ICED13
INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

A MODULAR DESIGN TOOL FOR VISUALIZING COMPLEX MULTISCALE SYSTEMS

Paul EGAN (1), Jonathan CAGAN (1), Christian SCHUNN (2), Philip LEDUC (1)
1: Carnegie Mellon University, United States of America; 2: University of Pittsburgh, United States of America

ABSTRACT
As engineers design ever more complex systems, such as bionanotechnology, it is particularly difficult for engineers to navigate the influences of parameters to design across multiple scales. We introduce a software tool that contains graphical interface modules that display multiple inputs and outputs simultaneously in real-time, across multiple scales of interest. To assess the utility of real-time feedback in the tool, the tool is validated through user studies involving counterbalanced sequences of optimization tasks with varying complexity and either real-time or delayed feedback. Real-time feedback consistently improved a user's ability to find better designs and the benefit increased as task complexity grew. Our scientific contribution is a greater understanding of how users search and understand complex design spaces. Users in the real-time condition were made fewer changes to designs each iteration and identifying such strategies could inform agent-based algorithms. Findings aid complex systems applications including virtual experiments, reverse engineering, and scientific discovery.

Keywords: complexity, GUI, myosin, cognitive study

Contact:
Prof. Jonathan Cagan
Carnegie Mellon University
Mechanical Engineering
Pittsburgh
15213
United States of America
cagan@cmu.edu

1 INTRODUCTION AND MOTIVATION

Complex systems are fundamentally different than merely complicated systems and present challenges for engineering design in many domains of application (Ottino, 2004). At a recent NSF workshop, the difficulties in engineering complex systems was outlined, and a particular importance was placed on the tendency that (1) Complex systems consist of many interacting parts across multiple scales of space and time and (2) Complex systems often exhibit counter-intuitive emergent behavior (Guckenheimer and Ottino, 2008). One domain that exemplifies both of these criteria is biological systems design at the nanoscale (Egan et al., 2012), such as the design of synthetic muscle tissue (Bach et al., 2004) and nano-actuators (Neiman and Varghese, 2011). These technologies are enabled from the coordinated interactions of many myosin molecular biomotors with sliding actin filaments (Figure 1) and have applications in medical treatments and lab-on-chip technologies. Our goal is to develop a multi-purpose design tool for this specific complex systems domain that is modular and is easily extendible to further domains and applications. In these complex systems applications, possibly quadrillions of myosins cyclically convert chemical energy into mechanical motion. There are also many design variables that influence emergent system performance, with changes to one design variable having multiple effects on other variables in counter-intuitive ways. For instance, increasing the number of myosins in the system will increase energy use (since there are more myosins cycling) and reduce the force each myosin must generate, resulting in a higher filament velocity. Current biophysical models are often inadequate for predicting when certain types of emergent behaviors occur (such as system dissociation) and therefore wet-lab experiments that are costly both in terms of times and materials must be conducted in order to measure a system's performance in different configurations.

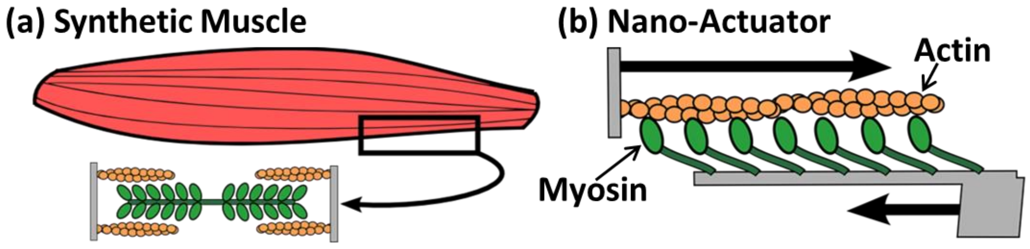


Figure 1. Schematics of (a) Synthetic Muscle and (b) Nano-Actuator Technologies

Developing myosin-based nano-technologies is growing in feasibility as synthetic biology (Heinemann and Panke, 2006) and systems biology (Purnick and Weiss, 2009) are emerging as prominent fields, and new techniques are more readily available for configuring individual molecules and systems. Synthetic biology refers to the process of developing artificial biological components whereas systems biology concentrates on controlling vast networks of biological components. Design applications of these fields are biopolymer materials (Koenderink et al., 2009), smart materials systems (Lv et al., 2010), lab-on-chip technologies (Bakewell and Nicolau, 2007), and biosensors (Agarwal and Hess, 2010). Additionally, improving design processes in these fields could lead to improvements in cancer treatments (Laubenbacher et al., 2009) and gene therapies (Sheridan, 2011). Software tools exist for modeling synthetic biological systems (Young and Alper, 2010), but tend to rely on biochemical rather than the mechanical approaches necessary for designing biomotor technologies.

Our approach is to build a tool that facilitates an engineer's ability to reason through many interactions of parameters present in a complex system, and therefore we focus on developing an intuitive representation and interface to aid the user. In our current implementation, the user may view real-time visual feedback of four independent and two dependent variables simultaneously which may promote a better understanding of how all of a systems' interacting parameters behave during the design process. Additionally, the tool may aid experts by allowing them to view the most crucial interactions and gain information about the system as a whole rather than through fragmented pieces of information and equations. Insights from how human designers search these spaces could inform future agent-based algorithms.

