A Data-driven Fault Detection Method Based on Dissipative Trajectories

Qingyang Lei ∗ Muhammad Tajammal Munir ∗∗ Jie Bao ∗ Brent Young ∗∗

∗ University of New South Wales, Sydney, NSW, 2052, Australia (e-mail: j.bao@unsw.edu.au).
∗∗ University of Auckland, Auckland, 1010, New Zealand (e-mail: b.young@auckland.ac.nz).

Abstract: Fault detection is becoming increasingly important as the complexity of industrial process develops. In this paper, a data-driven fault detection method is proposed. The dissipativity theory is adopted to find the appropriate dissipativity properties for the process input output trajectory. The dissipativity properties can be viewed as an abstract energy property, and the dissipativity properties of input output trajectories represent process dynamic features. As faults occur, the dissipative trajectories will change thus allow fault detection to be performed based on these dissipativity properties. A training algorithm is developed to search for the related properties using input output data. A prior knowledge of the process can be incorporated into the algorithm to facilitate the training. The proposed fault detection method is illustrated on a case study of a mono-chlorobenzene plant simulated using VMGSim.

Keywords: Dissipativity Theory, Dissipative Trajectory, Fault Detection, Data-driven.

1. INTRODUCTION

The growing complexity of modern industrial processes present a serious challenge for real time process monitoring. Considering the plant economy, industrial processes are often operated near their design constraints, thus failure in fault detection may rapidly lead to adverse consequences. It is reported in Izadi et al. (2009), due to abnormal events, the US petrochemical industry alone suffered approximately 20 billion USD in annual losses. As may be expected, fault detection is becoming steadily more crucial.

Existing fault detection approaches can be virtually classified into two categories, model based and data based approaches. Considering it is difficult to construct an accurate industrial process model, data based approaches have gained more attention from the industry. Various multivariate statistical process monitoring methods have been developed in the past few years, e.g., principal components analysis (PCA) (Kresta et al. (1991)), dynamic principal components analysis (DPCA) (Russell et al. (2000)).

Another data-driven fault detection methods received attention is support vector machine (SVM). Support vector machine is a comparatively new machine learning algorithm developed in Vapnik and Kotz (1982). The main idea of SVM is to find an optimal separation boundary (in the form of a hyper-plane) in a way that the margin between two sample classes is maximized (Mahadevan and Shah (2009)). There are reports of successful SVM application in various cases, e.g. in Mahadevan and Shah (2009) and Widodo and Yang (2007). More details on SVM can be found in Vapnik (2013).

First developed as a framework for analysing dynamical systems, the dissipativity theory was initially proposed in Willems (1972). The dissipativity properties represent the process dynamics features, e.g. the gain, phase or their combination (Rojas et al. (2008)). As such, the change of process dynamical behavior when a fault occurs may be reflected by change in the dissipativity properties. It has been shown that dissipativity properties derived from process models can be used for fault detection (Lei and Bao (2015)). In this paper, a data-driven fault detection approach is developed based on the behavior of processes captured by the dissipativity of the process input/output trajectories, which are determined through a training procedure using process data.

Dissipativity of a dynamic process often captures limited process features (e.g., the information included in the traditional QSR dissipativity is limited). As such, QSR-dissipativity can be too coarse for fault detection applications. It is shown in Chen et al. (2010), the passivity condition (a special case of dissipativity condition) was applied for fault detection and the results can be very conservative. Moreover, in traditional dissipativity, the storage function is defined on state variables. However state measurements are often unavailable, leading to the need of a demanding task of state estimation. In this paper, storage functions and supply rates are in Quadratic Difference Forms (QdF) (Willems and Trentelman (1998)) so that they are function of process input/output history. QdF based dissipativity may capture far more detailed process dynamic features compared to QSR dissipativity (Tippett and Bao (2011)) and thus allows much better fault detection performance.

Similar to SVM, the proposed method requires a training procedure. In the off-line design stage, a algorithm searches
for the dissipativity properties of the process input output dissipative trajectories. The training is performed based on process input and output data. The on-line implementation involves only the evaluation of the dissipativity inequality which is computational tractable. A comparison study between SVM and proposed method is conducted, and the proposed method outperformed SVM.

