SIMPLE RULES TO MODIFY PRE-PLANNED PATHS AND IMPROVE GROSS ROBOT MOTIONS ASSOCIATED WITH PICK & PLACE ASSEMBLY TASKS

Journal:	*Assembly Automation*
Manuscript ID:	AA-09-066.R1
Manuscript Type:	Original Article
Keywords:	Industrial Robotics, Assembly, Assembly < Assembly, Control systems < Assembly, Actuators < Industrial Robotics

SIMPLE RULES TO MODIFY PRE-PLANNED PATHS AND IMPROVE GROSS ROBOT MOTIONS ASSOCIATED WITH PICK & PLACE ASSEMBLY TASKS

Structured Abstract

Purpose: This paper describes real time improvements to the performance and trajectories of robots for which paths had already been planned by some means, automatic or otherwise. The techniques are applied to industrial robots during the gross motions associated with pick and place tasks. Simple rules for path improvement are described.

Design/methodology/approach: The dynamics of the manipulator in closed form Lagrange equations are used to represent the dynamics by a set of second-order coupled non-linear differential equations. The form of these equations is exploited in an attempt to establish some simple rules. Sub optimal paths are improved by considering simple rules developed from the model of the machinery dynamics. By considering the physical limitations of the manipulator, performance was improved by refining pre-calculated paths. Experiments were performed with a prototype robot and an old Puma 560 robot in a laboratory environment. Once the method had been tested successfully then experiments were conducted with a Kuka KR125 Robot at Ford Motor Company. The measured quantities for all the robots were drive currents to the motors (which represented the torques) and the joint angular positions.

Findings: The method of path refinement presented in this paper uses a simplified model of the robot dynamics to successfully improve the gross motions associated with a pick and place task. The advantage of using the input-output form described was that intermediate non-linearities (such as gear friction) and the motor characteristics were directly incorporated into the model.

Research limitations/implications: Even though many of the theoretical problems in manipulator dynamics have been solved, the question of how to best apply the theories to industrial manipulators is still being debated. In the work presented in this paper, information on system dynamics was used to produce simple rules for "path improvement".

Practical implications: Most fast algorithms are for mobile robots and algorithms are scarcer for manipulators with revolute joints (the most popular type of industrial robot). This work presents real time methods that allow the robot to continue working while new global paths are automatically planned and improved as necessary.

Originality/value: Motion planning for manipulators with many degrees of freedom is a complex task and research in this area has been mostly restricted to static environments, offline simulation or virtual environments. This research is applied in real time to industrial robots with revolute joints.

Keywords: *real-time; industrial-robot; pick-and-place; path-improvement; dynamics; gross-motions*.

SIMPLE RULES TO MODIFY PRE-PLANNED PATHS AND IMPROVE GROSS ROBOT MOTIONS ASSOCIATED WITH PICK & PLACE ASSEMBLY TASKS

Response to the comments of the reviewers

I was very pleased to read that the reviewers had recommended publication, but also suggested some revisions to my manuscript. The paper has been revised as suggested and my response to the reviewers' comments is:

Reviewer: 1

- The argument is made a little clearer.
- More explanation of the results has been included.
- Discussion and conclusions have been expanded a little.

Reviewer: 2

- more references have been included.
- The simple rules are stated more clearly.
- More explanation of the results has been included.
- Discussion and conclusions have been expanded a little.

Reviewer: 1 b / 3

- More discussion has been added throughout the paper.
- Neglecting the dynamics of the wrist is mentioned as part of the approximation.
- The improvements are more clearly stated.
- A paragraph has been included to explain that to increase base velocity joint, we need to reduce the mass moment of inertia by bringing the center of gravity close to center of rotation.
- Parameters in pseudo-code have been defined.

Reviewer: 4

- Grammatical errors have been corrected.
- It is made clear that the algorithm does not provide the entire robot trajectory between the initial and final robot positions, the output is an intermediate point that belongs to the new trajectory.
- It is explained that movements are assumed to be gross motions through free space and not fine motions near to objects.
- It is stated that the velocity profiles were trapezoidal.
- Figure 2 has been re-drawn again.
- The quality of the language has been improved and the specific corrections have been made.

SIMPLE RULES TO MODIFY PRE-PLANNED PATHS AND IMPROVE GROSS ROBOT MOTIONS ASSOCIATED WITH PICK & PLACE ASSEMBLY TASKS

1 Introduction.

This paper describes a system which improved the performance and trajectories of robots in real time for which paths had already been planned by some means, automatic or otherwise. The techniques are applied to industrial robots during the gross motions associated with pick and place tasks and are used to develop simple rules for path improvement.

The literature describes various planning algorithms (LaValle 2006) and automatic path planners designed to increase efficiency and productivity in a variety of tasks (Sampaio 2007), such as: production (Sanders 1995a, 2009a, b, c, 2010a), automatic driving (Sanders 2001a; Solea 2007), tele-operation (Sanders, 2008a & 2009c,d, h, 2010b), wheelchair navigation (Goodwin 1997; Stott 2000a+b; Sanders 1999, 2009e), welding (Sanders 2001b), disassembly tasks (Aguinaga, 2007), walking machines (Luk 2005, 2006; Urwin-Wright et al, 2002 & 2003), space (Huntsberger, 2006) and for tasks requiring more than one robot (Deshpande 2007).

These path planners usually required a geometric model of the world (Sanders 1995b) and sometimes that model was constructed from sensor information (Sanders 2008b, 2010c). Others have considered the steering of a robot in real-time according to the most recent sensor readings and different interfaces to program or control robots (Sanders 2009f), for example using pointers (Sanders 2005, 2009i).

Motion planning for manipulators with many degrees of freedom is a complex task (Sanders 2008c), sometimes requiring AI (Bergasa-Suso 2005; Chester 2006, 2007; Hudson 1996, 1997; Sanders 2009g; Stott 1997). Research in this area has been mostly restricted to static environments or virtual environments (Aguinaga 2007; Stott 2000a, Tewkesbury 1999a+b) or offline (Solea 2007). Solea for example considered trajectory planning to produce smooth simulated trajectories with low levels of acceleration and jerk by introducing a velocity planning stage in the trajectory planner. Others have suggested that dynamic models were necessary to produce smooth motion trajectories.

By considering the physical limitations of the manipulator, the performance can be improved by refining pre-calculated paths. The method of path refinement presented in this paper uses a simple model of the robot

dynamics to improve the gross motions associated with a pick and place task.

Most path planning work has tended to require computation time that makes the manipulator wait before carrying out the planned trajectories. Most fast algorithms are for mobile robots and algorithms are scarcer for manipulators with revolute joints, the most popular type of industrial robot. The methods presented in this paper allow the robot to continue working and new global paths are automatically planned and improved as necessary.

The method provides solutions to that problem which consider the geometric constraints of the obstacles and the restrictions of the world model. In this work, the sub optimal paths that are created are improved by considering simple rules developed from a model of the machinery dynamics.

Two major approaches in terms of the formulation of robot dynamics equations are the Newton-Euler method and the Lagrangian formulation. The Newton-Euler formulation has been employed to determine the inertial parameters of robot links, and these were then used in a recursive dynamics computation. Other authors adopted a hybrid procedure combining the Newton-Euler and Lagrange formulation of the dynamics to estimate the inertial parameters of the links.

