OPTIMIZED SYNTHESIS OF A FORCE GENERATING PLANAR FOUR-BAR MECHANISM INCLUDING DYNAMIC EFFECTS

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

Dr. Charles F. Reinholtz, Chairman

Dr. Larry D. Mitchell

Dr. Robert L. West

November 28, 2001

Blacksburg, Virginia

Keywords: Synthesis, Kinematics, Mechanisms, Linkages, Optimization, Dynamics, Force Generation

OPTIMIZED SYNTHESIS OF A DYNAMICALLY BASED FORCE GENERATING PLANAR FOUR-BAR MECHANISM

Brian Tavis Rundgren

Charles F. Reinholtz, Chairman

Mechanical Engineering

Abstract

This thesis presents a technique for designing planar four-bar linkages by coupling optimization, dynamics and kinematics. This synthesis technique gives the designer the ability to design linkages having a desired resistance profiles under an assumed motion profile.

The design approach presented in this thesis calculates the resistance forces by using both the static and the anticipated dynamic effects of the resistance loading. Almost all research to date has assumed that the static forces in the linkage dominate the dynamic forces; hence, the dynamic effects have been neglected. This thesis shows that this assumption is often invalid.

The traditional approach for designing resistance-generating mechanisms has been based on closed-form methods that attempt to exactly match the resistance at a small number of discrete positions. This work uses a numerical optimization method that allows for the matching of the entire resistance curve by approximately matching a set of positions that define the shape of the curve.

This work furthers the discipline of mechanism design by combining dynamics into existing linkage synthesis methods, resulting in an improved synthesis method that includes both static and dynamic effects. While this approach can be used in many applications, this work

focuses on the design of exercise equipment. This focus is because exercise equipment designed to optimally stress a specific muscle group usually have a specific “strength curve” used to design the resistance load. The “strength curve” is the locus of all maximum loads moveable by the exerciser in all body part positions over the full range of motion. This application ideally suits the specification of the problem addressed in this thesis.

Acknowledgments

There are many people that deserve a word of thanks for their various forms of support that they have freely given me during the creation of this thesis. I would like to start with my adviser, Dr. Reinholtz, for his expert advice. Another member of my committee that I particularly would like to thank is Dr. Mitchell for his insight into the world of weight training and in helping me gain access to various weightlifting machines through his connections with The Weight Club, in Blacksburg, VA, to whom I also owe a debt of gratitude for their open willingness to assist me by granting me access to their facility.

Last but not least, I would like to thank my parents for their endless support and patience during this endeavor.

Contents

List of Figures

List of Tables

Nomenclature

A	Substitution variable
a_i	Link mass coefficient for link i
α_i	Angular acceleration of link I
B	Substitution variable
β	Resistance mass offset angle
C	Substitution variable
γ	User input offset angle
δ	Angle that force F_{54} makes relative to the positive horizontal axis
e	Base of the natural logarithm
F	Resistance force
F_{gx}	Substitution variable (constant)
F_{gy}	Substitution variable (constant)
F_{ij}	Force of link i on link j
$\mathrm{f}(X^i)$	Evaluation of the objective function for the design variable set X^i
θ_i	Angular position of link i
$\dot{\theta}_i$	Angular velocity of link i
$\ddot{\theta}_i$	Angular acceleration of link i
I	Mass moment of inertia for the resistance mass
j	$\sqrt{-1}$, index
IO_2	Substitution variable (constant)
IO_5	Substitution variable (constant)
M	Resistance mass

O_2, O_5	Fixed pivot points/Linkage grounding points
q	Optimization iteration counter
R_i	Vector representation of link i
r_i	Length of link i (scalar)
t	Substitution variable
V_i	Substitution variables (constants)
φ	Angle that force F_{34} makes relative to the positive horizontal axis
X^0	Existing vector of design variables
X^1	New vector of design variables
ΔX	Optimization move vector

Chapter 1 Introduction

Section 1.1 Background and Motivation

Man's quest to be faster and stronger lead to the development of what is now know as resistance training, often called weightlifting. The first and still the most popular form of these exercises use free-weights as the source of resistance. Free-weights are masses that the exerciser (the term "user" is used in rest of the thesis) moves around and the weight of these masses provides a resistance to the motion consistent with Newton's Second Law. Over the years, free-weights have developed into the standard barbell/dumbbell form, we see today. One of the disadvantages of free-weights is that the resistance is limited by the direction of the acceleration of gravity and by the user's ability to accelerate the free-weights opposite to the direction of gravity over a range of motion.

At some point in history, the idea of using a mechanism instead of free-weights to provide the resistance was developed. While the exact date may never be known, weightlifting machines have a long an interesting history, an example of this is found in a patent issued for an exercise chair to a Mr. M. V. B. White (1879). These machines will be referred in this thesis as weightlifting machines. Weightlifting machines provide the user with many advantages over the use of free-weights. Some of these advantages include increased safety due to the machines providing protection for the user from injury from accidentally dropping weights and improved ability to do the exercises correctly due to the equipment's ability to enforce proper posture and positioning. Weightlifting machines also improve injury rehabilitation since the support the equipment provides the users body helps to prevent aggravation or generation of an injury.

The advantage that is of interest here is the ability of weightlifting machines to provide a resistance curve that can be customized to match the user's strength curve for a particular exercise, thus providing the user with a more effective and efficient workout (Herz, 1901). A human strength curve is a measure of the ability of a human to exert varying forces over the entire range of motion of a given exercise (Scardina, 1996). This advantage can be seen, for example, by plotting the resistance force curve generated by a free-weight during the standing bicep-curl exercise versus the desired curve for the exercise that matches the bicep strength curve that can be obtained by designing a weightlifting machine to match this curve. The free-weight curve was developed based on a person standing upright and where 0 degrees is horizontal. Figure 1-1 shows the increase in work required by the optimally designed machine over the free-weight load curve. The increase in work done during the exercise makes using a weightlifting machine more efficient than using a free-weight. In addition, notice that the maximum force the user experiences is roughly 30% higher with a machine than with free-weights, which causes the exercise to be more effective.

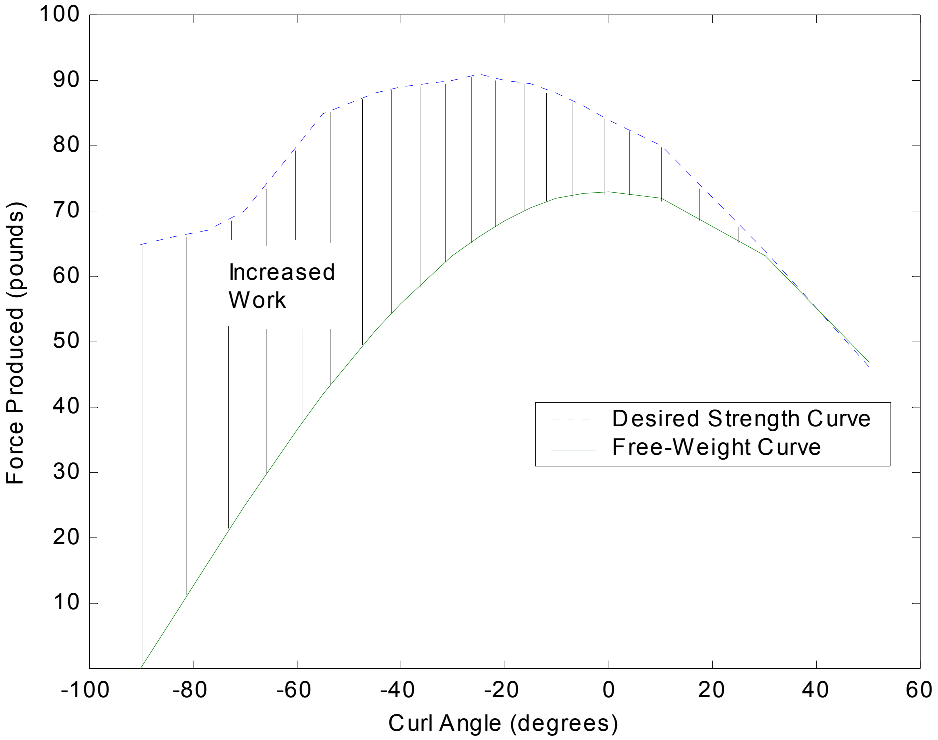


Figure 1-1 Work advantage of a weightlifting machine over free-weights for the bicep-curl.

Since the idea of human strength curves is important, one should know what the various published curves really represent. A number of decisions must be made when a strength curve is developed for a particular exercise. The most important is how to measure the strength curve. This question includes whether to measure the force generated statically at a series of discrete positions or to try to take force data dynamically throughout the exercise. Static data is easier to take and can eliminate muscle fatigue as a factor in the accuracy of the data, but the human strength curve for an exercise may differ when a person actually does the real (dynamic) exercise. When taking dynamic measurements, difficulty often arises in making sure that the instrumentation is always measuring the force correctly, since the instrumentation may move during the exercise and may not be measuring only the force normal to the users forearm. This problem is minor in static measurements since the instrumentation can be repositioned between each measurement. There is also the question of exactly how a person does a particular exercise, since each person often does an exercise a little differently. The biggest difference in this respect can come with positive (concentric) and negative (eccentric) resistance. For example, with the bicep-curl, one traditionally works the muscle during the contraction of the bicep (positive), but one can also work the muscle by carefully controlling the rate at which one re-extends the arm (negative). There is a great deal of research and debate regarding the subject of strength curves. For the purposes of this thesis, we will assume that there exists a human strength curve for each exercise and that a critical concept is that matching it provides an optimal workout for the user.

There are three main types of weightlifting equipment plus many minor types. One type uses a pin-selector plate stack that allows the user to select the number of plates that provide the resistance for the exercise. This controls the level of resistance. The stack is a permanent part of the machine and, thus, the maximum amount of resistance and the incremental size of the weight increase are fixed. An example of this type of weightlifting machine is pictured in Figure 1-2. The weight stack can be seen in the bottom center of the figure. This machine is typical of its class in that the weight stack is constrained to move only vertically.

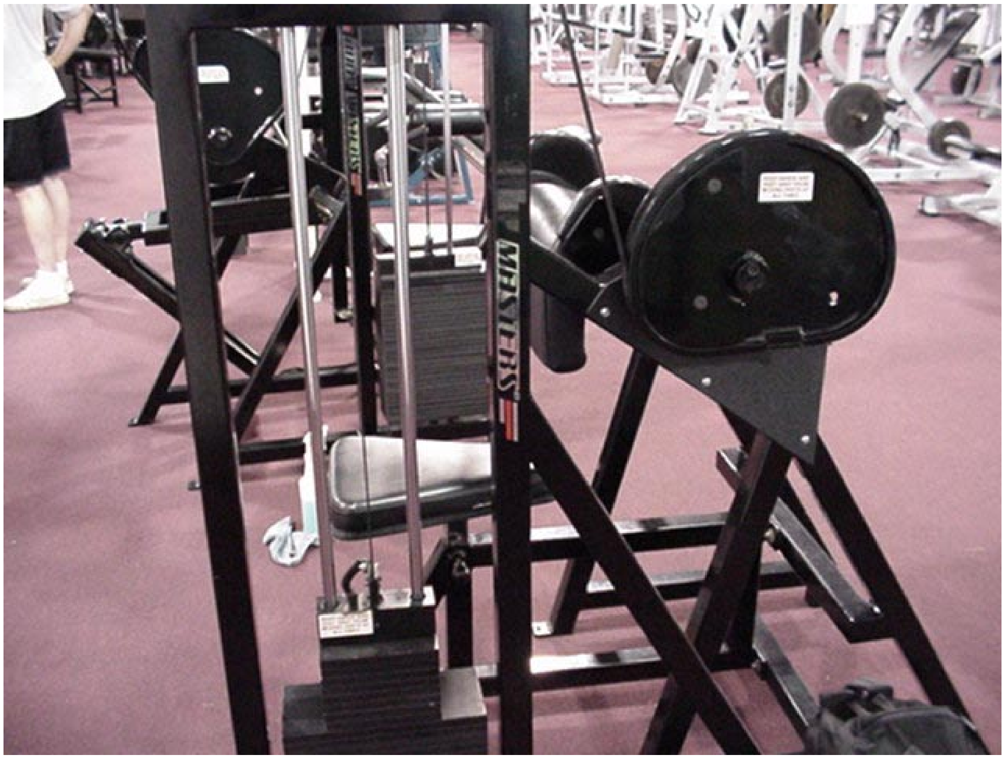


Figure 1-2 Body Masters™ stack-loaded bicep-curl exercise machine.

Another main type of machine has the weight added by the user to a post on the machine. This setup allows the weights to be used on multiple machines. Other advantages of this setup are that the basic machine is lighter, since no weights are built in, and the incremental increase in the loading is not predetermined, but can be selected by the user. A weightlifting machine of this type is shown in Figure 1-3. The weight loading post is located at the bottom right of the figure. Also visible is a feature common to almost all types of weightlifting machines, is a user handle grip system (center of picture) that accommodates a range of user sizes. This handle system attempts to keep the user input link from varying, resulting in each user having a similar mechanical advantage. The downside to this system is that force between the user's hand and the handle is not always perpendicular to the user's arm.

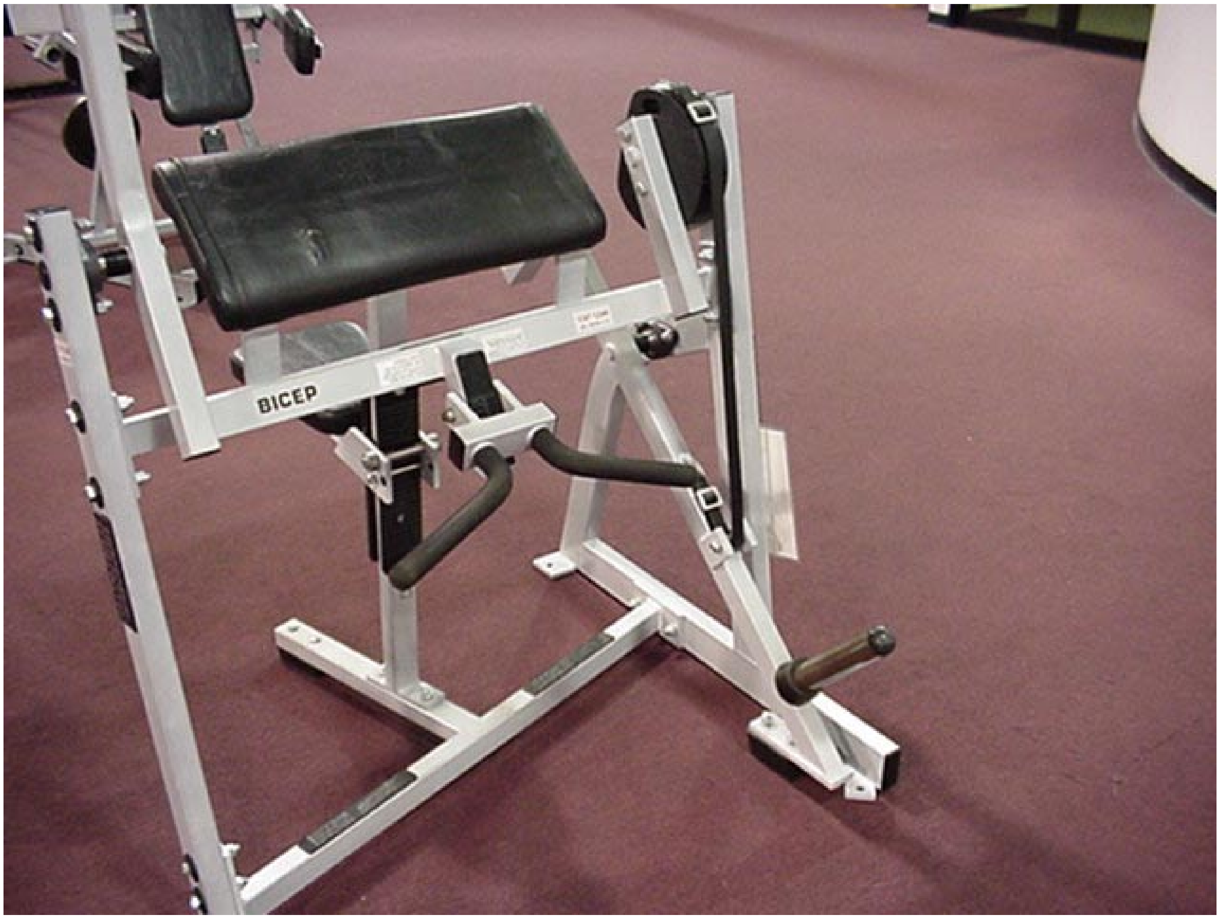


Figure 1-3 Hammer Strength^{TM} bicep-curl machine; uses independent weights.

The third type of machines produces resistance from the deformation of an elastic member. This design eliminates the need for weights, greatly reducing the weight of the machine. By switching between elastic members having different deformation properties, the user can vary the amount of resistance for each exercise. An example of this type of machine is shown in Figure 1-4. As with the stack-loaded machines, elastic resistance machines are typically limited in the number of different resistance levels they can provide the user, since the number and stiffness of the elastic members is preset when the machine is manufactured. Even though the elastic members are designed not to fail for millions of cycles, many potential users, who are use to steel weightlifting machines do not trust them.

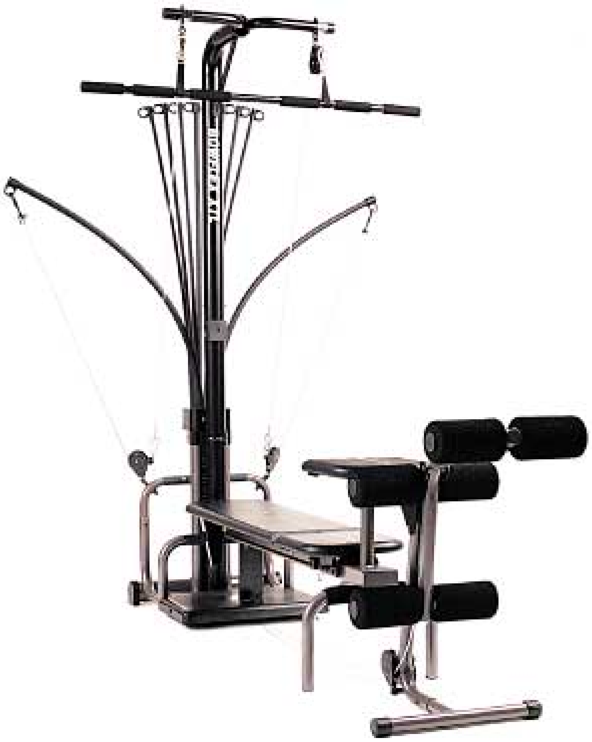


Figure 1-4 An example of an elastic resistance machine, the Bowflex XTL™.

There are also other minor types of weightlifting machines. Among these are machines that produce their resistance using a damper system, machines that use pneumatic or hydraulic springs, and machines that use electromagnetic/electromechanical devices. Damper-based machines produce resistance forces that are directly related to velocity, which limits their usefulness. The electromagnetic/electromechanical tend to be expensive, but may become a viable option in the future. These minor types currently do not occupy a large share of the weightlifting machine market.

This thesis will focus on the second type of setup, the user added weight setup. This setup is more interesting from a design point of view than the first setup because of the possible nonlinear motion of the load mass results in a more general problem. Moreover, the first setup is a constrained subset of the plate-loaded case. If the second setup is solved then the first setup design problem is solved. The major design issues in the third setup type result from the design of the elastic members rather than the kinematics and dynamics of the machine.

There are two common approaches to designing weightlifting machines to match human strength curves. One approach focuses on machines that use linkages and the other focuses on using cams. Cams traditionally require complex machining, whereas linkages are easy and inexpensive to produce. Linkages are also easier to assemble than cams, since unlike cams, links do not have to have their rotational position set during assembly to provide the correct relationship between the user input and the rotation of the cam. While the cam assembly problem can be made easier with the use of a keyway, it cannot be eliminated. The link's positions are inherently correct to within manufacturing tolerance given that the linkage was assembled in the desired closure. For these reasons, there exists a great deal of interest in the use of linkages in weightlifting machines, which partially motivates this thesis.

A good synthesis routine must include the dynamics of the mechanism, especially the dynamics associated with the weight mass. An example of this need is shown in Figure 1-5 where the strength curve of a row exercise was matched by Scardina (1996) using a static force only approach is compared with the same mechanism analyzed by the dynamic methods developed in this thesis. The input angle differs from what is shown in Scardina (1996). The angles displayed in Figure 1-5 are the supplementary angles to the ones in the original work due to difference in the angle definitions in the two analyses. A constant angular velocity of 1 radian per second was chosen for the input velocity. The exercise has a stroke of 40°, thus the entire exercise would be completed in approximately 1.4 seconds at the chosen velocity. The resistance mass was varied until the middle section of the dynamic roughly approximated the static curve. By examining Figure 1-5, one can see that the static and dynamic analyzes produce markedly different results. The difference can be directly attributed to the acceleration of the resistance mass. Figure 1-6 shows the angular acceleration of the resistance mass. The shape of the acceleration curve resembles the shape of the dynamic resistance curve, indicating a strong negative correlation between the resistance force and the inertia forces associated with the acceleration of the mass. These graphs clearly indicate that the dynamics of the mechanism dominates the static force at the two ends of the force curve even though the user is applying a time invariant input. The difference between the two curves may be greater if the user varies his or her input velocity during the exercise.

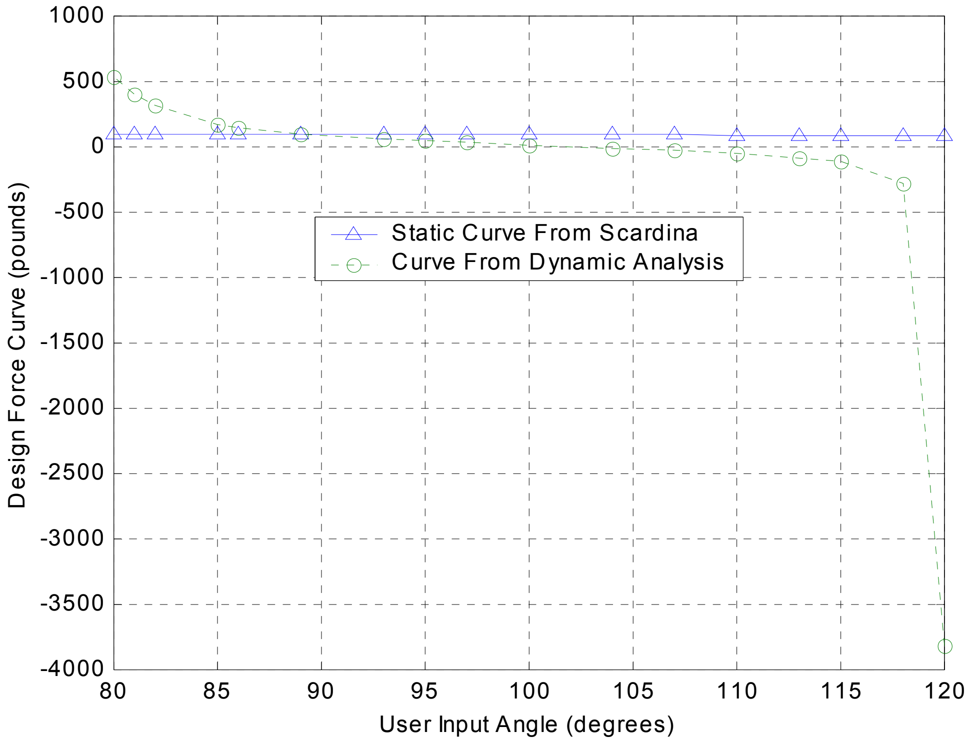


Figure 1-5 Comparison of static and dynamic results for the rowing machine.

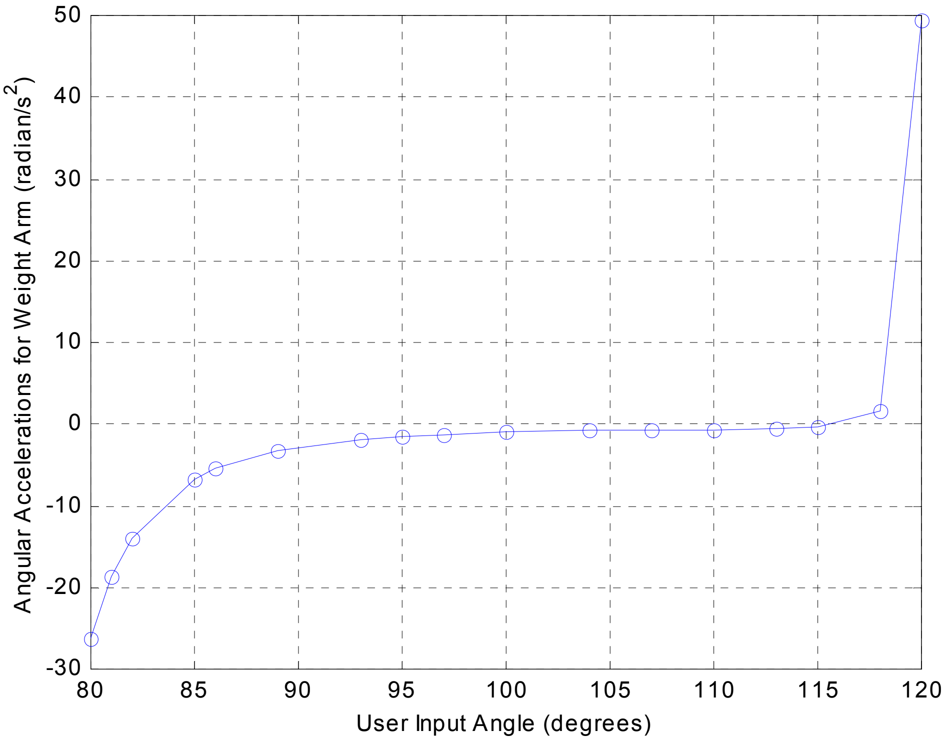


Figure 1-6 Angular acceleration of the weight mass for the row machine.

Section 1.2 Literature Review

The previous work done in kinematic synthesis that is of interest in this thesis falls into two categories; namely, optimal synthesis and synthesis involving the dynamics and forces of the linkage. An overview of relevant works follows.

Midha, Turcic and Bosnik (1984) published case studies in kinematic synthesis where complex number, algebraic, and numeric (optimal) synthesis techniques were applied to similar punching machine design problems. A discussion on the quality of the solutions to each problem is included in the paper. The optimized synthesis problem dealt with relating the position of the punch to the rotation of the input link. In other words, the case was a position synthesis problem. The optimization method did include checks on whether the mechanism would assemble and a check to insure that the input link rotated completely. Completely satisfactory solutions were not obtained in this case study.

