Abstract
This paper details two problems that employ Genetic-based Multi-objective Optimization algorithms on dynamic systems. The first is a simple, single input system and the second is a complex two input system. First, the two systems are detailed with the derivation of their models. From these models, an optimization algorithm based on NSGA-II is run, constructing the Pareto Frontiers. The results are presented and then analyzed.

1. INTRODUCTION
Many skilled, repetitive, physical tasks such as, shooting a rifle, bowling, or throwing a dart, have an infinite number of "strategies", or parameter combinations, that a user could employ to achieve the same end-result. For example, when bowling, the player can adjust their wrist angle, throwing height and throwing force in an infinite number of combinations that will each result in a strike. Each combination, though, would have a different accuracy, sensitivity to parameters, or expended energy associated with it. The user is interested in optimizing these multiple objectives simultaneously in order to formulate an optimal strategy, for example, one that is accurate at hitting a target while using as little energy as possible. In most systems, these two objectives cannot both be at their minimum values under any combination of parameters. However, there is a set of combinations where, the only feasible way to improve the value of one objective is to degrade the value of another. This collection of combinations is referred to as the Pareto Frontier.

Conflicting performance goals in dynamic systems have been investigated in so-called Goal Equivalence Manifolds (GEM). Succinctly, the GEM is a subspace of solutions in the parameter space where one objective is at its minimum. Deviation from the GEM imparts error on the objective with varying sensitivity. In comparing theoretical constructions of the GEM to data collected from human trials of the same task, the hypothetical objective can be tested for validity. As one becomes skilled at a task, it may be posited that their strategy will cluster around the GEM at the point of low sensitivity. If the data shows conflicting results, it is possible that the user has additional objectives that they are trying to extremize, but are not being modeled theoretically.

This paper attempts to construct multi-objective GEMS by means of a Pareto Frontier for two systems: one relatively simple with a single input and one more complicated with two inputs. Using the models, optimal strategies for completing the task are found along a Pareto Frontier using Evolutionary Multi-Objective Optimization. The parameters are then analyzed for their impact on the different objectives.

2. PROBLEM I

2.1 Description
The first problem found the optimal set of strategies for a shuffleboard game diagrammed in Figure 2.1. Starting from rest, the "player" is to slide the puck so that it comes to rest as close as possible to the end of the board, at L. The player is permitted to apply a force, \( F(t) \), to the puck until the puck reaches a specified percentage of the length of the board, \( \chi \). The puck is subject to the force of kinetic friction, \( \mu mg \), as it slides, and it is assumed that the applied force is always great enough to overcome the static friction at the start. In this consideration, the player simultaneously desires that the puck stop as close to the target at \( L \) as possible, and that small perturbations in the magnitude or the time of application will not significantly increase the error at the target. These objectives were classified as error and sensitivity.

![Figure 2.1: System of Problem I](image-url)
\[ \zeta''(\tau) = \frac{F(\tau)}{\kappa} - 1 \quad (2.1) \]
\[ \zeta(\tau) = \frac{x(\tau)}{L} \quad (2.2) \]
\[ \kappa = g \mu \quad (2.3) \]
\[ \tau = t \sqrt{\frac{g \mu}{L}} \quad (2.4) \]

Where \( \mu \) is the coefficient of kinetic friction, \( g \) is the local gravity, \( m \) is the mass of the puck, \( L \) is the length of the board, and \( t \) is the scaled, dimensionless time. \( x \) is a system parameter that is equivalent to the force due to kinetic friction.

### 2.2 Method

The applied force was in the form of a Heaviside (Step) function with two variable parameters as seen in Equation 2.5. This function was chosen in order to keep the number of design variables low compared to, for example, assuming the form of a truncated Fourier series. Additionally, the step function, or so-called "bang-bang" control, is commonly the most efficient input function in dynamic systems.

\[ F(t) = x_1 H(t - x_2) \quad (2.5) \]

The optimization program used the varying \( x \) during function evaluation. Using these parameters, the Ordinary Differential Equation (ODE) in Equation 2.1 was solved numerically using MATLAB's `ode45`. An additional consideration in the equation was that the friction force turned off when the velocity was equal to zero. Using the outputted states, the time when the position of the puck is equal to \( \chi \) was denoted as \( t_e \), and the time when the velocity of the puck returns to zero was denoted as \( t_f \).

