]^T$$

and

$$\theta_n = [\beta_1, \beta_2, \beta_3, \beta_4, \beta_5]^T.$$

These two algorithms are tested on the experimental data from all the candidate tests, compared with another three identification schemes, whose identification methods and corresponding model structures are described below:

1. PEM – General linear model

   pem is a Matlab function to estimate model parameters using an iterative prediction-error minimization method. Here, it is used for a general linear ARX model, which has the form:

   $$y(k) = \frac{b_0 q^{-d} + b_1 q^{-(d+1)} + \cdots + b_n q^{-(n+d)}}{1 + a_1 q^{-1} + \cdots + a_l q^{-l}} u(k) + \frac{1}{1 + a_1 q^{-1} + \cdots + a_l q^{-l}} v(k)$$

   Only a second order model was chosen because, when compared with higher order models, similar levels of fitting were observed. This is reflected in [Bobet et al., 2005] which compared several models for isometric muscles, and found that the general linear model with higher order gave fits that were only slightly better than those with a second order linear model. This suggests that the second order linear model provides a fit which is close to the best fit possible using a general linear model. Therefore, only a second order linear model is taken into account as the representation of the general linear model case, and the following (not necessarily critically damped) polynomial transfer function is chosen:

   $$G(q) = \frac{b_0 + b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}}$$

2. BAI – A modified Wiener-Hammerstein model

   This model structure and the corresponding identification comes from the recent paper [Bai et al., 2009] so that it is named after the author. A modified Wiener-Hammerstein model is proposed, shown in Fig. 2.5, which consists of two first-order linear blocks and a static nonlinearity in the middle. $v(k)$ and $w(k)$ are internal signals.

   $$v(k+1) = a_1 v(k) + a_2 u(k)$$

   $$w(k) = \frac{v(k)}{1 + v(k)}$$

   $$y(k+1) = b_1 y(k) + b_2 w(k)$$

   A Global Minimum Searching method is developed to find the four unknown parameters a_1, a_2, b_1 and b_2.

3. NLHW – Hammerstein structure

   nlhw is a Matlab function used to estimate Hammerstein-Wiener models which describes nonlinear dynamic systems using one or two static nonlinear blocks in series with a linear block. Only the linear block contains dynamic, time-variant elements. When excluding the output nonlinearity, the *nlhw* function can be used to estimate the Hammerstein structure.

   The linear block is a discrete-time transfer function and the order of the linear block can be specified by the following parameters:

   - n_b –The number of zeros plus one.
   - n_f –The number of poles.
   - n_k –The delay from input to the output in terms of the number of samples.

   The nonlinear block is implemented using nonlinearity estimators such as Dead Zone, Piecewise Linear and Saturation, etc. The estimator and number of units can be specified as well.

   In order to ensure continuity of the order of the linear block in the Hammerstein structure and comparability with the developed iterative algorithms, the parameters $n_b = 2$, $n_f = 2$ and $n_k = 1$ are chosen for the linear block and a piecewise estimator with the same amount of units as the knots in the cubic splines in (3.2) is used for the input nonlinear block. The output nonlinearity is excluded.

The results are all in terms of the Best Fit rate, defined in (3.35). It is noted that sometimes the best fit rates will go to negative numbers, which means the modeled outputs are even worse than a straight line representing the mean value of the measures outputs. Three aspects of results can be drawn:

- Identification results

  The identification results for each individual trial of four candidate tests are given together with the average results for all the trials, see Table 3.2 for three other identification methods and Table 3.3 for two iterative algorithms.

- Validation results

  To obtain the validation results, a model is firstly identified from the data of one trial and then is used to predict the outputs for all the trials in the same type of test. The results are the average values of all the prediction results in term of Best Fit rate. The validation results show the predictive ability within the same type of identification test. Due to the poor performance in identification results from the others, here only three algorithms are compared: Matlab function ‘nlhw’ and two iterative algorithms, listed in Table 3.4.

- Cross-validation results

TABLE 3.2: Identification results of three identification methods: Matlab Function 'pem' for a Linear model, Matlab function 'nlhw' for a Hammerstein Model and Global Minimum Searching method 'bai' for a modified Wiener-Hammerstein model for the four candidate tests. The results are in terms of the Best Fit Rate.

(a) PEM

	Triangular Ramp	Filtered Random Noise	Staircase	PRMS
1	64	41.1	89.54	53.95
2	poor	41.92	70.9	50.89
3	85.16	37.41	68.81	38.9
4	87.78	33.43	86.81	34.82
5	64.44	50.56	82.51	
6	89.57	59.01	67.23	
7	20.13	43.73	71.75	
8	83.59	57.16	32.93	
9	81.64	56.34	66.42	
10	88.09	57.48	70.91	
average	73.8	47.8	70.78	44.64

(b) BAI

	Triangular Ramp	Filtered Random Noise	Staircase	PRMS
1	67.54	35.47	87.33	43.63
2	74.78	41.24	67.99	42.62
3	59.79	35.29	73.48	40.85
4	66.05	27.56	77.24	24.78
5	63.37	46.22	84.70	
6	75.28	42.92	64.86	
7	68.54	39.11	70.47	
8	60.08	38.79	74.00	
9	62.88	44.98	60.99	
10	73.48	42.26	81.00	
average	67.18	39.38	74.21	37.97

(c) NLHW

	Triangular Ramp	Filtered Random Noise	Staircase	PRMS
1	89.66	40.33	84.85	55.67
2	53.68	11.75	85.43	75.15
3	92.31	1.55	86.74	57.14
4	92.18	39.43	91.31	41.49
5	92.87	81.08	83.32	
6	94.32	56.96	90.46	
7	85.72	58.43	83.68	
8	93.21	45.32	74.35	
9	94.42	80.75	85.54	
10	92.76	71.48	89.94	
average	88.11	48.71	85.56	57.36

Similarly, in order to show the predictive ability for different stimulation patterns, cross-validation analysis has been conducted, see Table 3.5. Firstly, a model is identified from

TABLE 3.3: Identification results of Algorithm 2 and Algorithm 3 for the four candidate tests. The results are in terms of the Best Fit Rate.

(a) Algorithm 2

	Triangular Ramp	Filtered Random Noise	Staircase	PRMS
1	85.88	42.54	87.62	50.34
2	88.34	36.39	89.16	52.48
3	91.33	36.09	85.18	52.43
4	89.23	36.58	89.68	36.69
5	92.25	63.38	89.84	
6	90.68	55.23	91.35	
7	89.14	48.12	88.33	
8	91.41	58.09	88.17	
9	94.25	74.74	83.46	
10	89.02	66.84	91.85	
average	90.15	51.8	88.46	48.00

(b) Algorithm 3

	Triangular Ramp	Filtered Random Noise	Staircase	PRMS
1	92.65	73.03	90.89	66.89
2	92.25	65.19	93.32	78.91
3	93.88	51.69	93.49	63.79
4	93.36	70.92	93.49	65.92
5	93.08	79.94	93.77	
6	91.98	68.46	92.34	
7	95.74	58.48	93.38	
8	92.41	61.50	94.66	
9	95.32	79.74	90.85	
10	92.60	71.32	94.23	
average	93.33	68.03	93.04	68.88

the data of all the trials in one type of test and then is used to predict the outputs for all the trials in one of the other tests. The results are again the average value of the Best Fit rate. Here only the TR, FRN, and Staircase tests are compared, due to the poor performance of the PRMS test in both identification and validation. For the same reason provided above, here only three algorithms are compared: Matlab function 'nlhw' and the two iterative algorithms.

3.4.4 Discussion

1. Model structure comparison

   It is obvious that the second-order linear model and its identification method 'pem' does not work very well, see Table 3.2(a). It is believed that this model structure is not complex enough to represent the muscle dynamics.

   A modified Wiener-Hammerstein model is used in [Bai et al., 2009] to model the force-

TABLE 3.4: Validation results of Algorithm 2, 3 and 'nlhw' for the four candidate tests. The model is identified from the listed data set and validated on all the data of the same test. The results are the average Best Fit Rate.

(a) Algorithm 2

	Triangular Ramp	Filtered Random Noise	Staircase	PRMS
1	82.28	28.01	73.99	11.17
2	82.78	45.00	83.63	43.45
3	79.01	40.52	77.77	46.32
4	82.51	8.79	77.27	44.10
5	82.83	37.62	82.72	
6	81.94	28.97	82.28	
7	78.51	40.67	81.20	
8	80.11	30.37	80.68	
9	82.80	44.77	78.09	
10	83.25	-45.65	81.04	
average	81.60	25.91	79.87	36.26

(b) Algorithm 3

	Triangular Ramp	Filtered Random Noise	Staircase	PRMS
1	76.12	16.90	75.86	46.08
2	80.94	26.34	84.39	32.15
3	76.58	36.32	83.42	29.50
4	81.80	16.31	83.47	50.22
5	81.79	24.37	75.82	
6	75.98	41.43	83.09	
7	68.03	29.49	83.32	
8	79.87	20.76	81.96	
9	80.32	46.80	80.67	
10	78.61	-30.94	83.81	
average	78.00	22.78	81.58	39.49

(c) NLHW

	Triangular Ramp	Filtered Random Noise	Staircase	PRMS
1	81.02	35.43	69.48	28.81
2	64.46	11.00	84.87	52.67
3	77.16	18.38	82.21	13.02
4	81.81	42.37	79.25	46.98
5	83.52	42.65	83.33	
6	81.13	29.58	82.25	
7	76.29	36.29	83.13	
8	80.42	36.68	84.20	
9	83.03	46.82	79.30	
10	83.76	5.44	80.03	
average	79.26	30.46	80.81	35.37

frequency relationship with fixed nonlinearity. However, it is found that in the identified nonlinearity by the iterative Algorithm there is a dead-zone between 0 to around $40\mu s$,

TABLE 3.5: Cross Validation results of Algorithm 2 and Algorithm 3 for the TR, FRN, and Staircase tests. The model is identified from all the data of one type of the test and validated on all the data of the other type of the test. The results are the average Best Fit Rate.

(a) Algorithm 2

	Triangular Ramp	Filtered Random Noise	Staircase
	(TR)	(FRN)	
(TR)	84.83	17.76	47.94
(FRN)	83.96	40.23	71.40
Staircase	79.53	42.67	84.48

(b) Algorithm 3

	Triangular Ramp	Filtered Random Noise	Staircase
	(TR)	(FRN)	
(TR)	81.80	41.08	64.11
(FRN)	68.75	46.80	67.86
Staircase	80.72	45.00	84.39

(c) NLHW

	Triangular Ramp	Filtered Random Noise	Staircase
	(TR)	(FRN)	
(TR)	85.55	-29.00	74.70
(FRN)	76.68	42.92	79.80
Staircase	71.26	39.71	82.71

which also can be illustrated by [Baker et al., 1993], while the fixed nonlinearity in the modified Wiener-Hammerstein model plotted in Fig. 3.8 does not contain dead-zone at all. This explains why the identification results of the modified Wiener-Hammerstein model are even worse than the linear model, Table 3.2(b). Thus, the modified Wiener-Hammerstein model is not suitable for the case of varying the stimulation intensity.

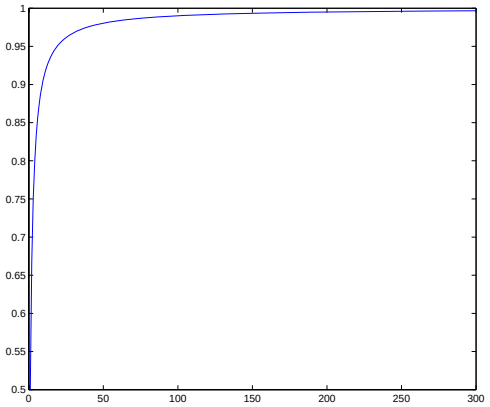
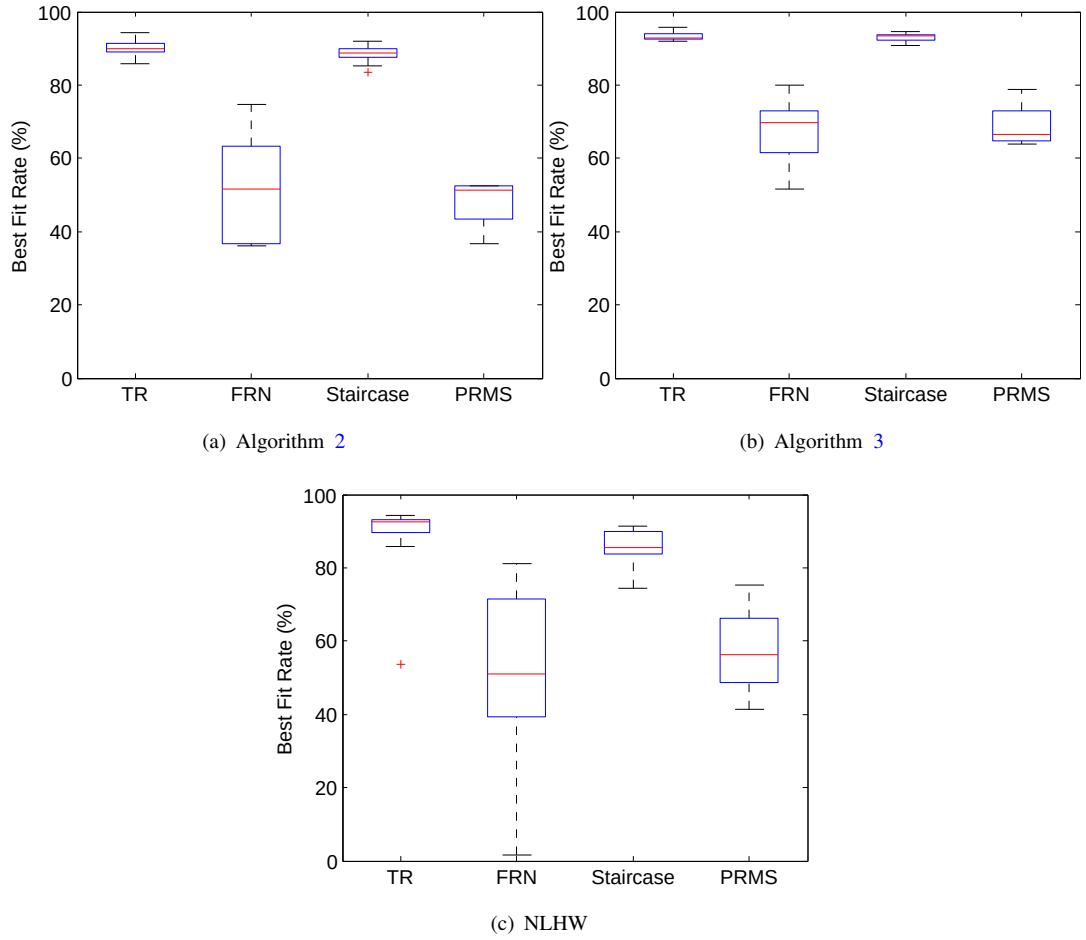


FIGURE 3.8: The fixed nonlinearity of the modified Wiener-Hammerstein model

The two iterative algorithms and the Matlab function 'nlhw' using the Hammerstein model outperform the others and shows significant improvement, Table 3.3 and Table 3.2(c).

Thus, it can be concluded that the Hammerstein model is the best choice based on the results above.

2. Comparison between the two iterative algorithms



(a) Algorithm 2

(b) Algorithm 3

(c) NLHW

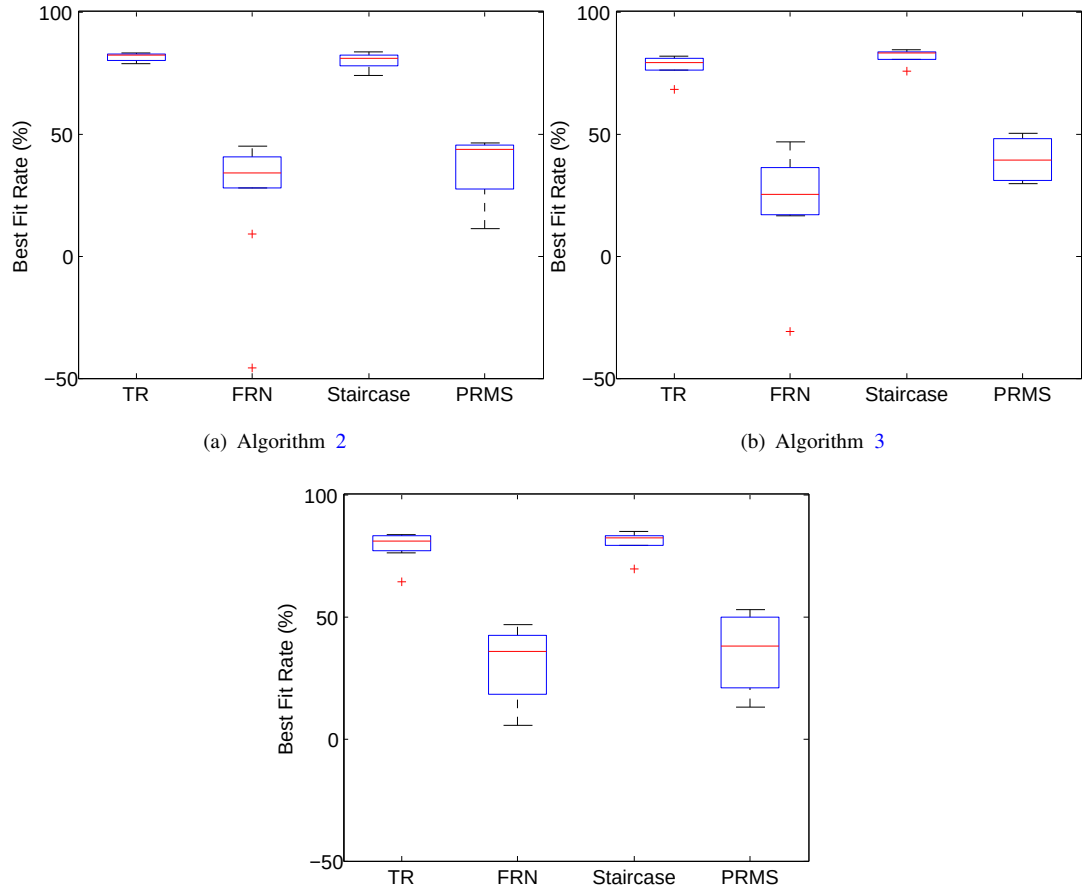
FIGURE 3.9: Box plots of identification results from two iterative algorithm and Matlab function 'nlhw'

In order to aid visual comparison, the box plots of the identification and validation results for the three comparable algorithms are presented in Fig. 3.9 and 3.10, respectively.

For the identification results, Algorithm 3 gives the best performance and the Matlab function 'nlhw' is the worst, even failing in one case (only achieving 1.55%). Algorithm 2 and 3 perform equivalently well in the validation results, easily observed from Fig. 3.10, but Matlab function 'nlhw' produces the largest box, which represents the Interquartile range of the results. Therefore, there is strong evidence that the two iterative algorithms are superior to the Matlab function 'nlhw' in most cases. Thus, Matlab function 'nlhw' will be excluded from the later comparison, which is between the two iterative algorithms in the following aspects:

- Initial values for linear parameters

  Algorithm 2 and Algorithm 3 both require the initial values of the linear parameters,



(a) Algorithm 2

(b) Algorithm 3

(c) NLHW

FIGURE 3.10: Box plots of validation results from the two iterative algorithms and Matlab function 'nlhw'

which can be obtained using any existing method which applies an input suitable for use with stroke patients. One such technique is the Ramp Deconvolution method [Durfee and MacLean, 1989], which was applied in the ILC stroke rehabilitation project reported in [Freeman et al., 2009b], and a representative choice of parameters may be taken. By using this representative estimate as the initial values, Algorithm 2 and Algorithm 3 can both achieve convergence after several iterations, illustrated by Fig. 3.11(a) and 3.11(b). However, irrespective of the iteration number, Algorithm 3 takes a longer period of time because in each iteration, an iterative search is applied.

In order to expedite the identification process of Algorithm 3, a better solution of the linear parameters is required as the initial values. The representative estimate from [Freeman et al., 2009b] is obviously not accurate enough and, moreover, the values of the linear parameters vary widely from subject to subject and it is difficult to find one representative estimate among all the subjects. Therefore, the optimal solution of the linear parameters from Algorithm 2 has been used to initialize Algorithm 3. This thereby unites the two algorithms in a single scheme which combines the speed of the first with the accuracy of the second. The results confirm high accuracy with

fewer iterations to converge, as illustrated by Fig. 3.11(c).

- Structure and unknown parameters

  A Hammerstein structure is used in both Algorithms, while the choice of the linear model is different. Algorithm 2 chooses the ARX linear model, where the white noise is assumed to pass through the denominator dynamics of the linear block before being added to the output. However, it is perhaps not the most natural form from a physical point of view. Thus, another linear model, OE model, is assumed in Algorithm 3, where white noise is added directly to the output, accounting for the measured errors from the equipment. The number of unknown parameters is kept the same for both Algorithms.

- Identification procedure

  The identification procedures of the two algorithms are not the same but they both alternatively optimize the nonlinear and linear parameters at each iteration.

  Algorithm 2 is a development of the Two-Stage identification method derived in Section 3.2, see also [Le et al., 2009], which has been shown to outperform the Ramp Deconvolution method and Separable Least Square method on a simulated muscle system with a range of noise levels. It alternatively solves the least squares problems to optimize the linear and nonlinear parameters. It is computationally easy and is reasonably fast in implementation.

  The identification procedure of Algorithm 3 is more complicated. In each iteration, the nonlinear parameters can be identified through use of transformations and related assumptions, in a least squares sense, while the identification of the linear parameters necessitates an iterative search technique to find the local optimal solution. Thus, it is more time consuming than Algorithm 2, but, by using the optimal solution from Algorithm 2 to provide initial values, the identification procedure of Algorithm 3 can be greatly speed up to the point where it is not a matter of concern.

- Performance

  Both Algorithm 2 and Algorithm 3 provide good fitting performance and predictive ability. Fig. 3.12 shows the fitting performance between the modeled outputs and measured outputs in both identification and validation cases.

  In terms of identification results, Algorithm 3 is superior to Algorithm 2, observed directly from Fig. 3.9(a) and 3.9(b). Numerically, Algorithm 3 improves the average results by up to 20% compared with Algorithm 2, as shown in Table 3.3.

  However, for validation results, both perform similarly, as shown by Fig. 3.10(a) and 3.10(b). Through examination of Table 3.4, Algorithm 2 is seen to be better for TR and FRN test data, while Algorithm 3 is better for the Staircase and PRMS test data. It is therefore reasonable to conclude that Algorithm 2 and Algorithm 3 have comparable performances in prediction.

The validation and prediction results provide the most direct indication of the models' accuracy when applied to the design of controllers for stroke rehabilitation. Since both

algorithms exhibit similar performance in this area, it is Algorithm 2's simpler implementation and faster computation that make it the preferable option. Whilst in this application Algorithm 3's increased complexity does not translate to improved results in validation and prediction, it is anticipated that applications exist in which it does outperform Algorithm 2.

3. Candidate tests comparison

   In order to visual aid the comparison, an example of the identification, validation and cross-validation results from four candidate tests are plotted in Fig. 3.13 and 3.14. In Fig. 3.13, for each candidate test, the identification result for one trial is plotted, followed by the validation result for another trial within the same type of identification test. In Fig. 3.14, the identified model from one type of candidate test is used to predict outputs for the other two. For example, in Fig. 3.14(d), a model is firstly identified from one trial of Staircase test data and then used to predict the output for one trial of FRN test data. The results plotted here are not average results as in the Table 3.4 and 3.5 but an example for one trial.

   Although the TR test is widely used in muscle tests such as [Freeman et al., 2009b], [Durfee and MacLean, 1989] and [Durfee and Palmer, 1994] and can achieve satisfactory fitting rates (almost the highest values in the identification case and approximately 80% for Algorithm 2 and a little lower for Algorithm 3 in the validation case), illustrated by Fig. 3.13(a), it shows poor capability in predicting other stimulation patterns, such as those in Table 3.5 and fitting plots in Fig. 3.14(c) and 3.14(e). This is due to its non-persistent excitation property discussed in the previous section, which leads to an unreliable model identified by this test.

