Eigenvalue Analysis of System Dynamics Models
Another Perspective

Mohamed Saleh

Decision Support Department
Cairo University

msaleh@idsc.net.eg

Rogelio Oliva

Mays Business School
Texas A&M University

roliva@tamu.edu

Pål Davidsen

System Dynamics Group
University of Bergen

davidsen@ifi.uib.no

Christian Kampmann

Copenhagen Business School

cek.lpf@cbs.dk

Abstract

Previous work related to eigenvalue analysis in the system dynamics field has primarily focused on linking the model structure to the modes of behavior -- i.e. the eigenvalues. While the system eigenvalues define the *characteristics* of the system's behavior modes (e.g., exponential growth, expanding oscillations), these behavior modes are not equally represented in all model variables, making it difficult to link the behavior of the behavior mode to the behavior of a particular variable. In this study we propose an alternative perspective and explicitly explore the *significance* that each behavior mode has on the system state variables. We achieve this by decomposing the behavior of a variable into a sum of the *weighted* behavior modes in the system (represented by the eigenvalues). We argue that focusing on the weights, rather than on the eigenvalues, is a more efficient way to develop policy recommendations and compute the elasticity of the weights to the gain on any link the model allowing for a more efficient and discriminate way to identify policies. A routine to estimate the weights of any linear model and compute the elasticity of those weights to model links is developed and made available at http://iops.tamu.edu/faculty/roliva/research/sd/.

Keywords: linear model analysis, eigenvalue analysis**,** leverage points.

Introduction

Previous work related to eigenvalue analysis in system dynamics (e.g., Abdel-Gawad *et al.* 2005; Forrester 1982; Gunerlap 2005; Kampmann 1996; Kampmann and Oliva 2006; Saleh and Davidsen 2001a, b) has focused on linking model structure to the system modes of behavior, expressed in the eigenvalues of the linearized model. It has proven elusive, however, to use these tools for policy design. Major obstacles for the broad adoption of these approaches have been the computational intensity required to perform the analysis and the difficulties in interpreting the results. While the eigenvalues define the *characteristics* of the system's behavior modes (e.g., exponential growth, exponential decay, expanding oscillations, dampened oscillations), these behavior modes are not equally manifested in the time path of a particular model variable, making it difficult to link the eigenvalue analysis to the observed simulated behavior (Kampmann and Oliva 2006).

In this study we propose an alternative perspective to by exploring the *significance* that each behavior mode has on the system state variables. We achieve this by decomposing the behavior of a given variable into the *weighted* sum of the behavior modes in the system (represented by the eigenvalues). Though all variables are driven by the same set of eigenvalues, each state variable has a different set of weights for these and thus show different behavior patterns. We argue that focusing on these weights, rather than on the eigenvalues, is a more efficient way to develop policy recommendations.

This paper is a proof of concept as opposed to a fully implemented and tested algorithm. In the following section we explain our basic notation (full mathematical development of the idea is in Appendix A) and we then proceed to illustrate the benefits of the analysis with a well-known model. The paper concludes with a summary of next steps necessary to operationalize the approach in a way that would be amenable to SD practitioners.

Decomposing Behavior

In Appendix A we show how the behavior of any state variable in a linear system can be decomposed into several modes of behavior – each characterized by an eigenvalue. (We confine ourselves in this paper to linear systems only, leaving consideration of how to analyze nonlinear systems for later work.) That is, the time trajectory of state variable i can be expressed as

$$x_i(t) = w_{i1}m_1(t) + \ldots + w_{ij}m_j(t) + \ldots + w_{in}m_n(t) + u_i; \qquad (1)$$

where $x_i(t)$ is the value of state variable i at time t; w_{ij} is a constant term representing the significance of mode j to state variable i, i.e. the weight of mode j on variable i; $m_j(t)$ is the value of the j^{th} mode of behavior at time t; and u_i is a constant term. The modes of behavior of a linear system are a function of the eigenvalues λ of the Jacobian matrix that characterizes the system (Ogata 1990).

$$m_j = \begin{cases} \exp\left(\mathrm{Re}[\lambda_j]t\right) & \text{if } \mathrm{Im}[\lambda_j] = 0 \\ \exp\left(\mathrm{Re}[\lambda_j]t\right)\sin\left(\mathrm{Im}[\lambda_j]t + \theta\right) & \text{otherwise} \end{cases} \qquad (2)$$

If the eigenvalue does not have an imaginary part the behavior mode is expressed by the first equation and is characterized by exponential growth (if the real part of the eigenvalue is positive) or decay (if the real part of the eigenvalue is negative). If an eigenvalue has an imaginary part different than zero, it means that two eigenvalues are a conjugated pair (with the same real part) and together they generate the oscillatory mode represented by the second expression. If the real part of the conjugate pair of eigenvalue is positive, it yields to an expanding oscillation mode; if it is equal zero, yields to a sustained oscillation mode; and if it is negative, this yields to a damped oscillation mode.

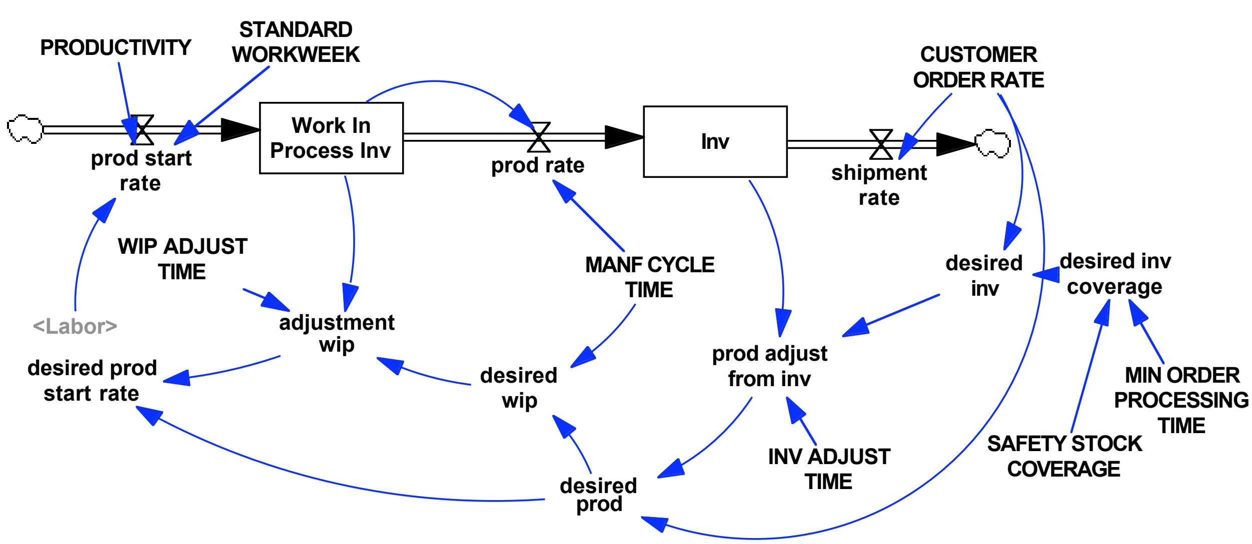
Decomposing the time trajectory of a state variable into its modes of behavior allows for a useful set of diagnosis, not only to understand the sources of the variable behavior, but also to identify the degree of interaction between different variables in the system. Furthermore, the significance of a behavior mode in a variable's behavior (w_{ij}) can also be used as a way to identify the elements of model structure most responsible for the observed behavior. We accomplish this by assessing the sensitivity of model weights to changes in the model's link gains. The gain of the link between two variables is defined as the partial derivative of the output variable with respect to the input variable ($g_{ab} = \partial a / \partial b$) and we define the elasticity of a weight to a gain (or elasticity of a weight to a link) as the ratio of the fractional change in the weight to the fractional change in the gain, i.e.,

