Introduction to the Loop Impact Method
of Feedback Dominance Analysis

Session 2

Measuring Impact and Dominance in System Dynamics Models

Notes

John Hayward

School of Computing & Mathematics

University of South Wales

john.hayward@southwales.ac.uk

sociomechanics.com

38th International Conference of the System Dynamics Society 2020

Abstract

A central premise of system dynamics is that dynamical behaviour can be explained by model structure, especially its feedback loops. For example, the familiar S-shaped growth of the limits-to-growth archetype is explained using shifting loop dominance. Although such an explanation appears clear, what is less clear is how a loop is quantified, how dominance is defined, and what aspect of dynamical behaviour is being explained. These issues become more pronounced as the number of variables and feedback loops increases.

This workshop will introduce participants to the Loop Impact Method of feedback dominance analysis. In this first session participants will explore a number of models using the method's definitions of behaviour, structure and dominance, and compare them with their own understanding of the model's behaviour and structure. Participants will be encouraged to work in groups and share ideas. Models will be accessed using the isee exchange web platform.

In the second session participants will learn how to implement the Loop Impact method by adapting standard models in Stella Architect.

Loop Impact

Loop impact is a measure of the effect that a feedback loop has on a stock contained in that loop.

Feedback loops, are a critical structure within a system dynamics model, used to explain behaviour in an endogenous manner. Feedback occurs when state conditions initiate a sequence of cause and effect that eventually returns to change those state conditions.

The loop impact is the effect that one stock – the **source** – has on the next stock in the causal chain in a feedback loop – the **target**. It is measured by the acceleration induced in the target stock by the source stock, divided by the rate of change of the target stock. In an *n*th order loop there will be *n* loop impacts, whose product is the loop gain.

The concept can be extended to exogenous effects and connected stocks not in a loop.

There are 3 key stages in implementing the loop impact method in an SD simulator[1]:

A. Isolate the pathway on which impact is required
B. Measure the impact
C. Determine the dominance

The first two stages will be looked at together.

Some familiarity with Stella Professional/Architect is assumed.

Instructions

The notes, with the web links should be complete. Instructions to be carried out in model construction and analysis are given as bullets.

[1] An alternative method involves computing the loop impacts using differentiation on a differential equation version of the SD model, which has been amended to preserve all pathways. Such an amended differential equation is called a causally connected differential equation (Hayward & Roach, 2018). The method in this workshop is numerical and does not require formal calculus.

1. Uniform Change

In a system dynamics model the fundamental state variables of the model are the stocks. Stocks are accumulations of material that can only change over time though their flows. Inflows add material to the stock, outflows remove material.

Consider a single stock *Y* with a constant inflow

- Start up Stella
- Place the following in the model window, Figure 1:

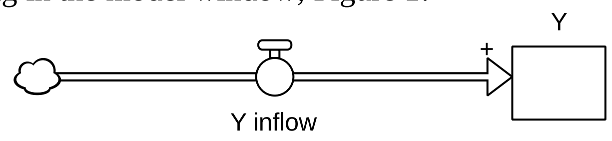


Figure 1: Uniform change – structure

- Right click the flow and set the polarity to plus (+).

The plus sign on an inflow means "add to". The flow adds material to *Y*.

- Set the initial value of *Y* to 20.
- Set *Y inflow* to 1.
- In Run Specs, set the method to Runge-Kutta-4, DT to 0.1, and the range from 0 to 20.
- Set up a graph for *Y* in the model window and run the model.

Your graph should look like Figure 2:

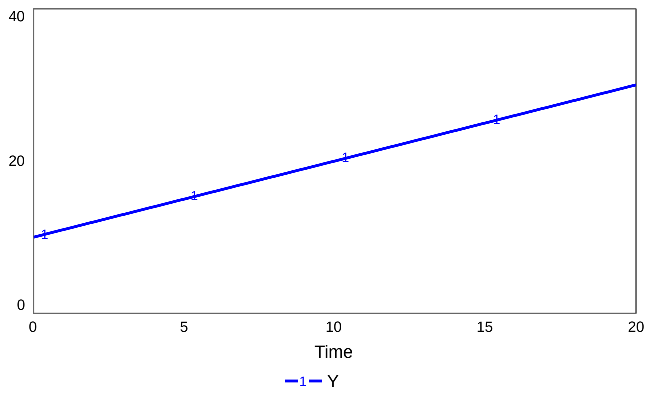


Figure 2: Uniform change - behaviour

The stock *Y* changes uniformly from its initial value at 10 to its final value at 30 though the addition of 1 per unit time. Uniform change shows itself as a straight line on a time graph, an absence of curvature. In this sense there is no force on the stock *Y*, i.e. there is no other stock, feedback loop or exogenous signal that has an impact on *Y* to change its uniform behaviour.

By contrast if the change in a stock is not uniform, showing itself as a curve, then there must be another stock, feedback loop or exogenous signal impacting the stock. The next system will show a stock impacted by another.

2. Stock Impact

Understanding Impact

Stock impact measures the effect one stock has on another. It is a generalisation of the loop impact of Hayward & Boswell (2014) as it does not require the stocks to be in a feedback loop.

The previous system is extended by allowing *Y* to impact another stock *X*.

- Add the following in the model layer of the previous one stock model, Figure 3:

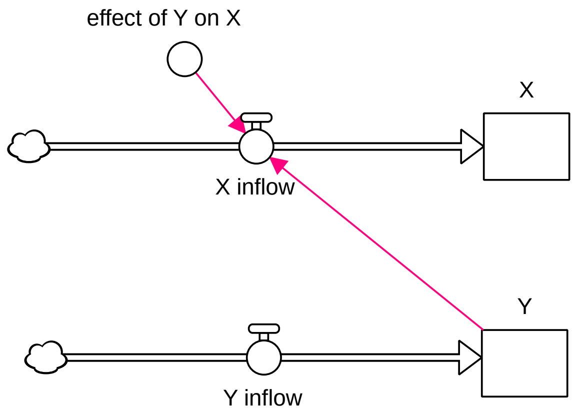


Figure 3: Stock Y influencing Stock X

- Set the values and formulae for the new elements using the following equations:

INIT X	=	*0*
X_inflow	=	*Y * effect_of_Y_on_X*
effect_of_Y_on_X	=	*0.2*
INIT Y	=	*10*
Y inflow	=	*1*

The last two equations should already be set.

Note that a stock can only be changed though its flow. Thus, for stock *Y* to directly influence stock *X*, then *Y* must be connected to the flow of *X*. **Thus, stock *Y* impacts stock *X***.

The connection from *Y* to *X* constitutes a **causal pathway** from *Y* to *X*. This is a one-way link representing the order of cause and effect in the model. In this case *Y* effects *X*, or more in the terminology used here, *Y* impacts *X*. *Y* is the **source stock**, as it is the origin of the cause; *X* is the **target stock**, the recipient of the effect of the cause.

The converter *effect of Y on X* controls how much *Y* impacts *X*. It is a unit conversion, allowing the two stocks to be compared even though they may be measured in different units. It is also a value conversion allowing the effect of *Y* on *X* to be adjusted.

- Set up a graph for *X* and run the model, Figure 4.

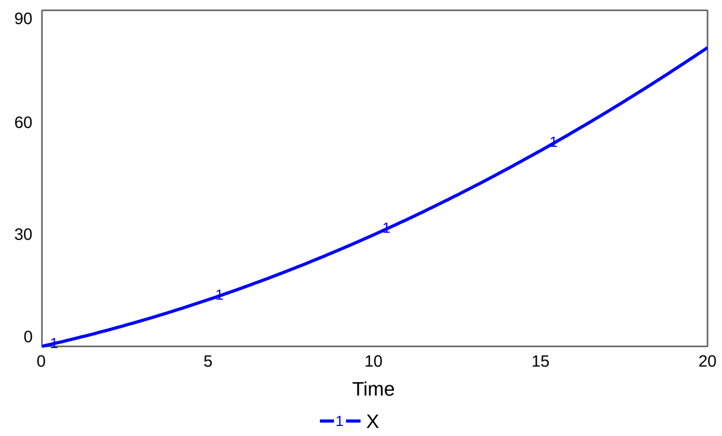


Figure 4: X is accelerating due to the effect of Y's uniform change

If *Y* had been constant, then *X* would have changed uniformly. However, in this case as *Y* is changing (Figure 2), then the change in *X* will not be uniform, as seen in Figure 4. The level of the effect of *Y* on *X* determines the extent to which the curvature of *X* deviates from a straight-line uniform behaviour.

When changes in stock *Y* induce a change in stock *X* then it shows itself in a deviation from uniform change in stock *X*, like a force shows itself as deviation from uniform motion. This deviation is the impact of stock *Y* on stock *X*.

Measuring Impact

In the model of Figure 3, "Stock Impact" is the measure of the impact of *Y* on *X* . It is defined by the amount of acceleration in *X* due to the effect of *Y*, measured relative to the change in *X*, (Hayward & Boswell 2014, Hayward & Roach 2018).

$$Impact = \frac{Acceleration}{Rate\ of\ change}$$

This can be computed as follows:

1. Determine the rate of change of the influence of *Y* on *X*. This is how much *Y* accelerates *X*.
2. Determine the net flow of *X*. This is the rate of change of *X*, (dX/dt).
3. Divide the result of 1. By the result of 2.

To carry out step 1, the effect of the pathway needs to be identified (key stage A, page 3). This can be done by adding an additional converter to the model which will be used to measure the influence of *Y* on *X*.

- Add the converter *Y on X identifier* in the link from *Y* to the flow of *X*, and move the converter *effect of Y on X* to be attached to the *Y on X identifier*. See Figure 5:

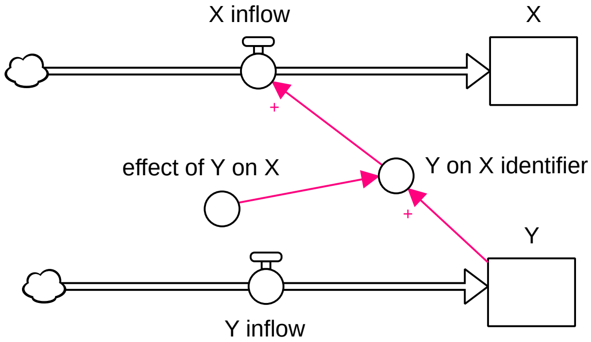


Figure 5: Pathway identifier added to model of Figure 3

- Set the formula for the inflow of *X:*
  X_inflow = Y_on_X_identifier

- Transfer the formula that had been in the *X* inflow to *Y on X identifier:*
  *Y_on_X_identifier = effect_of_Y_on_X*Y*

At the point the model should run and still give the same result. All that has happened is the final effect of the pathway from *Y* has been captured before entering the inflow of *X*. This procedure will be essential when there are multiple pathways to a flow.

