System Dynamics Model to Understand Demand Conditioning Dynamics in Supply Chains

Lianjun An and Bala Ramachandran

IBM T.J. Watson Research Center
Yorktown, New York 10598, U.S.A.
(914) 945-1565 & (914) 945-3768
{alianjun,rbala}@us.ibm.com

Abstract

Demand Conditioning is one of the methods used to address imbalances between supply and demand in supply chains. This requires the manufacturer to adjust the demand plan to respond to supply issues. The supply chain has several sources of delays and uncertainties such as lead times at different stages, forecast error, supply yield variability etc. that could potentially trigger or influence the conditioning process. In this paper, we examine dynamical effects in the conditioning process to study potential instabilities. We developed a Systems Dynamics model of a PC manufacturing supply chain to examine instabilities in the supply chain. This model provides insight on supply chain risks and error propagation due to unsynchronized execution. We also use the model to study the effect of different countermeasures to stabilize the supply chain.

Keywords: System Dynamics, Demand Conditioning, Supply Chain, Stability, on-demand operation.

1. Introduction

Managing uncertainties and dynamics in enterprise supply chains require an ongoing emphasis in aligning supply with demand. With the objective of using the supply chain as a source of competitive advantage, many enterprises are endeavoring not only to reactively responding to customer demand, but are also aspiring to proactively condition the supply chain to improve profits. A common example of this is the cross-selling of goods, using marketing promotions to avoid overstocking of specific products. The act of conditioning demand may have other consequences in the supply chain - in particular, due to lead times and uncertainties in the supply chain. Our objective in this paper is to research the dynamics of demand conditioning in supply chains using systems dynamics models.

The dynamic behavior of supply chains have been studied extensively, starting with the pioneering work of Forrester [1] in using systems dynamics models to demonstrate demand amplification in supply chains (otherwise known as bullwhip effect). Sterman [2] provides a nice overview of how systems dynamics can be used to study business dynamics. The sources of oscillations such as a failure to account for time delays are nicely illustrated for a variety of systems such as supply chains, labor markets etc. It is shown that perfectly rational strategies at a local level can cause system-wide oscillations, and control strategies to stabilize the system are proposed. Lee [3] has identified four drivers of the bullwhip effect – namely demand forecast updating, order batching, price fluctuation, rationing and shortage gaming and proposed strategies to counter them. The bullwhip effect is also taught in different management courses, due to its important practical implications.

From the perspective of demand conditioning it has been generally assumed that it is beneficial and little research has been done into dynamical effects associated with demand conditioning. Like any human system, conditioning demand in the supply chain is subject to time lags. If the demand can be conditioned very quickly relative to the time scale of supply variability, it is intuitive that demand conditioning can lead to supply chain benefits. However, the benefits are less clear if the time scale for conditioning is large compared with the time scale of supply variability. This scenario is very likely, since in many organizations, demand conditioning processes involve significant manual components such as instructing the sales force to tune what they are selling. In some situations, this may even involve developing new offerings that can be sold to the marketplace. Our main objective in this paper is to examine the dynamical effects if there are significant time lags in the conditioning process.

In the next section, we describe dynamical aspects of demand conditioning. In Section 3, we describe a systems dynamics model of demand conditioning actions in a PC supply chain. In Section 4, we discuss computational results from our Systems Dynamics model and discuss implications. We finish with our closing remarks in Section 5.

2. Demand Conditioning Dynamics

The conditioning processes in IBM PC Division are explained in [4] and can serve as a good example. When imbalance between demand and supply of components and products is detected, proactive actions can be taken to correct the situation [6]. The basic supply chain structure with conditioning process is shown in Figure 1. There are three decision points in this figure representing different types of conditioning.

- Supply conditioning: When the committed supply cannot meet the demand, it is possible that we can chase additional suppliers or adjust supply among different supply chain components.
- Demand conditioning: Through price change and promotion, we can provide incentives to customer to choose product alternatives.
- Offering conditioning: When there are some excessive parts, we can create and offer new configuration models to consume these parts.

We refer to assembled products as Machine Type Models (MTM). Components procured from suppliers are assembled to form major building blocks, which can be further assembled to make the MTM. Customer order creates demand on the MTMs and is backlogged into the order system (a pull model). The incoming part supply replenishes the inventory and makes parts being available for assembling (a push model). The demand-supply imbalance would be measured by the following expression:

$$\sum_i \left| p_i - \sum_j c_{i,j} m_j \right| ,$$

- where *m* is the vector representing demand amount for each MTM, *p* the vector of available parts for major building blocks, and *c* the BOM (bill of material) matrix – how a MTM is built-up from multiple parts.

When component supply is constrained, we have the option of choosing the allocations of components to different MTMs using different policies, such as priority, proportional allocation and optimizing allocations to maximize profit. Which rule is used might effect the long term instability and overall profit measure.

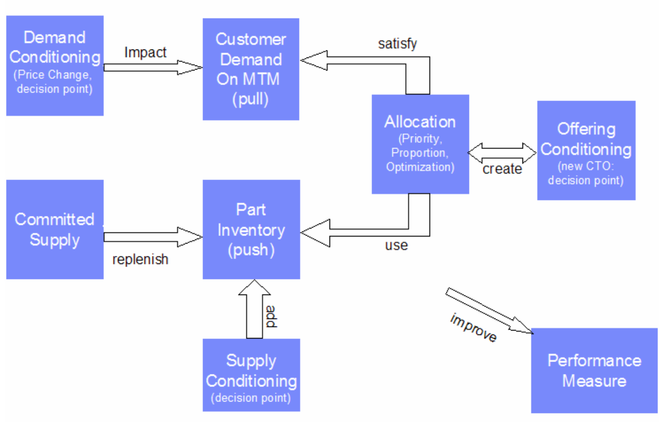

Figure 1: Supply Chain Conditioning Process

3. System Dynamics Model

In what follows, we study some instability issues related to the demand conditioning process. In particular, we examine the dynamic effects relating to time delays in the demand conditioning process. To gain a fundamental understanding, we only include uncertainty parameters related to synchronization in the conditioning process without explicitly including incentive actions to trigger demand shifting. We also further simplify the model by only considering two products in our model. We also do not consider allocation issues related to constrained supply between products. We present the stock and flow structure in section 3.1 and decision rules in section 3.2.

