Detection and Control of Invading Species in Autocatalytic Networks

Sukanya Balasubramanian,Fouad Teymour,Ali Cinar

Department of Chemical and Biological Engineering,
Illinois institute of Technology,Chicago,IL-60616,USA
Email: [balasuk,teymour,cinar] @iit.edu

1 Introduction

The autocatalytic paradigm is gaining increasing popularity for its diverse range of application in population biology concepts, DNA replication and RNA polymerization studies. When autocatalytic species are populated in a continuous stirred tank reactor (CSTR) environment, they display rich static and dynamic phenomena. In the natural setting, most of the living species exhibit features of survival, coexistence and different degrees of interaction with other species. Birol and Teymour (1) showed that competition of multiple species populating in a CSTR leads to survival of one of the populations. Chaivorapoj (2; 3) extended the idea by considering a single CSTR hosting a *robust* species, perturbed by a disturbance of a *non-robust* or *invading species* in the inflow. When the invading species gets a foothold in the system, concentration of the host species decreases considerably. A feedback control design was implemented to eliminate the invading species and restore original concentration of host species. For simple reactor configurations, bifurcation analysis serves as a potential tool to provide insight into species behavior. As the number of reactors increases, controller tuning using bifurcation techniques fail owing to a geometric increase in steady states. The control problem is all the more complex because each reactor contains an inlet, exit stream and interconnections with its neighbors.

In this paper, a hierarchical agent based system is implemented for detection and elimination of an invading species from a network of CSTRs. An interconnected grid network of 25 reactors is designed to produce a host species. At some time instant, a pulse disturbance of invading species enters a specific location in the network. The "poison" spreads in the

network, thereby causing deleterious effects to the system behavior. Agent based systems are a good option for use with large scale, distributed systems (4). Multi-agent systems exhibit intelligence through collective behavior of agents with simple functionalities (5; 6). In order to device detection and control strategies, the task of a subject matter expert is to perform a large number of simulations with pre-selected random values for parameters to collect information about process behavior. The process behavior is evaluated for conditions under normal operation and invasion. Depending on the magnitude of invasion, growth and death rates of the invading species, a variety of process behaviors evolve. A knowledge engineer characterizes the process behavior and stores the information in a knowledge base as rules. The knowledge structure is built using an expert system tool called "Knowledge Builder", where a Knowledge Base (KB) is configured and built using the tool's graphical user interface. Given a current state of the process, an agent invokes the expert system, which has the ability to make a decision on how to react, based on its configured knowledge. The output from the knowledge base in the form of actuator information is relayed to the system.

2 Mathematical Model

A network of *I* inter connected CSTRs (figure 1) is modeled by specifying material balances for each reactor *i* $(i = 1...N)$. The reactors in the network are designed such that the volume ratio of the reactors on the outer edge to reactors in the core is 1:3. The design is robust to disturbances since the smaller reactors shield the inner reactors, where the bulk of the product is being produced. In order to ensure long term survival of the host species, small concentrations are always present in the feed. The cubic autocatalytic reaction for *N* host species and *M* invading species is

$$
\begin{aligned}
&R + 2P_n \rightarrow 3P_n \\
&P_n \rightarrow D_P \\
&R + 2Q_m \rightarrow 3Q_m \\
&Q_m \rightarrow D_Q
\end{aligned}
$$

where R is the resource, P_n is the n^{th} host species, Q_m is m^{th} invading species and $D_{P(Q)}$ are the inert (dead) species. The growth rates and death rates define the strength

of invading species. The rate of change of resource, host species and invading species in reactor *i* can be written as

$$V_i\frac{dR_i}{dt} = F_0R_0 - F_{out,i}R_i + G(R' - R_i) - \sum_n k_{rp}R_iP_{n,i}^2$$
$$V_i\frac{dP_{n,i}}{dt} = F_0P_0 - F_{out,i}P_i + G(P' - P_{n,i}) - k_{rp}R_iP_{n,i}^2 - dAb_pAb_iP_{n,i}$$
$$V_i\frac{dP_{m,i}}{dt} = F_0Q_0 - F_{out,i}Q_i + G(Q' - Q_{m,i}) - k_{rq}R_iQ_{m,i}^2 - dAb_qAb_iQ_{n,i}$$
$$V_i\frac{dAb_i}{dt} = F_0Ab_0 - F_{out,i}Ab_i + G(Ab' - Ab_i) - \sum_n dAb_pP_{n,i} - \sum_m dAb_qQ_{m,i}$$

where R_0 is the resource concentration in feed, P_0 is the host species concentration in feed, Q_0 is the temporary invasion disturbance in reactor *i*, R_i is resource concentration in reactor *i*, $P_{n,i}$ is n^{th} host species concentration in reactor *i*, $Q_{m,i}$ is the m^{th} invading species concentration in reactor *i*, F_0 is the inlet flow rate, F_{out} is exit flow rate, G matrix defines the inter connection strength with neighboring reactors. Ab_i is the "antibiotic" concentration required to kill the invading species in reactor i. The presence of the antibiotic also leads to death of some of the desired species.

