CT561: Systems Modelling and Simulation

Week 9:
Delays and the Stock Management Structure

https://github.com/JimDuggan/CT561

Dr. Jim Duggan,

Information Technology,

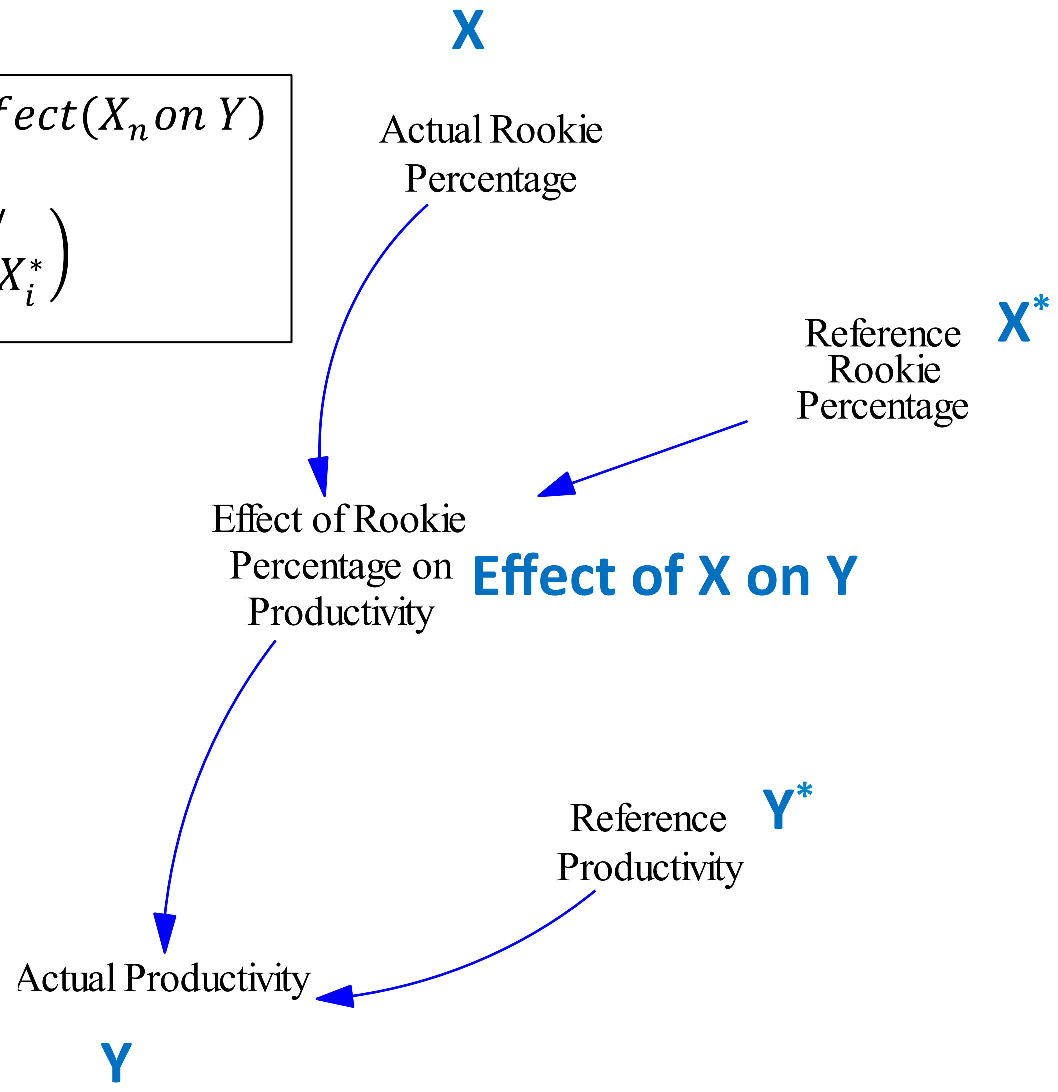
School of Engineering & Informatics

NUI Galway
OÉ Gaillimh

Recap – Formulating Effects

$$Y = Y^{*} \times Effect\,(X_1 on\, Y) \times \ldots \times Effect(X_n on\, Y)$$

$$Effect\,(\,X_i\; on\; Y) = f\left({X_i}/{X_i^{*}}\right)$$



Exploring the effect.

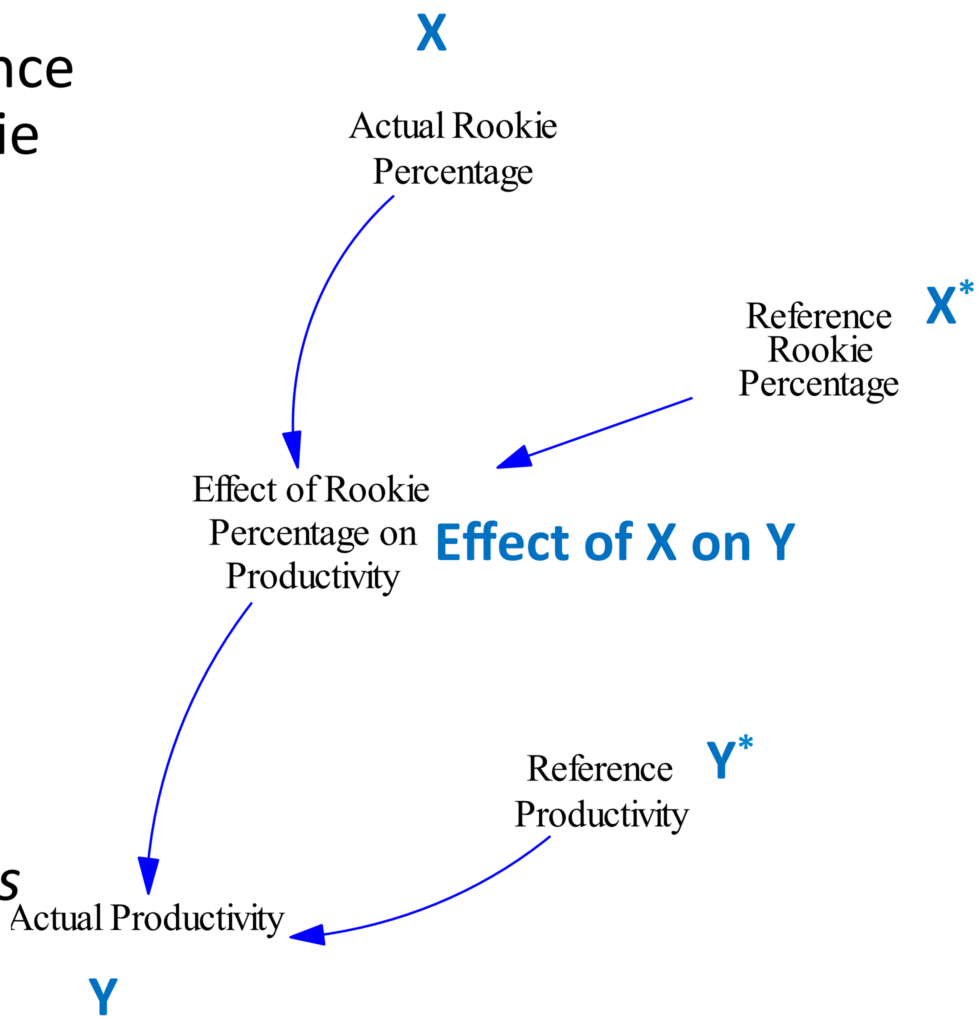
- Actual Productivity = Reference Productivity * Effect of Rookie Percentage on Productivity

$$Effect\ (X_i\ on\ Y) = f\left({X_i}/{X_i^*}\right)$$

Reference Productivity	Reference Rookie Percentage	Actual Rookie Percentage	Effect Multiplier	Actual Productivity
100	20%	20%	1	100
100	20%	40%	< 1	< 100
100	20%	10%	> 1	> 100

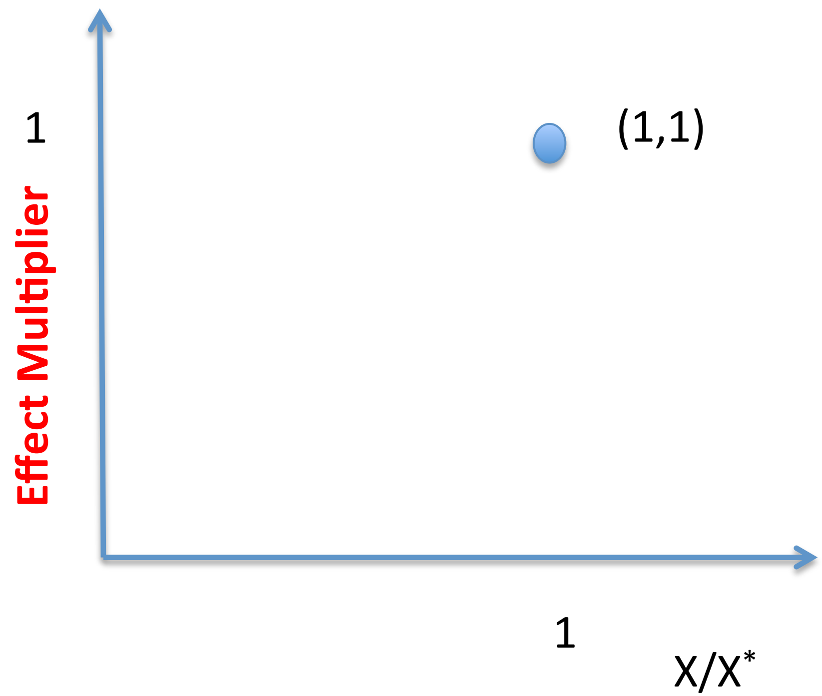
The Effect Equation

- Actual Productivity = Reference Productivity * Effect of Rookie Percentage on Productivity
- Effect of X on Y = $F(X/X^*)$
- Normalised Value
- *When $X = X^*$, $F(X) = 1$*
- *X^* and Y^* are reference values*



