Systems Cell Biology
University of California, Irvine
Prof. Lee Bardwell

Network Motif Modeling Assignments

For Bardwell Homework 2

Note: Most figures are from Shoval & Alon (2010), SnapShot: Network Motifs Cell 143(2):326.e1-2 (2010) DOI 10.1016/j.cell.2010.09.050

AUTOREGULATION

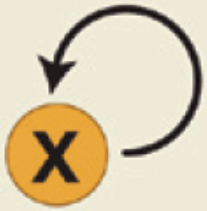
Negative

Speeds response time

Reduces cell-cell variability of X concentration

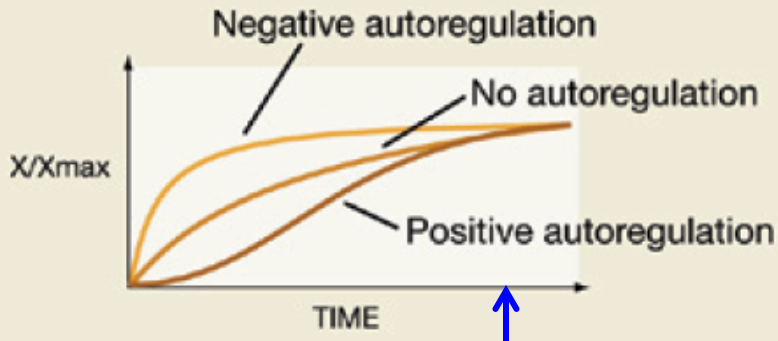
Makes input function (response to signal) more graded

