Problem Wk.5.5.1: Modeling the Conrolled/Sensor

Read the handout for Homework Assignment 2.

System Function

Enter the system function $\frac{V_c}{E}$ for the combined sensor/controller system below. Enter the numerator and denominator separately. You must enter them both (if the denominator is empty, enter a "`1`". The numerator and denominator do not need to exactly match our solution, but the ratio must be the same.

You can enter algebraic expressions in "standard" notation; the checker will try to turn your input into a valid Python expression. An example answer is something like:

```
2 (x + 3)
```

If you're having trouble with syntax, you can always type a legal Python expression, fully parenthesized and with all the operators, including `*` and `**`.

A few extra quick notes about syntax:

- These expressions are case-sensitive. `A` is not the same thing as `a`. Remember that, by convention, signals have capital letter names, and samples have lowercase letter names.
- To enter subscripts (e.g. k_s), use an underscore between the variable name and the subscript (e.g. `k_s`).
- To enter greek letters, just type the name of the letter. Note the difference between capital letters and lowercase letters. E.g., `Delta` becomes Δ, whereas `delta` becomes δ.
- Use a capital `R` for the delay operator R.

As an example, here is a list of some of the variables defined in the handout, along with the ASCII representation you should enter to represent each variable in the tutor:

Variable	ASCII
k_s	k_s
k_m	k_m
k_b	k_b
r_m	r_m
k_c	k_c
T	T
Ω_h	Omega_h
Θ_h	Theta_h

Numerator:

Denominator:

Block Diagram

Upload a PDF file containing your block diagram for this system. Please double-check that your file is a valid PDF before uploading. You will be able to check that the file is correctly uploaded.

Browse... Upload File

Code

Enter your code for `controllerAndSensorModel` below. The global variables defined in `hw2Work.py` can be used in your definition; do not redefine them here. Your code may use functions from the `sf` module, such as `sf.Gain(...)`; no `import` statements should be needed.

```
def controllerAndSensorModel(k_c):
    pass #your code here
```

6.01SC Introduction to Electrical Engineering and Computer Science

Spring 2011