Often, complex systems vary in the amount of information that is available to the designers. These systems are often differentiated by the terms Kind (information feedback is instant) or Wicked (information feedback is delayed) and offer a metric for measuring the effectiveness of the tool through assessing how users design in both Wicked and Kind conditions (Pretz, 2008). With respect to the biological domain, differences in feedback could reflect the computational cost required to evaluate a design or the availability of feedback from empirical studies. Therefore, in certain experimental contexts where it is expensive to design and manufacture a system to measure, there would be varying trade-offs in whether a Kind or Wicked design approach is more beneficial (i.e. building a design, then measuring, and redesigning based on information reflects Kind scenarios, or building multiple designs and evaluating them simultaneously reflects Wicked scenarios). By studying users in the context of Kind and Wicked feedback conditions, we are also able to validate that real-time feedback in a tool is beneficial to the design process and users' understanding of the system, and expect that users should benefit from immediate feedback conditions. The user's design proficiency in each environment must be measurable, so we develop NLP (nonlinear programming) optimization formulations of the design space to compare the evaluation of a user's design to a global optimum.

2 BACKGROUND

Two relevant areas of background literature were reviewed that include an overview of complex systems design tools and past models of myosin biomotors.

2.1 Complex Systems Cognition and Software

Previous studies have demonstrated that is difficult for people to understand the causal mechanisms in complex systems (Hmelo-Silver et al., 2007) and a subsequent software tool was developed (Vattam et al., 2011) to facilitate better qualitative understanding of natural complex systems. Our tool, however, focuses on the quantitative task of optimizing parameters once a complex design space is represented in an NLP scheme. Many cognitive studies have looked into the most beneficial way for users to find solutions in quantitative problems through controlling the manipulation of one variable at a time (Chen et al., 1999), but in complex design spaces where variables are sensitive to multiple influences it may be quite difficult for users to search efficiently (Kuhn et al., 2008). Our tool may aid users in searching these specific multivariable spaces, since it provides visual feedback that presents information about multiple parameter influences simultaneously.

A variety of tools exist for modeling, and sometimes designing synthetic biological systems. A few examples are the visual design tool TinkerCell (Chandra et al., 2009) for components inside cells, an agent-based biochemical reaction simulator NFSim (Sneddon et al., 2011), and Genedesigner (Villalobos et al., 2006) which is a visual design tool for *de novo* gene configurations. There is a current lack of design tools addressing complex mechanochemical systems although we previously developed an agent-based simulation for designing myosin systems (Egan et al., 2013). Our current approach is to leverage the graphical renderings and findings from our past approach, but utilize mathematical modeling to reduce computational effort and allow real-time feedback while integrating further useful features.

2.2 Myosin Modeling

A full account of myosin design is beyond this paper's scope, so only equations necessary for creating the interface is presented (Egan et al., 2011). Myosins interact with a moving actin filament subject to an applied force (Figure 2). Myosins have three structural states, attached and generating positive force, attached and generating negative force, or detached and generating no force. A myosin's performance is tunable via engineering its molecular configuration for altered mechanics (Anson et al., 1996) and chemistry (Murphy and Spudich, 1998). For our design interface, three myosin design parameters are considered with ranges informed by protein engineering experiments: (1) a myosin's lever arm length $\mathbf{l}$ which ranges from 4nm to 20nm, (2) a myosin's attachment rate $\mathbf{k_{on}}$ that ranges from 800s^{-1} to 4000s^{-1}, and (3) a myosin's detachment rate $\mathbf{k_{off}}$ that ranges from 400s^{-1} to 2000s^{-1}. Our interface allows variation of the number of myosins $\mathbf{N_{myo}}$ interacting with the filament that ranges from 20 to 100 myosins.

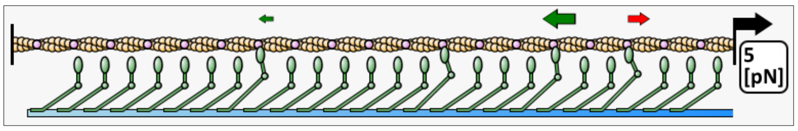


Figure 2. Rendering of Myosins Interacting in a Virtual Environment

The equations required to describe parameter influences across multiple scales are extensions of the Howard (2001) model. We assume an ideal quantity of energy $\boldsymbol{\varepsilon_{atp}}$ (62.5zJ) that all myosins utilize as their lever arms of length $\mathbf{l}$ rotate $\boldsymbol{\theta}$ (assumed as 30 degrees) relative to their head to determine their stiffness $\boldsymbol{\kappa}$:

$$\boldsymbol{\kappa} = \frac{\varepsilon_{atp}}{\left(\left[l \cdot sin(\theta)\right]^2 + l \cdot \left[1 - cos(\theta)\right]^2\right)}. \tag{1}$$

System force is a function of all input variables in addition to the filament velocity $\mathbf{v}$, spacing of myosin binding sites on actin $\mathbf{d}$ (assumed as 36 nm), and binding site size $\mathbf{x_z}$ (assumed as 1nm):

$$F_{sys} = N_{myo} \cdot \kappa \cdot \frac{l \cdot sin(\theta) + \frac{v}{k_{off}}}{\frac{d}{1 - exp\left(\frac{k_{on} \cdot x_z}{v}\right)}} \cdot \frac{\left[l \cdot sin(\theta)\right]^2 - 2 \cdot \left(\frac{v}{k_{off}}\right)^2}{2 \cdot l \cdot sin(\theta) + \frac{2 \cdot v}{k_{off}}}. \tag{2}$$