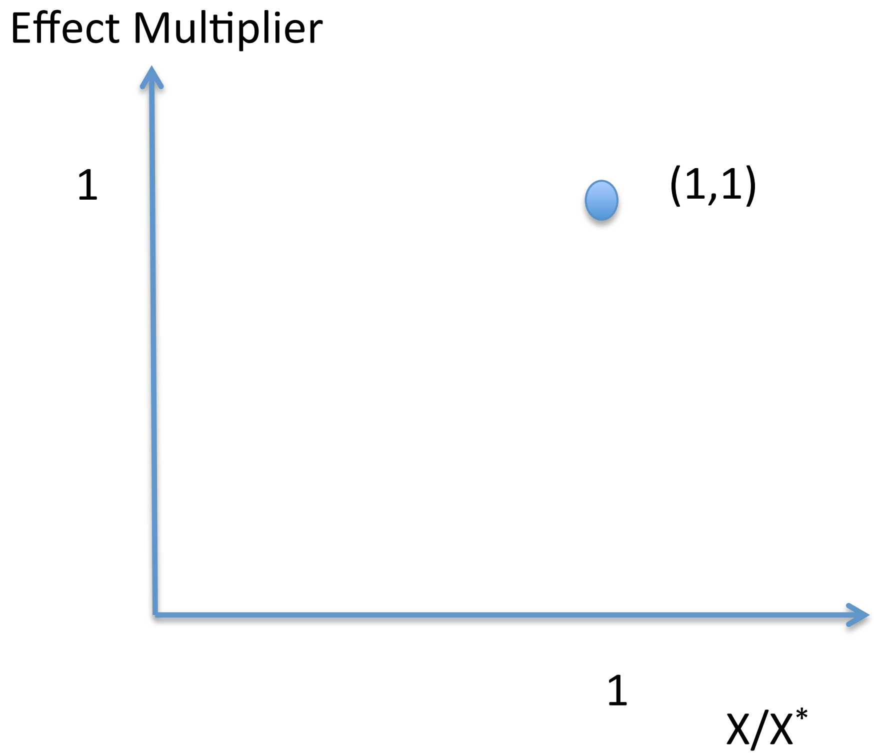
Example

- X = Rookie Percentage
- X^* = Reference Rookie Percentage
- Impact on experienced productivity?
- (1,1) is always on the line



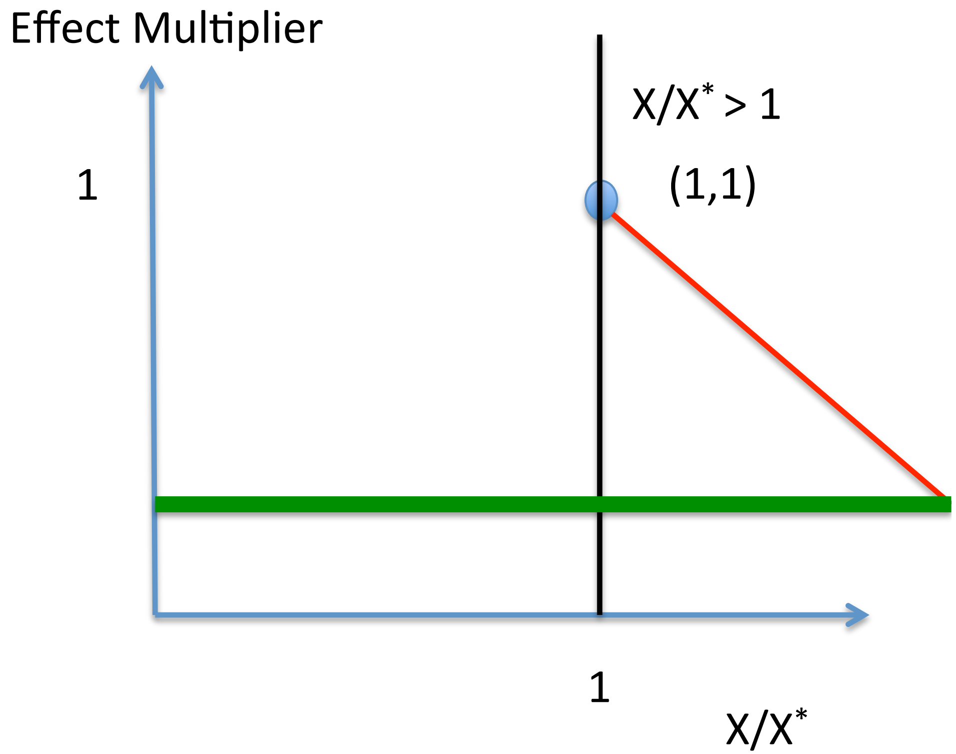
Thinking about the effects…

- X = Actual Rookie Percentage
- X^* = Reference Rookie Percentage (i.e. the number at which our experienced productivity is at its reference value)
- Question:
  - If $X > X^*$, Effect?
  - If $X < X^*$, Effect?



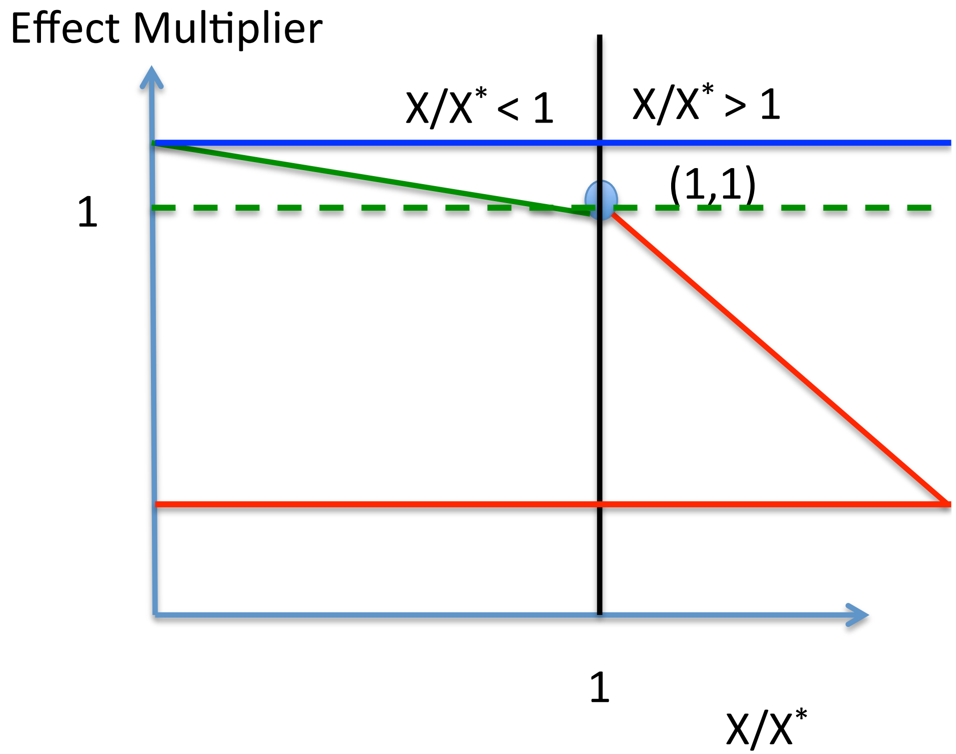
Sketching the relationship, More rookies than reference value

- $X > X^*$
  - We have more Rookies than our target level
  - This will reduce our experienced productivity
  - More work to train rookies
  - Effect will be lower than 1
  - Decide on minimum value (0.25)

