



## **A handbook for applying system dynamics techniques in value chains: An application to pig value chains**

Kanar Hamza.

PhD candidate at the University of New England

Karl M. Rich




Head of Research and Strategy, Lab 863 Ltd., Edgware, UK  
Professor at the University of New England

CGIAR Research Programs on Policies, Institutions, and Markets and Livestock and Fish

**May 2015**



This publication is licensed for use under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported Licence. To view this licence, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>. Unless otherwise noted, you are free to copy, duplicate, or reproduce and distribute, display, or transmit any part of this publication or portions thereof without permission, and to make translations, adaptations, or other derivative works under the following conditions:

-  **ATTRIBUTION.** The work must be attributed, but not in any way that suggests endorsement by the publisher or the author(s).
-  **NON-COMMERCIAL.** This work may not be used for commercial purposes.
-  **SHARE ALIKE.** If this work is altered, transformed, or built upon, the resulting work must be distributed only under the same or similar license to this one.

## Contents

Overview.....	4
How to use this manual .....	4
About the authors .....	4
Introduction: what is system dynamics? .....	5
First principles of system dynamics: stocks, flows, and feedbacks .....	6
System dynamics and value chains.....	19
Pig Supply .....	20
Pig sales .....	26
Pig demand.....	33
Pig price setting.....	41
Model extensions – integrating external animal health influences to the value chain.....	67
The impact of animal health policy on the value chain .....	73
Vaccination policy .....	73
Quarantine policy .....	76
Policy costs .....	78
References .....	95

## **Overview**

This handbook is designed to provide practitioners, researchers, students, and policymakers with the analytical tools required to use system dynamics (SD) tools in the context of analyzing value chains. In recent years, system dynamics tools have been identified as a means of adding value to conventional value chain analysis. System dynamics can transform value chain analysis from a qualitative diagnostic tool to a quantitative planning tool that can provide users with insights on the economic and social returns to specific policy scenarios.

The handbook emerged from three training sessions administered by the authors during 2013-2014 in Kenya, Colombia, and Vietnam. It specifically follows the training course administered in Vietnam in November 2014 for the International Livestock Research Institute (ILRI) through the CGIAR Research Programs on Policies, Institutions, and Markets (PIM) and Livestock and Fish (L&F). While the focus of that course – and this handbook – is on pig value chains, the lessons and methods presented here are replicable in many other livestock and agricultural settings.

We have organized the handbook as follows. First, we provide the reader with an introduction to SD and its use in value chains. We then provide a step-by-step guide in the construction of a value chain model using the iThink/STELLA software of ISEE Systems (<http://www.iseesystems.com>). We note that other software platforms for SD could be used, such as Vensim, Powersim, or AnyLogic, and adding complexity to the model as the handbook progresses. A number of hands-on examples that build upon each other are provided. The handbook is written in an interactive, informal style, with lots of practical examples and solutions provided. The reader is encouraged to work through each of the examples and think through some of the extensions provided at different points of the text.

## **How to use this manual**

This manual is meant for the prospective user of system dynamics. It is not meant to be read passively from cover-to-cover but rather is aimed at interactive self-study. The reader will be expected to have a background in basic economics and mathematics, but does not require any specialized technical background. Indeed, one of the benefits of system dynamics is in its interdisciplinary nature that is open to scholars and practitioners with different disciplinary and technical backgrounds. What is required is a willingness to learn and experiment, and persistence in understanding the model concepts described in the text. After finishing this manual, the reader should have some confidence in their ability to develop more complex value chain models using this very rich field of modeling!

## **About the authors**

Kanar Hamza is a Ph.D. student at the University of New England in Armidale, Australia. His research interests are in the application of system dynamics tools in different biological systems and value chains (livestock, fisheries, forestry). He has worked as a consultant for ILRI and WorldFish Centre, and previously worked at the Norwegian Institute of International Affairs (NUPI). He holds Masters degrees from the University of Bergen in System Dynamics, Radboud University Nijmegen in Business Administration, and James Madison University in Integrated Science and Technology.

Karl M. Rich is Head of Research and Strategy at Lab 863 Ltd., a UK-based boutique research company conducting applied research in value chain analysis, animal health economics, conservation, and regional development. He holds adjunct professorships at the University of New England and Scotland's Rural College (SRUC). He has over 18 years of experience in value chain analysis, policy modeling, and animal health economics, and has previously worked at ILRI, NUPI, and the International Food Policy Research Institute.

## **Introduction: what is system dynamics?**

System dynamics (or SD) is a modeling and analytical paradigm invented during the mid-1950s at the Massachusetts Institute of Technology (MIT) by Professor Jay W. Forrester. SD was developed initially as a means to provide quantitative and mathematically grounded insights to problems arising in industrial systems. In its early stages, SD was mainly applied in the fields of urban and industrial dynamics, with a heavy emphasis on developing decision-making frameworks for the private sector. However, the use of SD has expanded in recent years and has been applied in various fields such as economics, public policy, environmental studies, defense, commodity cycles, management, etc.

The System Dynamics Society, the professional association of system dynamics practitioners, defines SD as follows:

*[SD is] a computer-aided approach to policy analysis and design. It applies to dynamic problems arising in complex social, managerial, economic, or ecological systems — literally any dynamic systems characterized by interdependence, mutual interaction, information feedback, and circular causality.*

SD is, at its heart, an approach to solving problems based on dynamic behavior in complex systems. SD practitioners develop models as a means of understanding the consequences of behavior given interactions and feedbacks between different actors and/or decisions. An SD model need not necessarily be quantitative – qualitative models based on causal loop diagrams (CLD) that map the behavioral relationships between different actors or actions often can be a powerful means of helping stakeholders understand how the system works in the first place (Vennix 1996). However, quantitative models using software such as Dynamo, iThink, Powersim, Vensim, and Anylogic are generally constructed as a means of quantifying the dynamics within complex systems and conducting scenario analysis.

SD models are a relatively new feature in the context of value chain analysis, although early models by Slater et al. (1969) and Harrison et al. (1974) in the context of the agricultural marketing system in South America demonstrated their potential well ahead of their time. Rich et al. (2011) provided insights and motivation as to why SD models were particularly relevant in livestock systems. First, given the biological dynamics of livestock production, many of which take some time to manifest, SD models represent a useful way of modeling livestock systems in a way conventional economic models (e.g., partial equilibrium, general equilibrium) do not or cannot. Second, value chains themselves are complex systems of diverse actors, each with different motivations and capacities, and where temporal and spatial phenomena are of particular importance. Conventional value chain analyses (Kaplinsky and Morris 2001) focus primarily on mapping and diagnosing a given value chain at a given point of time and do not take into account explicitly (or quantitatively) how value chains could evolve in the future. Put differently, value chains themselves are movies, not pictures. Because of this, we require analytical frameworks that address those dynamics, something that SD models do nicely. Third, SD provides a useful means of combining market dynamics with other dynamic phenomena, such as those arising from animal disease or the environment that potentially have important feedback effects.

In this context, a number of SD models of value chains have been developed in recent years, most of which have been livestock-focused. Examples include Rich, Perry, and Kaitibie (2009) in the context of beef exports from Ethiopia; Hamza, Rich, and Wheat (2014) in the case of Norwegian salmon; Hamza et al. (2014a,b) on smallholder livestock markets in Southern Africa; and Naziri, Rich, and

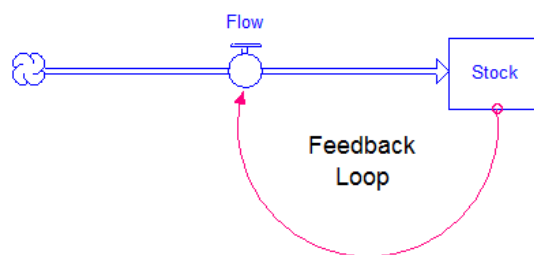
Bennett (2015) on beef exports from Namibia. There is currently ongoing work at ILRI using SD in the context of pig value chains in Vietnam, beef in Botswana, and dairy in Tanzania. SD methods are also being used in the context of analyzing dairy value chains in Nicaragua at CIAT.

### First principles of system dynamics: stocks, flows, and feedbacks

The central concepts of the SD paradigm are stocks, flows, and feedback loops. Later in the handbook, we will discuss the concept of delays, which play an important role in mediating the behaviors we predict in our systems.

*Stocks* represent the state of the system, such as the population of a country, water level in a reservoir, state of knowledge, etc. Put simply, stocks represent “stuff” at any given period of time. For example, a cup of coffee would be a stock, with the amount of coffee in the cup the value of the stock at this moment in time. *Flows* represent how the state of nature changes over time. In our coffee example, if I pour more coffee into the cup, there will be an inflow of coffee into my stock (the cup). If I drink from the cup, there will be an outflow of coffee from one stock (my cup) into another (my stomach!). Flows thus express the entry or exit of things to or from a stock. These could include, for example, the inflow of people into a country through births (birth rate) or the outflow of people through deaths (death rates). Neither stocks nor flows are required to be physical goods – intangible things like knowledge (a stock) or changes in the rate of knowledge acquisition (a flow) could also be modeled.

Stocks cannot change by themselves. Only flows (inflows and/or outflows) can change the level of a stock at any period of time, which are affected by other variables or technical relationships known as convertors and/or other stocks. In a similar vein, flows themselves are governed by *feedback loops*. Feedback loops represent circular connections among different elements of a model that influence the evolution of the system over time. Figure 1 shows a graphical representation of the central concepts of SD based on the software in the iThink software from which we base our models and instructions<sup>1</sup>.



**Figure 1: Graphical representation of stock, flow and feedback loop**

At this point, it is instructive to step back for a moment and first review the building blocks of SD using iThink so that you can follow along with the exercises that follow. To begin, first create a new model in iThink (go to File-New in iThink in either Windows or Mac). You will see a white, blank page and will notice a series of icons on the upper part of the screen; these represent the major tools we use to build our model. In figure 2, we provide a screenshot of the toolbar for iThink to help you follow along. The first icon (reading from left to right) is a rectangle shape that denotes stocks. If you click on the box, you can drag it to the blank screen; when you release, you will see a stock named

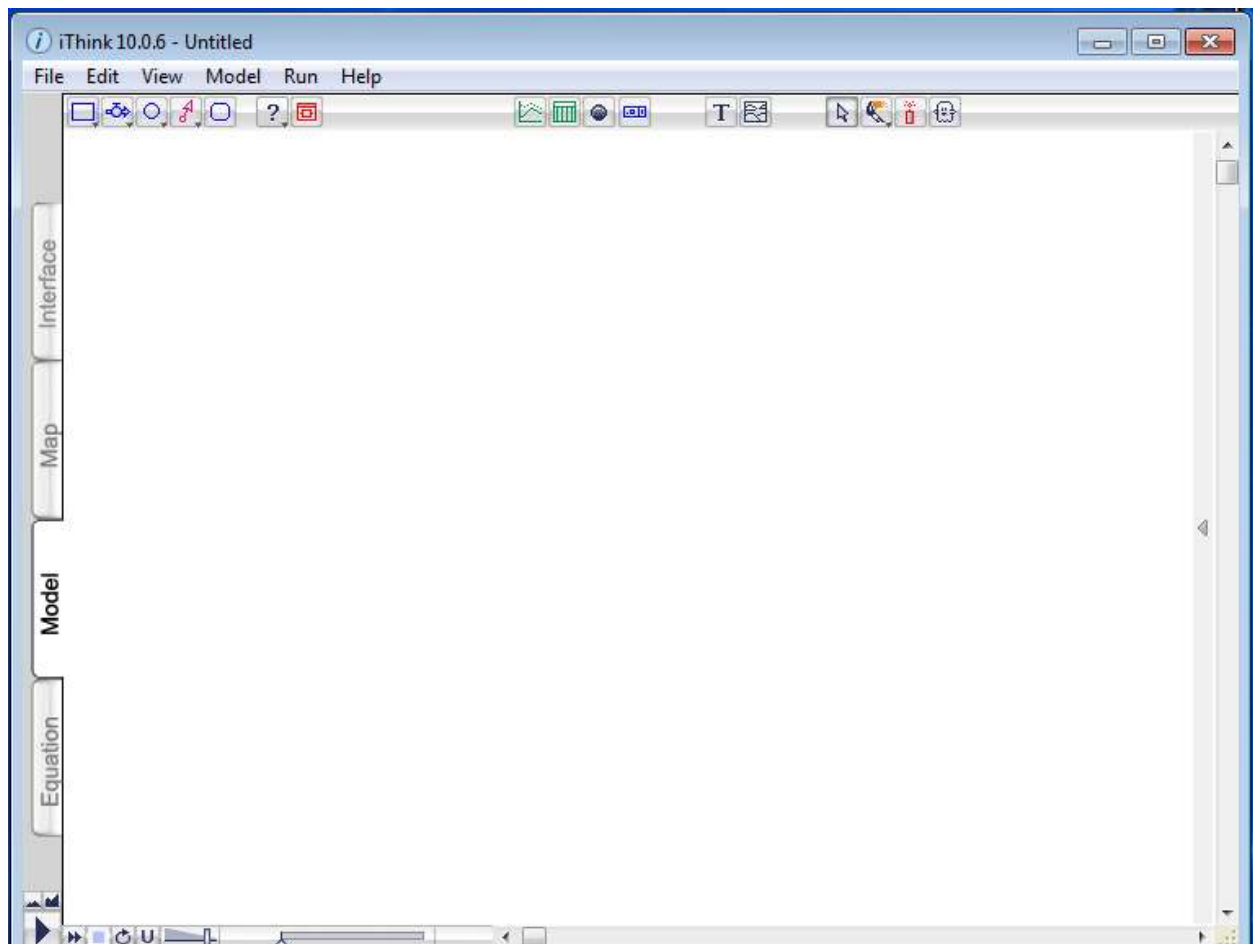
---

<sup>1</sup> We refer to iThink throughout the text, but note that our handbook instructions apply both to iThink and STELLA. Both are produced by ISEE Systems and are very similar, though there are slight differences in the applications in which they are used. For the purposes of this handbook, there is no functional difference between either software package or the commands mentioned. Note that other software systems use similar representations of stocks and flows, although the visualization and command structure are slightly different.

“Stock 1”. If you click on the text above the stock (not the stock itself), you can rename it. In order to be consistent with figure 1, let’s rename this as “Stock”.

To the right of the stock on the toolbar is a thick arrow – this denotes flows. You can drag this to the screen as with stocks, but remember to stretch it all with the mouse to show it. To connect the flow as an inflow to a stock, drag the “cloud” that is at the end of the arrow to the stock. As with stocks, you can rename flows; to be consistent with figure 1, change the name of this flow to “Flow”. If you wished to connect an outflow from a stock, you would drag the cloud at the beginning of the flow to the stock.

Continuing along the toolbar, the circular shape represents convertors. These are various technical parameters that mediate flows or can be used to create new convertors. Consider the coffee example from earlier. How much I drink depends on the speed (or rate) at which I do every time period. We could define this rate as a convertor that is connected to the flow (see the next section for more details and an example on this). But how do we connect convertors or other elements? This is where we use connectors, which is the red skinny arrow next to convertors. To complete figure 1, click on the connector icon and place it on the stock. Then, while holding the mouse or trackpad, drag it to connect to the flow.



**Figure 2: Screenshot of a blank iThink modeling frame**

You will notice other elements on the toolbar; we will discuss these later in the handbook as their use arises. In addition, on the left hand side of the screen, you will see four tabs: Interface, Map, Model, and Equations. The Interface provides a platform for model communication. The Map is just that – it shows a diagram of what your model looks graphically. The Model tab is where you do your

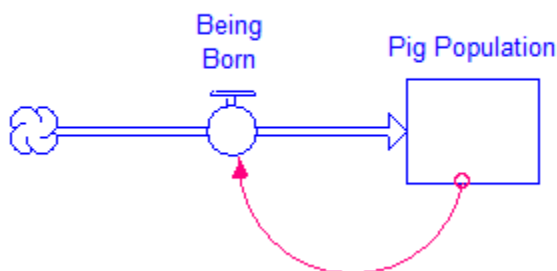
modeling and it tells you whether relationships have been defined. If you click on the Model tab for your model, you will see a “?” sign in the stock and flow. Finally, the Equation tab keeps track of your model definitions. As we will see shortly, modeling in iThink is conducted using graphical icons which mathematical relationships between stocks, flows, and convertors defined and stored in the Equations tab. For the purposes of the exercises that follow, you should be sure to be working in the Model frame.

Going back to figure 1, you will observe that information about the level of the “stock” is passed to the “flow” through a connector. This indicates that the “flow” rate depends on “stock” level. Similarly, the direction of “flow” is headed into the “stock” as an inflow. In figure 1, you will see that there is a circular connection defined here i.e. we start from “stock”, go through “flow” via a connector, and then back to the starting (“stock”) point through “flow”. This is an example of a feedback loop.

There are two types of feedback loops:

- *Reinforcing (R) or positive feedback loops* are self-reinforcing loops, in which an increase (decrease) in one element leads to an increase (decrease) in another. For instance, in the absence of any other flows, as population grows, more people are born, which leads to more people in the future, and more births, and so on.
- *Balancing (negative) feedback loops* are self-correcting loops, in which an increase (decrease) in one element leads to a decrease (increase) in another. For instance, in the absence of any other flows, as populations grow, there will be more deaths, leading to a reduction in the population in subsequent periods.

To graphically represent balancing and reinforcing feedback loops, let’s modify the names of stocks and flows of a simple stock and flow structure shown in figure 1 to represent a simple pig production model that includes “pig population”, and “being born”. In iThink, rename your stocks and flows accordingly. You should note that “Pig Population” is your stock (it is the number of pigs), while being born is the inflow of new pigs into the system. Your result should look like figure 3 below.

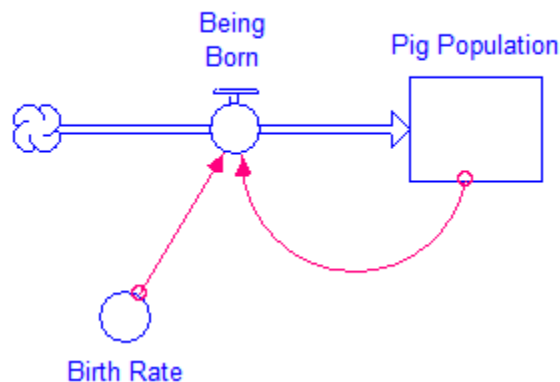


**Figure 3: Modified stock and flow structure to represent a simple pig population model**

As noted earlier, we defined a converter as an independent parameter or variable. Converters either directly influence flows (i.e. an independent parameter) or take inputs from other elements of the model to convert it to outputs that influence other sections of the model (whether another flow or other converters before reaching flows). The former case represents an exogenous parameter (independent parameter) while the latter represents a variable (dependent variable). To introduce converters to the simple model structure shown in figure 3, let’s add “birth rate” to the model, which we define as the (fixed) percentage of the population that is born every period. In iThink, go to the circular icon and drag a converter to the modelling frame, and name it “Birth Rate”. Then, drag a connector to connect “Birth Rate” to “Being Born”. This shows that the flow of “being born”



depends on the number of pigs in the farm or region (i.e. “pig population”) and pig productivity (i.e. “birth rate”).



**Figure 4: Expanding the simple pig model by adding a converter (birth rate)**

The model shown in figure 4 represents a qualitative model. To operationalize this model for simulation we need to add initial values and equations to the stock and flow structure. We do this as follows. We will first assume that this model runs on a weekly basis (we will define this explicitly in the model shortly). Let’s then define our stock, “Pig Population”. By definition, the stock is the state of the system. However, we need to first insert an initial value that represents the state of the system at time=0. This initial value can be an exogenously defined number, a converter from the model, or an equation. To define this stock in iThink, double click on the stock. You will get a diagram box to insert an equation or number for the stock. Place the value of 10,000 in the box and press the return key twice. Then in curly brackets, type {pigs}. What this element in the curly brackets denotes are the units of the stock, which in this case is pigs.<sup>2</sup> When you are finished, click on the green check icon.

Next, let’s define the converter “Birth Rate”. As “Birth Rate” is a constant parameter and is a percentage, we can insert any value between zero to one. Zero means no growth, while one represents a growth rate of 100%. We enter a value for “Birth Rate” in the same way – double click on the convertor and add a value. Our value will depend on the time period in our model. If we run a weekly model, for example, we would need to take an annual rate and convert to weeks. So, for example, if the annual birth rate is 3% per year, the weekly rate would be 3%/52. Thus, let’s first enter the value 0.03/52, hit the return key twice, and enter {1/week}. This denotes that the unit for “Birth Rate” is per week.

What about inflows? The inflow “being born” depends on both the stock of “pig population” and the parameter “birth rate”. This should be intuitive – the change in the number of pigs between time  $t-1$  and  $t$  will be the product of the previous stock of pigs at time  $t-1$  and the birth rate. Thus, because this flow has a functional form, we need to define an equation for “being born”. In the equation box, you will notice that in the upper left hand of the box, there is a list of “required inputs,” which in this flow are “Birth rate” and “Pig Population”. That alerts us to the fact that we have to use both of these elements in our definition. As noted, the entry of new pigs will be the product of the birth rate and the current population, so we define our equation as in equation (1), i.e.

$$\text{Being born} = \text{Birth rate} * \text{Pig population} \{ \text{pigs/week} \}^3 \dots\dots (1)$$

<sup>2</sup> It is good modeling practice to label the units of your model. It is not necessary to run the model, but it is an extremely useful way to check the consistency and validity of the dynamic relationships defined.

<sup>3</sup> It should be noted that the unit of each element of the model must be an outcome of the equation used. In this case, we multiplied pig population (the unit is pigs) by the birth rate (the unit is 1/week) and therefore the

Based on this simple model structure, consider the following exercise:

**Exercise 1:**

- 1- Based on equation 1 and the assumed values above, what sort of behavior would you expect for the stock “pig population” based on the model shown in figure 4 to generate (for example, linear, exponential, etc.)? Why? (Hint: Please remember the description of stocks and flows provided above).

-----

- 2- What type of feedback loop exists in the model shown in figure 4? Why? (Hint: please remember the definition of both reinforcing (R) and balancing (B) feedback loops provided above).

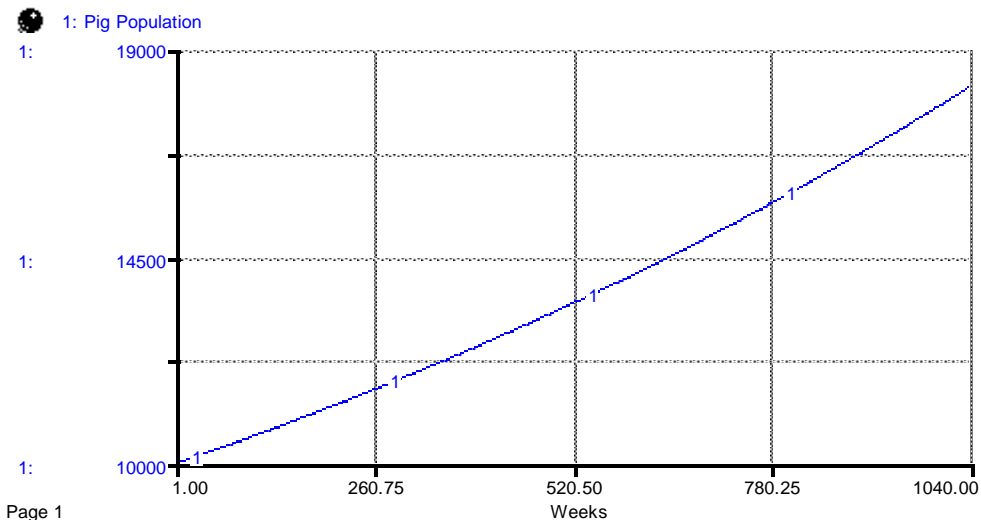
-----

**Solution 1:**

- 1- We should observe that this model exhibits exponential growth. In this model, recall that we only have an inflow into the stock “pig population” (figure 4). This inflow depends on the previous value of the stock and a positive birth rate, which means that the stock of “pig population” can only increase over time. The exponential pattern of behavior arises because each week, as the stock of pigs rises, the inflow (rate of change) of new pigs in the next period increases, generating an even larger cumulative stock of pigs, and an even faster rate of increase in the growth rate of pigs in the next period, and so on. If we run the model (more on this later), we would get the behavior as shown in figure 5 below.

---

unit of the flow is pigs/week. This is an important model validation procedure that must be satisfied to increase one’s confidence in any SD model. More on model validation will be presented as we progress.



?

**Figure 5: Behavior of “pig population” stock in our initial population model**

- 2- The feedback loop is a reinforcing loop. Recall figure 4. In this question, it is a reinforcing feedback loop because more “pig population” leads to more pigs “being born” which leads to a higher “pig population” and so on in a reinforcing manner.

Figures 4 and 5 show the structure and behavior of a simple pig production model, respectively. However, this model is quite simplistic. Thus, we need to expand the model boundary to make it more consistent with real life physical systems. For example, the behavior of the simple model that is shown in figure 4 indicated exponential growth of the pig population and assumed no sales or deaths, for instance. This is clearly unrealistic.

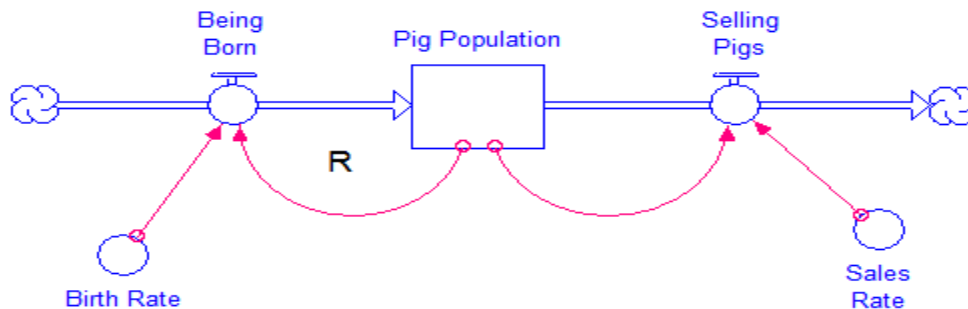
To solve this issue, let’s expand the model boundary by including a new structure to denote the sales of pigs to the market. Consider two new model elements, “sales rate” (i.e the percentage of the “pig population” that is sold over time) and “selling pigs” (i.e. number of pigs shipped to customers over time) to the market. How would we add these to our model? See exercise 2

**Exercise 2:**

- 1- Would we define “sales rate” as a stock, flow, or a parameter? Why?  
-----
- 2- Would we define “selling pigs” to the market as a stock, flow, or a parameter? Why?  
-----
- 3- Can you add the new elements of 1 and 2 above to the model structure shown in figure 4? Please use the space below to add model structure (be sure to do this in your model in iThink too!). Explain your reasoning for adopting this structure.  
-----

**Solution 2:**

- 1- "Sales rate" is a **parameter** (or an independent parameter or a constant) because it represents a fractional sales rate (that is, what proportion of "pig population" is sold per each time unit).
- 2- "Selling pigs" to the market is a **flow** (it is an outflow since it takes pigs out of the stock) because it denotes the number of pigs that are sold to the market over time (that is, pigs per week). As an aside, it is good modelling practice to define flows as action verbs (ending in -ing) to denote their dynamism over time, while stock and parameters are defined as nouns.
- 3- Our modified figure can be found in figure 6 below.



**Figure 6: An expanded pig population model**

SD models are generally built step-by-step. It is highly recommended that the modeler run the model each time after adding more structure to ensure that the modeler understands the model's behavior and whether the model and its behavior are consistent with real life conditions. This is also an important factor to decide whether we need to add more structure to the model or not (i.e. whether to challenge or expand the model boundary or not). Afterwards, we test different parameter values and evaluate model results based on the expected outcomes to reveal any structural flaws in the model.

Before we do these tests, let's first add some figures and equations to the added structure from figure 6. We will define sales rate at the same level as birth rate i.e. 0.03/52. For the outflow "selling pigs," we will have the same relationship as before, namely that this outflow will depend on the stock of pigs multiplied by the fractional rate of sales, or "sales rate". Enter the following equations for (2) and (3) into your model for "Sales rate" and "Selling Pigs," respectively, and complete exercise 3 below.

**Sales rate** = 0.03/52 {1/week} ..... (2)

**Selling pigs** = Sales rate \* Pig population {pigs/week} ..... (3)

**Exercise 3:**

- 1- Based on equations 1, 2, and 3 what sort of behavior (for "pig population") you expect the model shown in figure 6 to generate (for example, equilibrium, growth, decay, etc.)? Why?

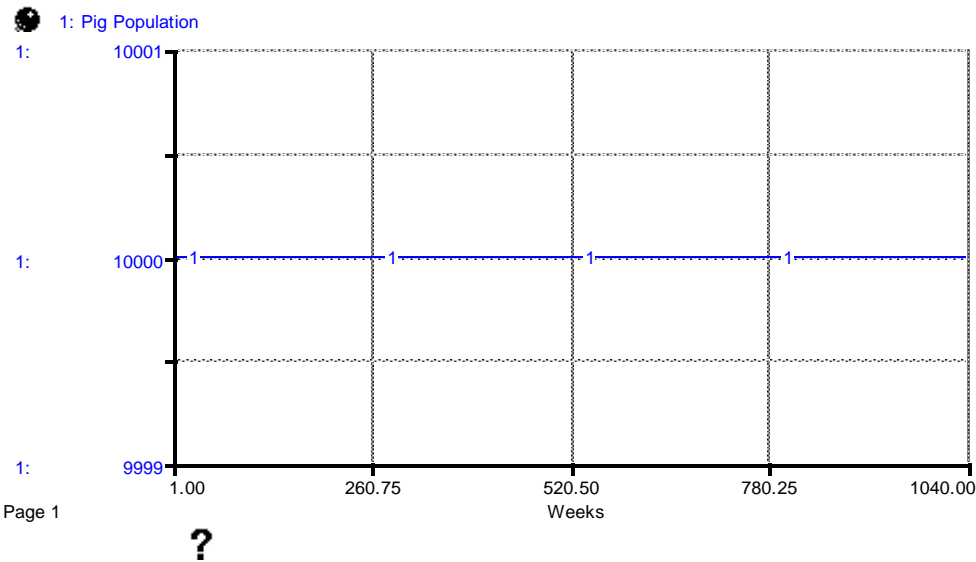
-----

- 2- What types of feedback loops exist in the model shown in figure 6? Why?

-----

**Solution 3:**

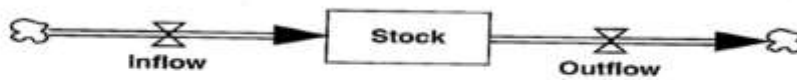
- 1- In this model, we would expect to observe equilibrium behavior in the stock “pig population”. This is because the inflow and outflow rates are the same, since the fractional rates of births and sales are the same. As stated earlier, stocks accumulate the differences between inflow and outflow. Thus, when the inflow equals the outflow, the stock remains unchanged (i.e. equilibrium). Figure 7 shows the model’s behavior under the initial assumptions.



**Figure 3: Behavior of the “pig population” stock of the model shown in figure 6 under initial assumptions**

The stock and flow structure is a graphical representation of SD models. That is, SD modelers use graphical representations to build any model. However, behind these graphics is an underlying mathematical structure in the background of the graphical representation of SD models. Figure 8 shows the standard graphical stock and flow structure and three alternative identical expressions of representing stocks and flows. In the hydraulic metaphor, the bathtub represents the stock and the faucets represent inflows and outflows. In the example above, we could interpret this that the water inflow is exactly the same as the water outflow, such that the level of water in the bathtub remains the same. Similarly, both the integral and differential equations in figure 8 mathematically formalize the stock and flow relationship.

Stock and Flow Diagram:



Hydraulic Metaphor:



Integral Equation:

$$\text{Stock}(t) = \int_{t_0}^t [\text{Inflow}(s) - \text{Outflow}(s)]ds + \text{Stock}(t_0)$$

Differential Equation:

$$d(\text{Stock})/dt = \text{Net Change in Stock} = \text{Inflow}(t) - \text{Outflow}(t)$$

Figure 8: Equivalent representation of a stock and flow structure (Source: Sterman 2000)

- 2- The new feedback loop is a **balancing loop**. Please note that when the direction of flows goes outside of the stock, this indicates that this flow negatively affects the stock (i.e. it takes material out of the stock). In this question, it is a balancing feedback loop because as “pig population” rises, this leads to an increase in “selling pigs” which leads to a lower “pig population” and so on in a balancing manner.