The goal was to simultaneously minimize both the error and sensitivity by adjusting \( x_1, x_2 \). The error, \( f_1 \), was calculated as the difference between the position of the puck when the velocity equaled zero and its desired position of one. The sensitivity, \( f_2 \), of a solution was calculated by re-performing the above calculations with both design parameters increased by 1%. This perturbed final position was compared to the unperturbed final position by a simple percentage-error calculation. Though there were more accurate methods for estimating sensitivity, most required many additional function evaluations around the point-of-interest. Using this method, only one evaluation was required, keeping the computational time reasonable to run locally on a laptop computer.

The constraints were turned into penalty functions to the error and sensitivity. The first part of the penalty calculates the impulse after the puck passes \( \chi \). Any force applied after this point increased the value of the objectives. The second penalty was placed on the final velocity. Since the states were evaluated numerically, the time when the velocity reached zero may not have been precise. This penalty added to the objective values if the velocity at \( t_f \) was not zero.

\[ f_1 = (1 - \zeta(t_f))^2 + \text{penalty} \quad (2.6) \]
\[ f_2 = \left| \frac{\zeta(t_f) - \zeta(t_f)}{\zeta(t_f)} \right| + \text{penalty} \quad (2.7) \]
\[ \text{penalty} = \int_{t_e}^{t_f} F(t) \, dt + v(t_f)^2 \quad (2.8) \]

### 2.2.1 Optimization Method

Because both Problem I and Problem II required continuous parameters as well as multiple objective functions, Deb’s Non-dominated Sorting Genetic Algorithm II (NSGA-II) was chosen as the optimization algorithm [2]. The MATLAB version of the algorithm that was built upon was downloaded from The MathWorks’s file-exchange community [1].

NSGA-II operates by first creating a population with random parameter values and then evaluating the fitness of each member with respect to each objective function. The population is then ranked and sorted into fronts based on fitness. The first front contains all members of the population that are non-dominated; meaning no other member in the population has all of its objective values greater than the member in focus. When all non-dominated members are found, the process repeats to find the second front, ignoring all members in the first front. This front-creation system repeats until all members are included in a front.

Next, the selection and mating operations occur. The sorted population is randomly partitioned into pairs or groups and placed into a tournament selection scheme. The member of the highest-ranked front in each grouping is selected to continue on to the mating operation. In the case of a tie in ranking, the winner is the member that is farthest from other solutions of that generation. This measurement is determined by the average distance of the point to its two closest neighbors in the objective space. The rank comparison encourages the fronts being pushed towards the actual Pareto Frontier, the subgroup of solutions that are not dominated by any other feasible solution in the objective space. The distance comparison encourages the members to spread across the fronts for a diverse collection of solutions. The members that have been selected for mating are randomly paired to produce the next generation of solutions. Depending on the algorithm settings, new members may be the result of crossover; possessing some parameter values from each parent. Finally, the mutation operation may be applied to the new child generation, randomly changing a parameter. The parent generation is then either removed from the system, or included with their children in the next round of evaluations. The new members are evaluated and the process returns to the selection step.

This cyclical procedure is repeated until convergence is reached, or for an arbitrary number of generations if convergence is not achieved. Convergence can be determined in numerous fashions, for example, multiple generations with little or no improvement in the fitness- or parameter values. The determination of how small the change is dependent on the problem being evaluated and the time allowed for the optimization to run. At this point, the rank-one solutions

---

2. This particular version of the algorithm did not include penalty functions directly in the sorting
now comprise the Pareto Frontier of the solution space.

2.3 Results

The experiments were conducted using $\chi = \frac{1}{3}$ and $\kappa = 15$. These values were selected arbitrarily for demonstration purposes. The probability of mutation was set to 10% and the probability of crossover was set to 100%. These were the program's default values and seemed to yield good results. The amplitude of the force, $x_1$, was permitted to have values in $[0,1000]$ and the application time, $x_2$, was permitted values in $[0,2]$. The NSGA-II algorithm was run with increasing population sizes and number of generations between 175 and 275, and the Pareto Frontiers were created and plotted in Figure 2.2.