The pathway identifier *Y_on_X_identifier* can now be used to compute the impact of *Y* on *X* (key stage B, page 3).

- Ghost the flow: *X inflow.*

- Determine the rate of change of *X* in the converter *X rate of change* from the ghost of *X inflow*, Figure 6.

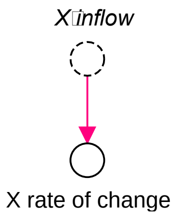


Figure 6: Computation of rate of change

using the formula: *X_ rate_of_change = X_inflow*

In models where a stock has multiple flows, defining a rate of change converter is essential.

- Ghost the converter *Y on X identifier*.

Determine the stock impact *Y on X impact* from the ghost of *Y on X identifier* and *X rate of change,*
- Figure 7:

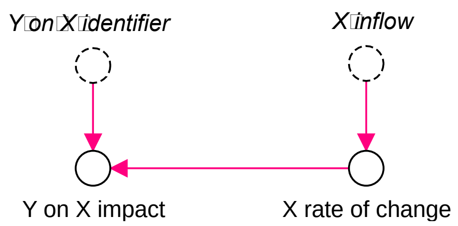


Figure 7: Computation of impact

using the formula:

Y_on_X_impact = DERIVN(Y_on_X_identifier, 1)/ X_ rate_of_change

The Stella built-in function DERIVN is the derivative with respect to time. The second parameter, 1, means it is the first derivative. The formula divides this time derivative by the time derivative of *X*. This formula measures how the pathway into *X*'s flow change compared with changes in *X* itself[2].

- Plot *Y on X impact* on a graph along with *X*, and place it on the right-hand axis (checkbox in graph dialogue box), Figure 8:

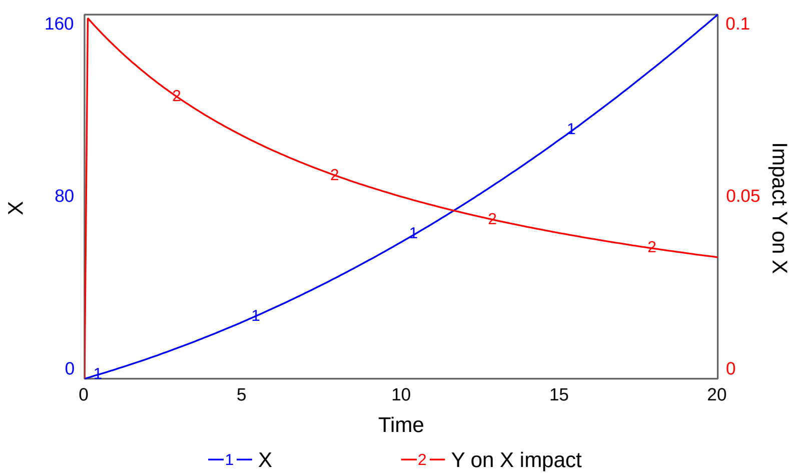


Figure 8: Stock X with impact of Y on X

Note that the stock impact of *Y* on *X* declines as *X* increases. This is because *Y* is changing uniformly, which has less impact on *X* as *X* gets larger.

Note the first few low impact values after the start of the simulation are spurious. This effect is due to the limitations of the numerical approximation and can be reduced by using a smaller value for DT[3].

The units of impact are (*1/time*), that is "per unit time". Impact measure acceleration and curvature independently of the units of the stock, enabling impacts from stocks with different units to be directly compared. Although beyond the scope of this workshop, impact is also a ratio measure of the force of one stock on another (Hayward & Roach, 2018).

[2] Mathematical note. Impact measures how changes in *y* accelerate *x*. Let $\dot{x} = f(y)$. Differentiating gives $\ddot{x} = \frac{d(f(y))}{dt} = \frac{\partial f}{\partial y}\dot{y}$. Impact compares this measure with changes in *x*. Thus $\ddot{x}/\dot{x} = \frac{\partial f}{\partial y}\frac{\dot{y}}{\dot{x}} = df/dt \div dx/dt$, the formula used.

[3] This definition of impact assumes time is modelled continuously, hence the RK method and DT much less than 1. Impact and feedback in discrete time systems is beyond the scope of this workshop.

3. Loop Impact in a First-Order Feedback Loop

First-order Feedback & Behaviour

When the influence of a stock on a stock is part of a feedback loop it makes sense to call it loop impact rather than stock impact. For a first-order feedback loop a stock influences itself. A first-order feedback loop is one with only one stock. The following model investigates the loop impact of such a system.

- Place the following in the model window, including the link polarities:

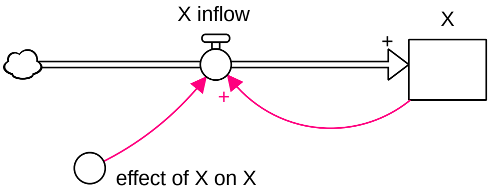


Figure 9: First-order reinforcing loop

The + on the flow means "add to". However, the + on the inflow means "same way". Thus, if *X* increases then its flow increases, but if *X* decreases then its flow decreases. Of course, unless other structure is added, *X* cannot decrease in this model!

There is now one causal pathway from *X* to itself, represented by the connecter from *X* to its inflow.

- Set the initial value of *X* to 1.
- Set *X inflow* to: *X_inflow = effect_of_X_on_X * X*
- Set the converter *effect of X on X* to 0.1.
- In Run Specs, set the method to Runge-Kutta-4, DT to 0.1, and the range from 0 to 20.
- Set up a graph for *X*, axis 0 to 10, and run the model.

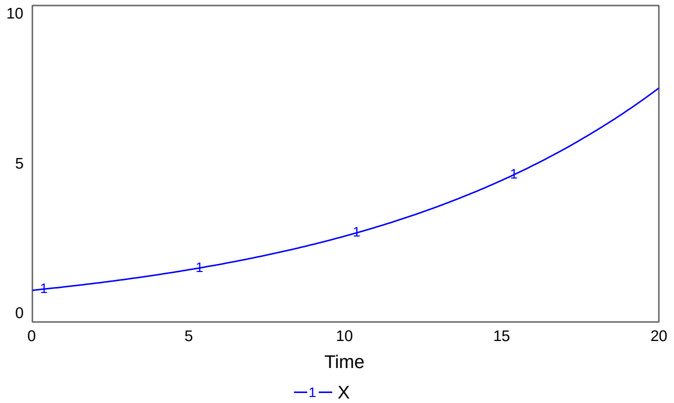


Figure 10: Exponential behaviour of first-order reinforcing loop Figure 9

The accelerating behaviour of *X*, Figure 10, is determined by the reinforcing feedback loop between itself and its flow, Figure 9, and the parameter *effect of X on X*. As *X* increases, its flow increases, thus *X* increases **faster.** Thus, *X* is causing itself it to accelerate. The curve is exponential, and the exponent is determined by the **loop gain**, which is *effect of X on X = 0.1* in this case.

The impact of *X* on itself is called **loop impact** which is a special case of stock impact when there is feedback.

Computing Loop Impact

The method is the same as the stock impact computation already introduced.

- Add the converter *X on X identifier* in the link from *X* to the flow of *X*, and move the converter *effect of X on X* to be attached to the *X on X identifier*. See Figure 11*:*

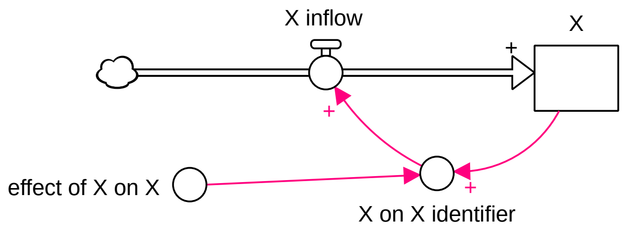


Figure 11: Loop identifier added to first-order reinforcing loop model of Figure 9

- Set the formula for the inflow of *X:*
  X_inflow = X_on_X_identifier

- Transfer the formula that had been in *X inflow* to *X on X identifier:*
  *X_on_X_identifier = effect_of_X_on_X * X*

- Ghost the inflow of *X* and the converter *X on X identifier*.

- Determine the rate of change of *X* in the converter *X rate of change* from the ghost of *X inflow*, Figure 12.

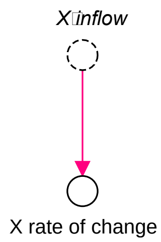


Figure 12: Computation of rate of change

using the formula: *X_rate_of_change = X_inflow*

- Determine the stock impact *Y on X impact* from the ghost of *Y on X identifier* and *X rate of change*, Figure 13:

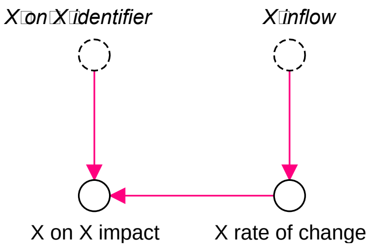


Figure 13: Computation of impact

using the formula: *X_on_X_impact = DERIVN(X_on_X_identifier, 1)/ X_rate_of_change*

- *Plot X on X impact* on a graph along with *X*, ensuring that the scale of the impact (right hand axis) is between 0 and 0.2,

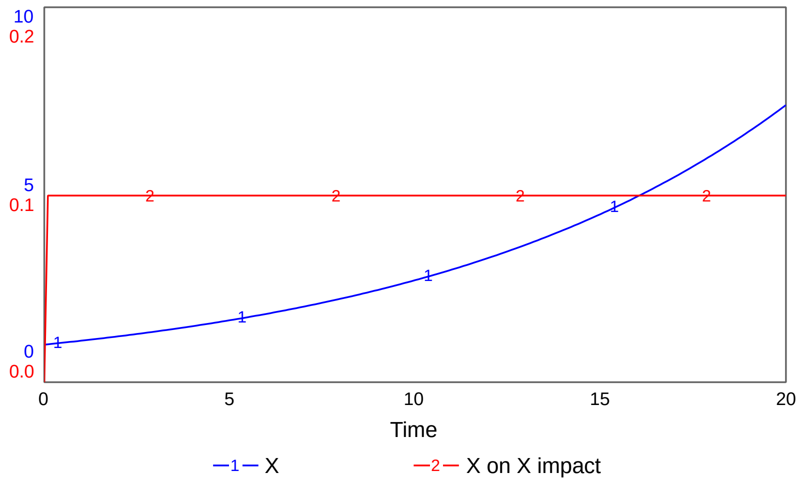


Figure 14: Behaviour of first order reinforcing loop Figure 9 with impact

The loop impact *X on X impact* is constant with a value of 0.1 (ignoring the initial numerical inaccuracy), Figure 14. This is the same as the loop gain, given by the parameter *effect of X on X*. **The loop impact in a first order loop is always equal to the loop gain whatever the formulae in the loop (Hayward & Boswell 2014).**

The reason for the constancy of loop impact is that the rate of change of stock *X* is proportional to *X*. Thus, the acceleration is proportional to the rate of change. Thus, the ratio of acceleration to rate of change is constant.