3.1. Stock and flow structure

The System Dynamics Model for demand conditioning process is shown in Figure 4. There are two basic stocks - order backlog and product inventory, in the model. The *Backlog* has in-and-out flows: *DemandRate* and *FulfilmentRate*. The *Inventory* has in-and-out flows: *ReplenishRate* and *ShipmentRate*. So the system would be formulated as the following

$$\begin{aligned} BackOrder[i](t) &= \int_{t_0}^{t} (DemandRate[i](\tau) - FulfilmentRate[i](\tau))d\tau + BackOrder[i](t_0) \\ Inventory[i](t) &= \int_{t_0}^{t} (ReplenishRate[i](\tau) - ShipmentRate[i](\tau))d\tau + Inventory[i](t_0) \end{aligned}, \quad (1)$$

where t is time, t_0 is the initial time for the processing, and $i=1,2$ is subscripted for product 1 and 2 in the model. The system consists of four equations here because two products are included. Note that both *BackOrder* and *Inventory* would enter the integrands on the right hand side since in-and-out flows are actually functions of them. In most cases, the in-and-out flows are nonlinearly dependent on the stocks variables with time-delays. As a result, the model cannot be solved analytically, but is amenable to solution through numerical methods.

In the system of equations, the *ReplenishRate* will be given and *DemandRate* will be adjusted based on decision rule given in the next subsection. In order to determine ShipmentRate, we introduce two variables (for each product, we omit subscripts from now on unless it is necessary): *DesiredShippingRate* and *MaximalShippingRate* defined by

$$\begin{aligned} DesiredShippingRate &= \frac{BackOrder}{DesiredShippingTime} \\ MaximalShippingRate &= \frac{Inventory}{Minimal\,ProcessingTime} \end{aligned} \quad . \qquad (2)$$

Since the *DesiredShippingRate* is determined by *BackOrder* and *MaximalShipping* is restricted by current Inventory *level*, it is obvious that *ShipmentRate* should be less than or equal to both in (2). The *DesiredShippingTime* is related to transportation delay, and the *MaximalShippingRate* is related to the time to process the order. Then *ShipmentRate* and *FulfilmentRate* are defined as follows:

$$\begin{aligned} &ShipmentRate = DesiredShippingRate * FulfilmentRatioFunc\left(\frac{MaximalShippingRate}{DesiredShippingRate}\right) \qquad (3) \\ &FulfilmentRate = ShipmentRate\,. \end{aligned}$$

The *FulfilmentRatioFunc* is given graphically as in Figure 2. When *MaximalShippingRate* is greater then *DesiredShippingRate*, *ShipmentRate* is close to *DesiredShippingRate* (*BackOrder* dominates the rate) since the value the fulfillment ratio function approaches one. When *MaximalShippingRate* is less then *DesiredShippingRate*, *ShipmentRate* is close to *MaximalShippingRate* since the slope of the curve is close to one (*Inventory* dominates the rate). As a lookup supported in Vensim [7], *FulfilmentRatioFunc* returns the nearest extreme value when the input goes outside the range of the lookup.

In this formulation, the following inequality holds

$$ShipmentRate \le Min(DesiredShippingRate, MaximalShippingRate)\,,$$

and *ShipmentRate* is a smooth function of the ratio of *MaximalShippingRate* over *DesiredShippingRate*.

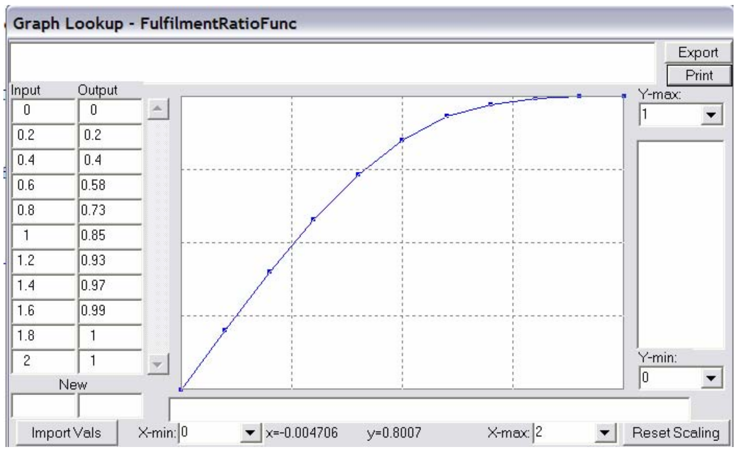

Figure 2: Fulfillment Ratio Function

3.2. Decision rules of conditioning process

Figure 3 shows the demand conditioning process. We periodically review the potential for demand conditioning – *ConditioningCycleTime* represents the review frequency, which triggers checking of the differences between *Inventory* and *BackOrder* (*IBDifference*) for both product 1 and 2. A positive *IBDifference* suggests an overage situation and a negative IBDifference suggests a shortage situation (we do not consider safety stock policy). When the shortage in one product can be compensated by an overage in the other product, and if both volumes are greater than the specified *Threshold*, the conditioning action is triggered and will initiate demand shifting between two products through some incentive means. We do not model the incentive process in our systems dynamics model, but directly compute the demand to be shifted and assume that the demand shift is somehow accomplished. The start and finish time for Conditioning would relate to the effectiveness of the mechanism to shift demand and may experience delays due to manual organizational process, etc. We do not investigate causal factors for synchronization of the conditioning process, but demonstrate the overall supply chain effects due to time delays in demand conditioning.

