Contents


```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Main Simulation Script for BoundingOverwatch Project with R Integration
% Randomly 10 trials is used to validate predictions
% DFT Parameters:
% phi1 - sensitivity to attribute differences (typically 0.5-2)
% phi2 - memory/feedback strength (0-1)
% tau - decision time steps (integer > 0)
% error_sd - noise standard deviation (σ_ε)
% beta - attribute weights from R estimation
% w -  attention weights (default [0.5;0.5])
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
clear all;
```

Step 1: Import CSV Data

(reference apolloMain_5 amd apolloMain_6 as example for data manipulation) biasData = readtable('user_choices.csv'); % Replace with the path to your data file disp('User bias data imported successfully.'); taskChoice_Data = readtable('user_choices.csv'); % Replace with the path to your data file disp('User task choice data imported successfully.');

```
robotChoice_Data = readtable('G:\My Drive\myResearch\Research Experimentation\Apollo\apollo\data\Bounding_Overwatch_Data\HumanData_Bounding_Overwatch.csv');
% Convert all column headers to lowercase
robotChoice_Data.Properties.VariableNames = lower(robotChoice_Data.Properties.VariableNames);
disp('User robot choice data imported successfully.');

% Randomly select 10 rows (or all rows if fewer than 10)
numRows = height(robotChoice_Data);
randomIndices = randperm(numRows, min(10, numRows));
robotChoice_Data = robotChoice_Data(randomIndices, :);

% Extract robot state attributes dynamically
robot_states = struct();
attributeSuffixes = {'traversability', 'visibility'}; % No leading underscores
for i = 1:3
    for attr = attributeSuffixes
        csvColName = sprintf('robot%d_%s', i, attr{1});  % Matches CSV column names
        structFieldName = attr{1};  % Valid field name
        if ismember(csvColName, robotChoice_Data.Properties.VariableNames)
            robot_states.(['robot' num2str(i)]).(structFieldName) = robotChoice_Data.(csvColName);
        else
            warning('Missing attribute column: %s', csvColName);
            robot_states.(['robot' num2str(i)]).(structFieldName) = NaN(height(robotChoice_Data), 1);
        end
    end
end

% Extract choice data and other metadata
choices = robotChoice_Data.choice;
participant_ids = robotChoice_Data.id;
stake_types = robotChoice_Data.stakes;
time_spent = robotChoice_Data.timeelapsed;
```

```
User robot choice data imported successfully.
```