This paper is organised as follows. A brief introduction on dissipivity system theory and dissipative trajectory is given in Section 2. The techniques of proposed data-driven fault detection method are presented in Section 3, an illustrative example and comparison study is included in Section 3 as well. The proposed fault detection method is demonstrated using a case study on a mono-chlorobenzene plant simulated using VMGSim in Section 4, followed by the conclusion.

2. SYSTEM BEHAVIOUR AND DISSIPATIVE TRAJECTORY

First developed as a framework for analysing dynamical systems, the dissipativity theory was initially proposed in Willems (1972). In this paper, the proposed data-driven fault detection approach is developed based on dissipativity inequality.

A general process can be described by a state space equation as follows

\[ x(t + 1) = Ax(t) + Bu(t) \]
\[ y(t) = Cx(t) + Du(t), \]

where \( x \in X \subset \mathbb{R}^n \) is state variable vector, \( u \in U \subset \mathbb{R}^p \) is the input variable vector, and \( y \in Y \subset \mathbb{R}^q \) is the output variable vector.

If a function defined on process input output variables denoted as the supply rate \( s(u(t), y(t)) \), and a function defined on process states, denoted as the storage function \( V(x(t)) \) can be found, such that

\[ s(u(t), y(t)) \geq V(x(t + 1)) - V(x(t)) \]

for all time steps (Willems (1972)), then the process in (1) is said to be dissipative with respect to that supply rate and storage function.

A commonly used supply rate (QSR dissipativity) takes the following form

\[ s(u(t), y(t)) = y^\top(t)Qy(t) + 2y^\top(t)Su(t) + u^\top(t)Ru(t). \]

However, the dissipativity properties of a system is generally difficult to determine from process input/output data as the dissipativity inequality (2) has to be satisfied for all input output trajectories (Tippett et al. (2013)). In this paper, a fault detection approach is developed based on the change of process behaviors which is manifested by input output trajectories, i.e., the dissipativity of the trajectories. The dissipative trajectory is defined as below.

**Definition 1 Dissipative Trajectory.** (Tippett and Bao (2013)) An input/output trajectory of a process is said to be dissipative with respect to supply rate \( Q_\phi \), if, at all time instances \( k \), the following inequality is satisfied:

\[ W_k = \sum_{t=0}^{k} Q_\phi(u(t), y(t)) \geq 0, \]

where \( Q_\phi \) is the supply rate induced by

\[ \phi_\phi(\zeta, \eta) = \begin{pmatrix} Q_\phi(\zeta, \eta) & S_\phi(\zeta, \eta) \\ S_\phi^\top(\zeta, \eta) & R_\phi(\zeta, \eta) \end{pmatrix}. \]

The concept of a dissipative trajectory can be understood as a weaker condition of classical dissipativity, see Tippett and Bao (2013). The definition of dissipative trajectory does not explicitly require the dissipativity inequality to be satisfied for all possible input. The concept from behavioural approach (Willems (2007)) also motivates the idea of finding the dissipativity properties for the input output trajectory. The adoption of dissipative trajectory concept is justified as the process operating data are by and large a fraction of all possible trajectories. It is generally not feasible to acquire input output data for all possible input trajectories. However, using proposed method the dissipativity properties for certain dissipative input output trajectories can be found.

Note that in this paper, both the supply rate and storage function are in quadratic difference form (QdF). The QdF supply rate \( Q_\phi(\hat{u}(t), \hat{y}(t)) \) for discrete time system is initially proposed in Kaneko and Fujii (2000), and takes a following form

\[ Q_\phi(\hat{u}(t), \hat{y}(t)) = \begin{pmatrix} \hat{y}(t) \end{pmatrix}^\top \begin{pmatrix} \tilde{Q} & \tilde{S} \\ \tilde{S}^\top & \tilde{R} \end{pmatrix} \begin{pmatrix} \hat{y}(t) \\ \hat{u}(t) \end{pmatrix} \]

with

\[ \hat{u}(t) = (u^\top(t) \ u^\top(t + 1) \ldots u^\top(t + N))^\top \]
\[ \hat{y}(t) = (y^\top(t) \ y^\top(t + 1) \ldots y^\top(t + N))^\top \]

The QdF supply rate \( Q_\phi \) can be understood as an extension to the traditional QSR-dissipativity. As the QdF can be rewritten into a following form

\[ Q_\phi(\hat{u}(t), \hat{y}(t)) = \hat{y}^\top(t) \tilde{Q} \hat{y}(t) + 2\hat{y}^\top(t) \tilde{S} \hat{u}(t) + \hat{u}^\top(t) \tilde{R} \hat{u}(t). \]

In this work, storage functions are defined as QdFs of process input and output trajectory (denoted as \( Q_\phi(\hat{u}(t), \hat{y}(t)) \)). As such, the proposed method does not need state measurement or estimation, leading to less computational complexity.

