DASA framework

Overview

Image-Based Control (IBC) systems are a class of data-intensive feedback control systems whose feedback is provided by image-based sensing using a camera. IBC has become popular with the advent of efficient image processing systems and low-cost CMOS cameras with high resolution. The combination of the camera and image processing (sensing) gives necessary information on parameters such as relative position, geometry, relative distance, depth perception and tracking of the object-of-interest. This enables the effective use of low-cost camera sensors to enable new functionality or replace expensive sensors in cost-sensitive industries like automotive.

The state-of-the-art design, analysis, and simulation of IBC assumes that the sensing algorithm is executing correctly with an assumed or estimated worst-case delay. The sensing algorithm is simulated and validated using static pre-captured image streams and is normally decoupled from the control algorithm. However, in reality, the camera is fixed to the vehicle body and any steering change would affect the region captured by the image. This dynamism cannot be captured in a static image stream and a dynamic image stream that considers the change in vehicle dynamics due to IBC actuation is needed.

We present an open-source design, analysis, and simulation framework for automotive IBC systems that can consider the change in vehicle dynamics in real-time and produces real-time dynamic image stream as per the control algorithm. Our framework models the 3D environment in 3ds Max, simulates the vehicle dynamics, camera position, environment and traffic in V-REP and computes the control output in Matlab. Our framework runs Matlab as a server and V-REP as a client in synchronous mode. We show the effectiveness of our framework using a vision-based lateral control system.

V-REP has remote API for interacting with. Here is a helpful link for integration between MATLAB and V-REP: http://www.edisondev.net/VREP/05MatlabTutorial.

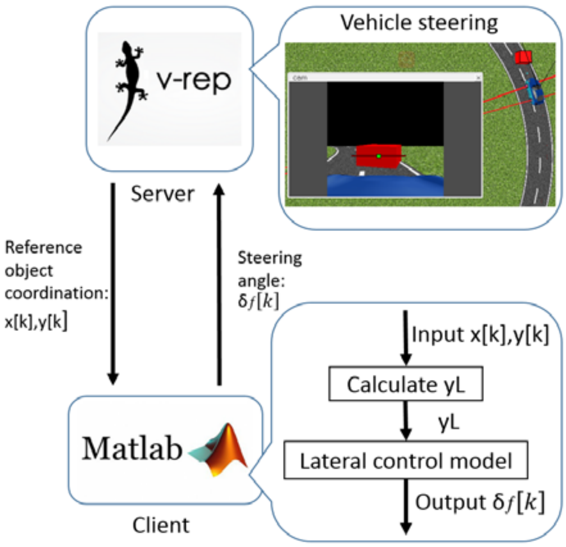

Figure 1. Architecture of the framework for validating vision based lateral control[1,2]

---

[1] Mohamed, S., Zhu, D., Goswami, D., & Basten, T. "Optimising quality-of-control for data-intensive multiprocessor image-based control systems considering workload variations." In DSD, 2018

[2] Zhu, D. "Simulation framework for design and analysis of vision-based lateral control of autonomous vehicle." Masters' Thesis, Eindhoven University of Technology, 2017.

Files: DASA framework

Folder: mexopencv-master

1. The file 'StraightTestTrack.mat' and '50mCurveTrackdata.mat' are the data for standard vehicle trajectory, they will be loaded at the initialization phase of simulation.
2. The code 'Calculation_VS_Framework.m' is to compare the difference between the framework and MATLAB based calculation.
3. The code 'LateralDeviation.m' is for look-ahead lateral deviation calculation. It will be called by framwork code during simulation.
4. The code 'control_model' is the control model configuration of switched controller. It will be called when 'ReportResult_SwitchedControl.m' is executed.
5. For varying sample period of worst-based controller case and switched-based controller case QoC result measurement (SSE calculation), in order to keep the measurement fair, the data volume has to be equal. Code 'SSE_for_Sample_SW.m' is for result measurement in such cases.

Folder: Data

This folder contains the data for all controller test. All the data are saved as .mat files. To reuse these data, you need use 'load' in MATLAB. Please check the 'load' in MATLAB help file. When you load the data for one test, copy the 'plot the results' part in framework code with appropriate legend into MATLAB 'Command Window', then the result could be plotted again.

Folder: QR-matix_Check

This folder stores MATLAB code for CQLF analysis for switched control system stability. 'yalmip' and 'SDPT3' are needed to be pre-installed.

1. The code 'control_model2' is the swtiched model setting to be done CQLF analysis. You can change model configuration.
2. The code 'cqlf_existence_clean' is for control model CQLF. Only the CQLF solution exists the overall switched control system is stable.

Folder: V-REP scene

This folder contains scenarios of two lane keeping test situations: small straight lane bias situation and entering curve situation.

Folder: 3dsMax_model

This folder contains the built curve lane model and center line model, the exported .obj files and the textures used for road and background field are stored in folder '3dsMax_model'.

1. '1to10_curve_track.max' is the curve model file.
2. 'obj file' folder contains the exported .obj files of curve lane model and curve lane center line model.
3. 'texture' folder contains the texture for road and field.
4. The straight lane could be built by using V-REP inner shape with road texture.
5. How to build the model and export the model as .obj file, you could refer to 3dsMax 2018 tutorial: http://help.autodesk.com/view/3DSMAX/2018/ENU/?guid=GUID-68FFEEDA-3801-444E-B450-942E03A7523D
6. How to import the .obj file and built the straight lane with road texture in V-REP, you could refer to V-REP user manual: http://www.coppeliarobotics.com/helpFiles/

Example: DASA framework

1. How to start simulation?

   a. open a v-rep scene in folder 'V-REP scene'
   b. open a framework matlab code (e.g., 'ReportResult_DifferParameters.m' or 'ReportResult_SwitchedControl.m') in folder 'mexopencv-master'.
   c. click 'Start' in matlab toolbar, then the whole framework simulation will be executed automatically. When the simulation finishes, the result will be plotted.