Step 2: R Bridge Implementation

```
disp('Initializing R bridge...');

% Configure paths
rscript_path = 'C:\Program Files\R\R-4.4.2\bin\x64\Rscript.exe';
r_script = 'G:\My Drive\myResearch\Research Experimentation\Apollo\apollo\example\DFT_Bounding_Overwatch.R';
csvFile = 'G:\My Drive\myResearch\Research Experimentation\Apollo\apollo\data\Bounding_Overwatch_Data\HumanData_Bounding_Overwatch.csv';
outputDir = 'G:\My Drive\myResearch\Research Experimentation\Apollo\apollo\Output_BoundingOverwatch';

% Verify installations
if ~isfile(rscript_path)
    error('Rscript.exe not found at: %s', rscript_path);
elseif ~isfile(r_script)
    error('R script not found at: %s', r_script);
elseif ~isfile(csvFile)
```

```
    error('Input CSV not found at: %s', csvFile);
elseif ~isfolder(outputDir)
    warning('Output folder does not exist, creating: %s', outputDir);
    mkdir(outputDir);
end

% Execute R with JSON output
try
    % Use proper argument formatting
    cmd = sprintf(['"%s" "%s" ', ...
                '-i "%s" -o "%s"'], ...
                rscript_path, r_script, csvFile, outputDir);

[status,result] = system(cmd);

    if status == 0
        % Handle output path (whether directory or file)
        if isfolder(outputDir)
            jsonFile = fullfile(outputDir, 'DFT_output.json');
        else
            jsonFile = outputDir;
        end

        % Parse JSON output
        if exist(jsonFile, 'file')
            jsonText = fileread(jsonFile);
            params = jsondecode(jsonText);

            % Extract parameters with validation
            %Boundedphi1, phi2 parameters
            phi1 = max(0, validateParam(params, 'phi1', 0.5)); % Ensure non-negative
            phi2 = min(max(0, validateParam(params, 'phi2', 0.8)), 1); % Constrain 0-1

            %Raw phi1, phi2 parameters
            %phi1 = validateParam(params, 'phi1', 0.5);
            %phi2 = validateParam(params, 'phi2', 0.8);
            tau = 1 + exp(validateParam(params, 'timesteps', 0.5));
            error_sd = min(max(0.1, validateParam(params, 'error_sd', 0.1)), 1); % still clip here

            % Extract attribute weights
            beta_weights = [
                params.b_attr1;
                params.b_attr2;
                params.b_attr3;
                params.b_attr4
            ];

            % Get initial preferences from ASCs
            initial_P = [
                validateParam(params, 'asc_1', 0);
                validateParam(params, 'asc_2', 0);
                validateParam(params, 'asc_3', 0);
            ];

            disp('Estimated Parameters:');
            disp(['phi1: ', num2str(phi1)]);
            disp(['phi2: ', num2str(phi2)]);
            disp(['tau: ', num2str(tau)]);
            disp(['error_sd: ', num2str(error_sd)]);
            disp('Initial Preferences (from ASCs):');
            disp(initial_P');
        else
            error('R output file not found');
        end
    else
        error('R execution failed: %s', result);
    end
catch ME
    disp('Error during R execution:');
    disp(getReport(ME, 'extended'));
    [phi1, phi2, tau, error_sd] = getFallbackParams();
    beta_weights = [0.3; 0.2; 0.4; 0.5]; % Default weights
    initial_P = zeros(3,1); % Neutral initial preferences
end
```

```
Initializing R bridge...
```

Step 3: MDFT Formulation to Calculate Preference Dynamics

(MDFT calculations based on estimated parameters) Create M matrix from current trial's attributes C11-C14 are consequence attributes for Robot 1 C21-C24 are consequence attributes for Robot 2 C31-C34 are consequence attributes for Robot 3

```
for current_trial = 1:height(robotChoice_Data)
    num_attributes = 4;

    M = [
        robotChoice_Data.c11(current_trial), robotChoice_Data.c12(current_trial), robotChoice_Data.c13(current_trial), robotChoice_Data.c14(current_trial);
        robotChoice_Data.c21(current_trial), robotChoice_Data.c22(current_trial), robotChoice_Data.c23(current_trial), robotChoice_Data.c24(current_trial);
        robotChoice_Data.c31(current_trial), robotChoice_Data.c32(current_trial), robotChoice_Data.c33(current_trial), robotChoice_Data.c34(current_trial)
    ];

    M = M ./ sum(M, 2);  % Normalize each row of M

    attributes = {'C1 - Easy Nav, Low Exposure', 'C2 - Hard Nav, Low Exposure', 'C3 - Easy Nav, High Exposure', 'C4 - Hard Nav, High Exposure'};
    beta = beta_weights ./ sum(abs(beta_weights));
    beta = beta';

    [E_P, V_P, choice_probs, P_tau] = calculateDFTdynamics(...
        phi1, phi2, tau, error_sd, beta, M, initial_P);

    % Display results for the trial
    disp('=== Trial Analysis ===');
    disp(['Trial: ', num2str(current_trial)]);
    disp(['Participant: ', num2str(participant_ids(current_trial))]);
    disp(['Actual Choice: Robot ', num2str(choices(current_trial))]);

    disp('M matrix (alternatives × attributes):');
    disp(array2table(M, ...
        'RowNames', {'Robot1','Robot2','Robot3'}, ...
        'VariableNames', attributes));

    disp('DFT Results:');
    disp(['E_P: ', num2str(E_P', '%.2f  ')]);
    disp(['Choice probabilities: ', num2str(choice_probs', '%.3f  ')]);
    [~, predicted_choice] = max(choice_probs);
    disp(['Predicted choice: Robot ', num2str(predicted_choice)]);
    disp(['Actual choice: Robot ', num2str(choices(current_trial))]);
    disp(' ');

    if predicted_choice == choices(current_trial)
        disp('✓ Prediction matches actual choice');
    else
        disp('✗ Prediction differs from actual choice');
    end

    % Plot evolution
    figure;
    plot(0:tau, P_tau);
    xlabel('Preference Step (\tau)');
    ylabel('Preference Strength');
    legend({'Robot1','Robot2','Robot3'});
    title(sprintf('Preference Evolution (Trial %d)', current_trial));
    grid on;
end
%{
%% Step 4: Output Results
disp('Saving results to CSV...');
output_table = table(E_P, V_P, P_tau(end,:)', ...
                     'VariableNames', {'ExpectedPreference', 'VariancePreference', 'FinalPreferences'});
writetable(output_table, 'results.csv');
disp('Results saved successfully!');
%}
```

```
=== Trial Analysis ===
Trial: 1
Participant: 181700
Actual Choice: Robot 1
M matrix (alternatives × attributes):
              C1 - Easy Nav, Low Exposure    C2 - Hard Nav, Low Exposure    C3 - Easy Nav, High Exposure    C4 - Hard Nav, High Exposure
              ___________________________    ___________________________    ____________________________    ____________________________

    Robot1              0.45463                        0.26797                        0.23203                         0.045371
    Robot2               0.4545                        0.26337                        0.23663                         0.045497
    Robot3              0.45462                        0.22627                        0.27353                         0.045588

DFT Results:
E_P: 5.39   2.08  -7.62
Choice probabilities: 0.965  0.035  0.000
Predicted choice: Robot 1
Actual choice: Robot 1

✓ Prediction matches actual choice
=== Trial Analysis ===
```