Even though many of the theoretical problems in manipulator dynamics have been solved, the question of how to best apply the theories to industrial manipulators is still being debated. In the work presented in this paper, information on system dynamics was used to produce a set of simple rules for an automatic path improvement system. Closed form Lagrange equations were selected to represent the dynamics by a set of second-order coupled non-linear differential equations. The form of these equations was exploited in an attempt to establish a set of simple rules.

Experiments were performed with a prototype robot and an old Puma 560 robot in a laboratory environment. The measured quantities for all the robots were drive currents to the motors (which represented the torques) and joint angular positions. The advantage of using this input-output form was that intermediate non-linearities (such as gear friction) and the motor characteristics were directly incorporated into the model.

Once the method had been tested successfully then experiments were conducted with a modular aluminum

Kuka KR125 Robot at Ford Motor Company (see figure 1). The Kuka KR125 robots were floor or ceiling mounted with handling loads of up to 125 kg, maximum reach of over 2.5 metres and joint rotational speeds of up to 150 degrees per second.

The rules developed during the work do not provide a complete new robot trajectory between the initial and final robot positions (that is left to the particular controller and the particular robot). The product from the rules is an intermediate via-point that the robot needs to move through in order to increase the speed of the movement. That was fed into the controller for each robot. In each case an improvement was made even though each controller used a different method of calculating specific trajectories.

- Figure 1 here -

In the next section the Lagrange formulation for these robots (with three revolute joints and two major links) is briefly outlined. In sections 3 and 4 the experimental identification procedure is described and in section 5 the results of this procedure are presented. The paper concludes with some discussion and conclusions.

2. The Lagrangian Formulation for a robot with three revolute joints

The Lagrangian equation in terms of the Lagrangian coordinates q is given by:

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial (dq_i/dt)} - \frac{\partial L}{\partial q_i}$$

where,

L	=	The Lagrangian function.
q_i	=	The coordinate of the i^{th} element used to express the kinetic and potential energies.
τ_i	=	The torque.

The relationships between the torques and the angular positions, velocities and accelerations of the links were obtained by considering the potential and kinetic energies. The Lagrangian L is defined as the difference between the kinetic and potential energy given by: $L = \mathbf{K} - \mathbf{P}$ where **K** is the total kinetic energy and **P** is the total potential energy. Using the expressions for **K** and **P** in terms of manipulator parameters, the equations for the dynamics of the three main links were obtained. An example for τ_i is shown:

$$\tau_i = \sum_{j=1}^{N} J_{ij} d^2\theta_j/dt + \sum_{j=1}^{N} \sum_{k=1}^{N} H_{ijk}(d\theta_j/dt)(d\theta_k/dt) + G_i$$

The revolute robots were assumed to consist of two main movable links, L_1 and L_2 (of masses m_1 and m_2) which could be rotated through angles θ_2 and θ_3, as shown in figure 2. The robot base L_0, with mass m_0 could rotate through θ_1. To simplify the model, only the three main revolute joints of the arm are considered during the development of the dynamic model. The dynamics of the wrist (and wrist joints) are ignored and the mass of the payload is included in m2. This approximation could cause errors, especially if the payload was heavy and an irregular shape. Those cases were not considered in this initial work.

- Figure 2 here -

The expressions of the kinetic energy of the links do not consider the exact inertia matrix. Instead, each link is considered as a lumped mass, without a moment of inertia. That approximation simplifies the dynamic equations and is shown to be acceptable during gross motions.

To determine the total kinetic and potential energy for the robot, each link was considered in turn to find the kinetic energy and potential energy equations. The cartesian coordinates of the assumed centre of mass (shown in figure 2) were considered in terms of the joint angles. For link L_1 this gave:

$$X_1 = L_1/2 \cos\theta_1\cos\theta_2$$

$$Y_1 = L_1/2 \sin\theta_1\cos\theta_2$$

$$Z_1 = L_0 + L_1/2 \sin\theta_2$$

Taking derivatives of the equations with respect to time gave:

$$dX_1/dt = -L_1/2\, d\theta_1/dt \sin\theta_1\cos\theta_2 - L_1/2\, d\theta_2/dt \cos\theta_1\sin\theta_2$$

$$dY_1/dt = L_1/2\, d\theta_1/dt \cos\theta_1\cos\theta_2 - L_1/2\, d\theta_2/dt \sin\theta_1\cos\theta_2$$

$$dZ_1/dt = L_1/2\, d\theta_2/dt \cos\theta_2$$

Considering V_1^2 where $V_1^2 = (dX_1/dt)^2 + (dY_1/dt)^2 + (dZ_1/dt)^2$ and using trigonometric identities to reduce the solution, the square of the velocity vector was:

$$V_1^2 = (L_1/2)^2 (d\theta_2/dt)^2 + (L_1/2)^2 (d\theta_1/dt)^2 \cos^2\theta_2$$

The kinetic energy term and the potential energy term of link L_1 were thus assumed to be:

$$K_1 = 1/2\ m_1 V_1^2 = 1/2\ m_1 (L_1/2)^2 \{(d\theta_2/dt)^2 + (d\theta_1/dt)^2 \cos^2\theta_2\}$$

$$P_1 = m_1 g L_0 + m_1 g (L_1/2) \sin\theta_2$$

where g = gravitational acceleration.

Similar kinetic energy and potential energy terms could be found for the other links and for the joints.

Having found the kinetic and potential energies for the three links / joints, then a Lagrangian of the robot;

$$\mathbf{L} = K_0 + K_1 + K_2 - (P_0 + P_1 + P_2)$$

was calculated. The partial derivatives $\partial L/\partial\theta_1$, $\partial L/\partial\theta_2$, $\partial L/\partial\theta_3$, $\partial L/\partial(d\theta_1/dt)$, $\partial L/\partial(d\theta_2/dt)$ and $\partial L/\partial(d\theta_3/dt)$ were then established so that the Lagrangian equation in terms of the robot joints;

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial(d\theta_i/dt)} - \frac{\partial L}{\partial\theta_i}$$

could be applied for each of the links θ_1, θ_2 and θ_3 in turn. The first dynamics equation was thus:

$$\begin{aligned}\tau_1 = {} & (d^2\theta_1/dt^2).\{I + m_1(L_1/2)^2\sin^2\theta_2 + m_2(L_1/2)\sin\theta_2 + m_2(L_2/2)\sin\theta_3\} \\ & + d\theta_1/dt\ d\theta_2/dt\ \ 2.\{m_1(L_1/2)\cos\theta_2 - m_2 L_1^2 \cos\theta_2 \sin\theta_2 + m_2 L_1 (L_2/2)\cos\theta_2 \cos\theta_3\} \\ & + d\theta_1/dt\ d\theta_3/dt\ \ 2.\{m_2(L_2/2)^2\cos\theta_2\cos\theta_3 + m_2 L_1 (L_2/2)\sin\theta_2\sin\theta_3\}\end{aligned}$$

This equation and the other torque equations had several components. They were:

- Effective inertias (and coupling inertias).
- Coriolis and centripetal coefficients.
- Gravity loadings.

so the equation for τ_1 could be expressed in the form:

$$\tau_1 = D_{1I}\, d^2\theta_1/dt^2 + D_{12}\, d\theta_1/dt\ d\theta_2/dt + D_{13}\, d\theta_1/dt\ d\theta_3/dt + D_{1g}$$

where:

D_{1I} =The effective moment of inertia about the Z1 axis

$D_{12}\ d\theta_1/dt\ d\theta_2/dt$= Coriolis torque acting at joint θ_1 due to the velocities of the base θ_1 and shoulder θ_2.