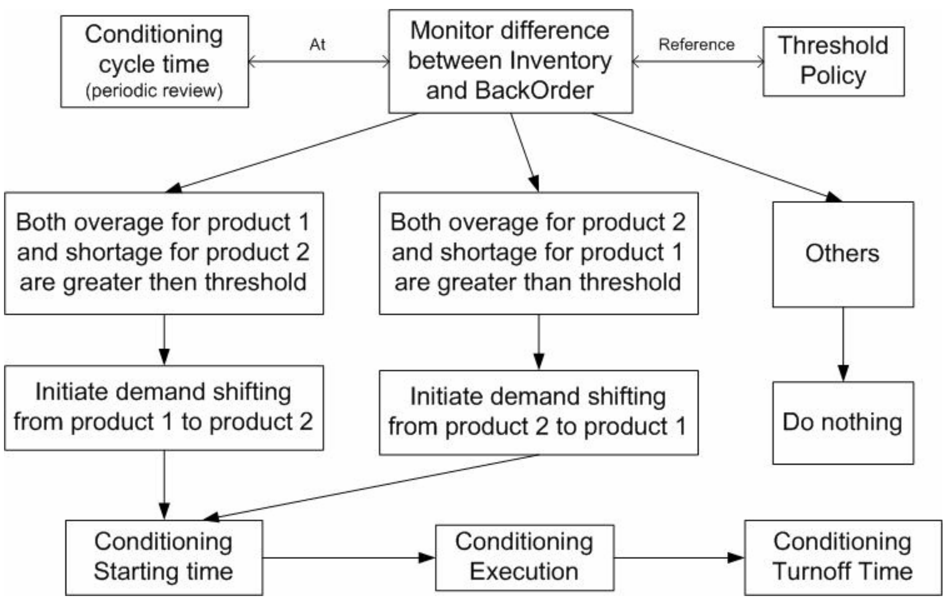

Figure 3: Demand conditioning process for two products

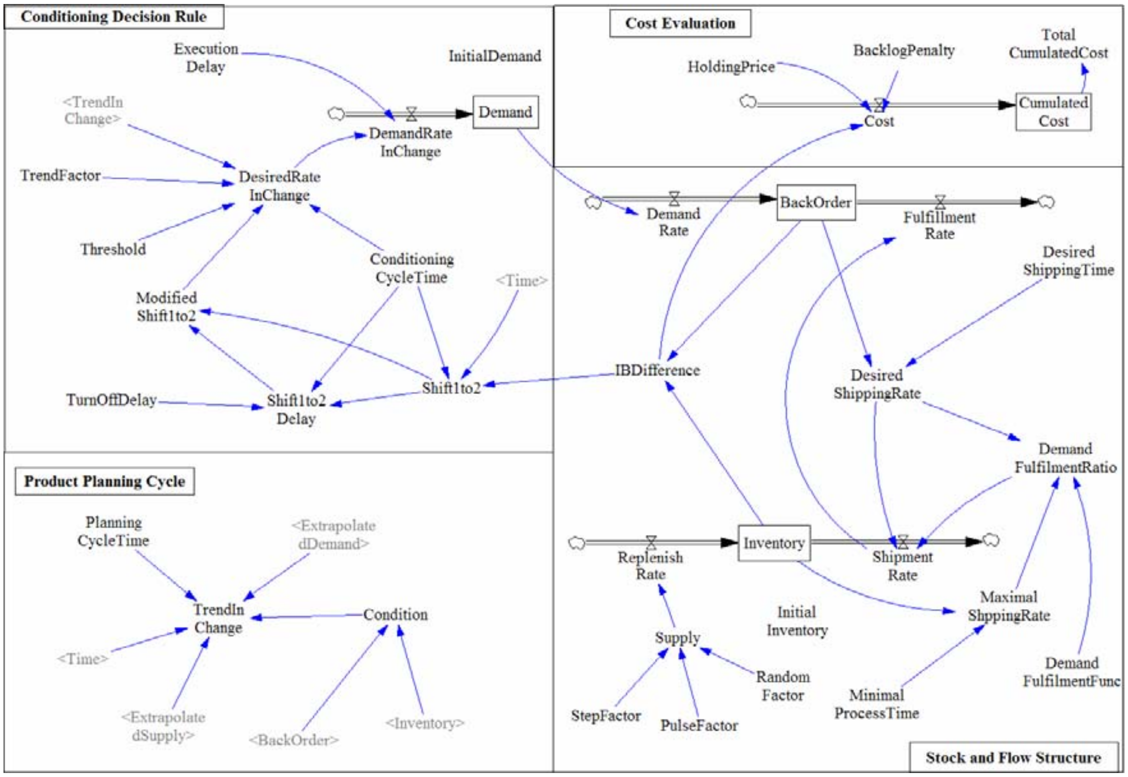

Figure 4: System Dynamics model of demand conditioning process

Now, we discuss the decision rules in the system dynamics model, shown in Figure 4. We have implemented the model through Vensim [7], and use the Vensim syntax to describe the decision rules. We start with periodic review of the difference of inventory and back order to determine the potential for demand conditioning. The demand shifting from product 1 to 2 *Shift1to2* is estimated based on the following:

$$\begin{array}{l} IfThenAlse\ \left((T > 0) : and : Modulo\ (T+1, CCT\ ) > 0 : and : Modulo\ (T+1, CCT\ ) <= 1, \right. \\ \quad IfThenAlse\ \left((IBD[1] > 0) : and : (IBD[2] < 0),\ Min\,(IBD[1], -IBD[2]), \right. \\ \quad\quad \left. \left. IfThenAlse\ \left((IBD[1] < 0) : and : (IBD[2] > 0),\ -Min\,(-IBD[1], IBD[2]),\ 0 \right), 0 \right), \right) \end{array}, \qquad (4)$$

where *T* is for *Time*, *CCT* for *ConditioningCycleTime*, and *IBD* for *IBDifference*. The review time constraint is expressed in the first part of equation (4). Note that the formulated condition 0 < *Modulo(T+1,CCT)* <= 1 guarantees that the system behavior does not depend on the time step chosen for numerical simulation. The second line is to check whether product 1 has overage and product 2 has shortage. Similarly, the third is to check whether product 1 has shortage and product 2 has overage. In both cases, the value would be the minimum of amplitudes of both overage and shortage. Otherwise, the value would be zero. The shifting is turned off by adding the negative of *Shift1to2* with delay *CCT+TurnOffDelay*. So the total possible shifting from product 1 to 2 would defined as

$$\begin{array}{l} ModifiedShift1to2[i] = \\ \quad\quad Shift1to2 - DelayFixed\,(Shift1to2, CCT + TurnOffDelay[i], Shift1to2) \quad i = 1,2 \end{array}.$$

