

Graph-theoretic Analysis of Complex Energy Integrated Networks^{*}

Sujit S. Jogwar^{*} Srinivas Rangarajan^{**}
Prodromos Daoutidis^{**}

^{*} *Praxair Technology Center, Tonawanda, NY 14150 USA (e-mail: sujit_jogwar@praxair.com).*

^{**} *University of Minnesota, Minneapolis, MN 55455 USA (e-mail: daout001@umn.edu)*

Abstract: Complex process networks featuring multiple energy integration agents (process-to-process heat exchangers) offer significant cost benefits while adding additional operational constraints. These networks show potential for multi-time scale dynamics owing to the presence of energy flows spanning multiple orders of magnitude. In previous work, we have developed a graph-theoretic framework to systematically uncover this time scale multiplicity. In this paper, we present an application of this framework to a reactor-heat exchanger system used for naphtha reforming. This system involves energy flows spanning three orders of magnitude and the underlying energy balance variables evolve over two time scales. The framework allows for the derivation of control-relevant models in each time scale and classifies the control objectives leading to a hierarchical control strategy. We demonstrate that the analysis uses minimum process information, is efficient, and scalable to large networks.

Keywords: Graph theory, Multi-time scale dynamics, naphtha reforming, hierarchical control.

1. INTRODUCTION

Increased global competition, high energy cost and the efforts towards sustainability have intensified the development of energy integrated process networks. While such integration results in cost savings, the resulting systems are quite difficult to operate and control. Specifically, such tightly integrated networks exhibit complex dynamics such as openloop instability, inverse response, multi-time scale dynamics, and in some cases, chaotic behavior (Chen and Yu, 2003; Morud and Skogestad, 1998; Zhu and Liu, 2005; Jogwar et al., 2009). Additionally, integration is associated with a reduction in operating degrees of freedom, resulting in loss of controllability. This has motivated the development of network-level control strategies (Rawlings and Stewart, 2008; Liu et al., 2008; Hudson and Bao, 2012; Hioe et al., 2012; Baldea et al., 2013).

In previous work (Jogwar et al., 2010), we have developed an analysis and control framework for typical energy integrated networks. The analysis of various energy integrated systems revealed the presence of two fundamental configurations that are at the core of these networks; those with *large energy recycle* and those with *large energy throughput*. These prototype networks have interesting time scale properties: large recycles exhibit two-time scale energy dynamics whereas large throughputs result in energy dynamics evolving at a faster rate compared to material dynamics (Jogwar et al., 2010). For such networks, singular perturbations can be used to reduce the underlying two-time scale dynamic model and the resulting reduced

models can be used for the design of a hierarchical control strategy in different time scales. Yet such simple prototype network configurations are rarely found in isolation. They are typically part of more complex networks with multiple integration loops. In the case of such complex networks, the rigorous analysis using singular perturbations, is in principle possible, but can be cumbersome. To this end, we have developed a graph-theory based analysis framework for complex networks which mimics the approach based on singular perturbations and provides a scalable reduction for large networks (Jogwar et al., 2011; Heo et al., 2012).

In this paper, we first review the key features of this graph-theoretic framework. In section 3, we consider an example complex network and apply this framework to establish the time scale properties and develop a hierarchical control scheme. Lastly, the advantages of this framework are illustrated and future research directions are highlighted.

2. GRAPH-THEORETIC ANALYSIS FRAMEWORK

In order to apply the graph-theoretic framework to analyze a complex network, it needs to be represented by an equivalent *energy flow graph*. This is defined as a weighted graph $\mathcal{G}(N,E)$ of the energy flows in a process network – the nodes N represent individual process entities (e.g. heat exchanger passes, trays of a distillation column) and the edges E (directed and weighted) represent the direction and the order of magnitude of the energy flows. Sources and sinks are not explicitly represented. Directed edges with no tail (or head) node are energy flows from sources (or to sinks) (Ahuja et al., 1993).

^{*} Partial financial support for this work by National Science Foundation (through Grant CBET-1133167) is gratefully acknowledged.

Once the network is represented by its energy flow graph, we have developed an algorithm that systematically reduces this graph to reveal the underlying time scale multiplicity, derive the scaled dynamic equations for each time scale and classify control objectives and manipulated inputs for a hierarchical control strategy. For simplicity of illustration, we have divided the algorithm in three parts.

