Visualization and Analysis of Multi-Robot Simulations

Laura Strickland, Gregory Cooke, Kartik Acharya, Kumbit Hwang, Nayeon Kim, Hubert Pan

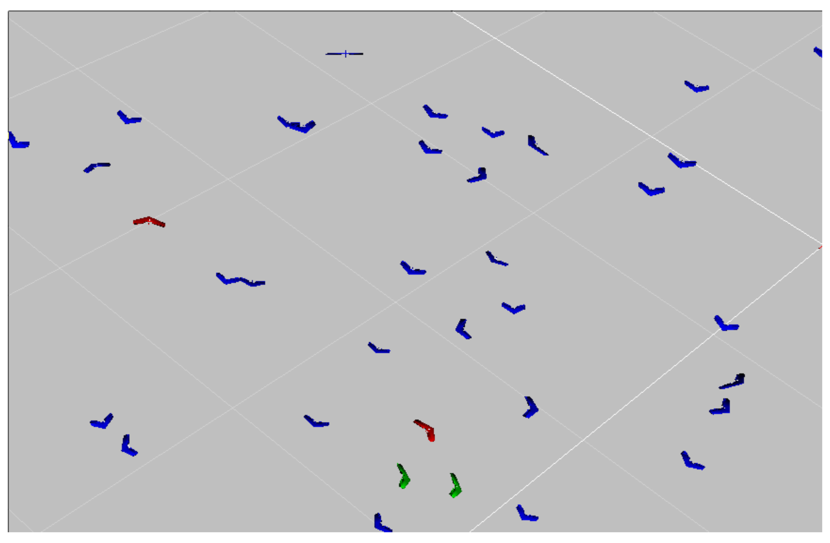
Georgia Tech

CSE 6242, Spring 2018

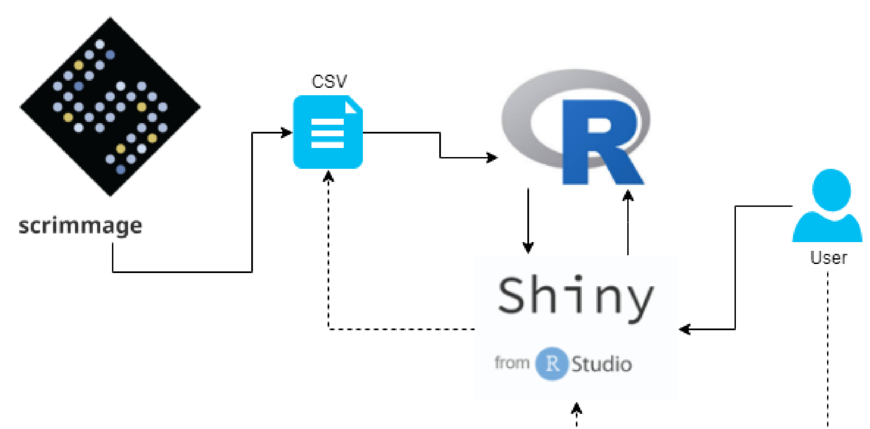
Motivation

There is no effective way currently to **visualize and analyze batch-run data output from SCRIMMAGE**, a multi-agent robotics simulator. Users of the simulator currently spend too much of their time creating custom data aggregation, plotting, and analysis scripts with a variety of languages including Python, Bash, and C++. An effective tool would enable users to gain a meaningful and critical insight on simulation output quickly and provide information to assist them in deciding whether additional simulation runs are needed for statistical significance of their results, and which analytical model(s) to use to interpret the results.

SCRIMMAGE Predator Prey Simulation



Overall Dataflow



Data

How did you get it? What are its characteristics?

We generated our data, which was non-temporal, by running over 600,000 simulations of predator-vs-prey multi-agent simulations in SCRIMMAGE. The metric characterizing simulation performance is the number of prey captured by predators by the end of each simulation. Parameters varied include maximum speeds and turn rates of the predators and prey, as well as whether predators could switch prey targets before their target was captured.

Our first set of raw data occupied over 15GB of disk space and contained results from simulations in which seven different parameters were varied using Latin Hypercube Sampling. After aggregation and cleaning, the data shrank to 181MB.

A smaller dataset (<10MB once aggregated and cleaned) was generated as well to test the stability and convergence analysis aspects of the tool. Only predator speed, prey max turn rate, and predator target switching were varied, and the levels of each parameter were predefined rather than sampled from a range. ~500 simulations were run for each combination of parameter values.