4 Loop Dominance in a First-Order Model

Introduction

We now want to apply what we have learnt about loop impact and its computation to a model with more than one loop. Although all loops will affect the behaviour of the stock to some degree, there may be periods where different loops have the controlling influence over behaviour. This is the dominance issue.

The model we will use is a limits-to-growth one where a population grows in a restricted environment, and is also subject deaths, Figure 15. There are three loops, all first-order: one reinforcing loop on the inflow, and two balancing loops, one on the inflow and one on the outflow.

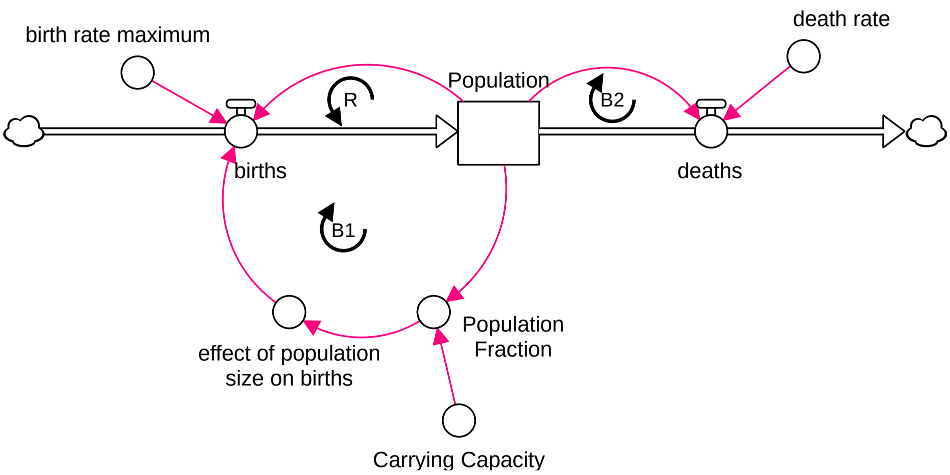


Figure 15: Limits-to-growth model of population

The behaviour, is the familiar S-shaped growth (if the birth rate is high enough), Figure 16:

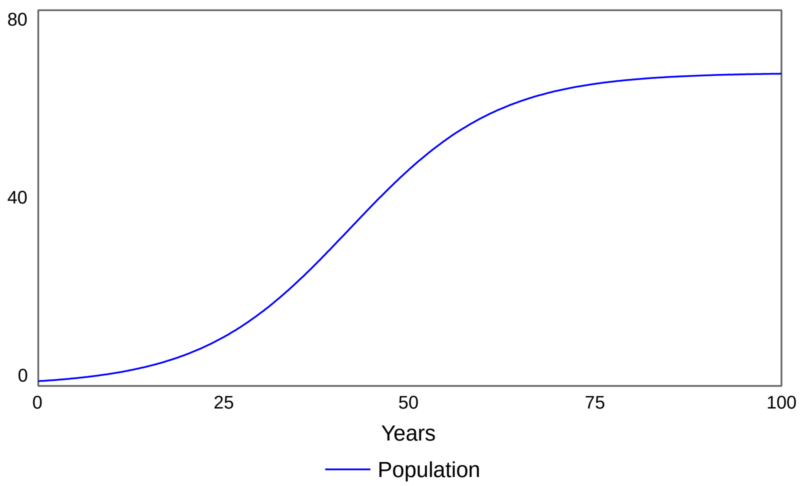


Figure 16: S-shaped behaviour of limits-to-growth model Figure 15

The question is: which loop is dominant in which period, i.e. which loop determines the curvature at any point in time? The method is to determine the three loop impacts, then apply an algorithm – **the loop picker algorithm** – to identify the loop, or loops, that are dominant. The use of the algorithm and the definition of dominance will be explained in the following sections.

The outline of the loop impact method is:

Step 1 – Insert Loop Identifiers
Step 2 – Compute Loop Impacts
Step 3 – Identify Possible Loop Combinations
Step 4 – Copy Loop Picker Algorithm
Step 5 – Connect Loop Picker Algorithm

a) Set Array Name and Element Names
b) Connect Loop Impacts
c) Edit Loop Combinations
d) Edit Combination Code

Refer back to this outline if needed.

Obtain the Model

To proceed, we need the model.

- Either download the model from: https://sociomechanics.com/loop-impact/intro/

  or construct the model in a Stella model window from Figure 15 using the equations:

  *Population(t) = Population(t - dt) + (births - deaths) * dt*
  INIT Population = 1
  INFLOWS:
  *births = birth_rate_maximum*effect_of_population_size_on_births*Population*
  OUTFLOWS:
  *deaths = death_rate*Population*
  birth_rate_maximum = 0.15
  Carrying_Capacity = 100
  death_rate = 0.05
  effect_of_population_size_on_births = 1-Population_Fraction
  Population_Fraction = Population/Carrying_Capacity

- If you have constructed the model ensure Run Specs has RK4 with DT of 0.02, start time 0, end time 100. Ensure there is also a graph as in Figure 16.

Step 1 – Insert Loop Identifiers

There are three loops, thus there are three loop impacts, thus three loop identifiers. Following the method described earlier, loop identifiers (converters) need to be placed in each loop, immediately before they enter the flows.

- Modify the model in Figure 15 with the three loop identifiers: *R Identifier*, *B1 Identifier* and *B2 Identifier*, Figure 17, moving the connections of the two parameters *birth rate maximum* and *death rate*:

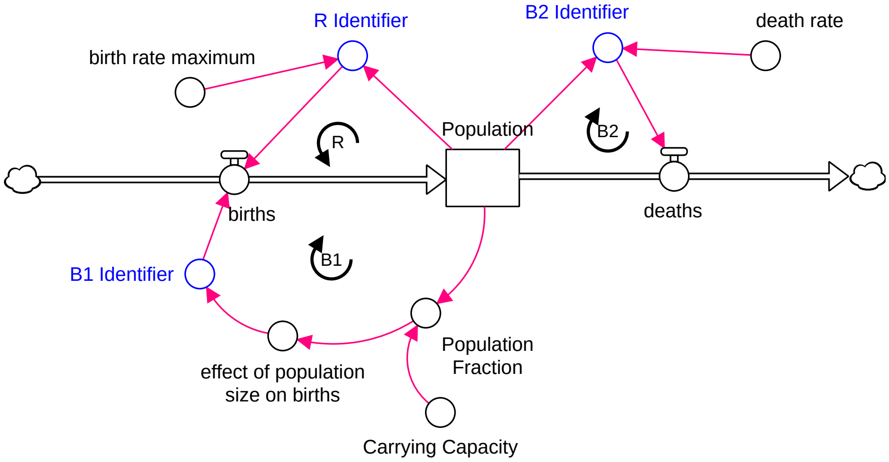


Figure 17: Limits-to-growth model with loop identifiers.

- The modified formulae should be[4]:

INFLOWS:		
births	=	*R_Identifier*B1_Identifier*
OUTFLOWS:		
deaths	=	*B2_Identifier*
B1_Identifier	=	*effect_of_population_size_on_births*
B2_Identifier	=	*Population*death_rate*
R_Identifier	=	*birth_rate_maximum*Population*

Step 2 – Compute Loop Impacts

Rate of Change

- <u>Ghost</u> the flows *births* and *deaths*, and <u>connect</u> to a new converter *population rate of change*, Figure 18, where the formula is: *population rate of change = births - deaths*

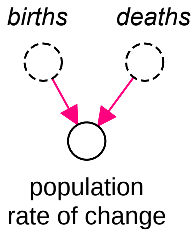


Figure 18: Computation of population rate of change

[4] The method shown here (Hayward & Boswell, 2014) modifies the model diagram (not structure or equations) with loop identifiers. If you do not want to modify your SD diagram, then the pathways can be ghosted and the loop identifiers inserted in the ghosted pathways. This alternative approach is beyond the scope of this workshop.

Loop Impacts

The loop *B2* is on its own, thus its loop impact follows as in previous examples:

- Ghost *population rate of change* and *B2 Identifier* and connect to a new converter *B2 Impact*, Figure 19, where the formula is:

  B2 Impact = - DERIVN(B2_Identifier, 1)/ population rate of change

  the time derivative of the identifier divided by the total rate of change of the stock. The minus sign is because the loop is on an outflow.

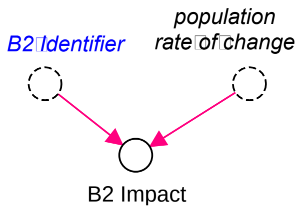


Figure 19: Computation of impact of loop B2

Because loops *R* and *B1* are multiplied together in the inflow of *Population*, then their impacts are computed using both loop identifiers following a simple rule: The derivative of the loop identifier of the **active** branch is multiplied by the loop identifier of the **passive** branch, where the **active** branch is the one whose impact is being found[5].