$$\varepsilon = \frac{\delta w_{ij} / w_{ij}}{\delta g_{ab} / g_{ab}}. \tag{3}$$

A routine to estimate the weight vector for all state variables in a linear model and compute the elasticity of those weights to model links was developed in *Mathematica,* and is available at: http://iops.tamu.edu/faculty/roliva/research/sd/. The routine takes as input a model representation created by the Vensim to *Mathematica* Utility developed by Kampmann and Oliva (2006) (also available at http://iops.tamu.edu/faculty/roliva/research/sd/) so any linear model represented in Vensim can be immediately analyzed. The routine first decomposes the behavior of the states variables by calculating the system's eigenvalues λ_j and the weights w_{ij} of the eigenvalues in each state variable's behavior. Plots of the decomposed behavior of the base case are automatically generated. Then, for each link in the model, the routine modifies the gain of the link $g^*_{ab} = (1+\delta) g_{ab}$ and recalculates eigenvalues λ^*_j and weights w^*_{ij}. The elasticity of the weights to the link gains is estimating by comparing the new calculated values to the base case as per equation 3, where $\delta w_{ij} = w^*_{ij} - w_{ij}$ and $\delta g_{ab} = g^*_{ab} - g_{ab}$. In the next session we illustrate the use of this analysis with a simplified version of a well-known system dynamics model. We first decompose the behavior of the state variables, interpret the significance of the results, and then illustrate how an evaluation of the weight elasticity to link gains could be used for policy analysis. All output shown was obtained directly from the developed routines.

Example

To illustrate the above concepts, we apply them to a simple linear model; a simplified version of the labor-inventory model described in chapter 19, in Sterman (2000). Sterman uses this model to make the argument that are interactions between inventory management policies and labor adjustments cause a dampened oscillation with frequency and amplitude similar to the business cycle. The stock and flow diagram of the simplified linear model is portrayed in Figure 1 and model equations are listed in Appendix B. A copy of the model file, and its translation into Mathematica to be processed by the developed routines is available at http://iops.tamu.edu/faculty/roliva/research/sd/.



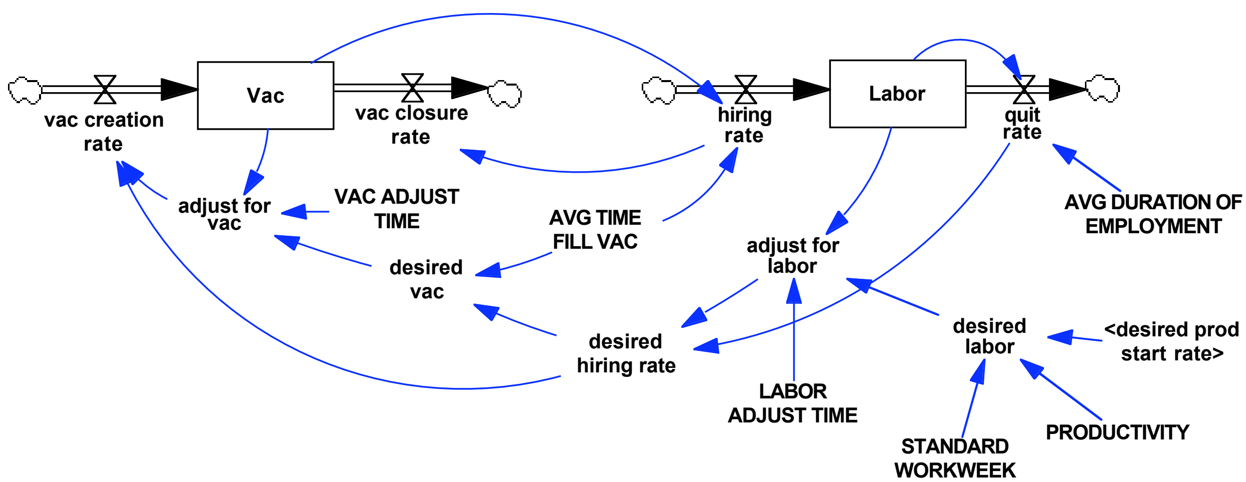


Figure 1. Stock & flow diagram of the model

The model consists of two sectors, the production and inventory sector, coupled through the variables desired production and labor. Labor explicitly controls production in the model. The model contains four state variables: 1) inventory, 2) labor, 3) vacancies, and 4) work in process inventory (WIP). The behavior of these state variables is illustrated in Figure 2.

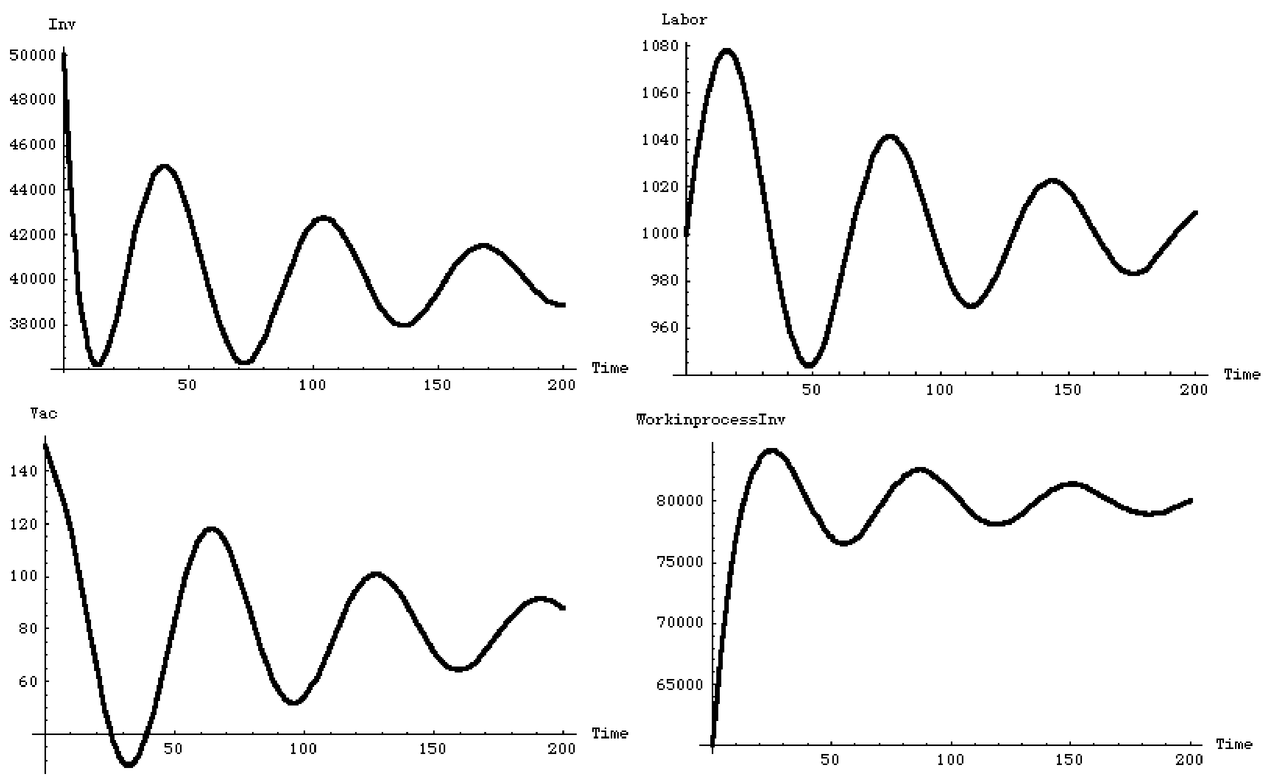


Figure 2. State variable behavior

Behavior Decomposition Weights (BDW)