   The FRN and PRMS tests are commonly used in system identification context but not widely applied to the electrically stimulated muscle fibres. The average identification results of these two tests by Algorithm 2 are 51.8% and 48% respectively, as shown in Table 3.3(a). Although Algorithm 3 improves on these by up to 20%, these tests are still far lower than those of the TR and Staircase tests. The validation results in Table 3.4 are even lower. In Fig. 3.13(b) and 3.13(d), the fitting plots for identification data are not bad but when using the identified models to predict the validation data, which is just a different trial within the same type of identification tests, the results are quite disappointing, not to mention the cross validation fitting plots for other types of identification tests in Fig. 3.14(f) and 3.14(a).

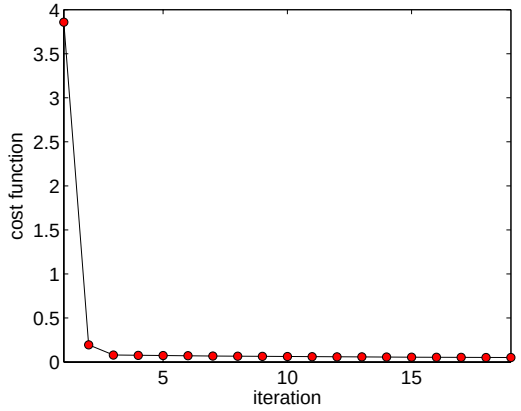
   There may be two reasons for this: the first is that the experimental data is not proper. Considering the effects caused by randomly exciting tests on the human subjects, it is believed that these signals elicit involuntary reflexes and subject discomfort, which results in noisy data. The second reason is that the model structure and the identification algorithms are not proper due to these tests containing higher frequency components than the other two. However, with respect to our particular control application, where the control inputs employed in clinical treatment are similar to TR and Staircase signals, see Fig. 5.14

for example, the models form from FRN and PRMS tests, even identified by improved identification algorithms or advanced model structures, e.g. high-order system, may be expected to lead to poor results when transferred to model-based control application due to its lack of output prediction for TR and Staircase tests, see Fig. 3.14(f) and 3.14(a).

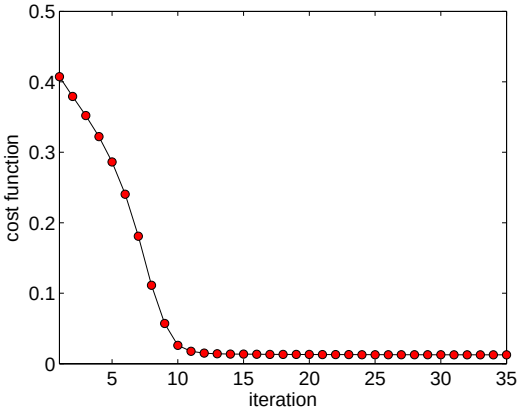
To the best of the authors knowledge, this is the first time the Staircase test has been used in the identification of electrically stimulated muscles, and it has shown clear advantage over alternatives, i.e. it is persistently exciting, gives high fitting rates in the identification case (the second highest one in Table 3.3) and in the validation case (surpassing even the TR test for Algorithm 3 in Table 3.4(b)), illustrated by Fig. 3.13(c) and shows accurate predictive ability across different stimulation patterns (see Table 3.5 and Fig. 3.14(d) and 3.14(b)). Therefore the Staircase test is highly recommended for the identification of electrically stimulated muscle.

4. Sample Deficiency

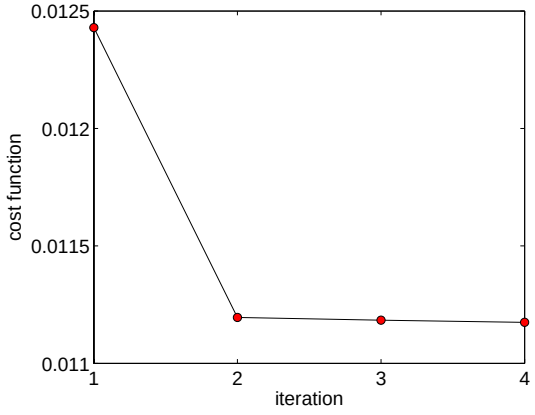
   Our study has some limitations which should be borne in mind when judging these comparisons. One is lack of samples. Only one unimpaired human sample is used and at least 34 identification tests were carried out on the same sample subject during several consecutive days with randomized order. When tested on more subject samples, it is expected that there will be variation in identified linear dynamics and nonlinear recruitment curves, which can be illustrated by several similar results, e.g. [Munih et al., 2000]. However, there is no evidence to show that subject sampling will affect the identification algorithm, the model structure or the identification tests, and this is reflected in [Munih et al., 2000], where the same model structure and identification algorithm were performed using experimental data from different subjects and similar results were obtained in each case. Thus, it can be concluded that the results here may be extended to the general cases but more identification tests on a wide range of human samples or even stroke patients will be carried out in the future.



(a)

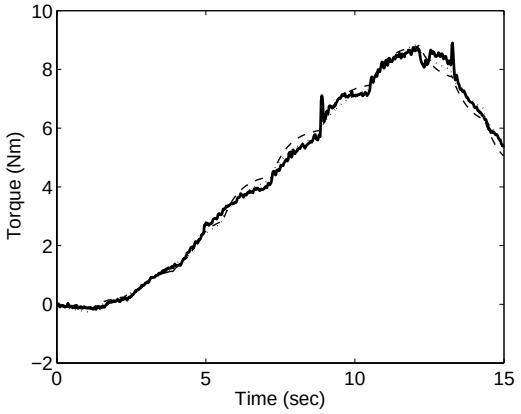


(b)

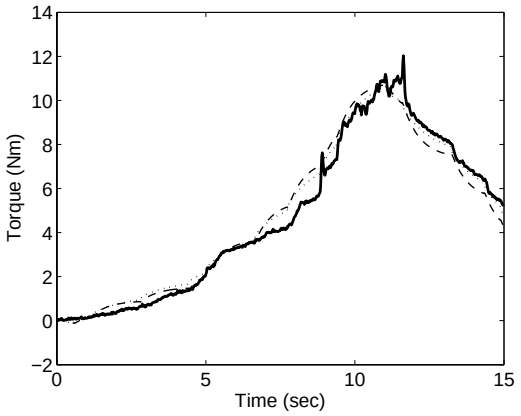


(c)

FIGURE 3.11: Examples of convergence properties for Algorithm 2 and Algorithm 3: (a) A representative estimate from [Freeman et al., 2009b] is used as the initial values and Algorithm 2 is applied. Convergence is achieved after 18 iterations, employing the tolerance ε defined in Algorithm 2; (b) A representative estimate from [Freeman et al., 2009b] is used as the initial values and Algorithm 3 is applied. Convergence is achieved after 35 iterations, using the tolerance ε defined in Algorithm 3; (c)The optimal solution from Algorithm 2 is used to provide the initial values and Algorithm 3 is applied. Convergence is achieved after 4 iterations, using the tolerance ε defined in Algorithm 3.

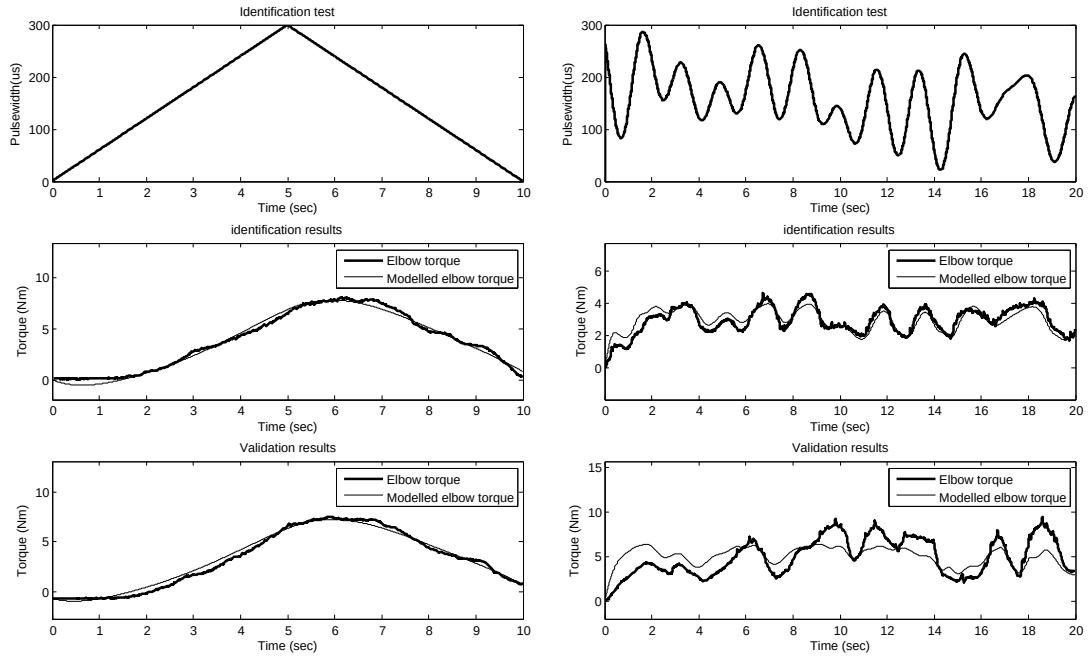


(a) Identification



(b) Validation

FIGURE 3.12: The force outputs of Algorithm 2 (dashed), Algorithm 3 (dotted) and the measured force outputs (solid) are plotted.



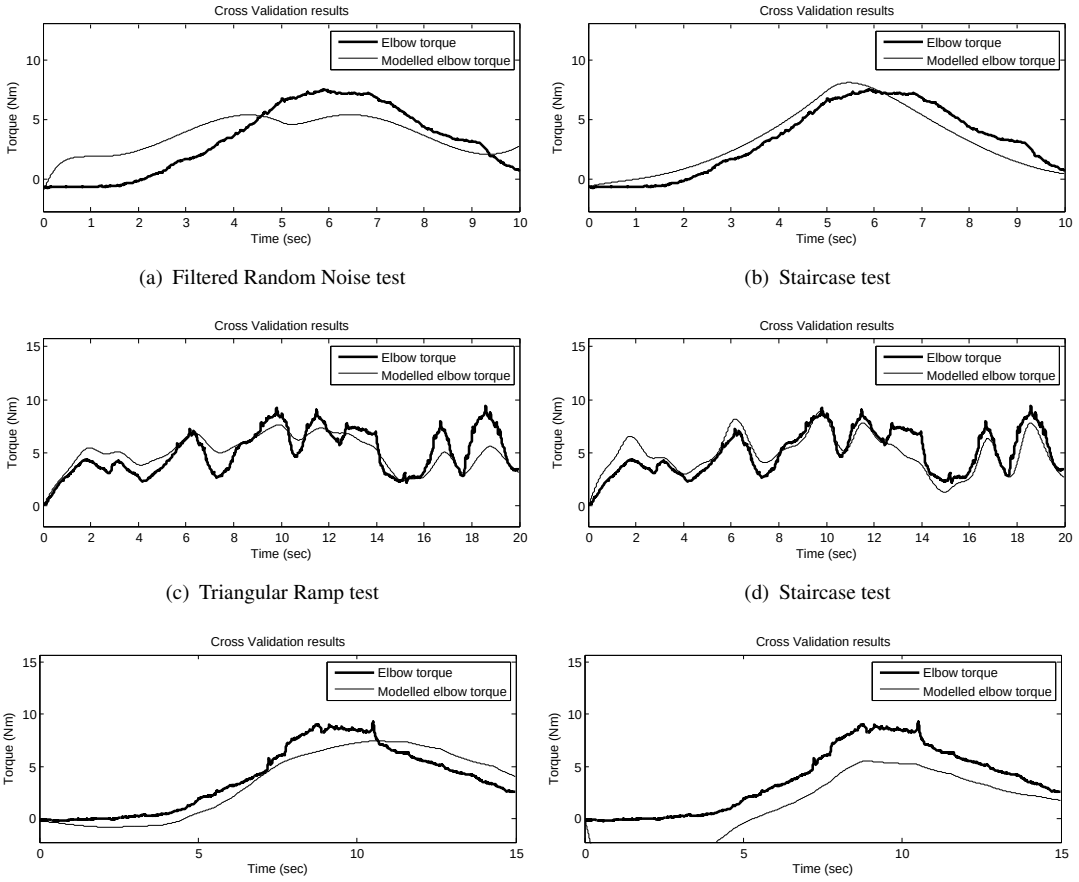
(a) Triangular Ramp test

(b) Filtered Random Noise test

(c) Staircase test

(d) Pseudo-Random Multi-level test

FIGURE 3.13: An example of the identification and validation results from four identification tests by Algorithm 2



(a) Filtered Random Noise test

(b) Staircase test

(c) Triangular Ramp test

(d) Staircase test

(e) Triangular Ramp test

(f) Filtered Random Noise test

FIGURE 3.14: An example of the cross validation results by Algorithm 2: (a) and (b) validate the models identified by Filtered Random Noise test and Staircase test, respectively, on the same trial of Triangular Ramp test data; (c) and (d) validate the models identified by Triangular Ramp test and Staircase test, respectively, on the same trial of Filtered Random Noise test data; (e) and (f) validate the models identified by Triangular Ramp test and Filtered Random Noise test, respectively, on the same trial of Staircase test data;

Chapter 4

Recursive Identification

The algorithms developed in the previous chapter constitute significant progress in the identification of electrically stimulated muscle, but the models were only verified over a short time interval (20 sec duration). However, in clinical trials, the duration of stimulation usually lasts between 30 minutes and 1 hour, where slowly time-varying properties of the muscle system arise due to fatigue, changing physiological conditions or spasticity. Motivated by this, online identification will be considered in this Chapter.

4.1 Problem Statement

Consider the discrete-time SISO Hammerstein model, shown in Fig. 4.1. The linear block is

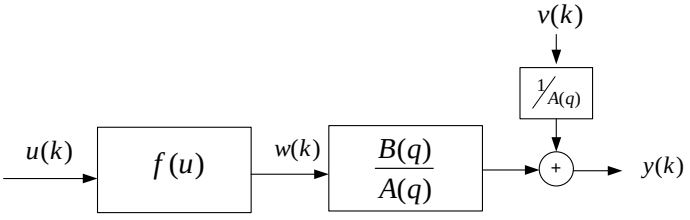


FIGURE 4.1: Hammerstein System

represented by ARX model:

$$y(k) = \frac{B(q)}{A(q)} w(k) + \frac{1}{A(q)} v(k) \tag{4.1}$$

and

$$B(q) = b_0 q^{-d} + b_1 q^{-(d+1)} + \cdots + b_n q^{-(n+d)} \tag{4.2}$$

$$A(q) = 1 + a_1 q^{-1} + \cdots + a_l q^{-l} \tag{4.3}$$

where q^{-1} is the delay operator and n, l and d are the number of zeros, poles and the time delay order, respectively. The parameters n, l and d are assumed to be known.

The nonlinearity is represented by a sum of the known nonlinear function $f_1, f_2, \ldots, f_m$ and a bias:

$$w(k) = f(u(k)) = \beta_0 + \sum_{i=1}^{m} \beta_i f_i(u(k)) \tag{4.4}$$

The considered identification problem is:

> Given N consecutive input-output data measurements $\{u(k), y(k)\}$ estimate recursively the linear parameters $[a_1, \ldots, a_l, b_0, \ldots, b_n]$ in (4.2,4.3) and the nonlinear parameters $[\beta_0, \ldots, \beta_m]$ in (5.1).

4.2 Recursive Algorithms

4.2.1 Recursive Least Square Algorithm

The well known RLS algorithm will be applied first, where in order to make the model linear in the parameters, over-parameterization of the Hammerstein structure is required. Then RLS is employed to recursively estimate the new parameter vector, and in the second step, SVD is used to recover the original parameters.

1. Over-parameterization

   Multiplying both sides of the difference equation (4.1) by $A(q)$ and rearranging the terms gives

$$\begin{aligned} y(k) = &- a_1 y(k-1) - \cdots - a_l y(k-l), \\ &+ b_0\beta_0 + b_0 \sum_{i=1}^{m} \beta_i f_i(u(k-d)) \\ &\vdots \\ &+ b_n\beta_0 + b_n \sum_{i=1}^{m} \beta_i f_i(u(k-d-n)) + v(k) \end{aligned} \tag{4.5, 4.6}$$

   Define the regressor ϕ as a combination of the past outputs and known nonlinear functions of the past inputs

$$\begin{aligned} \phi(k) = \Big[&- y(k-1), \ldots, -y(k-l), \\ &f_1(u(k-d)), \ldots, f_m(u(k-d)), \ldots, \\ &f_1(u(k-d-n)), \ldots, f_m(u(k-d-n)), 1 \Big]^T \end{aligned} \tag{4.7}$$

   and the extended parameter vector θ

$$\theta(k) = \Big[a_1, \ldots, a_l, \gamma_{01}, \ldots, \gamma_{0m}, \ldots, \gamma_{n1}, \ldots, \gamma_{nm}, \delta \Big]^T \tag{4.8}$$

where

$$\gamma_{ij} = b_i \beta_j \qquad i = 0,1,\ldots,n \quad j = 1,2,\ldots,m \tag{4.9}$$

and

$$\delta = \beta_0 \sum_{i=0}^{n} b_i \tag{4.10}$$

With these definitions, $y(k)$ can be expressed as linear in the new parameters

$$y(k) = \phi^T(k)\theta(k) + v(k) \tag{4.11}$$

Note that for the constant term in the nonlinearity, we only use one term in the ϕ vector as the last entry in (4.7) and merge the parameters of constant items into δ in (4.10). The RLS algorithm can now be applied as follows.

2. RLS algorithm

For a slowly time-varying system, a forgetting factor λ, $0 < \lambda \leq 1$ is introduced. This parameter weights the most recent data at unity, and data that is n time units old at λ^n.

The minimization criterion is

$$V(\theta,k) = \frac{1}{2}\sum_{i=1}^{k} \lambda^{k-i}\left(y(i) - \phi^T(i)\theta(i)\right)^2 \tag{4.12}$$

and the RLS algorithm for minimization of $V(\theta,k)$ over θ is given by the following equations:

$$P(k) = \frac{1}{\lambda}\left(P(k-1) - \frac{P(k-1)\phi(k)\phi^T(k)P(k-1)}{\lambda I + \phi^T(k)P(k-1)\phi(k)}\right) \tag{4.13}$$

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P(k)\phi(k)(y(k) - \phi^T(k)\hat{\theta}(k-1)) \tag{4.14}$$

3. Recover linear and nonlinear parameters

After each time instant, the new parameter vector θ is updated using (4.13) and (4.14). However, in order to feed into the controller, it is necessary to recover the linear and nonlinear parameters from the new parameter vector $\hat{\theta}$, which can be separated into the following three segments:

$$\hat{\theta}(k) = \begin{bmatrix} \hat{\theta}_a(k) \\ \hat{\theta}_{b\beta}(k) \\ \hat{\delta}(k) \end{bmatrix} \tag{4.15}$$

where

$$\hat{\theta}_a(k) = [\hat{a}_1(k) \quad \cdots \quad \hat{a}_l(k)]^T \tag{4.16}$$

and

$$\hat{\theta}_{b\beta}(k) = [\hat{\gamma}_{01}(k) \quad \cdots \quad \hat{\gamma}_{0m}(k) \quad \cdots \quad \hat{\gamma}_{n1}(k) \quad \cdots \quad \hat{\gamma}_{nm}(k)]^T \tag{4.17}$$

The estimated parameter $\hat{\gamma}_{ij}(k)$ are put in a $(n+1)\times m$ matrix

$$\hat{\Theta}'_{b\beta}(k) = \begin{bmatrix} \hat{\gamma}_{01}(k) & \cdots & \hat{\gamma}_{0m}(k) \\ \vdots & & \vdots \\ \hat{\gamma}_{n1}(k) & \cdots & \hat{\gamma}_{nm}(k) \end{bmatrix}$$

Note that in the absence of noise

$$\hat{\Theta}'_{b\beta}(k) = \begin{bmatrix} \hat{b}_0(k) \\ \vdots \\ \hat{b}_n(k) \end{bmatrix} \begin{bmatrix} \hat{\beta}_1(k) & \cdots & \hat{\beta}_m(k) \end{bmatrix}$$

so that $\hat{\Theta}'_{b\beta}(k)$ has rank equal to one. In the presence of noise, however, the estimated parameters $\hat{\gamma}_{ij}(k)$ need not form a rank-1 matrix $\hat{\Theta}'_{b\beta}(k)$. Consequently, $\hat{\Theta}'_{b\beta}(k)$ is approximated by a rank-1 matrix, which gives the estimated linear parameters $\hat{b}_i$ and estimated nonlinear parameters $\hat{\beta}_j$. The rank-1 approximation is achieved by computing the SVD

$$\hat{\theta}'_{b\beta}(k) = USV^T \tag{4.18}$$

Then

$$\hat{\theta}_b(k) = [\hat{b}_0(k) \quad \cdots \quad \hat{b}_n(k)]^T = U_1 S_1^{1/2} \tag{4.19}$$

$$\hat{\theta}_\beta(k) = [\hat{\beta}_1(k) \quad \cdots \quad \hat{\beta}_m(k)]^T = V_1 S_1^{1/2} \tag{4.20}$$

where U_1 and V_1 are the first columns of U and V, respectively, and S_1 is the first singular value. Finally,

$$\hat{\beta}_0(k) = \frac{\hat{\bar{\delta}}(k)}{\sum_{i=0}^{n} \hat{b}_i(k)} \tag{4.21}$$

and the estimated nonlinear and linear parameter vectors are

$$\hat{\theta}_n(k) = \begin{bmatrix} \hat{\beta}_0(k) \\ \hat{\theta}_\beta(k) \end{bmatrix} \quad \text{and} \quad \hat{\theta}_l(k) = \begin{bmatrix} \hat{\theta}_a(k) \\ \hat{\theta}_b(k) \end{bmatrix}$$

respectively.

4.2.2 Alternative Recursive Least Square Algorithm

The use of over-parameterization and subsequent rank-1 approximation often leads to a model which poorly fits the original data (as illustrated in Section 4.3). A new recursive identification method, named Alternative Recursive Least Square (ARLS) algorithm, is therefore developed which avoids over-parameterization by instead splitting the model into nonlinear and linear components, where each is identified independently using a parallel implementation. This method builds on the schemes described in Chapter 3 in which two iterative algorithms were developed for Hammerstein systems with differing noise models, and in each case nonlinear and linear

parameters were alternately optimized by different projection algorithms. Whilst both involved Least Squares (LS) optimization for offline identification, and therefore extend naturally to the online case through application of RLS, the one with simpler implementation and faster computation time will be taken as a starting point. By invoking certain approximations, this algorithm can be implemented recursively as follows:

- Recursive identification of linear parameters

  As described in Chapter 3, the parameters of the ARX model can be separated into linear and nonlinear parameter vectors

$$\theta_n = [\beta_0 \ \cdots \ \beta_m]^T \tag{4.22}$$

$$\theta_l = [a_1 \ \cdots \ a_l \ \ b_0 \ \cdots \ b_n]^T \tag{4.23}$$

  Assuming that the nonlinear parameter vector θ_n is known at the kth time instant, $y(k)$ can be expressed as a function of linear parameters $a_1(k),\ldots,a_l(k),b_0(k),\ldots,b_n(k)$ only

$$\begin{aligned} y(k) = &- a_1(k)y(k-1) - \cdots - a_l(k)y(k-l) \\ &+ b_0(k)f(u(k-d),\theta_n) + \cdots + b_n(k)f(u(k-d-n),\theta_n) + v(k) \end{aligned} \tag{4.24}$$

  or

$$y(k) = \phi_l^T(k,\theta_n)\theta_l(k) + v(k) \tag{4.25}$$

  where

$$\phi_l^T(k,\theta_n) = [-y(k-1) \ \cdots \ -y(k-l) \ \ f(u(k-d),\theta_n) \ \cdots \ f(u(k-d-n),\theta_n)] \tag{4.26}$$

  A forgetting factor λ_l is used in the recursive least squares algorithm to minimize the criterion

$$V_l(\theta_l,k) = \frac{1}{2}\sum_{i=1}^{k} \lambda_l^{k-i}\left(y(k) - \phi_l^T(k,\hat{\theta}_n(k-1))\theta_l(k)\right)^2 \tag{4.27}$$

  where the nonlinear parameter vector is approximated by the estimated value at the previous time instant $k-1$.

