Predator-Prey Dynamics and Wildlife Management: A System Dynamics Model

Kumar Venkat
Surya Technologies
February 10, 2005

Table of Contents

Problem Definition ... 1
Model Development ... 2
Case 1 – basic predator-prey-vegetation system: ... 2
Case 2 – higher initial predator population: ... 5
Case 3 – impact of bounty hunting: ... 6
Case 4 – impact of prey (deer) hunting: ... 8
Case 5 – increased prey (deer) hunting: ... 9
Case 6 – impact of crowding: ... 9
Case 7 – impact of hunting policy: ... 11
Case 8 – impact of information delay: ... 13
Conclusions ... 15
Attachment 1: Vensim Model Equations for Case 1 ... 16
Attachment 2: Vensim Model Equations for Case 4 ... 18
Attachment 3: Vensim Model Equations for Case 7 ... 20

Problem Definition

The problem is one of modeling the population dynamics of a 3-species system consisting of vegetation, prey and predator. The predator population was significantly reduced due to bounty hunting starting in 1900. If predators still remain, although in small numbers, then we still need to study the 3-species system with a low initial value for predators. The issue at hand is the boom-bust cycle that such coupled populations typically go through. The first task is to simply model this and reproduce the cyclic behavior of the 3-species system.

The next step is to study the effect of higher initial values for the predator population – as if they had not been decimated by bounty hunting. This should give some insight into whether the cyclic behavior might have existed – possibly with different means, amplitudes and periods – prior to the start of bounty hunting, as it does in many other real-life population systems studied by ecologists. Will the populations – especially the prey (deer) population – stabilize?

Following this, we need to add logic for bounty hunting of predators, and then hunting of the prey species as well. Will the populations stabilize under hunting scenarios?

Based on the insights emerging from these experiments, the final step is to design and experiment with some policy "levers" to see if the system can stabilize.

Model Development

The simplest model of predator-prey dynamics is known in the literature as the Lotka-Volterra model[1]. It is based on differential equations and applies to populations in which breeding is continuous. Continuous breeding throughout the year is one of my simplifying assumptions in this exercise, which doesn't detract from the real insights provided by the model.

The basic Lotka-Volterra model consists of the following system of differential equations:

dN/dt = r * N - s * P * N
dP/dt = f * s * P * N – q * P

N is the prey population (number of individuals or total biomass), P is the predator population, r is the fractional birth rate of the prey population, s is the search efficiency (or attack rate) of the predator, f is the food conversion efficiency of the predator (i.e., how good the predator is at turning food into offspring), and q is the fractional death rate of the predator.

When the predator's food supply (prey population) increases, the predator population also increases, with a delay, as food gets converted into offspring. As the predator consumes food, the prey population decreases, also with a delay. The reduced prey population then reduces the predator population by limiting food supply, again with a delay, which then allows the prey population to recover after a delay, and so on. The feedbacks here with delays cause this simple model to produce the well-known coupled oscillations of the two populations. As Sterman points out (pp. 114), oscillation always involves negative feedback loops with delays.

Note that death by starvation is not explicitly shown in this model, but is modeled indirectly. When food is in short supply, the predator's reproduction rate goes down, indirectly reducing its population.

After extending the Lotka-Volterra model to include vegetation as a third state variable (which will be the baseline case for this exercise, with results similar to the given RBP), I will proceed to explore the effects of higher initial values for the predators, bounty hunters, prey (deer) hunters, and policies that could help stabilize the populations.

Case 1 – basic predator-prey-vegetation system:

The above Lotka-Volterra model assumes an unlimited food supply for the prey population. I have <u>extended</u> it, for this exercise, by adding a third state variable to model vegetation as another continuously-changing population, which limits the food available to the prey population:

dV/dt = k * V + x *y – m * V * N
dN/dt = t * m * V * N - s * P * N

[1] Begon, M., J.L. Harper and C.R. Townsend, ***<u>Ecology: Individuals, Populations and Communities</u>***, Oxford, England: Blackwell Science Ltd, 1996.

$dP/dt = f * s * P * N - q * P$

V is the vegetation population, k is the fractional regeneration (birth) rate of the vegetation, m is the search efficiency (attack rate) of the prey, and t is the food conversion efficiency of the prey. Also, x is the rate at which seeds are dispersed (“imported” via dispersion) into the area of interest from external sources and y is the probability that an “imported” seed will successfully contribute to the vegetation biomass in the area. Other variables are as defined in the previous section.

So, we finally have a system of three state variables governed by three simultaneous differential equations. This defines the baseline case for this exercise. A note on terminology: I have generally used the general terms “vegetation”, “prey” and “predator” to refer to the state variables, instead of “deer” or “cougar”.

In this exercise, I will use biomass (rather than integer numbers) to measure populations without loss of generality. Fractional numbers can be justified because a population will consist of both fully-grown and efficient individuals, as well as small-sized, growing and/or otherwise inefficient individuals. (Efficiency refers to the ability to find food, as well as the ability to reproduce using that food.) This assumption is consistent with most ecological models of such populations.

The Vensim model for this system is shown below in Figure 1 and the Vensim equations are in Attachment 1. Here are some highlights of the model:

- I have mostly avoided complex equations and IF-THEN-ELSE structures, based on experience with the previous exercise and our discussion. As a result, the stock-flow diagram and the equations are easier to read and understand.
- I have ensured that all three stocks will remain positive by applying a maximum death rate constraint (first-order control), based on Sterman’s discussion (pp. 545-549).
- I have set the predator’s search efficiency 10 times smaller than the prey’s search efficiency for an obvious reason: It is much easier for a prey to find food (vegetation) than for a predator to hunt a moving and somewhat intelligent prey.
- I have set the predator’s food conversion efficiency 20 times larger than the prey’s food conversion efficiency, because plant food is less calorie and nutrient intensive. It is much more efficient to convert prey into predator’s offspring via food assimilation and reproduction.
- Initial values of populations are (I have experimented with other values and obtained similar results):
    - Vegetation: 100000
    - Prey: 500
    - Predator: 10
- External seed dispersion rate is 300/month, with a survival probability of 0.25.
- Time is measured in months. The simulation time step is also 1 month. I have tried other time steps and found no problems – the model behaves the same way.
- I have checked all models in this exercise for consistency of units and model correctness using Vensim’s checking features.