Approach

To tackle this problem we used a two-pronged approach:

1. **Provide a visualization interface** that allows the user to get a broad understanding of the data and how the different variables interact with each other by providing a quick, interactive data summary.

2. Provide **a set of analysis tools** that could be run on the data to provide a quantitative view of various factors such as simulation stability, inter-feature correlations, and model fitting.

How does the visualization work? We built an user-friendly interface using Shiny. **Shiny** is a package for R for constructing interactive web-apps. Both the visualization tab and the analysis plug-ins are written in R and are rendered by Shiny.

The interactive visualization plots a selected metric, or outcome variable, on the y-axis against a selected parameter on the x-axis. If the x-axis parameter is categorical, the data is plotted as a box plot, whereas for numeric parameters, it plots a scatter plot. Other parameters that were varied in simulation appear as sliders or drop-down menus, where the user can constrain their ranges/values to see relationships more clearly.

How does the analysis component work? To assist in **simulation stability** analysis, the tool displays a table of mean, variance, and coefficient of variation (COV) of the outcome variable for each parameter set. In general, the mean value of an outcome variable is considered to be an estimated result value. The outcome of the simulation is, however, governed not only by input parameters but also by randomness. The variance and COV are good indicators of convergence of simulation output. If the variance and COV are too large, the impact of randomness dominates the results. This tool can therefore assist a user in deciding whether he or she should run more simulations to obtain convergent results.

A correlation table shows the correlation coefficient between different features. The **inter-feature correlation** can inform the user's decision regarding which parameters to vary in future simulation runs.

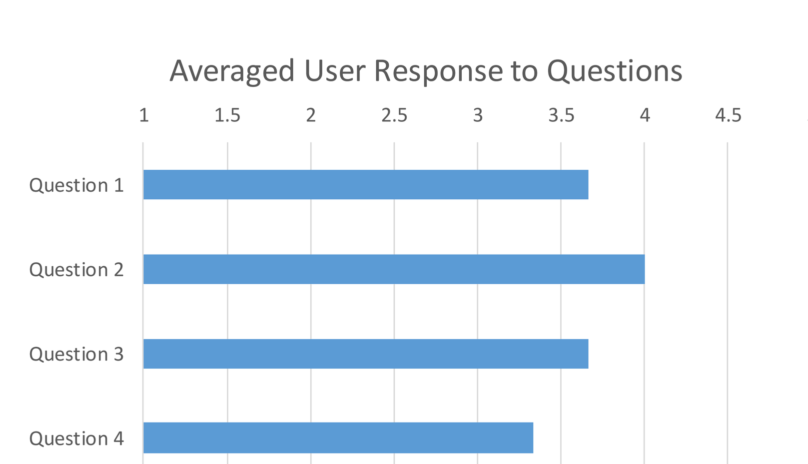
Multiple regression and data mining models are provided as click-and-run options for users to analyze the relationship between parameters and outcomes. Models provided are: (1) multivariate linear regression, (2) linear regression with model selection using AIC, (3) principal component regression using PCA, (4) partial least squares, (5) random forest regression, and (6) neural networks.

Why do you think they can effectively solve your problem? We believe that a **high degree of interactivity with the visualizations and analyses** will allow the user to quickly gain insight from their SCRIMMAGE batch-run data. By making the simulation analysis process less manual, we allow the user to focus on experiment design and analysis soon after their data is generated rather than requiring them to first write and debug one-off visualization and analysis scripts.

What is new in your approaches? This tool brings data visualization and analysis into one comprehensive package. With this tool able to provide quick feedback on existing simulation data, the user can quickly tell whether additional simulations are needed and for which parameter combinations.

We showed the tool to three researchers at GTRI and asked them to answer the following questions with values between 1 (highly disagree) and 5 (highly agree). Averaged responses are plotted to the right.