  The recursive algorithm for the linear parameter vector $\theta_l(k)$ is

$$P_l(k) = \frac{1}{\lambda_l}\left(P_l(k-1) - \frac{P_l(k-1)\phi_l(k,\hat{\theta}_n(k-1))\phi_l^T(k,\hat{\theta}_n(k-1))P_l(k-1)}{\lambda_l I + \phi_l^T(k,\hat{\theta}_n(k-1))P_l(k-1)\phi_l(k,\hat{\theta}_n(k-1))}\right) \tag{4.28}$$

$$\hat{\theta}_l(k) = \hat{\theta}_l(k-1) + P_l(k)\phi_l^T(k,\hat{\theta}_n(k-1))\left(y(k) - \phi_l^T(k,\hat{\theta}_n(k-1))\hat{\theta}_l(k-1)\right) \tag{4.29}$$

- Recursive identification for the nonlinear parameter vector

  As in the linear case, it is first assumed that the linear parameter vector θ_l is known.

Hence, at the kth time instant,

$$\underbrace{y(k)+a_1y(k-1)+\cdots+a_ly(k-l)}_{A(q,\theta_l)y(k)}=\beta_0(k)\sum_{i=0}^{n}b_i+$$
$$\beta_1(k)\sum_{i=0}^{n}b_if_1(u(k-d-i))+\cdots+\beta_m(k)\sum_{i=0}^{n}b_if_m(u(k-d-i))+v(k) \quad (4.30)$$

or, in matrix form,

$$A(q,\theta_l)y(k)=\phi_n^T(k,\theta_l)\theta_n(k)+v(k) \quad (4.31)$$

where

$$\phi_n^T(k,\theta_l)=[\sum_{i=0}^{n}b_i \quad \sum_{i=0}^{n}b_if_1(u(k-d-i)) \quad \cdots \quad \sum_{i=0}^{n}b_if_m(u(k-d-i))] \quad (4.32)$$

In order to recursively update the nonlinear parameter vector, the linear parameter vector is approximated by the estimated value from the previous time instant, resulting in the recursive least squares criterion

$$V_n(\theta_n,k)=\frac{1}{2}\sum_{i=1}^{k}\lambda_n^{k-i}\left(A(q,\hat{\theta}_l(k-1))y(k)-\phi_n^T(k,\hat{\theta}_l(k-1))\theta_n(k)\right)^2 \quad (4.33)$$

The recursive algorithm for the nonlinear parameter vector is

$$P_n(k)=\frac{1}{\lambda_n}\left(P_n(k-1)-\frac{P_n(k-1)\phi_n(k,\hat{\theta}_l(k-1))\phi_n^T(k,\hat{\theta}_l(k-1))P_n(k-1)}{\lambda_nI+\phi_n^T(k,\hat{\theta}_l(k-1))P_n(k-1)\phi_n(k,\hat{\theta}_l(k-1))}\right) \quad (4.34)$$

$$\hat{\theta}_n(k)=\hat{\theta}_n(k-1)+P_n(k)\phi_n^T(k,\hat{\theta}_l(k-1))\left(A(q,\hat{\theta}_l(k-1))y(k)-\phi_n^T(k,\hat{\theta}_l(k-1))\hat{\theta}_n(k-1)\right) \quad (4.35)$$

4.2.3 Initial Values for Two Algorithms

- RLS

  The initial values for the RLS are θ and P, which are calculated from several initial samples by the batch least squares algorithm. The number of samples is decided by the dimension of ϕ in order to obtain the unique solution.

$$\theta_{ini}=(\Phi^T\Phi)^{-1}\Phi^TY \quad (4.36)$$

$$P_{ini}=(\Phi^T\Phi)^{-1} \quad (4.37)$$

where

$$Y=\begin{bmatrix} y(1) \\ \vdots \\ y(T_{\text{ini}}) \end{bmatrix} \quad \text{and} \quad \Phi=\begin{bmatrix} \phi^T(1) \\ \vdots \\ \phi^T(N_{\text{ini}}) \end{bmatrix}$$

The matrix Φ may become singular or poorly conditioned and hence there exist problems with computing its inverse. Consequently, a regularization is applied, in which case (4.36) and (4.37) become

$$\theta_{ini} = (\Phi^T\Phi + \delta I)^{-1}\Phi^T Y \tag{4.38}$$

$$P_{ini} = (\Phi^T\Phi + \delta I)^{-1} \tag{4.39}$$

The regularization parameter δ is chosen to be small, say $\delta = 10^{-2} - 10^{-4}$, compared to the magnitude of the elements of Φ.

- ARLS

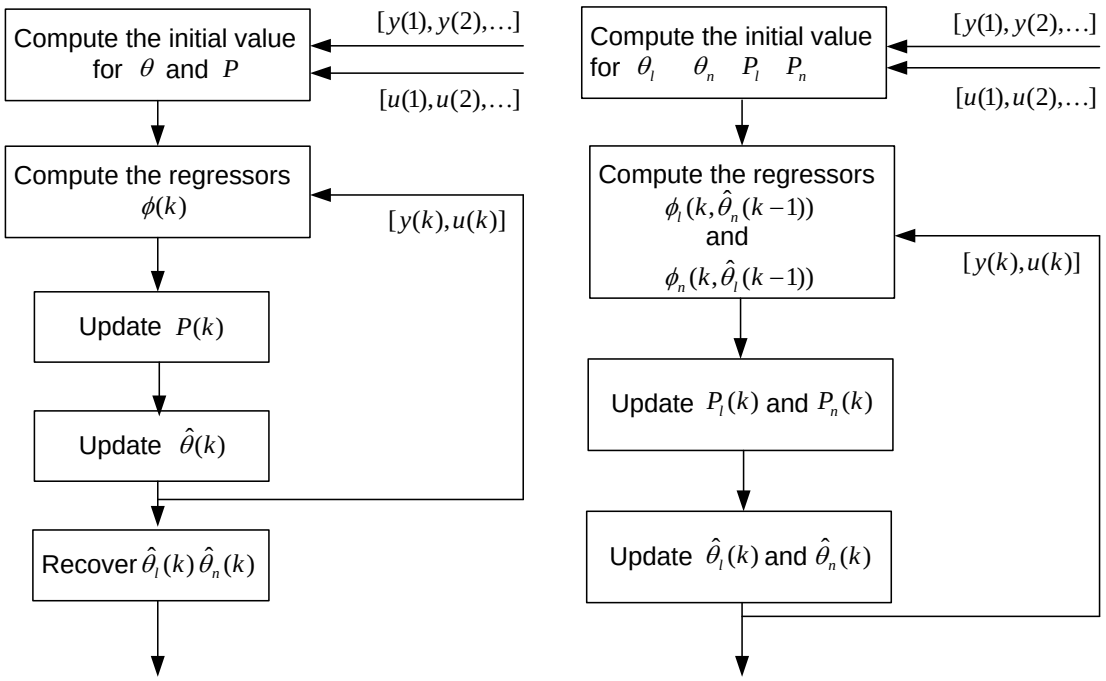
  For ARLS, the initial values are θ_l, θ_n, P_l and P_n. The initial values for θ_l and θ_n are found by applying rank-1 approximation, and then calculating Φ_l and Φ_n, where

$$\Phi_l = \begin{bmatrix} \phi_l^T(1,\theta_n) \\ \vdots \\ \phi_l^T(T_{\text{ini}},\theta_n) \end{bmatrix} \quad \text{and} \quad \Phi_n = \begin{bmatrix} \phi_n^T(1,\theta_l) \\ \vdots \\ \phi_n^T(T_{\text{ini}},\theta_l) \end{bmatrix}$$

  The initial values for P_l and P_n are therefore

$$P_l = (\Phi_l^T\Phi_l)^{-1} \quad \text{and} \quad P_n = (\Phi_n^T\Phi_n)^{-1}$$

  and again regularization may be applied to avoid ill-conditioning.



(a) RLS (b) ARLS

FIGURE 4.2: Flow Chart of both RLS and ARLS algorithms

4.3 Simulation Study

The two techniques are now compared in simulation across a number of criteria. Comparison is also made with their offline counterparts which, in both cases, involves exchanging the RLS update procedure for offline LS optimisation using full test data.

4.3.1 Numerical Example

The numerical example in [Boutayeb et al., 1996] is used, as it is highly relevant to the work reported in this paper

$$B(q) = q^{-1} + 0.6q^{-2} \tag{4.40}$$

$$A(q) = 1 - q^{-1} + 0.8q^{-2} \tag{4.41}$$

$$f(u) = 2.8u - 4.8u^2 + 5.7u^3 \tag{4.42}$$

The input signal used in [Boutayeb et al., 1996] is a zero mean white noise sequence, which is widely employed in recursive identification to guarantee persistent excitation. However, as previously noted, this is unsuitable for the present application, and will therefore be exchanged for a half cosine wave signal which has similar characteristics to signals used in rehabilitation (see [Hughes et al., 2009]). In order to make the half cosine wave signal persistently exciting, the diminishing excitation technique [Chen and Guo, 1991] has been applied:

$$u(k) = u_d(k) + \frac{\varepsilon(k)}{k^{\tau/2}} \tag{4.43}$$

where $u_d(k)$ is the designed input and $\varepsilon(k)$ is a bounded random sequence with $\tau > 0$ sufficiently small. The added measurement noise $v(k)$ is zero mean white noise such that the Signal-to-Noise Ratio

$$SNR = \left(\frac{var(y_{sig})}{var(y_{noi})} \right)^{1/2} \tag{4.44}$$

is equal to 10, 5 or 2. Here $y_{sig} = \frac{B(q)}{A(q)} w(k)$ is the noise-free output signal, $y_{noi} = \frac{1}{A(q)} v(k)$ is the correlated noise and $var(\cdot)$ the population variance of a finite-size sequence,

$$var(y) = \frac{1}{N-1} \sum_{t=1}^{N} (y_t - \bar{y})^2, \qquad \text{where} \quad \bar{y} = \frac{1}{N} \sum_{t=1}^{N} y_t \tag{4.45}$$

The input signals and the corresponding output signals with SNR=5 are given in Fig. 4.3.

4.3.2 Results

The two recursive algorithms, RLS and ARLS, are compared in terms of the following three aspects:

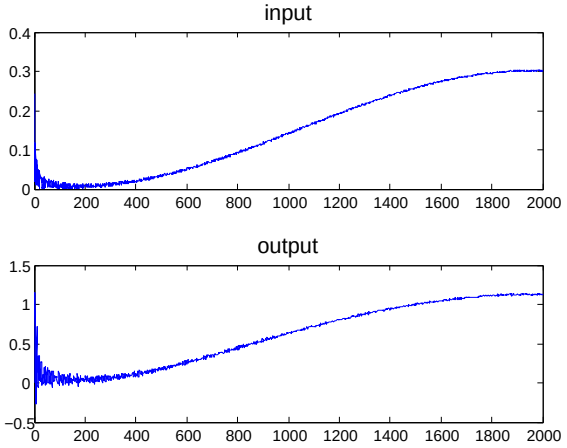


FIGURE 4.3: An example of the half cosine wave input and the corresponding output for the numerical example.

1. Error norm

   The error norm is the normalized error between the true values and the estimated values of the linear and nonlinear parameters, which is defined as:

$$Error\ Norm = \sqrt{\left(\frac{\|\theta_n - \hat{\theta}_n\|_2}{\|\theta_n\|_2}\right)^2 + \left(\frac{\|\theta_l - \hat{\theta}_l\|_2}{\|\theta_l\|_2}\right)^2}$$

   The recursive algorithms, together with their associated offline batch implementations, have been performed on 100 independent trials using different noise levels. The mean error norms of the updated parameter values at each time instant from the two recursive algorithms are traced in Fig. 4.4 and compared with the reference lines, that is, the mean error norms after 2000 samples from the two batch algorithms, LS and the first iterative algorithm (Iterative), developed in Chapter 3.

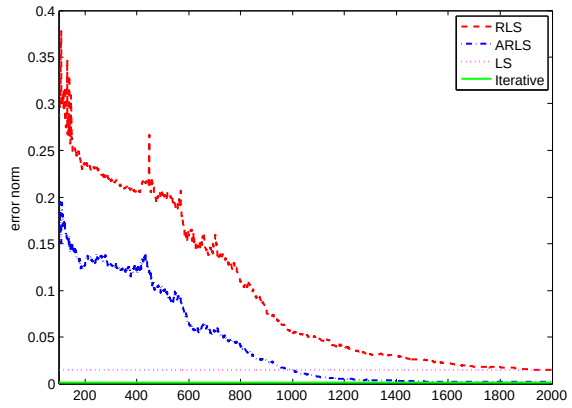
   Also the mean and standard deviation of the error norms after 2000 samples for 100 independent trials using different noise levels are listed in Table. 4.1.

TABLE 4.1: Numerical example: the mean and standard deviation of the error norms after 2000 samples for 100 independent trials using different noise levels (SNR=10, 5 and 2) from the two recursive algorithms (RLS and ARLS) and the two batch algorithms (LS and Iterative).

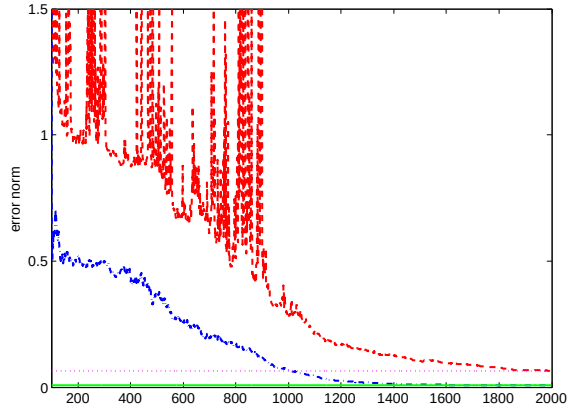
	Recursive		Batch	
	RLS	ARLS	LS	Iterative
SNR=10	0.0146±0.0102	0.0017±0.0010	0.0146±0.0102	0.0014±0.0008
SNR=5	0.0650±0.0420	0.0074±0.0041	0.0650±0.0420	0.0065±0.0034
SNR=2	0.7586±2.2713	0.0404±0.0253	0.7586±2.2713	0.0338±0.0224

2. Best Fit Rates

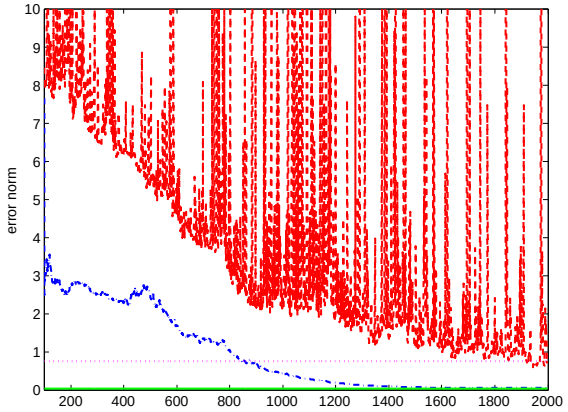
   In order to show how well the identified model can predict the output, Fig. 4.5 plots the measured output and the mean predicted output for 100 independent trials from two recursive algorithms after 2000 samples using different noise levels.



(a) SNR=10



(b) SNR=5

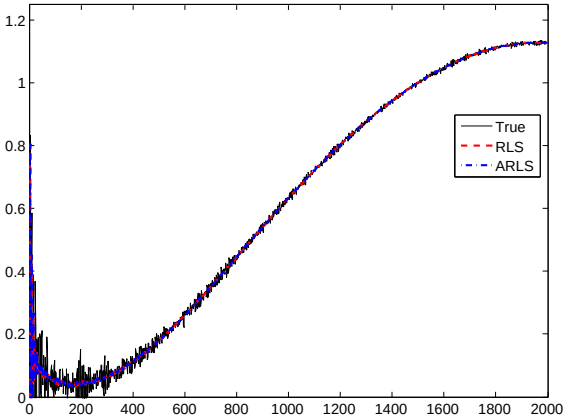


(c) SNR=2

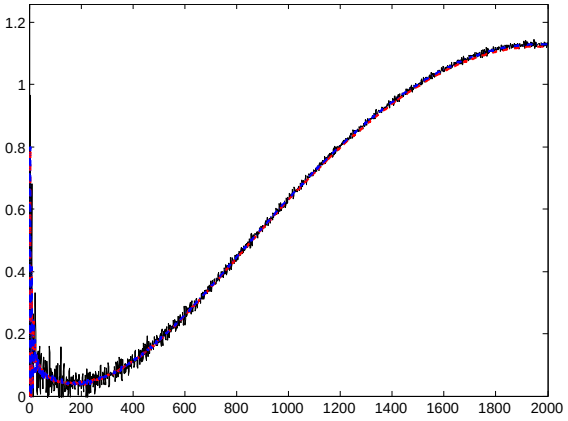
FIGURE 4.4: Numerical example: the mean error norms of the updated parameter values at each time instant for 100 independent trials using different noise levels (SNR=10, 5 and 2) from the two recursive algorithms (red dashed line for RLS and blue dash-dot line for ARLS) are compared with the mean error norms after 2000 samples from the two batch algorithms (magenta dotted line for LS and green solid line for Iterative).

Moreover, the mean and standard deviation of Best Fit rates after 2000 samples for 100 independent trials using different noise levels are listed in Table. 4.2.

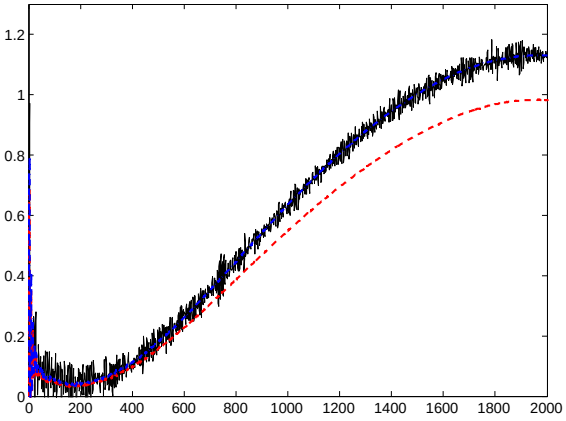
3. Convergence of parameter estimates



(a) SNR=10



(b) SNR=5



(c) SNR=2

FIGURE 4.5: Numerical example: the plots show the measured output and the mean predicted output for 100 independent trials using different noise levels (SNR=10, 5 and 2) from the two recursive algorithms (red dashed line for RLS and blue dash-dot line for ARLS) after 2000 samples. The x axis is the number of samples and the y axis is the output.

In order to show how fast the estimated values of the parameters converge to the true values, Fig. 4.6 plots the mean values of the updated nonlinear parameters for 100 independent trials using different noise levels.

TABLE 4.2: Numerical example: the mean and standard deviation of Best Fit rates after 2000 samples for 100 independent trials using different noise levels (SNR=10, 5 and 2) from the two recursive algorithms (RLS and ARLS) and the two batch algorithms (LS and Iterative).

	Recursive		Batch	
	RLS	ARLS	LS	Iterative
SNR=10	99.3354±0.3956	99.8009±0.0061	99.3354±0.3956	99.8011±0.0061
SNR=5	97.6243±1.5314	99.1991±0.0258	97.6243±1.5314	99.1998±0.0257
SNR=2	68.0062±35.3085	95.0197±0.1820	8.0062±35.3085	95.0225±0.1819

4. Effect of an abrupt change of the true model

   The results given above are from a time-invariant model and to determine how these two recursive algorithms track the time-variant model, an abrupt change in the true model after 2000 samples is introduced, and the nonlinear function becomes

$$f(u) = 2.8u - 5.1u^2 + 5.7u^3 \tag{4.46}$$

   where the coefficient of the term of the second degree changes from -4.8 to -5.1, which is such a slight change that it cannot be observed from the output plot, illustrated by Fig. 4.7.

   The convergence plots for the nonlinear parameter estimates from the two recursive algorithms are compared in Fig. 4.8(a) where $\lambda = 0.9993$ is chosen for RLS and $\lambda_l = 1$ and $\lambda_n = 0.9993$ for ARLS, and also the plot from ARLS has been magnified to show more clearly that the estimates converge to the true values after 5000 samples in Fig. 4.8(b).

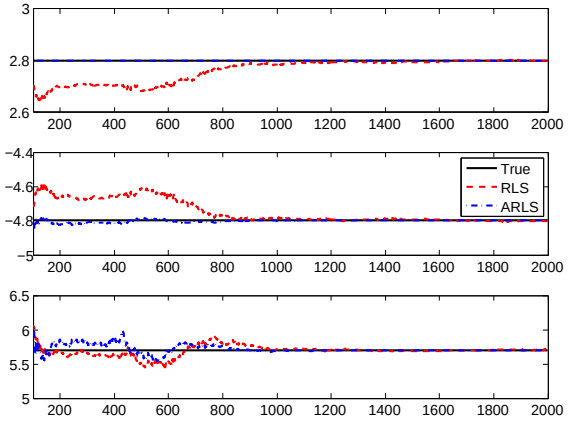
4.3.3 Discussion

ARLS algorithm is superior to the RLS algorithm in all the simulation results, especially in the noisy environment.

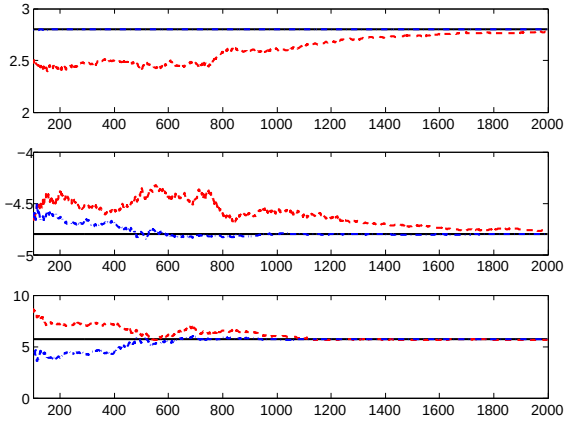
For ARLS, the error norms more quickly converge to lower values in all noise cases than RLS and even close to the batch algorithm which takes several iterations to optimize the parameters. This also can be illustrated by the convergence plot for the nonlinear parameter estimates, where the estimated values converge to the true value quickly even in noisy cases. When suffering from an abrupt change of the true model, ARLS still can quickly track the changed model.

RLS performs comparable well with ARLS at low noise levels. However, RLS takes quite a long time or even fails to converge to the true value for the noisy measurements. Furthermore, RLS is not good at tracking the time-varying systems based on the above simulation results, where the parameters estimates are subject to a greater oscillation and converge to the true values in a longer period of time after a slight change in the simulated model.

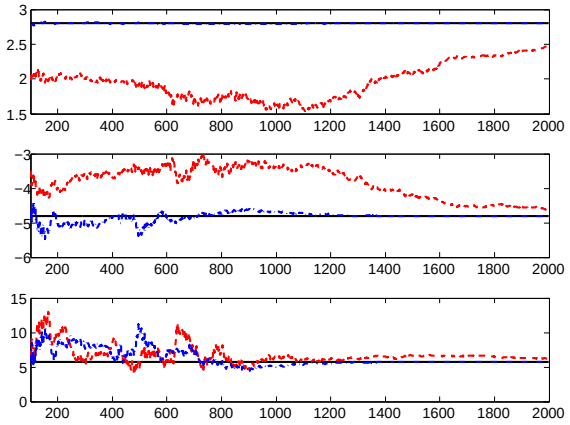
It is noted that the electrically stimulated muscle system is time-varying and the experimental results from such a system are very noisy. Thus, it is expected that ARLS will provide better



(a) SNR=10



(b) SNR=5



(c) SNR=2

FIGURE 4.6: Numerical example: the mean of the updated values for 100 independent trials using different noise levels (SNR=10, 5 and 2) from the two recursive algorithms (red dashed line for RLS and blue dash-dot line for ARLS) are compared with the true values (black solid line) of the nonlinear parameters. The x axis is the number of samples and the y axis is the value of the parameter.

performance than RLS. It also should be noted that the simulation with added white noise does not reflect the type of noise that might be expected during the FES application.