Positive



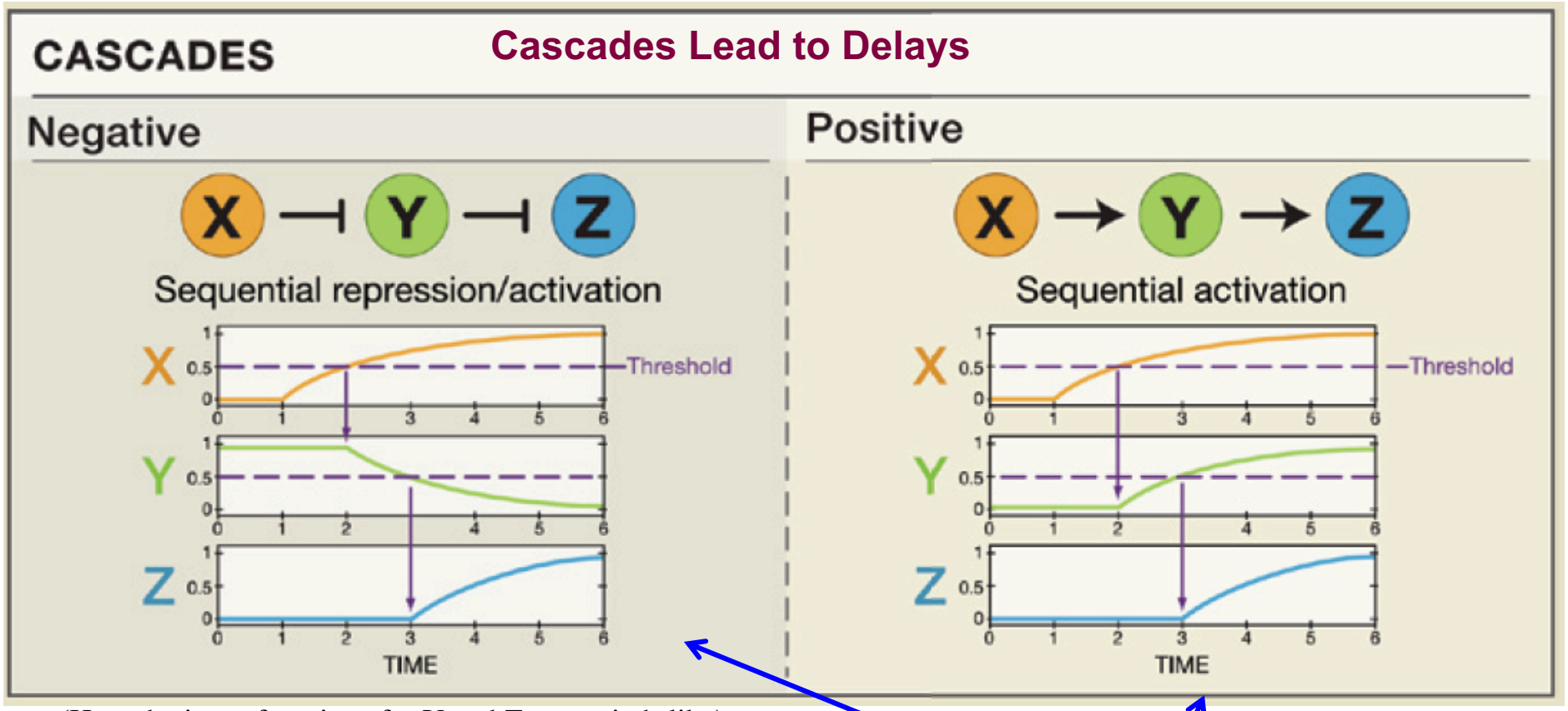
Slows response time

Possible bistability



1. Model negative, positive and no autoregulation and put all on the same graph so it looks like this. Note that all three reach the same steady state.

Hint: refer to the lecture on autoregulation to help you pick parameters so that all three graphs reach (within 1% is fine) the same steady state.



(Here the input functions for Y and Z are switch-like)

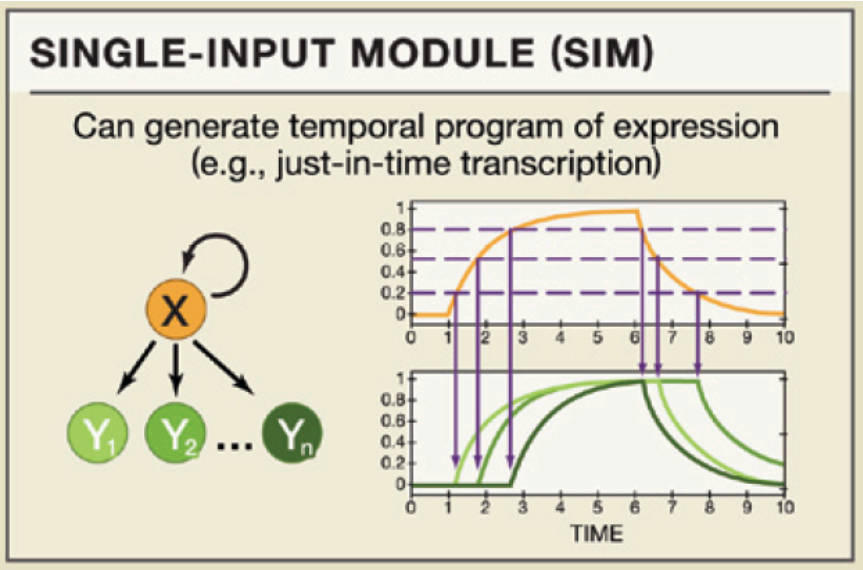
2A. Model one or the other of these cascades, and get graphs that look like this.

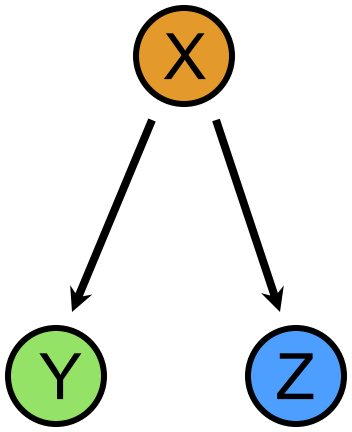
This will require high Hill numbers to get the sharp thresholds.

2B. What do your graphs look like if all Hill numbers are 1?

To do a valid comparison, get your graphs in 2B to reach ~ the same steady state as in 2A.