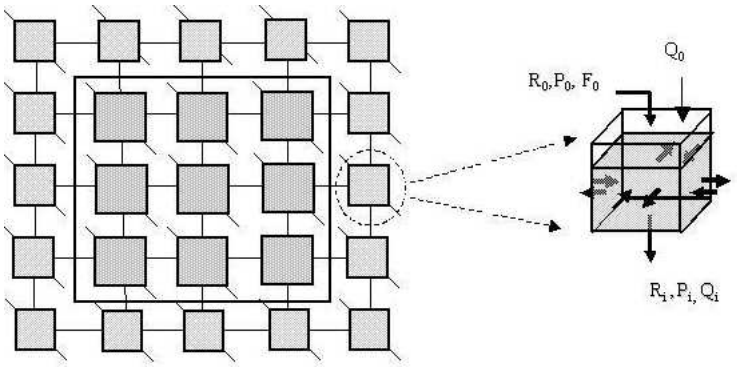


Fig. 1. (a) Network of interconnected reactors (b) Detail view of a reactor with inlet, outlet and interconnections.

3 Process Behavior

Detection and control strategies for a network poisoned by an external source cannot be designed from intuitive knowledge. Since the concentration of the invading species is

an immeasurable quantity, a detection scheme should be designed based on diminishing resource and product species concentrations. The control challenge will be to choose among different control alternatives depending on the extent of invasion in the network. In order to device these strategies, it is necessary to collect information about the behavior of the system under invasion and non-invasion conditions offline. The information contribute to building heuristic knowledge. As a knowledge building exercise, a series of open loop dynamic case studies are carried out for various operating parameters to account for transient and steady state characteristics.

Normal Operation. 'Normal Operation' refers to the network behavior when only a host species populates in the system for different parameter values. In each run, the network parameters, feed flow rates, interaction flow rates and initial resource concentrations are chosen in the range $F_0(0.15-1), g(0-3)$ and $R_0(0.8-3)$ respectively. A CVODE solver (7) is used to compute the time variation of resource and host species concentration states. A large number of such dynamic simulations are carried out to explore the boundaries of network behavior. The dynamic data organize themselves into a manifold, which relates the consumption of resource to production of host species (figure 2). The behavior for large values R, P correspond to large values of inlet resource concentration. In the ensuing sections we will focus on a fixed $R_0 = 1$ units. The steady states corresponding to lower values of R and P values concentrate on a continuous line with negative slope while the transient data appear as scatter points. A direct least squares algorithm (8) is used to fit an approximate ellipse around the cluster data. Since the dynamics are computed for random operating conditions, all the reactors produce the same control ellipse. For data points within the control ellipse, the gradient in concentration for host species and resource is often negative. A Matlab implementation of the ellipse-fitting algorithm proposed by Fitzgibon, results in an ellipse shown in figure 2.

Invasion. An invading species in the form of a concentration pulse disturbance in the feed is introduced at a specific location in the network. The invasion spreads to other reactors in the network because of inter connections with its neighbors. After the pulse disturbance is removed, the undesirable species can disappear from the environment, coexist with the host species, flush out the host species or propagate through the network as a disturbance. The extent to which the network is affected depends on the strength of invading species. A "strong invading species", characterized by large growth rates compared to the host species, gains a foothold in the system even after the disturbance is removed. Figure 3 illustrates the spectrum of network behavior under the effect of an strong invasion disturbance introduced in reactor 1. The initial conditions and final

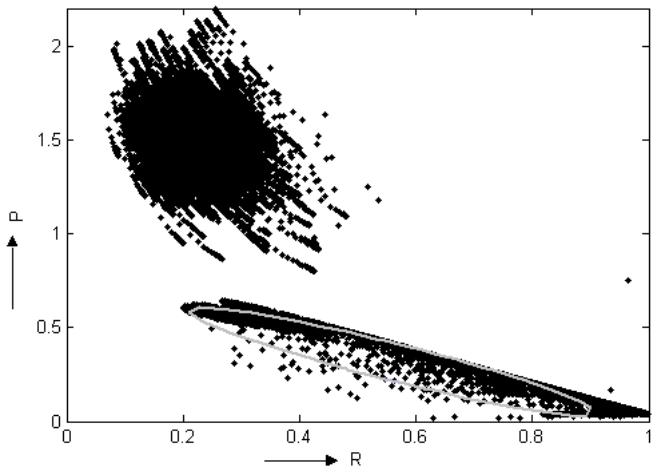


Fig. 2. Normal Operation behavior for operating parameters in the range $F_0(0.15-1), g(0-3)$ and $R_0(0.8-3)$