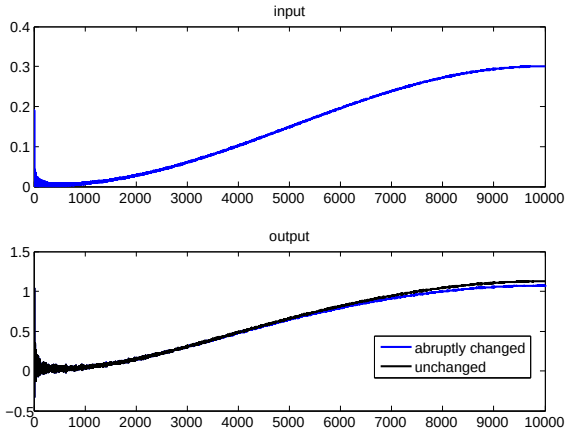


FIGURE 4.7: The half cosine wave input and the corresponding output when the model is changed after 2000 samples. The x axis is the number of samples and the y axis is the input (upper plot) or the output (lower plot).

4.4 Application to Electrically Stimulated Muscle

In this section, the recursive algorithms developed above are applied to online identification of the response of electrically stimulated muscle.

4.4.1 Experiment Set-up

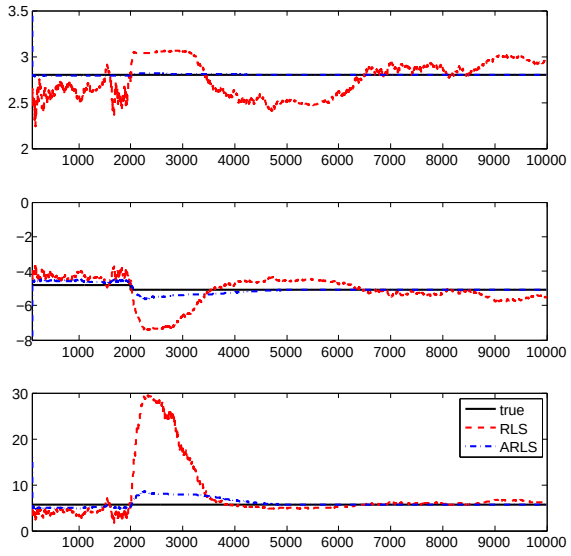
Recursive identification tests were performed on a single unimpaired subject, and took place on several independent days. The position of the robotic arm was then fixed at an elbow extension angle of approximately $\pi/2$ rads using a locking pin. This removes the non-isometric components of the biomechanical model, so that the resulting system corresponds to a Hammerstein structure (comprising the muscle model with the addition of passive elastic torque from the remaining arm which may also vary in time). The model's input is the stimulation pulsewidth, and its output is the torque about the elbow. The recursive identification tests last for 10 min, comprising 10 repeated waves of either a half-cosine function, or a staircase signal, added to which the diminishing excitation technique has been used to make the input signals persistently exciting. The two kinds of input signal have similar characteristics to those used in rehabilitation (see [Hughes et al., 2009]) and the corresponding output signals are plotted in Fig. 4.9.

4.4.2 Results

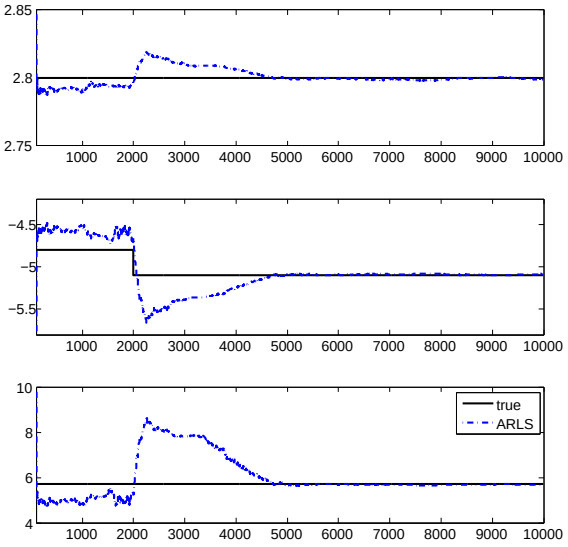
Here, the two recursive algorithms, RLS and ARLS, are compared in the following aspects:

1. One-step ahead prediction

   In order to evaluate the accuracy of the recursive algorithms, the measured torque outputs y are compared with the one-step ahead predicted outputs $\hat{y}$ in term of the Best Fit rate,



(a) RLS and ARLS



(b) close-up of ARLS trajectory

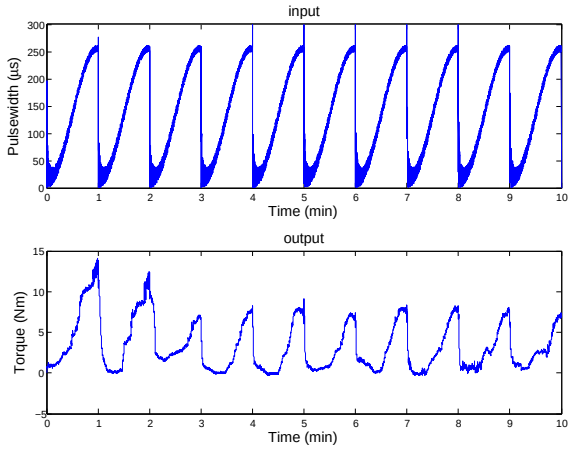
FIGURE 4.8: Numerical example with an abrupt change after 2000 samples: the time trajectory of the estimated nonlinear parameter values from the two recursive algorithms (red dashed line for RLS and blue dash-dot line for ARLS) at SNR=10. The x axis is the number of samples and the y axis is the value of the parameter.

defined in (3.35) and $\hat{y}$ is defined as

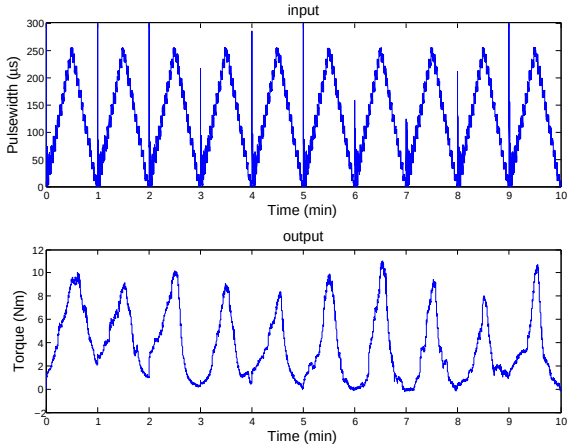
$$\hat{y}(k+1) = G(q, \hat{\theta}_l(k)) f(u, \hat{\theta}_n(k))$$

which is a one-step ahead prediction, using the updated model at the time instant k to predict the output at the next time instant k+1.

Table 4.3 lists the Best Fit rates for half cosine and staircase wave inputs respectively, and considers both the whole 10-minute dataset and the first 1-minute dataset, the latter of which contains less time-varying information. The corresponding plots are shown in Fig.



(a) half of the cosine wave



(b) staircase wave

FIGURE 4.9: The input and output signals for recursive identification tests

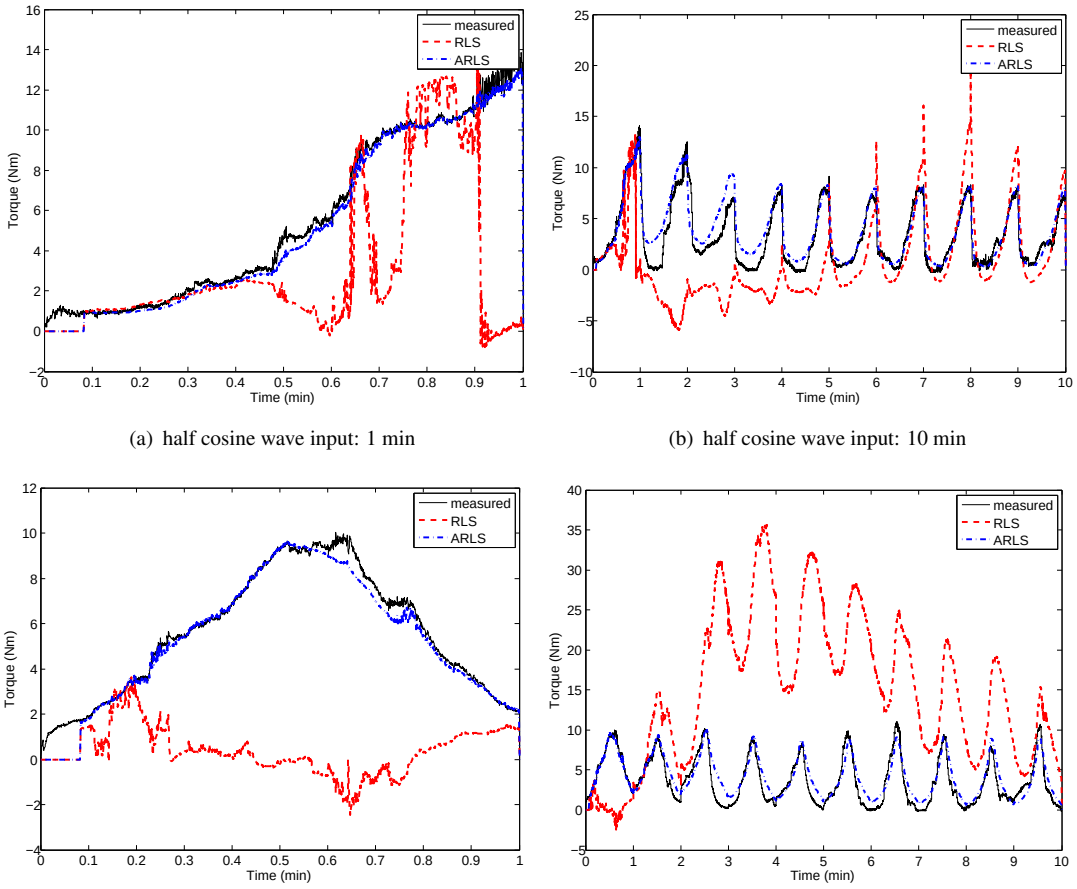
4.10.

TABLE 4.3: Muscle tests: Best Fit rates between the measured outputs and the one-step ahead predicted outputs from the two recursive algorithms, RLS and ARLS

	half cosine wave input		staircase wave input	
	RLS	ARLS	RLS	ARLS
1 min	-10.0244	87.9188	-130.4187	80.3162
10 min	-52.3874	61.3267	-408.2148	57.4049

2. Long-period prediction

In order to demonstrate the predictive ability for the longer period, the two recursive algorithms together and their corresponding offline batch implementations have been applied to the first 3, 4, 5, 6 and 7 minutes of the data respectively, and the resulting models then used to predict the corresponding outputs for the remaining time period. The Best Fit rates for identification and prediction are listed in Table 4.4. Fig. 4.11 shows the measured outputs and the modelled outputs in the case of the first 5 minutes for the identification phase



(a) half cosine wave input: 1 min

(b) half cosine wave input: 10 min

(c) staircase wave input: 1 min

(d) staircase wave input: 10 min

FIGURE 4.10: Muscle Tests: the plots show the measured outputs and the one-step ahead predicted outputs from the two recursive algorithms, RLS and ARLS.

as it is representative of all the results obtained.

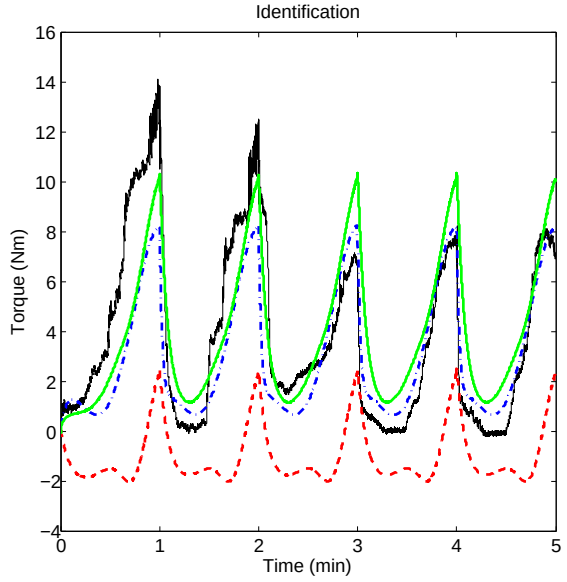
3. Computational time

   Since the algorithms are intended for online implementation in real-time, their computation time is an important factor. The time taken to perform a single updating step for both recursive algorithms is listed in Table. 4.5.

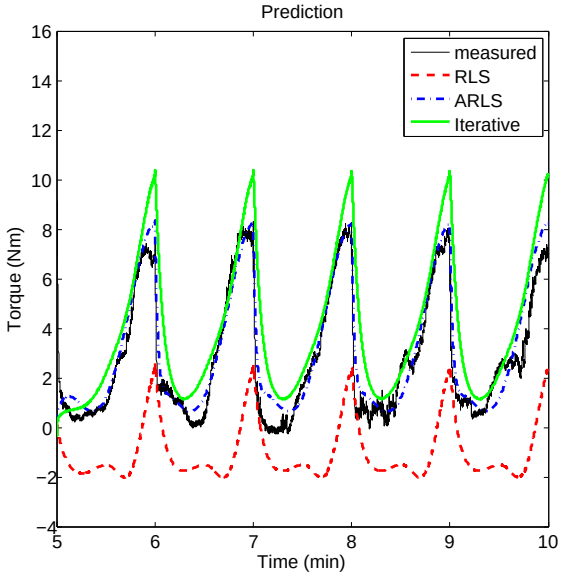
4.4.3 Discussion

- Batch algorithms vs recursive algorithms

  Batch algorithms are offline and use all the data at hand to perform the identification process in order to find the best model according to the minimization criterion. Table 4.6 shows the identification results for the two batch algorithms: LS and iterative respectively. It is clear that LS cannot deal with the noisy and time-varying experimental data and the iterative algorithm greatly improves the Best Fit rates. For 1 minute data, iterative algorithm achieves around 85% fitting rates, which is in agreement with the results



(a) Identification



(b) Prediction

FIGURE 4.11: Muscle Tests: the plots show the measured outputs and the modelled outputs for identification validation data, respectively, in the case of both halves of the data for identification and prediction.

reported in Chapter 3, where a 20 second test was used. However, for the 10 minute data, which contains more time-varying information, even the iterative algorithm cannot find a time-invariant model to fit all the data and it only yields 12% fitting rate for the staircase input.

It follows from Table 4.4 that the iterative algorithm provides the best identification fitting rates in all cases but performs very poorly for prediction. However, ARLS is very good at prediction and gives even higher fitting rates for prediction compared to identification. The results of Fig. 4.11 reflect that the iterative algorithm uses all the identification

TABLE 4.4: Muscle tests: Identification and validation Best Fit rate (%).

		Recursive		Batch	
		RLS	ARLS	LS	Iterative
first 3 min	Identification	-115.2605	40.0999	-115.2605	46.4750
next 7 min	Prediction	-119.4234	30.4174	-119.4234	-0.1979
first 4 min	Identification	-90.0323	38.0301	-90.0323	40.5572
next 6 min	Prediction	-100.0545	60.3873	-100.0545	19.5024
first 5 min	Identification	-60.7013	40.6237	-60.7013	42.2075
next 5 min	Prediction	-70.6340	66.5179	-70.6340	29.9268
first 6 min	Identification	-10.9030	40.7501	-10.9030	41.7214
next 4 min	Prediction	-7.5807	69.0567	-7.5807	37.4819
first 7 min	Identification	23.4563	44.5587	23.4563	45.0254
next 3 min	Prediction	17.2580	63.8627	17.2580	40.9369

TABLE 4.5: Muscle tests: computational time in seconds for a single updating step for the two recursive algorithms: RLS and ARLS

	RLS	ARLS
computational time	0.0019	1.0989×10^{-4}

data to calculate the best model, which, due to the time-varying properties of the system, produces an identified model which may be interpreted as an 'average' response, see Fig. 4.11(a). However, since ARLS updates the estimated model so that it is responsive to changes in underlying dynamics, the model produced after 5 min, even when it has not corresponded with particularly high fitting rates for the past data, is the best model to predict the future output, as illustrated by Fig. 4.11(b). Finally, the batch algorithm is not good at identification of long-period data from a time-varying system.

TABLE 4.6: Muscle tests: Best Fit rates between the measured outputs and modeled outputs from two batch algorithms: LS and Iterative

	half cosine wave input		staircase wave input	
	LS	Iterative	LS	Iterative
1 min	-49.5367	86.9901	-133.4340	85.2571
10 min	23.8095	44.6698	-46.7207	12.2363

On the other hand, batch algorithms are computationally heavy and not suitable for real-time implementation, as illustrated by Table. 4.7, where the computational times for 1 min and 10 min data from the two batch algorithms, LS and Iterative, are listed. The computational time grows considerably with the increase in samples, so that there comes a point when calculations cannot be completed before the arrival of new data.

- RLS vs ARLS

  From above analysis, it is necessary to perform recursive rather than batch identification

TABLE 4.7: Muscle tests: computational time for 1 min and 10 min data from the two batch algorithms, LS and Iterative, in seconds

	LS	Iterative
1-min	0.1155	1.9881
10-min	28	70

for the experimental data. Here the two recursive identification algorithms are applied, RLS and ARLS. For both cases, they first use several samples to generate an initial estimate, less than 0.1 minutes of data, and then update the linear and nonlinear parameters at each time instant.

It is clear that ARLS is far superior in this respect to RLS, which is in accord with the expected outcome after the simulation study. In a lot of cases, RLS will generate negative Best fit rates, which means the predicted values are even worse than a straight line, standing for the average value of the output. For the noisy experimental data and slowly time-varying muscle system, ARLS is the best choice. For 1 minute data, one-step ahead prediction can track the output well, shown in Fig. 4.10(a) and 4.10(c) and for 10 minute data, it also can capture long term variation in the muscle properties, as illustrated by Fig. 4.10(b) and 4.10(d).

Moreover, ARLS is even faster than RLS, because ARLS splits the algorithm into two parallel ones, each of which entails low-dimensional matrix multiplication.

Another advantage of ARLS over RLS is that ARLS has two separate weighting parameters for linear and nonlinear parameters, λ_l and λ_n. In the real muscle system, the linear and nonlinear parameters represent two different mechanisms (muscle activation and recruitment respectively) which change over time at different rates. The ability to choose individual weighting parameters for each mechanisms provides clear selection and performance advantages over a single λ parameter.

In the previous recursive process, the weighting parameters λ, λ_l and λ_n are fixed at 1, and the implications of this choice are now considered using Tables 4.8 and 4.9. For RLS, there is no improvement when tuning the λ parameter, while for ARLS, the fitting rate reaches 70% for $\lambda_l = 0.9995$ and $\lambda_n = 0.9997$.

TABLE 4.8: 10 min data of half cosine wave input: the Best Fit rates between the measured outputs and the one-step ahead predicted outputs from RLS with difference choices of λ

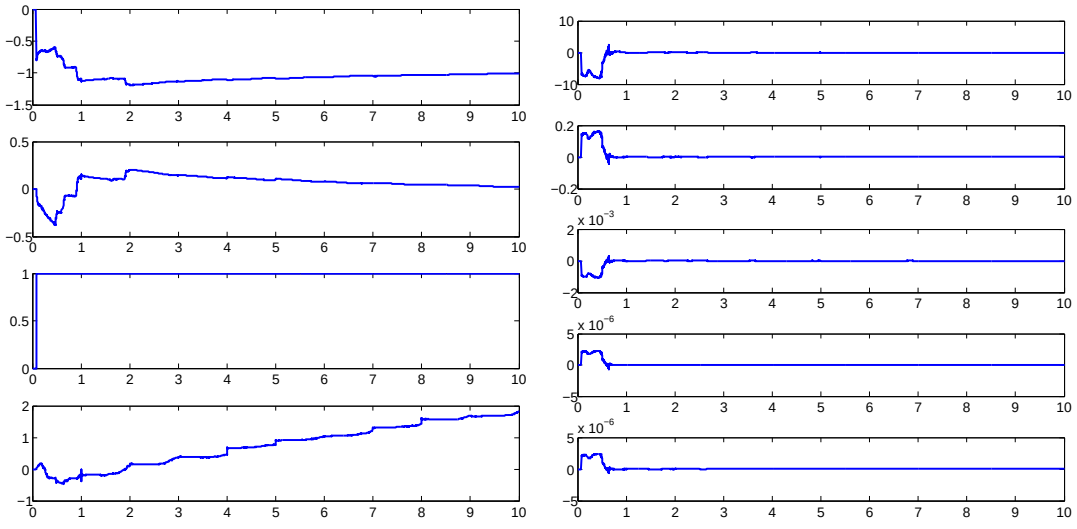
λ	Best Fit Rate (%)
1	-52.3874
0.9999	-109.1885
0.9998	-141.6934
0.9997	-103.2831
0.9990	-64.0752

TABLE 4.9: 10 min data of half of cosine wave input: the Best Fit rates between the measured outputs and the one-step ahead predicted outputs from ARLS with difference choices of λ_l and λ_n

λ_l	λ_n	Best Fit Rate (%)
1	1	61.3267
0.9999	0.9999	63.6053
0.9998	0.9999	65.8187
0.9997	0.9999	67.6394
0.9996	0.9998	68.7207
0.9995	0.9997	70.8437
0.9994	0.9996	43.0805

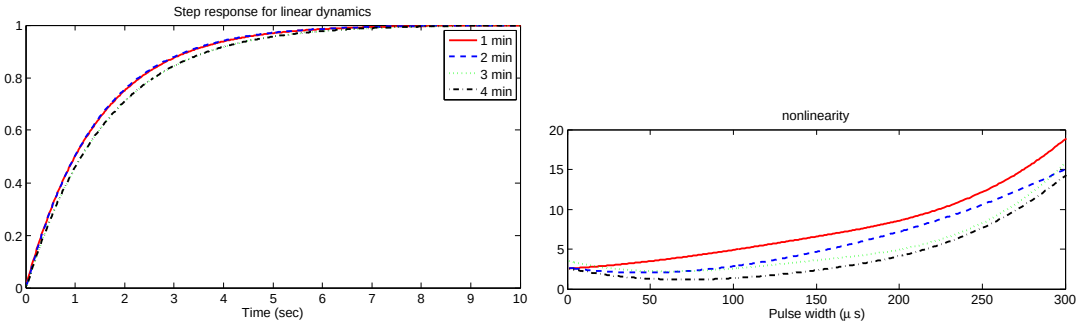
- Time-variance of the muscle model

  Fig. 4.12 shows the time trajectory of the estimated values for the linear and nonlinear parameters from ARLS and the time variation of the estimated values of all the parameters, except the third linear parameter, which was normalized as 1 during the identification. In Fig. 4.12, it is a little difficult to observe with the naked-eye the variance of the estimated nonlinear parameters and the variance of the absolute values of the estimated linear parameters is greater. In order to provide visual aid, the step response for the identified linear block and IRC for the identified nonlinearity are plotted for four time instants at equal intervals, 1, 2, 3 and 4 mins, in Fig. 4.13, where the slower step response and lower muscle gain are observed, due to fatigue. This illustrates the variance of the underlying physiological mechanisms of the muscles.



