Trajectory Tracking Control of Robotic Jaw Actuators via Galil Motion System

A thesis presented in partial fulfilment of the requirements for the degree of

Master of Engineering

in

Mechatronics

At

Massey University, Auckland

New Zealand

Biqing Chen

2008

Abstract

A mechatronic chewing robot of 6-DOF mechanism which consists mainly of the skull, six crank actuators, end effector and motion control system has been designed and is required to simulate human chewing behaviours while the chewed food properties are evaluated. The robotic mechanism is proposed and its kinematic parameters are defined according to the biomechanical findings and measurements of the human masticatory system.

This thesis is concerned with the design and implementation of trajectory tracking control for robotic jaw actuators via Galil motion controller. The aim of this project is to simulate the dynamics behaviour and force-motion control of the robot, and to quantitatively assess food texture changes during chewing. A control system based Galil motion control card has been formed to achieve the motion of simulated human mastication. Some real human mastication motion have been tracked and used as targeted trajectories for the robot to reproduce.

Several experiments have been executed to measure the jaw movements and chewing forces. To reduce the vibration of the actuators and protect sensitive linkage part of the robot, the traditional PID control and some advanced control theories were implemented to achieve most effective efforts. A mathematical model was also designed at the first stage when a test actuator powered by brushless motor was formed; however, it is finally proven not well controlled in either mechanical and control ways. Major features of the built robot including the motion control system are presented and tested. Experimental results including free chewing, soft-food and hard-food chewing are given where the foods are simulated by foam and hard objects. Also the joint actuations and driving torques required are compared for the chewing of different foods.

In conclusion, tracking motion control has been attempted on the physical robot and a solution to the trajectory control has been developed.

Acknowledgement

I would like to thank the following people for their contributions:

- Professor Peter Xu for his guidance and assistance during the project. The willingness and enthusiasm he showed have helped me develop in my understanding of the research program.
- Doctors Johan Potgieter and John Bronlund for their great academic support and look after.
- Mr. Raymond Hoffman, technician at Albany workshop.
- Fellow students for their support and friendship that made the research life enjoyable.
- My wife and parents for their love and support.

Table of Contents

List of Figures

List of Tables

Abbreviations and Acronyms

DOF	Degree of freedom
AI	Artificial Intelligence
SCSI	Small Computer System Interface
MI	Maximum Intercuspation
CC	Cascade control
RSS	Revolute joint, Spherical joint
WJ	Waseda Jaw
TMJ	Temporomandibular Joint

Chapter 1 Introduction

1.1 Background

Food industry is tending to offer more tasty and specified food for particular consumer group. It is consequently difficult to satisfy everyone with a single selection. For certain categories of foods, texture is predominantly important. Therefore, the development of texturally attractive food products increasingly interests food industries. To do this, the study on the relationships between food properties, mastication process and texture perception can be categorized into three main areas, sensory assessment such as time intensity or progressive profiling, physiological study such as electromyography recording or jaw movement recording, and mechanical measurement such as compression or shear test of food. However, the sensory way may only provide qualitative feedback while instrumental measurements are too abstract for characterizing the dynamic interactions between texture perceptions and chewing patterns that occur during the process of food structure breakdown. And the physiological method has not been used for food evaluation due to sophisticated mastication process and mechanism of texture perception. [20]

There have been a lot of researches regarding machine to simulate people's chewing sensation and measure food's texture. The measurements have to be continuously made over the masticating process and my include frequency, length of chewing, tracking of jaw movement, force distribution, application of compression and shear forces on food, particle size and structure of the bolus just prior to swallowing. These quantities vary between subjects (e.g. because of differences in jaw geometry, teeth shape, sensitivity to pain) and food texture (e.g. elasticity, hardness, adhesion particularly to dentures, etc.). Because of the complex nature of the chewing process, there is a real need for the development of quantitative methods for evaluating the capability of a person to effectively chew foods.

According to recent research, most machines and devices available for evaluating food texture properties only have one particular function (mostly food compression) and are not able to simulate the entire suite of complex functions and movements involved during mastication. Since early 1990s there have been a number of attempts in developing masticatory robots such as dental patient training, jaw simulation, food texture assessment and speech therapy. The heuristic value of the robotic approach lies in the fact that it is able to perform actual mastication which enables one to understand different scenarios, explore different ideas, develop novel hypotheses, and gain insight into the consequences of variations in masticatory function between and within individuals and species. [21]

At Massey University, New Zealand, We began our robotic jaw research program at Massey University in 2002 with the aim of analytically characterizing food texture. The design of a mechatronic machine that can reproduce the jaw movements with motion and force requirements specified according to human chewing behavior has been attempted. A physical model has been manufactured where each muscle group of mastication is modeled by a RSS spatial linkage. Their placement between the mandible (or the end-effector) and the skull (or the ground) resulted in a spatial mechanism of 14 links, six linear actuators and 12 spherical joints. The machine is intended for evaluation of new types of foods in terms of food texture perception. Some simulation results of the jaw motion under open and close-loop controls have been attempted by SimMechanics, one of the toolboxes for Matlab.[20] the result of open-loop simulations have shown that the jaw mechanism was modeled properly that enables the jaw movements to be reproduced. Closed-loop simulations have shown that the jaw movements are highly dynamically coupled and their tracking is hardly achievable using a set of simple independent controllers.

In order to get better trajectory tracking performance, a multivariable control should be pursued. To do this, a linear time-invariant model of the mechanism dynamics must be extracted. While this work presented in this paper is preliminary, it does provide a viable jaw mechanism model to be worked on. The issues that are being researched currently under the project include solid model and animation, physical mechanism building, motion control and design, dynamics and multivariable control of the jaw mechanism.

1.2 Aim and Objectives

The thesis research is part of the research program of a chewing robot with 6 RSS parallel mechanism by Massey University, Auckland, New Zealand.

The aim of this project is to investigate into various of control algorithms and implement a range of trajectories via Galil controller that considers the dynamics of the robotic jaw, such as nonlinearities in its actuator, friction, stiffness, inertia and gravity effects. This required the chewing device to approximate and track the human chewing cycles. The control system must meet desired specification in accuracy and range of motion. The novelty of this research is therefore the development of a control algorithm to simulate the dynamics behavior, force-motion control of the model parameters in the robotic jaw. The inputs for the control system will be a set of six trajectories of the six actuators, as well as position feedback from actuators for closed loop trajectory tracking. The outputs of the control system will be the motion control signals at the appropriately amplified level to drive the six actuators.

To achieve this goal, the following objectives were identified:

a. Extensive research and understanding of the mechanism used

There are several articles published by the Massey University robotic jaw research group regarding on the previous research and design of the mechanism, inverse kinematics and motion control, and various control algorithms have been currently published.

Research regarding on how humans chew and several experiments should be attempted to select the most fitted control methods.

Mathematical modeling for an actuator could be attempted to provide extra information including the maximum current, output force which is difficult to gather in laboratory.

b. Implement and control the robotic jaw by Galil devices through suitable setting
   The understanding of Galil control system such as amplifier board, controller, and software is necessary because there are many functions that the system can be set at various situations.

c. Implement some typical chewing trajectories based on previous researches by using several control algorithms.
   The trajectories produced by tracking human chewing behaviours must be run on the robot first in order to select a suitable control theory. The feasibility of the hardware design must be proven by implementing a given mandibular trajectory to be tracked. Before the robot is given samples of simulated food, a range of data including the maximum motor force, PID value in case in PID control, tracking time interval, and other motion control methods should be tested.

d. Testing simulated food for this model to see its ability to work
   After a sample food is given, the robot should be able to apply little extra force to achieve the biting depth while the extra force must be controlled and prevented to harm all the linkages. It is necessary to capture the re-shaped trajectories due to added food and analysis the value of the extra force to gather the property information of the sample food.

Chapter 2 Literature Review

2.1 Food Texture Analyser

Food texture analysis is measuring the properties related to how a food feels in people's mouth. This can be done using a sensory panel—a group of people selected to taste food and provide feedback— or by instrumental methods. Compared with sensory panels, which are costly and time consuming, instrumental methods can save time, reduce costs, and provide more consistent, objective results. However, since it is difficult for machines to imitate biting and chewing, the need for sensory panels as a correlative test method will continue for the foreseeable future. [2]

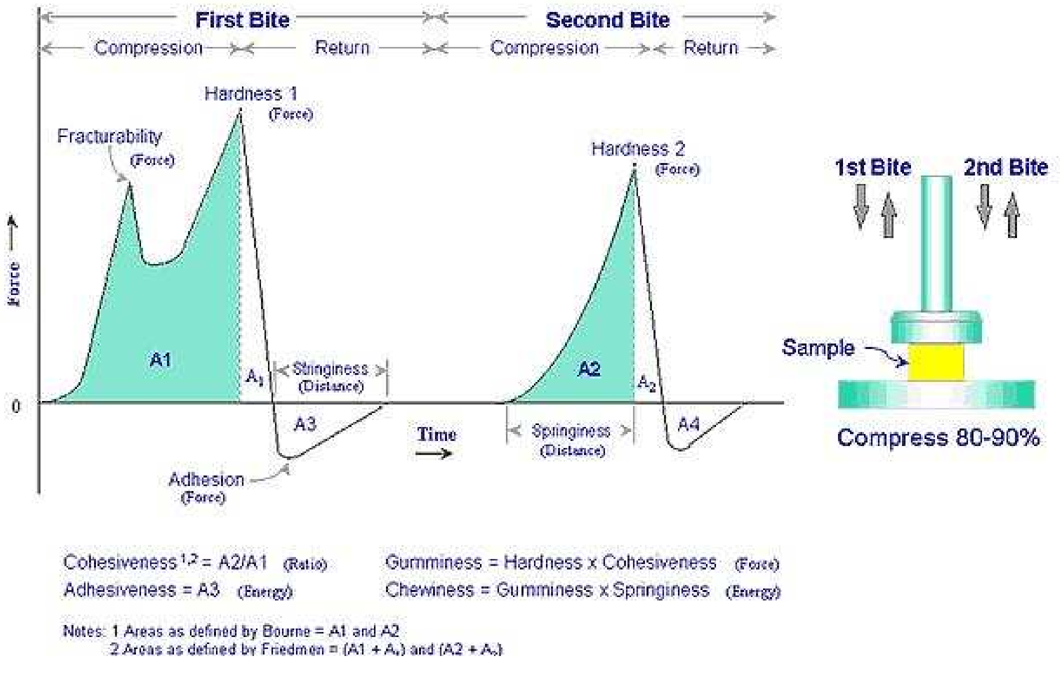


Figure 2.1 Instrumented Food Texture Analysis (*Adapted from INSTRON [2]*)

Today's improved instruments with digital control circuits provide the following benefits for food texture analysis:

a. Force is applied at a precise rate
b. Results are accurate and repeatable
c. Food texture analysis software performs complex calculations in seconds and produces detailed reports
d. Researchers can develop new test methods and automate the data reduction and reporting process
e. Cost-effective way to provide repeatable results quickly
f. Monitor product quality
g. Provide an objective measure of customer acceptance
h. Shorten new product development circles
i. Provide process control feedback

2.2 Foods Chewing Robots

The variety of mechanisms available for evaluating food texture properties in today's market is abundant; however, most machines only have one particular function (mostly food compression) and are not able to simulate the entire suite of complex functions and movements involved during mastication [3]. From early 1990s, there have been some attempts in developing robotic jaw for particular use, such as motion in speech and mastication [6], biting force information [7],and for dental patient training [3,4,5]. These ideas of the robotic approach are able to let machine perform actual mastication.

The dental robotics group in Takanishi Laboratory, Waseda University had developed mastication robot since 1986 to mimic human's mastication movement. They have developed series of robots primarily for patients' treatment [3, 4, and 5]. The treatment robots WY (Waseda Yamanashi) use master-slave system and capability for mouth opening and closing training that simulates doctor's hand motion during training. Six linear actuators manipulate the patient jaw like a parallel mechanism's end-effector. Patient bite the end-effector of the parallel mechanism and the doctor manipulates the master manipulator to train the patient's lower jaw. The upper mouth piece of the robot holds a patient's upper jaw and the patient's lower jaw are moved accordingly by the robot. Instead 3-DOF of WY-3 treatment robot, from 1998, the WY-5 series robots had been based on a parallel mechanism of 6-DOF and, the most advanced versions WY-5 and WY-6 were developed between 1998 and 2003, able to reproduce the same movable range and force as the human's jaw [1]. However, due to the main research area of the WY robots, their design is limited in the assistance for disabled people while cannot perform detailed human-like chewing behaviours. Although it can measure the state of lower and upper jaw, the information is only for monitoring mouth opening and closing status.

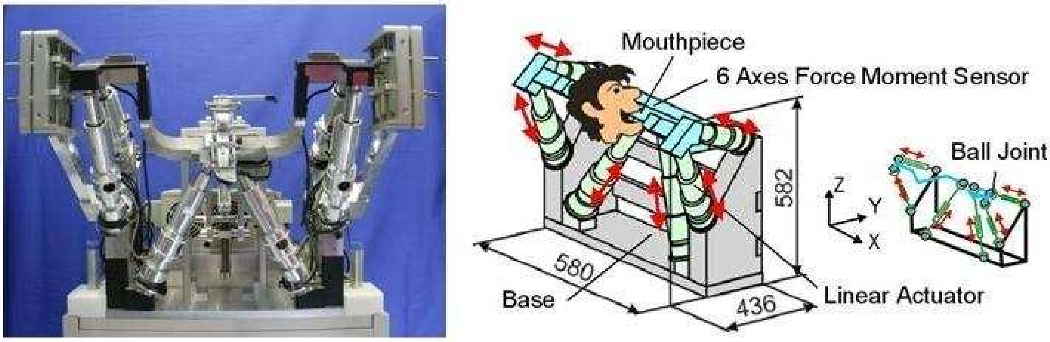


Figure 2.2 Patient Manipulator (*Adapted from Waseda-Yamanashi [1]*)

Alemzadeh, an engineer at Bristol University, UK is developing a robotic system called Dento-Munch that mimics the biting, chewing and munching of the human jaw. This robot is used to test materials life for use in dentistry. It can reproduce the full movement of the human jaw. Two platforms act as the upper and lower jaws, with the lower jaw capable of moving in 6 DOF. It also has closed loop control where can slow the chewing when resistance is high or speed up when is no resistance.

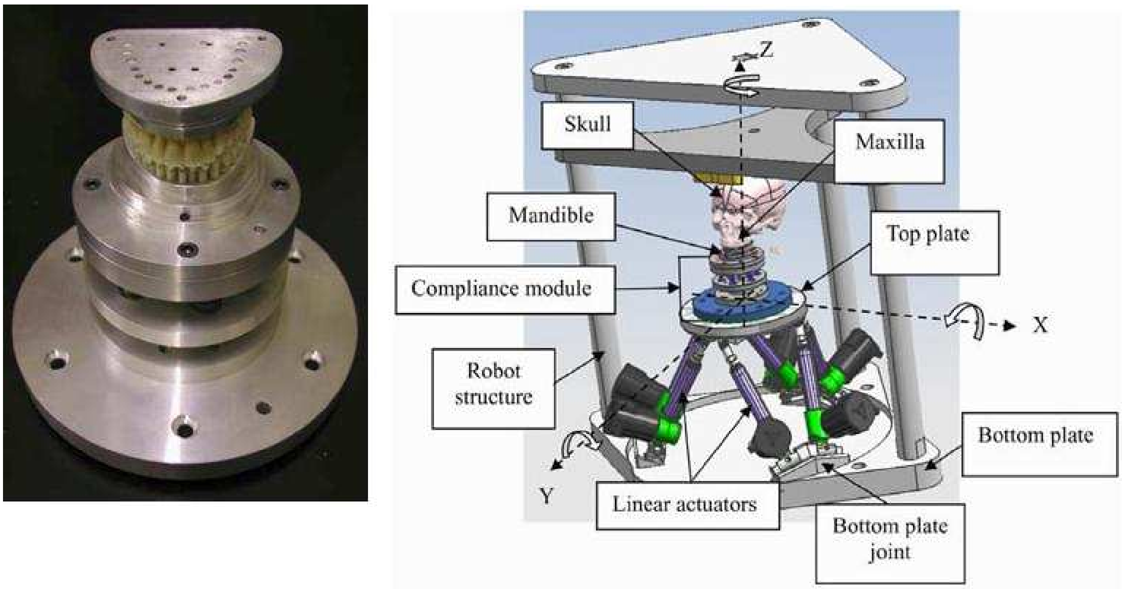


Figure 2.3 Dento-Munch Robotic Jaw(*Adapted from NewScientist.com [19]*)

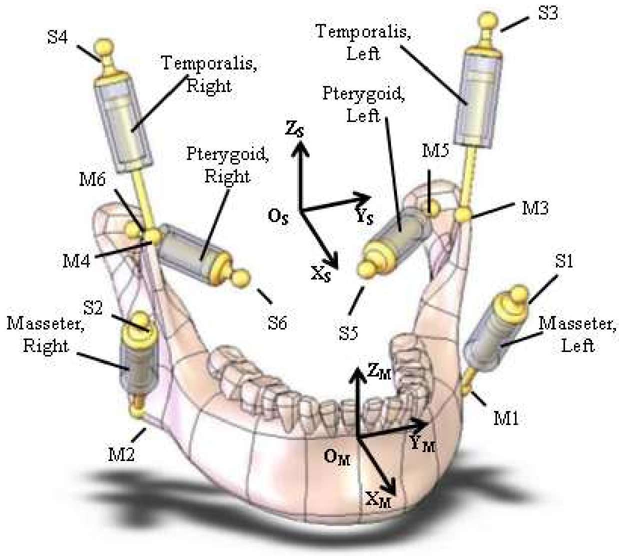
2.3 Foods Chewing Machines

The robotic jaw research and experiment program has been started since 2002 in Massey University to quantitatively evaluate the dynamic changes to the texture of foods during chewing [20]. The robot intends to chew foods in a human way (in terms of reproducing human masticatory movements and chewing forces) and, during the chewing the food particles are collected and analyzed for food property evaluation [21]. To address WJ robot’s problems mentioned above, our robot was modeled following a review of the biomechanical findings about the jaw structure and muscles of mastication [21]. Each of major mouth-closing muscles (temporalis, masseter and pterygoid muscle group) was represented by a linear actuator. Since the linear actuators are double-acting, the mouth-opening muscles are no longer required to be presented in the model. Muscle origin and insertions were modeled as spherical joints, by means of which the actuators are placed between the mandible and the skull so that each actuator acts always in the direction of the resultant muscular force. This resulted in a robotic platform mechanism with the mandible being a moving plate and the skull a ground plate. However, building such a robot based on the above model was beset by technical challenges. Among them was the fact that the size, motion and power specified for the linear actuator could not be satisfied all together using existing techniques [21]. Thanks to the investigation into the biological process of mastication and jaw structure, each of major mouth-closing muscles (temporalis, masseter and pterygoid muscle group) was represented by a linear actuator. Simulations for motion and control have been conducted using Matlab SimMechanics toolbox, and results have shown the jaw mechanism enables the jaw movements to be reproduced [20].

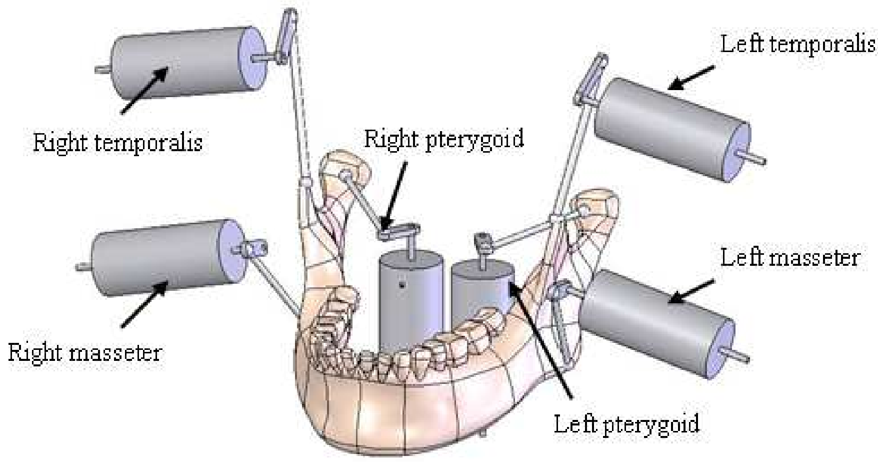
The robotic mechanism of a life-sized 6RSS parallel masticatory robot is proposed and its kinematic parameters are defined according to the biomechanical findings and measurements of the human masticatory system. For a given mandibular trajectory to be tracked, the closed form solution to inverse kinematics of the robot is found for joint actuations while differential kinematics is derived in Jacobian. [3]

A mechanism for life-sized masticatory robot was proposed early according to the biomechanical findings of the physiological jaw structure and muscles of mastication [42], as shown in Figure 2.4 (a). The mechanism is driven only by replicate mouth-closing muscles (temporalis, masseter and pterygoid muscle groups) in double-acting linear actuators placed between the mandible and maxilla via ball-socket joints. The dimension and specification for the robot were defined according to the life-size requirement and the masticatory patterns to be reproduced [21].

As existing technologies were challenged in building such small robot, a new mechanism, as shown in Figure 2.4(b), got into attention. This is a 6RSS parallel mechanism and the mandible is actuated by 6 RSS linkages with the couplers being in place of the linear actuators of the model in Figure 2.4(a). Each RSS linkage was designed such that the coupler aligns itself with the muscle's line of action. A special attention was put on the linkage's transmission angle for efficient delivery of the coupler's force onto the mandible [21]. The mandible was replicated from a human cadaver by computer tomography in the School of Dentistry at the University of Otago, New Zealand. The three-dimensional coordinates of the attachment/insertion positions of the major masticatory muscles were registered from eight cadavers [43]. However, it should be noted that the new model is intuitive and a rigorous proof of the equivalence of the two models in Figure 2.4 is to be studied.



(a) Robotic model of linear actuation



(b) Robotic model of crank actuation

Figure 2.4 Two mechanisms for life-sized masticatory robot

This new robotic model differs from the WJ's in its mechanism and biological foundation. Specifically, it has six DOFs sufficient for all possible masticatory trajectories in three-dimensional space; it employs a double-acting actuator for each group of muscles;

it attaches the actuators on the mandible and skull, via RSS linkage, according to the registered muscle position of cadavers and; it relaxes TMJ for free movement within its envelop.

2.4 Human Chewing Movements

The chewing circle can be divided into three phases (Figure 2.5) [8]: the preparatory phase, during which the mandible drops down and moves into position to crush the food bolus; the crushing phase, which lasts from contact with the bolus of food until contact with the tooth; and finally, the grinding phase. During the first two phases, the motion of the jaw is highly variable, and is influenced by the type of food being chewed. During the grinding phase, the motion of chewing is dictated by the anatomy of the teeth [9].

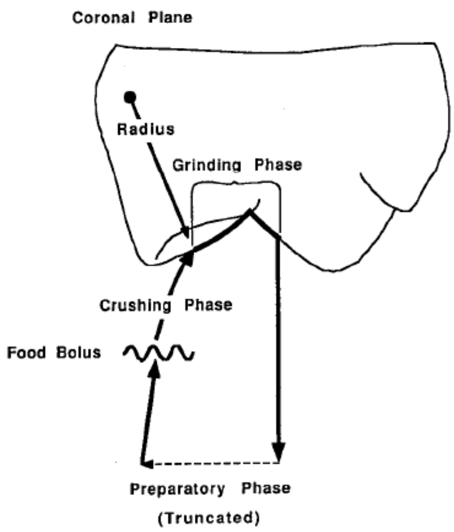


Figure 2.5 Cross section through a maxillary first molar showing the masticatory circle in the coronal plane

In the horizontal plane, motion can be described as an arc formed by rotation about the condyle on the same side as the teeth of interest. Because the amount of movement during the grinding phase is small (0.35 mm) [9] compared to the radius (approximately 10 cm), this motion can be approximated by straight-line motion. A similar approximation will not work in the coronal plane. Here the path of movement is highly dependent on the Occlusal anatomy of the teeth [13] (Figure 2.5).

Masticatory movements are described as closely repeated and the trajectory can be recorded by various approaches, such as: video camera, tracking sensors or even Face Recognition System. Figure 2.6 illustrates a representative cycle of an incisor path recorded from a healthy subject by using an Optotrak camera.

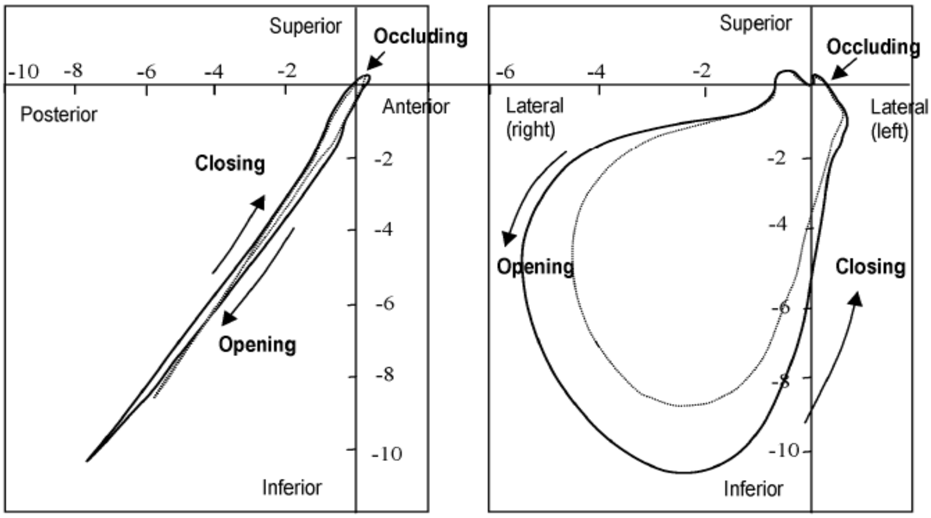


Figure 2.6 Trajectories of the incisor point in the sagittal plane and the frontal plane. The thin line represents soft food and the bold line for hard food. The source has been adapted from *Anderson et al. (2002)*.

Ogawa et al. (2001) defined that the occlusal area ends 0.5 mm from the maximum intercuspation (MI). The opening phase starts at the 0.5 mm point to the MI and ends at the most inferior point of the chewing circle. The closing phase starts from the most inferior point and ends at the 0.5 mm point to the MI [14].

On the other hand, in the Bioengineering Institute at the University of Auckland, Auckland, New Zealand a research was undertaken to use a custom made brace (Figure 2.7) to capture human chewing motion with all three-dimensional space through a motion capturing system VICON MX.

Figure 2.7 Brace for jaw tracking

The incisor trajectories recorded with this method are well suited for the chewing robot application as all the 3D movements were specified. Figure 2.8 shows a time-function of an OXYZ trajectory that was recorded using the method above.

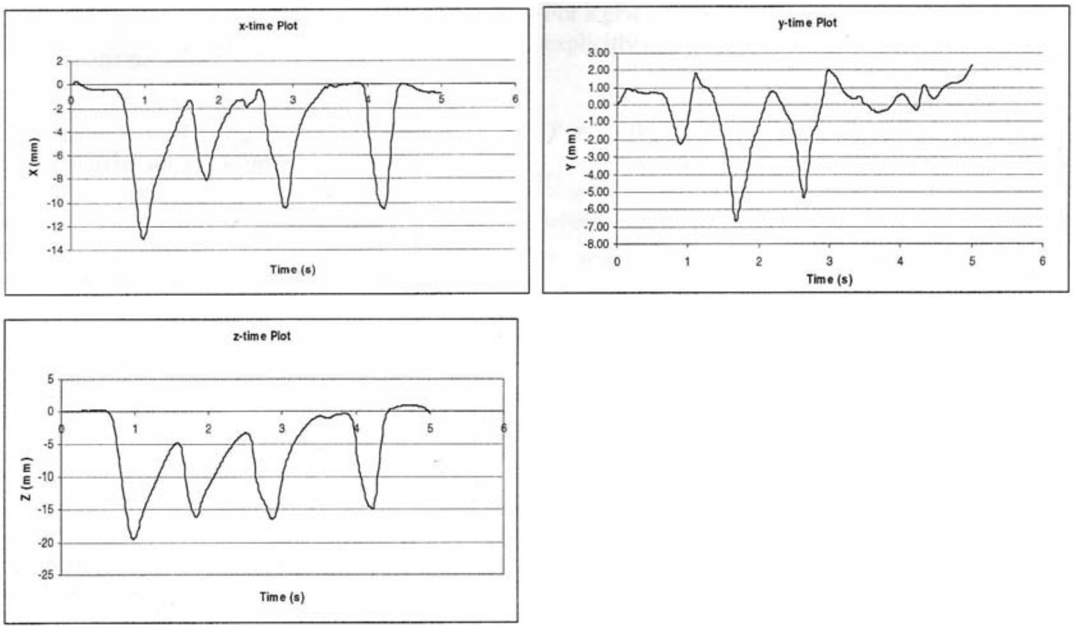


Figure 2.8 Time-Functions of displacement of incisor point

Jaw movements in mastication are also characterized by independent motion in four degrees of freedom. As shown in Figure 2.9, the figure shows bilateral chewing movements for six types of food with a normal bolus size (large diameter boluses give the same pattern). The slopes and intercepts relating pitch to horizontal position are more restricted in their range than that observed in speech but nevertheless differ statistically with factors such as bolus material and size. Jaw vertical and horizontal positions were observed to vary independently, particularly as a consequence of changes to the size of the food bolus (Figure 2.10). The timing of pitch and yaw movements was also independent with the phasing of movement involving motion in either one degree of freedom, the other, or the two combined[10].

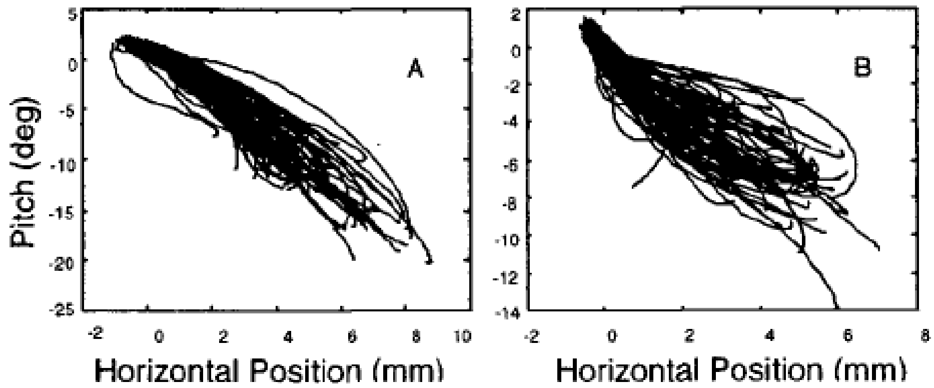


Figure 2.9 Jaw motion in mastication for two subjects (A. B) shown as a function of sagittal plane orientation and horizontal position.

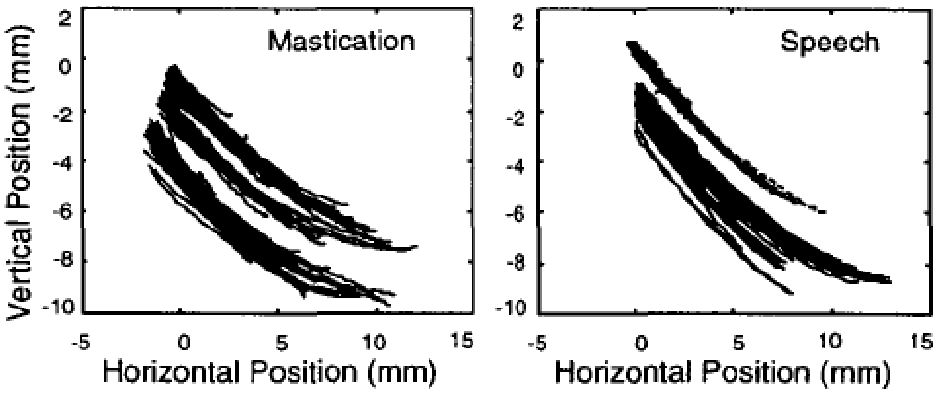


Figure 2.10 Sagittal plane vertical position varies independently of horizontal position, due to changes of bolus size and food material (left panel) and loudness of speaking voice (right panel).

2.5 Contributions of the Thesis

Instead of using costly sensory panel, the life-sized chewing robot of 6RSS parallel mechanism can save time, budget, and provide connivance in measuring the properties of food. The intelligent robotic jaw system would have characters such as learning, intuition, reasoning, understanding, autonomous behavior, and adaptive. It would include a system with such software that could perform quickly with a certain logical reasoning. At this stage, we might only know the jaw feedback through computer plotting or the data reading; however, this information would be either translated to human voice or user-friendly application interface.

Major features of the built robot including the motion control system are presented. The control system has been constructed and both hardware and software setup were done before the experiments started.

Experimental results for free chewing, soft-food chewing and hard-food chewing are given where the foods are simulated by foam and hard objects, and crank actuations and driving torques (an indication of muscular activities) required are compared for the chewing of different foods. A sophisticated control technology was then selected to control the Massey robotic jaw running with food samples. Thanks to the DMC1880 controller, all six actuators were tested in the same time and commands send to six channels were implemented in real time. The final results show the jaw can achieve most requirements including the required chewing force, speed and trajectory tracking under a stable and safe working condition. In the following chapters, trajectories are run on the robotic jaw and proved to be implementable. Problems, such as time interval difference, out of force limit, and linkage fault due to the excess motor power, are well solved.