It is not possible to solve for velocity as a function of force and therefore velocity is found through iteration of equation 2 for a single design embodiment. We assume a system force of 10pn for all applications. The number of myosins attached to actin on average is:

$$N_{att} = N_{myo} \cdot \frac{l \cdot sin(\theta) + \frac{v}{k_{off}}}{\frac{d}{1 - exp\left(\frac{k_{on} \cdot x_z}{v}\right)}}. \tag{4}$$

The average number of myosins attached is a stability metric of the system, and it is suggested that at least two or more myosins must remain attached on average in order to maintain constant directionality of the actin filament and avoid system dissociation due to diffusion (Uyeda et al., 1990). The amount of ATP a system utilizes which is a metric of its system energy is provided by:

$$E_{sys} = N_{myo} \cdot \frac{v}{\frac{d}{1 - exp\left(\frac{k_{on} \cdot x_z}{v}\right)}}. \tag{5}$$

It is not necessary to calculate the precise area of the system for the purposes of this study, and a relative area $\mathbf{A_{sys}}$ is found as a proportion between the number of myosins and lever arm length:

$$A_{sys} \propto \frac{N_{myo}}{l}. \tag{6}$$

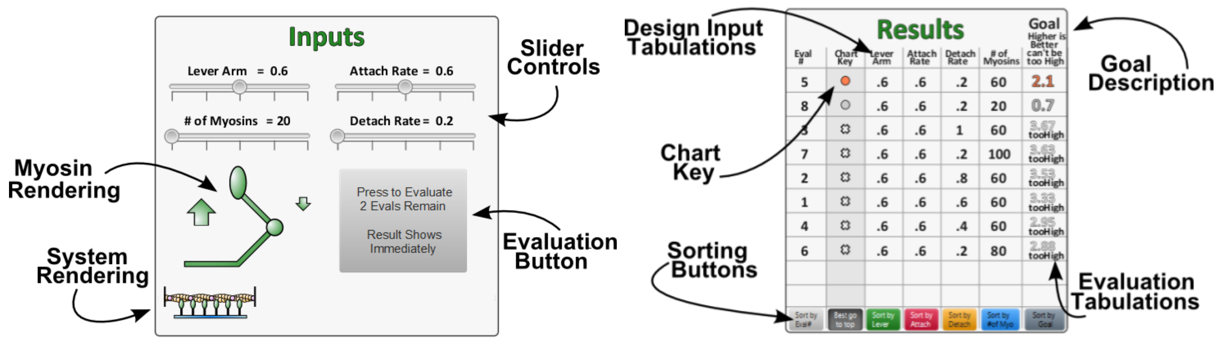
These equations were chosen because they enable the mathematical model to be expressed as an NLP optimization problem. In the NLP space, input and output variables are identified and constraints placed on the values of each variable. The complexity of all the parameter influences makes it difficult to predict how changing one input may affect the output of a chosen goal/objective variable.

3 SOFTWARE INTERFACE

Our visual design tool contains sequences of reconfigurable scenes that house varied interactive modules. The tool is implemented in Java and uses JavaFX 2 for graphics renderings, scene creation, and user interaction tools. The presentation throughout Section 3 is presented with consideration of how these tools appear to a user configuring designs formulated as NLP optimization problems.

3.1.1 Design Inputs Module

The design inputs module (left of Figure 4) allows a user to alter the values of myosin design parameters with slider bars and evaluate designs with a button. Each of the four design parameters from Section 2.2 have a corresponding graphic that refreshes to reflect one of five slider values (represented as ticks on the slider bars), they are: (1) a myosin's lever arm length is represented by the height of the myosin graphic, (2) a myosin's attach rate is represented by the size of the 'up' arrow to the left of the myosin, (3) a myosin's detach rate is represented by the size of the 'down' arrow to the right of the myosin, and (4) the number of myosins is reflected by the number of myosins in the system rendering graphic. The lever arm, attach rate, and detach rate are represented in a non-dimensionalized quantity to avoid overly confusing units and values for users who are unfamiliar with the system. Each design parameter has five possible values based on the ranges provided in Section 2, therefore 625 design possibilities exist, which is a quite complex design space for users to navigate when considering the counter-intuitive parameter interactions present and limited time a user has to search the space during our timed users studies.



Eval #	Chart Key	Lever Arm	Attach Rate	Detach Rate	# of Myosins	Goal Higher is Better can't be too High
5	●	.6	.6	.2	60	2.1
8	○	.6	.6	.2	20	0.7
3	✕	.6	.6	1	60	3.67 tooHigh
7	✕	.6	.6	.2	100	3.63 tooHigh
2	✕	.6	.6	.8	60	3.53 tooHigh
1	✕	.6	.6	.6	60	3.33 tooHigh
4	✕	.6	.6	.4	60	2.95 tooHigh
6	✕	.6	.6	.2	80	2.88 tooHigh
Sort by Eval#	Best go to top	Sort by Lever	Sort by Attach	Sort by Detach	Sort by #of Myo	Sort by Goal

Figure 4. Labeled Renderings of Design Inputs (left) and Results Table (right) Modules

3.1.2 Tabulated Outputs Module

The tabulated outputs module (right of Figure 4) records all input and output information for each design evaluation. It consists of a series of columns and rows. Each column represents a different design metric. The columns, from left to right are 'Eval #,' which represents the order a design was evaluated. The 'Chart Key' column has symbols depicting how the outputs are plotted if charts are present. The 'Lever arm,' 'Attach rate,' 'Detach rate,' and '# of Myosins' columns represent their respective design inputs. The 'Goal' column has values of the objective function and details about the constraint (e.g. "Higher is Better, can't be too High" in Figure 5). If there is a second output variable, it appears in the column to the right of the 'Goal' column. Each row represents a single design evaluation with the exception of the top row being a header and the bottom row housing sort buttons.