(a) linear parameters (b) nonlinear parameters

FIGURE 4.12: Muscle tests: the time trajectory of the estimated values of the linear and nonlinear parameters from ARLS.



(a) step response for linear block

(b) nonlinearity

FIGURE 4.13: Muscle tests: the step response for identified linear block and identified nonlinearity are plotted along the time from ARLS

Chapter 5

Functional Electrical Stimulation Control

In this chapter the identified muscle models that have been developed will be directly used in FES control schemes for electrically stimulated muscle under isometric conditions. The experimental tests will be undertaken using the same equipment employed in the program of stroke rehabilitation, in order to confirm performance under conditions as similar as possible to those used clinically. In particular, FES is applied to the subject's triceps muscle when the arm is strapped to the robotic arm, and the task is to repeatedly track a reference trajectory of torque about the elbow. As explained, iterative learning controllers will be used since the repeated nature of the task, together with resetting between trials, exactly corresponds to the ILC methodology. Since its formal conception [Arimoto et al., 1984] more than two decades ago, ILC theory has been developed and applied to a huge range of systems and applications. The basic concept of ILC is based on the notion that the performance of a system conducting the same task many times can be improved by using information gathered from previous executions. In this Chapter, the plant model will be firstly described and then two nonlinear ILC approaches will be theoretically explained and implemented. Later on, their trial-dependent counterparts are investigated and compared in terms of their performance in simulation and experimental tests. Finally, online identification algorithms developed in the last Chapter will be used to produce adaptive ILC methods, which will also be implemented on the simulated muscle system.

5.1 Hammerstein Plant Description

As in the previous tests, the plant is the isometric muscle system which means that the muscle will held at a fixed length during all experiments. The input of the plant is the stimulation pulsewidth, u, in μs and the output is the torque generated at the elbow joint, y. The Hammerstein structure is used as a model for the plant, which consists of a static nonlinear block followed by

a linear dynamics, see Fig. 3.1. In detail, the static nonlinearity is represented by a cubic spline

$$f(u) = \sum_{i=1}^{m-2} \beta_i |u - u_{i+1}|^3 + \beta_{m-1} + \beta_m u + \beta_{m+1} u^2 + \beta_{m+2} u^3, \tag{5.1}$$

$u_{min} = u_1 < u_2 < u_3 < \cdots < u_m = u_{max}$ are the spline knots and the linear block is a discrete-time transfer function

$$G(q) = \frac{b_0 q^{-d} + b_1 q^{-(d+1)} + \cdots + b_n q^{-(n+d)}}{1 + a_1 q^{-1} + \cdots + a_l q^{-l}} \tag{5.2}$$

For implementation in the ILC framework which follows, it is transformed to a state-space model:

$$\begin{aligned} x(k+1) &= Ax(k) + Bf(u(k)) \\ y(k) &= Cx(k) \qquad\qquad x(0) = x_0 \end{aligned} \tag{5.3}$$

where

$$A = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{l-1} & -a_l \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & & 1 & 0 \end{bmatrix} \tag{5.4}$$

$$B(k) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5.5}$$

$$C(k) = \begin{bmatrix} 0 & \cdots & 0 & b_0 & \cdots & b_n \end{bmatrix} \tag{5.6}$$

and

$$f(u(k)) = \sum_{i=1}^{m-2} \beta_i |u(k) - u_{i+1}|^3 + \beta_{m-1} + \beta_m u(k) + \beta_{m+1} u^2(k) + \beta_{m+2} u^3(k) \tag{5.7}$$

Moreover, the model is not limited to the time-invariant case. In Chapter 4, a slowly time-varying Hammerstein model is assumed, which is the optimal solution after reviewing the defects of the Hammerstein structure as the most popular candidate for modelling the isometric muscle dynamics in Section 2.1.5. Thus, a slowly time-varying state-space model is considered in later sections and a subscript or an index is used to indicate the time-varying parameters.

The system is defined over the finite time interval $k \in [0, 1, 2, \ldots, N-1]$. The single input and single output sequences are given by the vectors

$$u = [u(0), u(1), \ldots, u(N-1)]^T \tag{5.8}$$

$$y = [y(0), y(1), \ldots, y(N-1)]^T \tag{5.9}$$

The goal of the standard ILC framework is to construct a series of input sequences which drive

the system to track a reference sequence

$$y_d = [y_d(0), y_d(1), \ldots, y_d(N-1)]^T \tag{5.10}$$

Let u_j and y_j be the input and output vectors respectively on the j^{th} trial, and the tracking error be defined as $e_j = y_d - y_j$. Then the ILC problem is to find a sequence of control inputs that satisfies

$$\lim_{j\to\infty} \|e_j\| = 0, \qquad \lim_{j\to\infty} \|u_j - u_d\| = 0 \tag{5.11}$$

where u_d is the ideal control input.

Over each trial the relationship between the input and output time-series can be expressed by the following algebraic functions

$$\begin{aligned}
y(0) &= Cx(0) = g_0(x(0)) \\
y(1) &= Cx(1) = C(Ax(0) + Bf(u(0))) \\
&= g_1(x(0), u(0)) \\
y(2) &= Cx(2) = C(Ax(1) + Bf(u(1))) \\
&= CA(Ax(0) + Bf(u(0))) + CBf(u(1)) \\
&= g_2(x(0), u(0), u(1)) \\
&\vdots \\
y(N-1) &= Cx(N-1) = C(Ax(N-2) + Bf(u(N-2))) \\
&= g_{N-1}(x(0), u(0), u(1), \ldots, u(N-2))
\end{aligned} \tag{5.12}$$

This allows a rigid connection to be made between ILC and the techniques from nonlinear optimization which are employed in the next sections, and provides an analytic basis for their extension to ILC. Using the relations (5.12), the system (5.3) can be represented by the algebraic function $g(\cdot)$

$$y = g(u), \quad g(\cdot) = [g_0(\cdot), g_1(\cdot), g_2(\cdot), \ldots, g_{N-1}(\cdot)]^T \tag{5.13}$$

This plant description is now used to demonstrate the implementation of two existing ILC approaches: Gradient descent ILC and Newton method based ILC.

5.2 Control Schemes

The standard ILC set-up is shown in Fig. 5.1, which comprises the update equations:

$$u_{j+1} = u_j + Le_j \tag{5.14}$$

$$e_j = y_d - y_j \tag{5.15}$$

where y_d is the reference trajectory, u_j, y_j and e_j are the input applied on the jth trial, the corresponding plant output and the error, respectively and L is the learning operator. In terms of designing the learning operator, ILC algorithms can be roughly divided into two categories. The first category does not assume any prior knowledge of the plant model and the corresponding controllers generally comprise a simple mathematical operation, such as a derivative or time-shift. The second uses knowledge of the plant to seek to guarantee favourable convergence or robustness properties. The model-based schemes that will be considered have both been used in the stroke rehabilitation project, and so are known to be suitable for use in the intended application area. These are Gradient descent ILC and Newton method based ILC. In the following two sections, the detailed theory about these two algorithms will be given.

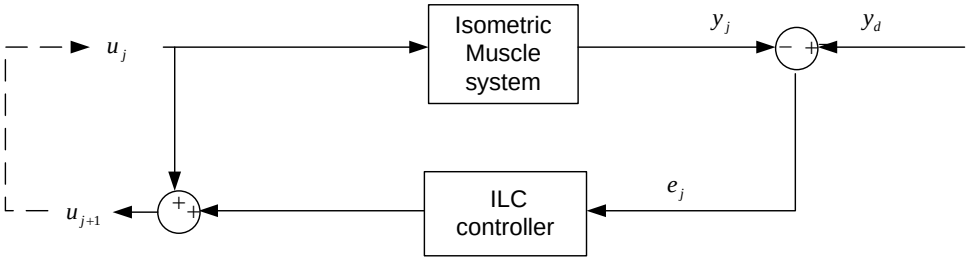


FIGURE 5.1: Block diagram of control scheme

5.2.1 Gradient Descent ILC

ILC can be considered as an iterative numerical solution to the problem

$$\min_u \|e\|_2^2 = \|y_d - g(u)\|_2^2 = p(u) \tag{5.16}$$

The gradient descent method is a nonlinear minimization technique which solves this iteratively with the update

$$\begin{aligned} u_{j+1} &= u_j - \frac{\varepsilon}{2}\nabla p(u_j) \\ &= u_j + \varepsilon \nabla g(u_j)(y_d - g(u_j)) \\ &= u_j + \varepsilon g'(u_j)^T (y_d - g(u_j)) \end{aligned} \tag{5.17}$$

where ε is a positive scalar gain. In the ILC framework the term $y_d - g(u_j)$ is replaced by the experimentally obtained error signal, e_j, to produce

$$u_{j+1} = u_j + \varepsilon g'(u_j)^T e_j \tag{5.18}$$

The derivative $g'(u_j)$ is equivalent to the system linearization around u_j and is represented by the $N \times N$ matrix

$$g'(u_j) = \begin{bmatrix} \frac{\partial g_0}{\partial u_j(0)} & \frac{\partial g_0}{\partial u_j(1)} & \cdots & \frac{\partial g_0}{\partial u_j(N-1)} \\ \frac{\partial g_1}{\partial u_j(0)} & \frac{\partial g_1}{\partial u_j(1)} & \cdots & \frac{\partial g_1}{\partial u_j(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{N-1}}{\partial u_j(0)} & \frac{\partial g_N}{\partial u_j(1)} & \cdots & \frac{\partial g_{N-1}}{\partial u_j(N-1)} \end{bmatrix} \tag{5.19}$$

since, from (5.12), $g_s(\cdot)$ is not a function of $u_j(t),\ t > s$, this can be simplified to

$$g'(u_j) = \begin{bmatrix} \frac{\partial g_0}{\partial u_j(0)} & 0 & 0 & 0 & 0 \\ \frac{\partial g_1}{\partial u_j(0)} & \frac{\partial g_1}{\partial u_j(1)} & 0 & 0 & 0 \\ \frac{\partial g_2}{\partial u_j(0)} & \frac{\partial g_2}{\partial u_j(1)} & \frac{\partial g_2}{\partial u_j(2)} & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{N-1}}{\partial u_j(0)} & \frac{\partial g_{N-1}}{\partial u_j(1)} & \frac{\partial g_{N-1}}{\partial u_j(2)} & \cdots & \frac{\partial g_{N-1}}{\partial u_j(N-1)} \end{bmatrix} \tag{5.20}$$

Here, firstly, a time-invariant Hammerstein plant is considered and $g'(u_j) =$

$$\begin{bmatrix} 0 & 0 & 0 & \cdots 0 \\ CB\frac{\partial f}{\partial u_j(0)} & 0 & 0 & \cdots 0 \\ CAB\frac{\partial f}{\partial u_j(0)} & CB\frac{\partial f}{\partial u_j(1)} & 0 & \cdots 0 \\ \vdots & \vdots & \vdots & \ddots \vdots \\ CA^{N-2}B\frac{\partial f}{\partial u_j(0)} & CA^{N-3}B\frac{\partial f}{\partial u_j(1)} & CA^{N-4}B\frac{\partial f}{\partial u_j(2)} & \cdots 0 \end{bmatrix} \tag{5.21}$$

where A, B and C are the state, input and output matrices of the nominal model defined in (5.4,5.5,5.6) and $\frac{\partial f}{\partial u_j(k)}$ is the derivative of $f(\cdot)$, defined in (5.7), with respect to u at the point $u = u_j(k)$.

For an iteration-invariant plant it can be shown that there exists a scalar parameter $\varepsilon > 0$ in (5.18) which guarantees convergence to a local minimum error norm (global for the LTI case). Moreover, for the LTI case it is shown in [Owens et al., 2009], that ε can be chosen before each trial to minimise the subsequent error norm, using the iteration-varying value

$$\varepsilon_{j+1} = \frac{\|g_e^T e_j\|^2}{\|g_e g_e^T e_j\|^2} \tag{5.22}$$

where $g_e =$

$$\begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-2}B & CA^{N-3}B & CA^{N-4}B & \cdots & D \end{bmatrix} \tag{5.23}$$

which contains the Markov parameters of the nominal model, which correspond to components of the discrete FIR of the plant. This algorithm has been found to possess extremely desirable robust convergence properties and has been tested using a gantry robot facility [Ratcliffe et al., 2008], a non-minimum phase testbed [Freeman et al., 2007b], and in our project [Freeman et al., 2009a]. When testing on a single unimpaired human subject for a slow and fast trajectories, the phase-lead algorithm and adjoint algorithm (called Gradient Decent ILC in our thesis) show comparable good performances with the far more complex norm-optimal ILC (NOILC) algorithm, see Fig. 5.2.

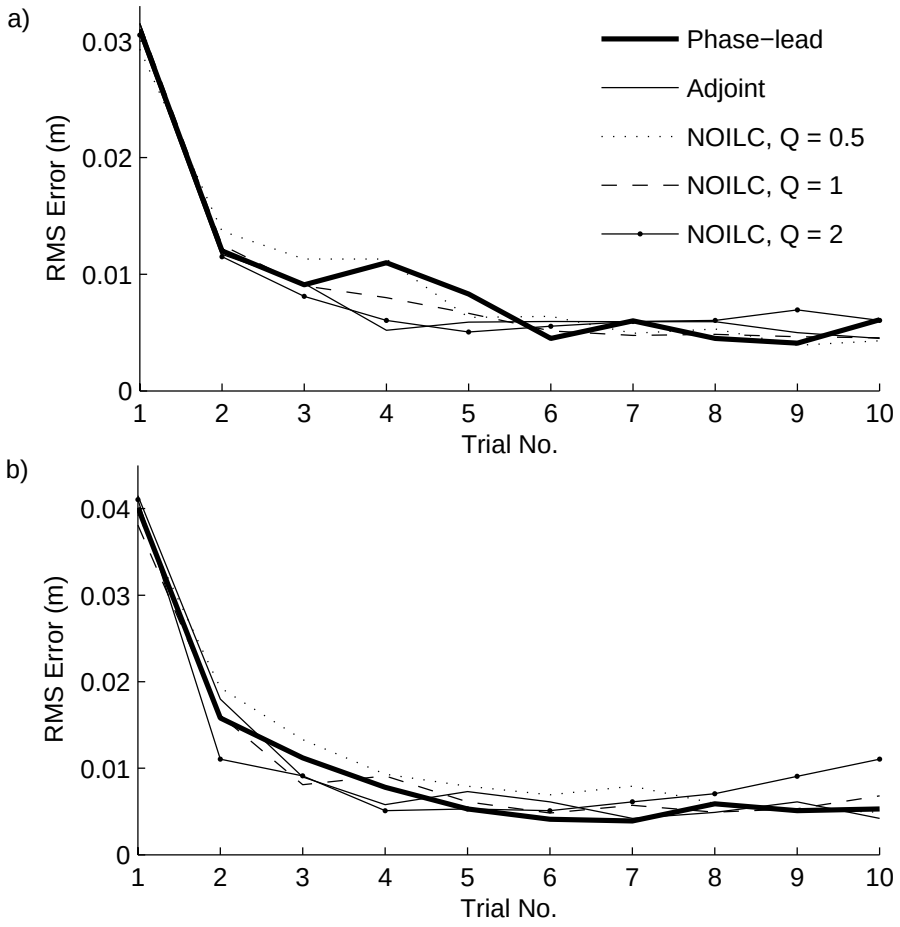


FIGURE 5.2: Single subject comparison of phase-lead and adjoint (Gradient Decent in our thesis) algorithms with NOILC for a) the slow trajectory, and b) the fast trajectory. [Freeman et al., 2009a]

5.2.2 Newton Method Based ILC

This approach is based on the Newton method, which is perhaps the best known method for finding successively better approximations to the roots of a real-valued function. Given a function $f(x)$ and its derivative $f'(x)$, an estimation of the roots can be obtained iteratively by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{5.24}$$

where n is the iteration number. This method converges very fast since its rate is quadratic rather than linear but it is only feasible when the inverse of $f'(x_n)$ exists and also it requires efficient computational implementation of inversion.

When this advanced iterative optimization technique is applied to (5.14,5.15), the corresponding learning gain L is given by

$$\begin{aligned} L &= \left(\frac{\partial(y_d - g(u_j))}{\partial u_j}\right)^{-1} \\ &= g'(u_j)^{-1} \end{aligned} \tag{5.25}$$

which results in

$$u_{j+1} = u_j + g'(u_j)^{-1} e_j \tag{5.26}$$

The next step is the calculation of the derivative $g'(\cdot)$ and then the inverse $g'(\cdot)^{-1}$ if they exist. In order to avoid the problem of infeasibility, ill-conditioning, or the presence of high amplitudes/frequencies in the update, another ILC scheme is used to find the solution. This is given as follows: firstly, use z_{j+1} to substitute $g'(u_j)^{-1} e_j$ in (5.26)

$$u_{j+1} = u_j + z_{j+1} \tag{5.27}$$

then z_{j+1} can be computed by solving the equation

$$g'(u_j) z_{j+1} = e_j \tag{5.28}$$

Because $g(\cdot)$ represents the nonlinear dynamic system, the derivative $g'(u_j)$ is equivalent to the linearization of the system about u_j. Then solving (5.28) becomes finding the z_{k+1} that drives the linearized system $g'(u_j)$ to track the desired output e_j, which can be achieved by any globally convergent ILC algorithm for LTV systems.

Specifically, consider the Hammerstein state-space plant in (5.3), and assume the parameters are time-invariant. In this case the Newton method based ILC can be summarized as:

On the jth trial, run the system on the current trial input u_j and record the output y_j and the tracking error $e_j = y_d - y_j$.

1. System linearization

The linearization of the system is

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{A}(k)\tilde{x}(k)+\tilde{B}(k)\tilde{u}(k) \\ \tilde{y}(k) &= \tilde{C}(k)\tilde{x}(k) \end{aligned} \tag{5.29}$$

where

$$\tilde{A}(k) = A = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{l-1} & -a_l \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & & 1 & 0 \end{bmatrix}$$

$$\tilde{B}(k) = B \times \frac{\partial f}{\partial u_j(k)} = \begin{bmatrix} \frac{\partial f}{\partial u_j(k)} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\tilde{C}(k) = C = \begin{bmatrix} 0 & \cdots & 0 & b_0 & \cdots & b_n \end{bmatrix}$$

2. ILC to find z_{j+1}

   For the linear time-varying system (5.29), a linear time-varying ILC algorithm, such as NOILC, is adopted to find the control input z_{j+1} in order to track the reference signal e_j.

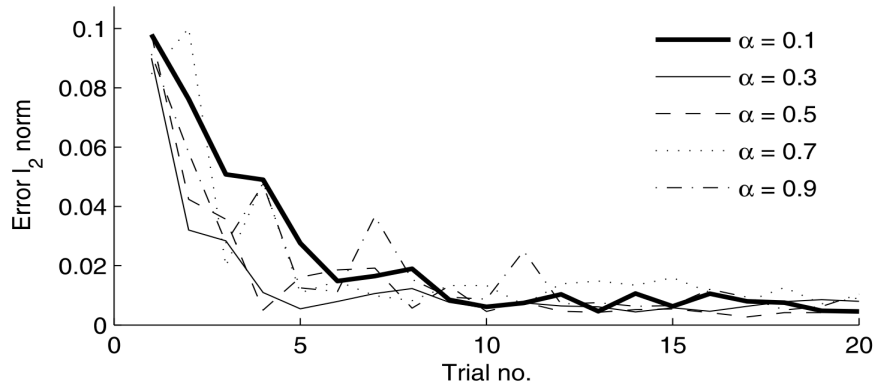
3. Get the control input u_{j+1}

   After calculating z_{j+1}, the control input for the next trial is updated by (5.27).

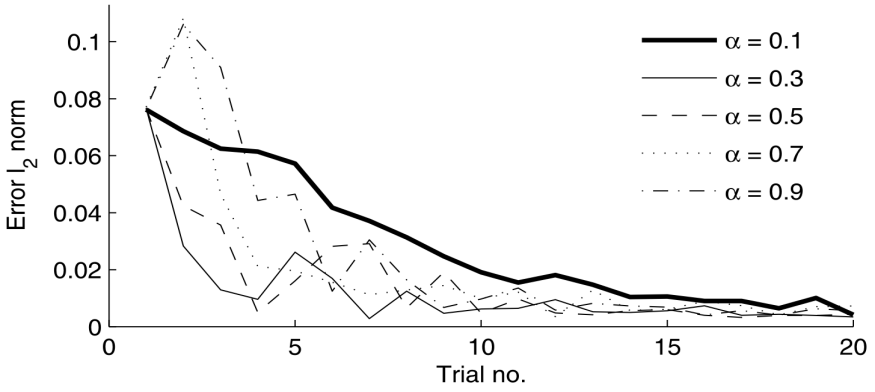
This algorithm has been implemented in our application in [Davies et al., 2008], where through simulation studies and experimental evaluation, it was ascertained that Newton Method Based ILC algorithm provided a good improvement from Gradient Descent ILC in terms of reduction of the tracking error norm as the number of trials increased, see Fig. 5.3.

5.3 Adaptive Control Scheme

From inspection of the general class of ILC algorithm considered, the performance of the controller clearly depends on three factors, the physical plant (as opposed to the plant model), the desired output, and the learning gain. The latter operator is applied to the previous trial error to produce the updating step (see (5.14)). Usually, a rule is defined to describe how the learning gain depends on the data, e.g., $\varepsilon g'(u_j)^T$ in (5.18) for Gradient descent ILC and $g'(u_j)^{-1}$ in (5.25) for Newton method based ILC. However, the two algorithms in the last section are confined to so-called non-adaptive update laws, which means the rule defined for calculating the learning



(a) Medium trajectory



(b) Fast trajectory

FIGURE 5.3: Medium and fast trajectory error results using various α. [Davies et al., 2008]

gain is fixed, typically determined by the prior knowledge of the plant model. Due to the plant being potentially highly time-varying and the fact that the more precise the plant model is, the better performance the controller may be expected to obtain, adaptive ILC algorithms will be investigated here. These update the plant model, and, in so doing, also update the ILC algorithm. Firstly, the trial-dependent adaptive ILC is considered, where the update rule is trial dependent and hence not updated during the trial. Thus, online identification is not required and there is plenty of time between trials to re-identify the plant model. Moreover, it is observed from Fig. 4.13 that linear parameters are more time-varying than the nonlinear ones. Thus, for simplicity, not the whole Hammerstein plant model, but only the linear part, is updated and then fed into the ILC controller to revise the update rule used for the learning gain. The detailed implementation is as follows.

5.3.1 Linear Adaptor plus Trial-dependent ILC

The whole control scheme, plotted in Fig. 5.4, consists of a linear adaptor and a trial-dependent ILC, two types of which are considered: Gradient descent ILC and Newton method based ILC.

- Linear adaptor

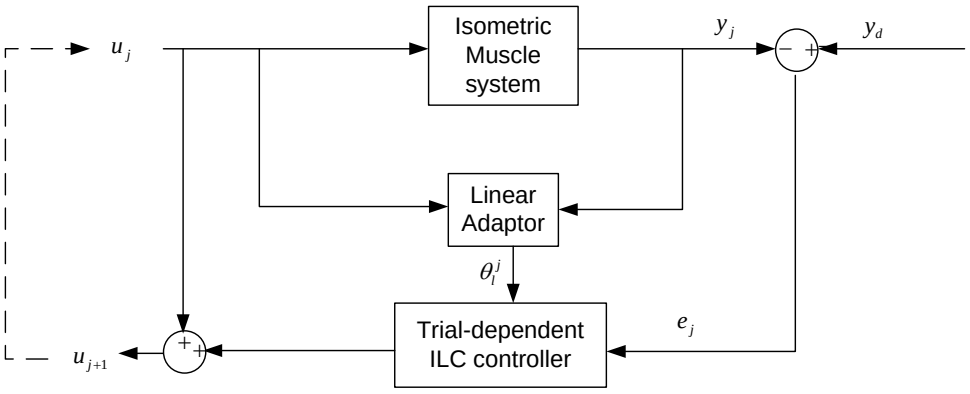


FIGURE 5.4: Block diagram of Linear Adaptor plus Trial-dependent ILC Controller

As its name implies, the linear adaptor only updates the linear block in the Hammerstein model and the nonlinearity identified beforehand is fixed in the plant model throughout the control test. The batch of data at the current trial and the memory of only the last trial's linear parameters are used in the updating algorithm, which is defined as

$$\hat{\theta}_l^j = \arg\min_{\theta_l} \|Y_j - \Phi_j \theta_l^j\|_2 + \Lambda \|\theta_l^j - \theta_l^{j-1}\|_2 \tag{5.30}$$

where Λ is a diagonal matrix with each diagonal element representing the variability of the corresponding linear parameter.