conditions for dynamics are represented by gray clusters and the transients are indicated by black dots. A pulse disturbance is introduced at $t = (150 - 200)$ time units with a magnitude of $Q_0 = 1$ units and strength $(k_{rq}, d_q) = (24.576, 0.3840)$ units . The initial conditions for dynamics are chosen to be normal operating steady states arising from random configurations of feed flow rates. The plots are constructed on a $R - P$ domain. Since Q is an immeasurable quantity, the growth and death of the invading species with measured in terms of consumption of resource and diminishing host species. In the invading reactor and its neighbors, Q gains a foothold in the system very quickly, whereas the distant reactors hit invasion after a delay. Figure 4 shows the response of the network to a "moderate invading species" with strength $(k_{rq}, d_q) = (10.0, 0.3840)$ introduced in reactor 1. The invasion responses (gray dots) are projected over the normal operating data (black dots). At the end of the simulation, the possible invasion responses are: the invading reactor may flush out the host species permanently or temporarily; the invading reactor neighbors hit invasion or contain traces of Q; in the distant reactors, the invasion propagates as a disturbance. A weak invading species with strength $(k_{rq}, d_q) = (2.0, 0.3840)$ stays within the normal operation thresholds.

Characterizing transients One striking observation from figure 4 is the invasion data deviate from normal operating regions. The presence of an invading species causes a drastic decrease in resource concentration and indirectly affects the production of the host species. The phenomenon is a strong indication of invasion and can be interpreted quantitatively as the occurrence of a positive concentration gradient. Starting from the normal operating steady states (continuous line), the large amount of network behavior data can be distilled into follow categories - the system immediately goes to an invasion

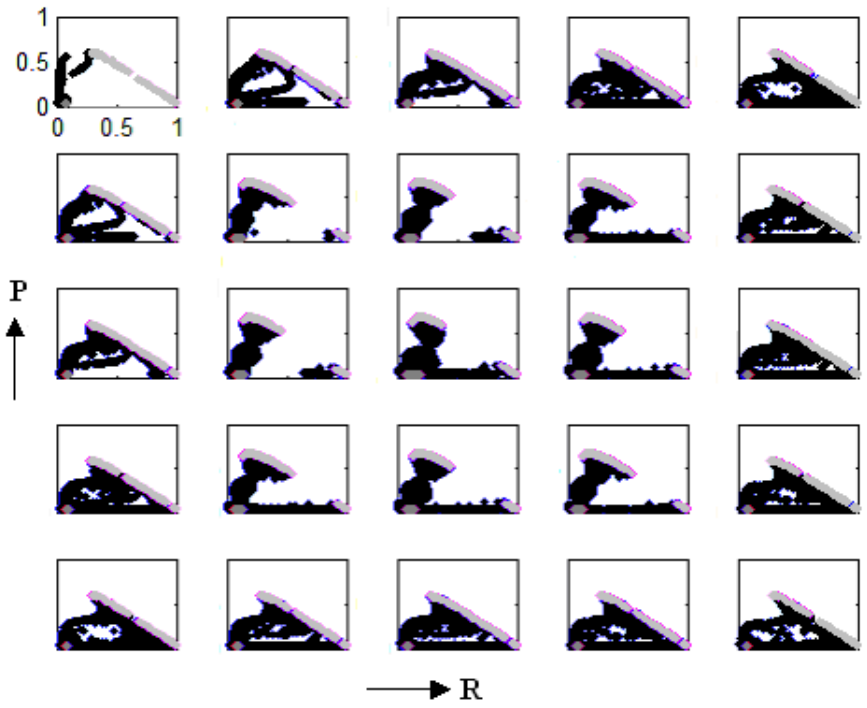


Fig. 3. Open loop responses when the network is attacked by a strong invading species.