Sort buttons always move higher value designs with respective to a column to the top, with the exception of the 'chart key' column which has no values. The 'Chart Key' column is always sorted with better designs appearing higher (i.e. maximization problems have highest values at the top and minimization at the bottom; infeasible designs are always at the bottom). In the case of two designs having the same value for a sorting metric, a secondary sort is done with respect to the goal variable.

3.1.3 Plotted Outputs Module

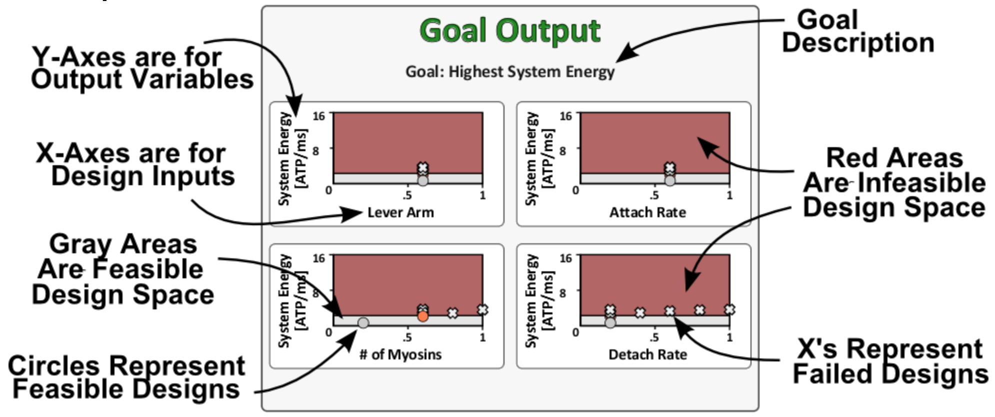

Goal Output
Goal: Highest System Energy
System Energy [ATP/ms]
16
8
0
.5
1
Lever Arm
Attach Rate
# of Myosins
Detach Rate
Y-Axes are for Output Variables
X-Axes are for Design Inputs
Gray Areas Are Feasible Design Space
Circles Represent Feasible Designs
Goal Description
Red Areas Are Infeasible Design Space
X's Represent Failed Designs

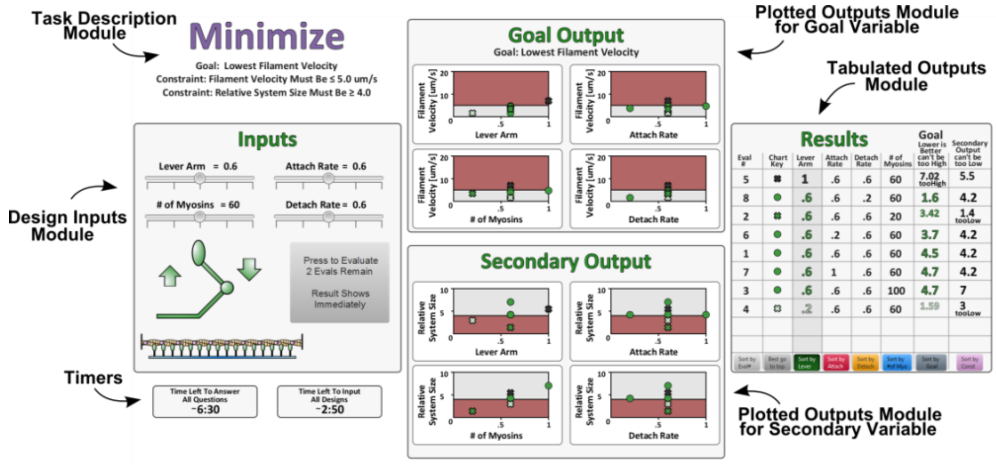
Figure 5. Labeled Rendering of Plotted Outputs Module

The plotted outputs module contains one chart for each input parameter as x-axes and a common output variable for y-axes. Therefore, for a single design evaluation there is one y-value corresponding to four different x-values that appears once on each of the four charts. At the top of the module is a reminder of the goal for a given design task. The charts also have red or gray areas depending on the constraints for a task; red areas represent the infeasible design space. When plotted, outputs for feasible designs appear as circles while infeasible designs appear as crosses.

3.3 Design Scenes with Module Integration

A complete design scene consists of an integrated set of all mentioned interface modules as rendered in Figure 6. In this case, there are two output variables being tracked with their own set of constraints.

Therefore, outputs for a design appear in up to three locations: each plotted output module and the tabulation module. Timers are also included for user studies.