By introducing *TurnOffDelay*, we could study the influence from the uncertainty associating with the switching off of the conditioning process. The *DesiredRateInChange*, which causes the demand change, is formulated as

$$\begin{aligned}&DesiredRateInChange[1]=\\&\quad IfThenAlse(Abs(ModifiedShift1to2[1])>Threshold,\ ModifiedShift1to2[1]/CCT,\ 0)\\&DesiredRateInChange[2]=\\&\quad IfThenAlse(Abs(ModifiedShift1to2[2])>Threshold,\ -ModifiedShift1to2[2]/CCT,\ 0)\ .\end{aligned} \tag{5}$$

When the *ModifiedShift1to2* is greater then *Threshold*, demand for product 1 is increased by *ModifiedShift1to2*[1]/*CCT* and demand for product 2 is decreased by *ModifiedShift1to2*[2]/*CCT*. Our formulation implies that we intend to correct the situation in a time period of duration *CCT*. To model delays in the execution of conditioning actions, we introduce another parameter *ExecutionDelay*. The real *DemandRateInChange* would be delayed with respect to *DesiredRateInChange*

$$\begin{aligned}&DemandRateInChange=\\&\quad DelayFixed\,(DesiredRateInChange, ExecutionDelay, DesiredRateInChange)\ .\end{aligned}$$

Finally, the *Demand* is determined through integrating *DemandRateInChange*

$$Demand(t)=\int_{t_0}^{t} DemandRateInChange(\tau)d\tau + Demand(t_0)\ .$$

DemandRate is non-negative and can be expressed as

$$DemandRate = Max\,(Demand\ ,0)\ . \tag{6}$$

Without the above constraint, *Demand* could become negative with execution delay and turnoff delay, This can happen because the effects of previous conditioning actions may not be seen due to the delays.

3.3. Cost formulation

We associate excessive inventory with holding costs and backlogged order with penalty costs. The excessive inventory is represented by the positive part of *IBDifference* and the backlogged order is represented by the negative part of *IBDifference*. So the cost is given by

$$Cost[i]=HoldingCost*Max(IBD[i],0)+BacklogPenalty*Max(-IBD[i],0)\ ,$$

where *IBD* is for *IBDifference*. And *CumulatedCost* will be integrated from *Cost*

$$CumulatedCost[i]=\int_{t_0}^{t} Cost[i](\tau)d\tau\ .$$

As seen from formulation that, we end up with the system of differential equations (four from subsection 3.1, two from 3.2 and two from 3.3) with nonlinearities, time delays and uncertainty. The work on the stability of stochastic delay differential equations in literature [8] is related to our context here and may be applicable to study the stability of demand conditioning processes. In this paper, we limit ourselves to a system dynamics study of the stability of demand conditioning processes.

4. Computational Results and Implications

In order to study dynamical effects in Demand Conditioning, we consider a simple two product supply chain. We introduce a supply spike in one product, along with a corresponding shortfall in another product that can trigger conditioning actions. We then explore the dynamics of inventories, backorders and costs for different values of the delays in the start and finish of the conditioning actions. These delays can be interpreted to be the result of different manual processes in Demand Conditioning.

4.1. Ideal case

We use the following for *ReplenishRate*

$$\begin{aligned} ReplenishRate[1] &= 100 + Pulse(10,20) \\ ReplenishRate[2] &= 90 - Pulse(10,20) \end{aligned}, \qquad (7)$$

which is shown in Figure 5. The function *Pulse* has two parameters: starting time (10) and duration (20). We use the Pulse function here to introduce spike in product 1 and drop in product 2 that can trigger conditioning actions. The initial inventory levels are set to be 200 and 100 for two products, and the initial backorders are set to be 100 and 90.

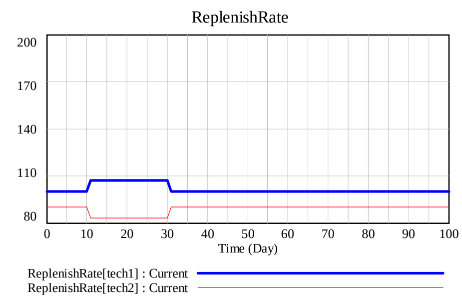

Figure 5: Replenishing rate from suppliers

For the demand rate, initially we also have 100 for product 1 and 90 for product 2. Because of the change in supply (Pulse in equation (7)), *IBDifference* for product 2 becomes negative and the imbalance occurs at *Time*=11 as shown in the right of Figure 6. First we demonstrate system behavior in an idealized case in which there are no execution delay (*ExecutionDelay*=0) and no conditioning turnoff delay (*TurnOffDelay*=0). We also set the conditioning cycle time (*CCT*) to be 7 and threshold to be 0. When Time reaches 14, demand conditioning is triggered based on decision rules expressed in (4), as shown in the left of Figure 6. In fact, there are subsequent demand conditioning actions at Time=21 and 28. Finally, the conditioning action is totally turned off at *Time*=42. The right of Figure 7 shows corresponding *IBDifference* curve for both products. The *IBDifference* roughly returns to the initial state after *Time*=42.

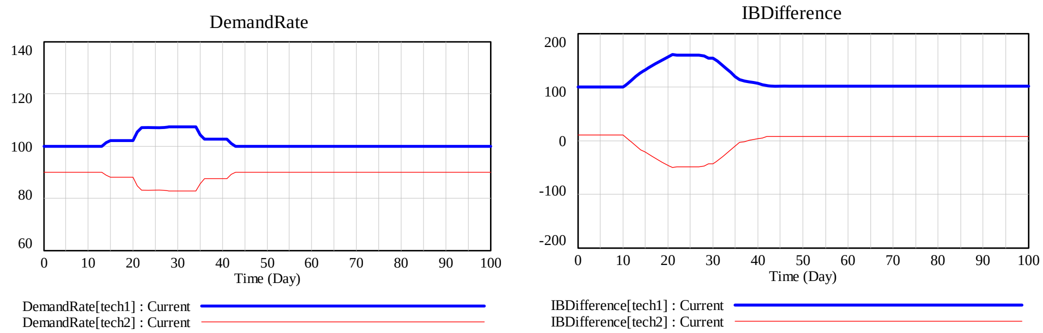

Figure 6: Demand conditioning effect in ideal case

4.2. Influence from delays

In reality, we do not have control on execution delay and conditioning turnoff delay. Figure 7 shows the demand rate and *IBDifference* profiles, when the execution delay is 4 and turnoff delay is 5. Due to the execution delay, the demand rate changes at Time=18 instead of 14. The *IBDifference* for product 1 changes from a large positive number to zero. In the other words, the product 1 changes from an overage to being just on the verge of a shortage situation.