$D_{13}\ d\theta_1/dt\ d\theta_3/dt$= Coriolis torque acting at joint θ_1 due to the velocities of the base θ_1 and the elbow θ_3.

D_{1g} = The gravitational torque.

The second dynamic equation in coefficient form was:

$$\tau_2 = D_{2I}\, d^2\theta_2/dt^2 + D_{22}\, d\theta_2/dt\, d\theta_3/dt + D_{2cI}\, d^2\theta_3/dt^2 + D_{24}\, (d\theta_3/dt)^2 + D_{25}\, (d\theta_1/dt)^2 + D_{26}$$

where

D_{2I}	=	The effective moment of inertia about the Z_2 axis
$D_{22}\, d\theta_2/dt\, d\theta_3/dt$	=	Coriolis torque due to velocities of the shoulder and elbow.
D_{2cI}	=	Coupling inertia term between links L_1 and L_2.
$D_{24}\, (d\theta_3/dt)^2$	=	Centripetal torque at θ_2 due to the velocity of θ_3.
$D_{25}\, (d\theta_1/dt)^2$	=	Centripetal torque at θ_2 due to the velocity of θ_1.
D_{2g}	=	The gravitational torque.

The third dynamics equation in the coefficient form was,

$$\tau_3 = D_{3I} d^2\theta_3/dt^2 + D_{3cI} d^2\theta_2/dt^2 + D_{33}(d\theta_1/dt)^2 + D_{34}(d\theta_2/dt)^2 + D_{3g}$$

where

D_{3I}	=	The effective inertia term at joint 3.
D_{3cI}	=	The coupling inertia term between links L_1 and L_2.
$D_{33}(d\theta_1/dt)^2$	=	Centripetal torque acting at θ_3 due to velocity $d\theta_1/dt$.
$D_{34}(d\theta_2/dt)^2$	=	Centripetal torque acting at θ_3 due to velocity $d\theta_2/dt$.
D_{3g}	=	The gravitational torque.

The expressions for the dynamics consisted of variables, which were functions of sines and cosines of the joint positions, and constants which depended on the manipulator link parameters such as link mass, centre of mass, and radii of gyrations. Measurements could have been taken of the links to obtain the dimensions of centres of mass and radius of gyration for each link. The link masses could have been calculated from the measurements and the density of the materials and then the dynamics constants calculated. That process would have been tedious and the measurement of parameters such as location of centre of masses and exact shapes would have been susceptible to errors. An alternative approach was to obtain the constants by actually running the manipulator. The approach exploited direct input-output measurements during actual motion and then employed the results (presented in section 4) to produce simple rules for robot path improvement.

3 Formulation of the Experiments.

There is a disparity in the roles that different terms play in the dynamics equations. The importance of the velocity dependent terms has been controversial and there are situations where centripetal and Coriolis forces dominate inertial forces. That idea can be extended to eliminate less significant dynamics terms and expressions within terms when using the equations for manipulator control. The manipulator joints experience high velocities during gross motions when controller accuracy is not critical. During fine motions when the control accuracy is important, joints move with high accelerations and low velocities so that the gravitational and inertial forces become dominant and velocity dependent forces are not so important.

The inertial terms were assumed to be less significant as the work described here was concerned with the gross motions associated with path planning and not the fine motions associated with approach paths or fine detailed tasks. The inertial and coupling inertia terms were excluded to give the following simplified equations:

$$\tau_1 = d\theta_1/dt\ d\theta_2/dt\ 2\{m_1(L_1/2)\cos\theta_2 - m_2L_1^2\cos\theta_2\sin\Theta_2 + m_2L_1(L_2/2)\cos\theta_2\cos\theta_3\} + d\theta_1/dt\ d\theta_3/dt\ 2\{m_2(L_2/2)^2\cos\theta_2\cos\theta_3 + m_2L_1(L_2/2)\sin\theta_2\sin\theta_3\}$$

$$\tau_2 = d\theta_2/dt\ d\theta_3/dt\ 2m_2L_1(L_2/2)\cos(\theta_2+\Theta_3-\pi) - (d\theta_3/dt)^2\ 2\{m_2L_1(L_2/2)\cos(\theta_2+\Theta_3-\pi)\} - (d\theta_1/dt)^2\ \{m_1(L_1/2)^2\cos\theta_2\sin\theta_2 + m_2L_1^2\cos\theta_2\sin\theta_2 + m_2L_2(L_1/2)\cos\theta_2\cos\theta_3\} - m_1g(L_1/2)\cos\theta_2 - m_2gL_1\cos\Theta_2 - m_2g(L_2/2)\cos(\theta_2+\theta_3)$$

$$\tau_3 = (d\theta_1/dt)^2\{m_2(L_2/2)^2\sin\theta_3 + m_2L_1(L_2/2)\sin\theta_2\sin\theta_3\} + (d\theta_2/dt)^2\{m_2L_1(L_2/2)\cos(\theta_2+\Theta_3-\pi)\} - m_2g(L_2/2)\cos(\theta_2+\theta_3-\pi)$$

so that:

$$D_{12} = 2\{m_1(L_1/2)\cos\theta_2 - m_2L_1^2\cos\theta_2\sin\Theta_2 + m_2L_1(L_2/2)\cos\theta_2\cos\theta_3\}$$

$$D_{13} = 2\{m_2(L_2/2)^2\cos\theta_2\cos\theta_3 + m_2L_1(L_2/2)\sin\theta_2\sin\theta_3\}$$

$$D_{1g} = 0$$

$$D_{22} = 2m_2L_1(L_2/2)\cos(\theta_2+\Theta_3-\pi)$$

$$D_{24} = 2m_2L_1(L_2/2)\cos(\theta_2+\Theta_3-\pi)$$

$$D_{25} = \cos\theta_2\sin\theta_2 + m_2L_1^2\cos\theta_2\sin\theta_2 + m_2L_2(L_1/2)\cos\theta_2\cos\theta_3\}$$

$D_{2g} = m_1 g(L_1/2)\cos\theta_2 + m_2 g L_1 \cos\theta_2 + m_2 g(L_2/2)\cos(\theta_2 + \theta_3)$

$D_{33} = m_1(L_2/2)^2 \sin\theta_3 + m_2 L_1(L_2/2)\sin\theta_2 \sin\theta_3$

$D_{34} = m_2 L_1(L_2/2)\cos(\theta_2 + \theta_3 - \pi)$

$D_{3g} = m_2 g(L_2/2)\cos(\theta_2 + \theta_3 - \pi)$

To determine the dynamics constants experimentally, it was important to know the joint torques of all the joints at any time instant. This was achieved by monitoring joint motor currents. The output torque was approximately linear to the motor current except for an offset at the origin and a diverging curvature on both curves, which corresponded to the two directions of motion. The offset at the origin was caused by static friction that the joint must overcome before any motion at the joint could result. The diverging characteristic is explained by the load dependent nature of joint friction which increases non linearly with an increase in load. In this work the functional relationship between joint torque and current was assumed to be a linear relationship so that the process of computing torque from current was a simple linear mapping and in practice the torque constants provided by the manufacturer were used in converting currents to torques.

