LECTURE OUTLINE
TODAY, TOMORROW

Review of numerical analysis, control theory, and software engineering with applications to beam physics and control

Formal Theory

We review the following subjects (in roughly that order):

- Linear Algebra
- Control Theory
- Linear Beam Optics
- Software Engineering

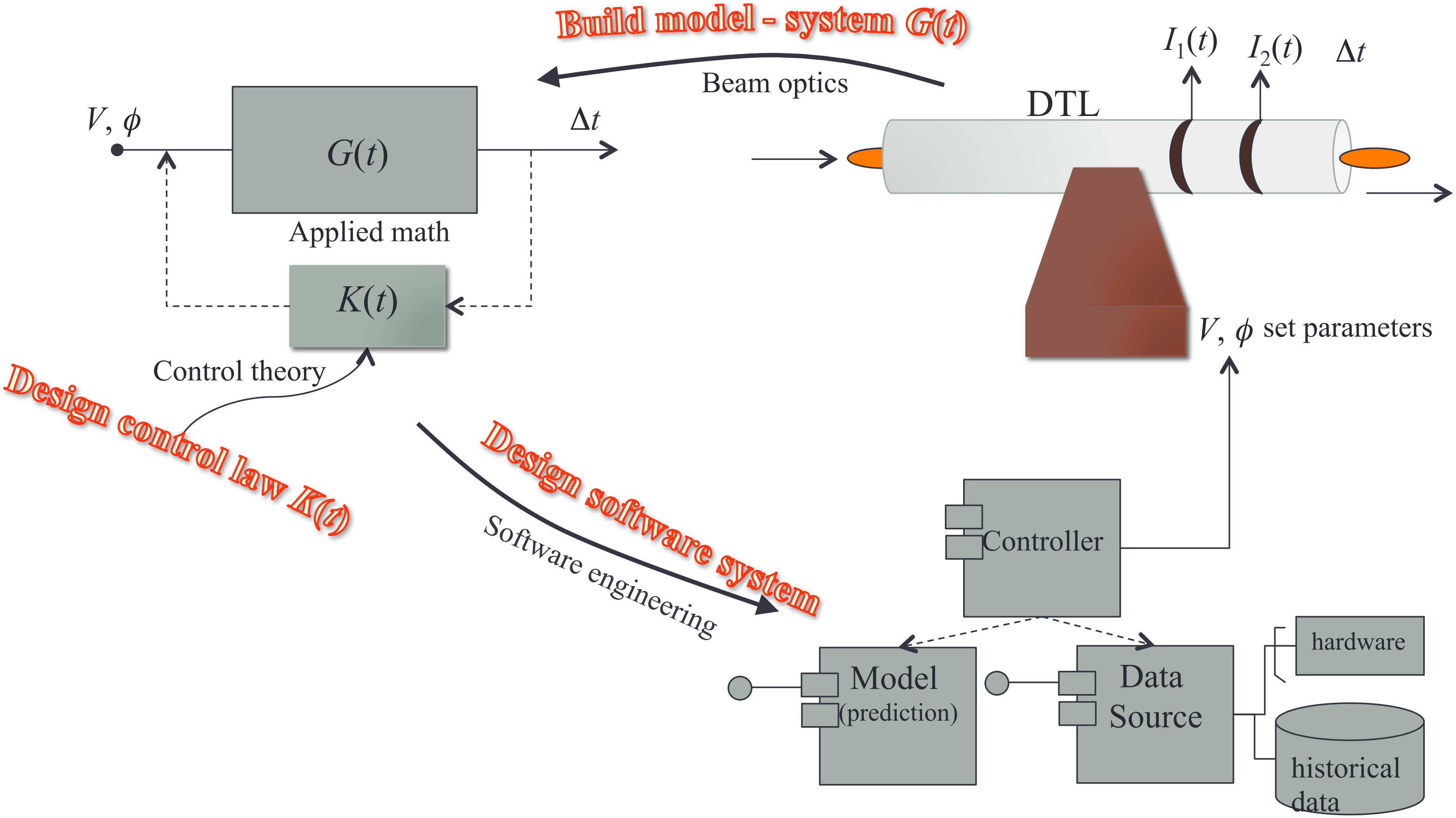
High level application development borrows heavily from theses areas (formally and informally)

MOTIVATION: We wish to abstract our application from the problem domain in order to apply standard solution techniques without regard to the original situation (avoid getting lost in detail).

Descriptions

- Linear Algebra
  - Provides general techniques for solving systems abstracted from problem domain
- Control Theory
  - Formalizes the notion of control (of a system), provides techniques for designing controllers and control laws
- Linear Beam Optics
  - Basic models for accelerator systems allows us to create mathematical abstractions (models) of our problem domain
- Software Engineering
  - Once the problem domain is modeled, a controller designed, a solution technique chosen, then a suitable software system must be designed to implement all of the above.

Example: Klystron Phase Settings



Motivation

Abstract the problem into a domain where a wealth solution techniques exist

- Utilize as many tools in the chest as possible

- For example

  How much salt is contained in a 100 gallon tank initially filled with fresh water being filled at a rate 2 gallons per minute with a 10% brine solution and overflowing at the same rate? Assume the overflow is fully mixed.

Analysis

$$V\dot{n}(t) = rn_0 - rn(t)$$

V = volume of tank

r = fill/overflow rate

$n(t)$ = salt concentration

n_0 = fill salt concentration

MODEL - abstraction

Calculus – tool

$$n(t) = n_0\left(1 - e^{-\frac{r}{V}t}\right)$$

SOLUTION

Outline

- Today (Tuesday)
  - Linear Algebra
  - Control Theory and Linear Systems
  - Linear Beam Optics
  - (Primer on the Open XAL Online Model)
- Tomorrow - Software Engineering
  - The Software Process
  - Software Architecture and Design
  - The XAL Solver

Architecture

- Although building an application from the “physics perspective” might seem appealing, it is typically not.
  - Brittle
  - Hard to maintain
  - Hard to upgrade
  - Difficult to understand and document
- Although the treatment is rather formal, we provide practical instruction.
  - The format I call “Theorem-Example”