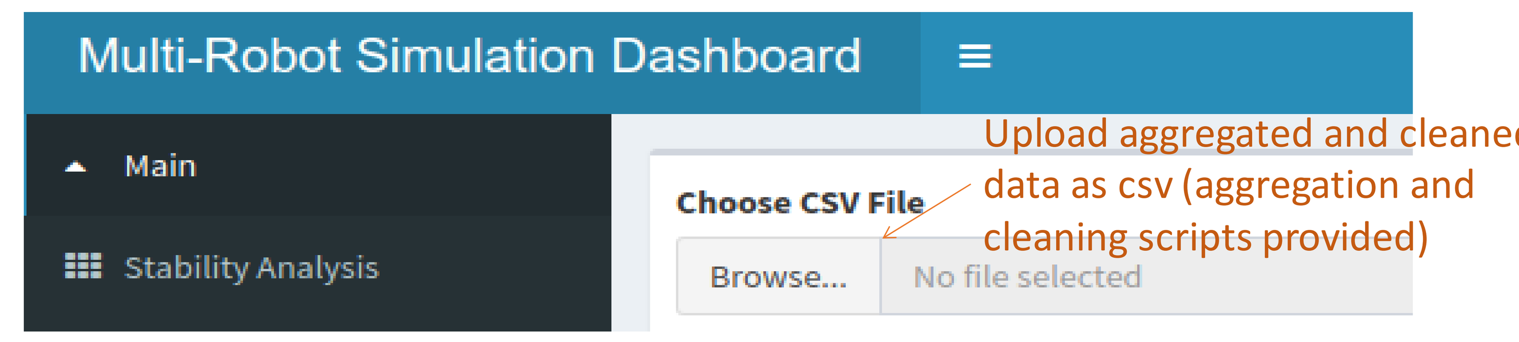
1) This tool is easy to use
2) This tool will save me time in my typical analysis workflow
3) If I need figures for future reports based on experiments in SCRIMMAGE, I would want to use this tool.
4) I enjoyed using this tool.