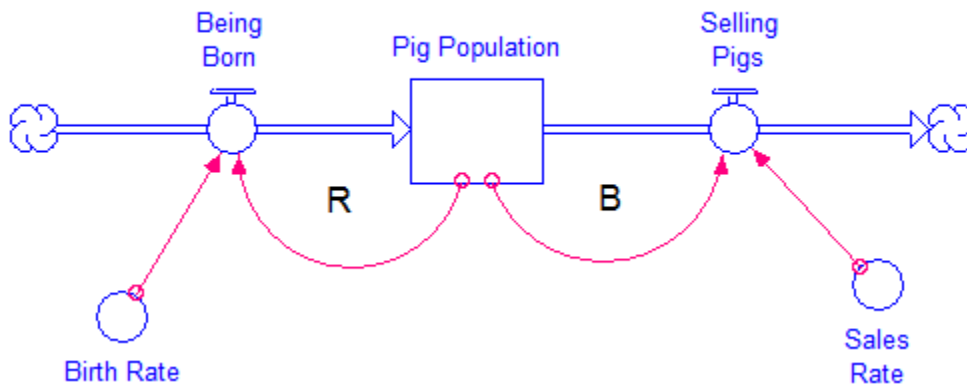


Figure 9: Feedback loops in the expanded pig population model

**Exercise 4:**

- 1- Please adjust the birth rate to read

**Birth rate** =  $0.01/52$  {1/week} ..... (4)

Assuming the rest of the model remains unchanged, how would you expect the model behavior shown in figure 7 to change in terms of the population of pigs? Could you draw a graph to show the pattern of behavior in the pig population (for example, would you expect to see exponential growth, linear growth, exponential decay, linear decay, or something else)? Hint: please refer to figure 8 to see how changes in flows affect the model to answer this question.). Plot time on the X-axis and pig population on the Y-axis for this question and subsequent ones.



- 2- Return the birth rate in question 1 to its original value ( $0.03/52$ ). Now, please adjust equation 2 (sales rate) to

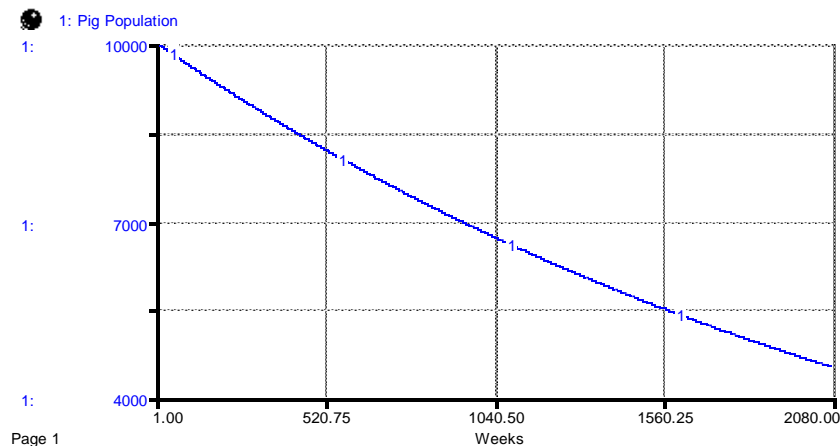
**Sales rate** =  $0.01/52$  {1/week} ..... (5)

Assuming the rest of the model remains unchanged, how would the model behavior shown in figure 7 change? Could you draw a graph to show the expected pattern of behavior (for example, exponential growth, linear growth, exponential decay, linear decay, or something else)?



#### Solution 4<sup>4</sup>:

- 1- The observed model behavior is **exponential decay** because in this particular case the inflow “birth rate” is lower than the outflow “sales rate”. This leads to a depletion of the stock “pig population” in a decreasing manner. Think of this using the hydraulic metaphor in figure 8. If more water leaves the bath tub than water comes in, the level of water in the bath tub decreases.

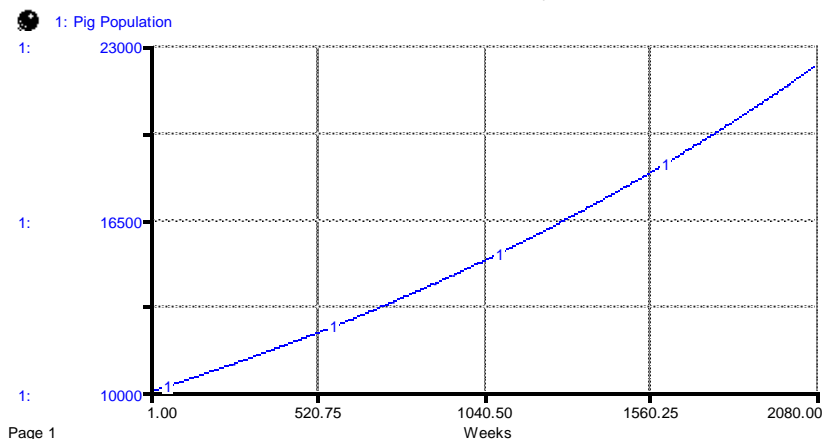


Page 1

?

**Figure 10: Updated model behavior by changing the birth rate inflow**

- 2- The expected model behavior is **exponential growth** because in this particular case the inflow “birth rate” is higher than the outflow “sales rate”. This leads to an increase in the stock of “pig population”. Again, use the hydraulic metaphor in figure 8. If more water comes into the bath tub than water exits, then the level of water in the bath tub increases.



Page 1

?

**Figure 11: Updated model behavior by changing the sales rate outflow**

So far, we have gone through a simple example of the stock and flow structure and different types of simple model behavior in SD in an intuitive fashion. However, as we will be building more complex models of value chains shortly, it is good at this stage to demonstrate how we translate this intuitive analysis into the running of such models in iThink itself.

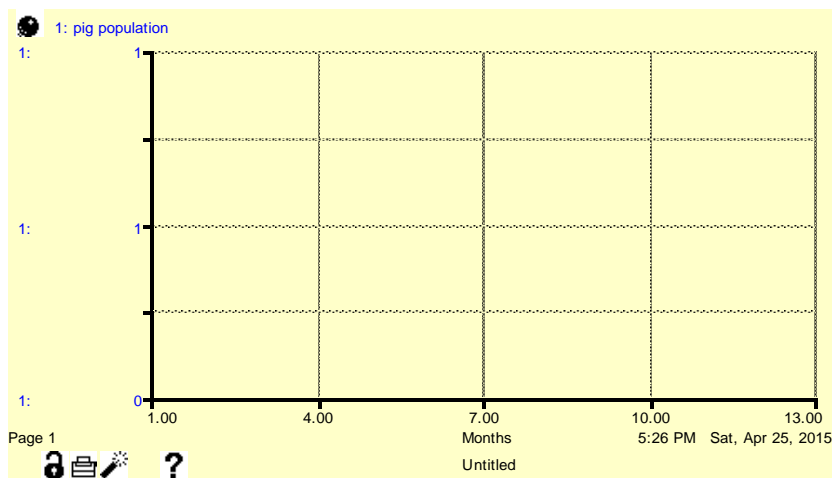
---

<sup>4</sup> Please note that we extended the time horizon on x-axis of figures 10 and 11 to make sure that the pattern of behaviour is clearly visible.



Let's start with the model developed above in figure 6, based on the equations (1)-(3) above. Using the instructions given earlier on inserting stocks, flows, converters, and equations into iThink, reproduce figure 6 into iThink. When all model elements have been defined (in the Model frame, there should be no more "?" signs in your icons.

You will want to see your results from your model, and in iThink you can generate both graphical and tabular forms of model outputs. To create a graph, on the icon toolbar at the top of the modeling frame, you will see a green line graph icon. When you click on this and drag to the modeling frame, a blank graph will appear (see figure 12 below). To add things to this graph, double click on the blank graph. You will obtain a box entitled "Define Graph" which allows you to choose stocks, flows, and converters from the "Allowable" window on the left side of the box. Highlight "pig population". That will highlight the ">>" button and allow you to move this stock to "Selected". Then click OK. You will see "pig population" on the y-axis of your newly created graph.



**Figure 12: A blank graphical output in iThink**

To run SD models in iThink, you can either click on the play arrow (▶) on the bottom-left corner of iThink model (see figure 2) or via the task list above the tool bar by clicking on "Run" and then selecting the first choice ("Run") on the list. You should obtain the same results as in figure 7. Repeat Exercise 4, changing the values of birth and sales rates to obtain the graphs reproduced in figures 10 and 11.

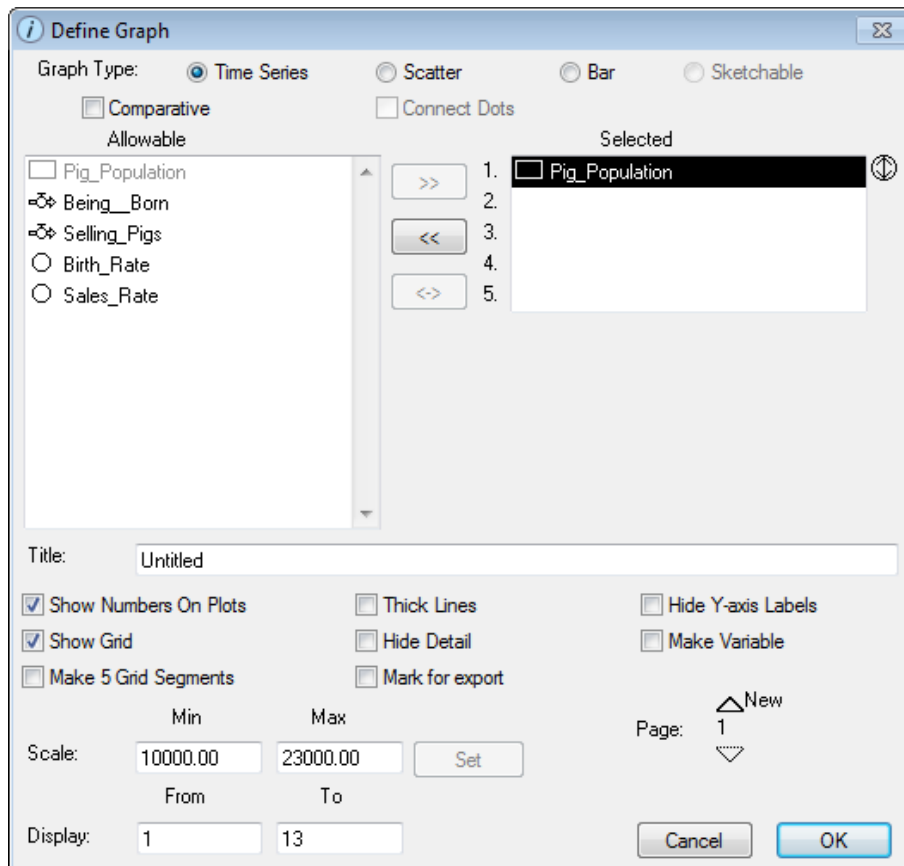
Under the "run" menu, you will find other elements, one of which is called "Run specs". Run specs is the seventh element on the list under the run menu. The "Run specs" menu is important because here you can change various model settings, such as the simulation time unit (day or week or month etc.), simulation length (from week X to week Y), and DT in the box that appears after you select run specs.

While simulation time units and length are pretty straightforward and based on the system the user defines, the concept of DT is not. DT is defined as the frequency by which the model updates itself. A DT of 1 for a weekly model, for example, means that the model is a discrete-time model that updates every week. The smaller and smaller DT gets means that we increase the precision and approximation to continuous time. The default DT setting of iThink model is 0.25. You can change DT as needed but a very low DT (for example 0.0007) needs to be justified. As a rule of thumb, DT must be equal to or lower than one-quarter of the shortest time delay found in the model (we will discuss delays in the next section). DT has significant impacts on SD models and must be selected carefully. A good test of DT selection is to re-run your model with a lower DT to see if the pattern of model parameters is the same as with your chosen DT. If your model behavior changes significantly (for

example, oscillation changes to linear or exponential growth or decay, etc.), this indicates that your selected DT is not precise. In this case, lower the DT value by half and run the model until the behavior remains about the same as when you lowered DT to half again. When the pattern of behavior does not change when you lower DT, this means your model DT is appropriate.

Note: the default setting on iThink for time unit is months, the simulation time horizon is 13, and DT is 0.25. You can change these to different values and units under run specs as we explained earlier. Similarly, when you use the graph icon to create a graph (as you did it in figure 12), the default setting of the graph is the same as your model default setting in terms of the time horizon and units (month or week or day). However, as you can see, there are more details on the figure you created in figure 12 compared to the other figures presented in this manual. In order to remove some of the unnecessary details in the figure, label the figure, and scale the y-axis, double click on the figure you created in iThink - the box shown in figure 13 will open.

On the box shown in figure 13, the items listed in the box under allowable represents model elements that can be plotted on a graph. To plot any of those items listed under “allowable” on a graph, double click on any of them. They will automatically move to the next box on the right hand side. Any item that has been selected for plotting on the graph is listed in the box under “selected” on the right hand side of the box. To change the scale on the y-axis, click (select) the item (or items, if apply) in the box under “selected”. The selected item will be highlighted (pig population in figure 13). To make sure the change in the scale on the y-axis is correctly applied, make sure the double-headed arrows next to the item listed on the box under “selected” is surrounded by a circle. If it is not, click on the double headed arrow once or twice until it is surrounded by a circle. At this stage, the boxes (Min and Max) next to scale at the bottom of the box shown in figure 13 will allow you to make changes on the scale of the y-axis. The boxes next to display at the bottom of the box show the scale on the x-axis. The x-axis is automatically set in a similar ways to the model time horizon (simulation length). But you can also change it if needed. However, the minimum on the X-axis can only be greater than or equal to the starting time of the simulation and the maximum should less than or equal to the finish time of the simulation. In other words, if your simulation time starts from week 1 to week 13, then the minimum on the x-axis should be 1 or more and the maximum should be 13 or less. To remove any unnecessary details on the figure, check the box “hide details” and delete (or change) the word untitled in the box next to the title.



**Figure 13: Toolbox to change graph settings**

### System dynamics and value chains

Different interpretations and definitions of the value chain have been used in the literature (Tchale & Keyser 2010; UNIDO 2009; FAO 2012) but the most common definition is the one by Kaplinsky & Morris (2000, page 4) as the “full range of activities which are required to bring a product or service from conception, through the different phases of production (involving a combination of physical transformation and the input of various producer services), delivery to final consumers, and final disposal after use.”

Traditionally, value chain analysis (VCA) has been conducted in a qualitative manner, following the principles of Kaplinsky and Morris (2000), or later development handbooks (e.g. M4P 2007<sup>5</sup>; GIZ 2008<sup>6</sup>). A complete VCA consists of a mapping of the value chain; an analysis of chain governance, or understanding the power and transaction mechanisms that exist within and among different chain nodes; an assessment of upgrading strategies taken by chain actors; and the distributional benefits of chain participation (Rich et al. 2011). While quantitative data is used in the analysis to look at the distribution of value and flows of goods throughout the value chain, such analyses are incapable of looking at the returns to potential identified policy options.

However, as noted earlier, SD modeling gives us a unique platform to look more rigorously at the value chain. A naïve way of looking at the value chain in an SD framework would be to consider each node or actor in the chain as a stock, with various flows connecting different nodes depending on the marketing channel in question. However, certain nodes are much more complicated in their setup, particularly at the production and retail sides of the chain. This is even more pressing in

<sup>5</sup> <http://www.value-chains.org/dyn/bds/docs/574/M4PComMarkConferenceSummary.pdf>

<sup>6</sup> [http://www2.gtz.de/wbf/4tDx9kw63gma/ValueLinks\\_Manual.pdf](http://www2.gtz.de/wbf/4tDx9kw63gma/ValueLinks_Manual.pdf)

livestock systems given the influence of biological parameters and time delays in production. This requires a more rigorous means of modeling both the supply and demand sides of livestock systems as we detail below.

An SD model of the value chain can include a number of interacting modules:

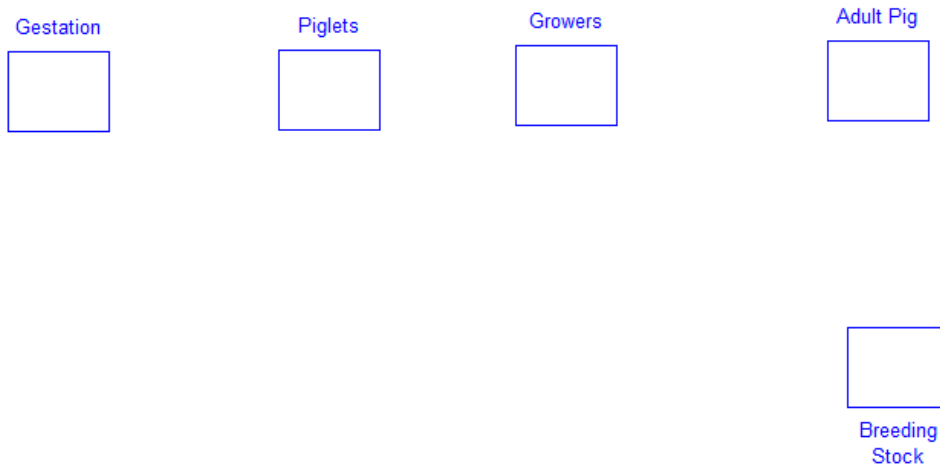
- A module that describes the supply side of the value chain, in terms of modeling explicitly the dynamics of the production of a good;
- A module that describes the demand side of the value chain, in terms of modeling explicitly the dynamics of sales and parameters that influence consumer demand;
- A module that shows and links the intermediaries that exist between production and consumption and the different channels of distribution that may exist;
- A module that describes price formation within the value chain and how producers and other value chain actors respond in the short-term to price;
- A module that looks at long-term investment behavior of relevant value chain actors to build and expand (or reduce) capacity;
- A module that elaborates the profitability of different actors in the chain and how those influence short- and long-run investment behavior;
- A module that looks at adoption behavior and how that influences behavior in the value chain;
- A module that looks at biophysical or other external influences on behavior in the value chain, such as environmental factors, land constraints, or animal disease considerations.

In the next three sections, we will create a value chain model of the pig sector. We will do that by creating a number of modules associated with supply, demand, and price setting that reflect the dynamics of production, trade, and sales within the value chain. The next sections should be done interactively, following closely in iThink with the instructions and discussion given.

### Pig Supply

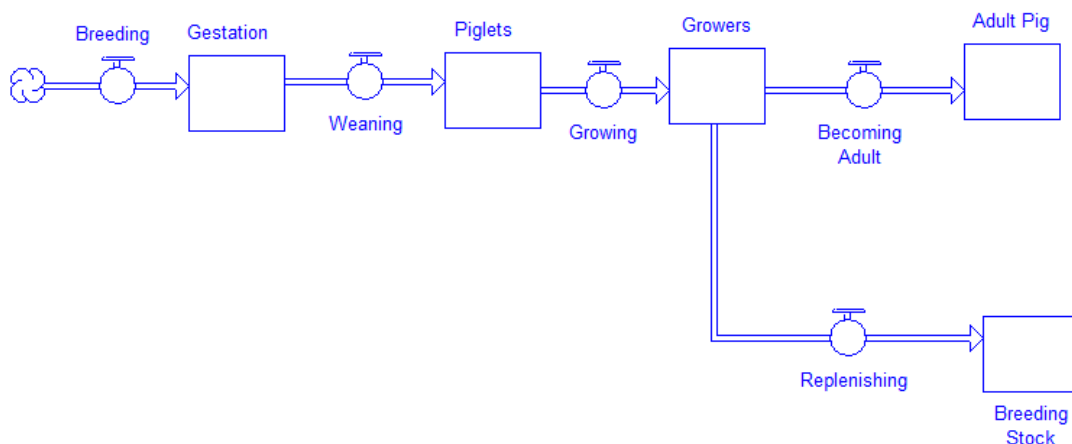
Let's first begin with modeling pig production by expanding the model that we constructed in the first section. Before, you will recall that we consider a very simple model of population growth, with one stock and two flows representing births (inflow) and sales (outflow). However, pig (or any livestock or fish) production should be modeled in more detail to reflect the evolution of growth over time. In particular, we can consider the life cycle of animals in our model by separating out the different stages of life experienced by pigs. Pigs start out in gestation prior to birth, and then become piglets after birth. After some weeks, piglets grow into young pigs, or growers. Growers are fattened for many weeks after which they become adult pigs, or finishers. Finishers are finally either sold for slaughter or die naturally. Some growers are reserved for breeding stock, which are then used to generate more pigs. Each of these transitions between age classes will depend on various factors. Further disaggregation (such as based on gender) can be done but this depends on the purpose of the model.

In order to conceptualize this model, let's try to develop this age cohort using the language of SD. We will first define the different pig age groups ("breeding stock", "gestation", "piglets", "growers", and "adult pigs") as stocks. We represent these age classes as stocks because they represent the number of pigs in each age group on a farm or in a region (for example, 10 breeding pigs). In our model, let's define five such stocks: "Breeding stock", "gestation", "piglets", "growers", and "adult pigs". We denote this visually in iThink as shown in figure 14 below.



**Figure 14: Age cohort stocks of pigs**

Let's next consider the transition of pigs from one age stage to another. In the language of SD, we represent this by flows. These flows define the transition from one age class to another, and are defined as action verbs e.g., weaning as the transition from gestation to piglets, growing as the transition from piglets to growers, and so on (figure 15). In our model, we define a transition (flow) from growers to breeding stock to model the behavior of reserving some animals for breeding stock. New pigs enter the system via the inflow "breeding" into the gestation stock.

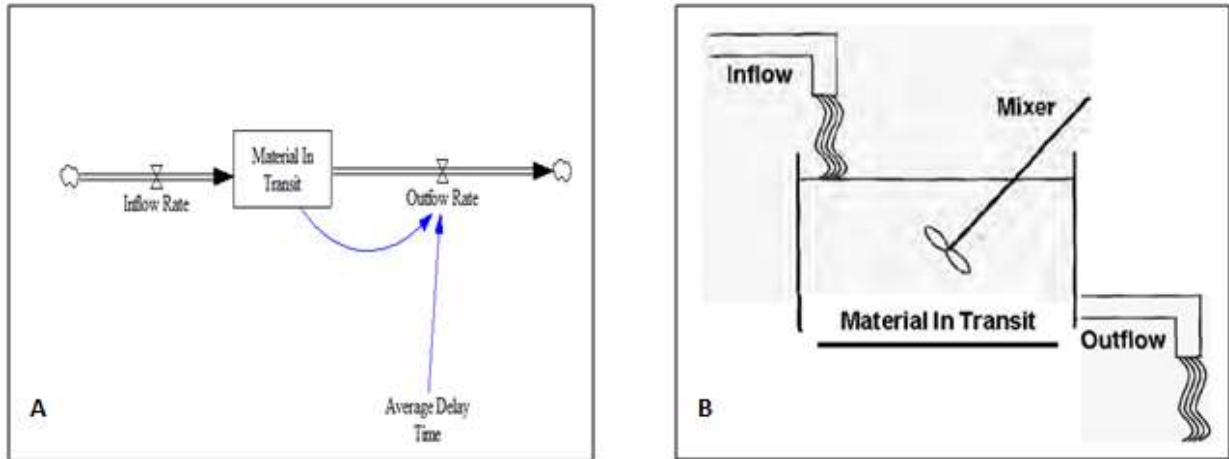


**Figure 15: Inflows and outflows of pigs in an age cohort model**

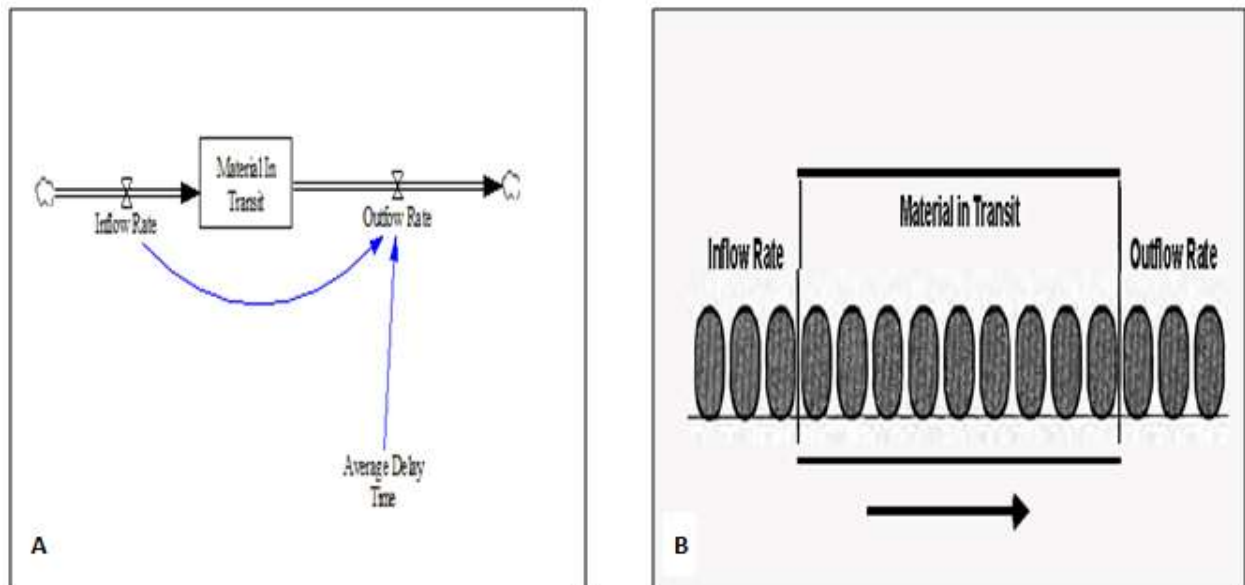
As we described earlier, flows are regulated via stocks and parameters. What elements might affect the transition of pigs from one stage to another? Let's start with "breeding". Breeding will depend firstly on the availability of pigs to breed with, or put differently, the number of animals in "Breeding Stock". Second, breeding will depend on the number of piglets that result from breeding. On average, about 8-12 pigs are generated per litter and there are just over two litters per sow per year. Given this, we'll need to create convertors for "Head per litter" and "Litters per year" that will be multiplied by the number of "breeding stock". If our model is a weekly model, we will need to adjust "litters per year" accordingly i.e., by dividing the annual rate by 52 weeks.

Let's move next to the flow of "weaning". The rate of outflow from "Gestation" to "Piglets" will depend on the number of pregnant pigs we have and the time it takes for a pregnant pig to give birth. In other words, there will be a time delay between conception and birth. How we model this delay depends on our assumptions regarding it. In SD, we distinguish between two types of delays: material delays and pipeline delays. A material delay defines the outflow from a stock as the quantity held in the stock at time  $t$  divided by the length of delay. The assumption here is that there

is perfect mixing of all elements in the stock at any given time so that the probability of exit by any element is the same and independent of when the item entered the stock (figure 16, Sterman 2000). A pipeline delay, by contrast, acts like a conveyor belt where elements that enter the stock first exit the stock first (figure 17). We model a pipeline delay by using the *inflow* into the stock (not the stock itself) and applying a DELAY function to the outflow that depends on the inflow and the delay time.



**Figure 16: a stock and flow structure of first order material delay (A) and an equivalent bath-tub metaphor (B). Source (Sterman 2000)**

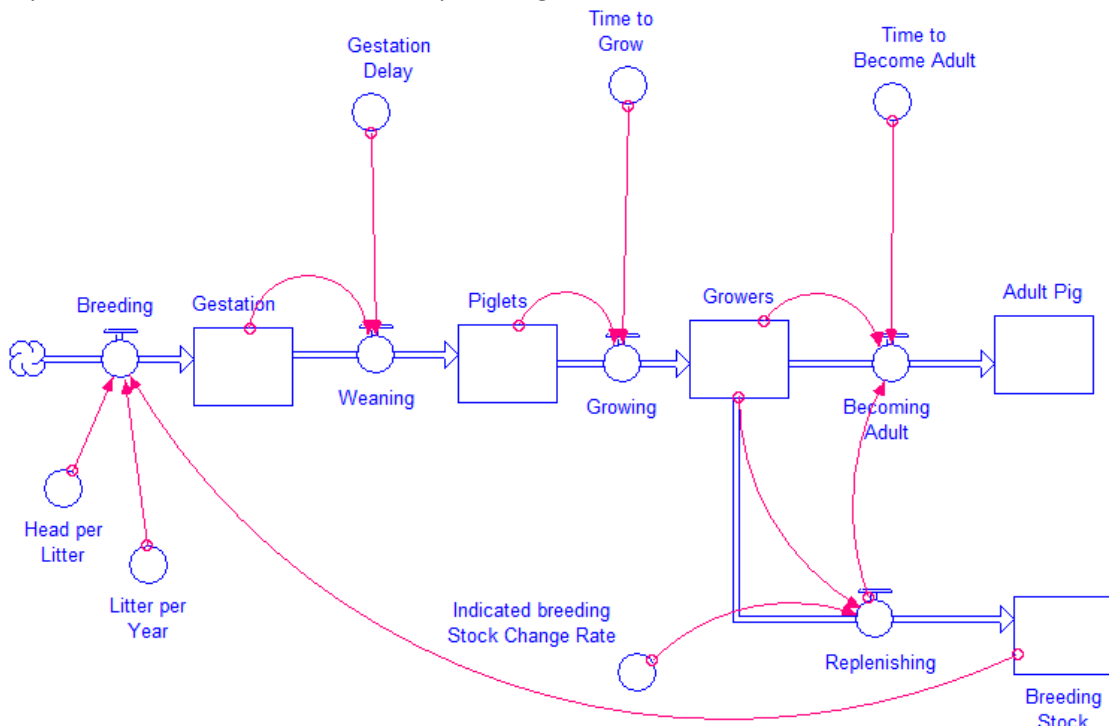


**Figure 47: a stock and flow structure of first pipeline delay (A) and an equivalent metaphor (B). Source (Sterman 2000)**

While a pipeline delay is a more realistic means of marketing livestock systems intuitively, it may not always be the case that producers strictly manage age cohorts of animals, especially in developing countries and in smallholder systems. For the purposes of exposition for this model, we use a material delay in this model. Therefore, our “weaning” flow would be our “Gestation” stock divided by a converter “Gestation Delay”. Similarly, our “growing” stock would be defined similarly, as the stock of “Piglets” divided by the converter “Time to Grow”.

When animals are in the stock “Growers,” one of two things can happen. First, a proportion of female pigs will be reserved for breeding stock to create pigs in the future. To model this, we define a flow called “replenishing” that depends on number of pigs in the “growers” stock and a parameter called “indicated breeding stock change rate” that is the proportion of pigs in “growers” stock producers decide to keep for breeding purposes. Second, animals not kept for breeding will eventually move to the “Adult Pig” stock through the “becoming adult” flow. We define the flow “Becoming adult” as the number of pigs in the “growers” stock divided by “time to grow” minus pigs from the “replenishing” flow to ensure consistency. In other words, those pigs that are kept for breeding are not moved to the stock of “adult pig”.

Figure 18 provides an illustration of the completed age structure.



**Figure 18: An SD model of a pig production system**

It is important to recognize what is exogenous and endogenous in SD models. An endogenous variable, stock, or flow is the one that is part of a feedback loop. That is, it affects a model component or components and in return is affected by those components of the model. An exogenous parameter (or variable) is a parameter or a variable that is not part of any feedback loop. That is, it affects a component or components of the model but it is not itself affected by the rest of the model. In other words, it is exogenous to the model or outside the model boundary.

Based on figure 18 above, consider the following exercise:

**Exercise 5 (all questions relevant to figure 18):**

- 1- Are “breeding stock” and “growing” flow exogenous or endogenous components of the model? Why?

-----

- 2- Are parameters “gestation delay”, “indicated breeding stock change rate”, and “head per litter” endogenous or exogenous components of the model? Why?

-----

- 3- Can you show how many feedback loops are in the model shown in figure 18? Please address whether they are reinforcing or balancing feedback loops? (Hint: recall that a feedback loop is a circular causality. That is, we start from point **X**, we go through **Z** and then move back to **X** in a circular connection.)