If we continue to increase the turnoff delay, we end up with oscillations alternating with overages and shortages for products 1 and 2. Figure 8 shows the *IBDifference* for different *TurnOffDelay* with given *ExecutionDelay*=4. When the turnoff delay is 8, oscillation is dampened after two cycles and both products end up in an overage state, as shown in the left of Figure 8. However, when the turnoff delay is 10, the oscillation continues, growing in amplitude as shown in the right of Figure 8.

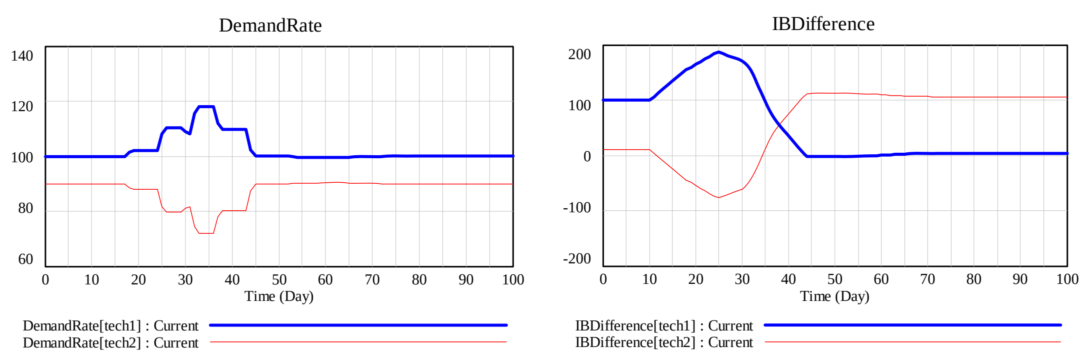

Figure 7: Demand conditioning effect in
the case of *ExecutionDelay*=4 and *TurnOffDelay*=5

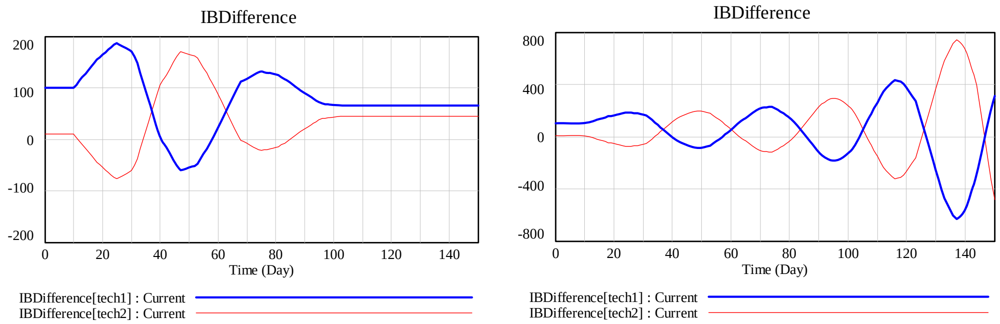

Figure 8: *IBDifference* in case of *ExecutionDelay*=4 and *TurnOffDelay*=8, 10

Figure 9 shows the *IBDifference* profile for a longer time horizon, under the same parameter settings as in Figure 8. The oscillation stops at a point with both products experiencing large shortages. In the context of this paper, we refer to this as unstable, even though the values converge to a large product shortage. The reason is that the inherently unstable parameter selection converges to a large shortage due to the demand non-negativity constraint on the demand rate (Equation. 6). Interestingly, if we remove this constraint, the oscillation continues with growing amplitude as shown in the right of Figure 9. However, it is unrealistic to remove equation 6 from the formulation. Hence, the model ends up with both products having shortages with amplified magnitudes for the unstable case compared with both products stabilizing to a reasonable profile for the stable case.

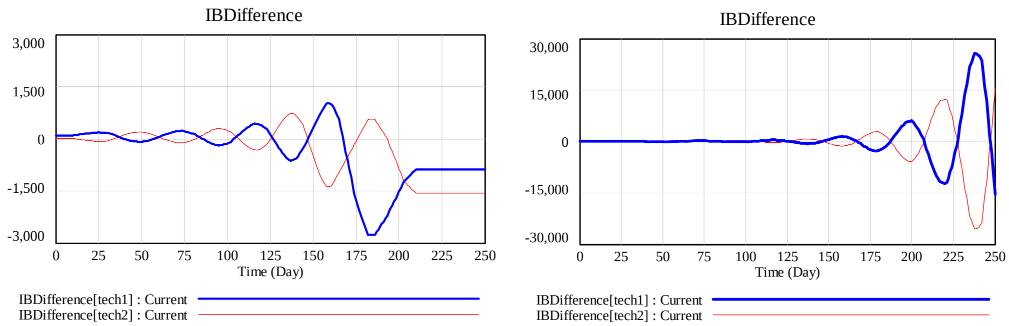

Figure 9: *IBDifference* in case of *ExecutionDelay*=4 and *TurnOffDelay*=10

If we study the effect of varying the execution delay keeping the turnoff delay fixed, we observe the same qualitative behavior. Figure 10 shows *IBDifference* for different *ExecutionDelay* with given *TurnOffDelay*=4. When the execution delay is 6, oscillation dampens after one cycle in the left of Figure 10; when the execution delay is 8, the oscillation continues to grow in amplitude, as shown in the right of Figure 10.