In the graphs above, nothing happens for the first minute. You don't need to do this; your simulation (and graphs) can begin where X starts rising. Thus, you won't need to use a Piecewise function for this problem





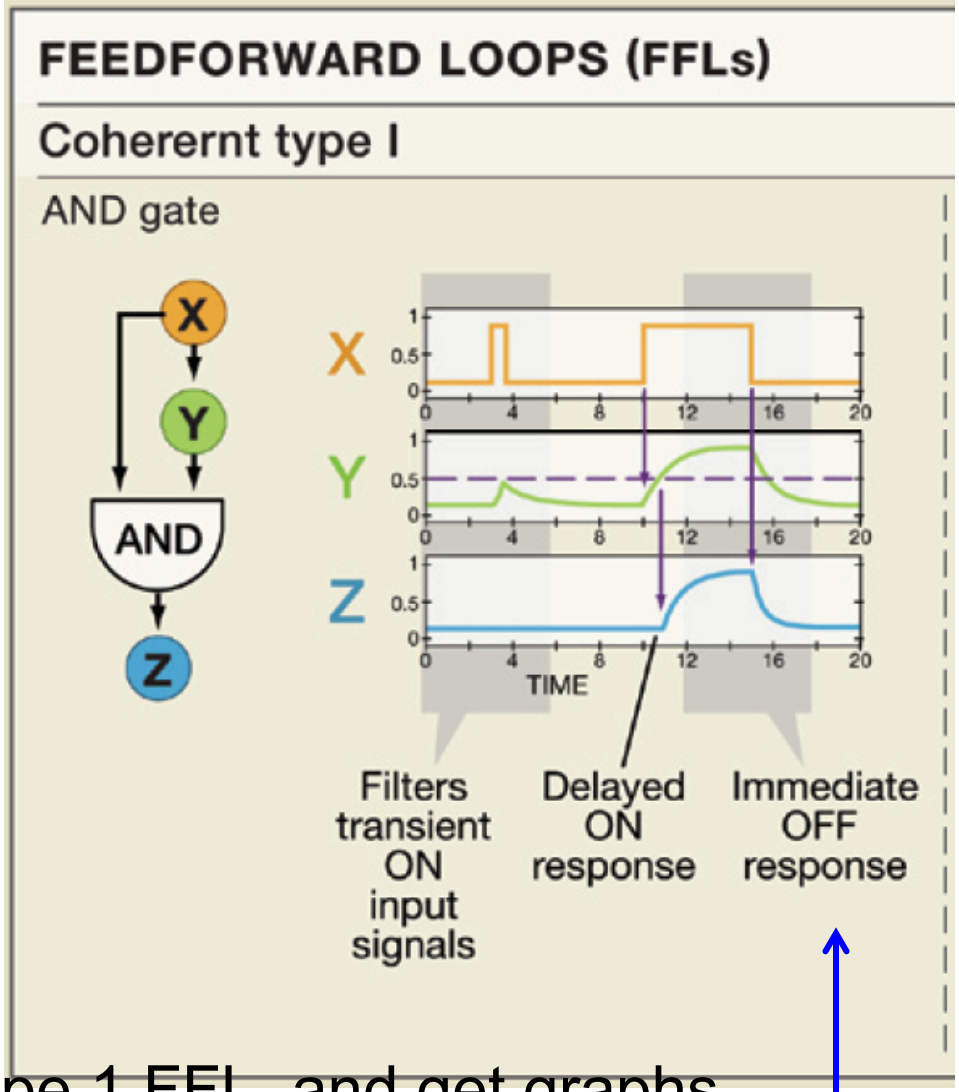
3.

←Model the system shown here. It is simpler then the one shown above. However, try to get your output more or less like the plots shown in the figure. Adjust the parameters so that X will turn on Y 1 minute before it turns on Z. Then have X shut off (use piecewise fxn) and have Y decay quickly and Z decay more slowly.