The dissipativity inequality in (2) can be easily written into a QdF form, based on extended input and output variables, as follows

\[ Q_\phi(\hat{u}(t), \hat{y}(t)) \geq Q_\phi(\hat{u}(t + 1), \hat{y}(t + 1)) - Q_\phi(\hat{u}(t), \hat{y}(t)). \]

A training algorithm is developed in the next section to search for the appropriate dissipativity properties \( (Q_\phi, Q_\phi) \) based on process input output data.

3. DISSIPATIVITY TRAJECTORIES BASED FAULT DETECTION METHOD

The basic idea of proposed method is to search for the dissipativity properties based on input and output data. The process input output data trace certain dissipative trajectories as the process is operated under nominal condition. When faults occur the trajectory is no longer dissipative, thus allow changes to be detected by checking the dissipativity properties.
The decision making is based on the following dissipativity inequality. If the inequality holds, the data are tracing a dissipative trajectory, and the process is assumed to be operating normally. However, if the dissipativity condition is breached, the data are no longer tracing the dissipative trajectory and one can conclude there may be a fault in the process.

\[ Q(\hat{u}(t), \hat{y}(t)) \geq Q(\hat{u}(t+1), \hat{y}(t+1)) - Q(\hat{u}(t), \hat{y}(t)) \]  

(10)

3.1 Off-line Training

A recursive algorithm is developed to search for the appropriate dissipativity properties using process input output data.

First, \( m \) training samples are collected into the extend input and output form. For example, set \( N = 2 \) in equation (7), this indicates the QdF order is 2. The order of QdF in this proposed method is chosen on a trial-and-error basis. One can increase the QdF order if low order can not deliver satisfactory performance. As more data been included into QdF, performance would be improved at the expense of computational efforts. Class label \( y_i \) is also assigned to each sample when collecting the data, samples from normal operating condition are labelled as \( y_i = 1 \) while samples from faulty condition are labelled as \( y_i = -1 \).

Then the search starts with a initial guess of the properties (coefficients for \( Q_{\Phi}, Q_{\Psi} \)), where prior knowledge on the process can be considered. If no prior knowledge is available, the initial guess can be a all-one matrix of proper dimension. The coefficient matrix for \( Q_{\Phi} \) and \( Q_{\Psi} \), denoted as \( \phi \) and \( \psi \) respectively, are in the following form

\[
\phi = \begin{pmatrix} \bar{Q} & \bar{S} \\ \bar{S}^T & \bar{R} \end{pmatrix}, \quad \psi = \begin{pmatrix} \bar{X} & \bar{Y} & \bar{Z} \end{pmatrix}
\]  

(11)

The decision variables for the training algorithm are the matrices coefficients \( \bar{Q}, \bar{S}, \bar{R}, \bar{X}, \bar{Y}, \bar{Z} \). The dimension of these decision variables is a function of the QdF order \( N \). The training algorithm updates the dissipativity properties in each iteration through minimization of a objective function \( \theta \). The objective function is used to evaluate if the input output data are tracing a dissipative trajectory. The objective function, denoted as \( \theta \), is simply the summation of all the scores \( \alpha_i \), as follows

\[
\theta = \sum_{i=1}^{m} \alpha_i
\]  

(12)