Another work on optimized synthesis was published by Bagci and Rieser (1984). In their work, Bagci and Rieser successfully utilized optimization in designing four-bar linkages for position function generation. Unlike Midha, et al (1984), the synthesis dealt not just with position, but also with the velocity and acceleration of the linkage. By including the velocity and acceleration, the authors were able to design linkages that better represented the desired function. The optimization technique used rigidly defined set of equations for the objective function. This rigid definition of the objective function made mathematical manipulations easy, but limits the range of problems that can be dealt with. The optimization routine uses derivates of the objective function in finding the optimized solution. This method used by Bagci and Rieser is similar to the Linear Quadratic Regulator method frequently used in optimal control (Friedland, 1986).

Another notable work is that of Venkataraman et al (1992) where optimization was used to handle the synthesis of a four-bar linkage with inexact links that was to be used for body guidance; a task impossible to do using closed-form methods. As with many other works, this one uses a two-step optimization process. In this case, the process is the combination of traditional precision-point approach (Venkataraman, et al, 1992) to arrive a general idea of where in the solution space the optimal solution would lie followed by what Venkataraman called a Sequential Unconstrained Minimization Technique to refine the linkage design to get the final optimal solution. The importance of this work is that it shows the flexibility and robustness that optimal synthesis offers the designer.

The need to include dynamics has been known for some time (Starr, 1974 and Reinholtz, 1983), but little work has been done in this area. Starr (1974) discussed various works where dynamics were included in the synthesis routines and emphasized the advantage of dealing with the kinematics and dynamics of the linkage simultaneously rather than in a trial-and-error loop. The examples dealt with the forces generated by the masses and inertias in the linkage, but none of the works tried to *specify* the forces generated by the linkage.

The best work in the area of optimized synthesis that included dynamics was done by Rigelman and Kramer (1988). While their work includes the dynamics of the links with mass and inertia, it does not address the issue of force generation. In their synthesis method, they require the use of two different optimization routines to produce acceptable results. Also, the link design from which the mass and inertia was calculated was not verified as to whether the links could handle the forces (stresses) generated during the loading of the mechanism. As a result the links may not be an accurate representation of the links actually used in the final production design.

Soper (1995) produced an excellent compilation of analytical synthesis techniques for the force-generation problem. While Soper did include a discussion of the dynamic effects on the force generated by a linkage, the synthesis methods Soper developed were based on static forces. Soper also included a brief introduction to optimization, but did not utilize it in any way.

Some good work on optimized force-generating-mechanism synthesis has been done by Scardina (1996). Scardina developed an approach that found an optimized planar four-link mechanism that produced a resistance force curve that matched a desired human strength curve. The force analysis was based on a static loading of the mechanism by a mass attached to the linkage. No attempt was made to incorporate the dynamic effects of this load mass on the shape of the resistance force curve. As shown in Section 1.1, the lack of the dynamic effects may produce erroneous results.

Section 1.3 Review of Planar Linkage Kinematics

The position analysis of a four-bar linkage can easily be done by using Loop-Closure equations (Mabie and Reinholtz, 1986). The nomenclature used in the following derivation comes from the four-bar linkage shown in Figure 1-7. Writing the closure equation in complex polar form results in Equation (1.1),

$$r_7 e^{j\theta_7} + r_3 e^{j\theta_3} - r_5 e^{j\theta_{51}} - r_4 e^{j\theta_4} = 0 \qquad \textbf{(1.1)}$$

The ground link is customarily taken to lie along the real axis and thus the position equation can be rewritten as

$$r_7 + r_3 e^{j\theta_3} - r_5 e^{j\theta_5} - r_4 e^{j\theta_4} = 0 \qquad \textbf{(1.2)}$$

The link lengths, the r_i's and the input angle, θ_2, are generally known, but the angles, θ_3 and θ_4, are unknown and need to be solved for. First, Equation (1.2) must be rearranged so that $r_4 e^{j\theta_4}$ (or $r_5 e^{j\theta_5}$) is isolated on one side of the equation as shown in Equation (1.3)

$$r_4 e^{j\theta_4} = r_7 + r_3 e^{j\theta_3} - r_5 e^{j\theta_5} \qquad \textbf{(1.3)}$$

After forming the complex conjugate of Equation (1.3)

$$r_4 e^{-j\theta_4} = r_7 + r_3 e^{-j\theta_3} - r_5 e^{-j\theta_5} \qquad \textbf{(1.4)}$$

and then multiplying the conjugate with the original equation yields

$$r_4^2 = r_7^2 + r_3^2 + r_5^2 + r_7 r_3 \left(e^{j\theta_3} + e^{-j\theta_3}\right) - r_7 r_5 \left(e^{j\theta_5} + e^{-j\theta_5}\right) - r_3 r_5 \left(e^{j\theta_3} e^{-j\theta_5} + e^{-j\theta_3} e^{j\theta_5}\right) \qquad \textbf{(1.5)}$$

Expanding Equation (1.5) using the identity of $e^{j\theta} = \cos\theta + i\sin\theta$

$$r_4^2 = r_7^2 + r_3^2 + r_5^2 + 2r_7 r_3 \cos\theta_3 - 2r_7 r_5 \cos\theta_5 - 2r_3 r_5 \cos\theta_3 \cos\theta_5 - 2r_3 r_5 \sin\theta_3 \sin\theta_5 \qquad \textbf{(1.6)}$$

One is then in a position to take advantage of the following trigonometric identities:

$$\cos\theta_5 = \frac{\left(1-t^2\right)}{\left(1+t^2\right)}$$
$$\sin\theta_5 = \frac{2t}{\left(1+t^2\right)} \qquad \textbf{(1.7)}$$

where,

$$t = \tan\left(\frac{\theta_5}{2}\right) \qquad \textbf{(1.8)}$$

After applying these simplifications to Equation (1.6) and rearranging, one arrives at an equation in quadratic form with respect to *t*;

$$At^2 + Bt + C = 0 \qquad \textbf{(1.9)}$$

where

$$\begin{aligned} A &= r_4^2 - r_7^2 - r_3^2 - r_5^2 - 2r_7r_3\cos\theta_3 - 2r_7r_5 - 2r_3r_5\cos\theta_3 \\ B &= 4r_3r_5\sin\theta_3 \\ C &= r_4^2 - r_7^2 - r_3^2 - r_5^2 - 2r_7r_3\cos\theta_3 + 2r_7r_5 + 2r_3r_5\cos\theta_3 \end{aligned} \qquad \textbf{(1.10)}$$

Equation (1.9) can be solved for t using the quadratic formula, since the expressions for *A*, *B*, and *C* contain only known quantities. Once *t* has been found, θ_5 can be solved for using Equation (1.8). With θ_5 now known, Equation (1.4) can be solved for θ_4. If the linkage cannot be assembled given the input angle and link lengths, both of the angles θ_5 and θ_4 will come out as complex numbers.

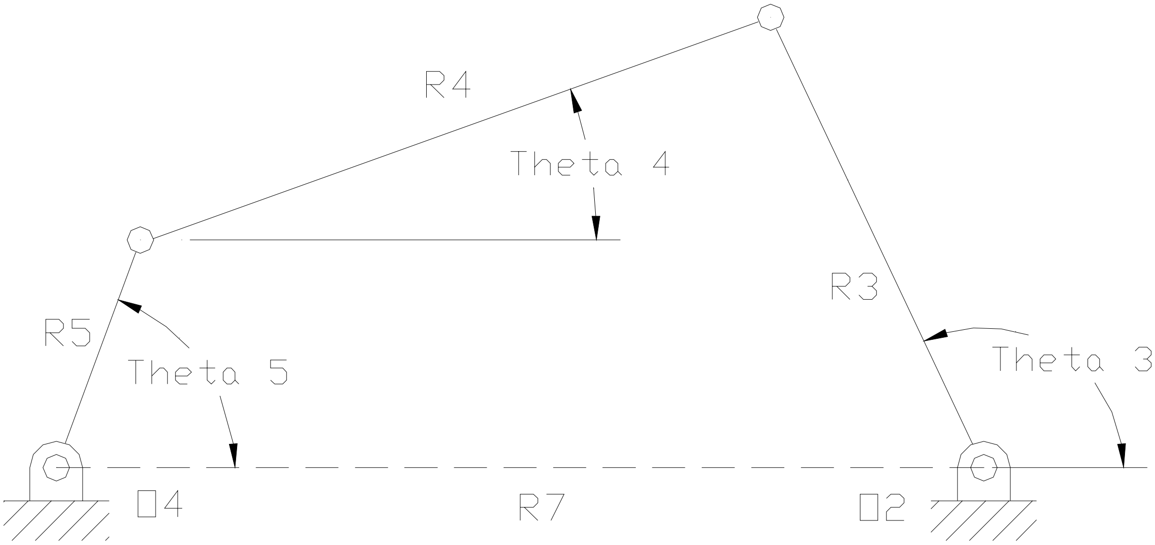


Figure 1-7 Drawing of a four-bar linkage with descriptive notation.

Chapter 2 Problem Definition

Based on the motivation and background given in Section 1.1, it has been deemed desirable to develop a synthesis method for designing weightlifting equipment. The focus of the equipment design has been selected to concentrate on planar four-bar linkages coupled with a resistance mass that produce a resistance curve that matches the human strength curve. The method must include the dynamics, as well as the kinematics, associated with the linkage. To produce the best match to the resistance curve, rather than a few precision points on the curve, an optimization technique will be used for the synthesis. Details regarding optimization will be discussed in Chapter 3. This chapter discusses the kinematic and dynamic analyses that arise from the requirements of the stated problem.

Section 2.1 Kinematics of a Planar Weightlifting Mechanism

The starting point for any mechanical analysis is to define the system being analyzed. As has previously been stated, the goal is to design a weightlifting machine that uses a linkage to achieve a specified resistance curve. Practical consideration dictated that the linkage should have one resistance mass and one user input. Such requirements limit the design to single degree-of-freedom mechanisms. Good engineering practice implores the engineer to try to keep the machine being designed as simple as possible. For this reason, a planar linkage is preferable to a spatial linkage. With these considerations, the desired mechanism is a planar four-bar linkage with the resistance mass loaded on one link and the user input applied to another link, such as is shown in Figure 2-1.

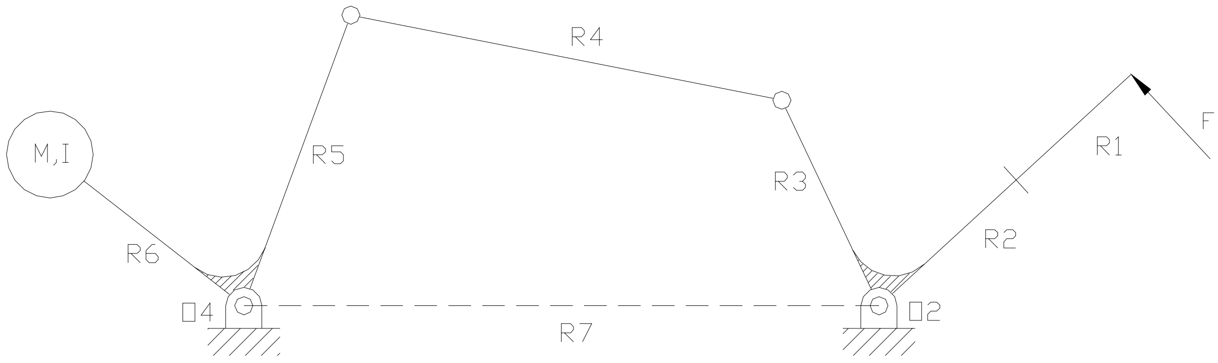


Figure 2-1 Generic Four-Bar Linkage with Resistance Mass

The links of the mechanism, denoted by the *Ri*'s, are assumed to be rigid, massless links whose lengths are available as design variables with the exception of R_1 and R_2. The notation of an upper case R denotes that both the length (r) and orientation (θ) are included in the link designator. The links R_1 and R_2 make up the user input link. The ground link (R_7), shown as a dashed line) is taken as a fixed link from O_5 to O_2. The angles γ (Gamma) and β (Beta) in Figure 2-1 are rigid offset angles for the user input and the resistance mass, respectively. These angles are available as design variables. The inertia of the mass (I) is not a design variable; rather it is the intrinsic inertia associated with the design variable mass (M) and is shown only to indicate that the dynamics of the mass are included in the problem.

The kinematic analysis of the four-bar linkage was developed in Section 1.3. The additions of the mass on link R_6 and user-input (R_1+R_2) links does not significantly impact the kinematic analysis since they are rigidly attached to links R_5 and R_3, respectively. Therefore, the angular velocities and accelerations of the mass and user-input links are the same angular velocities and accelerations as links R_5 and R_3, respectively. Since for this problem, the angular position, velocity and acceleration will be assumed for the user input link, the position, velocity and acceleration of $R3$ are the known inputs used in the four-bar kinematic analysis.

The assumption that the user input is known is based on a couple of aspects. The first is that the human body is a feedback-controlled machine that, within a reasonable operational range, is capable of controlling the motion and force generation of any of its members. This aspect allows the designer to assume that the user's angular input (acceleration and velocity) and the applied force are independent of each other as long as both are within the maximums that the person may achieve. Another is the idea that one can measure the motion of a person doing a bicep-curl with sufficient accuracy to allow the designer to claim a set of input values, position, velocity and acceleration, as being the known inputs to the system. When making this decision, the designer must recognize that each person is different. The user input will vary to a certain degree. The designer must examine these differences once a design has been made.

Optimization is being used to find the set of design variables that best solves the problem. For each analysis, the following quantities will always be known: the *R*i's, β, γ, M, I and all of the input link motion data. The unknowns are, thus, limited to the angular position, velocity and acceleration. The angular position can be solved used the previously mentioned methods. Angular velocity and acceleration equations can then be developed by differentiating the position equations. Now all of the kinematic properties of the mechanism have been found and are available for dynamic analysis.

Section 2.2 Dynamics of a Planar Four-Bar Linkage

Since the mechanism is being driven in a controlled fashion, the motion of all of the members of the mechanism can be found from the assumed user input. Before beginning the dynamic analysis, two decisions must be made. The first decision is on whether the links have mass properties or not. If the links are to have mass, one must know how the links are designed, or failing that, have a reasonable estimate of the inertial properties of the links. It sounds simple just to design the links and incorporate them into the analysis, but since the dynamic analysis is part of an optimized synthesis routine, the links will change for each iteration, meaning that the design of the link must also be updated for each iteration. A simple link design would be to choose a cross-sectional area for each link and assume that the area will give the link sufficient strength to prevent any failure, such as yielding or buckling, for all loading and link length conditions. The link mass and inertia would then simply be functions of the link length. The problem with this method is that the link-strength assumption may not be valid. The links could be over-engineered and, thus, could be larger than need be. One must also consider how important the mass of the links really is to the problem. If the mass is considered to contribute little to the forces of interest, then the mass could be left out with minimum impact on accuracy of the design. The actual results can be checked after the mechanism is constructed to verify or disprove this assumption. In this thesis, the dynamic effects of the load weight are assumed to be the dominant dynamic effect and the mass of the other linkage members was neglected.

The other issue that needed to be resolved was how to represent the load mass. The problem here is not whether to include it, but exactly how should it be modeled. In particular, one must determine how the mass is distributed when calculating the inertia. Usually weights in user-loaded weightlifting machines are thin disks. However, a given weight can be accomplished by using different combinations of plates. Each combination would generally have a different inertia. Since there is no way of knowing how each user will load the machine, there is no easy answer as to which set of plates will be used to make up each weight. In addition, while the drawing in Figure 2-2 shows the weight plates aligned such that the disks are shown in their circular view, this may not be the best arrangement for them. The actual production model of the machine may have the weights oriented in a different way. Thus, the inertia would need to be recalculated. One solution to this problem would be to choose a radius and an orientation for the load weight and then calculate the inertia. In this thesis, the decision was made to set the radius to zero, that is to make the mass a point mass. This decision was based on the difficulty of choosing an accurate radius and on the treatment that the mass receives in the dynamic equations. Since O_5 is a fixed point, the most natural way of solving the dynamics of the linkage subsection shown in Figure 2-2 is to take moments about O_5. The inertia of mass M about O_5 will include an term from the Parallel Axis Theorem, $M(r_6)^2$. So, if r_6 is larger than the radius of the weight plate, the Parallel Axis term will, be significantly larger due to the squaring of the radius, than the term due to the geometry of the plates. Due to possible clearance issues with the fixed pivot at O_5, one would in general want r_6 to be larger than the largest possible plate radius. For example, if the weight plate is oriented in its circular view, the orientation of highest inertia, it has a moment of inertia of ½ Mr_w^2. If r_w equals r_6, then the inertia due to the geometry of the plate is 33% of the total inertia. If r_w is one-half of r_6, the inertia of the plate is 11% of the total inertia. If r_w is one-quarter of r_6, the inertia of the plate is 3% of the total inertia. So, by choosing the mass to be a point mass, the dominant component of the inertia is retained.

To begin the analysis of the linkage substructure shown in Figure 2-2, some clarification must be made on the nomenclature used in this and subsequent figures. The force F_{ij} is the force applied by link i onto link j, taken as a positive force on link j. Thus, F_{54} is negative in Figure 2-2 since we are looking at the force from the reference frame of link 5 instead of the link 4 frame. Taking moments about O_5 results in

$$-r_5\sin(\theta_5)\cdot F_{54}\cos(\theta_4)+r_5\cos(\theta_5)\cdot F_{54}\sin(\theta_4)+Mgr_6\cos(\theta_5+\beta)=Mr_6^2\ddot{\theta}_5 \qquad \textbf{(2.1)}$$

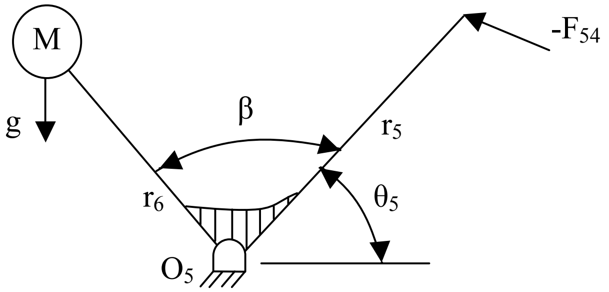


Figure 2-2 Diagram of weight loaded link.

Since the links are assumed massless, the coupler link becomes a simple two-force member. A free-body diagram of the coupler link (link R_4) is shown in Figure 2-3. The analysis of this link is trivial and yields that F_{54} = -F_{34}. This relationship allows for the coupling of the equations for the two "ends" of the linkage.

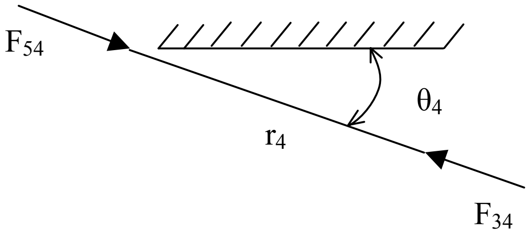


Figure 2-3 Free-body diagram of coupler link.

A diagram of the input subsection of the linkage is shown in Figure 2-4. As with the mass side of the linkage, moments were summed about the fixed pivot point, O_2, which yields,

$$F \cdot (r_1 + r_2) - F_{54}\cos(\theta_4) \cdot r_3 \sin(\theta_2 + \gamma) + F_{54}\sin(\theta_4) \cdot r_3 \cos(\theta_2 + \gamma) = 0 \qquad \textbf{(2.2)}$$

Examining Equations (2.1) and (2.2), one sees that the only unknowns are F and F_{54}. By rearranging the Equation (2.1), F_{54} is found to be

$$F_{54} = \frac{M(r_6^2\ddot{\theta}_5 + gr_6\cos(\theta_5+\beta))}{r_5\sin(\theta_5)\cos(\theta_4) - r_5\cos(\theta_5)\sin(\theta_4)} \qquad \textbf{(2.3)}$$

Equation (2.3) can now be substituted in to Equation (2.2) and the resulting equation solved for *F*. Finally, *F* is found to be

$$F = \left[\frac{M(r_6^2\ddot{\theta}_5 + gr_6\cos(\theta_5+\beta))}{r_1+r_2}\right] \cdot \left[\frac{r_3\left(\cos(\theta_4)\sin(\theta_2+\gamma) - \sin(\theta_4)\cos(\theta_2+\gamma)\right)}{r_5\sin(\theta_5)\cos(\theta_4) - r_5\cos(\theta_5)\sin(\theta_4)}\right] \qquad \textbf{(2.4)}$$

The resistance force, *F*, can now be calculated at any user input angle. The linkage's resistance curve can be compared to the desired strength curve. Notice that the Equation (2.4) is linear with respect to mass and the user-input-link length. This result guarantees that the final linkage design will provide the desired level of resistance, irrespective to how strong the user is. In addition, the curve is preserved for any sized user, though the necessary load mass will change. These properties are desirable since one wants a weightlifting machine to be useful to the maximum number of possible users.

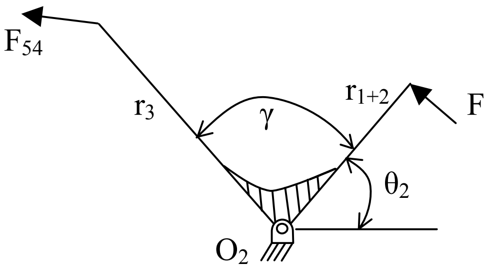


Figure 2-4 Diagram of user input and four-bar input links.

Throughout the dynamic analysis, there has been an assumption of frictionless joints. The joints on weightlifting machines often include rolling element bearings that keep the friction to a minimum. Another assumption has been that the links are rigid. The assumption of rigidity is common and usually valid in design problems since during the design process one can usually control the rigidity of the mechanism. One assumption that was made, and is usually made by others, but is often over looked, is the assumption of sea-level value for the acceleration of gravity, *g*. However, since almost all mass-based weightlifting equipment is used on the Earth's surface, there is little need to question this assumption.

Section 2.3 Dynamics when the Links Have Mass

Let us re-examine the dynamics of the four-bar linkage for the case when the links have mass. In this case, the link masses are considered to be point masses located at the ends of each link that vary in magnitude by the linear relationship, $m_i = 2a_i r_i$. The a_i's are constant, positive real numbers representing the mass per unit length of the beam. While representing the mass in this manner is a rough approximation, it does create a potentially more accurate model than the single-load mass model. These additional masses increase the complexity of the synthesis routine.

The dynamic analysis must be redone, since many of the simplifications from the previous section no longer apply. The biggest difference is in the dynamics of link R_4. For this reason, the dynamic analysis will begin with link R_4. A free-body diagram of link R_4 is shown in Figure 2-5. Notice that the addition of the mass means that the link is no longer a simple two-force member. The angles δ and φ are necessary since the force F_{34} and F_{54} can no longer be assumed to act along Link R_4.

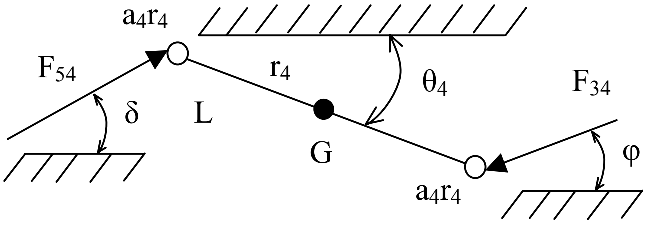


Figure 2-5 Diagram of the coupler link with point mass.

To perform a complete dynamic analysis of link R_4, the motion of the center of mass needs to be known. Since the exact geometry of the links is unknown, we will assume that the centers of mass are at the link midpoints. To find the acceleration of the center of mass, we will define the acceleration relative to the acceleration of an endpoints, since the acceleration of the endpoints are known. Therefore, for the endpoint shared with link R_5 (point L), see Figure 2-1 and Figure 2-5, we have,

$$\begin{aligned}
&\vec{a}_L = \vec{\alpha} \times R + \vec{\varpi} \times (\vec{\varpi} \times R) \\
&\vec{\varpi} = \dot{\theta}_5 \hat{k} \\
&\vec{\alpha} = \ddot{\theta}_5 \hat{k} \\
&R = r_5 \cos\theta_5 \hat{i} + r_5 \sin\theta_5 \hat{j} \\
&\vec{\varpi} \times R = -\dot{\theta}_5 r_5 \sin\theta_5 \hat{i} + \dot{\theta}_5 r_5 \cos\theta_5 \hat{j} \\
&\vec{\varpi} \times (\vec{\varpi} \times R) = -\dot{\theta}_5^2 r_5 \cos\theta_5 \hat{i} - \dot{\theta}_5^2 r_5 \sin\theta_5 \hat{j} \\
&\vec{\alpha} \times R = -\ddot{\theta}_5 r_5 \sin\theta_5 \hat{i} + \ddot{\theta}_5 r_5 \cos\theta_5 \hat{j} \\
&\vec{a}_L = \left(-\dot{\theta}_5^2 r_5 \cos\theta_5 - \ddot{\theta}_5 r_5 \sin\theta_5\right)\hat{i} + \left(-\dot{\theta}_5^2 r_5 \sin\theta_5 + \ddot{\theta}_5 r_5 \cos\theta_5\right)\hat{j}
\end{aligned} \qquad \textbf{(2.5)}$$

To compute $\mathrm{a_G}$ from a_L,

$$\vec{a}_G = \vec{a}_L + \vec{\alpha} \times R + \vec{\varpi} \times (\vec{\varpi} \times R)$$

$$\vec{\varpi} = \dot{\theta}_4 \hat{k}$$

$$\vec{\alpha} = \ddot{\theta}_4 \hat{k}$$

$$R = \frac{r_4}{2} \cos\theta_4 \hat{i} + \frac{r_4}{2} \sin\theta_4 \hat{j}$$

$$\vec{\varpi} \times R = -\dot{\theta}_4 \frac{r_4}{2} \sin\theta_4 \hat{i} + \dot{\theta}_4 \frac{r_4}{2} \cos\theta_4 \hat{j}$$

$$\vec{\varpi} \times (\vec{\varpi} \times R) = -\dot{\theta}_4^2 \frac{r_4}{2} \cos\theta_4 \hat{i} - \dot{\theta}_4^2 \frac{r_4}{2} \sin\theta_4 \hat{j}$$

$$\vec{\alpha} \times R = -\ddot{\theta}_4 \frac{r_4}{2} \sin\theta_4 \hat{i} + \ddot{\theta}_4 \frac{r_4}{2} \cos\theta_4 \hat{j}$$

$$\begin{aligned} \vec{a}_G = &\left(-\dot{\theta}_4^2 \frac{r_4}{2} \cos\theta_4 - \ddot{\theta}_4 \frac{r_4}{2} \sin\theta_4 - \dot{\theta}_5^2 r_5 \cos\theta_5 - \ddot{\theta}_5 r_5 \sin\theta_5 \right) \hat{i} \\ &+ \left(-\dot{\theta}_4^2 \frac{r_4}{2} \sin\theta_4 + \ddot{\theta}_4 \frac{r_4}{2} \cos\theta_4 - \dot{\theta}_5^2 r_5 \sin\theta_5 + \ddot{\theta}_5 r_5 \cos\theta_5 \right) \hat{j} \end{aligned} \qquad \textbf{(2.6)}$$

Free-body diagrams of the other two substructures of the linkage are shown in Figure 2-6 and Figure 2-7.