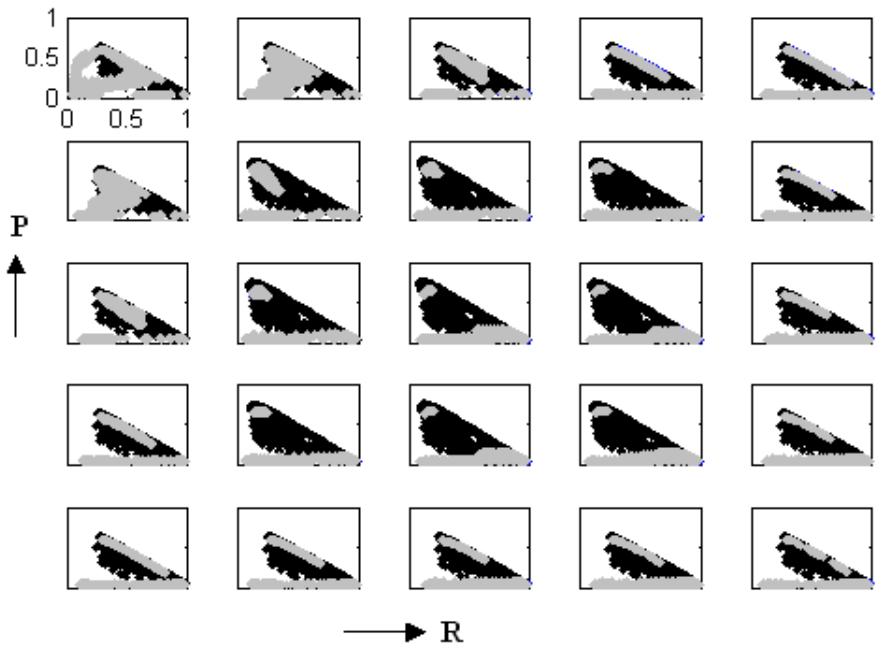


Fig. 4. Open loop responses under moderate invasion

steady state (figure 5a), heads to invasion state after a delay (figure 5b), instantly heads in invasion direction but loops back to its original steady state (figure 5c), loops back to

its original state with a delay term (figure 5d) or settles down at coexistence steady state (figure 5e). The gradient sign changes are reflected in the figure.

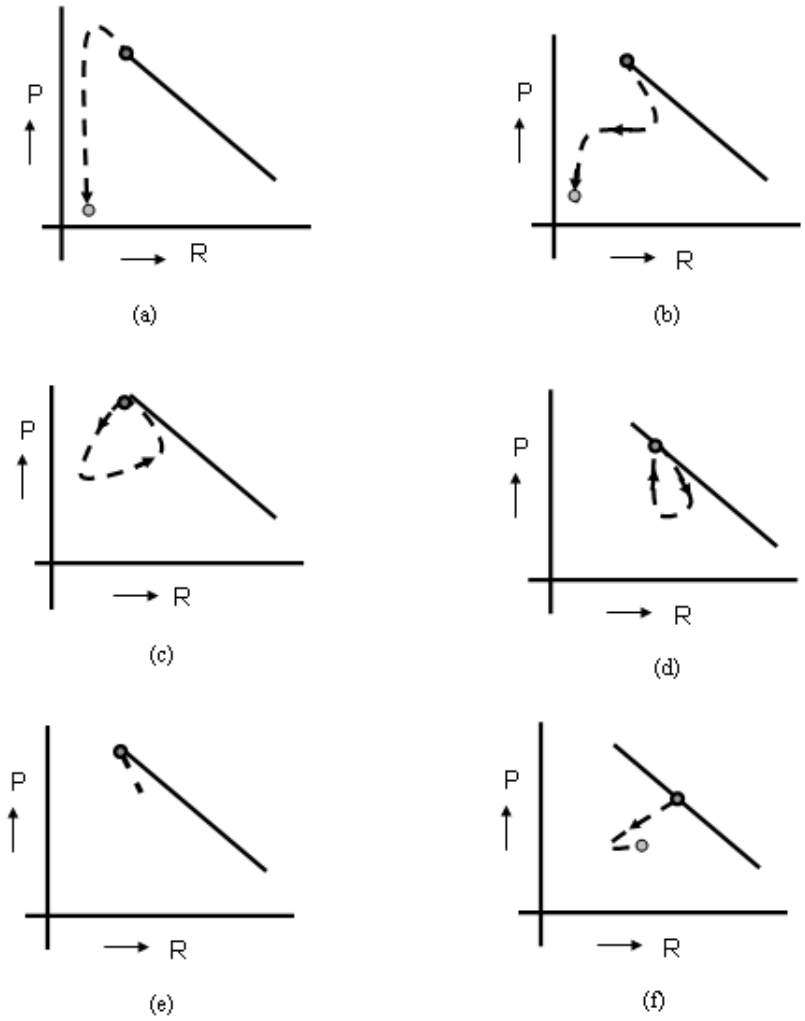


Fig. 5. Transients during invasion (a) Immediate invasion, $dpdr(+ve)$,(b) Invasion after delay $dpdr(-ve,+ve)$, (c) Immediate Looping $dpdr(+ve,-ve)$ (d) Looping after delay $dpdr(-ve,+ve,-ve)$ (e) Disturbance propagation, (f) Coexistence steady state

4 Knowledge Representation

Knowledge from the process must be translated into a more formalized representation to be implemented into a knowledge base. One type of knowledge representation scheme is "production systems" where rules are used as a mode of reasoning (9). In a rule-based system environment, an inference engine determines which rules are applicable and which

of these candidate rules should executed in a given situation. The rules are expressed in the form

$IF < conditions > THEN < actions >$

The control mechanism is defined by a recognize-act cycle. The "recognize" step determines which rule to fire and the "act" step executes the conclusion of the rule.