Algorithm 1: *MultiTimeScale*(\mathcal{G}, W)

```

1: Sort W in descending order
2: for  $i = 1$  to  $\text{Size}(W)$  do
3:    $m = W[i]$ 
4:    $\mathcal{H} = \text{InducedSubgraph}(\mathcal{G}, m)$ 
5:    $\mathcal{T}(\tau_m) = \text{nodes} \in \mathcal{H}$ 
6:    $C = \text{SmallestElementaryCycle}(\mathcal{H})$ 
7:   while  $C \neq \phi$  do
8:      $\text{GraphReduce}(\mathcal{G}, C, m)$ 
9:      $\text{GraphReduce}(\mathcal{H}, C, m)$ 
10:     $C = \text{SmallestElementaryCycle}(\mathcal{H})$ 
11:   end while
12:   for all node  $N \in \mathcal{H}$  such that  $\text{degree}(N) \neq 0$  do
13:     Remove  $N$  from  $\mathcal{G}$ 
14:   end for
15: end for
16: return  $\mathcal{T}$ 

```

The inputs to the algorithm are the graph $\mathcal{G}(N, E)$ and a vector W of the various orders of magnitude exhibited by different energy flows. The output of the algorithm is \mathcal{T} , a set such that $\mathcal{T}(\tau_m)$ refers to the set of units (nodes) evolving in the time scale τ_m . Since one seeks the evolution of the system for times $t = 0 \rightarrow \infty$, the algorithm begins with the largest order of magnitude energy flows (corresponding to the fastest time scale) and proceeds to the smallest. For a given order of magnitude ‘ m ’, an induced subgraph ‘ \mathcal{H} ’ is formed from \mathcal{G} such that all the edges in \mathcal{H} are of the order ‘ m ’. All the nodes in this subgraph evolve at the time scale τ_m . A graph reduction procedure (steps 7 through 14) is applied to this subgraph before proceeding to the next time scale.

Algorithm 2: *Dynamics*(\mathcal{G}, W)

```

1: Sort W in descending order
2: for  $i = 1$  to  $\text{Size}(W)$  do
3:    $m = W[i]$ 
4:    $\mathbf{g}_m(\mathbf{u}_m) = h_{1,s} \times \text{Ebalance}(\mathcal{G}, m)$ 
5:    $\mathcal{H} = \text{InducedSubgraph}(\mathcal{G}, m)$ 
6:    $C = \text{SmallestElementaryCycle}(\mathcal{H})$ 
7:   while  $C \neq \phi$  do
8:      $\text{GraphReduce}(\mathcal{G}, C, m)$ 
9:      $\text{GraphReduce}(\mathcal{H}, C, m)$ 
10:     $C = \text{SmallestElementaryCycle}(\mathcal{H})$ 
11:   end while
12:   if  $\text{size}(\text{RecycleTimes}) \neq 0$  then
13:      $DAE_m = \mathbf{C}_m \mathbf{B}_m \mathbf{z}_m$ 
14:      $\text{Constraint}_m = \bar{\mathbf{g}}_{m-}(\mathbf{u}_{m-})$ 
15:     if  $\text{size}(\text{RecycleTimes}) > 1$  then
16:        $\text{AddConstraints}_m = \sum_{j=1}^{\text{size}(\text{RecycleTimes})} DAE_j$ 

```

```

17:   end if
18:   end if
19:   if  $\text{degree}(N) = 0$  for any node  $N \in \mathcal{H}$  then
20:     Add  $\tau_m$  to  $\text{RecycleTimes}$ 
21:     Add  $N_i$  to  $\text{PureRecycles}$ 
22:   end if
23:   if  $\text{degree}(N) \neq 0$  for all nodes  $N \in \mathcal{H}$  then
24:     Clear  $\text{RecycleTimes}, \text{PureRecycles}$ 
25:   end if
26:   for all node  $N \in \mathcal{H}$  such that  $\text{degree}(N) \neq 0$  do
27:     if  $N$  is a composite node then
28:       Remove  $N_i$  from  $\text{PureRecycles}$ 
29:     end if
30:     Remove  $N$  from  $\mathcal{G}$ 
31:   end for
32: end for
33: Energy balance equations are
34:

```

$$\frac{dH}{dt} = \sum_{i=W[1]}^{\text{size}(W)} \frac{1}{\varepsilon^i} \mathbf{g}_i(\mathbf{u}_i)$$

```

35: for all  $m \in W$  do
36:   Reduced order model in  $\tau_m$  is
37:

```

$$\begin{aligned} \frac{dH_m}{d\tau_m} &= \bar{\mathbf{g}}_m(\mathbf{u}_m) + DAE_m \\ 0 &= \text{Constraint}_m + \text{AddConstraints}_m \end{aligned}$$

```

38: end for

```

Having identified the various time scales exhibited by the energy dynamics of a complex network, this algorithm derives the scaled equations describing the energy dynamics in each time scale. Graph traversing algorithms (Cormen, 2001) are used for this and the subroutine *Ebalance* performs this operation. Depending on the structure of the subgraph \mathcal{H} , *i.e.* whether it corresponds to a recycle or a throughput, the resulting scaled dynamic equations are ODEs (ordinary differential equations) or DAEs (differential algebraic equations). Line 34 gives the overall energy dynamics whereas line 37 gives the reduced order dynamics in a particular time scale.

Algorithm 3: *Control*(\mathcal{G}, W)

```

1: Sort W in descending order
2: for  $i = 1$  to  $\text{Size}(W)$  do
3:    $m = W[i]$ 
4:    $\mathcal{H} = \text{InducedSubgraph}(\mathcal{G}, m)$ 
5:   for each node  $N \in \mathcal{H}$  do
6:     if  $N$  is a composite node then
7:       add  $\sum N_i$  to  $\mathcal{Y}(\tau_m)$ 
8:     else
9:       add  $N$  to  $\mathcal{Y}(\tau_m)$ 
10:    end if
11:   end for
12:    $\mathcal{U}(\tau_m) = \text{Edges in } \mathcal{H}$ 
13:    $C = \text{SmallestElementaryCycle}(\mathcal{H})$ 
14:   while  $C \neq \phi$  do
15:      $\text{GraphReduce}(\mathcal{G}, C, m)$ 

```

```

16:   GraphReduce ( $\mathcal{H}, C, m$ )
17:    $C = \text{SmallestElementaryCycle}(\mathcal{H})$ 
18: end while
19: if degree( $N$ ) = 0 for any node  $N \in \mathcal{H}$  then
20:   All but 1 out of  $N_i$  should be controlled in this
   time scale
21: end if
22: for all node  $N \in \mathcal{H}$  such that degree( $N$ )  $\neq 0$  do
23:   Remove  $N$  from  $\mathcal{G}$ 
24: end for
25: end for
26: return  $\mathcal{Y}, \mathcal{U}$ 

```

Lastly, algorithm 3 classifies different control objectives to be addressed in each time scale (\mathcal{Y}) as well as the potential manipulated inputs (\mathcal{U}) available to achieve those objectives. This is based on the fact that flowrates of different magnitude have been documented to act (and thus be available for manipulation) at different time scales (Jogwar et al., 2009, 2010).

Let us now apply this framework to an example complex network.

3. ANALYSIS OF REACTOR-HEAT EXCHANGER SYSTEM

Energy integrated reactor-heat exchanger systems are quite common in chemical industries and typically incorporate recycle of energy using numerous process-to-process heat exchangers. The simplest case involves using a hot reactor effluent to preheat cold feed, resulting in a configuration with large energy recycle (Jogwar et al., 2009). The design of heat exchanger networks to minimize utility consumption in reactor systems leads to complex configurations. Fig. 1 shows one such example system used for naphtha reforming (Varghese and Bandyopadhyay, 2007). This system has two endothermic reactors (reformers) operating at elevated temperature compared to the feed. Nine process-to-process heat exchangers (PPX_1 through PPX_9) are used to reduce external energy consumption by transferring energy from a hot effluent stream to the cold inlet stream, and thus each one of these acts as a feed effluent heat exchanger (FEHE). The energy flow structure for this system involves nine energy recycle loops. Table 1 tabulates the nominal values of the duties of these FEHEs, along with the heating/cooling duties of the utilities.