- Crowding – limitations in physical space to accommodate predators and prey, regardless of food supply – is not included in Case 1, but is included in another model extension discussed later.

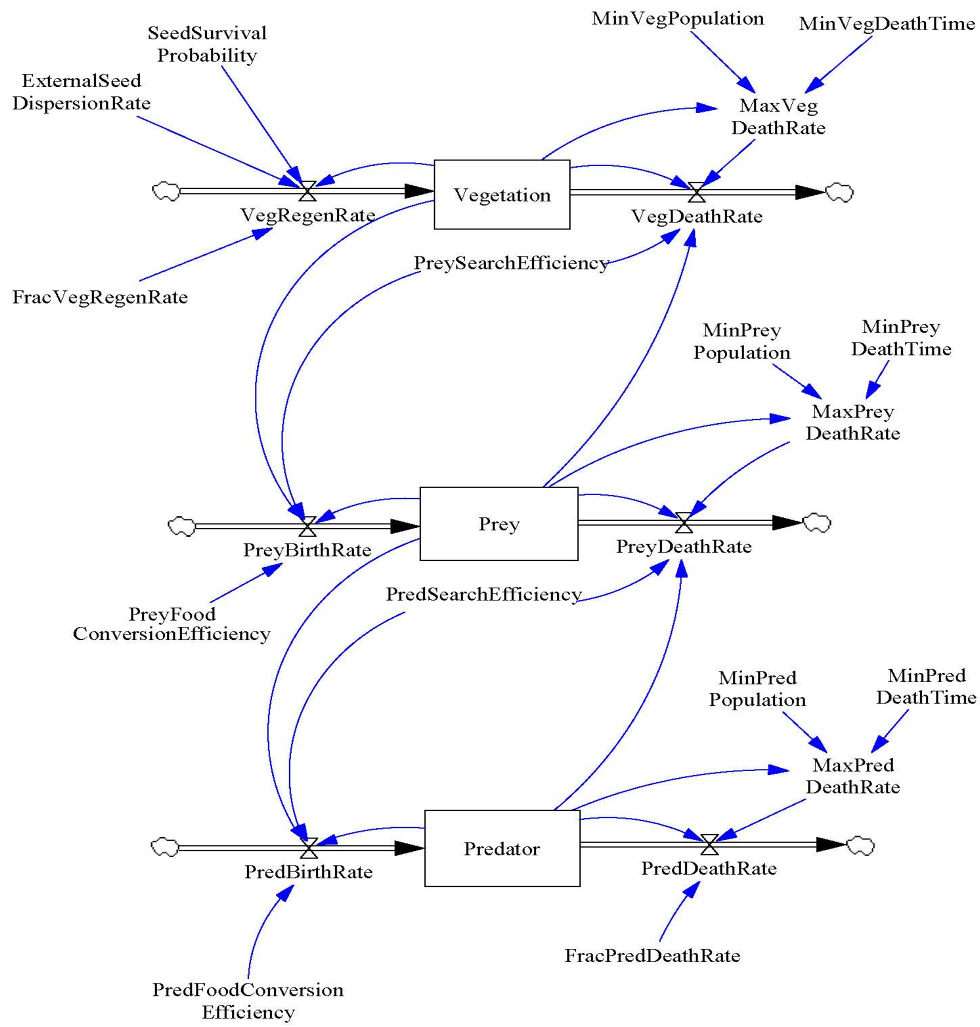
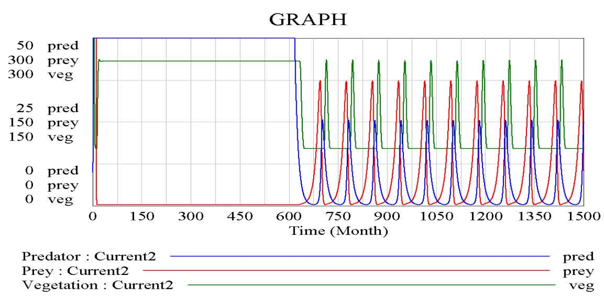


Figure 1: Case 1 (baseline) model of the Predator-Prey-Vegetation system

The results of the simulation are shown below in Figure 2. Once the system removes the excess levels of vegetation, it settles into the familiar coupled oscillations of all three populations. These results are similar to the given RBP, but obviously the values and curve shapes are different due to my specific assumptions and approach. These oscillations exhibit neutral stability: they continue indefinitely if undisturbed, and each disturbance leads to a new series of stable cycles around the same means but with different amplitudes. (Sterman refers to such oscillations as limit cycles, pp. 131.)

The vegetation population has a flat non-zero bottom due to the external seed source – I have verified this by running simulations without this external seed source. The external seed source itself is realistic (although actual values may vary widely), and it certainly prevents the vegetation from dying off completely during the cycles and keeps the system viable.



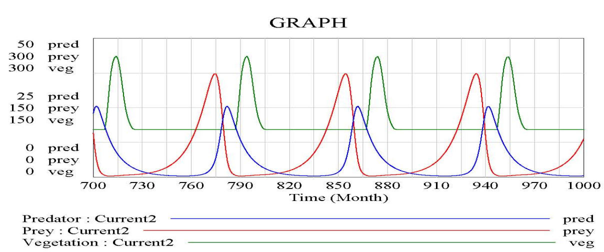


Figure 2: Case 1 results showing coupled oscillations

Case 2 – higher initial predator population:

I tried increasing the initial predator population (biomass) from 10 to 1000 (and even higher) and found no essential difference at all in the final behavior of the model. The only difference is in the early parts of the simulation, where the prey population becomes very small. Other than that, once the excess vegetation has been consumed by the prey-predator food chain without making any population extinct, all three populations settle into coupled oscillations.

Based on the results so far, for the particular model and assumptions that I am using, I don't see a way to stabilize the prey population simply by starting with a higher number of predators. A large number of predators could possibly kill off the entire prey population, in which case the predator population will also collapse to zero eventually. In my simulations, I did not allow any population to become entirely extinct – the combination of available vegetation and predator search efficiency allowed just enough prey biomass to survive and eventually grow.