-----

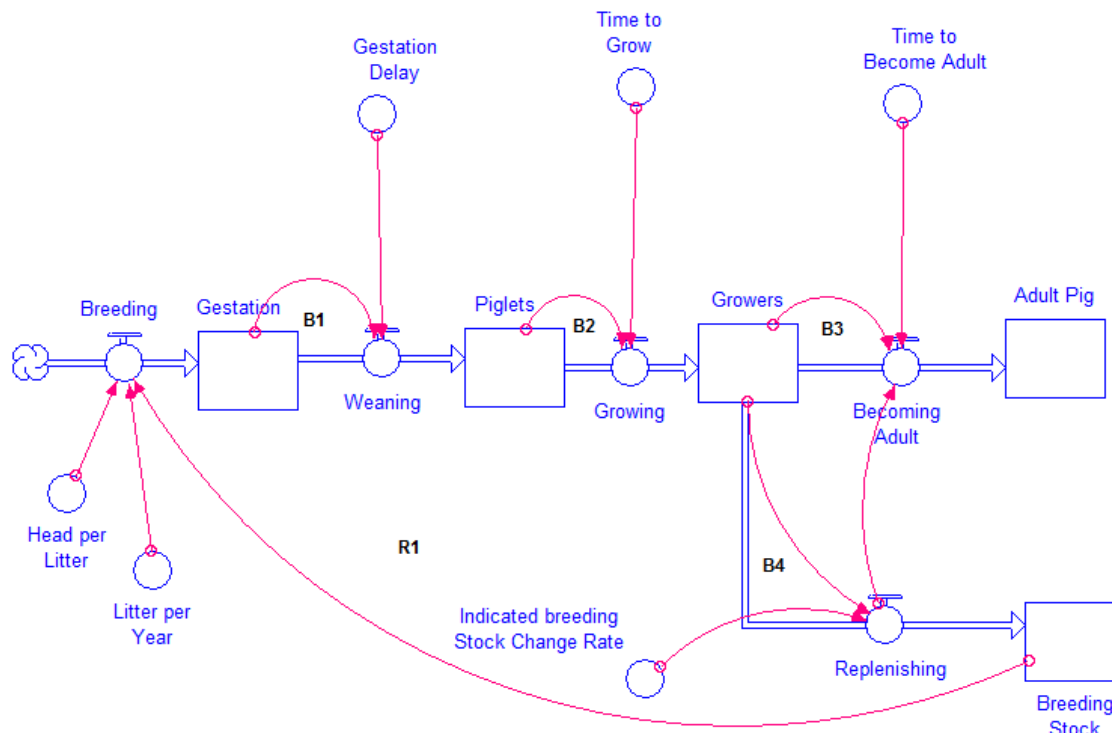
**Solution 5:**

- 1- Both “breeding stock” and “growing” flow are endogenous components of the model because they are part of a feedback loop. That is, “breeding stock” determines “breeding” flow which in turn affects “gestation” stock. In a similar vein, “gestation” stock determines “weaning” flow which affects “piglets” stock and goes through “growing” flow, “growers” stock, “replenishing” flow and feeds back to the “breeding stock”. “Breeding stock” is connected back to “breeding” flow via a connector that closes the feedback loop. This indicates that all elements involved in this feedback loop are endogenous to the model as they affect each other (directly or indirectly) in a circular relationship. This feedback loop is a reinforcing feedback loop (R1<sup>7</sup> in figure 15) because, for example, more “breeding stock” leads to more “breeding” which in turn leads to higher “gestation”, “piglets”, “growers” and then even more “breeding stock” which leads to a further increase in “breeding” flow and so on in a reinforcing manner.
- 2- These parameters are all exogenous to the model. That is, “gestation delay”, “indicated breeding stock change rate”, and “head per litter” affect the flows “weaning”, “replenishing”, and “breeding”, respectively, but they are not affected by any components in the model. That is, they are fixed parameters and do not change over time or interact with the rest of the model.
- 3- There are at least five feedback loops in the model. See figure 19 for feedback loop locations.

---

<sup>7</sup> R and B signs are not mandatory to be written on the SD model structure. But for the sake of clarity they are provided in a few places in this manual.





**Figure 59: Feedback locations in the model<sup>8</sup>**

R1 is described in point 1 above. The balancing feedback loop B1 involved two endogenous (“gestation” and “weaning”) and one exogenous (“Gestation delay”) elements of the model. B1 shows that more “gestation” leads to more “weaning” which in turns lowers “gestation” stock which leads to lower “weaning” flow in the next rounds and so on. This process happens in all balancing feedback loops.

As we model the supply side of this model, we should be aware of the following issues. First, it is best to endogenize SD model behavior as much as possible (i.e. so that the behavior is driven by closed feedback loops rather than exogenous parameters and variables). The modeler should justify the choice of exogenous elements (parameters and variables) of any model to ensure the validity of the model (in SD, we call this the model boundary test). If we run a model for 10 years, the exogenous elements of the model must not change significantly over this time horizon. For instance, the parameters “litter per year” and “head per year” can be estimated based on historic values (i.e. taking a 10 year average). If they do change in real life, then we must endogenize them to ensure model validity. For instance, consider the parameter “indicated breeding stock change rate”. This represents the producer’s decision on the percentage of pigs that are kept for breeding purposes. If we run the model over a one-year (52 weeks) time horizon, it is unlikely that this parameter would change. However, simulating over a ten-year time horizon might be more problematic since producers will decide what percentage of pigs to keep as breeding stock based on pig price movements. Thus, if the pig price is high for one to two years, producers might decide to keep more breeding pigs for future investment and vice versa. Therefore, in this case we would need to endogenize this parameter to increase our confidence in the model (we will come back to this point later).

<sup>8</sup> Parameters “head per litter” and “litter per year” represent number of offspring per litter and number of litter per year, respectively. Parameters “gestation delay”, “time to grow”, and “time to become adult” represent pregnancy duration, time required to grow from piglets to growers, and time required to grow from growers to adult pigs, respectively. Parameter “indicated breeding stock change rate” represents the percentage of grower pigs farmers decide to keep for breeding stock over time.

Before we move to the next section, let's parameterize the model in figure 19 with the figures found in table 1 below. In table 2, we review the definitions of the flows defined earlier. For now, keep the initial values of stocks undefined – we will define these in the next section.

**Table 1: Initial parameter value of the pig model**

Parameter	value	Unit
Gestation delay	17	{week}
Time to grow	5.5	{week}
Time to become adult	16	{week}
Feeding time	1	{week}
Average breeding time	260	{week}
Minimum processing time	1	{week}
Indicated breeding stock change rate	0.000781	{1/week}
Litter per year	2	{litter/pig/year}
Pig per litter	8	{pig/litter}

**Table 2: Initial value of the flows of the pig production model**

Flow	Value	Unit
Breeding	$(\text{Breeding Stock} * \text{Litter per Year} * \text{Head per Litter}) / 52$	{pig/week}
Weaning	$\text{Gestation} / \text{Gestation Delay}$	{pig/week}
Growing	$\text{Piglets} / \text{Time to Grow}$	{pig/week}
Becoming adult	$(\text{Growers} / \text{Time to Become Adult}) - \text{Replenishing}$	{pig/week}
Replenishing	$\text{Growers} * \text{Indicated breeding stock change rate}$	{pig/week}

#### Pig sales

In this section, we expand the model structure to move downstream in the pig value chain. Our model expansion will necessarily challenge the model boundary to correctly represent the system or the problem that we intend to model. As a rule of thumb, expanding the model boundary should aim to improve model behavior and to verify the model structure (i.e. structure verification test) as a means of increasing our confidence in the model. If expanding the model boundary does not have any impact on model behavior, then it is an unnecessary addition. However, when our model structure is incomplete (i.e. the model structure cannot be verified compared to the real life physical system), then challenging the model boundary is necessary. The structure verification test is simple. This test is conducted by looking at the model structure to verify if there is any parameter or structure that has no real life meaning. If there is a parameter used that has no real life meaning, then this represents a shortcoming of the model. Similarly, if the model structure is not consistent with real life physical systems, then challenging the model boundary is necessary.

As constructed at present, the structure of the model is incomplete if we compare it with a real life pig production system. For instance, both of the stocks “adult pig” and “breeding stock” are only affected by inflows. In real life, however, farmers grow pigs to supply the market and fulfill customer demand.

Let's think through the model structure and determine where we need to expand. First, farmers do not keep “adult pigs” (if not kept for breeding purposes) for a long time. Rather, adult pigs are sold for slaughter to meet customer demand. In SD terms, that means we should create a flow that comes out of the stock of “adult pig” called “slaughtering”. The flow of “slaughtering” is based on the number of pigs in “adult pig” stock and the amount of time pigs are fed prior to slaughter, which

we can define as a parameter called “feeding time”. Second, pigs in “breeding stock” are kept for a few years and then shipped to the market after they are past productive age. In SD terms, we model this through a flow called “breeding stock slaughtering” that comes out of the stock of “breeding stock”. This flow depends on the number of pigs in “breeding stock” divided by the “average breeding time,” which is defined as the average time (measured in the number of years or weeks or any other time units) that farmers keep breeding pigs on their farm prior to sending them to slaughter.

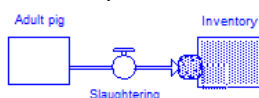
When farms sell pigs for slaughter, they are purchased either directly by slaughterhouses or other intermediaries (traders, collectors, etc.). For simplicity and exposition, let’s assume a simple value chain structure by which slaughterhouses buy directly from farmers (it is possible to expand this – see Hamza et al. 2014a for an example). Slaughterhouses will transform pigs into meat. The amount of pigs they process into meat can be represented as a stock, which we define as “inventory”. In this example, we keep all units in our model in liveweight equivalent even though it is possible to convert pigs into meat directly (can you think how you would model this?) The stock of “inventory” represents available meat (in pig equivalent) to the market. In your model in iThink, create a stock named “inventory” and then drag the cloud at the end of the flow of “slaughtering” to drop it<sup>9</sup> on the stock of “inventory”. Repeat the same process to connect the flow of “breeding stock slaughtering” to the stock of “inventory”.


In SD models, actors sell from inventory. On the supply side, producers sell from the stock of adult pigs and old breeding stocks. Same way, processors will sell meat from inventory to consumers. This means that we need to create an outflow named “selling” that exits the stock of “inventory” to supply meat (in pig equivalent) to the market. The flow of “selling” depends on two things. First, “selling” will depend on consumer demand, which we can define as a converter that denotes the number of pigs demanded by customers per week. Second, the amount processors can sell will depend on their ability to meet customer demand. For instance, it might be that processors may have limitations in meeting consumer demand if there are capacity constraints in slaughter. In other words, sales will be a fraction of consumer demand that is adjusted by the fraction of demand that can be met by processors. That means we should adjust the amount that can be sold based on these capacity constraints.

To model these capacity constraints, we first define a converter that we name “order fulfilment ratio”. “Order fulfilment ratio” depends on “customer demand” and the ability of processors to meet demand from inventory, which we call “maximum selling rate”. This converter will be a fraction ranging from 0 to 1. If “order fulfilment ratio” is 1, then all “customer demand” is realized, while if it is 0.5, this means only 50% of “customer demand” is realized. Thus, we will need to create a converter called “maximum selling rate” and connect both “maximum selling rate” and “customer demand” using connectors to “order fulfilment ratio”.

“Order fulfilment ratio” will be defined as a graphical (or table or lookup) function. We define the x-axis, as “maximum selling rate” divided by “customer demand”. When the x-axis is less than one, it should be the case that processors will not be able to fully meet demand. Once the x-axis is greater than one, processors should be able to meet all, or nearly all, demand. To define this as a graphical function in iThink, first, define the equation “order fulfilment ratio” as “maximum selling rate”



<sup>9</sup> Make sure that the cloud is attached to the stock before dropping, usually when you drag the cloud of any flow to any stock, the stock will be filled with dots before dropping as it is shown in figure below.



divided by “customer demand”. The equation box looks like figure 20 (A). Second, at the bottom of equation box, next to (  $x^2$  ), where you insert the equation to the model elements, select graphical function (  ). The equation box will then look like figure 20 (B). To switch to graphical function, on the top-left corner of the equation box (shown in figure 20 (B)), check the box next to graphical. When the graphical function is switched on, the equation box will look like figure 20 (C).

Now, note that the x-axis is the same as the equation you defined initially for the variable (figure 20 – A). To define the graph, you can simply either use the cursor on the space provided to draw a graphical relationship or under the x-axis enter specific points (see figure 20 – D) to insert values directly to define the relationship between the x-axis and y-axis. You can also set min and max values of both the x-axis and y-axis on specific numeric boxes provided next to each axis. You can adjust the number of data points to figures you have in your model. The default value for the number of data points is 11 but if you do not have a specific reason to change or keep the default value, change it to 5 points to make it easier to draw.

Now, going back to finalize the graphical equation inside our variable “order fulfilment ratio”, let’s define the min and max for x-axis as 0 and 2, respectively. Zero means that “inventory” is empty and 2 means that number of pigs in the “inventory” is twice the size of “customer demand”. Set the min and max value of the y-axis as 0 and 1, respectively. Zero means 0% of “customer demand” is realized, while 1 means 100% of “customer demand” is realized. To set values on y-axis, use the table you opened (as in figure 20 – D). While the x-axis will be automatically set, you can change the values of the x-axis by unlocking the key on the top of the table (do not forget to lock it again when finished editing). To fill the graphical function you developed, use the following values as (x, y): (0, 0), (0.5, 0.3), (1, 0.9), (1.5, 1), (2, 1). The expression (0,0) means that when inventory is empty, then 0% of customer demand is realized, while (0.5, 0.3) means that when inventory equals 50% of customer demand, then only 30% of customer demand is realized. Note that these are suggested values and can be adjusted if needed.

When you have finished defining all data points on the x and y-axis, click the check sign (  ) on the bottom-right corner of the equation box. Variables (or parameters) that are defined as graphical equations will have a small curvy line in it (  ).

The converter “Maximum sales rate” represents the number of pigs that can be sold to the market given the availability of pigs in the stock of “inventory” and time required for processing, which we define as a converter called “minimum processing time”. We define “maximum selling rate” as “inventory” divided by “minimum processing time”. Finally, we will define the flow of sales as the product of “consumer demand” and “order fulfillment ratio”, which ensures that consumer demand is adjusted by the fraction of inventory that can be sold to the market.

Figure 21 shows the updated model structure that shows the flow of pigs from production to consumption (selling to the market), while table 3 and figure 22 define the new parameters and flows added to the model.

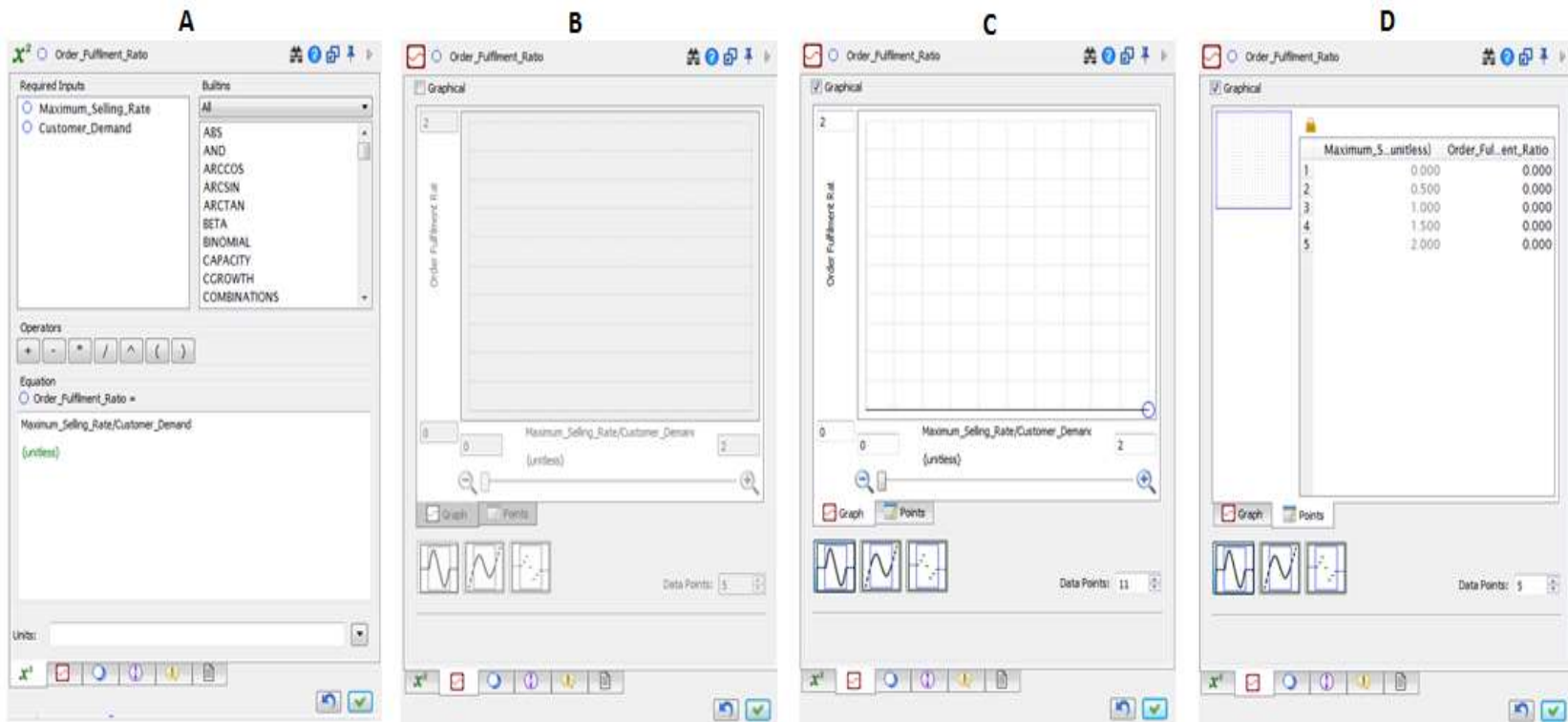
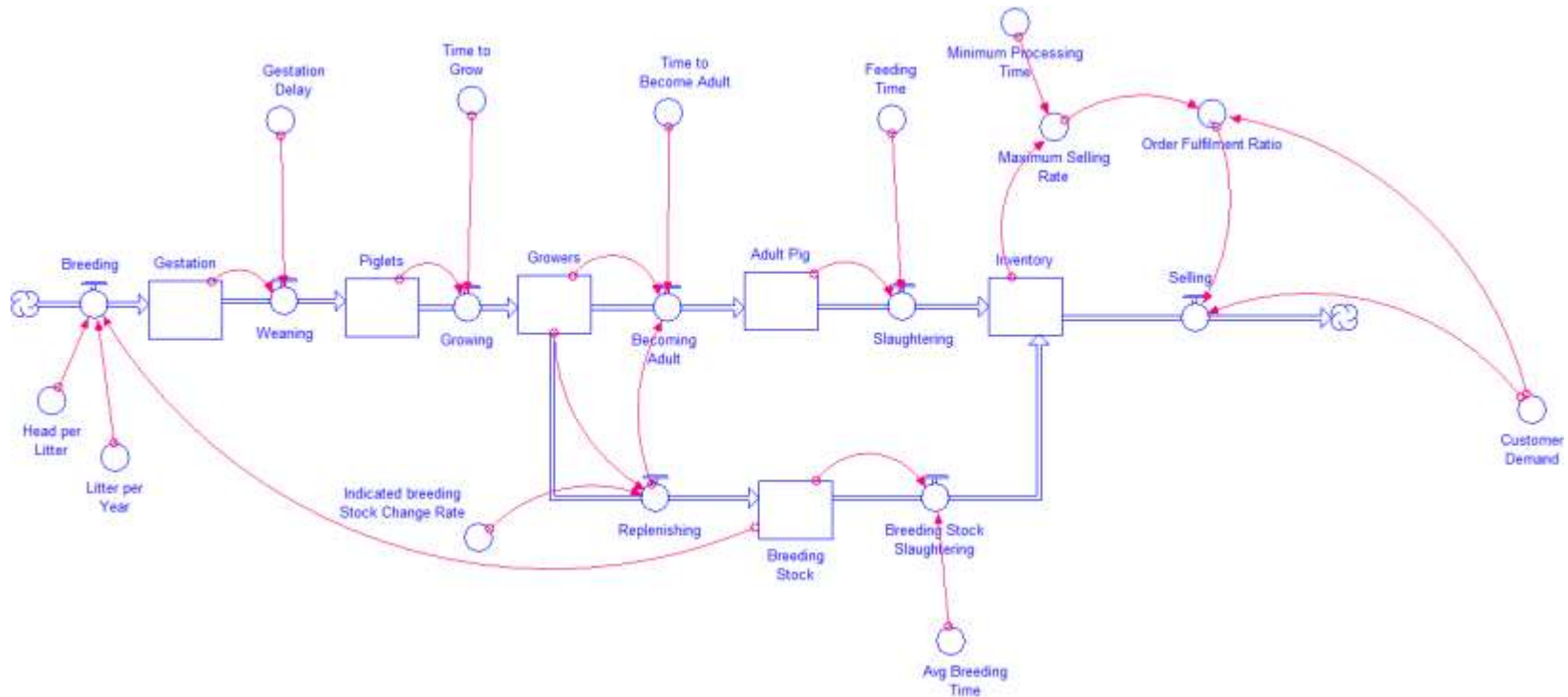


Figure 20: The procedure to make a graphical function in iThink

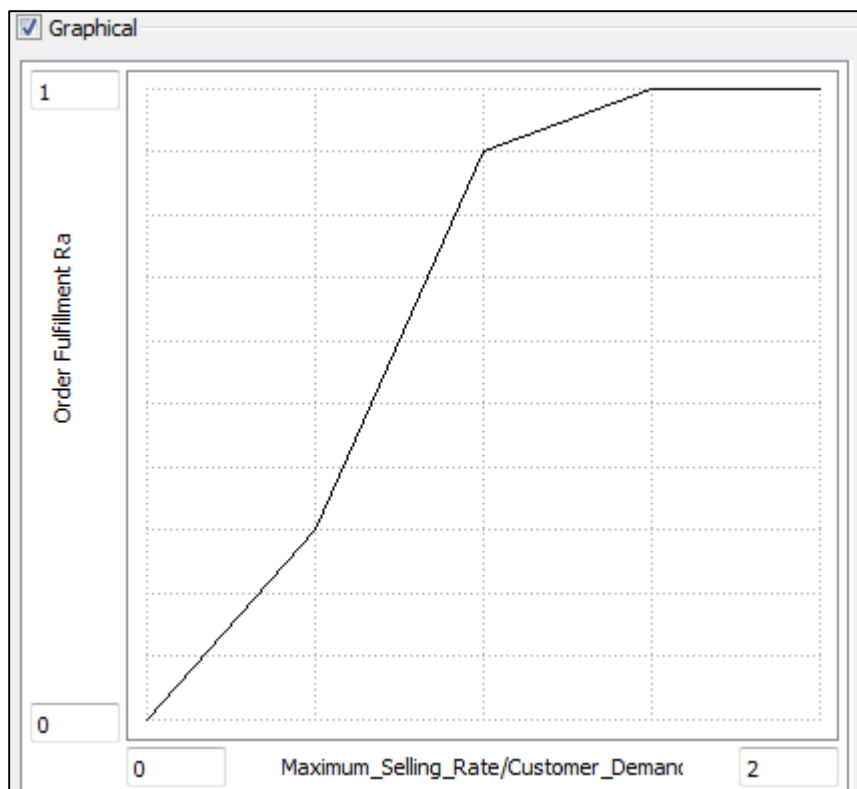


**Figure 21: Revised structure of the pig supply chain model to include processing and sales to the market<sup>10</sup>**

<sup>10</sup> Both flows of “breeding stock slaughtering” and “slaughtering” represent number of breeding pigs that is been removed from breeding stock and number of adult pigs being slaughtered over time, respectively. Parameters “average breeding time” and “feeding time” represent number of weeks (or years depending on model time unit) producers keep breeding pigs and the number of weeks producers keep adult pigs before slaughtering them, respectively. Stock of “inventory” represents inventory of pigs available to be shipped to the market. Flow of “selling” represents number of pigs shipped to the market to fulfil customer demand. Parameter “minimum processing time” represents time required to process a pig for selling. Variables “maximum selling rate” and “order fulfilment ratio” represent number of pigs producers can sell over time and portion of customer demand that can be satisfied by “inventory”, respectively. Parameter “customer demand” represents number of pigs demanded by customers.

**Table 3: New parameter and flow values for the demand side of the pig model**

Parameter	Value	Unit
Customer demand	390	{pig/week}
Order fulfillment ratio	Graphical function (see figure 21)	unitless
Maximum selling rate	Inventory/Minimum Processing Time	{pig/week}
Flows	Value	Unit
Slaughtering	Adult Pig/Feeding Time	{pig/week}
Breeding stock slaughtering	Breeding Stock/Avg Breeding Time	{pig/week}
Selling	Customer Demand*Order Fulfilment Ratio	{pig/week}



**Figure 22: Graphical function for order fulfillment ratio**

To completely operationalize the latest version of the model shown in figure 21, we need to provide initial values for the stocks. Table 4 shows the name of the stocks, the equation that is used to calculate the initial value of each stock, and the unit of each stock in the pig model.

**Table 4: Initial value of the stocks in the pig model**

<b>Stock<sup>11</sup></b>	<b>Value</b>	<b>Unit</b>
<b>Gestation</b>	Customer Demand <sup>12</sup> *Gestation Delay	{pig}
<b>Piglets</b>	Customer Demand*Time to Grow	{pig}
<b>Growers</b>	Customer Demand*Time to Become Adult	{pig}
<b>Adult pig</b>	Customer Demand*Feeding Time	{pig}
<b>Breeding stock</b>	Customer Demand*(52/ (Litter per Year*Head per Litter) <sup>13</sup>	{pig}
<b>Inventory</b>	Customer Demand*Inventory Goal	{pig}

Based on figure 21 above, consider the following exercises:

**Exercise 8:**

- 1- Is it possible to run the model shown in figure 21 in equilibrium? How?

-----

- 2- Are “customer demand” and “indicated breeding stock change rate” exogenous or endogenous element of the model shown in figure 21? Why?

-----

- 3- Does the model structure make sense? Do we need to further challenge the model boundary?

-----

---

<sup>11</sup> Note: equations inside stocks are only used to provide initial value to the stock. You can also enter a number directly inside a stock to initialize it. But the use of question is desired. For example, if you change the initial value of “customer demand”, then the initial value of all stocks will update automatically.

<sup>12</sup> We multiply the equation inside the stocks by customer demand to ensure that the values of all stocks are set to run the model in equilibrium.

<sup>13</sup> The expression (52/ (Litter per Year\*Head per Litter) is used in this equation to correctly convert model time unit to weeks and to adjust to the fact that pig (unlike sheep and cows) has more than 1 litter per year and more than 1 offspring per litter. That is, 52 represent 52 weeks of a year (note: if we want to convert model unit to months, then we will use 12 or 365 for days). Then we divide 52 by (litter per year \* head per litter) to make sure that we can run the model in equilibrium regardless of parameter (litter per year and head per litter) values.



- 4- Is there any new feedback loops introduced to the model provided in figure 21 (compare to the mode shown in figure 19)? If yes, what type of feedback loops (reinforcing or balancing)?
- 

**Solution 8:**

- 1- Yes, it is possible to run the model in equilibrium. In order to run the model in equilibrium, the inflows and the outflows of any stock must be equal (we will illustrate this through running the model after this exercise).
- 2- “Customer demand” and “indicated breeding stock change rate” are exogenous parameters in the model because they affect the flow of “selling” and “replenishing”, respectively, but they are not (neither directly or indirectly) affected by any element of the model (they are not part of any feedback loops).
- 3- The flow of pigs from breeding to selling seems reasonable and consistent with real life smallholder livestock producers supply chains. However, “customer demand” is exogenous to the model which is unrealistic. In other words, “customer demand” should be an outcome of the interaction of supply and demand and price movements rather than a constant parameter (more on this will come later). Similarly, “indicated breeding stock change rate” should change based on price movement (more precisely, producer’s perception about) the price of pigs in the market) rather than a constant parameter because farmers decision to manage breeding stock depends on price movements.
- 4- There are at least two new balancing feedback loops:
  - A- The loop of “breeding stock” and “breeding stock slaughtering” flows.
  - B- The loop that starts from “inventory” and goes through “maximum sales rate”, “order fulfilment ratio”, and the flow of “selling” and feeds back to the stock of “inventory”.

**Pig demand**

In this section, we move to explain and elaborate on the demand side of the value chain. To begin, let’s first think through what factors affect “customer demand”. From neoclassical economic theory, the demand curve is defined as a function of prices (own price, prices of complements and substitutes), income, and other factors such as preferences, tastes, etc. Moreover, our demand curve can shift (to the right or left) based on changes in income, prices of complements and substitutes, and other factors. Accordingly, we can define demand in a similar way in iThink, while translating it into the language of stocks and flows.

Recall before that we already created a converter called “Customer demand”. What would this converter depend on? From the discussion above, we should ensure that demand is based on standard demand factors (prices, incomes) and shift parameters that might cause demand to change based on changes in preferences and tastes. In our model, we will thus define a stock called “Demand” that

determines what the value of “Customer demand” will be, and whose value will change based on changes in demands that depends on prices, income, and other factors

Remember that the value of stocks can only change from inflows and/or outflows. That means we need to define some type of flow that allows demand to change, but which also takes into account that the change in demand could be either positive or negative. We model these demand changes by creating what’s called a bi-flow. A bi-flow is double-headed flow that shows the flow can either be positive (inflow to the stock of “demand”) or negative (outflow from the stock of “demand”). To create a biflow in iThink, simply draw an inflow or an outflow as you usually do, but when you double click on the flow to open the equation box, on the top-left corner, check biflow instead of uniflow. Call this biflow “Change in demand” and be sure to connect to the stock “Demand”. Note that one arrow head of the biflow is filled while the other is not. When the biflow head that is not filled is attached to the stock, this means when the value of biflow is positive, meaning that the stock increases.

The biflow “change in demand” is affected by three elements. First, it will depend on the stock of “demand” to represent the current demand or demand at time t (make a link from the stock of “demand” connect to the biflow “change in demand” using a connector). Second, it will depend on a parameter “demand adjustment time” that represents the time it takes consumers to adjust their demand for pig meat. In other words, if price goes up or down, it takes some period of time for consumers to adjust their purchasing behavior. Use a converter to create a parameter named “demand adjustment time” and then make a connection from this parameter to the biflow “change in demand” using a connector. Finally, changes in demand will depend on price changes that will determine what value demand will take. This means that we will need to define some type of price relationship that relates demand to prices. In your iThink model, use a converter to create a parameter called “demand schedule” and then link it to the biflow “change in demand”.

Let’s define the parameters just added to the model. Define “demand adjustment time” as 4 weeks. To define “demand schedule,” we will create a simple demand function that depends on its own price, the response of demand to price (the elasticity of demand), and the initial level of demand that determines where the demand curve intercepts the y-axis. In your model, create the converters “price”, “demand intercept” and “price elasticity of demand” and link them to the converter “demand schedule” using connectors. Double click on “demand schedule” and then insert the following equation:

$$\text{Demand schedule} = \text{EXP} (\text{Demand Intercept}) * \text{Price}^{\text{Price Elasticity of Demand}} \{\text{pig/week}\}$$

Let’s assume that “price elasticity of demand” is -0.5 {unitless}. The “demand” intercept” will depend on initial values of price and demand based on the demand schedule equation above. If we rearrange the equation for “demand schedule,” solving for demand intercept, and replacing “initial demand” for demand and “initial price” for price, we get:

$$\text{Demand intercept} = \text{LN} (\text{Initial Demand} / (\text{Initial Price}^{\text{Price Elasticity of Demand}}))$$

Make sure the demand intercept converter is connected to initial demand and initial price. Insert 390 {pig/week} and 50 {\$/pig} for the parameters “initial demand” and “initial price”, respectively. Let’s also define the initial value of the stock of “demand” equal to “initial demand”. To do this in iThink, double click on the stock of “demand”, start to write initial and it will automatically show up on among a list of parameters used in the model. Make sure to use the correct parameter of “initial demand”. Note that this will only allocate the value to the stock at time 1 (starting time of the simulation).

“Demand intercept” is also affected by exogenous factors called “other factors”. “Other factors” is an exogenous variable that exogenously introduces shocks to the model. The shock might be a long-term shock due to animal disease or a short term shock. For the sake of testing the model response to various shocks, we will focus on the latter in this manual. Now, on your model in iThink, create a converter called “other factors” and link and use a connector to connect it to “demand intercept”. The converter “other factor” is defined in a way to represent percentage change. In other words, if “other factors” is 0.8, it means that “other factors” lower the “demand intercept” to 80% of what it would normally be; if “other factors” is 1.2, it means that “other factors” increase the “demand intercept” to a 120% of what it would normally be.