The amplitude of the force, $x_1$, was permitted to have values in $[0,1000]$ and the application time, $x_2$, was permitted values in $[0,2]$. The NSGA-II algorithm was run with increasing population sizes and number of generations between 175 and 275, and the Pareto Frontiers were created and plotted in Figure 2.2.

The results from these runs were aggregated and run through the non-dominated sort to create an ensemble Pareto Frontier in Figure 2.3.

Finally, a zoomed-in ensemble Pareto Frontier plot containing the both the objective- and parameter values was plotted in Figure 2.5.

2.4 Analysis

As seen in Figure 2.2, the multiple seeds converged at higher population and generation size as there is no consistent improvement of the Pareto Frontier. The ensemble Pareto Frontier in Figure 2.3 shows that the ideal value of error, zero, was never reached using the NSGA-II. The distribution across the frontier appears poor in the graph, but solutions with higher error or sensitivity were not included in the plot.

Even the most accurate solution still had noticeable error at the target as seen in the state plot in Figure 2.4; the puck
The data was assumed to be well-distributed along the Pareto Frontier, as is the intention of NSGA-II. Even though the relationship between input and output may not be linear, as partial-correlation assumes, the smaller subset of solutions gave a good estimate of the relationship around the ideal point of zero error and zero sensitivity. From the partial-correlation data and the graph, it was observed that error and sensitivity are inversely correlated with the input parameters, which was expected considering the objectives were in competition. It was also seen that the force parameter has a high correlation and significance with both the error and sensitivity. The time of application, however, showed a weak correlation but good significance with both error and sensitivity.

The implication of this result was that applying a high force for a short duration yielded more accurate strategies while applying a low force for a longer time yielded more robust solutions.

3. PROBLEM II

3.1 Description

The second analyzed system was a simplified basketball throwing device, illustrated in Figure 3.1. Beginning from rest, a force, $F(t)$, is applied to a mass, $m_1$, accelerating it in the $+y$-direction. Additionally, a moment, $M(t)$, is applied to a slender rod (pinned to the first mass) with mass of $m_2$, polar inertia of $I$, and length of $L$. A ball, with mass of $m_3$, rests at the top of the rod and constrained to only move perpendicularly to the rod. The ball is released from the rod when its acceleration drops to zero. At release, the ball flies through the air towards a basket positioned at a specified height, $H$, and distance, $D$, from the center of the device, shown in Figure 3.2. The ball is released with initial horizontal and vertical positions and velocities denoted by $X$, $Y$, $X'$, and $Y'$ with all parameters normalized with respect to the rod length, $L$, and non-dimensionalized. This system roughly modeled a player extending their legs and swinging their arms over their head to shoot a basketball. For this analysis, the player’s goals were to hit the target (basket) as closely as possible and use the minimal amount of energy. For this section, the impulse – the change in velocity of the mass – and the energy were treated as equivalent.

Using Lagrange’s equations for the system, the non-linear, non-dimensional governing equations of motion for this problem were:

$$Y''(t) = -\left( -2L \left( 12n_1 + 5n_2 - 12 \right) F(t) + 3 \right)$$

$$L \text{Mass} \left( g \left( 3 \left( 2n_1 + n_2 - 2 \right)^2 \cos(2\theta(t)) \right) - 3 \left( 2n_1 + n_2 \right)^2 - 2 \left( n_2 + 6 \right) \right) + L \left( 2n_1 + n_2 - 2 \right)$$

$$\left( 12n_1 + 5n_2 - 12 \right) \theta'(t)^2 \sin(\theta(t)) + 12 \left( 2n_1 + n_2 - 2 \right) M(t) \cos(\theta(t)) / \left( 2LW \left( 3 \left( 2n_1 + n_2 - 2 \right)^2 \cos^2(\theta(t)) + 12n_1 + 5n_2 - 12 \right) \right)$$

$$\theta''(t) = \left( 3 \left( L \left( 2n_1 + n_2 - 2 \right) \right) \right)$$