Summary: *The position and velocity were measured for various inputs. The joint torques necessary to generate motion were observed while the manipulator moved along trajectories with known motion parameters. Since the joint torque was directly related to the constants by the dynamics equations and the intermediate joint positions were known, a set of equations linear to the constants could be established from the readings of joint current and joint position and used to solve for the constants in the equations of the dynamics. This method accounted for the non linearity of the manipulator.*

4 The Experimental Method.

The procedures were initially applied to a prototype robot base and arm, and then to the base, shoulder and elbow joint of a Puma 560 robot, both with end effector loads of one kilogram. Once the method had been tested then the method was applied to a Kuka KR125 Robot. The angular positions of the joints were fed back from encoders mounted on the robots. The encoder outputs were converted to a count representing position. Software was developed in C and Quick-Basic and a series of three tests were conducted:

(i) Static Tests.
(ii) Single Joint Motion Tests.
(iii) Multiple Joint Motion Tests.

(i) Static Tests: To obtain the gravitational constants from the knowledge of joint torques, the effects due to other dynamics terms were eliminated so that the joint torque became a function of gravity loading. Only the joint of interest was moved and the other joints were stationary. Under these test conditions, velocity and acceleration dependent terms disappeared as the other joints were stationary and the motion of the non-stationary joint was very slow. With the other joints locked in a particular configuration, the torque or force required to move each joint was measured. The torques required to overcome gravity in each configuration were estimated by moving the manipulator to a desired configuration and then incrementing the output through D/A converters one bit at a time until motion was detected. The result of these measurements was a table of gravitational torques (D_{ig} for link i) for varying θ_1, θ_2 and θ_3.

If τ_{pi} was the torque in one direction and τ_{mi} in the other, and F_{is} represented static friction for joint i, the following equations were obtained: $\tau_{pi} = D_{ig} + F_{is}$ and $\tau_{mi} = - D_{ig} + F_{is}$ so that:

$$D_{ig} = (\tau_{pi} + \tau_{mi})/2$$

This procedure was repeated for each ten degree increment of each joint angle that occurred as a basis function for D_{ig}. Two constants, **A** and **B** were determined for each robot to satisfy $\mathbf{A} = m_2 g L_2/2$ and $\mathbf{B} = gL_1(m_2+m_1/2)$ so that:

$D_{3g} = \mathbf{A} \cos(\theta_2+\theta_3-\pi) = -\mathbf{A} \cos(\theta_2+\theta_3)$

$D_{2g} = \mathbf{B} \cos(\theta_2) - D_{3g}$

$D_{1g} = \mathbf{0}$

(ii) Single Joint Motion Tests: These were achieved by driving the motors at a constant velocity. Practically, this was achieved by outputting a step velocity demand and running the joints through 10 degrees before taking any readings to avoid the inertial effects. Only one joint was moved at a time so that the governing equation was:

$$\tau_i = b_i(d\theta_i/dt) + F_i + D_{ig}$$

With gravitational compensation this could be reduced to $\tau_i = b_i(d\theta_i/dt) + F_i$ where F_i is the Coulomb

friction and b_i is the overall viscous damping coefficient, so that the steady-state velocity was:

$$(d\theta_i/dt)_{ss} = \frac{\tau_i - F_i}{b_i}$$

The current required to maintain a constant velocity, and the velocity of the base joint for a constant demand output, were recorded for various configurations.

(iii) Multiple Joint Motion Tests: To estimate the coupling terms in the dynamic equations, motions requiring joints to move simultaneously were applied. The same input was applied to a joint, i, first with a joint, j, stationary and then with joint j also in motion. The response in the two cases with gravitational compensation was assumed as:

with coupling $\tau_{ic} = H_{ij}(d\theta_{ic}/dt)(d\theta_{jc}/dt) + b_i(d\theta_{ic}/dt) + F_i$

and without coupling $\tau_i = b_i(d\theta_i/dt) + F_i$

so that $H_{ij}\theta_{ic}\theta_{jc} = \tau_{ic} - \tau_i$

where the subscript c indicated the presence of coupling. The measured motion responses together with previously computed values of b_i and F_i were to be used to evaluate the coupling coefficients in the above equations. In the case of this work the practical evaluation was not necessary.

5 Results from the Static and Motion Tests.

(i) Static Tests: The shoulder currents required to overcome gravity and the static friction of the shoulder joint for various configurations of the elbow joint were recorded with τ_{pi} (the torque in one direction) and τ_{mi} (the torque in the other direction). As discussed in section 4, F_{is}, the static friction for joint i could be removed as $\tau_{pi} = D_{ig} + F_{is}$ and $\tau_{mi} = -D_{ig} + F_{is}$ so that: $D_{ig} = (\tau_{pi} + \tau_{mi})/2$. The remaining D_{2g} is shown in the graphs of figure 3 with the Elbow angle marked.

- Figure 3 here -

(ii) Single Joint Motion Tests: The graphs in figure 4 show the current required to maintain a constant velocity for each joint for different configurations.

- Figure 4 here -

(iii) Multiple Joint Motion Tests: The noise in the system was greater than any effects due to coupling between joints.

6 Discussion of the results.

(i) Static Tests: The equations for the manipulator dynamics developed in the paper suggested that the maximum gravitational effect would be felt by joints θ_2 and θ_3 at $\theta_2 = 0°$, $\theta_3 = 180°$ and the minimum effect at $\theta_2 = 90°$, $\theta_3 = 180°$ as the equations for the static case were:

$$\tau_2 = \mathbf{B}\cos(\theta_2) + \mathbf{A}\cos(\theta_2 + \theta_3) + \mathbf{F}_{is}$$

$$\tau_3 = -\mathbf{A}\cos(\theta_2 + \theta_3) - \mathbf{F}_{is}.$$

This was confirmed.

(ii) Single Joint Motion Tests: Considering the equation from section 4.ii:

$$(d\theta_i/dt)_{ss} = \frac{\tau_i - F_i}{b_i}$$

joints θ_2 and θ_3 performed as expected as shown in figure 4, in that they were not affected by the configuration of the other joints. Figure 4 also shows that the base joint had a steady state velocity which was partly dependent on joint angles θ_2 and θ_3. The velocity of θ_1 was greater as the mass moved towards the Origin.

(iii) Multiple Joint Motion Tests: There were no measurable velocity effects due to coupling effects between the joints. Although results are not recorded here, there was an inertia coupling between joints θ_2

and θ_3. This could be considered in future work.