We assume that “other factors” is exogenous to the model that depends on a parameter called “demand shock”. The parameter “demand shock” allows us to introduce exogenous changes to the demand side of the model. If the value of this parameter is 0, it means there are no external changes on “demand intercept”. If it is 0.2, it means we introduce a 20% increase in demand. If it is -0.2, it means we introduce a 20% decrease in demand. In your model, use a converter to create a parameter “demand shock” and then connect it with “other factors” using a connector. To define the extent to which “demand shock” affects “other factors” and then “demand intercept”, we use a STEP function in iThink. A STEP function in iThink will increase a parameter by a certain amount (height) at a certain period of time. The STEP function takes the form “STEP (height, time)”. Height represents the magnitude of change (or shock). Time represents the starting time of the STEP change. To find STEP function in iThink, simply look for it in the builtin<sup>14</sup> function list on the right side of the equation box (the equation box can be opened by double-clicking on the targeted parameter, variable, flow, and stock). The below equation is used to define how any “demand shock” introduced to the model:

$$\text{Other factors} = 1 + (\text{STEP}(\text{Demand Shock}, 10) - \text{STEP}(\text{Demand Shock}, 12)) \{\text{unitless}\}$$

Let’s review what this function says. STEP(Demand Shock, 10) tells the program to increase the parameter “other factors” by the amount of “Demand Shock” at time=10. We add 1 to this since “Demand Shock” is given as a percentage change. The term “-STEP (Demand Shock, 12)” is introduced to make sure that the STEP length will be for only 2 weeks. That is, we are subtracting the same “Demand Shock” two weeks later, so that the net effect after week 12 is zero (i.e., other factors reverts to its baseline value of 1). If we only included the first step function, then the STEP change will be permanent from week 10.

The equation inside “demand intercept” now reads as

$$\text{Demand intercept} = \text{LN}(\text{Initial Demand} / (\text{Initial Price}^{\text{Price Elasticity of Demand}})) * \text{other factors}$$

Note that the parameter “demand shock” is an input to the STEP function inside “other factors”. For now, let’s assume that parameter “demand shock” equals to zero.

For the remainder of the simulation, the stock of “demand” will change through the biflow “change in demand”.

For the biflow “change in demand,” insert the following equation:

$$\text{Change in demand} = (\text{Demand Schedule} - \text{Demand}) / \text{Demand Adjustment Time} \{\text{pig/week/week}\}$$

Note how this equation works. When the price increases, the demand schedule will change. If this value is larger than the stock of demand, prices will rise. This change in the demand will be mediated by the demand adjustment time, such that if the demand adjustment time is large, demand will adjust much more slowly.

Figure 23 shows the new model structure. Compare your model to the one shown in figure 23 to make sure that you have correctly specified all the connections and directions of flows and connectors. Be particularly careful with the direction of the biflow “change in demand”.

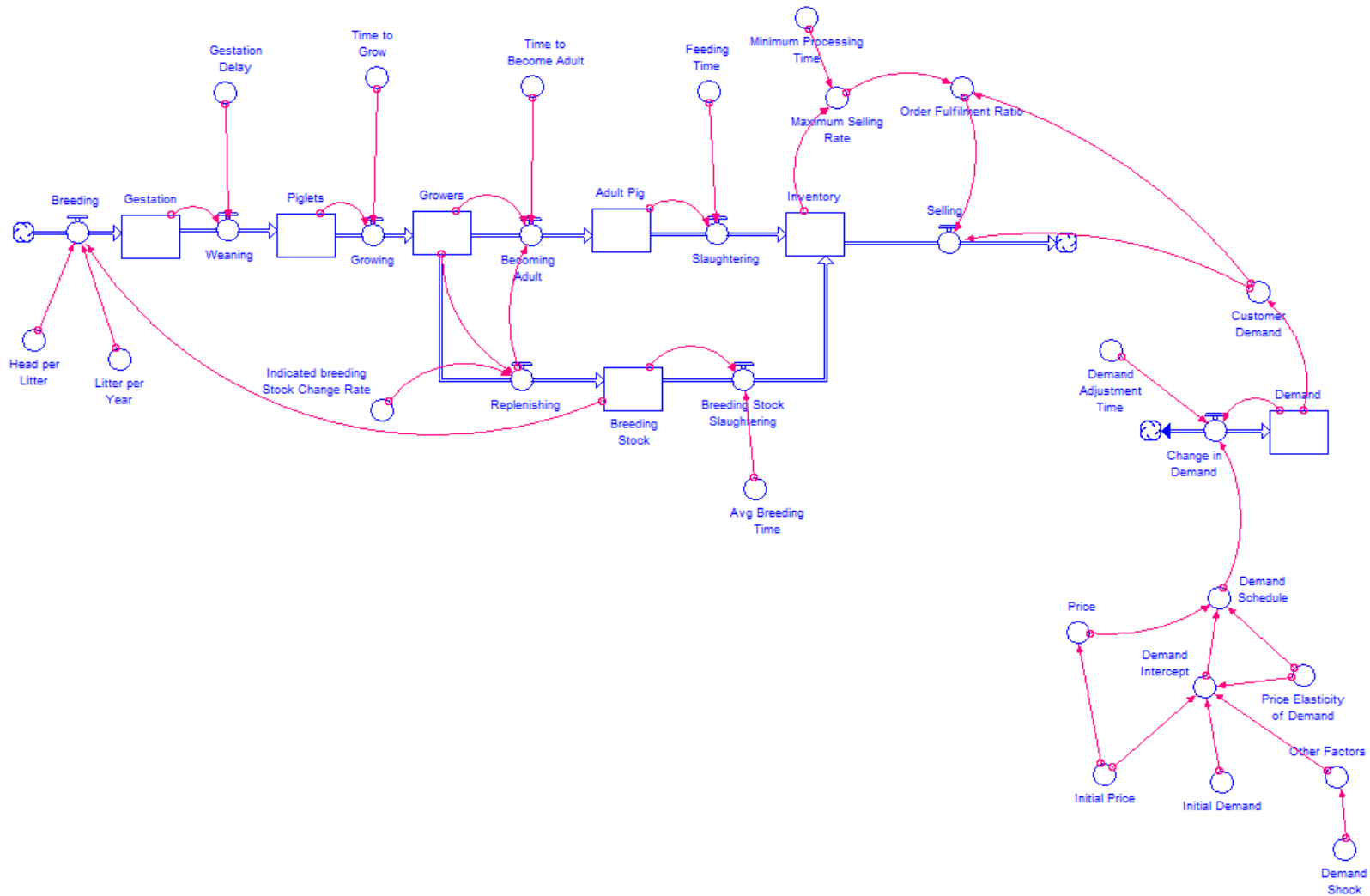
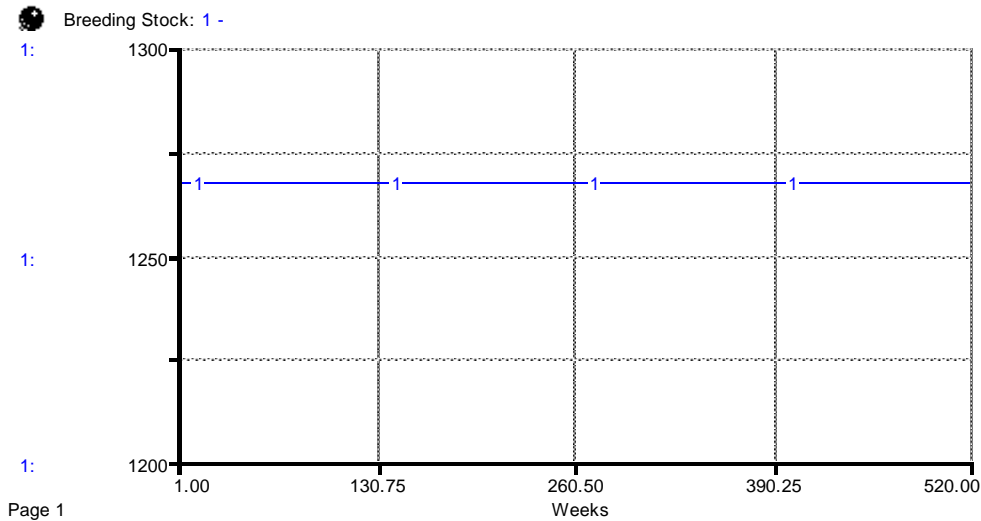


Figure 23: Incorporating added structure to the pig value chain model to include factors that affect demand<sup>15</sup>

<sup>15</sup> “Initial price” and “initial demand” parameters are introduced to initialize the model (only initial values of the affected model component are affected by these two parameters). Otherwise, both price and demand are determined by the model endogenously (more on this will come later).

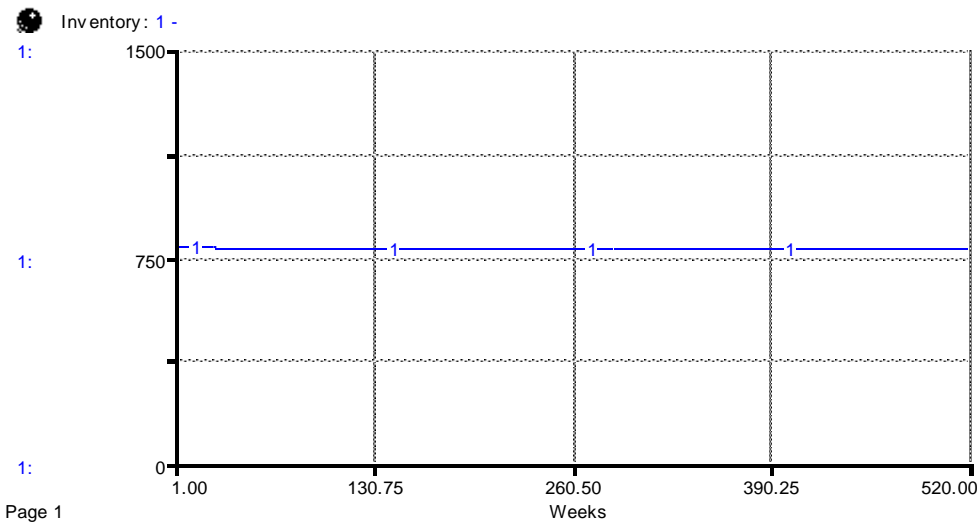
**Note:**

After using the parameter value in tables 1 and 2 above and the equations provided in the text above figure 23 as suggested to your model in iThink to operationalize the latest version of the model shown in figure 23, the model should run in equilibrium. Below are a few figures from the model behavior. Running model in equilibrium provides a benchmark to check the model validity under shocks. In other words, running the model in equilibrium ensures that the modeler understands how the model works and provides a benchmark to compare the model behavior under a shock.



?

**Figure 24: Breeding stock behavior of the model shown in figure 23**



?

**Figure 25: Inventory behavior of the model shown in figure 23**

Running model in equilibrium assists modelers to understand the model behavior as model structure expands and to conduct tests to ensure model validity. The tests include an equilibrium shock and behavior prediction tests. These tests are conducted by shocking (change in a parameter) the model in

which the modeler predicts how the model behavior changes under a shock. In the latest version of the model shown in figure 21, the model runs in equilibrium (figure 24 and 25), to conduct an equilibrium shock and behavior perturbation tests, let's increase "customer demand" by 20% from week 10 to 12 (i.e. from week 10 to 12 there will be 20% increase in demand but the demand heads back to its original value afterwards).

**Exercise 9:**

- 1- Can you predict the behavior of "breeding stock" under the proposed demand shock (please use the space provided below to draw a figure)? If your figure is not similar (only qualitatively, do not worry about quantitative precision) to the figure provided in solution 9? Why?

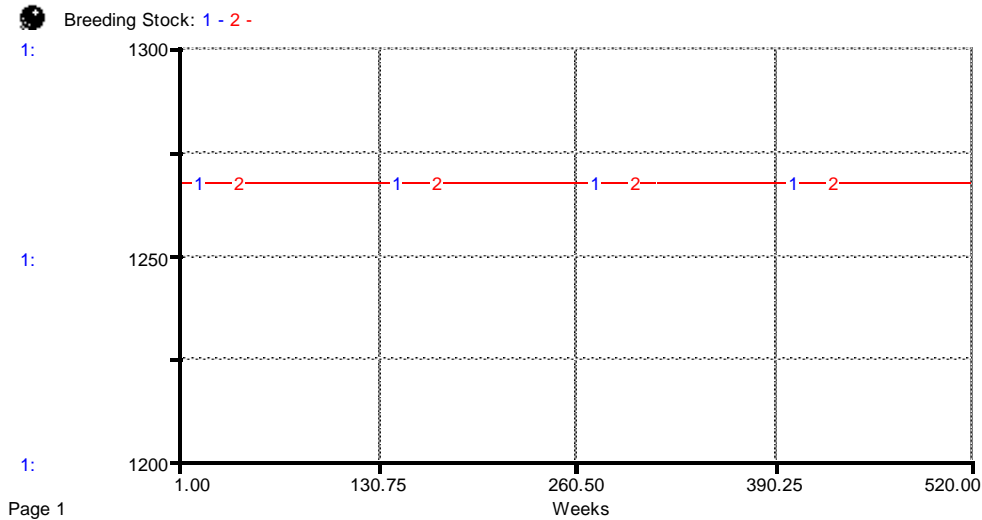


- 2- Can you predict the behavior of "inventory" under the proposed demand shock (please use the space provided below to draw a figure)? If your figure is not similar (only qualitatively, do not worry about quantitative precision) to the figure provided in solution 9? Why?



**Solution 9:** (note that we compare behavior of "breeding stock" and "inventory" here with those shown in figures 23 and 24 – run 1 is the same as in figures 23 and 24; run 2 is the model behavior under demand shock)

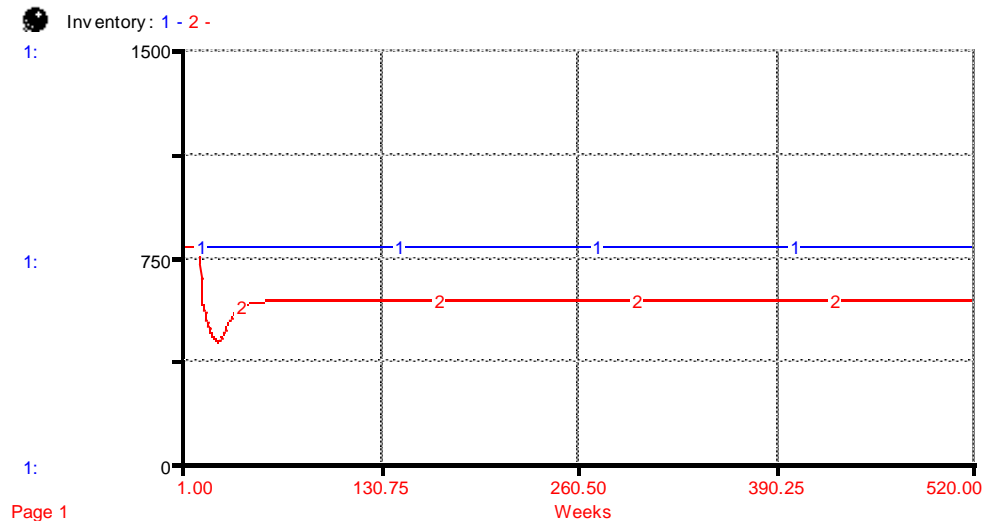
- 1- We expect that the behavior of breeding stock will not change under demand shock because producer's decision to manage pig production (i.e. "indicated breeding stock change rate") is not endogenous). That is, pig supply remains the same because the model does not have "indicated breeding stock change rate" endogenized to ensure that any change in demand is reflected in the supply. In other words, the supply of pigs remains the same regardless of changes in the demand side.



?

Figure 26: undated breeding stock behavior under demand shock (run 2)

- 2- We expect that the stock of "inventory" will decline and reach a new equilibrium at a lower level than before the shock (figure 24 above or run 1 in figure 27) because a 20% demand shock leads to more pigs moving out of inventory without any increase in pig supply.



?

Figure 27: undated inventory behavior under demand shock (run 2)



**Note:**

Figures 26 and 27 showed that the latest version of the model shown in figure 23 above makes a correct response to the demand shock. However, the model behavior showed the need to challenge the model boundary. For example, economic theory expects that change in demand (for example, the 20% demand shock we introduced to the model) should trigger changes in price and the quantity supplied. But currently price is outside the boundary (exogenous), and thus to improve the realism of the model, we need to endogenize price, which we will do in the next section.

**Exercise 10:**

- 1- Is “customer demand” in the model shown in figure 27 an exogenous or endogenous component of the model? Why?

-----

- 2- Is “price” an exogenous or endogenous component of the model shown in figure 27? Why?

-----

**Solution 10:**

- 1- Though there are various factors affecting “customer demand”, “customer demand” is still an exogenous element of the model because it is not part of any feedback loop. That is, to be an endogenous component of the model, the component needs to be part of a feedback loop (circular connection, which is we start from a point and need to go through the model until we go back to the starting point). In this example, if we start from “customer demand, we cannot go through the model and head back to “customer demand in a circular connection).
- 2- “Price” is an exogenous component of the model because it is not part of any feedback loops.

**Pig price setting**

In SD models, the process by which equilibrium occurs is modeled somewhat differently than in neoclassical economic theory. In neoclassical economic theory, shifts in the supply and/or demand curve (arising from changes in technology, income, or external market forces) will define a new equilibrium point. In an SD model, by contrast, price changes (and subsequently changes in supply and demand) occur based on product availability as defined by the stock of inventory (Whelan and Msefer 1996).

We assume in an SD model that products (pigs or meat in our model) are sold from the inventory producers or processors have at hand. This should make intuitive sense if we think about the process of sales in industrial supply chains in the real world. The decision for a producer in our model is to first define the desired amount of inventory and to make their stocking decisions based on the difference between what they actually have as inventory and desired inventory. If a producer has more inventory than they desire, they would want to get rid of the excess. In doing so, this will place more supply on the

market and reduce prices. Same way, if the producer does not have a sufficient level of inventory, they will want to increase their stocks, increasing demand for product and raising the price.

In our model, we will use the stock of “inventory” to represent the availability of pigs in the market and the flow of “selling” to represent pig demand. The ratio of “inventory” to “selling” represents actual inventory coverage at each period of time. In other words, actual inventory coverage gives a signal to the market about the relative availability of pigs in the market. If this ratio is 1 (that is, one week of inventory coverage), it means the inventory of pigs is equal to market demand. If the ratio is greater than 1, it means more pigs are available in the market than market demand. If the ratio is less than 1, it means fewer pigs are available in the market than market demand. To add this to your SD model in iThink, use a converter to create a variable named “actual inventory coverage” and then make a link using a connector from the stock of “inventory” to “actual inventory coverage”. Similarly, make another link through a connector from the flow of “selling” to “actual inventory coverage”. We then define the equation of “actual inventory coverage” as:

$$\text{Current inventory coverage} = \text{Inventory} / \text{Selling} \{ \text{week} \}^{16}$$

“Current inventory coverage” represents the immediate or actual inventory coverage in the market. However, such information is not immediately available to the market. In other words, there may be a delay in how quickly this information affects decision-making. We can account for this delay by creating two new variables – the first named “perceived inventory” and the other called “time to perceive inventory”. “Perceived inventory” takes into account the delay in which this information is acted upon, while “time to perceive inventory” is the number of weeks of this delay; it is the time needed for the information about actual inventory coverage to be perceived by market actors.

In iThink, make a connection from “current inventory coverage” and one from “time to perceive inventory” to “perceived inventory coverage” using connectors. Let’s assume that “time to perceive inventory” is 2 weeks. The “perceived inventory” coverage is a smoothed version of “current inventory coverage”. In other words, value chain actors adjust their perceptions gradually over a certain time delay (i.e. “time to perceive inventory”). To define such an equation in iThink, we use a builtin function called smooth (SMTH):

$$\text{Perceived inventory} = \text{SMTH1} (\text{Current Inventory Coverage}, \text{Time to Perceive Inventory})^{17} \{ \text{week} \}$$

Perceptions about inventory are only one part of the product availability story. Producers (or other chain actors) will also have a level of inventory desired based on various factors related to the nature of the product or the amount of inventory they want to keep in stock to ensure the market is satisfied. We thus need to compare our perceptions about inventory with these desired levels of inventory, which in turn will determine whether we want to liquidate or add stocks, and will subsequently influence what happens to prices.

---

<sup>16</sup> Note how the unit of “week” is generated. We divide “inventory” which it has a unit of {pigs} by “selling” which it has a unit of {pigs/week}. Thus, we get pigs/(pigs/week) which can be rewritten as pigs\*(week/pigs) which results in the unit of {week}. That is, week or weeks of inventory coverage.

<sup>17</sup> “The SMTH1 builtin calculates a first-order exponential smooth of *input*, using an exponential averaging time of *averaging time*, and an optional initial value *initial* for the smooth. If you do not specify an initial value, SMTH1 assumes the value to be the initial value of *input*”. (iThink help command)

In your model, define a converter named “inventory goal” to represent the desired number of weeks of inventory coverage. Let’s assume that our “inventory goal” parameter is 2 weeks. In other words, if the flow of “selling” equals to 100 pigs/week, then we desire to have 200 pigs in our inventory. Next, let’s define a new converter that defines the relationship between perceived and desired inventory (or inventory goal). Call this converter “inventory ratio” and define it as the ratio between “perceived inventory coverage” and “inventory goal”, ensure that you use connectors to link perceived inventory and inventory goal.

**Inventory ratio** = Perceived Inventory/Inventory Goal {unitless}

“Inventory ratio” represents market signals about the availability of pigs in the market. If the inventory ratio is greater than 1, then perceived inventories are greater than what we want. In that case, this ratio should send a signal that prices should fall. To enable that those signals affect price with this parameter, first create a new parameter named “price elasticity of inventory ratio” to represent the sensitivity of the “inventory ratio” to price. Let’s assume that the “price elasticity of inventory ratio” is -0.3 {unitless}.

Next, make connections from “inventory ratio” and “price elasticity of inventory ratio” to “price” using connectors. Similarly, make a connection from “price elasticity of inventory ratio” to “price” using a connector. Note that earlier in this manual we defined “price” to be equal to “initial price”. However, now there are two additional factors (“inventory ratio” and “price elasticity of inventory ratio”) that affect “price” which requires updating the way we defined “price”. The new definition (or equation) of price is:

**Price** = Initial Price\*Inventory Ratio<sup>Price Elasticity of Inventory Ratio</sup> {\$/pig}

Figure 28 presents the addition of the price setting process to the model. Make sure you got all connections especially direction of flows and connectors as shown in figure 28.

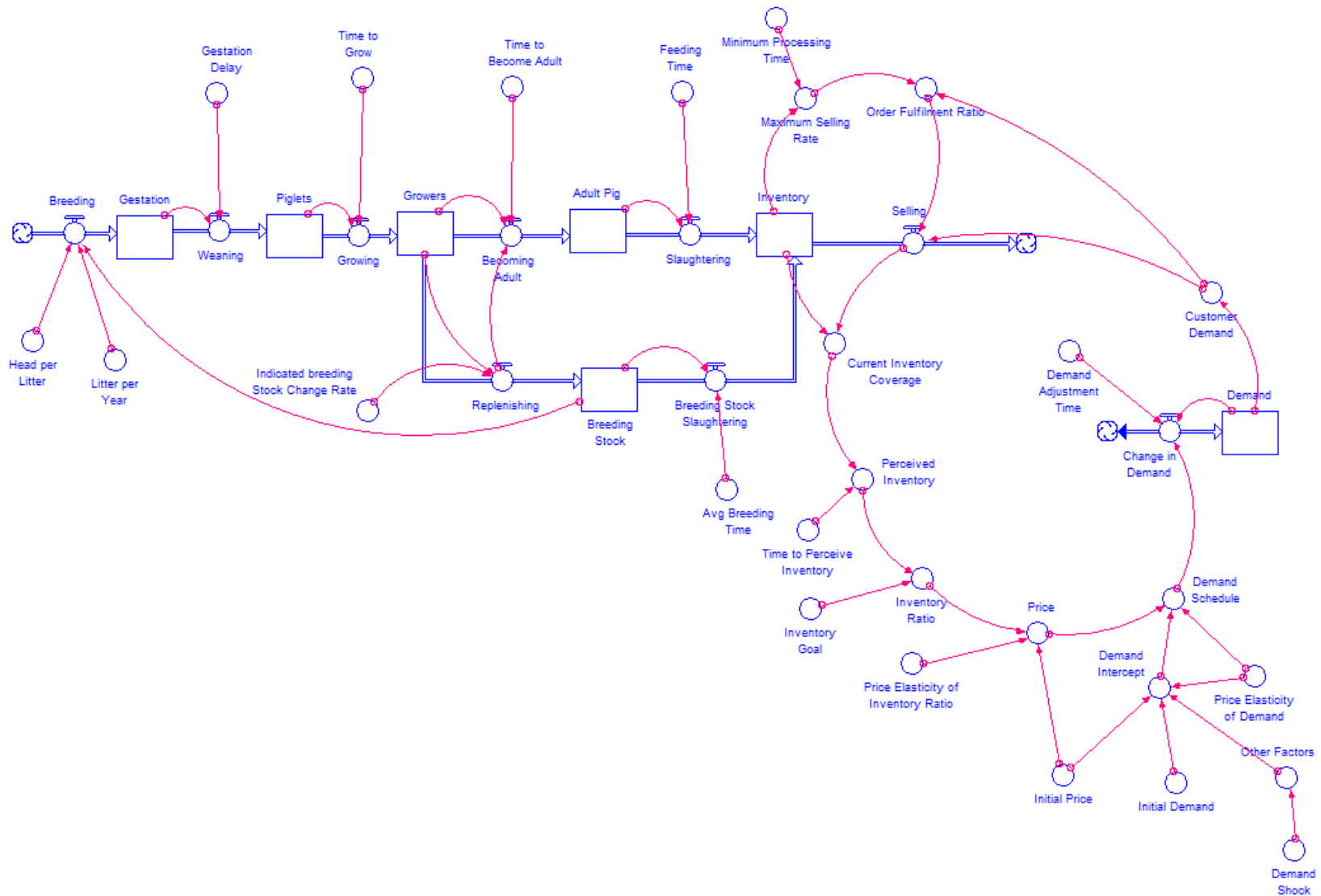


Figure 28: Incorporating added structure to the pig value chain model to include factors that affect prices

**Exercise 11:**

- 1- Can you identify any new feedback loops (whether reinforcing or balancing loop) in the added model structure in figure 28?

-----

- 2- Are “customer demand” and “price” endogenous or exogenous variables in the model show in figure 28? Why?

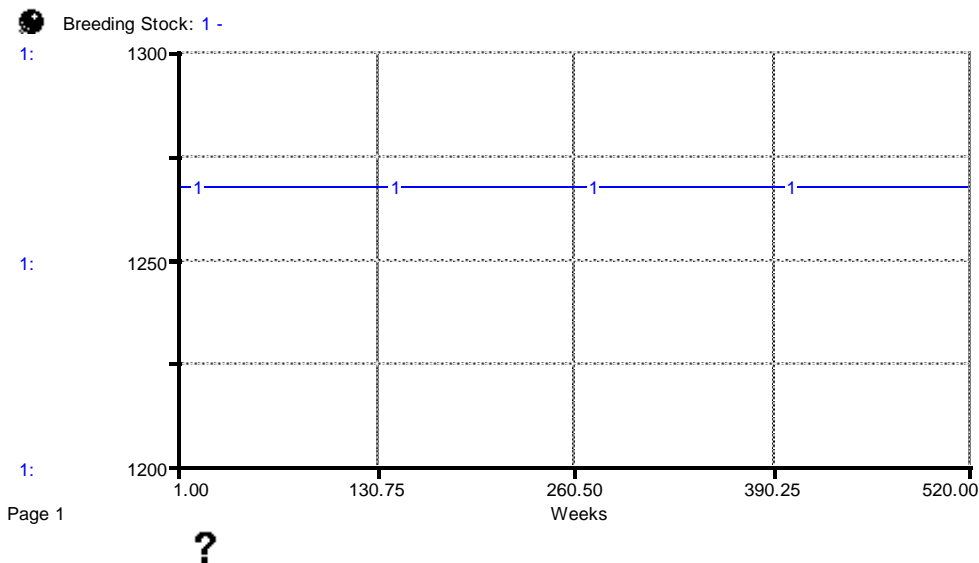
-----

- 3- Is it possible to run the latest version of the model (figure 28) in equilibrium? How?

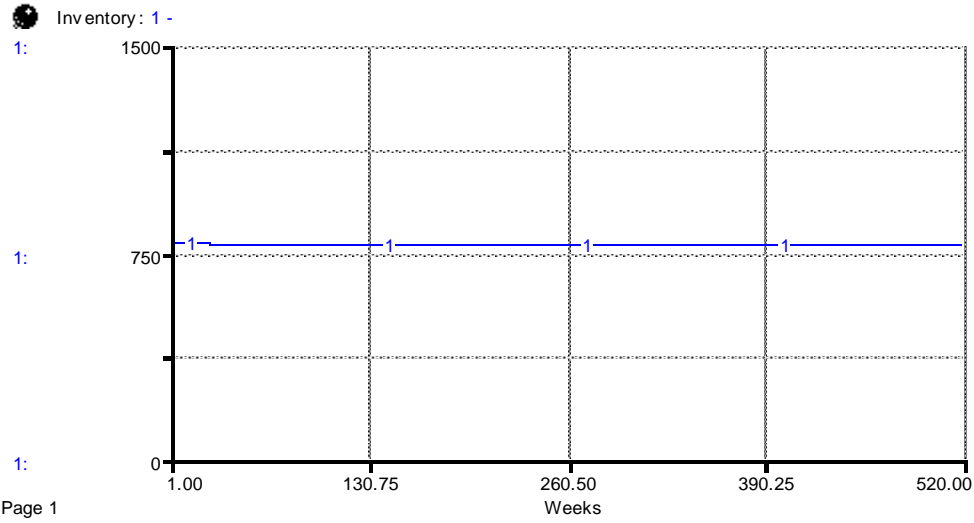
-----

**Solution 11:**

- 1- A main feedback loop is introduced to the model. This feedback loop begins at “customer demand”, goes through the flow of “selling” and the stock of “inventory”, then through “current inventory coverage” and “inventory ratio” which affects “price” which in turn affects “demand” and feeds back to “customer demand.” To determine whether it is a reinforcing or balancing feedback loop, let’s assume that “customer demand” increases by 20%. This leads to higher “selling” which lowers the stock of “inventory” because sales increased by 20%. Higher “selling” and lower “inventory” lead to lower “inventory coverage” which in turn leads to higher “price” because low inventory coverage sends signals of a supply shortage to the market. Higher “price” leads to lower “customer demand” which lowers “selling” in the next round. This indicates that this feedback loop is self-correcting and therefore it is a balancing feedback loop.
- 2- Both variables “customer demand” and “price” are endogenous to the model because they are part of a feedback loop described in point 1.
- 3- Yes, it is possible to run this model in equilibrium. To run the model in equilibrium, all inflow(s) and outflow(s) of each stock in the model must be equal. Using the equations provided earlier to define model elements to operationalize the added structure of the model in figure 28 will make the model to run in equilibrium. Below are a few figures of the model behavior in equilibrium.