Table 1. Nominal values of heat exchanger duties for the reactor - HE network

Exchanger	Duty	Exchanger	Duty
H_1	3.72 kW	PPX_1	4.88 kW
H_2	12.55 kW	PPX_2	17.20 kW
H_3	37.77 kW	PPX_3	1.88 kW
C_1	3.80 kW	PPX_4	8.50 kW
C_2	4.77 kW	PPX_5	10.80 kW
C_3	6.20 kW	PPX_6	29.30 kW
C_4	0.35 kW	PPX_7	4.10 kW
C_5	12.50 kW	PPX_8	3.84 kW
		PPX_9	7.90 kW

The duty of any FEHE represents the magnitude of the corresponding energy recycle stream. As the duties of these FEHEs span multiple orders of magnitude, there is

a potential for the presence of several large energy recycle loops. On the other hand, the duty of the heating/cooling utility represents an external energy flow and the presence of different orders of magnitude in their values suggests that this system also involves large energy throughputs. In essence, the prototype networks with large energy recycle and throughput form the backbone of this integrated system.

Let us now apply the analysis framework presented in section 2 to the reactor-HE system in Fig. 1. The three algorithms presented in section 2 are applied to this system. The cycle finding algorithm proposed in Tarjan (1973) has been used in the subroutine *SmallestElementaryCycle*. All the algorithms are coded in C++. To begin with, the energy flow graph for this system is constructed. All the process-to-process heat exchangers are represented by two nodes, one for the cold side and the other for the hot side, connected by a directed edge from the hot node to the cold node (see Fig. 2). The definitions of the various nodes of this energy flow graph are tabulated in Table 2.

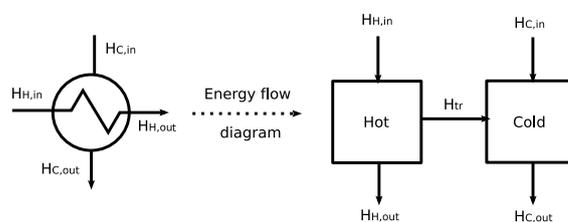


Fig. 2. Energy flow structure for a process-to-process heat exchanger

Table 2. Node details for the reactor-HE system

Index	Node details	Index	Node details
Primary nodes			
1	PPX_1 -cold	18	PPX_7 -cold
2	PPX_1 -hot	19	PPX_7 -hot
3	PPX_2 -cold	20	C_5
4	PPX_2 -hot	21	H_2
5	H_1	22	Reformer-1
6	Desulphurisation reactor	23	H_3
7	PPX_3 -cold	24	Reformer-2
8	PPX_3 -hot	25	PPX_8 -cold
9	PPX_4 -cold	26	PPX_8 -hot
10	PPX_4 -hot	27	C_2
11	C_1	28	C_3
12	Gas separation	29	Mixer
13	Stripping column	30	Compressor
14	PPX_5 -cold	31	PPX_9 -cold
15	PPX_5 -hot	32	PPX_9 -hot
16	PPX_6 -cold	33	C_4
17	PPX_6 -hot	34	Stabilizing column
Auxiliary nodes			
35	H_1 -head	43	Stripping column energy input
36	H_2 -head	44	Reformer-1 heat of reaction
37	H_3 -head	45	Reformer-1 heat of reaction
38	C_1 -tail	46	Stabilizing column energy input
39	C_2 -tail	47	Compressor power input
40	C_3 -tail	48	Desulphurisation heat of reaction
41	C_4 -tail	49	Naphtha feed-head
42	C_5 -tail	50	Aromatics product-tail