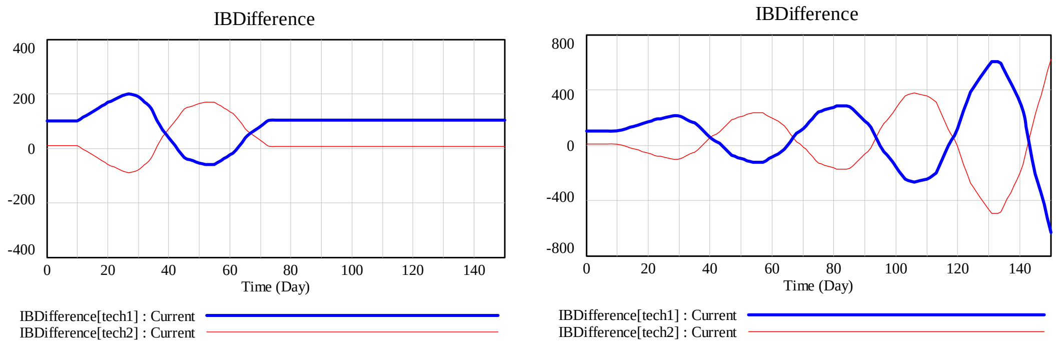

Figure 10: *IBDifference* in case of *ExecutionDelay*=6, 8 and *TurnOffDelay*=4

The right of Figure 11 shows the phase plot of *IBD*[1] versus *IBD*[2] for the case where the execution delay is (4, 5) (i.e. 4 for product 1 and 5 for product 2) and the turnoff delay is (8,9). It starts as a big oval and then shrinks in its diameter, ultimately stabilizing at the point (65.55, 56.06). The left of Figure 11 is the corresponding state plot. Since this choice of parameters corresponds to a stable case, both products stabilize to an overage state. Note that if we use the delay parameter settings (4,4) instead of (4, 5), the phase plot ends up as a straight line, oscillating between line segments of smaller lengths before converging to a point.

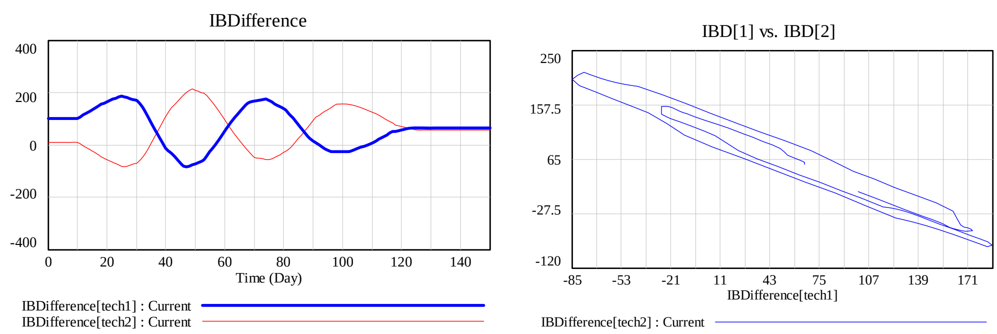

Figure 11: *IBDifference* for *ExecutionDelay*=(4,5) and *TurnOffDelay*=(8,9)

The right of Figure 12 shows the phase plot of *IBD*[1] versus *IBD*[2] for the case where the execution delay is (4, 5) and the turnoff delay is (9, 10). It starts a small oval, then continues to grow in diameter, ultimately stabilizing to the point (-126.27, -2476). The left of Figure 12 is the corresponding state plot. Since this parameter setting corresponds to an unstable case, both products stabilize to a shortage state with increased magnitude.

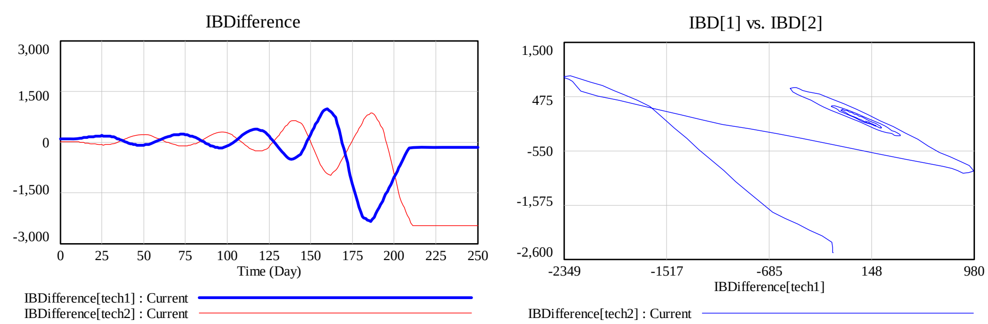

Figure 12: *IBDifference* for *ExecutionDelay*=(4,5) and *TurnOffDelay*=(9,10)

4.3. Cost Impacts

The left of Figure 13 shows the cost plot for *ExecutionDelay*=(4,5) and *TurnOffDelay*=(8,9). The cost curves have several peaks with the sum of amplitudes gradually becoming smaller, and then the costs stabilize at certain levels. Note that if the first bump is due to overage and holding cost of excessive inventory, then the next bump would be due to shortage and the penalty cost of backlogged order. The points at which the curves cross zero are switching points between overage and shortage. Starting at Time=14, the first bump for product 2 is shortage cost and the first bump for product 1 is overage cost.

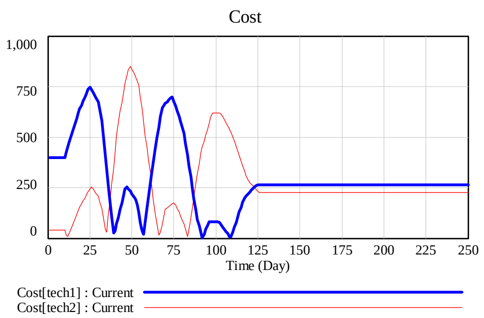

Figure 13: Cost for ExecutionDelay=(4,5) and TurnOffDelay=(8,9)

The left of Figure 14 shows the cost plot for *ExecutionDelay*=(4,5) and *TurnOffDelay*=(9,10). The cost curves keep growing in amplitude, ultimately stabilizing at very high levels due to a large shortage. The right is the corresponding phase plot of *Cost*[1] vs. *Cost*[2]. The phase change range becomes bigger and bigger before stabilizing at a point far away from the origin.