Results

Eval #	Chart Key	Lever Arm	Attach Rate	Detach Rate	# of Myosins	Goal Lower is Better can't be too High	Secondary Output can't be too Low
5	✖	1	.6	.6	60	7.02 tooHigh	5.5
8	●	.6	.6	.2	60	1.6	4.2
2	✖	.6	.6	.6	20	3.42	1.4 tooLow
6	●	.6	.2	.6	60	3.7	4.2
1	●	.6	.6	.6	60	4.5	4.2
7	●	.6	1	.6	60	4.7	4.2
3	●	.6	.6	.6	100	4.7	7
4	✖	.2	.6	.6	60	1.59	3 tooLow

Figure 6. Labeled Screen Capture of Design Scene Integrating all Interface Modules

4 DESIGN TOOL ASSESSMENT WITH HUMAN PARTICIPANTS PROTOCOL

In order to validate the usefulness of the design interface, a user study protocol was developed that measured user design performance with the tool in real-time (Kind) and delayed feedback (Wicked) environments for problems of varying complexities. The feedback conditions allow for determining how beneficial it is to incorporate real-time feedback in the tool and also simulates real-world design scenarios when obtaining information may be costly. For example, when performing wetlab experiments there is a trade-off in how many new system configurations are built in relation to when they are measured (since both processes, and transitioning may take long periods of time). Additionally, assessing users' problem solving strategies could inform agent-based algorithms. Participants were also asked a series of post-study questions about the tool's utility.

4.1 Design Tasks

There were six NLP formulations for the myosin system produced, separated into three categories depending on the complexity of their problem space (Table 1). Each task had a design goal with respect to one output variable as an objective function and then constraints either on the goal output or a secondary output. Designs are all scored relative to the global objective function; if any constraint was violated the design was considered infeasible and given a score of zero. Each constraint added to the task effectively cuts the feasible design space by 50%, which was implemented to maintain consistency in design task difficulty. For two constraint tasks, the secondary constraint is the only active constraint limiting global optimality while the constraint on the goal variable is inactive.

Table 1. Six NLP Design Tasks with Varied Numbers of Variables and Constraints

Design Task	Design Goal	Goal Constraint	Secondary Constraint
A1	Highest System Energy Use	≤ 2.3	none
A2	Lowest Avg# of Attached Myos	≥ 5.7	none
B1	Highest Filament Velocity	none	System Energy Use ≥ 2.3
B2	Lowest System Energy Use	none	Avg# of Attached Myos ≥ 5.0
C1	Lowest Filament Velocity	≤ 5.0	System Size ≥ 4.0
C3	Highest Avg# of Attached Myos	≥ 5.4	Filament Velocity ≤ 3.2

To ensure that the design spaces represented problems of roughly the same difficult for each pairing of design tasks, all 625 designs were evaluated in every space and graded relative to the global optimum. Figure 7 displays the proportion of designs that had at least a given relative objective function for each task. As expected, more highly constrained designs have fewer feasible designs, and less designs near the optimal solution. More critically, the pairs of problems were very close in difficulty.

4.2 Participant Grouping and Task Sequencing

The user study consisted of four participant groupings that each received a unique sequence of tasks to form a within subjects experiment that investigates the effects of feedback and task complexity on solution quality (Table 2). Task Sequences One and Two order tasks from A1 to C2 and alternate tasks between Wicked (delayed feedback) and Kind (real-time feedback), with differing initial feedback conditions. Wicked and Kind conditions are common across many complex systems, and refer to the availability of information received once a system is perturbed (Pretz, 2008). Sequences Three and Four reversed the ordering of Sequences One and Two. If feedback is Kind, output information about a design is plotted in real-time. However, if feedback is Wicked then output information is only plotted in real-time after eight evaluations are conducted. For each task, a user is allowed ten evaluations and a maximum time of 3.5 minutes. Sixteen mechanical engineering students (3rd/4th year seniors and graduate students) were recruited from Carnegie Mellon University and randomly and evenly assigned to each of the four study conditions.

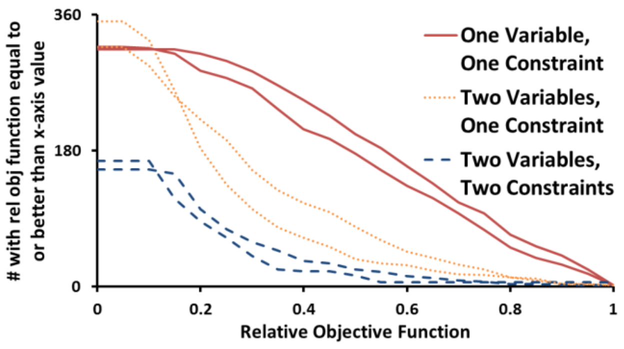


Figure 7. Design Space Comparisons

Table 2. Four Different Sequences of Design Tasks and Feedback Conditions

	First Task	Second Task	Third Task	Fourth Task	Fifth Task	Sixth Task
Sequence One	A1: Kind	A2: Wicked	B1: Kind	B2: Wicked	C1: Kind	C2: Wicked
Sequence Two	A1: Wicked	A2: Kind	B1: Wicked	B2: Kind	C1: Wicked	C2: Kind
Sequence Three	C2: Kind	C1: Wicked	B2: Kind	B1: Wicked	A2: Kind	A1: Wicked
Sequence Four	C2: Wicked	C1: Kind	B2: Wicked	B1: Kind	A2: Wicked	A1: Kind

5 RESULTS OF DESIGN TOOL ASSESSMENT

Results were aggregated by task complexities and feedback conditions; findings demonstrate that real-time feedback enhanced solution quality. Post-study responses were also analyzed.

