Lecture 00.02 Embedded control of mechanical systems

feedback control

Feedback control is a powerful idea: a desired system state is compared to a measurement of its actual state and some actuation is applied such that the difference between the desired and the actual state (the error) is minimized. This concept has been realized in many ways since around 300 BC, when the Greeks used a float valve to regulate the flow of water to a relatively constant rate, allowing them to measure time.

The 20th Century, especially its latter half, saw a mathematization and vast improvement of control. Electronic circuits began to perform the key "logic" function: for example, "if a is measured, then do b." Previously, the mechanical organization of the system had to to perform the logic, as in the flow-regulation example illustrated in Figure 00.2.

As circuits have evolved into increasingly powerful microprocessors connected to sensors and actuators, the complexity of implementable logic has grown drastically. But one constraint is persistent: decisions about how the control system should respond with its actuators, given the system's current state, must happen in *real-time*—that is, *now*! (Rather, as close to "now" as possible.) Real-time computing for control must not only be fast, but reliably so; that is, the programmer must be able to direct *timing*. Although computing power has improved steadily, real-time computing remains among the greatest challenges for control systems engineers.

real-time

timing

Most feedback control is instantiated with *embedded computers* because sensors and actuators must be nearby to increase reliability and decrease lag. Therefore, designing *controllers* (that is, embedded computers and peripherals used for control) requires an understanding of embedded

controllers

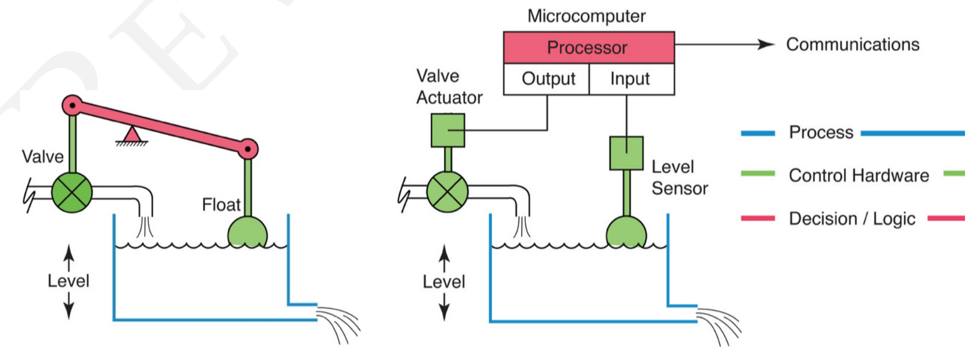

Figure 00.2: level control with (left) a mechanical and with (right) an embedded computer.

computing hardware and its programming.