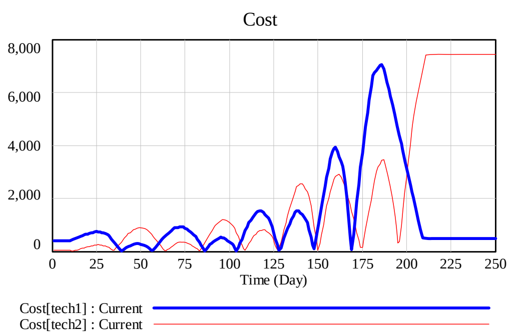

Figure 14: Cost for ExecutionDelay=(4,5) and TurnOffDelay=(9,10)

These simulations suggest that there exists a transition point at which the system transitions from a stable behavior to an unstable behavior. Figure 15 shows the plot of the average cost versus turnoff delay with given *ExecutionDelay*=4 and initial inventory level being (200,100). When the turnoff delay is less than or equal to 9, the average cost has small variation and is small. When the turnoff delay is beyond 9, the average cost has a sharp transition.

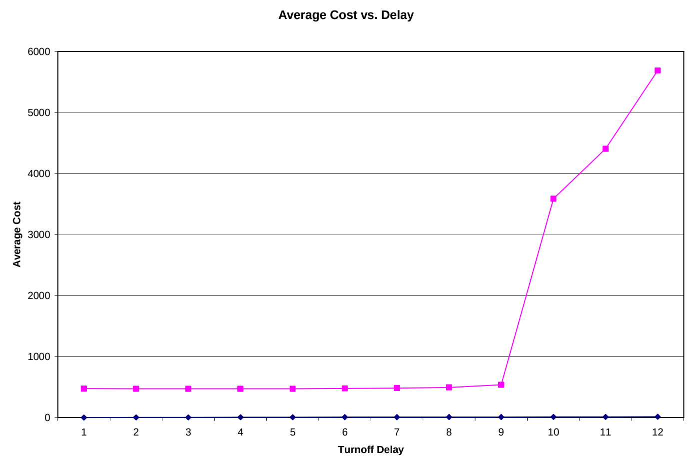

Figure 15: Cost Jump during Transiting from Convergence to Divergence

4.4. Countermeasures

Both subsections 4.2 and 4.3 demonstrate dynamical effects for different values of time delays in the conditioning process. We now discuss how these instabilities can be potentially managed.

4.4.1. Inventory level influence

Increasing inventory levels would in general increase the time before product shortages are seen and further reduce the frequency of oscillations. Figure 16 shows that, when we change initial inventory for product 2 from 100 to 150 and keep *TurnOffDelay*=10 and *ExecutionDelay*=4, behavior of *IBDifference* changes significantly comparing with the right of Figure 8 in which inventory baseline is (200,100) and oscillates one cycle, before stabilizing.

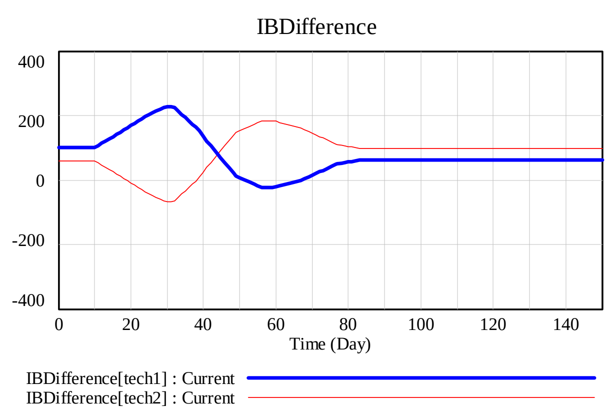

Figure 16: IBDifference after increasing inventory level
for *ExecutionDelay*=4 and *TurnOffDelay*=10

Figure 17 shows plot of average cost versus the turnoff delay for different initial inventory levels. When the initial inventory is (100, 90), average cost will jump when the turnoff delay is 3. When the initial inventory is (200,100), average cost will jump when the turnoff delay is 9. When the initial inventory moves up to (200,150), average cost will jump when the turnoff delay is 13. It implies that, in order to make the system more stable, a potential countermeasure is to increase the inventory level (safety stock level). However, notice from Figure 17 that the average costs are higher when the inventory levels are increased. This suggests a tradeoff between operational costs and risk of instability. This also suggests that stability issues need to be considered when enterprises drive cost reduction initiatives, because inventory reduction brings along with it an increased risk of instability.

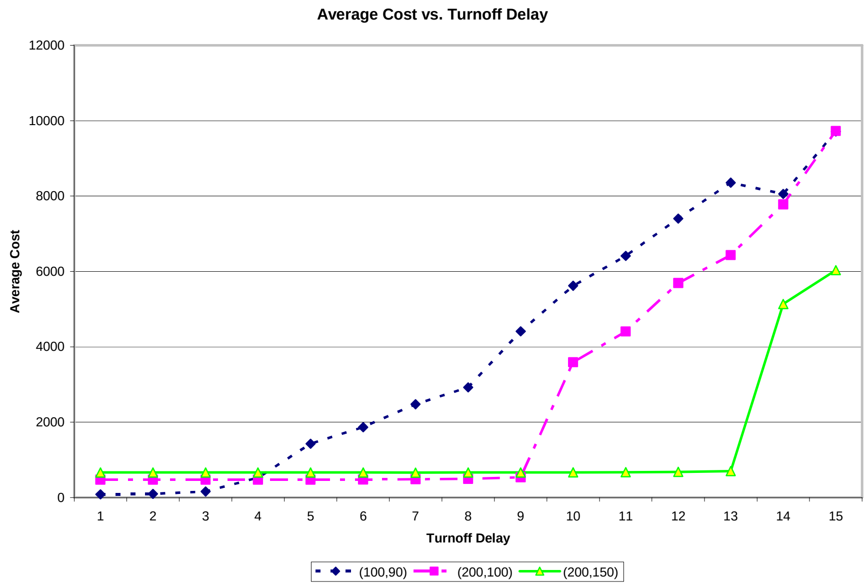

Figure 17: Cost Transition Change for Different Inventory Levels

4.4.2. Conditioning line consideration

Another countermeasure is to avoid excessive conditioning which could happen if conditioning actions are done based on partially or completely ignoring the effects of earlier conditioning actions. This is important because in the presence of time delays, the effects of past conditioning actions may not be immediately manifested in the inventories and backlogs that drive conditioning actions according to equations (4) and (5). In the ideal case, this situation never happens since there are no delays and the effects of past conditioning actions are seen before any further conditioning actions are decided upon. This situation is not unlike the systems dynamics models in Reference [2], where it is shown that ignoring the supply lines partially or completely is an important source of amplifications. Hence, the decision rule for new conditioning actions should account for the conditioning line, as shown in Figure 18.