- Using ghosting, connect the converters to new converters for the impacts, as in Figure 20:

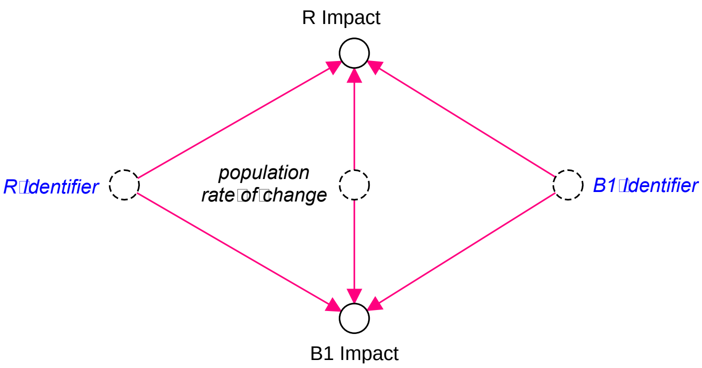


Figure 20: Computation of impact of loops R & B1.

- Set the formula for the impact of *R* by differentiating The *R identifier*, multiplying by the passive branch, and dividing by the rate of change of the population:

  *R_Impact = B1_Identifier*DERIVN(R_Identifier, 1)/ population rate of change*

- Likewise set the formula for *B1 Impact*, where *B1 identifier* is active and *R identifier* is passive:

  *B1_Impact = R_Identifier*DERIVN(B1_Identifier, 1)/ population rate of change*

As the loops *R* and *B1* are on the inflow the formula for the impacts are positive.

NB Inflow impacts have positive formulae, outflow impacts have negative formula

[5] This rule is the product rule from differential calculus.

Step 3 – Identify Possible Loop Combinations

There are three loops: *R*, *B1* and *B2*. Because this is a first order model the loop impacts will not change polarity because impact is the same as the loop gain which does not normally change polarity[6]. Thus, there is only one loop with positive polarity on *Population*, i.e. *R*, and two with negative polarity on *Population*, i.e. *B1* and *B2*. Thus, these two are the only loops that can combine. Name it B1B2.

Thus, the four possible conditions of dominance are: R, B1, B2 and B1B2. This information will be used to connect the loop impacts to the algorithm to determine dominance.

Step 4 – Copy Loop Picker Algorithm

Download and Copy

The loop picker algorithm determines the dominance from the loop impacts. Its inner workings need not concern us here. The main issue will be its use, as it will nearly always require some modification to connect to your SD model.

- Ensure your Population model Stella file is open.
- Download *Loop Picker.stmx* from https://sociomechanics.com/loop-impact/intro/
- Open the Stella file *Loop Picker.stmx*
- Select and copy all the converters in the model, e.g. cmd A, cmd C. (or ctrl A, ctrl C)
- Close the Stella WINDOW by clicking the top left icon on the window for *Loop Picker.stmx* i.e. do not quit Stella.

This is to ensure the Stella code remains intact in the memory buffer.

- Paste the algorithm in the model, e.g. cmd V (or ctrl V). Save the file.

Identify Parts of Algorithm

The algorithm is shown in Figure 21. *single loops* is the input. *loop picker* is the output, and contains an index that indicates the dominant loops at any one time. *change of dominance* can be used to indicate the time that dominance changes, though the loops are not identified.

Three converters need modifying: *single loops*, *loop combinations*, *combination code*, all given in large font. Other converters can be ignored.

Dominance

A loop, or set of loops, is deemed dominant if they are of the same type (reinforcing or balancing) as the stock behaviour, and combined impact exceeds the sum of the total impacts of the opposite type. Often this will be a single loop, but where no single loop has more impact than the opposing loops, then the smallest number of loops, with the largest impacts is used. This is called the dominant loop set[7].

[6] In a nonlinear model the loop gain may be a function of the variables and will change over time. Mathematically it is possible to construct a model where the loop gain can change polarity within the bounds of the model, however such models are rare in system and population dynamics. Normally, loop gain keeps a fixed polarity, thus the impact of a first order loop will retain a fixed polarity. Higher order loops may have impacts that change polarity such that the polarity of the gain is preserved, see section 5.

[7] This definition of dominance was explored in the first part of this workshop.

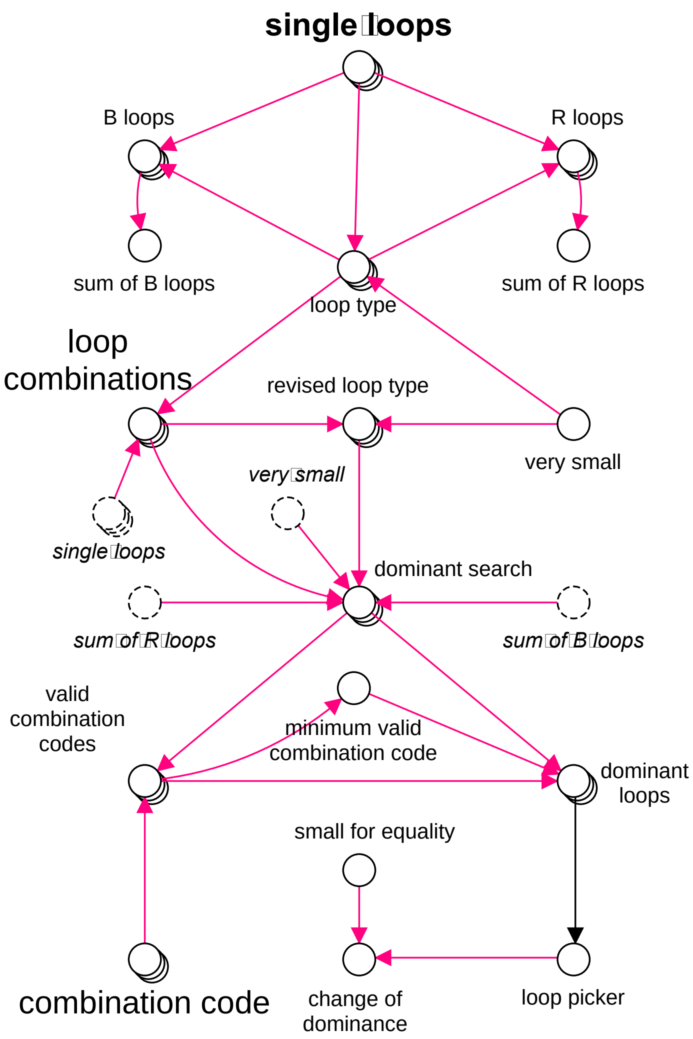


Figure 21: Loop picker algorithm. "Single loops" is the input; "loop picker" and "change of dominance" are the outputs

Step 5 – Connect Loop Picker Algorithm

a) Set Array Name and Element Names

The loop picker algorithm uses an array to hold the impacts of the different loops. The array indices can use the loop names. You do not need to know how to use arrays to follow these steps. The array is already set up but needs more elements.

The model has four possible conditions of dominance: R, B1, B2 and B1B2.

- Open the array editor on the "model" menu and change the array name from *Loops* to *Population Loops*[8].
- Increase the array size to 4.
- Name the dimension element names R, B1, B2 and B1B2 in that order
- Close the array editor.

Check with Figure 22.

[8] Renaming avoids name clashes in models with more than one stock that need additional loop picker algorithms.

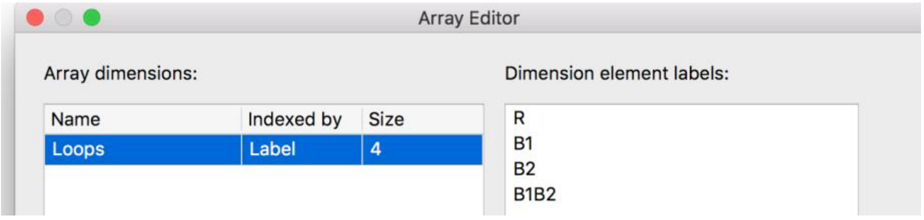


Figure 22: Array editor

Thus, there is a mapping from the array index to the names:

Index	Element Name
1	R
2	B1
3	B2
4	B1B2

b) Connect Loop Impacts to *single loops*

- Ghost and connect the 3 impact converters from step 3 to *single loops*, Figure 23:

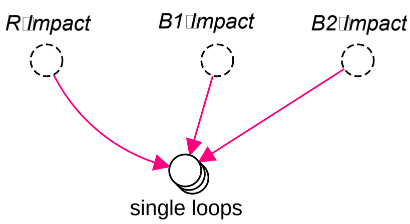


Figure 23: Connect impacts to "single loops"

- Set the formula in *single loops* so that the correct impact is placed next to the correct array element label, with the combination B1B2, having zero, see Figure 24:

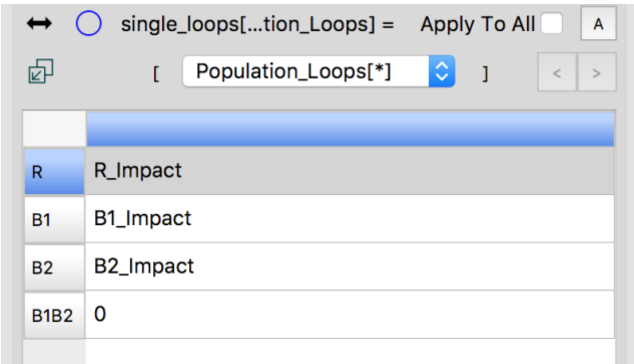


Figure 24: Formulae in "single loops".

The combined impact of *B1* and *B2* is set correctly elsewhere, step c).

c) Edit Loop Combinations

- <u>Set</u> *loop combinations*, so that the first three elements are the same as single loops, e.g. *single_loops[R]*. This can be done by clicking single_loops from the options, which initiates a menu so that you can choose the element R, B1 etc.
- <u>Set</u> the B1B2 element of loop combinations to *single_loops[B1]+single_loops[B2]*. That is the sum of the impacts of the two loops. *loop combinations* should look like Figure 25:

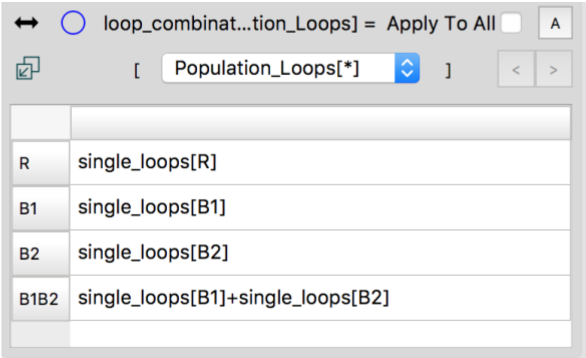
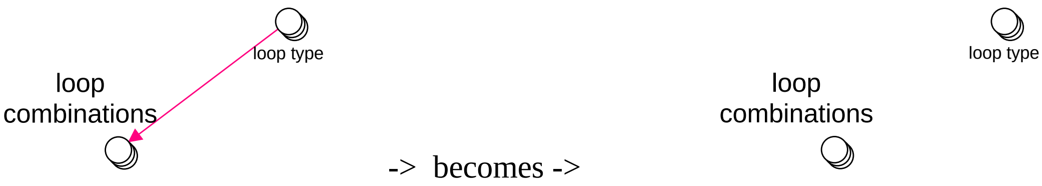


Figure 25: Set loop combinations

Closing the dialogue box should give an error as loop type has not been used. It is only used when loops may change the polarity of their impact. As this does not normally happen in a first order model the connection from *loop type* is not needed here.