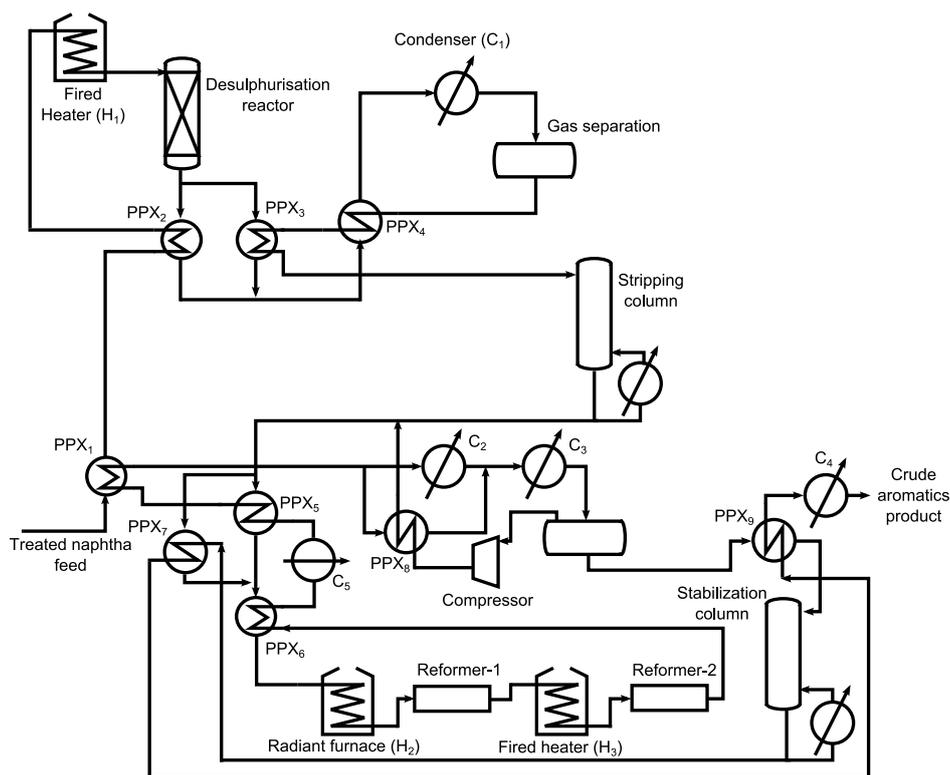


Fig. 1. Energy integrated reactor-heat exchanger network for naphtha reforming

The advantage of the proposed framework is that it requires minimum process information, specifically the orders of magnitude (rather than actual magnitudes) which can be easily inferred based on unit-level and overall energy balances. For example, for the heat exchanger PPX_2 in this system, the only information available is that the rate of heat transfer in this exchanger is large (say $\mathcal{O}(1/\varepsilon)^1$). The energy balance across this exchanger sets the hot inlet flow and the cold outlet flow to be $\mathcal{O}(1/\varepsilon)$. This, however, does not fix the orders of magnitude for hot exit flow and the cold inlet flow (which can be either $\mathcal{O}(1)$ or $\mathcal{O}(1/\varepsilon)$). The energy balance across the exchanger PPX_1 provides the solution. It indicates that all the energy flows in this exchanger should be of the same order of magnitude. As the energy flow corresponding to the naphtha feed is assumed to be $\mathcal{O}(1)$, all the energy flows in PPX_1 and the remaining energy flows in PPX_2 are $\mathcal{O}(1)$. A feasibility condition (as shown in the following subroutine) can be easily added in the energy flow graph definition algorithm. Based on the energy balance across each node, the validity of the arguments made about the orders of magnitude of the various energy flows can be verified.

Subroutine *FeasibilityCheck(G)*

- 1: Feasibility = TRUE
- 2: **for all** node $N \in G$ **do**
- 3: MaxInOrder = highest order of magnitude among the edges entering node N

- 4: MaxOutOrder = highest order of magnitude among the edges leaving node N
 - 5: **if** MaxInOrder \neq MaxOutOrder **then**
 - 6: "Energy flow structure is infeasible"
 - 7: Feasibility = FALSE
 - 8: **end if**
 - 9: **end for**
 - 10: **return** Feasibility
-

The analysis for this system (or any other system) is performed in the following sequence.

- (1) Input energy flow graph G .
- (2) Check feasibility of the graph.
- (3) Identify time scales in the network using algorithm 1.
- (4) Check feasibility of the recycle structures (sub-graphs).
- (5) Generate dynamic models in each time scale using algorithm 2.
- (6) Classify controlled variables and manipulated inputs using algorithm 3.

The results from the application of algorithm 1 to this system are included in Table 3. It can be noted that the various energy balance variables evolve over two time scales. Specifically, the enthalpies of the cold and hot sides of PPX_6 (16 and 17), the heaters H_2 (21) and H_3 (23) and the two reformers (22 and 24) evolve only in the fast time scale. It can be noted that these variables are a part of a large throughput from the large heater H_2 duty to the endothermic heat of reaction in reformer-1 (via reformer-2, PPX_6 , H_2 and reformer-1). The enthalpies of the cold

¹ $\varepsilon = 0.1$, a small number

and hot sides of PPX_2 (3 and 4), the heater H_1 (5) and the desulphurisation reactor (6) evolve in both the fast and the slow time scale, owing to the presence of a large recycle loop (via PPX_2 , H_1 , desulphurisation reactor). All the other energy balance variables evolve only in the slow time scale.