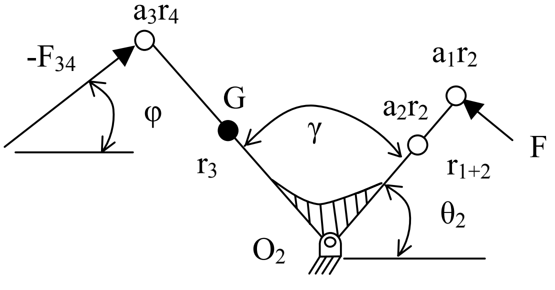


Figure 2-6 Diagram of the input link with point masses.

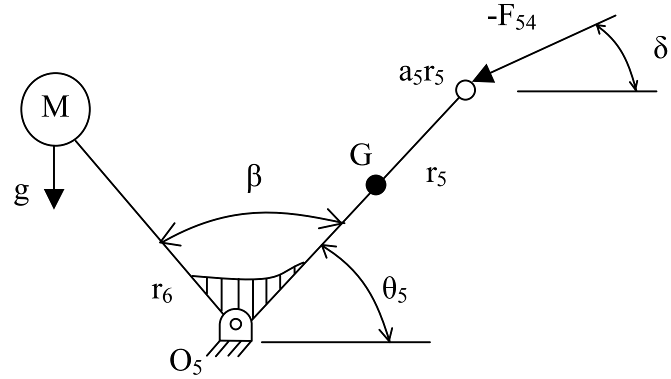


Figure 2-7 Diagram of the output link with point masses.

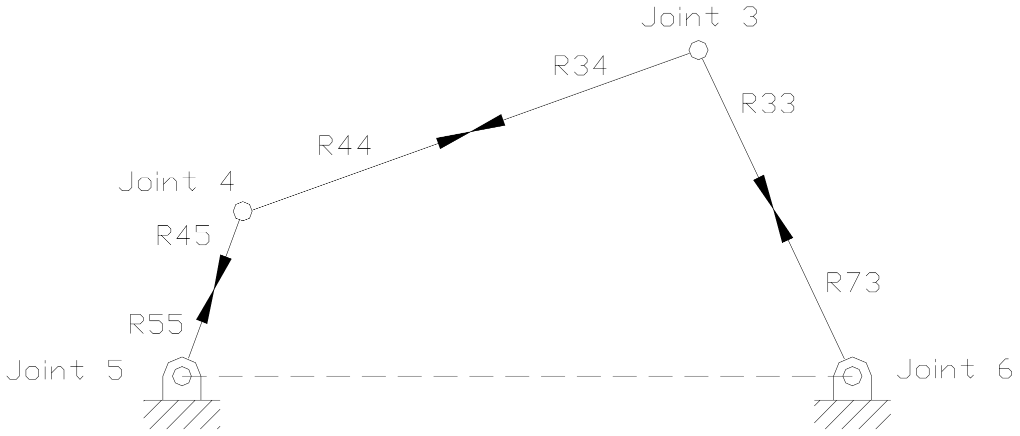


Figure 2-8 Joint and center of mass locating vectors diagram.

Analyzing the links shown in Figure 2-5, Figure 2-6 and Figure 2-7 by summing forces and summing moments about the center of gravity of each link yields the following equations,

$$\Sigma F : F_{43} + F_{73} = M_3 Ag_3 + M_3 g \tag{2.7}$$

$$\Sigma F : F_{45} + F_{75} = M_5 Ag_5 + M_5 g \tag{2.8}$$

$$\Sigma F : -F_{43} - F_{45} = M_4 Ag_4 + M_4 g \tag{2.9}$$

$$\Sigma M : R_{33} \times F_{43} + R_{63} \times F_{73} + T_s = I_3 \ddot{\theta}_3 \tag{2.10}$$

$$\Sigma M: R_{45} \times F_{45} + R_{55} \times F_{75} + T_w = I_5 \ddot{\theta}_5 \tag{2.11}$$

$$\Sigma M: -R_{34} \times F_{43} - R_{44} \times F_{45} = I_4 \ddot{\theta}_4 \tag{2.12}$$

where the mass and inerita properties for all links are defined as,

$$M_i = 2a_i r_i \qquad I_i = \frac{M_i r_i^2}{4} \tag{2.13}$$

Note that the inertial torque due to the load mass, T_w, is considered separately. The vectors, R_{ij} are defined as the vectors that located link j's center of mass relative to the joint i. The joint, link and locating vector notations are shown in Figure 2-8. To find $T_{w,}$ moments are summed about O_5 for the weight link,

$$T_w = -Mr_6^2 \ddot{\theta}_5 - Mgr_6 \cos(\theta_5 + \beta) \tag{2.14}$$

Rewriting the system equations by expanding the force equations into *x* and *y* components and then putting the resulting set of 9 scalar equations into matrix form yields,

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -R_{33y} & R_{33x} & -R_{63y} & R_{63x} & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -R_{45y} & R_{45x} & -R_{55y} & R_{55x} & 0 \\ R_{34y} & -R_{34x} & 0 & 0 & R_{44y} & -R_{44x} & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} F_{43x} \\ F_{43y} \\ F_{73x} \\ F_{73y} \\ F_{45x} \\ F_{45y} \\ F_{75x} \\ F_{75y} \\ T_s \end{bmatrix} = \begin{bmatrix} M_3 Ag_{3x} \\ M_3 Ag_{3y} + M_3 g \\ M_5 Ag_{5x} \\ M_5 Ag_{5y} + M_5 g \\ M_4 Ag_{4x} \\ M_4 Ag_{4y} + M_4 g \\ I_3 \ddot{\theta}_3 \\ I_5 \ddot{\theta}_5 - T_w \\ I_4 \ddot{\theta}_4 \end{bmatrix}$$

T_s can now be calculated. Note that T_s is the torque applied to link R_3 by the user-input link and is not directly the torque due to the user force *F*. By summing moments on the user-input link, which was not included in the matrix equation above, yields the following equation for *F*,

$$F = \frac{T_s}{r_1 + r_2} + \frac{r_2 a_2 g \cos\theta_2}{2} + r_1 a_1 g \cos\theta_2 + \ddot{\theta}_2 \left(\frac{r_2 a_2}{4} + r_1 a_1 \right) \qquad \textbf{(2.15)}$$

Chapter 3 Optimization

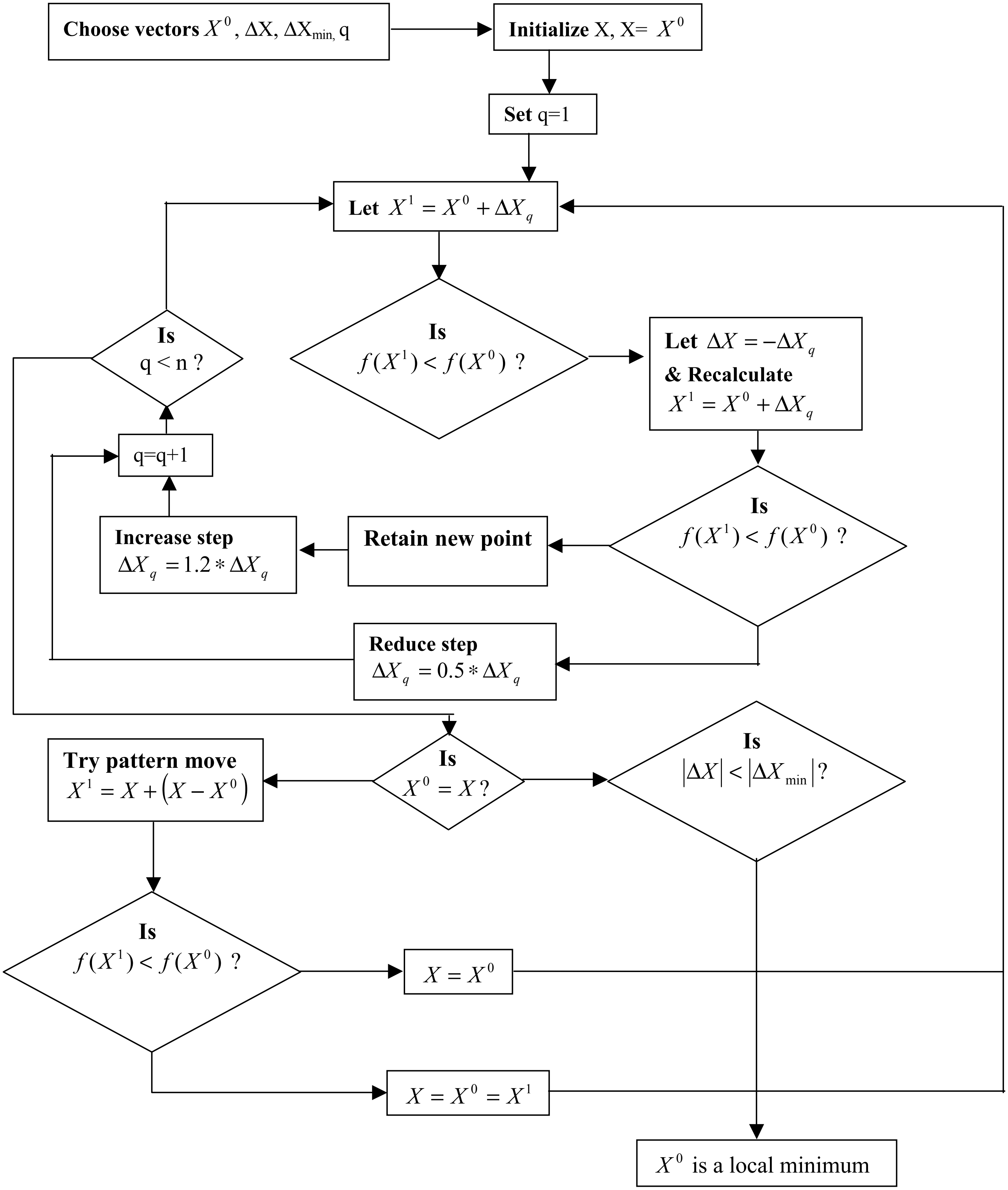
Optimization, as defined by Reinholtz (1983), "… the process of seeking the best result under a given set of circumstances." Based on this definition, optimization can be broken down into two parts: the *optimization routine* (the process) that finds minimum values of the second part, the *objective function* (the circumstances). Optimization routines are based on one or more numerical or closed-form-root finding techniques. These discussed in the first section of this chapter. Objective functions are defined by the user to quantify the characteristics that a perfect solution should, or should not have. The design of objective functions is discussed in the second section of this chapter.

Section 3.1 Optimization Routine

Many references on optimization exist and the examination of just a few can give one an excellent overview of the optimization routines available to the designer. Many of the more "intelligent" optimization routines rely on differentiating the objective function (Rao, 1984). The need to take partial derivatives is almost guaranteed when one looks at the so-called "global" optimization routines (Floudas and Pardalos, 1992). Obviously, this requirement creates a significant hurdle when dealing with complex functions often encounters in linkage synthesis. This requirement also forces the use of only continuous functions, which may not be appropriate for the problem (Vanderplaats, 1984). Not only is rewriting the objective function to include the linkage evaluation functions time consuming, the resulting function may not be differentiable. Often for problems with difficult derivatives, approximations of the functions are made allowing for easier derivatives, which allows gradient methods to be used (Vanderplaats, 1984). These methods has been found to greatly increase solution times, making the gradient routines slower than direct search routines for these types of problems (Eason and Fenton, 1974). Even for the routines that do not require differentiation, the fact that objective functions may return imaginary numbers causes problems with implementing them. For these reasons, direct search, often called "brute force", techniques are popular. These techniques use the simplest method of optimization that is choosing an initial set of values for the design variables, evaluating the objective function at the initial position and then deciding where to choose the next position. The selection of the next position is based on examining the results of a pattern search in the area of the first position. The pattern search tells the routine where the lowest or highest value, depending on the problem, of the objective function is in the search area, and this location is taken for the next position. Techniques of this type usually differ primarily in the pattern that is used to select the next set of points.

The Hooke and Jeeves (1961) optimization routine was chosen for this thesis. Aside from the reasons listed above, one reason this routine was chosen is its relative simplicity as a routine, making it easy to program and to use. In addition, this author has had some experience with this routine and has found that it produces good results on problems of this type. A flow chart for the Hooke and Jeeves (1961) optimization routine is shown in Figure 3-1. In the figure, the *X*'s are the different sets of values for the design variables and *q* is a counting variable that keeps track of the number of iterations completed. Hooke and Jeeves is based on a two step routine. In the first step, small incremental changes (positive and negative) to each design variable with a check on the effect these small changes (step) have on the objective function. This first step is used to determine the "direction" for the next move. The next stage consists of proceeding to move in this "direction" (pattern move) until such a move no longer reduces the value of the objective function. To improve the convergence rate of the routine, after each successful pattern move, the step size increases, allowing the routine to proceed quickly along a good direction. A failed pattern move resets the step size to the original user defined default. A failed exploratory move causes a reduction in the step size. The routine terminates when the exploratory move does not significantly change the value of the objective function or if the number of iterations reaches a predetermined limit. Hooke and Jeeves evaluates the changes in the objective function for a change in each design variable independent of the changes in the other variables. When making its exploratory move, there does exists a highly contrived case where the routine will not converge on a minimum. This case assumes if the initial variable values place the initial objective function value exactly in the middle of a “ridge” and the lay of the ridge is aligned relative to the variable axes in such a way that the direction dictated by the step sizes lie along the direction of the ridge. The routine will end up walking along the ridge, or terminating at the initial position. This case primarily arises because Hooke and Jeeves evaluates each variable independently as to the effect it has on the value of the objective function. So, the program has no way of knowing that a combined variable change will land position the next evaluation point on the ridge until the routine tries to make its first move. Since the step size is known, as well as any change in step size, the relationship between the design variables and the objective function could be easily determined. As such there would be little need for an optimization routine for a case of this type. So, for all practical purposes, the assumption that Hooke and Jeeves is always locally convergent is acceptable.

Figure 3-1 Hooke and Jeeves Optimization Flowchart


Choose vectors X^0, ΔX, $\Delta X_{min,}$ q
Initialize X, X= X^0
Set q=1
Let $X^1 = X^0 + \Delta X_q$
Is $f(X^1) < f(X^0)$?
Let $\Delta X = -\Delta X_q$
& Recalculate
$X^1 = X^0 + \Delta X_q$
Is q < n ?
q=q+1
Is $f(X^1) < f(X^0)$?
Increase step
$\Delta X_q = 1.2 * \Delta X_q$
Retain new point
Reduce step
$\Delta X_q = 0.5 * \Delta X_q$
Try pattern move
$X^1 = X + (X - X^0)$
Is $X^0 = X$?
Is $|\Delta X| < |\Delta X_{min}|$?
Is $f(X^1) < f(X^0)$?
$X = X^0$
$X = X^0 = X^1$
X^0 is a local minimum

Section 3.2 Objective Function Design

The objective function defines the problem to be solved by the optimization routine. As a result, the solution that the optimization routine produces is a function of the problem definition. The optimization routine minimizes (or maximizes) a set of mathematical relationships. The objective function must be designed in such a way that the conditions of accomplishing a desired task are represented in a well-defined mathematical form. Because the value of the objective function is usually designed to increase as the solutions deviate from the ideal solution, or in other words, penalize the solution for failing to meet the specifications, objective functions are also referred to as penalty functions. Clearly then, the design of the objective function is critical to having the optimization routine produce good results.

Ideally, an objective function will result in just one minimum that corresponds to the best possible mechanism for the particular application, but this standard is difficult to achieve when dealing with complex problems. The general trend is that the more complex the problem, the greater the number of local minima, and the more prevalent the ridges and undulations are in the solution space. While the solution space is rarely as low order as three dimensions, the three dimensional topographical terms are often used to provide the reader and the designer a better visual understanding of the nature of the solution space. A more practical goal is to develop an objective function that results in an easily discernable, finite region in which the optimal solution exists, but that also contains many "good" solutions that may be acceptable to the designer. In view of this more generalized goal of objective function design, the objective function needs to have a large, in a relative sense, value in the region outside of the acceptable design space and to have a slope that will force the optimization routine back into the acceptable region if routine tries to leave the region. One way of accomplishing this goal is to add a large number (a penalty) to the value of the objective function for any solutions outside of the acceptable region coupled with a function that increases rapidly, such as an exponential or power function, as the solutions move further from the acceptable region. For example, in the example problems discussed in this thesis, the step component of the penalty for closure failure is 10^{40}. To achieve the same objective function value from the quality of fit assessment would require each design point to be 36,889,000 times larger than desired. This type of objective function is often called a penalty function, since it adds a penalty when a constraint is violated. Defining the acceptable region in this manner is often categorized as imposing a set of inequality constraints on the problem. One should note that this method creates a discontinuity in the objective function and this may cause problems with optimization routines that depend on taking or estimating derivatives.

The question is then raised, what defines the acceptable region. The acceptable region is defined by a fundamental relationship that applies to the problem that cannot be violated. Some examples of this would be the Conservation of Energy and Mass, and the limitation of no physical object being able to travel faster than light. For linkage synthesis, this region would be defined by the requirement that the linkage must actually assemble in all of the desire positions. Quantitatively this requirement can be stated as: all the angles and link length calculations for the linkage must result in real numbers. While fulfilling this requirement is necessary, the fulfillment of this requirement does not guarantee that the linkage will work or that it will have the desired characteristics. Further requirements may need to be included in the objective function to handle other defects that the mechanism may have. Among these defects are branch defects, order defects and Grashoff defects (Mabie and Reinholtz, 1986). One may also need to set limits on the size or another linkage property that must be met. If this need exists then, similarly designed objective functions will need to be implemented.

Within the constrained region, the objective function is inversely proportional to how closely the mechanisms actual properties match the desired properties. Before the objective function can be designed, careful consideration must be given to what criteria the mechanism should be judged by, as well as the relative importance of each criterion.

Obviously, in this case, a cost associated with the deviation of the mechanism's resistance curve from the desired curve must be included. In the case being dealt with here, this part of the objective function contains the primary objective that is to produce a mechanism with a desired resistance curve. Since the difference between the actual and desired curves can be both positive and negative and penalty functions need to be positive, some means must applied to the difference to convert it to a positive only values. One way of handling the negatives is to simply raise the difference by an even power and the resulting value will always be positive. The choice of which power to use will depend on the magnitude relative to the other objective function terms that works best for the particular problem. For example, if the cost is too small, it will not provide the optimization routine with enough information on the fit for it to be able to improve it. If the cost is too large, it can overwhelm the other costs, and can cause the optimization routine to seek a solution outside the constrained region. Clearly the number of points that one is trying to match will impact the choice of function for the fit cost, since 20 points will result in a larger cost than 10 points will for the same function. So, caution must be used. For example, if the objective function has been decided on, and the designer changes the number of points being matched, the objective function will not perform as originally intended, affecting the overall performance of the optimization routine. Optimization presents no limit on how many properties one may try to match at once, but the odds of finding a solution to a problem decrease with the number of properties being matched. For this reason, one should closely examine whether a property needs to be matched exactly or whether it could be simply given a one-sided constraint.

Frequently, one may need to include in the objective function a representation of any secondary objectives in the problem. Secondary objectives represent properties or conditions that one would *like* the solution to have, but are not critical. An example from the weightlifting problem is that there is a desire to keep the resistance mass as small as possible, but a solution with a large resistance mass would be accepted if no other solution exists. Clearly, the costs associated with the secondary objectives need to be relatively smaller than cost associated with the primary objective. In addition, one sees that secondary objectives are usually represented by one-sided constraints that try to minimize or maximize a property. Secondary objectives allow the optimization routine to see differences in otherwise identically valid solutions.

Another consideration in designing objective functions is computational complexity of the function. Since the objective function is used many times by the optimization routine (see Figure 3-1), an overly complex objective function can slow down the entire process. In addition, often the more complex objective functions are likely to have more local minima, increasing the possibility of the optimization routine finding a solution that is not desired.

The last issue in objective function design is testing and iteration. Before the objective function is used in optimization, it needs to be tested. The designer needs to check the values of each of the sub-functions of the objective function in relation to the solutions that the objective function produces. Often one or more of the sub-functions needs to be adjusted to achieve the correct balance between the sub-functions. This testing is best done by starting from the same starting point for the optimization and seeing if the routine produces improved solutions for each variation of the objective function. Several iterations of this step may be necessary.

Chapter 4 Example Problem 1: Bicep-curl Exercise

Section 4.1 Bicep-curl Exercise

As discussed in Section 1.1, the goal of this thesis is to develop a method for designing a weightlifting machine with a resistance curve that matches the human-strength curve. For the example problem used in this chapter, the exercise of interest is the bicep-curl. A strength curve for a bicep-curl is shown in Figure 4-1. The curve was plotted with 180 degrees corresponding to the arm being fully extended. The force data was recorded in a series of static situations where the force measured was the force generated in the direction normal to the forearm and in the plane of motion. Several data sets were averaged to produce the smooth curve plotted on the graph. No specific physiological information was given about user or users on whom the strength-curve experiment was done (Clarke et al, 1950).

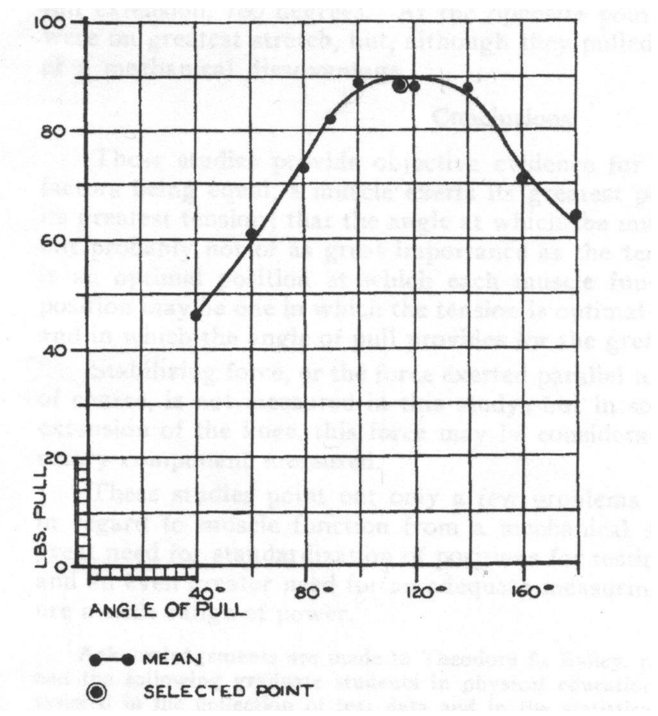


Figure 4-1 Human-strength curve for a bicep-curl taken from Clarke, et al (1950).

Discrete points were read from the graph in Figure 4-1 for use in the synthesis process. While reading points off of a graph does introduce some uncertainty, the uncertainty in the formulation of the original strength curve and general variation in human response makes the graph reading error insignificant. Strength curves may be formulated in many ways. Strength curves can be made for concentric and eccentric parts of the exercise, and the strength can be measured in either a static or a dynamic manner. Since these points must reasonably represent the strength curve, proper spacing of the points and having a sufficient number of points was critical. The angular position of the design points was converted into a coordinate frame where the user's arm is oriented along the vertical axis and where all angles are measured relative to the positive horizontal axis. Thus at a straight down position in Figure 4-1 the angle is 180°, but with reference to the new coordinates, the angle to be used is would be –90°. This conversion was necessary to have the data agree with the standard kinematic convention on the angular measurement of linkage positions which the kinematic and dynamic analyzes use. The selected design points are listed in Table 4-1.

Table 4-1 List of Design Points

Angular Position	Force
(degrees)	(pounds)
-90.0	65
-83.7	66
-77.3	67
-70.0	70
-60.0	80
-55.0	85
-45.0	88
-39.5	89
-30.0	90
-25.0	91
-20.0	90
-15.0	89.5
-10.0	88
-5.0	86
0.0	84
10.0	80
30.0	64
50.0	46

The spacing of these points was determined by first taking 10° increments, then subdividing that increment to 5° in the critical areas of the curve, areas with the greatest change in function value. Further subdivisions were done during the design process as the need for closer spacing in certain areas of the curve became apparent.

While the values in Table 4-1 are static measurements, the actual exercise is done with in a continuous motion. Therefore, a velocity and acceleration must be assigned to each of the design points. Based on observations by this author, one reasonable approximation of the motion is that other than the accelerations at the beginning and end of the exercise, the bicep-curl is a constant velocity motion with an angular velocity around 1.8-1.9 radians per second. So, for all but the beginning and end points, the design points were given a velocity of 1.9 radians per second and zero acceleration. This velocity profile is shown in Figure 4-2. While the average velocity could be roughly measured using a stopwatch, measuring the acceleration would require a more complicated setup. This thesis deals with mechanism synthesis and not exercise physiology. Because of this, the decision was made to approximate the acceleration curve. In addition, if one assumes an acceleration and designs the machine with it, the effects any error with the assumed acceleration will have on the strength curve will be partially contained. What will happen is that the user may not be able generate an acceleration greater than the assumed acceleration since any acceleration greater than the assumed may result in a higher resistance force, an increase that the user will not have the strength to overcome assuming that the strength curve is approximately matched. In addition, the spacing of the design points is too coarse to allow for a proper representation of an accurate acceleration curve. For these reasons, the acceleration of the end points was set as a constant equaling to the slope of the velocity curve. This slope is equal to the steady state velocity squared divided by 0.22 in units of radians/s^2. So, for a velocity of 1.9 radians per second, the acceleration is ±16.4 radians/s^2.