5.1 Design Task Performance

All designs were graded relative to the optimal for a given problem. For each set of two tasks of a given level of complexity, data was averaged and standard error was found for all Kind and Wicked environments separately. An automated script was also run and averaged for solving the problems by inputting solutions randomly and repeating the process under the same rules as the users for 10,000 runs, and acts as a control. Since Kind and Wicked conditions varied between problems for a user, each participant is represented once in both the Wicked and Kind aggregations for a pair of tasks and the user acts as their own control for individual variability in human problem solving performance.

The best solution at each evaluation point was also aggregated, based on a user's best design by that particular evaluation number. For example, if a user found a design with a relative objective function of 0.5 for their first evaluation and then 0.25 for their second evaluation, the first evaluation value is used for data aggregation at evaluation number 2 because it is their best design by that point in the study. Figure 8 shows three scatter plots demonstrating the differences between Kind and Wicked conditions for each pair of tasks in addition to a bar graph that summarizes the average highest design score users had at the end of all evaluations (equivalent to the tenth evaluation on the scatter plot). Infeasible designs are given a score of zero and the error bars indicate the standard error of the mean.

For all three task complexities, the first evaluation value of both the Kind and Wicked groups are similar and then the Kind group rapidly increases in solution quality until about the fourth evaluation. Also, the random solver steadily finds better solutions at a similar rate and is outperformed by the Kind group in all tasks, and the Wicked group in the two variables two constraints task. For the one variable one constraint problem, the Kind group stops increasing in solution quality after the fourth evaluation because they have hit the ceiling for the problem, while the Wicked group remains increasing at a much lower rate. The ceiling is reached because this is the easiest set of design tasks and many users in the Kind group found the optimal solution in the design space. In the second and third problems a ceiling is not reached, although the Kind group's rate of increase in solution quality decrease sharply after the fourth evaluation and remains roughly equal to the rate of the Wicked group for the rest of the evaluations. It is possible that users in the Kind group are more able to quickly avoid highly suboptimal solutions due to the immediate feedback that occurs with much more frequency at the beginning of a design task, when it may be easier to find much higher quality solutions more quickly. Similar growth rates afterwards may be due to random searches always providing a steady increase in solution quality if no ceiling is reached.

The bar charts show that at the end of a task, the Kind group always has found a better solution on average than the Wicked group and this difference increases as more variables and constraints are added to the task. That is, the benefit of the feedback feature of the design tool (i.e. the effect of Kind vs. Wicked) is larger for the more complex tasks than the less complex tasks (see Figure 8 bar chart). Counter-intuitively, the Kind group does approximately as well on the two variables and two constraints problem as on the two variables and one constraint problem. We theorize this occurs because the extra constraint actually simplifies the design search even though it increases the complexity of the problem statement. It simplifies the space because the extra constraint is a non-active constraint on the goal output and therefore informs the users of what are considered low values for the goal. This theory is further supported when considering the Wicked group does worse on this task because of the smaller feasible design space (and therefore more 0 scoring designs) and the user gains no useful feedback for how to avoid the extra infeasible designs.

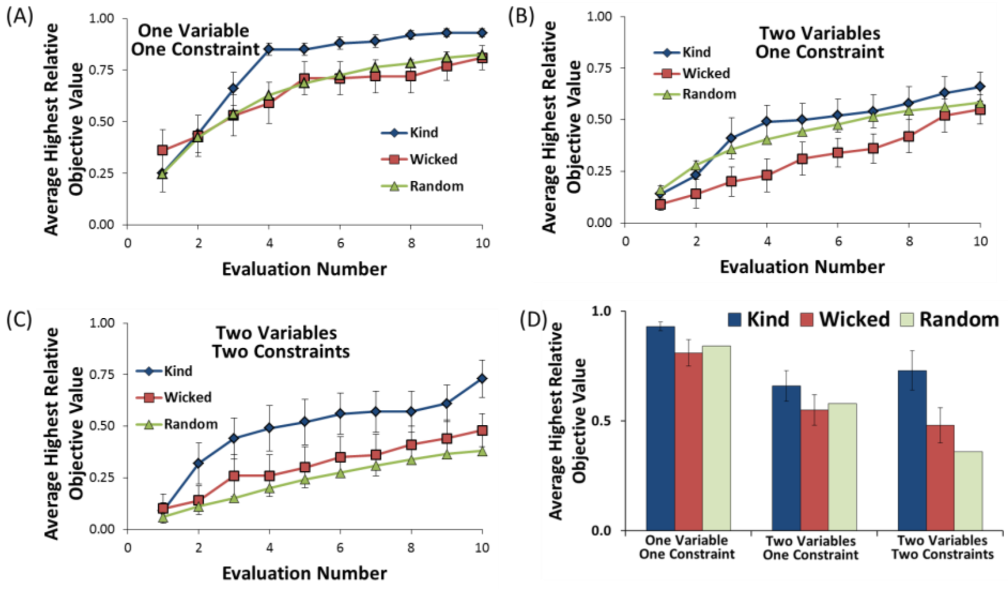


Figure 8. Design Evaluation Results

We further support our theory by aggregating users' strategies for each problem type and demonstrate that users' search strategies were different in Wicked than in Kind environments and affects how often better solutions are found. We tracked how much a user changes their design with respect to all of their previous solutions (small changes allow for better causal inferences) by counting how many 'ticks' (each design input has five 'ticks' on a GUI slider describing its configuration, see Figure 4) were different in comparison to design inputs for all previous solutions. For instance, if a user had design inputs with tick values of 3-3-3-3, 3-3-3-1, and 3-3-1-3, the comparison of the 1st and 2nd inputs would have a total difference of two, while the minimum difference between the third and all previous solutions would also be a total difference of two because it has a total of two ticks difference from the first solution and four ticks difference from the second solution. Our findings show that users in the Wicked condition made larger changes to their designs (Figure 9A). Users in the Wicked condition were also less likely to improve their global best solution with each guess (Figure 9B), which was tracked by keeping a running total of how often a user had improved their design relative to their best solution found for each evaluation number. These findings suggests that users in the Kind condition benefitted from their real-time feedback by utilizing information to make smaller, more useful changes to their designs.