To calculate the objective function, one needs to check the dissipativity inequality for each training sample at each iteration (with temporary \( \phi \) and \( \psi \)). If the inequality (9) holds for data from nominal operating condition, assign a negative score \( \alpha_i = -a \) to that sample. If the inequality holds for data from faulty operating condition (which leads to a missed alarm), assign a positive score \( \alpha_i = b \). Similarly, assign a negative score if the inequality does not hold for data from faulty condition (successful detection), positive score if the inequality does not hold for data from normal condition (false alarm). The scoring rules for each sample can be summarised as

\[
\alpha_i = \begin{cases} 
-\text{a} & \text{if } y_i = 1 \text{ and } Q_{\Phi,i} - \Delta Q_{\Psi,i} > 0 \\
\text{b} & \text{if } y_i = 1 \text{ and } Q_{\Phi,i} - \Delta Q_{\Psi,i} < 0 \\
-\text{a} & \text{if } y_i = -1 \text{ and } Q_{\Phi,i} - \Delta Q_{\Psi,i} < 0 \\
\text{b} & \text{if } y_i = -1 \text{ and } Q_{\Phi,i} - \Delta Q_{\Psi,i} > 0 
\end{cases}
\]  

(13)

In each iteration, the algorithm updates the decision variables. And the training can be organized as a minimization problem with a form given below

\[
\text{minimize } \theta = \sum_{i=1}^{m} \alpha_i \\
\text{subject to } \alpha_i y_i (Q_{\Phi,i} - \Delta Q_{\Psi,i}) < 0, \quad i = 1, 2, \ldots m
\]  

(15)

In each iteration, the scores \( \alpha_i \) are calculated using the temporary dissipativity coefficients, and \( \theta \) for that iteration is the summation of all \( \alpha_i \) in that iteration. Once the stopping criterion is satisfied (objective function \( \theta \) hits certain value), the training algorithm is stopped and the dissipativity properties \( (Q_{\Phi}, Q_{\Psi}) \) for a dissipative input output trajectory is found to be the properties in the last iteration.

3.2 On-line Implementation

At every sampling step \( t \), evaluate the dissipativity inequality

\[
Q(\hat{u}(t), \hat{y}(t)) - (Q(\hat{u}(t+1), \hat{y}(t+1)) - Q(\hat{u}(t), \hat{y}(t))) \geq 0
\]  

(16)

If (16) holds, then these process input output data are tracing a dissipative trajectory that determined in the off-line training stage, and no fault is detected. Otherwise the trajectory is not dissipative and a fault has been detected.

The on-line implementation of the proposed approach only involves the evaluation of a quadratic form of the processes inputs and outputs over a moving window. Thus the computational burden is reasonably tractable.

3.3 Illustrative Example

A heat exchanger as shown in Fig. 1 is studied. In this process, steam mass flow \( m_s \) is measured as input variable, and the fluid outlet temperature \( v_o \) is measured as

Fig. 1. A Heat Exchanger Example (Isermann (2011))
output variable, sampling period is 5 seconds. There is a frequently found fault for heat exchanger, i.e. air (inert gas) in steam space. Input output samples are obtained from simulation studies.

The QdF order used for this example is 2, input output data are collected as in (7) with \( N = 2 \). To choose \( N = 2 \) is because with 2 extra step data, the proposed method is capable of delivering satisfactory detection results. Generally speaking, as more data been included into QdF, performance would be improved at the expense of computational efforts. While having large \( N \) benefits the performance, time needed for collecting the data would also increase, thus having \( N = 2 \) gives the balance between performance and efficiency. The training fault detection result can be seen in Fig. 2. The vertical axis represents the value for \( Q_{\Phi} - \Delta Q_{\Phi} \). According to (9), if \( Q_{\Phi} - \Delta Q_{\Phi} \) is positive, the inequality holds and the process input output data are tracing a dissipative trajectory. When fault occurs (after the vertical dashed line), the data are no longer tracing the dissipative trajectory and \( Q_{\Phi} - \Delta Q_{\Phi} \) is negative. As can be seen from Fig. 2, the dissipativity properties are able to detect the change in operating condition.

### 3.4 Support Vector Machine Comparison Study

The proposed method requires a training procedure with a complexity similar to that of SVM. There are also many reported successful applications of SVM in fault detection. Thus a comparison study is done in this subsection. The basic concept of SVM is to find a separation boundary, such that the margin between two sample classes is maximized. For a typical two-class problem (i.e. classify normal and faulty condition), after training, the support vector machine can find an optimal hyper-plane that separates these two classes, see Abe (2005).

As shown in Figure 3, the optimal plane is found such that the margin between two classes is maximized. The circles/squares with grey colour that are used in constructing the separation boundaries are called support vectors (SV). The SVs contain all information for constructing the separation plane.