As outlined in Appendix A, it is possible to decompose the behavior of each state variable into three modes of behaviors: two exponential decay modes and a damped oscillation mode, presented in Table 1. The decomposition and the associated weights are presented in Table 2. The table also shows the weights normalized by the constant term as well as the phase lag expressed in degrees.

Mode no. *i*	**Unit**	l_1	l_2	l_3
Real part Re{*l*}	1/week	–0.353	–0.138	–0.009
Imaginary part Im{ *l*}	1/week	0	0	0.098
Exp. adj. time *t*	weeks	2.83	7.25	105.7
Oscillation period *T*	weeks	–	–	63.6

Table 1. Behavior modes λ and the corresponding exponential adjustment times $\tau = |1/\mathrm{Re}\{\lambda\}|$ and periods of oscillation $T = 2\pi/\mathrm{Im}\{\lambda\}$, respectively.

The first two modes are exponential adjustments that die out relatively quickly (a few weeks), while the third mode is a damped oscillation with a period of 63 that which takes much longer to die out (106 weeks).

Variable $x(t)$	***Inv***	***Labor***	***Vac***	***WIP***
Constant *u*	40,000	1,000	80	80,000
Weight w_1	–122.22	–7.87	21.61	345.24
Weight w_2	14,432	20.72	–21.22	–15,934
Weight w_3	7,384.1	–89.09	70.40	–5,861.3
Phase *q*	3.76	0.15	1.42	0.85
w_1/u	–0.003	–0.007	0.270	0.004
w_2/u	0.361	0.021	–0.265	–0.199
w_3/u	0.185	–0.089	0.880	–0.073

Table 2. Behavior decomposition weights for the four state variables

$$x(t) = u + w_1 \exp(\lambda_1 t) + w_2 \exp(\lambda_2 t) + w_3 \sin(\mathrm{Im}(\lambda_3) t + \theta)$$

The decomposition of the behavior of the state variables can be observed in Figure 3. In each panel of Figure 3, the four components of behavior (three behavior modes plus the steady-state constant) are been plotted with a thin line and the overall behavior of the variable –the addition of the four components– with a broader line. With the exception of vacancies, the component from behavior mode 1 is hardly visible in these plots. By scaling the decomposition equations by the steady-state constant we can assess the relative low impact of behavior mode 1 on the other three variables, cf. Table 2.

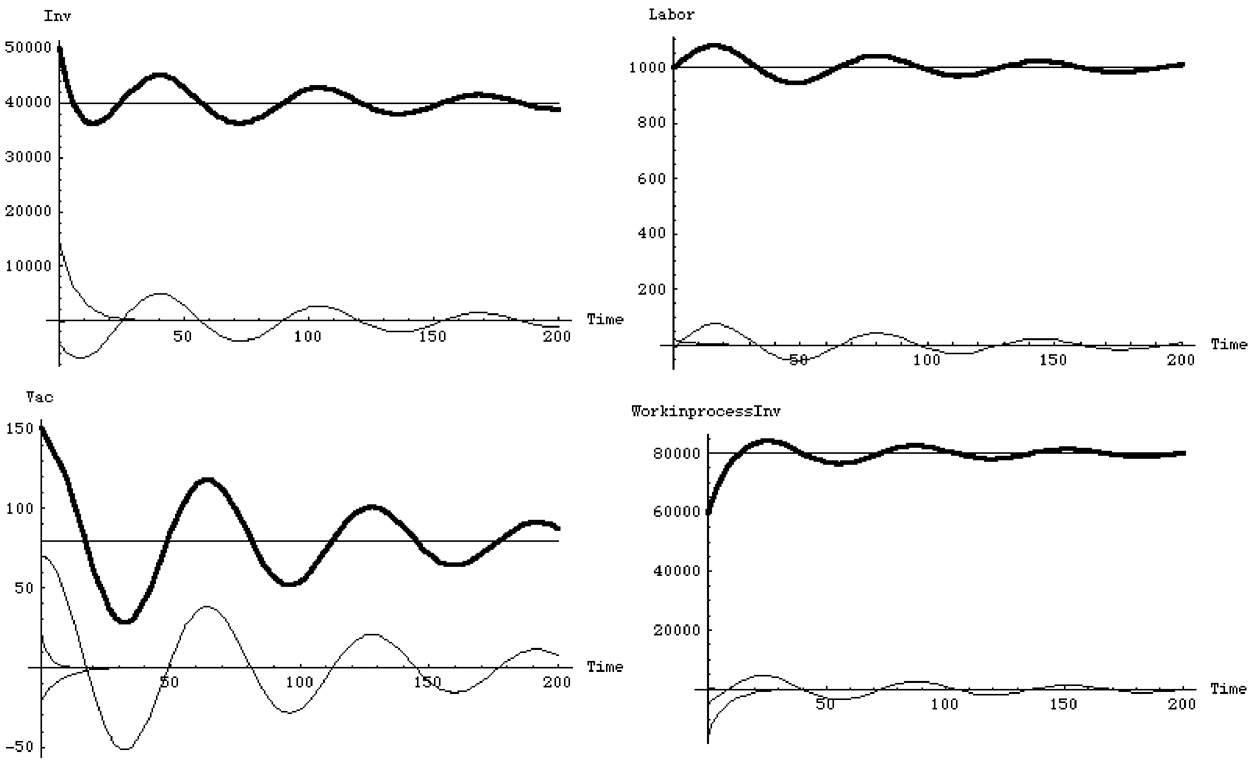


Figure 3. State variable behavior decomposition

From the scaled weights and the corresponding plots in Figure 3, it is relatively easy to perform a set of diagnostics. First, modes one and two represent rapid adjustments at the beginning of the simulation. Note that those two modes have little or no impact on those variables that started the simulation close to their steady-state value, e.g. labor. Relatively quickly, these modes die out and the variable behavior is controlled by the damped oscillation of mode three. Second, all variables are oscillating with the same frequency, corresponding to mode 3, i.e., with a period of around 63 weeks, but with significant lags between them; θ_i values (measured in radians) range from 0.15 to 3.76, representing a 206° phase lag between inventory and labor.