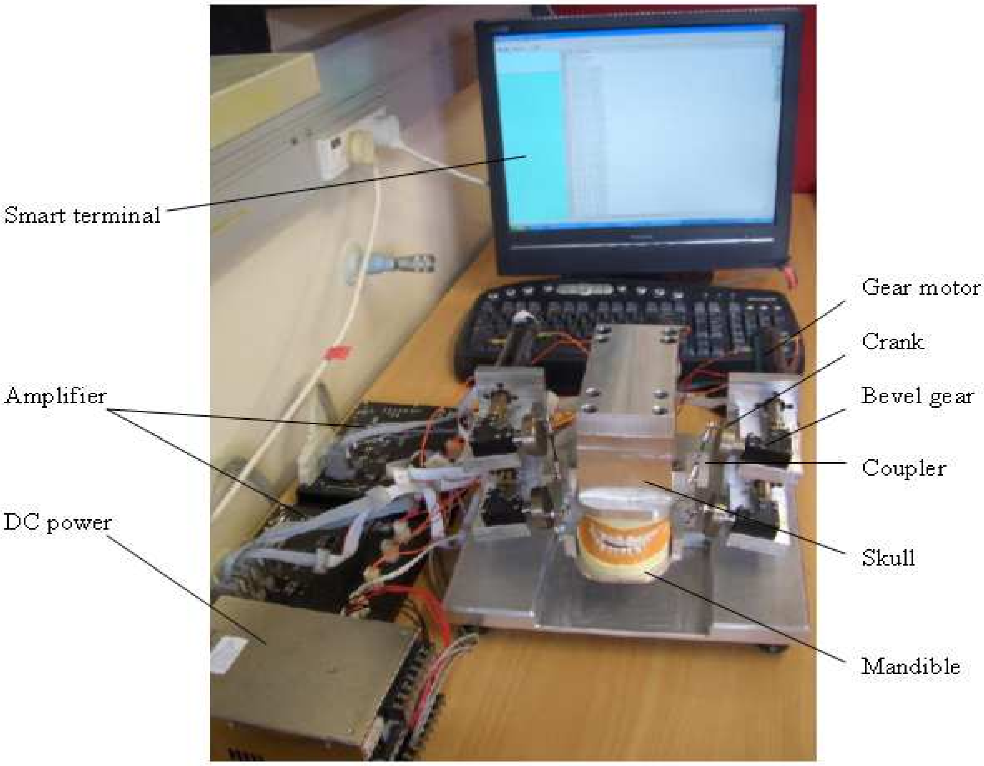
Chapter 3 Motion Control System

3.1 Introduction

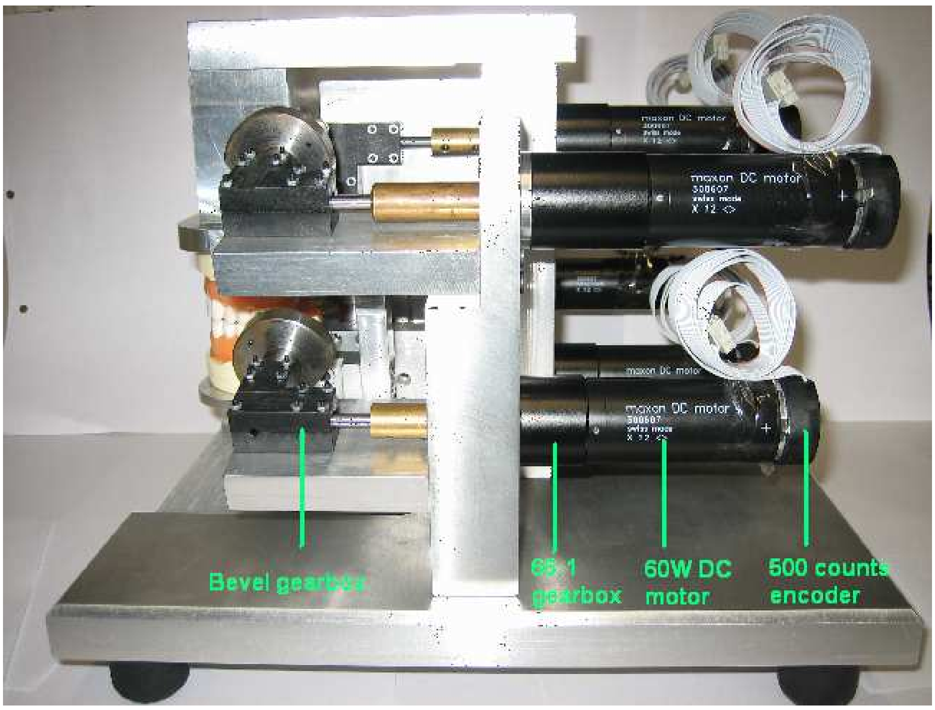
In a control system we typically have to deal with two signals only, the control signal and the measured variable. The control of jaw motion may be described for mastication. Modern control theory has proved to be very useful in solving problems when the system is well defined, but it fails to cope with some complex processes due to the non-linearly and time- varying behaviour of the system. Thus, the need of increasing speed, precision, and sophistication has resulted in a new generation of intelligent motion controllers.

3.2 The Massey Robotic System

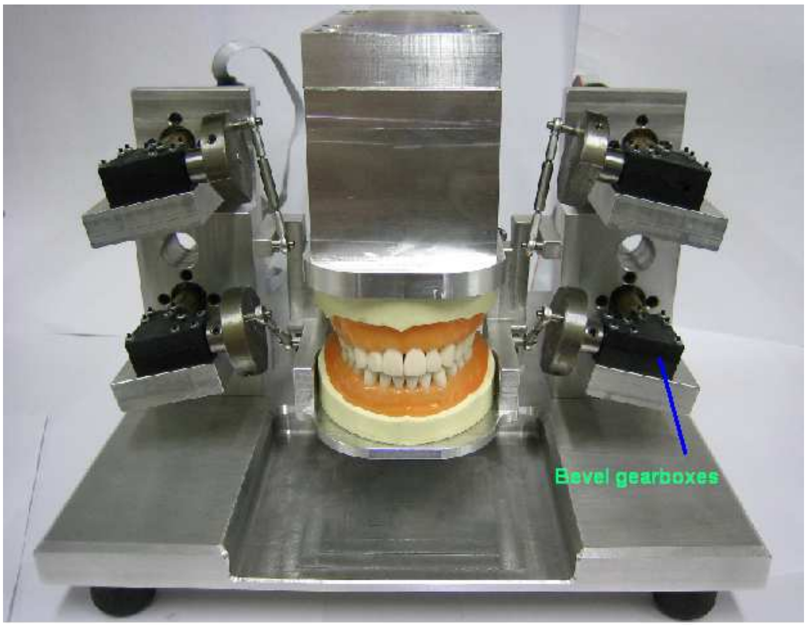
The built robot is shown in photographs (Figure 3.1). Each actuation unit consists of a MAXON 60W DC motor attached with a MR, Type L 500 counts encoder, and a 66:1 gearbox. This system gives rise to a constant speed torque of 86.2 mNm and maximum speed of 121.2 RPM at the crank end. By using bevel gearboxes with a gear ratio of 1:1 in front of each motor unit, all six actuation units were oriented in the same direction. The crank was manufactured out of mild steel while the coupler was assembled of two rod ends and a shaft. According to a test conducted, the coupler can withstand a pulling force of 988 N. The crank and the coupler were connected in a way that the coupler was able to rotate around its connection shaft and could move freely in the frontal angle direction. The teeth were made out of melamine, strong enough for chewing tests. During the construction, the life-sized upper and lower jaws were shortened in depths of about 10 mm in order to have a space to accommodate the bevel gearbox mounting units for the left/right pterygoid actuators. [3]



(a) major parts



(b) motors and attached encoders



(c)

Figure 3.1 Built masticatory robot

3.3 The Galil System

3.3.1 Programming and Executing

Galil terminal program, where the motion control commands were written that can be either online or offline downloaded to the card. Its WSDK package allows automatic tuning to provide software control. Galil DMC motion controller has the feature of instant control without host intervention plus multitasking allows up to eight user programs to be executed simultaneously. Then the commands transaction delay between host and controller can be ignored; in addition, six servo motors can run together permitting concurrent operation of independent tasks.

The motion control system consists of a six-axis motion control card (Galil DMC-1860), two amplifiers (each driving up to four motors), a power supplier, and a program (Galil DMC Smart Terminal), as shown in Figure 3.1. The motion control card outputs a control signal for each actuator that varies between -10V and 10V.

The low level motion control of the robot involves a trajectory-tracking PID control for each of the six actuators. In Galil PID gains are tuned for each actuator which acts on differences between commanded position and position fed back by the encoders. Independent PID control is expected to work well because of the gear reduction between motor and crank, which reduces greatly the effects of dynamic coupling between cranks.

The trajectory data for each actuator can be in the form of a series of points versus time. This type of motion suits a mode of motion control called contour mode in Galil and implemented via Galil Smart Terminal program. To program the motion, two arrays must be defined, one for the absolute position data and the other for the difference between consecutive points. The absolute position data can be imported explicitly from the output of the inverse kinematics program in Matlab. The difference data array is then calculated in the terminal program. A constant time-interval between consecutive points is also defined. Contour mode works by moving the specified difference (in encoder counts) over the fixed time interval. Using this method, the commanded motion of all 6 actuators can be synchronized, giving rise to a coordinated movement to the robotic jaw.

3.3.2 Amplifier Board

Galil AMP-19540 multi-axis amplifier has been designed to work with both brushless and brush-type servomotors and installation will vary depending upon which type of motor is used. With two AMP-19540, the amplifiers can control up to eight axes per Galil software.

The amplifier boards in computer receive a signal range from 10V to -10V and convert it to the required level of current. The amplifiers can be configured in the current or velocity mode. The velocity mode is preferred when velocity feedback is utilized; otherwise, a power amplifier typically is configured in the current mode.

In the current mode, the amplifier produces a current that is directly proportional to the input voltage. This is achieved by a current feedback loop which monitors the current and assures that it is proportional to the command signal. Current amplifiers are characterized by the current gain K_a which indicates the amplification per 1V of command signal. Amplifiers can also be configured in velocity mode. Such an amplifier can be seen as the accompanying figure

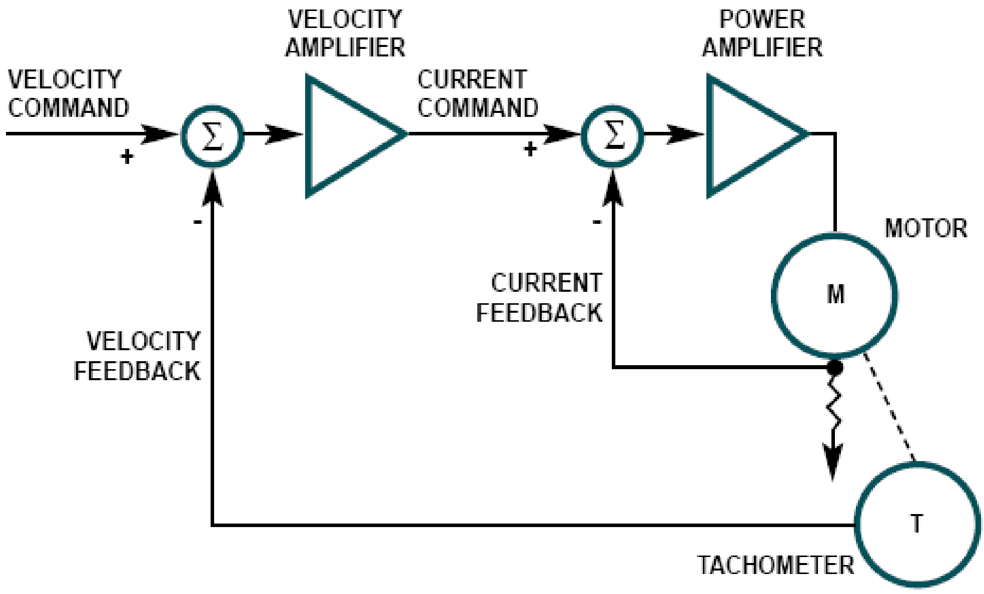


Figure 3.2 Amplifier control diagram

Table 3.1 Electrical Specifications

Input Voltage	18-80V DC
Continuous Current	7A
Peak Current	10A
Amplifier Gain	0.4,0.7 or 1.0A/V(jumper adjustable
Switching Frequency	60kHz
Continuous power per axis	500W
Require thermal cooling	All axes>3A

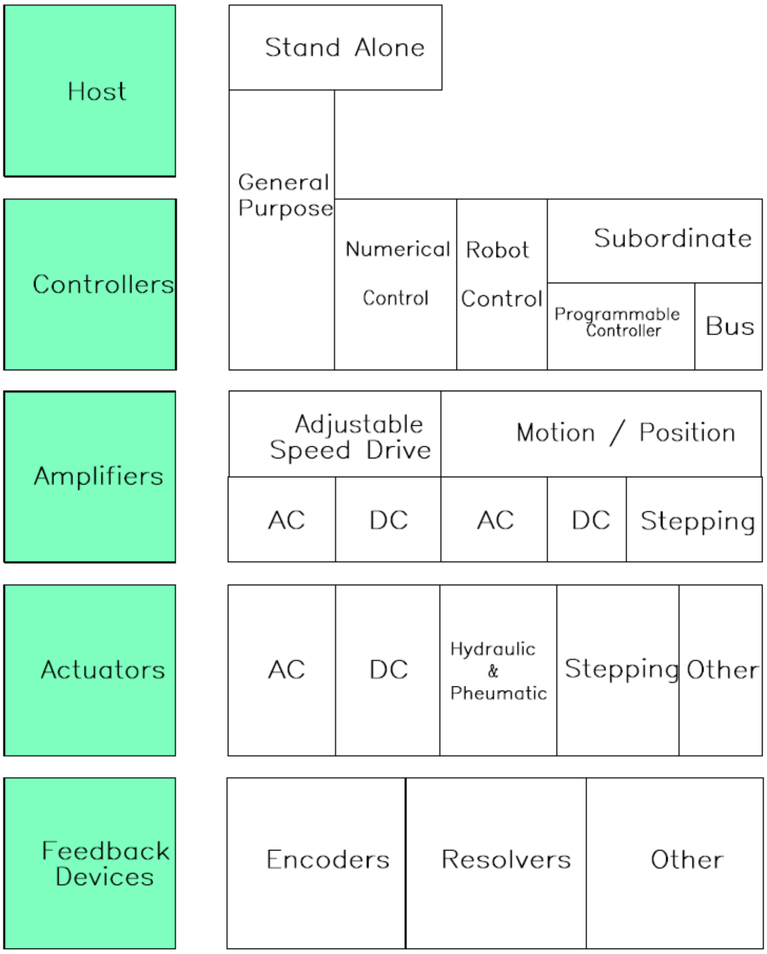


Figure 3.3 Typical Programmable Motion Control Systems

3.3.3 System Setup

Galil control system was constructed by a control card, two amplifier board and software. The control system must be set up before the physical attempts started.

The DMC1880 control card was installed in a computer and powered by PC motherboard. It has two 100-pin SCSI connectors for each set of 4 axes. Therefore, in order to control six actuators, two AMP19540 amplifier boards were connected. The first AMP19540 has been wired to four actuators (X, Y, Z, W) through MX, MY, MZ, MW motor output

4-pin power ports and the second controls the other two (E, F) through MX, MY motor output 4-pin power ports. Encoder of each actuator was connected to each 15-pin hi-density female d-sub X, Y, Z, W ports.

Figure 3.4 AMP19540

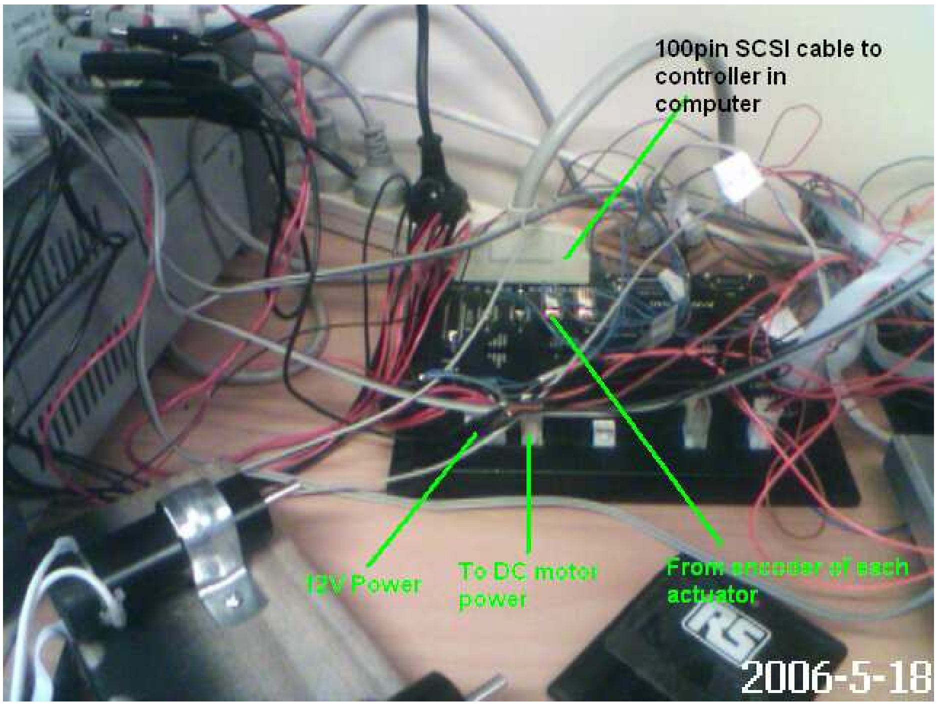


Figure 3.5 Single amplifier board connection

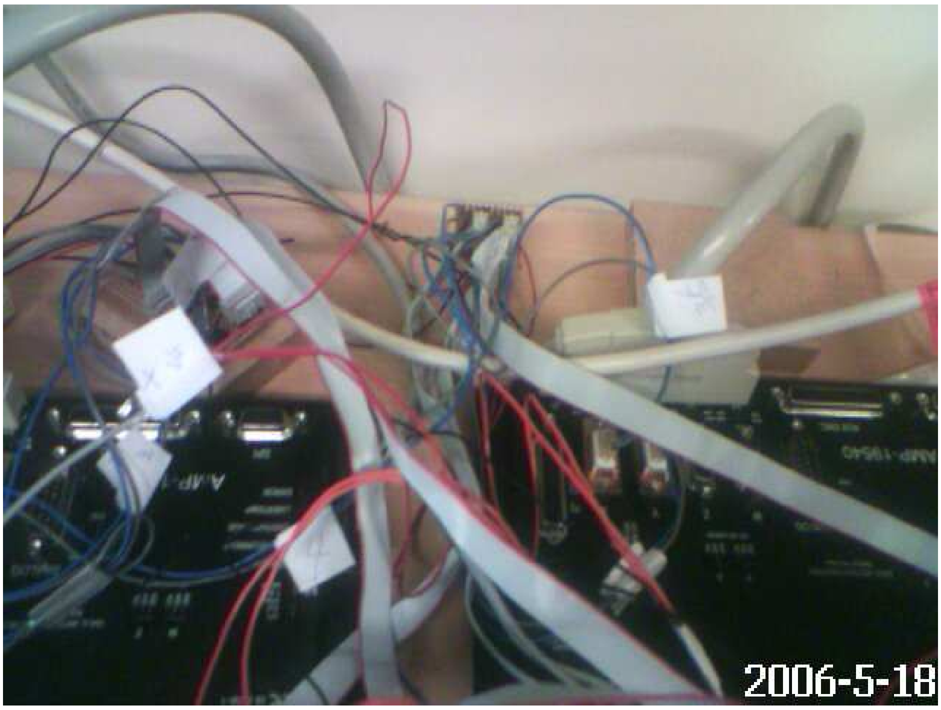


Figure 3.6 Dual amplifier boards connection

3.4 PID Tuning

PID controller has some important features, like providing feedback; the integral action is able to eliminate steady state offsets; the derivative action can anticipate the future. PID controllers are included in many special-purpose control systems. Typically, mode switches and anti-windup are more difficult in the features of PID control.

In the field of process control systems, it is well known that the basic and modified PID control schemes have proved their usefulness in providing satisfactory control, although in many given situations they may not provide optimal control. [32]

PID control is used at the lowest level; the multivariable controller gives the setpoints to the controllers at the lower level. Galil controller has given opportunities of additional features like automatic tuning, gain scheduling, and continuous adaptation. Auto-tuning means that the controller parameters are tuned automatically on demand from an operator or an external signal. Adaptation means that the parameters of a controller are continuously updated [31]. Tuning and adaptation can be done in many different ways. There are several methods available from Galil Windows Servo Design Kit providing the automatic PID tuning for a wide variety of systems and loads.

PID controller is adjusted on-site; there are many different types of PID tuning available in Galil WSDK software kit. The simplest move is trapezoidal velocity profile. The move is completely characterized by the distance, slew velocity, acceleration and deceleration rates. After manual tuning, delicate and fine results of PID values are made on-site. The accuracy is how close to the accepted or true value a measurement is. It specifies the maximum error in achieving a desired position.

The performance of any PID controller depends on the specific parameters of the system. Once these parameters change, a significant amount of effort required to manually may be tune the gains. In some systems, gains are usually kept very high in order to suppress the effects of change of load and disturbances. A simple fuzzy tuning procedure for the PID gains Kp, KI and KD can vary the PID gains according to the momentary robot configuration and assumes large gain values only when really needed[38].

Modern control theory has proved to be very useful in solving problems when the system is well defined, but it fails to cope with some complex processes due to the non-linearly and time- varying behaviour of the system. However, human operators often successfully control these complex systems. It would therefore seem logical to design a controller that is based on the experience of the human operator. The essence of fuzzy control is to exploit this human operator's knowledge [38].

3.5 Control Options

Basically, the motion control system’s purpose is to control any one, or combination, of the following parameters: Position/Velocity/Acceleration/Torque [29].

The types of feedback devices used in a motion control system will depend on both the control element (position, velocity, and/or torque), as well as the accuracy required. Another parameter used in the selection of a feedback device might be environmental considerations, such as temperature, sensitivity, and stiffness.

Modern control methodologies including nonlinear control, optimal control, variable structure control, and adaptive control for different types of motors to deal with the uncertainties, a conventional PID controller may not behave satisfactorily.

3.5.1 Velocity Control

Velocity control or speed control needs to be quantified with respect to several issues. First, what is the speed required to do the application? Further, does this load vary with speed or is load constant? In general, constant thrust (torque) over a fairly wide range of cutting speeds plus has a high speed requirement at low load for rapid traversing. This would result in an overall speed range of hundreds or thousands to one. Another consideration of velocity control is speed regulation. Speed regulation is generally expressed in percent of a set speed.

Although the system did allow us to control the motor torque of each joint, we preferred to conduct our experiments in velocity mode when possible. The use of velocity mode allowed for better high-gain trajectory tracking because of the increased feedback-loop rate and the use of the tuned proportional-integral controller provided by Mitsubishi. [25] Therefore, velocity mode experiments provided a better means of using feedback torque as a disturbance estimator for identifying model parameters. The torque data for the joints in velocity mode were communicated from the servo driver to the host computer as calculated by

$$T_m = k_T * I = Td * (0.001 Nm/digit) \quad (3.1)$$

We consider the model of the harmonic drive to be composed of friction, gravity, and stiffness. Because we are primarily concerned with low-velocity and low-acceleration applications (such as in surgery), the only torque components from the robot arm dynamics will be the gravitational torque, friction torque, and stiffness, no inertia. [25]

$$TinN = Tcf(\theta) + Tvf(\dot{\theta}) + Tg(\theta) + Tc(Tcf, Tvf, Tg) \quad (3.2)$$

Where Tin is the input torque, N is the transmission ratio (N is 50 for all joints), Tcf is the coulomb friction, Tvf is the velocity dependent friction torque, Tg is the gravity torque, and Tc is the torque used to deform the wave generator. We were able to achieve good velocity tracking even at very low velocities (<0.01 rad/s).

Velocity-Dependent Friction (friction–velocity relationship): each joint of the robot was commanded to move at a constant velocity and the mean torque required maintaining the velocity was taken to be the friction for that value of velocity.

Position-Dependent Friction: Friction in the HDT is strongly position dependent because of kinematics error in the transmission. Harmonic drives display kinematics error that causes the torque transmission characteristics of the drive to deviate from the ideal transmission model. This error is caused by a number of factors, such as tooth-placement errors on both the circular spline and flexspline, out-of-roundness in the three transmission components, and misalignment during assembly.

3.5.2 Torque Control

Torque control suggests the need to control the torque or force in a system independent of speed. An example would be a simple feed or take-up rolls in a web application for which web tension is controlled. Maintaining constant tension on the web results in varying torque at rolls as a function of roll diameter, result in a constant power requirement. A more complex tension control might require a changing or tapered tension as a function of roll diameter. As in the evaluation of a velocity-controlled system, a torque controlled system needs to be quantified by a number of parameters. The required torque range, the speed range must the torque be provided, and the character of the torque such as ripple of concern, and if so, the frequencies of ripple present a problem.

Control of the robot can be achieved in either “velocity mode” or “torque mode [25].” In velocity mode, the desired velocity for each joint is sent to the amplifier from the host computer. A high gain digital PI feedback loop running at 1504 Hz on the servo driver controls the joint velocity. In torque mode the desired joint torque (in this case, the motor torque constant times the motor current), is sent to the amplifier.

3.5.3 Position Control

Position control entails the control of motion displacement which is the change of motion with respect to time. This includes command, control, and the monitoring of motion. This can be as simple as the change in velocity command by limit switches on a simple slide drive, or as complex as linear and circular interpolation among axes on a multi-axis machine. Within the discipline of position control, numerous issues need to be quantified or measured. The resolution of the position control being the smallest unit of displacement needs to be defined. Along with the resolution, the accuracy and repeatability of the motion displacement needs to be determined. Resolution, accuracy, and repeatability are common quantities associated with position feedback devices like encoders and resolvers; but, the specification of a complete motion system must also take into account the mechanical system and position controller.

It typically involves motion acceleration, or the change in velocity with respect to time. The acceleration rate will affect the forces in the system since torque is the product of inertia and acceleration rate. It is important to include the inertia of the actuator (typically a motor) in any torque calculation of this kind since its inertia may contribute considerably to the torque required.

Flexibility regarding the need to change certain parameters of the required motion is also required by position control. For example, the length of a move, or speed of the system may be changing based on variables in the process or parts being manufactured. For this reason, a programmable motion controller is needed along with specific application software. The application software can be canned or bundled software that might come with the system, or custom software requiring additional cost or effort. When considering flexibility in programming, it is also important to define the degree of operator interface for implementing changes. A simple operator interface could be a thumb wheel switch allowing selection of particular predetermined functions. A more complex and flexible interface might entail an alphanumeric display and keypad.

3.5.4 Force Feedback Control

Many applications require some sort of force feedback control in order to provide a proper accurate amount of torque applied at all time. With the torque control system, the motor not only offers a constant torque, but also protects the motor from overloading. There are two basic approaches to complete the force feedback loop [22]: use force feedback sensor directly or use an encoder on the motor and close an outside sensor loop.

In the first case, the sensor just needs to be simply connected to the system but the position of the sensor must be well selected as it should always be in contact with the object being applied the force. Another issue may be the gain of the sensor. Tuning may prove difficult in some sensor configurations because of the high gain of the sensor. The advantages of this configuration are loop update speed and ease of setup.

In the second choice (Figure 3.7), the motor is always in control. Program on the controller reads the sensor input from the analog inputs and there is no need for the sensor to touch the object and tuning is a simple task. However, the system can be slow down and update rate will decrease from 1ms in the first case to 2-20ms.

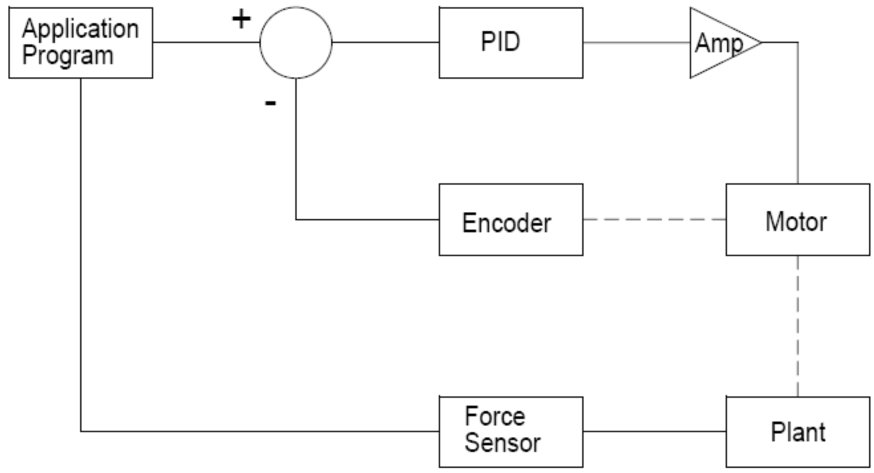


Figure 3.7 Simple control diagram of a system with an outside force sensor loop

An example of Force Feedback Control is a Bone-drilling experimental system designed by W.Y Lee [23]. The drilling motor may in fact stall if the feed rate is exceeded or too high, and one approach to prevent the situation is add a drilling motor torque feedback. The drilling motor will run at a higher speed in order to prevent a stall then the torque controller will maintain a constant cutting force with variable speed. If the drilling motor's output torque never achieves the torque command, then the drilling motor will run at its maximum speed.

With this torque control scheme, the drilling motor can be protected from overloading as well as provide constant drilling torque. The constant torque output will be kept up even when the motor is stalled. Thus, the motor will not be burnt down under any operational case. Moreover, in the case of drilling a porous bone, the output drilling torque may change unexpectedly and rapidly, depending on the robot's feed forward speed and the drilling target. If a lag-lead controller is used, then the controller will result in many overshooting torques during the drilling process and cause a lot of noise.

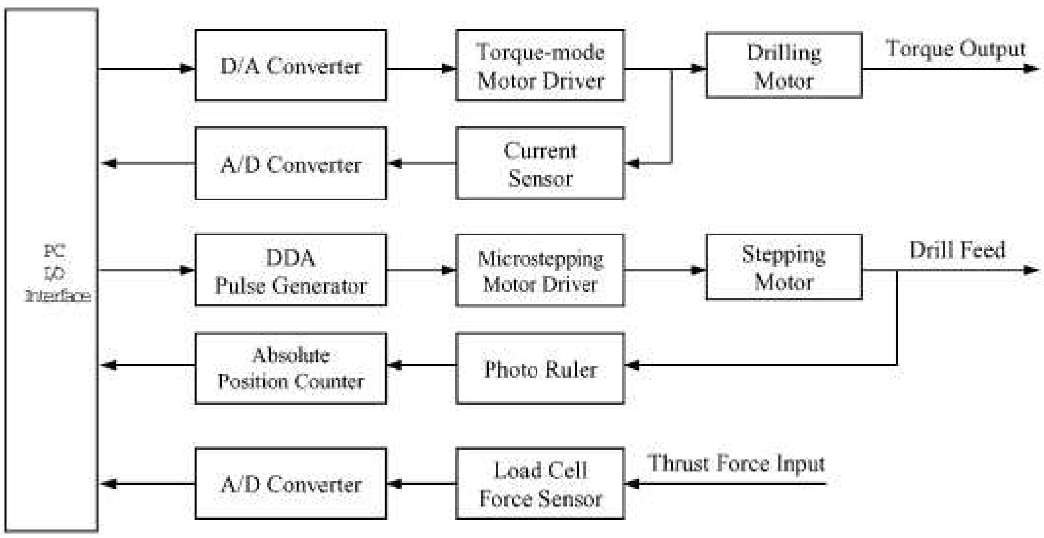


Figure 3.8 Hardware block diagram of the bone-drilling system

3.5.5 Feed Forward Control

Nowadays with development of usage of high speed feed drive machining and ultra precision machining is required high acceleration, high speed, fast response, and very small tracking error in such systems. Controllers take action to an error before it causes disturbance to processes.

Most of the servo systems in industrial mechatronic are being controlled by PID method with feedback to controller, which is called "Model Base" controller.

Figure 3.9 illustrates the overall feedforward control diagram. This controller is being used for fast response of feedforward loop in affront of un-modeled disturbance respect to feedback loop. Because the ripple, friction and unknown force are ignored in feedback controller, they should be compensated by this controller. This controller is a neural network adaptive controller. Feedforward controller includes feedback loop. At the beginning of training, feedback controller is controlling the system and then in continues by increasing the training process; the system control will be dominated exactly by feedforward controller [24].

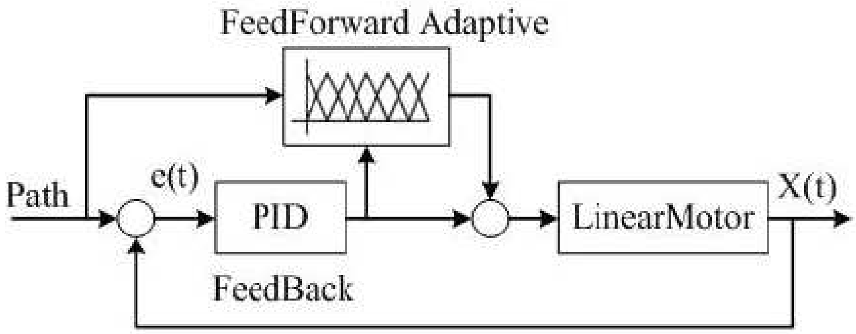


Figure 3.9 Overall feedforward control diagram

The main advantages of this control algorithm is no change in the controlled variable at all, disturbances have no effect on the process. One drawback of this configuration is the measured signal goes to the controller parallel to the process. Other drawbacks include complex Computation and, unknown and immeasurable disturbances in advance.

3.5.6 Cascade Control

Cascade control (CC) was first introduced many years ago by Franks and Workey [39] which is an alternative to conventional single feedback control to improve the performance of a control system particularly in the presence of disturbances.

In conventional single feedback control, the corrective action for disturbances does not begin until the controlled variable deviates from the set point. A secondary measurement point and a secondary controller, Gc2, in cascade to the main controller, Gc1, can be used to improve the response of the system to load changes, D2. as shown in Figure 3.10

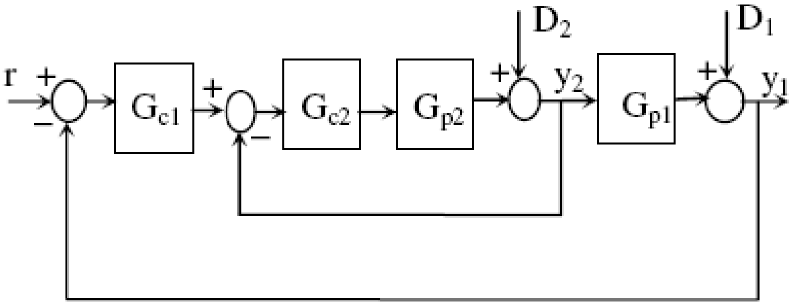


Figure 3.10 Cascade Control System

The system has two loops. The outer loop deals with the primary measured signal, and the response of inner loop is quite faster than the outer loop. Cascade control is especially effective if the inner loop is much faster than the outer loop and if the main disturbances affect the inner loop first. It is not a simple feedback control strategies and can be used when there are several measurement signals and one control variable. Also, it is possible to use high loop gains in the inner loop and the dynamics for setpoints to process output can be sometimes faster than open loop dynamics, thus, higher controller gains can be used in the outer loop.

There are still some disadvantages for the cascade control, the main one is we need carefully chose the secondary measured variables. The essential disturbances should act in the inner loop and it should be possible to have a high gain in the inner loop. Others include difficulty of tuning. Cascade controller must be tuned in a correct sequence. The outer loop should first be put in manual when the inner loop is tuned. The inner loop should then be put in automatic when tuning the outer loop.

3.6 Modelling and Simulation of Robotic Actuators

3.6.1 Robotic Mathematical Modelling

A real DC motor is a converter which can convert current into rotational torque [26]. Motor system identification allows building mathematical models of a dynamic system based on measured data [27]. A virtual motor system can be simulated by computer software and performs virtual tests to prevent the risks of having mistake in real demonstrations. A virtual motor block will be modelled with sophisticated PID control technology

Modelling a DC motor can help us to identify the system parameters and then attempt to design an optimum controller for the motor.

The actuator in the jaw control system contains the DC motor. It provides rotary motion, and coupled with a crank or a gearbox can further provide linear transition. The electric circuit of the armature and the free body diagram of the rotor can be seen from the following picture.