Where:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Objective</th>
<th>r-value</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force</td>
<td>Error</td>
<td>-0.8076</td>
<td>0.001</td>
</tr>
<tr>
<td>Force</td>
<td>Sensitivity</td>
<td>0.8210</td>
<td>0.003</td>
</tr>
<tr>
<td>Time</td>
<td>Error</td>
<td>-0.1046</td>
<td>0.0052</td>
</tr>
<tr>
<td>Time</td>
<td>Sensitivity</td>
<td>0.1118</td>
<td>0.0377</td>
</tr>
</tbody>
</table>

Figure 3.1: Dynamic System of Problem II

Figure 3.2: Normalized Ballistic System of Problem II
The parameters for the system were: $m_1$: the mass of the base, $m_2$: the mass of the arm, $m_3$: the mass of the projectile, $L$: the length of the arm, $g$: the local gravity, $F$ is an applied force in the $y$-direction and $M$ is an applied moment in the $θ$-direction. $τ$ is the scaled non-dimensional time, $Y$ is the non-dimensional vertical distance from the datum line.

### 3.2 Method

The applied force and -moment were composed of Heaviside functions with variable parameters as expressed in Equations (3.9) and (3.10). The values of $x_1, x_4$ determine the magnitudes, $x_2, x_5$ determine the turn-on time and $x_3, x_6$ determine the turn-off time. For this problem, a Heaviside function was used as in Problem I for the same reasons.

\[
F(t) = x_1(H(x_2 - t) - H(x_3 - t)) \quad (3.9)
\]

\[
M(t) = x_4(H(x_5 - t) - H(x_6 - t)) \quad (3.10)
\]

To solve this problem, the optimizer took a set of parameters, $x$, and applied them to the force and moment definitions and then solved the ODEs in Equations (3.9) and (3.10). At time, $t_r$, where $θ''(t_r) = 0$ for the first time, the projectile is released from the swinging arm. This time, $t_r$ is also when $θ'$ reaches its first local maximum. The velocity and position of the projectile at $t_r$ were calculated from the solution to the ODE and then used as the initial conditions in a 2-D ballistic model. The second model was used to calculate how close the projectile lands horizontally to the target. This was done by first finding the time when the ball was at the same height as the target after descending from its apex. The time was multiplied by the initial horizontal velocity to find the projectile’s distance from its release point. This measurement was subtracted from the actual distance to the apex. The time was multiplied by the initial horizontal velocity to find the projectile’s distance from its release point.

The goal of this analysis was to simultaneously minimize both the error and total impulse by adjusting $x$. The error, $f_1$, was determined as described above. The impulse, $f_2$, was calculated as sum of the time integrals of the applied force and -moment functions. The first value in the penalty function was set to minimize the idle time before the force or moment is activated. This prevented an infinite number of equivalent solutions that simply waited different lengths of time to activate the system. The second and third facets of the penalty function ensure that the turn-off times occur after the turn-on times. Lastly, the fourth factor ensures only solutions that release the ball before the rod is vertical were considered as these solutions would not reach the basket.

\[
y(τ) = \frac{y(τ)}{L} \quad (3.3)
\]

\[
τ = t_1\sqrt{\frac{2}{L}} \quad (3.4)
\]

\[
\text{Mass} = m_1 + m_2 + m_3 \quad (3.5)
\]

\[
n_1 = \frac{m_1}{\text{Mass}} \quad (3.6)
\]

\[
n_2 = \frac{m_2}{\text{Mass}} \quad (3.7)
\]

\[
W = g\text{Mass} \quad (3.8)
\]

### 3.3 Results

The optimization was run using the following SI-based parameters: $m_1 = 80, m_2 = 3, m_3 = 0.8, g = 9.81, L = 0.83, D = 4, H = 3$. The mass values were selected as the average values of a human torso, human arm and standard basketball. The height, $H$, is the standard height for a basketball hoop, and the distance, $D$, has the player positioned 4m away. $L$ is the average value of the length of a human arm.