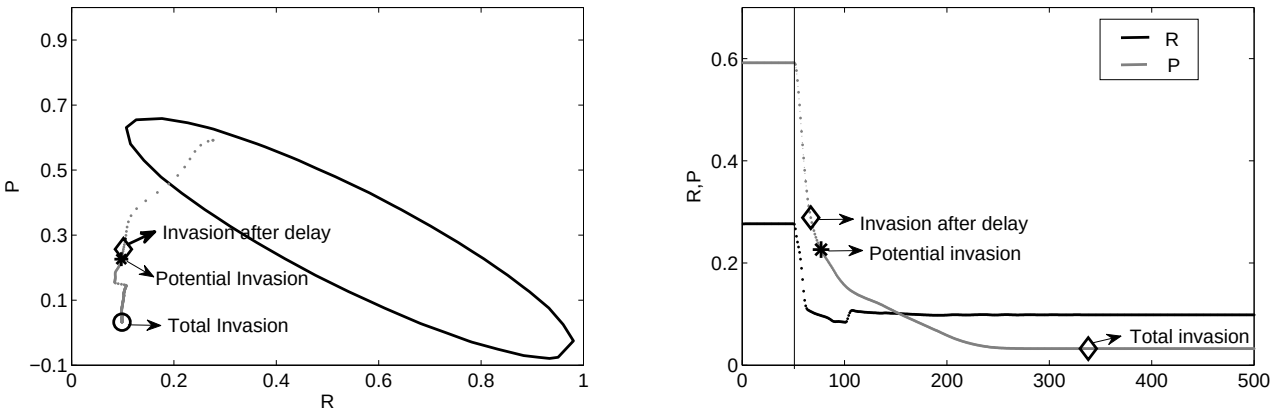
To illustrate knowledge representations, we will build a complete program using "MADCABS Knowledge Builder", which is a novel tool for analyzing production systems. The knowledge builder overlooks process operation conditions and provides necessary detection and control.In the ensuing section, we will describe the detection and control strategies for our sample system, give an overview of knowledge builder, and discuss a very simple implementation and operation of the program.

System. The network of 25 reactors (figure 1) is desired be operating under "normal" conditions. At each time instant, a detection scheme checks whether reactors in the network give rise to any of the following flags,

- Total invasion flag - Raised when the system reaches an invasion steady state
- Potential invasion flag - Raised when the system proceeds in an invasion direction
- Indirect invasion alert- One of neighboring reactors is affected by invasion
- Invasion delay flag - Raised when the system proceeds in an invasion direction after some time delay.

The detection algorithm computes the invasion flags based on the relative location of the (R,P) projections wrt the control ellipse and gradient changes with time. The detection scheme accounts for minor fluctuations in gradient values and outliers in data. For example, a potential invasion flag is raised when an (R, P) projection returns "*outside the ellipse*" and the *dp/dr* gradient is positive for about 20 time units. The indirect invasion alerts facilitate faster detection. Figure 6 shows the flags raised in the R-P domain and concentration time domain in reactor 2.Table 1 shows the results from the detection algorithm designed using simple IF-THEN rules in MATLAB. We observe that the indirect invasion flags and potential flags provide faster detection capabilities.

The control strategies are designed based on the detection flags raised. In case of a potential invasion flag, the control strategy adopted is to isolate the reactor, add an "antibiotic" to flush out the undesirable species and reconnect the interactions. Other



(a) (b)

Fig. 6. Flags raised by the detection algorithm in reactor 2 in (a)R-P domain (b)Concentration vs time domain

#	Total	Potential	Indirect	#	Total	Potential	Indirect
1	329	71	-	**10**	338	177	170
2	338	77	71	**13**	419	321	200
3	364	110	95	**14**	398	293	205
4	338	154	141	**15**	366	198	189
5	334	170	157	**17**	332	194	169
7	333	134	84	**19**	363	290	205
8	398	200	112	**20**	339	220	211
9	332	194	169	**25**	334	232	228

Table 1
Invasion flags raised by the detection algorithm as time progresses

control strategies such as a slight tweaking of flow rates may be necessary to remove trace quantities of undesirable species. Figure 7 shows the implementation of the control strategy on the invaded reactor. We observe that the invading species introduced between $t = (150 - 200)$, kills the host species. At $t = 400$, an antibiotic addition, eliminates the invading species, while a small amount of host species is still present in the system. The original concentration of host species can be recovered by adding additional species in the feed.