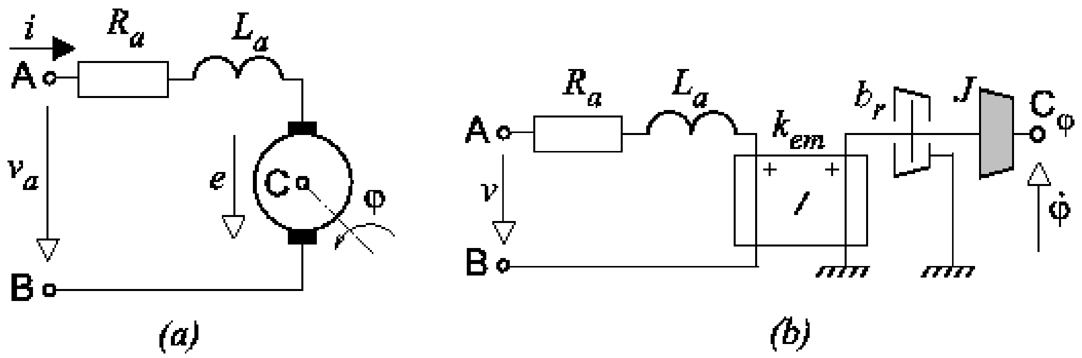


Figure 3.11 Dynamic model of a DC permanent magnet machine

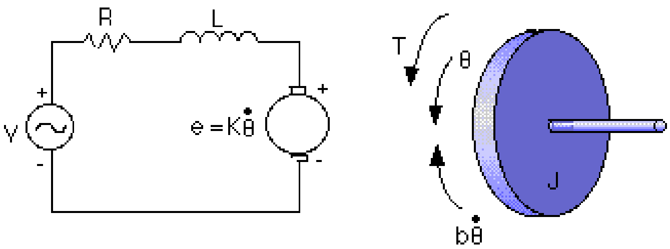


Figure 3.12 Normal DC motor armature

Figure 3.11 and Figure 3.12 show a configuration of a permanent magnet DC machine in which **R** and **L** represents the armature circuit resistance and inductance, respectively. Other values should be allocated into following physical parameters: rotor inertia (**J**); damping ratio of the mechanical system (**b**); torque constant (**Km/Kt**); back emf constant (**K_e/K_b**);

The equation below presents the calculations of motor torque (T_m) and back emf (e):

$$e = K_e \omega = K_e \frac{d\theta}{dt} \tag{3.3}$$

In Equation 3.3, T_m (N-m or oz-in) is the motor torque; K_t (N-m/A or oz-in/A) is the torque constant; I (Amps) is the current through the armature; K_e (V/rpm) is the back EMF constant; and ω (rpm) is the armature speed; Viscous friction coefficient b (N-m/rpm); Moment of Inertia J (N-m-s^2 or Kg-m^2).

Using Newton's law of Motion and Kirchhoff's current law, the linear equation can be written as:

$$\begin{aligned} J\ddot{\theta} + b\dot{\theta} &= K_t I - T_{applied} \\ L\frac{dI}{dt} + RI &= U_n - K_e \omega \end{aligned} \tag{3.4}$$

The law applies to the motor system give the following equations:

$$
\begin{aligned}
&J\frac{d^2\theta}{dt^2} = T_m - b\frac{d\theta}{dt} - T_{applied} \Rightarrow \frac{d^2\theta}{dt^2} = \frac{1}{J}(K_t I - b\frac{d\theta}{dt} - T_{applied}) \\
&L\frac{dI}{dt} = U_n - e - RI \Rightarrow \frac{dI}{dt} = \frac{1}{L}(U_n - K_e\frac{d\theta}{dt} - RI)
\end{aligned}
\tag{3.5}
$$

In state-space representation of the dynamic system, the equations above can be expressed by choosing the rotating speed and electric current as the state variables and the voltage and applied torque as inputs. The output is chosen to be the rotating speed.

$$
\begin{aligned}
&\frac{d}{dt}\begin{bmatrix}\dot{\theta} \\ I\end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{K_t}{J} \\ -\frac{K_e}{L} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}\dot{\theta} \\ I\end{bmatrix} + \begin{bmatrix}\frac{1}{L} & 0 \\ 0 & -\frac{1}{J}\end{bmatrix}\begin{bmatrix}U_n \\ T_{applied}\end{bmatrix} \\
&y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\dot{\theta} \\ I\end{bmatrix} + \begin{bmatrix}0 & 0\end{bmatrix}\begin{bmatrix}U_n \\ T_{applied}\end{bmatrix}
\end{aligned}
\tag{3.6}
$$

3.6.2 Motor Parameter Identification

The key parameters of a typical DC motor are in Table 3.3. A brushless motor was first selected in this system and then the brushed Maxon DC RE-30 motor was used to replace it due to the difficulty of finding a suitable encoder. Motor torque capability is expressed by continuous value and the peak value. The continuous torque is what the motor can produce continuously without overheating and the peak value is the maximum that can be generated without causing mechanical damage.

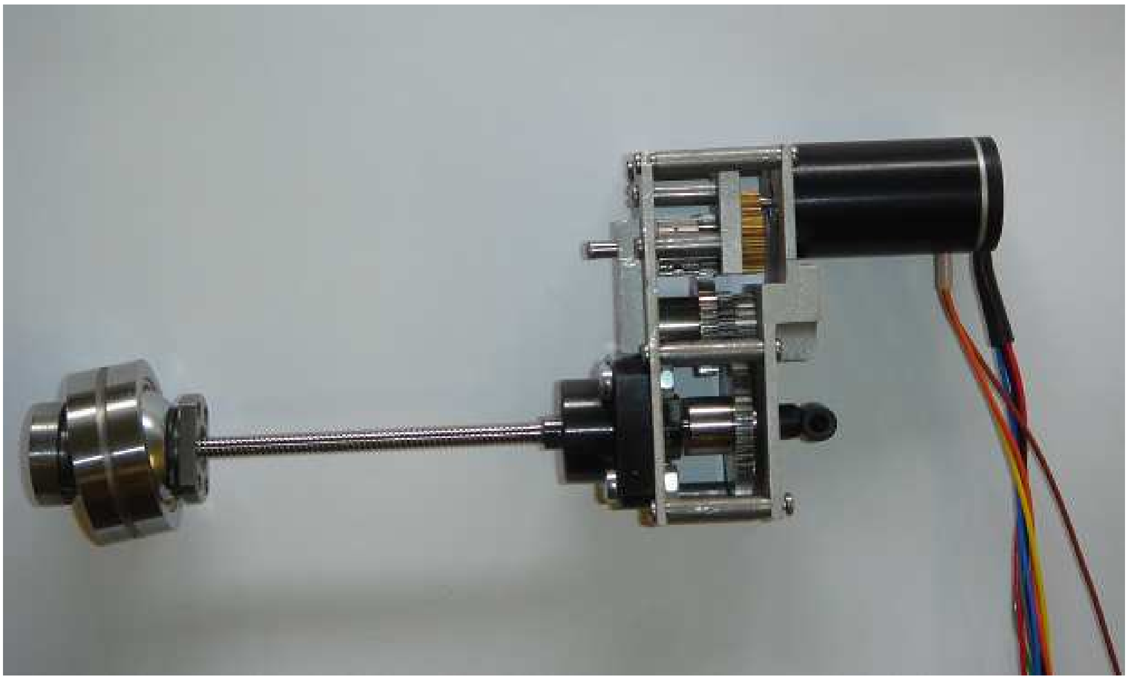


Figure 3.13 Brushless Motor

Table 3.2 Characters of Actuators

	Sy	Required	Brushless	Brush
Actuator type			Linear	Crank
Actuator stroke		20mm	>20mm	N/a
Motor type			Brushless	Brushes
Motor speed			49’000RPM	8’810RPM
Gear ratio	G_r		10:1	66:1
Gear end speed	G_s		81.7RPS	2.2RPS
Actuator velocity	A_v	70mm/s	81.7mm/s	14rad/s
Actuator acceleration	A_a	3000mm/s^2		
Actuator force	A_f	150N	>150N	>150N
Minimal feed		0.01mm	0.01mm	0.01mm
Angular movement at joints		25^o		

Table 3.3 Essential characters of DC motors

		Brushless	Brush
Model		Faulhaber	Maxon motor RE 30
Type		2036 012B/Brushless	∅30mm/Graphite brushes
Power rating	P	20W	60W
Nominal voltage	U_N	12Volt	24Volt
No load speed	n_o	49’000/min	8810/min
Stall torque		N/A	1.02N-m
Torque constant	K_m/K_t	0.898 oz-in/A	25.9 mN-m/A
Back-emf constant	K_e/K_b	0. 664mV/rpm	1/369 V/rpm
Rotor inertia	J	$2.761*10^{-5}$ oz-in-sec^2	33.3 g-cm^2
Terminal resistance	R	3.4Ohm (phase-phase)	0.661Ohm
Terminal inductance	L	148uH(phase-phase)	0.12mH
Viscous friction coefficient/dynamic	C_v/B	$3.030*10^{-6}$ oz-in/rpm	0.115mNm/rpm
Max Load	$M_{e\ max}$	0.736 oz-in	86.2mNm

If the torque constant is specified in N-m / A and the back EMF constant in V-sec/rad, then both the constants can be replaced by a constant K_{EM} called *electromechanical constant* :

$$K_{EM} = K_t = K_e = 1 / K_v$$

As we know 1rpm= 0.10472 rad/s. When K_t is specified in N-m / A and the K_e in V/rpm, we can derive:

$$K_e = 0.10472\ K_t$$

With Kt in units of oz-in/A, Ke in units of V/rpm:

$$K_e = K_t/0.0013524$$

In addition:

$$1\ \text{oz-in-sec}^2 = 0.007062\ \text{kg-m}^2$$

$$1\text{kg-m}^2 = 0.102\text{kg-m-s}^2 = 1\text{N-m-s}^2$$

$$1\ \text{oz-in} = 0.007062\ \text{N-m}$$

$$1\text{rad/s} = 9.55\text{rpm}$$

Then, according to international System of Units, we will have the following table:

Table 3.4 Essential characters of DC motors (SI)

		Brushless	Brush
Model		Faulhaber	Maxon motor RE 30
Type		2036 012B/Brushless	∅30mm/Graphite brushes
Power rating	P	20W	60W
Nominal voltage	U_N	12Volt	24Volt
No load speed	n_o	49’000RPM	8810RPM
Stall torque		N/A	1.02N-m
Torque constant	K_m/K_t	0.006341 N-m/A	0.0259N-m/A
Back-emf constant	K_e/K_b	0.000664 V/rpm	0.00271V/rpm
Rotor inertia	J	0.000000195 kg-m^2	0.00000333 kg-m^2
Terminal resistance	R	3.4Ohm (phase-phase)	0.661Ohm
Terminal inductance	L	0.000148H (pha-phase)	0.00012H
Friction coefficient	C_v/B	0.0000000214N-m/rpm	0.000115N-m/rpm
Max Load	$M_{e\ max}$	0.0052Nm	0.0862Nm

3.6.3 Simulation and Visual Testing

MATLAB/SIMULINK is one of the most successful software packages currently available for performing mathematical computations and plotting capabilities for displaying information [28]. For our exposition we use a pc with Windows XP on which MATLAB version 7.0 and SIMULINK version 7 have been installed.

The armature voltage Ea is input to the block. The output information is the multiplexed signal, containing information about the Speed of the Motor shaft “v” in RPM, Position of the Shaft “Pos” in Revolution, and Output “Pos_rad” in rad respectively. The model of the permanent magnet DC motor is as shown in Figure 3.14.

We require the steady state error of the motor velocity should be less than 1 %(0.01mm) when the motor rotates at the desired speed. The other performance requirement is the acceleration of the motor. In this case, the settling time have to be of 2 seconds. Finally, in order to prevent high speed damaging the motor, we will have an overshoot smaller than 5%.

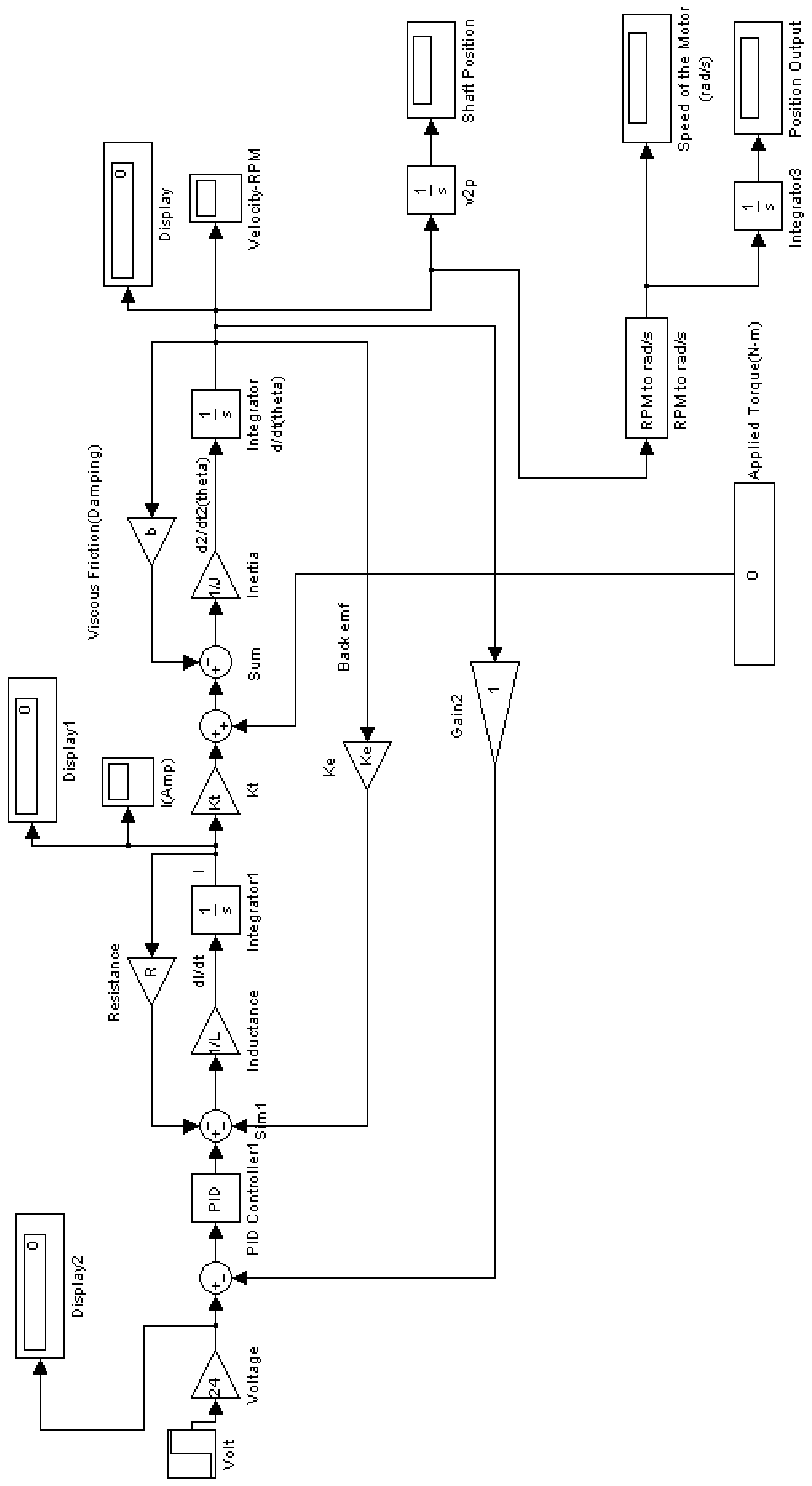


Figure 3.14 MAXON RE 30 (310007/268214) Graphite Brushes 60W DC Motor Block

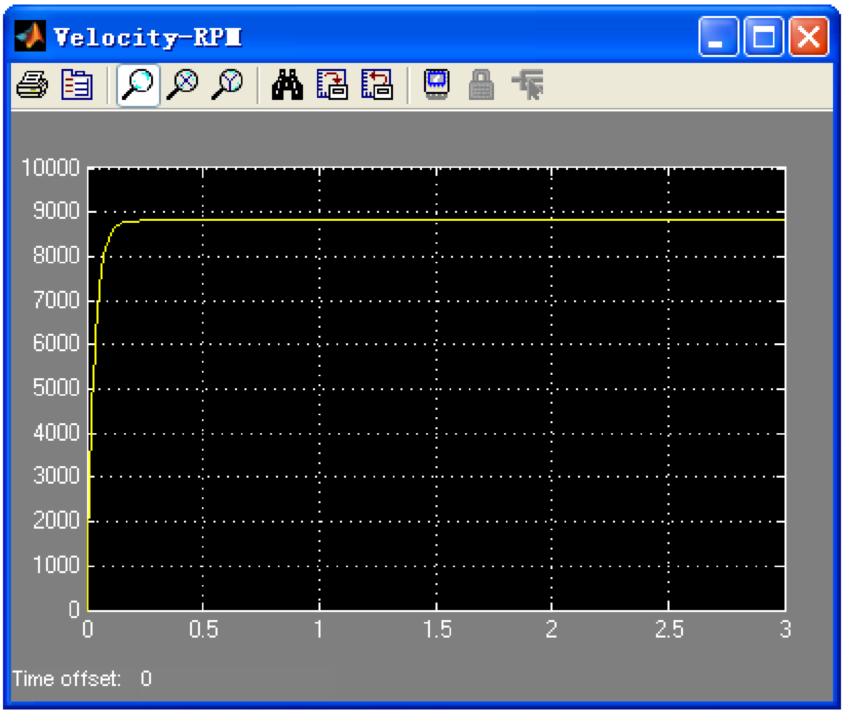


Figure 3.15 Open loop simulation(No load)-speed of motor

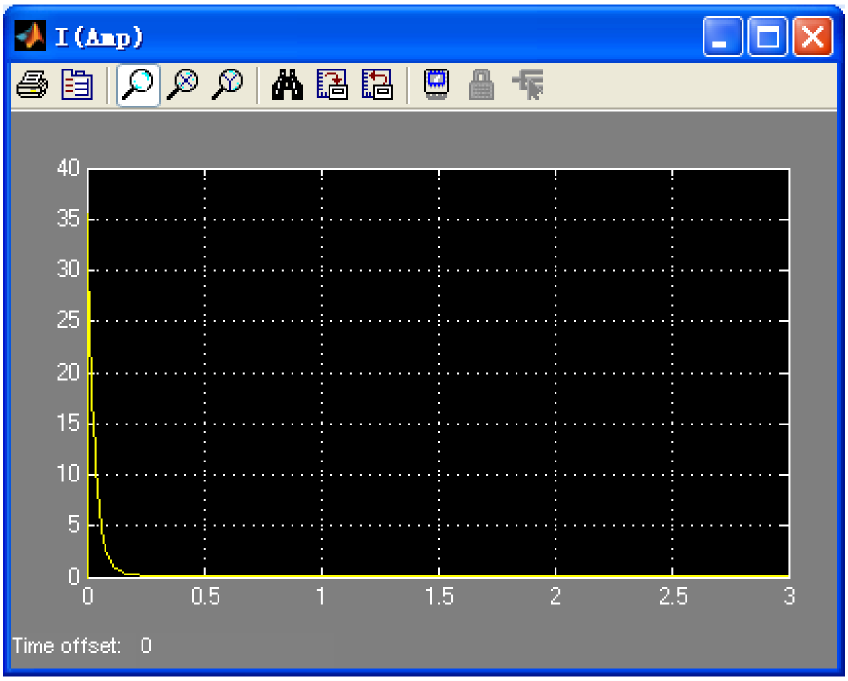


Figure 3.16 Open loop simulation(No load)-Current of motor

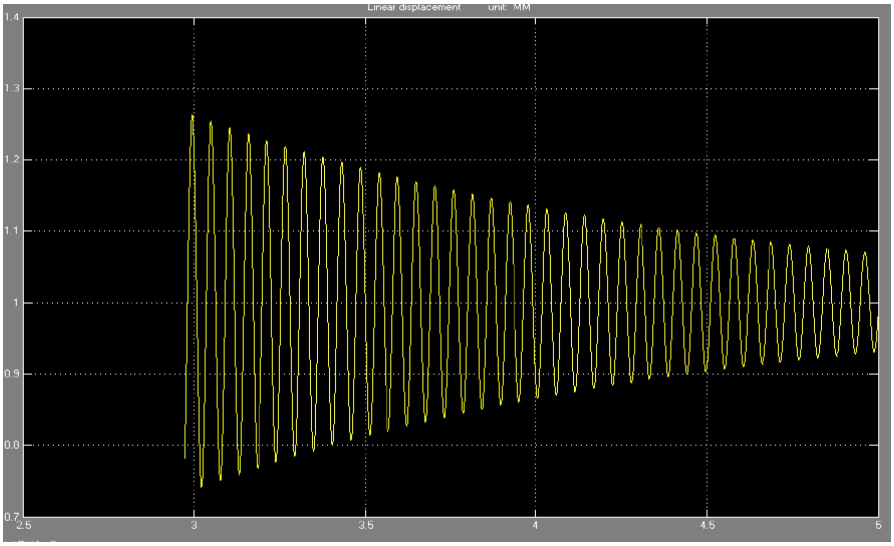


Figure 3.17 PID tuning simulation(No load)

3.7 Conclusions

The Massey robot jaw which is composed of six Massey designed actuators, one Galil controller board, two Galil amplifier board and software running on a PC has been set up and connected. Each actuator has a 60W DC motor attached with an encoder, and a 66:1 gearbox plus a bevel gearbox in front of each motor unit. All six actuation units were oriented in the same direction.

At first, each actuator was removed from the main structure and independently tested through WSDK, a Galil control kit, to automatically find proper PID values to achieve stable motor spin in a simple speed control mode, where the speed is set in advance. Some research regarding on the control theories were done and there is a find the Contour Control Mode provided by Galil can perform a required control under PID mode in most circumstances. Therefore the subsequent trajectory tracking experiments are mainly operated under the Contour Mode.

The controller in this system was also tried to be software written in Matlab. This idea is to minimize the physical experimental time of the controller. A simulation test shows less efficiency of a simple PID controller. However, after a series of experiments done through the Galil controller, it is found that the traditional PID control method can fully control the system. The difference is due to the internal design of the Galil controller; it is hard to model it. Therefore, only some simple motor control methods, ex: PID control, were tested.

Although the test results are different, motion of motors can still be simulated, some force tests could be applied on a single motor and avoid potential hazards. If a proper controller model could be found, the model can further be synchronized with main process event from the jaw. Furthermore, some control algorithms for a mastication robot to simulate these adaptive human jaw motions could be attempted. The algorithm could includes the jaw opening and closing angle and the jaw opening reflex motion are defined by using the monitored biting force, not only as functions of the chewing angle.

Chapter 4 Trajectory Tracking and Experiments

4.1 Introduction

The purpose of the experiments was to test if the robot could follow a real masticatory movement, the robot could chew on foods of different texture properties and the robot could have a compliant ability in giving way to the very hard food in next chapter. In the following experiments the masticatory movement to be reproduced by the robot is specified by the trajectory of incisor point (IP) (or Om in Figure 2.4(a).). The masticatory movement was modified from a masticatory measurement made with a subject to chew a hard food. The mastication lasted 45.32 seconds and was sampled in 10 ms and 8ms.

Given the above mandible movement, inverse kinematics was solved to find individual trajectory of each of the six crank actuations required for the robot to follow the mandible movement. As Galil controller used can only implement a trajectory of time interval of 2^n ms, where n is an integer between 1 and 8, the actual mandible movement that was sampled in 10 ms needs to be interpolated before its implementation in Galil. Cubic polynomial was found able to convert 10 ms interval trajectory to executable 8 ms interval trajectory without significant distortion in motion profile (in terms of angular displacement, velocity and acceleration).

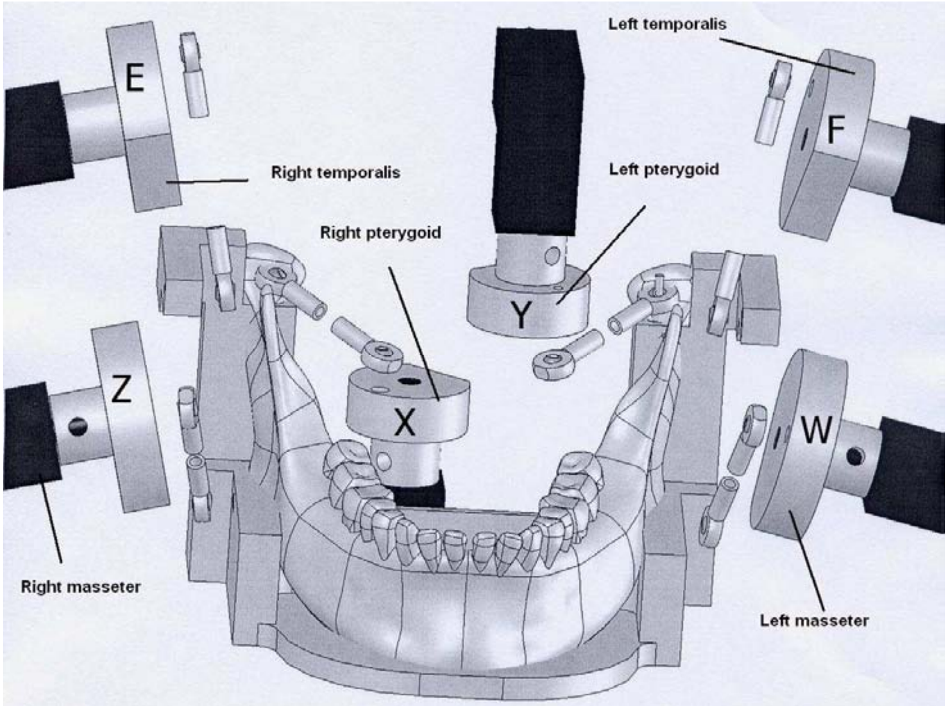


Figure 4.1 Components making up the Massey Jaw Robot

4.2 Basic Chewing Trajectories

A variety of human chewing trajectories can be measured from different positions on the mandible, such as incisor teeth or molar teeth. They are necessary tools for specified purposes. For example, the WY (Waseda Yamanashi) serious robot [3, 4, and 5] is for the treatment for jaw disorder patient.

A chewing trajectory of the bottom incisor was inducted in order to calculate the displacement change, velocity, acceleration, and angular movement of each of six actuators while chewing [14] and it is named “Motion 2003”. “Motion 2003” is a simulated jaw chewing behaviours and consists of thousands position segments with 10ms interval each.

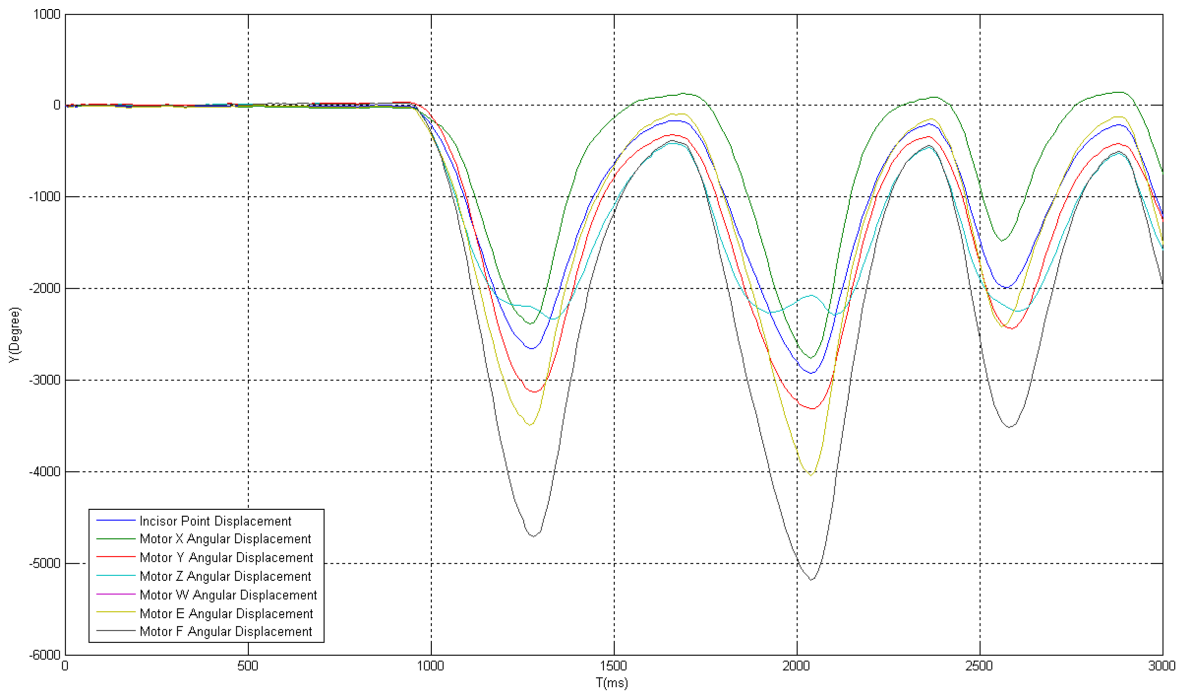


Figure 4.2 Trajectory displacement compare between six motors and incisor point

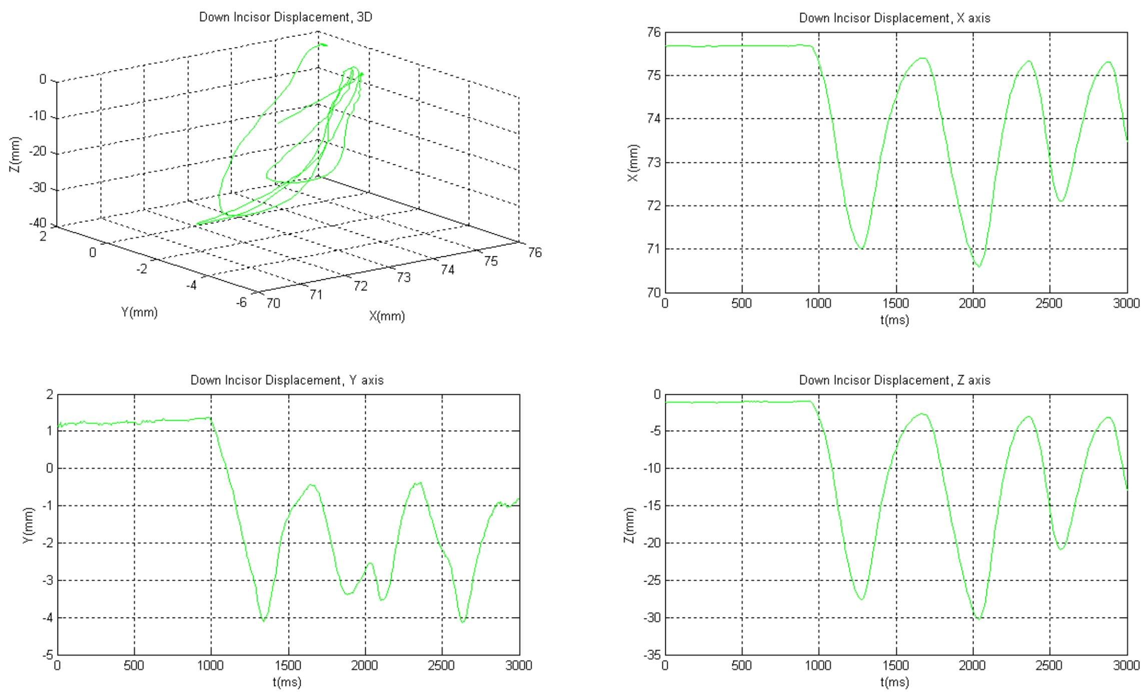


Figure 4.3 Trajectory displacement of incisor point in X,Y,Z axis and 3D

4.3 Trajectories and Control Problems

According to the control specification of the Contour Mode provided by DMC-1880 Galil controller, it is found that the selectable position segment time intervals have to be defined as 2^n ms, where n is an integer between 1~8 [33]. Therefore, "Motion 2003" trajectories with 10ms interval can no longer work properly on the Galil controller or have to be resolved and fitted to 2^3=8ms which is close to 10ms in order to perform synchronized contour control between the Solidwork simulation and the real motion control of the robotic jaw.

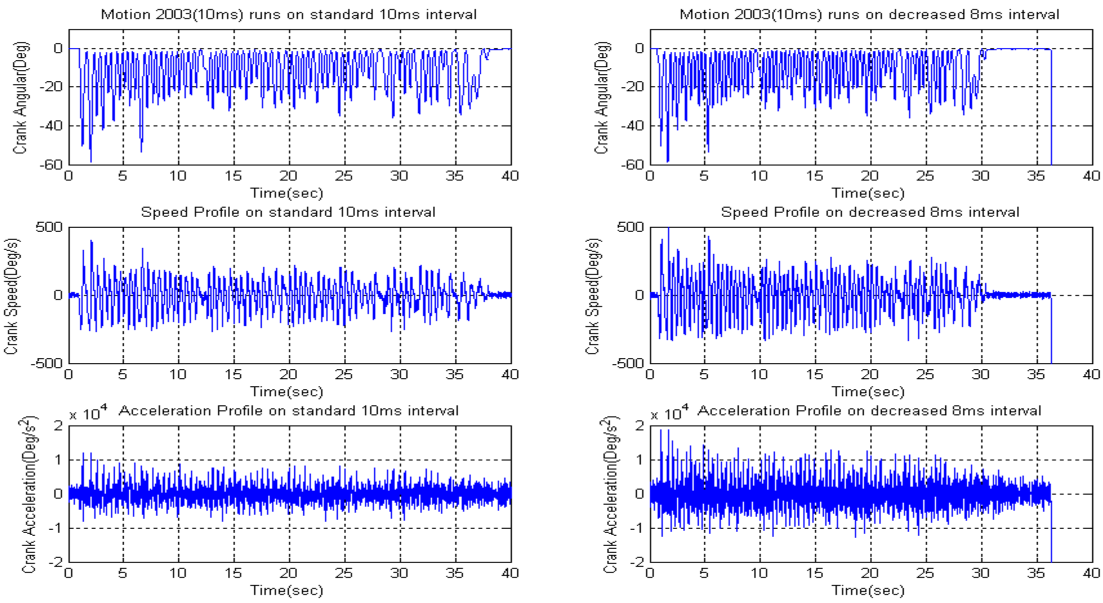


Figure 4.4 Mathematical Motion Profiles Analysis of "Motion 2003" Crank E runs on standard 10ms segment interval and fast 8ms segment interval during the full simulated run with auto-adapted speed and acceleration

The unsynchronized phenomenon can be seen from Figure 4.4. When a position segment implementing time decreased to 8ms (right sub-figures), the entire motion finishes quicker and is in 80% designed time. In Figure 4.4, length of trajectory profile is like "compressed" but for the speed profile, it has increase change on its amplitudes, averagely, 25%. The maximum speed can reach up to 500 ^{o}/s. According to the actuator property (Table 4.1), the peak speed is out of the range, and finally can cause mechanical faults. The acceleration profile is much worse, average amplitude jumps 65% and more points exceed the actuator tolerance.

Table 4.1 Actuator Property

	Crank (rad)	Crank (Deg)	Motor Shaft(rad)	Motor Shaft(Deg)	Motor Encoder (2000)
Maximum Angular Movement	1.308r	75^{o}	86.35r	4950^{o}	275000
Angular Movement Accuracy	0.00175r	0.1^{o}	1.155r	6.6^{o}	37
Maximum Angular Velocity	8rad/s	$458.6\,^{o}/s$	528 rad/s	30267.6 $^{o}/s$	168153 counts/s
Maximum Angular Acceleration	$200rad/s^2$	$11465\,^{o}/s^2$	13200 rad/s^2	756688114 65 $^{o}/s^2$	4203822 $counts/s^2$

A new trajectory can be made from other measurements but it costs time to retrace and rebuild motion. Another applicable is called trajectory interpolation. From the research, interpolation was already used by the contour mode inside of the controller. A function in contour mode automatically linear interpolates trajectory segment to 1 millisecond path long regardless the original length of trajectory segment. As a result, interpolation way has been selected to generate the new crank trajectories and the original Motion 2003 will still be used as a faster motion control to be compared in the results of trajectory, speed and acceleration profiles.

4.4 Interpolations

There are several kinds of interpolant that will be discussed below: linear, cubic, and spline. They are the most commonly used interpolants in Matlab which we used to make new crank trajectories with.

An interpolation form “Collins English Dictionary” is [35]: 1. Insert between or among others. 2. Change by putting in new material. 3. Estimate a missing value by taking an average of known values at neighboring points.

Given a sequence of n distinct numbers x_k called nodes and for each x_k a second number y_k, we are looking for a function f so that [36]:

$$f(x_k) = y_k, k = 1,...,n$$

A pair x_k, y_k is called a data point and *f* is called the interpolating function.