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_{l+n+1} \end{bmatrix}$$

where Y_j is the vector of the measured output at the jth trial

$$Y_j = \begin{bmatrix} y_j(l) \\ y_j(l+1) \\ \vdots \\ y_j(N) \end{bmatrix}$$

and Φ_j is the same as (4.26), while u_j, y_j is used instead

$$\Phi_j = \begin{bmatrix} -y_j(l) & \cdots & -y_j(1) & f(u_j(l+1-d),\hat{\theta}_n) & \cdots & f(u_j(l+1-d-n),\hat{\theta}_n) \\ -y_j(l+1) & \cdots & -y_j(2) & f(u_j(l+2-d),\hat{\theta}_n) & \cdots & f(u_j(l+2-d-n),\hat{\theta}_n) \\ \vdots & & \vdots & \vdots & & \vdots \\ -y_j(N-1) & \cdots & -y_j(N-l) & f(u_j(N-d),\hat{\theta}_n) & \cdots & f(u_j(N-d-n),\hat{\theta}_n) \end{bmatrix}$$

and

$$\theta_l^j = \begin{bmatrix} a_1^j \\ \vdots \\ a_l^j \\ b_0^j \\ \vdots \\ b_n^j \end{bmatrix}$$

The first part of the right hand side in (5.30) is to minimize the cost function and the second part is to minimize the updating step with a factor matrix Λ. The solution is

$$\hat{\theta}_l^j = (\Phi_j^T \Phi_j + \Lambda)^{-1}(\Phi_j^T Y_j + \Lambda \theta_l^{j-1}) \tag{5.31}$$

- Trial-dependent ILC

  For the class of model-based ILC algorithms, performance can be improved by revising the update rule to account for new plant data. By responding to change in the assumed plant model, this clearly provides a more precise updating step for the input in order to track the desired reference. In the sections which follow, the exact design of the update rule is considered for both Gradient descent ILC and Newton method based ILC will be explained.

  – Gradient descent ILC

    For the Gradient descent ILC algorithm, the trial-dependent update law becomes

$$u_{j+1} = u_j + \varepsilon_j g_j'(u_j)^T e_j \tag{5.32}$$

    where

$$g_j'(u_j) = \begin{bmatrix} 0 & 0 & 0 & \cdots 0 \\ C_j B_j \frac{\partial f}{\partial u_j(0)} & 0 & 0 & \cdots 0 \\ C_j A_j B_j \frac{\partial f}{\partial u_j(0)} & C_j B_j \frac{\partial f}{\partial u_j(1)} & 0 & \cdots 0 \\ \vdots & \vdots & \vdots & \ddots \vdots \\ C_j A_j^{N-2} B_j \frac{\partial f}{\partial u_j(0)} & C_j A_j^{N-3} B_j \frac{\partial f}{\partial u_j(1)} & C_j A_j^{N-4} B_j \frac{\partial f}{\partial u_j(2)} & \cdots 0 \end{bmatrix} \tag{5.33}$$

    The subscript j in A, B and C denotes the linear plant model for the jth trial, which can be recalculated by (5.4,5.5,5.6) according to the updated linear parameters θ_j from the Linear Adaptor.

    Moreover, ε_j is also adaptive and the optimal solution is

$$\varepsilon_j = \frac{\|g_j'(u_j)^T e_j\|^2}{w + \|g_j'(u_j) g_j'(u_j)^T e_j\|^2} \tag{5.34}$$

and

$$w = w_0 + w_1 \|e_j\|^2 \tag{5.35}$$

where w_0 and w_1 are the tuning factors necessary to keep the magnitude of ε_j small in order to enhance the resulting robustness properties.

– Newton method based ILC

Similarly, the trial-dependent update law for Newton method based ILC necessitates adding a subscript j to the adaptive term:

$$u_{j+1} = u_j + g'_j(u_j)^{-1} e_j \tag{5.36}$$

where $g'_j(u_j)^{-1} e_j = z_{j+1}$, which is the norm optimal ILC solution that drives the linearized system $g'_j(u_j)$ to track the desired output e_j. The only difference is that the linearized system is also trial-dependent. The $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ matrices in (5.29) are recalculated at each trial based on the updated linear parameters θ_j from the Linear Adaptor.

5.3.2 Online Identification plus Adaptive ILC

By extension of the methodology of the previous section, if online identification has been implemented, the plant model could be updated at each sampling time instant. In this case the update rule used for the learning gain is updated more often, not simply trial-dependent but taking each sampling time instant as an updating unit. The control scheme is shown schematically in Fig. 5.5, and consists of an online identification block and an adaptive ILC.

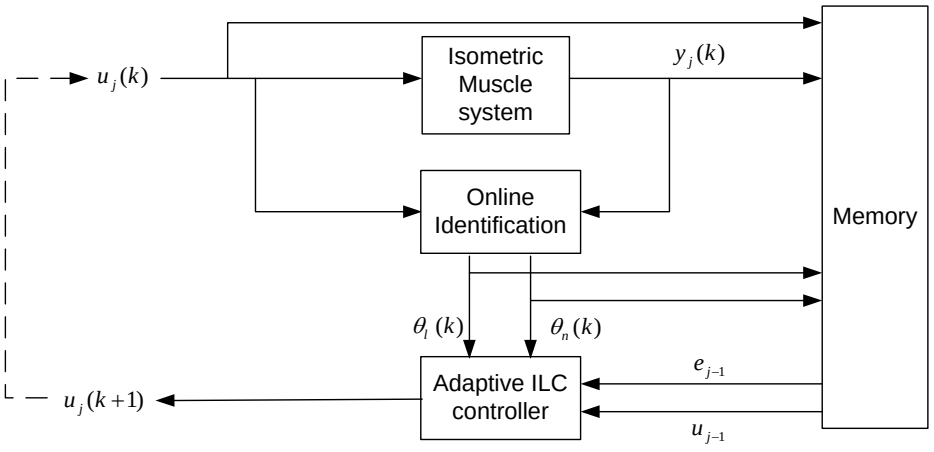


FIGURE 5.5: Block diagram of Online Identification plus Adaptive ILC

- Online identification

This online Identification block operates in real time to update the linear and nonlinear parameter vectors for the Hammerstein plant model at each sampling time instant. The

ARLS algorithm is employed, which is fast enough for real-time implementation, see Table 4.7 and has satisfactory performance for the noise experimental data and slowly time-varying muscle system, illustrated by Table 4.6. For the ARLS algorithm, the linear and nonlinear parts are separated and updated individually. Firstly, the information state for the linear and nonlinear part $P_l(k)$ and $P_n(k)$ is updated by (4.28) and (4.34) and then the linear and nonlinear parameter vectors $\hat{\theta}_l(k)$ and $\hat{\theta}_n(k)$ are updated as well according to (4.29) and (4.35), respectively. For added clarity, these updates are restated below:

$$
\begin{aligned}
P_l(k) &= \frac{1}{\lambda_l}\left(P_l(k-1) - \frac{P_l(k-1)\phi_l(k,\hat{\theta}_n(k-1))\phi_l^T(k,\hat{\theta}_n(k-1))P_l(k-1)}{\lambda_l I + \phi_l^T(k,\hat{\theta}_n(k-1))P_l(k-1)\phi_l(k,\hat{\theta}_n(k-1))}\right)\\
P_n(k) &= \frac{1}{\lambda_n}\left(P_n(k-1) - \frac{P_n(k-1)\phi_n(k,\hat{\theta}_l(k-1))\phi_n^T(k,\hat{\theta}_l(k-1))P_n(k-1)}{\lambda_n I + \phi_n^T(k,\hat{\theta}_l(k-1))P_n(k-1)\phi_n(k,\hat{\theta}_l(k-1))}\right)\\
\hat{\theta}_l(k) &= \hat{\theta}_l(k-1) + P_l(k)\phi_l^T(k,\hat{\theta}_n(k-1))\left(y(k) - \phi_l^T(k,\hat{\theta}_n(k-1))\hat{\theta}_l(k-1)\right)\\
\hat{\theta}_n(k) &= \hat{\theta}_n(k-1) + P_n(k)\phi_n^T(k,\hat{\theta}_l(k-1))\left(A(q,\hat{\theta}_l(k-1))y(k) - \phi_n^T(k,\hat{\theta}_l(k-1))\hat{\theta}_n(k-1)\right)
\end{aligned}
$$

- Adaptive gradient ILC

Only the Gradient descent ILC algorithm has been implemented adaptively since every update of Newton method based ILC requires an operation consisting of several-trials of norm optimal ILC, which is time-consuming and difficult to accomplish before the next sampling time arrives. However, a method to speed up the Newton adaptive implementation is presented, which may facilitate future implementation. The adaptive Gradient descent ILC is intended to respond to the newly updated plant model and to generate the input signal for the next sampling time instant according to the updated update law, which is derived as follows:

First, the non-adaptive update law is rewritten as

$$u_j = u_{j-1} + \varepsilon g'(u_{j-1})^T e_{j-1} \tag{5.37}$$

and for the Hammerstein plant model, when substituting (5.21) in (5.37), the $k+1$th entry becomes

$$u_j(k+1) = u_{j-1}(k+1) + \varepsilon \sum_{i=1}^{N-k-1} CA^{i-1}B\frac{\partial f}{\partial u_{j-1}(k+1)}e_{j-1}(k+i+1) \tag{5.38}$$

Similarly, for the online version, the plant model is updated at each sampling time instant, and the matrix $g'(\cdot)$ is updated using the most recent plant model. From (5.18) we can write

$$u_j(k+1) = u_{j-1}(k+1) + \varepsilon\left[g'(u_{j-1})\big|_{k+1}^T e_{j-1}\right](k+1) \tag{5.39}$$

where $g'(u_{j-1})\big|_{k+1}$ is the linearised system matrix on sample $k+1$ which uses the current

plant information as the best estimate for future samples. It is defined by

$$g'(u_j)\big|_k = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ C(1_k)B(1_k)\frac{\partial f(1_k)}{\partial u_j(0)} & 0 & \cdots & 0 \\ C(2_k)A(2_k)B(2_k)\frac{\partial f(2_k)}{\partial u_j(0)} & C(2_k)B(2_k)\frac{\partial f(2_k)}{\partial u_j(1)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C((N-1)_k)A((N-1)_k)^{N-2}B((N-1)_k)\frac{\partial f((N-1)_k)}{\partial u_j(0)} & C((N-1)_k)A((N-1)_k)^{N-3}B((N-1)_k)\frac{\partial f((N-1)_k)}{\partial u_j(1)} & \cdots & 0 \end{bmatrix} \tag{5.40}$$

where $x_k = \min\{x,k\}$. In particular, the k^{th} column of (5.40) is

$$\left[\underbrace{0,\ldots,0}_{k}, C(k_k)B(k_k)\frac{\partial f(k_k)}{\partial u_j(k)}, \ldots, C((N-1)_k)A((N-1)_k)^{N-k-1}B((N-1)_k)\frac{\partial f((N-1)_k)}{\partial u_j(k)}\right]^T \tag{5.41}$$

$$= \left[\underbrace{0,\ldots,0}_{k}, C(k)B(k)\frac{\partial f(k)}{\partial u_j(k)}, C(k)A(k)B(k)\frac{\partial f(k)}{\partial u_j(k)}, \ldots, C(k)A(k)^{N-k-1}B(k)\frac{\partial f(k)}{\partial u_j(k)}\right]^T \tag{5.42}$$

Application in the gradient algorithm (5.18) gives

$$\begin{aligned} u_j(k+1) &= u_{j-1}(k+1) \\ &+\varepsilon\left[\underbrace{0,\ldots,0}_{k+1}, C(k+1)B(k+1)\frac{\partial f(k+1)}{\partial u_j(k+1)}, \ldots, C(k+1)A(k+1)^{N-k}B(k+1)\frac{\partial f(k+1)}{\partial u_j(k+1)}\right]^T e_{j-1} \end{aligned}$$

This leads to the adaptive update law:

$$\begin{aligned} u_j(k+1) &= u_{j-1}(k+1) \\ &+\varepsilon(k+1)\sum_{i=1}^{N-k-1} C(k+1)A(k+1)^{i-1}B(k+1)\frac{\partial f(k+1)}{\partial u_{j-1}(k+1)}e_{j-1}(k+1+i) \end{aligned} \tag{5.43}$$

- Adaptive Newton method based ILC

Consider the Newton update (5.28) which must be resolved on each update using the most recent plant model. We therefore desire a recursive relationship in order to reduce computation. For the k^{th} sample (5.28) is given by

$$g'(u_j)\big|_k z_{j+1,k} = e_j \tag{5.44}$$

where $g'(u_j)\big|_k$ is given by (5.40). For the $k+1^{th}$ sample we can write

$$g'(u_j)\big|_{k+1} z_{j+1,k+1} = g'(u_j)\big|_k \left(z_{j+1,k} + \Delta z_{j+1,k}\right) = e_j \tag{5.45}$$

where $\Delta z_{j+1,k}$ is the increment to $z_{j+1,k}$ due to the new model information. Now from

(5.28)

$$g'(u_j)\big|_{k+1} = g'(u_j)\big|_k + \begin{bmatrix} 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & \underline{g'(u_j)\big|_{k+1}^k} - \underline{g'(u_j)\big|_k^k} \end{bmatrix} \tag{5.46}$$

where $\underline{g'(u_j)\big|_k^m}$ is the bottom left $m \times m$ subarray of the lower-triangular matrix $\underline{g'(u_j)\big|_k}$. Application in (5.45) gives

$$g'(u_j)\big|_k z_{j+1,k} + g'(u_j)\big|_k \Delta z_{j+1,k} + \begin{bmatrix} 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & \underline{g'(u_j)\big|_{k+1}^k} - \underline{g'(u_j)\big|_k^k} \end{bmatrix} z_{j+1}^k$$
$$+ \begin{bmatrix} 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & \underline{g'(u_j)\big|_{k+1}^k} - \underline{g'(u_j)\big|_k^k} \end{bmatrix} \Delta z_{j+1}^k = e_j \tag{5.47}$$

Then application of (5.44) yields

$$\left(g'(u_j)\big|_k + \begin{bmatrix} 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & \underline{g'(u_j)\big|_{k+1}^k} - \underline{g'(u_j)\big|_k^k} \end{bmatrix} \right) \Delta z_{j+1}^k =$$
$$- \begin{bmatrix} 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & \underline{g'(u_j)\big|_{k+1}^k} - \underline{g'(u_j)\big|_k^k} \end{bmatrix} z_{j+1}^k \tag{5.48}$$

The lower triangular structure of $g'(u_j)\big|_k$ means that the first k elements of Δz_{j+1}^k are zero, and so the solution to (5.48) can be written as

$$\underline{g'(u_j)\big|_{k+1}^k}\,\underline{\Delta z_{j+1}^k\big|^k} = \left(\underline{g'(u_j)\big|_k^k} - \underline{g'(u_j)\big|_{k+1}^k}\right) \underline{z_{j+1}^k\big|^k} \tag{5.49}$$

where $\underline{z_{j+1}^k\Big|^k}$ denotes the lower $N-k$ elements of z_{j+1}^k. The relationship (5.49) is a recursive formula for the calculation of the Newton direction z_{j+1} on trial k, whose dimension reduces with every sample. This effectively allows the Newton method based ILC law to be implemented recursively since:

- The gradient ILC algorithm can be used in place of NOILC to solve (5.49) between each trial

– The update (5.49) does not have to be solved at every sample, but can be computed at every $M > 1$ samples. This provides additional time to perform the computation, and M can be chosen to ensure the first, largest computation is not excessive
– Updating every M samples also introduces time-scale separation between the identification and control algorithms which stabilizes the dual system since it can ensure convergence of the identification routine prior to use in the controller.

5.4 Simulation Results

In order to investigate the performance of the adaptive control schemes, a simulated muscle system is firstly used to demonstrate the feasibility and superiority over the non-adaptive counterparts.

5.4.1 Simulated Muscle System

The simulated muscle system is designed as follows:

1. The nonlinearity $f(\cdot)$, shown in Fig. 5.6, is a cubic splines function with one knot at 150:

$$f(u) = -0.0278 + 0.0019u - 7.83 \times 10^{-6}u^2 + 1.78 \times 10^{-8}u^3 + 2.36 \times 10^{-8}|u - 150|^3 \tag{5.50}$$

2. The linear dynamics $G(q)$ is a second-order linear system with one unit time delay, one zero at 0.2493 and two poles at 0.9830 and 0.1110.
3. $e(t)$ is normally distributed random noise with the noise power 0.001.

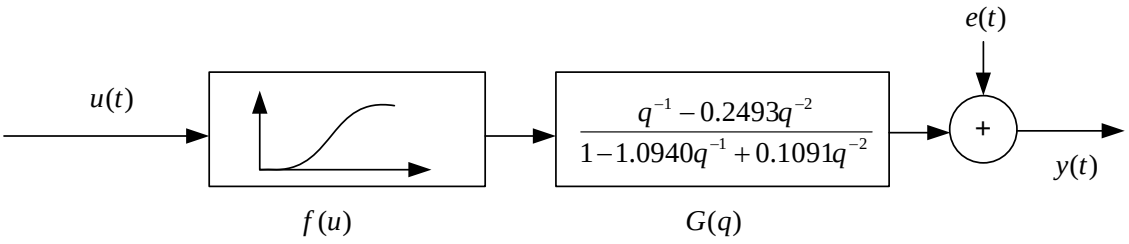


FIGURE 5.6: Simulated Muscle System

In real experimental situations, two kinds of problem are usually encountered: Firstly, due to inaccuracy of the identification procedure and the time span between the identification tests and the control tests, the identified plant model and the true plant are likely to be different. Moreover, even if the plant model is reasonably accurate, the true plant is actually time-varying and the fixed plant model must therefore contain inaccuracies. Thus, the adaptive control schemes and their non-adaptive counterparts will be evaluated using the following three situations:

- Case 1: exact plant model + time-invariant system

  Here it is assumed that the identified plant model is exactly the same as the true plant and the true plant is time-invariant.

- Case 2: wrong plant model + time-invariant system

  For this case, the linear dynamics of the identified plant model are different from those of the true plant:

$$\hat{G}(q) = \frac{q^{-1} - 0.2493q^{-2}}{1 - 1.1830q^{-1} + 0.1966q^{-2}} \tag{5.51}$$

  where the two poles are 0.9830 and 0.2000, only one of which is different. The true plant is kept time-invariant.

- Case 3: exact plant model + time-variant system

  In this situation, the true plant is assumed to be time-variant. Here, for simplicity, only a time-varying gain factor is applied to multiply the output. Two functions of time are applied to the gain factor:

  1. Trial-variant gain factor

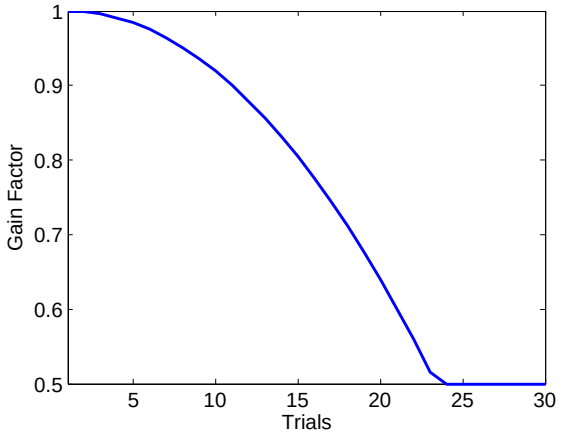


FIGURE 5.7: The trial-variant gain factor

     This is a reasonably simple case, where the gain factor in the same trial is kept constant, but decreases from 1 to 0.5 along each trial, see Fig. 5.7.

  2. Time-variant gain factor

     In this case, the gain factor is truly time-varying, like Fig. 5.8. With the assumption that there is no rest between the trials, this time-varying gain factor works as a look-up table, with the output being multiplied by the value at the corresponding time instant.

For each case, a 30-trial ILC implementation is applied to the stimulated muscle system and the reference trajectory is chosen as half of the cosine wave, shown in Fig. 5.9. The tracking errors

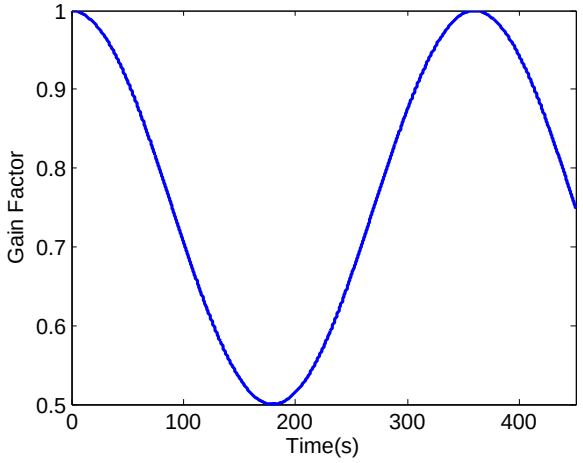


FIGURE 5.8: The time-varying gain factor

are normalized using

$$Normalized\ Error = \frac{\|e_j(k)\|_2}{\|r(k)\|_2}$$

and plotted against trial number.

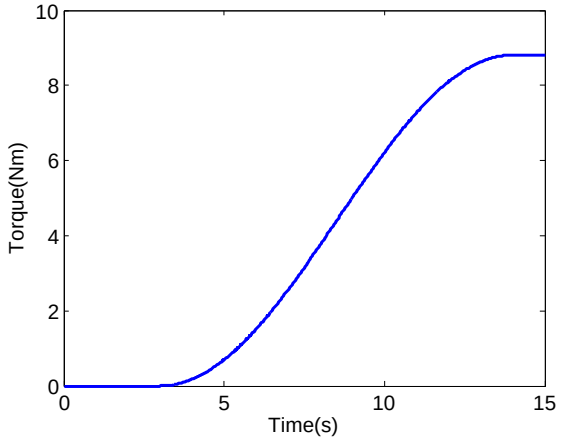
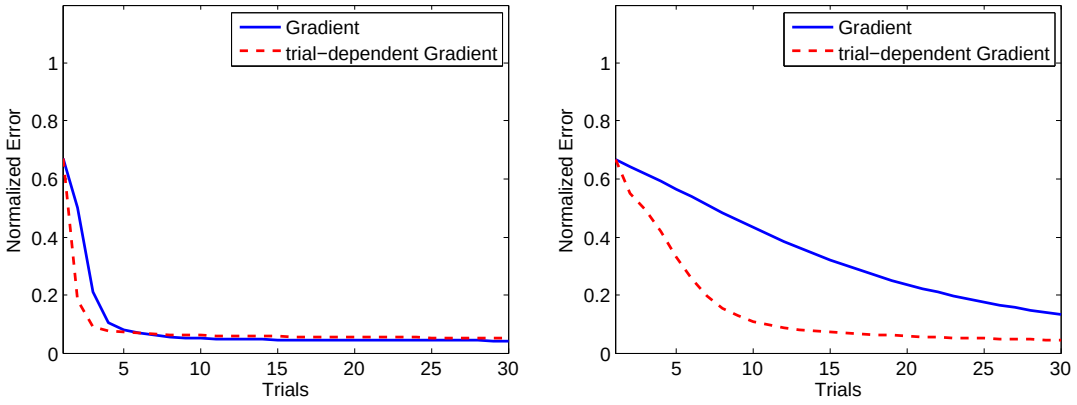


FIGURE 5.9: The reference trajectory

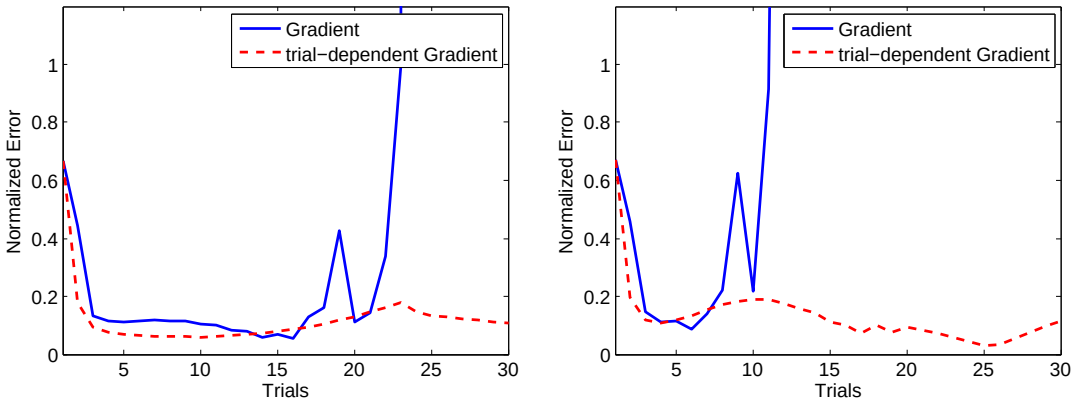
5.4.2 Gradient vs Trial-dependent Gradient ILC

For the Gradient ILC scheme, ε is only tuned for the first trial and fixed throughout the remainder of the test. However, the trial-dependent Gradient ILC will update the linear plant model, and ε and g' are also updated at each trial. Thus, the tracking error of the Gradient ILC converges particularly slowly when the plant model is not correct, compared with that of the trial-dependent one in Fig. 5.10(b). Moreover, when the simulated muscle system is time-varying, the tracking errors from the trial-dependent Gradient ILC still can remain stable and are under 0.2. However, for the Gradient ILC scheme, the tracking errors are not robust and instability results, see Fig. 5.11.



