MK3 Capstone project – Cyber Secure LEGO System

User's Guide

Welcome!

This folder contains all the software achievement of our Capstone project. **MATLAB** is the only environment for launching these software scripts and the content in the folder can be classified as follows:

1. **RWTH - Mindstorms NXT Toolbox** for MATLAB;
2. A script for initializing the NXT robot;
3. A script and a Simulink object for Modelling the servomotor of NXT robot;
4. A script for measuring the sampling period of NXT brick;
5. Several self-design function for achieving specific goals in sensor attack detection and correction;
6. Widget files support USB connection between MATLAB and NXT brick;
7. **Psychtoolbox-3 package** which supports using Keyboard to Simulate the injection of sensor attack;
8. Several versions of Main function **Psychtoolbox-3 package** for controlling the car demo and simulate the sensor attack, sensor attack detection and sensor attack correction. Each version shows a milestone of this project.

The tips of demonstrating our achievement are as follows:

1. Connection:
    a. Connect the PC and NXT brick with a USB cable;
    b. Tap the orange button of NXT brick to power it on;
    c. Open MATLAB environment;
    d. Type ‘Initialization’ command to setup the connection, this command will both save the connection handle variable to MATLAB and active three servomotors objects *‘mA’*, *’mB’* and *‘mC’*. And a beep will indicate the success of connection;

```
>> Initialization
>> |
```

Remark: libusb-win32 should also be used to generate a connection drive file for connection.

2. Connect a Gamepad to PC which will be used as a control tool for demonstrating the detection-and-correction algorithm;

3. Demonstration:
    a. Open MATLAB file ‘Capstone_v13’;

```
Capstone_v13.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    MX3 CAPSTONE PROJECT MATLAB CODE   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Initialize the behaviour of Car;
theta = 10;                          % Initial outboard angular velocity, In degree/s;
theta_standby = 0;                   % LMS algoithm for extrating the redundancy steering angle;
right_standby = 0;                   % LMS for right driving motor;
left_standby = 0;                    % LMS for left driving motor;
w_ref = 0;                          % Initial Steering Angle;
Duration = 20;                       % Define how long enables the navigation;
load('Matrices_dic_driving.mat');    % Load the pre-derived state space matrices;
mat_driving = Matrices_dic_driving;  % Assign 'mat' as the reference matrices dictionary for driving motor;
load('Matrices_dic_steering.mat');   % Dictionary for steering motor;
mat_steering = Matrices_dic_steering;

%% Threshold Value
thre_steering = 10;                  % Define the detection threshold of steering subsystem;
thre_driving = 10;                   % Define the detection threshold of driving subsystem;
joy = vrjoystick(1);                 % Enable the gampad controlling method;

%% Reserve SIX factors for Steering PID controller;
% Sum of all error history for integral control;
% Record previous one error for diffrential control;
%  For Steering Motor;
error_sum_mB = 0;
error_Pre_mB = 0;
%  For Right Driving Motor;
error_sum_mA = 0;
error_Pre_mA = 0;
%  For Left Driving Motor;
error_sum_mC = 0;
error_Pre_mC = 0;

%%  Reserve a variable for Attack Signal (software vulnerability);
attack_data_mB = 0;                   % Constant attack for Steering Motor;
attack_data_mA = 0;                   % Constant attack for Right Driving Motor;
attack_data_mC = 0;                   % Constant attack for Left Driving Motor;
```

b. Click 'Run' button of MATLAB;

   Remark:

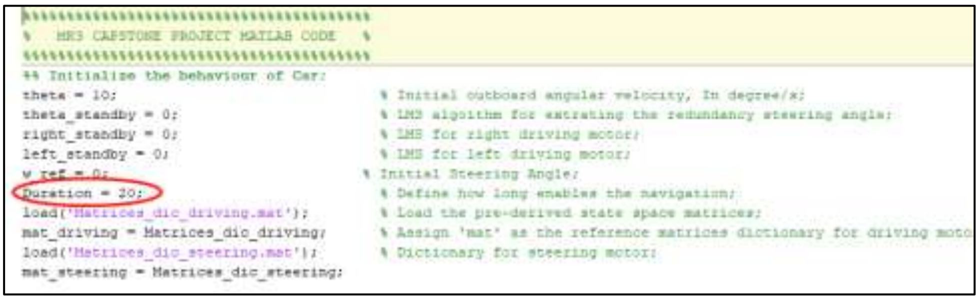
   To successfully run this script, Psychtoolbox-3 package should be correctly installed.