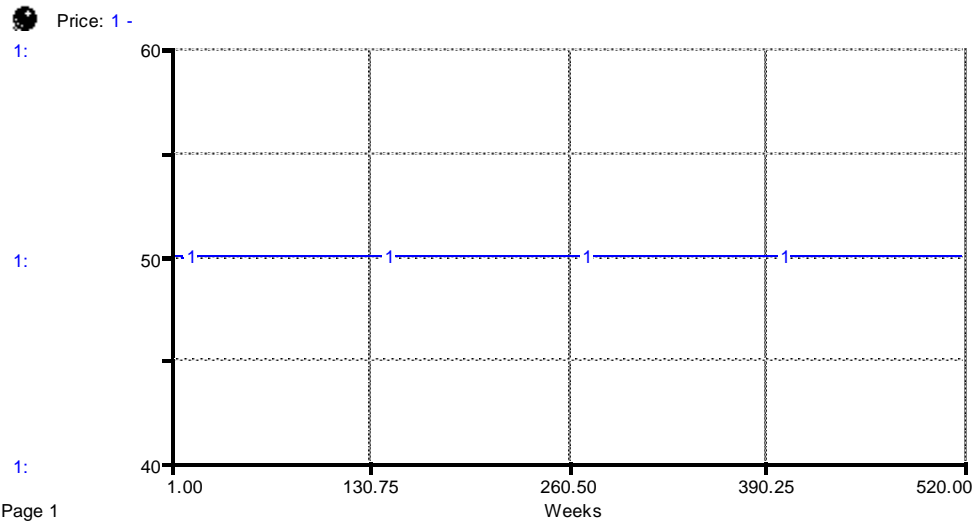


**Figure 29: Breeding stock behavior of the model shown in figure 28**



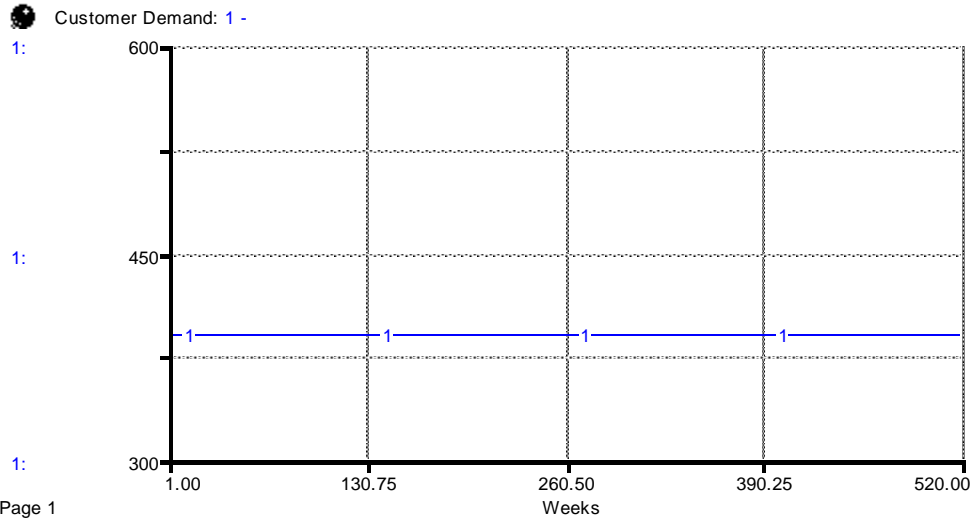
?

Figure 30: Inventory behavior of the model shown in figure 28



?

Figure 31: Price behavior of the model shown in figure 28



?

**Figure 32: Customer demand behavior of the model shown in figure 28**

In the latest version of the model shown in figure 28, the model is run in equilibrium (figure 29 and 32). To conduct an equilibrium shock and behavior prediction test, let's increase demand by 20% from week 10 to 12 (i.e. from week 10 to 12 there will be a 20% increase in demand but demand then heads back to its original value afterwards).

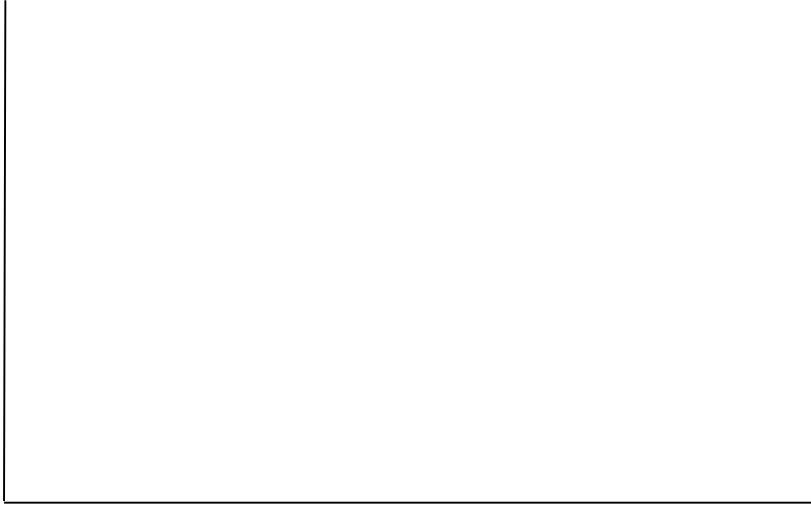
**Exercise 12:**

- 1- Can you predict the behavior of "breeding stock" under the proposed demand shock (please use the space provided below to draw a figure)? If your figure is not similar (only qualitatively, do not worry about quantitative precision) to the figure provided in solution 12? Why?

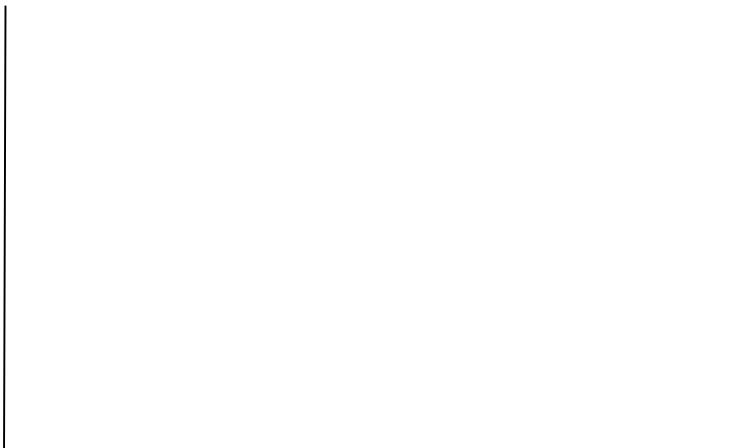




- 2- Can you predict the behavior of “inventory” under the proposed demand shock (please use the space provided below to draw a figure)? If your figure is not similar (only qualitatively, do not worry about quantitative precision) to the figure provided in solution 12? Why?



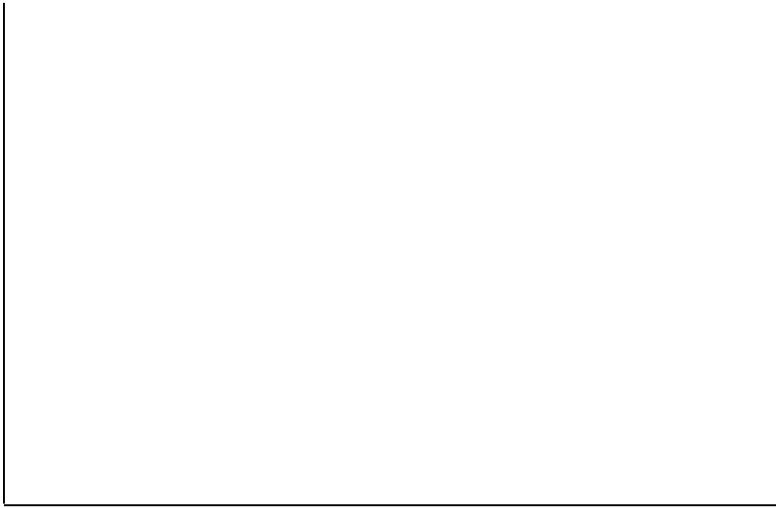
- 3- Can you predict the behavior of “price” under the proposed demand shock (please use the space provided below to draw a figure)? If your figure is not similar (only qualitatively, do not worry about quantitative precision) to the figure provided in solution 12? Why?<sup>18</sup>



---

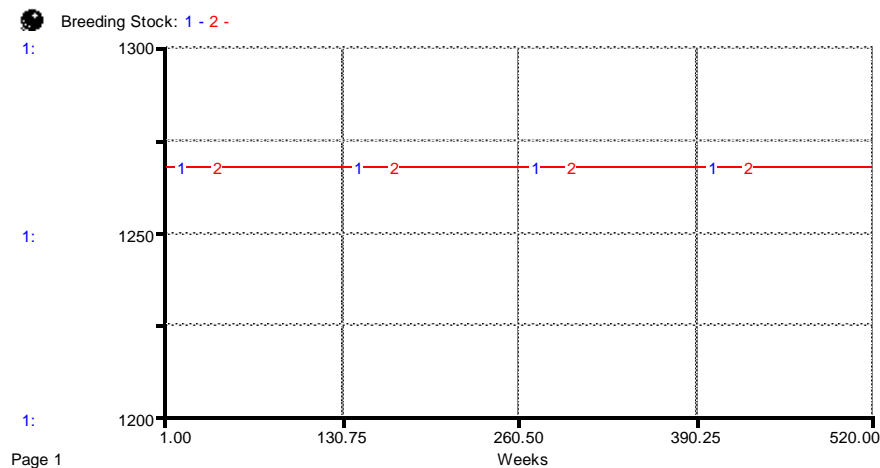
<sup>18</sup> The x-axis of the figure is time in weeks; y-axis is  $\$/\text{pig}$

- 4- Can you predict the behavior of “customer demand” under the proposed demand shock (please use the space provided below to draw a figure)? If your figure is not similar (only qualitatively, do not worry about quantitative precision) to the figure provided in solution 12? Why?<sup>19</sup>



**Solution 12 (in figures of this solution, run 1 represents behavior without a shock; run 2 represents behavior under the shock):**

- 1- We expect that the behavior of “breeding stock” will not change under the demand shock because the producer’s decision to manage pig production (i.e. “indicated breeding stock change rate”) is exogenous. That is, pig supply remains the same because the model does not have “indicated breeding stock change rate” endogenized to ensure that any changes in demand and price are reflected in supply.

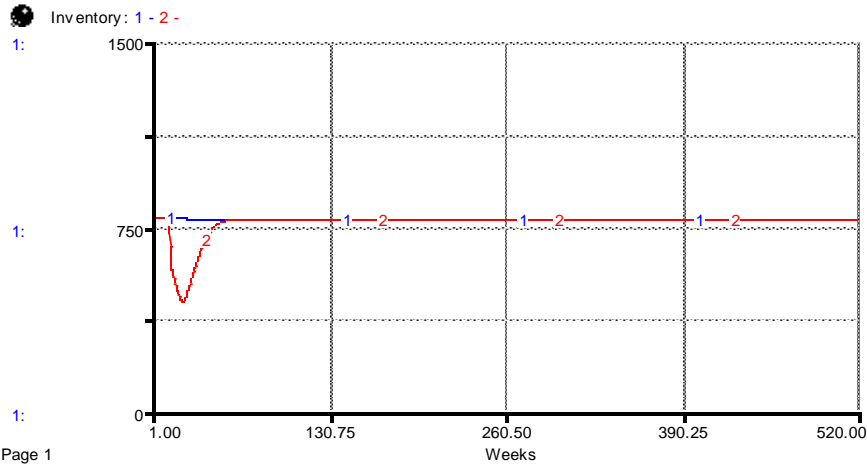


?

**Figure 33: updated breeding stock behavior under a demand shock (run 2)**

<sup>19</sup> The x-axis of the figure is time in weeks; y-axis is pig/week

- 2- During weeks 10 to 12 (20% increase in demand), the inventory level declines to satisfy the 20% increase in demand. However, decreased “inventory” leads to increase in “price” which lowers “customer demand”. This leads “inventory” to head back to the original equilibrium level after some time lag when the demand shock finishes.

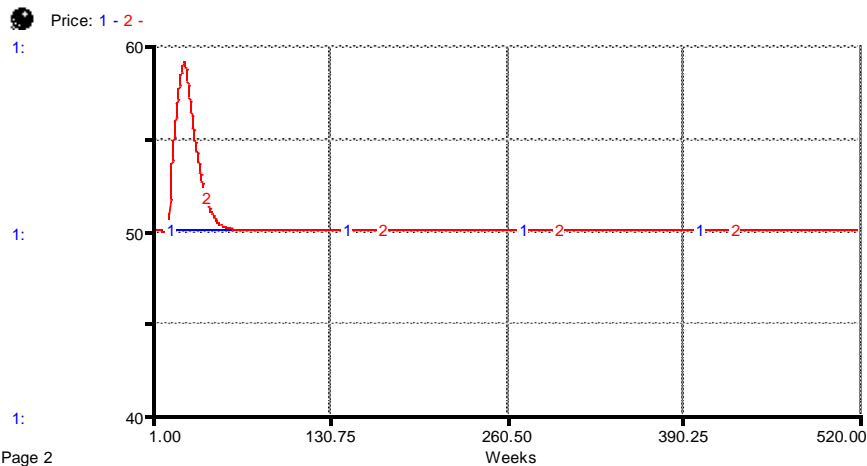


Page 1

?

**Figure 34: updated inventory behavior under a demand shock (run 2)**

- 3- During weeks 10 to 12 (20% increase in demand), “price” increases because of a lower “inventory” level. However, increased “price” leads to lower “customer demand”. This leads “inventory” to head back to the original equilibrium level after some time lag when the demand shock finishes which in turn makes price to move towards the original equilibrium.

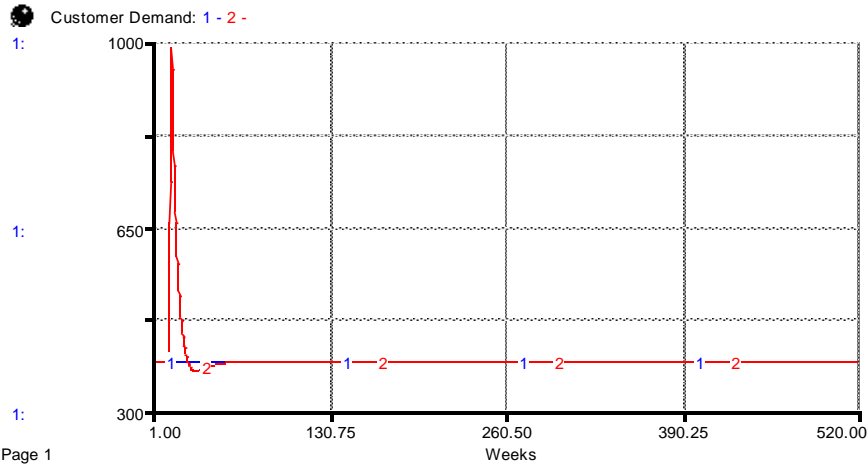


Page 2

?

**Figure 35: updated price behavior under a demand shock (run 2)**

- 4- "Customer demand" increases sharply at the first few weeks of the demand shock (20% increases in demand from week 10 to 12). However, this temporary shock leads to a decline in "inventory" which leads to a higher "price". Higher "price" affects consumers which leads them to lower "customer demand". Lower "customer demand" leads to increase in "inventory" in the next rounds which in turn lowers "price" until "customer demand" moves toward its original equilibrium level.



?

Figure 36: updated customer demand behavior under demand shock (run 2)

**Note:**

Comparing “breeding stock” and “inventory” figures of exercise 9 with exercise 12 shows the impact of endogenizing “customer demand” and “price” on the model behavior at the demand side. However, in both exercises (9 and 12) “breeding stock” behavior remains intact. This is an indication that the producer’s decision to manage supply of pigs to the market is exogenous to the model. It is unrealistic to assume the price movement has no impact on the producer’s decision. Thus, further expansion of the model is needed to comply with real life systems.

To link the movement in pig “price” to producer’s behavior, we need to endogenize “indicated breeding stock change rate”. To do this, let’s define the factors that affect “indicated breeding stock change rate”. These factors include “breeding stock adjustment” and number of pigs in the stock of “growers”. “Breeding stock adjustment” represents the number of pigs producers aim to add to “breeding stock” each period of time. However, the number of pigs that producers can actually add to “breeding stock” depends on number of “grower” pigs available at producer’s facility. Thus, “indicted breeding stock change rate” is a function of “breeding stock adjustment” and “grower” pigs. In your SD model, create a stock named “breeding stock adjustment” and then use a connector to make a connection from “breeding stock adjustment” to “indicted breeding stock change rate”. In a similar vein, make a connection from the stock of “growers” to “indicted breeding stock change rate” using a connector. Earlier in this manual (table 1), “indicted breeding stock change rate” was assigned a fixed value. However, now “indicted breeding stock change rate” is affected by both stocks of “breeding stock adjustment” and “growers”. Therefore, the definition of “indicated breeding stock change rate” needs to be updated as follows:

**Indicated breeding stock change rate** = MAX ((Breeding Stock Adjustment/Growers), 0) {1/week}

We used a MAX function to ensure that “indicted breeding stock change rate” cannot be negative because a breeding pig cannot revert to being a grower (if a breeding pig is not needed, it will be sold). Let’s define factors that affect the stock of “breeding stock adjustment”. We know that stocks can only change through flows. “Breeding stock adjustment” can change in either direction (increase or decrease) each time. Thus, we can model this via a biflow. In your iThink model, drag a flow into the stock of “breeding stock adjustment” and then name it “change in breeding stock adjustment”. Double click on “change in breeding stock adjustment” flow and make it a biflow (note that we already described how to switch a flow to a biflow in iThink earlier in this manual). Multiple factors regulate the biflow of “change in breeding stock adjustment” such as the stock of “breeding stock adjustment”, the gap between actual number of pigs in breeding stock and desired number of pigs in breeding stock, and the time it takes producers to fill the gap between actual and desired breeding stock. In your SD model, make a connection using a connector from “breeding stock adjustment” and the biflow “change in breeding stock adjustment”. Use converters to create a parameter called “gap adjustment time” and a variable called “breeding stock gap”. Then connect both “gap adjustment time” and “breeding stock gap” to the biflow “change in breeding stock adjustment”. The initial value of the stock of “breeding stock adjustment” and the biflow of “change in breeding stock adjustment” are as follows:

**Breeding stock adjustment** = Breeding Stock Gap<sup>20</sup> {pig/week}

---

<sup>20</sup> Note this only represents the initial value. That is, the initial value of “breeding stock adjustment” equals to initial value of “breeding stock gap”.

**Change in breeding stock adjustment** = (Breeding Stock Gap – Breeding Stock Adjustment)/Gap Adjustment Time {pig/week/week}

The variable “breeding stock gap” depends on the producer’s desired number of breeding pigs, number of breeding pigs slaughtered each time, and the actual number of breeding pigs held by a producer. In your SD model, make a connection from “breeding stock” to “breeding stock gap” using a connector. To account for the number of breeding pigs slaughtered each time, create a variable called “perceived breeding stock slaughter rate” to represent the number of pigs slaughtered each time. Then make a connection from the flow of “breeding stock slaughtering” to “perceived breeding stock slaughter rate” using a connector. Create a parameter named “time to perceive slaughter rate” to represent the time that it takes producers to get informed and account for breeding pigs that have been sold and exited the farm. Make a connection from parameter “time to perceive slaughter rate” to “perceived breeding stock slaughter rate” using a connector. Then make a link from “perceived breeding stock slaughter rate” to “breeding stock gap”. Next, use a converter to create a variable named “desired breeding stock”, then connect it using a connector to “breeding stock gap”. Use the following equations to define the elements described above:

**Breeding stock gap** = (Desired Breeding Stock + Perceived Breeding Stock Slaughter Rate) - Breeding Stock {pig/week}

**Perceived breeding stock slaughter rate** = SMTH1 (Breeding Stock Slaughtering, Time to Perceived Slaughter Rate) {pig/week}

**Time to perceive slaughter rate** = 1 {week}

**Gap adjustment time** = 26 {week}

“Desired breeding stock” represents the desired number of breeding pigs producers aim to have. One factor that affects “desired breeding stock” is production capacity (that is, producer’s farm size in terms of number of pigs that can be kept). In your SD model, use a converter to create a parameter called “production capacity” and then connect it to “desired breeding stock” using a connector. Define parameter “production capacity” as:

**Production capacity** = (Initial Demand\*1.1111)\*(52/ (Litter per Year\*Head per Litter))<sup>21</sup> {pig}

We will define “desired breeding stock” after exercise 13. Note: make sure you got all connections especially direction of flows and connectors as shown in figure 37.

---

<sup>21</sup> Production capacity is a constant. The outcome of this equation is used as an initial value. The reason for using this equation is to make sure to be able to run the model in equilibrium if the modeler decided to change parameter value (such as litter per year or head per litter). That is, using equation rather than a direct number updates initial value of the model automatically without a need to check and change initial value of stocks and relevant parameters one by one if modeler decided to use different parameter values.

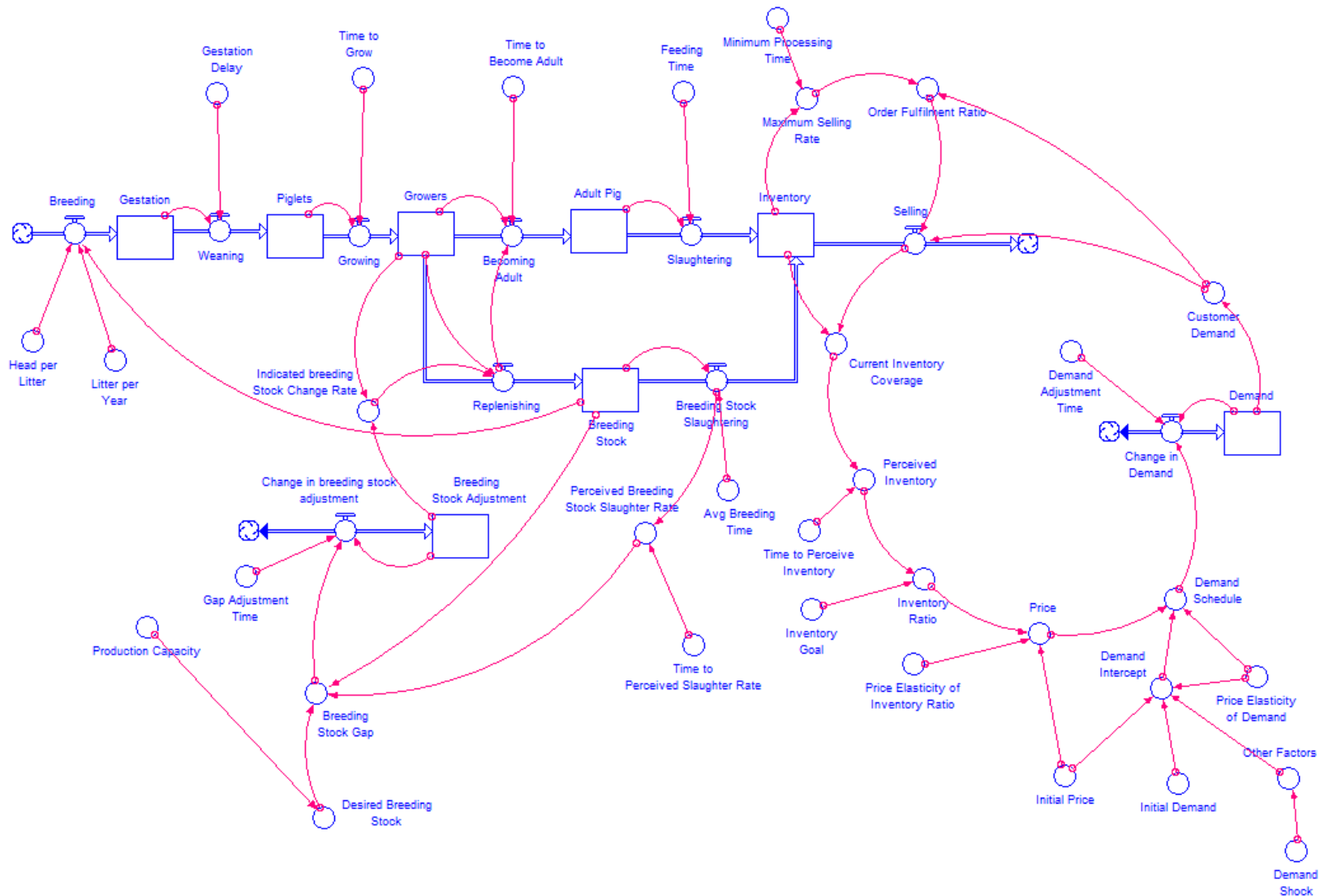


Figure 37: Added structure to the model to address factors that affect the producer’s decision to supply pigs to the market (i.e. “indicated breeding stock change rate”)

**Exercise 13:**

- 1- Would “desired breeding stock” in the latest version of the model shown in figure 37 be an exogenous or endogenous element of the model? Why?

-----

- 2- Should pig “price” in the latest version of the model shown in figure 37 affect “desired breeding stock”? Why?

-----

**Solution 13:**

- 1- “Desired breeding stock” as shown in figure 37 is an exogenous parameter because it is not part of any feedback loops.
- 2- Pig “price” is a very important factor. Pig “price” should definitely affect “desired breeding stock” to ensure that any movement in pig “price” is taken into account by producer’s to manage “breeding stock” and determine pig supply to the market.

In the latest version of the model in figure 37, “desired breeding stock” is exogenous to the model which indicates that the producer’s decision to manage “breeding stock” is made without taking pig “price” into account. This violates real life market dynamics. In a real life markets, price movements along the value chain will influence producer behavior. To introduce the effect of price movements on producer behavior, we need to introduce the investment effects of a pig “price” movement on producer decisions to utilize their farm capacity.

To do this, first, we need to smooth pig “price” over some time period because producers usually do not make decisions on the percentage of breeding pigs kept on farm based on short term (for example, week to week) price movements. Rather, producers perceive pig “price” over some time period before making any decision to avoid sudden reactions to short term price volatility. In other words, if prices keep moving up for a year or so, producer expectations of future profitability of the pig farming business will increase. In turn, farmers might decide to expand their farm capacity to produce more pigs.

To smooth pig “price” movements, create a stock and name it “short term expected price”. Then drag a flow into the stock of “short term expected price” and name it “change in short term expected price”. Since price can either increase or decrease, switch the flow of “change in short term expected price” to a biflow. In order to form an expectation of expected pig price, we need to define a parameter to define the amount of time that it takes producers to smooth their price expectations. Let’s create a parameter called “price adjustment time and assume that it equals to half the length of a pig production cycle (i.e. about 12 weeks). To operationalize the stock and flow structure used to form short term expected price, connect “price”, “short term expected price”, and “price adjustment time” to the biflow of “change in



short run price” using connectors. Use the following equations to define the structure to form short term expected price:

**Short term expected price** = Price {\$/pig}

**Change in short term price** = (Price-Short Term Expected Price)/Price Adjustment time {\$/pig/week}

Note that the initial value of short term expected price is equal to initial value of price. To do this in iThink, double click on the stock of “short term expected price”, then start to write price in the equation box and it will automatically show up on the list of parameters and other elements used in the model. Make sure to select the intended element on the list. Keep in mind that this will only define the initial value (the value at starting time).

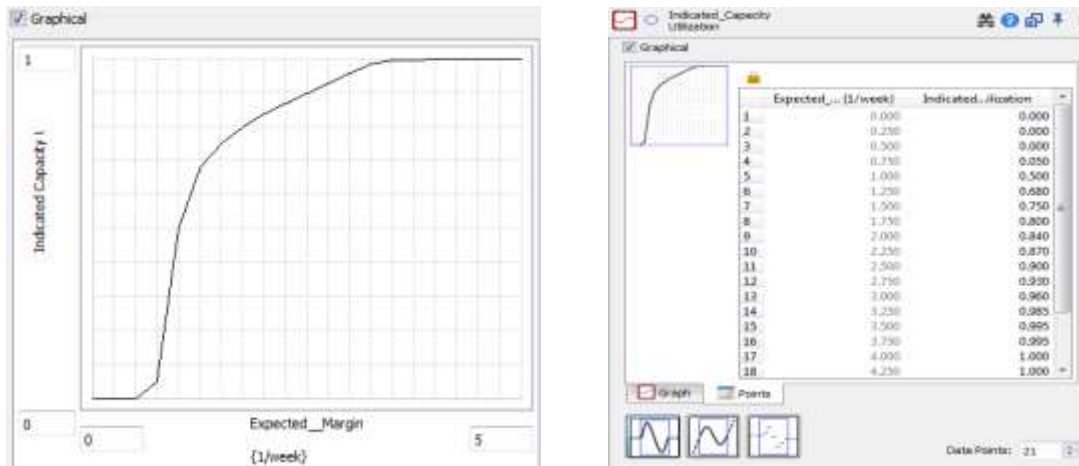
In our model, the profitability of operations will guide decision-making. We define profitability quite simply as the ratio of price to production costs i.e. our expected markup. To make this definition, we first need to define production costs. Let’s use a converter to create a parameter named “production cost”. “Production cost” represents overall costs to produce one pig expressed as the percentage of the initial price. Let’s assume that this is a fixed percentage (0.4) of “initial price”. To define this in iThink, make a link from “initial price” to “production cost”. We then create a second variable using a converter and name it “expected margin”. Make a link using a connector from the stock of “short term expected price” to “expected margin”. Similarly, make a link using a connector from “production cost” to “expected margin”. Define the two equations as:

**Production cost** = 0.4\*Initial Price {\$/pig}

**Expected margin** = Short Term Expected Price/Production Cost {unitless}

“Expected margin” is a profitability indicator that affects producer farm capacity utilization. In other words, if “expected margin” is higher than 1, it indicates that pig farming is a profitable business. We need next to relate profitability to capacity utilization and investment decisions. Farms typically operate at excess capacity so that when prices rise, we would expect farmers to first use unutilized capacity to expand production. When that is exhausted, farmers would have to invest in new capacity. To model this, let’s first use a converter to create a variable named “indicted capacity utilization” to represent producer’s desired capacity utilization based on “expected margin”. Use a connector to make a link from “expected margin” to “indicated capacity utilization”. Let’s define “indicated capacity utilization” in a way that reflects movement in “expected margin”. In other words, as “expected margin” increases, so does “indicated capacity utilization”. In this case, we will use a graphical function (see figure 20 on how to create a graphical function in iThink).

**Indicated capacity utilization**<sup>22</sup> = (graphical function)



“Indicted capacity utilization” represents the immediate reaction by producers to movements in “expected margin”. However, in real life capacity utilization cannot be changed immediately if “expected margin” moves upward or downward. Rather, it takes some time for producers to adjust their farm capacity utilization. To do this in iThink, use a converter to create a variable named “capacity utilization” to represent actual (or current) capacity utilization. Similarly, use a converter to create a parameter named “adjustment time” to represent the amount of time it takes producers to change their farm’s capacity utilization (“adjustment time” is equal to one production cycle, about 26 weeks). Make a link using a connector from “indicated capacity utilization” to “capacity utilization”. Similarly, make a link from “adjustment time” to “capacity utilization”. Use smooth function (SMTH1) to define the equation inside “capacity utilization” to reflect the gradual adjustment process over time.

**Capacity utilization** = SMTH1 (Indicated Capacity Utilization, Adjustment Time) {1/week}

Next, make a link using a connector from “capacity utilization” to “desired breeding stock”. This will close the feedback loop of the impact of price movement on producer’s behavior. Let’s define “desired breeding stock” based on the factors that affect it (“production capacity” and “capacity utilization”).

**Desired breeding stock** = Production Capacity\*Capacity Utilization {pig/week}

Figure 38 shows an expanded version of the model structure that includes effect of pig “price” on producer’s behavior. Hint: make sure you have all connections especially direction of flows and connectors as shown in figure 38.

<sup>22</sup> Note that figure on the left-hand side shows graphical representation of the relationship between “expected margin” and “indicted capacity utilization”. The table on the right-hand side represents precise values on both the x-axis and the y-axis. These are assumed value and can be changed as needed.