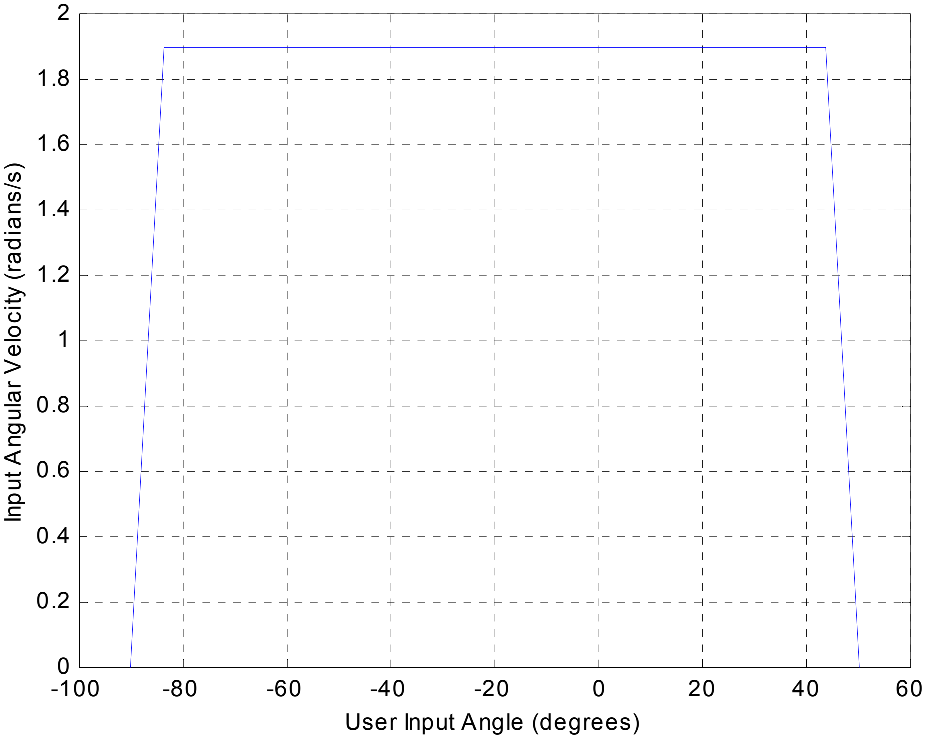


Figure 4-2 Velocity profile of the exercise.

One more assumption must be made before the synthesis can begin. The length of the user input link must be selected. To best maintain the normalcy of the resistance force to the user's forearm, the user-input link needs to be approximately the length as the distance between the users elbow and the middle of the users palm. Using the average hand length of 7.6 inches and the average elbow to fingertip length of 19.0 inches, a user input arm length of 15 inches was taken as approximately average (Woodson et al, 1992).

Section 4.2 Implementation of Synthesis Routine

The actual process of solving the synthesis problem was done on a computer. The large numbers of calculations need to perform the optimization precludes other solution methods. All of the kinematic, dynamic and optimization equations and routines were programmed into The Math Works technical computing language, Matlab^{TM}. Matlab provides the user with a higher level programming environment with many built-in functions that are designed specifically with engineering computational needs in mind.

The programming consisted of creating a main program and two separate subroutines. The main program contains the code for the Hooke and Jeeves optimization routine (Figure 3-1). As this program runs, it repeatedly calls the first sub-routine, which contains the kinematic and dynamic analysis as well as the objective function. The output of this subroutine to the main program is a single value for the objective function. This output allows the main program to determine the move direction and duration. This output is often displayed so that the user may track the progress of the optimization routine. The main program calls the second routine only once per run. This subroutine contains all of the code from the first sub-routine plus code to create plots of various linkage properties, such as the resistance curve, position, velocity and acceleration data. The sub-routine also outputs to the display the final value of the objective function. This sub-routine can also be easily manipulated to provide any additional outputs that may be desired. The final values for all of the design variables are output to display by the main program. These programs are included in Appendix A.

The main program and the subroutines purposely lack graphical user interfaces (GUIs). The programs are designed to be used, not as a "black box" where one simply inputs a couple of parameters and lets in run, but as a tool for a knowledgeable kinematic designer. The user needs to have some understanding of kinematic synthesis and numerical optimization to implement the changes to the program that are required for the unique aspects of the specific problem one is trying to solve. For example, each problem will almost certainly require changes to the objective function and will require various initial design variable values. These changes must be made by the user. One also can easily fall into believing the results of a program without questioning them when one takes the "black box" viewpoint. For this reason, any results need to receive a "does this make sense" check from the user.

Another reason for the lack of GUIs is to keep the program as simple as possible. As stated above, the user will need to make changes to the program. Adding GUIs complicates the program and making changes becomes more difficult. In addition, keeping the program simple makes the program smaller and quicker to run. Also, by maintaining the simplicity, the program can more easily be incorporated into other programs as desired. Any GUIs would have to be eliminated or reworked to allow the program to be combined with other programs. This change would take time and expertise that could be better used elsewhere.

Section 4.3 Synthesis Results

When examining the synthesis results, one should recall the goal of this thesis. The goal was develop a synthesis routine to design force generating planar four-bar linkages. In particular, this problem focused on matching the resistance curve of the linkage to the human-strength curve for the bicep-curl exercise. After the synthesis routine was applied to the bicep-curl problem, a satisfactory solution to the problem was found. The values for all of the design variables for the final optimized solution are given in Table 4-2. This linkage had an associated objective function value of 65,611. This value is in and of itself, meaningless, since it is by nature a relative number. The value is only recorded for possible future comparisons with other linkages and objective functions. Please note that this solution is an *optimized solution*, that is the best solution found from an optimization routine, and not necessarily the *optimal solution*, that is the best solution possible.

Table 4-2 Final design variable values. In units of feet, radians, and slugs.

Variable	$R_1 + R_2$	R_3	R_4	R_5	R_6	R_7	β	γ	M
Value	1.4	0.744	2.435	1.403	0.656	1.838	1.680	-0.100	10.04

Drawings of the linkage in its starting and final positions are shown in Figure 4-3 and Figure 4-4, respectively. The circle at the end of the one link represents the resistance mass, and as such, it clarifies which link is the output and which is the user input links.

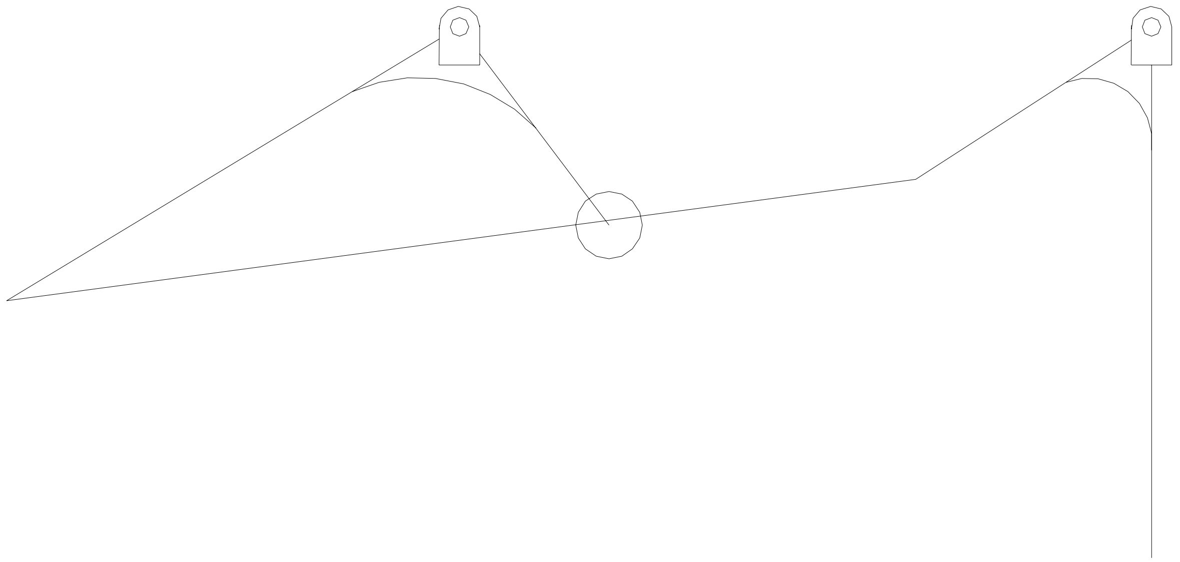


Figure 4-3 Optimized bicep-curl exercise mechanism in the initial position.

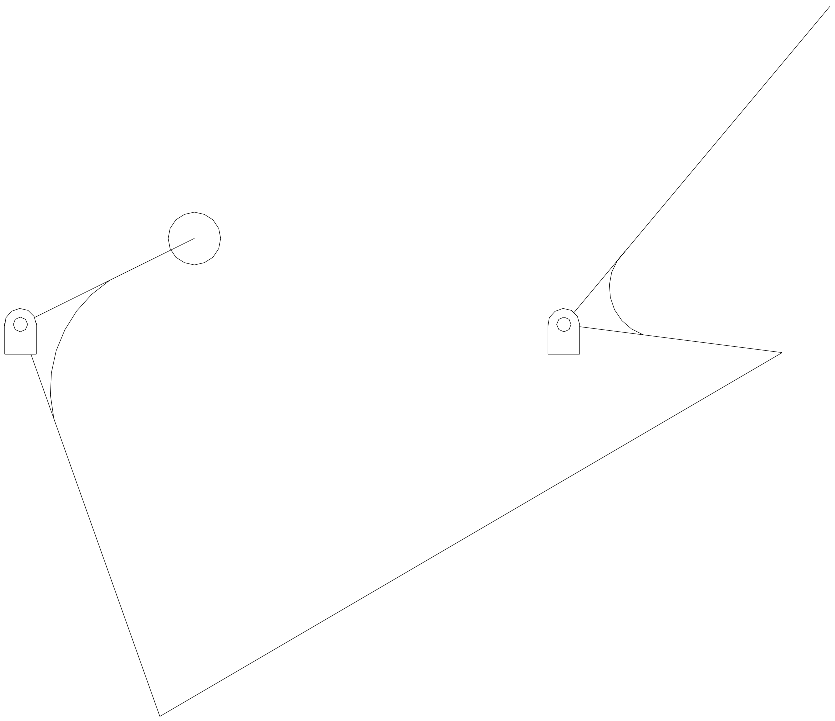


Figure 4-4 Optimized bicep-curl exercise mechanism in the final position.

The final linkage design produces the resistance curve shown in Figure 4-5. While the resistance curve is not an exact match, it is close enough for the mechanism to be usable in its desired role in a weightlifting machine. Where the quality of the match may be a concern is in the regions where the resistance curve is greater than the strength curve since this will cause the curl motion to stall. Since the user strength cannot be increased to bring the entire resistance curve under the strength curve. The resistance curve must be shifted down below the strength curve. This shift can be easily accomplished without distorting the curve by reducing the resistance mass. Reducing the mass will reduce the maximum resistance, but the overall effect of matching the resistance curve to the strength curve is what is most desired and that effect will not be significantly affected by this change.

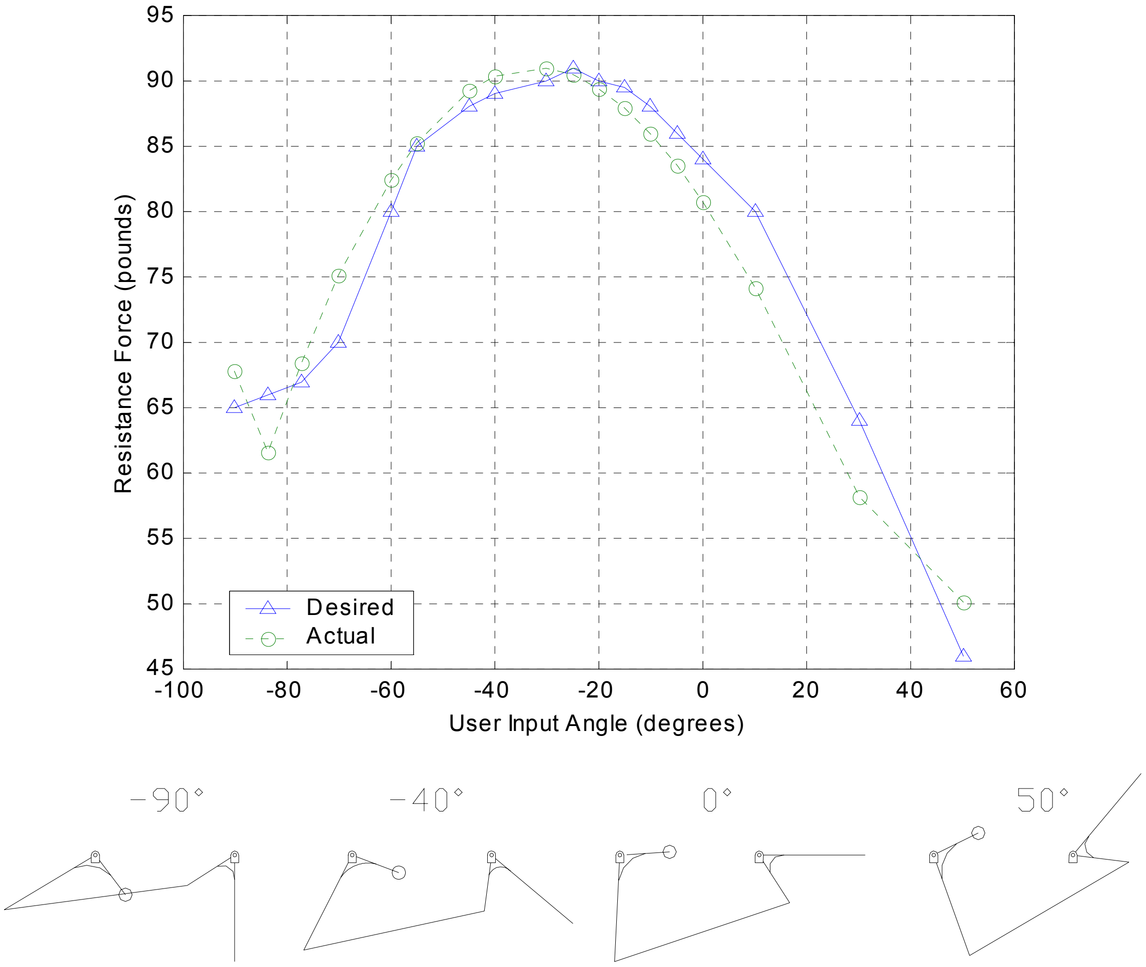


Figure 4-5 Resistance curve of the final optimized solution.

To better understand force generating synthesis, a breakdown of the effects on the force curve of the three main factors, static loading, dynamic loading and kinematics, would be useful.

The static loading in this problem is developed from the weight of the resistance mass. The moment generated by the resistance mass is function of the magnitude of the mass and the length of the moment arm. The moment arm length consists of the length of R_6 and its angle relative to weight vector. This angle is the only component of the static loading that changes during the exercise, which affects the shape of the resistance curve. For this reason, the deviation of this angle from the maximum moment angle of 90° has been plotted in Figure 4-6. One can see that the static moment increases almost linearly to a peak at –5° and then decreases with the same linear relationship. This peak is 25° out-of-phase with the resistance curve. Also, since the resistance curve is not perfectly symmetrical, the changing static moment cannot be solely responsible for the shape of the resistance curve. This fact indicates that the other two force factors, dynamic loading and kinematics, play an important role in the determination of the resistance force.

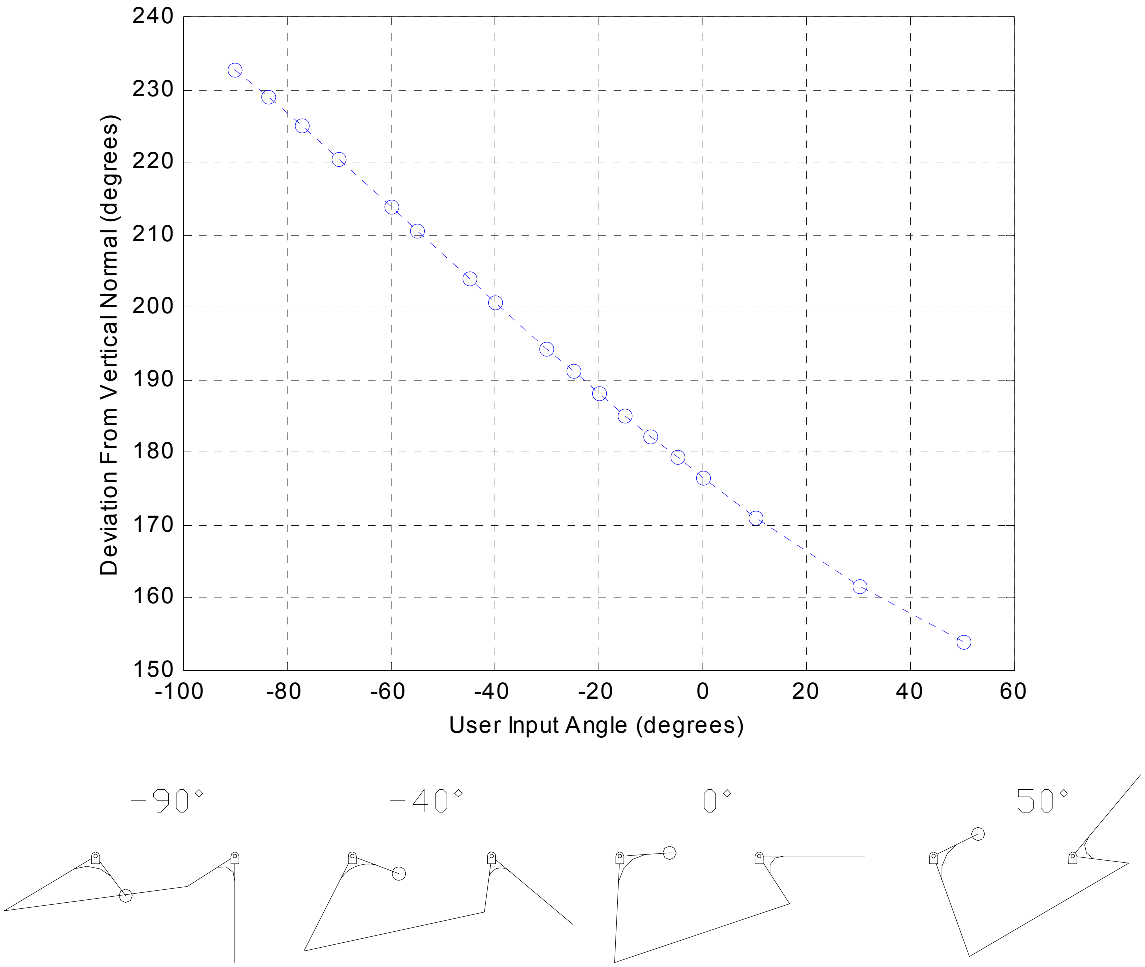


Figure 4-6 Relative static moment angle deviation during the exercise.

The dynamic loading is determined by the acceleration of the mass, so an examination of the acceleration will provide the insight that is needed into the effects of the dynamic loading. The angular acceleration for the coupler (α_4) and output (α_5) links is plotted in Figure 4-7.

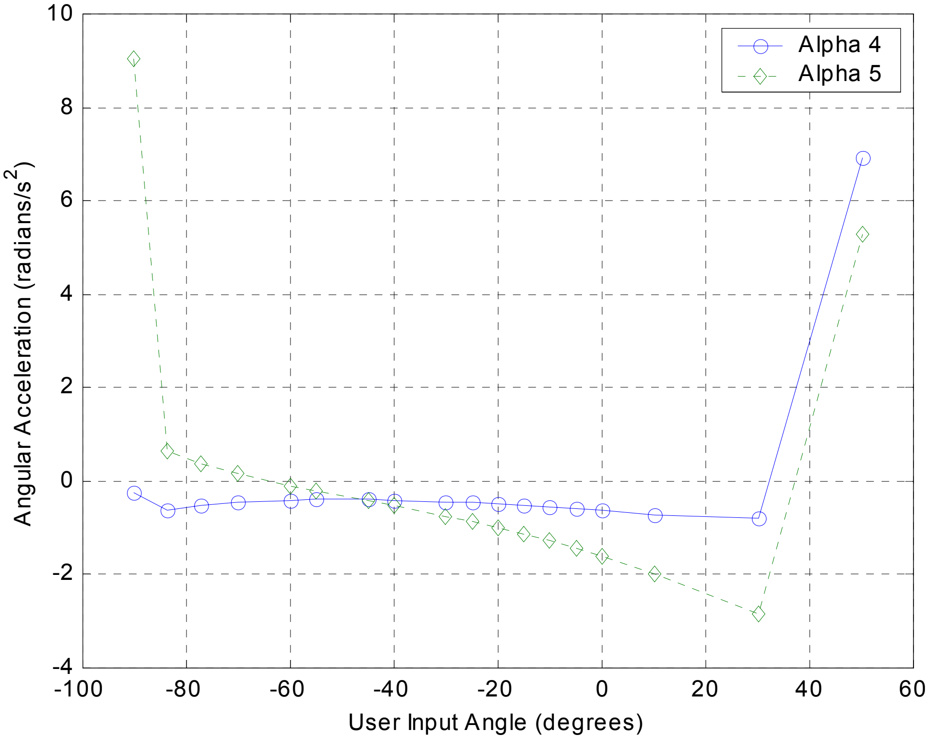


Figure 4-7 Angular accelerations for the coupler and output links.

Comparing the shape of the output acceleration curve (Figure 4-7) with the resistance curve (Figure 4-5), we see that the shapes are not the same. However, we see that the acceleration curve for the resistance mass is not symmetrical about the –30° line. These facts indicate that the acceleration of the mass does affect the resistance curve.

To obtain a complete picture of how the resistance is generated, the last force component, kinematics, needs to be examined. A kinematic property of interest is the transmission angle. The transmission angle is the angle between the coupler link and the output link (Mabie and Reinholtz, 1987). This angle indicates the efficiency of the force transfer between the output link and the coupler link. As the transmission angle deviates from 90°, more of the force is directed along the link and into the ground, rather than moving the resistance mass. The result is that more force must be applied by the input link to achieve the same motion of the resistance mass. The transmission angle is plotted in Figure 4-8.

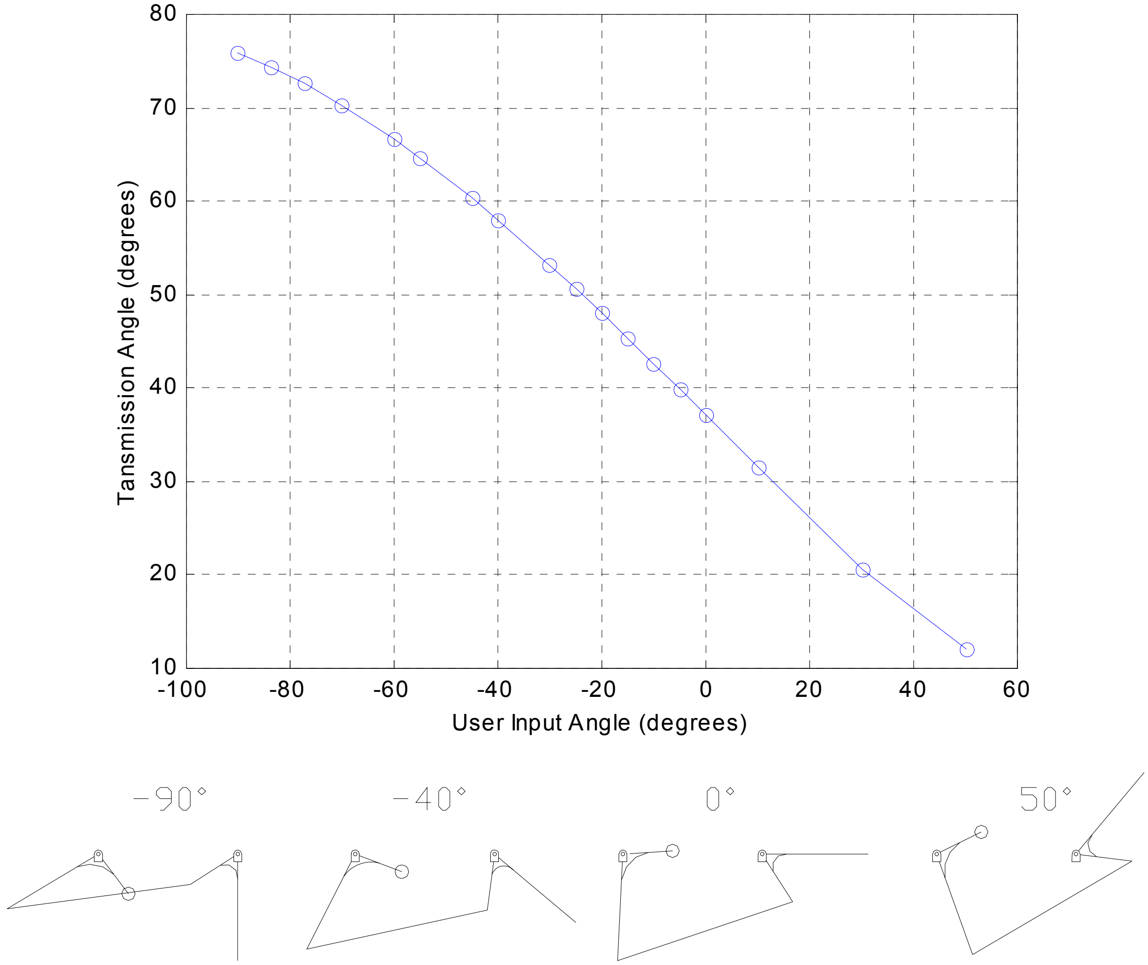


Figure 4-8 Variation of the transmission angle during the bicep-curl exercise.

Examining all three of the force components shows that a combination of the three produce the desired resistance curve. None of the three can be ignored or simplified out of the problem without impacting the solution. How much impact do these factors have on the solution? To answer this question, the resistance curve produced by the design linkage when only static effects are considered was plotted with the resistance curve for when the dynamic effects are also included. This comparison is shown in Figure 4-9. The figure shows that the static forces completely dominate over most of the exercise. However, by comparing this figure with the acceleration plot in Figure 4-7 one sees that where the combined effects curve deviates from the static curve is in the same regions where angular acceleration of the output link is not near zero. This attribute is especially true at the end points, where the user applied acceleration causes large output link accelerations. The conclusion that can be made is that when a problem involves an input acceleration, one needs to include the dynamics. In cases where there is no input acceleration, the acceleration of the output link needs to be checked. If the output acceleration is near zero, then one may ignore the dynamic effects, but if the output acceleration is not near zero, one should include the dynamic effects.