Focusing on inventory, for example, we can see that w_{12} (the weight of the second eigenvalue on Inventory) is much larger than the other two weights on inventory, and analysis of Figure 3 reveals that while significant, the second behavior mode is only active during the first 25 simulation periods. After period 25, the whole behavior of the inventory variable is controlled by the third behavior mode.

BDW Elasticity to Links

In this example, it might be of interest to study both the second and third modes of behavior. By focusing on the weights of the behavior modes for the variable of interest – rather than the eigenvalues – we can identify leverage points to increase or decrease the influence of a behavior mode in he variable. Table 3 lists the elasticities of weights into the Inventory (*Inv*) stock (w_1, w_2 and w_3) to the model's link gains–in this case sorted by the absolute value of the elasticity of w_2. The table does not report elasticities to links from constants to auxiliary variables since constants, being system parameters, might be used in more than one model equation. In this case, the analysis should focus on the sensitivity of the weights to those constants or parameter. We discuss this strategy in a separate section.

From Table 3, the highest leverage point for weight w_2 is the link from Work_In_Process_Inventory to Production_Rate. An increase of the gain of this link (by reducing the length of the manufacturing cycle time; Production_Rate = WIP/Manf_cycle_time) would significantly decrease w_2 reducing the effect of the second mode on the behavior of inventory. Reducing the manufacturing cycle time would make the production rate faster and make it more difficult for WIP to increase and inventory to decrease, to their steady state values. The interpretation of positive elasticities is similar. (A positive elasticity indicates that an increase in the numerical value of the link gain increases the numerical value of the weight, hence making it more prominent in the behavior regardless of its sign.) For example, an increase in the gain of the link from Work_In_Process_Inventory to Adjustment_For_WIP (fourth row) would significantly increase w_2; hence making the second mode of behavior more salient in the behavior of the inventory. The gain of this link can be increased by reducing the value of the WIP_adjustment_time constant in the model (Adjustement_For_WIP=(Desired_WIP-WIP)/WIP_adjustment_time). A reduction of the WIP_adjustment_time from its initial value of 6 weeks would mean that the model would be more aggressive in closing gaps to desired inventory and would allow for the

Input Variable	Output Variable	w_1	w_2	w_3
WorkinprocessInv	ProdRate	0.063	−4.316	−0.305
DesiredWIP	AdjustmentWIP	−22.724	−3.162	7.253
DesiredProd	DesiredWIP	−22.724	−3.162	7.253
WorkinprocessInv	AdjustmentWIP	16.407	3.147	−7.192
ShipmentRate	Inv	11.452	3.088	−1.982
ProdRate	Inv	−5.743	−2.934	3.255
DesiredProdStartRate	DesiredLabor	−23.327	−2.391	5.498
DesiredLabor	AdjustForLabor	−23.327	−2.391	5.498
DesiredProd	DesiredProdStartRate	−17.042	−2.372	5.441
Labor	AdjustForLabor	21.347	2.052	−5.721
DesiredInvCoverage	DesiredInv	−16.173	−1.705	4.559
DesiredInv	ProdAdjustfromInv	−16.173	−1.705	4.559
ProdStartRate	WorkinprocessInv	−8.212	1.525	2.497
Labor	ProdStartRate	−8.212	1.525	2.497
ProdRate	WorkinprocessInv	5.802	−1.382	−3.470
Inv	ProdAdjustfromInv	24.946	1.279	−5.530
DesiredVac	AdjustForVac	−4.042	−0.488	0.581
DesiredHiringRate	DesiredVac	−4.042	−0.488	0.581
HiringRate	Labor	0.913	−0.444	0.864
Vac	AdjustForVac	1.042	0.425	−0.505
ProdAdjustfromInv	DesiredProd	8.766	−0.423	−0.976
QuitRate	DesiredHiringRate	−4.052	−0.390	1.087
AdjustForLabor	DesiredHiringRate	−2.011	−0.342	−0.214
VacCreationRate	Vac	−5.002	−0.305	0.368
DesiredHiringRate	VacCreationRate	−2.021	−0.244	0.291
QuitRate	Labor	0.266	0.243	−1.074
Vac	HiringRate	1.437	−0.231	0.613
VacClosureRate	Vac	0.523	0.213	−0.252
HiringRate	VacClosureRate	0.523	0.213	−0.252
Labor	QuitRate	−3.786	−0.147	0.013
AdjustForVac	VacCreationRate	−2.987	−0.062	0.077
AdjustmentWIP	DesiredProdStartRate	−6.294	−0.020	0.060

Table 3. Weight elasticities to links associated with Inventory stock – sorted by w_2

WIP stock to reach its steady-state value much faster. Note that while the inventory adjustment time (the parameter governing the gain on links into Prod_Adjust_form_Inv, rows 12 and 16 in Table 3) is twice as long as the WIP adjustment time, the gap between initial inventory and its steady-state value is much smaller than the gap between the initial WIP and its steady state value. The analysis of weight elasticity to links, assuming that individual links can be independently adjusted, identifies this larger gap and points to WIP_adjustment_time as having bigger leverage on the weight of the second mode of behavior on inventory. In the next section we provide an analysis of weight elasticity to parameters that provides a more realistic identification of leverage points.

Further scrutiny of the table also points to the unique leverage points for each weight. Through cross comparisons of the elasticity tables it is possible to identify those links that have the highest leverage to the weight of interest, but that that have relative low influence on the weights of other modes on other state variables. Consider, for example, the Work_In_Process_Inventory to Production_Rate link, the relatively low elasticity of the w_1 and w_3 weight to that link suggest that the link is particularly effective in solely affecting the effect of the second behavior mode on Inventory.

To facilitate this analysis, Table 4 lists the same results as Table 3 but now sorted by the absolute value of the elasticity of w_3. From table 4, we can see that an increase in the gain of the link from Desired_Production to Desired_WIP (second row) would significantly increase w_3; hence making the third mode of behavior more salient. By increasing the manufacturing cycle time (Desired_WIP = Desired_production*Manf_cycle_time) one strengthens the multiplier effect that desired WIP coverage has, thus further amplifying the desired production rate and increasing the amplitude of the model's oscillating mode—the Bullwhip effect experienced in the Beer Distribution Game.

Input Variable	Output Variable	w_1	w_2	w_3
DesiredWIP	AdjustmentWIP	−22.724	−3.162	7.253
DesiredProd	DesiredWIP	−22.724	−3.162	7.253
WorkinprocessInv	AdjustmentWIP	16.407	3.147	−7.192
Labor	AdjustForLabor	21.347	2.052	−5.721
Inv	ProdAdjustfromInv	24.946	1.279	−5.530
DesiredProdStartRate	DesiredLabor	−23.327	−2.391	5.498
DesiredLabor	AdjustForLabor	−23.327	−2.391	5.498
DesiredProd	DesiredProdStartRate	−17.042	−2.372	5.441
DesiredInvCoverage	DesiredInv	−16.173	−1.705	4.559
DesiredInv	ProdAdjustfromInv	−16.173	−1.705	4.559
ProdRate	WorkinprocessInv	5.802	−1.382	−3.470
ProdRate	Inv	−5.743	−2.934	3.255
ProdStartRate	WorkinprocessInv	−8.212	1.525	2.497
Labor	ProdStartRate	−8.212	1.525	2.497
ShipmentRate	Inv	11.452	3.088	−1.982
QuitRate	DesiredHiringRate	−4.052	−0.390	1.087
QuitRate	Labor	0.266	0.243	−1.074
ProdAdjustfromInv	DesiredProd	8.766	−0.423	−0.976
HiringRate	Labor	0.913	−0.444	0.864
Vac	HiringRate	1.437	−0.231	0.613
DesiredVac	AdjustForVac	−4.042	−0.488	0.581
DesiredHiringRate	DesiredVac	−4.042	−0.488	0.581
Vac	AdjustForVac	1.042	0.425	−0.505
VacCreationRate	Vac	−5.002	−0.305	0.368
WorkinprocessInv	ProdRate	0.063	−4.316	−0.305
DesiredHiringRate	VacCreationRate	−2.021	−0.244	0.291
VacClosureRate	Vac	0.523	0.213	−0.252
HiringRate	VacClosureRate	0.523	0.213	−0.252
AdjustForLabor	DesiredHiringRate	−2.011	−0.342	−0.214
AdjustForVac	VacCreationRate	−2.987	−0.062	0.077
AdjustmentWIP	DesiredProdStartRate	−6.294	−0.020	0.063
Labor	QuitRate	−3.786	−0.147	0.010