Thus a comparison study between SVM and proposed method is done. The training procedure of a SVM is performed using the same samples as previous subsection (input output data as in (7) with \( N = 2 \)). And the SVM detection accuracy is 10% less than the proposed approach.

### Table 1. Illustrative Example Result Summary

| Method   | Training Accuracy | Testing Accuracy |
|----------|-------------------|------------------|
| SVM      | 91%               | 88%              |
| Dissipativity | 100%             | 98%              |

Results of SVM using input output at each step is shown in Fig. 4. The normal operating samples are marked as squares, whereas samples from faulty condition are marked as circles. The misclassified samples are marked by a cross in the centre. As shown in Figure 4, there are significant overlapping between the two classes, therefore it is difficult to acquire satisfactory result. This somehow explains why SVM is not performing well in this particular example.

The fault detection performance of the dissipativity based SVM based methods are summarised in Table 1. The dissipativity based method outperformed the SVM based method in terms of accuracy.

### 4. CASE STUDY ON MONO-CHLOROBENZENE PLANT

A monochlorobenzene (MCB) plant is studied in this section. The plant is simulated using VMGSim, the process flowsheet is depicted in Fig. 5.

A mixture of benzene, monochlorobenzene (MCB), and HCl is present in the feed of this MCB separation process. The vapour stream coming out of the flash tank is fed into the absorber (T1) where it is put into contact with recycled MCB. Most of the HCl product comes out of the absorber as vapour. The liquid product (L2) coming out of the absorber is mixed with liquid product (L1) from the flash vessel (F1) in a mixer (M1). The mixture coming out of the mixer (M1) is then fed into the distillation column (T2). In this column a fixed amount (1% of inlet feed to the
Fig. 5. Mono-chlorobenzene plant under study

The distillate product (D) contains most of the benzene and bottom product (B) contains most of the MCB. A fraction of the bottom product stream is recycled back into the absorber.

The feed flow rate is in constant oscillation, and the primary fault in this process takes place in the recycle valve (V-7). A change of flow rate through the valve is used to simulate a block in the valve. The initial fault in recycle valve changed the flow rate to around 90% of the set point. Due to the presence of other controllers, the system can recover from the change. Then the next fault in recycle valve changed the flow rate to around 40% of the nominal value. The system can not to recover from this change even with the help of other controllers. All other controllers was working fine except the recycle valve, which was too small to handle the required flow.

We took 100 samples (50 nominal operating samples and 50 faulty operating samples) out of all the 2000 samples to do the training for both SVM and proposed method. The first fault occurred at around the 1000th samples (after the vertical dashed line in Fig. 6 and 7), while the second fault occurred at around the 1600th samples. Each sample contains the flow rate and composition from the input and output streams of the process ('Feed', 'HCl', 'Benzene' and 'MCB' in Fig. 5). It is assumed we do not have access to the measurements of the recycle stream where the fault had occurred.

The SVM testing result is shown in Fig. 6, the dissipativity method result is shown in Fig. 7. The Y axis in Fig. 6 and 7 represents the class label, where +1 indicates nominal condition and -1 indicates faulty condition. A summary of these two methods is presented in Table 2. It is observed that the proposed method outperformed the SVM based method, as the proposed method can detect a wider range of faulty operating condition. It is observed that both methods did not perform well around the first 100 samples after the fault. This can be explained as, after the first stage of fault, the control system is able to rectify the changes in the process, for example, in Fig. 8 the benzene flow out of the system was temporarily restored to its nominal value. It also worth noting that the proposed method has better detection performance after the second fault, as in Fig. 6 one can observe a long series of missed alarms after the second fault around the 1600th samples

Table 2. Case Study Result Summary

| Method     | Training Accuracy | Testing Accuracy |
|------------|-------------------|------------------|
| SVM        | 100%              | 87.2%            |
| Dissipativity | 100%              | 93.6%            |

Fig. 6. SVM Result

5. DISCUSSION AND CONCLUSION

In this paper, an approach to train the dissipativity properties using process date is developed. The idea of finding the dissipativity properties for process input output trajectories leads to simplified fault detection approach. The efficacy of proposed method is demonstrated in a MCB plant case study, where the dissipativity motivated data-driven method outperformed SVM.