In real life, coupled oscillations of predator and prey populations is common (a well-known example is that of the snowshoe hare and Canada lynx), but there are also significant cases where they are not so coupled. Achieving a stable population, or at least greatly reducing the boom-bust cycles, will almost certainly require some policy intervention, and I will examine this later.

Case 3 – impact of bounty hunting:

The Vensim model in Figure 3 below includes hunters killing off some predators. The results are shown in Figure 4. For these representative results, I used a total of 10 hunters each operating at an efficiency of 0.005 (probability of a predator being killed by a hunter in one time unit). The results, not surprisingly, show that the peak values of the predator and prey populations almost double and cycles become very pronounced. In other words, the systematic and sustained disturbance caused by bounty hunting simply shifts the system to a new series of neutrally stable cycles.

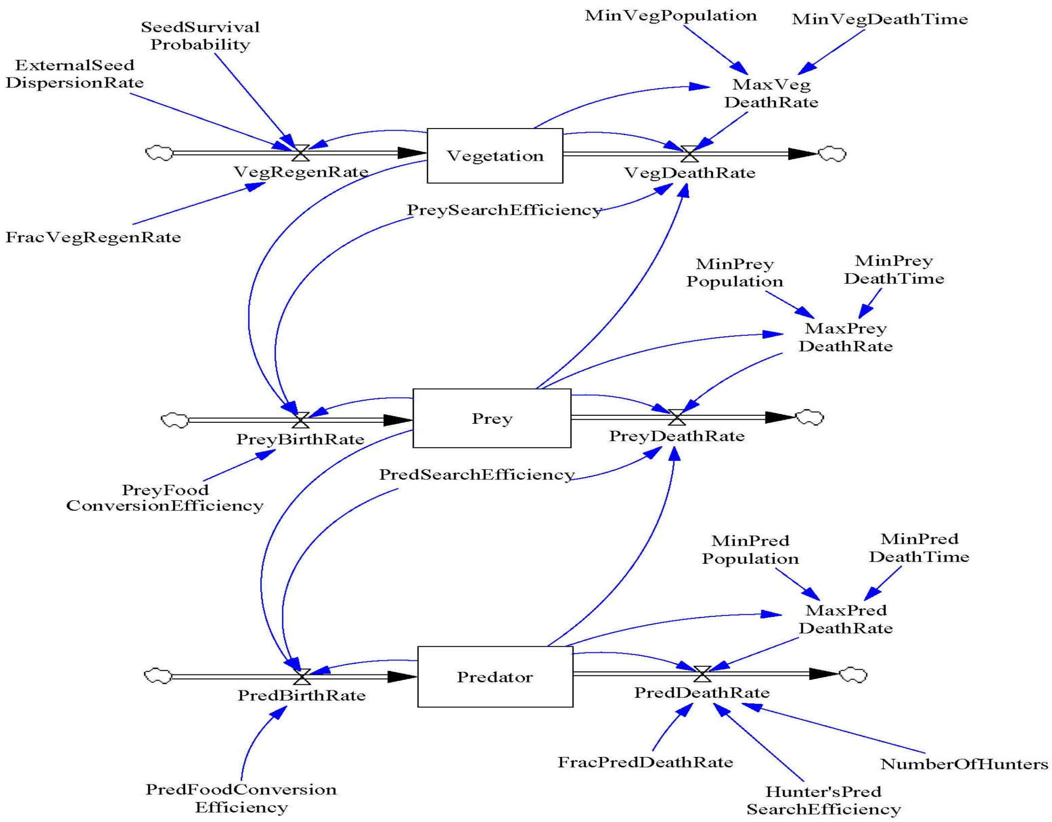


Figure 3: Case 3 model – impact of bounty hunting

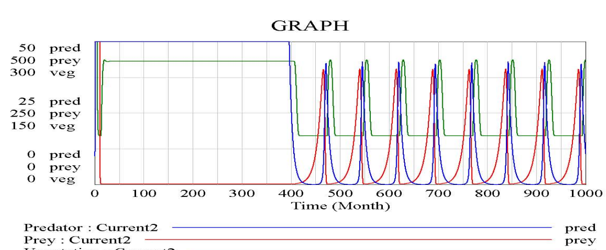


Figure 4: Case 3 results

Case 4 – impact of prey (deer) hunting:

The Vensim model in Figure 5 below includes hunters killing off some prey (deer) and some predators. I have added 10 prey hunters at an efficiency of 0.01 – this specifies that it is easier to successfully hunt deer than it is to hunt cougar. The results are shown in Figure 6, and the Vensim model equations are in Attachment 2.

The coupled oscillations are still there, but the period of the oscillation is much larger (increased from about 100 months to 2000 months) – a new series of neutrally stable cycles. The predator population is depressed due to hunting and lack of food (since prey is also being hunted by humans). Both prey and predator populations have narrow peaks and are depressed otherwise. Vegetation is now fairly stable as a result, with narrow peaks immediately following the prey population peaks.