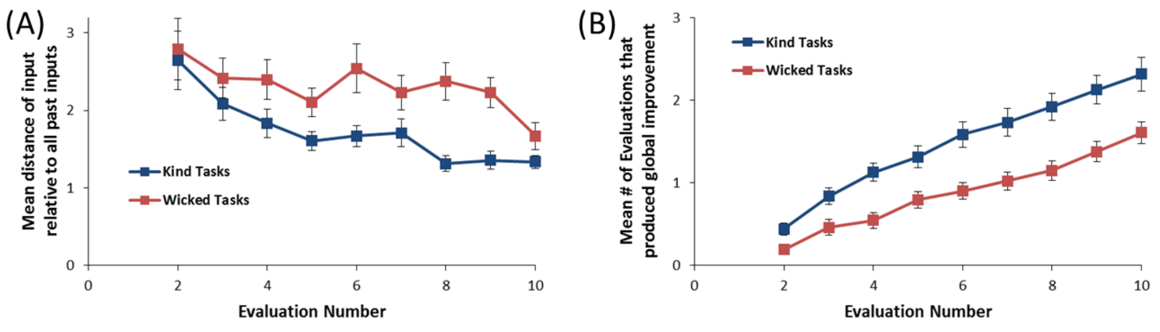


Figure 9. User Search Strategies

5.2 Post-study Questions

After participants were finished with all of their design tasks, they were asked open-ended questions. Each answer was categorized into a different bin of answers (i.e. each bin reflects a positive/negative/indifferent response) depending on the question context and response variability.

The first post-study question asked "Did you tend to use the charts or results table more," and results show that the majority of participants (70%) preferred the results table more. One participant with a high preference for the table responded "*I used the results in the table more… because I could quickly see if a design was meeting the constraints or not.*" Whereas one of the few participants with preference for the plots responded "*Charts…I found it annoying to identify which column corresponded to which parameter, because all of the numbers looked the same…*" Thus, the software modularity is beneficial because not all users prefer to explore complex systems in the same manner. The high preference for the table suggests there could be improvements made to the plots, which was investigated by the second question.

The second post-study question inquired "Was it difficult to read data on the charts?" and it was found that most participants found the charts very difficult (30%), or at least somewhat difficult to use (45%). One common complaint was the charts were difficult to read "*especially when the points were clustered together.*" Another complaint said it was "*hard to make a judgment on the optimal design which is governed by more than one parameter.*" These difficulties arise because there were many parameters to plot and consider simultaneously; in the future it may be possible to give more control over the plot visualization such as zooming in and out or changing axes labels to explore different aspects of variable interactions more easily. One of the users that enjoyed the charts stated "…*it made the table even more helpful because I could quickly evaluate my data to try and figure out a given trend.*" This suggests that although many users preferred the tables over charts, including both may still be advantageous as they provide qualitatively different forms of information and users may benefit from a synergistic effect of having them both present.

The final question asked "Did you find sorting the results useful?" and there was an overwhelming positive response (85%) with answers such as "*Definitely. The buttons were straightforward and labeled well.*" This response demonstrates that the feature was used by designers and could also explain why tables were preferred more since the sorting functionality was tied directly to them. The sorting functionality could have been a big contributor to the Kind group performing better, since sorting was only useful for comparing output values once feedback was received.

7 CONCLUSION

In this study a modular design tool was developed to visually aid engineers in configuring designs of complex systems and study their strategies for solving NLP problems. The tool was implemented in the specific domain of myosin-based nanotechnologies, which addresses mechanical and spatial considerations that are required for embodiment of biological systems, but are often not considered in biochemically-focused design tools. The modular nature of our tool ensures that it is easily extendible to future domains and applications of complex systems.

We conducted a user study to determine whether the tool aided their design of myosin systems and found that it's real-time feedback (Kind) increased solution quality in comparison to situations where feedback was delayed (Wicked). Findings suggest that Kind feedback is increasingly helpful in more complex problem statements, and that users in the Kind condition were more likely to make smaller, helpful changes to their design inputs. We also found through post-study questions that users preferred different aspects of the tool, so modularity could enable tailoring to individual preferences.
We envision future implementations of our modular tool to enable users to reverse engineer complex systems and design scientific experiments, which is highly beneficial in design applications that depend on results from wet-lab experiments, where obtaining information describing system performance may be costly in terms of time and computational effort (i.e., Wicked conditions). Additionally, findings from how users successfully solved problems could inform future agent-based algorithms and these scientific contributions should aid in further enabling engineers to find solutions for high performance technologies in design applications of ever increasing complexity.