Knowledge Builder and its implementation. In order to understand the working and implementation of Knowledge Builder, we shall apply it to a simple application, an interconnected network of 2 reactors. Each reactor contains an inlet and outlet feed stream and inter connections with its neighbors. Each reactor contains a stable configuration of

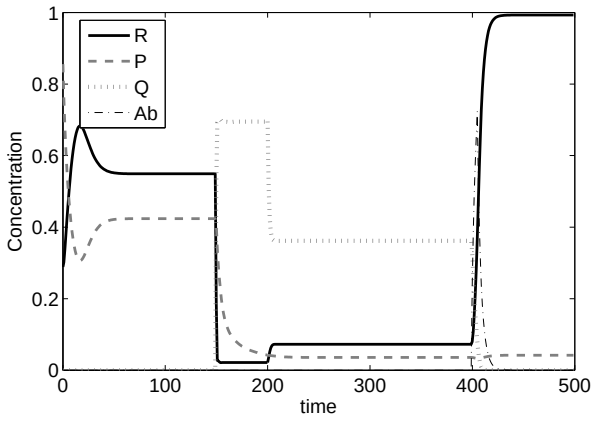


Fig. 7. Antibiobic action

different autocatalytic species. The control objective calls for replacing species in reactor 2. The reactor providing the desired species is designated as "source reactor" and the reactor to be changed is the "target reactor". The desired transformation can be achieved by (a) Increasing interconnection flow rate to target reactor (b) Incase the increase is not sufficient, seek assistance from target reactor.

The knowledge builder components for the source reactor is shown in figure 8. The knowledge builder's GUI consists of panels which explain different parts of the rule base. The "subject panel" shows the working memory or the current state of the process. State variables, input variables and local variables populate the knowledge base. The process simulator operates in a continuous mode. At each time instant, the simulator provides the state and input information. The knowledge builder program updates the state and input variables based on changes in a "watch file". Variables such as *controller status, arbitrator status* are local to the knowledge base and are set to default values. "Status Panel" shows the matching routines for the components. For example, in figure 8, the controller checks if the subject "controller status" is assigned to value "attack". The program will test conditions based on changes to the status of the system. It will then fire all the rules that depend on this status being true. For each of the rules to be fired, it will execute all of the rule actions that are assigned weight one. The "rule panel" shows what actions to execute based on the predicates specified in the "action panel". For example, the rule named "Rule controller" initiates the action "Proportional Controller" which increases interaction flow rate. The output from the agents is relayed to the system in the form of an "out file". The process reads the actuator settings from an out file. Each reactor has an instance of the knowledge base, which fire different rules based on the initial conditions. The knowledge base for each reactor is maintained by a "Knowledge Modeler" GUI figure

(9). The knowledge modeler allows negotiation protocols between the knowledge engines of each reactor.

The performance of the control system is demonstrated in the following example. Two species with identical growth and death rates $k_r = 25, d = 0.1$ populate in the reactor network. Initially, feed flow rates $f = 0.008$ and interaction flow rates $g = 0.001$ are identical in both reactors. When the interaction flow rates are manipulated, the outflows are adjusted to maintain a constant volume. The initial species concentration profile for reactor 2 is shown in region(a) of figure 10. In times $t = (0 - 1000)$, the knowledge builder only updates the process variables. At $time = 1000$ units, the *controller status* of the source reactor is set to *attack*. The knowledge base fires rules to increase the interaction flow rate from the source to the target reactor. We observe that this change is not sufficient to overtake reactor 2 (region (b)). In region (c), the target reactor reactor is exploited to increase its interaction flow rate to the source reactor. The target reactor requests the source reactor reactor to double its output. The interaction flow rate bounds are continuously updated and maintained in an arbitrator agent. Requests are accepted if the requested changes are within the bounds.

5 Research Ahead

The appropriate tools required for the scalability of knowledge builder are under development. A general template will be used to describe the basic knowledge of each reactor in the network. Once the network topology and inter connections are determined, an instance of the knowledge base is copied to every reactor. Additional information (subjects, statuses, rules) specific to each knowledge base can be incorporated into the builder. The knowledge builder should be able to support access settings such as private (local to a KB), protected (visible to its environment) and public (visible to the network) for the subject variables. The environment for a reactor is defined by its neighbors. A reactor engine can have access to protected and public subjects for its neighbors. Generalized request handling procedures for subjects need to be implemented. Another direction of research will be to explore the panorama of conflict resolution strategies (10). Some of the strategies available in literature are: rule selection based on a priority order, newly activated rules are placed above all activations, random activation of rules and many others.