7. Development of Simple Rules for Path Improvement.

Considering the results of the position and velocity tests, only two effects dominated the dynamics of the two robots. They were the varying effect of θ_2 and θ_3 on the base joint, and the gravity effect of θ_3 upon θ_2. These suggested two simple rules by which the robot path could be improved.

1. The base velocity was related to the controller demand input, the robot configuration, joint limitations and payload. The first rule was:

-To increase the base velocity the arm should attempt to move the centre of mass towards the centre of rotation by moving θ_2 towards 90° and θ_3 towards 90°.

2. Gravity loading was related to the configuration of joints θ_2 and θ_3. The second rule was:

-To reduce the effects of gravity loading, the arm should move θ_3 towards 90° during motions of θ_2.

8 Results from applying the simple rules.

Once these rules had been established, motion tests were undertaken for various paths during pick and place operations. The times for the revised paths were recorded. The tests were repeated with all three robots and typical results were:

To test for the reduction in coulomb friction, each robot arm was initially moved from [140°,0°,180°] to [-140°,0°,180°] via [0°,90°,180°]. For the prototype robot the movement took an average of 4.49 seconds. When the test path was modified to use the same START and GOAL, but to move through a via-point at [0°, 90°, 90°] the robot took an average of 4.16 seconds; a saving of 0.33 seconds. Savings for the other two robots were .41 seconds for the Puma 560 robot in the laboratory environment and 0.52 seconds for the Kuka KR125 Robot.

To test for the reduction in gravity loading, similar tests were conducted for the shoulder and elbow, with the waist still (at 0°). The shoulder was moved from -10° to 90° with the elbow at 180°; this gave an average time of 1.98 seconds for the prototype robot . When the path was modified so that the elbow moved in

towards 90° until the shoulder reached 50° and then moved out to 180°, an average time of 1.75 seconds was recorded; a saving of 0.23 seconds. Savings for the other two robots were .21 seconds for the Puma 560 robot in the laboratory environment and 0.32 seconds for the Kuka KR125 Robot.

The adaption rules were included in an automatic path planning and adaption system and the two sets of pseudo-code are shown below:

```
ShoulderDiff = Shoulder(n+1) - Shoulder(n)
NewShoulder(n) = Shoulder(n) + ShoulderDiff/2
ElbowDiff = Elbow(n+1) - Elbow(n)
IF SGN(ElbowDiff) = 1 THEN
        NewElbow(n) = Elbow(n) - ShoulderDiff/6
ELSE
        NewElbow(n) = Elbow(n+1) - ShoulderDiff/6
END IF

BaseDiff = base(n+1) - Base(n)
NewBase(n) = Base(n) + BaseDiff/2
ShoulderDiff(n) = Shoulder(n+1) - Shoulder(n)
ElbowDiff = ElbowDiff(n) - ElbowDiff(n+1)
IF BaseDiff <>   0 THEN
        IF (Shoulder(n+1) > 0) AND SGN(ShoulderDiff) = 1 THEN
                NewShoulder(n) = Shoulder(n) - BaseDiff/2
                IF SGN(ElbowDiff) = 1 THEN
                        NewElbow(n) = Elbow(n) + BaseDiff/4
                ELSE
                        NewElbow(n) = Elbow(n+1) + BaseDiff/4
                END IF
        ELSE IF (Shoulder(n+1) > 0) AND SGN(ShoulderDiff) = 0 THEN
                NewShoulder(n) = Shoulder(n+1) - BaseDiff/2
                IF SGN(ElbowDiff) = 1 THEN
                        NewElbow(n) = Elbow(n) + BaseDiff/4
                ELSE
                        NewElbow(n) = Elbow(n+1) + BaseDiff/4
                END IF
        ELSE IF (Shoulder(n+1) < 0) AND SGN(ShoulderDiff) = 1 THEN
                NewShoulder(n) = Shoulder(n+1) + BaseDiff/2
                IF SGN(ElbowDiff) = 1 THEN
                        NewElbow(n) = Elbow(n) + BaseDiff/4
                ELSE
                        NewElbow(n) = Elbow(n+1) + BaseDiff/4
                END IF
        ELSE IF (Shoulder(n+1) < 0) AND SGN(ShoulderDiff) = 0 THEN
                NewShoulder(n) = Shoulder(n) + BaseDiff/2
                IF SGN(ElbowDiff) = 1 THEN
                        NewElbow(n) = Elbow(n) + BaseDiff/4
                ELSE
                        NewElbow(n) = Elbow(n+1) + BaseDiff/4
                END IF
        END IF
END IF
IF NewElbow(n) > 180 THEN NewElbow(n) = 180
IF NewShoulder(n) < -30 THEN NewShoulder(n) = -30
```

Where:

ShoulderDiff = Movement of the shoulder joint.
NewShoulder(n) = New value of the shoulder joint.
ElbowDiff = Movement of the shoulder joint.
NewElbow(n) = New value for the elbow joint.
BaseDiff = Movement of the base joint.
NewBase(n) = New value for the base joint.

An example of initial paths and their adapted paths after applying the rules developed in Section 7 is shown below:

```
 70 ,  0 , 100        70 ,  0 , 100
150 ,-30 , 100       110 , 10 , 120
                     150 ,-30 , 100

 90 , 20 , 150        90 , 20 , 150
 30 , 75 , 125        60 , 45 , 140
                      30 , 75 , 125
```

Two simple example paths are shown on the left and the result of applying the rules are shown on the right. In both cases a via-point was generated which moved the shoulder and elbow through configurations which tended to move the centre of mass closer to the centre of rotation during motions of the robot base.

As an example, the simple rules were applied to a Puma 560 robot. The robot arm was initially moved from [140°,0°,180°] to [-140°,0°,180°] via [0°,90°,180°]. The movement took an average of 3.34 seconds. When the test path was modified to use the same START and GOAL, but to move through a via-point at [0°, 90°, 90°] the robot took an average of 3.05 seconds; a saving of 0.29 seconds. Similar tests were conducted for the shoulder and elbow, with the waist still (at 0°). The shoulder was moved from -10° to 90° with the elbow at 180°; this gave an average time of 1.45 seconds. When the path was modified so that the elbow moved in towards 90° until the shoulder reached 50° then moved out to 180°, an average time of 1.34 seconds was recorded. This represented a saving of 0.11 seconds.

9 Discussion and Conclusions.

A successful method of path improvement has been presented. A method for calculating the manipulator dynamics model for a robot with revolute joints based on the Lagrange formulation was presented. The model was refined and confirmed through a sequence of static tests, single joint and multiple joint motion tests. The model included the effects of gear transmission and friction. Simple rules for path improvement

were developed from the simplified model. These rules were applied to adapt the paths of three revolute robots during various gross motions associated with pick and place assembly tasks. The method reprogrammed a path during the first sequence of a set of repeated paths by adding via-points which moved the manipulator through more profitable configurations. The rules developed were specific to the revolute robots tested during this work but the new concept of using the manipulator dynamics to produce simple path reprogramming rules can be applied to any open kinematic chain.