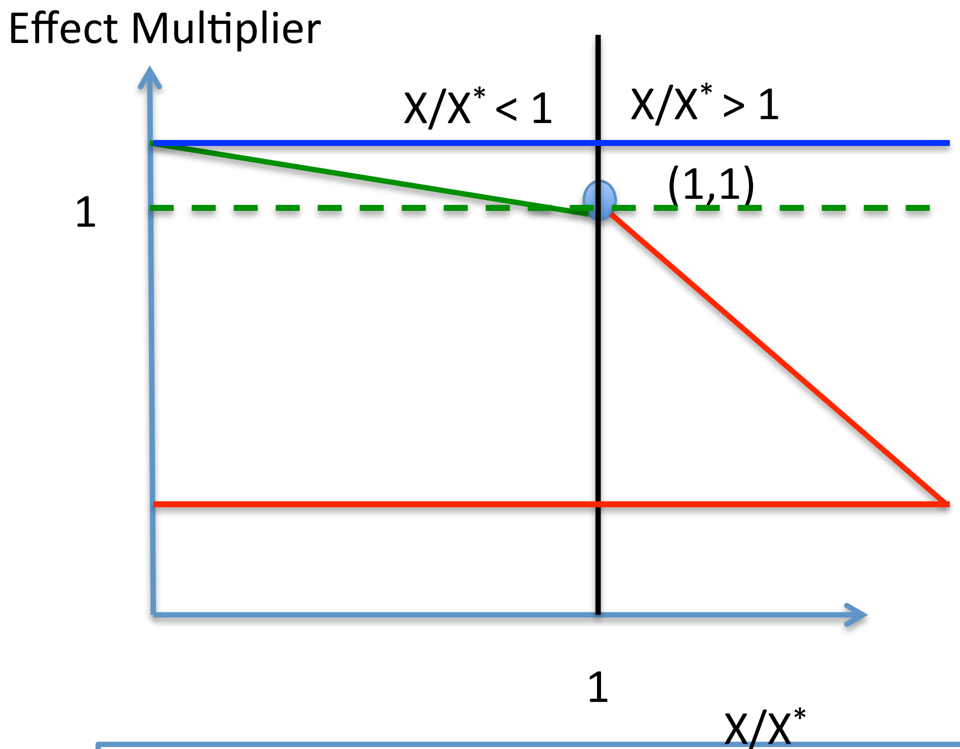


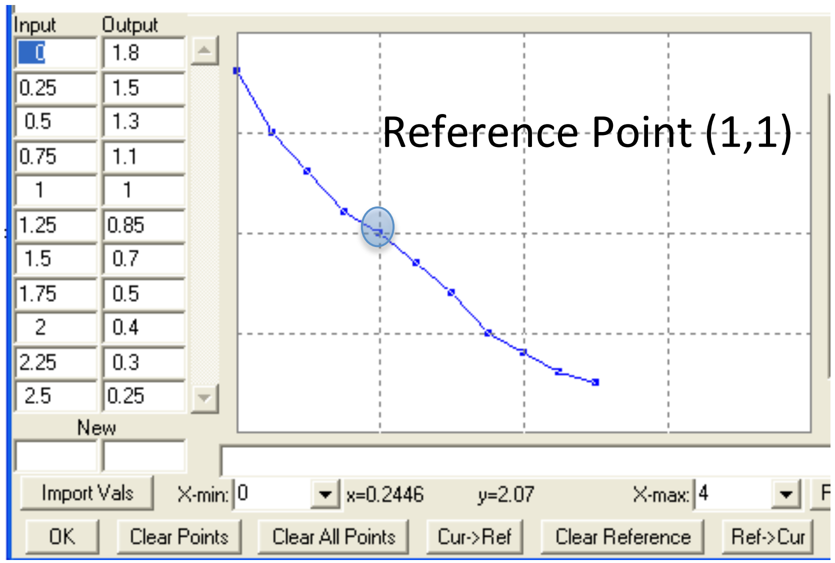
Sketching the relationship, Less rookies than reference value

- $X < X^*$
  - We have less Rookies than our target level
  - This will increase our experienced productivity
  - Less work to train rookies
  - Effect will be greater than 1
  - Decide on a maximum value (1.8)



Additional Information





Effect of Rookie Percentage on Productivity = WITH LOOKUP(Actual Rookie Percentage / Reference Rookie Percentage , ([(0,0)-(4,2)],(0,1.8),(0.25,1.5) , (0.5,1.3),(0.75,1.1),(1,1),(1.25,0.85),(1.5,0.7),(1.75,0.5),(2,0.4),(2.25,0.3),(2.5,0.25)))

Full Equations

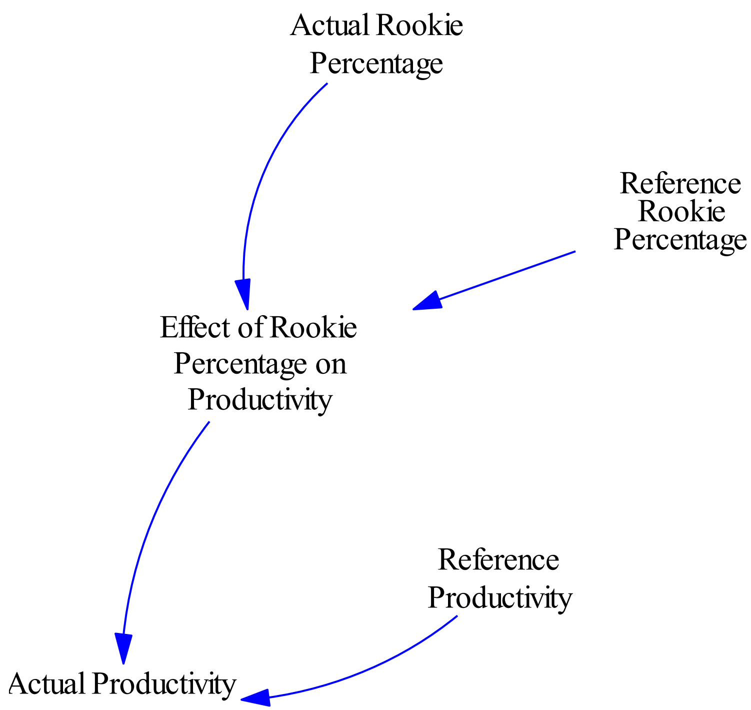
Actual Productivity = Effect of Rookie Percentage on Productivity * Reference Productivity

Actual Rookie Percentage = 0.2 + ramp (0.03, 10, 20)

Effect of Rookie Percentage on Productivity = WITH LOOKUP(Actual Rookie Percentage / Reference Rookie Percentage , ([(0,0)-(4,2)],(0,1.8),(0.25,1.5) ,(0.5,1.3),(0.75,1.1),(1,1),(1.25,0.85),(1.5,0.7),(1.75,0.5),(2,0.4),(2.25,0.3),(2.5,0.25)))

Reference Productivity = 100

Reference Rookie Percentage = 0.2



(1) Delays

- "Delays are pervasive.
  - It takes time to measure and report information.
  - It takes time to make decisions.
  - It takes time for decisions to affect the state of the system" (Sterman 2000)
- We need to use delays in many of our models

The output of a delay lags behind the input:

General structure of a material delay:

The post office as a delay:

Delay Distributions

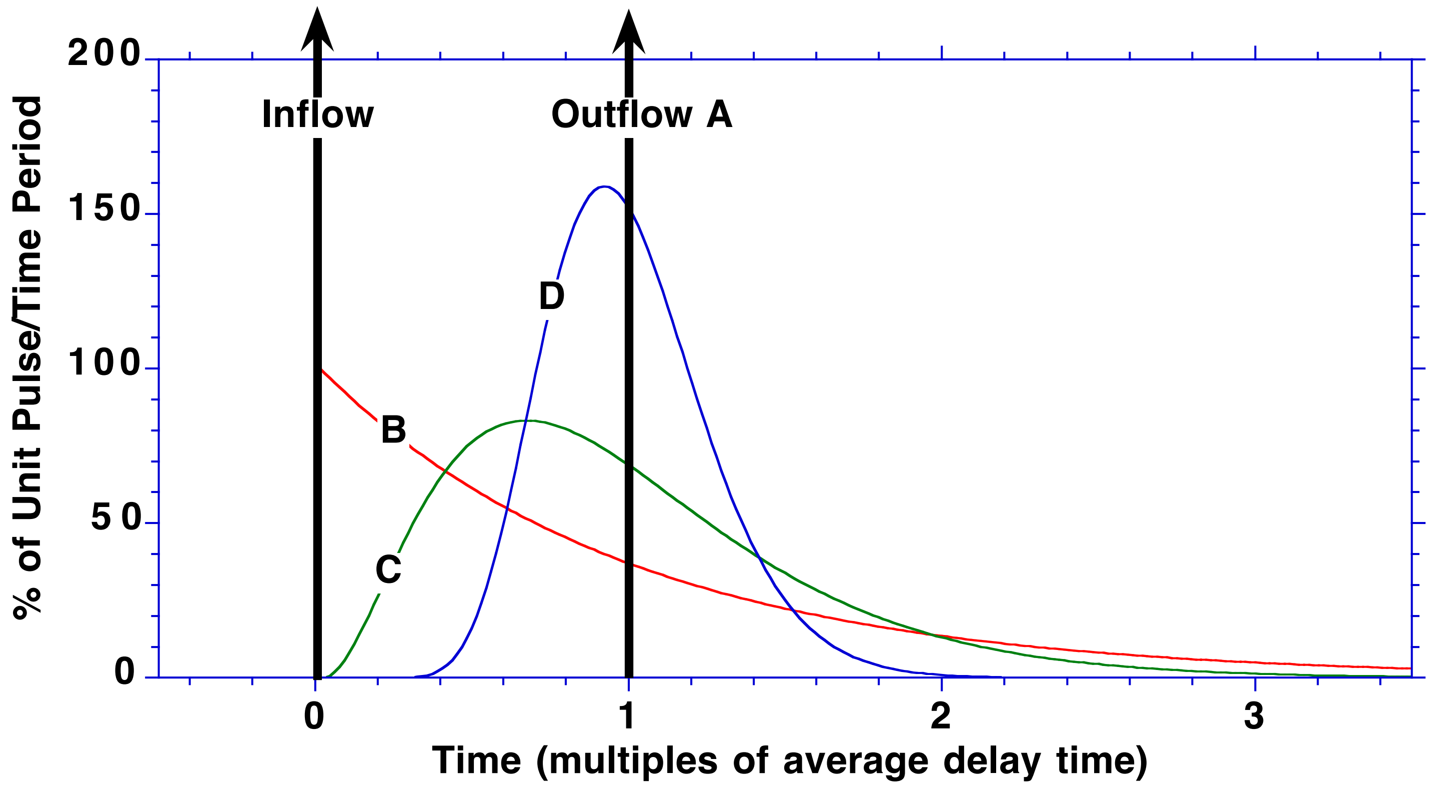


Figure 11-2 Some distributions of the outflow from a delay

The input I all cases is a unit pulse at time zero. Outflow A is a pipeline delay in which all items arrive together exactly 1 delay time after they enter. Outflow distributions B-D exhibit different degrees of variation in processing times for individual items so some arrive before and some after the average delay time. In all cases the average delay time is the same and the areas under each distribution are equal.

