Intro to Dr. Krauss’ Block Diagram Code Generation

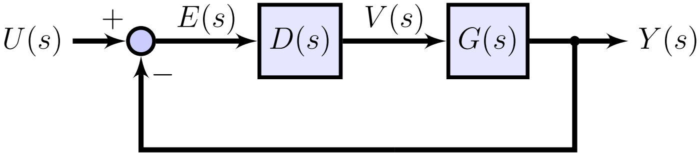
EGR 345

GVSU

Background

- I have been working on how to help students perform feedback control experiments using Arduino and Raspberry Pi for a decade or so.
    - this is year 10 of doing it in 345 at GVSU
- this is an active area of pedagogical research for me

How do we convert this to C code?



- there is an implied `for` loop
- the code must execute every 2 milliseconds
- there are a lot of details you have to get right
- students can easily get bogged down in the weeds

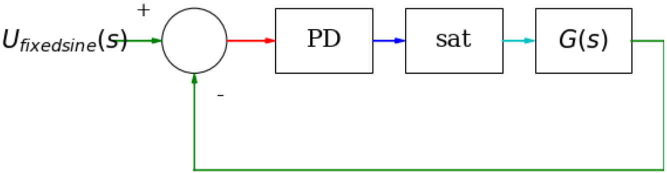
Goal

I want 345 students to focus on learning dynamic systems and feedback control content without getting bogged down in low-level coding details.

- I don’t want you writing Arduino code “by hand”.

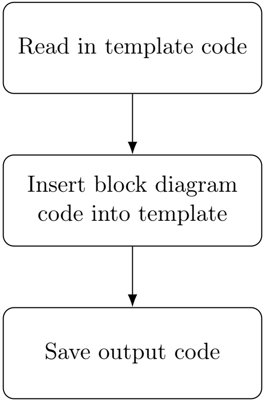
Block Diagram Code Generation

- you create a block diagram model of your system:



- my Python library auto-generates the C code for Arduino or Raspberry Pi

Code Generation Process



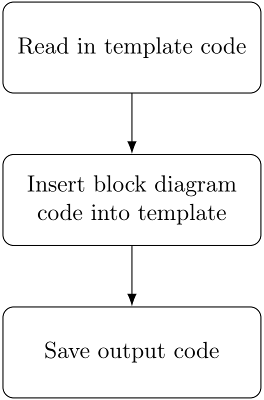
Read in template code

Insert block diagram code into template

Save output code

Code Generation Gotchas

- template code is read-only
    - do not edit or overwrite it
- template and output **must** be in different folders
- template code is not meant to be compiled



Read in template code

Insert block diagram code into template

Save output code

Block Diagram Challenges

- block diagram creation needs to be learned
    - there are steps to be followed in a specific order
- block diagram coding can feel restrictive and take some getting used to

Block Placement

- I know, you want to click and drag the blocks
  - not yet, maybe some day
- the block diagram is a graph drawn with `matplotlib`, the same Python library we use for plotting
- you tell my library how the blocks are arranged and it generates the block diagram for you

Block Placement Continued

- there are two types of placements: absolute and relative
- you must have one absolute block
- the first block you create will be absolute at `(0,0)`
    - if you ever delete this block, you will make the software mad
- making the rest of the blocks relative makes it easier if you have to change things
    - but deleting a block can mess up blocks that were relative to it
- **work around:** some students have found it helpful to sketch the block diagram ahead of time to guide them when creating it in my software

Other Issues with My Software

- it still has some bugs
    - I am not a software engineer
    - I am just one person
- editing block diagrams is a work in progress
    - when I created the software, I didn’t practice making mistakes and editing things
- **bottom line:** it is still *way* better than writing Arduino code by hand