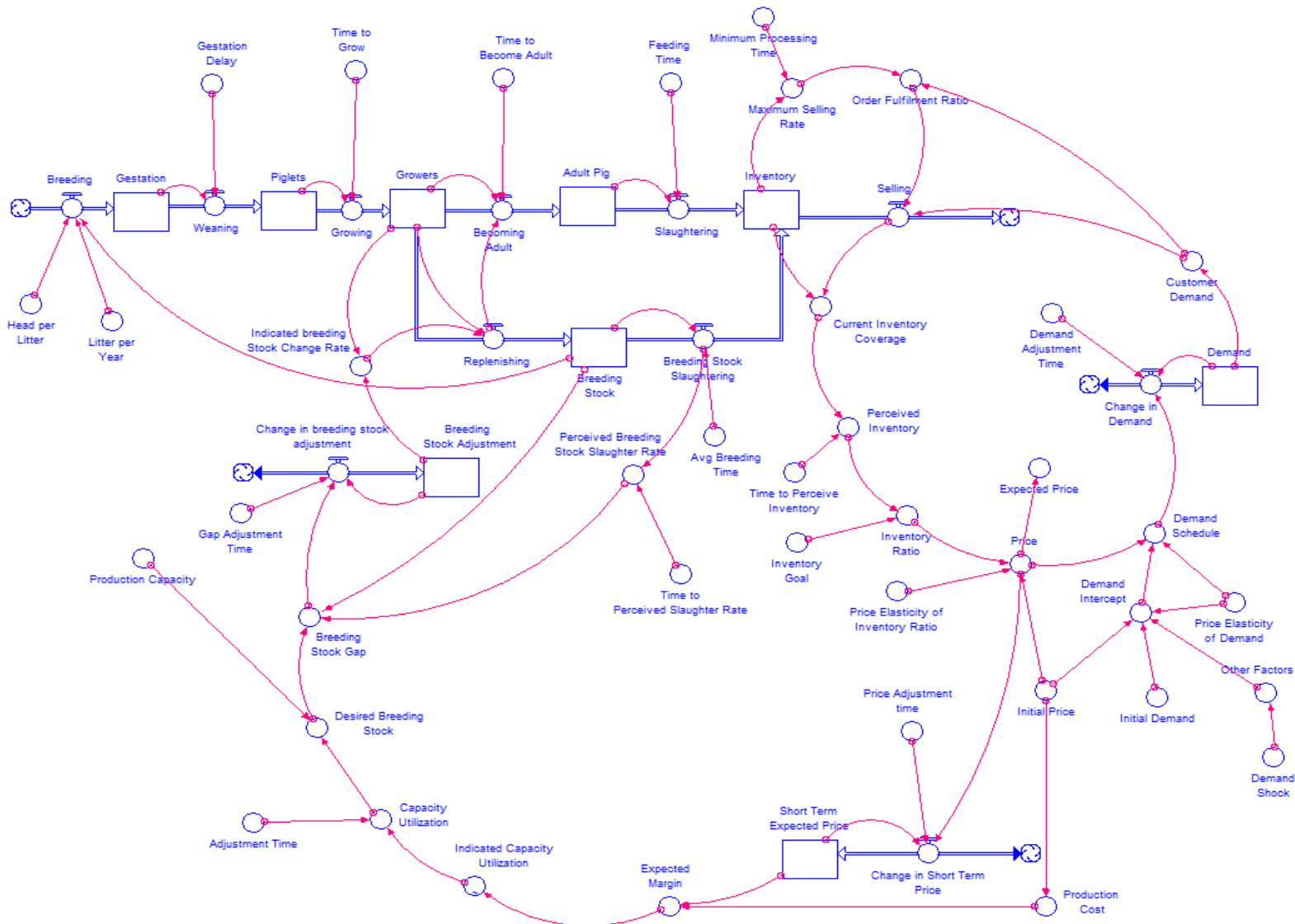


Figure 38: Added structure to present effect of price on producer behavior

**Exercise 14:**

- 1- Can you explain the feedback loop that “desired breeding stock” is part of? Is it a balancing or a reinforcing feedback loop?

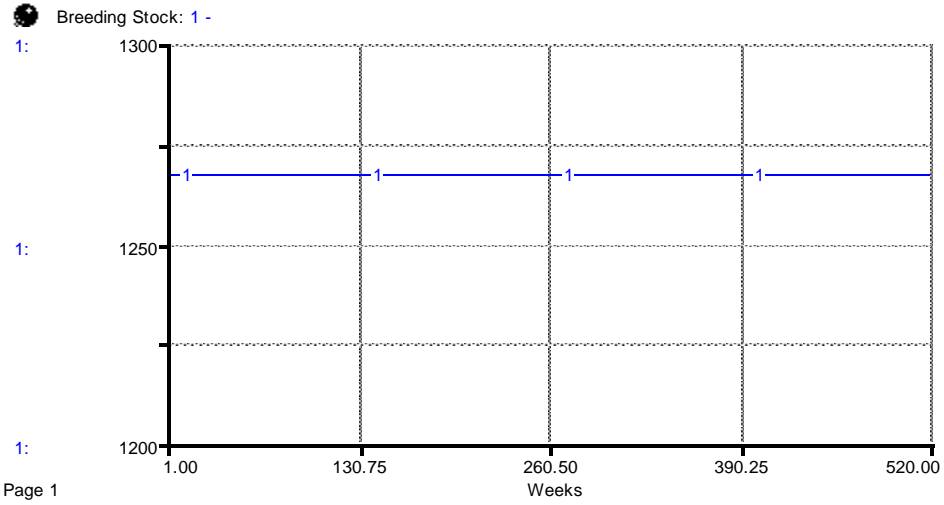
-----

- 2- Is it possible to run the model shown in figure 38 in equilibrium? How?

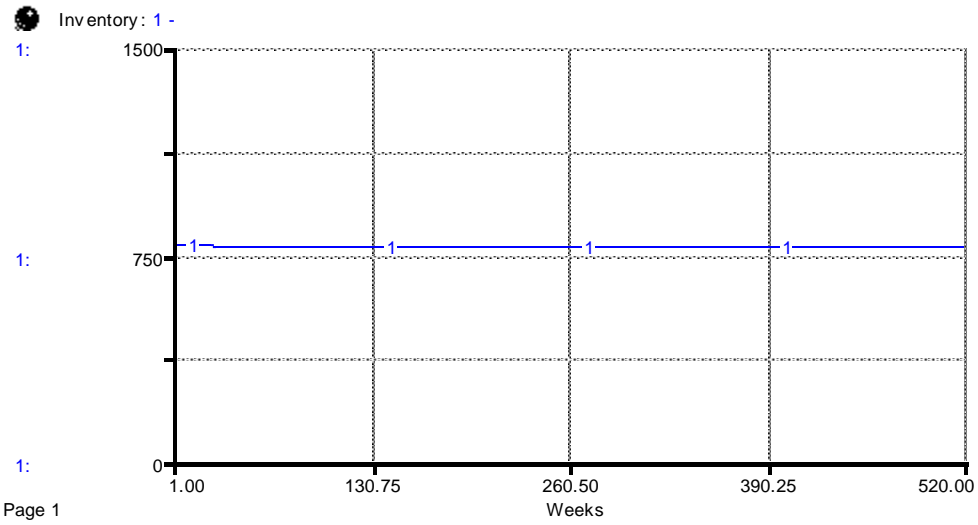
-----

**Solution 14:**

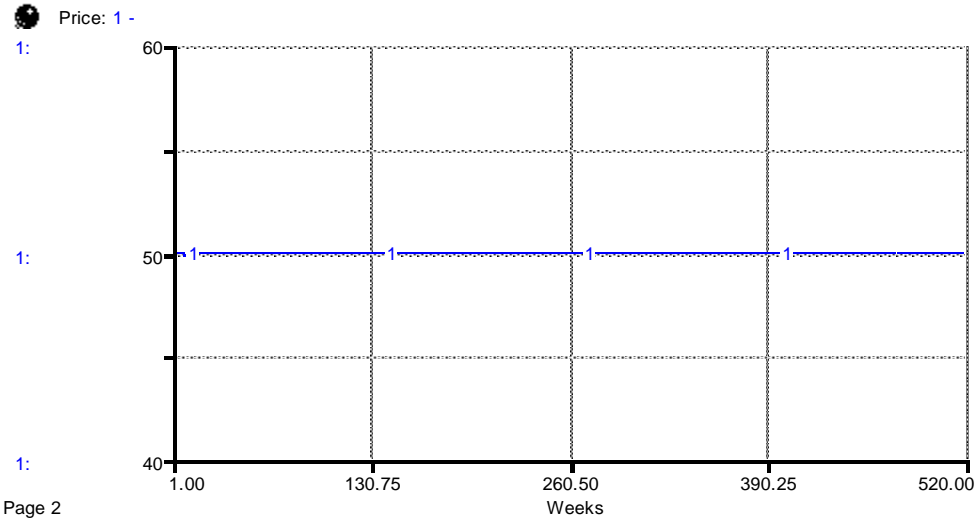
- 1- “Desired breeding stock” is part of an important feedback loop that connects “price” with the supply side of the model. This feedback loop begins where “desired breeding stock” affects “breeding stock” through “breeding stock adjustment”, “indicated breeding stock change rate” and the flow of “replenishing”. Then the change in “breeding stock” goes all the way through aging model (“gestation” to “piglets” to “growers” to “adult pigs”) to affect “inventory”. Change in “inventory” affects “price” through “current inventory coverage” and “inventory ratio” which in turn affects “expected short term price”, “expected margin”, “capacity utilization” and feeds back to “desired breeding stock”. This is a balancing (self-correcting) feedback loop. Please note that we went through a main feedback loop but there are more than one loop that “desired breeding stock” is part of.
- 2- Yes, it is possible to run this model in equilibrium. To run the model in equilibrium, all inflow(s) and outflow(s) of each stock in the model must be equal. Using the equations provided above to operationalize the added structure in figures 37 and 38 will make the model to run in equilibrium. Below are a few figures of the model behavior in equilibrium.



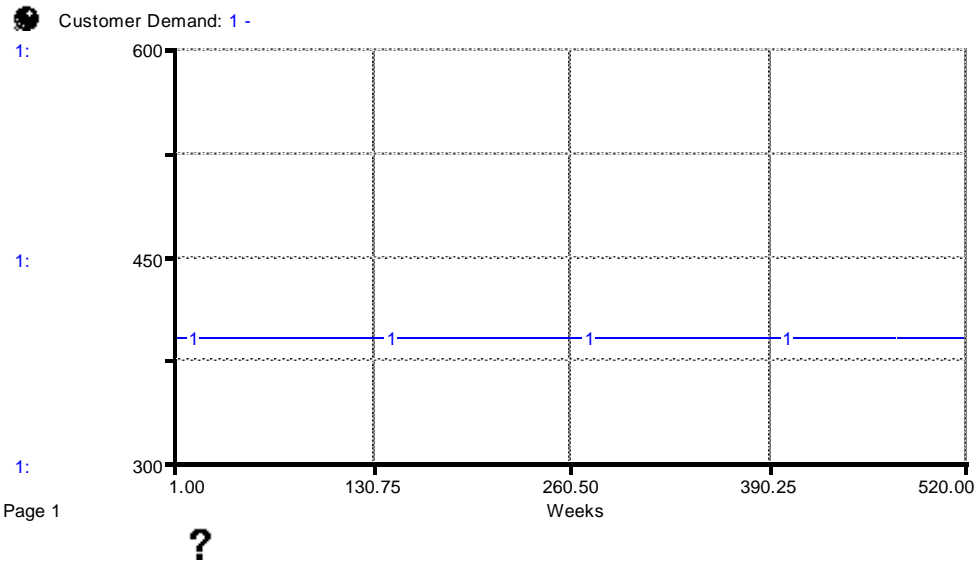
**Figure 39: Breeding stock behavior of the model shown in figure 38**



**Figure 40: Inventory behavior of the model shown in figure 38**



**Figure 41: Price behavior of the model shown in figure 38**

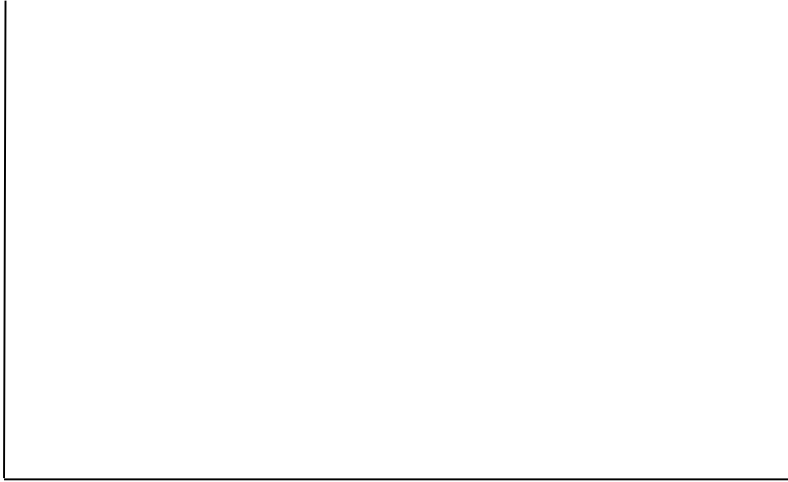


**Figure 42: Customer demand behavior of the model shown in figure 38**

The latest version of the model in figure 38 runs in equilibrium (see figure 39 and 42). To conduct an equilibrium shock and behavior prediction test, let's increase demand by 20% from week 10 to 12 and do the exercise below.

**Exercise 15:**

- 1- Can you predict the behavior of “breeding stock” under the proposed demand shock (please use the space provided below to draw a figure)? If your figure is not similar (only qualitatively, do not worry about quantitative precision) to the figure provided in solution 15? Why?



- 2- Can you predict the behavior of “inventory” under the proposed demand shock (please use the space provided below to draw a figure)? If your figure is not similar (only qualitatively, do not worry about quantitative precision) to the figure provided in solution 15? Why?



- 3- Can you predict the behavior of “price” under the proposed demand shock (please use the space provided below to draw a figure)? If your figure is not similar (only qualitatively, do not worry about quantitative precision) to the figure provided in solution 15? Why?<sup>23</sup>



- 4- Can you predict the behavior of “customer demand” under the proposed demand shock (please use the space provided below to draw a figure)? If your figure is not similar (only qualitatively, do not worry about quantitative precision) to the figure provided in solution 15? Why?<sup>24</sup>



---

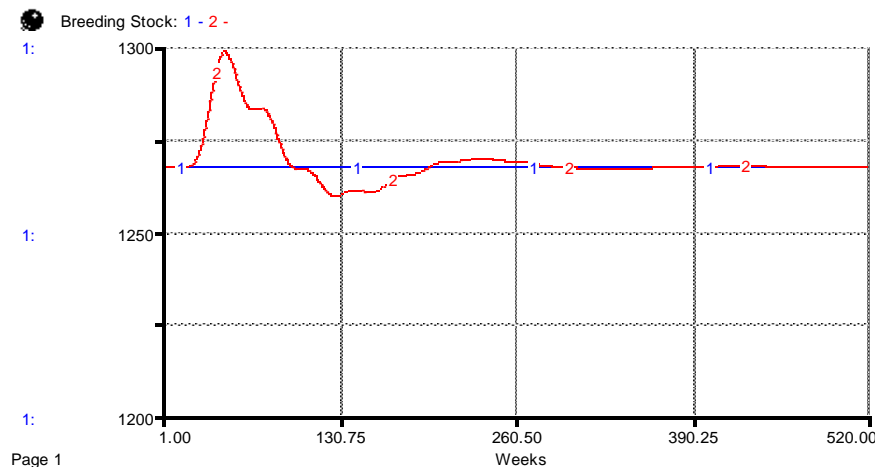
<sup>23</sup> The x-axis of the figure is time in weeks; the y-axis is \$/pig

<sup>24</sup> The x-axis of the figure is time in weeks; the y-axis is pig/week



**Solution 15:**

- 1- “Breeding stock” reacts smoothly to a 20% increase in demand and takes a long period of time to get back to the original equilibrium level after the demand shock is over. A 20% increase in demand leads to a higher pig “price” which in turn motivates producers to increase “breeding stock” for future investments to increase supply in the future. However, there is a long time lag (one production cycle) from the moment producers make a decision until the consequences of that decision to reach market (i.e. “inventory”). Therefore, when the demand shock (week 10 to 12) finishes, it takes a long time for “breeding stock” to reach original equilibrium because there are pigs in the supply line that will reach market and “breeding stock” with a lag. Note here that “price” affects both the demand (i.e. “customer demand”) and the supply (i.e. “indicated breeding stock change rate”) side at the same time and both sides try to get the model back to the equilibrium point when the demand shock is over. But because there are time lags (especially in the supply side, longer than 26 weeks), the model oscillates slightly until it heads back toward its original equilibrium level.

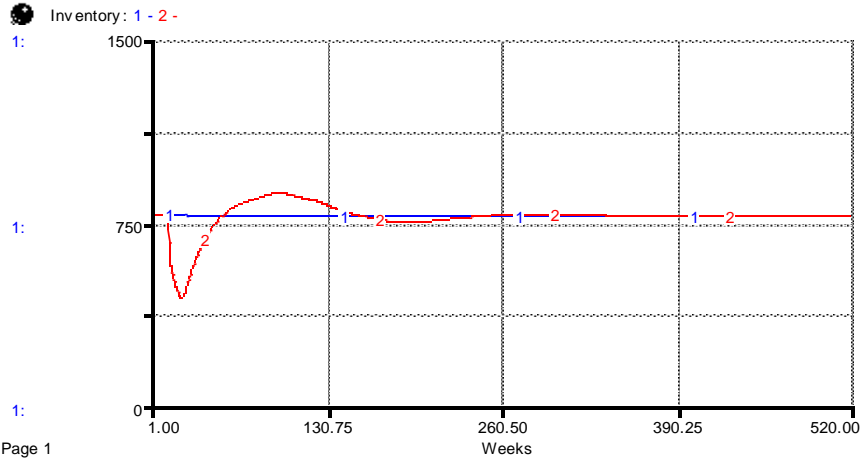


Page 1

?

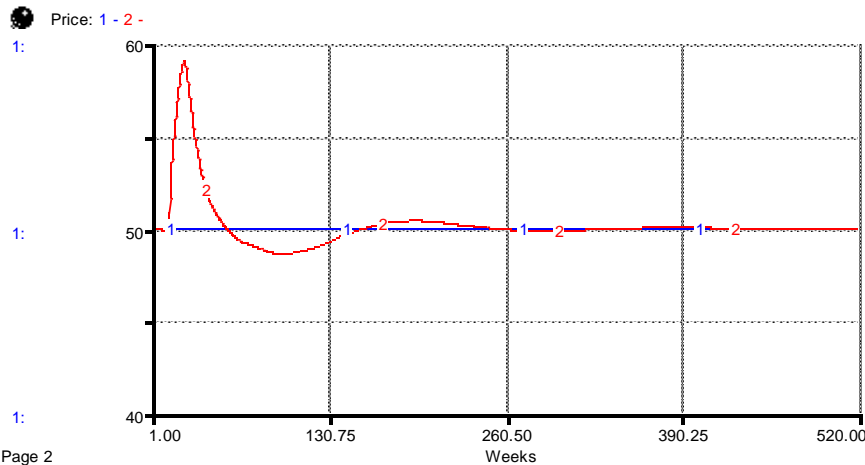
**Figure 643: updated breeding stock behavior under demand shock (run 2)**

- 2- During weeks 10 to 12 (20% increase in demand), the “inventory” level declines to satisfy the 20% increase in demand. However, on the demand side, decreased “inventory” leads to an increase in “price” which lowers “customer demand”. This leads “inventory” to head back to the original equilibrium level after some time lag when the demand shock finishes. Unlike exercise 12 – question 2, “price” movement not only affects the demand side, but also the supply side. Therefore, the higher pig “price” induced by the demand shock motivates producers to increase “breeding stock” to increase supply in the future. This indicates that the higher “price” leads to lower “customer demand” but triggers producers to produce more in expectation of a higher “expected margin” in the future. However, unlike the demand side where “customer demand” reacts quickly to higher “price”, the consequences of producers decisions (i.e. “indicated breeding stock change rate”) to increase “breeding stock” take longer to reach the market, given biological delays equivalent to one production cycle. Therefore, “inventory” declines during the demand shock, then heads towards the original equilibrium level but slightly overshoots because of pigs that are already in the supply line as a consequence of past decisions by producers. Eventually, excess pigs in the supply line are absorbed by the market and the model moves back towards its original equilibrium.



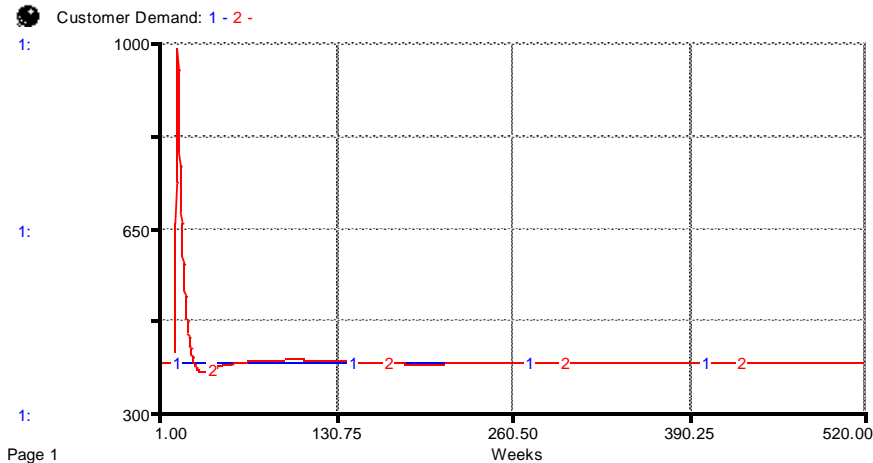
**Figure 44: updated inventory behavior under demand shock (run 2)**

3- Similar to points 1 and 2, the demand shock (weeks 10 to 12) leads to a higher pig “price”. Pig “price” moves smoothly towards original equilibrium after the shock but slightly undershoots because excess pigs in the supply line increase “inventory” level that leads to a slightly lower “price” until the excess pigs in the supply line are absorbed by the market.



**Figure 45: updated price behavior under demand shock (run 2)**

4- The spike in “customer demand” is the demand shock (weeks 10 to 12). An increase in “customer demand” leads to a higher pig “price” which leads to a quick consumer reaction to lower demand. However, producer reactions to higher pig “price” leads to an excess of pigs in supply line as a consequence of investment decisions to increase supply. This leads to a higher “inventory” and, hence, a slightly lower pig “price”. This in return leads to slightly higher “customer demand” until excess pigs in the supply line are absorbed by the market.



?

**Figure 46: updated customer demand behavior under demand shock (run 2)**

### Model extensions – integrating external animal health influences to the value chain

So far, we have showed a simple pig value chain model of pig production and marketing and provided insights by which such a model could be expanded to add other market channels and actors. This model has been used to highlight and simulate a few simple scenarios and even in its simple form demonstrates the power of SD models in analyzing different value chain phenomena.

In this section, we add an external animal health component to the model to simulate the impacts that animal diseases might have on the value chain and, in turn, how changes in the value chain engendered by disease could influence the evolution of disease in future periods (Rich 2007; Rich and Roland-Holst 2013). The basis for our disease model is the standard SIR (Susceptible-Infectious-Recovered) model from the epidemiology literature (Anderson and May 1991). This model separates pig population into three groups.

- 1- Susceptible (S): disease free pigs but vulnerable to be infected.
- 2- Infectious (I): diseased pigs.
- 3- Recovered (R): diseased pigs that become disease free.

Pigs move from one stage to another depending on their health condition. In the meantime, new pigs enter the model through births and adult pigs move out of the model through slaughtering (these two flows are modeled in the value chain section). The differential equations that characterize the S-I-R model are presented below in equation, where  $S$ ,  $I$ , and  $R$  are the population states as defined above and the parameters  $\beta$  and  $\alpha$  represent the transition rates from Susceptible to Infected and Infected to Removed, respectively:

$$\frac{dS}{dt} = -\beta SI$$

$$\frac{dI}{dt} = -\alpha I + \beta SI$$

$$\frac{dR}{dt} = \alpha I$$

The model above is simplistic in the sense that it assumes a closed population. In reality, however, animals will enter and leave the population based on migration, births, market movements (e.g., sales of animals for slaughter or export), and mortality that are either attributable to disease or natural. In addition, control measures such as vaccination will further modulate the dynamics of the system – in such cases, there will be an additional transition between susceptible animals and removed animals that will depend on the rate of vaccination. We will adapt these changes – and their interface with the value chain – in our model.

Consider the questions in the exercise below.

**Exercise 16:**

- 1- Would “Susceptible population”, “infectious population” and “recovered population” be stocks or flows? Can you draw them in space provided below (or on iThink software)?

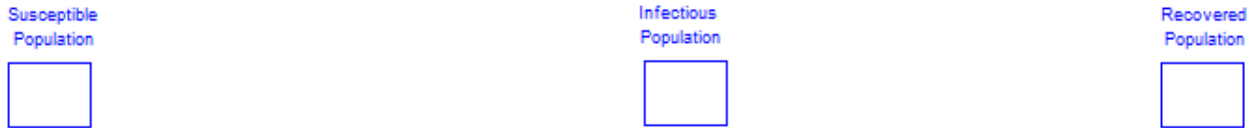
-----

- 2- Would the movements among “susceptible population”, “infectious population” and “recovered population” be stocks or flows? Can you add this to the figure you drew in first question (or in the iThink software)?

-----

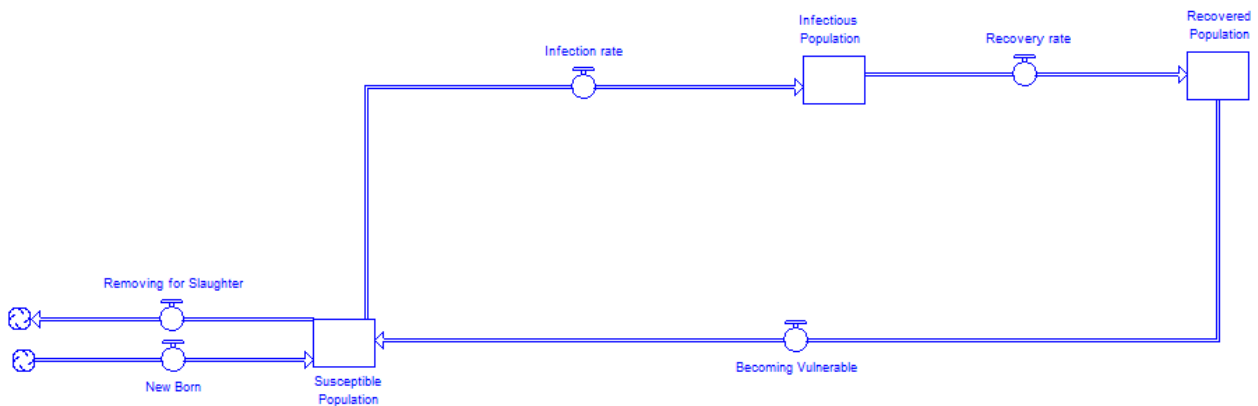
**Solution 16:**

- 1- “Susceptible population”, “infectious population” and “recovered population” are stocks; they are the number of pigs in each state at time  $t$ . In your SD model, use a stock to create a stock named “susceptible population”. Repeat the process to create two more stocks named “infectious population” and “recovered population”.



**Figure 47: Stocks of susceptible, infectious, and recovered pig populations**

- 2- First, we will have the straightforward connections: there will be a flow from susceptible to infectious (“infection rate”) and from infectious to recovered (“recovery rate”). These flows can be deduced from the differential equations given above. However, there will be other possible flows too. In particular, consider possible other flows into the susceptible population. Depending on the type of disease, it is possible that recovered pigs could become susceptible again if immunity from disease wanes over time. This means there could be a flow from the recovered population to susceptible (“becoming vulnerable”). Second, consider a system that is not closed (as opposed to the standard SIR model which assumes a closed system). In this case, we could see the entry of both newly born pigs into the susceptible population and the exit of pigs sold for slaughtered exiting the susceptible stock. See figure 48 below for the representation of these flows in iThink.



**Figure 48: Stock and flow structure of the SIR model**

Let’s next address factors that affect these flows as illustrated in figure 48. The flow “Infection rate” depends on multiple factors such as the size of vulnerable pig population (i.e. “susceptible population”), the size of the diseased pig population (i.e. “infectious population”), how often pigs contact each other (i.e. “contact rate”), the probability of catching disease if an infected pig contacted a healthy one (i.e. “infectivity”), and the total pig population.

To define the “infection rate” in your model, first, use a converter to create a variable named “total pig population” to represent the total pig population from the value chain model. Now, use connectors to

make connections from each stock that represents the different pig age groups (“breeding stock”, “piglets”, “growers” and “adult pigs”) in the value chain model to the variable “total pig population”. Then make a link using connector from “total pig population” to the flow of “infection rate”. Similarly, use a connector to make a link from the stock “infectious population” to the flow of “infection rate”. Then make a connection from the stock “susceptible population” to the flow of “infection rate”. To relate the infectivity of disease to the flow of “infection rate”, use a converter to define a parameter named “infectivity”. Then link “infectivity” using a connector to the flow “infection rate”. Similarly, use a converter to create a parameter named “contact rate”. Finally, use a connector to make a link from “contact rate” to “infection rate”.

The equations below define this added structure in SIR model:

**Infection rate** = Susceptible Population\*Contact rate\*Infectivity\*Infectious Population/Total Pig Population {pig/week}

**Total population** = Piglets + Growers + Adult Pig + Breeding Stock {pig}

**Infectivity** = 0.05 {unitless}

**Contact rate** = 4 {pig/week}

**Susceptible population** = Total Pig Population – Infectious Population – Recovered Population<sup>25</sup> {pig}

**Infectious population** = 0 {pig}

**Recovered population** = 0 {pig}

Both of the flows “recovery rate” and “becoming vulnerable” represent respectively the number of animals that recover from disease and those recovered that will return to “susceptible population”. “Recovery rate” depends on the stock of “infectious population” and the amount of time that it takes an infectious pig to become disease free. In iThink, use a converter to create a parameter named “average duration of illness” and then make a connection from “average duration of illness” to the flow “recovery rate”. Similarly, make a link using a connector from the stock of “infectious population” to the flow “recovery rate”.

The flow “Becoming vulnerable” depends on the stock “recovered population” (i.e. number of pigs in the recovered stock) and the amount of time that it takes a recovered pig to become vulnerable again. In other words (and depending on the disease), pigs that recover from a disease will eventually become vulnerable again. In iThink, use a converter to create a parameter named “average immunity time” to represent the time a pig remains immune from disease after recovering from it. Then make a link using a connector from “average immunity time” to the flow “becoming vulnerable”. In a similar vein, make a link using a connector from the stock “recovered population” to the flow “becoming vulnerable”. Use equations below to add the new structure to the SIR model:

**Recovery rate** = Infectious Population/Average duration of illness {pig/week}

**Becoming vulnerable** = Recovered Population/Average Immunity Time {pig/week}

---

<sup>25</sup> Note that under this equation, we assume that all pigs are in “susceptible population” because we assume that both stocks of “infectious population” and “recovered population” are zero. But if initial value of both or either “infectious population” and “recovered population” to more than 0, then the initial value of “susceptible population” will automatically update itself. This way, we will be sure that we have the same number of pigs in both value chain and SIR model. In other words, we will be sure that we will not have more “susceptible pigs” than pigs exist in the value chain model or more “infectious pigs” than number of pigs a farmer has.

**Average duration of illness** = 12 {week}

**Average immunity time** = 1 {week}

The two flows “new born” and “removing for slaughter” on the left-hand side of the SIR model are exogenous and are directly taken from the value chain model constructed earlier in this manual (refer to figure 38 to see the latest version of the model). To connect value chain model to the SIR model, make a link using a connector from the flow of “weaning” in the value chain model to the inflow “new born”. In other words, the moment a pig is born, it becomes susceptible to disease. In a similar vein, to remove pigs that are slaughtered from the SIR model, make a link using a connector from the flow “slaughtering” in the value chain model to the flow “removing for slaughter”. In a similar vein, make a link using a connector from the flow “breeding stock slaughtering” in the value chain model to the flow of “removing for slaughter”.

**New born** = Weaning {pig/week}

**Removing for slaughter** = Slaughtering + Breeding Stock Slaughtering {pig/week}

Compare your model with that illustrated below in figure 49.