Hint: For X, you will need to use the methods shown in the Pulsed_Simple_Gene_Regulation notebook to generate a pulse in NDSolve. Your equation for x will look something like:

x'[t] == pulse[t, xmax] – d x[t]

But unlike in the notebook, you will not need d to be huge

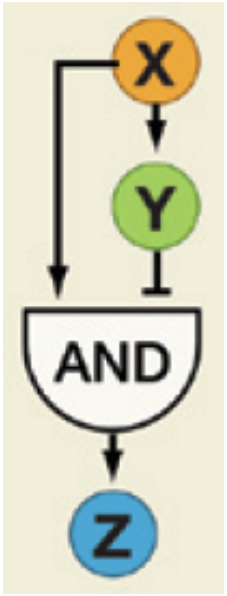


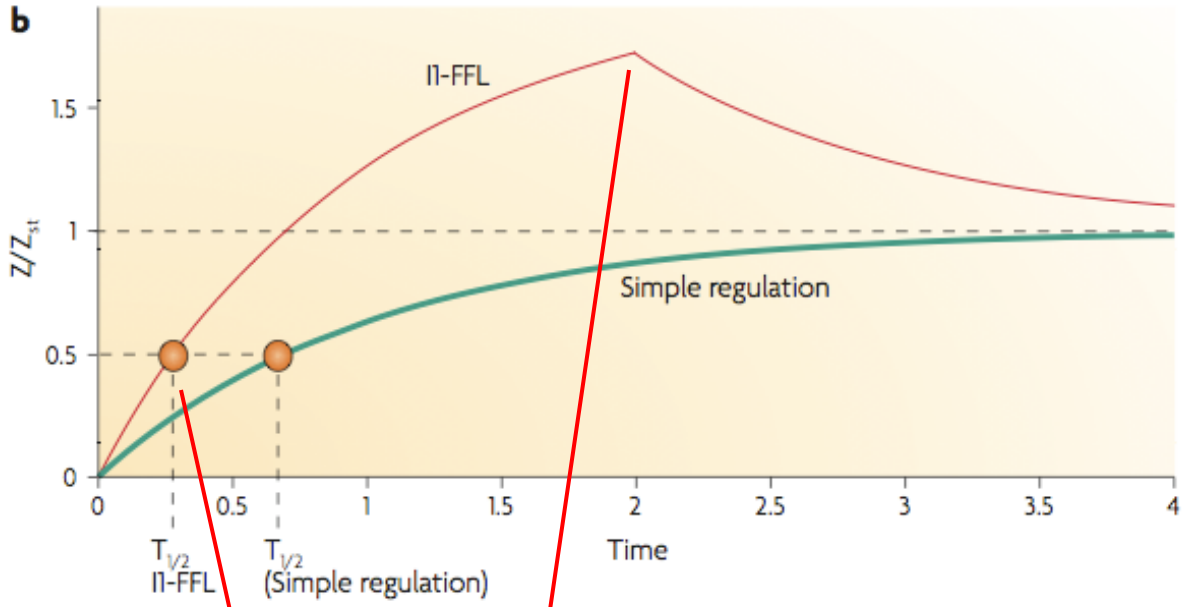
4. Model a coherent type 1 FFL, and get graphs that look like this

Note the pulsed nature of the input TF X. You will only model Y and Z as genes that get transcribed/translated into proteins. X is modeled as per "pulsed simple gene regulation" notebook