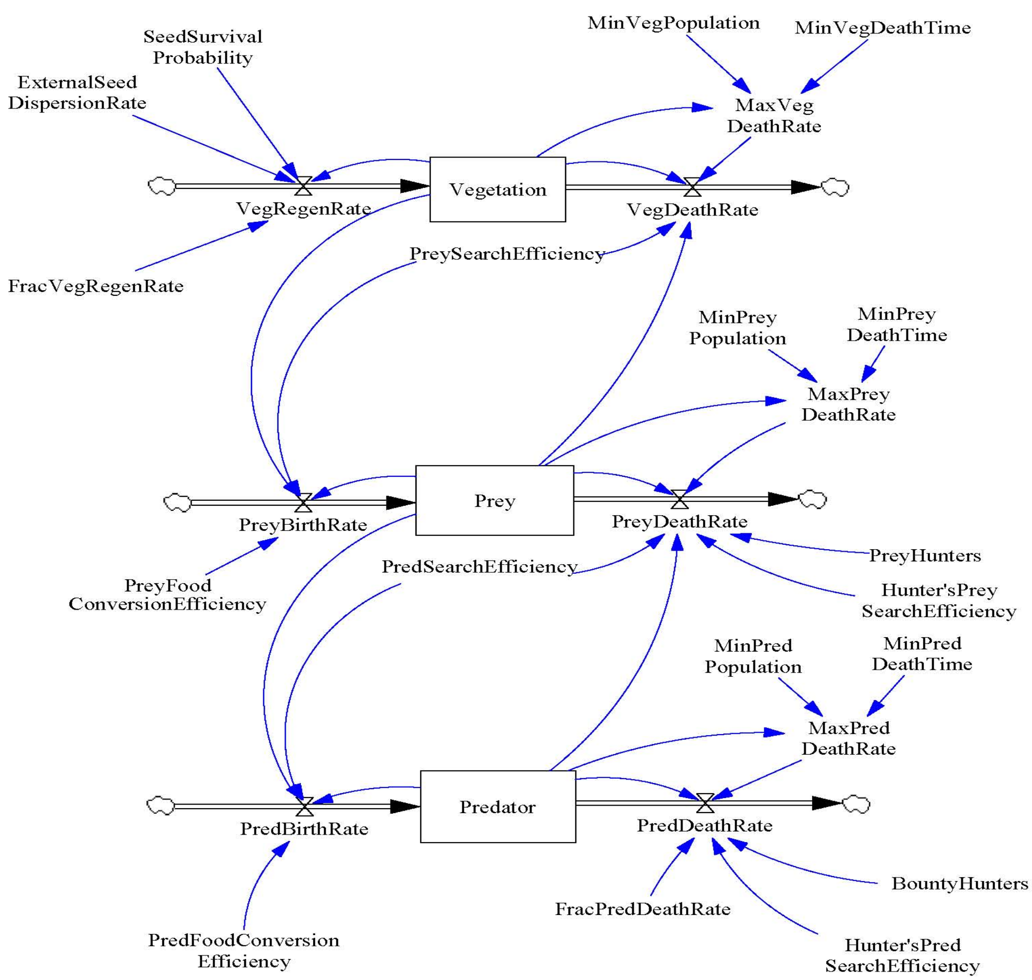


Figure 5: Case 4 model – impact of prey (deer) hunting

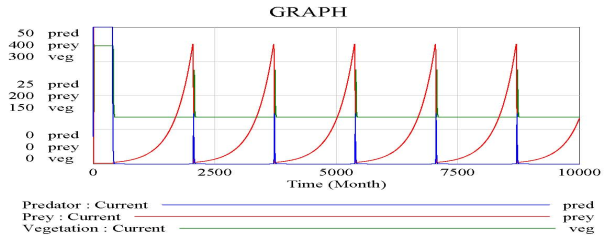


Figure 6: Case 4 results

Case 5 – increased prey (deer) hunting:

I ran several more simulations of hunting scenarios, with more hunters engaged in prey hunting, and the same number hunting predators or no hunting of predators. At higher hunting rates, the prey population drops to small values and the predator becomes extinct due to decreased availability of food. Both prey and vegetation stabilize at non-zero values without oscillations. Oscillations go away when predators die off. Human hunting alone does not introduce oscillations because the model includes no delay for humans to respond to increases in any population. The following table summarizes some of these scenarios and results.

Predator Hunters	Prey Hunters	Final Predator Biomass	Final Prey Biomass	Final Vegetation Biomass
10	11	0	4.65	110
0	12	0	4.375	120
0	15	0	3.75	150
0	11	0	4.65	120

Case 6 – impact of crowding:

An interesting ecological scenario is where resources other than food are fixed and must be shared by individuals of each species. One such constraint is physical space to accommodate a population, including typically non-overlapping ranges for predators. The Vensim model in Figure 7 limits both predator and prey populations to certain maximum values to reflect space limitations (one set of values is shown in the table below), but includes no hunting.

	Initial Value	**Maximum Value**
Vegetation	100000	None
Prey	50	60
Predators	12	15

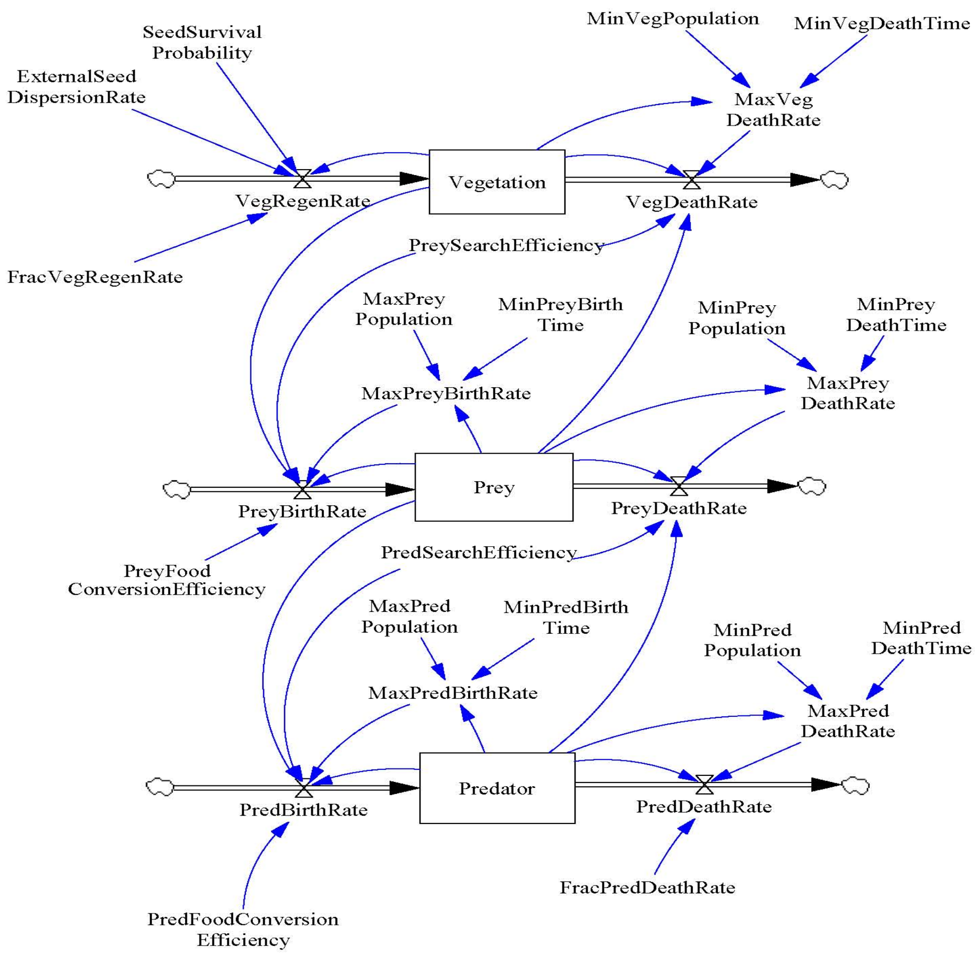


Figure 7: Case 6 model – impact of crowding

For the parameters shown in the table above, the results are shown below in Figure 8. Similar results can be obtained for a wide range of parameter values. The coupled oscillations are still present for prey and predator populations, but the amplitudes are quite small and the populations are nearly stable – possibly as stable as they can get endogenously without policy intervention.