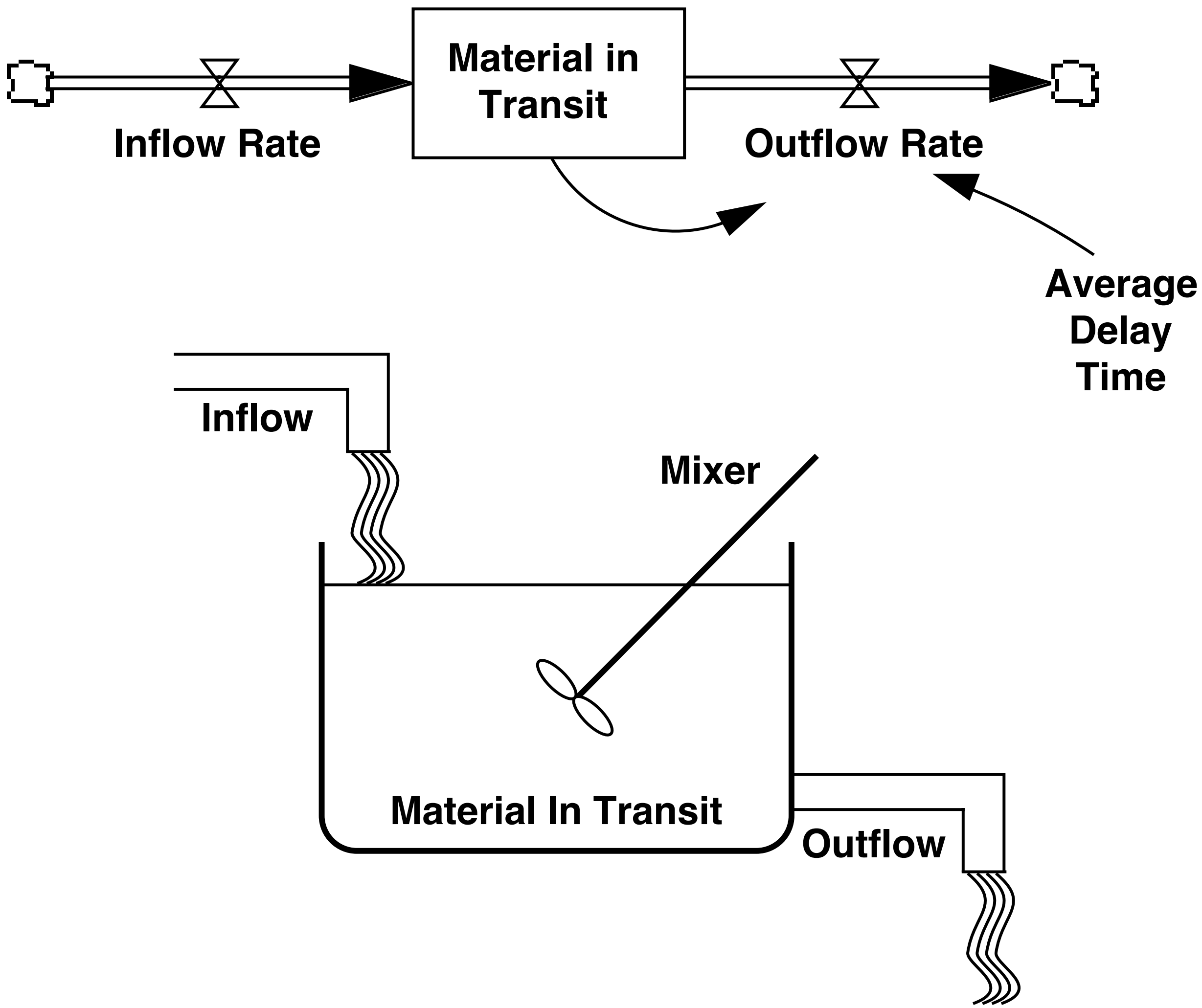
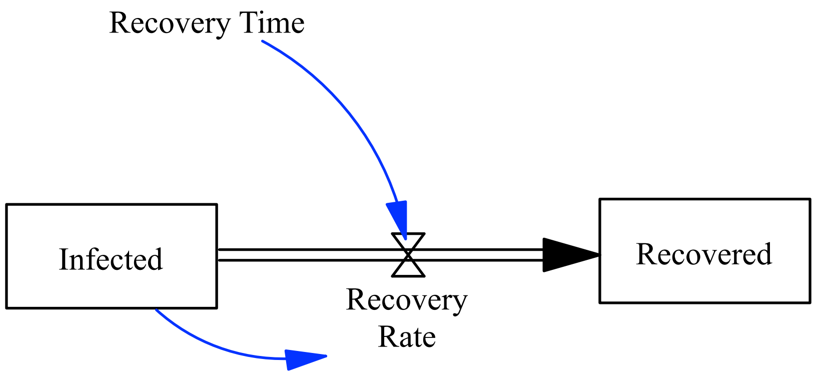


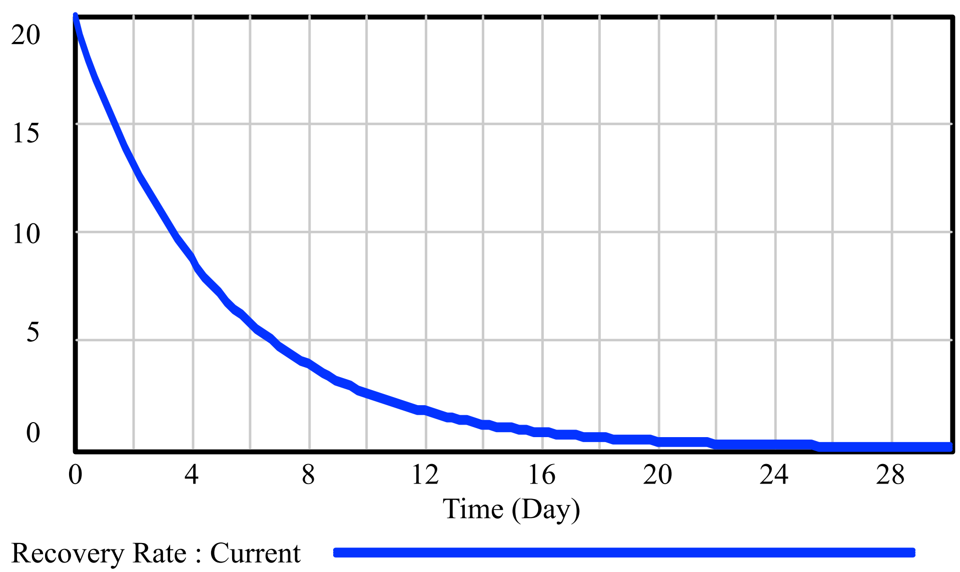
Figure 11-4 First-order material delay: structure

The outflow is proportional to the stock of material in transit. The contents of the stock are perfectly mixed at all times, so all items in the stock have the same probability of exit, independent of their arrival time.

Example 1: Recovery Delay



Infected= INTEG (-Recovery Rate, 100)
Recovered= INTEG (Recovery Rate,0)
Recovery Rate= Infected/Recovery Time
Recovery Time= 5



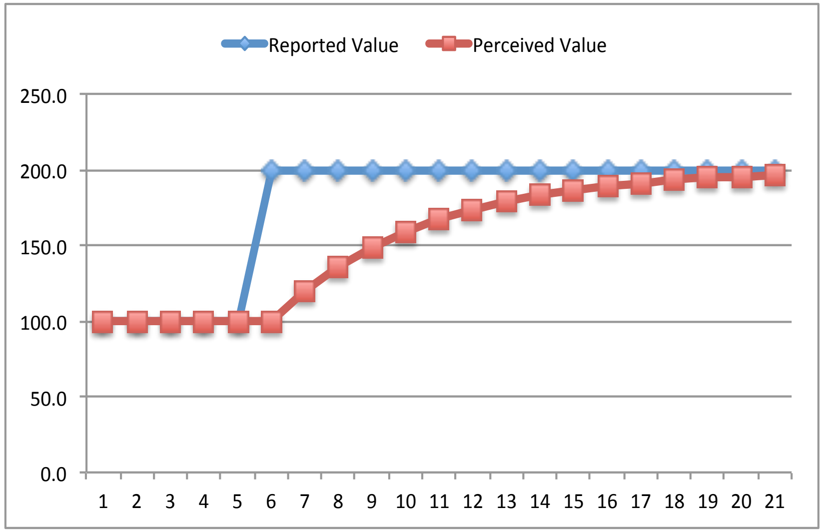
Time (Day)	Infected	Recovery Time	Recovery Rate
0	100	5	20
0.25	95	5	19
0.5	90.25	5	18.05
0.75	85.74	5	17.15
1	81.45	5	16.29
1.25	77.38	5	15.48
1.5	73.51	5	14.7
1.75	69.83	5	13.97
2	66.34	5	13.27
2.25	63.02	5	12.6
2.5	59.87	5	11.97
2.75	56.88	5	11.38
3	54.04	5	10.81
3.25	51.33	5	10.27
3.5	48.77	5	9.753
3.75	46.33	5	9.266
4	44.01	5	8.803