Table 4: Weight elasticities to links associated with Inventory stock – sorted by w_3

BDW Elasticity to Parameters

An alternative exploration of the policy design space is achieved assessing the weight elasticity to model parameters. While changes to model parameters might not have the ability to identify a unique leverage points for a particular weight on a stock as the link elasticities do, parameters reflect policies and various "physical" realities in the system and as such represent more intuitive intervention points. Furthermore, assessing weight elasticities to parameter changes allows for a more realistic assessment of the policy design space since in most instances changes to link gains would be implemented through parameter changes, and parameters might impact several links simultaneously. Our algorithms also support the calculation of weight elasticity to parameters and these can be reported either by mode (assessing the impact across different stocks) or by stock (comparing the impact of the parameter changes across behavior modes). Tables 5 and 6 report these two modes of output for $w_{2,}$ again in the case of the Inventory variable (*Inv*).

Analysis and interpretation of this output is similar to the one performed for the tables with weight elasticities to links (tables 1 and 2), but in the interest of brevity are not performed here. Note that the weight elasticity to a parameter cannot be estimated directly by adding the weight elasticities of the links into the variables affected by it. For example, Table 3 reports a weight elasticity to Inv_Adjust_Time of 0.850, while the sum of the weight elasticities to the links into Production_Adjust_from_Inventory, the only variable affected by Inv_Adjust_Time, in Table 1 is –0.426 (–1.705+1.279). This is due to the fact that parameters enter in different algebraic forms, not just additive terms, in the various equations.

Parameter	Inv	Labor	Vac	WIP
InvAdjustTime	0.850	-6.289	-6.948	0.235
WIPAdjustTime	-0.467	2.208	2.457	-0.236
ManfCycleTime	0.323	4.190	3.562	0.742
StandardWorkWeek	-0.170	-1.169	-1.169	-0.170
Productivity	-0.170	-1.169	-1.169	-0.170
VacAdjustTime	0.102	0.417	0.446	0.129
LaborAdjustTime	0.037	-0.872	-0.957	-0.041
AvgTimeFillVac	-0.009	0.148	1.163	0.005
AvgDurationofEmployment	-0.007	0.030	0.112	-0.003
SafetyStockCoverage	0.000	0.000	0.000	0.000
MinOrderProcessingTime	0.000	0.000	0.000	0.000
CustomerOrderRate	0.000	0.000	0.000	0.000

Table 5: Weight elasticities to parameters associated to 2nd behavior mode – sorted by *Inv*

Parameter	1	2	3
InvAdjustTime	–4.712	0.850	–0.162
WIPAdjustTime	1.677	–0.467	1.240
ManfCycleTime	0.730	0.323	–0.169
StandardWorkWeek	0.367	–0.170	0.811
Productivity	0.367	–0.170	0.811
VacAdjustTime	3.375	0.102	–0.184
LaborAdjustTime	–0.877	0.037	1.028
AvgTimeFillVac	1.117	–0.009	0.068
AvgDurationofEmployment	0.066	–0.007	–0.009
SafetyStockCoverage	0.000	0.000	0.000
MinOrderProcessingTime	0.000	0.000	0.000
CustomerOrderRate	0.000	0.000	0.000

Table 6: Weight elasticities Parameter associated with Inventory stock – sorted by w_2

Next Steps

An obvious next step for the development of this tool is the application of the method to nonlinear models typically developed in system dynamics. The linearization of an SD model around an instantaneous model operating point has proven an effective strategy to use insights and tools from linear dynamics in SD models (Abdel-Gawad *et al.* 2005; Forrester 1982; Gunerlap 2005; Kampmann 1996; Kampmann and Oliva 2006; Saleh and Davidsen 2001a, b). We are optimistic about our ability to operationalize the computations and insights form this analysis into a tool that can be reliably used by the SD community. These explorations are left as further developments for this line of research.

Another issue outstanding is that the weights are partly a function of the initial conditions of the system. One may say that certain initial conditions may excite particular behavior modes more than others. It would be useful, therefore, to separate the effect of changing initial conditions on the weights from the part that comes more from structural features.

We end with a word of caution. While changes in parameters would impact the weight matrix as described in the previous section, it should be noted that these changes would also change the eigenvalues themselves. That is, once parameter changes are made, the explanation of the impact of parameter changes on the behavior of inventory above suggest, the Jacobian of the system and consequently its eigenvalues also change. Changing the time it takes stocks to reach their steady state values is equivalent to stating a change on the real part of the eigenvalue controlling that behavior, regardless of the weight applied to it. The fact that weights and eigenvalues are not independently determined is perhaps one of the biggest shortcomings of this proposed method of analysis. The method, however, is effective in identifying leverage points for intervention in the models behavior and as such a promising tool for policy design.

References

Abdel-Gawad A, Abdel-Aleem B, Saleh M, Davidsen P. 2005. Identifying dominant behavior patterns, links and loops: Automated eigenvalue analysis of system dynamics models. *Proceedings of the 2005 Int. System Dynamics Conference*. Boston.

Edwards C, Penny D. 2005. *Differential equations and linear algebra*. 2nd ed. Pearson Education: Upper Saddle River, NJ.

Forrester N. 1982. A Dynamic Synthesis of Basic Macroeconomic Policy: Implications for Stabilization Policy Analysis. PhD Thesis, Sloan School of Management, Mass. Inst. of Technology, Cambridge, MA.

Gunerlap B. 2005. Towards Coherent Loop Dominance Analysis: Progress in Eigenvalue Elasticity Analysis. *Proceedings of the 2005 Int. System Dynamics Conference*. Boston.

Kampmann CE. 1996. Feedback Loop Gains and System Behavior (unpublished manuscript). Summarized in *Proceedings of 1996 Int. System Dynamics Conference*. Cambridge, MA.

Kampmann CE, Oliva R. 2006. Loop Eigenvalue Elasticity Analysis: Three Case Studies. *System Dynamics Review* **22**(2): 146-162.

Luenberg D. 1979. *Introduction to dynamic systems: Theory, models and applications*. Wiley: New York.

Ogata K. 1990. *Modern Control Engineering*. 2nd ed. Prentice Hall: Englewood Cliffs, NJ.