Incoherent FFL

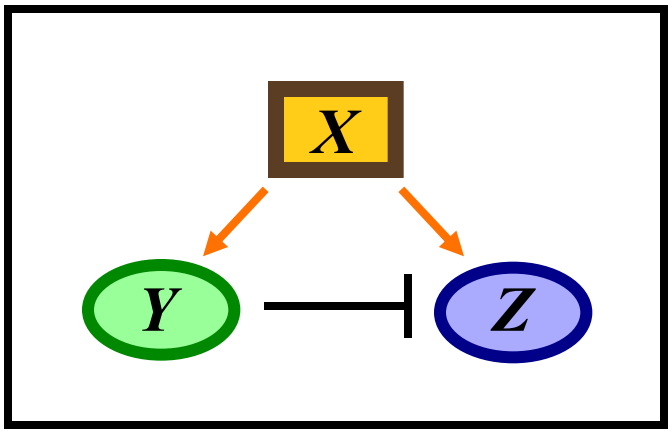
Incoherent type 1



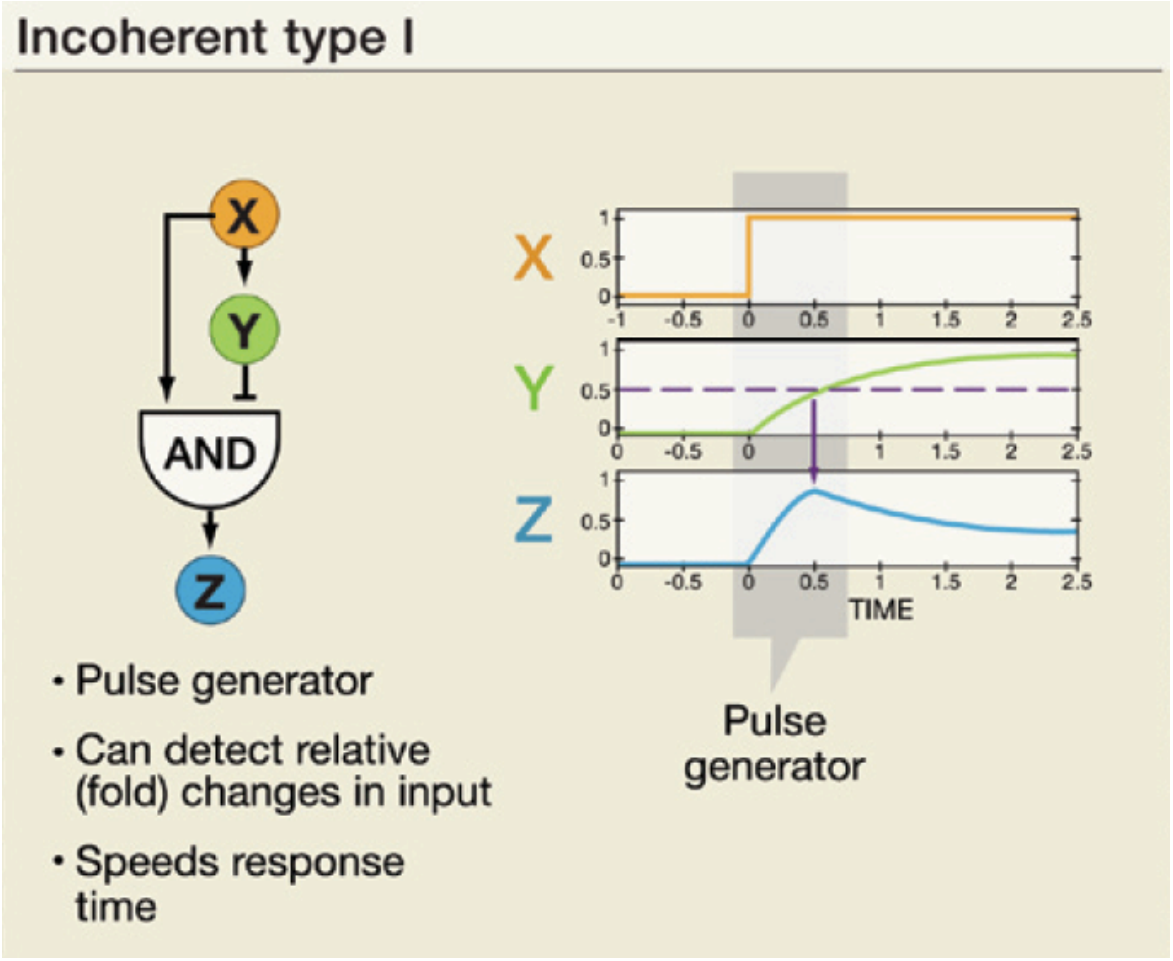


Pulse generation

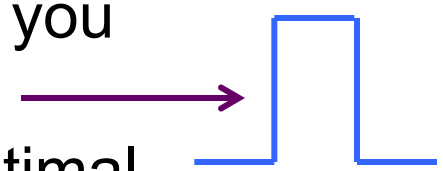
Response acceleration