The results suggested a maximum improvement of ≈10%. In practice after considering 50 random paths for each robot, the average improvement was only 3% for the prototype robot, 2.5% for the Puma 560 robot and 3.4% for the Kuka KR125 Robot. This is a satisfactory improvement but the selection of the via-points could be improved in future work.

All the research was conducted with a standard load of 2 Kg and with the robots mounted on the floor. Future work could investigate different mounting arrangements and motions with varying loads. The algorithms do not provide the robot trajectory between the initial and final robot positions, the output is an intermediate point that belongs to the new trajectory. Because the gross movements are assumed to be through free space (as apposed to fine motions close to obstructions), the specific path planning for the two parts of the new trajectory are not considered.

In order to increase the velocity of the base joints, the mass moment of inertia of the arm is brought closer to the center of rotation so that when applying the same current (torque) to each actuator, then higher velocities can be achieved. By bringing θ_3 to 90 degree as θ_2 is moved, the lever of torque created by gravity on θ_2 is reduced, so that θ_2 and θ_1 can move faster. The time saved is small but over a long series of repeated operations even that small saving could be significant.

Ongoing research is investigating the automation of the whole process so that a robot can move around in free space with a specific load or loads and model itself in order to create some simple rules for itself. These rules would then be specific to that robot and load(s) and the particular configuration (wall mounted, floor mounted or ceiling mounted) but could reduce the time taken for repeated movements in free space.

References

Aguinaga I, Borro D, Matey L, (2007). Path-planning techniques for the simulation of disassembly tasks. Assembly Automation 27 (3), pp 207-214.

Bergasa-Suso, J; Sanders, DA; Tewkesbury, GE (2005). Intelligent browser-based systems to assist Internet users. IEEE Transactions (E) 48 (4) pp 580-585.

Chester, S; Tewkesbury, G; Sanders, D, et al (2006). New electronic multi-media assessment system. WEBIST 2006: Proceedings of the Second International Conference on Web Information Systems and Technologies Pages: 320-324.

Chester, S; Tewkesbury, G; Sanders, D, et al. (2007). New electronic multi-media assessment system. Web Information Systems and Technologies Volume: 1 Pages: 414-420 Published: 2007

Deshpande A, and Luntz J (2007). Behaviors for physical cooperation between robots for mobility improvement. AUTONOMOUS ROBOTS 23 (4), pp 259-274.

Goodwin MJ, Sanders DA amd Poland GA (1997). Navigational assistance for disabled wheelchair-users Jnl of Systems Architecture 43 (1-5), pp 73-79.

Helgason RV, Kennington JL and Lewis KR (2001). Cruise missile mission planning: A heuristic algorithm for automatic path generation. Jnl of Heuristics 7 (5), pp 473-494.

Hudson AD, Sanders DA and Tewkesbury GE (1996). Simulation of a high recirculation airlift reactor for steady-state operation. Water science and technology Volume: 34 Issue: 5-6 Pages: 59-66.

Hudson, AD; Sanders, DA; Golding, H, et al. (1997). Aspects of an expert design system for the wastewater treatment industry. Jrnl of systems architecture 43 (1-5) pp 59-65.

Huntsberger TL and Sengupta A (2006). Game theory basis for control of long-lived lunar/planetary surface robots. Autonomous Robots 20 (2), pp 85-95.

LaValle (2006). Planning Algorithms. Cambridge University Press

Luk BL, Cooke DS, Galt S, Collie AA and Chen S(2005), “Intelligent legged climbing service robot for remote maintenance applications in hazardous environments”, Journal of Robotics and Autonomous Systems, Vol. 53/2, pp. 142-52.

Luk BL, Collie AA, Cooke DS and Chen S. (2006). Walking and climbing service robots for safety inspection of nuclear reactor pressure vessels. Measurement & Control 39 (2), pp 43-47.

Mediavilla M, Fraile JC, Gonzalez T and Galindo IJ (2003). "Selection of strategies for collision-free motion in multi-manipulator systems,"J. Intell. Robot Syst 38, PP 85-104.

Sampaio JHB (2007) Planning 3D well trajectories using spline-in-tension functions. Jnl Energy Resources Technology-Trans of ASME 129 (4), pp 289-299.

Sanders DA (1995a). The modification of pre-planned manipulator paths to improve the gross motions associated with the pick and place task. Robotica 13, pp 77-85 Part 1.

Sanders DA (1995b). Real time geometric modeling using models in an actuator space and Cartesian space. Jnl of Robotic Systems 12 (1), pp 19-28.

Sanders DA & Stott IJ (1999). A new prototype intelligent mobility system to assist powered wheelchair users. Ind Robot 26 (6), pp 466-475.

Sanders, D (1999b) Perception in robotics. Ind Robot 26 (2), pp 90-92.

Sanders DA and Baldwin A (2001a). X-by-wire technology. Total vehicle technology: Challenging current thinking, pp 3-12.

Sanders DA and Rasol Z (2001b) An automatic system for simple spot welding tasks Total vehicle technology pp 263-272.

Sanders DA, Urwin-Wright SD, Tewkesbury GE, et al (2005). Pointer device for thin-film transistor and cathode ray tube computer screens. ELECTRONICS LETTERS, Volume: 41, Issue: 16, pp 894-896.

Sanders, D (2008). Controlling the direction of "walkie" type forklifts & pallet jacks. Assem' Automation 28 (4), pp 317-324.

Sanders, D (2008b) Environmental sensors and networks of sensors. Sensor Review 28 (4) pp 273-274.

Sanders, D (2008c) Progress in machine intelligence. Ind Rob – an Int jrnl 35 (6) pp 485-487.

Sanders DA (2009a). Recognizing shipbuilding parts using artificial neural networks and Fourier descriptors. Proc of Institution of Mechanical Engineers Part B – Journal of Engineering Manufacture, Volume: 223, Issue: 3, Pages: 337-342.

Sanders DA, Lambert G and Pevy L (2009b) Pre-locating corners in images in order to improve the extraction of Fourier descriptors and subsequent recognition of shipbuilding parts. Proc of Institution of Mechanical Engineers Part B – J Engineering Manufacture, Volume: 223 Issue: 9 Pages: 1217-1223.

Sanders DA, Yong Chai Tan, Rogers I and Tewkesbury GE (2009c). An expert system for automatic design-for-assembly. Assembly Automation, Volume 29, Issue 4, Pages: 378-388. .

Sanders D (2009d) Comparing speed to complete progressively more difficult mobile robot paths between human tele-operators and humans with sensor-systems to assist. Assembly Automation 29 (3) pp 230-248.

Sanders DA, Langner and Tewkesbury (2009e) Improving wheelchair-driving using a sensor system to control wheelchair-veer and variable-switches as an alternative to digital-switches or joysticks. Industrial Robot IR-08-714.

Sanders D (2009f). Ambient-intelligence, rapid-prototyping and where real people might fit into factories of the future. Assembly Automation 29 (3) pp 205-208.

Sanders DA (2009g). Introducing AI into MEMS can lead us to brain-computer interfaces and super-human intelligence (invited viewpoint review paper). Assembly Automation 29 (4), pages: 309-312.

Sanders DA (2009h). Analysis of the effects of time delay on the tele-operation of a mobile robot in various modes of operation. Industrial Robot, Volume: 36, Issue: 6, pages: 570-584.