When using the Matlab interpolates 10ms interval crank trajectories to 8ms interval crank trajectories, information of speed (first derivative) and acceleration (second derivative) are also changed as well. Then, it is necessary to control new speed and acceleration under the specified values nominated for the cranks (Table 4.1) to prevent linkage broken and control errors. On the other hand, interpolated trajectories of actuators need to be correctly calculated by means that no dead positions will be reached and the motion profile needs to be as same as the original one(Motion 2003) running on simulated 10ms interval motion, especially at these Maximum Intercuspation (MI).

4.4.1 Piecewise Linear Interpolation

Piecewise linear interpolation is one of the simplest methods. Generally, it takes two data points and there is $x_k \le x < x_{k+1}$ while *k* is the interval index. Then the interpolant is given by

$$f(x) = y_k + (x - x_k)\frac{y_{k+1} - y_k}{x_{k+1} - x_k} \qquad (41)$$

For example, x_1 =1, y_1 =0.22, x_2 =2, y_2 =0.88. Consider the two points and determine $f(1.5)$. Since 1.5 is midway between 1 and 2, it is reasonable to make $f(1.5)$ midway between $f(1)$=0.22 and $f(2)$=0.88, which is 0.55.

This is clearly a linear function that passes through (x_k, y_k) and (x_{k+1}, y_{k+1}).

Linear interpolated crank trajectories have been converted by "ConvertToLinInterpolation.m" (Appendix 4.1) in Matlab.

4.4.2 Cubic Interpolation

Polynomial interpolation is a generalization of linear interpolation [36]. In Matlab, Piecewise Cubic Hermite Interpolation Polynomial is also called pchip. It is a shape-preserving interpolant.

We have $f(x) = y_k + (x - x_k)\dfrac{y_{k+1} - y_k}{x_{k+1} - x_k}$ in formula equation (41)

Let d_k denote the slope of the interpolant at x_k:

$$d_k = f'(x_k)$$

For piecewise linear interpolation, $d_k = \dfrac{y_{k+1} - y_k}{x_{k+1} - x_k}$

In the polynomial interpolation, we need to define the slopes d_k so that the function values do not overshoot the data values in the known area.

Cubic interpolated trajectories have been converted by "ConvertToCubInterpolation.m" (Appendix 4.2) in Matlab.

4.4.3 Spline Interpolation

One other interpolating function is a cubic spline. A reference text for both the mathematics and the software is *A Practical Guide to Splines* by Carl de Boor [37]. De Boor is also the author of the spline function and the Spline Toolbox for Matlab.

In the spline functions distributed with Matlab and the Spline Toolbox, the slopes are computed by the Matlab bash-slash operator.

Figure 4.5 shows the compare results of the spline interpolant and pchip interpolant. We can see the first derivative of spline is smooth, while the first derivative of pchip is continuous but has some kinks [34]. In the second derivative (acceleration), the spline is still continuous, while the pchip has jumps at every knot.

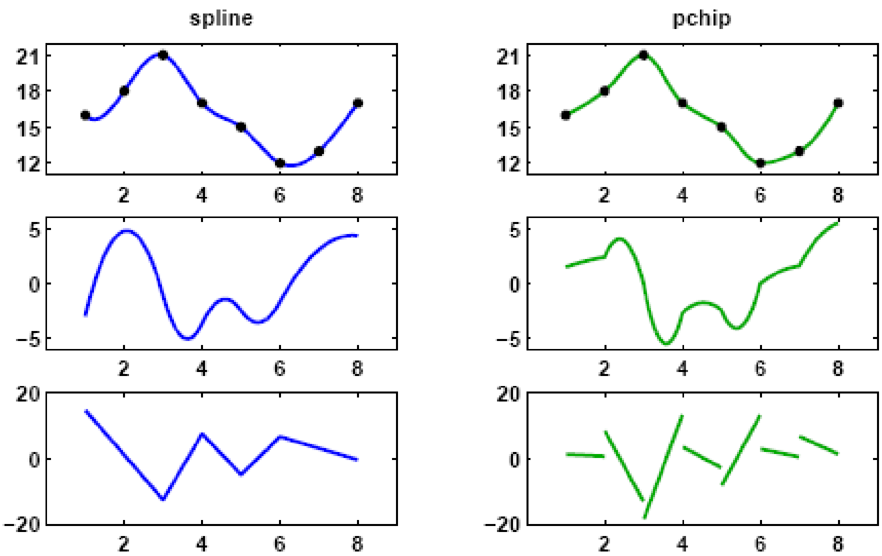


Figure 4.5 Spline and pchip interpolants, and their first two derivatives

Spline interpolated trajectories have been converted by “ConvertToSpiInterpolation.m” (Appendix 4.3) in Matlab.

4.4.4 Mathematical Analysis

So far, three groups of interpolated crank trajectories (linear, cubic, and spline) have been generated by the Matlab. Assuming in a random period of time, trajectory profile on one actuator is representative of all actuators in entire motion time. Then, motion profiles of crank F (Figure 4.6) was selected to analyse as it is one of the actuators that provide most force during occlusal area (0.5mm from the MI), and the motion time period is from 4^{th} to 7^{th}.

In past research, it is proven that the top point of each crank angular displacement curve is the maximum intercuspation (MI). In other words, in a chewing cycle, the occlusal phase ends 0.5mm from MI in vertical opening. Ogawa et al. (2001).

- Motion Profiles Overview

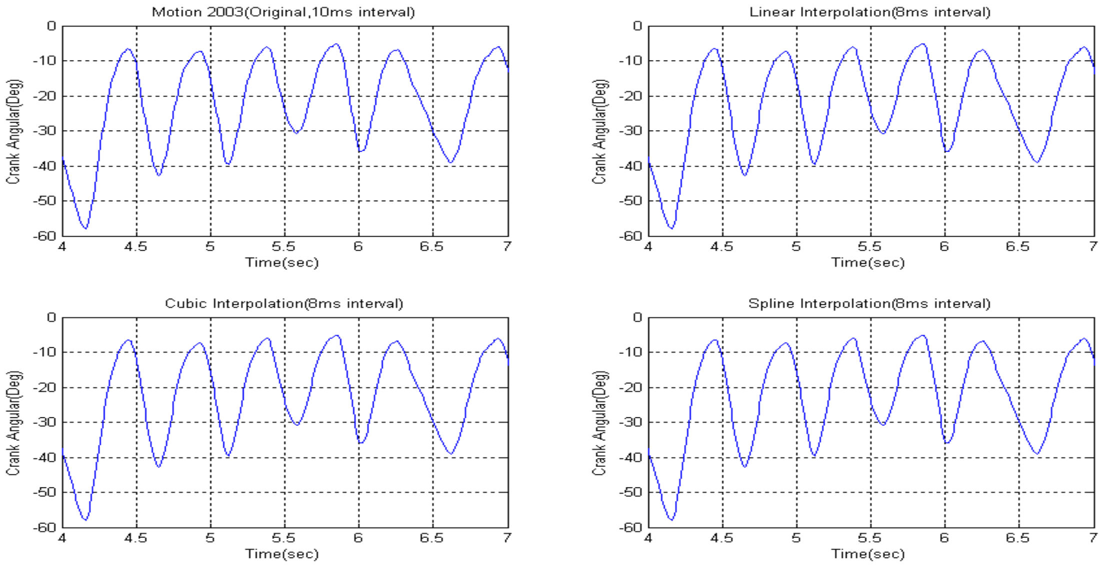


Figure 4.6 Crank F Trajectory Profiles from 4^{th} second to 7^{th} second

Figure 4.6 displays three mathematically interpolated crank trajectories and the original crank trajectory for crank Z between 4^{th} second and 7^{th} second from Motion 2003. Each three-second trajectory includes six Maximum Intercuspations (MI) where are between about -8^{o} to -6^{o}. The overall impression of all interpolated trajectories is favourable. There are no new breaks, jumps, peak cuts or obvious shape changes, which means interpolation is acceptable for the new jaw crank trajectories.

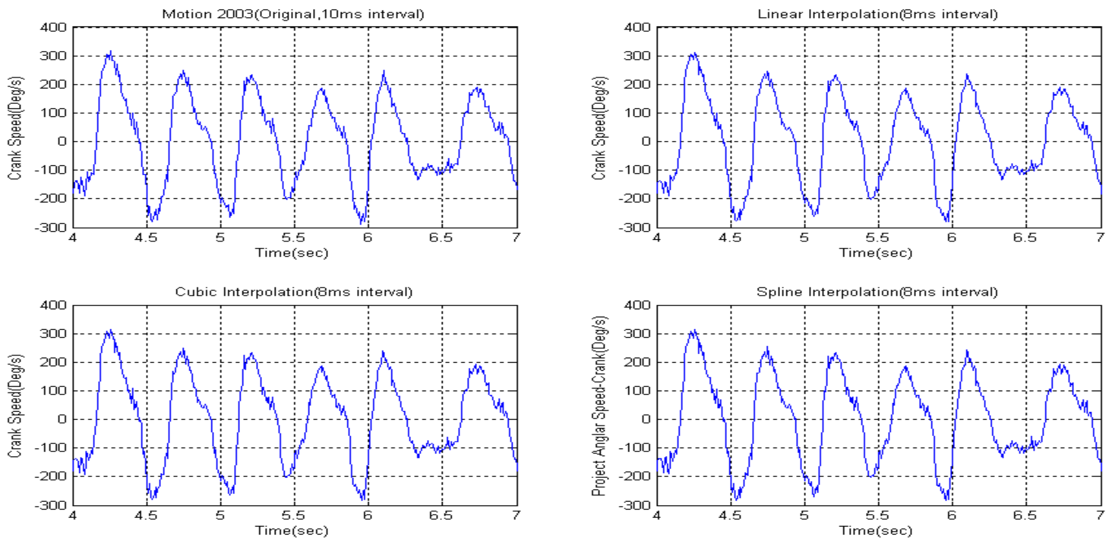


Figure 4.7 Crank F Angular Speed Profiles from 4th second to 7th second

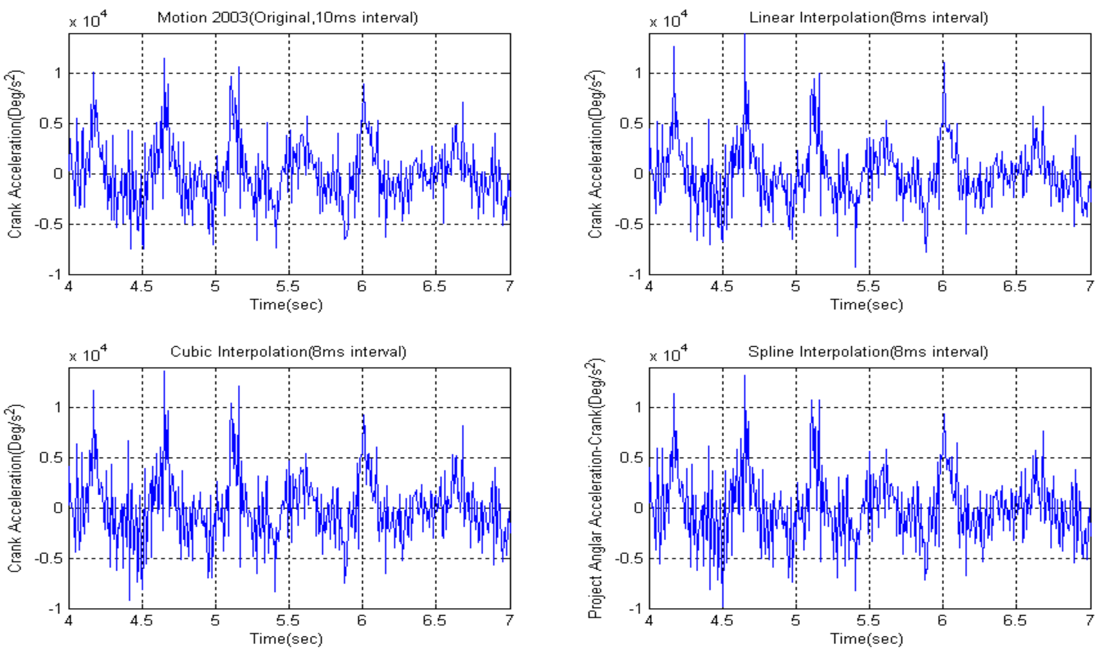


Figure 4.8 Crank F Acceleration Angular Profiles from 4th second to 7th second

Speed profile (Figure 4.7) and acceleration profile (Figure 4.8) are other two important factors in control. Mathematically, speed is the first derivative of position displacement and one more derivate of speed gives acceleration. In Figure 4.7, speed displacement from every interpolated trajectory has similar shape as the one from original Motion 2003. There are same jumps and breaks in four sub-figures. Unfortunately, neither interpolant has significantly smoothed speed profile because the time interval change is not very big and the original trajectory already contains some abrupt speed change.

We can see every acceleration profile (Figure 4.8) has totally lost smoothness even in the original Motion 2003 and they have similar jumps in some area. According to Newton's law, acceleration determines the moving torque of each crank. We can assume the no-load (no-food) torque have the same shape as the acceleration profiles. Here, we can understand after interpolated, acceleration can still obey the actuator range (Table 4.1).

- Motion Profile in an occlusal phase

In order to well explain what happens in the occlusal area, trajectories of Crank F from 5.32s to 5.44s are illustrated in Figure 4.9. In terms of actuator property criteria (Table 4.1), the mechanical crank angular displacement accuracy is limited within 0.1^{o} in any chewing area. We can see the maximum MI error of Linear Interpolant compared to original trajectory is 0.0946^{o}; maximum MI error of Cubic Interpolant is 0.0515^{o}; maximum MI error of Cubic Interpolant is 0.0598^{o}. Errors can be contributed to the function and polynomials used by the interpolant.

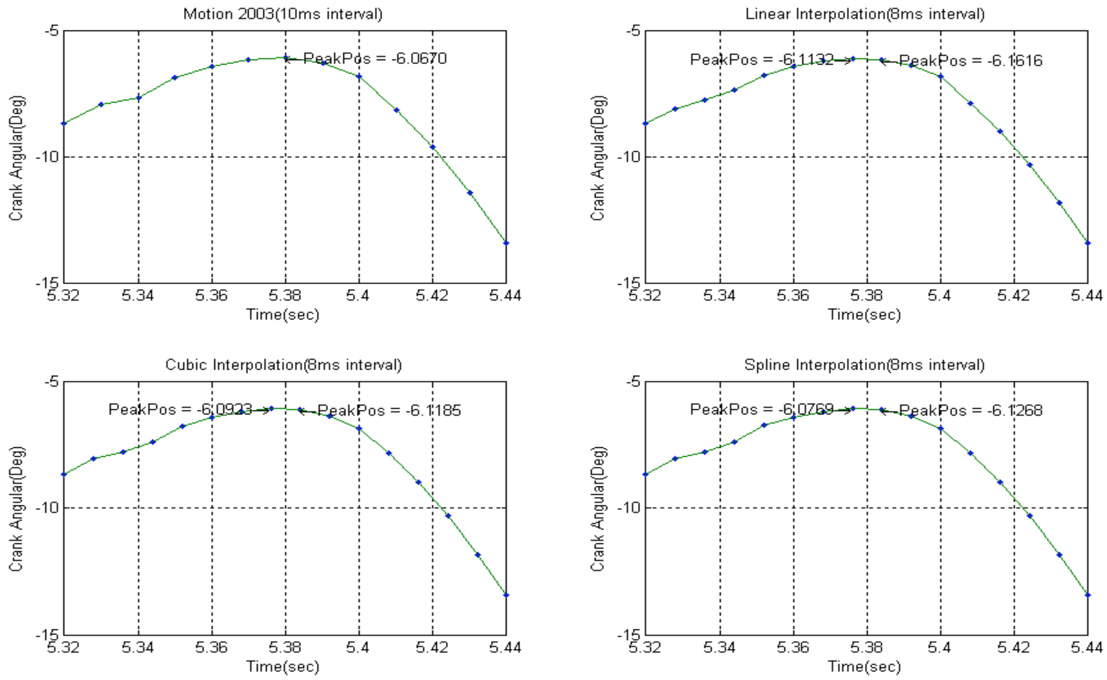


Figure 4.9 Mathematical Analysis of Crank Angular Profile in one Random Occluding Area (Crank F)

Inescapably, interpolation causes little position error; however, the error in MI point of each interpolant is limited in crank accuracy and the error size sequence is cubic<spline<linear so the interpolation works.

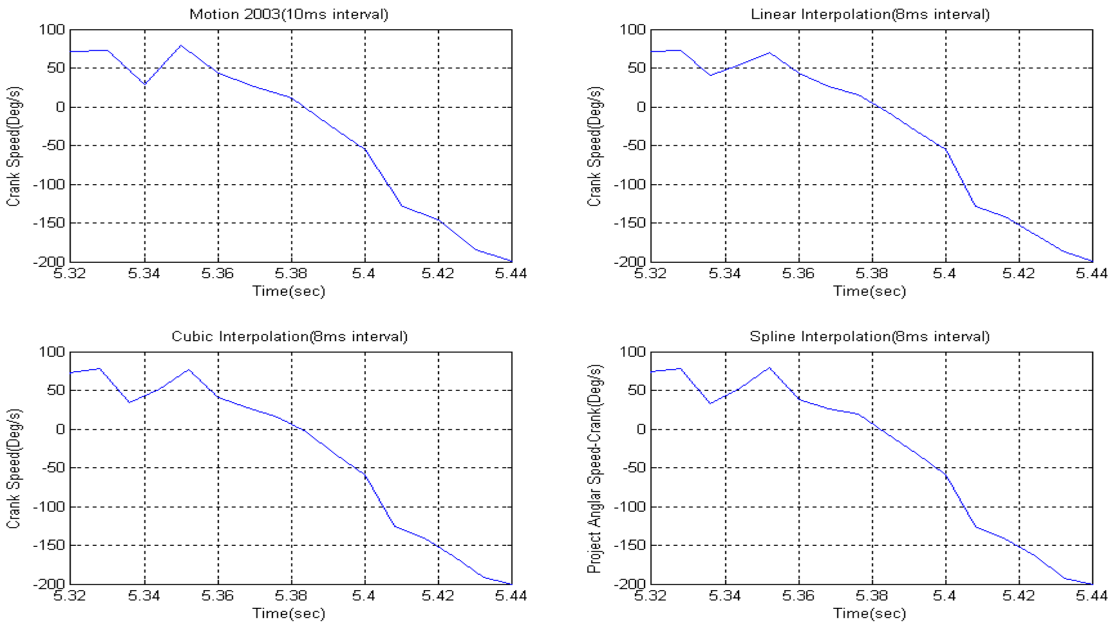


Figure 4.10 Mathematical Analysis of Crank F Speed Profile in a Random Occluding Area

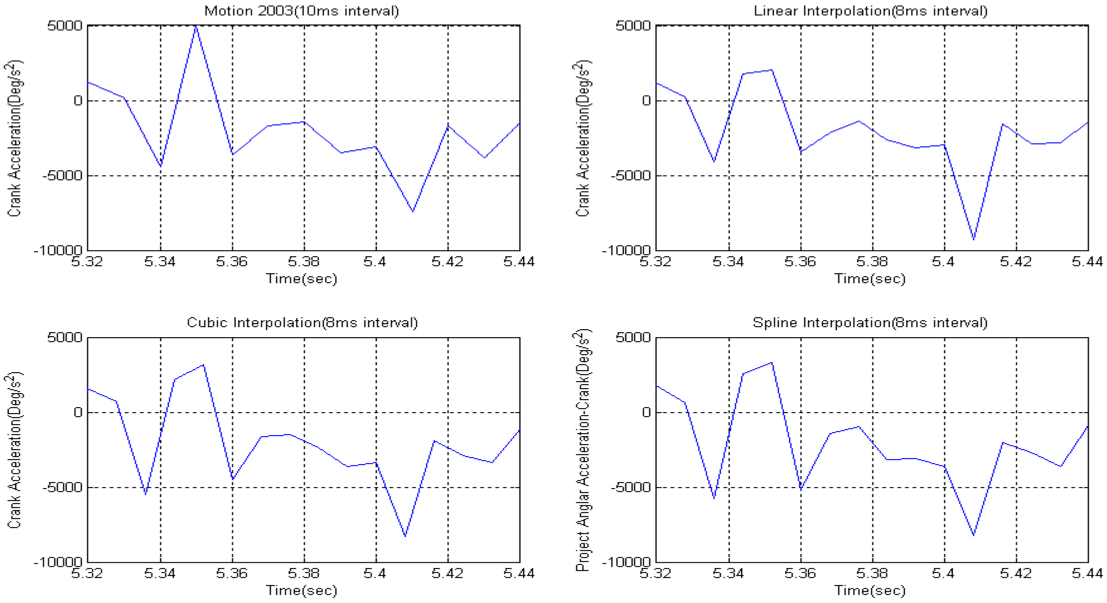


Figure 4.11 Mathematical Analysis of Crank F Acceleration Profile in one Random Occluding Area

A mathematic trajectory interpolation can smooth the motion profile and has little influence in velocity and acceleration. Errors in the occlusal area are well controlled in designed accuracy in Figure 4.10, Figure 4.11.

4.4.5 Optimal Option

The goal of interpolation is to make motion segment interval down from original 10ms to executable 8ms without shape change and give smooth motion profile if applicable under specified actuator accuracy. Comparing the three groups of interpolated crank trajectories to “Motion 2003”, it is found that none of them overshoots the peak and they increase, decrease on the same trajectory segment as in “Motion 2003”. As well as the trajectories, speed and acceleration were also considered and they are in desirable ranges too.

Linear interpolation is the easiest way and the controller uses this method inside as well, but, there are both discontinuous derivative (speed) happening in occluding area, and large position error at MI points. The pchip (cubic interpolation) is guaranteed to preserve shape, especially in the occlusal and maximum open area while profile is not smooth enough as spline which cannot give same trajectory shape. Considering the aim of the selection, cubic interpolated trajectory is finally selected to be tracked by the jaw cranks and named as “CI Motion 2003”.

Turn to the contour mode inside controller, as declared in section 4.2, it automatically linear interpolate any input position section to 1ms segment interval. Therefore, the smaller the given position segment, the well the nonlinear motion profile shape.

4.5 Conditions of No-load (No-food) Experiments

In the actual testing, two groups of crank trajectories were selected, which are from Motion 2003 and Cubic Interpolated (CI) Motion 2003. “Motion2003” in this experiment is the original incisor trajectory formed in Solidwork. Cubic interpolated crank trajectories were deduced from cranks trajectories of Motion 2003 in Matlab after mathematical calculations and stored in data files. The difference of two groups of trajectory is: Motion 2003 defines time interval as 10ms but will have to be implemented as 8ms by the controller instead. CI motion 2003 was designed to have the 8ms segment interval and can be exactly executed by controller synchronously. In other words, 5 seconds Motion 2003 trajectory finishes in 4 seconds because of the interval decrease.

For both crank trajectory groups, the motors were controlled by Galil 1880 under contour mode with KP=8, KD=290, and KI=0.2. Specified time interval of contour segment is 8ms, which is close to 10ms and not cause excessive segments. Motion was performed at auto-adaptive speeds and given 60% in maximum control torque (±6V control signal) over the first five-second Motion 2003 incisor trajectory. The limited torque is in order to prevent linkage broken by sudden force change. No extra load or food was applied to the bottom plate of the robotic jaw. Table 4.2 specifies list of detailed experimental conditions.

For each run, commanded crank angular displacement, actual crank angular displacement, crank angular speed, crank angular acceleration, and actual applied torque at each motor (control signal) feed-backed to the controller RAM had been recorded into six text documents and sent to Matlab in order to view and analysis effects of cubic interpolant on crank trajectories.

Table 4.2 Experimental Conditions

Parameter	Value
PID	KP=8, KD=290, KI=0.2 applied to every motor
Motor Speed	Arbitrary velocity profile, unlimited maximum velocity
Motor Acceleration	Arbitrary acceleration profile, unlimited maximum acceleration
Motor Torque	Within 60% maximum control torque (±6V control signal)
Path length	First 5 seconds of incisor trajectory based on 10ms interval (seven

	chewing cycles): 500 position points for the Motion 2003(500*0.008=4 seconds) 625 position points for the CI Motion 2003(625*0.008=5 seconds)
Crank Load	No load, bottom teeth plate is installed on the jaw
Food Force	No food applied
Control Mode	Contour mode, position increments described with command from trajectory data files
Segment Interval	DT=3; interval=2^3=8ms
Trajectories	Two groups of crank trajectories(X,Y,Z,W,E,F) from Motion 2003 and CI Motion 2003
Data Collect Interval	8ms each position collected

4.6 Experimental Results

Under the previous specified conditions, experimental motion profile results of six cranks for both trajectory groups are displayed in Appendix 1. Each crank motion profile figure contains information of actual trajectory, commanded trajectory, and error between actual and commanded trajectory, applied torque, crank angular speed, and crank angular acceleration.

Still assuming during the first 5-second period (Motion 2003; 10ms interval), motion profile of crank A is representative of all actuators in entire motion time. Regardless of the running time, tests of both trajectories were completed in seven chewing cycles.

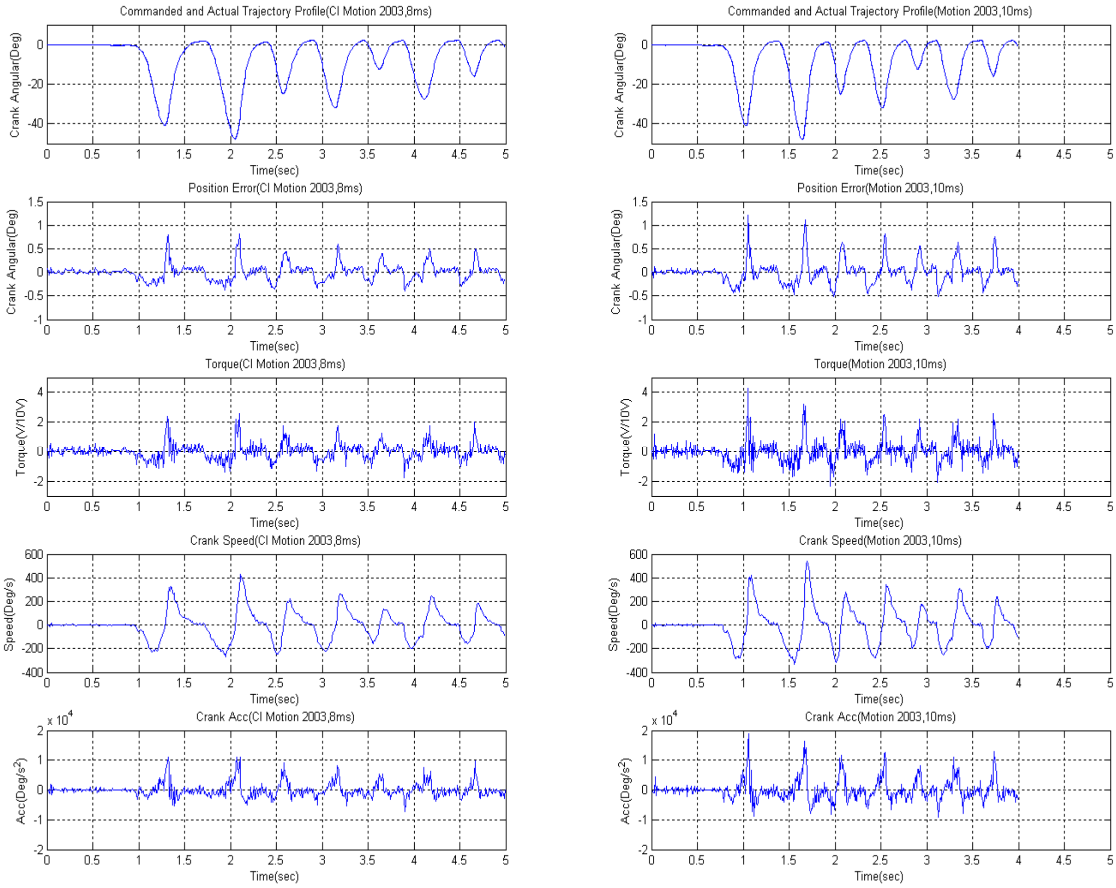


Figure 4.12 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, Torque, Crank Speed, Crank Acceleration Results Running on CI Motion 2003 (left) and Motion 2003 (right) during First Seven Chewing Cycles on Crank X

4.6.1 Crank Trajectory Profiles

For the commanded and actual trajectory profiles in Figure 4.12, the only difference between Motion 2003 and CI Motion 2003 is the motion finishing time due to the different trajectory segment time intervals. Actually, motion was compressed to 80% in the right sub-figures. Therefore, we can define the CI Motion trajectory is the trajectory on standard segment rate, while the original Motion 2003 is the speedup trajectory with the rate of 1.25 times.

There is no obvious (position error>1^{o}) peak-cut or backlash during occlusal or maximum opening area in the two trajectory sub-figures. This means we can accelerate motion up to 25% based on standard trajectory without losing control when no food has been applied to the bottom jaw.

On the other hand, cubic interpolated crank F trajectory from Motion 2003 can be exactly executed by the controller as it is anglicised in the mathematical part.

4.6.2 Motion Control Position Accuracy

Crank position accuracy in mechanical aspect of the robotic jaw is defined as 0.1 crank degree in Table 4.1 where the errors normally happen in transmission parts between motor shaft and crank end. On the motion control hand, errors appear between the Galil controllers to motors, in other words, is in the control loop, where motor shaft encoder counts feedback to controller and are compared with the counted position. Motion control errors mainly caused by: 1. PID value not well tuned and overshoot occurs in some control segments; 2. A sudden change in speed orientation or acceleration orientation and incur motor shaft jerking; 3. A big position change in an interval and motor not able to reach commanded position under specified torque limit; 4. exceed mechanical force like hard food.

The two sorts of errors can be positive or negative, respectively. The combinational effect of mechanical and control error can be measured by adding a linear encoder on each jaw linkage. In a manner, when the orientations of the two errors are opposite and they have same error value, the final impact error then may be zero.

Fig 4.13 displays filtered position points on Crank A. At these position times, the motion control position error is more than 0.13 degree and 0.14 degree for both trajectories, respectively. Percentage of the position points with exceeded error is also showed at the corner of each sub-figure.

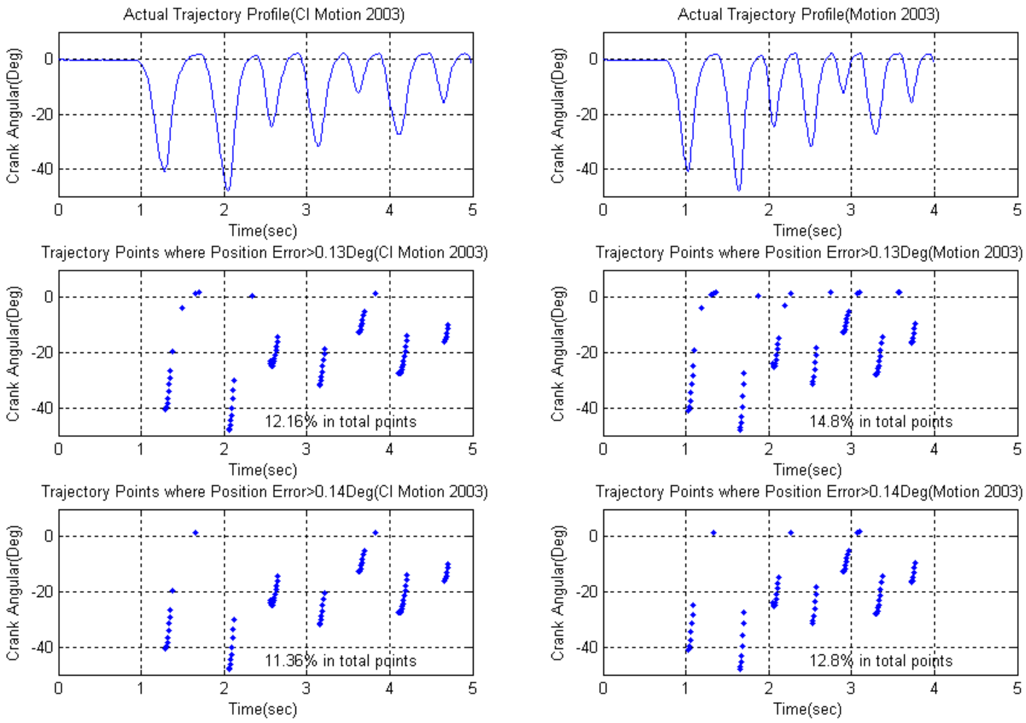


Fig 4.13 Filtered Trajectories with Errors under Specified Position Accuracy (Crank X)

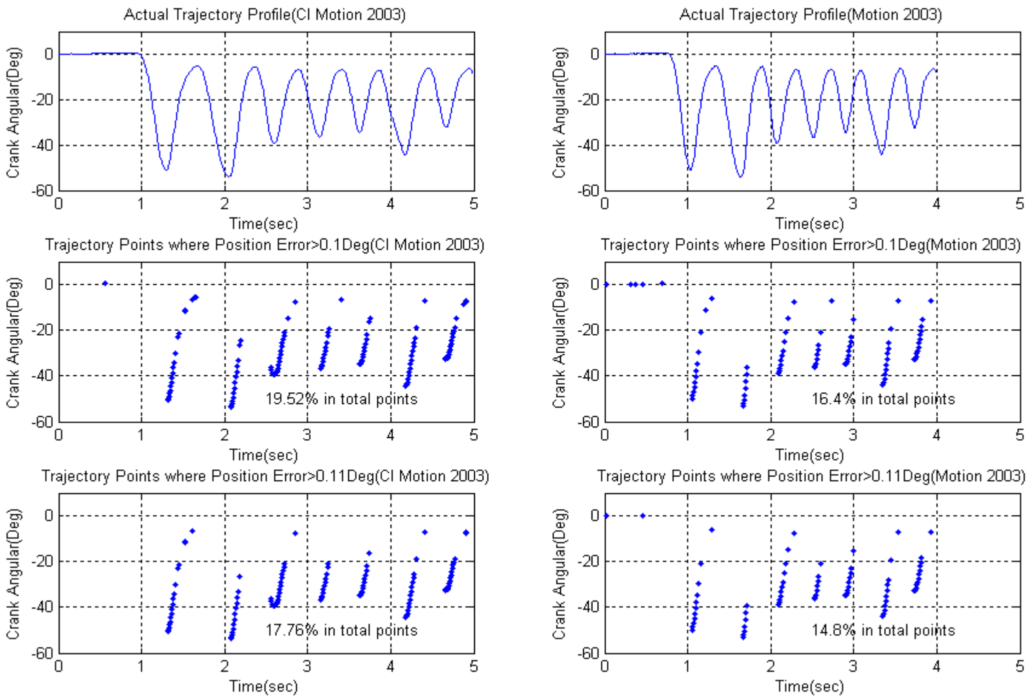


Fig 4.14 Filtered Trajectories with Errors under Specified Position Accuracy (Crank Y)