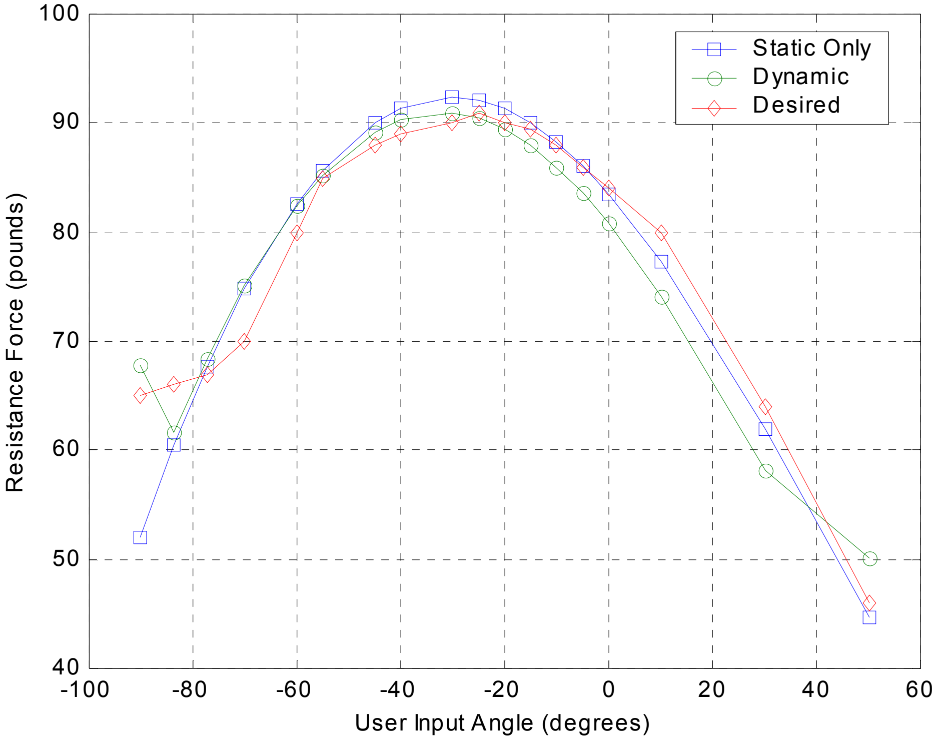


Figure 4-9 Contributions of the static and dynamic force components.

Due to uncertainty in the exact manner in which the weightlifting machine will be used, the linkage needs to be analyzed for its performance in a range of situations. In Figure 4-10, the resistance curve for the linkage given a ±50% variation in the user input velocity from the nominal design velocity. The general trend is as the velocity increases the resistance force decreases. This decrease is caused by the acceleration increasing in magnitude, but the acceleration is negative, thus reducing the resistance force. The starting point experiences the greatest change. The system must accelerate at a greater to reach the higher velocities since the acceleration region is fixed. This assumption in the velocity profile is clearly not accurate over the span of velocities. Therefore, the end point forces should not be taken as exact. This graph also illustrates the importance of dynamics when calculating the resistance force, because the change in the resistance curve in Figure 4-10 is due solely to the dynamics of the system, since gravity and the mass are not changing.

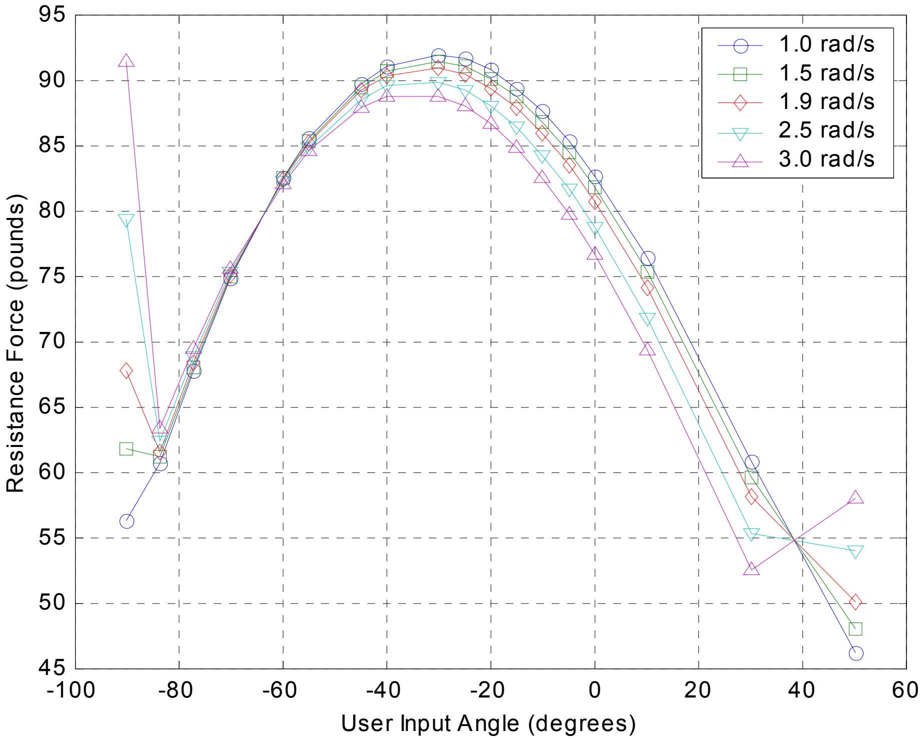


Figure 4-10 Effects of input velocity on the resistance force curve.

Two other usage considerations that exist are possible variations in the user input link length and variations in the magnitude of the resistance mass. The need for a longer or shorter input arm length is due to the natural differences in user body size. A higher or lower resistance force than was used in the initial design is expected since the magnitude of the strength of each user is different. The equations in Section 2.2 show that the resistance force is linear with respect to these two variables. The affects of changing mass and user input link length are shown in Figure 4-11.

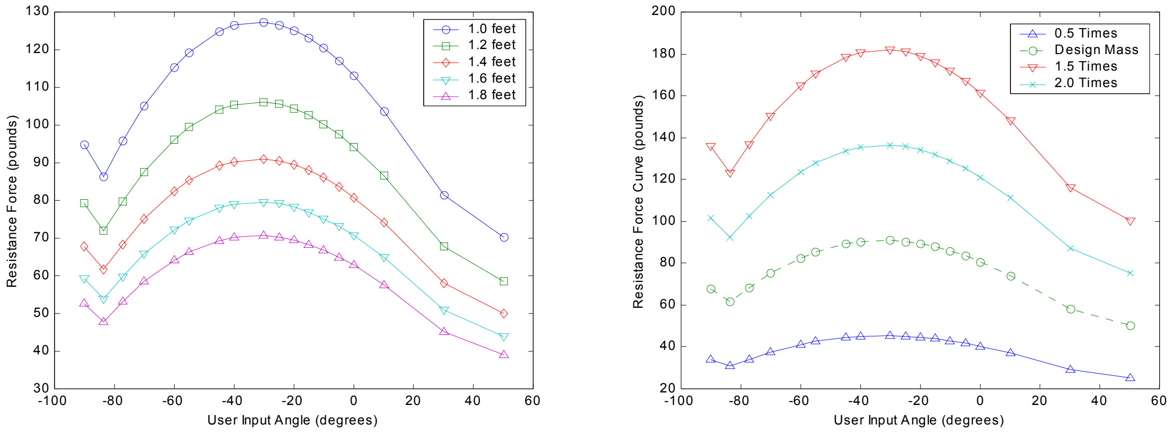


Figure 4-11 Resistance force as a function of user input length (left) and of the load mass.

The resistance mass is more important than just its effect on the resistance curve. Controlling the amplitude of the mass was a secondary objective derived from the practical desire to have a resistance mass of the same order of magnitude as the peak resistance. The resistance mass of the final design was around 10.0 slugs or 322 pounds. Since this resistance mass produces a peak resistance of 91 pounds, the mass is of an unacceptable magnitude. This criteria was not incorporated into the final objective function since the restrictions on the solution space was found to be too stringent, and that they needed to be relaxed so that a solution with an acceptable resistance curve could be found.

The angular jerk of the mechanism is an important property from the stand point of user safety and comfort. If the jerk is too high, the exercise motion will be uncomfortable for the user. The angular jerk of the coupler and output links are shown in Figure 4-12. The jerk curve of the output link is small in magnitude for most of the curve. Near the end points the jerk is higher since the end points experience the user-applied accelerations.

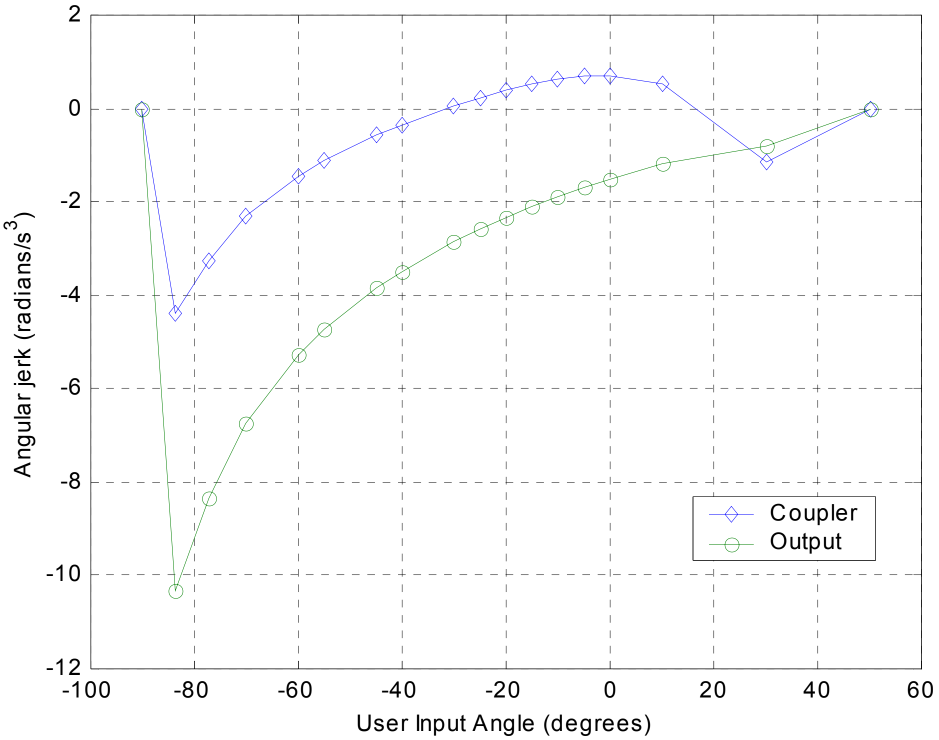


Figure 4-12 Angular jerk for the coupler and output links.

The sensitivity of the system to small changes in the design variables needs to be examined since there is always uncertainty in the dimensions of any real object. A low sensitivity to changes in the design variables is desired. A design that exhibits this behavior is said to be robust. Each of the design variables in this problem were subjected to a ±1% change in value. The resistance curves for these changes were plotted in Figure 4-13 for R_3 and R_4, Figure 4-14 for R_5 and R_6, Figure 4-15 for β and γ, and Figure 4-16 for R_7. As these figures show, the design is generally insensitive with the exception of the link changes that cause the acceleration at the starting position to increase or decrease. These acceleration changes cause the resistance force to change significantly at this point. Only the cases where the force increases is there a concern, since the increase takes the resistance force far above the strength curve.

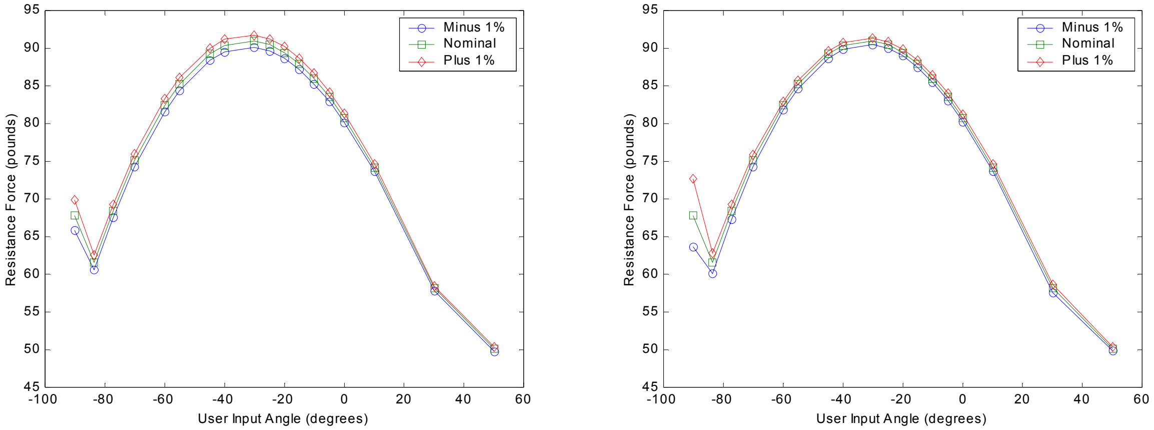


Figure 4-13 Sensitivity plots for R_3 (left) and R_4 (right).

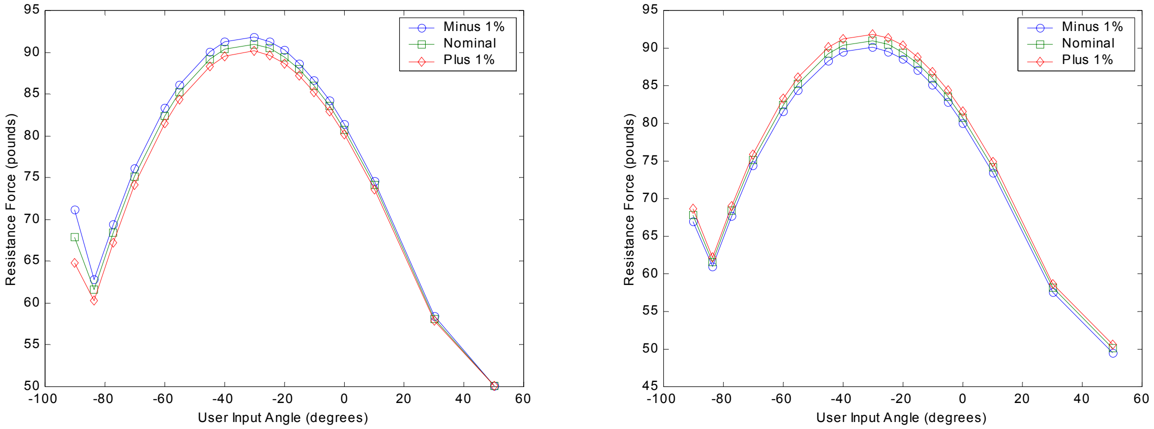


Figure 4-14 Sensitivity plots for R_5 (left) and R_6 (right)

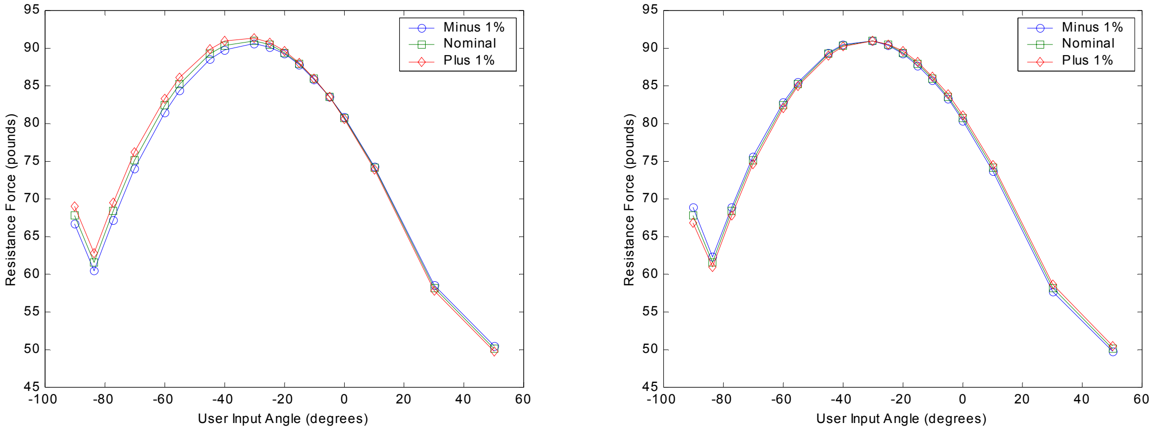


Figure 4-15 Sensitivity plots for the weight offset angle (l) and the user input angle (r)

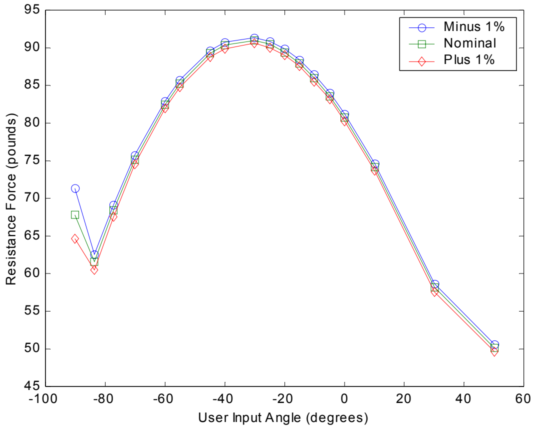


Figure 4-16 Sensitivity plot for the ground link (R_7)

Chapter 5 Example Problem 2: Bicep-curl Revisited

Section 5.1 Purpose of Re-examining this Problem

This example problem deals with the same bicep-curl exercise as in the first example problem (Chapter 4), except that the dynamic model of the mechanism has changed. In this problem, the links that make up the mechanism have their mass modeled as point masses located at the ends of each link, producing a center of mass at the midpoint of each link. The dynamics of this model was discussed in Section 2.3. While this mass model is not exact, it demonstrates that the real links will have dynamic properties and that these properties give rise to a more complex force generation problem. Since the inclusion of the dynamic properties of the links makes for a more difficult problem, the question is logically raised; do we really need to include them? For an answer, let us examine the resistance curve of the mechanism synthesized in Chapter 4 for the original single-mass case, and the case of all the links having mass. The two resistance curves are plotted in Figure 5-1. The two resistance curves, while similar, are not similar. This difference indicates that the link masses can be important factors when designing for force generation. The link mass coefficients (a_i's) for this example problem were given the value of 0.03 slugs per foot. This value means that a 1 ft. long link would weigh a, 1.9 lbs. For comparison, a foot of round steel tubing of 1.5 inch diameter and 1/8 inch wall weights 1.8 lbs.

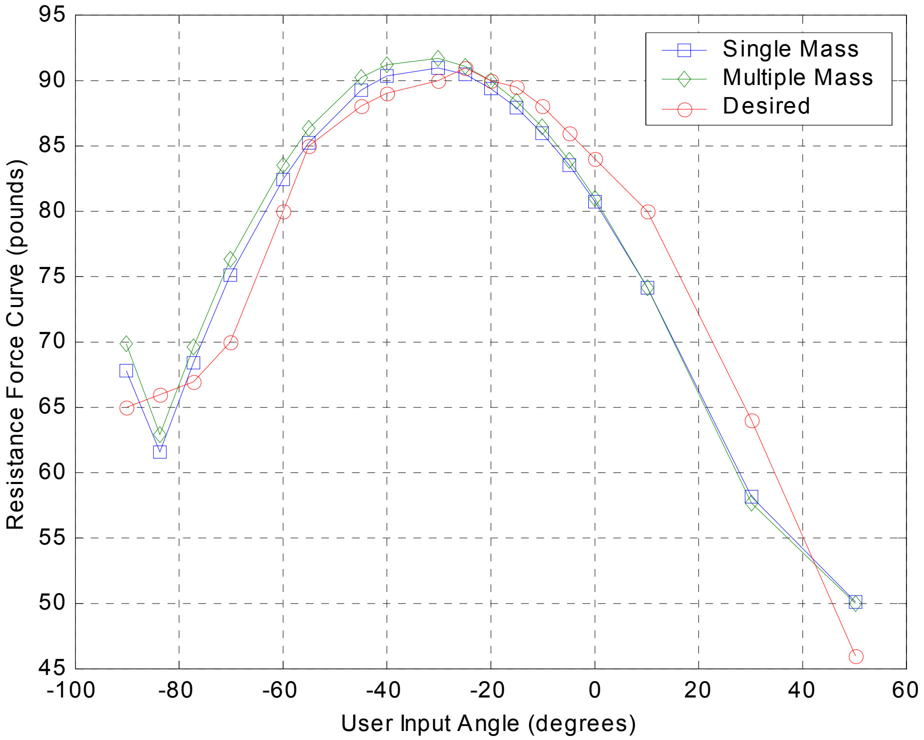


Figure 5-1 Comparison of the dynamic forces for the single and multiple mass systems.

Since the resistance-force curve changes with the addition of link masses, the linkage will need to be resynthesized for the case where the links have mass. The synthesis routine will be reworked to account for the dynamics of the links. A new optimized solution will then be found using this reworked dynamic analysis. The implementation of the improved synthesis routine will be the same as for the original routine. The analysis equations and the optimization routine will be programmed in Matlab and run until a satisfactory solution is found. Then the results will be output, analyzed and plotted.

Section 5.2 Synthesis Results

When examining the synthesis results, one should recall the goal of this thesis. The goal was develop a synthesis routine to design force generating planar four-bar linkages. In particular, this problem focused on matching the resistance curve of the linkage to the human strength curve for the bicep-curl exercise. After the synthesis routine was applied to this more complete version of the bicep-curl problem, an acceptable solution to the problem was found. The values for all of the design variables for the final optimized mechanism are given in Table 5-1. This linkage had an associated objective function value of 160,100. This value is in and of itself, meaningless, since it is by nature a relative number. The value is only recorded for possible future comparisons with other linkages and objective functions. Please note that this solution is an *optimized solution*, that is the best solution found from an optimization routine, and not necessarily the *optimal solution*, that is the best solution possible.

Table 5-1 Final design variable values. In units of feet, radians, and slugs.

Variable	$R_1 + R_2$	R_3	R_4	R_5	R_6	R_7	β	γ	M
Value	1.4	0.636	1.215	0.856	0.363	0.993	1.806	-1.307	14.26

Drawings of the linkage in its starting and final positions are shown in Figure 5-2 and Figure 5-3 respectively. The circle at the end of the one link represents the resistance mass and, as such, it clarifies which link is the output and which is the user input links. While the peak resistance force is 91 pounds, the resistance mass weights 459 pounds. This weight to peak resistance force advantage is not desirable from a practical viewpoint. Just looking at the resistance mass can be misleading since the links themselves provide an additional 8.0 pounds of the resistance mass, so that the overall there is 467 pounds of mass in the linkage to produce the 91 pounds of peak resistance force.

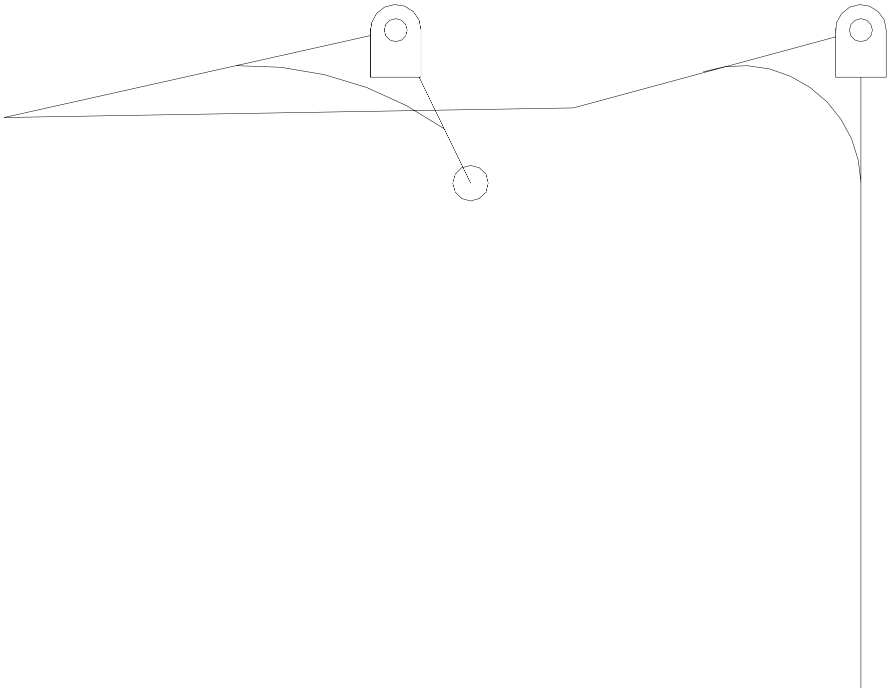


Figure 5-2 Final multiple-mass design drawn in the initial position.

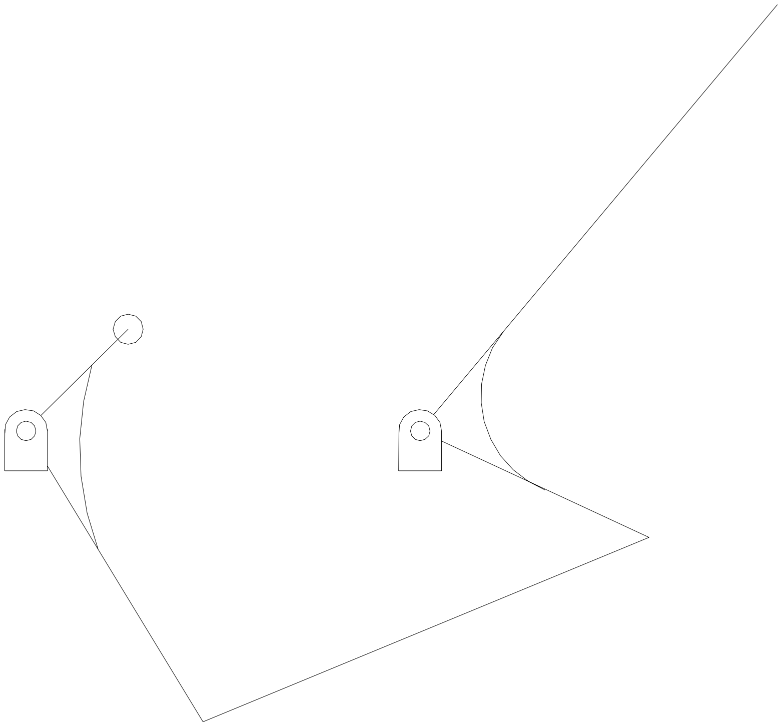


Figure 5-3 Final multiple-mass design drawn in the final position.