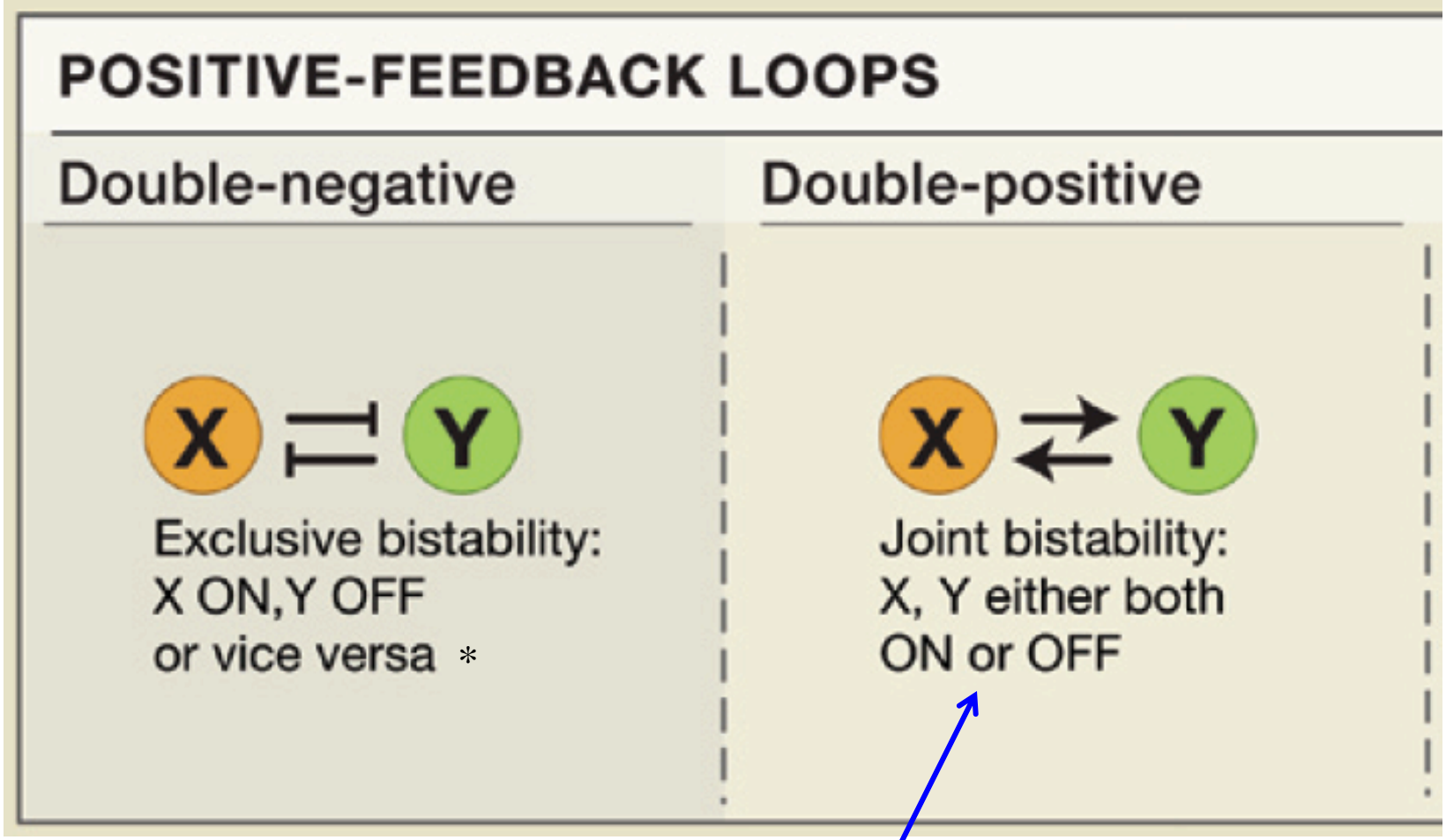
5A. Model an incoherent type 1 FFL, and get graphs that look like this (see also next page). DO NOT USE A PULSED INPUT FOR X, rather, treat X as a constant.