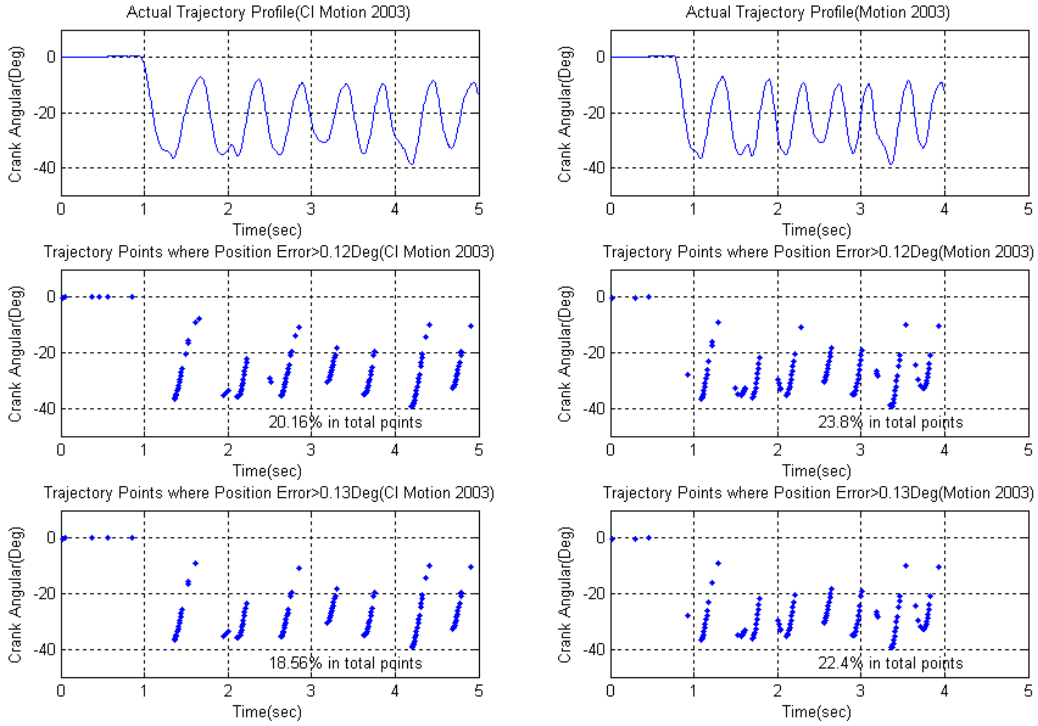


Fig 4.15 Filtered Trajectories with Errors under Specified Position Accuracy (Crank Z)

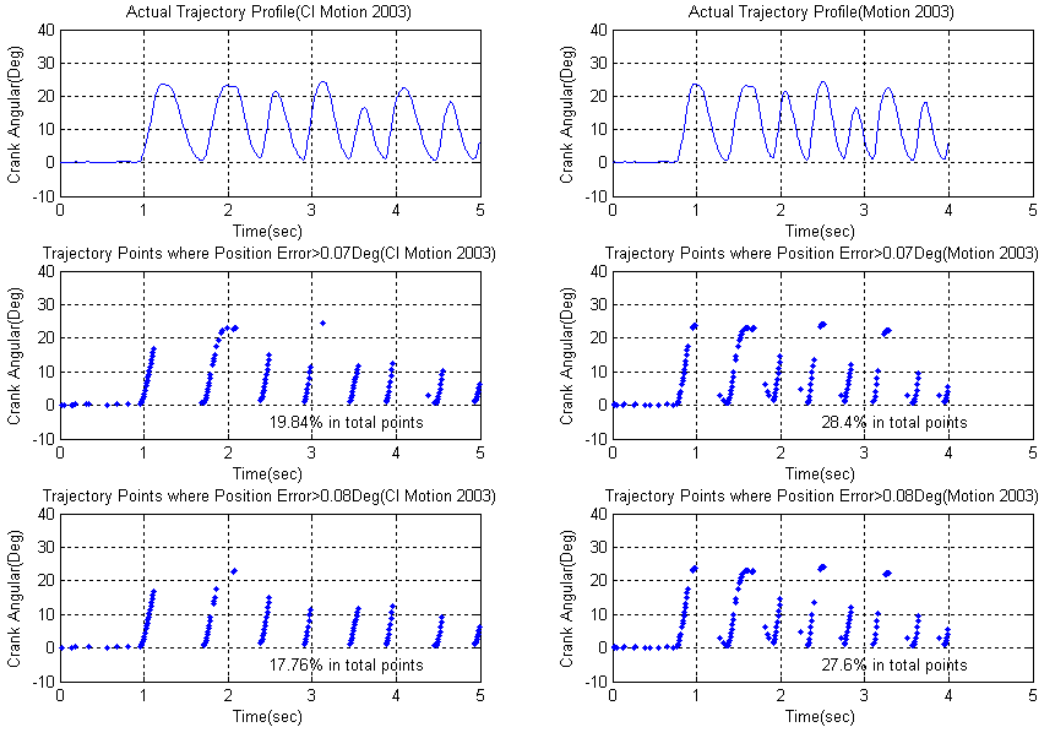


Fig 4.16 Filtered Trajectories with Errors under Specified Position Accuracy (Crank W)

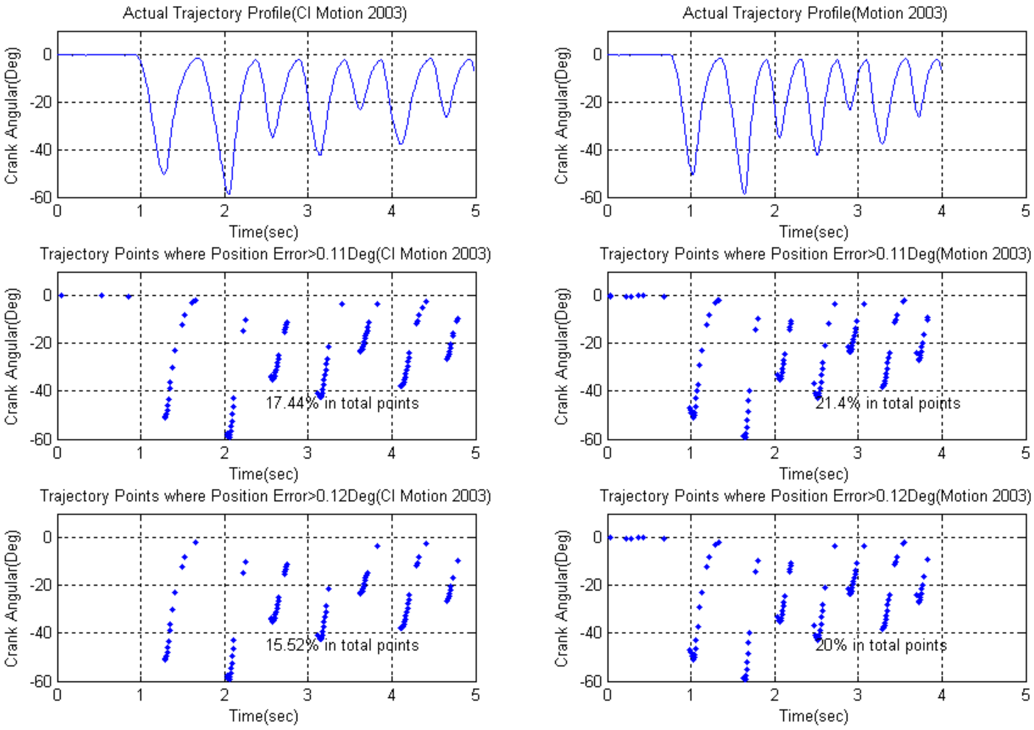


Fig 4.17 Filtered Trajectories with Errors under Specified Position Accuracy (Crank E)

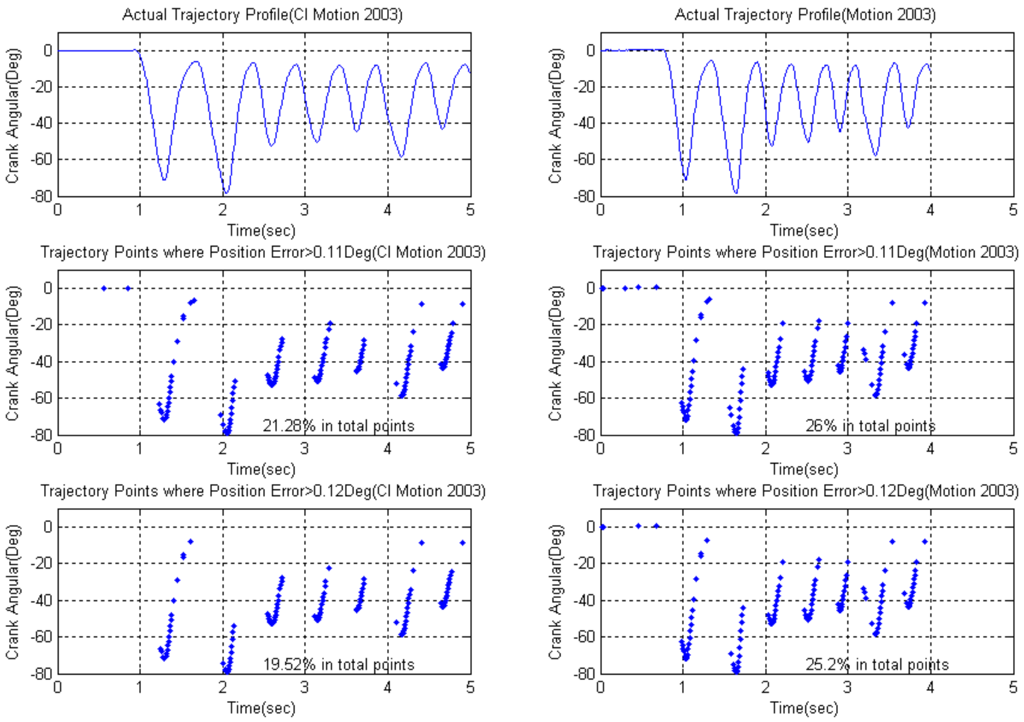


Fig 4.18 Filtered Trajectories with Errors under Specified Position Accuracy (Crank F)

Definitely, the quicker the jaw runs, the more inaccurate points we got under same specified motion control position accuracy. The relationship of the position error and torque will be talked next. Here, suitable motion control position accuracy will be selected first.

The top two actual trajectory profiles in Fig 4.13 were picked up from Figure 4.12 as a compare to the filtered trajectories. Basically, in Fig 4.13, trajectory positions with error fill the starting path of jaw closing area and the starting area of occlusal area. We knew the most important part in a chewing cycle is the occlusal area, so the inaccurate trajectory in this area should be eliminated or as less as possible. Now, only consider the normal chewing behaviour (CI Motion 2003) from crank A. Clearly, when the accuracy allowance is set to 0.13^{o}, there are two position points in the first occlusal area; however, given 0.01^{o} more allowance, only up to one trajectory points has got position error out of range. This excessive error is because the plate's up and down teeth touched. Therefore, motion control position accuracy of crank A can be defined as 0.14 crank degrees, and the total accuracy of each crank is 0.1+0.14=0.25 crank degree.

Under the same criteria: no more than one position error exceeds accuracy allowance in occlusal area. From the results shown in Fig.4.13 to 4.18, we get:

Table 4.3 Motion Control Position Accuracy of Cranks

	PID	Accuracy (crank degree)	Total Accuracy (crank degree)	Error Occurs Percentage
Crank X	KP=8, KD=290, KI=0.2	0.14	0.25	11.36%
Crank Y	KP=8, KD=290, KI=0.2	0.11	0.21	17.76%
Crank Z	KP=8, KD=290, KI=0.2	0.13	0.23	18.56%
Crank W	KP=8, KD=290, KI=0.2	0.08	0.18	17.76%
Crank E	KP=8, KD=290, KI=0.2	0.12	0.22	15.52%
Crank F	KP=8, KD=290, KI=0.2	0.12	0.22	19.52%

It is also essential to know what happens after a motion control position error appeared. From position error in Figure 4.12, we can see the error can be nearly 1^{o} in some contour segment; however, the commanded trajectory and actual trajectory is closely matched. As a conclusion, the controller adds compensation to the next contour segment and tries to minimize the total trajectory error. This is the reason why the trajectory with position error can be ignored during jaw closing area in Fig 4.13.

4.6.3 Torque and Torque Limits

A run torque at each actuator sent to the amplifier is represented by a control signal ranging from -10V to +10V. Under free food contour mode, without limiting the maximum torque and given a big step contour segment (large position change), controller will generate a big control signal (large torque) to the amplifier in order to push motor speed up while causes both motor speed and acceleration jump. The absolute value of the dynamic torque has the positive rate compared to the statistic torque of a motor, which no load is applied and holding motor shaft under a same tuned PID value.

Disasters, like linkage broken, gear teeth cracked, happen when the torque exceeds jaw mechanical compliance. When hard food is placed between the teeth plates, controller is willing to apply more torque to motors in order to try to successfully achieve the position segment without motion control position error and break the “food”, although this work is not possible. In this case, after touching the “hard food”, controller torque quickly increases to |10V| or the mechanical linkage breaks, whichever happens first.

For example, during the maximum opening area with trend to close in Figure 4.12, due to the inertia of the bottom jaw plate, a large positive acceleration was required and they caused a significant torque increase to nearly 2.5V in CI Motion 2003. The torque peak appears worse to be more than 4 in the right motion because a quicker acceleration (1.6 times bigger) was required. This also proves the assumption of mathematical analysis of the original trajectory property in Table 4.1 is correct.

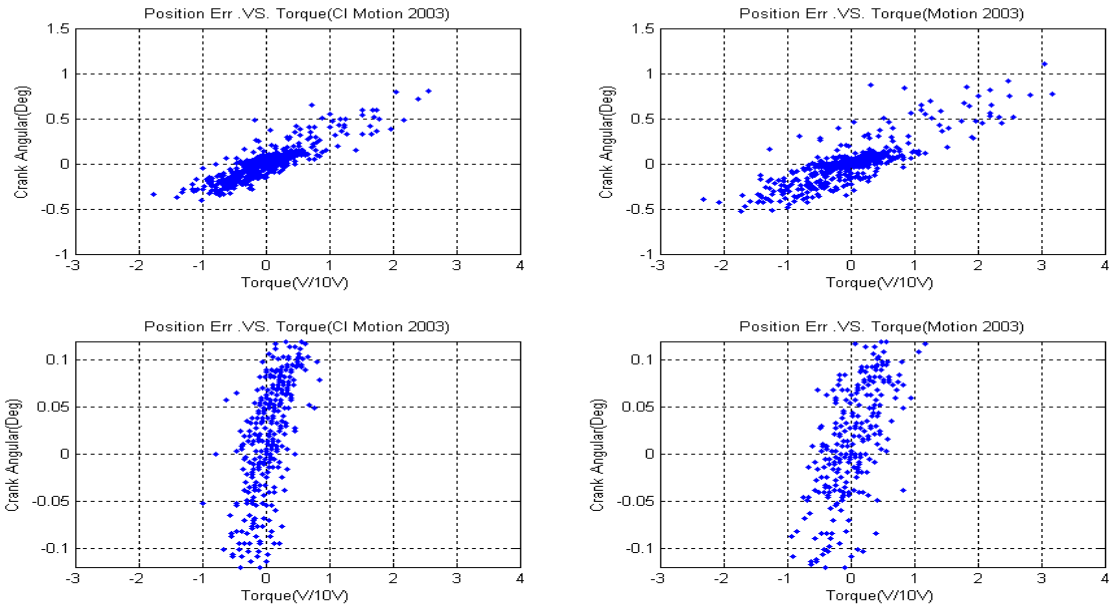


Figure 4.19 Position Errors and Torque (Crank A)

Another issue to the torque is motion accuracy including position, speed, and acceleration. Top two sub-figures of Figure 4.19 are the point-clouds figure of the value of position error and torque at different time during the experimental period. As described in 4.5.2, a direct ratio links the position error and torque when no food applied. Motion control position accuracy was defined as $|1.4^{o}|$ for Crank A, so the last two sub-figures of Figure 4.19 are the torque of motion trajectory which has the allowed errors. Only consider the normal trajectory behaviour (CI Motion 2003) at left, we get without food, when the motion control position accuracy is controlled within 0.14 crank degree, the torque applied to motor is less or equal to 1V/10V. $\text{Torque}_{\text{accuracy} \le 0.14} \le |1|$V. As a definition, when motor is applied with less than 1V torque, it is always running under specified motion control position accuracy.

Return to the top two sub-figures in Figure 4.19, we see before torque=-1 and after torque=1, position error is $>0.14^{o}$. Then, we must make sure torque>|1| only happens out of the occlusal area, except the position point where teeth touch.

4.6.4 Crank Speed and Acceleration

According to Table 4.1, an actuator only can perform linear speed change within its mechanical speed and acceleration tolerance (maximum speed) which are ±458.6°/s and ±11465 °/s², respectively. When motors run on Motion 2003 trajectory, according to Appendix 1, acceleration often exceeds tolerance after maximum opening area and speed has the same problem as well. So they both tell the quicker trajectory is not suitable for food-loaded test in the future.

Turning to the speed and acceleration profiles of CI Motion 2003, the displacements are well shaped as they were calculated in the mathematical analysis (Figure 4.7, Figure 4.8). As a conclusion, the maximum controller speed and acceleration output can be much bigger than what the mechanical part can handle.

One of the operational solutions to control the limit of acceleration is to change the control mode from contour mode to speed mode. The speed mode is a basic control mode provided by Galil. Operators can force the controller to fix the maximum acceleration and speed but has to estimate the time of the "speed freezing" period in order to calculate how far the end-point has moved. Figure 4.20 illustrates this possible solution.

		Trigger
Acceleration	0°<<<<<<<<<<<<<<<<<<<<<<<<<<<<<	>>>>>>>>>>>11465°/s²
	Contour Mode	Speed Mode
	Speed adjusted by controller	Speed set by user
	Acceleration adjusted by controller	Acceleration set by user
	More torque required for motor moving, torque must be limited in order to control the motor acceleration	Torque can be adjust first and manually fix trajectory error next

Figure 4.20 Optimal operation options

According to the exam results in Appendix 1, during some segments period, where requires some huge speed changes, there is no sufficient torque for both accelerating and handling external force. In this manner, these points would be dead angle for monitoring the force feedback, although linkages are well protected. This can be done by setting up a conditional code in programming although it will double the programming load.

The advantages of two-mode control are:

- Less torque is required for free-food chewing so more spare torque can be arranged for the external force.
- It does not require hardware work to add force sensors

The disadvantages of two-mode control are:

- The position length of next contour segment after the speed mode will be automatically compensated by the controller. There will be bigger acceleration for this point than original.
- Programming is complex and the code length is limited by Galil software, which is only 8000 lines including uploading trajectory data.

4.7 Conclusions

A new group of interpolated crank trajectory had been generated under specified condition and is proved executable by the Galil controller. Most hazardous errors are eliminated and a stable control torque can be set to prevent linkage broken. Motion control accuracies of each crank are defined. Also, the no-food test tells, in actual control, traditional PID is applicable and the occlusal area is well controlled.

However, the error occur percentage looks very big for every crank. Although the number can be decreased by increase accuracy allowance, the total accuracy will be too big and worse the motion especially at the MI point. Actually, error happen percentage can be changed by either adjust PID value or give addition control loop as compensation. The advantage of adjust PID is the convenience, but it will cause control torque change and a new torque limit has to be found. New control loops like fuzzy logic or fuzzy system for auto-adaptive PID gain are selectable to be attached. Nevertheless, the total accuracy including mechanical backlash can only cut by installing a linear encoder to measure the jaw linear displacement.

Control torque, speed, acceleration all displayed the power of the motor for each crank is enough to push the jaw move understand chewing behaviours. Maximum value of the three factors can be increased several times if we upgrade linkage material.

Chapter 5 Force Compliance Control and Experiments

5.1 Introduction

In the previous free chewing tests, position accuracy for each individual crank was deduced and the robotic jaw can follow trajectory to perform chewing under designed speed and acceleration (Table 4.1); however, in order to know the torque tolerance of linkage and the maximum torque needed for No-food chewing, the tests were undertaken without crank torque limit.

The force testing is to assess feasibility of passive force control by limiting individual torque on the actuators of the robotic jaw. Torque required by each actuator will increase when external force is added. In the other words, jaw used bigger force to chew when food is put into mouth. The output force at the bottom plate can be passively controlled when the maximum torque supplied to each actuator is limited. In case of exceeding the selected maximum torque on an actuator, the motor should stop and hold until the overload force is released. During the holding time, position error of this actuator occurs but other actuators can still run unless they reach their own maximum force.

The property of the food, such as hardness, shape, as well as stickiness can be calculated from the feedback of position error and the difference of crank torque supplied.

5.2 Experiment Analysis without Force Compliance Control

At the first stage of the experiments, motors were supplied without any force limit and powered by maximum current when required in order to monitor the behaviours of the trajectory trace. According to the motor property, each 12V motor has assigned 60W power rating and they can be briefly overloaded in a short term, therefore, this experiment is designed to see how the motors behave to follow their trajectories and increase the accuracy. In addition, the practical mechanical force tolerance of each linkage can be tested and recorded.

The six motors were powered by laboratory power supply with the maximum 6A continuous current. The power supply is protected by internal overload circuit and will automatically recover while the overload released so that the overall current to six motors can only be less than 6A. The normal 50A power supply was not used for the exam because the motors may spin madly if given incorrect commands and break the entire linkage system. Then, we can know the actual maximum power to the motors was 1/6 of the maximum motor power.

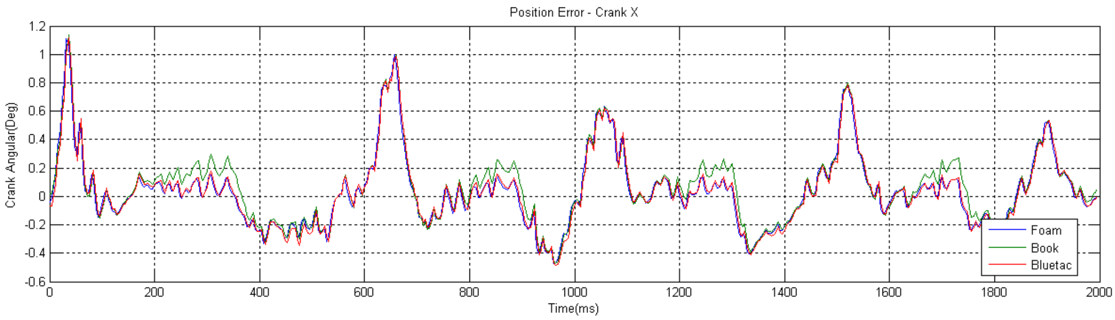
Table 5.1 Experimental Conditions

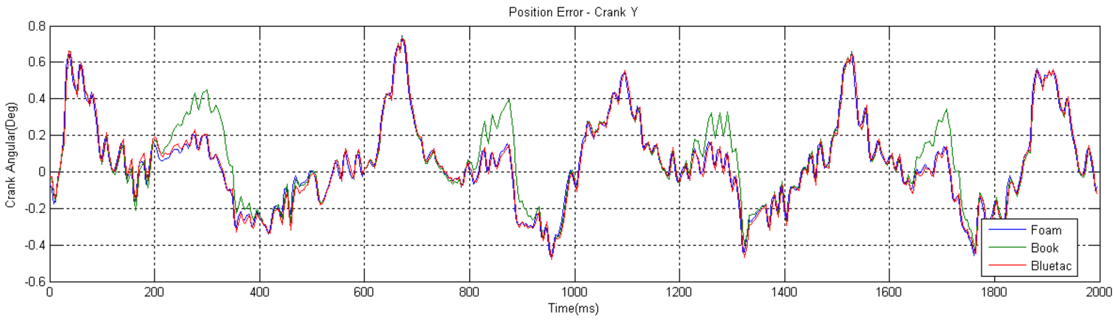
Parameter	Value
PID	KP=8, KD=290, KI=0.2 applied to each motor
Motor Speed	Arbitrary velocity profile, unlimited maximum velocity
Motor Acceleration	Arbitrary acceleration profile, unlimited maximum acceleration
Motor Torque Limitation	Six motors share 6A which is 1/6 of the max continuous current of these motor need
Path length	First 7.5 seconds of incisor trajectory (Seven chewing cycles): A pause is inserted between 1.24s and 3.74s for placing simulated food.
Crank Load	Top and bottom teeth plate are installed on the jaw.
Food Force	No food 10mm aluminium plate 10mm soft foam 10mm sticky gum
Control Mode	Contour mode, position increments described with command from trajectory data files

Segment Interval	DT=3; interval=2^3=8ms
Trajectories	Crank trajectories(X,Y,Z,W,E,F) from CI Motion 2003
Data Collect Interval	One position point per 8ms collected

Without setting the force limit value, a 2B book, a 5cm thick foam, and 1cm thick Bluetac® have been tested to evaluate the performance of the six actuators during first 2 second chewing area. Graphs of text result are showing in Appendix 2.

Here, the position error among the three testing material has been shown in Figure 5.1.





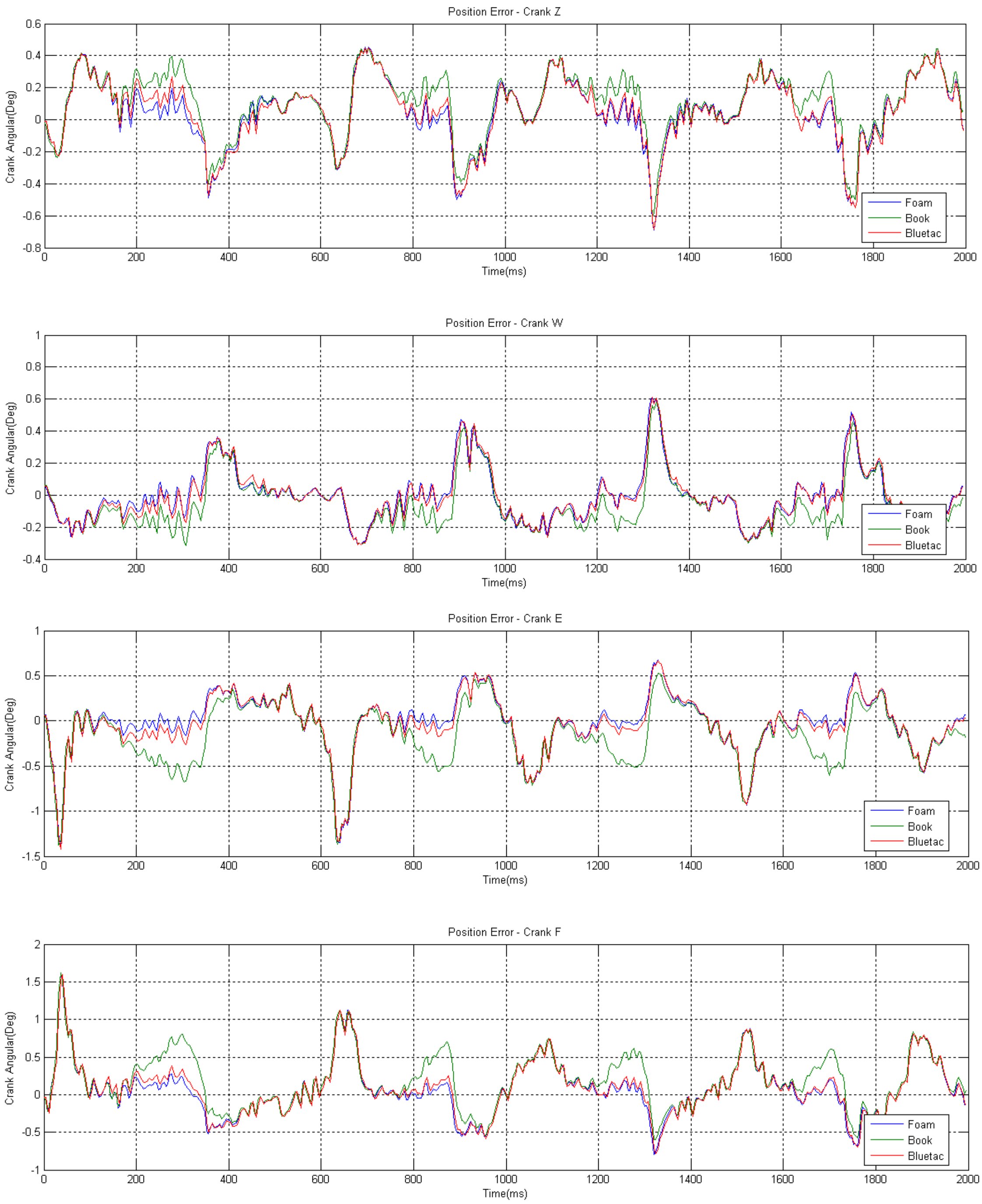
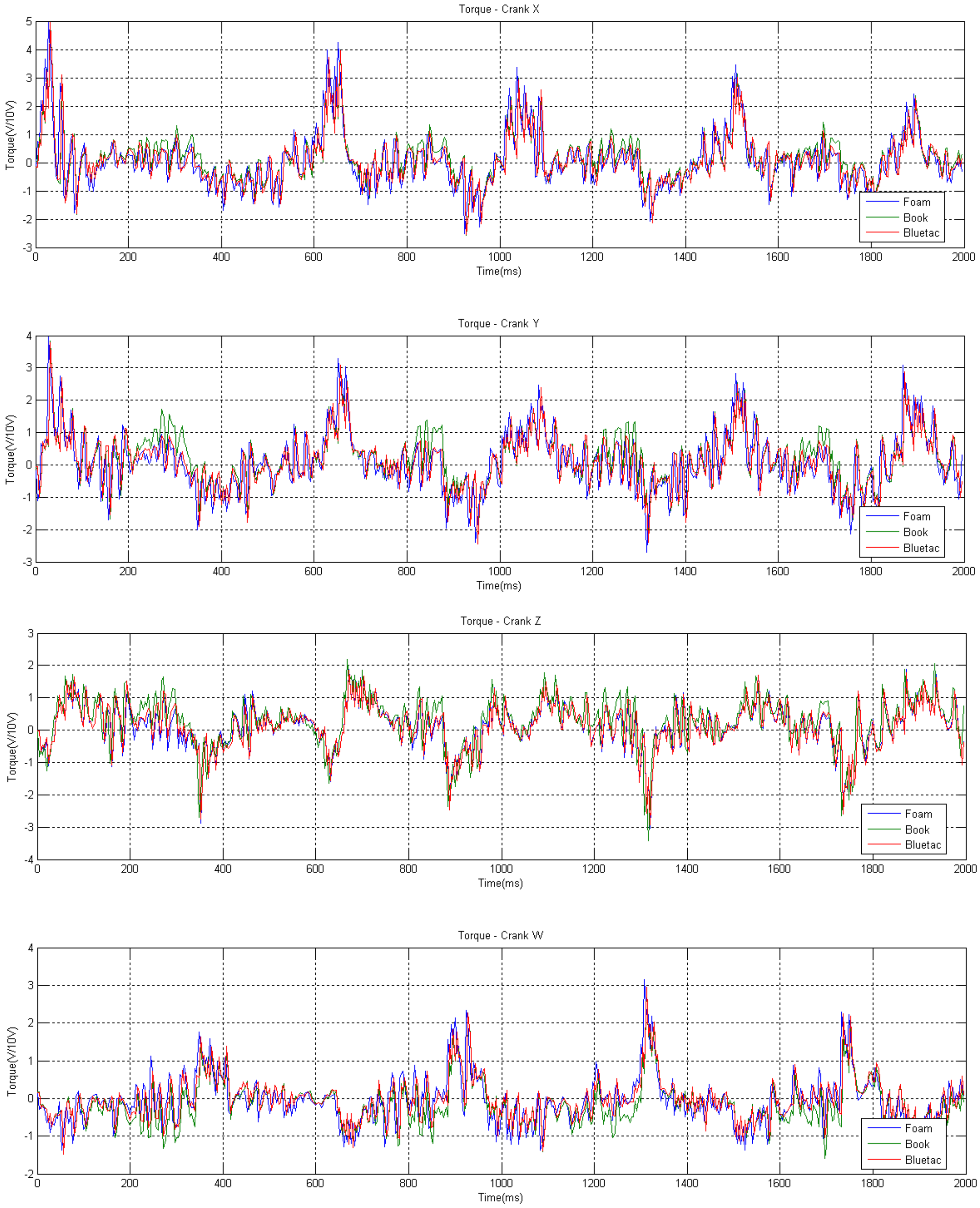


Figure 5.1 Position error values compare (2B book, 5cm thick foam, and 1cm thick Bluetac) of 6 cranks

Obviously, for the position error of each crank, 2B book has the biggest value and Bluetac has got the second while foam last. Compare to the crank trajectory. The difference of error occurred during the occlusal area and eased after teeth left the material.

The torque value among the three testing material have been shown inFigure 5.4.



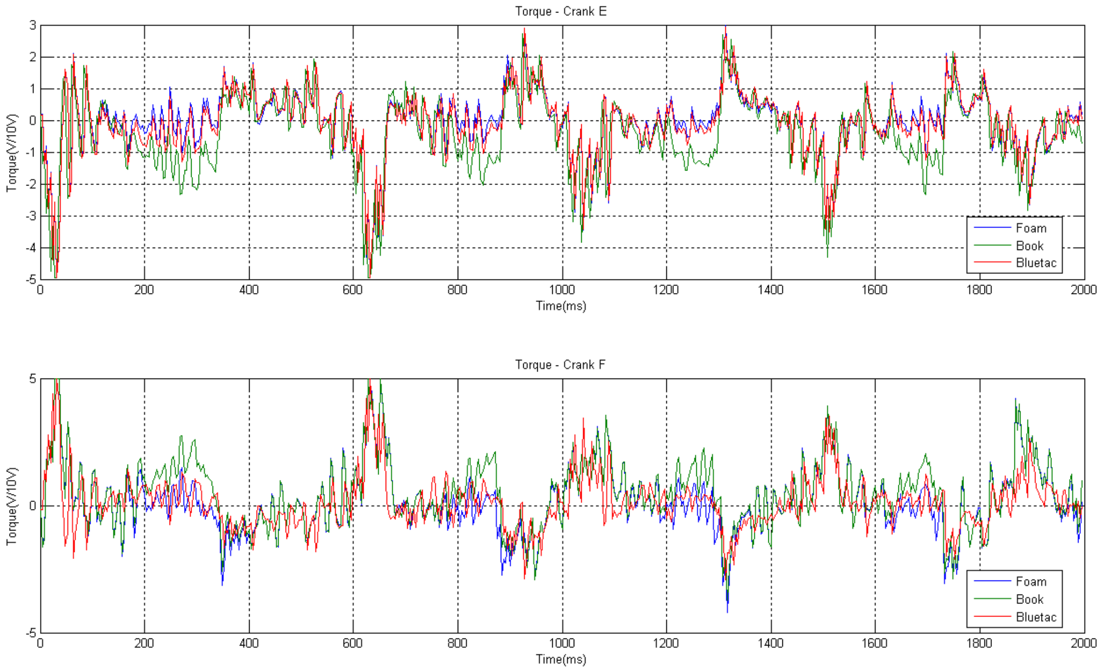


Figure 5.2 Torque values compare (2B book, 5cm thick foam, and 1cm thick Bluetac) of 6 cranks

The torque graphs displays same symptom as the position error graphs shown. Without setting a force limit for each actuator, the jaw could try it best to break the food without considering the structure of it. During the test, several coupler shafts（Figure 5.3）between actuator and end joint were broken.

Figure 5.3 Physical coupler, which includes one thread shaft and two rod ends

Thus, it is necessary to set a reasonable limit to minimize the damage to the hardware. On the nature way, the setting of the limit is also correct because no one would like to use his entire force to break nuts by teeth. The teeth broken are equal to the coupler broken in the previous experiment.

5.3 Experiment Analysis with Passive Force Control

5.3.1 Experimental Conditions

With the teeth removed, in order to fix the simulated food while chewing properly, the top plate of the robotic jaw was replaced by a flat thick hard plate. The bottom surface of the hard plate just touches the top part of bottom teeth when it closes. A 10mm aluminium plate, simulating hard food and a 10mm foam, simulating soft food were then inserted between the hard plate and bottom teeth during a 2.5 jaw open period. Both of the simulated foods were fixed under the top plate during the chewing.

The Cubic Interpolated (CI) Motion 2003 chewing trajectory was run in the experiments. The trajectories included seven chewing cycles over the first 5 seconds. A 2.5 seconds pause phase, was inserted into each crank trajectory after the jaw first opened, for the convenience of inserting the simulated food. As the trajectories have been interpolated, the running speeds, accuracy of control, and other properties, are exactly same as they were specified in the No-Food experiment. However, in order to see the effect of passive force control, individual crank torque limits (Table 5.2) were set to each actuator. The value of torque limits are the reasonable torque required for motor running in acceptable accuracy in the CI Motion 2003 experiment without food.