- Remove the connector from *loop type* to loop *combinations:*



d) Edit Combination Code

This converter indicates how many loops are combined in each element. The first three elements are single loops, hence coded 1. B1B2 is two loops combined thus coded 2.

- <u>Set</u> *combination code* to:

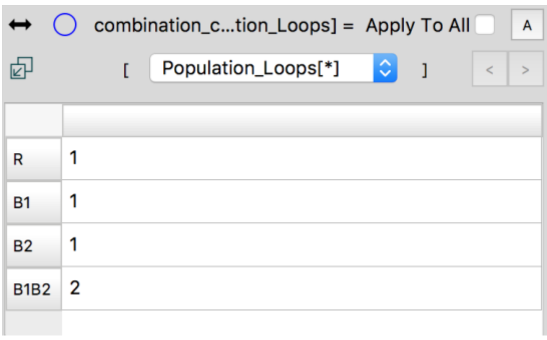


Figure 26: Set combination code.

Check that there are no remaining errors on converter formulae.

Interpreting Results

This can be done in a graph.

Loop Picker

The Loop Picker identifies which loop or loop combination is dominant, i.e. explains the behaviour of the stock *Population*. It can be displayed on the same graph as *Population.*

- Add *loop picker* to the population graph on the **right hand** axis.
- Fix the scale of *loop picker* to something larger than needed, e.g 0 .. 5, to make the graph easier to read, Figure 27
- Run the model.

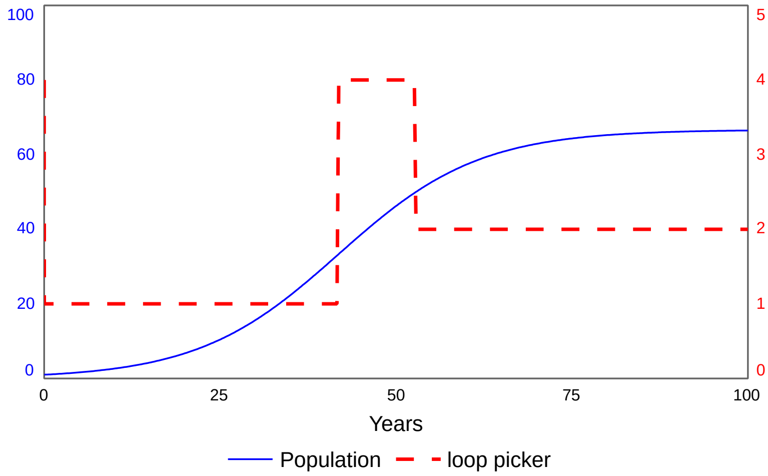


Figure 27: Population and loop picker

Holding the mouse with left button down over the graph should show the loop picker change from 1 to 4 and finally to 2.

Remember the identification of index value with loop:

Index	Element Name
1	R
2	B1
3	B2
4	B1B2

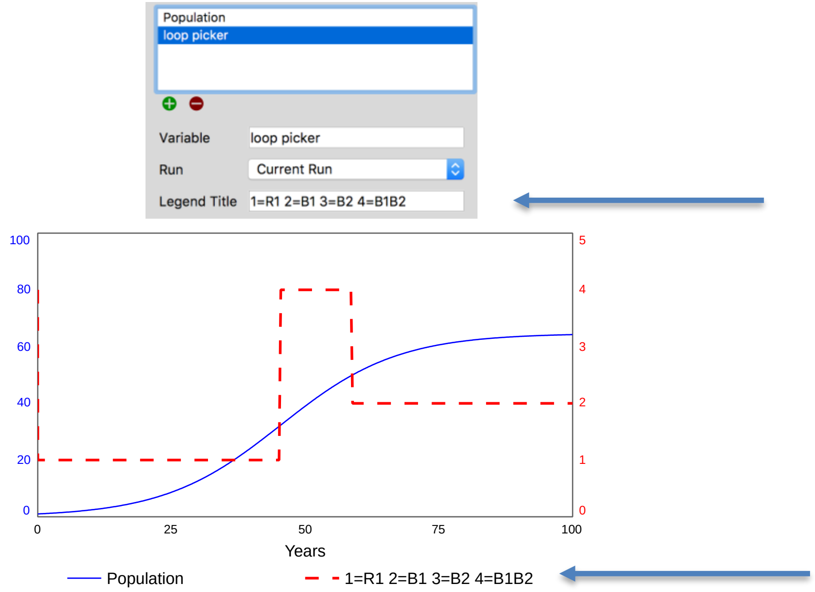
Thus, the three regions of dominance are firstly *R*, then the combination of *B1* and *B2*, and finally *B1* on its own.

Thus, the interpretation is that the reinforcing loop *R* causes the accelerated growth in the population. However, the dominance shifts to the balancing loops which slow the growth. Hence a limits-to-growth model is sometimes called shifting loop dominance. The reinforcing dominance is before the point of inflection, whereas dominance by the balancing loops is after the point of inflection.

The balancing dominance has two phases: firstly, where both balancing loops are needed to overcome the reinforcing loop *R*; then a phase where *B1* is sufficiently strong on its own. That loop

B2 has brought the change of dominance in early helps explain why the population does not reach its carrying capacity.

Unfortunately, Stella does not display the array element name on the graph, only the numerical index when mouse is held down. Thus, it is a good idea to put in a text box with the mapping between the index and the loop/loop combination. Or you could modify the graph legend for loop picker, e.g.:



You can now experiment with different parameter values and observe the effect on the loop dominance.

Change of Dominance

As an alternative, the population can have the points where dominance changes superimposed, Figure 28. Place the converter *change of dominance* on the right-hand axis of the graph. Although this display does not say which loop dominates in the three regions, it may be useful for presentation purposes, with the loop information extracted from the loop picker.

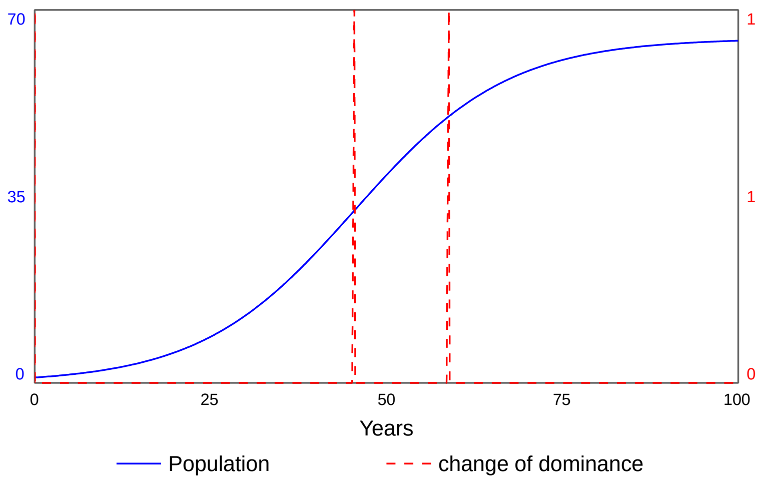


Figure 28: Population and change of dominance

5 Loop Dominance in a Second-Order Model

Introduction

When applying the loop impact method to a model with second (or higher) order loops the chief thing to remember is that the impact on a stock in such loops **may** change polarity. Thus, as a precaution when using the impacts of second order loops you need to consider that they could act positively – an accelerating/reinforcing effect, or negatively – a decelerating/balancing effect. This has implications in the edits of loop combinations.

The model we will use is an overshoot-and-collapse archetype – in this case a deer population living in a finite area, which places limits on their food supply, *Vegetation*, Figure 29.

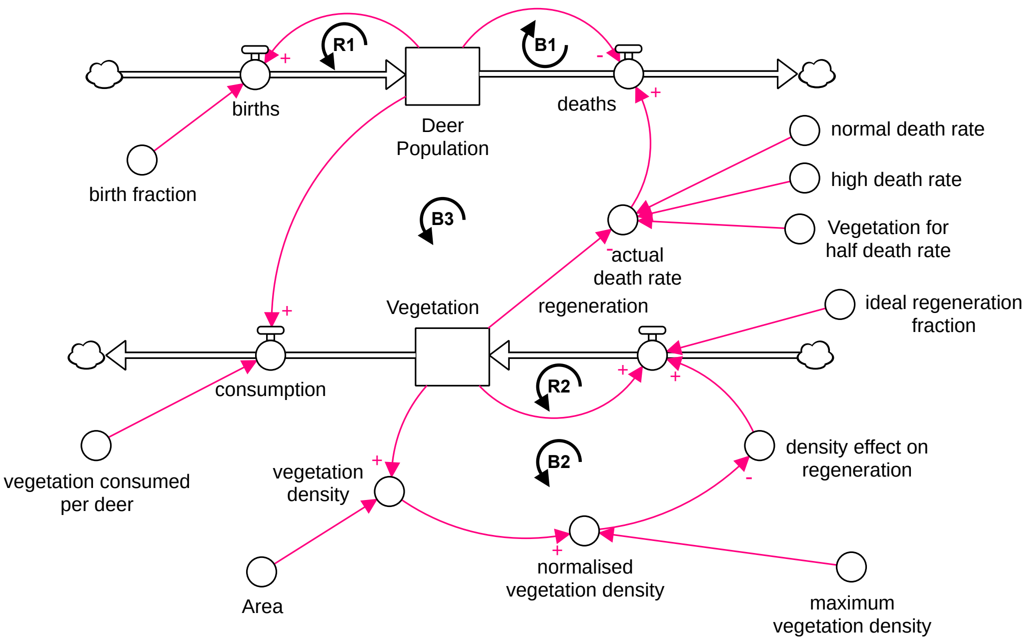


Figure 29: Deer and vegetation, overshoot and collapse model

The fine detail of the formula can be largely ignored; the key is the overall structure. There are 5 loops: *R1* and *B1* on the deer alone, *R2* and *B2* on the vegetation alone, and *B3* which contains both stocks and is thus a second-order loop.