Trial: 2
Participant: 214504
Actual Choice: Robot 1
M matrix (alternatives × attributes):

	C1 - Easy Nav, Low Exposure	C2 - Hard Nav, Low Exposure	C3 - Easy Nav, High Exposure	C4 - Hard Nav, High Exposure
Robot1	0.45443	0.30157	0.19843	0.045568
Robot2	0.45475	0.27828	0.22172	0.045249
Robot3	0.45465	0.29843	0.20157	0.045353

DFT Results:
E_P: 4.43 -6.19   1.60
Choice probabilities: 0.944 0.000 0.056
Predicted choice: Robot 1
Actual choice: Robot 1

✓ Prediction matches actual choice
=== Trial Analysis ===
Trial: 3
Participant: 214504
Actual Choice: Robot 1
M matrix (alternatives × attributes):

	C1 - Easy Nav, Low Exposure	C2 - Hard Nav, Low Exposure	C3 - Easy Nav, High Exposure	C4 - Hard Nav, High Exposure
Robot1	0.45446	0.28277	0.21739	0.045382
Robot2	0.45513	0.27949	0.22051	0.044872
Robot3	0.45449	0.28518	0.21497	0.045363

DFT Results:
E_P: 1.80 -3.33   1.37
Choice probabilities: 0.603 0.004 0.393
Predicted choice: Robot 1
Actual choice: Robot 1

✓ Prediction matches actual choice
=== Trial Analysis ===
Trial: 4
Participant: 214504
Actual Choice: Robot 1
M matrix (alternatives × attributes):

	C1 - Easy Nav, Low Exposure	C2 - Hard Nav, Low Exposure	C3 - Easy Nav, High Exposure	C4 - Hard Nav, High Exposure
Robot1	0.45507	0.26728	0.23272	0.044931
Robot2	0.45455	0.28379	0.21621	0.045455
Robot3	0.45473	0.26169	0.23831	0.045274

DFT Results:
E_P: -1.04   3.52 -2.63
Choice probabilities: 0.010 0.988 0.002
Predicted choice: Robot 2
Actual choice: Robot 1