Sanders, DA; Tewkesbury, GE (2009i). A pointer device for TFT display screens that determines position by detecting colours on the display using a colour sensor and an Artificial Neural Network. DISPLAYS, Volume: 30, Issue: 2, pp 84-96.

Sanders DA, Lambert, Graham-Jones, Tewkesbury, Onuh and Ndzi (2010a) A robotic welding system using image processing techniques and a CAD model to provide information to a multi-intelligent decision module. Assem' Automation AA-09-021.R1

Sanders DA and Stott IJ (2010b). - Analysis of failure rates with a tele-operated mobile robot between a human tele-operator and a human with a sensor system to assist. Robotica. Paper ROB-REG-08-0199.

Sanders DA (2010c) Comparing ability to complete simple tele-operated rescue or maintenance mobile robot tasks with and without a sensor system. Sensor Review. Paper SR-09-609.

Solea R and Nunes U (2007). Trajectory planning and sliding-mode control based trajectory-tracking for cybercars. Integrated CAE, 14 (1), pp 33-47.

Stott, IJ; Sanders, DA; Goodwin, MJ (1997). A software algorithm for the intelligent mixing of inputs to a tele-operated vehicle Jrnl of systems architecture 43 (1-5) pp 67-72.

Stott IJ and Sanders DA (2000a). The use of virtual reality to train powered wheelchair users and test new wheelchair systems. Int Jnl of Rehabilitation Research 23 (4), pp 321-326.

Stott I, Sanders D (2000b). New powered wheelchair systems for the rehabilitation of some severely disabled users. Int Jnl of Rehab Research 23 (3) pp 149-153.

Tewkesbury, G; Sanders, D (1999a) A new robot command library which includes simulation. Ind Rob 26 (1) pp 39-48.

Tewkesbury GE and Sanders DA (1999b). A new simulation based robot command library applied to three robots. Jnl of Robotic Systems 16 (8), pp 461-469.

Urwin-Wright S, Sanders D, Chen S (2002). Terrain prediction for an eight-legged robot. Jnl of Robotic Systems 19 (2), pp 91-98.

Urwin-Wright S, Sanders D, Chen S (2003). Predicting terrain contours using a feed-forward neural network. Eng' App' of AI 16 (5-6), pp 465-472.

Figure 1 - *Industrial robot KUKA KR125.*

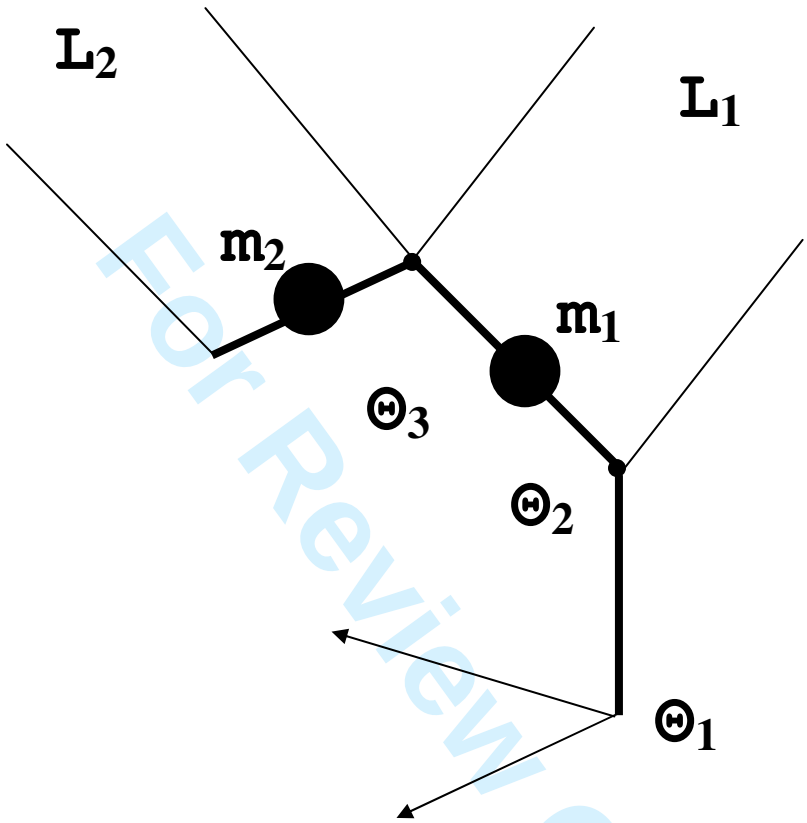
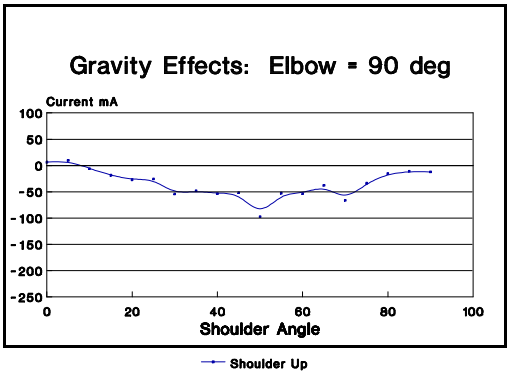
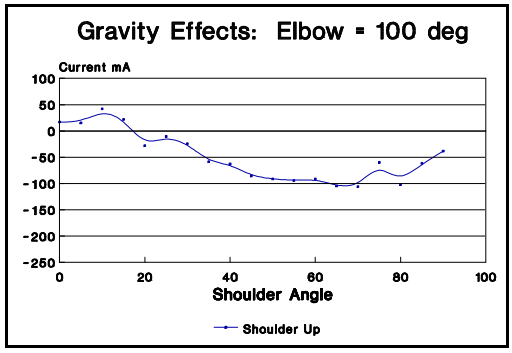
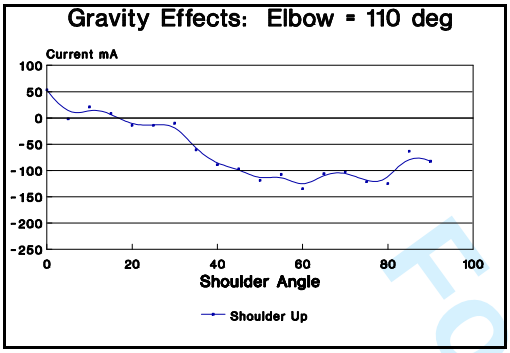
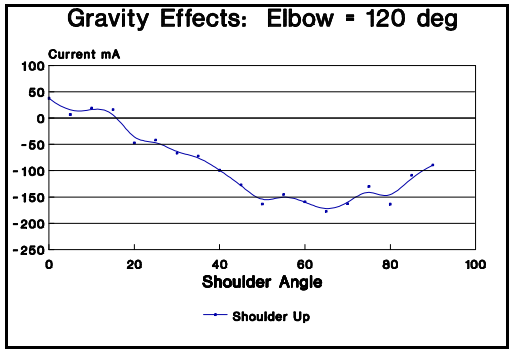