Each stock needs to be analysed separately. The following deals with the loop impact analysis on *Deer Population*. The analysis of *Vegetation* is left as an exercise.

Obtain the Model

- Download the model from: https://sociomechanics.com/loop-impact/intro/

Step 1 – Insert Loop Identifiers (*Deer*)

Note the pathways into the stock *Deer Population*[9]. The stock has three pathways into its flows. Two from itself, *R1* and *B1*; and one from *Vegetation*, which we associate with loop *B3*. *B1* and *B3* combine in the flow *deaths*, thus the formula in *deaths* needs to be examined. It is multiplication.

- Modify the model in Figure 29 with the three loop identifiers: *R1 Identifier*, *B1 Identifier* and *B3 D Identifier*, Figure 30, moving the connections of the parameter *birth fraction*:

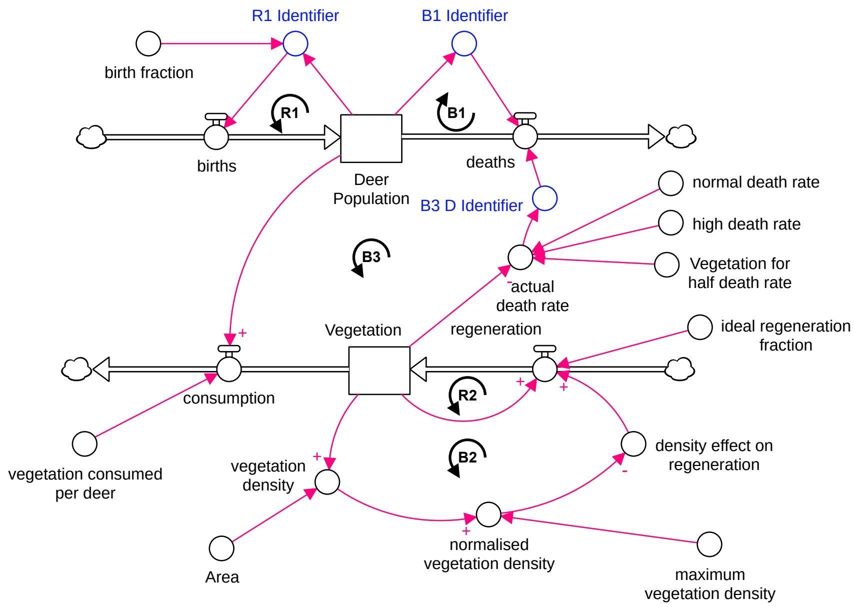


Figure 30: Overshoot and collapse with loop identifiers on Deer Population

- The modified formulae should be:

INFLOWS:		
births	=	*R1_Identifier*
OUTFLOWS:		
deaths	=	*B1_Identifier*B3_D_Identifier*
B1_Identifier	=	*Deer_Population*
B3_Identifier	=	*actual_death_rate*
R_Identifier	=	*Deer_Population*birth_fraction*

Step 2 – Compute Loop Impacts (*Deer*)

Rate of Change

- Ghost the flows *births* and *deaths*, and connect to a new converter *deer rate of change*, Figure 31, where the formula is: *deer rate of change = births – deaths*

[9] In more complex models the number of independent loops is not unique, but the number of independent pathways is always unique, Kampman, 2012; Hayward & Roach, 2018.

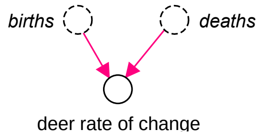


Figure 31: Deer rate of change

Loop Impacts

- Ghost *deer rate of change* and *R1 Identifier* and connect to a new converter *R1 Impact*, Figure 32, where the formula is:

  R1 Impact = DERIVN(R1_Identifier, 1)/ (deer_rate_of_change + 0.0000001)

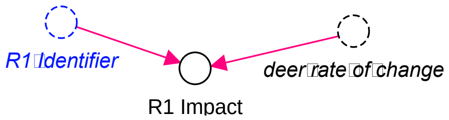


Figure 32: Impact of R1 on Deer Population

Note the addition of a very small number in the denominator. This is to avoid the possibility of a zero divide and will also generally improve the numerical accuracy of the loop impact calculation. Models with two or more stocks are prone to these problems. The value of the number is one of trial and error.

The impacts for *B1* and *B3* on the deer population use the multiply rule as they combine in the outflow multiplicatively.

- Ghost *deer rate of change*, *B2 Identifier*, and *B3 D Identifier* and connect as in Figure 33
-
- Set the formula for the impact of *B1* where *B1 identifier* is active and *B3 D identifier* is passive:

  *B1_Impact = -B3_D_Identifier*DERIVN(B1_Identifier, 1)/(deer_rate_of_change + 0.0000001)*

- Likewise set the formula for *B3 D Impact*, where *B3 D identifier* is active and *B1 identifier* is passive:

  *B3_D_Impact = - B1_Identifier *DERIVN(B3_D_Identifier, 1)/ (deer_rate_of_change + 0.0000001)*

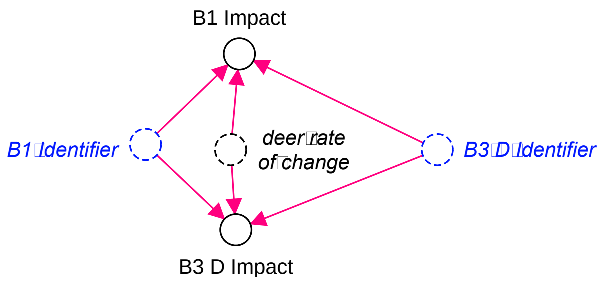


Figure 33: Impacts of B1 and B3 on Deer Population

As the loops *B1* and *B3* are on the outflow, the formula for the impacts are negative.

Step 3 – Identify Possible Loop Combinations (*Deer*)

The three loops are *R1*, *B1* and *B3*. As *R1* and *B1* are first order they are likely to be fixed in polarity: *R* will have positive impact, and *B1* will have negative impact. However, *B3* is second order and could change the polarity of its impact on a given stock. Thus, *B3 D Impact* could have either positive or negative impact and thus combine with either of the first order loops.

Thus, there are 5 combinations: R1, B1, B3, R1B3, B1B3. Thus, the array in the loop picker will have 5 elements.

Step 4 – Copy Loop Picker Algorithm

- Ensure your overshoot-and-collapse model Stella file is open.
- Download *Loop Picker.stmx* from https://sociomechanics.com/loop-impact/intro/ if you have not already done so.
- Open the Stella file *Loop Picker.stmx*
- Select and copy all the converters in the model
- Close the Stella WINDOW by clicking the top left icon on the window for *Loop Picker.stmx* i.e. do not quit Stella.
- Paste the algorithm in the overshoot-and-collapse model, Save the file.

Step 5 – Connect Loop Picker Algorithm (*Deer*)

a) Set Array Name and Element Names

- Open the array editor on the “model” menu and change the array name from *Loops* to *Deer Loops*.
- Increase the array size to 5.
- Name the dimension element names R1, B1, B3, R1B3 and B1B3 in that order
- Close the array editor.

Check with Figure 34.

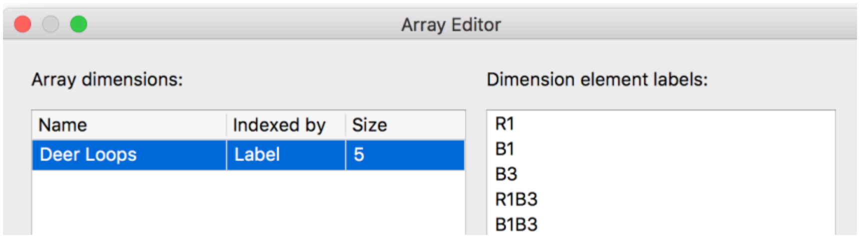


Figure 34: Array editor

Thus, the mapping from array index to loop name is :

Index	Element Name
1	R1
2	B1
3	B3
4	R1B3
5	B1B3

b) Connect Loop Impacts to *single loops*

- <u>Ghost</u> and <u>connect</u> the 3 impact converters from step 3 to *single loops*, Figure 35

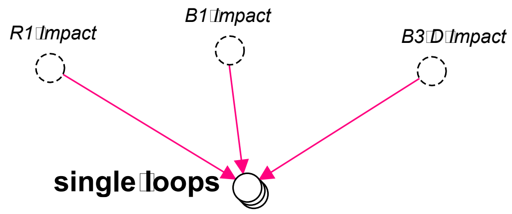


Figure 35: Connect impacts to "single loops"

- <u>Set</u> the formula in *single loops* so that the correct impact is placed next to the correct array element label, with the combinations, having zero, see Figure 36

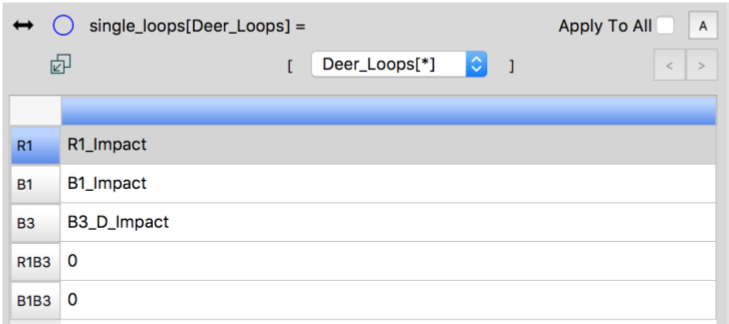


Figure 36: Formulae in "single loops".

c) Edit Loop Combinations

- <u>Set</u> *loop combinations*, so that the first three elements are the same as single loops, e.g. *single_loops[R1]*, etc (check Figure 37).

For the loop combinations we now need to determine whether the impact of *B3* on *Deer Population* is positive (reinforcing), or negative (balancing). For this we need to use the converter *loop_type* which will determine that effect at any given point in time. If *loop_type* of *B3* is 1, then its effect is reinforcing, if *loop type* is 0 then it is balancing.