Table 5.2 Experimental Conditions

Parameter	Value
PID	KP=8, KD=290, KI=0.2 applied to each motor
Motor Speed	Arbitrary velocity profile, unlimited maximum velocity
Motor Acceleration	Arbitrary acceleration profile, unlimited maximum acceleration
Motor Torque Limitation	Crank X=±2.5V/10V Crank Y=±2.0V/10V Crank Z=±1.0V/10V

	Crank W=±1.0V/10V Crank E=±3.0V/10V Crank F=±3.0V/10V
Path length	First 7.5 seconds of incisor trajectory (Seven chewing cycles): A pause is inserted between 1.24s and 3.74s for placing simulated food.
Crank Load	Bottom teeth plate is installed on the jaw. Top plate is replaced by a same high hard metal plate.
Food Force	No food 10mm aluminium plate 10mm soft foam
Control Mode	Contour mode, position increments described with command from trajectory data files
Segment Interval	DT=3; interval=2^3=8ms
Trajectories	Crank trajectories(X,Y,Z,W,E,F) from CI Motion 2003
Data Collect Interval	One position point per 8ms collected

5.3.2 Food Simulated by Aluminium Plate

Under the specified conditions in Table 5.2, a 10mm aluminium plate was inserted into the right side of the robotic jaw in the first jaw open cycle. Motion profile results including actual trajectory, position error, and torque, are compared with those of No-Food tests displays in Figure 5.4, Figure 5.5, and Figure 5.6.

The behaviour of actuator can be affected when a food of any shape or material is placed into the mouth. According to the property (hard material and slippery surface) and placed position (right side) of the aluminium plate, the trajectory profiles of actuators are divided into three pairs. Crank X and crank Y are mainly responsible for the horizontal movement; Crank Z and crank E mainly look after left side upright action; and the other two were mainly taking action to the right side chewing where the aluminium plate was placed.

- Crank X and Crank Y

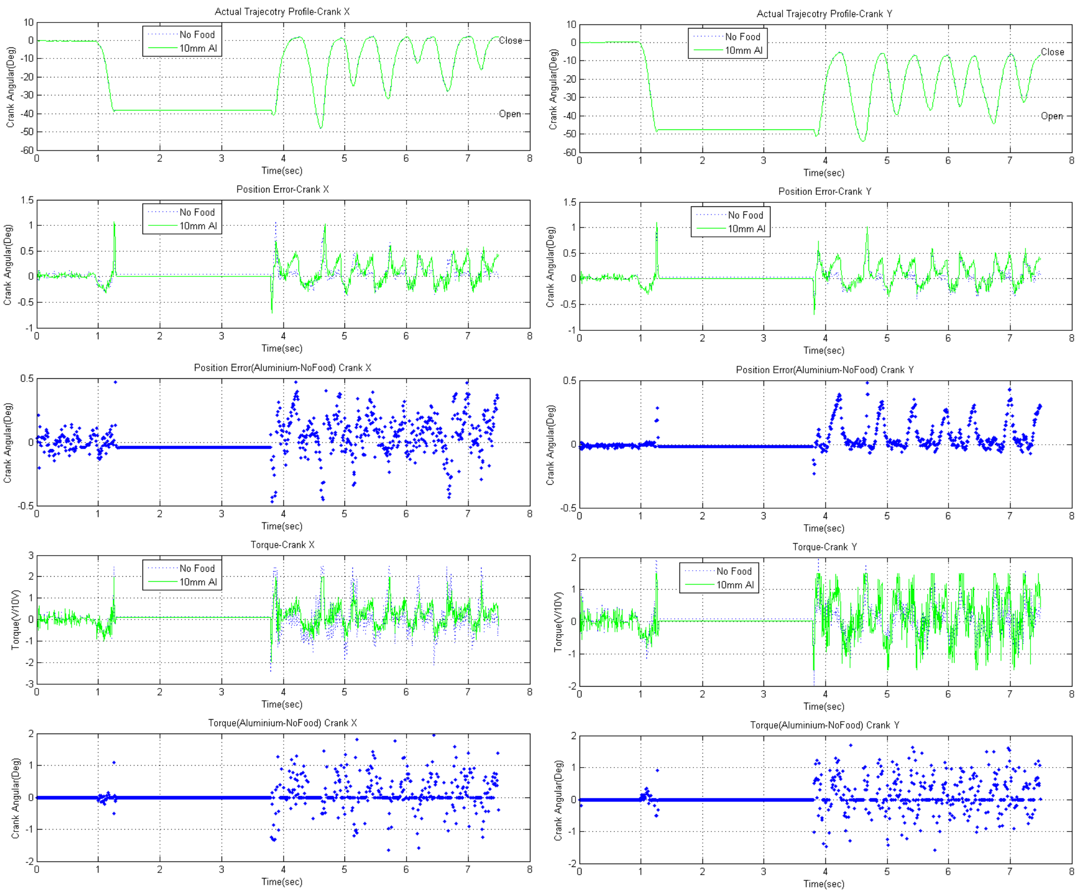


Figure 5.4 Robotic chewing results of hard food simulated by 10mm hard aluminium plate (CI Motion 2003)-Crank X and Crank Y

Although the aluminium plate was placed at the right side of the jaw, motion profile of both Crank X and Crank Y were affected; however, the change is very slight because the surface of the aluminium plate is very slippery and smooth. In the original design, Crank X and crank Y mainly do the X-Y plane action; therefore, in this experiment, only the dynamic and static friction between the plate and teeth can incur position error change and force increase during the chewing area.

Table 5.3 Properties of Crank X and Crank Y

	Crank X	Crank Y
Control Position Accuracy(Crank Degree)	0.14^o	0.11^o
Torque Limit	2.5V/10V	2.0V/10V

Since the surface of aluminium plate is neither sticky nor uneven, the overall actual trajectories of No-food and Aluminium are basically matched. In the actual trajectory profile of Figure 5.4, there is no visible change in either close or open area.

Refer to the position error graphs; we can see some slight error increase after the aluminium plate was inserted. The difference/increase of the position error between No-Food and Aluminium test are calculated by error of Aluminium minus error of No-food and is shown in the third two graphs in Figure 5.4. Based on the control accuracy of crank X and crank Y, which is 0.14^o and 0.11^o respectively, we can see the aluminium plate gave Crank X about 0.35^o incremental position error and Crank Y about 0.30^o incremental position error during the experimental period. Clearly, the tighter the jaw chewed, the larger the error occurred.

Torque graphs show the limit of the force was not constantly reached during the chewing period. This means the force effects from the acceleration of the jaw on the surface movement is under the torque limit controlled.

The torque difference is calculated by aluminium torque minus no-food torque when crank position error bigger than crank position accuracy and are shown in the last two graphs in Figure 5.4. Interestingly, the peak and bottom value of the torque difference is not obvious and all the torque difference points scatters between -2V/10V and +2V/10V.

- Crank Z and Crank E（Left）

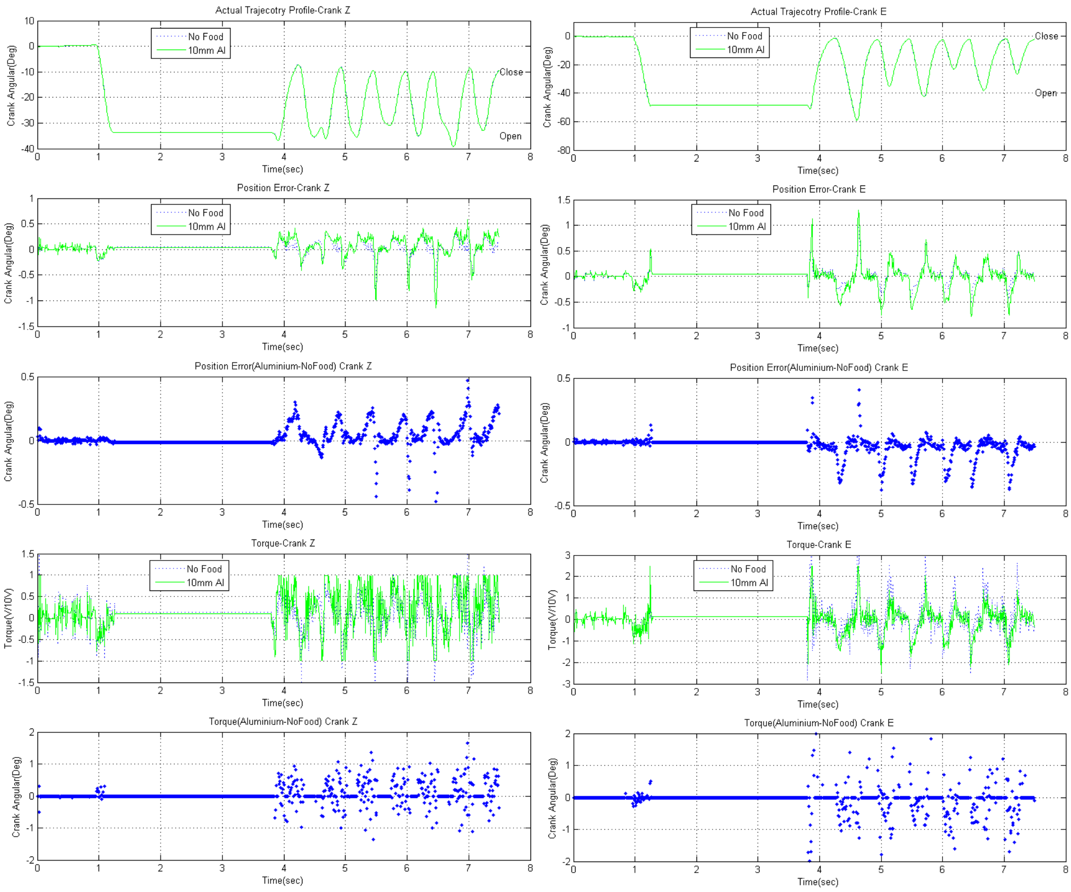


Figure 5.5 Robotic chewing results of hard food simulated by 10mm hard aluminium plate (CI Motion 2003)-Crank Z and Crank E (Left)

Table 5.4 Properties of Crank Z and Crank E (Left)

	Crank Z	Crank E
Control Position Accuracy(Crank Degree)	0.13^{o}	0.12^{o}
Torque Limit	1.0V/10V	3.0V/10V

Theoretically, the motion profile of Crank Z and Crank E in this experiment should be same as those in No-food test as no stuff was at the left side. Nevertheless, the simulated food is 10mm thick and the left actuators had to be affected when the right side cranks extremely tightly touched the aluminium plate. This is just like a “help” given to right cranks by their opposite mates. Definitely, the external force for left cranks is not as much as the right cranks so the actual trajectory profile in Figure 5.5 cannot show a remarkable difference.

The influenced results of position error are similar as Crank X and Crank Y. we can see the small “help” made about 0.2^o incremental position error (maximum position error-accuracy tolerance) for both cranks. The shape of position error difference of Crank Z and Crank E is same as the first two cranks. Error difference increased and decreased with similar gradient.

According to the torque limit in Table 5.4, we can see the torque for Crank Z often touched but did not keep at the torque limit line. Compared to the torque difference of Crank X and Crank Y, these points still scatter but limited within the occlusal range. The torque difference amplitude of Crank Z is ±1V/10V; Crank E is ±2V/10V.

● Crank W and Crank F (Right)

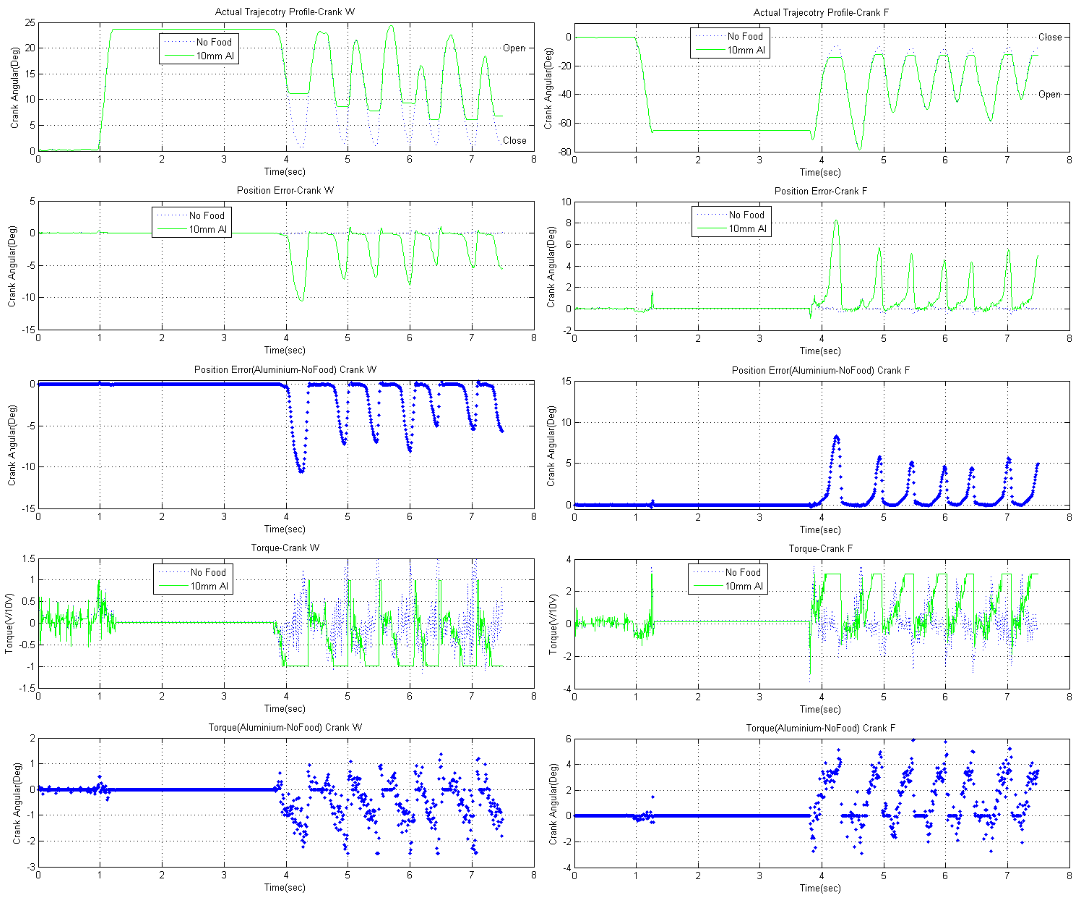


Figure 5.6 Robotic chewing results of hard food simulated by 10mm hard aluminium plate (CI Motion 2003)-Crank W and Crank F (Right)

Crank W and F were directly affected by the external force. In this experiment, the aluminium plate is too hard to be crashed by the teeth; therefore the torque should increase significantly when teeth touched the plate until it reached the torque limit (Table 5.5). After that, the movement is temporarily disabled until the overload force is released.

The position holding can be seen from the actual trajectory profile in Figure 5.6. In the graphs, both cranks had a peak cut during the close phase. Ideally, the movement holding trigger points for crank actual trajectory should be the same value because the shape of aluminium plate is not changeable. Actual trajectory profile (Crank F) exactly displayed this feature; however, the holding values of Crank W fluctuated. By making the experiment again, it is found that the fluctuation is caused by the sampling point interval.

Table 5.5 Properties of Crank W and Crank F (Right)

	Crank W	Crank F
Control Position Accuracy(Crank Degree)	0.08^{o}	0.12^{o}
Torque Limit	1.0V/10V	3.0V/10V

This time, a very huge position error is displayed and the error difference is also very remarkable. For Crank W, the occlusal phase is 5^{o} to 10^{o} in a 25^{o} range and it is 5^{o} to 8^{o} in a whole 80^{o} range for Crank F. It is interesting that the releasing half way has less big error difference points than the closing half way. This proves the motor torque can be quickly released by the gravity and a slight spring power within the linkage.

Torque displacement directly represents the strength of the external force from aluminium plate. Both cranks reached the maximum limit in the close phase and kept torque output at the maximum value until the external force dropped. Torque difference points in the last two graphs in Figure 5.6 are concentrated during the time when aluminium plate was touched by the bottom teeth, and they clearly explained the additional torque applied to the cranks. The maximum torque used for Crank W and crank F are 3V/10V and 6 V/10V, respectively.

The passive force control has been proven correct in that a small increase in external force applied to the robotic jaw has caused an increase in required motor torque of the actuators.

5.3.3 Food Simulated by Foam

Under the same conditions in Table 5.2, 10mm foam was inserted into the robotic jaw in the first jaw open cycle where both side of the bottom molar teeth can chew the foam. Motion profile results including actual trajectory, position error, position error difference, torque, and torque difference are compared with those of No-Food tests displays in Figure 5.8, Figure 5.9, Figure 5.10. The No-food experiment was done again under previous conditions in order to get more accuracy results.

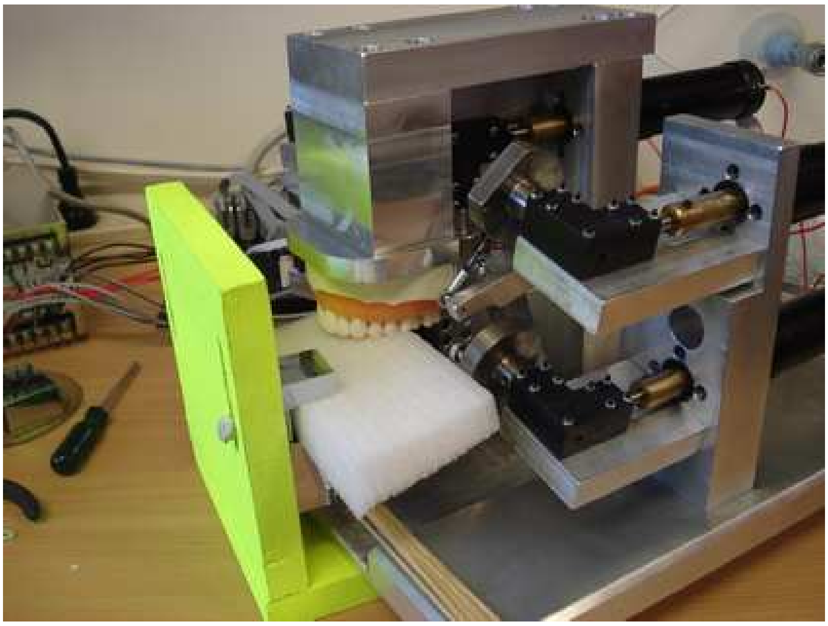


Figure 5.7 Massey robotic jaw chewing form

The behaviour of actuator should be slightly affected when a soft food is placed into the mouth. According to the property (soft material) and placed position (both side) of the foam, the trajectory profiles of actuators are anglicised as follows.

- Crank X and Crank Y

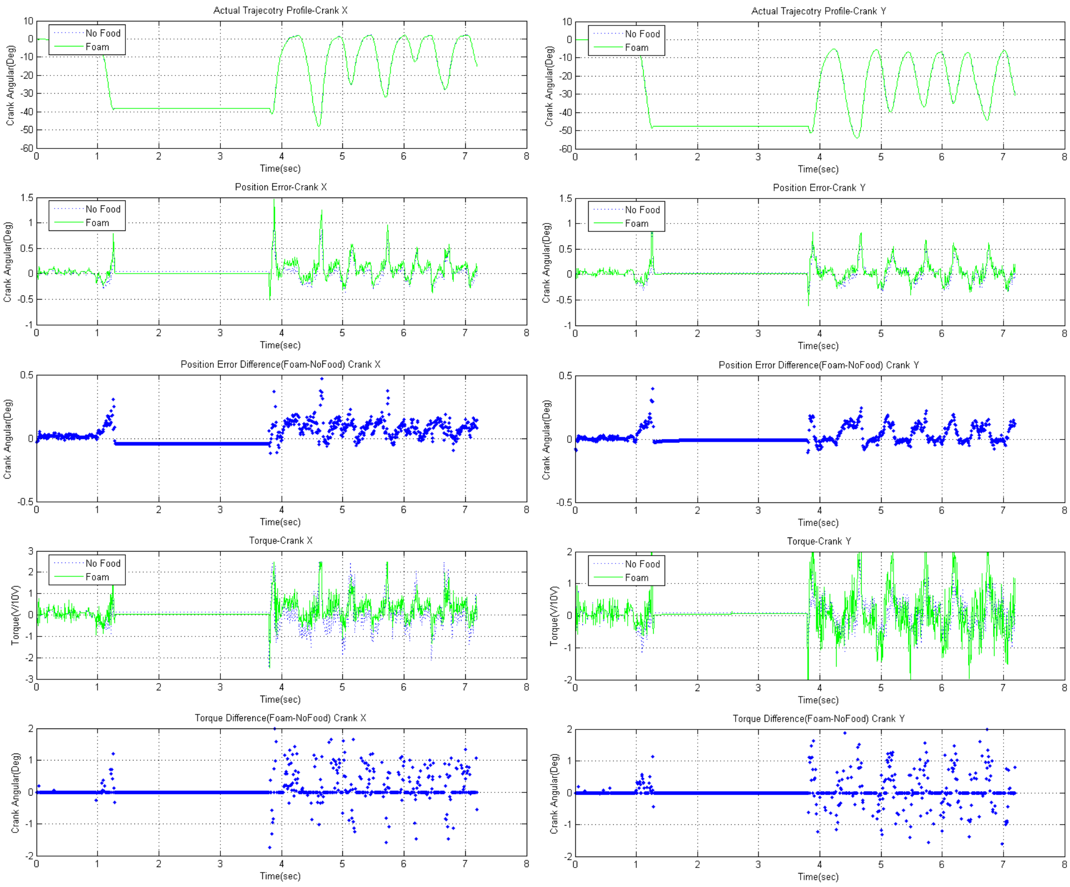


Figure 5.8 Robotic chewing results of hard food simulated by Foam (CI Motion 2003)-Crank X and Crank Y

The surface of the foam is little rough and can be broken by bottom teeth after bit. Once the teeth can go into the foam, the Crank X and Crank Y will meet additional force from the inside of the foam while the surface friction is no longer existed. Therefore, the little external force is always there.

The position error difference graphs in Figure 5.8 tells the foam had given less increased position error to the two actuators compare with aluminium plate. Torque graphs also prove less additional force was given to the actuators during the entire motion.

- Crank Z, W and Crank E,F

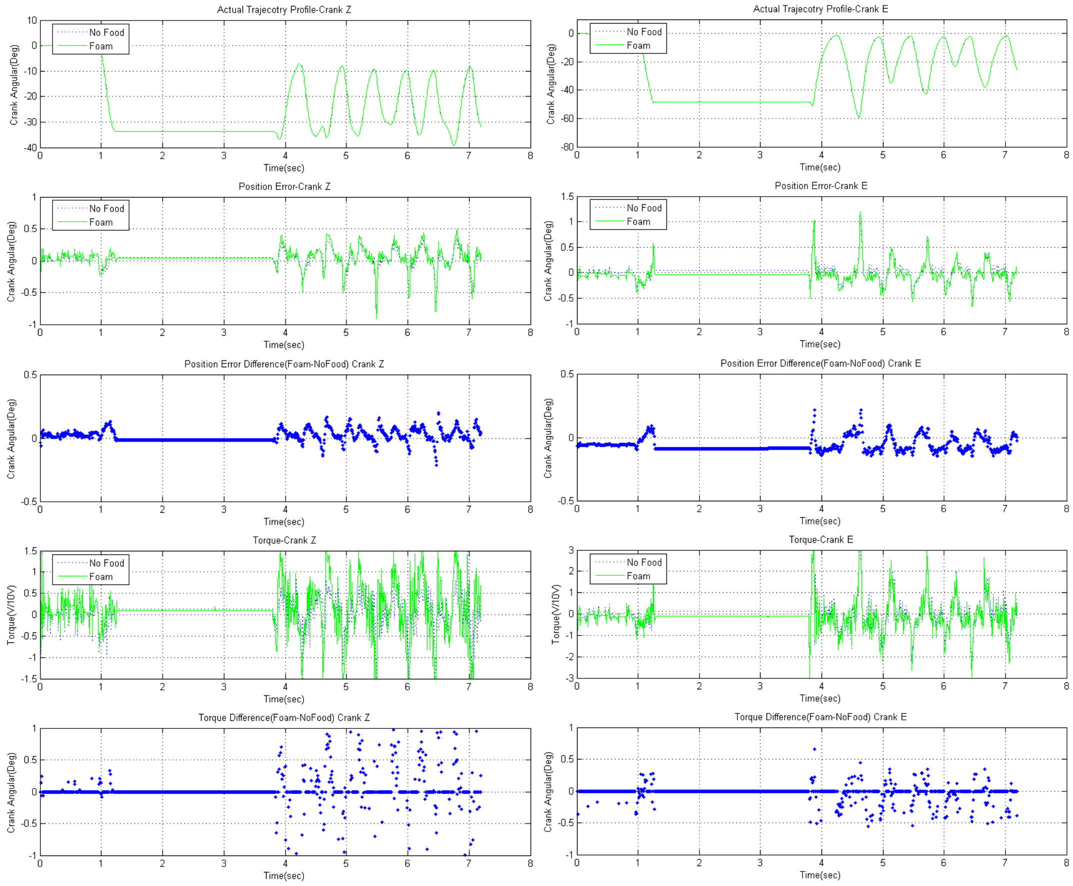


Figure 5.9 Robotic chewing results of hard food simulated by Foam (CI Motion 2003)-Crank Z and Crank E (Left)

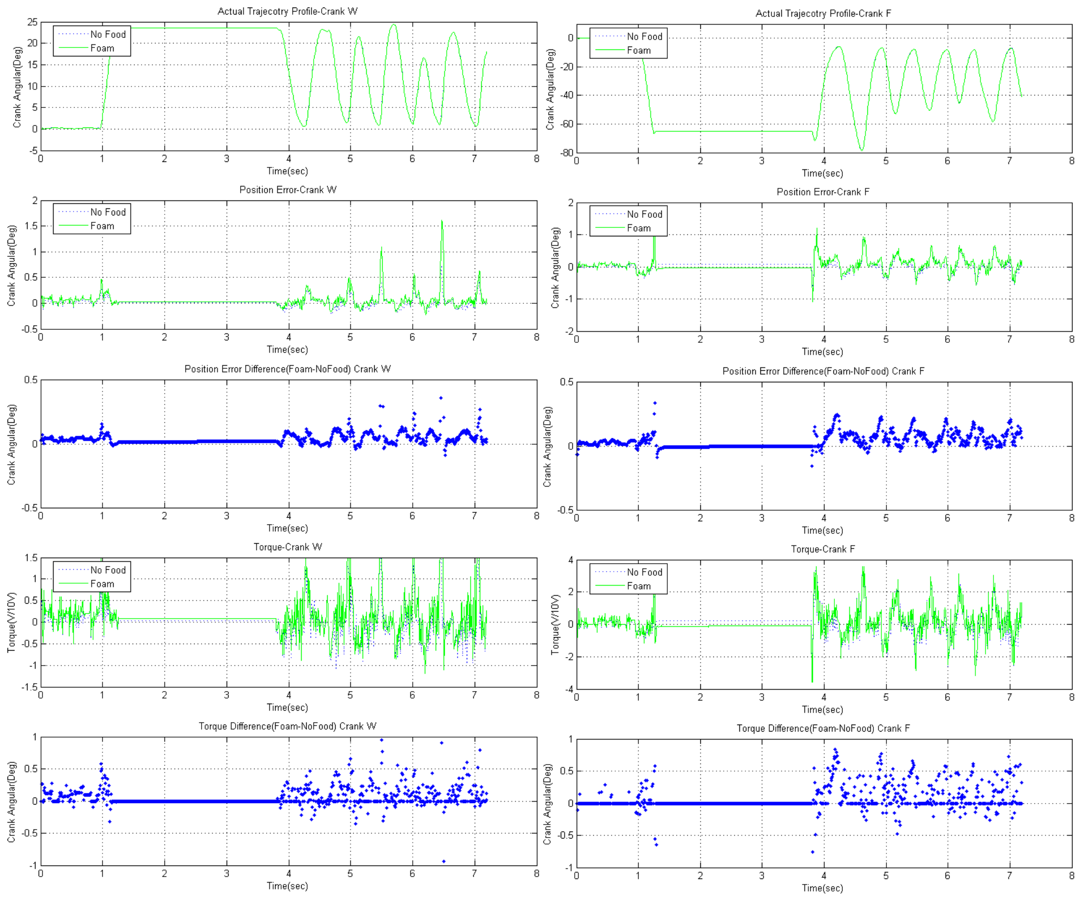


Figure 5.10 Robotic chewing results of hard food simulated by Foam (CI Motion 2003)-Crank W and Crank F (Right)

In this foam test, both side cranks touched the simulated soft food and chewed together. The foam is so soft that the position error difference and torque difference has only minor increase in Figure 5.9 and Figure 5.10.

5.4 Conclusions

The torque limit command (TL) was used as a means of passive force control by holding with the torque limit until an accessible position point comes during the occluding phase. The passive control can eliminate the excessive effects from an accelerating jaw on the torque applied from motors, and enable to control the torque applied from motors to only be dependent on force at jaw. The position error, especially during the occlusal area, was significantly controlled after adding a force limit. Also, the action of the robot chewing aluminium plate was no longer “rude” and out of control. This passive force control algorithm helped the PID control provided by Galil controller and achieved the goal of stable control.

The evidence shows that the passive force control can be used as an auto-repair fuse to protect motors and can monitor the additional force feed backed to the motors. In the other words, the property of the chewed food can be sponsored through the torque graph; however, the limit of actuator torque is affect by both output maximum jaw force and the acceleration, which relates to the accuracy.

Chapter 6 Conclusion and Recommendation

6.1 Conclusion

A proper control system for the Massey robotic jaw to mimic human chewing behaviours was studied and the following conclusions could be made. There were firstly some control algorithms that have been tested for processing stable and reliable control when the gain was unknown. Thanks to the advanced Galil control system, most of them had been run and the control torque, speed, acceleration can all be displayed. The power of the motor for each crank is enough to push the jaw move under most chewing behaviours. Maximum value of the factors such as torque limit or chewing speed can be increased several times if we upgrade linkage material.

An effective control method for linear motor is reported. A desired controller is used and tuned by this method with low computation to the computer. The forces and its' effects which are complex or unknown can be limited with learning experiments accurately. By this method, one can attain to good performance in comparison with other complex method.

The robot enables to run chewing trajectories and to apply estimated chewing forces. The control of the chewing robot has been successful. If more complex programming and fuzzy logic theory can be adopted, a probably more smooth and safe chewing motion would have been possible to produce

6.2 Recommendation

There are still some things that can be improved. Although Galil says the controller could run itself offline without host interference, the size of the board memory is limited. According to the user manual, the microcomputer of controller has only 512K RAM which has been assigned to variables, array elements and application programs. Therefore, when applying a long trajectory with 2ms segment, the data will exceed the capacity of the controller. In this case, instead using the terminal software accompanied with the controller, a VB programming needs to be written and performs continuous trajectory downloading to the controller card as well as drawing feed-backed trajectory displacement.

Furthermore, the system could achieve more accuracy during the occlusal area when more sensors are used to provide addition feedback loop or using an advanced control algorithm.

Reference

[1] "Dental Robotics Groupe- WY-5 Series", *http://www.takanishi.mech.waseda.ac.jp/research/jaws/index.htm*

[2] INSTRON, "Benefits of Food Texture Analysis", *http://www.instron.cn/wa/applications/food/benefits.aspx*

[3] W.L. Xu, J. Torrance, "Kinematics and Experiments of a Life-Sized Masticatory Robot", Vol.55, No.5, 2008. pp. 2121-2132.

[4] H. Takanobu, A. Takanishi, "Integrated dental robot system for mouth opening and closing training", *2002 IEEE International Conference on Robotics & Automation*, Washington, DC, May, 2002. pp. 1428-1433.

[5] H. Takanobu and A. Takanishi, "Dental robotics and human model", *International IEEE EMBS Conference on Neural Engineering, Capri Island*, May 20-22, 2003. pp. 671-674.

[6] Vatikiotis.B, E.; Ostry. "Analysis and modeling of 3D jaw motion in speech and mastication". *1999 IEEE International Conference*, Page(s):442 - 447 vol.2, Oct. 1999

[7] Takanobu, H.; Tanase, T. "Adaptive masticatory jaw motion using jaw position and biting force information", 1994. *IEEE International Conference on MFI '94*, Oct. 1994 Page(s):360 - 365

[8] T. R. Murphy, "The timing and mechanism of the human masticatory stroke", *Arch. Oral. Biol.*, vol. 10, no. 6, pp. 981-993, Nov.-Dec. 1965.

[9] C. H. Gibbs, H. C. Lundeen, P. E. Mahan, and J. Fujimoto, "Chewing movements in relation to border movements at the first molar," *J. Prosthet. Dent.*, vol. 46, no. 3, pp. 308-322, Sept. 1981.

[10] Vatikiotis-Bateson, E.; Ostry, D.J. "Analysis and modeling of 3D jaw motion in speech and mastication". *IEEE SMC '99 Conference Proceedings*, 12-15 Oct. 1999 Page(s):442 - 447 vol.2

[11] S.L. Andresen. Herbert A. Simon: AI Pioneer. *IEEE Intelligent Systems*, 16(4):71-72, July/August 2001.

[12] P.Kyberd. “The intelligent hand”. *IEEE Review*, pages 31-35, September 2000.

[13] DeLong, R.; Douglas, W.H. “An artificial oral environment for testing dental materials. Biomedical Engineering”, *IEEE Transactions*, Volume 38, Issue 4, April 1991 Page(s):339 - 345

[14] Pap, “A Chewing Robot Based on Parallel Mechanism-Analysis and Design”, *Massey University, Palmerston North,* 2006

[15] T.Tsuji, A. Jazidie, and M. Kaneko. “Cooperative multi-arm robots”. *IEEE Transactions on Systems Man and Cybernetics*, 27(5):862-867, October 1997.

[16] G. Muscato. “Soft computing techniques for control of walking robots”. *IEEE Computing and Control Engineering Journal*, pages 193-200, August 1998.

[17] Gee W. Ng, “Intelligent Systems-Fusion, Tracking and Control”. *DSO National Laboratories and National University of Singapore.* 2003.