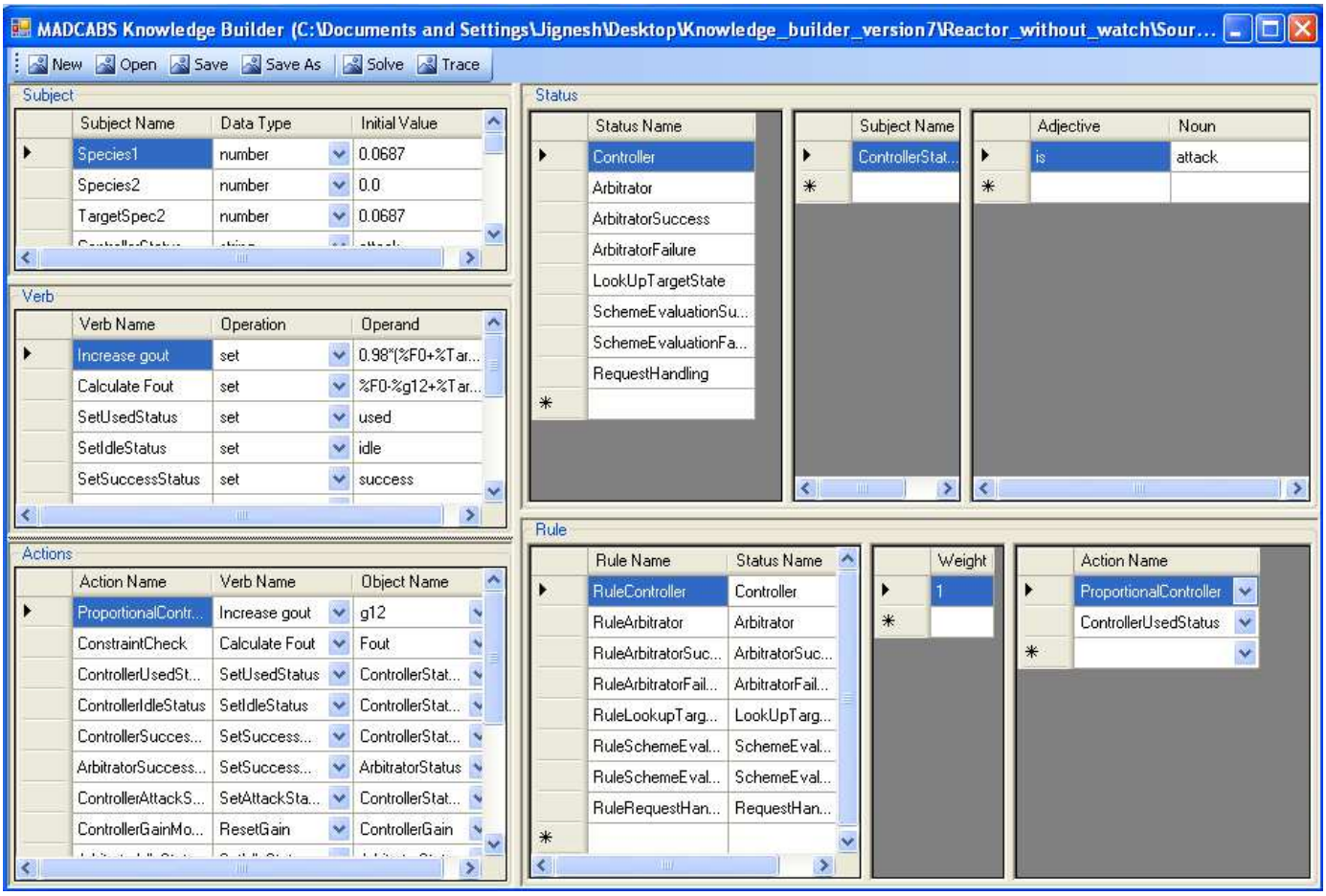


Fig. 8. Knowledge Builder representation for a 2 reactor system

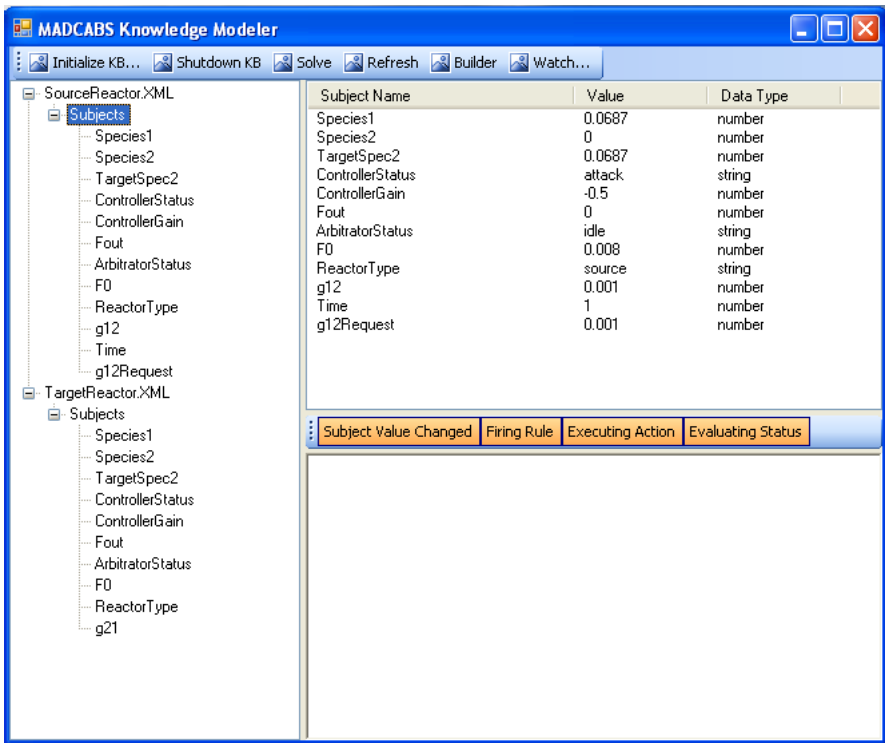


Fig. 9. Knowledge Modeler illustration for a SourceReactor and Target reactor

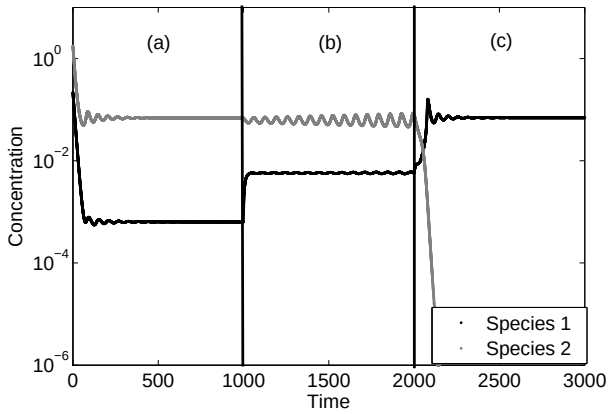


Fig. 10. Strategy for replacing autocatalytic species in reactor 2 (a) $(g_{12}, g_{21}) = (0.001, 0.001)$ (b)$(g_{12}, g_{21}) = (0.0088, 0.001)$),$(g_{12}, g_{21}) = (0.0165, 0.0088)$

References

[1] Birol, I., Teymour, F., "Statics and Dynamics of Multiple Cubic Autocatalytic Reactions," *Physica D*, Vol. 144, pp.

[2] Chaivorapoj, W., Birol, I., Cinar, A., and Teymour, F.,, "Feedback Control of a Continous-Flow Stirred Tank Reactor with Competing Autocatalators," *Ind. Eng. Chem. Res*, Vol. 42, pp. 3765-3785, 2003.

[3] Chaivorapoj, W., Birol, I., and Teymour, F.,, "Competition between Robust and Non-Robust Autocatalytic Replicators," *Ind. Eng. Chem. Res*, Vol. 42, pp. 3765-3785, 2003.

[4] Tatara,E., Birol,I.,Teymour,F.,Cinar,A.(2005),"Agent Based Control of Autocatalytic Replicators in Networks of Reactors," *Computers and Chemical Engg*,Vol 29, pp.807-815.