5B. By messing with parameters, (but working within the framwork of the I-FFL type 1), can you make the "pulse" of Z look more pulse-like?



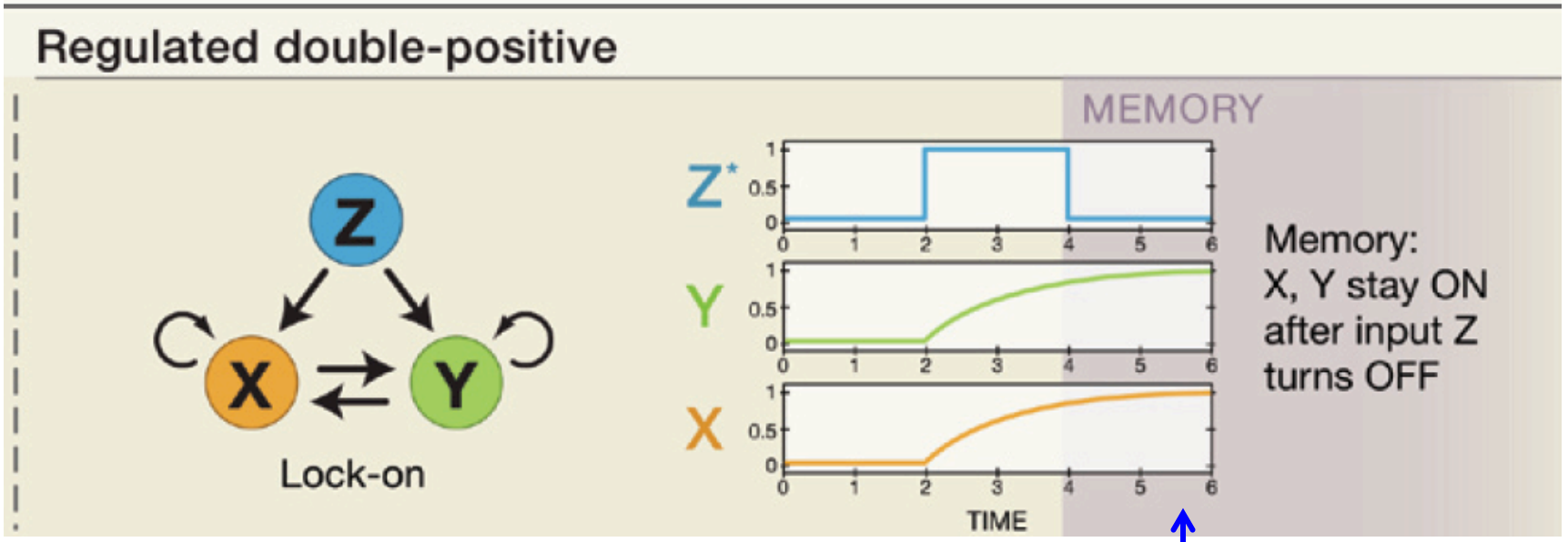
(Note – I don't think it's possible to get an optimal solution, but do your best)



6A. Model a double-positive feedback loop. Demonstrate joint bistability (that is, both off or both on depending on the starting amount of X, assuming Y[0]=0)

6B. Is joint OFF stable or unstable? Under what conditions?

The type of answers I am expecting for 5B are examples taken from your model. I am not expecting a full-blown mathematical analysis. The autoregulation demo notebook shows how to do a dose response curve; this is a useful approach here. Also, don't give me a trivial answer where joint off is stable because the motif can never be switched on.