The resistance force curve for the final solution is plotted with the human strength curve in Figure 5-4. One quickly notices that the resistance force curve in this problem is nearly as good of a match to the strength curve as the single-mass system curve is. This result is unexpected. The multiple-mass system has the same number of design variables as the single-mass system, but multiple-mass system has more force generating components (link mass and inertia) that the linkage design must handle. This arrangement creates an expectation that one may not be able to control the performance of the design of the multiple-mass system to the degree possible with the single-mass system. The performance of the synthesis routine in dealing with the multiple-mass system is, thus, satisfactory.

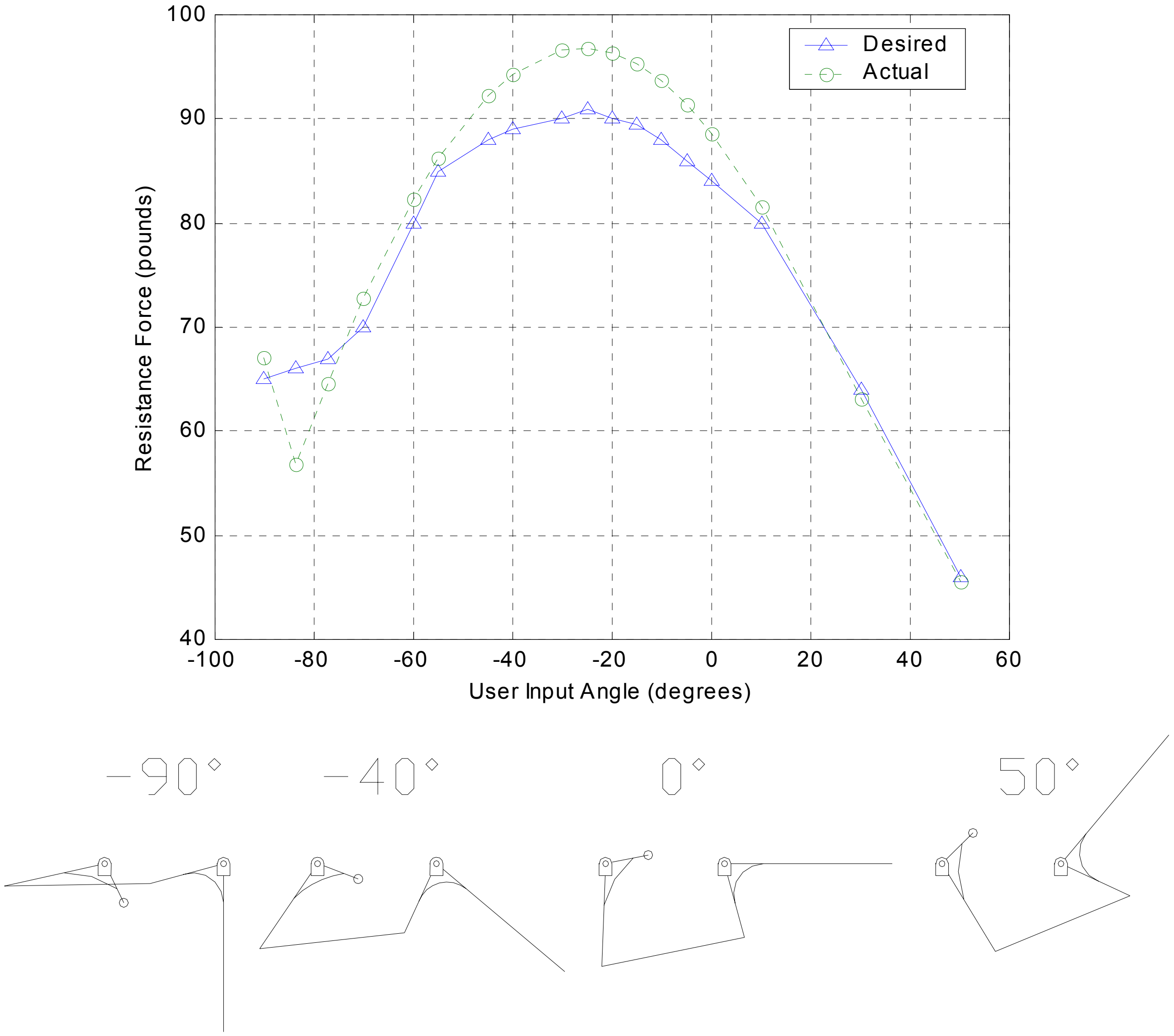


Figure 5-4 Resistance curve for the final multiple-mass solution.

From examining the graph in Figure 5-4, a few interesting observations can be made. The resistance curve in the graph does not match the strength curve well in the rise section. This mismatch can also be seen in the single mass system resistance curve, Figure 4-5. In both cases; the linkage is unable to produce a concave rise in the resistance curve that would be able to match the strength curve in this region. However, both linkages have little difficulty in matching the roughly linear fall in the strength curve. One would logically expect to find a linkage with the opposite trait, being able to match the rise, but not the fall. The optimization routine did not produce such a design with a similar objective function value.

In the region around the peak resistance force, the resistance force curve is greater than the strength curve. This difference will cause a problem when the exercise is performed. The user will not be able to overcome the resistance in this region and the exercise motion will stall. To correct for this problem, the mass can be reduced. The resistance mass was reduced by 6%, and the new resistance curve was plotted as shown in Figure 5-5. While this new curve solves the problem of the excessive resistance in the peak region, the resistance curve does not match the rest of the strength curve as well.

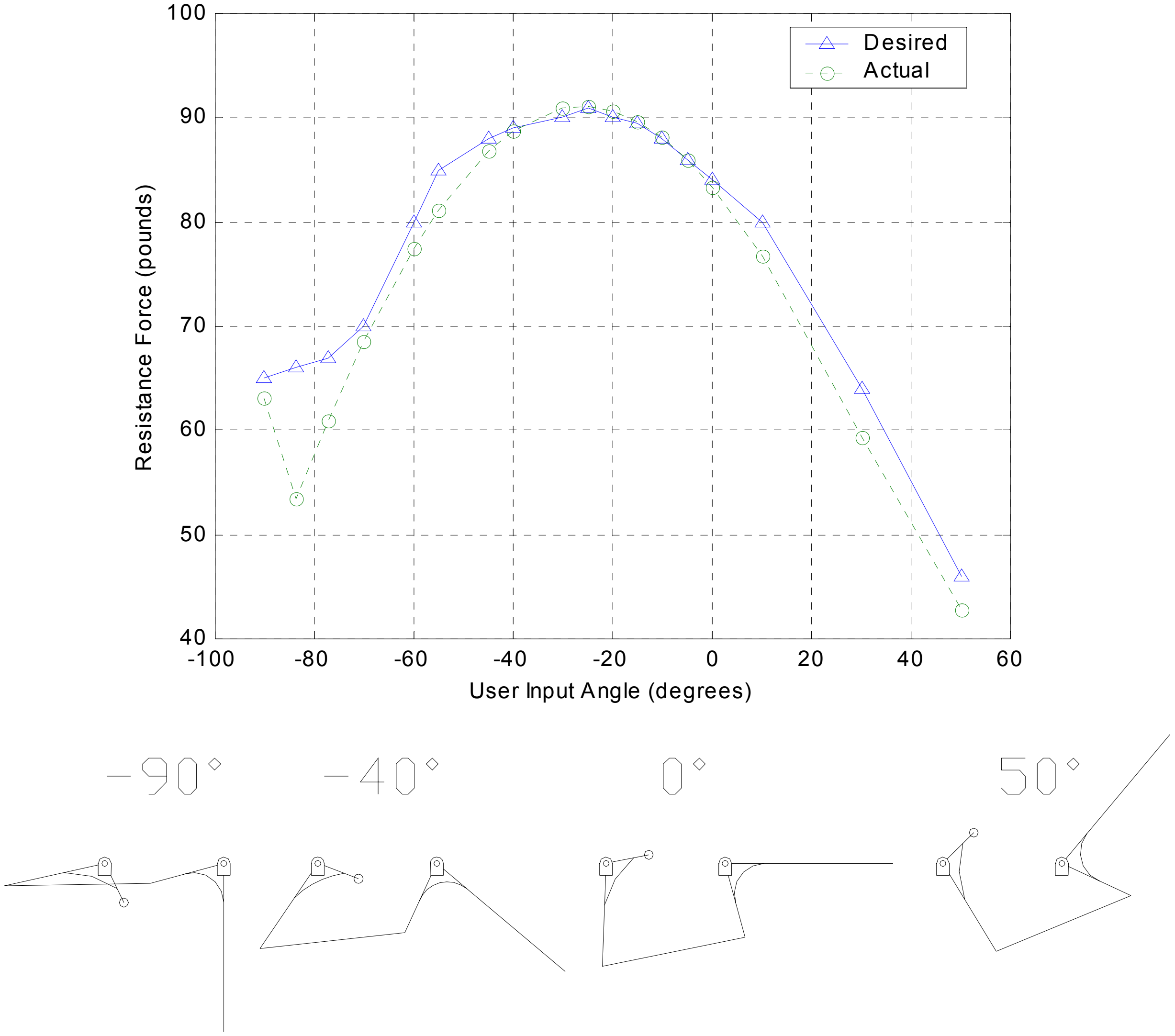


Figure 5-5 Resistance curve for the mass adjusted case.

There are four components involved in producing the resistance force; the static and dynamic effects of the resistance mass, and the static and dynamic effects of the links. In Section 4.3, a comparison of the static and dynamic effects of the resistance mass showed that the dynamic effects should be included for the problem of the bicep-curl exercise. How much impact do the links have on the solution? To answer this question, the resistance curves produced by the design linkage when only resistance mass effects are considered, and when only the static effects of the links are considered were plotted with the resistance curve for when the all of the effects are included. This comparison is shown in Figure 5-6. The figure shows that the links do have a significant effect on the resistance curve. While most of the effect is due to the static link effects, the dynamic effects have an impact similar to what was seen in the single mass system.

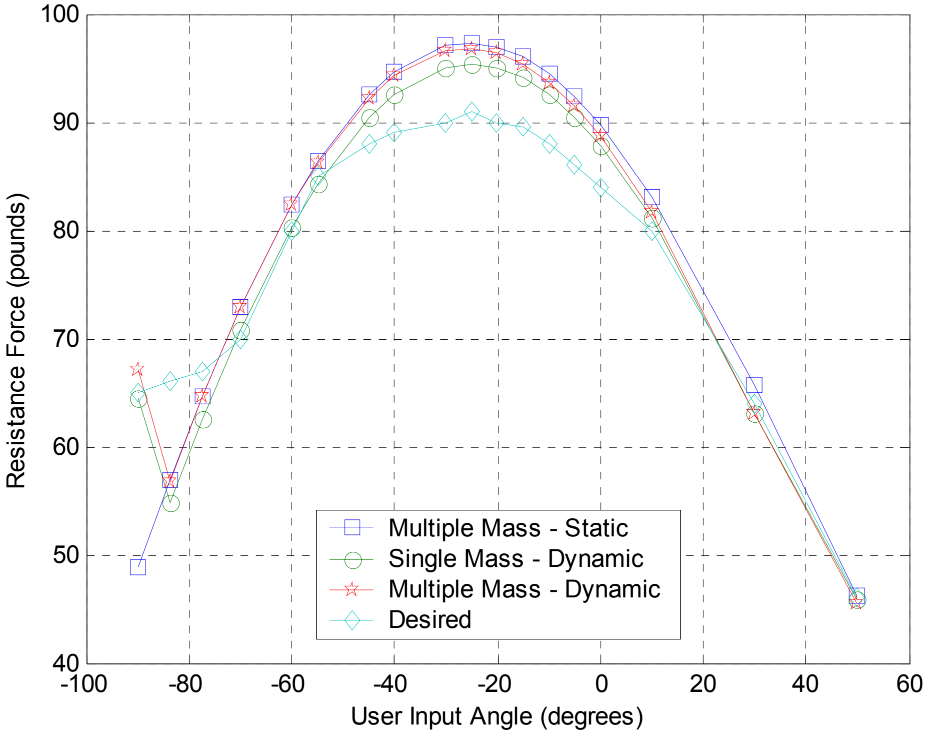


Figure 5-6 The effect of the links on the resistance curve.

The angular jerk of the mechanism is an important property from the standpoint of user safety and comfort. If the jerk is too high, the exercise motion will be uncomfortable for the user. The angular jerk of the coupler and output links are shown in Figure 5-7. The jerk curve of the output link is small in magnitude for most of the curve. Near the end points the jerk is higher since the end points experience the user-applied accelerations.

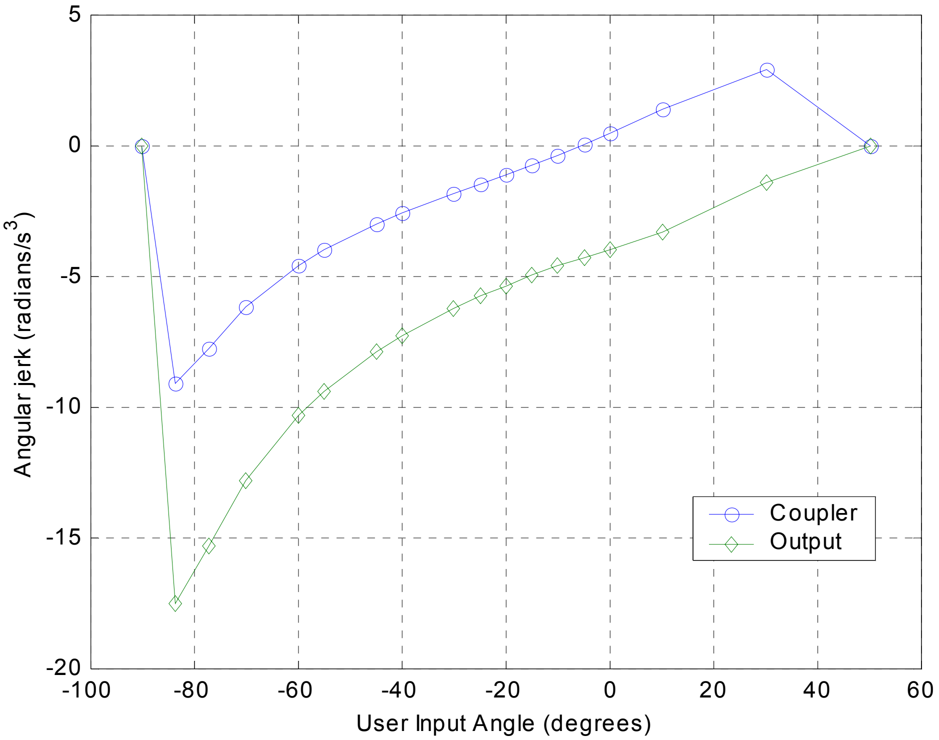


Figure 5-7 Angular jerk for the coupler and output links.

Due to uncertainty in the exact manner in which the weightlifting machine will be used, the linkage needs to be analyzed for its performance in a range of situations. In Figure 4-10, the resistance curve for the linkage given a ±50% variation in the user input velocity from the nominal design velocity. The general trend is as the velocity increases the resistance force decreases. This decrease is caused by the acceleration increasing in magnitude, but the acceleration is negative, thus reducing the resistance force. The starting point experiences the greatest change. The system must accelerate at a greater to reach the higher velocities since the acceleration region is fixed. This assumption in the velocity profile is clearly not accurate over the span of velocities. Therefore, the end point forces should not be taken as exact. This graph also illustrates the importance of dynamics when calculating the resistance force, because the change in the resistance curve in Figure 5-8 is due solely to the dynamics of the system, since gravity and the masses are not changing.

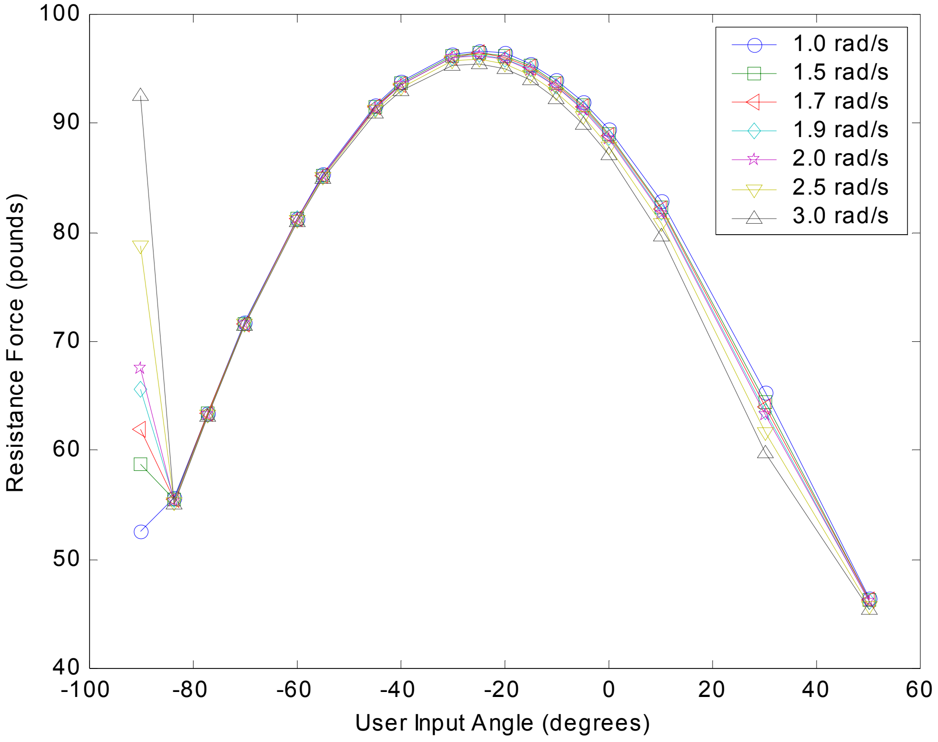


Figure 5-8 Resistance force as a function of user input velocity.

Two other usage considerations that exist are possible variations in the user input link length and variations in the magnitude of the resistance mass. The need for a longer or shorter input arm length is due to the natural differences in user body size. A higher or lower resistance force than was used in the initial design is expected since the magnitude of the strength of each user is different. The affects of changing mass and user input link length are shown in Figure 5-9. The variation in resistance force is linear with respect to resistance mass. As expected, the resistance force is almost linear with varying user input link lengths. A slight nonlinearity exists because the mass of the user input link varies with its length, but the resistance generated by this mass is quite small.

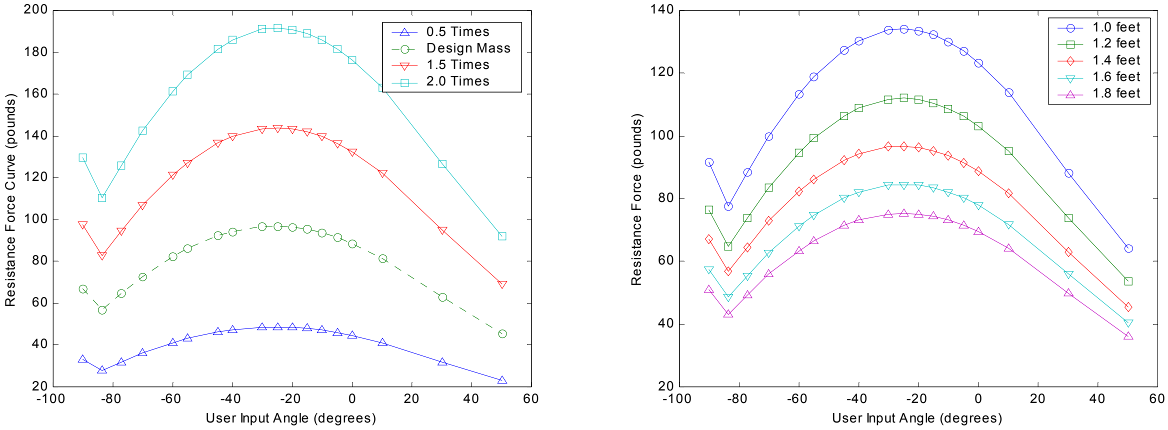


Figure 5-9 Resistance force as a function of user input length (left) and of the load mass.

The last aspect of the linkage that should be examined is how tolerant the linkage is to changes in the mass of the links. This property is important because the link mass coefficients, a_i, was assumed to be 0.03 slugs/ft, but there is a great deal of uncertainty in this value. The resistance force curve was plotted for a range of values for the a_i's, with all other factors held constant and is shown in Figure 5-10. While the resistance force increase with increasing a_i's, the impact on the shape of the resistance curve is minimal. This result means that the mechanism can tolerate a range of linkage designs without needing to be re-synthesized.

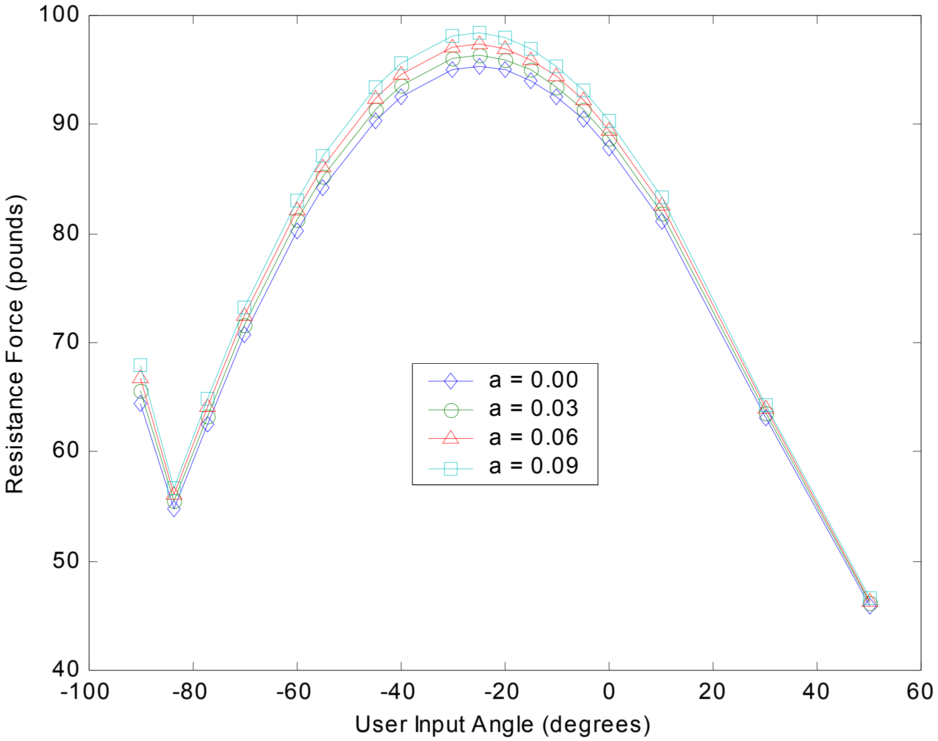


Figure 5-10 Effects of the link masses on the resistance-force curve.

The sensitivity of the system to small changes in the design variables needs to be examined since there is always uncertainty in the dimensions of any real object. A low sensitivity to changes in the design variables is desired. A design that exhibits this behavior is said to be robust. Each of the design variables in this problem were subjected to a ±1% change in value. The resistance curves for these changes were plotted in Figure 5-11 for R_3 and R_4, Figure 5-12 for R_5 and R_6, Figure 5-13 for β and γ, and Figure 5-14 for R_7. As these figures show, the design is generally insensitive with the exception of the link changes that cause the acceleration at the starting position to increase or decrease. These acceleration changes cause the resistance force to change significantly at this point. Only the cases where the force increases is there a concern, since the increase takes the resistance force far above the strength curve.

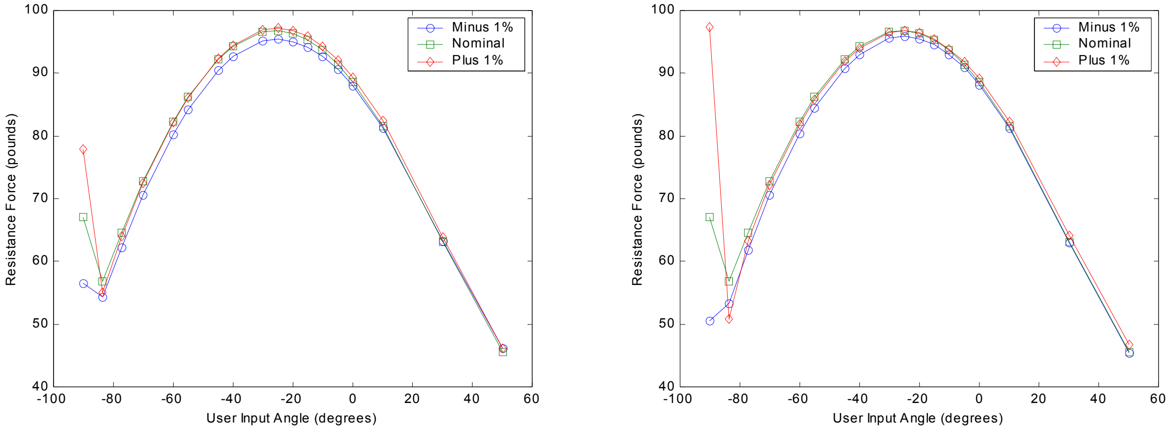


Figure 5-11 Sensitivity plots for R_3 (left) and R_4 (right).

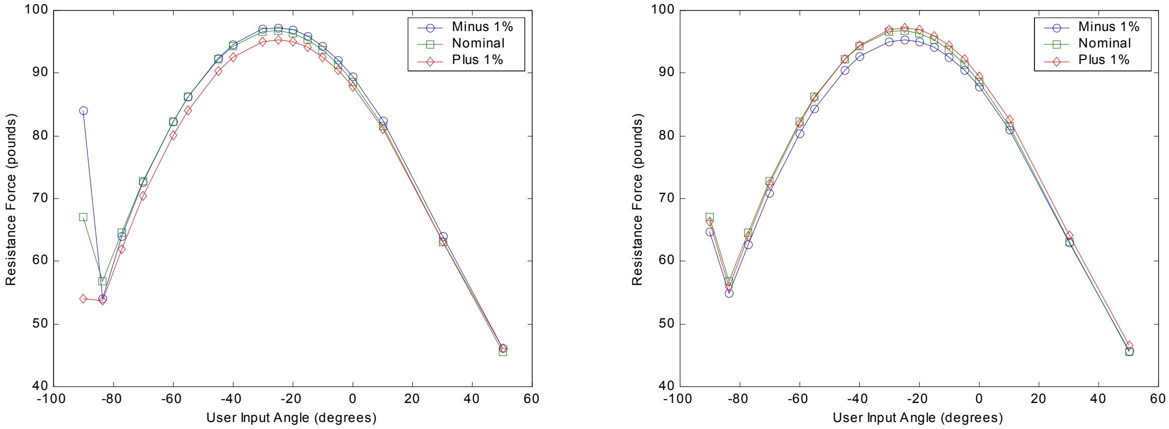


Figure 5-12 Sensitivity plots for R_5 (left) and R_6 (right).