✗ Prediction differs from actual choice
=== Trial Analysis ===
Trial: 5
Participant: 214504
Actual Choice: Robot 2
M matrix (alternatives × attributes):

	C1 - Easy Nav, Low Exposure	C2 - Hard Nav, Low Exposure	C3 - Easy Nav, High Exposure	C4 - Hard Nav, High Exposure
Robot1	0.45464	0.3002	0.1998	0.045364
Robot2	0.45419	0.30019	0.20013	0.045484
Robot3	0.45458	0.29493	0.20507	0.045425

DFT Results:
E_P: 1.46 -0.58 -1.04
Choice probabilities: 0.824 0.108 0.068
Predicted choice: Robot 1
Actual choice: Robot 2

✗ Prediction differs from actual choice
=== Trial Analysis ===
Trial: 6
Participant: 181700
Actual Choice: Robot 2
M matrix (alternatives × attributes):

	C1 - Easy Nav, Low Exposure	C2 - Hard Nav, Low Exposure	C3 - Easy Nav, High Exposure	C4 - Hard Nav, High Exposure

Robot1	0.4546	0.27689	0.22311	0.045396
Robot2	0.45447	0.27255	0.22756	0.045424
Robot3	0.45458	0.27395	0.22605	0.045417

DFT Results:
E_P: 1.79 -1.82 -0.13
Choice probabilities: 0.853 0.023 0.125
Predicted choice: Robot 1
Actual choice: Robot 2

✗ Prediction differs from actual choice
=== Trial Analysis ===
Trial: 7
Participant: 214504
Actual Choice: Robot 2
M matrix (alternatives × attributes):

	C1 - Easy Nav, Low Exposure	C2 - Hard Nav, Low Exposure	C3 - Easy Nav, High Exposure	C4 - Hard Nav, High Exposure
Robot1	0.45458	0.27868	0.22132	0.04542
Robot2	0.45448	0.27546	0.22454	0.045525
Robot3	0.4545	0.29995	0.20029	0.045258

DFT Results:
E_P: -0.65 -3.61   4.10
Choice probabilities: 0.009 0.000 0.991
Predicted choice: Robot 3
Actual choice: Robot 2

✗ Prediction differs from actual choice
=== Trial Analysis ===
Trial: 8
Participant: 214504
Actual Choice: Robot 2
M matrix (alternatives × attributes):

	C1 - Easy Nav, Low Exposure	C2 - Hard Nav, Low Exposure	C3 - Easy Nav, High Exposure	C4 - Hard Nav, High Exposure
Robot1	0.45471	0.28335	0.21665	0.045288
Robot2	0.454	0.26998	0.23077	0.045249
Robot3	0.45398	0.29953	0.20125	0.045242

DFT Results:
E_P: 1.08 -5.75   4.51
Choice probabilities: 0.031 0.000 0.969
Predicted choice: Robot 3
Actual choice: Robot 2

✗ Prediction differs from actual choice
=== Trial Analysis ===
Trial: 9
Participant: 181700
Actual Choice: Robot 2
M matrix (alternatives × attributes):

	C1 - Easy Nav, Low Exposure	C2 - Hard Nav, Low Exposure	C3 - Easy Nav, High Exposure	C4 - Hard Nav, High Exposure
Robot1	0.4548	0.33594	0.16406	0.045201
Robot2	0.45448	0.30089	0.19911	0.045522
Robot3	0.45436	0.31394	0.18606	0.045637