Challenge 9.1

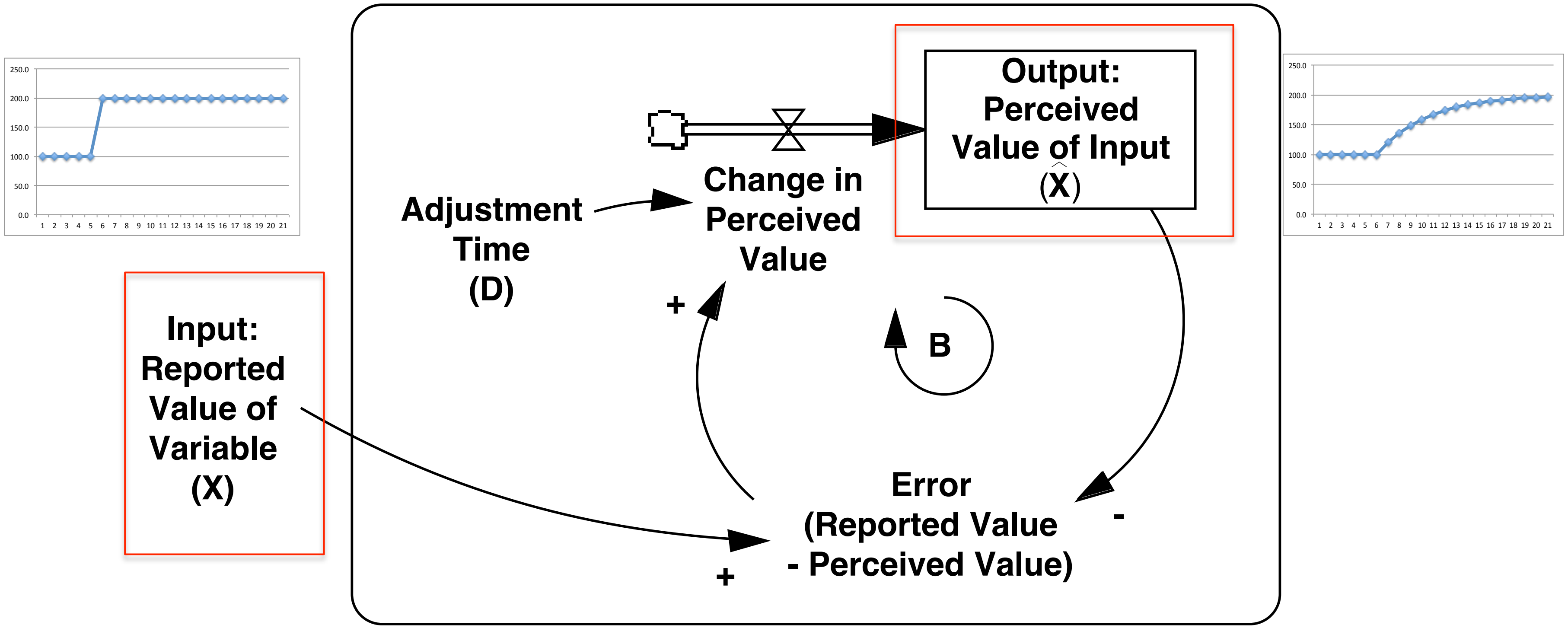
- Build an aging chain model career progression, where software engineers have the following trajectory:
  - Graduate Engineers (24 months)
  - Software Engineers (36 months)
  - Senior Software Engineers (48 months)
  - Consulting Engineers (60 months)

Information Delay (Smoothing)

- Model of decision maker's expectation (what value might a variable take on?)
- Similar to a forecast (exponential smoothing)
- Input is an actual value, output an expectation (perceived value)



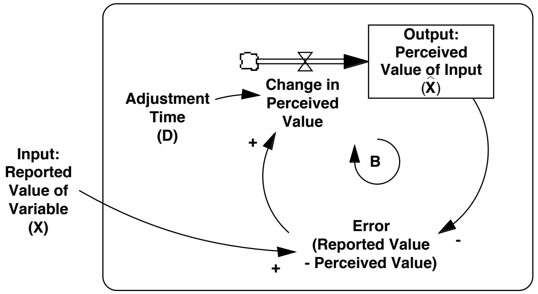
Information Delay Stock and Flow Model



$\hat{X}$ = INTEGRAL(Change in Perceived Value, $\hat{X}(0)$)

Change in Perceived Value = Error/D = $(X - \hat{X})/D$

Excel Example – Exponential Smoothing



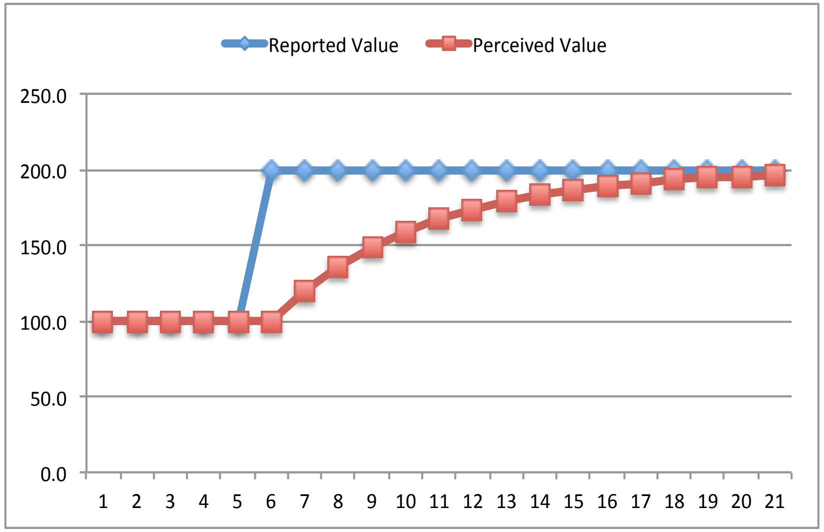
$\hat{X}$ = INTEGRAL(Change in Perceived Value, $\hat{X}(0)$)
Change in Perceived Value = Error/D = $(X - \hat{X})/D$

Time	Reported Value	Perceived Value	Error	AT	Change in Perceived Value
0.0	100.0	100.0	0.0	5.0	0.0
1.0	100.0	100.0	0.0	5.0	0.0
2.0	100.0	100.0	0.0	5.0	0.0
3.0	100.0	100.0	0.0	5.0	0.0
4.0	100.0	100.0	0.0	5.0	0.0
5.0	200.0	100.0	100.0	5.0	20.0
6.0	200.0	120.0	80.0	5.0	16.0
7.0	200.0	136.0	64.0	5.0	12.8
8.0	200.0	148.8	51.2	5.0	10.2
9.0	200.0	159.0	41.0	5.0	8.2
10.0	200.0	167.2	32.8	5.0	6.6
11.0	200.0	173.8	26.2	5.0	5.2
12.0	200.0	179.0	21.0	5.0	4.2
13.0	200.0	183.2	16.8	5.0	3.4
14.0	200.0	186.6	13.4	5.0	2.7
15.0	200.0	189.3	10.7	5.0	2.1

Error = Reported - Perceived

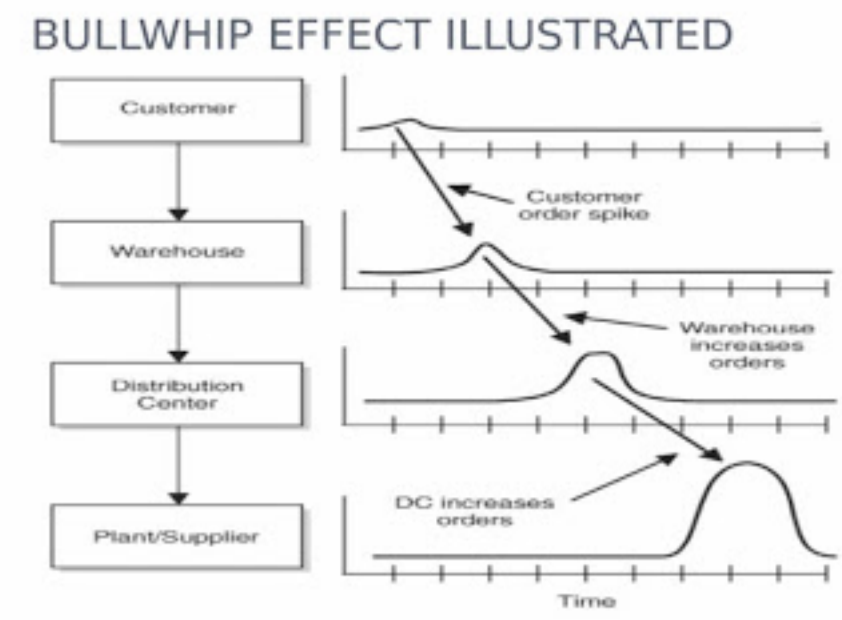
Change in Perceived = Error / AT

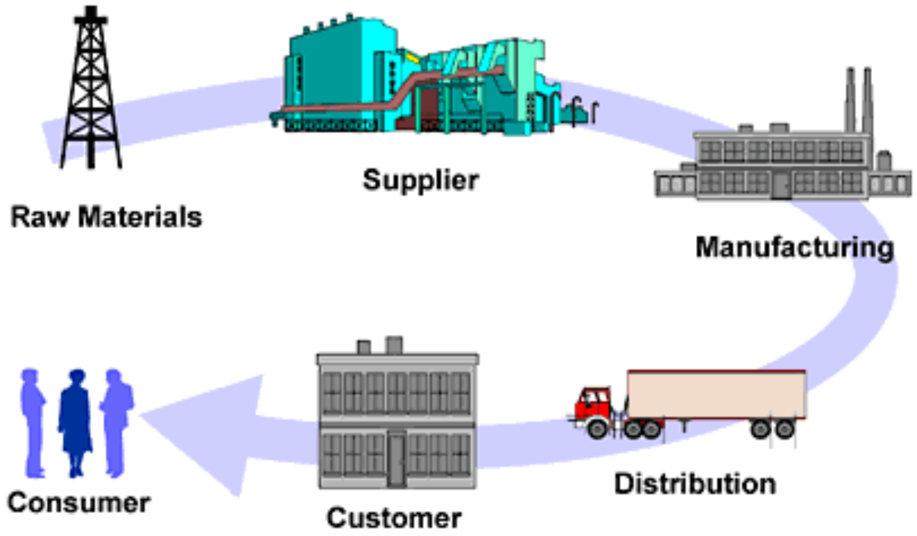
$\text{Perceived}_t = \text{Perceived}_{t-1} +$ Change in Perceived