An exact integer test in a numerical model can be subject to inaccuracies. Thus, the test will be

loop_type[B3] > 0.5 Reinforcing effect
loop_type[B3] < 0.5 Balancing effect

- <u>Set</u> the R1B3 element of loop combinations to
  IF(loop_type[B3]>0.5) THEN(single_loops[R1]+single_loops[B3]) ELSE(0)

  That is, the combination is only the sum of the impacts of the two loops when *B3* is acting in a reinforcing/accelerating way on *Deer Population*. Otherwise the combination is rejected.

- <u>Set</u> the B1B3 element of loop combinations to
  IF(loop_type[B3]<0.5) THEN(single_loops[B1]+single_loops[B3]) ELSE(0)

Check your converter with Figure 37

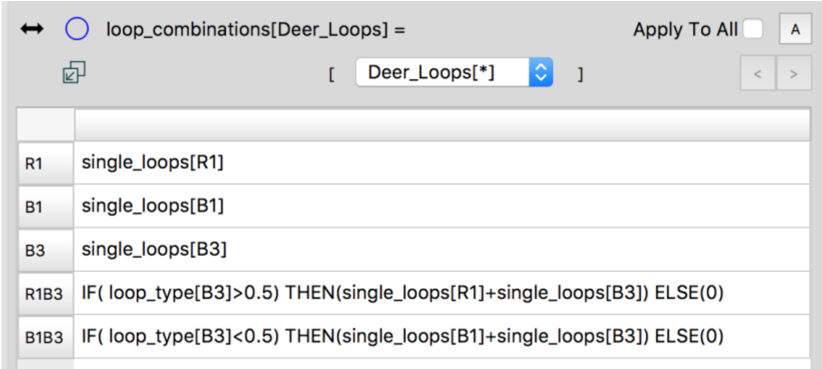


Figure 37: Set loop combinations

Thus, *B3* will be combined with the appropriate type of loop depending on its effect on the stock *Deer Population.*

d) Edit Combination Code

- Set *combination code* to:

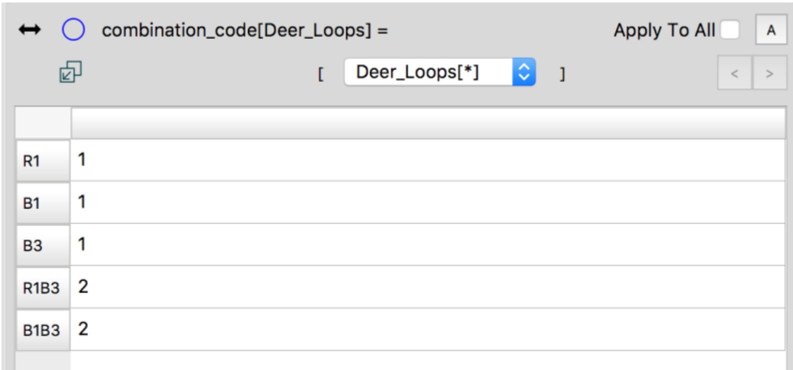


Figure 38: Combination codes

Interpreting Results

Set up a graph with *Deer Population* on the left-hand axis and *loop picker* on the right-hand axis. To help, rename the loop picker legend to 1=R1; 2=B1; 3=B3; 4=R1B3; 5=B1B3

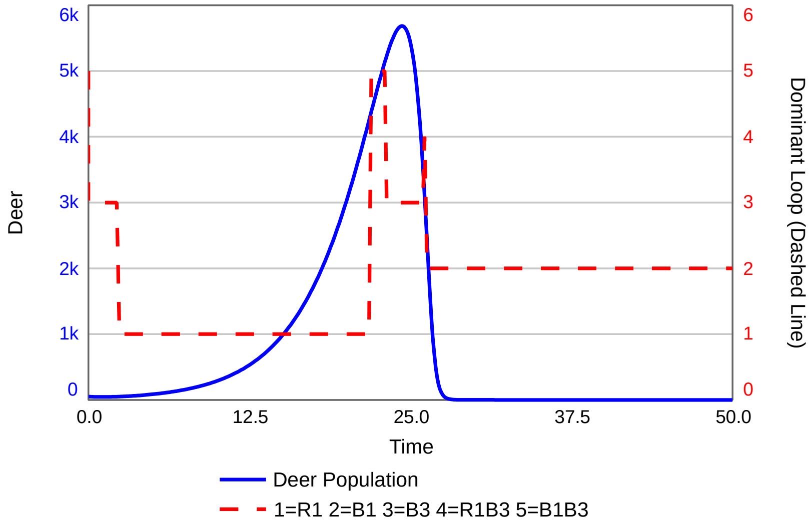


Figure 39: Deer Population and loop dominance

Initially, the largest impact has come from loop *B3*. This is transient behaviour induced by the initial conditions used. The initial food supply (*Vegetation*) has been set too low for the *Deer Population*, which falls slightly during that period. Once the vegetation has increased, the deer population grows dominated by loop *R1* (from time 2.3).

R dominates until time 22 when the combined impact of *B1* and *B3* acts to slow the growth. From time 23.1 to time 26.1 *B3* is dominates. It is *B3* that is responsible for turning growth to decline. Up to the maximum *B3* acts in a balancing way on *Deer Population*. After the maximum it has flipped polarity from minus to plus, acting in a reinforcing way[10].

After time 26.1 the decline in the deer population slows due to the effect of *B1*. It looks as if at time 26.1 there may be a momentary period where *R1* and *B3* dominate together. This may just be a numerical error, which can be checked by decreasing DT. But even if it is not, such a brief phenomenon adds nothing to the explanation.

The problem in this system is that the *B3*, the loop meant to correct the deer population, is too weak for too long. This can be seen by plotting the 3 impacts on the same scale, Figure 40 dotted line.

[10] Of course the impact of B3 on *vegetation* must have flipped polarity at the same time, so that the product of the impacts, the loop gain, keeps its polarity intact. This will be discovered in the exercise.

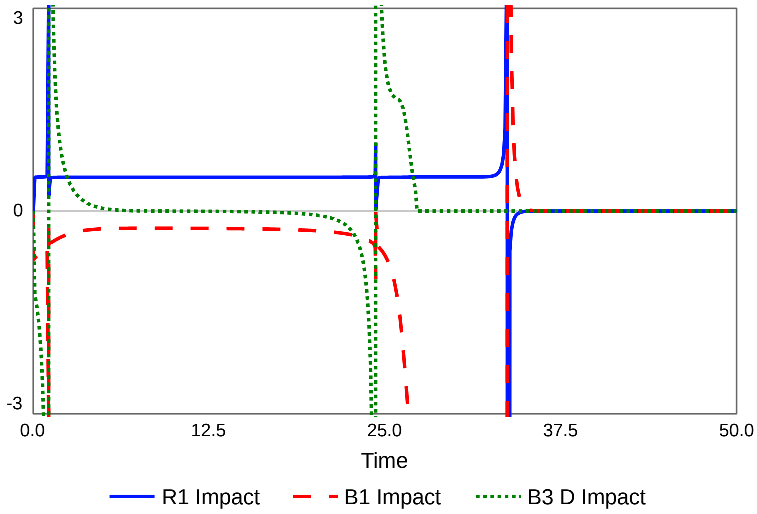


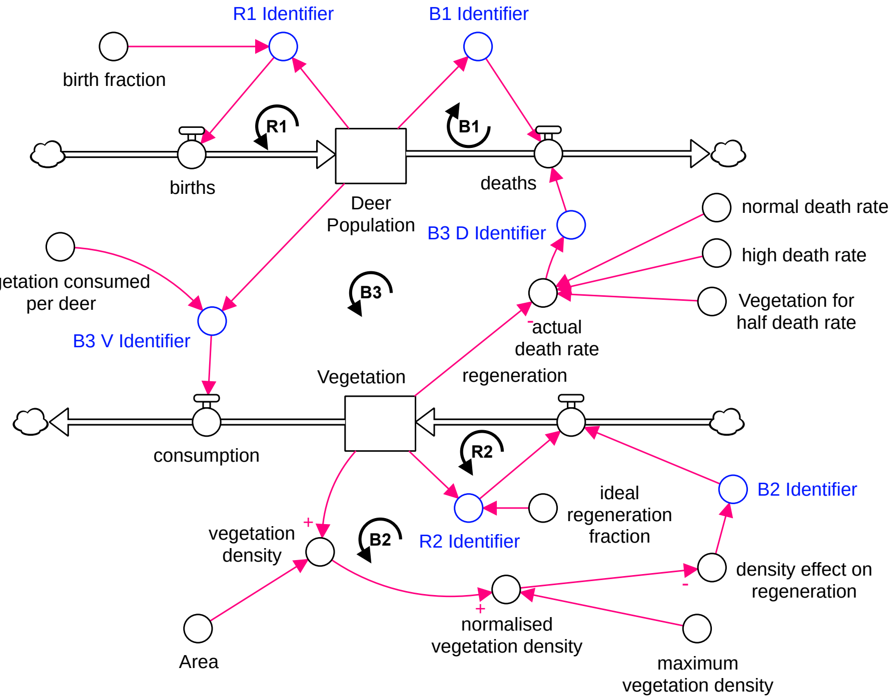
Figure 40: Loop impacts compared

Plotting the impacts is a useful guide to loop behaviour even if a loop is not dominant.

6 Exercise – Loop Dominance on Vegetation

As an exercise do a loop impact analysis on Vegetation. The analysis on both stocks is need for a system-wide explanation of the behaviour. As a guide, the headings, with screen shots are below.