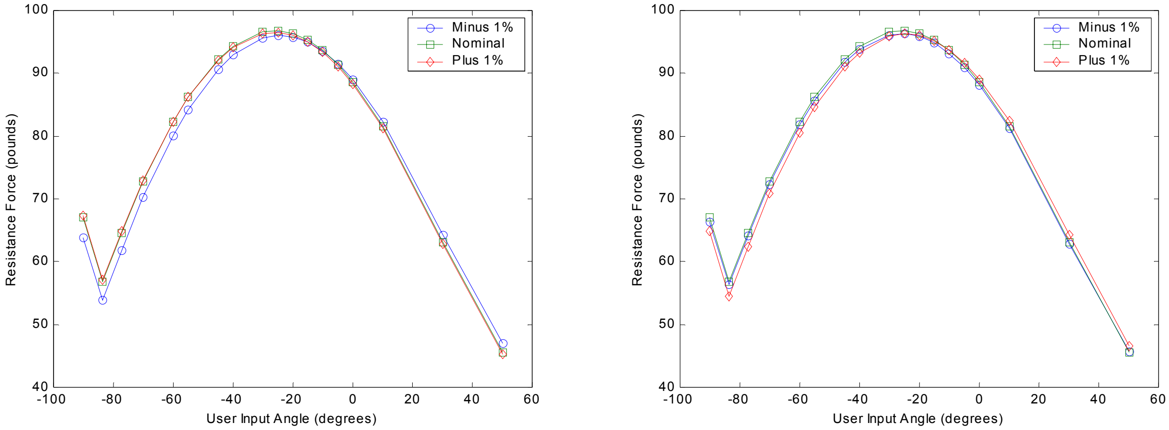


Figure 5-13 Sensitivity plots for weight offset angle (left) and user input angle (right)

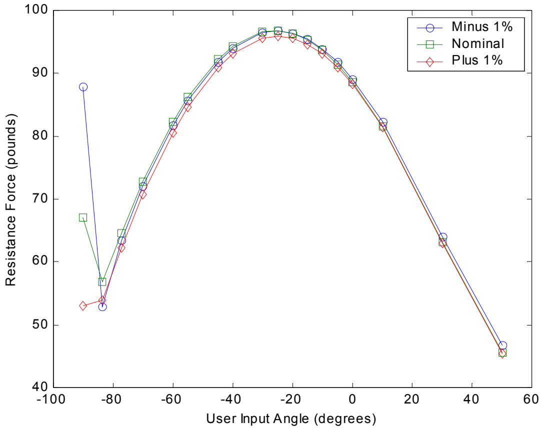


Figure 5-14 Sensitivity plot for the ground link (R_7).

Chapter 6 Conclusions

Section 6.1 Summary

As demonstrated in Chapter 1, there exists a need to develop a linkage-synthesis routine that incorporates the dynamics of the linkage and provides the designer with the ability to handle large numbers of design points. In addition, this need was shown to best be satisfied by a routine that incorporates numerical optimization. The shortcomings of existing tools were discussed. Furthermore, an example of an application in need of such a design tool, a bicep-curl weightlifting machine, was presented.

The components of the synthesis routine, the kinematics, the dynamics and the numerical optimization were discussed in detail in Chapter 2 and Chapter 3. The equations and routines developed from these components were incorporated into a number of Matlab^{TM} program files, as discussed in Section 4.2.

The optimized synthesis process did produce two useable linkages that improve the performance of the standard bicep-curl exercise while meeting a range of practical considerations. The two linkages were the solutions to two different definitions of the bicep-curl problem. The first linkage was for a simple model that did not assign any mass properties to the mechanism’s links. Details on this example problem are given in Chapter 4. The second linkage was designed using a model where the links were assumed to have point masses located at the ends of the links and that these masses were linearly dependent on the length of the links. Details on this example problem are given in Chapter 5. As these example problems illustrate, the development of the synthesis routine was successful. Modifications can easily be made to the code included in this thesis to allow for the creation of solution to other application problems. The extent of the modifications will depend on how much the application varies from the example given in this thesis, but the general framework of this synthesis method allows for a wide range of problems to be solved. This general applicability is the strength and purpose behind the development of this tool.

Section 6.2 Future Work

While this thesis presents a meaningful step forward in the field of linkage synthesis, more can be done. The most obvious step is to include the properties of the links. The difficulty here is determining what those properties should be. Clearly, the properties would need to be defined as a function of the link length, but even a rough approximation of these functions would require the inclusion of other areas of machine design, since details regarding the design of the links must be determined. Designing the links would require a stress analysis that dictates the cross section and thickness of the links, which affects the mass and inertia of the links, which then influences the forces experienced by the links and the overall force curve. Using links with more realistic properties would also invalidate the rigid link assumption. The integration of these design areas would result in a total *machine synthesis* method that would provide a powerful, but complex, design tool for solving many problems.

References

1. Bagci, C., and Rieser, G. M., “Optimum Synthesis of Function Generators Involving Derivative Constraints”, *Mechanism and Machine Theory*, 1984, Vol. 19, No. 1, pp. 157-164.
2. Clarke, H. H., Elkins, E. C., Martin, G. M., and Wakim, K. G., “Relationship Between Body Position and the Application of Muscle Power to Movements of the Joints”, *Archives of Physical Medicine and Rehabilitation*, 1950, pp. 81-89.
3. Eason, E. D., and Fenton, R. G., “A Comparison of Numerical Optimization Methods for Engineering Design”, *Journal of Engineering for Industry*, 1974, Vol 96, Series B, No 1, ASME, New York, NY, pp. 196-200
4. Floudas, C. A., and Pardalos, P. M., *Recent Advances in Global Optimization*, Princeton University Press, Princeton, New Jersey, 1992.
5. Friedland, R., *Control System Design: An Introduction to State-Space Methods*, McGraw-Hill, Inc., New York, NY, 1986, pp 337-377.
6. Herz, M., United States Patent #684,688, October 15, 1901.
7. Hooke, R., and Jeeves, T. A., “‘Direct Search’ Solution of Numerical and Statistical Problems”, *Journal of the Association for Computing Machinery*, 1961, Vol. 8, No. 2, pp. 212-229.
8. Mabie, H. H., and Reinholtz, C. F., *Mechanisms and Dynamics of Machinery*, 4th Edition, John Wiley and Sons, New York, NY, 1987.

9. Midha, A., Turcic, D. A., and Bosnik, J. R., “Creativity in the Classroom – A Collection of Case Studies in Linkage Synthesis”, *Mechanism and Machine Theory*, 1984, Vol. 19, No. 1, pp. 25-44.

10. Rao, S. S., *Optimization: Theory and Applications (second edition)*, John Wiley & Sons, New York, NY, 1984.

11. Reinholtz, C. F., *Optimization of Spatial Mechanisms*, Doctorate Dissertation, University of Florida, Gainesville, FL, 1983

12. Rigelman, G. A., and Kramer, S. N., “A Computer-Aided Design Technique for the Synthesis of Planar Four Bar Mechanisms Satisfying Specific Kinematic and Dynamic Conditions”, *Journal of Mechanisms, Transmissions, and Automation in Design,* 1988, Vol. 110, September, pp. 263-268.

13. Scardina, M. T., *Optimal Synthesis of Force-Generating Planar Four-Link Mechanism* MS Thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, October, 1996.

14. Soper, R. R., *Synthesis of Planar Four-Link Mechanisms for Force Generation*, MS Thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, August, 1995.

15. Starr, P. J., “Dynamic Synthesis of Linkages: An Emerging Field”, *ASME Technical Paper 74-DET-64*, 1974.

16. Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design, with Applications*, McGraw-Hill, New York, NY, 1984.

17. Venkataraman, S. C., Kinzel, G. L., Waldron, K. J., “Optimal Synthesis of Four-Bar Linkages for Four Position Rigid Body Guidance with Selective Tolerance

Specifications", *Mechanical Design and Synthesis (ASME DE-Vol. 46), Proceedings of the 22nd Biennial Mechanism Conference*, 1992, Scottsdale, AZ, USA, September 13-16, pp. 651-659.

18. White, M. V. B., United States Patent #217,918, July 29, 1879.
19. Woodson, W. E., Tillman, B., Tillman, P., *Human Factors Design Handbook*, Second Edition, McGraw-Hill, New York, NY, 1992.

Appendix A: Matlab Programs

```
% Hook and Jeeves optimization code
% Brian Rundgren - Thesis Work - Fall 2001
clear all;
close all;

% Define initial parameters
param = [0 0 0 0 0 0 0 0 0];
newparam = zeros(size(param));
q = length(param);
% Initial drop into design space (the first guess for mechanism dimensions)
r1 = 0.7;  % r1 and
r2 = 0.7;  % r2 make up the user input link
r3 = 0.36; % 4-Bar input link
r4 = 6.20; % Coupler link
r5 = 5.60; % Output link
r6 = 8.40; % Weight arm length
r7 = 1.00; % Ground link
mass = 2.00;    % Resistance mass
beta = 1.95;  % Weight offset angle
gama = 0.98;  % Input offset angle

% Initalize the variable param.
param(1) = r1+r2;
param(2) = r7;
param(3) = r3;
param(4) = r4;
param(5) = r5;
param(6) = r6;
param(7) = mass;
param(8) = beta;
param(9) = gama;

% Set step size (these are the starting delta link lengths)
% The first step size = 0 since param 1 is fixed by user
% for this particular application.
StepSize = [0 0.1 0.1 0.1 0.1 0.1 0.1 0.01 0.01];
%  Compute current cost function
%  and check parameters
CurrentCost = OF2(param);
% Maybe not the best programming practice, but this gives a starting values
for comparison
OldCost = 1000000;
CostDifference = 1;
StepCheck = [0 0.01 0.01 0.01 0.01 0.01 0.001 0.001 0.001];
iteration=0

% The comparison below uses the or command to check to see if the
% cost is still improving and, at the same time, checks to see of the
% if all of the steps sizes are below their preset minimum

while abs(OldCost-CurrentCost) > CostDifference | sum(StepCheck<StepSize)>0

newparam=param;

OldCost = CurrentCost;

% Exploratory Search, the steps have a magnitude (stepsize) and a + or - sense
% determined by the direction vector.  During the exploratory loop below, the
direction may also be
% set to zero to indicate that neither a plus or a minus step in that
direction lowered the cost.

Direction = [0 1 1 1 1 1 1 1 1];
```

```
% Step in each direction and check for reduction in objective function (OF2).
% Direction(j)= 0 if both + and - directions fail to reduce objective function
value
for j=1:q
 newparam(j) = param(j)+StepSize(j)*Direction(j);
 NewCost = OF2(newparam);
   if NewCost > CurrentCost
    Direction(j) = -Direction(j);
    newparam(j) = param(j)+StepSize(j)*Direction(j);
    NewCost = OF2(newparam);
    if NewCost > CurrentCost
      Direction(j) = 0;
      StepSize(j) = StepSize(j)/2;
     end
  end
end

% End of the Exploratory Search
move = StepSize

% Start the Pattern Move
newparam = param + StepSize.*Direction;
newcost = OF2(newparam);

while newcost < CurrentCost
   param = param + move.*Direction;
   CurrentCost = newcost;
   move = move*1.25
   newparam = param + move.*Direction
   param
   newcost = OF2(newparam);

end

iteration=iteration+1;
end

% Export linkage dimensions and call plotting routine.
param
PLT(param)
```

The following program is the first subroutine used in example problem number 1 called OF2.m. The file must be saved with this name or the main program and sub-routine must be edited for the new name.

```
function OFtotal = OF2(param)
% Brian Rundgren - Thesis Work - Fall 2001
% Kinematic and Dynamic Analysis of Four-bar Linkage
% With Objective Function Calculation.


% Input design values.
theta2 = [-1.571 -1.46 -1.35 -1.222 -1.0472 -0.9599 -0.7854 -0.6981 -0.524 -
0.436 -0.3491 -0.2618 -0.175 -0.0873 0 0.175 0.524 0.873];
Fdesired = [65 66 67 70 80 85 88 89 90 91 90 89.5 88 86 84 80 64 46];

value = 1.9;   % Steady state angular velociy.

% Initalize matrices
theta2dot = [0 value value value value value value value value value value
value value value value value value 0];
% The following is the user input angular acceleration. Calculated based on
%  the spacing between the first two design points and the steady state
velocity.
theta2ddot = [value^2/0.22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -value^2/0.22];
theta4dot = zeros(size(theta2dot));
theta5dot = zeros(size(theta2dot));
theta4ddot = zeros(size(theta2dot));
theta5ddot = zeros(size(theta2dot));
F = zeros(size(theta2dot));
TA = zeros(size(theta2dot));
OFi = 0;
indx = 1;
q = 18;

% Position Analysis of Fourbar component.
for indx = 1:q
   theta3(indx) = theta2(indx) + param(9);
   C = param(4)^2-param(2)^2-param(3)^2-param(5)^2-
2*param(2)*param(3)*cos(theta3(indx))+2*param(2)*param(5)+2*param(3)*param(5)
*cos(theta3(indx));
   B = 4*param(3)*param(5)*sin(theta3(indx));
   A = param(4)^2-param(2)^2-param(3)^2-param(5)^2-
2*param(2)*param(3)*cos(theta3(indx))-2*param(2)*param(5)-
2*param(3)*param(5)*cos(theta3(indx));

   % Solve for theta5.
   t1 = (0-B+(B^2-4*A*C)^(0.5))/(2*A);
   theta5(indx) = 2*atan(t1);

   % Imaginary solution penalty (Checks closure).
   if (abs(imag(theta5(indx))))>0
      OFi = OFi + 100000*abs(imag(theta5(indx)))^8+10^30
   end

   % Solve theta4.
   theta4(indx) = asin((param(3)*sin(theta3(indx))-
param(5)*sin(theta5(indx)))/param(4));

   % Imaginary solution penalty (Checks closure).
   if (abs(imag(theta4(indx))))>0
      OFi = OFi + 100000*abs(imag(theta4(indx)))^8+10^30
   end

end

% Velocity Analysis.

indx = 1
for indx = 1:q
   A = param(4)*sin(theta4(indx));
   B = param(5)*sin(theta5(indx));
   C = param(3)*theta2dot(indx)*sin(theta3(indx));
   D = param(4)*cos(theta4(indx));
   E = -param(5)*cos(theta5(indx));
   F = -param(3)*theta2dot(indx)*cos(theta3(indx));
   num4d = ((F*B)-(E*C));
   dend = ((D*B)-(E*A));
   num5d = ((D*C)-(F*A));
   theta4dot(indx) = num4d/dend;
   theta5dot(indx) = num5d/dend;
end

% Acceleration Analysis.
indx = 1;
for indx = 1:q
   A = param(4)*sin(theta4(indx));
   B = param(5)*sin(theta5(indx));
   CP =
param(3)*(theta2dot(indx)^2)*cos(theta3(indx))+param(3)*sin(theta3(indx))*the
ta2ddot(indx)+param(4)*(theta4dot(indx)^2)*cos(theta4(indx))-
param(5)*(theta5dot(indx)^2)*cos(theta5(indx));
   D = param(4)*cos(theta4(indx));
   E = -param(5)*cos(theta5(indx));
   FP = param(3)*(theta2dot(indx)^2)*sin(theta3(indx))-
param(3)*(theta2ddot(indx))*cos(theta3(indx))+param(4)*(theta4dot(indx)^2)*si
n(theta4(indx))-param(5)*(theta5dot(indx)^2)*sin(theta5(indx));
   dendd = (D*B-E*A);
   num4dd = (FP*B)-(E*CP);
   num5dd = (D*CP-FP*A);
   theta4ddot(indx) = num4dd/dendd;
   theta5ddot(indx) = num5dd/dendd;
end

% Dynamic Analysis.

indx = 1;
for indx = 1:q
num1 =
param(7).*(param(6).^2.*theta5ddot(indx)+32.174.*param(6).*cos(theta5(indx)+p
aram(8)));
den1 = param(5).*sin(theta5(indx)).*cos(theta4(indx))-
param(5).*sin(theta4(indx))*cos(theta5(indx));
```

```
F54(indx) = num1./den1;
F(indx) =
(F54(indx)*(cos(theta4(indx)).*param(3).*sin(theta2(indx)+param(9))-
sin(theta4(indx)).*param(3).*cos(theta2(indx)+param(9))))/param(1);
end

% Calculate Transmission Angle.

indx = 1;
for indx = 1:q
   Z = param(2)^2+param(3)^2-2*param(2)*param(3)*cos(theta2(indx)+param(9));
   TA(indx) = acos((Z-param(4)^2-param(5)^2)/(-2*param(4)*param(5)));
end

% Find link length ratio penalty.
indx = 1;
Rmax = 0; Rmin = 1000;
for indx = 1:6
Rmin = min(abs(param(indx)), Rmin);
Rmax = max(abs(param(indx)), Rmax);
end
if Rmax/Rmin >= 15
   OFb = 400 + (Rmax/Rmin)^2;
else
   OFb = 0;
end

% Find penalty for negative link length.
OFc = 0;
indx = 1;
for indx = 1:6
   if param(indx)<0
   OFc = OFc + (100-param(indx))^8;
end
end

% Find penalty for force difference. (Checks quality of fit).
OFa = 0;
diff = 0;
indx = 1;
for indx = 1:q
   diff = 3*abs(100*(Fdesired(indx)-F(indx))/Fdesired(indx))^4+diff;
end
OFa = diff;

% Negative force penalty.
indx = 1;
for indx = 1:q
   if F(indx)<0
      OFa = OFa + abs(F(indx))^4;
   end
end

% Minimize the Magnitude of the Weight Relative to Max Force.
OFm = 0;
OFm = 1000^(param(7)*32.174*10000/max(Fdesired));
```

```
% Sum Penalities and Export Total.
OFtotal = OFa + OFb + OFc + OFi
```

The following program is the second subroutine used in the first example problem called PLT.m. The file must be saved with this name or the main program and subroutine must be edited for the new name. This subroutine contains all of the code from the first subroutine (OF2) plus code to calculate additional properties, to produce graphs and to output numerical data of interest.

```
function OFtotal = PLT(param)
% Brian Rundgren - Thesis Work - Fall 2001
% Kinematic and Dynamic Analysis of Four-bar Linkage
% With Objective Function Calculation and Plots Results.


% Input design values
theta2 = [-1.571 -1.46 -1.35 -1.222 -1.0472 -0.9599 -0.7854 -0.6981 -0.524 -
0.436 -0.3491 -0.2618 -0.175 -0.0873 0 0.175 0.524 0.873];
Fdesired = [65 66 67 70 80 85 88 89 90 91 90 89.5 88 86 84 80 64 46];

value = 1.9; % Steady state angular velocity

% Initalize matrices
theta2dot = [0 value value value value value value value value value value
value value value value value value 0];
% The following is the user input angular acceleration. Calculated based on
%  the spacing between the first two design points and the steady state
velocity.
theta2ddot = [value^2/0.22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -value^2/0.22];
theta4ddot = zeros(size(theta2dot));
theta5ddot = zeros(size(theta2dot));
theta2tdot = zeros(size(theta2dot));
theta4tdot = zeros(size(theta2dot));
theta5tdot = zeros(size(theta2dot));
F = zeros(size(theta2dot));
TA = zeros(size(theta2dot));
OFi = 0;
indx = 1;
q = 18;

% Position Analysis of Fourbar component.
for indx = 1:q
   theta3(indx) = theta2(indx) + param(9);
   C = param(4)^2-param(2)^2-param(3)^2-param(5)^2-
2*param(2)*param(3)*cos(theta3(indx))+2*param(2)*param(5)+2*param(3)*param(5)
*cos(theta3(indx));
   B = 4*param(3)*param(5)*sin(theta3(indx));
   A = param(4)^2-param(2)^2-param(3)^2-param(5)^2-
2*param(2)*param(3)*cos(theta3(indx))-2*param(2)*param(5)-
2*param(3)*param(5)*cos(theta3(indx));

   % Solve for theta5.
   t1 = (0-B+(B^2-4*A*C)^(0.5))/(2*A);
   theta5(indx) = 2*atan(t1);

   % Imaginary solution penalty (Checks closure).
   if (abs(imag(theta5(indx))))>0
      OFi = OFi + 100000*abs(imag(theta5(indx)))^8+10^30
   end
   % Solve theta4.
   theta4(indx) = asin((param(3)*sin(theta3(indx))-
param(5)*sin(theta5(indx)))/param(4));

   % Imaginary solution penalty (Checks closure).
   if (abs(imag(theta4(indx))))>0
      OFi = OFi + 100000*abs(imag(theta4(indx)))^8+10^30
   end
   devFvert(indx) = abs(pi-(theta5(indx)+param(8)));
end

% Velocity Analysis.

indx = 1
for indx = 1:q
   A = param(4)*sin(theta4(indx));
   B = param(5)*sin(theta5(indx));
   C = param(3)*theta2dot(indx)*sin(theta3(indx));
   D = param(4)*cos(theta4(indx));
   E = -param(5)*cos(theta5(indx));
   F = -param(3)*theta2dot(indx)*cos(theta3(indx));
   num4d = ((F*B)-(E*C));
   dend = ((D*B)-(E*A));
   num5d = ((D*C)-(F*A));
   theta4dot(indx) = num4d/dend;
   theta5dot(indx) = num5d/dend;
end

% Acceleration Analysis.
indx = 1;
for indx = 1:q
   A = param(4)*sin(theta4(indx));
   B = param(5)*sin(theta5(indx));
   CP =
param(3)*(theta2dot(indx)^2)*cos(theta3(indx))+param(3)*sin(theta3(indx))*the
ta2ddot(indx)+param(4)*(theta4dot(indx)^2)*cos(theta4(indx))-
param(5)*(theta5dot(indx)^2)*cos(theta5(indx));
   D = param(4)*cos(theta4(indx));
   E = -param(5)*cos(theta5(indx));
   FP = param(3)*(theta2dot(indx)^2)*sin(theta3(indx))-
param(3)*(theta2ddot(indx))*cos(theta3(indx))+param(4)*(theta4dot(indx)^2)*si
n(theta4(indx))-param(5)*(theta5dot(indx)^2)*sin(theta5(indx));
   dendd = (D*B-E*A);
   num4dd = (FP*B)-(E*CP);
   num5dd = (D*CP-FP*A);
   theta4ddot(indx) = num4dd/dendd;
   theta5ddot(indx) = num5dd/dendd;
end

% Dynamic Analysis.

indx = 1;
for indx = 1:q
num1 =
param(7).*(param(6).^2.*theta5ddot(indx)+32.174.*param(6).*cos(theta5(indx)+p
aram(8)));
```

```
den1 = param(5).*sin(theta5(indx)).*cos(theta4(indx))-
param(5).*sin(theta4(indx))*cos(theta5(indx));
F54(indx) = num1./den1;
F(indx) =
(F54(indx)*(cos(theta4(indx)).*param(3).*sin(theta2(indx)+param(9))-
sin(theta4(indx)).*param(3).*cos(theta2(indx)+param(9))))/param(1);
end

% Jerk Analysis
indx = 1;
for indx = 1:q
   theta4tdot(indx) =
((sin(theta5(indx))/cos(theta5(indx)))*(3*param(5)*theta5dot(indx)*theta5ddot
(indx)*sin(theta5(indx)) ...

+param(5)*theta5dot(indx)^3*sin(theta5(indx))+3*param(4)*theta4dot(indx)*thet
a4ddot(indx)*sin(theta4(indx))+param(4)*theta4dot(indx)^3*sin(theta4(indx))
...
      +param(3)*theta2tdot(indx)*cos(theta3(indx))-
3*param(3)*theta2dot(indx)*theta2ddot(indx)*sin(theta3(indx))-
param(3)*theta2dot(indx)^3*cos(theta3(indx))) ...

+3*param(5)*theta5dot(indx)*theta5ddot(indx)*cos(theta5(indx))+param(5)*theta
5dot(indx)^3*cos(theta5(indx))+param(4)*theta4dot(indx)^3*cos(theta4(indx))
...
      +3*param(4)*theta4dot(indx)*theta4ddot(indx)*cos(theta4(indx))-
param(3)*theta2tdot(indx)*sin(theta3(indx))-
3*param(3)*theta2dot(indx)*theta2ddot(indx)*cos(theta3(indx)) ...
      +param(3)*theta2dot(indx)^3*sin(theta3(indx)))/(-
param(4)*sin(theta4(indx))+param(4)*cos(theta4(indx))*sin(theta5(indx))/(cos(
theta5(indx))));
   theta5tdot(indx) =
(3*param(5)*theta5dot(indx)*theta5ddot(indx)*sin(theta5(indx)) ...

+param(5)*theta5dot(indx)^3*sin(theta5(indx))+3*param(4)*theta4dot(indx)*thet
a4ddot(indx)*sin(theta4(indx))+param(4)*theta4dot(indx)^3*sin(theta4(indx))-
param(4)*theta4tdot(indx)*cos(theta4(indx)) ...
      +param(3)*theta2tdot(indx)*cos(theta3(indx))-
3*param(3)*theta2dot(indx)*theta2ddot(indx)*sin(theta3(indx))-
param(3)*theta2dot(indx)^3*cos(theta3(indx)))/(param(5)*cos(theta5(indx)));
end



% Calculate Transmission Angle

indx = 1;
for indx = 1:q
   Z = param(2)^2+param(3)^2-2*param(2)*param(3)*cos(theta2(indx)+param(9));
   TA(indx) = acos((Z-param(4)^2-param(5)^2)/(-2*param(4)*param(5)));
end



% Plot results
```

```
figure(1)
plot(theta2*180/pi, theta4ddot, '-o', theta2*180/pi, theta5ddot, ':d')
xlabel('User Input Angle (degrees)')
ylabel('Angular Acceleration (radians/s^2)')
legend('Alpha 4', 'Alpha 5',0)
grid on

figure(2)
plot(theta2*180/pi, Fdesired, '-^',theta2*180/pi, F, ':o')
xlabel('User Input Angle (degrees)')
ylabel('Resistance Force (pounds)')
legend('Desired', 'Actual',0)
grid on
F
F54
figure(3)
plot(theta2*180/pi, theta4*180/pi, ':+',theta2*180/pi, theta5*180/pi, ':o')
xlabel('User Input Angle (degrees)')
ylabel('Theta 4,5 (degrees)')
legend('Theta 4', 'Theta 5 ', 0)
grid on

figure(4)
plot(theta2*180/pi, TA*180/pi,'-o')
xlabel('User Input Angle (degrees)')
ylabel('Tansmission Angle (degrees)')
grid on

figure(5)
plot(theta2*180/pi, theta4dot, ':+',theta2*180/pi, theta5dot, ':o')
xlabel('User Input Angle (degrees)')
ylabel('Omega 4,5 (radians/second)')
legend('Omega 4', 'Omega 5 ', 0)

figure(6)
plot(theta2*180/pi, devFvert*180/pi, ':o')
xlabel('User Input Angle (degrees)')
ylabel('Deviation From Vertical Normal (degrees)')
grid on

figure(7)
plot(theta2*180/pi, theta4tdot, '-d', theta2*180/pi, theta5tdot, '-o')
xlabel('User Input Angle (degrees)')
ylabel('Angular jerk (radians/s^3)')
legend('Coupler','Output',0)
grid on

% Find link length ratio penalty.
indx = 1;
Rmax = 0; Rmin = 1000;
for indx = 1:6
Rmin = min(abs(param(indx)), Rmin);
Rmax = max(abs(param(indx)), Rmax);
end
if Rmax/Rmin >= 15
   OFb = 400 + (Rmax/Rmin)^2;
else
```

```
   OFb = 0;
end

% Find penalty for negative link length
OFc = 0;
indx = 1;
for indx = 1:6
   if param(indx)<0
   OFc = OFc + (100-param(indx))^8;
end
end

% Find penalty for force difference. (Checks quality of fit).
OFa = 0;
diff = 0;
indx = 1;
for indx = 1:q
   diff = 3*abs(100*(Fdesired(indx)-F(indx))/Fdesired(indx))^4+diff;
end
OFa = diff;

% Negative force penalty
indx = 1;
for indx = 1:q
   if F(indx)<0
      OFa = OFa + abs(F(indx))^4;
   end
end

% Minimize the Magnitude of the Weight Relative to Max Force.
OFm = 0;
OFm = 1000^(param(7)*32.174*10000/max(Fdesired));

% Sum Penalities and Export Total.
OFa
OFb
OFc
OFtotal = OFa + OFb + OFc + OFi
```