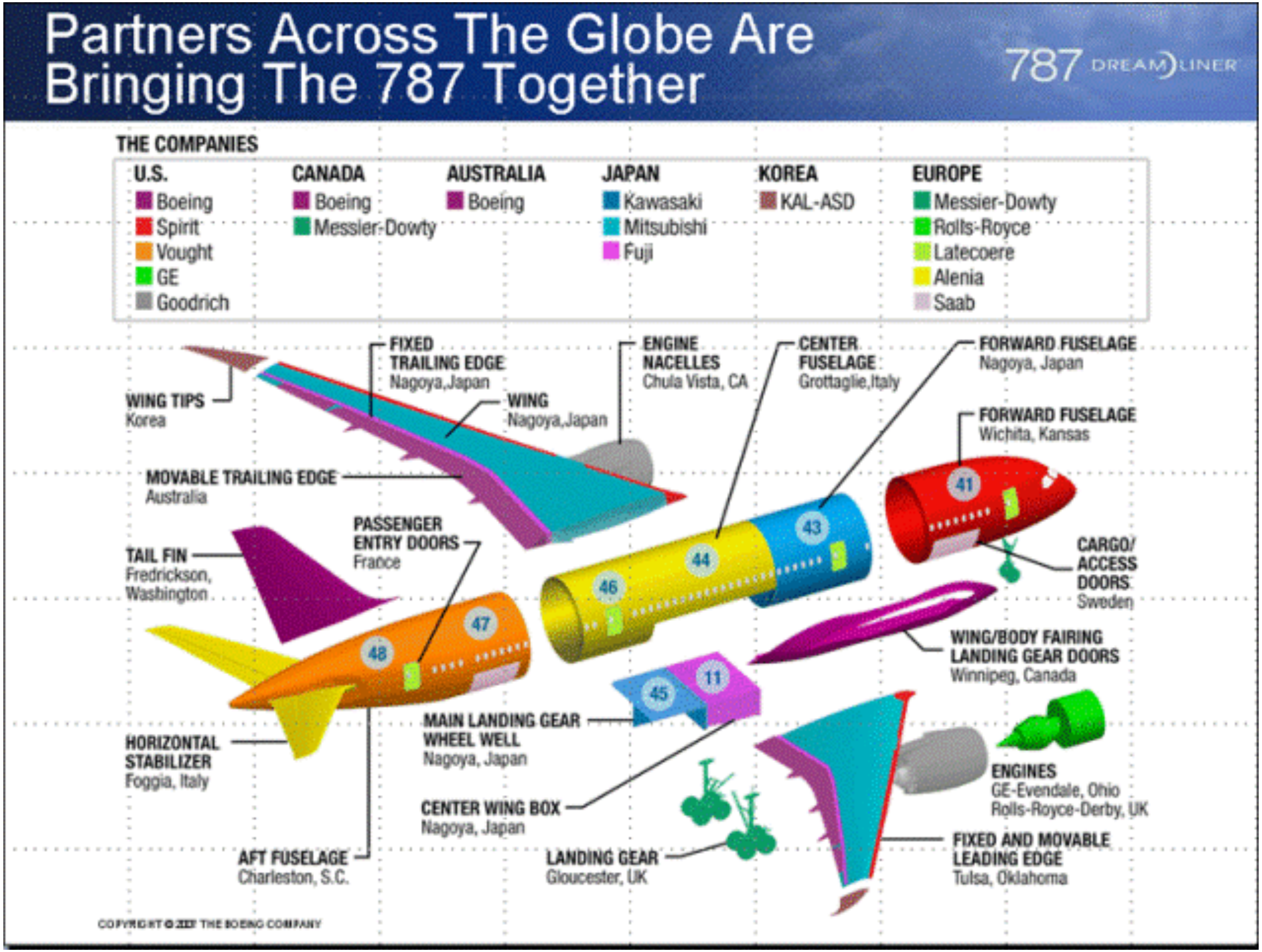
The Stock Management Structure

http://sinaslogisticsblog.blogspot.ie/2010/04/bullwhip-effect.html



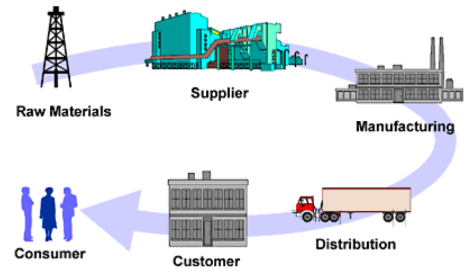


http://www.brightonsbm.com/news/sup ply-chain/



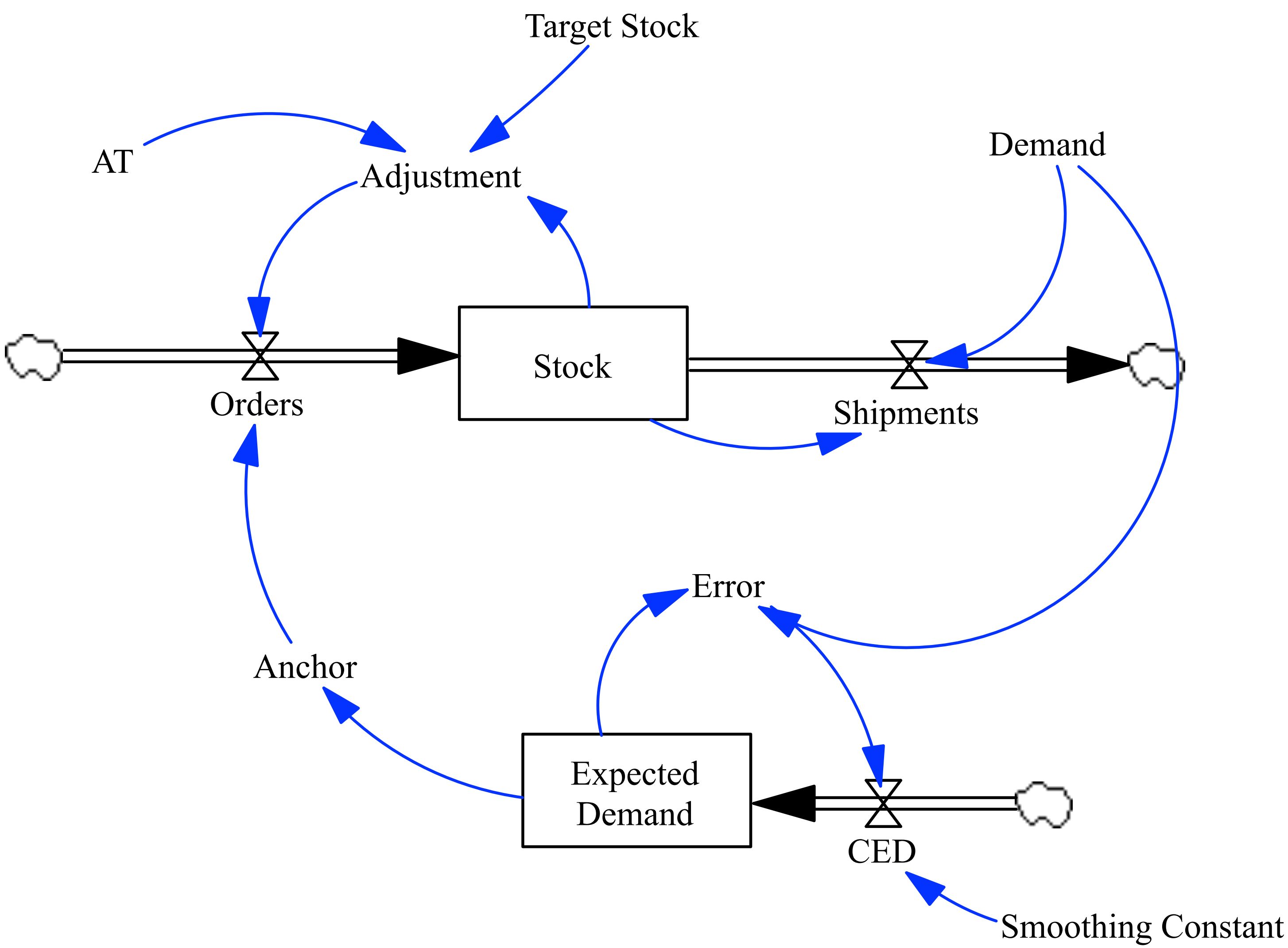
http://supply-chain-data-mgmt.blogspot.ie/2012/10/the-size-of-boeings-supply.html

Managing a Stock

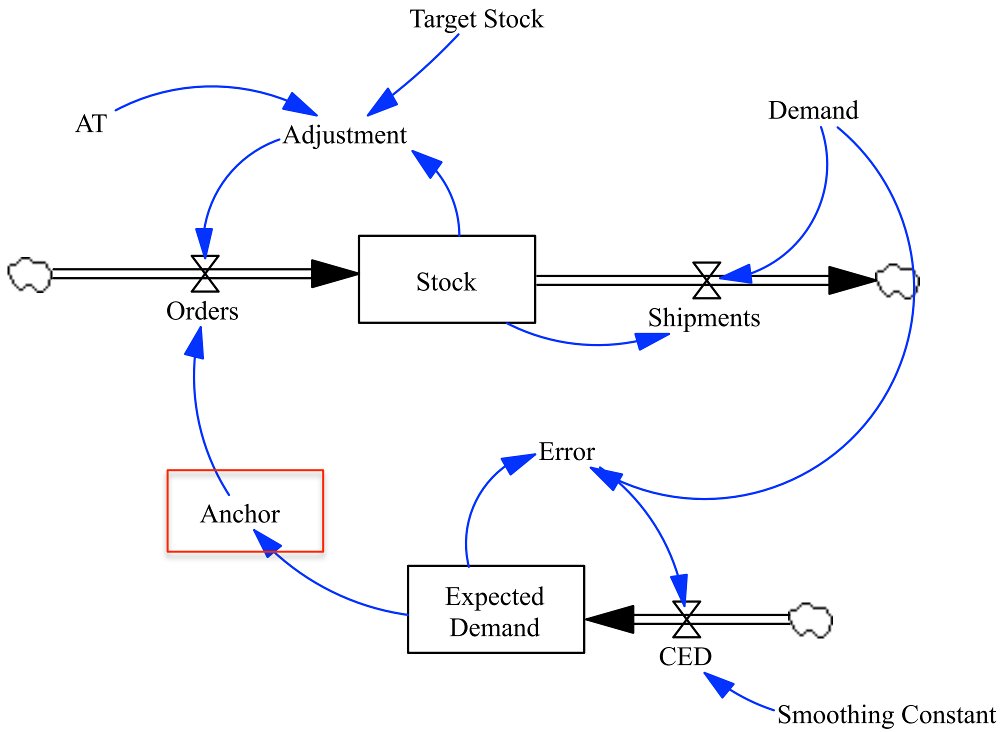


- Managers should replace expected losses from the stock **(the anchor)**
- Managers should reduce the discrepancy between the desired and actual stock **(the Adjustment)**
- Acquire:
  - more than the expected losses when the stock is less that the desired,
  - less than the expected losses when there is a surplus.