Table 3. Results of algorithm 1

Timescale	Nodes evolving	# of pure recycles
τ_2	3, 4, 5, 6, 16, 17, 21, 22, 23, 24	1
τ_1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 18, 19, 20, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34	0
τ_0		0

Algorithm 2 was then applied to this graph. The code successfully generated scaled energy balance equations. Equation (1) represents the original scaled energy balance equations. It can be noted that this equation is stiff and shows a potential for multi-time scale dynamics. H represents the vector of all the enthalpy balance variables. $g_0(u_0)$, $g_1(u_1)$ and $g_2(u_2)$ are scaled vectors.

$$\frac{dH}{dt} = \epsilon g_0(u_0) + g_1(u_1) + \frac{1}{\epsilon} g_2(u_2) \quad (1)$$

The corresponding scaling rules (2) and (3) were also generated:

$$\mathcal{O}(1) \text{ steady state ratios, } k_{i-j} = \frac{h_{i-j,s}}{h_{w,s}} \quad (2)$$

$$\mathcal{O}(1) \text{ scaled flows, } u_{i-j} = \frac{h_{i-j}}{h_{i-j,s}} \quad (3)$$

where, $h_{w,s}$ is a reference energy flow of weight w , where w is the weight of the energy flow h_{i-j} .

The dynamic equations in the fast time scale $\tau_1 = t/\epsilon$ are given by Eq. (4).

$$\frac{dH_2}{d\tau_1} = g'_2(u_2) \quad (4)$$

where $H_2 = \{H_3, H_4, H_5, H_6, H_{16}, H_{17}, H_{21}, H_{22}, H_{23}, H_{24}\}$. $g'_2(u_2)$ represents a subset of the vector $g_2(u_2)$ corresponding to enthalpy vector H_2 . As there is a large energy recycle loop, the constraints from the fast time scale are linearly dependent. The dynamics in the slower time scale is given by Eq. (5).

$$\begin{aligned} \frac{dH_1}{d\tau_0} &= g'_1(u_1) + C_1 B_1 z_1 \\ \tilde{g}'_2(u_2) &= 0 \end{aligned} \quad (5)$$

where $\tilde{g}'_2(u_2) = 0$ represents linearly independent constraints. H_1 represents the corresponding vector of energy balance variables evolving in this time scale. z_1 represents algebraic variables, thus making this slow dynamics a DAE system.

Note that the exact forms of the various vectors and matrices are generated by the code but, for the sake of brevity, are not reproduced here.

Lastly, the results from the application of algorithm 3 are presented in Table 4. Variables that should be controlled in the fast time scale are the temperatures of cold and hot sides of PPX_2 (3 and 4) and PPX_6 (16 and 17), the heater H_1 (5), the desulphurisation reactor (6), the heaters H_2 (21) and H_3 (23) and the two reformers (22 and 24). It was also indicated that one out of the temperatures of cold and hot sides of PPX_2 , the heater H_1 and the desulphurisation reactor (6) should be left uncontrolled in this time scale. Some of the potential manipulated inputs include the heating duty of H_3 (37→23), the heat transfer rate across PPX_2 (4→3) which can be varied using a bypass stream across this exchanger. In the slow time scale, most of the variables evolving in that time scale are to be controlled along with the total enthalpy of the recycle block comprising of PPX_2 , H_1 and desulphurisation reactor.

Table 4. Results of algorithm 3

Timescale	Potential control variable Node	Potential manipulated input edge
τ_2	16	3 → 5 (Internal flow)
	17	4 → 3 (Internal flow)
	21	5 → 6 (Internal flow)
	22	6 → 4 (Internal flow)
	23	16 → 21 (Internal flow)
	24	17 → 16 (Internal flow)
	any 3 out of 3, 4, 5 & 6	21 → 22 (Internal flow)
		22 → 44 (External flow)
		23 → 24 (Internal flow)
		24 → 17 (Internal flow)
	37 → 23 (External flow)	
τ_1	1, 2, 7, 8,	1 → 3 (Internal flow)
	9, 10, 11, 12,	2 → 1 (Internal flow)
	13, 14, 15, 18,	2 → 26 (Internal flow)
	19, 20, 25, 26,	2 → 27 (Internal flow)
	27, 28, 29, 30,	7 → 13 (Internal flow)
	31, 32, 33, 34	8 → 7 (Internal flow)
	3 + 4 + 5 + 6	8 → 10 (Internal flow)

This example clearly illustrates the predictive power of the proposed analysis framework. It should be emphasized that minimal process information is utilized to generate these results very efficiently (it took less than 1 second to run all these algorithms together for this system on a Intel(R) Core(TM)2 Quad CPU with 2.40 GHz clock speed and 4 GB of RAM).