Saleh M, Davidsen P. 2001a. The origins of behavior patterns. *Proceedings of the 2001 Int. System Dynamics Conference*. Atlanta.

Saleh M, Davidsen P. 2001b. The origins of business cycles. *Proceedings of the 2001 Int. System Dynamics Conference*. Atlanta.

Sterman JD. 2000. *Business dynamics: Systems thinking and modeling for a complex world*. Irwin McGraw-Hill: Boston.

Appendix A: Eigenvalue Analysis of Linear Models

Mathematical Background

In this appendix, we will decompose the time trajectory of a state variable into several modes of behavior. The time trajectory of a state variable is a mathematical function that specifies the value of the state variable at any time instant. The point of departure is the structure of the model, which, in the case of linear model can be represented by the following compact matrix equation (we denote scalars using lower case letters; vectors in bold lower case letters; and matrices in bold capital letters):

$$\dot{\mathbf{x}}(t) = \mathbf{G}\mathbf{x}(t) + \mathbf{b} \tag{A.1}$$

where **x** is the vector of state variables; $\dot{\mathbf{x}}$ is the vector of first time derivates of state variables (rates), **b** is a constant vector; and **G** is the Jacobian or gain matrix $\left(\mathbf{G_{ij}} = \partial \dot{x}_i / \partial x_j\right)$. In linear systems, **G** is constant unlike nonlinear systems, where it is a function of the state variables and exogenous inputs and consequently varies over time. **b** is likewise constant in a linear model with zero or constant exogenous variables, unlike the case in nonlinear systems.

Differentiating equation A.1 with respect to time yields

$$\ddot{\mathbf{x}}(t) = \mathbf{G}\dot{\mathbf{x}}(t) \tag{A.2}$$

where $\ddot{\mathbf{x}}$ is the curvature vector –the vector of second time derivates of state variables. Thus, the gain matrix **G** relates the slope vector to the curvature vector in an n dimensional standard space R^n.

The solution to the system of differential equations specified by equation A.2 provides us with the time trajectory of the slope vector of the model. We use the eigenvalue method for solving differential equations (Luenberg 1979) to solve for the time trajectory of the slope.

The n eigenvalues and their associated right eigenvectors of the gain matrix **G** are defined as $\mathbf{Gr_k} = \lambda_k \mathbf{r_k}$. The default case is to have n distinct eigenvalues, and consequently the right eigenvectors will be linearly independent (Luenberg 1979), and span the n-dimensional space, R^n. Consequently, the slope vector can be expressed as a linear combination of the right eigenvectors,

$$\dot{\mathbf{x}}(t) = \alpha_1(t)\mathbf{r}_1 + \alpha_2(t)\mathbf{r}_2 + \mathbf{...} + \alpha_n(t)\mathbf{r}_n, \tag{A.3}$$

where α_k are the components of the slope vector in the new coordinate system and $\mathbf{r}_i$ are the constant set of eigenvectors. Differentiating equation (A.3) with respect to time yields the components $\dot{\alpha}_k$ of the curvature vector in the new coordinate system

$$\ddot{\mathbf{x}}(t) = \dot{\alpha}_1(t)\mathbf{r}_1 + \dot{\alpha}_2(t)\mathbf{r}_2 + \mathbf{...} + \dot{\alpha}_n(t)\mathbf{r}_n. \tag{A.4}$$

Substituting equation A.3 into equation A.2 yields

$$\ddot{\mathbf{x}}(t) = \mathbf{G}\left[\alpha_1(t)\mathbf{r}_1 + \alpha_2(t)\mathbf{r}_2 + \mathbf{...} + \alpha_n(t)\mathbf{r}_n\right].$$

Rearranging,

$$\ddot{\mathbf{x}}(t) = \alpha_1(t)\mathbf{Gr}_1 + \alpha_2(t)\mathbf{Gr}_2 + \mathbf{...} + \alpha_n(t)\mathbf{Gr}_n, \tag{A.5}$$

and replacing the definition of eigenvalues $\mathbf{Gr_k} = \lambda_k \mathbf{r_k}$, we obtain:

$$\ddot{\mathbf{x}}(t) = \alpha_1(t)\lambda_1\mathbf{r}_1 + \alpha_2(t)\lambda_2\mathbf{r}_2 + \mathbf{...} + \alpha_n(t)\lambda_n\mathbf{r}_n. \tag{A.6}$$

Equating A.4 and A.6 we obtain a differential equation that describe the dynamics that take place along the coordinate specified be the right eigenvector $\mathbf{r_k}$:

$$\dot{\alpha}_k(t) = \lambda_k \alpha_k(t)$$

The solution of the above differential equation is

$$\alpha_k(t) = \alpha_k^0 e^{\lambda_k(t-\tau)}$$

where τ is the initial time, i.e. the starting time of the analysis period, and α_k^0 is the corresponding initial value of α_k.

It is clear that the only factor determining the dynamics along a particular coordinate –i.e., a right eigenvector– is the eigenvalue associated with that coordinate itself. Substituting the solution of the dynamic behavior of each alpha α_k, into equation A.3 yields the time trajectory of the slope vector along the dimensions of the eigen-space

$$\dot{\mathbf{x}}(t) = \alpha_1^0 e^{\lambda_1(t-\tau)}\mathbf{r_1} + \alpha_{21}^0 e^{\lambda_2(t-\tau)}\mathbf{r}_2 + \ldots + \alpha_n^0 e^{\lambda_n(t-\tau)}\mathbf{r}_n. \tag{A.7}$$

Integrating the above slope trajectory equation with respect to time (from time = τ to time = t), yields

$$\mathbf{x}(t) = \left(\alpha_1^0/\lambda_1\right)\left(e^{\lambda_1(t-\tau)} - 1\right)\mathbf{r_1} + \left(\alpha_2^0/\lambda_2\right)\left(e^{\lambda_2(t-\tau)} - 1\right)\mathbf{r}_2 + \ldots + \left(\alpha_n^0/\lambda_n\right)\left(e^{\lambda_n(t-\tau)} - 1\right)\mathbf{r}_n + \mathbf{x_0},$$

where $\mathbf{x_0}$ is a constant vector representing the initial values of the state variables. Expanding the above equation,

$$\mathbf{x} = \left(\alpha_1^0/\lambda_1\right)e^{\lambda_1(t-\tau)}\mathbf{r_1} + \ldots + \left(\alpha_n^0/\lambda_n\right)e^{\lambda_n(t-\tau)}\mathbf{r}_n - \left(\alpha_1^0/\lambda_1\right)\mathbf{r_1} - \ldots - \left(\alpha_n^0/\lambda_n\right)\mathbf{r}_n + \mathbf{x_0},$$

defining $\mathbf{w_k} = \left(\alpha_k^0/\lambda_k\right)\mathbf{r_k}$ and $\mathbf{u} = -\left(\alpha_1^0/\lambda_1\right)\mathbf{r_1} - \ldots - \left(\alpha_n^0/\lambda_n\right)\mathbf{r}_n + \mathbf{x_0}$ we obtain an expression

$$\mathbf{x}(t) = \mathbf{w_1}e^{\lambda_1(t-\tau)} + \ldots + \mathbf{w_n}e^{\lambda_n(t-\tau)} + \mathbf{u} \tag{A.8}$$

that decomposes the state trajectory into several modes of behavior, each characterized by an eigenvalue. Below we discuss special cases.