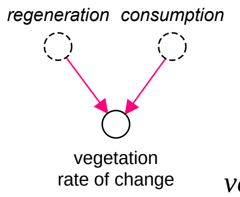
Step 1 – Insert Loop Identifiers (*Vegetation*)



*regeneration = R2_Identifier*B2_Identifier*
consumption = B3_V_Identifier
*R2_Identifier = Vegetation*ideal_regeneration_fraction*
B2_Identifier = density_effect_on_regeneration
*B3_V_Identifier = Deer_Population*vegetation_consumed_per_deer*

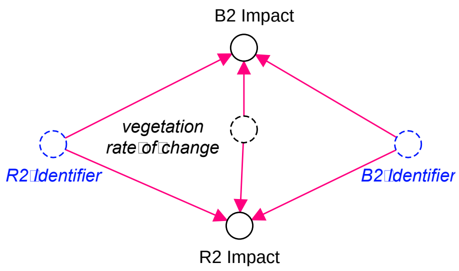
Step 2 – Compute Loop Impacts (*Vegetation*)

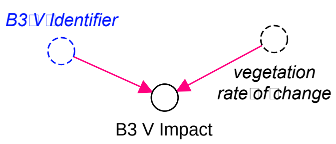
Rate of Change



vegetation_rate_of_change = regeneration-consumption

Loop Impacts





*R2_Impact = B2_Identifier*DERIVN(R2_Identifier, 1)/(vegetation_rate_of_change+0.0000001)*
*B2_Impact = R2_Identifier*DERIVN(B2_Identifier, 1)/(vegetation_rate_of_change+0.0000001)*
B3_V_Impact = -DERIVN(B3_V_Identifier, 1)/(vegetation_rate_of_change+0.0000001)

(*B3* is on an outflow)

Step 3 – Identify Possible Loop Combinations (*Vegetation*)

R2, B2, B3, R2B3, B2B3. Remember *B3* will change polarity.

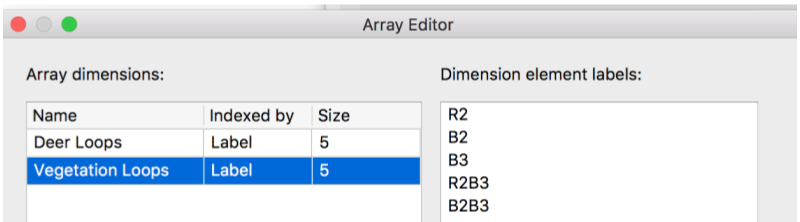
Step 4 – Copy Loop Picker Algorithm

IMPORTANT. Copy the loop picker from the original file. This will make sure a new array for the Vegetation loops are set up.

Note that all the converter names have a 1 put on the end.

Step 5 – Connect Loop Picker Algorithm (*Vegetation*)

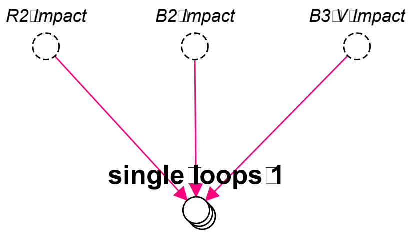
a) Set Array Name and Element Names

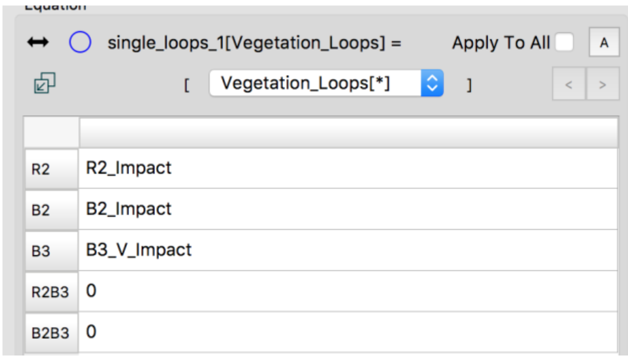


Thus the identification is

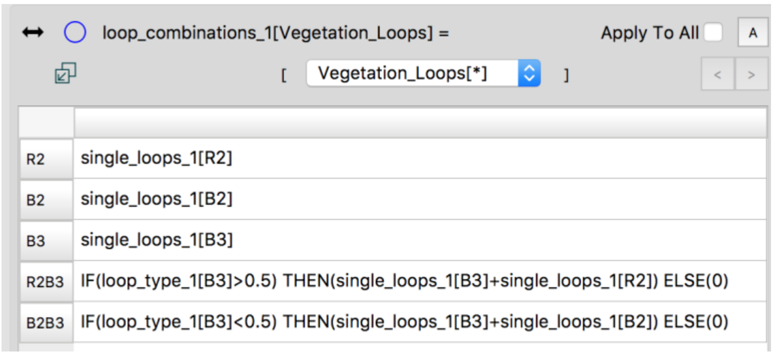
Index	Element Name
1	R2
2	B2
3	B3
4	R2B3
5	B2B3

b) Connect Loop Impacts to *single loops*

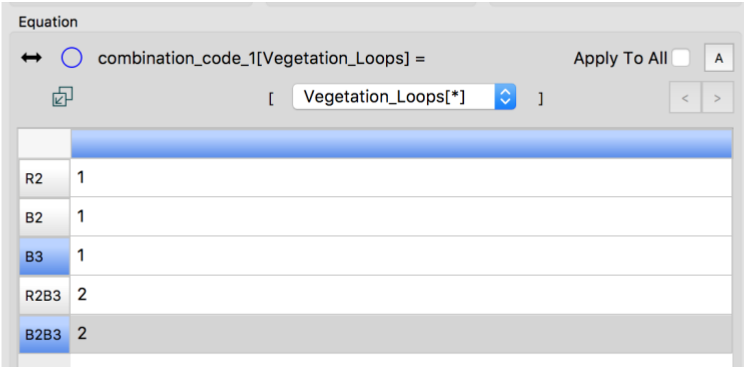




c) Edit Loop Combinations

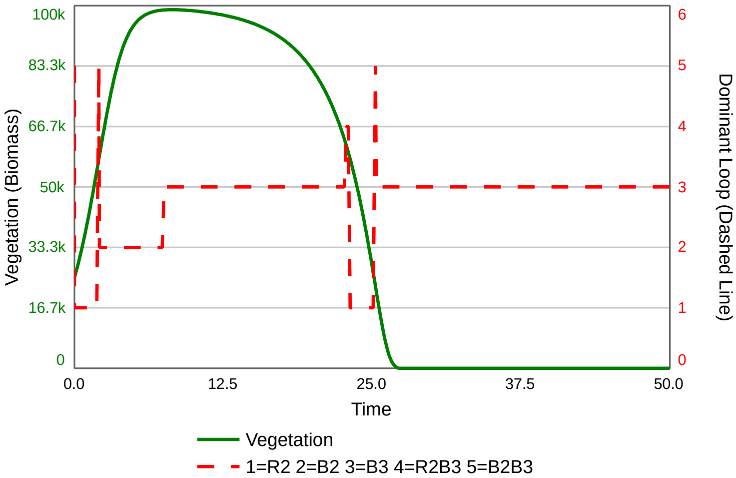


d) Edit Combination Code



Interpreting Results

Use *loop picker 1* to identify the loops in the graph and relabel the legend



Growth is through *R2*, slows through *B2*, the capacity loop. The Vegetation nears its maximum before the loop *B3* is strong enough to help slow growth. Collapse occurs because the environment, not the deer's food consumption, dominates the corrective slowing period, until it is too late.

After the maximum, *B3* accelerates the decline, followed by *R2* accelerating the decline. *B3* only flips to a balancing impact near the end of the vegetation's decline, the same point as the peak in the deer numbers. There is only a short period near the end where *B3* dominates both stocks at the same time. With dominance, it is important to say which stock is affected by which loop (or stock).

You might like to adjust parameters to see if collapse can be avoided and investigate what changes in the loop dominance. You might have to change the time horizon.

References

The method is described in the 2014 paper and extended in the 2018 paper.

Hayward J, Boswell GP. 2014. Model behaviour and the concept of loop impact: a practical method. System Dynamics Review 30(1): 29–57.

Hayward J, Roach PA. 2017. Newton's laws as an interpretive framework in System Dynamics. System Dynamics Review 33(3-4), 183-218. DOI: 10.1002/sdr.1586.

Jeremy Sato compared Loop Impact with other methods and gave valuable insights on the method

Sato JB. 2016. State space analysis of dominant structures in dynamic social systems. PhD thesis. Washington University, St Louis, MO.

The Loop Impact method is built on previous work, particularly the Pathway Participation Metric of Mohammed Mojtahedzadeh and the Loop Deactivation method of David Ford.

Duggan J, Oliva R. 2013. Methods for identifying structural dominance. System Dynamics Review, special virtual issue. Available: http://onlinelibrary.wiley.com/journal/10.1002/(ISSN)1099-1727/homepage/VirtualIssuesPage.Html#methods

Ford DN. 1999. A behavioural approach to feedback loop dominance analysis. System Dynamics Review 15(1): 3–36.

Forrester NB. 1982. A dynamic synthesis of basic macroeconomic theory: Implications for stabilization policy analysis, PhD thesis, Sloan School of Management, MIT: Cambridge, MA.

Gonçalves P. 2009. Behavior modes, pathways and overall trajectories: eigenvector and eigenvalue analysis of dynamic systems. System Dynamics Review 25(1): 35–62.

Güneralp B. 2006. Towards coherent loop dominance analysis: progress in eigenvalue elasticity analysis. System Dynamics Review 22(3): 263–289.

Huang J, Howley E, Duggan J. 2009. The Ford method: a sensitivity analysis approach. In Proceedings of the 27th International Conference of the System Dynamics Society. System Dynamics Society: Albuquerque, NM.

Kampmann CE. 2012. Feedback loop gains and system behaviour (1996). System Dynamics Review 28(4): 370–395.

Kampmann CE, Oliva R. 2008. Structural dominance analysis and theory building in system dynamics. Systems Research and Behavioral Science 25(4): 505–519.

Mojtahedzadeh M. 2008. Do parallel lines meet? How can pathway participation metrics and eigenvalue analysis produce similar results? System Dynamics Review 24(4): 451–478.

Mojtahedzadeh M, Richardson GP. 1995. Confusion in the polarity of major loops. In Proceedings of the 13th International Conference of the System Dynamics Society, Tokyo, Japan. Albany, NY: System Dynamics Society.

Mojtahedzadeh M, Anderson D, Richardson GP. 2004. Using Digest to implement the pathway participation method for detecting influential system structure. System Dynamics Review 20(1): 1–20.

Moxnes E, Davidsen PI. 2016. Intuitive understanding of steady-state and transient behaviors. System Dynamics Review 32(2): 130–155.

Richardson GP. 1986. Problems with causal loop diagrams. System Dynamics Review 2(2): 158–170.

Richardson GP. 1995. Loop polarity, loop dominance, and the concept of dominant polarity. System Dynamics Review 11(1): 67–88.

Richardson GP. 1997. Problems in causal loop diagrams revisited. System Dynamics Review 13(3): 247–252.