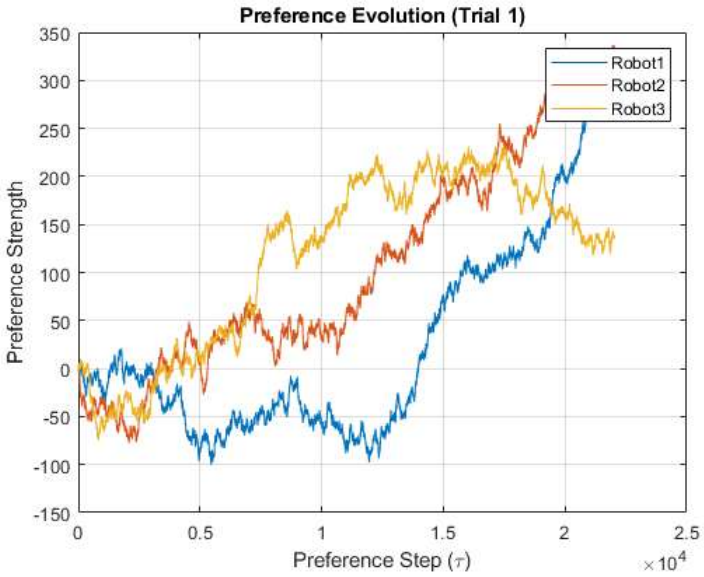
DFT Results:
E_P: 6.01 -5.91 -0.26
Choice probabilities: 0.998 0.000 0.002
Predicted choice: Robot 1
Actual choice: Robot 2

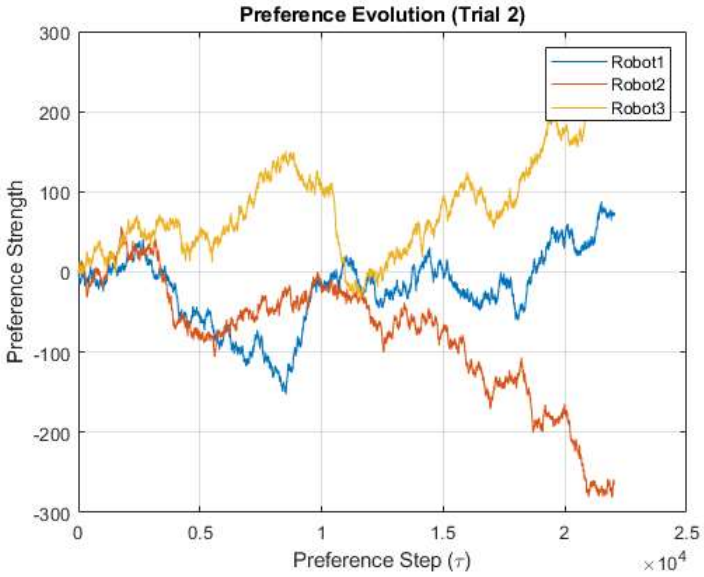
✗ Prediction differs from actual choice
=== Trial Analysis ===
Trial: 10
Participant: 181700
Actual Choice: Robot 1
M matrix (alternatives × attributes):

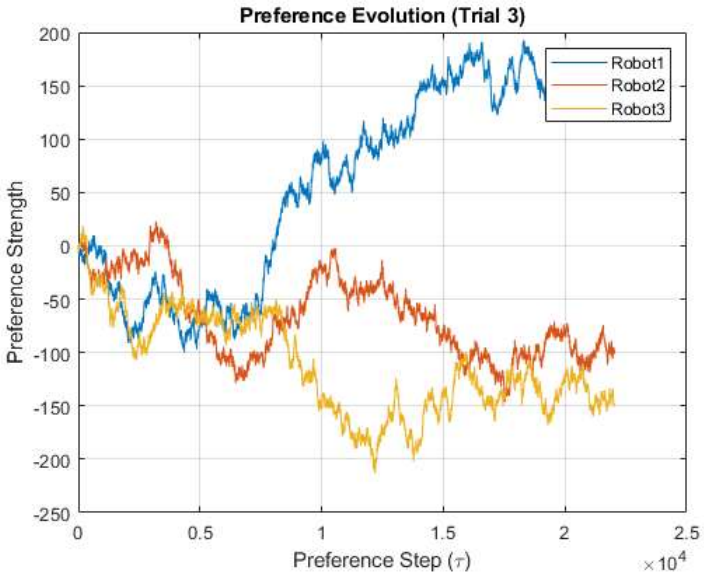
	C1 - Easy Nav, Low Exposure	C2 - Hard Nav, Low Exposure	C3 - Easy Nav, High Exposure	C4 - Hard Nav, High Exposure
Robot1	0.45455	0.26197	0.23803	0.045455
Robot2	0.45451	0.26568	0.23432	0.045487
Robot3	0.45455	0.25071	0.24921	0.045538