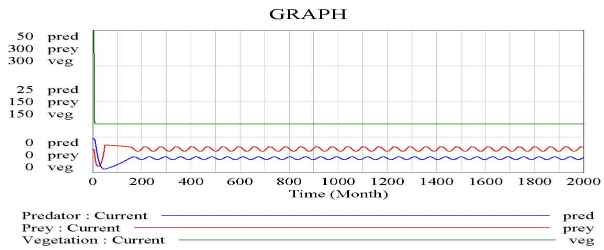


Figure 8: Case 6 results – near stability via crowding (endogenous)

Case 7 – impact of hunting policy:

The previous case shows how the predator-prey-vegetation system could reach a nearly stable state endogenously. When crowding is not an issue – i.e., when sufficient space and other resources are available, or when the desired population levels must be set well below what would result from crowding alone – the previous case suggests a possible solution. Hunting of both predator and prey populations could artificially enforce some upper limits for the populations.

The following Vensim model in Figure 9 is an extension of Case 4 (with both kinds of hunting), where hunting is allowed only when each population exceeds a certain minimum value. The Vensim model equations are in Attachment 3.

This, of course, assumes that frequent census efforts will be undertaken to dynamically control the hunting. With hunting of the prey population allowed only when it exceeds 75 (and no such limit on hunting of predators), the populations reach stability, as shown in Figure 10 below. There is still a small-amplitude noise-like fluctuation in the prey population, but it is not significant at all, and the population hovers right around 75. (More on this in Case 8.)

Interestingly, when the minimum prey population is set at 100, the predator-prey oscillations start again at a higher frequency than in many of the previous cases. When the minimum prey population is reduced to 50, it results in a lower frequency oscillation. This suggests there may well be an optimum stable value for the prey population, which can be found by running multiple simulations, as I have done. This ideal value is likely to be a function of the exact model, assumptions, parameters, etc. For my final model here, along with its underlying assumptions and parameters, 75 seems to be a very good value for the minimum prey population.

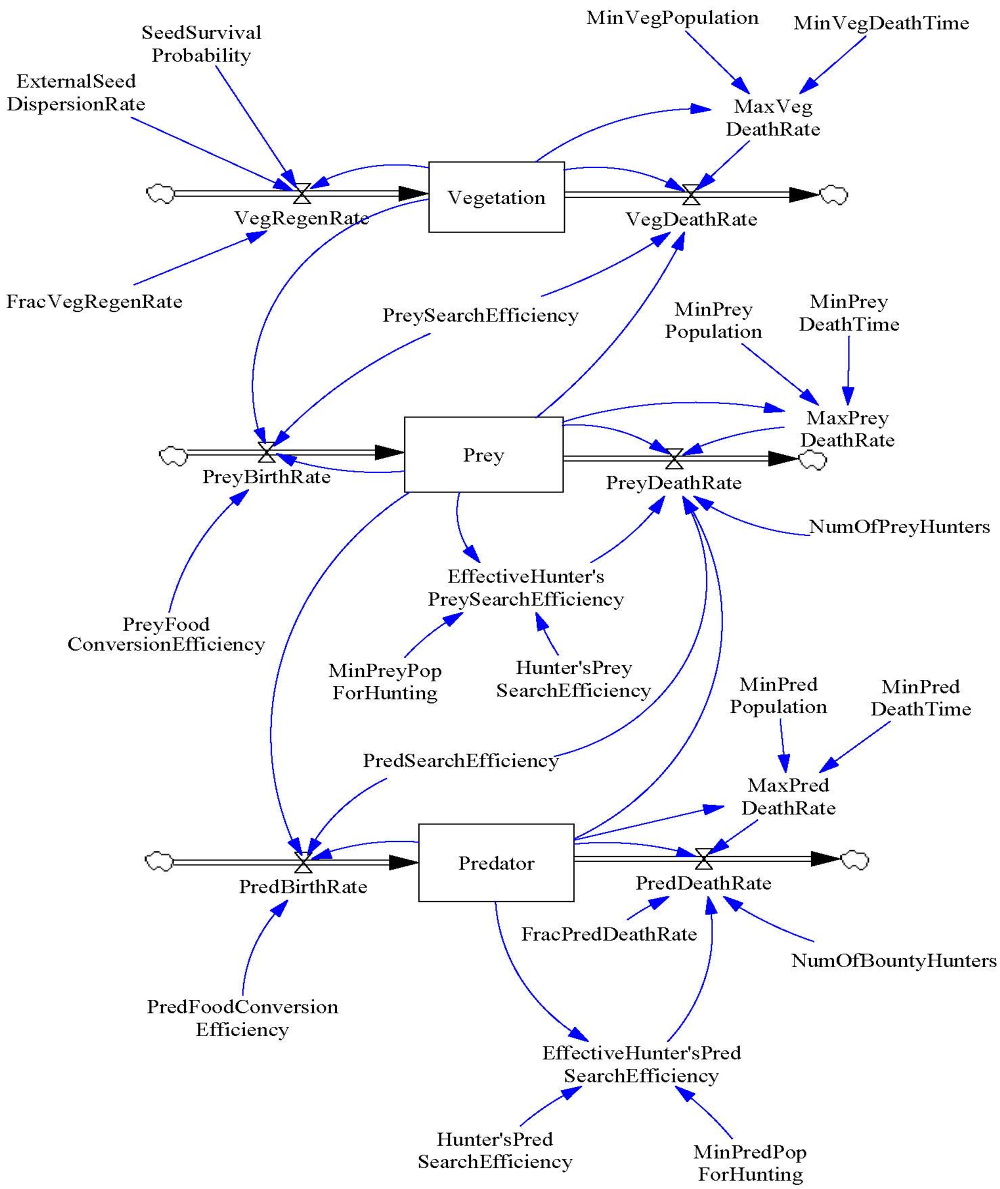


Figure 9: Case 7 model – impact of hunting policy

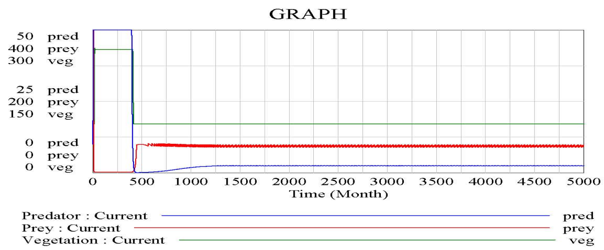


Figure 10: Case 7 results – stability via hunting policy

Case 8 – impact of information delay:

In Case 7, the small-amplitude fluctuations in the prey population indicate that negative feedbacks and delays are still interacting. But since there is no delay in measuring the populations and turning hunting on/off in the Case 7 model, the only remaining delays from a hunting perspective are those inherent in the population stocks. These stocks are part of the negative feedback loops. To the extent that additional delays are introduced in the hunting loops, including information delays and response delays, more significant oscillations could be expected.

I have verified this using the model in Figure 11, where there is an added information delay in perceiving the prey population (based on discussion in Sterman pp. 428-429). As the adjustment time is increased gradually from one month to several months, the prey population oscillates with higher and higher amplitudes! Figure 12 shows the results for an adjustment time of 6 months, just to exaggerate the effect of information delay.

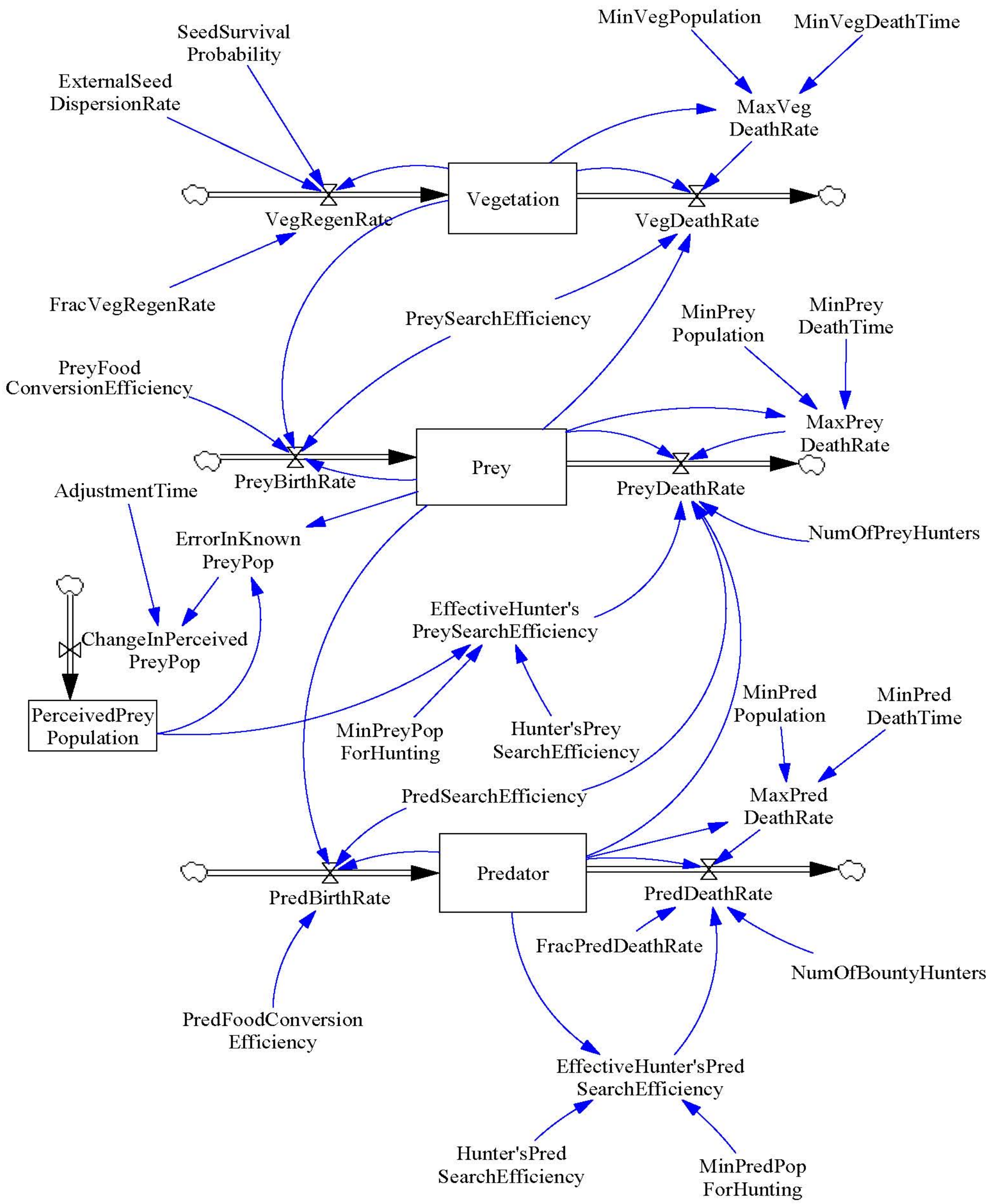


Figure 11: Case 8 model – impact of information delay

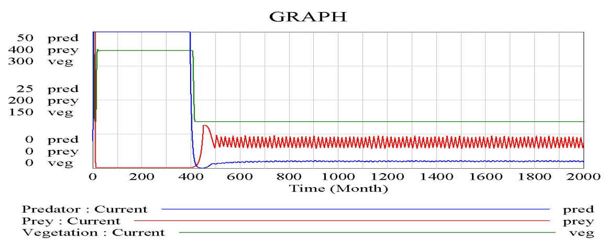


Figure 12: Case 8 results

Conclusions

Extension of the basic Lotka-Volterra model to a 3-species system results in neutrally-stable coupled oscillations of all three populations. This is expected behavior for this type of a simple population model, and to an extent reflects real-life population dynamics as well as seen in the ecology literature. What is interesting, however, is that the oscillations persist for a wide range of enhancements to the model. Scenarios such as increased initial predator population, bounty hunting of predators, prey (deer) hunting, etc., all lead to different series of neutrally-stable cycles. Cases 1-4 illustrate these scenarios.

