January 7, 2008

The following is a guide to the simulation files for the manuscript "Theory Driven Bias in Ideal Point Estimates - A Monte Carlo Study."

To generate the plots in the paper, the files must be run in the following sequence.

1. **calibration.m** and **calibration_un**. These files determine how many status quo points must be drawn in each legislative session, for each specification of the simulation, in order to generate an expected number of successful roll calls equal to 241.
2. **Generate_Output_Normal_House.m** and **Generate_Output_Uniform_House.m**. These files run the simulations and output the estimated ideal points from each simulation run in the folders /outnormal and /outuniform. The first file runs through 16 specifications of the model, changing the variance of the normally distributed random utility shock σ^2. The second file runs through a single specification of the model with uniformly distributed random utility shocks.
3. **BlockzoneWidth_EstPiv_Plots.m**, **BlockzoneWidth_Uniform_EstPiv_Plots.m**, **IdlPt_Means_Uniform_Plots**, **Mean_ClassSucc_Plots**, **Mean_Corellation_Uniform.m**, and **Mean_Plots_Normal.m**. These files produce the plots and figures in the manuscript.

Below is a key to the plots.

Figure	FileName	Creator
1	**Figure1.pdf**	Plot_Example_Final
	Effect of Voting Error on Conditional Voting Probabilities	
2	**TrueVEstimated_0.pdf**	Mean_Plots_Normal
	Mean Estimated vs. True Ideal Points, $\sigma = 0$	
3	**TrueVEstimated_p4.pdf**	Mean_Plots_Normal
	Mean Estimated vs. True Ideal Points, $\sigma = .4$	
4	**class_success.pdf**	mean_classsucc_plots
	Mean Correct Voting and Classification Success as Function of σ	
5	**Mean_Correllation.pdf**	Mean_Plots_Normal
	Mean Correlation of Blockzone Legislators as Function of σ	
6	**BZ_Densities_EstPiv_XX.pdf**	BlockzoneWidth_EstPiv_Plots
	Distribution of Estimated Blockzone Widths, $\sigma = XX$	
7	**BZ_Densities_EstPiv_Un_1.pdf**	BlockzoneWidth_Uniform_EstPiv_Plots
	Distribution of Estimated Blockzone Widths, Uniform Errors	
8	**IdlPt_Means_Uniform_1.pdf**	IdlPt_Means_Uniform_Plots
	Estimated Ideal Points, Party Cartel, Uniform Errors	

The remaining matlab files are accessory functions for running the simulations. They are organized in a modular fashion to permit tweaking the model without changing all of the code. Of particular importance are the functions **generate_session.m** and **generate_runs.m.** The former generates votes from a **t** length legislative session. The latter generates estimated ideal points from **z_iter** separate legislative sessions of length **t**. Both functions take the following parameters -

1. **legislators** - an n by 3 matrix where n is the number of legislators. Must be presorted by rank. The first column is a numerical label, the second column is the legislator's ideal point, and the third column contains the party. The code 100 is Democrats and 200 for Republicans.
2. **which_model** - specifies which complete information game is played
   - 1 - proposer acts as a Romer-Rosenthal agenda setter
   - 2 - pivots act as gatekeepers; if choose not to gatekeep proposer proposes own ideal point
   - 3 - proposer proposes his own ideal point and the pivots may not block any bill
3. **l_piv**, **r_piv**, and **p**. The spatial locations of the left pivot, right pivot, and proposer.

   Together (2) and (3) specify the complete information game. So for example, **which_model**=2, **p**=**l_piv**=*ideal point of median*, and **r_piv**=*ideal point of majority party median* is the party cartel game. Pivotal politics is **which_model**= 1, **l_piv**=*filibuster pivot*, **p**=*median*, and **r_piv**=*veto pivot*.
4. **vote_dist -** distribution of voter utility shocks. Takes 3 values
   - 1 - normally distributed random utility shock with variance **sigma**
   - 2 - uniformly distributed random utility shock. The baseline support of the shock is the minimal support consistent with voting probabilities strictly interior to $[0, 1]$ for all legislators. The variable **sigma** acts as a multiplier that scales the baseline support.
   - 3 - perfect voting.
5. **sigma -** accessory parameter to **vote_dist**
6. **sqdist -** distribution from which status quo points are drawn. Takes 2 values
   - 0 - Drawn uniformly from $[-1/2, 3/2]$
   - 1 - Drawn from a uniform distribution whose lower bound is **lpoint** and whose support length is **supplength**
7. **lpoint** - accessory parameter to sqdist
8. **supplength** - accessory parameter to sqdist