The amplitudes of force and moment, $\{x_1, x_4\}$, were permitted values in $[0, 1000]$. The time parameters, $\{x_2, x_3, x_5, x_6\}$, were permitted values in $[0, 2]$. After running the NSGA-II algorithm with a population and number of generations ranging from 150 to 250, the Pareto Frontiers were created in Figure 3.3 and the aggregated Pareto Frontier was created in Figure 3.4.

![Figure 3.3: Pareto Frontiers of System 2](image)

A plot of the ODE of one efficient solution is shown in Figure 3.5.

A zoomed-in section of the Pareto front is shown in Figure 3.6 with the vertical axis denoting the impulse of the moment, $x_4(x_6 - x_5)$, and the color denoting the impulse of the force, $x_1(x_3 - x_2)$.

### 3.4 Analysis

As seen in Figure 3.3, the Pareto Frontier began to converge by the seed of 225 population and generations as there is no improvement of the front between that and the seed of 250 of each. The ensemble Pareto Frontier in Figure 3.4 displays a well-spread frontier. However, the frontier does not reach the ideal zero-value for error, but hits a wall of approximately 23. Further work on the problem would determine where in the model this limit on error is coming.
Figure 3.4 and the following partial correlation table show the relationship between force impulse and moment impulse with the objectives.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Objective</th>
<th>r-value</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force Impulse</td>
<td>Error</td>
<td>-0.1280</td>
<td>0.0604</td>
</tr>
<tr>
<td>Force Impulse</td>
<td>Total Impulse</td>
<td>0.1419</td>
<td>0.0372</td>
</tr>
<tr>
<td>Moment Impulse</td>
<td>Error</td>
<td>-0.2678</td>
<td>0.0001</td>
</tr>
<tr>
<td>Moment Impulse</td>
<td>Total Impulse</td>
<td>1.0000</td>
<td>0</td>
</tr>
</tbody>
</table>

As seen from the partial-correlation values, both the force- and moment impulse were positively correlated with the total impulse of the system, which was an obvious result as the total impulse is the sum of the two parameters. The inputs showed a weak negative linear correlation with the error.

More compelling were qualitative observations from the scatter plot. For the system parameters approximating a human throwing the ball, the force impulse was about two orders of magnitude less than the moment impulse. This implied that it was not worth the energy to accelerate the heavy $m_1$ (the torso); all energy was spent on rotating the lighter $m_2$ (the arms). The best solutions, however, were those that added some force impulse to the moment impulse. These solutions had both less error and less total energy spent than some solutions that had almost exclusively moment impulse.

Because of the number of parameters, it was difficult to analyze and visualize the effect of timing when the force or moment were applied.

4. CONCLUSIONS

The optimization for Problem I successfully found multiple solutions on the Pareto Frontier, but failed to locate the extreme point of zero error. Based a subset of solutions near the ideal point of zero error and zero sensitivity, it was found that more accurate solutions tended to have a higher force amplitude and shorter duration. Conversely, more robust solutions tended to have smaller amplitude but a longer duration.

From the results of Problem II, almost the entire Pareto Frontier was populated with solutions, making an extremely diverse collection. However, an error in either the model or the optimization prevented solutions being found that had a error near its minimum of zero. From the data based on human-scaled parameters, applying a small impulse of force and a large impulse of moment created solutions that used the least energy and, according to the model, were the most accurate.

Using NSGA-II proved to be a successful method for finding optimal strategies for both problems. Derivative-based methods would have been difficult or impossible to implement for both problems, especially the second due to its high complexity and physical constraints. For the same reasons, the second problem would not have been solved well with variational methods. Additionally, considering that both problems were multi-objective, finding the Pareto Frontier would have been troublesome even if other standard optimization methods could have been used.
5. FUTURE WORK

Further work on this project could include the following additions and refinements:

- Improve sensitivity-calculation technique in Problem I
- Investigate how changing $\chi$ changes the strategy of Problem I
- Compare results from a human trial to simulated objective space in Problem I
- Fix the modeling or optimization error in Problem II that prevents the error from reaching zero
- Add sensitivity as a third objective in Problem II
- Develop a feedback system to explore how repeated trials change based on perception of performance

6. REFERENCES