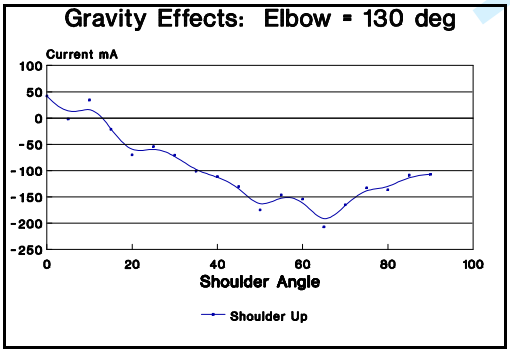
Figure 2:*The Simple Model used for the Three Main Links of the Robot.*

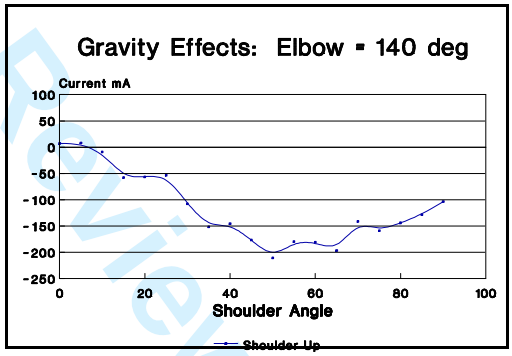


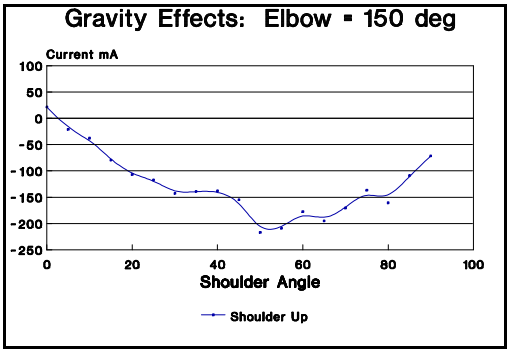


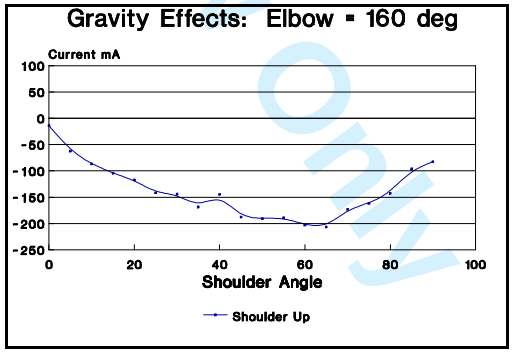


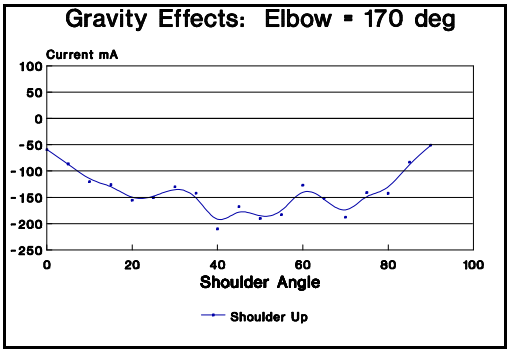












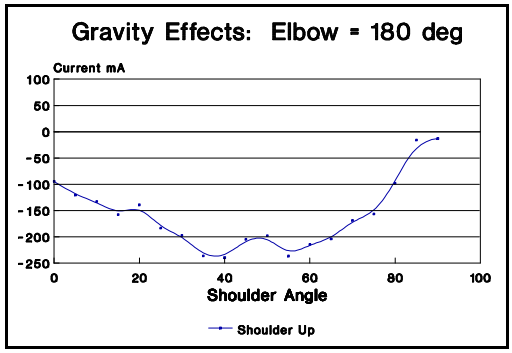
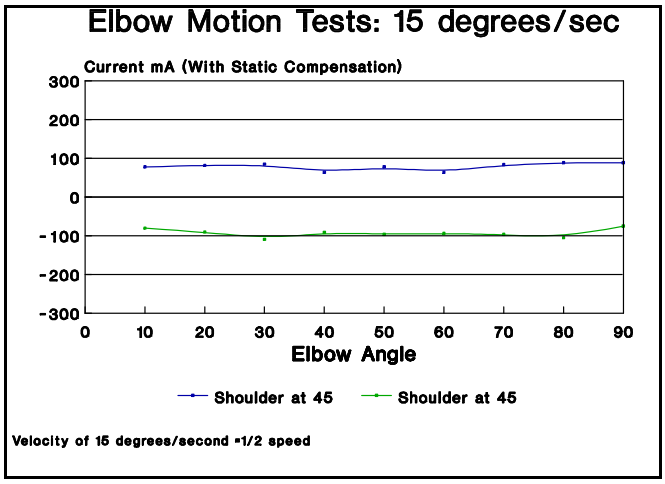
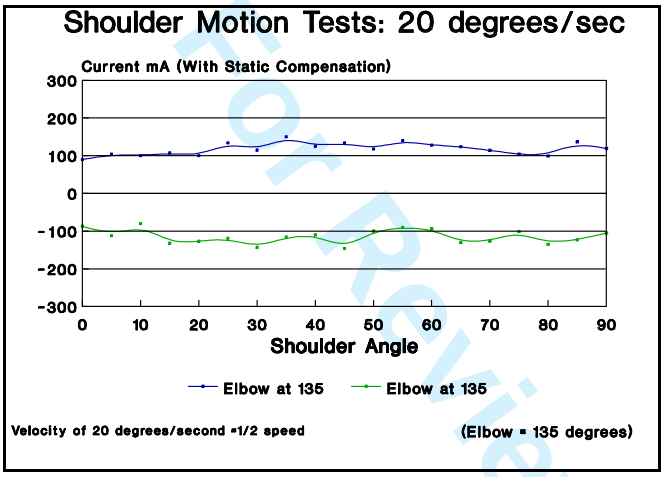


Figure 3 - *Remaining D_{2g} shown for the Prototype Robot with the Elbow angle marked.*

(





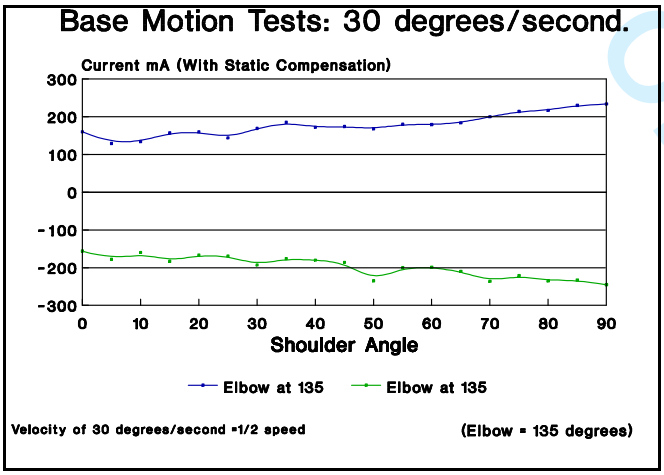


Figure 4 - *Current required to maintain a constant velocity for each joint for different configurations.*