The following program was used in solving the second example problem. The program, called OF4.m is a replacement for the subroutine OF2.m used with the first example problem. The main program was varied only with respect to changing the subroutine calls from OF2 to OF4 and PLT to PLT3 (see next program). Since these changes are minor, the main program has not been included with these changes.

```
function OFtotal = OF4(param)
% Brian Rundgren - Thesis Work - Fall 2001
% Kinematic and Dynamic Analysis of Four-bar Linkage
% With Objective Function Calculation and Plots Results.


% Input design values
theta2 = [-1.571 -1.46 -1.35 -1.222 -1.0472 -0.9599 -0.7854 -0.6981 -0.524 -
0.436 -0.3491 -0.2618 -0.175 -0.0873 0 0.175 0.524 0.873];
Fdesired = [65 66 67 70 80 85 88 89 90 91 90 89.5 88 86 84 80 64 46];
a = 1.0.*[0.03 0.03 0.03 0.03 0.03]; % Link mass coefficients.

value = 1.9; % Steady state angular velocity

% Initalize matrices
theta2dot = [0 value value value value value value value value value value
value value value value value value 0];
% The following is the user input angular acceleration. Calculated based on
%  the spacing between the first two design points and the steady state
velocity.
theta2ddot = [value^2/0.22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -value^2/0.22];
theta4ddot = zeros(size(theta2dot));
theta5ddot = zeros(size(theta2dot));
F = zeros(size(theta2dot));
TA = zeros(size(theta2dot));
OFi = 0;
indx = 1;
q = 18;

% Position Analysis of Fourbar component.

for indx = 1:q
   theta3(indx) = theta2(indx) + param(9);
   C = param(4)^2-param(2)^2-param(3)^2-param(5)^2-
2*param(2)*param(3)*cos(theta3(indx))+2*param(2)*param(5)+2*param(3)*param(5)
*cos(theta3(indx));
   B = 4*param(3)*param(5)*sin(theta3(indx));
   A = param(4)^2-param(2)^2-param(3)^2-param(5)^2-
2*param(2)*param(3)*cos(theta3(indx))-2*param(2)*param(5)-
2*param(3)*param(5)*cos(theta3(indx));

   % Solve for theta5.
   t1 = (0-B+(B^2-4*A*C)^(0.5))/(2*A);
   theta5(indx) = 2*atan(t1);

   % Imaginary solution penalty (Checks closure).
   if (abs(imag(theta5(indx))))>0
      OFi = OFi + 100000*abs(imag(theta5(indx)))^8+10^40
   end
   % Solve theta4.
   theta4(indx) = asin((param(3)*sin(theta3(indx))-
param(5)*sin(theta5(indx)))/param(4));

   % Imaginary solution penalty (Checks closure).
   if (abs(imag(theta4(indx))))>0
      OFi = OFi + 100000*abs(imag(theta4(indx)))^8+10^40
   end
   devFvert(indx) = abs(pi-(theta5(indx)+param(8)));
end

% Velocity Analysis.

indx = 1
for indx = 1:q
   A = param(4)*sin(theta4(indx));
   B = param(5)*sin(theta5(indx));
   C = param(3)*theta2dot(indx)*sin(theta3(indx));
   D = param(4)*cos(theta4(indx));
   E = -param(5)*cos(theta5(indx));
   F = -param(3)*theta2dot(indx)*cos(theta3(indx));
   num4d = ((F*B)-(E*C));
   dend = ((D*B)-(E*A));
   num5d = ((D*C)-(F*A));
   theta4dot(indx) = num4d/dend;
   theta5dot(indx) = num5d/dend;
end

% Acceleration Analysis.

indx = 1;
for indx = 1:q
   A = param(4)*sin(theta4(indx));
   B = param(5)*sin(theta5(indx));
   CP =
param(3)*(theta2dot(indx)^2)*cos(theta3(indx))+param(3)*sin(theta3(indx))*the
ta2ddot(indx)+param(4)*(theta4dot(indx)^2)*cos(theta4(indx))-
param(5)*(theta5dot(indx)^2)*cos(theta5(indx));
   D = param(4)*cos(theta4(indx));
   E = -param(5)*cos(theta5(indx));
   FP = param(3)*(theta2dot(indx)^2)*sin(theta3(indx))-
param(3)*(theta2ddot(indx))*cos(theta3(indx))+param(4)*(theta4dot(indx)^2)*si
n(theta4(indx))-param(5)*(theta5dot(indx)^2)*sin(theta5(indx));
   dendd = (D*B-E*A);
   num4dd = (FP*B)-(E*CP);
   num5dd = (D*CP-FP*A);
   theta4ddot(indx) = num4dd/dendd;
   theta5ddot(indx) = num5dd/dendd;
end

% Dynamic Analysis.

indx = 1;
for indx = 1:q
   ag4x = -theta4dot(indx)^2*param(4)*cos(theta4(indx))/2-
theta4ddot(indx)*param(4)*sin(theta4(indx))/2-
```

```
theta5dot(indx)^2*param(5)*cos(theta5(indx))-
theta5ddot(indx)*param(5)*sin(theta5(indx));
ag4y = -theta4dot(indx)^2*param(4)*sin(theta4(indx))/2-
theta4ddot(indx)*param(4)*cos(theta4(indx))/2-
theta5dot(indx)^2*param(5)*sin(theta5(indx))-
theta5ddot(indx)*param(5)*cos(theta5(indx));
ag5x = -theta5dot(indx)^2*param(5)*cos(theta5(indx))/2-
theta5ddot(indx)*param(5)*sin(theta5(indx))/2;
ag5y = -
theta5dot(indx)^2*param(5)*sin(theta5(indx))/2+theta5ddot(indx)*param(5)*cos(
theta5(indx))/2;
ag3x = -theta2dot(indx)^2*param(3)*cos(theta3(indx))/2-
theta2ddot(indx)*param(3)*sin(theta3(indx))/2;
ag3y = -
theta2dot(indx)^2*param(3)*sin(theta3(indx))/2+theta2ddot(indx)*param(3)*cos(
theta3(indx))/2;
agmx = -theta5dot(indx)^2*param(6)*cos(theta5(indx)+param(8))-
theta5ddot(indx)*param(6)*sin(theta5(indx)+param(8));
agmy = -
theta5dot(indx)^2*param(6)*sin(theta5(indx)+param(8))+theta5ddot(indx)*param(
6)*cos(theta5(indx)+param(8));
R63 = [param(3)*cos(theta3(indx))/2 param(3)*sin(theta3(indx))/2];
R44 = [param(4)*cos(theta4(indx))/2 param(4)*sin(theta4(indx))/2];
R55 = [param(5)*cos(theta5(indx))/2 param(5)*sin(theta5(indx))/2];
m3 = 2*a(3)*param(3);
m4 = 2*a(4)*param(4);
m5 = 2*a(5)*param(5);
I3  = m3*param(3)^2/4;
I4  = m4*param(4)^2/4;
I5  = m5*param(5)^2/2;
I6 = param(7)*param(6)^2
Tw = -param(7)*32.174*cos(theta5(indx)+param(8))*param(6)-
theta5ddot(indx)*param(6)^2*param(7);

uns = [1 0 1 0 0 0 0 0 0
        0 1 0 1 0 0 0 0 0
        0 0 0 0 1 0 1 0 0
        0 0 0 0 0 1 0 1 0
       -1 0 0 0 -1 0 0 0 0
       0 -1 0 0 0 -1 0 0 0
       R63(2) -R63(1) -R63(2) R63(1) 0 0 0 0 1
       0 0 0 0 2*R55(2) -2*R55(1) 0 0 0
       -R44(2) R44(1) 0 0 R44(2) -R44(1) 0 0 0];

kns = [m3*ag3x
       m3*ag3y+m3*32.174
       m5*ag5x+param(7)*agmx
       m5*ag5y+m5*32.174+param(7)*32.174+param(7)*agmy
       m4*ag4x
       m4*ag4y+m4*32.174
       I3*theta2ddot(indx)

(I5+I6)*theta5ddot(indx)+m5*32.174*param(5)*cos(theta5(indx))+param(7)*32.174
*param(6)*cos(theta5(indx)+param(8))
       I4*theta4ddot(indx)];

   forces = inv(uns)*kns;
   F54(indx) = forces(2)/sin(theta4(indx));
   F(indx) =
(param(1)^3/2*a(1)+param(1)^3*a(2)/8+forces(9))/param(1)+a(1)*32.174*param(1)
*cos(theta2(indx))/2+a(1)*32.174*param(1)*cos(theta2(indx))/4;
end

% Calculate Transmission Angle

indx = 1;
for indx = 1:q
   Z = param(2)^2+param(3)^2-2*param(2)*param(3)*cos(theta2(indx)+param(9));
   TA(indx) = acos((Z-param(4)^2-param(5)^2)/(-2*param(4)*param(5)));
end

% Find link length ratio penalty.
indx = 1;
Rmax = 0; Rmin = 1000;
for indx = 1:5
Rmin = min(abs(param(indx)), Rmin);
Rmax = max(abs(param(indx)), Rmax);
end
if Rmax/Rmin >= 4
   OFb = 400^2 + (Rmax/Rmin)^3;
else
   OFb = 0;
end

% Find penalty for negative link length
OFc = 0;
indx = 1;
for indx = 1:6
   if param(indx)<0
   OFc = OFc + (100-param(indx))^8;
end
end

% Find penalty for force difference. (Checks quality of fit).
OFa = 0;
diff = 0;
indx = 1;
for indx = 1:q
   diff = 3*abs(100*(Fdesired(indx)-F(indx))/Fdesired(indx))^4+diff;
end
OFa = diff;

% Negative force penalty
indx = 1;
for indx = 1:q
   if F(indx)<0
      OFa = OFa + abs(F(indx))^4;
   end
end

% Minimize the Magnitude of the Weight Relative to Max Force.
OFm = 0;
OFm = 1000^(param(7)*32.174*10000/max(Fdesired));
```

```
% Sum Penalities and Export Total.
OFtotal = OFa + OFb + OFc + OFi
```

The following program is the subroutine PLT3.m, and is a replacement for the subroutine PLT.m for when the second example problem was being solved. PLT3 has the same relationship with OF4 as PLT has with OF2.

```
function OFtotal = PLT3(param)
% Brian Rundgren - Thesis Work - Fall 2001
% Kinematic and Dynamic Analysis of Four-bar Linkage
% With Objective Function Calculation and Plots Results.


% Input design values
theta2 = [-1.571 -1.46 -1.35 -1.222 -1.0472 -0.9599 -0.7854 -0.6981 -0.524 -
0.436 -0.3491 -0.2618 -0.175 -0.0873 0 0.175 0.524 0.873];
Fdesired = [65 66 67 70 80 85 88 89 90 91 90 89.5 88 86 84 80 64 46];
a = (3/3).*[0.03 0.03 0.03 0.03 0.03]; % Link mass coefficients.

value = 1.9; % Steady state angular velocity

% Initalize matrices
theta2dot = [0 value value value value value value value value value value
value value value value value value 0];
% The following is the user input angular acceleration. Calculated based on
%  the spacing between the first two design points and the steady state
velocity.
theta2ddot = [value^2/0.22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -value^2/0.22];
theta4ddot = zeros(size(theta2dot));
theta5ddot = zeros(size(theta2dot));
theta2tdot = zeros(size(theta2dot));
theta4tdot = zeros(size(theta2dot));
theta5tdot = zeros(size(theta2dot));
F = zeros(size(theta2dot));
TA = zeros(size(theta2dot));
OFi = 0;
indx = 1;
q = 18;

% Position Analysis of Fourbar component.
for indx = 1:q
   theta3(indx) = theta2(indx) + param(9);
   C = param(4)^2-param(2)^2-param(3)^2-param(5)^2-
2*param(2)*param(3)*cos(theta3(indx))+2*param(2)*param(5)+2*param(3)*param(5)
*cos(theta3(indx));
   B = 4*param(3)*param(5)*sin(theta3(indx));
   A = param(4)^2-param(2)^2-param(3)^2-param(5)^2-
2*param(2)*param(3)*cos(theta3(indx))-2*param(2)*param(5)-
2*param(3)*param(5)*cos(theta3(indx));

   % Solve for theta5.
   t1 = (0-B+(B^2-4*A*C)^(0.5))/(2*A);
   theta5(indx) = 2*atan(t1);

   % Imaginary solution penalty (Checks closure).
   if (abs(imag(theta5(indx))))>0
      OFi = OFi + 100000*abs(imag(theta5(indx)))^8+10^40
   end
   % Solve theta4.
   theta4(indx) = asin((param(3)*sin(theta3(indx))-
param(5)*sin(theta5(indx)))/param(4));

   % Imaginary solution penalty (Checks closure).
   if (abs(imag(theta4(indx))))>0
      OFi = OFi + 100000*abs(imag(theta4(indx)))^8+10^40
   end
   devFvert(indx) = abs(pi-(theta5(indx)+param(8)));
end

% Velocity Analysis.

indx = 1
for indx = 1:q
   A = param(4)*sin(theta4(indx));
   B = param(5)*sin(theta5(indx));
   C = param(3)*theta2dot(indx)*sin(theta3(indx));
   D = param(4)*cos(theta4(indx));
   E = -param(5)*cos(theta5(indx));
   F = -param(3)*theta2dot(indx)*cos(theta3(indx));
   num4d = ((F*B)-(E*C));
   dend = ((D*B)-(E*A));
   num5d = ((D*C)-(F*A));
   theta4dot(indx) = num4d/dend;
   theta5dot(indx) = num5d/dend;
end

% Acceleration Analysis.
indx = 1;
for indx = 1:q
   A = param(4)*sin(theta4(indx));
   B = param(5)*sin(theta5(indx));
   CP =
param(3)*(theta2dot(indx)^2)*cos(theta3(indx))+param(3)*sin(theta3(indx))*the
ta2ddot(indx)+param(4)*(theta4dot(indx)^2)*cos(theta4(indx))-
param(5)*(theta5dot(indx)^2)*cos(theta5(indx));
   D = param(4)*cos(theta4(indx));
   E = -param(5)*cos(theta5(indx));
   FP = param(3)*(theta2dot(indx)^2)*sin(theta3(indx))-
param(3)*(theta2ddot(indx))*cos(theta3(indx))+param(4)*(theta4dot(indx)^2)*si
n(theta4(indx))-param(5)*(theta5dot(indx)^2)*sin(theta5(indx));
   dendd = (D*B-E*A);
   num4dd = (FP*B)-(E*CP);
   num5dd = (D*CP-FP*A);
   theta4ddot(indx) = num4dd/dendd;
   theta5ddot(indx) = num5dd/dendd;
end

% Jerk Analysis
indx = 1;
for indx = 1:q
   theta4tdot(indx) =
((sin(theta5(indx))/cos(theta5(indx)))*(3*param(5)*theta5dot(indx)*theta5ddot
(indx)*sin(theta5(indx)) ...

+param(5)*theta5dot(indx)^3*sin(theta5(indx))+3*param(4)*theta4dot(indx)*thet
a4ddot(indx)*sin(theta4(indx))+param(4)*theta4dot(indx)^3*sin(theta4(indx))
...
      +param(3)*theta2tdot(indx)*cos(theta3(indx))-
3*param(3)*theta2dot(indx)*theta2ddot(indx)*sin(theta3(indx))-
param(3)*theta2dot(indx)^3*cos(theta3(indx))) ...

+3*param(5)*theta5dot(indx)*theta5ddot(indx)*cos(theta5(indx))+param(5)*theta
5dot(indx)^3*cos(theta5(indx))+param(4)*theta4dot(indx)^3*cos(theta4(indx))
...
      +3*param(4)*theta4dot(indx)*theta4ddot(indx)*cos(theta4(indx))-
param(3)*theta2tdot(indx)*sin(theta3(indx))-
3*param(3)*theta2dot(indx)*theta2ddot(indx)*cos(theta3(indx)) ...
      +param(3)*theta2dot(indx)^3*sin(theta3(indx)))/(-
param(4)*sin(theta4(indx))+param(4)*cos(theta4(indx))*sin(theta5(indx))/(cos(
theta5(indx))));
   theta5tdot(indx) =
(3*param(5)*theta5dot(indx)*theta5ddot(indx)*sin(theta5(indx)) ...

+param(5)*theta5dot(indx)^3*sin(theta5(indx))+3*param(4)*theta4dot(indx)*thet
a4ddot(indx)*sin(theta4(indx))+param(4)*theta4dot(indx)^3*sin(theta4(indx))-
param(4)*theta4tdot(indx)*cos(theta4(indx)) ...
      +param(3)*theta2tdot(indx)*cos(theta3(indx))-
3*param(3)*theta2dot(indx)*theta2ddot(indx)*sin(theta3(indx))-
param(3)*theta2dot(indx)^3*cos(theta3(indx)))/(param(5)*cos(theta5(indx)));
end



% Dynamic Analysis.

indx = 1;
for indx = 1:q
   ag4x = -theta4dot(indx)^2*param(4)*cos(theta4(indx))/2-
theta4ddot(indx)*param(4)*sin(theta4(indx))/2-
theta5dot(indx)^2*param(5)*cos(theta5(indx))-
theta5ddot(indx)*param(5)*sin(theta5(indx));
ag4y = -theta4dot(indx)^2*param(4)*sin(theta4(indx))/2-
theta4ddot(indx)*param(4)*cos(theta4(indx))/2-
theta5dot(indx)^2*param(5)*sin(theta5(indx))-
theta5ddot(indx)*param(5)*cos(theta5(indx));
ag5x = -theta5dot(indx)^2*param(5)*cos(theta5(indx))/2-
theta5ddot(indx)*param(5)*sin(theta5(indx))/2;
ag5y = -
theta5dot(indx)^2*param(5)*sin(theta5(indx))/2+theta5ddot(indx)*param(5)*cos(
theta5(indx))/2;
ag3x = -theta2dot(indx)^2*param(3)*cos(theta3(indx))/2-
theta2ddot(indx)*param(3)*sin(theta3(indx))/2;
ag3y = -
theta2dot(indx)^2*param(3)*sin(theta3(indx))/2+theta2ddot(indx)*param(3)*cos(
theta3(indx))/2;
agmx = -theta5dot(indx)^2*param(6)*cos(theta5(indx)+param(8))-
theta5ddot(indx)*param(6)*sin(theta5(indx)+param(8));
agmy = -
theta5dot(indx)^2*param(6)*sin(theta5(indx)+param(8))+theta5ddot(indx)*param(
6)*cos(theta5(indx)+param(8));
R63 = [param(3)*cos(theta3(indx))/2 param(3)*sin(theta3(indx))/2];
R44 = [param(4)*cos(theta4(indx))/2 param(4)*sin(theta4(indx))/2];
```

```
R55 = [param(5)*cos(theta5(indx))/2 param(5)*sin(theta5(indx))/2];
m3 = 2*a(3)*param(3);
m4 = 2*a(4)*param(4);
m5 = 2*a(5)*param(5);
I3  = m3*param(3)^2/4;
I4  = m4*param(4)^2/4;
I5  = m5*param(5)^2/2;
I6 = param(7)*param(6)^2;
Tw = -param(7)*32.174*cos(theta5(indx)+param(8))*param(6)-
theta5ddot(indx)*param(6)^2*param(7);

uns = [1 0 1 0 0 0 0 0 0
       0 1 0 1 0 0 0 0 0
       0 0 0 0 1 0 1 0 0
       0 0 0 0 0 1 0 1 0
      -1 0 0 0 -1 0 0 0 0
      0 -1 0 0 0 -1 0 0 0
      R63(2) -R63(1) -R63(2) R63(1) 0 0 0 0 1
      0 0 0 0 2*R55(2) -2*R55(1) 0 0 0
      -R44(2) R44(1) 0 0 R44(2) -R44(1) 0 0 0];

kns = [m3*ag3x
      m3*ag3y+m3*32.174
      m5*ag5x+param(7)*agmx
      m5*ag5y+m5*32.174+param(7)*32.174+param(7)*agmy
      m4*ag4x
      m4*ag4y+m4*32.174
      I3*theta2ddot(indx)

(I5+I6)*theta5ddot(indx)+m5*32.174*param(5)*cos(theta5(indx))+param(7)*32.174
*param(6)*cos(theta5(indx)+param(8))
      I4*theta4ddot(indx)];

   forces = inv(uns)*kns;
   F54(indx)  = forces(2)/sin(theta4(indx));
   F(indx)  =
(param(1)^3/2*a(1)+param(1)^3*a(2)/8+forces(9))/param(1)+a(1)*32.174*param(1)
*cos(theta2(indx))/2+a(1)*32.174*param(1)*cos(theta2(indx))/4;
end

% Calculate Transmission Angle

indx = 1;
for indx = 1:q
   Z = param(2)^2+param(3)^2-2*param(2)*param(3)*cos(theta2(indx)+param(9));
   TA(indx)  = acos((Z-param(4)^2-param(5)^2)/(-2*param(4)*param(5)));
end



% Plot results


figure(1)
plot(theta2*180/pi, theta4ddot, '-o', theta2*180/pi, theta5ddot, ':d')
xlabel('User Input Angle (degrees)')
ylabel('Angular Acceleration (rad/s/s)')
```

```
legend('Alpha 4', 'Alpha 5',0)
grid on

figure(2)
plot(theta2*180/pi, Fdesired, '-^',theta2*180/pi, F, ':o')
xlabel('User Input Angle (degrees)')
ylabel('Resistance Force (pounds)')
legend('Desired', 'Actual',0)
grid on
F
F54
figure(3)
plot(theta2*180/pi, theta4*180/pi, ':+',theta2*180/pi, theta5*180/pi, ':o')
xlabel('User Input Angle (degrees)')
ylabel('Theta 4,5 (degrees)')
legend('Theta 4', 'Theta 5 ', 0)
grid on

figure(4)
plot(theta2*180/pi, TA*180/pi,'-o')
xlabel('User Input Angle (degrees)')
ylabel('Tansmission Angle (degrees)')
grid on

figure(5)
plot(theta2*180/pi, theta4dot, ':+',theta2*180/pi, theta5dot, ':o')
xlabel('User Input Angle (degrees)')
ylabel('Omega 4,5 (radians/second)')
legend('Omega 4', 'Omega 5 ', 0)

figure(6)
plot(theta2*180/pi, devFvert*180/pi, ':o')
xlabel('User Input Angle (degrees)')
ylabel('Deviation From Vertical Normal (degrees)')
grid on

figure(7)
plot(theta2*180/pi, theta4tdot, '-o',theta2*180/pi, theta5tdot, '-d')
xlabel('User Input Angle (degrees)')
ylabel('Angular jerk (radians/s^3)')
legend('Coupler','Output',0)
grid on

% Find link length ratio penalty.
indx = 1;
Rmax = 0; Rmin = 1000;
for indx = 1:5
Rmin = min(abs(param(indx)), Rmin);
Rmax = max(abs(param(indx)), Rmax);
end
if Rmax/Rmin >= 4
   OFb = 400^2 + (Rmax/Rmin)^3;
else
   OFb = 0;
end

% Find penalty for negative link length
```

```
OFc = 0;
indx = 1;
for indx = 1:6
   if param(indx)<0
   OFc = OFc + (100-param(indx))^8;
end
end

% Find penalty for force difference. (Checks quality of fit).
OFa = 0;
diff = 0;
indx = 1;
for indx = 1:q
   diff = 3*abs(100*(Fdesired(indx)-F(indx))/Fdesired(indx))^4+diff;
end
OFa = diff;

% Negative force penalty
indx = 1;
for indx = 1:q
   if F(indx)<0
      OFa = OFa + abs(F(indx))^4;
   end
end

% Minimize the Magnitude of the Weight Relative to Max Force.
OFm = 0;
OFm = 1000^(param(7)*32.174*10000/max(Fdesired));

% Sum Penalities and Export Total.
OFa
OFb
OFc
OFtotal = OFa + OFb + OFc + OFi
```

Vita

Brian T. Rundgren was born on February 21, 1977 in Harrisonburg, VA. He is the son of David W. and Millicent Q. Rundgren. Brian spent most of his childhood in Eggleston, VA, learning how to make the best of having an older brother, Shane. After graduating from Giles High School in 1995, Brian attended college at Clemson University, where he graduated *magna cum laude* in December 1999. During his studies at Clemson, Brian participated in the cooperative education program, spending 3 semesters working for Advanced Automation in Greenville, SC. Immediately after receiving his Bachelor's in Mechanical Engineering; Brian entered the graduate program at Virginia Tech. Brian completed his Master's in Mechanical Engineering in December 2001.