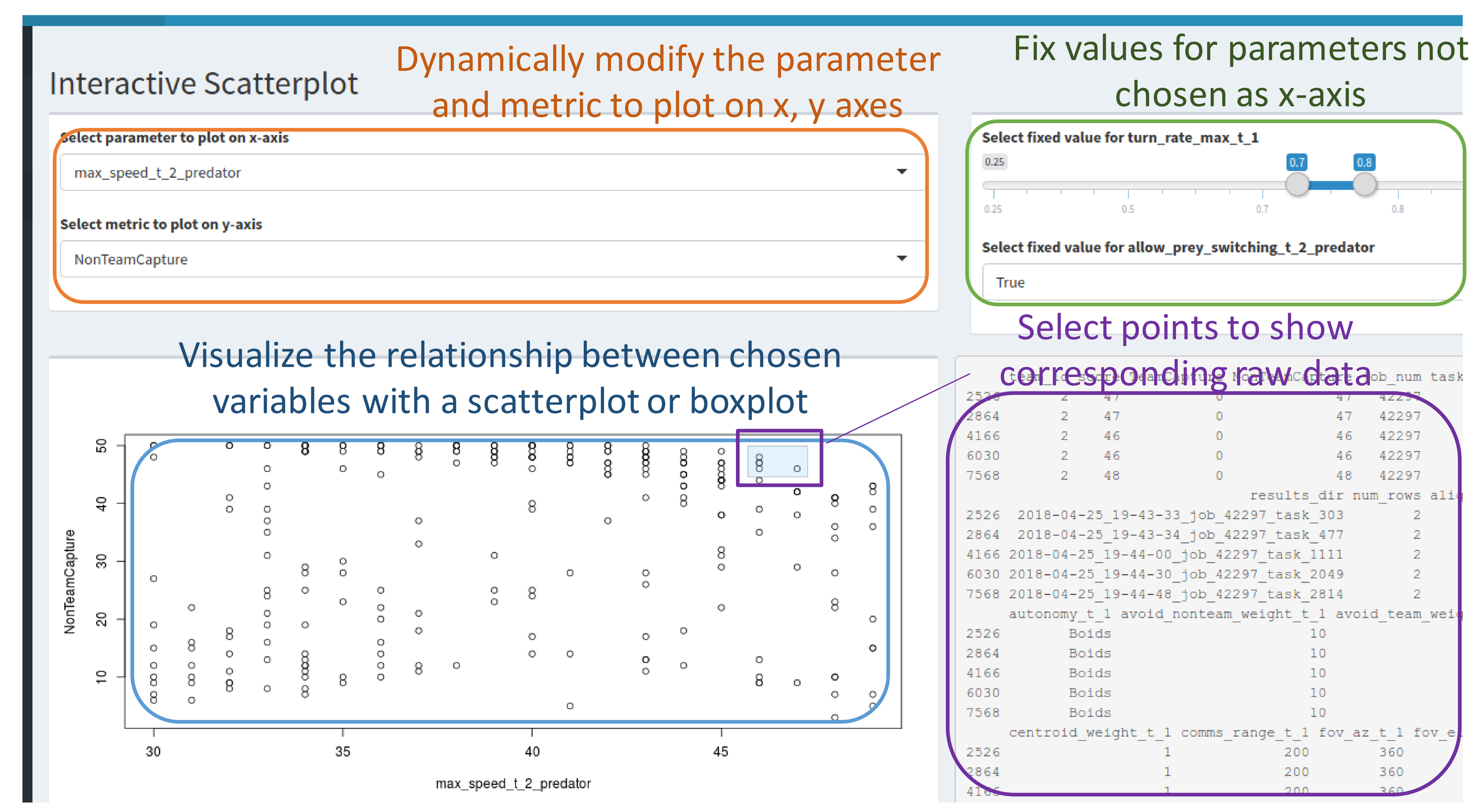
Experiments and Results

We have two separate evaluations: what are the most important parameters for our predator-prey model and do SCRIMMAGE users believe that our tool will be a valuable resource. Here we present the parameter analysis results by following a user's steps. A user study with GTRI employees who use SCRIMMAGE frequently will occur in the coming weeks.