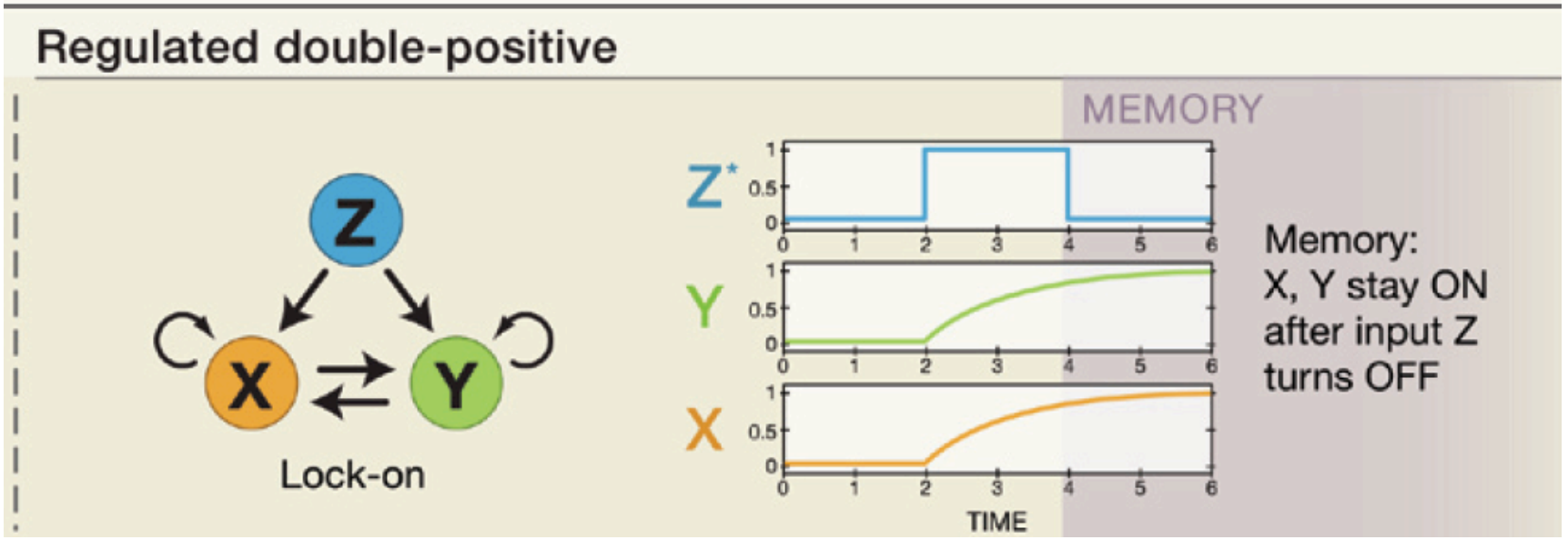


7. Model a regulated-double positive feedback loop and get graphs that look like this

The above model has 6 arrows. Some of these are not essential, and you may drop them. Doing this will simplify your input functions. Just make it clear which ones you dropped. You also need to figure what type of logic you want to use (AND, OR, etc, and for what TF combinations).

Note 1: *Your **Z** input should start at 0.05, not 0. This will show that the "off" steady states for X and Y are stable. The off steady-states for X and Y need to be near 0, but not exaclty = 0.

Note 2: Each distinct phase of the pulsed input should last at least 10 half-lives for the longest-lived protein in your model. That way we know the system has essentially come to steady state. Here the pulse times are two minutes, so half lives need to be < 12 seconds (or rescale time appropriately, e.g. if the decay rates are set to 1 min^{-1}, the pulse phases should last ≥ 10 minutes each)



As you start deleting or weakening arrows, you will find that you can make several distinct circuits that can all reproduce the above graphs, for example:

1.Activation detector with memory: X and Y will lock if either one of them, or both, is activated by Z.

2.X does not lock on unless it gets an input from Z, but Y can lock on if X locks on, regardless of whether or not Y gets an input from Z

3.Coincidence detector with memory (more difficult): X and Y will lock on only if they both get an input from Z. That is, if you delete the arrow from Z to X (or the arrow from Z to Y), the "lock on" no longer works.

(You only have to do one of these)