README for "Kalman filtering with synthetic measurements under an event-triggered sensor scheduler" JAVA source code

Ye Sun and Daniel B. Work

July 24, 2015

Abstract

This document describes the implementation of the Kalman filter with synthetic measurements, introduced in the article "Kalman filtering with synthetic measurements under an event-triggered sensor scheduler" by Sun and Work, submitted to the 54th IEEE Conference on Decision and Control. The source code is available to be downloaded at https://github.com/Lab-Work/KFSMcdc2015.

1 License

2 Publishing results using this software

We kindly ask any future publications using this software include a reference to the following publication:

Y. Sun and D.B. Work, "Kalman filtering with synthetic measurements under an event-triggered sensor scheduler," *submitted to the 54th IEEE Conference on Decision and Control*, 2015.

3 General Instruction

3.1 General usage notes

1. This release of software is intended to complement a paper submission to the the IEEE Transactions on Control of Network Systems;

2. This software is intended to be run on Eclipse. It was developed using Eclipse 4.3.0, and the development environment is Java SE Development Kit 8u40.

3.2 How to use the software

1. Download the folder **Sensor Schedule_CDC_kfsm** and import it into Eclipse.

2. To generate Fig. 2a in the paper, set $NumKFSM = NumKFDT = 1$ in *Estimation.java* and run *Test.java*, and plot the last columns of the generated *KFSMerror.csv* and *KFDTerror.csv* in the result folder;

3. To generate Fig. 2b in the paper, set $NumKFSM = NumKFDT = 100$ in *Estimation.java* and run *Test.java*, and plot the last columns of the generated *KFSMerror.csv* and *KFDTerror.csv* in the result folder. The trace of the error covariance is in *Trace.csv*, where the first and second columns are for the KF-SM and KF-DT, respectively.

4 Package List

1. *DoubleMatrix*, some operations on matrix and *GaussianGenerator.java* for generating Gaussian (and other types of) noise;

2. *targetGroundTruth*, where ground truth of the target to be tracked is computed;

3. *trueSolution*, where the sensor data is generated based on the true state;

4. *triggerer*, where the deterministic and stochastic threshold-based sensor schedulers are located;

5. *model*, used to specify parameters used for estimation (e.g. model error covariance matrix, initial guess of the estimation, and system dynamics used in estimation);

6. *filters*, the KF-SM, KF-DT [1, 2] and KF-ST [3, 4], the entire estimation process is in *Estimation.exportResult*, which includes all the steps in estimation and functions to export result.

References

[1] J. Wu, Q. Jia, K. H. Johansson, and L. Shi, "Event-based sensor data scheduing: trade-off between communication rate and estimation quality," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 1041–1046, 2013.

[2] K. You and L. Xie, "Kalman filtering with scheduled measurements," *IEEE Transactions on Signal Processing*, vol. 61, no. 6, pp. 1520–1530, 2013.

[3] D. Han, Y. Mo, J. Wu, S. Weerakkody, B. Sinopoli, and L. Shi, "Stochastic event-triggered sensor schedule for remote state estimation," in *Proceedings of the 52nd IEEE Conference on Decision and Control*, 2013, pp. 6079–6084.

[4] S. Weerakkody, Y. Mo, B. Sinopoli, D. Han, and L. Shi, "Multi-sensor scheduling for state estimation with event-based, stochastic triggers," in *Proceedings of the 4th IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 2013.