(a) exact plant model (b) wrong plant model

FIGURE 5.10: The normalized tracking errors of Gradient and trial-dependent Gradient ILC on Case 1: exact plant model + time-invariant system and Case 2: wrong plant model + time-invariant system



(a) trial-variant gain factor (b) time-varying gain factor

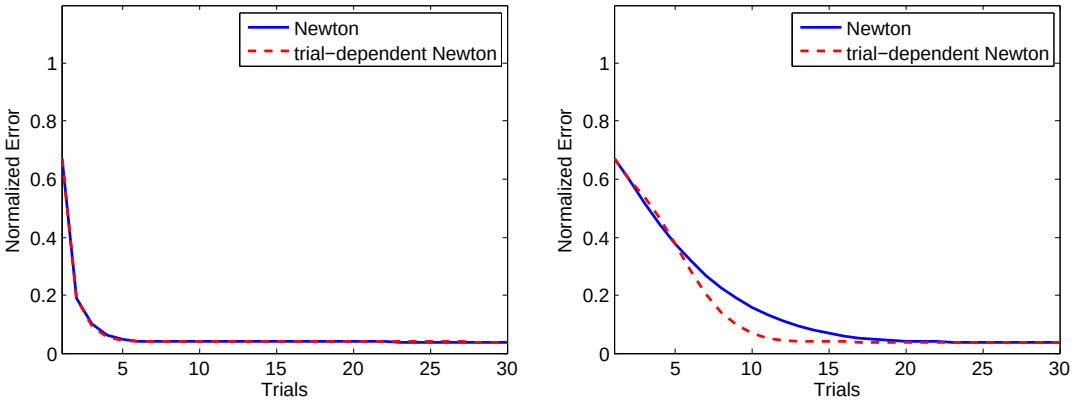
FIGURE 5.11: The normalized tracking errors of Gradient and trial-dependent Gradient ILC on Case 3: exact plant model + time-variant system

5.4.3 Newton vs Trial-dependent Newton Based ILC

For the Newton method based ILC and its trial-dependent version, the performances are both quite satisfactory and the tracking errors converge very fast as expected, except in Fig. 5.12(b) where the plant model is not quite accurate. Because the simulated muscle system does not vary very much with time and the wrong plant model does not vary significantly from the true plant, the trial-dependent Newton method based ILC only slightly improves the performance, illustrated by Fig. 5.12 and 5.13. Thus, in real experiments, only the Newton method based ILC has been implemented.

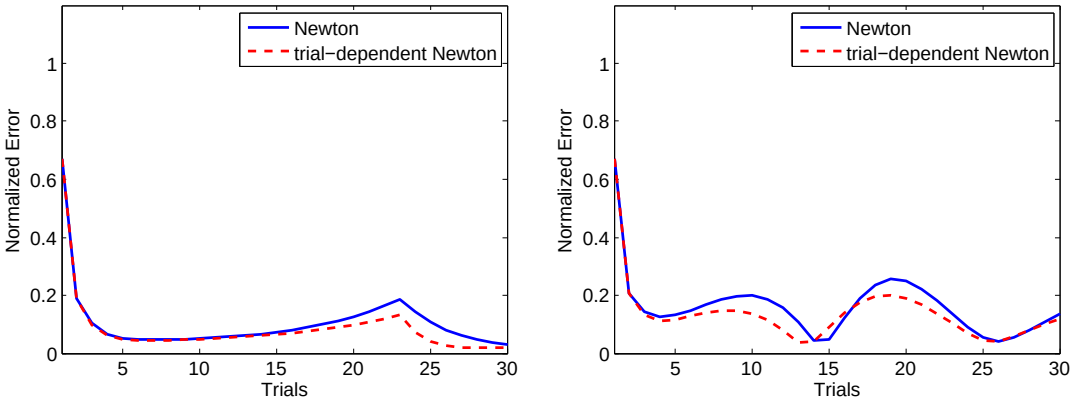
However, although the Newton method based ILC performs very well, it has not yet been used in combination with online identification plus adaptive ILC control scheme, because this necessitates a large amount of computation online whilst a separate ILC law is employed to solve the

intermediate problem (5.28). This is true even with the recursive formulation of the Newton update (5.49).



(a) exact plant model (b) wrong plant model

FIGURE 5.12: The normalized tracking errors of Newton and trial-dependent Newton based ILC on Case 1: exact plant model + time-invariant system and Case 2: wrong plant model + time-invariant system



(a) trial-variant gain factor (b) time-varying gain factor

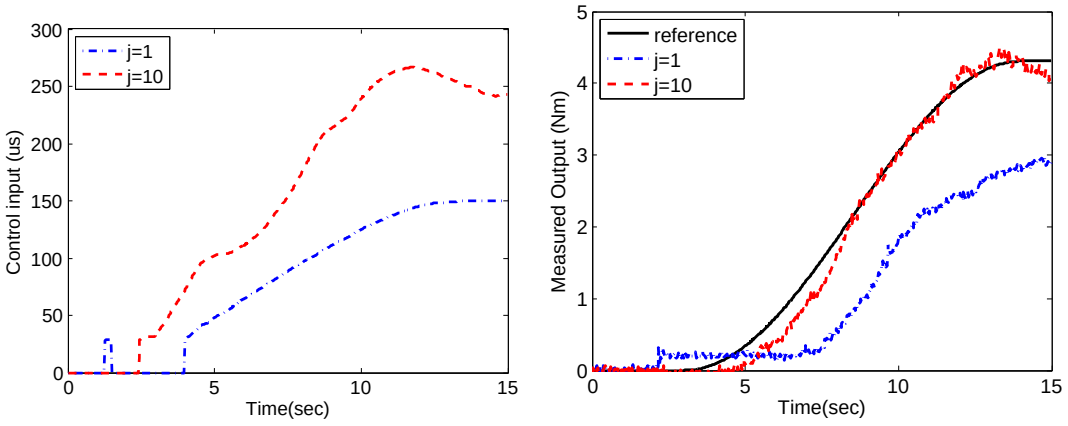
FIGURE 5.13: The normalized tracking errors of Newton and trial-dependent Newton based ILC on Case 3: exact plant model + time-variant system

5.5 Experimental Results

Experiments have so far been carried out using a single unimpaired subject. The reference signal y_d comprises half of a cosine wave whose amplitude is chosen as a percentage of the maximum force calculated during the identification test. The Gradient and Newton method based ILC algorithms are implemented for the non-adaptive control scheme and only Gradient ILC has been implemented for the trial-dependent adaptive control scheme so far. Ten individual tests are performed on the same subject over the course of different days for these three control schemes.

- Gradient ILC

  Fig. 5.14 shows an example of the results using Gradient ILC, where the control inputs and the measured outputs for the first and last trials are plotted and in the last trial, the measured output torque could track the reference quite well. However, it do not happen all the cases, see Fig. 5.15, where three typical examples for the normalized tracking errors against trial number are plotted. For this non-adaptive control scheme, the plant model and ε are not updated. Therefore, if the ε is small enough, the performance is more robust but the tracking error converges very slowly, as shown in Fig. 5.15(a). When increasing ε, the controller will suffer from instability, such as shown in Fig. 5.15(b) or even significant fluctuations as in Fig. 5.15(c).



(a) Control Input (b) Control Output

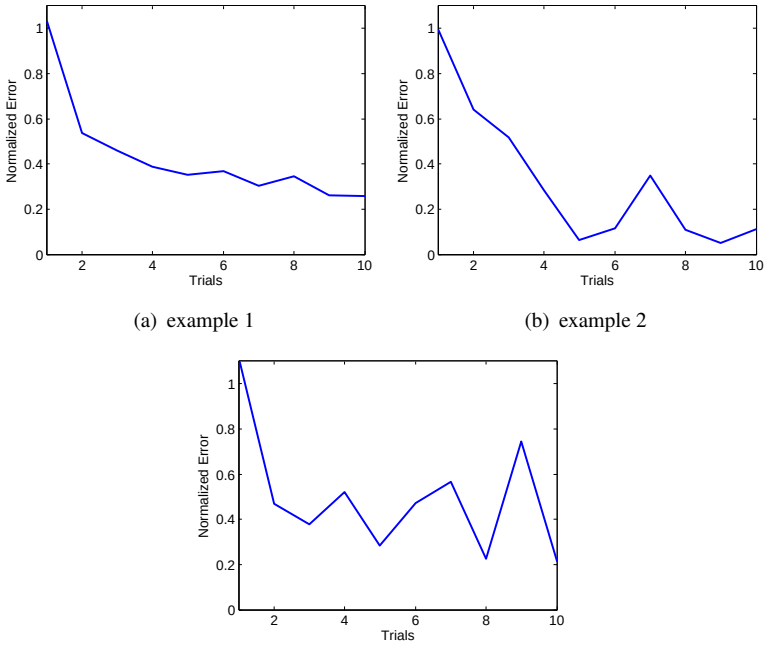
FIGURE 5.14: An example of isometric control experiments using Gradient ILC: (a) updated stimulation pulsewidth as control input for the first and last trials; (b) the measured output torques for the first and last trials, compared with the reference.

- Newton method based ILC

  Fig. 5.16 shows an example of the results using Newton method based ILC, where in the first trial, the measured output is not far away from the reference due to its fast convergence, also illustrated by Fig. 5.17. When compared with Gradient ILC in Fig. 5.18, Newton method based ILC is more robust and converges faster than Gradient ILC.

- Trial-dependent gradient ILC

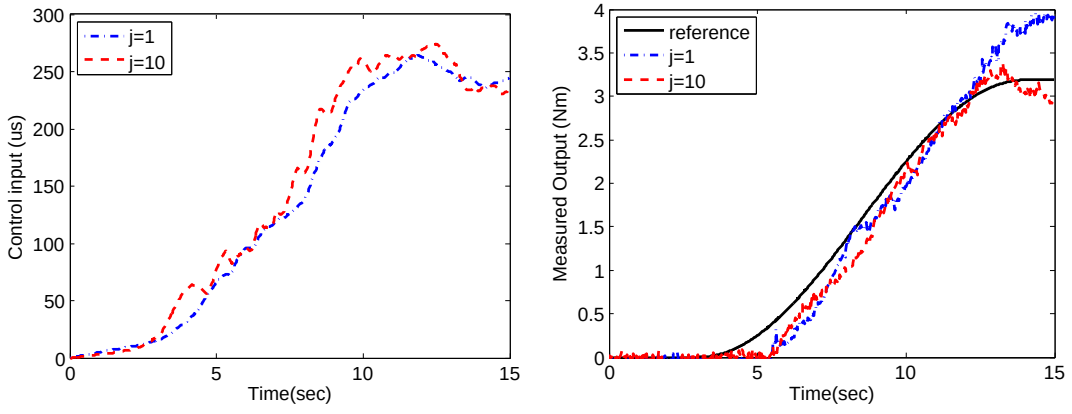
  Similar example can be found for the trial-dependent Gradient ILC as in Fig. 5.14, while the performance of trial-dependent Gradient ILC is more robust and the fluctuations are within a certain acceptable region. Moreover, the tracking errors for 10 tests are averaged and plotted against trial number in Fig. 5.20, for both Gradient and trial-dependent Gradient ILC in order to aid visual comparison. It is obvious that the the trial-dependent Gradient ILC is superior to the standard Gradient ILC law when applied to a real muscle system.



(a) example 1 (b) example 2

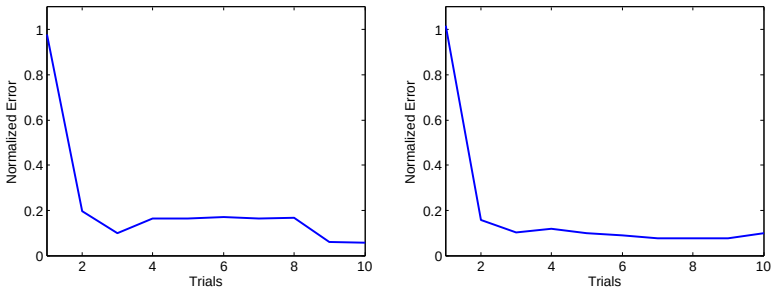
(c) example 3

FIGURE 5.15: Three examples of the normalized tracking errors of the Gradient ILC from real experimental data



(a) Control Input (b) Control Output

FIGURE 5.16: An example of isometric control experiments using Newton method based ILC: (a) updated stimulation pulsewidth as control input for the first and last trials; (b) the measured output torques for the first and last trials, compared with the reference.



(a) example 1 (b) example 2

FIGURE 5.17: Two examples of the normalized tracking errors of Newton method based ILC from real experimental data

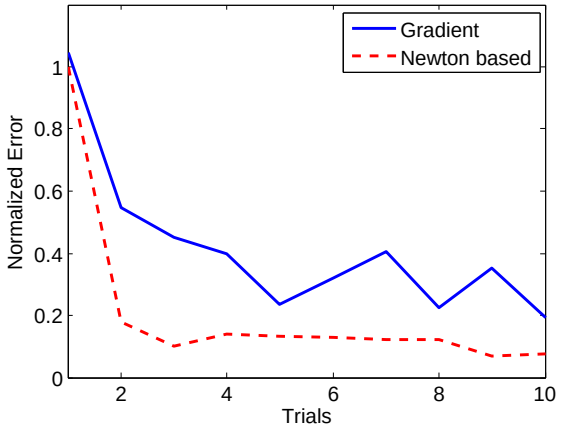
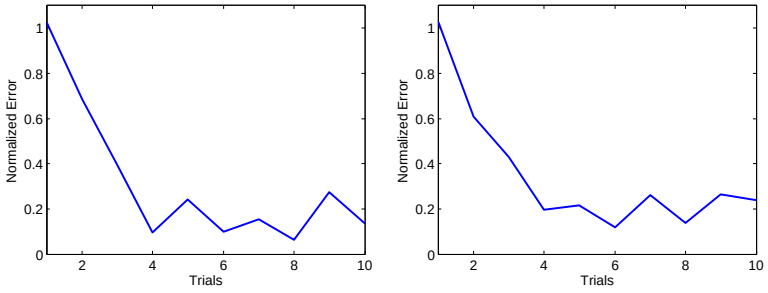
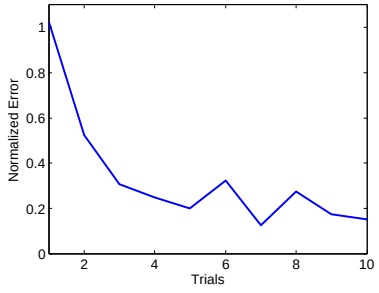


FIGURE 5.18: The averaged tracking errors for both Gradient and Newton method based ILC



(a) example 1 (b) example 2



(c) example 3

FIGURE 5.19: Three examples of the normalized tracking errors of the trial-dependent Gradient ILC from real experimental data

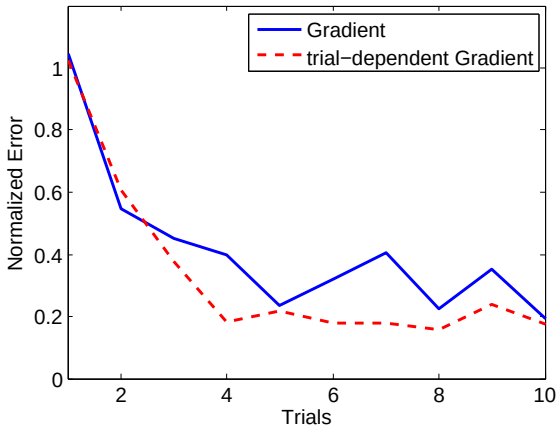


FIGURE 5.20: The averaged tracking errors for 10 individual tests of both Gradient and trial-dependent Gradient ILC

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis concentrates on the identification of electrically stimulated muscle after stroke and aims to provide model-based controllers with more accurate plant models in order to improve the effectiveness of FES treatments and hence speed the recovery of the impaired arms.

After reviewing muscle behavior and existing dynamic models, the nonlinear muscle dynamics under isometric conditions is adopted as the modelling subject in this thesis and a Hammerstein structure is chosen as the model. This consists of a static nonlinearity described by a cubic spline function and linear dynamics represented by a transfer function.

In Chapter 3, batch identification is first considered. By separating the parameters into linear and nonlinear parts, the Hammerstein model was expressed in terms of bilinear equations, which involves fixing one part, resulting in a linear system which is then solved for the other part using a least squares algorithm. The later researches all share this idea. At the initial phase, a two-stage algorithm is presented, which starts from an estimate of the linear parameters from the preliminary study, and then identifies the nonlinear and optimizes the linear parameters in two consecutive stages. When compared with the extended Ramp Deconvolution Method and Separable Least Squares (SLS) Optimization Algorithm on a simulated muscle system, this simple algorithm outperforms the others. In order to further improve the accuracy, a direct way is to repeatedly execute the two stages until the convergence is achieved, which leads to the iterative algorithm. Note that this iterative algorithm is not new in the literature, with several having been applied to the IIR linear part. Furthermore, the iterative algorithm has been extended to the OE linear model and formed the second iterative algorithm, where some extra schemes were applied to solve the difficulties encountered by using an OE linear model instead of an ARX linear model. These two iterative algorithms, the second of which is totally new, have been compared with another three identification methods on the experimental data from a single unimpaired subject. Four candidate tests are chosen through intensive study of related issues concerning

identification tests, and a comprehensive comparison is carried out on the identification, validation and cross validation results, showing: (1) the Hammerstein model is the best choice after comparison with a second-order linear model and the modified Wiener-Hammerstein model; (2) Two iterative algorithms both exhibit similar performance in prediction results, while due to the simpler implementation and faster computation, the first one is preferable; (3) The Staircase test, which is used for the first time in muscle tests, shows clear advantages over alternatives and is highly recommended.

Furthermore, with respect to the time-varying properties of the muscle models, recursive identification is considered in Chapter 4, where a novel recursive identification algorithm is investigated for the Hammerstein structure. This algorithm is derived from the iterative algorithm and the idea is to separate the linear and nonlinear parameters and construct their own information states and updating algorithms, based on Recursive Least Square (RLS) algorithms with exponential forgetting factors. Although the linear and nonlinear parts are recursively estimated in a parallel manner, the updating algorithms are related to each other's estimations from the previous time instant, so that it is named the Alternately Recursive Least Square (ARLS) Algorithm. This algorithm does not require a particular type of input signal and outperforms the leading RLS alternative in both numerical simulation, and when applied to the experimental identification of electrically stimulated muscle. RLS is the most promising technique in the literature and has been applied to the relevant work, which employs the RLS algorithm to the over-parameterization of the Hammerstein structure and applies SVD to recover the original parameters. However, due to the ignorance of the rank constraint in the over-parametric vector, the performance is not fully satisfactory, especially in noisy environments. The proposed ARLS can overcome this problem and become even faster in computational time for a single updating step and more reasonable compared to RLS due to two separate forgetting factors.

Eventually, the identified muscle model is used in FES control schemes for the electrically stimulated muscle under isometric conditions in Chapter 5. Firstly, two advanced model-based ILC schemes: Gradient descent ILC and Newton method based ILC have been used to control the Hammerstein plant. Then, two adaptive control schemes are proposed, that is, Linear Adaptor plus Trial-dependent ILC and Online Identification plus Adaptive ILC. The performance of these control schemes is investigated on a simulated muscle system and several are implemented on the robotic workstation. Future work will include experimental validation of all control schemes.

6.2 Future Work

The work reported in this thesis can be divided into two main topics: identification and control. The former is well-structured and heavily investigated and in the future, it could be expanded in terms of generality and individuality:

1. The identification algorithms developed in this thesis will be extended to more general

Hammerstein structures and even the other transformations in the Hammerstein Wiener model family.

So far, the iterative algorithms and their recursive counterpart, ARLS, both are built for the specified Hammerstein structure with a cubic spline nonlinearity followed by a transfer function linear dynamics. However, it is quite straightforward to extend them to more general Hammerstein representations. With respect to the models of the linear dynamics, generally, they are input-output polynomial models with the equations:

$$y(k) = \frac{B(q)}{A(q)} f(u(k)) + \frac{C(q)}{D(q)} v(k) \tag{6.1}$$

where $B(q)$ and $A(q)$ are the same as described in (3.1) and

$$C(q) = 1 + c_1 q^{-1} + \cdots + c_r q^{-r} \tag{6.2}$$

$$D(q) = 1 + d_1 q^{-1} + \cdots + d_s q^{-s} \tag{6.3}$$

When $A(q) = 1$, they become impulse-response models, in which case the problem will become much more easy and the solution can be found in [Bai and Li, 2004]. In this thesis, ARX and OE cases are considered, where $C(q) = 1$, $D(q) = A(q)$ and $C(q) = 1$, $D(q) = 1$, respectively. It will be very useful to expand the algorithms to this general linear model. For example, the over-parameterization identification algorithm is extended to MISO Hammerstein model with the same linear form as in (6.1) and some extra stages are required to identify the C and D in [Boutayeb et al., 1996]. Taking account of the nonlinearity, the algorithms in this thesis already can be widely used in the case that the nonlinearity can be expressed as

$$f(u) = \sum_{i=1}^{p} \beta_i f_i(u) \tag{6.4}$$

where f_i are known functions. However, when considering the real situation, some more complex nonlinearities are presented, such as hard nonlinearities with the dead-zone, saturation, preload, relay and hysteresis in [Bai, 2002b]. These particular cases are worth studying in the future. Moreover, there are quite a lot of related works in the literature widening their objective models to the other three in the Hammerstein Wiener model family, e.g. [Bai, 2002a] and [Gomez and Bayens, 2005]. The Wiener system, where the linear subsystem is in front of the static nonlinearity, has a similar forms to the Hammerstein structure. The other two, shown in Fig. 2.3, with another static nonlinearity at the beginning or linear dynamics in the end, are all very useful in modelling the nonlinear system, even for muscle systems. Thus, the algorithms will be developed for the other three model structures.