[5] Tatara,E.,Cinar,A.,Teymour,F.(2007),"Control of Complex distributed Systems with Distributed Agents," *Journal of Process Control*,Vol 17, pp.415-427.

[6] Tetikar,D.,Artel,A.,Tatara,E.,Teymour,F.,North,M.,Cinar.A.,(2006),"Agent Based System for Reconfiguration of Distributed Chemical Reactor Network Operation," *Proceedings of American Control Conference*,Minneapolis.

[7] Hindmarsh, A., Taylor (1998), A. PVODE and KINSOL: Parallel Software for Differential and Nonlinear Systems; Lawrence Livermore National Laboratory Technical Report UCRL-ID-129739; Lawrence Livermore National Laboratory: Livermore, CA.

[8] Fitzgibbon,A.,Pilu,M.,Fisher,R.(1999),"Direct Least Square Fitting of Ellipses," *IEEE transactions on Pattern Analysis and Machine Intelligence*,Vol 21, pp.476-480.

[9] Cronk,R.,Callahan,P.,Bernstein,L.(1988),"Rule based Expert Systems for Network Management and Operations: An Introduction," *IEEE Network*,September, pp.7-18.

[10] Hicks,R.,(2007),"The No-Inference Engine Theory - Performing Conflict Resolution during Development," *Decision Support Systems*,pp.434-444.