ACKNOWLEDGMENTS

Funding support was provided by the National Defense Science and Engineering Fellowship and the National Science Foundation under grant number CMMI-1160840. The authors thank Patra Virasathienpornkul, Tiffany Ho, and Matthew Wood for providing feedback on the study during its development. Patra additionally helped recruit participants during the experimental phase of the study.

REFERENCES

Agarwal, A., and H. Hess (2010). Biomolecular motors at the intersection of nanotechnology and polymer science. *Prog. Polym. Sci.*, Vol. 25, pp. 252-277.
Anson, M., M. A. Geeves, S. E. Kuzawa, and D. J. Manstein (1996). Myosin motors with artificial lever arms. *The EMBO Journal,* Vol. 15, pp. 6069-6074.
Bach, A. D., J. P. Beier, J. Stern-Staeter, and R. E. Horch (2004). Skeletal muscle tissue engineering. *Journal of Cell. Mol. Med.,* Vol. 8, pp. 413-422.
Bakewell, D. J. G., and D. V. Nicolau (2007). Protein linear molecular motor-powered nanodevices. *Australian Journal of Chemistry,* Vol. 60, pp. 314-332.
Chandra, D., F. Bergmann, and H. Sauro (2009). TinkerCell: modular CAD tool for synthetic biology. *Journal of Biological Engineering,* Vol. 3, pp. 1-17.
Chen, Z., and D. Klahr (1999). All other things being equal: Acquisition and transfer of the control of variables strategy. *Child Development,* Vol. 70, pp. 1098-1120.
Egan, P., J. Cagan, C. Schunn, and P. LeDuc (2013). Design of complex biologically-based nanoscale systems using multi-agent simulations and structure-behavior-function representations. *ASME Journal of Mechanical Design (accepted).*
Egan, P., P. LeDuc, J. Cagan, and C. Schunn. (2011) A design exploration of genetically engineered myosin motors. ASME Design Automation Conference, Washington DC.
Guckenheimer, J., and J. M. Ottino (2008). Foundations for Complex Systems Research in the Physical Sciences and Engineering, Report from an NSF Workshop.
Heinemann, M., and S. Panke (2006). Synthetic biology-putting engineering into biology. *Bioinformatics,* Vol. 22, pp. 2790-2799.
Hmelo-Silver, C. E., S. Marathe, and L. Liu (2007). Fish swim, rocks sit, and lungs breathe: Expert-novice understanding of complex systems. *Journal of the Learning Sciences,* Vol. 16, pp. 307-331.
Koenderink, G. H., Dogic, Z., Nakamura, F., Bendix, P. M., Mackintosh, F. C., Hartwig, J. H., Stossel, T. P. & Weitz, D. A. (2009). An active biopolymer network controlled by molecular motors. *PNAS,* Vol. 106, pp. 15192-15197.
Kuhn, D., K. Iordanau, M. Pease, and C. Wirkala (2008). Beyond control of variables: What needs to develop to achieve skilled scientific thinking? *Cognitive Development,* Vol. 23, pp. 435-451.
Laubenbacher, R., V. Hower, A. Jarrah, S. Torti, V. Shuleav, P. Mendes, F. Torti, and S. Akman (2009). A systems biology view of cancer. *Biomchimeca et Biophysica Act,* Vol. 1796, pp. 129-139.
Lv, S., D. M. Dudek, Y. Cao, M. M. Balamurall, J. Gosline, and H. Li (2010). Designed biomaterials to mimic the mechanical properties of muscles. *Nature,* Vol. 465, pp. 69-73.
Murphy, C. T., and J. A. Spudich (1998). Dictyostelium myosin 25-50K loop substitutions specifically affect ADP release rates. *Biochemistry,* Vol. 37, pp. 6738-6744.

Neiman, V. J., and Varghese, S. (2011). Synthetic bio-actuators and their applications in biomedicine. *Smart Structures and Systems,* Vol. 7, pp. 185-198.
Ottino, J. M. (2004). Engineering complex systems. *Nature*, pp. 399.
Pretz, J. (2008). Intuition versus analysis: strategy and experience in complex everyday problem solving. *Memory and Cognition,* 36, pp. 554-566.
Purnick, P., and R. Weiss (2009). The second wave of synthetic biology: from modules to systems. *Nature Reviews,* Vol. 10, pp. 410-422.
Sheridan, C. (2011). Gene therapy finds its niche. *Nature Biotechnology,* Vol. 29, pp. 121-128.
Sneddon, M., J. R. Faeder, and T. Emonet (2011). Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. *Nature Methods,* Vol. 8, pp. 177-185.
Uyeda, T., S. Kron, and J. Spudich (1990). Myosin step size estimation from slow sliding movement of actin over low densities of heavy meromyosin. *J. Mol. Biol.,* Vol. 214, pp. 699-710.
Vattam, S. S., A. K. Goel, S. Rugaber, C. E. Hmelo-Silver, R. Jordan, S. Gray and S. Sinha (2011). Understanding complex natural systems by articulating structure-behavior-function models. *Educational Technology and Society,* Vol. 14, pp. 66-81.
Villalobos, A., J. Ness, C. Gustafsson, J. Minshull, and S. Govindarajan (2006). Gene Designer: a synthetic biology tool for constructing artificial DNA segments. *BMC Bioinformatic,* Vol. 7, pp. 1-8.
Young, E., and H. Alper (2010). Synthetic biology: Tools to design, build, and optimize cellular processes. *Journal of Biomedicine and Biotechnology,* pp. 1-12.