2. After identification of the nonlinear muscle dynamics under isometric conditions, the remaining components in the muscle model will be investigated. These comprise the passive force-length and force-velocity relationship $f_{passive}(x, \dot{x})$ and the active force-length and

force-velocity relationship $f_{active}(x,\dot{x})$, described in Section 2.1.2.

For the purpose of off-line identification, it is easy to identify these two by applying separate identification tests and fitting strategies, introduced by [Durfee and Palmer, 1994] and implemented in the project by [Freeman et al., 2009b]. After identifying the nonlinear muscle dynamics as a Hammerstein model, a passive test, where the force is generated only by passive manipulation of the muscle tendon, is firstly applied to estimate $f_{passive}(x,\dot{x})$ by piecewise linear fitting in a least squares sense. Then $f_{active}(x,\dot{x})$ is identified from the test subjecting the muscle to constant stimulation and random length inputs. By eliminating the effects from the other two, $f_{active}(x,\dot{x})$ is estimated. However, because of the individual identification procedures, the error caused by one process will propagate to the later, resulting in an inaccurate muscle model, especially when one or two of these procedures failed. Thus, further optimization strategies will be considered as well as the case in which separate processes are avoided and they are identified together.

More importantly, online identification of the whole muscle model will be investigated. This topic has been studied before: in [Schauer et al., 2005b], a Hammerstein-Wiener model was used to model the whole muscle system and extended Kalman filter is exploited to online identify the model. However, the experimental results between modeled and measured output are not in close agreement. In the literature, online identification of Hill-type muscle model has not been implemented yet and must be enforced in the future. As stated in the previous Chapter, muscle system is found to be time-varying due to fatigue, spasticity or changing physiological conditions so that the parameters of the most trustworthy Hill-type model should be estimated online.

The control component of this thesis is still at an early stage, with only simulation and a few experimental results presented so far. In order to provide the evidence that the model identified from the proposed algorithms indeed improves the performance of the controller, some experiments comparing the old identification scheme and the one proposed in this thesis will be carried out, including a number of subjects and repeated tests. These Two adaptive control schemes, Linear Adaptor plus Trial-dependent ILC and Online Identification plus Adaptive ILC, are suggested in order to respond to the time-variant muscle system. Unfortunately, only the first one has been implemented on the robotic workstation with just a few tests. Thus, in the future, the second one will be implemented and together with the first one, a large number of experiments will be carried out to evaluate the performance of adaptive control schemes and online identification.

Bibliography

H. S. Ahn, Y. Q. Chen, and K. L. Moore. Iterative learning control: brief survey and categorization. *IEEE Transactions on Systems, Man, and Cybernetics Part C - Applications and Reviews*, 37(6):1099–1121, November 2007.

S. Arimoto, S. Kawamura, and F. Miyazaki. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2):123–140, 1984.

E. W. Bai. An optimal two-stage identification algorithm for Hammerstein-wiener nonlinear systems. *Automatica*, 34(3):333–338, March 1998.

E. W. Bai. A blind approach to the Hammerstein-wiener model identification. *Automatica*, 38 (6):967–979, June 2002a.

E. W. Bai. Identification of linear systems with hard input nonlinearities of known structure. *Automatica*, 38(5):853–860, May 2002b.

E. W. Bai. Decoupling the linear and nonlinear parts in Hammerstein model identification. *Automatica*, 40(4):671–676, 2004.

E. W. Bai. Least squares solutions of bilinear equations. *System and Control Letters*, 55(6): 466–472, June 2006.

E. W. Bai, Z. Cai, S. Dudley-Javorosk, and R. K. Shields. Identification of a modified Wiener-Hammerstein system and its application in electrically stimulated paralyzed skeletal muscle modeling. *Automatica*, 45(3):736–743, March 2009.

E. W. Bai and M. Fu. A blind approach to Hammerstein model identification. *IEEE Transactions on Siganal Processing*, 50(7):1610–1619, July 2002.

E. W. Bai and D. Li. Convergence of the iterative hammerstein system identification algorithm. *IEEE Transactions on Automatic Control*, 49(11):1929–1940, November 2004.

L. L. Baker, D. R. McNeal, L. A. Benton, B. R. Bowman, and R. L. Waters. *NeuroMuscular Electrical Stimulation: A Practical Guide*. 3rd edition, 1993.

L. Bako, G. Mercere, S. Lecoeuche, and M. Lovera. Recursive subspace identification of Hammerstein models based on least squares support vector machines. *IET Control Theory and Applications*, 3(9):1209–1216, September 2009.

R. Baratta and M. Solomonow. The dynamic response model of nine different skeletal muscles. *IEEE Transactions on Biomedical Engineering*, 37(3):243–251, March 1990.

L. Bernotas, P. E. Crago, and H. J. Chizeck. A discrete-time model of electrically stimulated muscle. *IEEE Transactions on Biomedical Engineering*, 33(9):829–838, September 1986.

J. Bobet, E. R. Gossen, and R. B. Stein. A comparison of models of force production during stimulated isometric ankle dorsiflexion in humans. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(4):444–451, December 2005.

J. Bobet and R. B. Stein. A simple model of force generation by skeletal muscle during dynamic isometric contractions. *IEEE Transactions on Biomedical Engineering*, 45(8):1010–1016, August 1998.

J. Bobet, R. B. Stein, and M. N. Oguztoreli. A linear time-varying model of force generation in skeletal muscle. *IEEE Transactions on Biomedical Engineering*, 40(10):1000–1006, October 1993.

M. Boutayeb, D. Aubry, and M. Darouach. A robust and recursive identification method for MISO Hammerstein model. In *UKACC International Conference on Control*, volume 1, pages 234–239, 2-5 September 1996.

M. Boutayeb and M. Darouach. Recursive identification method for MISO Wiener-Hammerstein model. *IEEE Transactions on Automatic Control*, 40(2):287–291, February 1995.

D. A. Bristow, M. Tharayil, and A. G. Alleyne. A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(3):96–114, June 2006.

J. G. Broeks, G. J. Lankhorst, K. Rumping, and A. J. Prevo. The long-term outcome of arm function after stroke: Results of a follow-up study. *Disability and Rehabilitation*, 21:357–364, 1999.

J. H. Burridge and M. Ladouceur. Clinical and therapeutic applications of neuromuscular stimulation: A review of current use and speculation into future developments. *Neuromodulation*, 4(4):147–154, October 2001.

S. G. Carroll, R. J. Triolo, H. J. Chizeck, R. Kobetic, and E. B. Marsolais. Tetanic responses of electrically stimulated paralyzed muscle at varying interpulse intervals. *IEEE Transactions on Biomedical Engineering*, 36(7):644–653, July 1989.

M. Castro-Alamancos, L. Garcia-Segura, and J. Borrell. Transfer of function to a specific area of the cortex after induced recovery from brain damage. *European Journal of Neuroscience*, 4(9):853–863, 1992.

F. H. I. Chang and R. Luus. A non-iterative method for identification using Hammerstein model. *IEEE Transactions on Automatic Control*, 16(5):464–468, 1971.

H. F. Chen. Pathwise convergence of recursive identification algorithms for Hammerstein systems. *IEEE Transactions on Automatic Control*, 49(4):1641–1649, October 2004.

H. F. Chen and L. Guo. *Identification and Stochastic Adaptive Control*. Boston, MA: Birkhuser, 1991.

T. L. Chia, P. C. Chow, and H. J. Chizeck. Recursive parameter identification of constrained systems: An application to electrically stimulated muscle. *IEEE Transactions on Biomedical Engineering*, 38(5):429–442, May 1991.

H. J. Chizeck, S. Chang, and R. B. Stein. Identification of electrically stimulated quadriceps muscles in paraplegic subjects. *IEEE Transactions on Biomedical Engineering*, 46(1):51–61, January 1999.

H. J. Chizeck, P. E. Crago, and L. S. Kofman. Robust closed-loop control of isometric muscle force using pulsewidth modulation. *IEEE Transactions on Biomedical Engineering*, 35(7): 510–517, July 1988.

P. E. Crago, P. H. Peckham, and G. B. Thrope. Modulation of muscle force by recruitment during intramuscular stimulation. *IEEE Transactions on Biomedical Engineering*, 27(12): 679–684, 1980.

I. L. Davies, C. T. Freeman, P. L. Lewin, E. Rogers, and D. H. Owens. Newton method based iterative learning control of the upper limb. In *Proceedings of the 2008 American Control Conference*, pages 3887–3892, Seattle WA, United States, June 11-13 2008.

R. Davoodi and B. J. Andrews. Computer simulation of FES standing up in paraplegia: A self-adaptive fuzzy controller with reinforcement learning. *IEEE Transactions on Rehabilitation Engineering*, 6(2):151–161, 1998.

R. Davoodi, B. J. Andrews, and G. D. Wheeler. Manual and automatic control of fes-assisted indoor rowing exercise. In *Proceedings of the 2nd Joint EMBS/BMES Conference*, Houston, Texas, October 23-26 2002.

J. R. de Kroon, M. J. Ijzerman, J. Chae, G. J. Lankhorst, and G. Zilvold. Relation between stimulation characteristics and clinical outcome in studies using electrical stimulation to improve motor control of the upper extremity in stroke. *Journal of Rehabilitation Medicine*, 37:65–74, 2005.

J. R. de Kroon, J. H. van der Lee, M. J. Ijzerman, and G. J. Lankhorst. Therapeutic electrical stimulation to improve motor control and functional abilities of the upper extremity after stroke: a systematic review. *Clinical Rehabilitation*, 16:350–360, 2002.

E. J. Dempsey and D. T. Westwick. Identification of Hammerstein models with cubic spline nonlinearities. *IEEE Transactions on Biomedical Engineering*, 51(2):237–245, February 2004.

J. Ding, L. W. Chou, T. M. Kesar, S. C. K. Lee, T. E. Johnston, A. S. Wexler, and S. A. Binder-MacLeod. Mathematical model that predicts the force-intensity and force-frequency relationships after spinal cord injuries. *Muscle and Nerve*, 36(2):214–222, August 2007.

J. Ding, A. S. Wexler, and S. A. Binder-MacLeod. A mathematical model that predicts the force-frequency relationship of human skeletal muscle. *Muscle and Nerve*, 26(4):477–485, October 2002.

S. J. Dorgan and M. J. O'Malley. A nonlinear mathematical model of electrically stimulated skeletal muscle. *IEEE Transactions on Rehabilitation Engineering*, 5(2):179–194, June 1997.

H. Dou, K. K. Tan, T. H. Lee, and Z. Zhou. Iterative learning control of human limbs via functional electrical stimulation. *Control Engineering Practice*, 7:315–325, 1999.

W. K. Durfee and K. E. MacLean. Methods for estimating isometric recruitment curves of electrically stimulated muscle. *IEEE Transactions on Biomedical Engineering*, 36(7):654–667, July 1989.

W. K. Durfee and K. I. Palmer. Estimation of force-activation, force-length, and force-velocity properties in isolated, electrically stimulated muscle. *IEEE Transactions on Biomedical Engineering*, 41(3):205–216, March 1994.

W. Farahat and H. Herr. A method for identification of electrically stimulated muscle. In *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 1–4, Shanghai, China, September 2005.

M. Ferrarin, F. Palazzo, R. Riener, and J. Quintern. Model-based control of FES-induced single joint movements. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 9 (3):245–257, September 2001.

C. T. Freeman, A. M. Hughes, J. H. Burridge, P. H. Chappell, P. L. Lewin, and E. Rogers. An experimental facility using functional electrical stimulation for stroke rehabilitation of the upper limb. In *Proceedings of 10th IEEE International Conference on Rehabilitation Robotics*, pages 393–400, Noordwijk aan Zee, The Netherlands, June 2007a.

C. T. Freeman, A.-M. Hughes, J. H. Burridge, P. H. Chappell, P. L. Lewin, and E. Rogers. Iterative learning control of FES applied to the upper extremity for rehabilitation. *Control Engineering Practice*, 17(3):368–381, March 2009a.

C. T. Freeman, A. M. Hughes, J. H. Burridge, P. H. Chappell, P. L. Lewin, and E. Rogers. A model of the upper extremity using surface FES for stroke rehabilitation. *ASME Journal of Biomechanical Engineering*, 131(1):031011–1–031011–12, 2009b.

C. T. Freeman, P. L. Lewin, and E. Rogers. Further results on the experimental evaluation of iterative learning control algorithms for non-minimum phase plants. *International Journal of Control*, 80(4):569–582, April 2007b.

I. Goethals, K. Pelckmans, J. A. K. Suykens, and B. DeMoor. Subspace identification of hammerstein systems using least squares support vector machines. *IEEE Transactions on Automatic Control*, 50(10):1509–1519, October 2005.

H. Gollee, D. J. Murray-Smith, and J. C. Jarvis. A nonlinear approach to modeling of electrically stimulated skeletal muscle. *IEEE Transactions on Biomedical Engineering*, 48(4):406–415, April 2001.

J. C. Gomez and E. Bayens. Subspace-based identification algorithms for hammerstein and wiener models. *European Journal of Control*, 11(2):127–136, 2005.

C. Gowland, H. deBruin, J. V. Basmajian, N. Plews, and I. Burcea. Agonist and antagonist activity during voluntary upper-limb movement in patients with stroke. *Physical Therapy*, 72 (9):624–633, September 1992.

G. M. Graham, T. A. Thrasher, and M. R. Popovic. The effect of random modulation of functional electrical stimulation parameters on muscle fatigue. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(1):38–45, 2006.

W. Greblicki. Nonlinearitv estimation in hammerstein systems based on ordered observations. *IEEE Transactions on Signal Processing*, 44(5):1224–1233, May 1996.

W. Greblicki. Stochastic approximation in nonparametric identification of Hammerstein systems. *IEEE Transactions on Automatic Control*, 47(11):1800–1810, November 2002.

W. Greblicki and M. Pawlak. Identification of discrete hammerstein systems using kernel regression estimation. *IEEE Transactions on Automatic Control*, 31(1):74–77, January 1986.

E. Heinneman and C. Olson. Relations between structure and function in design of skeletal muscle. *Journal of Neurophysiology*, 28(3):581–598, 1965.

A. V. Hill. Then heat of shortening and the dynamic constants of a muscle. *Proceedings of the Royal Society*, 126:136–195, 1938.

T. C. Hsia. A multi-stage least squares method for identifying hammerstein model nonlinear systems. In *Proceedings of the 1976 IEEE Conference on Decision and Control including the 15th Symposium on Adaptive Processes*, pages 934–938, Clearwater, FL USA, 1976.

A. M. Hughes, C. T. Freeman, J. H. Burridge, P. H. Chappell, P. Lewin, and E. Rogers. Feasibility of iterative learning control mediated by functional electrical stimulation for reaching after stroke. *Journal of Neurorehabilitation and Neural Repair*, 23(6):559–568, 2009.

K. J. Hunt, R. P. Jaime, and H. Gollee. Robust control of electrically-stimulated muscle using polynomial H_∞ design. *Control Engineering Practice*, 9:313–328, 2001.

K. J. Hunt, M. Munih, and N. de N. Donaldson. Feedback control of unsupported standing in paraplegia - part I: Optimal control approach. *IEEE Transactions on Rehabilitation Engineering*, 5(4):331–340, 1997.

K. J. Hunt, M. Munih, N. N. Donaldson, and F. M. D. Barr. Investigation of the Hammerstein hypothesis in the modeling of electrically stimulated muscle. *IEEE Transactions on Biomedical Engineering*, 45(8):998–1009, August 1998.

I. W. Hunter. Frog muscle fiber dynamic stiffness determined using nonlinear system identification techniques. *Biophysical Journal*, 47(2):287a, February 1985.

I. W. Hunter. Experimental comparison of wiener and Hammerstein cascade models of frog muscle fiber mechanics. *Biophysical Journal*, 49(2):81a, February 1986.

I. W. Hunter and M. J. Korenberg. The identification of nonlinear biological systems: Wiener and Hammerstein cascade models. *Biological Cybernetics*, 55(2-3):135–144, November 1986.

A. F. Huxley. Muscle structure and theories of contraction. *Progress in Biophysics and Biophysical Chemistry*, 7:257–318, 1957.

S. Jezernik, R. G. V. Wassink, and T. Keller. Sliding mode closed-loop control of FES: Controlling the shank movement. *IEEE Transactions on Biomedical Engineering*, 51(2):263–272, 2004.

E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Principles of Neural Science*. McGraw-Hill, 4rd edition, 2000.

R. F. Kirsch, D. Boskov, and W. Z. Rymer. Muscle stiffness during transient and continuous movements of cat muscle: Perturbation characteristics and physiological relevance. *IEEE Transactions on Biomedical Engineering*, 41(8):758–770, 1994.

N. Lan, H. Q. Feng, and P. E. Crago. Neural network generation of muscle stimulation patterns for control of arm movements. *IEEE Transactions on Rehabilitation Engineering*, 2(4):213–224, December 1994.

L. A. F. Law and R. K Shields. Mathematical models use varying parameter strategies to represent paralyzed muscle force properties: a sensitivity analysis. *Journal of NeuroEngineering and Rehabilitation*, 2:12, May 2005.

L. A. F. Law and R. K. Shields. Mathematical models of human paralyzed muscle after long-term training. *Journal of Biomechanics*, 40:2587–2595, 2007.

F. Le, I. Markovsky, C. T. Freeman, and E. Rogers. Identification of electrically stimulated muscle after stroke. In *In European Control Conference, Budapest, Hungary, 23-26 August 2009 (Accepted)*, 2009.

Y. Liu and E. W. Bai. Iterative identification of hammerstein systems. *Automatica*, 43(2):346–354, February 2007.

L. Ljung. *System Identification: Theory for the User*. Upper Saddle River, NJ: Prentice Hall, 2nd edition, 1999.

P. H. Mccrea, J. J. Eng, and A. J. Hodgson. Time and magnitude of torque generation is impaired in both arms following stroke. *Muscle and Nerve*, 28:46–53, 2003.

M. Munih, K. J. Hunt, and N. N. Donaldson. Variation of recruitment nonlinearity and dynamic response of ankle plantarflexoes. *Medical Engineering and Physics*, 22(2):97–107, 2000.

K. S. Narendra and P. G. Gallman. An iterative method for the identification of nonlinear systems using a Hammerstein model. *IEEE Transactions on Automatic Control*, 11(3):546–550, July 1966.

D. H. Owens, J. J. Hatonen, and S. Daley. Robust monotone gradient-based discrete-time iterative learning control. *International Journal of Robust and Nonlinear Control*, 19(6):634–661, April 2009.

H. C. Park, D. G. Koo, J. H. Youn, , and J. Lee. Relay feedback approaches for the identification of Hammerstein-type nonlinear processes. *Industrial and Engineering Chemistry Research*, 43(4):735–740, February 2004.

V. M. Parker, D. T. Wade, and R. Langton-Hewer. Loss of arm function after stroke: Measurement, frequency and recovery. *International Rehabilitation Medicine*, 8(2):69–73, 1986.

M. Pawlak. On the series expansion approach to the identification of hammerstein systems. *IEEE Transactions on Automatic Control*, 36(6):763–767, June 1991.

M. Ponikvar and M. Munih. Setup and procedure for online identification of electrically stimulated muscle with matlab simulink. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 9(3):295–301, September 2001.

F. Previdi, M. Ferrarin, S. M. Savaresi, and S. Bittanti. Closed-loop control of FES supported standing up and sitting down using virtual reference feedback tuning. *Control Engineering Practice*, 13:1173–1182, 2005.

J. D. Ratcliffe, J. J. Haotaonen, P. L. Lewin, E. Rogers, and D. H. Owens. Robustness analysis of an adjoint optimal iterative learning controller with experimental verification. *International Journal of Robust and Nonlinear Control*, 18(10):91–115, July 2008.

D. N. Rushton. Functional electrical stimulation and rehabilitation - an hypothesis. *Medical Engineering and Physics*, 25(1):75–78, January 2003.

T. Schauer, N. O. Negard, F. Previdi, K. J. Hunt, M. H. Fraser, E. Ferchland, and J. Raisch. Online identification and nonlinear control of the electrically stimulated quadriceps muscle. *Control Engineering Practice*, 13(9):1207–1219, September 2005a.

T. Schauer, N. O. Negard, F. Previdi, K. J. Hunt, M. H. Fraser, E. Ferchland, and J. Raisch. Online identification and nonlinear control of the electrically stimulated quadriceps muscle. *Control Engineering Practice*, 13:1207–1219, 2005b.

R. A. Schmidt and T. D. Lee. *Motor Control and Learning: A Behavioural Emphasis*. Human Kinetics Europe Ltd, 3rd edition, 1998.

R. Shi and J. F. MacGregor. A framework for subspace identification methods. In *Proceedings of the 2001 American Control Conference*, pages 3678–3693, Arlington, VA, June 25-27 2001.

G. Shue, P. E. Crago, and H. J. Chizeck. Muscle-joint models incorporating activation dynamics, moment-angle, and moment-velocity properties. *IEEE Transactions on Biomedical Engineering*, 42(2):213–223, February 1995.

T. Soderstrom and P. Stoica. *System Identification*. Upper Saddle River, NJ: Prentice Hall, 1989.

P. Stoica. On the convergence of an iterative algorithm used for Hammerstein system identification. *IEEE Transactions on Automatic Control*, 26(4):967–969, August 1981.

L. Sun, W. Liu, and A. Sano. Identification of a dynamical system with input nonlinearity. *IEE Proceedings-Control Theory Applications*, 146(1):41–51, January 1999.

S. W. Sung. System identification method for Hammerstein processes. *Industial and Engineering Chemistry Research*, 41(17):4295–4302, August 2002.

R. Thorsen, R. Spadone, and M. Ferrarin. A pilot study of myoelectrically controlled FES of upper extremity. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 9(2): 161–168, June 2001.

P. Tresadern, S. Thies, L. Kenney, D. Howard, and J. Y. Goulermas. Artificial neural network prediction using accelerometers to control upper limb fes during reaching and grasping following stroke. In *Proceedings of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2916 – 2919, New York, USA, 2006.

M. Verhaegen and D. Westwick. Identifying mimo Hammerstein systems in the context of subspace model idenfication methods. *International Journal of Control*, 63(2):235–258, January 1996.

L. Vodovnik, W. J. Crochetiere, and J. B. Reswick. Control of a skeletal joint by electrical stimulation of antagonists. *Medical and Biological Engineering and Computing*, 5(2):97–109, March 1967.

J. Wang, A. Sano, T. Chen, and B. Huang. Identification of Hammerstein systems without explicit parameterisation of non-linearity. *International Journal of Control*, 82(5):937–952, May 2009.

J. Wang, A. Sano, D. Shook, T. Chen, and B. Huang. A blind approach to closed-loop identification of hammerstein systems. *International Journal of Control*, 80(2):302–313, February 2007.

T. Watanabe, K. Iibuchi, K. Kurosawa, and N. Hoshimiya. A method of multichannel PID control of two-degree-of-freedom wrist joint movements by functional electrical stimulation. *Systems and Computers in Japan*, 34(5):319328, 2003.

D. T. Westwick and R. E. Kearney. Separable least squares identification of nonlinear Hammerstein models: Application to stretch reflex dynamics. *Annals of Biomedical Engineering*, 29 (8):707–718, August 2001.

A. S. Wexler, J. Ding, and S. A. Binder-Macleod. A mathematical model that predicts skeletal muscle force. *IEEE Transactions on Biomedical Engineering*, 44(5):337–348, May 1997.

H. Wu, Z. Zhou, S. Xiong, and W. Zhang. Adaptive iteration learning control and its applications for fns multi-joint motion. In *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference*, Baltimore, Maryland, USA, 2000.

G. I. Zahalak and S. Ma. Muscle activation and contraction: Constitutive relations based directly on cross-bridge kinetics. *ASME Journal of Biomechanical Engineering*, 112(1):52–62, February 1990.

W. X. Zhao and H. F. Chen. Adaptive tracking and recursive identification for Hammerstein systems. *Automatica*, 45(12):2773–2783, 2009.

Y. Zhu. Identification of Hammerstein models for control using ASYM. *International Journal of Control*, 73(18):1692–1702, July 2000.