Conjugated eigenvalues

In case of conjugate eigenvalues, their weights will also be conjugated, and they will combine into an oscillating behavior mode. Assume, without loss of generality, that $\tau = 0$ and consider a pair of conjugated eigenvalues in equation A.8.

$$x_i(t) = w_{i1}e^{\lambda_1 t} + w_{i2}e^{\lambda_2 t} + u_i,$$

where $w_{i1} = a + ib$, $w_{i2} = a - ib$, $\lambda_1 = c + id$, and $\lambda_2 = c - id$, and, consequently, $u_i = -2a$, i.e.,

$$x_i(t) = e^{ct}(a+ib)e^{idt} + e^{ct}(a-ib)e^{-idt} - 2a. \tag{A.9}$$

Using Euler's formula in complex analysis $e^{i\zeta} = \cos\zeta + i\sin\zeta$, A.9 becomes

$$x_i(t) = e^{ct}(a+ib)\left[\cos(dt) + i\sin(dt)\right] + e^{ct}(a-ib)\left[\cos(-dt) + i\sin(-dt)\right] - 2a$$

that simplifies to

$$x_i(t) = 2e^{ct}\left[a\cos(dt) - b\sin(dt)\right] - 2a.$$

Taking $\sqrt{a^2+b^2}$ as a common factor on the first term yields

$$x_i(t) = 2\sqrt{a^2+b^2}e^{ct}\left[\frac{a}{\sqrt{a^2+b^2}}\cos(dt) + \frac{-b}{\sqrt{a^2+b^2}}\sin(dt)\right] - 2a,$$

defining an angle θ such that $\sin(\theta) = a/\sqrt{a^2+b^2}$ and $\cos(\theta) = -b/\sqrt{a^2+b^2}$, i.e., $\theta = \operatorname{atan}(a/-b)$, and replacing

$$x_i(t) = 2\sqrt{a^2+b^2}e^{ct}\left[\sin(\theta)\cos(dt) + \cos(\theta)\sin(dt)\right] - 2a,$$

and simplifying

$$x_i(t) = 2\sqrt{a^2+b^2}e^{ct}\sin(dt+\theta) - 2a,$$

we reduce the conjugate pair to a single oscillatory behavior mode $\left(e^{ct}\sin(dt+\theta)\right)$, and identified the weight of that behavior mode to x_i $\left(2\sqrt{a^2+b^2}\right)$ and its contribution to the **u** vector (-2*a*).

Zero eigenvalue

Assume a zero eigenvalue in the slope trajectory equation A.7

$$\dot{\mathbf{x}}(t) = \alpha_1^0 e^{0(t-\tau)}\mathbf{r_1} = \alpha_1^0\mathbf{r_1}.$$

Integrating the above equation (from time = τ to time = t) yields

$$\mathbf{x} = (t-\tau)\alpha_1^0\mathbf{r_1}\ .$$

The above equation represents a linear mode of behavior. The weight vector associated with this linear mode is equal to $\alpha_1^0 \mathbf{r}_1$, and this mode does not contribute at all to the **u** vector since at the starting time of the analysis period this linear mode equals zero.

Incomplete set of right-eigenvectors

The term “complete right-eigenvectors” means that the right-eigenvectors span the whole n-dimensional space (the default case developed above). Incomplete set of right-eigenvectors can only occur where there are non-distinct (repeated) eigenvalues but non-distinct eigenvalues is not a sufficient condition (Edwards and Penny 2005).

An eigenvalue is of multiplicity *k* if it is repeated *k* times—i.e. if it is a *k*-fold root of the equation Det($\mathbf{G} - \lambda\, \mathbf{I}$) = 0. An eigenvalue of multiplicity *k* is said to be complete if it has *k* linearly independent associated right-eigenvectors. If every eigenvalue of the Jacobian matrix is complete then, since right-eigenvectors associated with different eigenvalues are always linearly independent (Edwards and Penny 2005), it follows that **G** does have a complete set of *n* linearly independent right eigenvectors. An eigenvalue λ of multiplicity $k > 1$ is called defective if it is not complete and has only *p* linearly independent right-eigenvectors, where $0<p<k$. The number *d* is called the defect of the defective eigenvalue λ and denotes the number of "missing" independent right-eigenvectors ($d=k-p$).

Since any *n*x*n* matrix **G** has *n* linearly independent right-eigenvectors (Edwards and Penny 2005, p. 449), in an incomplete case, i.e. if a defective eigenvalue exits), it is necessary to compute *d* "generalized" right-eigenvectors linearly independent from each other and the original *p* right-eigenvectors.

Without loss of generality, assume that matrix **G** has a single eigenvalue λ with multiplicity n ($k=n$), and that there is only one linearly independent right-eigenvector ($\mathbf{r}_1$) associated with λ ($p=1$), thus the defect of λ is $d=n-1$.

The initial right-eigenvector $\mathbf{r}_1$ can be computed from the definition of eigenvalue $\mathbf{G}\mathbf{r}_1 = \lambda\mathbf{r}_1$ and the d generalized right-eigenvectors, $\mathbf{r}_2 \ldots \mathbf{r}_n$, which are based on $\mathbf{r}_1$, can be computed from the recursive equation $\mathbf{G}\mathbf{r}_\mathbf{i} = \lambda\mathbf{r}_\mathbf{i} + \mathbf{r}_{\mathbf{i-1}}$ (Edwards and Penny 2005), thus:

$$\begin{gathered}\mathbf{G}\mathbf{r}_2 = \lambda\mathbf{r}_2 + \mathbf{r}_1 \\ \mathbf{G}\mathbf{r}_3 = \lambda\mathbf{r}_3 + \mathbf{r}_2 \\ \ldots \\ \mathbf{G}\mathbf{r}_n = \lambda\mathbf{r}_n + \mathbf{r}_{n-1}\end{gathered}$$

Replacing these definitions of the generalized right-eigenvectors in equation A.5 and arranging yields

$$\ddot{\mathbf{x}}(t) = (\alpha_1(t)\lambda + \alpha_2(t))\,\mathbf{r}_1 + (\alpha_2(t)\lambda + \alpha_3(t))\,\mathbf{r}_2 + \ldots + (\alpha_j(t)\lambda + \alpha_{j+1}(t))\,\mathbf{r}_j + \ldots + (\alpha_n(t)\lambda)\,\mathbf{r}_n . \text{(A.10)}$$