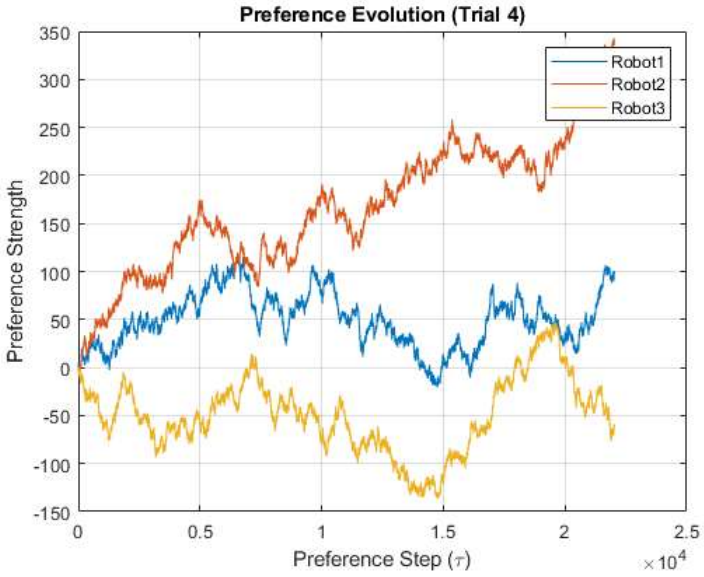
DFT Results:
E_P: 1.72   0.62 -2.50
Choice probabilities: 0.743 0.246 0.011

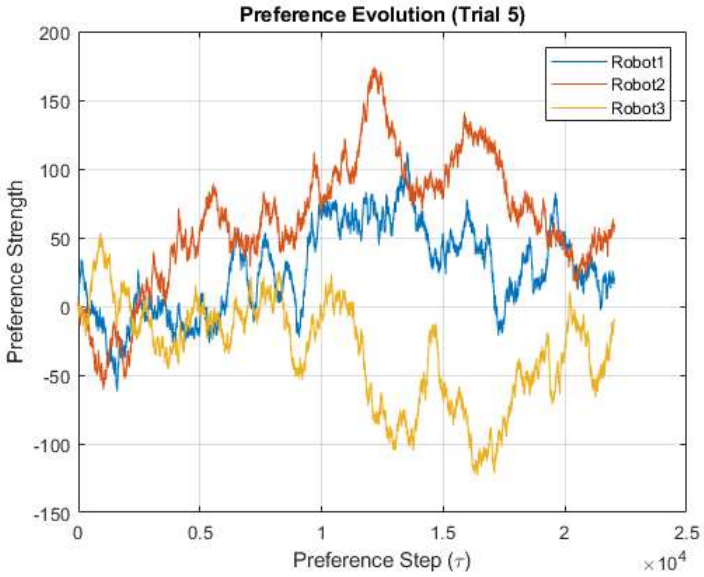
```
Predicted choice: Robot 1
Actual choice: Robot 1

✓ Prediction matches actual choice
```