c. Follow the comment in this script to examine;

d. The speed and steering angle can be controlled by the joysticks of the Gamepad, the attack signal can be launched by press the numeric keyboard. The attack signal is launched for steering subsystem if no change is made;

e. By partially commenting out the script from line 238 to line 271, the detection- and -correction algorithm can be disabled for comparison;

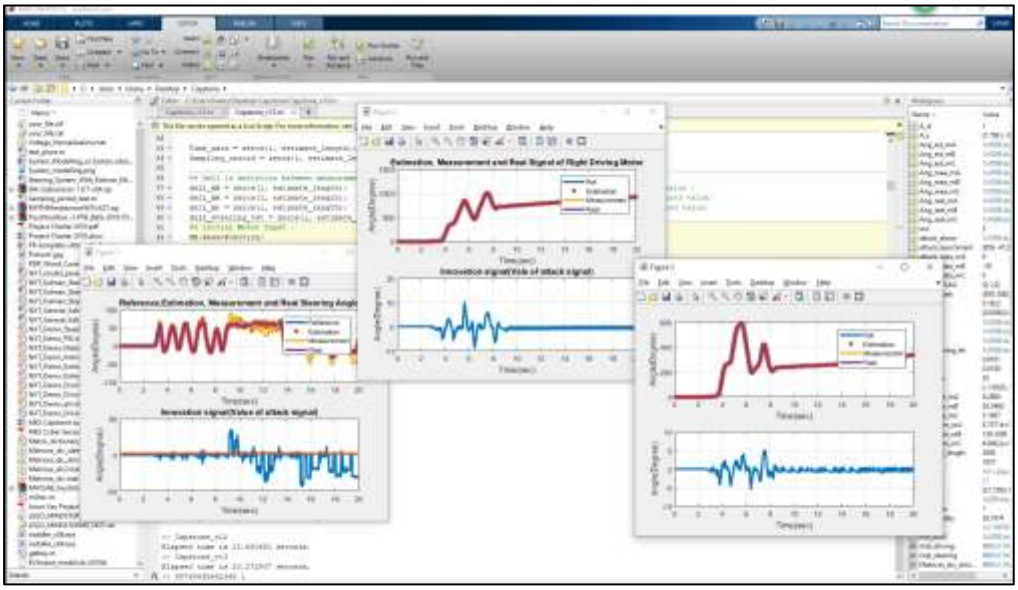
f. The total time length of the simulation would be 20 seconds if no change has been made, and the length could be changed by switching the variable 'Duration':



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   MK3 CAPSTONE PROJECT MATLAB CODE   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Initialize the behaviour of Car:
theta = 10;                          % Initial outboard angular velocity, In degree/s;
theta_standby = 0;                   % LMS algoithm for extrating the redundancy steering angle;
right_standby = 0;                   % LMS for right driving motor;
left_standby = 0;                    % LMS for left driving motor;
w_ref = 0;                          % Initial Steering Angle;
Duration = 20;                       % Define how long enables the navigation;
load('Matrices_dic_driving.mat');    % Load the pre-derived state space matrices;
mat_driving = Matrices_dic_driving;  % Assign 'mat' as the reference matrices dictionary for driving moto
load('Matrices_dic_steering.mat');   % Dictionary for steering motor;
mat_steering = Matrices_dic_steering;
```

g. Try to play with the Keyboard and Gamepad to see the influence on the car demo;

h. After simulation, several plotting will be generated to demonstrate the process of attack signal and the performance of security solution;



All the scripts including each self-design function has been commented in line, to get the overview of the project, it is recommended to read the script 'Capstone_v13'. Or, for better understand, we have setup an website and upload an earlier version of the script on http://mk3.epizy.com/

References

Psychtoolbox-3