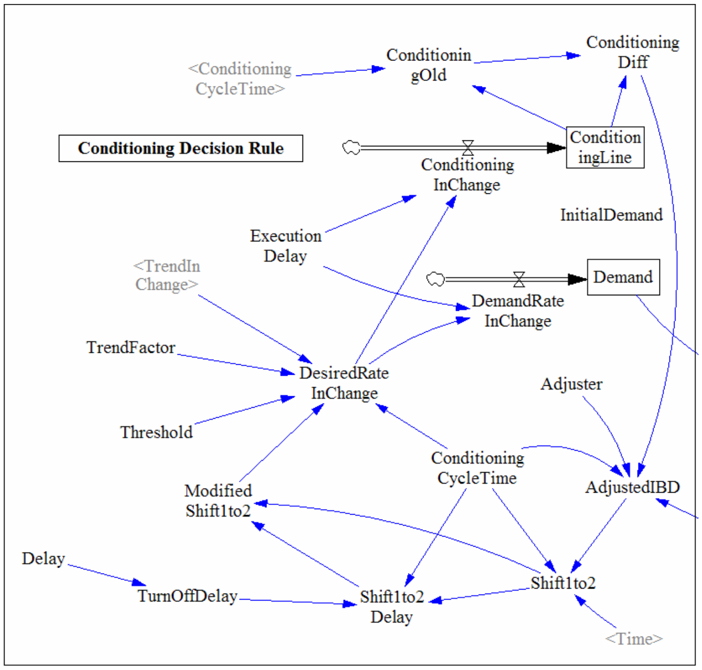

Figure 18: Decision Rule with Conditioning Line

Instead of using *IBDifference* to determine *Shift1to2* in Eq. 4, we used *AdjustedIBD* (this is adjusted by the current conditioning pipeline),

$$AdjustedIDB = IBDifference - ConditioningDiff * Adjuster \ ,$$

$$ConditioningDiff = ConditioningLine - FixedDelay(ConditioningLine, CCT, ConditioningLine) \ .$$

By choosing *Adjuster*=1, we take the last *CCT* period conditioning lines into account. The conditioning line is calculated as the following

$$ConditioningLine(t) = \int_{t_0}^{t} FixedDelay(DRIC(\tau), ExecutionDelay, DRIC(\tau))d\tau \ ,$$

where *DRIC* is for *DesiredRateInChange*. Figure 19 shows the result for the execution delay being 4 and turnoff delay being 19. We observe stable behaviors even for large turnoff delays.

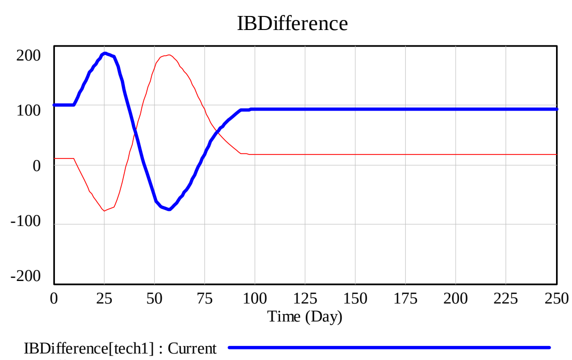

Figure 14: *IBDifference* after Conditioning Line Adjustment

5. Closing Remarks

In this paper, we have discussed dynamical effects arising out of demand conditioning in the supply chain, using a systems dynamics model. We showed the presence of oscillations in supply shortages created due to the conditioning actions, if the time delays in managing the conditioning are long. There are other interesting dynamical issues relating to demand conditioning, arising from the presence of uncertainties in the supply chain. For instance, we may forecast a future inventory surplus for a specific product and trigger conditioning actions to sell the excess inventory. However, since supply is uncertain, it is possible that the inventory surplus may not materialize due to a number of reasons – for example, the suppliers yield may be variable. If this happens, we may have triggered the conditioning action too soon and as a result, create an inventory shortage instead of a surplus, as an unintended consequence of the action taken. An interesting question for future work in this regard is how to trigger the conditioning actions, given the supply and demand uncertainty and lead times in the supply chain.

We suspect that the instabilities discussed in this paper are equally applicable to supply conditioning. This is particularly relevant, since all the suppliers in different tiers are not integrated; hence, it is likely that some of the supply lines are not accounted for in supply chain planning. Since there are considerable time delays in supply management, this is a potential source of instabilities. It would be interesting to explore this deeper, to understand the drivers of instability in supply conditioning and also to examine potential countermeasures.

References

[1] J.W. Forrester, Industry Dynamics, MIT Press, Cambridge, MA, 1961.

[2] J.D. Sterman, Business Dynamics: System Thinking and Modeling for a Complex World, Irwin McGraw-Hill, Boston, 2000.

[3] Hau L. Lee, V. Padmanabhan, S. Whang, Information Distortion in a Supply Chain: the Bullwhip Effect, Management Science, Vol. 43, No. 4, April 1997, pp. 546-558.

[4] S. Kapoor, K. Bhattacharya, S. Buckley, P. Chowdhary, M. Ettl, K. Katircioglu, E. Mauch, L. Phillips, "A Technical Framework for Sense & Respond Business Management", *IBM Systems Journal*, Vol. 44, No. 1, pg 5, 2005.

[5] J.D. Sterman, Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment, Management Sciences, Vol. 35, No. 3, March 1989, pp. 321-339.

[6] P. Huang, Y.M. Lee, L. An, M. Ettl, K. Soururajan, S. Buckley, Utilizing Simulation to Evaluate Business Decisions in Sense-and-Respond Systems, Proceedings of the 2004 Winter Simulation Conference, 2004.

[7] Vensim: http://www.vensim.com

[8] A.F. Ivanov, Yu. Kazmerchuk and A.V. Swichchuk, *Theory, Stochastic Stability and Applications of Stochastic Delay Differential Equations: A Survey of Recent Results*, Differential Equations and Dynamical Systems, Vol. 11, Nos. 1&2, 2003, pp. 55-115.