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an application of a graph theoretic framework for the analysis of a complex energy integrated network. The framework offers significant advantages over an analytical approach using singular perturbations. Specifically,

- the graph-based approach is scalable to large networks
- it requires minimum process information to uncover dynamic time scale multiplicity or identify controllability issues
- it can be easily automated and can serve as an efficient tool to compare and screen different design alternatives based on controllability of the integrated system

The graph-theoretic framework is generic and can find applications in a wide range of energy integrated systems. One particular area we are currently exploring is the analysis of cryogenic systems. Energy integration is prevalent in cryogenic systems as the supply of refrigeration at cryogenic temperatures is highly energy (and thus cost) intensive. These systems are, however, characterized by the presence of recovery and recycle of refrigeration. These systems work the opposite way as compared to conventional high temperature integrated systems. For example, a colder stream carries higher refrigeration than a hotter stream. The graph-theoretic framework will therefore be modified to analyze such refrigeration flow graphs.

ACKNOWLEDGEMENTS

Partial financial support for this work by National Science Foundation (through Grant CBET-1133167) is gratefully acknowledged.

REFERENCES

- Y.H. Chen and C.C. Yu. Design and control of heat-integrated reactors. *Ind. Eng. Chem. Res.*, 42(12):2791–2808, 2003.
- S.S. Jogwar, M. Baldea, and P. Daoutidis. Dynamics and Control of Process Networks with Large Energy Recycle. *Ind. Eng. Chem. Res.*, 48(13):6087–6097, 2009.
- S.S. Jogwar, M. Baldea, and P. Daoutidis. Tight energy integration: Dynamic impact and control advantages. *Comput. Chem. Eng.*, 34(9):1457–1466, 2010.
- J.C. Morud and S. Skogestad. Analysis of instability in an industrial ammonia reactor. *AIChE J.*, 44(4):888–895, 1998.
- R.E. Tarjan. Enumeration of the elementary circuits of a directed graph. *SIAM J. Comput.*, 2(3):211–216, 1973.
- J. Varghese and S. Bandyopadhyay. Targeting for energy integration of multiple fired heaters. *Ind. Eng. Chem. Res.*, 46(17):5631–5644, 2007.
- Y. Zhu and X. Liu. Dynamics and control of high purity heat integrated distillation columns. *Ind. Eng. Chem. Res.*, 44(23):8806–8814, 2005.
- J.B. Rawlings and B.T. Stewart. Coordinating multiple optimization-based controllers: New opportunities and challenges. *J. Process Contr.*, 18(9):839–845, 2008.
- J. Liu, D. Muñoz de la Peña, B.J. Ohran, P.D. Christofides, and J.F. Davis. A two-tier architecture for networked process control. *Chem. Eng. Sci.*, 63(22):5394–5409, 2008.
- D. Hioe, J. Bao, and E.B. Ydstie. Dissipativity Analysis for Networks of Process Systems. *Comput. Chem. Eng.*, 50:207–219, 2013.
- M. Baldea, N.H. El-Farra, and E.B. Ydstie. Dynamics and Control of Chemical Process Networks: Integrating Physics, Communication and Computation. *Comput. Chem. Eng.*, 51:42–54, 2013.
- N. Hudson and J. Bao. Dissipativity-based decentralized control of interconnected nonlinear chemical processes. *Comput. Chem. Eng.*, 45:84–101, 2012.
- S. Heo, S.S. Jogwar, S. Rangarajan, and P. Daoutidis. Graph reduction for hierarchical control of energy integrated process networks. *IEEE 51st Annual Conference on Decision and Control*, 6388–6393, 2012.
- S.S. Jogwar, S. Rangarajan, and P. Daoutidis. Multi-time Scale Dynamics in Energy-integrated Networks: A Graph Theoretic Analysis. *IFAC World Congress*, 18(1):6085–6090, 2011.
- R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, 1993.
- T.H. Cormen. *Introduction to algorithms*. The MIT Electrical Engineering and computer Science Series. MIT Press, 2001.