The Stock and Flow Model



Stock Management: The anchor



Anchor=Expected Demand

Expected Demand= INTEG (CED, 100)

CED=Error/Smoothing Constant

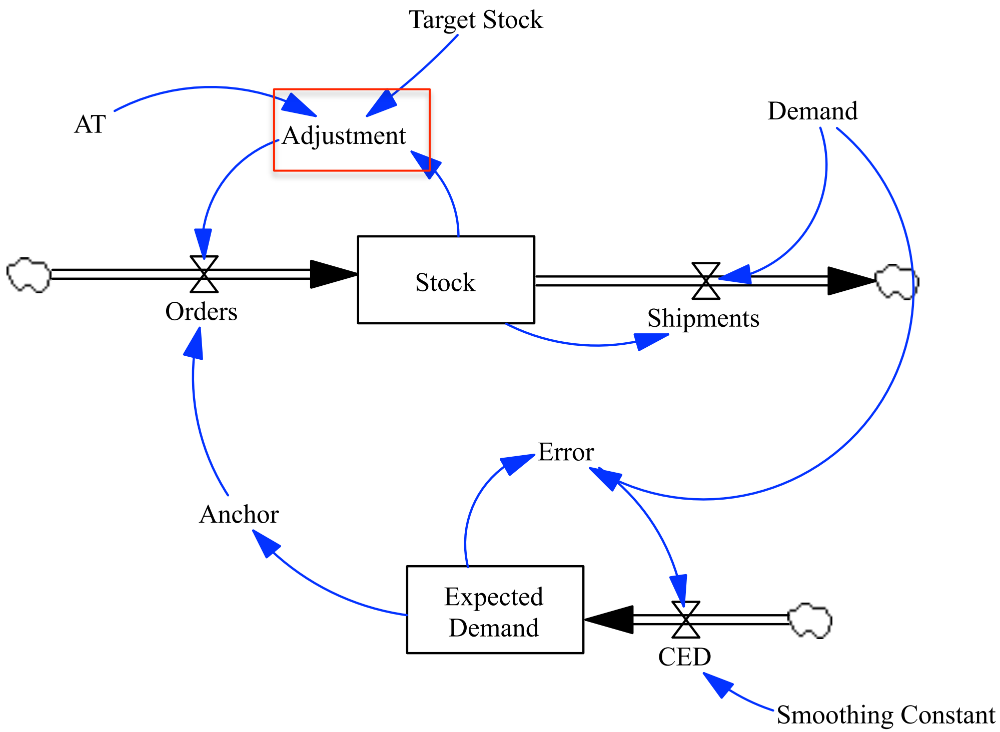
Error=Demand-Expected Demand

Smoothing Constant= 2

Demand=100+Step(100,10)-Step(150,20)

Managers should replace expected losses from the stock

Stock Management: The adjustment



Adjustment = (Target Stock-Stock)/AT

Target Stock=400

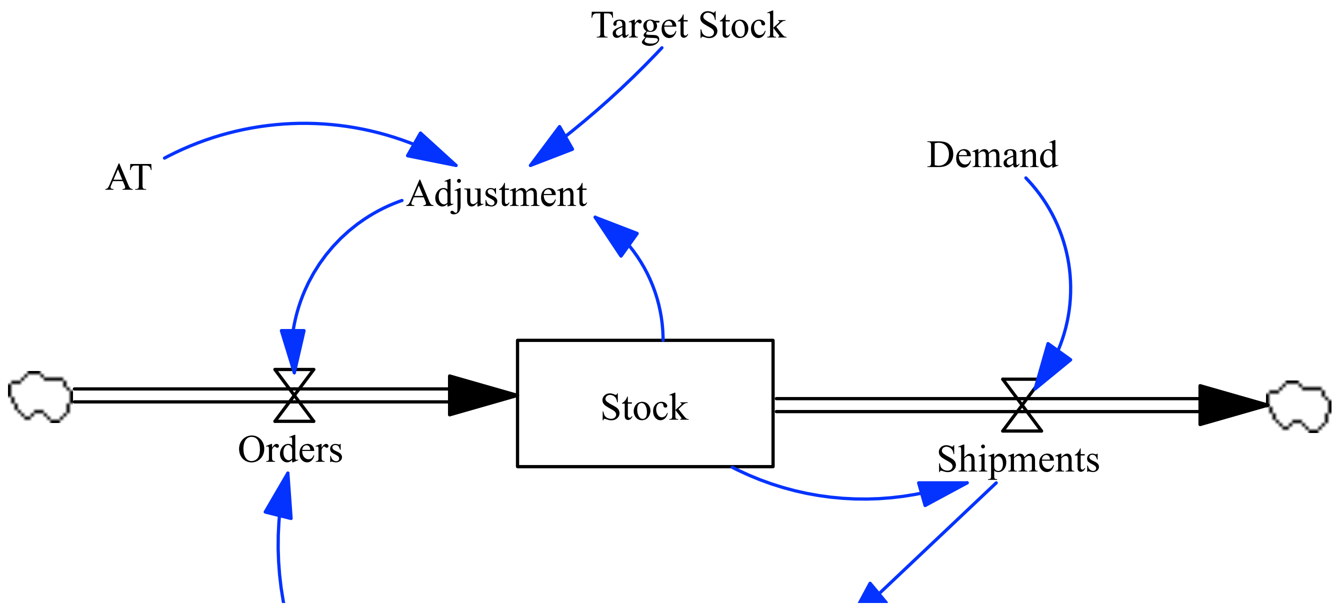
AT=1

Stock= INTEG (Orders-Shipments, 400)

Orders=max(0,Anchor+Adjustment)

Managers should reduce the discrepancy between the desired and actual stock

Inventory Management: The adjustment



Adjustment=(Target Stock-Stock)/AT

AT=3

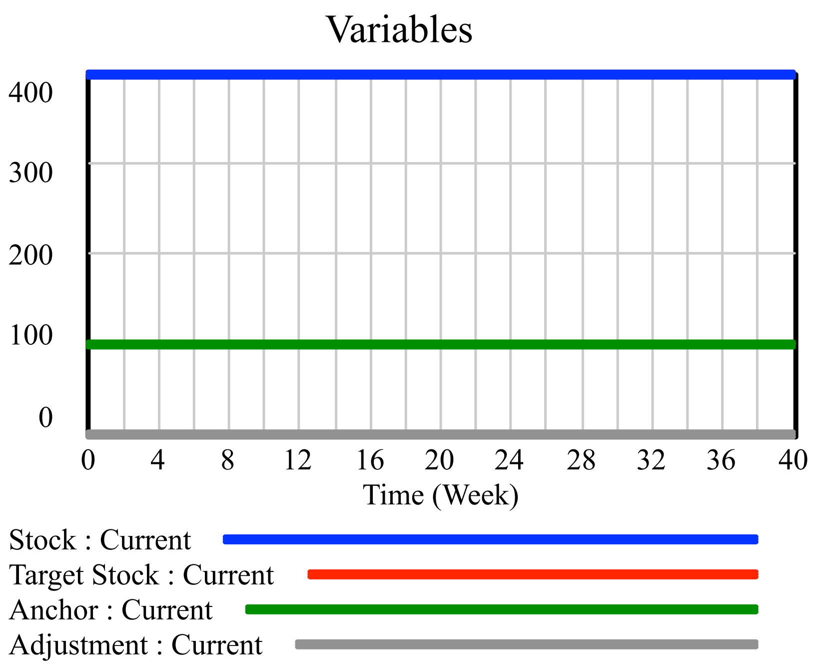
Demand=10+Step(10,20)

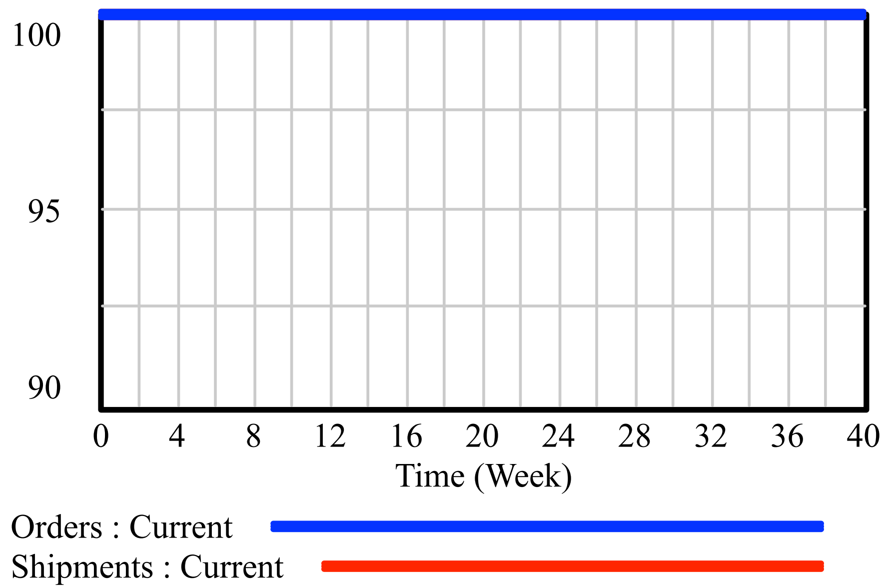
Shipments=min(Demand,Stock)