Equating A.4 and A.10 we obtain the following differential equations

$$\begin{gathered}\dot{\alpha}_1(t) = \alpha_1(t)\lambda + \alpha_2(t) \\ \dot{\alpha}_2(t) = \alpha_2(t)\lambda + \alpha_3(t) \\ \ldots \\ \dot{\alpha}_n(t) = \alpha_n(t)\lambda\end{gathered}$$

and, starting from the last differential equation, one can recursively solve all differential equations obtaining

$$\alpha_n(t) = \alpha_n^0 e^{\lambda t}$$

$$\alpha_{n-1}(t) = \left\{ \alpha_{n-1}^0 + \alpha_n^0 \frac{t}{1!} \right\} e^{\lambda t}$$

$$\alpha_{n-2}(t) = \left\{ \alpha_{n-2}^0 + \alpha_{n-1}^0 \frac{t}{1!} + \alpha_n^0 \frac{t^2}{2!} \right\} e^{\lambda t}$$

$$\ldots$$

$$\alpha_j(t) = \left\{ \alpha_j^0 + \alpha_{j+1}^0 \frac{t}{1!} + \ldots + \alpha_n^0 \frac{t^{n-j}}{(n-j)!} \right\} e^{\lambda t}$$

$$\ldots$$

$$\alpha_1(t) = \left\{ \alpha_1^0 + \alpha_2^0 \frac{t}{1!} + \alpha_3^0 \frac{t^2}{2!} + \ldots + \alpha_{n-1}^0 \frac{t^{n-2}}{(n-2)!} + \alpha_n^0 \frac{t^{n-1}}{(n-1)!} \right\} e^{\lambda t}$$

Substituting the above equations into A.3 yields

$$\dot{\mathbf{x}}(t) = \mathbf{z}_1 e^{\lambda t} + \mathbf{z}_2 t\, e^{\lambda t} + \ldots + \mathbf{z}_j t^{j-1} e^{\lambda t} + \ldots + \mathbf{z}_n t^{n-1} e^{\lambda t}, \qquad \text{(A.11)}$$

where

$$\mathbf{z}_1 = \alpha_1^0 \mathbf{r}_1 + \alpha_2^0 \mathbf{r}_2 + \ldots + \alpha_n^0 \mathbf{r}_n$$

$$\mathbf{z}_2 = \alpha_2^0 \mathbf{r}_1 + \alpha_3^0 \mathbf{r}_2 + \ldots + \alpha_n^0 \mathbf{r}_{n-1}$$

$$\mathbf{z}_3 = \frac{\alpha_3^0}{2!}\mathbf{r}_1 + \frac{\alpha_4^0}{2!}\mathbf{r}_2 + \ldots + \frac{\alpha_n^0}{2!}\mathbf{r}_{n-2}$$

$$\ldots$$

$$\mathbf{z}_j = \frac{\alpha_j^0}{(j-1)!}\mathbf{r}_1 + \frac{\alpha_{j+1}^0}{(j-1)!}\mathbf{r}_2 + \ldots + \frac{\alpha_n^0}{(j-1)!}\mathbf{r}_{n-j+1}$$

$$\ldots$$

$$\mathbf{z}_n = \frac{\alpha_n^0}{(n-1)!}\mathbf{r}_1$$

Integrating A.11 yields

$$\mathbf{x}(t) = \mathbf{w}_1 e^{\lambda t} + \mathbf{w}_2 t\, e^{\lambda t} + \ldots + \mathbf{w}_j t^{j-1} e^{\lambda t} + \ldots + \mathbf{w}_n t^{n-1} e^{\lambda t} + \mathbf{u}$$

where

$$\mathbf{w}_1 = \frac{\mathbf{z}_1}{\lambda} - \frac{\mathbf{z}_2}{\lambda^2} + \ldots + \frac{(-1)^{i-1}\,(i-1)!\,\mathbf{z}_i}{0!\ \lambda^i} + \ldots + \frac{(-1)^{n-1}\,(n-1)!\,\mathbf{z}_n}{0!\,\lambda^n}$$

$$\mathbf{w}_2 = \frac{\mathbf{z}_2}{\lambda} - \frac{2\,\mathbf{z}_3}{\lambda^2} + \ldots + \frac{(-1)^{i-2}\,(i-1)!\,z_i}{1!\ \lambda^{i-1}} + \ldots + \frac{(-1)^{n-2}\,(n-1)!\,\mathbf{z}_n}{1!\ \lambda^{n-1}}$$

$$\ldots$$

$$\mathbf{w}_j = \frac{\mathbf{z}_j}{\lambda} - \frac{j\,\mathbf{z}_{j+1}}{\lambda^2} + \ldots + \frac{(-1)^{i-j}\,(i-1)!\,\mathbf{z}_i}{(j-1)!\ \lambda^{i+1-j}} + \ldots + \frac{(-1)^{n-j}(n-1)!\,\mathbf{z}_n}{(j-1)!\ \lambda^{n+1-j}}$$

$$\ldots$$

$$\mathbf{w}_n = \frac{\mathbf{z}_n}{\lambda}$$

$$\mathbf{u} = -\mathbf{w}_1 + \mathbf{x}_0$$

These values of $\mathbf{w}_i$ and $\mathbf{u}$ have the same interpretation as the values with the same labels in equation A.7, but in this case the $\mathbf{w}_i$ coefficients represent the weight of a hyper-exponential mode.

Appendix B: Model Equations

```
Independent_Variables:
init  Inventory = 50000
init  Labor = 1000
init  Vacancies = 150
init  Work_In_Process_Inventory = 60000

const Customer_Order_Rate = 10000
const Vacancy_Adjustment_Time = 4
const Productivity = .25
const WIP_Adjustment_Time = 6
const Inventory_Adjustment_Time = 12
const Manufacturing_Cycle_Time = 8
const Desired_Inventory = 40000
const Labor_Adjustment_Time = 19
const Average_Time_To_Fill_Vacancies = 8
const Average_Duration_Of_Employement = 100
const Standard_Workweek = 40

Dependent_Variables:
aux   Production_Adjustment_From_Inventory = (Desired_Inventory-Inventory)
                                             /Inventory_Adjustment_Time
aux   Desired_Production = Customer_Order_Rate+Production_Adjustment_From_Inventory
aux   Desired_WIP = Manufacturing_Cycle_Time*Desired_Production
aux   Adjustment_For_WIP = (Desired_WIP - Work_In_Process_Inventory)
                                             /WIP_Adjustment_Time
aux   Desired_Production_Start_rate = Desired_Production+Adjustment_For_WIP
aux   Desired_Labor = Desired_Production_Start_rate/(Productivity*Standard_Workweek)
aux   Adjustemnt_For_Labor = (Desired_Labor-Labor)/Labor_Adjustment_Time
aux   Quit_Rate = Labor/Average_Duration_Of_Employement
aux   Desired_Hiring_Rate = Quit_Rate+Adjustemnt_For_Labor
aux   Desired_Vacancies =  Desired_Hiring_Rate*Average_Time_To_Fill_Vacancies
aux   Adjustment_For_Vacancies = (Desired_Vacancies-Vacancies)
                                             /Vacancy_Adjustment_Time
aux   Vacancy_Creation_Rate = Adjustment_For_Vacancies+Desired_Hiring_Rate
aux   Hiring_Rate = Vacancies/Average_Time_To_Fill_Vacancies
aux   Vacancy_Closure_Rate = Hiring_Rate
aux   Production_Start_Rate = Labor*Standard_Workweek*Productivity
aux   Production_Rate = Work_In_Process_Inventory/Manufacturing_Cycle_Time
aux   Shipment_Rate = Customer_Order_Rate

Flows:
flow  Work_In_Process_Inventory = Production_Start_Rate  - Production_Rate
flow  Inventory = Production_Rate  - Shipment_Rate
flow  Vacancies = Vacancy_Creation_Rate - Vacancy_Closure_Rate
flow  Labor = Hiring_Rate - Quit_Rate
```