Oscillations go away when all the predators die off for some reason while the prey population is also hunted at some fixed level. One way this can happen is by increased hunting of the prey population, which then decreases the food available for predators. But this allows only the prey (deer) population and vegetation to stabilize, without any predators remaining in the area. Case 5 illustrates this scenario. However, this is not a satisfying solution because we have entirely lost one of the species from the ecosystem.

A study of what happens due to crowding (i.e., limited space) in Case 6 suggests that limiting both prey and predator populations, regardless of food supply, could take the populations close to stable values. Oscillations still remain, but are very low amplitude. This is a natural, endogenous solution (without any hunting by humans or other human policies) that probably does occur in real life, because some real populations do reach such near-stability.

Based on this, I developed a policy scenario in Case 7, where hunting is allowed only when a population exceeds a certain minimum value. It turned out that it was sufficient to apply this only to the prey population in my model, with predators hunted at all times but at hunting levels that don't drive them to extinction. By adjusting the minimum prey population – which is essentially the **policy lever** in this case – I found that it is possible to choose a value for this parameter which results in stable populations.

Finally, Case 8 shows the definitive link between delays and oscillations in negative feedback systems. Any hunting policy developed along the lines of Case 7 will be effective only if information and response delays are minimized on the human side of the overall system.

This project has pushed me to consider feedbacks and model calibration very carefully. It was also necessary to run a large number of simulations, with different model extensions and parameter changes, to really understand the population dynamics. In the end, I was able to look at the insight provided by the endogenous behavior of a crowded system and formulate a policy which would achieve a similar result – in fact, a better result. I was also able to extract a fundamental lesson about the effects of material and information delays in systems with negative feedbacks.

Attachment 1: Vensim Model Equations for Case 1

```
(01)    ExternalSeedDispersionRate=
                300
        Units: veg/Month
(02)    FINAL TIME  = 1500
        Units: Month
 (03)   FracPredDeathRate=
                0.1
        Units: 1/Month
(04)    FracVegRegenRate=
                0.25
        Units: 1/Month
(05)    INITIAL TIME  = 0
        Units: Month
 (06)   MaxPredDeathRate=
                IF THEN ELSE(Predator >= MinPredPopulation, (Predator - MinPredPopulation
        )/MinPredDeathTime, 0)
        Units: pred/Month
(07)    MaxPreyDeathRate=
                IF THEN ELSE(Prey >= MinPreyPopulation, (Prey -
MinPreyPopulation)/MinPreyDeathTime
        , 0)
        Units: prey/Month
(08)    MaxVegDeathRate=
                IF THEN ELSE(Vegetation >= MinVegPopulation, (Vegetation -
MinVegPopulation
        )/MinVegDeathTime, 0)
        Units: veg/Month
(09)    MinPredDeathTime=
                1
        Units: Month
(10)    MinPredPopulation=
                0
        Units: pred
(11)    MinPreyDeathTime=
                1
        Units: Month
(12)    MinPreyPopulation=
```

2
Units: prey
(13) MinVegDeathTime=
1
Units: Month
(14) MinVegPopulation=
2
Units: veg

(15) Predator= INTEG (
PredBirthRate-PredDeathRate,
10)
Units: pred
(16) PredBirthRate=
PredFoodConversionEfficiency * PredSearchEfficiency * Prey * Predator
Units: pred/Month

(17) PredDeathRate=
MIN(FracPredDeathRate * Predator, MaxPredDeathRate)
Units: pred/Month

(18) PredFoodConversionEfficiency=
0.1
Units: pred/prey

(19) PredSearchEfficiency=
0.02
Units: 1/pred/Month

(20) Prey= INTEG (
PreyBirthRate-PreyDeathRate,
500)
Units: prey
(21) PreyBirthRate=
PreyFoodConversionEfficiency * PreySearchEfficiency * Vegetation * Prey
Units: prey/Month
(22) PreyDeathRate=
MIN(PredSearchEfficiency * Predator * Prey, MaxPreyDeathRate)
Units: prey/Month
(23) PreyFoodConversionEfficiency=
0.005
Units: prey/veg
(24) PreySearchEfficiency=
0.2
Units: 1/prey/Month
(25) SAVEPER =
TIME STEP
Units: Month [0,?]
(26) SeedSurvivalProbability=
0.25
Units: Dmnl
(27) TIME STEP = 1
Units: Month [0,?]
(28) VegDeathRate=
MIN(PreySearchEfficiency * Prey * Vegetation, MaxVegDeathRate)
Units: veg/Month

(29) Vegetation= INTEG (
+VegRegenRate-VegDeathRate,
100000)
Units: veg
(30) VegRegenRate=
FracVegRegenRate * Vegetation + ExternalSeedDispersionRate *
SeedSurvivalProbability
Units: veg/Month

Attachment 2: Vensim Model Equations for Case 4

(01) BountyHunters=
10
Units: hunter
(02) ExternalSeedDispersionRate=
300
Units: veg/Month
(03) FINAL TIME = 10000
Units: Month
(04) FracPredDeathRate=
0.1
Units: 1/Month
(05) FracVegRegenRate=
0.25
Units: 1/Month
(06) Hunter'sPredSearchEfficiency=
0.005
Units: 1/hunter/Month
(07) Hunter'sPreySearchEfficiency=
0.01
Units: 1/hunter/Month
(08) INITIAL TIME = 0
Units: Month
(09) MaxPredDeathRate=
IF THEN ELSE(Predator >= MinPredPopulation, (Predator - MinPredPopulation
)/MinPredDeathTime, 0)
Units: pred/Month
(10) MaxPreyDeathRate=
IF THEN ELSE(Prey >= MinPreyPopulation, (Prey -
MinPreyPopulation)/MinPreyDeathTime
, 0)
Units: prey/Month
(11) MaxVegDeathRate=
IF THEN ELSE(Vegetation >= MinVegPopulation, (Vegetation -
MinVegPopulation
)/MinVegDeathTime, 0)
Units: veg/Month
(12) MinPredDeathTime=
1
Units: Month
(13) MinPredPopulation=
0
Units: pred
(14) MinPreyDeathTime=