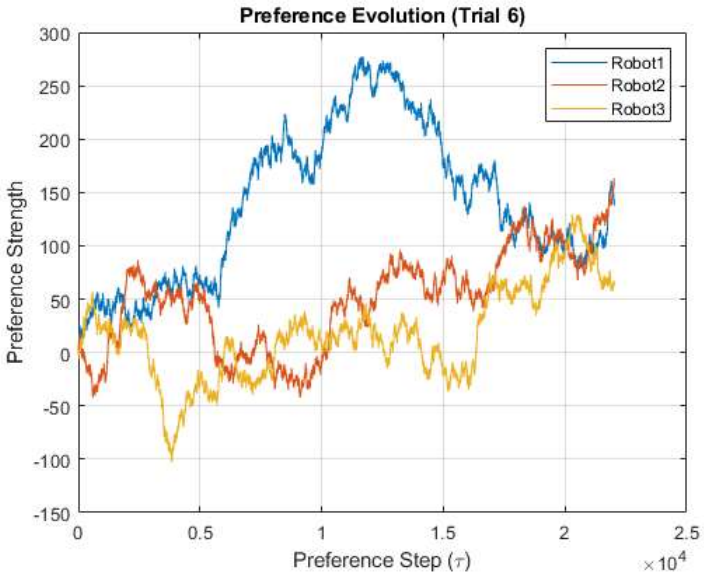


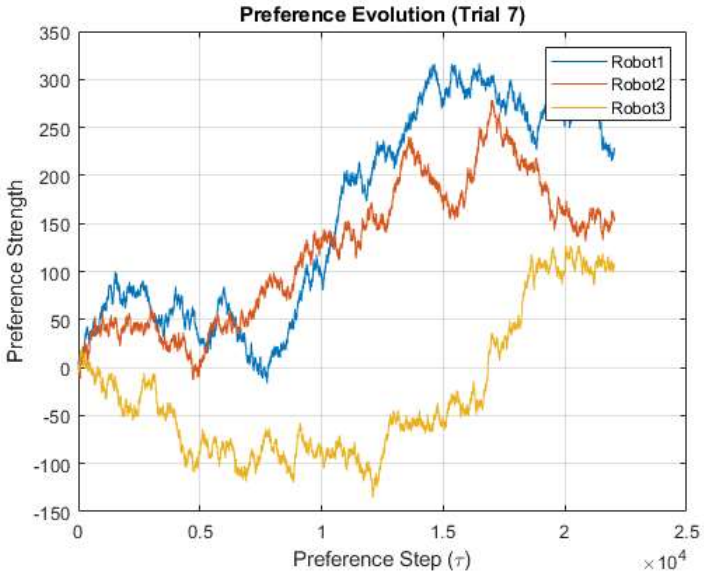


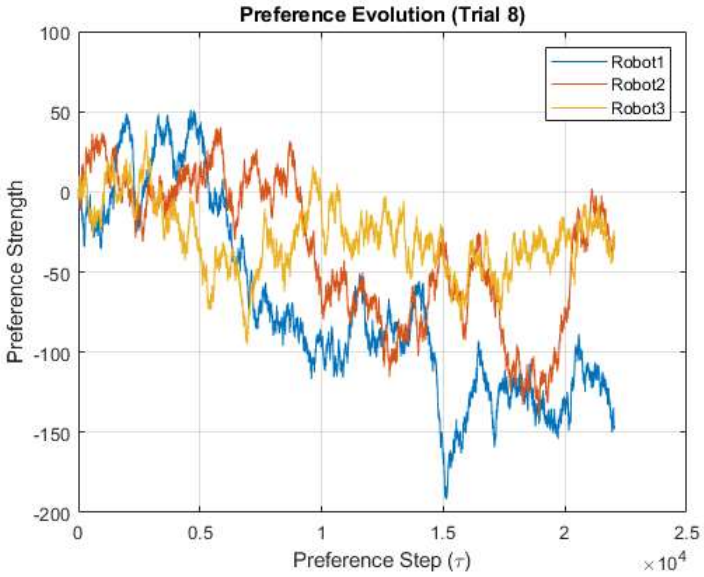

Preference Evolution (Trial 3)
Robot1
Robot2
Robot3
Preference Strength
Preference Step (τ)
$\times 10^4$