[18] Perzanowski, D.; Schultz, A.C. “Building a multimodal human-robot interface”. *IEEE Intelligent Systems*, 16(1):16-21, Jan-Feb 2001

[19] Tom Simonite, “Dental robot chews over a toothy problem”, *NewScientist.com news service*, 28 June 2007

[20] B. Daumas, W.L. Xu, J. Bronlund, “Jaw mechanism modeling and simulation”, *Mechanism and Machine Theory,* MAMT1048. 5 Feb.2005

[21] W. L. Xu,, J.-S. Pap, “Design of a Biologically Inspired Parallel”, *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, VOL. 55, NO. 2, FEB 2008,

[22] Galil Motion Control, Inc., “Force Feedback, Application Note #2419”

[23] W.Y Lee, C,L Shih, S.T Lee, “Force Control and Breakthrough Detection”, *IEEE/ASME TRANSACTIONS ON MECHATRONICS*, VOL.9, NO. 1, MARCH 2004

[24] Imen, S. J.; Shakeri, M. “Feed forward adaptive control of a linear brushless DC motor”, *SICE, 2007. Annual Conference*, 17-20 Sept. 2007 Page(s):2200 - 2204

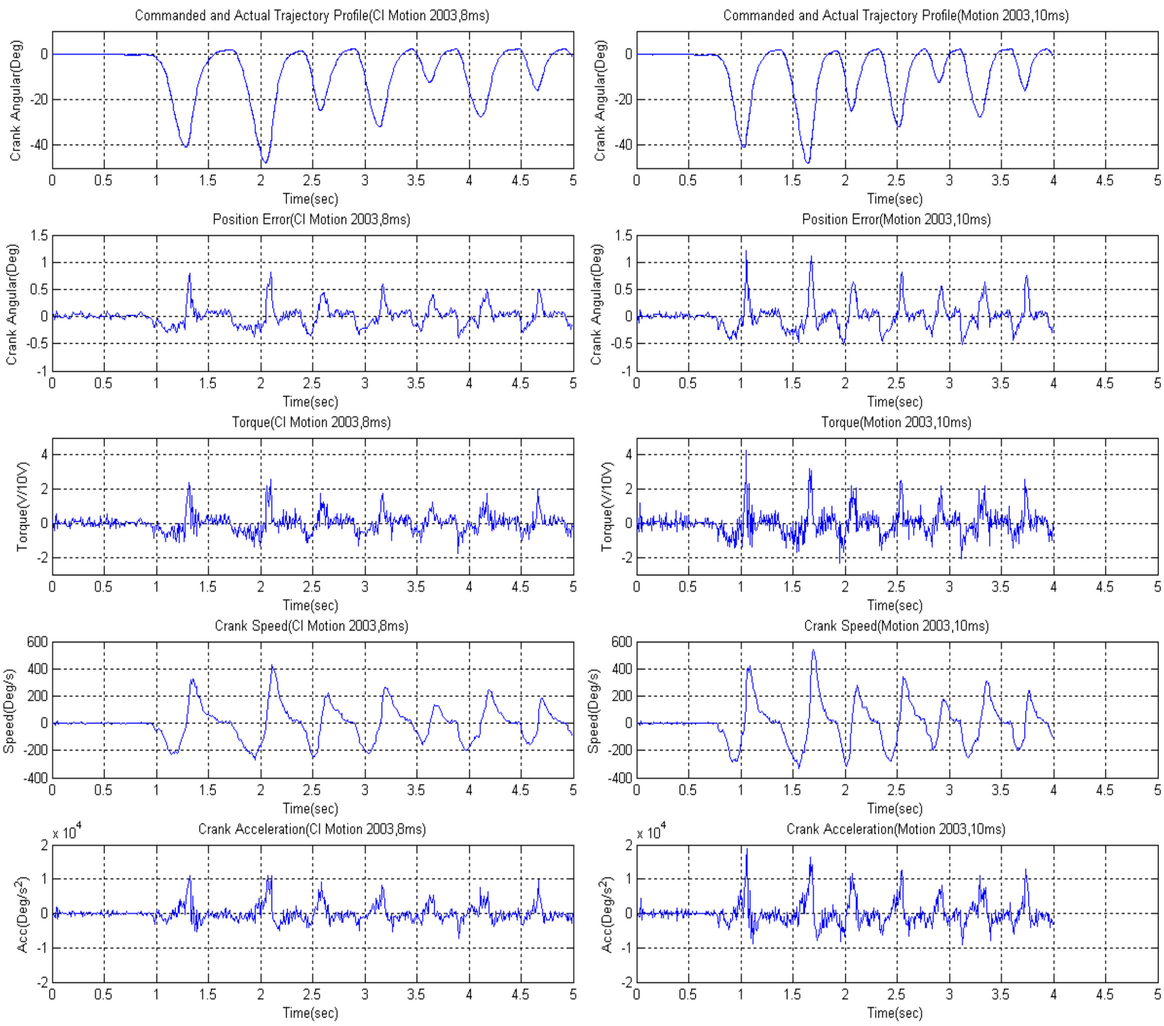
[25] C.W. Kennedy and J.P. Desai, “Modelling and Control of the Mitsubishi PA-10 Robot Arm Harmonic Drive System,” *IEEE/ASME Trans. Mechatronics,* vol.10, no.3, pp.263-274, Jun. 2005.

[26] Galil Motion Control Inc., “Technical References”, www.galilmc.com 2002. P1

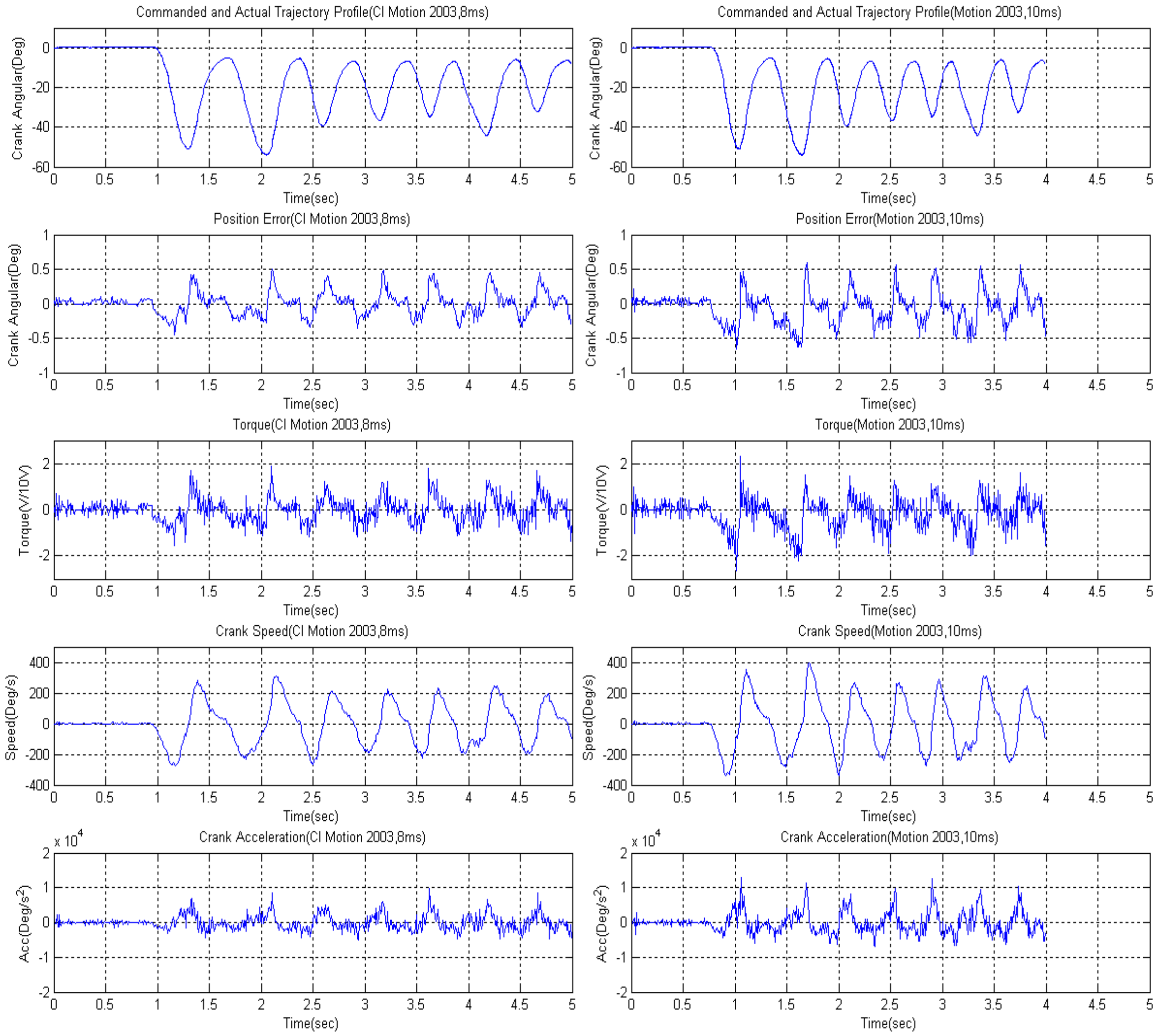
[27] S.H. Sadiq and K. Sultan, “DC Motor Identification”, P1
[28] T.E. Djaferis “Automatic Control- The Power of Feedback using MATLAB”, *University of Massachusettes*, Amherst, P63, P64
[29] S.J. O’Neil, “Motion Control Handbook,” *in* [*www.micromo.com*](http://www.micromo.com), Micro Mo Electronics, Inc. 1998
[30] Z. Man, M. O’Day, and X, Yu, “A robust adaptive terminal sliding mode control for rigid robotic manipulator,*” J. Intell. Robot. Syst: Theory Appl.*, vol. 24, pp.23-41, 1999
[31] K.Astrom and T.Hagglund, “PID Controllers: Theory, Design, and Tuning” 1988
[32] K. Ogata, “Modern Control Engineering-Fourth Edition”, *University of Minnesota*, 2002
[33] “ DMC-1700/1800 Manual Rev. 1.2l”, *Galil Motion Control*, Inc. pp119, Nov/2005
[34] C. Moler, “Numerical Computing with MATLAB”, *http://www.mathworks.com/moler/interp.pdf*, Chapter 3
[35] R. F. Woolley, “Interpolation”, *http://www.dtp-aus.com/interpol.htm*, 1997-2003
[36] Wikipedia, “Interpolation”, *http://en.wikipedia.org/wiki/Interpolation*, Wikimedia Foundation, 2006
[37] C. DE Boor, “A Practical Guide to Splines”, *Springer-Verlag*, New York, 1978.
[38] Bekit, B.W.; Seneviratne, L.D.; Whidborne, “Fuzzy PID tuning for robot manipulators”, IECON '98. *Proceedings of the 24th Annual Conference of the IEEE*, 31 Aug.-4 Sept. 1998 Page(s):2452 - 2457 vol.4
[39] R.G. Franks and C.W. Worley, “Quantitive Analysis of Cascade Control”, *Industrial and Engineering Chemistry*, vol. 48, no. 6, pp. 1074-1079, 1956.
[40] Kaya, I.; Atherton, D.P, “Improved Cascade Control Structure for Controlling Unstable and Integrating Processes”, *Decision and Control, CDC-ECC '05.* 12-15 Dec. 2005 Page(s):7133 – 7138
[41]J.D. Torrance, J.-S. Pap, W.L. Xu, J.E. Bronlund, K.D. Foster, “Motion control of a chewing robot of 6 RSS parallel mechanism”, *in Proceedings of 3rd International Conference on Autonomous Robots and Agents (ACARA 2006)*, 12-14 Dec 2006, Palmerston North, New Zealand, pp.593-598.

[42] W.L. Xu, J. Bronlund and J. Kieser, “A robotic model of human masticatory system for reproducing chewing behaviours”, *IEEE Robotics and Automation Magazine,* vol. 12, no.2, 2005, pp90-98.
[43] J.P. Merlet, “Parallel Robots”, *2nd edition*, *Springer*, the Netherlands, 2006, 394p.

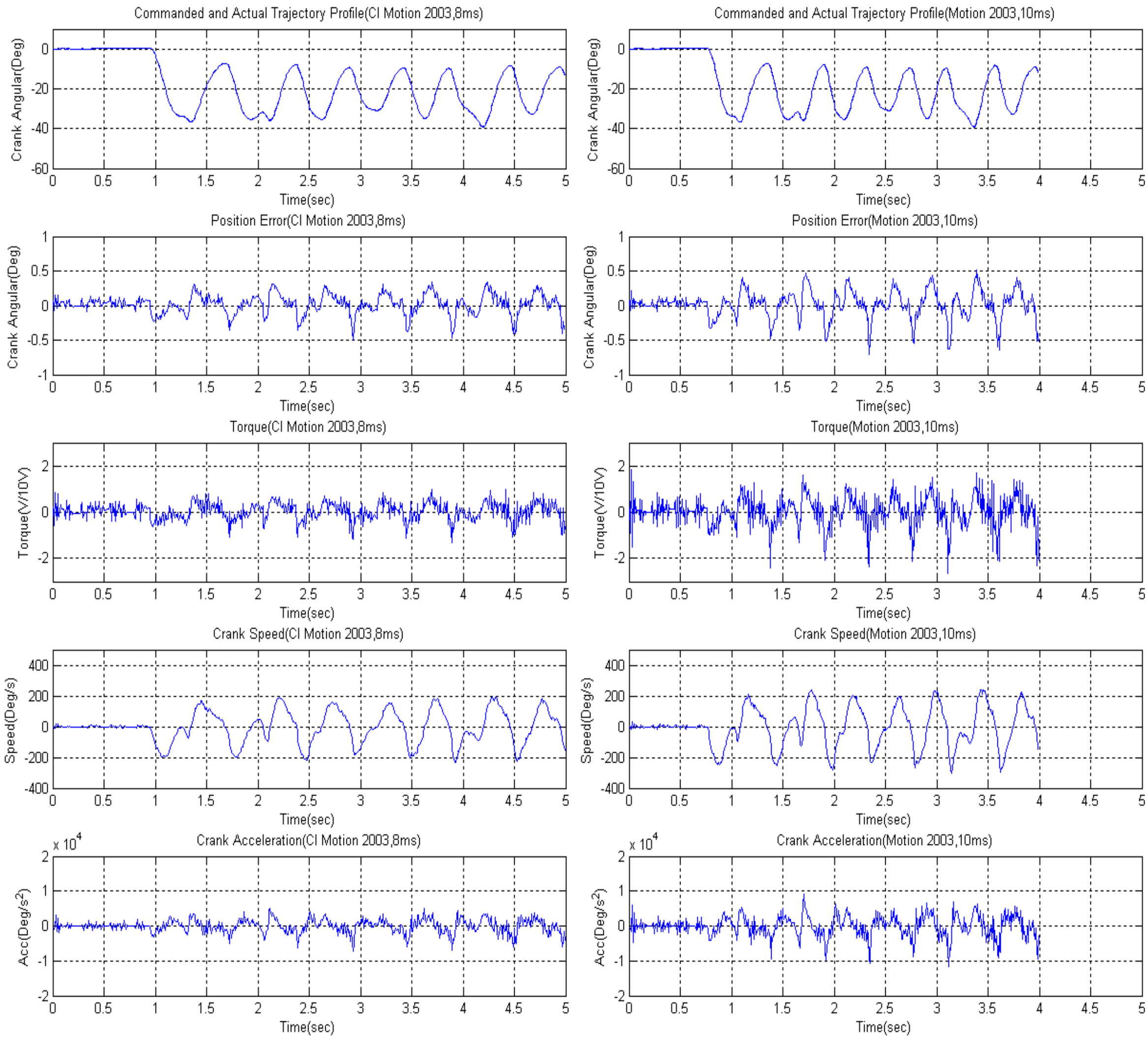
Appendix 1 Experiment results of Motion 2003 and CI Motion 2003 during first seven chewing cycles without load (no food)



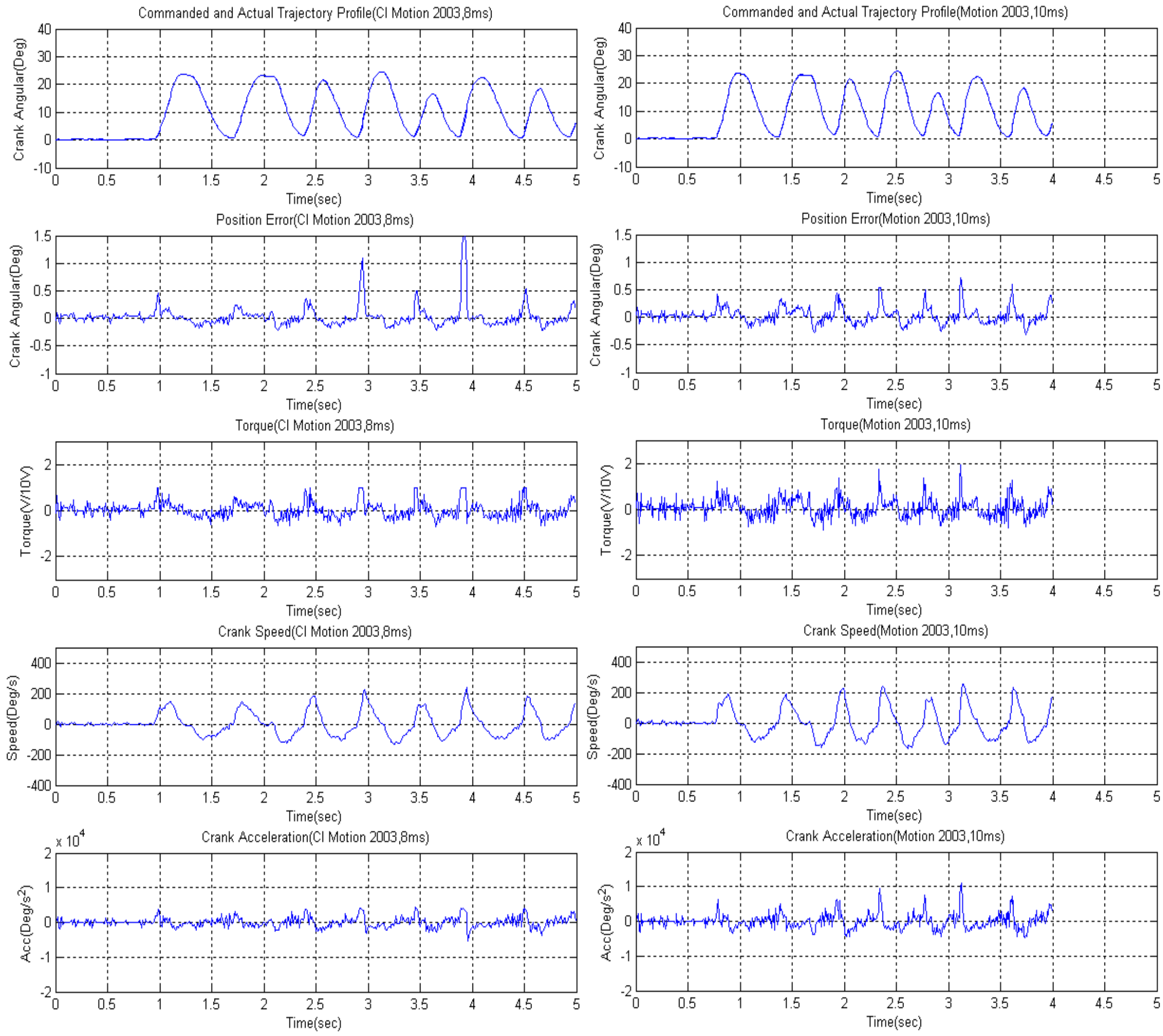
Appendix Figure 1.1 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, Torque, Crank Speed, Crank Acceleration Results Running on CI Motion 2003 (left) and Motion 2003 (right) during First Seven Chewing Cycles on Crank X



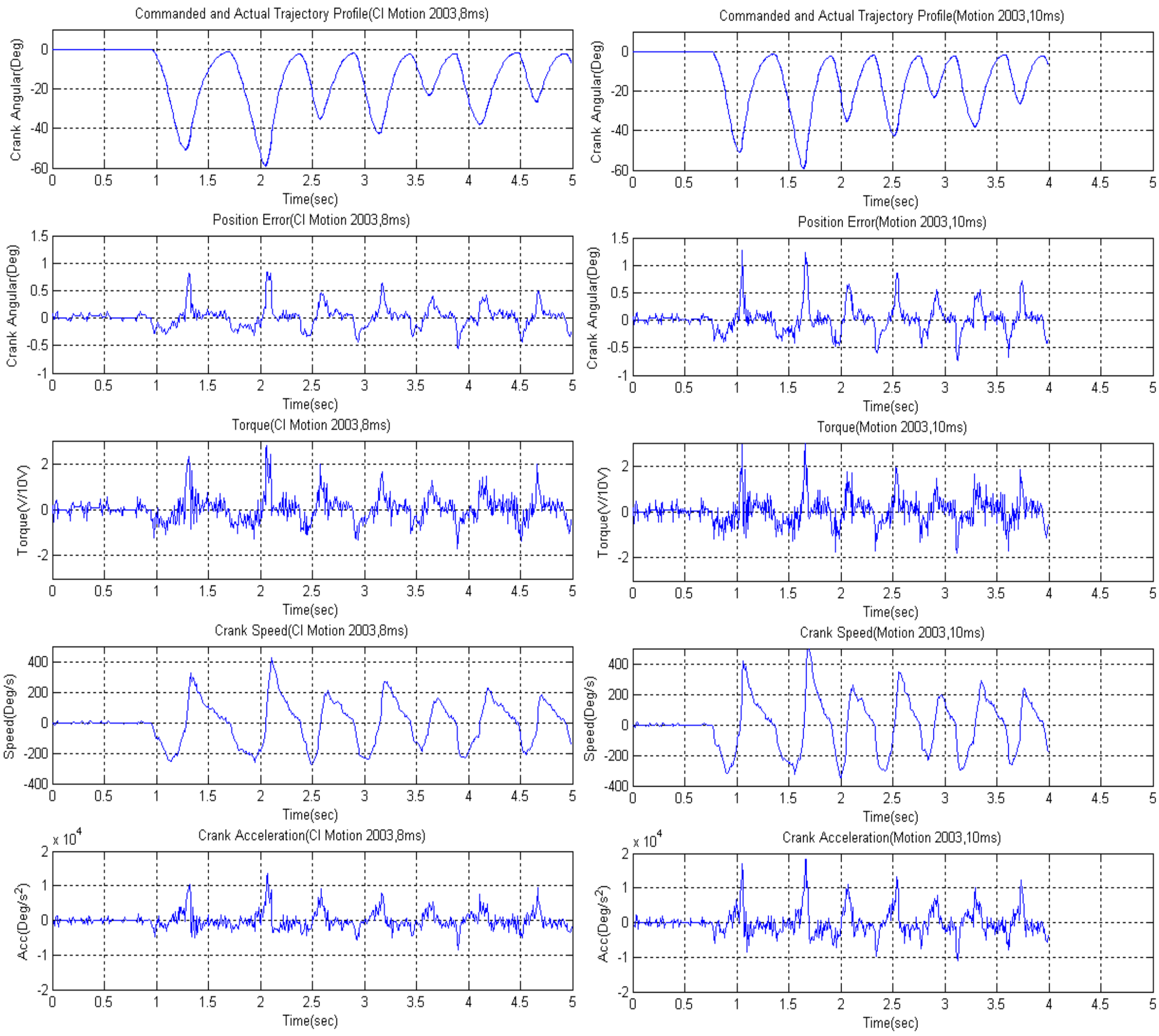
Appendix Figure 1.2 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, Torque, Crank Speed, Crank Acceleration Results Running on CI Motion 2003 (left) and Motion 2003 (right) during First Seven Chewing Cycles on Crank Y



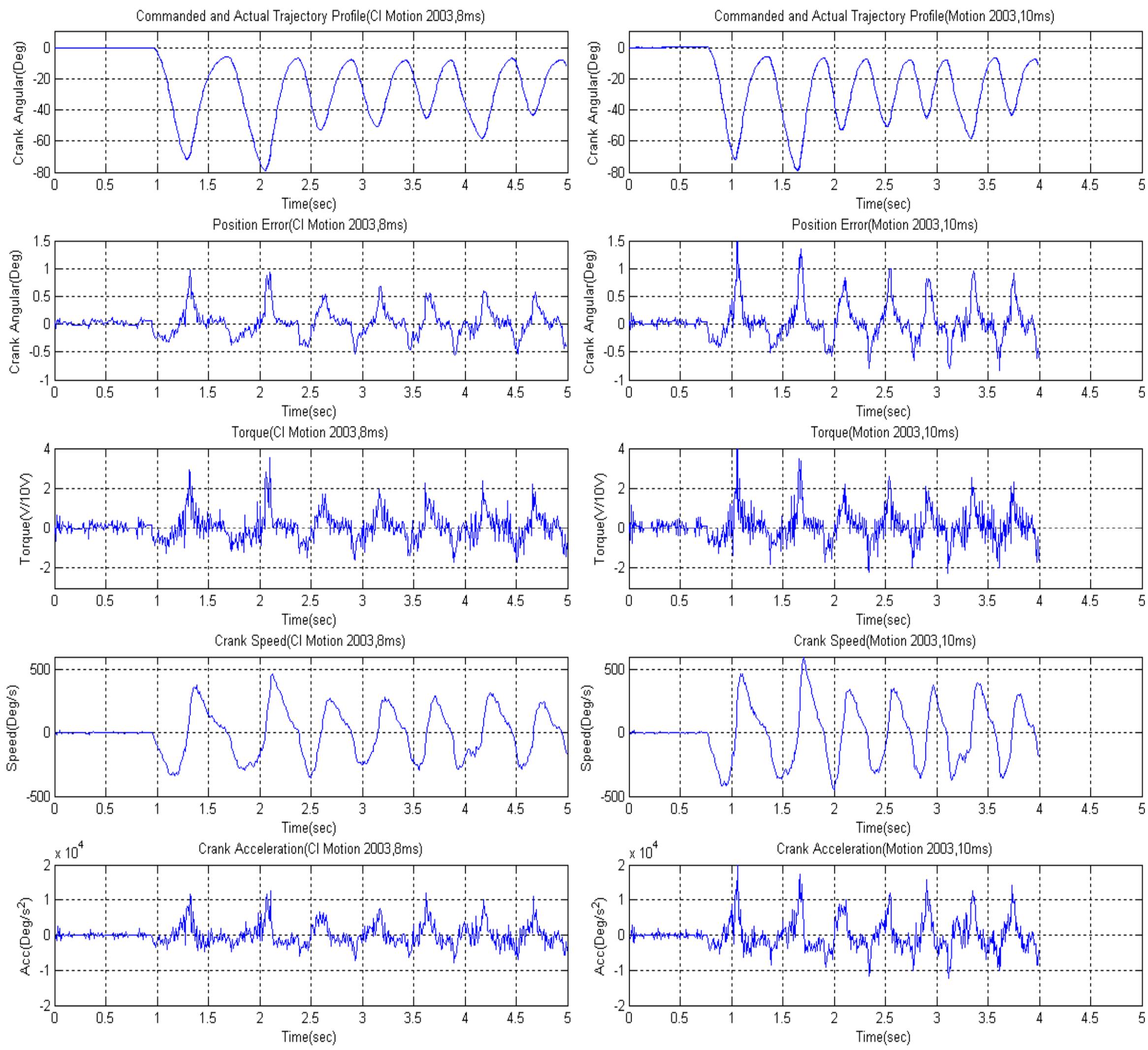
Appendix Figure 1.3 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, Torque, Crank Speed, Crank Acceleration Results Running on CI Motion 2003 (left) and Motion 2003 (right) during First Seven Chewing Cycles on Crank Z



Appendix Figure 1.4 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, Torque, Crank Speed, Crank Acceleration Results Running on CI Motion 2003 (left) and Motion 2003 (right) during First Seven Chewing Cycles on Crank W

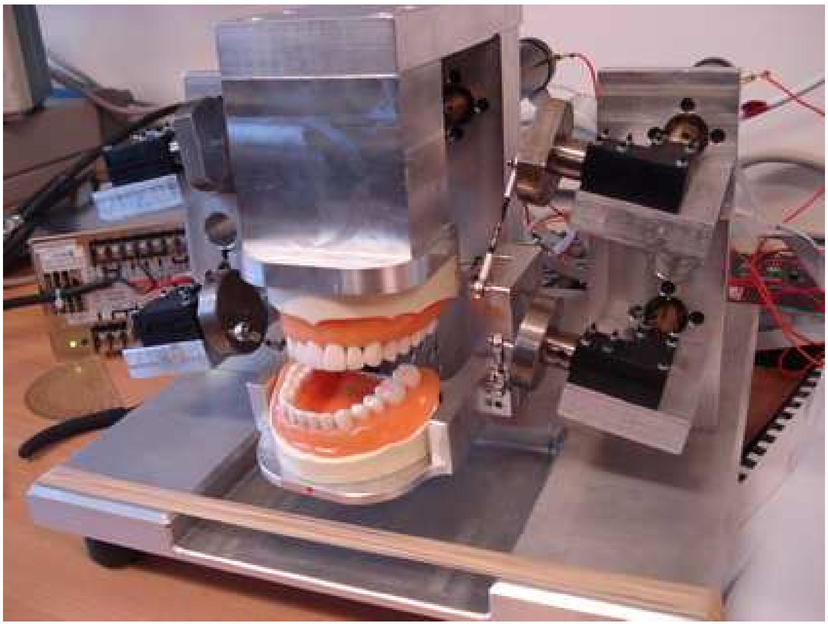


Appendix Figure 1.5 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, Torque, Crank Speed, Crank Acceleration Results Running on CI Motion 2003 (left) and Motion 2003 (right) during First Seven Chewing Cycles on Crank E

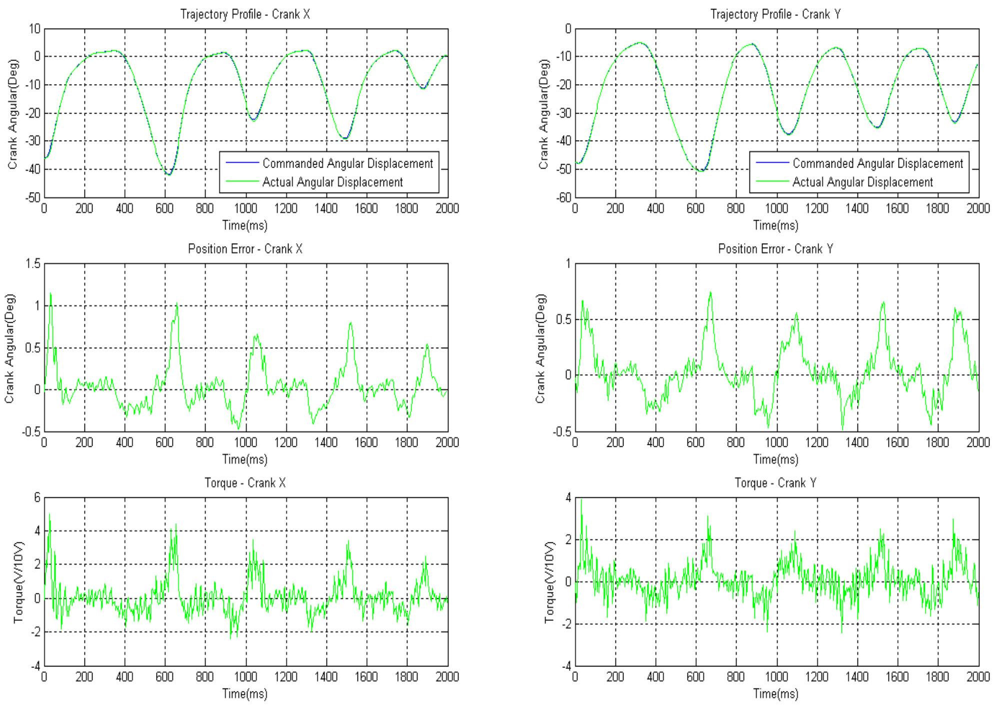


Appendix Figure 1.6 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, Torque, Crank Speed, Crank Acceleration Results Running on CI Motion 2003 (left) and Motion 2003 (right) during First Seven Chewing Cycles on Crank F

Appendix 2 Experiment results of Motion 2003 during first four chewing cycles

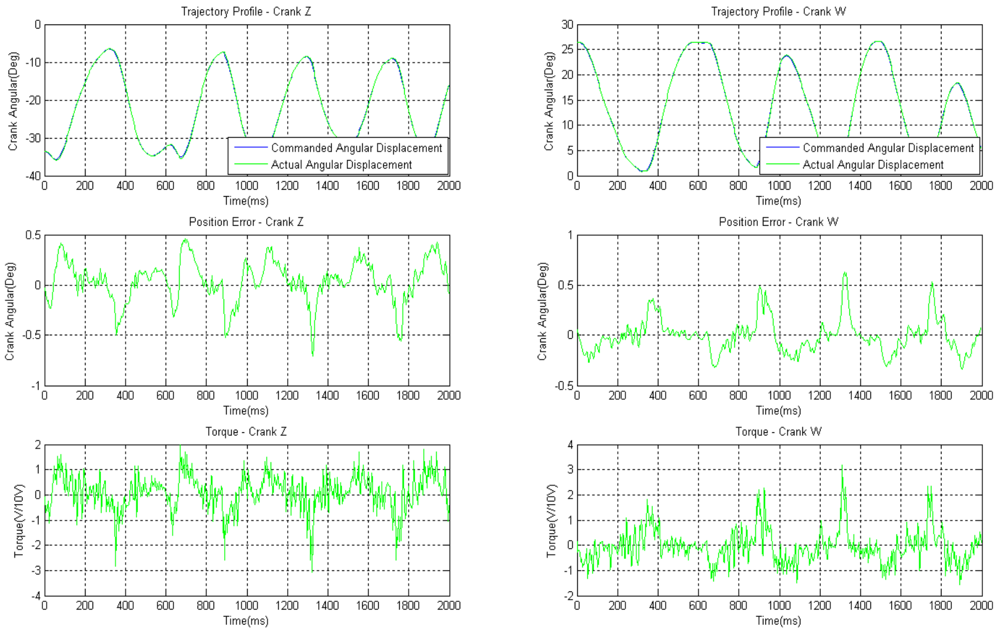


Appendix Figure 2.1 Massey robotic jaw running free

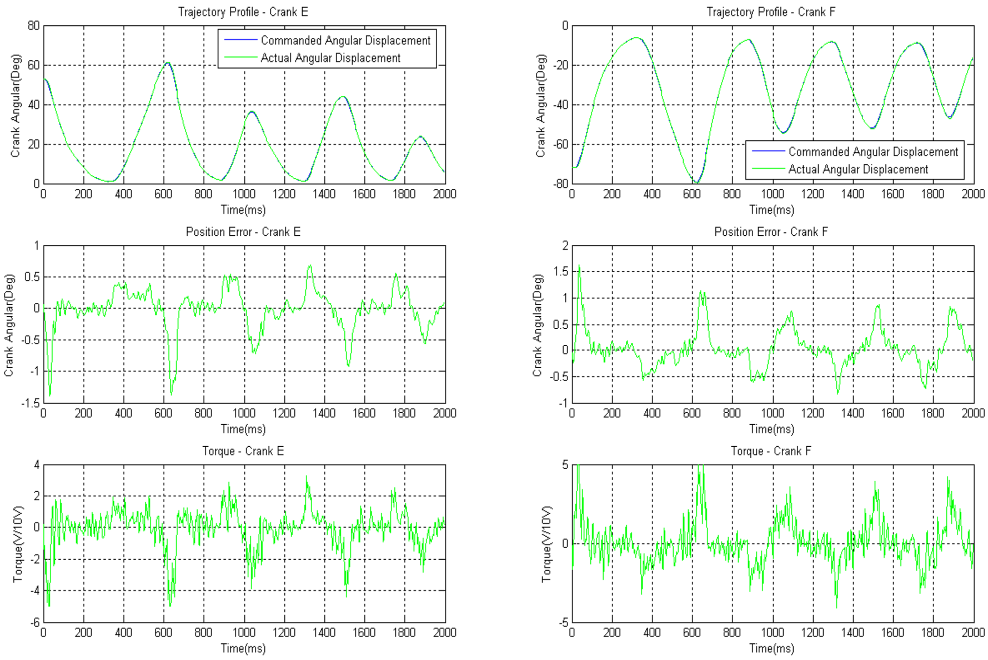


Appendix Figure 2.2 Experimental Results of Commanded and Actual Trajectory

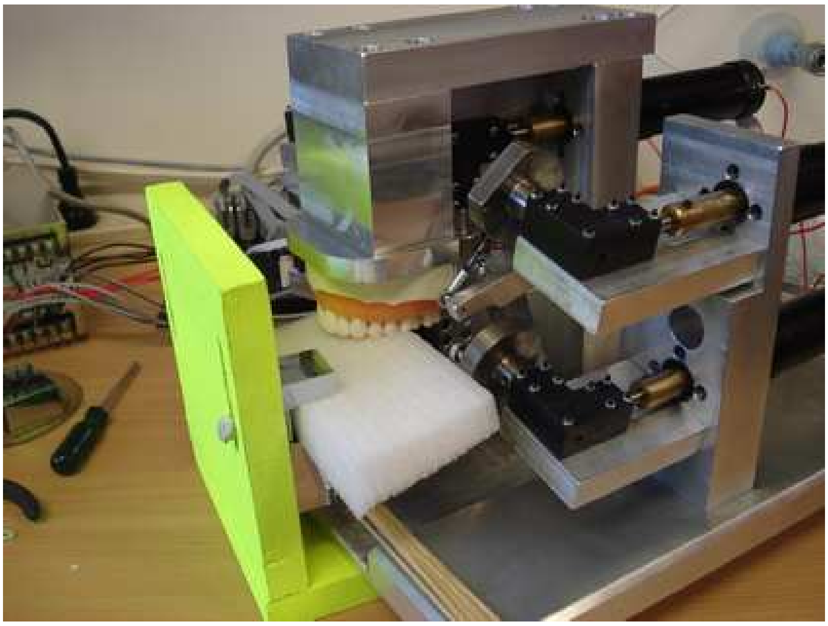
Profiles, Position Error, and Torque Running on Motion 2003 trajectory without food during the first four Chewing Cycles on Crank X and Crank Y