Stock= INTEG (Orders-Shipments, 50)

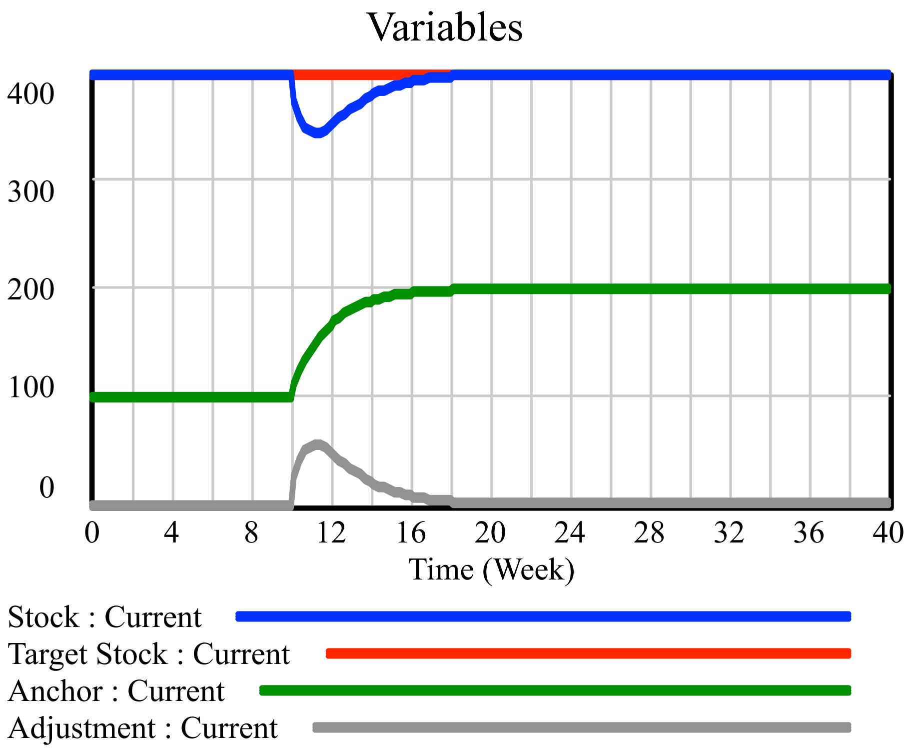
Target Stock=50

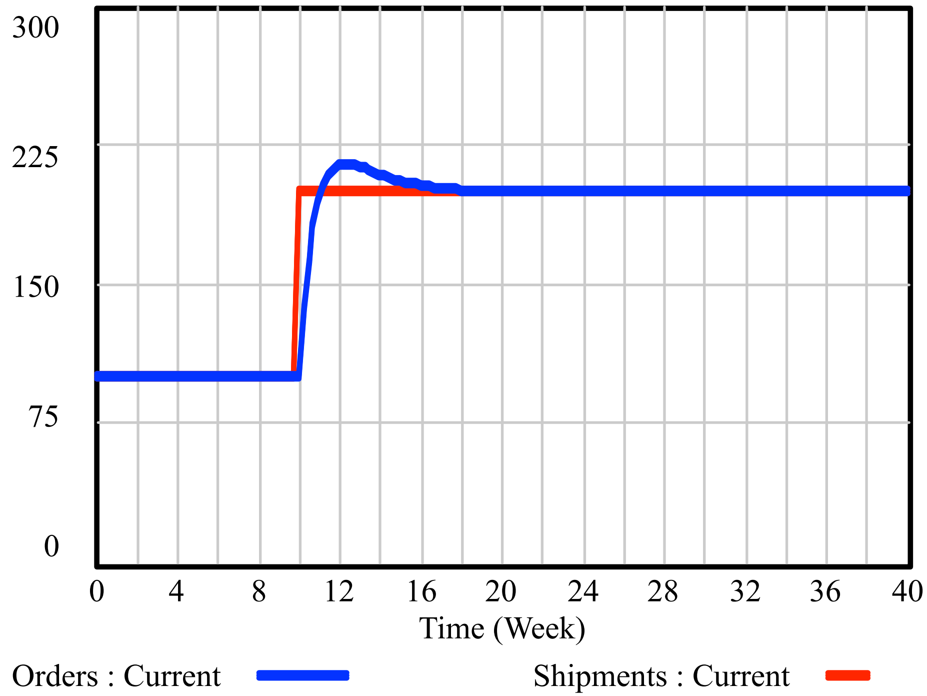
Equilibrium



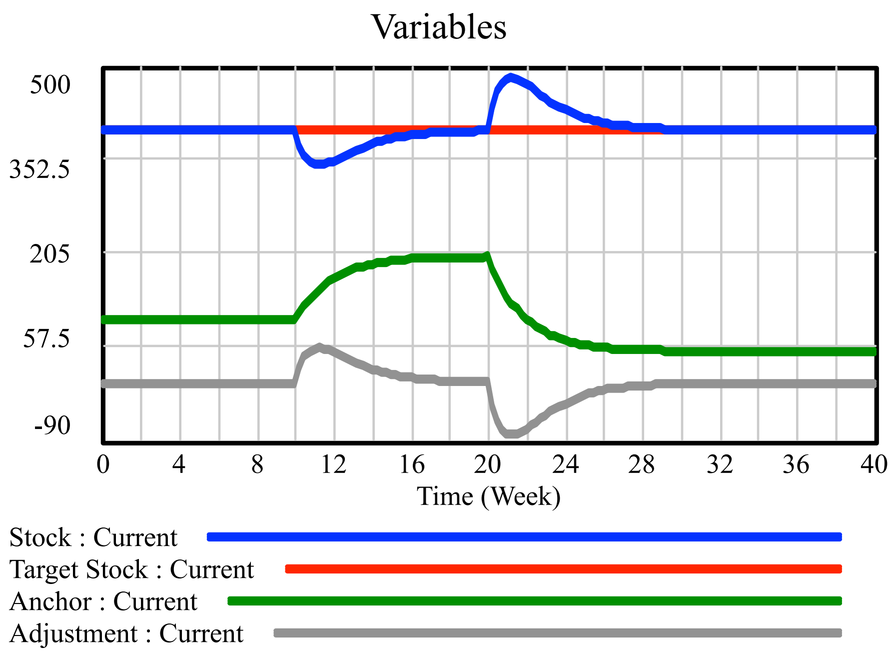


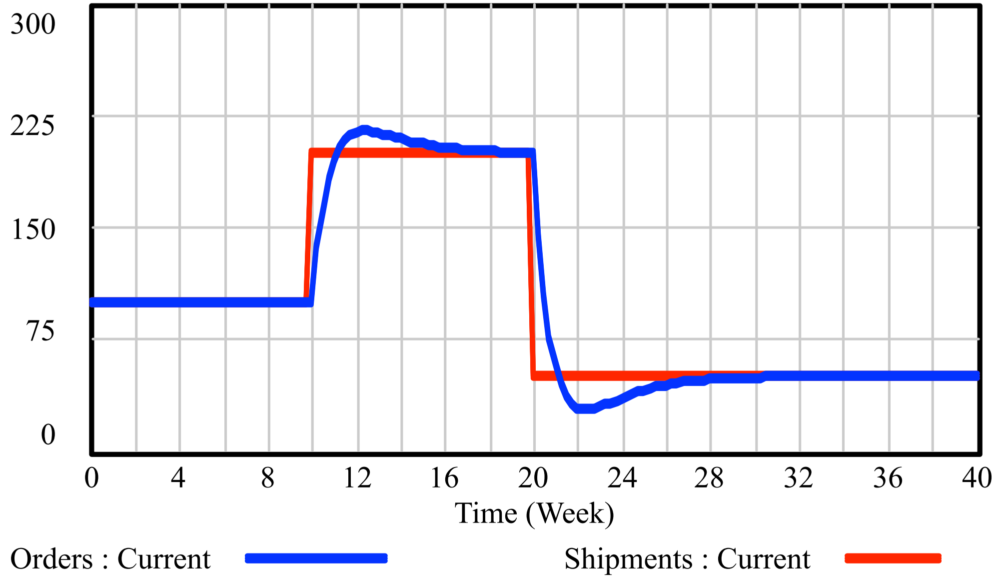
Demand = 100+Step(100,10)





Demand = 100+Step(100,10) – Step(150,20)





Challenge 9.2

Build a workforce (aging chain) stock and flow model for a software organisation. Employees are hired at graduate level, and from there they can stream into a programming career, or a consulting career.

There are three sequential levels of programmers: junior, senior and architect.

On the consulting side, there are also three sequential levels: junior, senior and partner.

The flow into the programmer stream is governed by a variable called *Programmer Fraction,* and the remaining graduates choose the consulting stream.

At any time, programmers may change to the consulting stream (to a similar level, for example, a junior programmer could become a junior consultant, but not vice-versa).

The model must allow for attrition at each stage (i.e. where employees leave the organisation). Formulate the model, with equations, and also provide dimensions for each variable.