The primary advantage of proposed approach is its simplicity, particularly in terms of on-line implementation.
In this paper, once the dissipativity properties are found off-line, on-line evaluation of the dissipativity inequality can be readily carried out through simple calculations. The dissipativity properties also provide insights into the dynamic features of the processes, e.g. the gain, phase or their combination. Whilst SVM does not provide details of the physical features of that system.

One possible extension to proposed method is data-driven fault diagnosis. The off-line training stage requires samples from both nominal and faulty operating condition, with modest alteration the proposed algorithm can be used to find dissipativity properties that are sensitivity to individual faults. Thus allow fault diagnosis to be performed based on process data.

Another possible future extension is to develop an integrated fault detection and fault-tolerant control design structure. As reported in Bao et al. (2003), the dissipativity properties are used in fault-tolerant control design and showed promising results. There is also a general consensus in the process control community that fault detection and fault tolerant control should be integrated to offer the most efficient approach as stated in Mhaskar et al. (2012).

REFERENCES

Abe, S. (2005). Support vector machines for pattern classification, volume 2. Springer.

Bao, J., Zhang, W.Z., and Lee, P. (2003). Decentralized fault-tolerant control system design for unstable processes. Chemical Engineering Science, 58(22), 5045–5054.

Chen, W., Ding, S., Khan, A., and Abid, M. (2010). Energy based fault detection for dissipative systems. In Control and Fault-Tolerant Systems (SysToI), 2010 Conference on, 517–521. IEEE.

Isermann, R. (2011). Fault-diagnosis applications: model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems. Springer.

Izadi, I., Shah, S.L., Shook, D.S., and Chen, T. (2009). An introduction to alarm analysis and design. In 7th IFAC International Symposium on Fault Detection, Supervision and Safety of Technical Systems, SAFEPROCESS’09, June 30, 2009 - July 3, 2009.

Kaneko, O. and Fujii, T. (2000). Discrete-time average positivity and spectral factorization in a behavioral framework. Systems & control letters, 39(1), 31–44.

Kresta, J.V., MacGregor, J.F., and Marlin, T.E. (1991). Multivariate statistical monitoring of process operating performance. The Canadian Journal of Chemical Engineering, 69(1), 35–47.

Lei, Q. and Bao, J. (2015). Dissipativity based fault detection and diagnosis for linear systems. In 2015 IEEE Conference on Control Applications (CCA), Sydney, Australia.

Mahadevan, S. and Shah, S.L. (2009). Fault detection and diagnosis in process data using one-class support vector machines. Journal of Process Control, 19(10), 1627–1639.

Mhaskar, P., Liu, J., and Christofides, P.D. (2012). Fault-tolerant process control: methods and applications. Springer Science & Business Media.

Rojas, O.J., Bao, J., and Lee, P.L. (2008). On dissipativity, passivity and dynamic operability of nonlinear processes. Journal of Process Control, 18(5), 515–526.

Russell, E.L., Chiang, L.H., and Braatz, R.D. (2000). Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. Chemometrics and Intelligent Laboratory Systems, 51(1), 81–93.

Tippett, M.J. and Bao, J. (2011). Dissipativity based analysis using dynamic supply rates. In Proceedings of the 18th IFAC World Congress, volume 28.

Tippett, M.J. and Bao, J. (2013). Distributed model predictive control based on dissipativity. AIChE Journal, 59(3), 787–804. doi:10.1002/aic.13868.

Tippett, M.J., Hoe, D., and Bao, J. (2013). Integrated approach to identification and control of multivariable processes based on dissipativity. In Thermodynamic Foundations of Mathematical Systems Theory, volume 1, 54–59.

Vapnik, V. (2013). The nature of statistical learning theory. Springer Science & Business Media.

Vapnik, V.N. and Kotz, S. (1982). Estimation of dependences based on empirical data, volume 40. Springer-verlag New York.

Widodo, A. and Yang, B.S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. Mechanical Systems and Signal Processing, 21(6), 2560–2574.

Willems, J.C. and Trentelman, H.L. (1998). On quadratic differential forms. SIAM Journal on Control and Optimization, 36(5), 1703–1749.

Willems, J.C. (1972). Dissipative dynamical systems part i: General theory. Archive for rational mechanics and analysis, 45(5), 321–351.

Willems, J.C. (2007). The behavioral approach to open and interconnected systems. Control Systems, IEEE, 27(6), 46–99.