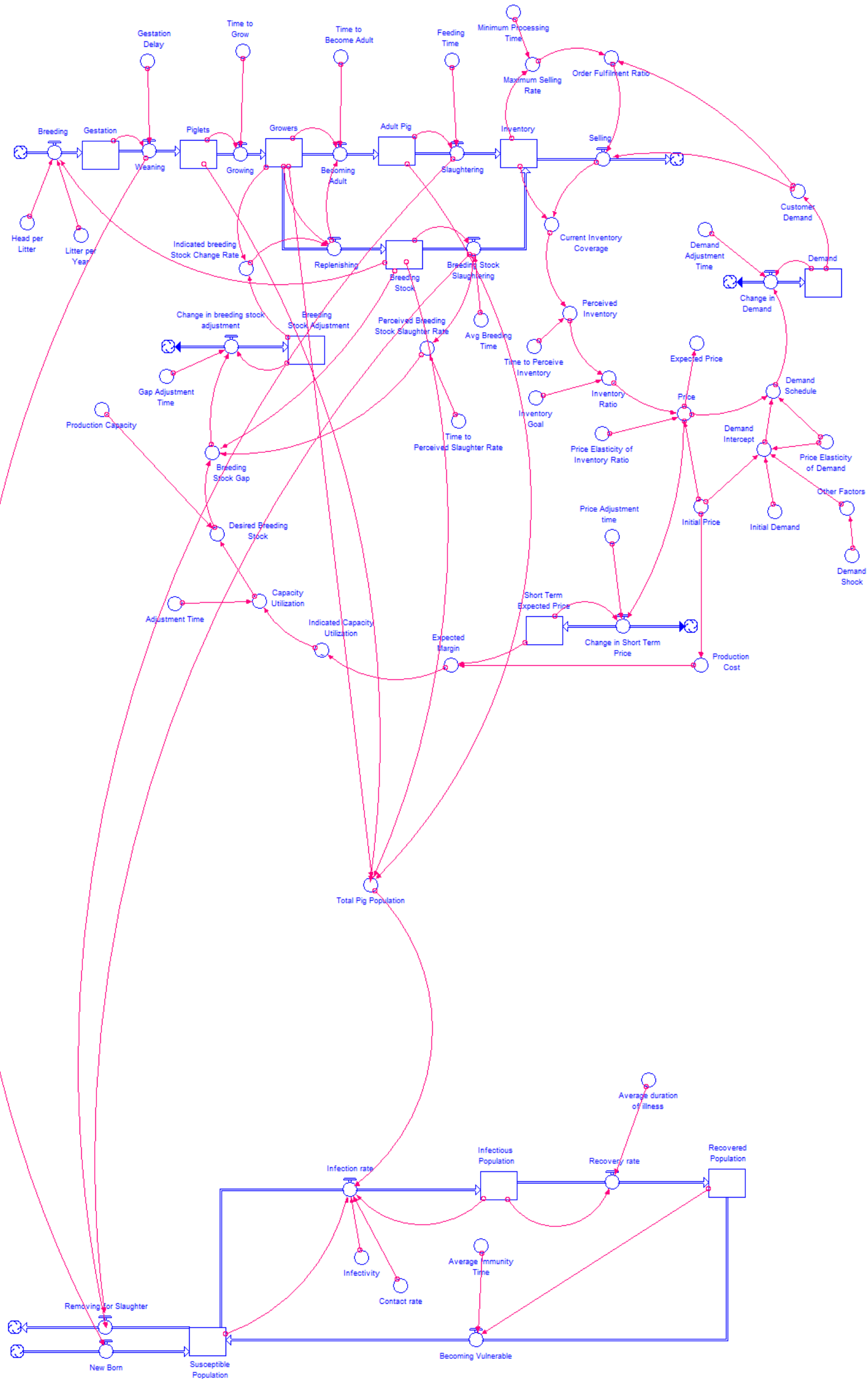
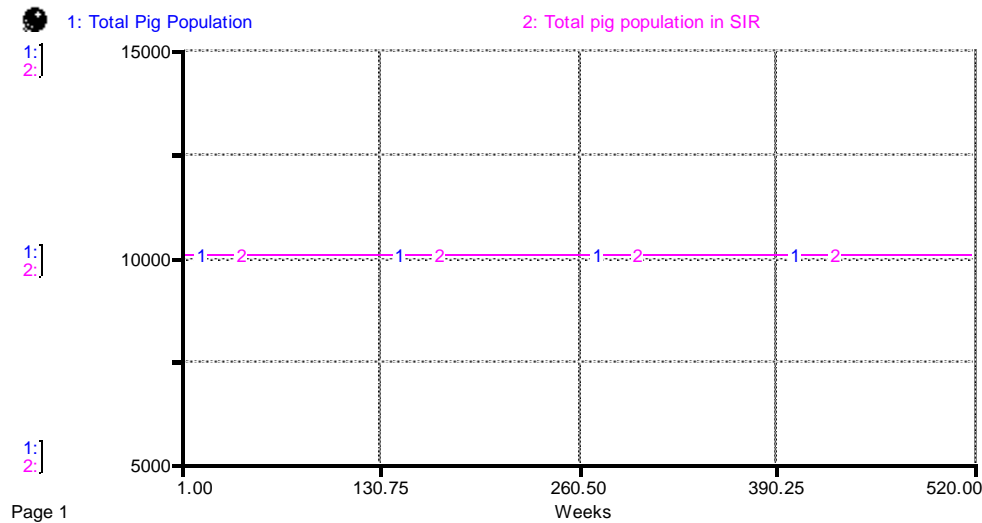


Figure 49: Adding the SIR model to the value chain model



An important check of this model is that the number of pigs in the SIR model **must** be equal to the number of pigs in the value chain model. That is, “breeding stock” + “piglets” + “growers”+ “adult pigs” must equal the sum of “susceptible population”, “infectious population” and “recovered population”. In order to ensure that both the value chain model and the SIR model are consistent, we can check this by adding a graph to our model that plots “Total pig population” (already defined) to a new parameter called “Total pig population in SIR”, defined as “susceptible population” + “infectious population” + “recovered population”<sup>26</sup>. If we run this in iThink, we should get a graph like that shown in figure 50 where both population levels are equal.



Page 1

?

**Figure 50: Consistency check of total pig population in the value chain model and total pig population in the SIR model**

### The impact of animal health policy on the value chain

The latest version of the model in figure 49 integrates both the value chain and the SIR model. The SIR model in its current form does not include any structure to manage or control disease. In this section, we will design two different interventions to manage animal disease and simulate their impacts on both the evolution of disease and the pig value chain.

- 1- Vaccination policy: vaccinating pigs in “susceptible population” before getting infected.
- 2- Quarantine policy: removing pigs in “infectious population” to limit contact with healthy pigs.

#### Vaccination policy

This policy removes pigs from “susceptible population” to a new stock “vaccinated population” where pigs are immune from an infectious disease. To introduce a vaccination policy to the SIR model, create a stock named “vaccinated population”. Then create a flow from the stock of “susceptible population” to the stock of “vaccinated population”, and name this flow “vaccinating”.

<sup>26</sup> Note: you can create a variable called “total pig population in SIR” and add all SIR pig population stocks to it. For simplicity of figure, we have not included “total pig population in SIR” in figure 49.

The flow “vaccinating” depends on three factors. First, it will depend on the number of pigs that can be vaccinated each time. Create a converter named “vaccination rate”. “Vaccination rate” will be regulated by (1) the “time to deploy vaccine” (i.e. time needed to vaccinate all pigs), (2) “fraction vaccinated” (i.e. the percentage of the vulnerable pig population we aim to vaccinate, where 1 denotes 100% and 0 denotes 0%), and (3) the number of pigs that potentially can be vaccinated (i.e., pigs in the stock “susceptible population”). Create two converters “time to deploy vaccine” and “fraction vaccinated”. Then connect “susceptible population”, “time to deploy vaccine”, and “fraction vaccinated” to the variable “vaccination rate”.

The equations below summarize the additions noted above:

**Vaccination rate** = Fraction Vaccinated\*(Susceptible Population/Time to Deploy Vaccine) {pig/week}  
**Time to deploy vaccine** = 52 {week}  
**Fraction vaccinated** = 0.3 {unitless}

Second, while “vaccination rate” represents the ability to deliver vaccines each time, it might not be guaranteed that vaccinated pigs will be fully immune from disease. In iThink, use a converter to create a parameter named “effectiveness of the vaccine”. Make a link from the parameter “effectiveness of the vaccine” to the flow “vaccinating”. For now, let’s assume that the vaccine is 100% effective. We enter a value of 1 to denote this percentage effectiveness.

**Effectiveness of the vaccine** = 1 {unitless}

Third, vaccination is an expensive procedure and is not implemented until a certain percentage of the pig population actually catches an infectious disease. Therefore, let’s create a variable called “threshold to start vaccination” that equals to 0.5% of “total pig population”. What this means is that if 0.5% of “total pig population” is infected (i.e. the number of pigs in “infectious population”), then the vaccination policy is automatically launched. Use a connector to link “threshold to start vaccination” to the flow “vaccinating”. The equation “threshold to start vaccination” is thus:

**Threshold to start vaccination** = 0.005\*Total Pig Population {pig}

Finally, we define the flow “vaccinating” as follows:

**Vaccinating** = (IF (Infectious Population>Threshold to Start Vaccination) THEN (Vaccination Rate\*Effectiveness of the Vaccine) Else (0)) {pig/week}

This equation says that when number of infected pigs (i.e., the number of pigs in “infectious population”) exceeds 0.5% of total pig population, then the vaccination policy is launched. Otherwise, there is no vaccination policy implemented and the rate of vaccinating is zero

Figure 51 below summarizes graphically the connections generated from the instructions above.

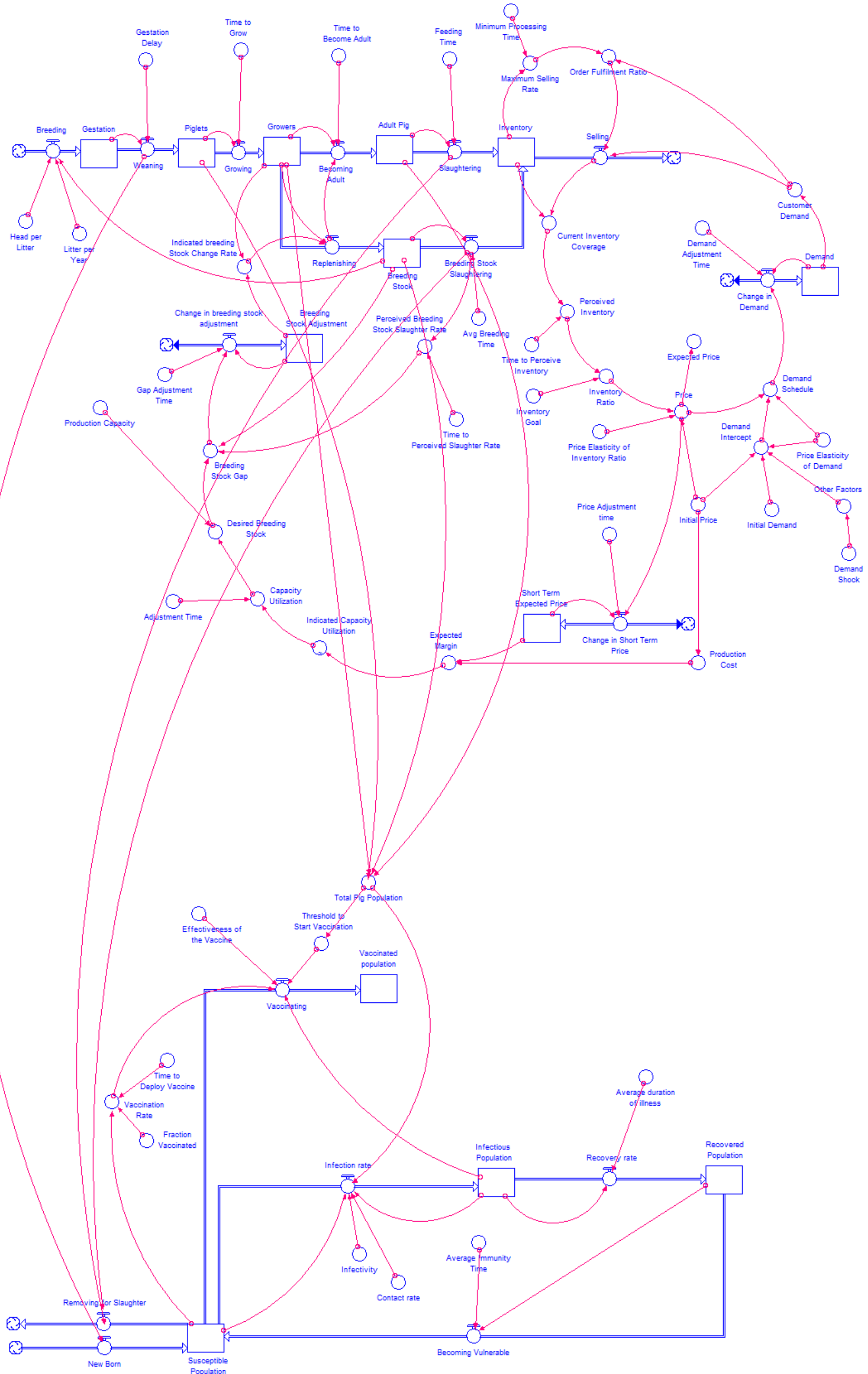


Figure 51: Implementing a vaccination policy in the integrated SIR-value chain model

## Quarantine policy

In this policy, infected pigs are quarantined. To do this, we remove infected pigs from “infectious population” to “quarantined population” using a new flow called “quarantine rate”. The purpose of placing infected pigs in quarantine is to restrict their contact rate with healthy pigs.

In your model, create a stock called “quarantined population”. Then create a flow from the stock of “infectious population” into the stock of “quarantined population” and name it “quarantine rate”. The flow “Quarantine rate” depends on the stock of “infectious population” (i.e. number of infected pigs), “quarantine time” (i.e. the time that it takes to quarantine a pig from the moment it enters the stock of “infectious population”), and “quarantine fraction” (i.e. what percentage of the infectious pig population we aim to place in quarantine, where 1 equals 100% and 0 equals 0%). Use a connector to link the stock “infectious population” to the flow “quarantine rate”. Create converters called “quarantine time” and “quarantine fraction”. Use a connector to link “quarantine fraction” to the flow “quarantine rate”. Similarly, use connectors to link the parameter “quarantine time” to the flow “quarantine rate”.

Define the equations below to add the quarantine policy to the SIR model:

**Quarantine rate** = ((Infectious Population\*Quarantine Fraction)/Quarantine Time)  
{pig/week}

**Quarantined population** = 0 {pig}

**Quarantine fraction** = 0.3 {unitless}

**Quarantine time** = 8 {week}

Pigs that are placed in quarantine will remain quarantined until they recover and become disease free. In other words, when infected pigs in “quarantined population” become disease free, they move to the stock of “recovered population” via the flow of “quarantine population recovery rate”. In your SD model, create a flow out of the stock “quarantined population” into the stock of “recovered population” and name it “quarantine recovery rate”. “Quarantine recovery rate” will depend on the number of pigs in the stock of “quarantined population” and the parameter “average duration of illness”<sup>27</sup> that represents the average period of time it takes each infected pig to recover from disease. Use the equation below to define the flow of “quarantine recovery rate” and make the relevant links using connectors to generate this equation:

**Quarantine recovery rate** = Quarantined Population/Average duration of illness {pig/week}

Figure 52 below summarizes graphically the connections generated from the instructions above.

---

<sup>27</sup> This parameter is already added to the model.

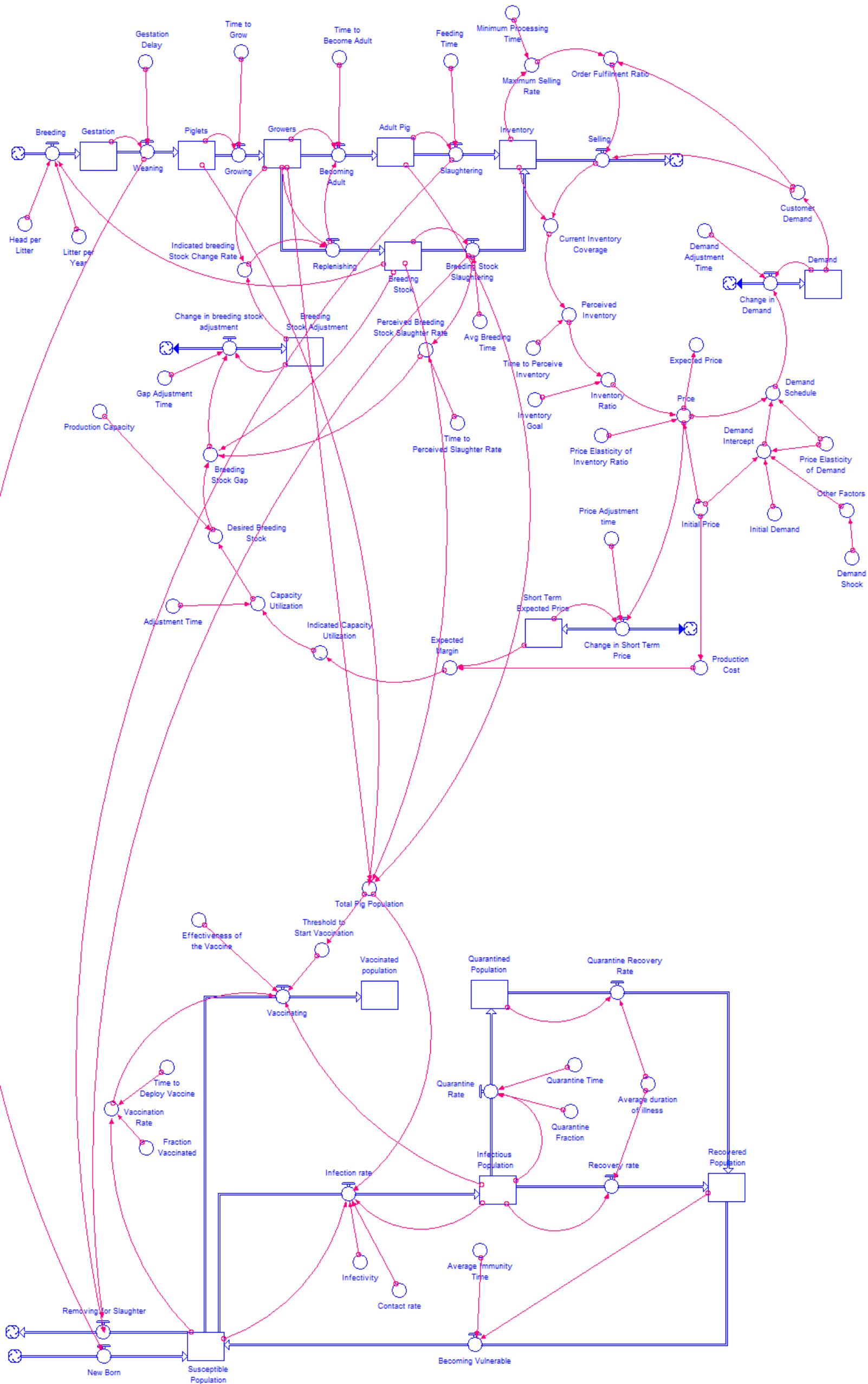


Figure 52: Implementing a quarantine policy in the integrated SIR-value chain model

## Policy costs

Vaccination and quarantine policies are potentially expensive interventions. The costs of any policy intervention need to be included in the model to analyze the cumulative costs of the interventions and their cost over time.

To include policy costs in our model, create two stocks, one that is named “cumulative vaccine cost” and the other named “cumulative quarantine cost”. Similarly, drag a flow into the stock of “cumulative vaccine cost” and name it “vaccine cost over time”. Similarly, drag a flow into the stock of “cumulative quarantine cost” and name it “quarantine cost over time”. Note that we used inflows because neither quarantine cost nor vaccine costs can be negative. “Vaccine cost over time” will depend on the number of pigs that are vaccinated each period (i.e. flow of “vaccinating”) times the “average cost of vaccine per pig”. Use connectors to link the flow of “vaccinating” to the flow of “vaccine cost over time”. Then use a converter to create a parameter named “average cost of vaccine per pig”.

“Quarantine cost over time” will depend on the number of pigs that are placed in quarantine over time (i.e. the flow “quarantine rate”) times the “average quarantine cost per pig”. Use a connector to make a link from the flow “quarantine rate” to the flow “quarantine cost over time”. Then use a converter to create a parameter named “average quarantine cost per pig”. Then use a connector to link “average quarantine cost” to the flow “quarantine cost over time”.

Use the following equations to define the cost features defined above:

**Vaccine cost over time** = Vaccinating\*Average vaccine cost per pig {\$/week}

**Cumulative vaccine cost** = 0 {\$}

**Average vaccine cost per pig** = 5 {\$/pig}

**Quarantine cost over time** = Quarantine Rate\*Average Quarantine cost per pig {\$/week}

**Cumulative quarantine cost** = 0 {\$}

**Average quarantine cost per pig** = 10 {\$/pig}

Figure 53 below summarizes graphically the connections generated from the instructions above.

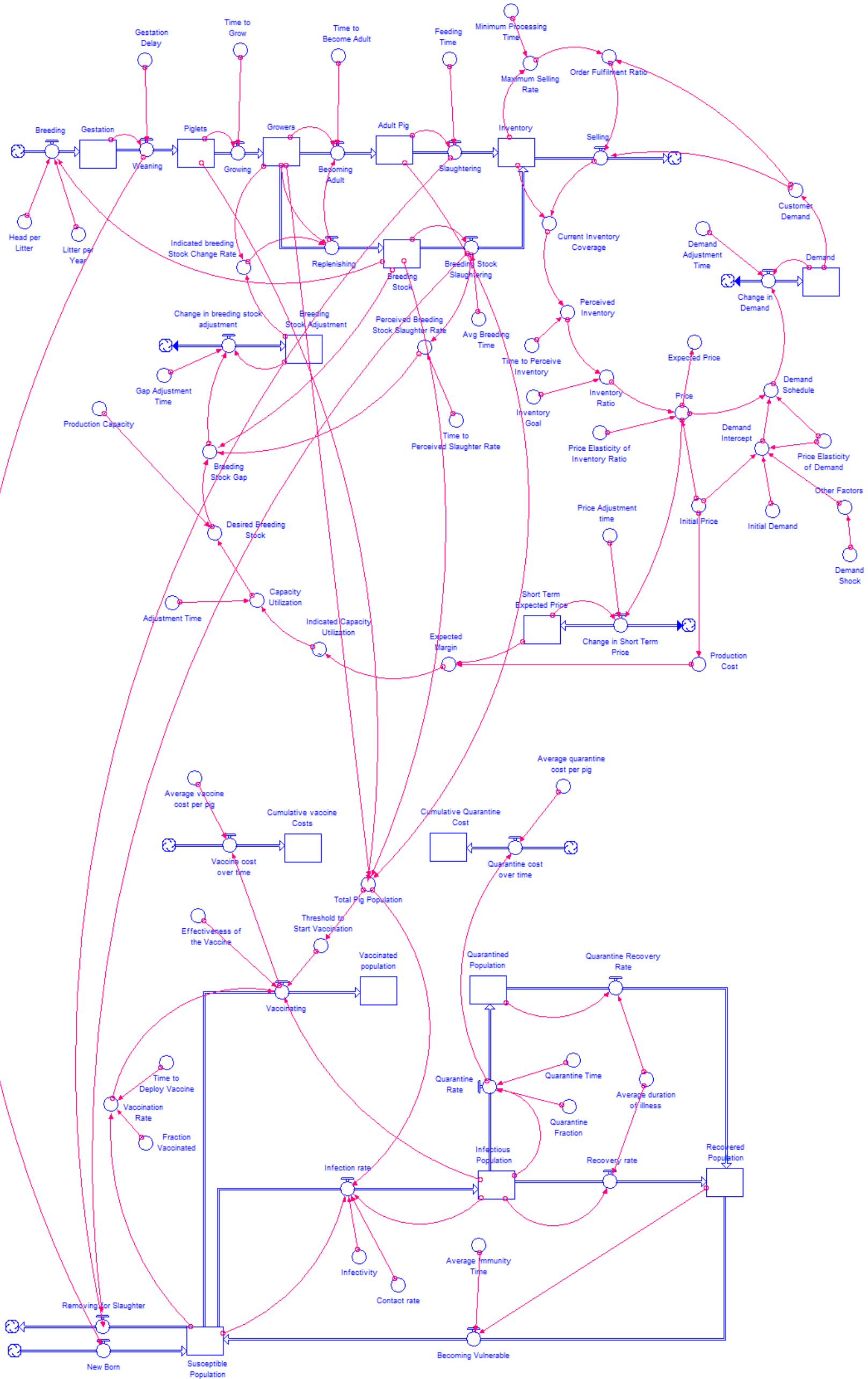


Figure 53: Implementing policy costs in the integrated SIR-value chain model

**Exercise 17**

- 1- Under current assumptions in SIR model (i.e. initial value of “infectious population” is equal to zero), how would stocks of “susceptible population”, “infectious population”, “recovered population”, “Vaccinated population” and “quarantined population” behave? Why? (Note: please be aware that the value chain model is running in equilibrium (no shocks tested))

-----

- 2- How would stocks of “susceptible population”, “infectious population”, and “recovered population” behave if the initial value of “infectious population equals to 1? Why?<sup>28</sup> Please draw possible behavior of each stock on three spaces to draw figures provided below (please pay attention to pattern of behavior (i.e. exponential growth or exponential decay or linear growth or a combination of these patterns of behavior rather than quantitative precision).

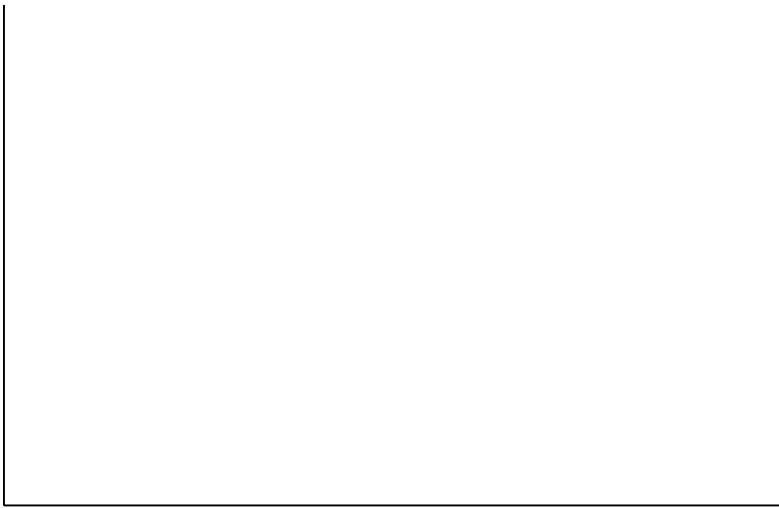
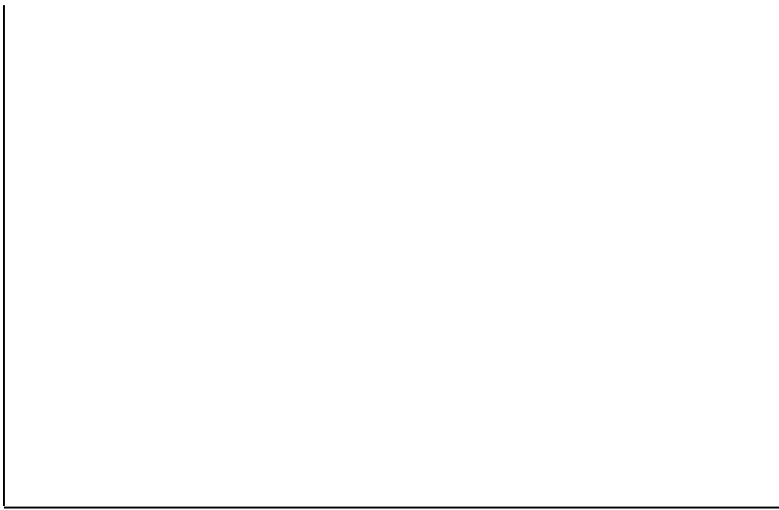


---

<sup>28</sup> Note:

- 1- be aware that the value chain model is running in equilibrium (no shocks test here)
- 2- For now, we assume policy interventions are not activated (we have no policy (vaccine and quarantine) in place).

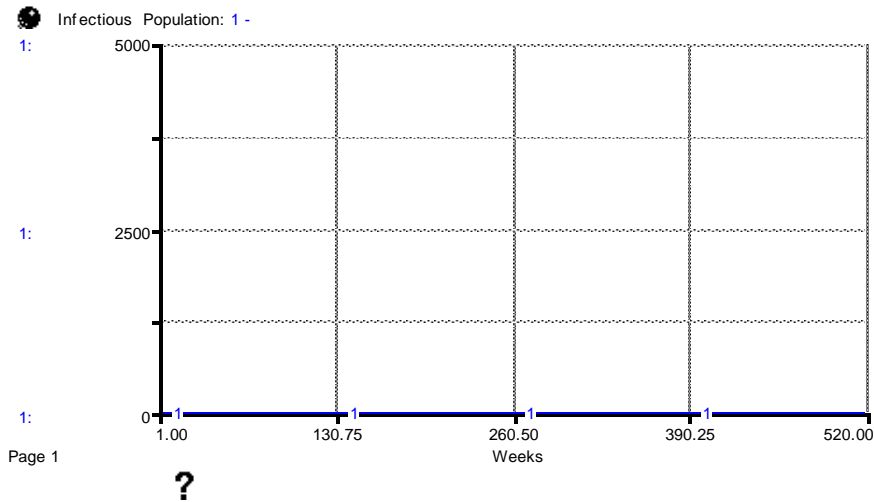
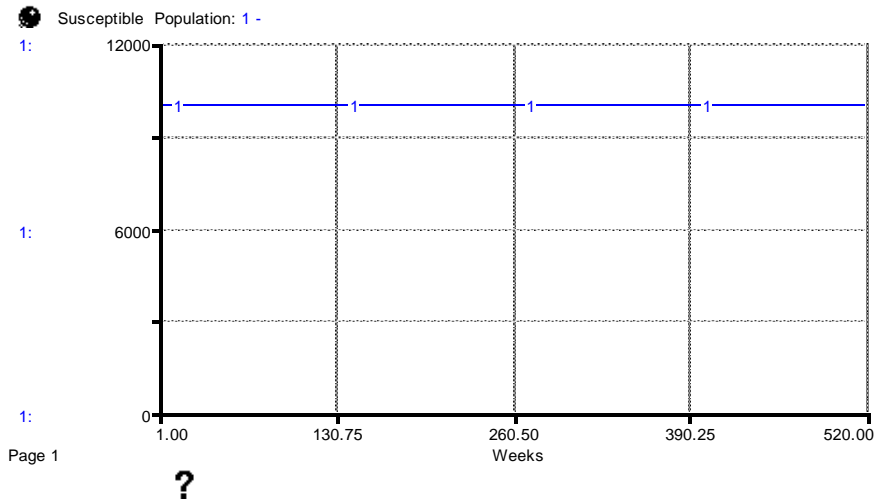




### Solution 17<sup>29</sup>

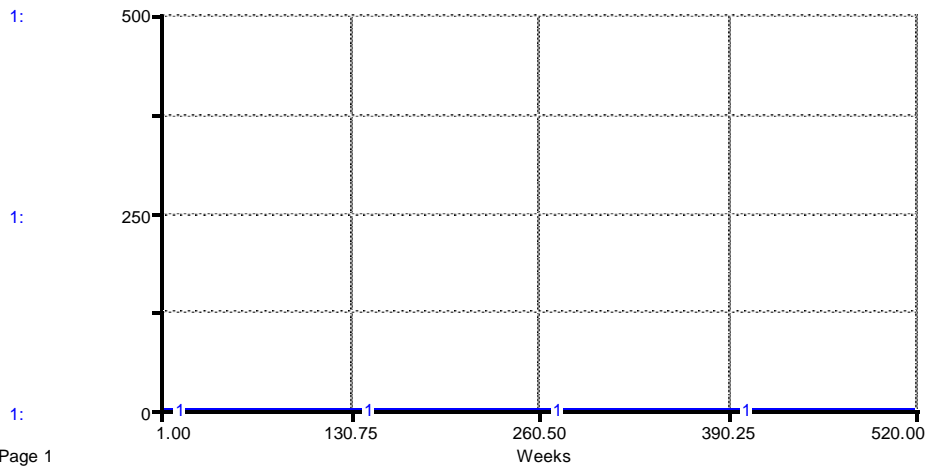
- 1- The stocks “susceptible population”, “infectious population”, “recovered population”, “vaccinated population” and “quarantined population” in the SIR model will remain unchanged because when “infectious population” is equal to zero it means there are no infected pigs to start disease outbreaks. Additionally, all stocks of the SIR model except “susceptible population” are equal to zero over a 520 week simulation time horizon because the farm or the region is disease free (there need to be at least one infected pig to start a disease outbreak). In this case, “Susceptible population” is equal to “total pig population”. Figure 54 shows the behavior of targeted stocks in the SIR model when “infectious population” is equal to zero.

**Figure 54:** shows behavior of stocks of “susceptible population”, “infectious population”, “recovered population”, “vaccinated population” and “quarantined population” in SIR model, respectively



<sup>29</sup> In **solutions 17 and 18 (next exercise)**, **run 1** represents no disease case; **run 2** represents disease outbreak without policy interventions; **run 3** represents disease outbreak with vaccination policy; **run 4** represents disease outbreak with quarantine policy in place.