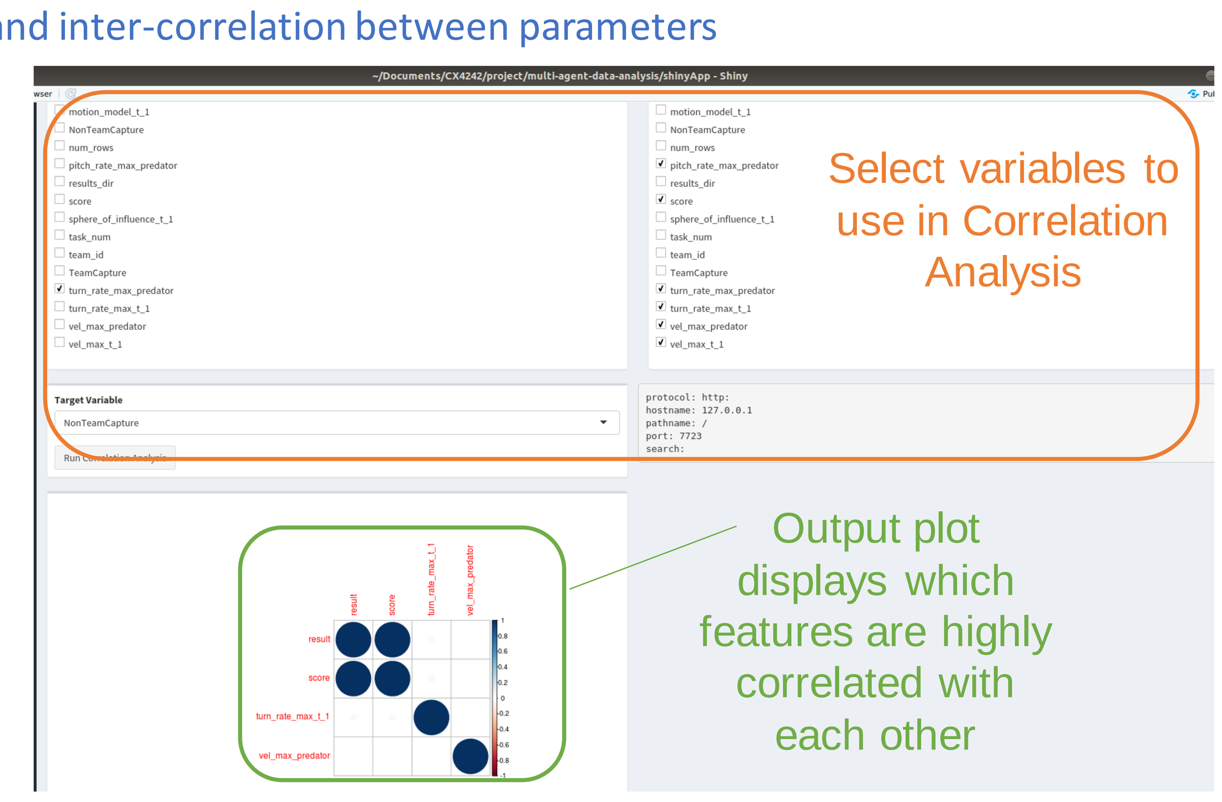
1. Upload simulation data



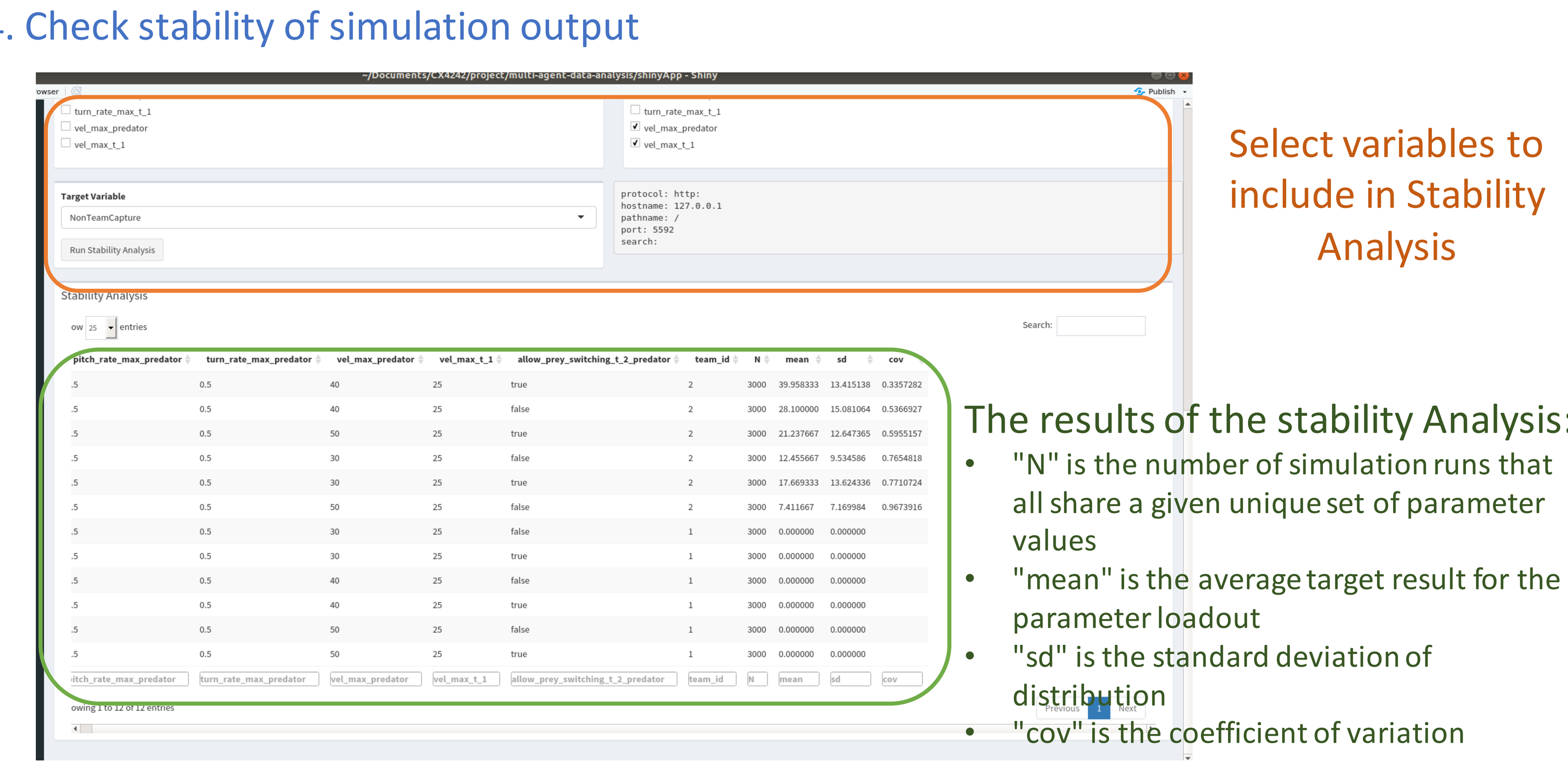
2. Explore and visualize the data



3. Understand inter-correlation between parameters



4. Check stability of simulation output



5. Understand the impact of parameters on the outcome (e.g. Multiple Linear Regression)