1
Units: Month
(15) MinPreyPopulation=
2
Units: prey
(16) MinVegDeathTime=
1
Units: Month
(17) MinVegPopulation=
2
Units: veg
(18) Predator= INTEG (
PredBirthRate-PredDeathRate,
10)
Units: pred
(19) PredBirthRate=
PredFoodConversionEfficiency * PredSearchEfficiency * Prey * Predator
Units: pred/Month
(20) PredDeathRate=
MIN(FracPredDeathRate * Predator + Hunter'sPredSearchEfficiency * BountyHunters
* Predator, MaxPredDeathRate)
Units: pred/Month
(21) PredFoodConversionEfficiency=
0.1
Units: pred/prey
(22) PredSearchEfficiency=
0.02
Units: 1/pred/Month
(23) Prey= INTEG (
PreyBirthRate-PreyDeathRate,
500)
Units: prey
(24) PreyBirthRate=
PreyFoodConversionEfficiency * PreySearchEfficiency * Vegetation * Prey
Units: prey/Month
(25) PreyDeathRate=
MIN(PredSearchEfficiency * Predator * Prey + Hunter'sPreySearchEfficiency
* PreyHunters * Prey, MaxPreyDeathRate)
Units: prey/Month
(26) PreyFoodConversionEfficiency=
0.005
Units: prey/veg
(27) PreyHunters=
10
Units: hunter
(28) PreySearchEfficiency=
0.2
Units: 1/prey/Month
(29) SAVEPER =
TIME STEP
Units: Month [0,?]
(30) SeedSurvivalProbability=
0.25
Units: Dmnl
(31) TIME STEP = 1

Units: Month [0,?]
(32) VegDeathRate=
MIN(PreySearchEfficiency * Prey * Vegetation, MaxVegDeathRate)
Units: veg/Month
(33) Vegetation= INTEG (
+VegRegenRate-VegDeathRate,
100000)
Units: veg
(34) VegRegenRate=
FracVegRegenRate * Vegetation + ExternalSeedDispersionRate *
SeedSurvivalProbability
Units: veg/Month

Attachment 3: Vensim Model Equations for Case 7

(01) EffectiveHunter'sPredSearchEfficiency=
IF THEN ELSE(Predator > MinPredPopForHunting,
Hunter'sPredSearchEfficiency
, 0)
Units: 1/hunter/Month
(02) EffectiveHunter'sPreySearchEfficiency=
IF THEN ELSE(Prey > MinPreyPopForHunting, Hunter'sPreySearchEfficiency, 0
)
Units: 1/hunter/Month
(03) ExternalSeedDispersionRate=
300
Units: veg/Month
(04) FINAL TIME = 5000
Units: Month
(05) FracPredDeathRate=
0.1
Units: 1/Month
(06) FracVegRegenRate=
0.25
Units: 1/Month
(07) Hunter'sPredSearchEfficiency=
0.005
Units: 1/hunter/Month
(08) Hunter'sPreySearchEfficiency=
0.01
Units: 1/hunter/Month
(09) INITIAL TIME = 0
Units: Month
(10) MaxPredDeathRate=
IF THEN ELSE(Predator >= MinPredPopulation, (Predator - MinPredPopulation
)/MinPredDeathTime, 0)
Units: pred/Month
(11) MaxPreyDeathRate=
IF THEN ELSE(Prey >= MinPreyPopulation, (Prey -
MinPreyPopulation)/MinPreyDeathTime
, 0)
Units: prey/Month
(12) MaxVegDeathRate=

IF THEN ELSE(Vegetation >= MinVegPopulation, (Vegetation - MinVegPopulation
)/MinVegDeathTime, 0)
Units: veg/Month
(13) MinPredDeathTime=
1
Units: Month
(14) MinPredPopForHunting=
0
Units: pred
(15) MinPredPopulation=
0
Units: pred
(16) MinPreyDeathTime=
1
Units: Month
(17) MinPreyPopForHunting=
75
Units: prey
(18) MinPreyPopulation=
2
Units: prey
(19) MinVegDeathTime=
1
Units: Month
(20) MinVegPopulation=
2
Units: veg
(21) NumOfBountyHunters=
10
Units: hunter
(22) NumOfPreyHunters=
10
Units: hunter
(23) Predator= INTEG (
PredBirthRate-PredDeathRate,
10)
Units: pred
(24) PredBirthRate=
PredFoodConversionEfficiency * PredSearchEfficiency * Prey * Predator
Units: pred/Month
(25) PredDeathRate=
MIN(FracPredDeathRate * Predator + EffectiveHunter'sPredSearchEfficiency * NumOfBountyHunters * Predator, MaxPredDeathRate)
Units: pred/Month
(26) PredFoodConversionEfficiency=
0.1
Units: pred/prey
(27) PredSearchEfficiency=
0.02
Units: 1/pred/Month
(28) Prey= INTEG (
PreyBirthRate-PreyDeathRate,
500)
Units: prey
(29) PreyBirthRate=

PreyFoodConversionEfficiency * PreySearchEfficiency * Vegetation * Prey
Units: prey/Month

(30) PreyDeathRate=
MIN(PredSearchEfficiency * Predator * Prey + EffectiveHunter'sPreySearchEfficiency
* NumOfPreyHunters * Prey, MaxPreyDeathRate)
Units: prey/Month

(31) PreyFoodConversionEfficiency=
0.005
Units: prey/veg

(32) PreySearchEfficiency=
0.2
Units: 1/prey/Month

(33) SAVEPER =
TIME STEP
Units: Month [0,?]

(34) SeedSurvivalProbability=
0.25
Units: Dmnl

(35) TIME STEP = 1
Units: Month [0,?]

(36) VegDeathRate=
MIN(PreySearchEfficiency * Prey * Vegetation, MaxVegDeathRate)
Units: veg/Month

(37) Vegetation= INTEG (
+VegRegenRate-VegDeathRate,
100000)
Units: veg

(38) VegRegenRate=
FracVegRegenRate * Vegetation + ExternalSeedDispersionRate * SeedSurvivalProbability
Units: veg/Month