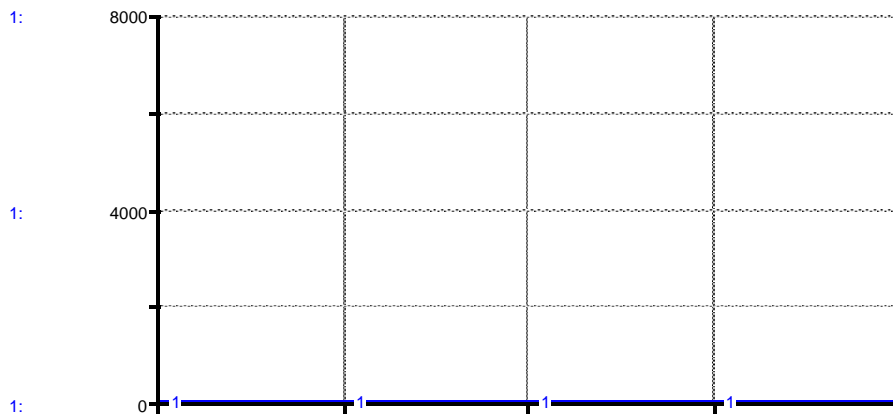
Recovered Population: 1 -



Page 1

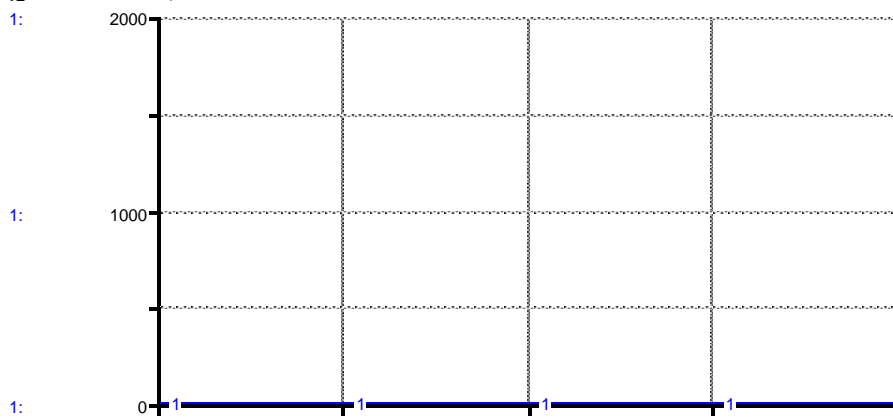
?

Vaccinated population: 1 -



?

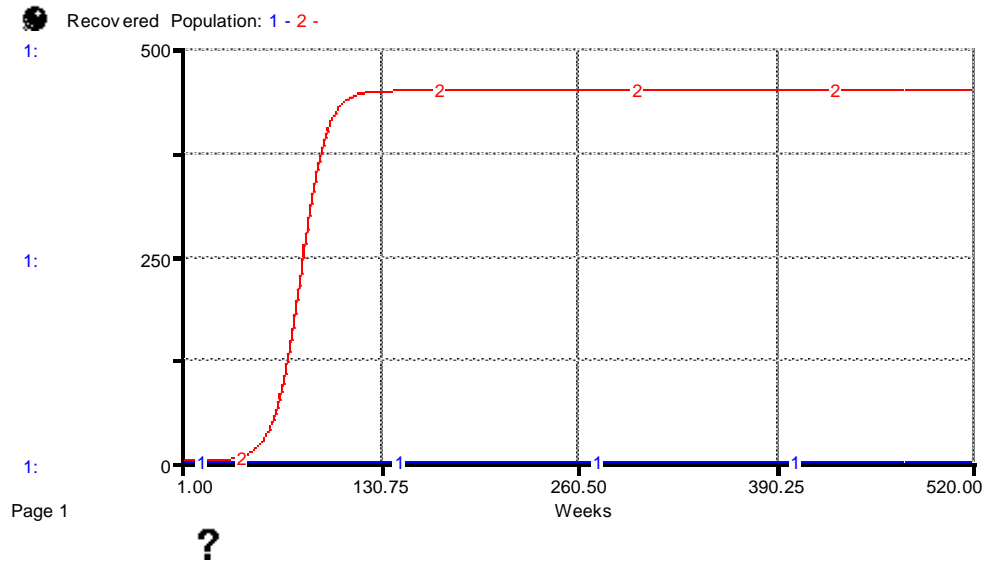
Quarantined Population: 1 -



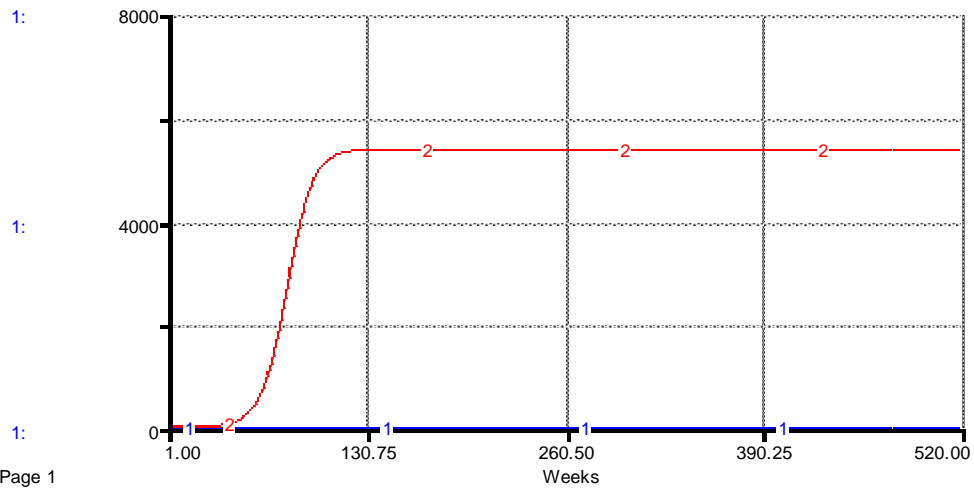
?

- 2- When the initial value of “infectious population” is equal to 1, this will lead to a disease outbreak. Initially, there is only one infected pig. This infected pig will infect other pigs in the farm or region. At the beginning, “infection rate” is very slow because one pig cannot cause a large-scale outbreak. However, more pigs get infected over time which accelerates “infection rate”. This leads to an outbreak and an increasing decline in “susceptible population” because more pigs get infected which in turn increases the level of “infectious population” in an increasing rate. However, as “infectious population” increases and “susceptible population” declines, the flow of “infection rate” begins to decrease in an accelerating rate because there are less vulnerable pigs (i.e. lower “susceptible population”). As “infectious population” grows, so does the flow of “recovery rate” and stock of “recovered population” which in turn increases flow of “becoming vulnerable” and feeds back to the stock of “susceptible population”. Then the infection cycle repeats itself. Figure 55 shows model behavior of “susceptible population”, “infectious population”, and “recovered population”, respectively.

**Figure 55: behavior of stocks of “susceptible population”, “infectious population”, and “recovered population”, respectively**

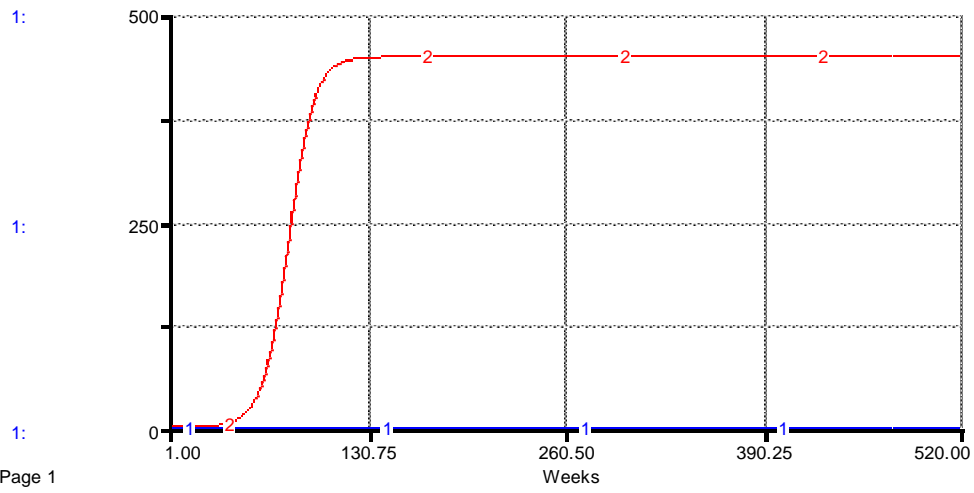


Infectious Population: 1 - 2 -



?

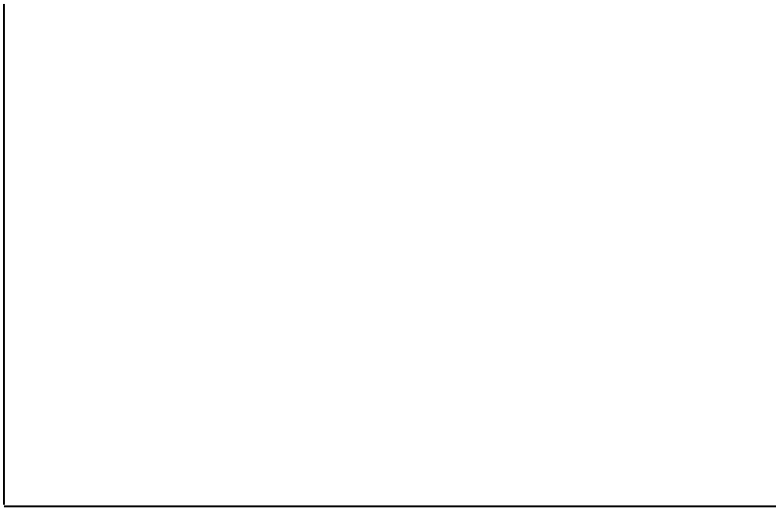
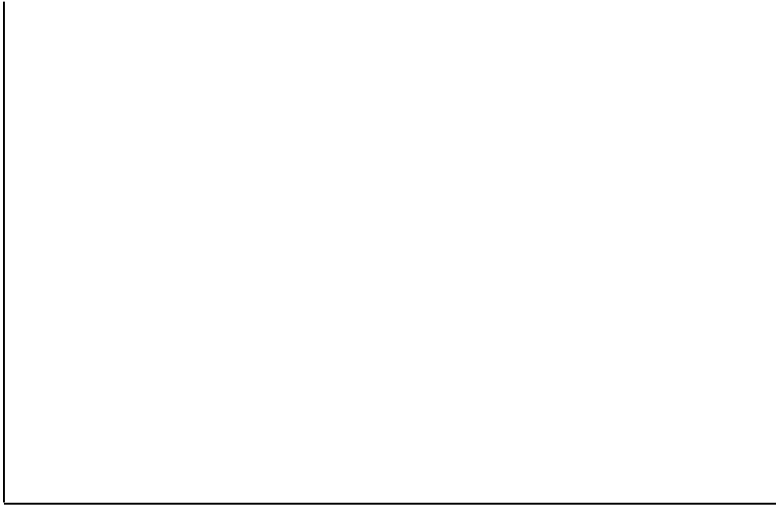
Recovered Population: 1 - 2 -



?

**Exercise 18**

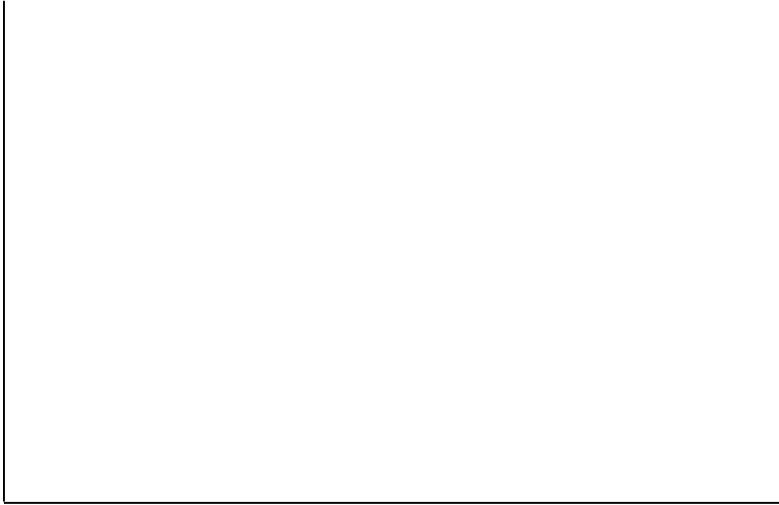
- 1- If a vaccination policy is applied, how would stocks of “susceptible population”, “infectious population”, “recovered population”, and “vaccinated population” behave if the initial value of “infectious population equals to 1? Why?<sup>30</sup> Please draw the possible behavior of each stock on four spaces to draw figures provided below (please pay attention to pattern of behavior i.e. exponential growth or exponential decay or linear growth or a combination of these patterns of behavior rather than being quantitatively precise).



---

<sup>30</sup> Note:

- 3- be aware that the value chain model is running in equilibrium (no shocks tested here)
- 4- For now, we assume quarantine policy intervention is not activated.



- 2- If a quarantine policy is applied, how would stocks of “susceptible population”, “infectious population”, “recovered population”, and “quarantined population” behave if the initial value of “infectious population equals to 1? Why?<sup>31</sup> Please draw the possible behavior of each stock on four spaces to draw figures provided below (please pay attention to pattern of behavior i.e. exponential growth or exponential decay or linear growth or a combination of these patterns of behavior rather than being quantitatively precise).

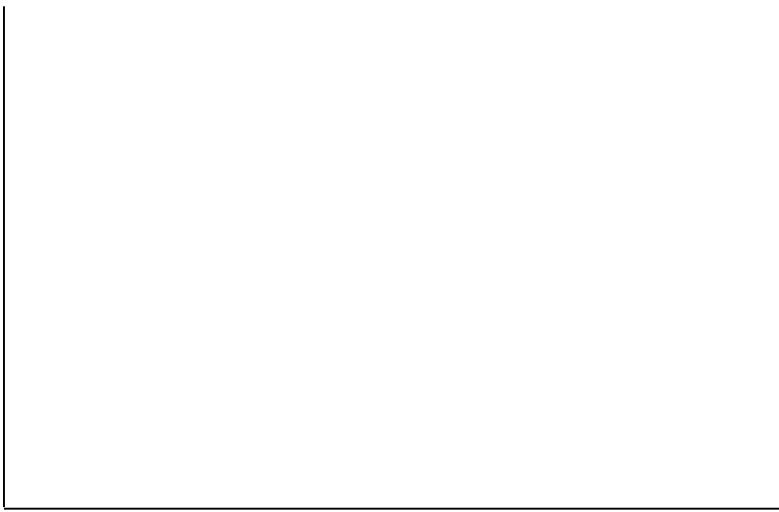


---

<sup>31</sup> Note:

- 5- be aware that the value chain model is running in equilibrium (no shocks tested here)
- 6- For now, we assume vaccination policy intervention is not activated.

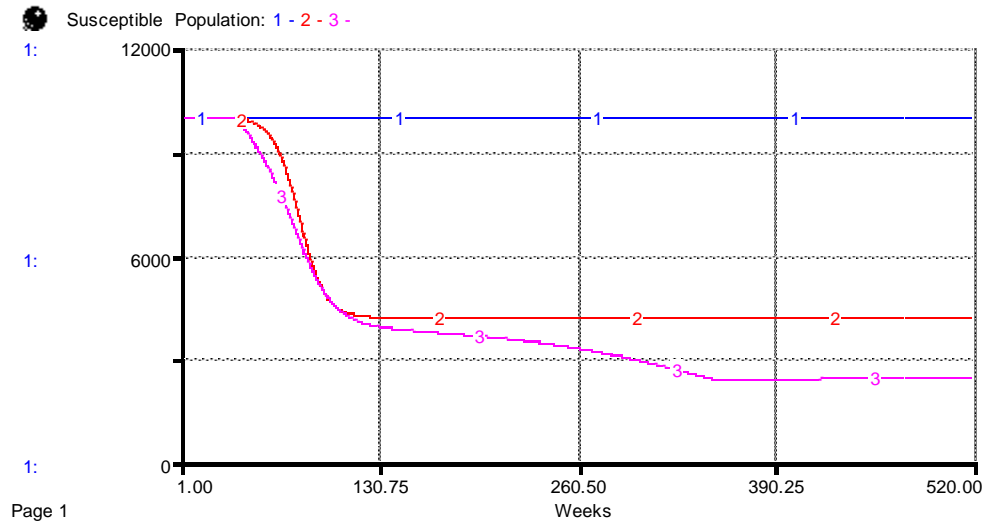




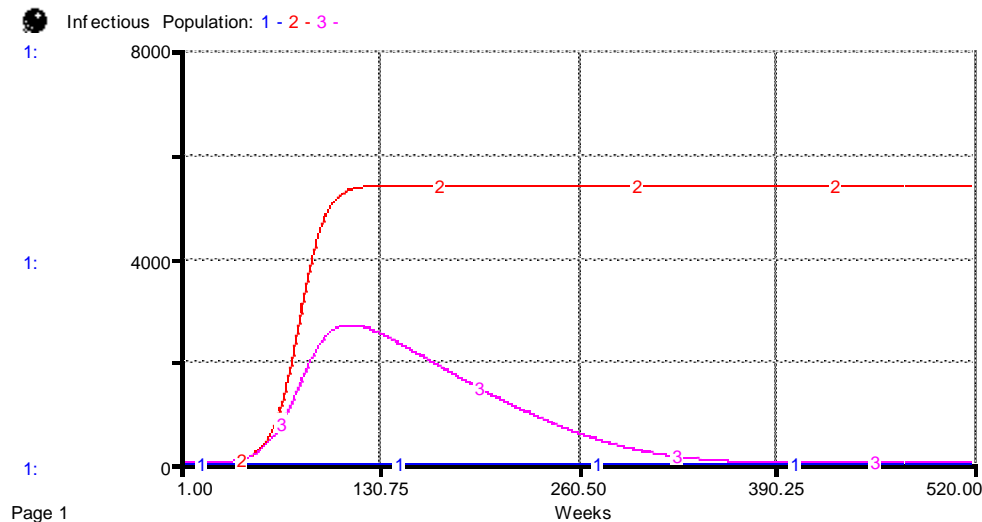
**Solution 18**

- When a vaccination policy is active, vulnerable pigs are continuously vaccinated and move to “vaccinated population” where pigs are immune from disease. This means that the “vaccinated population” continuously increases until it reaches the pig population limit. Since there are more outflows of pigs out of “susceptible population” compared to the previous exercise, “susceptible population” declines at a faster rate because a percentage of pigs become infected and another percentage gets vaccinated. Both the “infectious population” and “recovered population” peak at a much lower magnitude and then decline to reach very low levels compared to question 2 of exercise 17 because as infected pigs recover and move back to “susceptible population”, a fraction of those pigs get vaccinated until most pigs are immune from disease.

**Figure 56: behavior of stocks of “susceptible population”, “infectious population”, “recovered population”, and “vaccinated population, respectively**

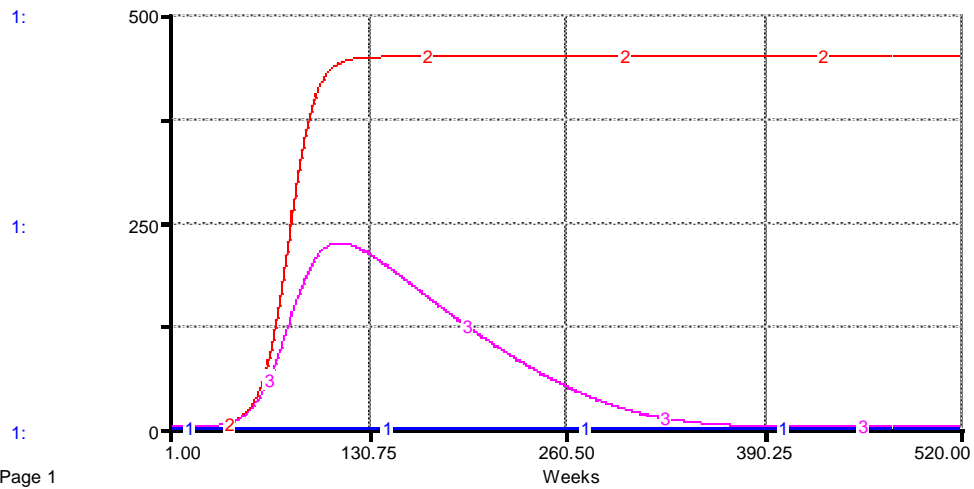


?



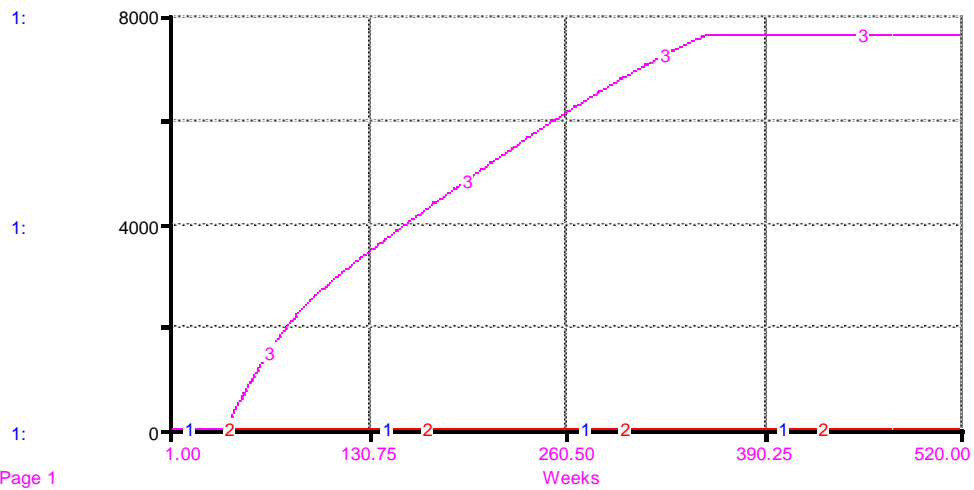
?

Recovered Population: 1 - 2 - 3 -



?

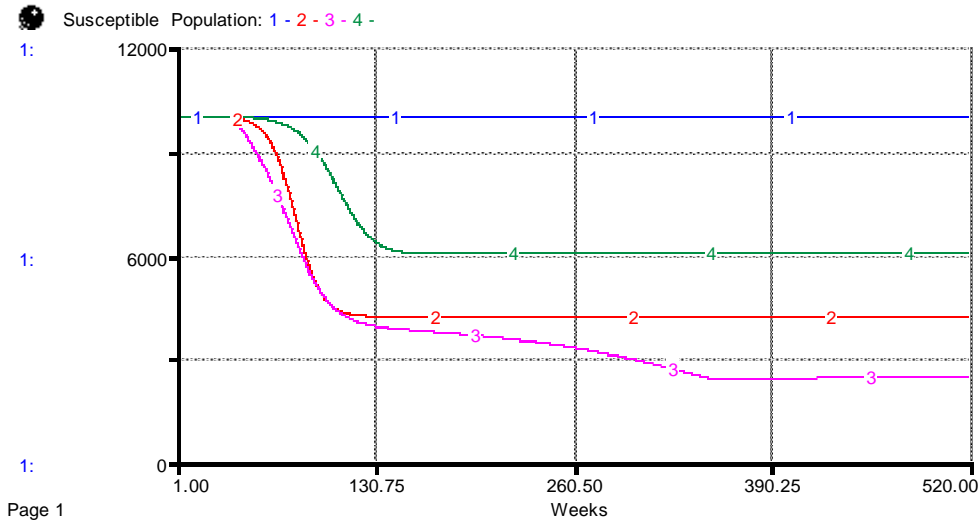
Vaccinated population: 1 - 2 - 3 -



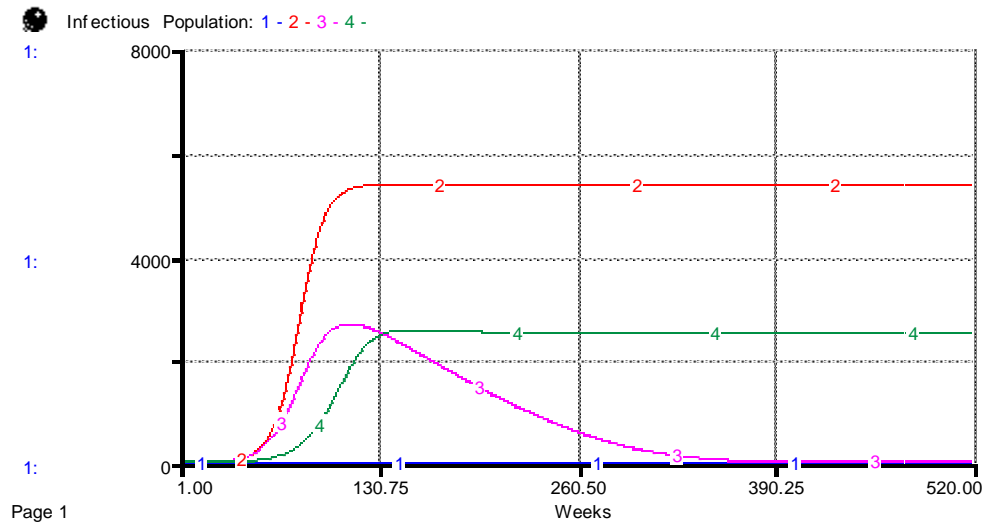
?

- 2- When a quarantine policy is active, the ‘susceptible population’ declines at a slower rate than results of question 2 of exercise 17 because infectious pigs are placed in quarantine to restrict their contact rate with healthy pigs. This greatly reduces the infection rate. However, infected pigs in “quarantine population” recover from disease and move to “recovered population” and then become vulnerable again. But since the infection rate is lower than the no quarantine case, both “infectious population” and “recovered population” peak at a much lower level compared to the no quarantine case.

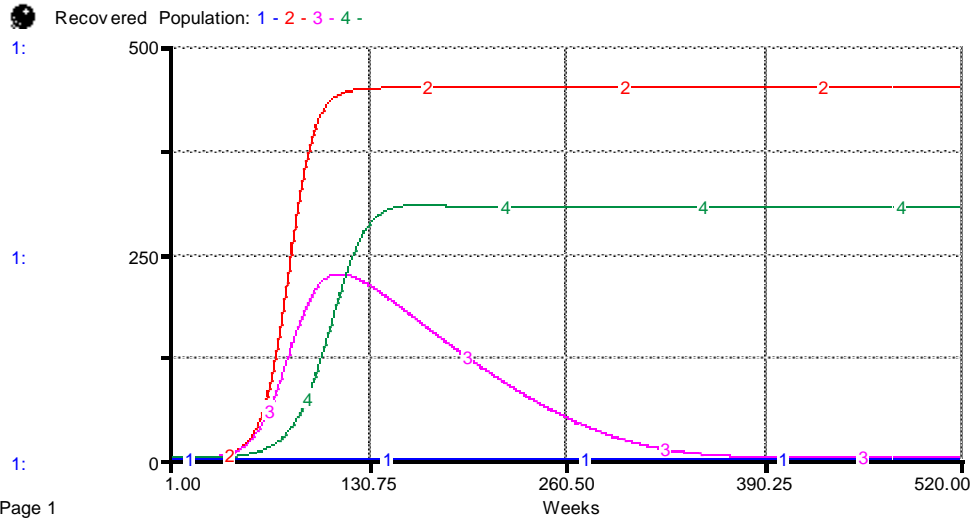
**Figure 57: behavior of stocks of “susceptible population”, “infectious population”, “recovered population”, and “quarantined population, respectively**



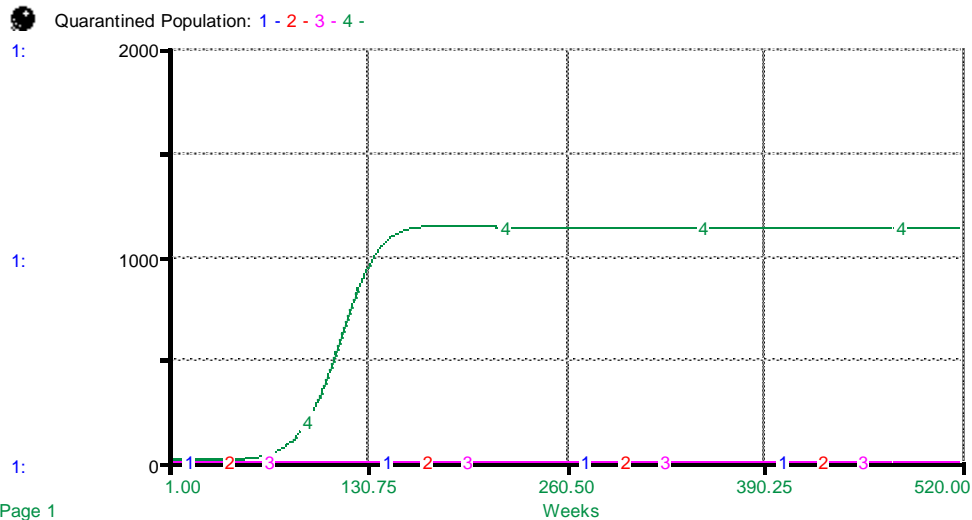
?



?



?



?

### Concluding remarks

The integrated value chain and SIR model of the previous section can be further expanded and integrated to provide a more realistic assessment of the joint epidemiological and economic impacts of different disease interventions. For example, we could include pig death rates in the value chain model in a manner that is consistent with death rate parameters in the SIR model. As disease management interventions directly affect pig fecundity and mortality, there will be an important feedback loop between disease and value chain dynamics and demonstrate how the evolution of disease can affect chain behavior, and how chain behavior can influence subsequent waves of infection over time (Rich 2007). Similarly, we could model vaccination in a manner that allows pigs to become susceptible again and require re-vaccination in order to stay immune from disease.

The models developed in this manual provide a first glimpse into the power of using SD in a value chain setting. As value chains are complex systems replete with a variety of interactions among different actors, biological parameters, environmental variables, and behavioral considerations, SD models are a natural means to model and conceptualize the structure of these chains and possible entry points for intervention. Future research will be needed to refine our thinking on how to model different agricultural and livestock systems that take into account the contextual differences associated with different chains, although it is hoped that this work provides a lens into how new models could be adapted from the lessons generated here.

## References

1. Hamza, K., Rich, K.M., Wheat, I.D. (2014). A system dynamics approach to sea lice control in Norway. *Aquaculture Economics and Management* 18(4), 344-368.
2. Hamza, K., Rich, K.M., Bahta, S., Katjuongua, Baker, D. (2014a). An analysis of beef market liberalization in Botswana: A quantitative value chain approach. Paper presented at the International Agribusiness and Food Management Association Annual Meetings, Cape Town, South Africa, 17-19 June 2014.
3. Hamza, K., Rich, K.M., Baker, A.D., Hendrickx, S. (2014b). Commercializing smallholder value chains for goats in Mozambique: A system dynamics approach. *Proceedings in Food System Dynamics 2014*, 117-134.
4. Harrison, K., Henley, D., Riley, H., Shaffer, J. (1974). *Improving Food Marketing Systems in Developing Countries: Experiences from Latin America*, Research Report No. 6. Latin American Studies Center, Michigan State University.
5. Kaplinsky, R., & Morris, M. (2000). *A Handbook for Value Chain Research*. Sussex, UK: International Development Research Center (IDRC).
6. Naziri, D., Rich, K.M., Bennett, B. (2015). Would a commodity-based trade approach improve market access for Africa? A case study of the potential of beef exports from communal areas of Namibia. *Development Policy Review* 33(2).
7. Rich, K.M. (2007). New methods for integrated models of animal disease control. Selected paper presented at the American Agricultural Economics Association Annual Meetings, Portland, OR. Accessible at [http://agecon.lib.umn.edu/cgi-bin/pdf\\_view.pl?paperid=26240&ftype=.pdf](http://agecon.lib.umn.edu/cgi-bin/pdf_view.pl?paperid=26240&ftype=.pdf).
8. Rich, K.M., Ross, R.B., Baker, D.A., Negassa, A. (2011). Quantifying value chain analysis in the context of livestock systems in developing countries. *Food Policy* 36(2), 214-222.
9. Rich, K.M., Perry, B.D., Kaitibie, S. (2009). Commodity-based trade and market access for developing country livestock products: the case of beef exports from Ethiopia, *International Food and Agribusiness Management Review* 12 (3), 1-22.
10. Slater, C.C. et al. (1969). *Market Processes in the Recife Area of Northeast Brazil*. Research Report #2. Latin American Studies, Center Michigan State University, East Lansing, MI.
11. Sterman, J.D. (2000). *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin-McGraw Hill, Boston.
12. Vennix, J. A. (1996). *Group model building: Facilitating team learning using system dynamics*. Chichester: Wiley.
13. Whelan, J. and Msefer, K. (1996). "Economic supply & demand," Paper prepared for the MIT System Dynamics in Education Project, Report No. D-4388.