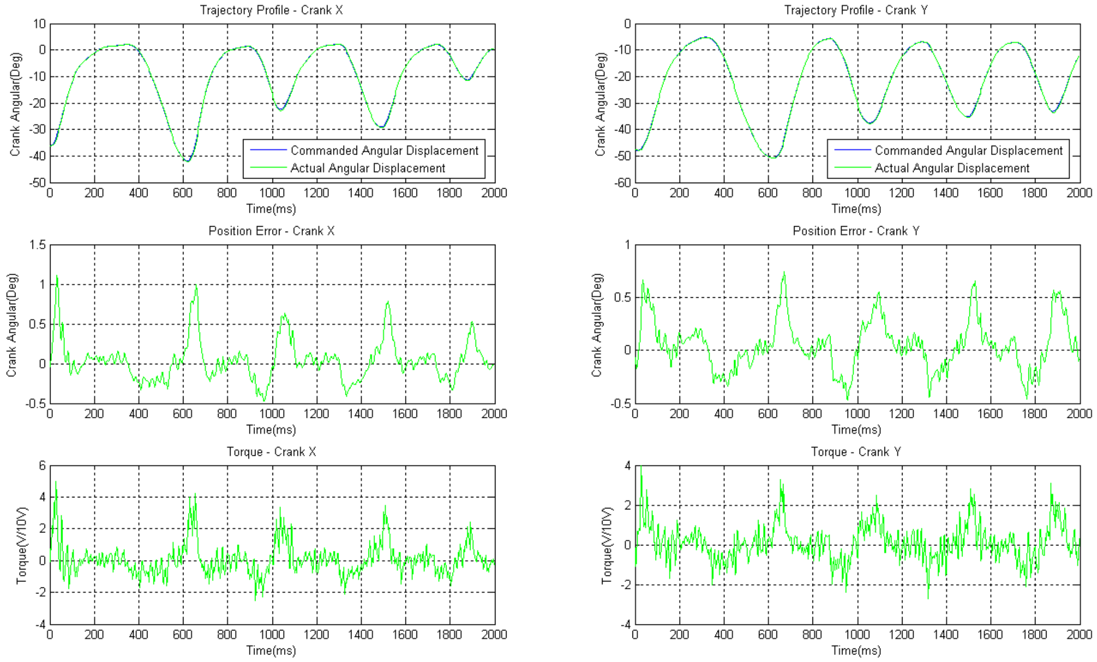
Appendix Figure 2.3 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, and Torque Running on Motion 2003 trajectory without food during the first four Chewing Cycles on Crank Z and Crank W



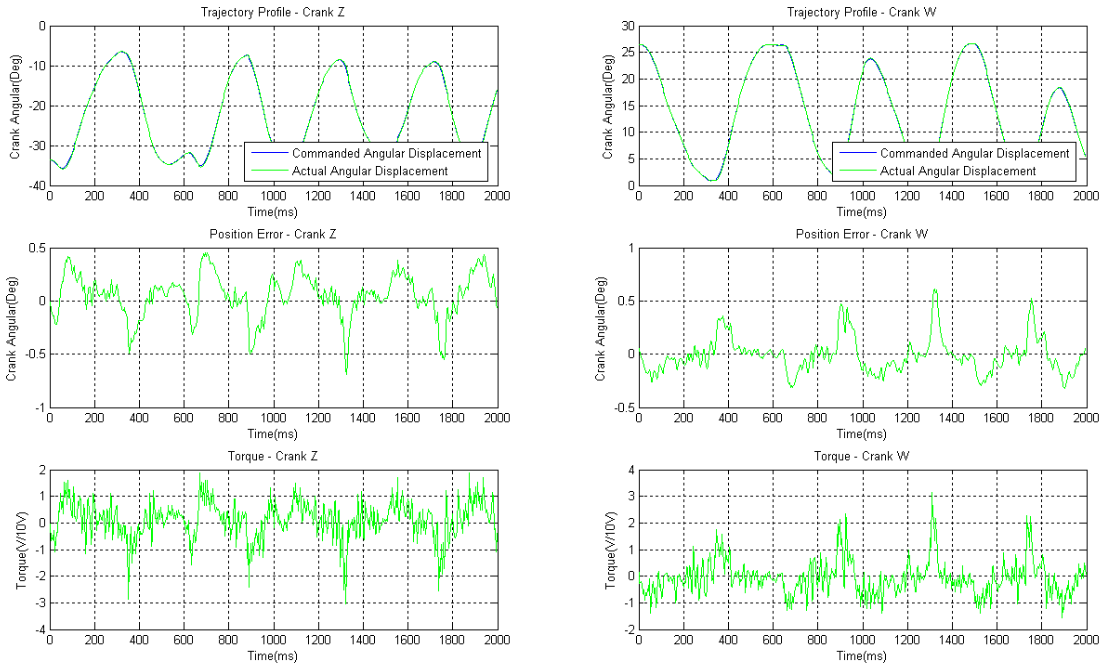
Appendix Figure 2.4 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, and Torque Running on Motion 2003 trajectory without food during the first four Chewing Cycles on Crank E and Crank F



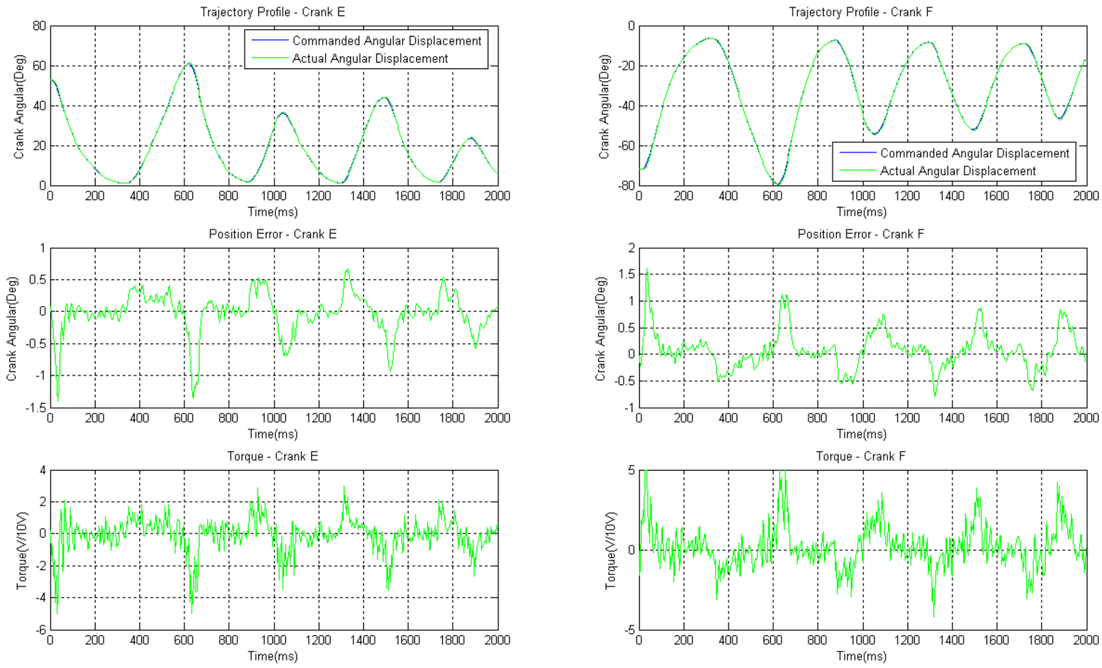
Appendix Figure 2.5 Massey robotic jaw chewing form



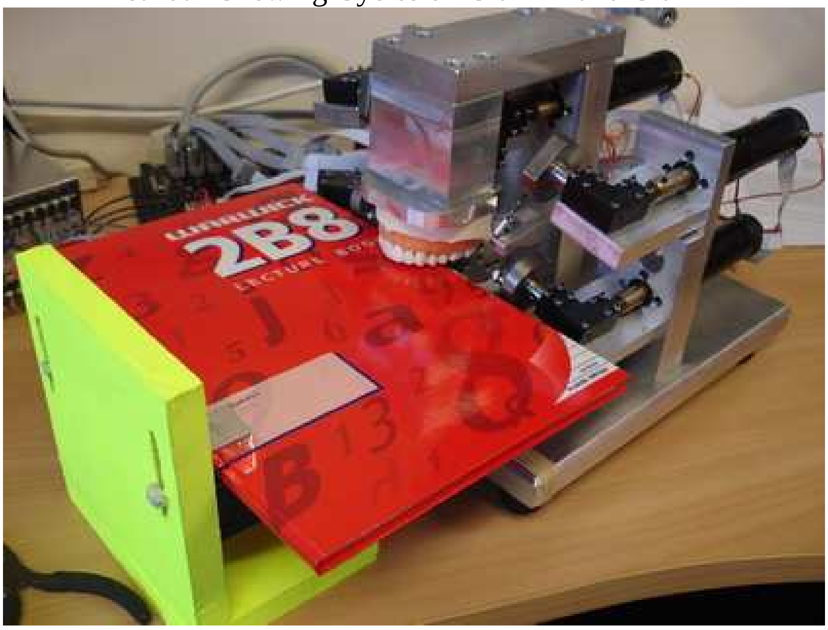
Appendix Figure 2.6 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, and Torque Running on Motion 2003 trajectory on 10mm foam during the first four Chewing Cycles on Crank X and Crank Y



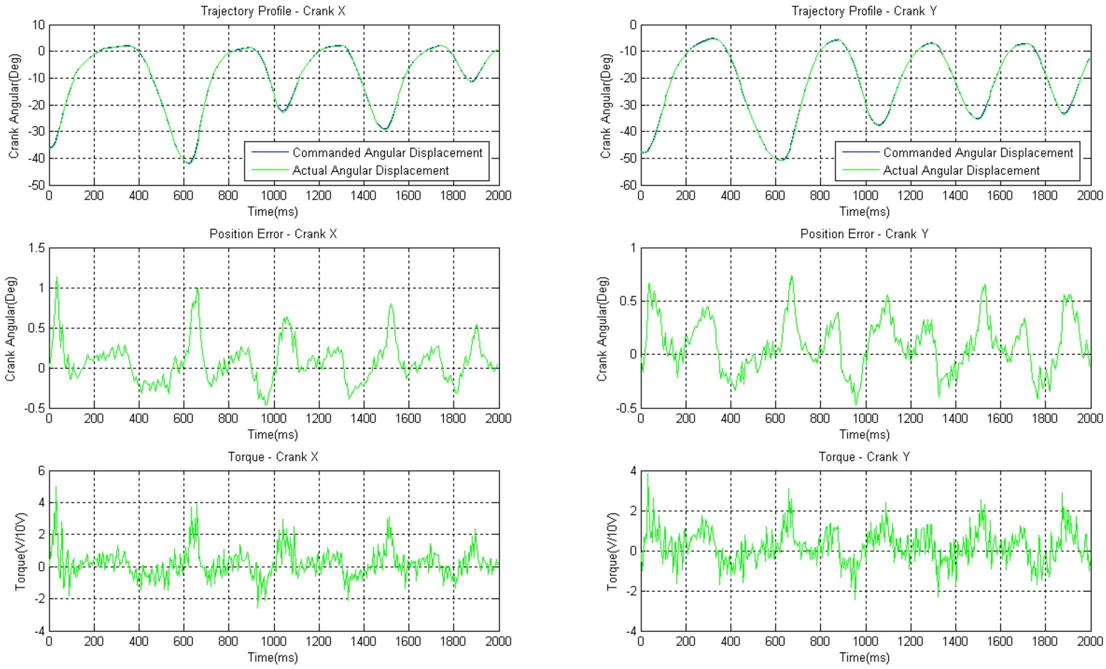
Appendix Figure 2.7 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, and Torque Running on Motion 2003 trajectory on 10mm foam during the first four Chewing Cycles on Crank Z and Crank W



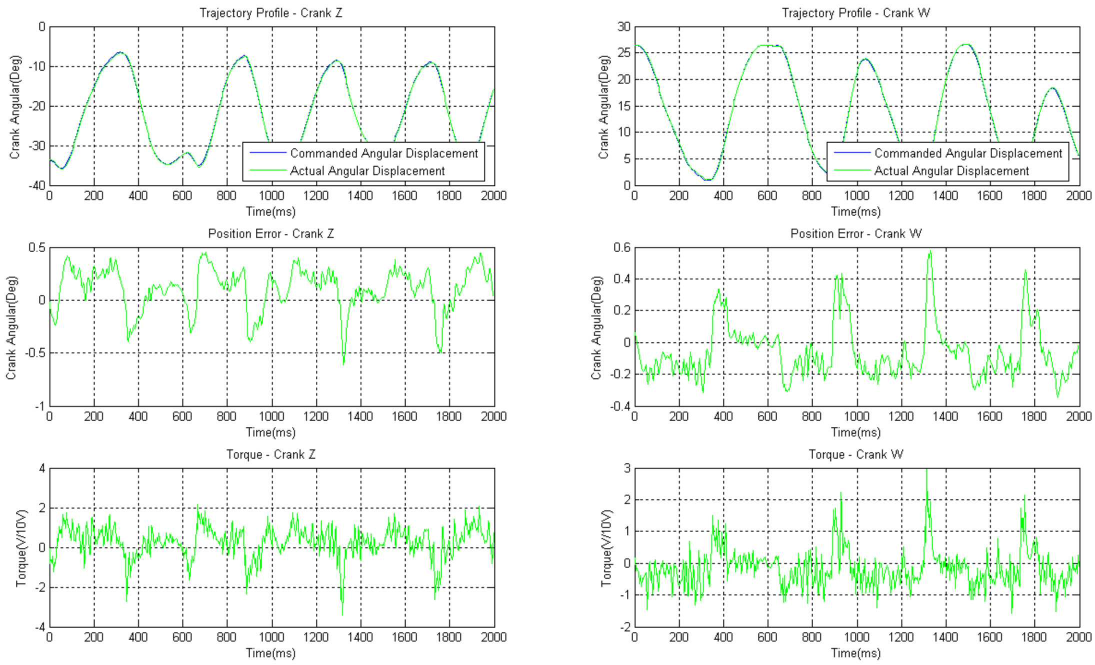
Appendix Figure 2.8 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, and Torque Running on Motion 2003 trajectory on 10mm foam during the first four Chewing Cycles on Crank E and Crank F



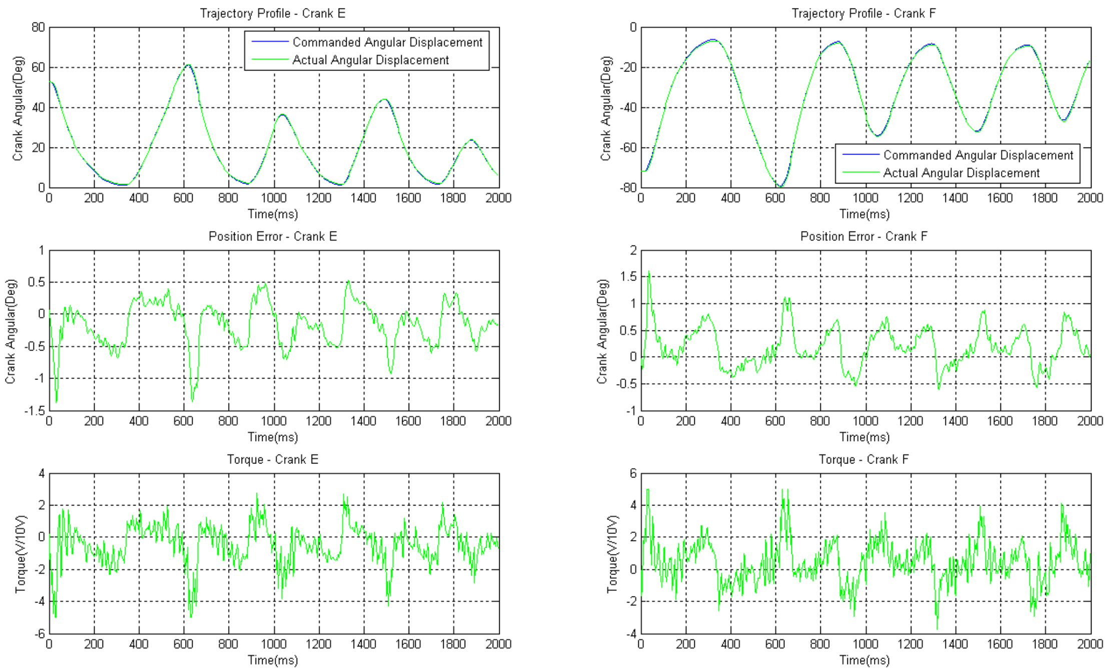
Appendix Figure 2.9 Massey robotic jaw chewing 2B book



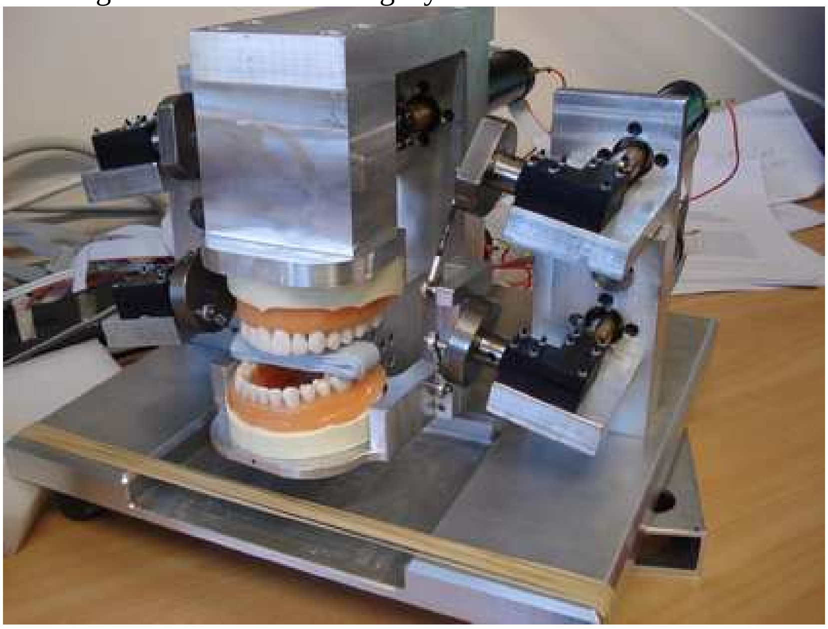
Appendix Figure 2.10 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, and Torque Running on Motion 2003 trajectory on 5mm book during the first four Chewing Cycles on Crank X and Crank Y



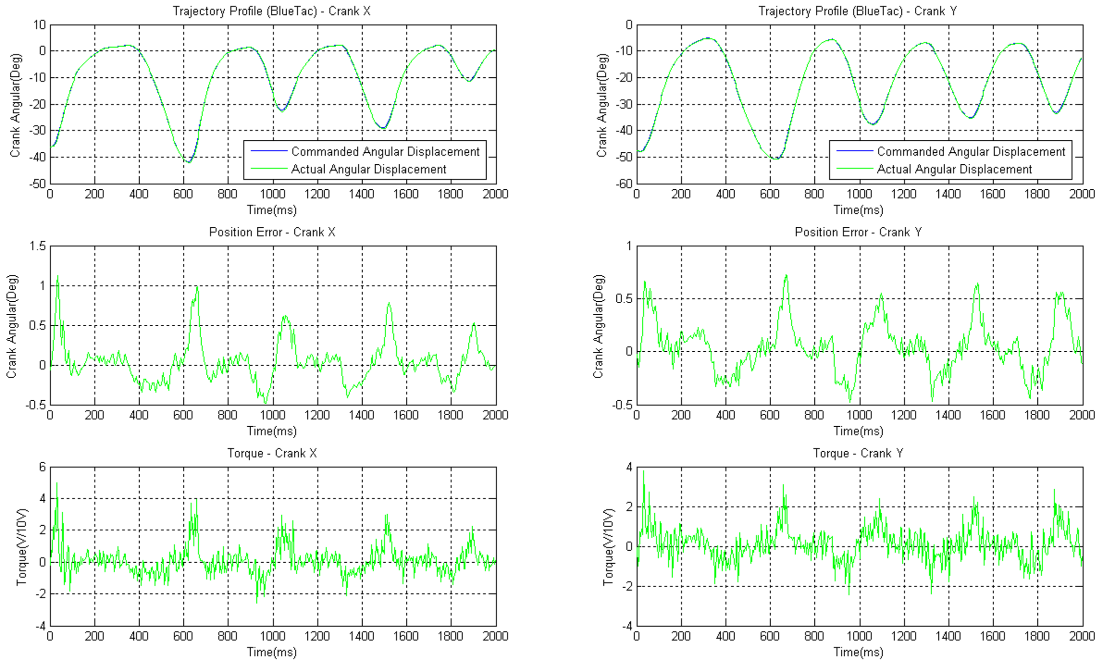
Appendix Figure 2.11 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, and Torque Running on Motion 2003 trajectory on 5mm book during the first four Chewing Cycles on Crank Z and Crank W



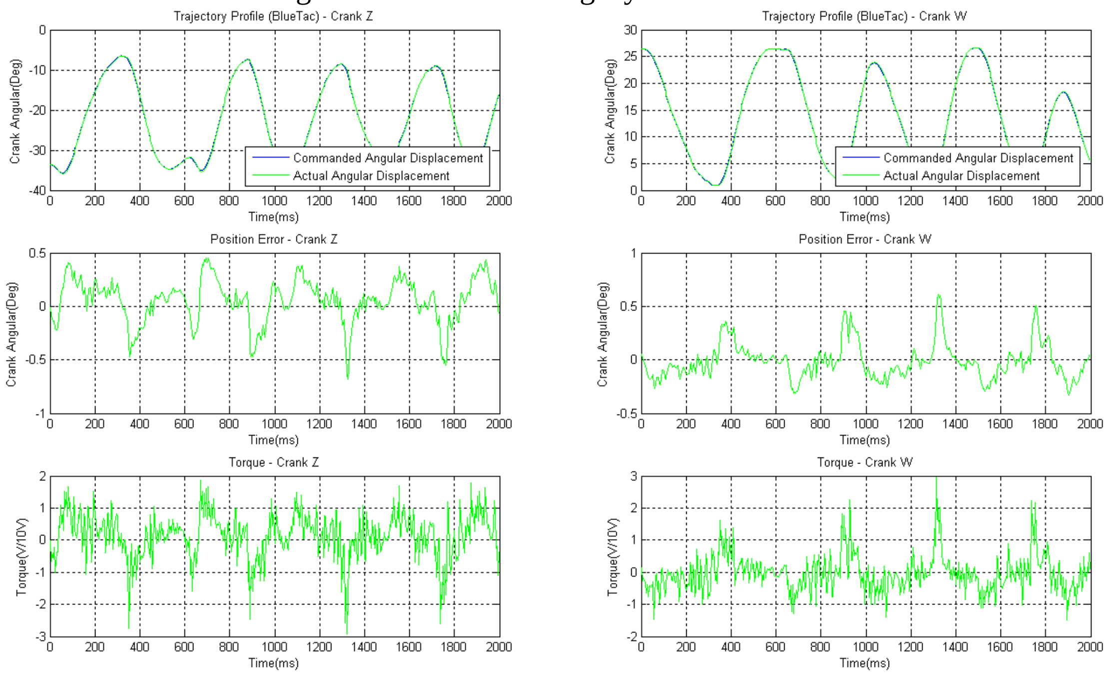
Appendix Figure 2.12 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, and Torque Running on Motion 2003 trajectory on 5mm book during the first four Chewing Cycles on Crank E and Crank F



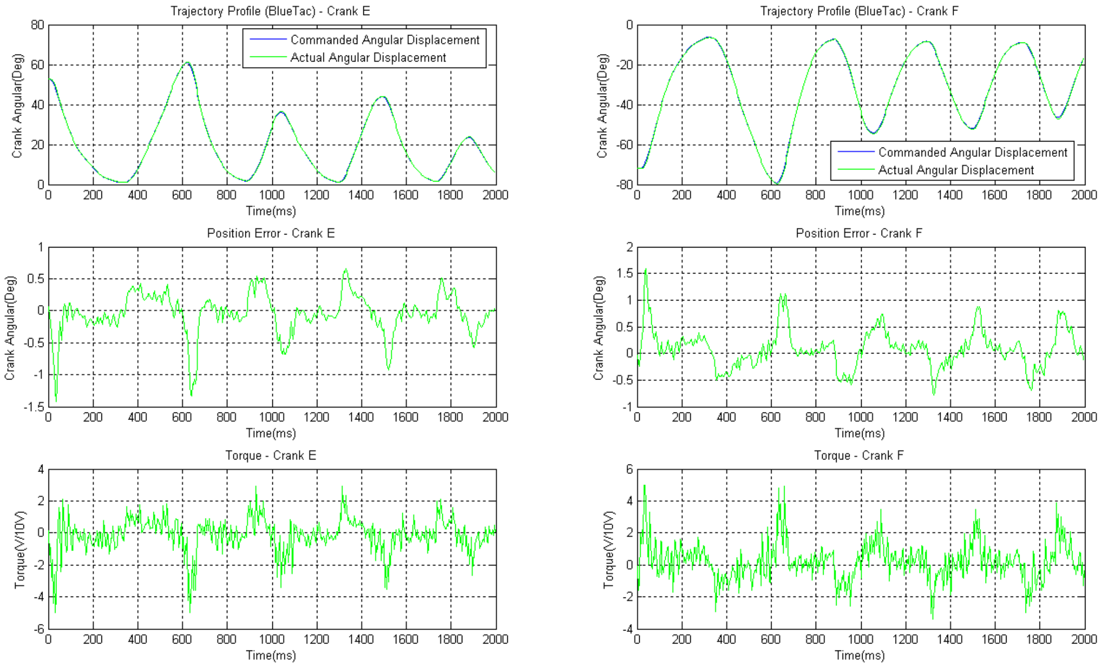
Appendix Figure 2.13 Massey robotic jaw chewing sticky material



Appendix Figure 2.14 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, and Torque Running on Motion 2003 trajectory on 5mm sticky Blue-tac during the first four Chewing Cycles on Crank X and Crank Y



Appendix Figure 2.15 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, and Torque Running on Motion 2003 trajectory on 5mm sticky Blue-tac during the first four Chewing Cycles on Crank Z and Crank W

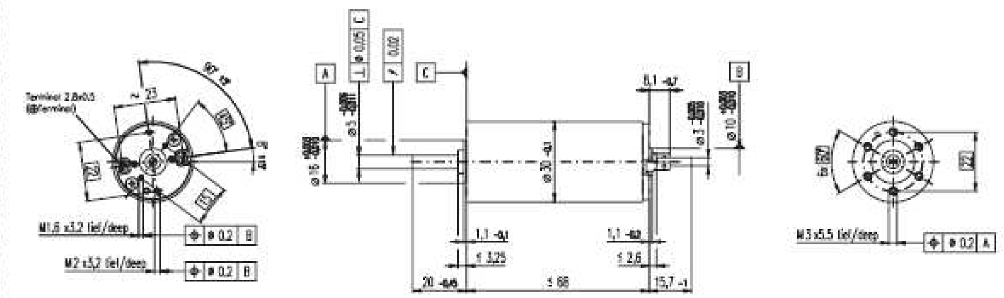


Appendix Figure 2.16 Experimental Results of Commanded and Actual Trajectory Profiles, Position Error, and Torque Running on Motion 2003 trajectory on 5mm sticky Blue-tac during the first four Chewing Cycles on Crank E and Crank F

Appendix 3 Mexon RE 30 Brush DC Motor

maxon DC motor

RE 30 ∅30 mm, Graphite Brushes, 60 Watt



M 1:2

- Stock program
- Standard program
- Special program (on request!)

Order Number

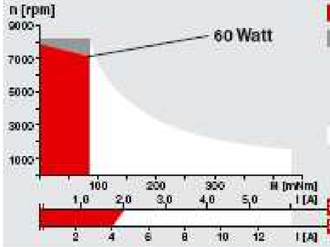
		according to dimensional drawing	310005	310006	310007	310008	310009
		Shaft length 15.7 shortened to 8.7 mm	268193	268213	268214	268215	268216
Motor Data							
1	Assigned power rating	W	60	60	60	60	60
2	Nominal voltage	Volt	12.0	18.0	24.0	36.0	48.0
3	No load speed	rpm	8170	8590	8810	8590	8490
4	Stall torque	mNm	844	991	1020	936	1020
5	Speed / torque gradient	rpm / mNm	9.74	8.71	8.70	9.22	8.33
6	No load current	mA	279	212	154	99.1	73.2
7	Starting current	A	60.5	49.8	39.3	23.5	19
8	Terminal resistance	Ohm	0.198	0.362	0.611	1.53	2.52
9	Max. permissible speed	rpm	8200	8200	8200	8200	8200
10	Max. continuous current	A	4	4	3.81	2.42	1.88
11	Max. continuous torque	mNm	47.5	68.6	86.2	84.1	88.6
12	Max. power output at nominal voltage	W	177	220	233	209	226
13	Max. efficiency	%	87	87	88	88	88
14	Torque constant	mNm / A	13.9	19.9	25.9	38.8	53.8
15	Speed constant	rpm / V	685	479	369	240	178
16	Mechanical time constant	ms	3	3	3	3	3
17	Rotor inertia	gcm²	33.5	35.7	33.3	32.9	34.5
18	Terminal inductance	mH	0.03	0.07	0.12	0.28	0.51
19	Thermal resistance housing-ambient	K / W	6	6	6	6	6
20	Thermal resistance rotor-housing	K / W	1.7	1.7	1.7	1.7	1.7
21	Thermal time constant winding	s	16	17	16	16	17

Specifications

- Axial play 0.05 - 0.15 mm
- Max. ball bearing loads
  axial (dynamic)
  not preloaded 5.6 N
  preloaded 2.4 N
  radial (5 mm from flange) 28 N
  Force for press fits (static) 110 N
  (static, shaft supported) 1200 N
- Radial play ball bearing 0.025 mm
- Ambient temperature range -20 ... +100°C
- Max. rotor temperature +125°C
- Number of commutator segments 13
- Weight of motor 238 g
- 2 pole permanent magnet
- Values listed in the table are nominal.
  For applicable tolerances see page 43.
  For additional details please use the maxon selection program on the enclosed CD-ROM.

⚠ Tolerances may vary from the standard specification.

Operating Range



Comments Details on page 49

Recommended operating range

Continuous operation
In observation of above listed thermal resistances (lines 19 and 20) the maximum permissible rotor temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

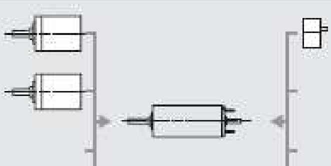
Short term operation
The motor may be briefly overloaded (recurring).

310009 Motor with high resistance winding
310005 Motor with low resistance winding

maxon Modular System Overview on page 17 - 21

Planetary Gearhead
∅32 mm
0.75 - 4.5 Nm
Details page 219

Planetary Gearhead
∅32 mm
1.0 - 6.0 Nm
Details page 220



Encoder MR
256 - 1024 CPT
3 channels
Details page 239

Recommended Electronics:

ADS 50/5	page 259
ADS_E 50/5	260
EPOS 24/5	271
MIP 50	273
Notes	17

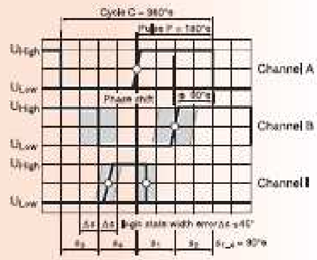
80 maxon DC motor

April 2005 edition / subject to change

Appendix 4 500 Counts Encoder MR

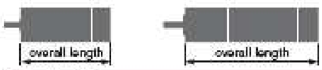
Encoder MR, Type L, 256 - 1024 Counts per turn, 3 Channels, with Line Driver

maxon tacho



■ Stock program
□ Standard program
Special program (on request!)

	Order Number 225783	228452	225785	228456	225787
Type					
Counts per turn	256	500	512	1000	1024
Number of channels	3	3	3	3	3
Max. operating frequency (kHz)	80	200	160	200	320



Combination

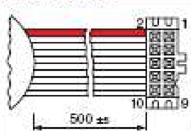
+ Motor	Page	+ Gearhead	Page	+ Brake	Page	Overall length [mm] / ● see: + Gearhead				
RE 30	80					79.4	79.4	79.4	79.4	79.4
RE 30	80	GP 32, 0.75 - 6.0 Nm	219/220			●	●	●	●	●
RE 35	81					82.3	82.3	82.3	82.3	82.3
RE 35	81	GP 32, 0.75 - 6.0 Nm	219/220			●	●	●	●	●
RE 35	81	GP 42, 3 - 15 Nm	224			●	●	●	●	●
RE 36	82					82.6	82.6	82.6	82.6	82.6
RE 36	82	GP 32, 0.75 - 6.0 Nm	219/220			●	●	●	●	●
RE 36	82	GP 32, 0.4 - 2.0 Nm	224			●	●	●	●	●
RE 36	82	GP 42, 3 - 15 Nm	224			●	●	●	●	●
RE 40	83					82.4	82.4	82.4	82.4	82.4
RE 40	83	GP 42, 3 - 15 Nm	224			●	●	●	●	●
RE 40	83	GP 52, 4 - 30 Nm	227			●	●	●	●	●
A-max 32	122/124					72.7	72.7	72.7	72.7	72.7
A-max 32	122/124	GP 32, 0.75 - 6.0 Nm	219/221			●	●	●	●	●
A-max 32	122/124	GS 38, 0.1 - 0.6 Nm	223			●	●	●	●	●
EC-max 40, 70 W	177						73.9		73.9	
EC-max 40, 70 W	177	GP 42, 3 - 15 Nm	225				●		●	
EC-max 40, 120 W	178						103.9		103.9	
EC-max 40, 120 W	178	GP 52, 4 - 30 Nm	228				●		●	

Technical Data

Supply voltage	5 V ± 5 %
Output signal	TTL compatible
Index pulse width (nominal)	90°e
Operating temperature range	-25 ... +85°C
Moment of inertia of code wheel	≤ 1.7 gcm²
Output current per channel	max. 5 mA

Attention: The index signal I is synchronised with channel A or B

Pin Allocation



1 N.C.
2 Vcc
3 GND
4 N.C.
5 Channel $\bar{A}$
6 Channel A
7 Channel $\bar{B}$
8 Channel B
9 Channel $\bar{I}$ (Index)
10 Channel I (Index)

DIN Connector 41651
flat band cable AWG 28

Connection example