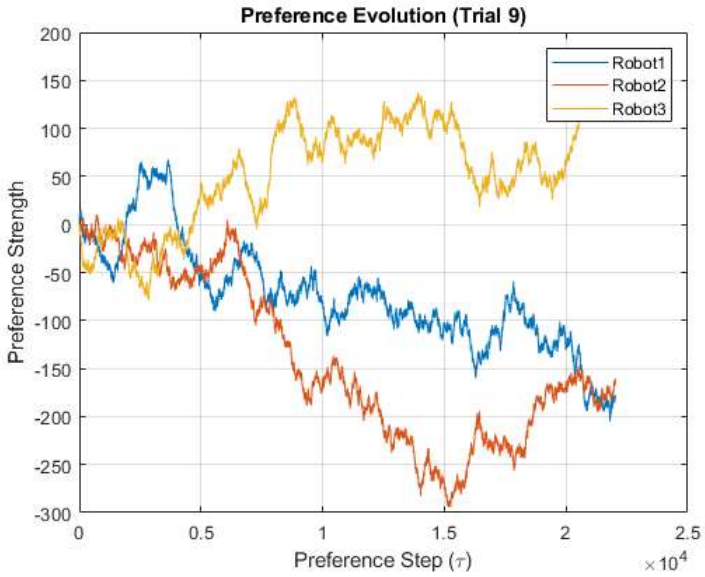

Preference Evolution (Trial 4)
Robot1
Robot2
Robot3
Preference Strength
Preference Step (τ)
$\times 10^4$

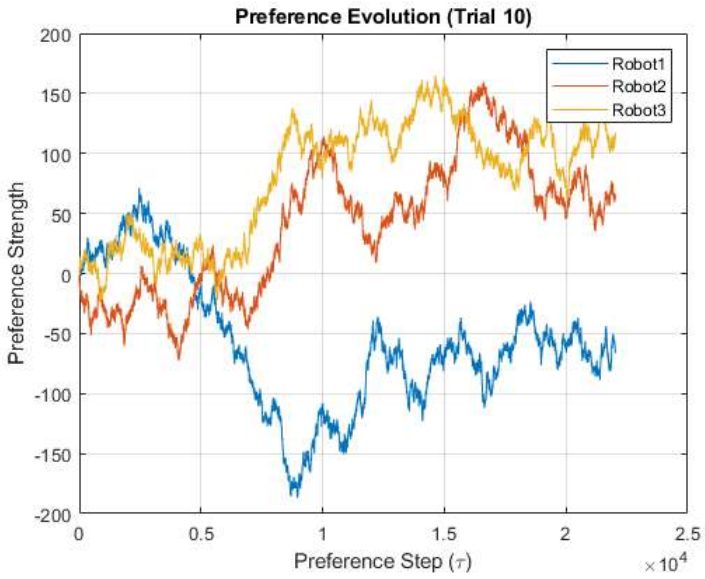

Preference Evolution (Trial 5)
Robot1
Robot2
Robot3
Preference Strength
Preference Step (τ)
$\times 10^4$


Preference Evolution (Trial 6)
Robot1
Robot2
Robot3
Preference Strength
Preference Step (τ)
$\times 10^4$


Preference Evolution (Trial 7)
Robot1
Robot2
Robot3
Preference Strength
Preference Step (τ)
$\times 10^4$


Preference Evolution (Trial 8)
Robot1
Robot2
Robot3
Preference Strength
Preference Step (τ)
$\times 10^4$





Helper Functions

```
function param = validateParam(params, name, default)
    if isfield(params, name) && isnumeric(params.(name))
        param = params.(name);
    else
        warning('Using default for %s', name);
        param = default;
    end
end

function [phi1, phi2, tau, error_sd] = getFallbackParams()
    phi1 = 0.5 + 0.1*randn();
    phi2 = 0.8 + 0.1*randn();
    tau = 10 + randi(5);
    error_sd = 0.1 + 0.05*rand();
    warning('Using randomized default parameters');
end
```

```
Estimated Parameters:
phi1: 1.4903
phi2: 0
tau: 22027.4658
error_sd: 1
Initial Preferences (from ASCs):
    1.0988   -1.2563         0
```

Published with MATLAB® R2022b