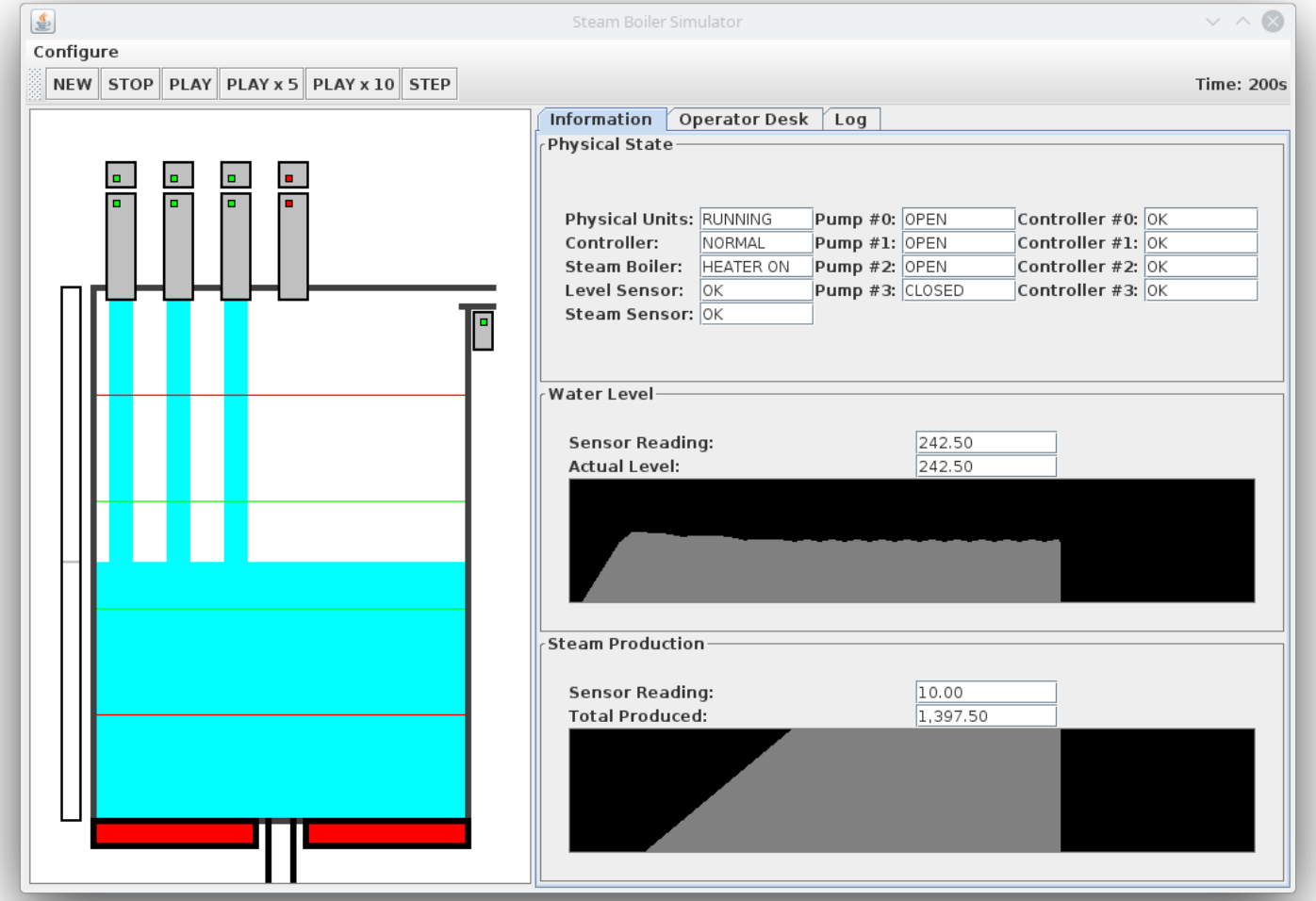
Lecture 6 — Steam Boiler Problem

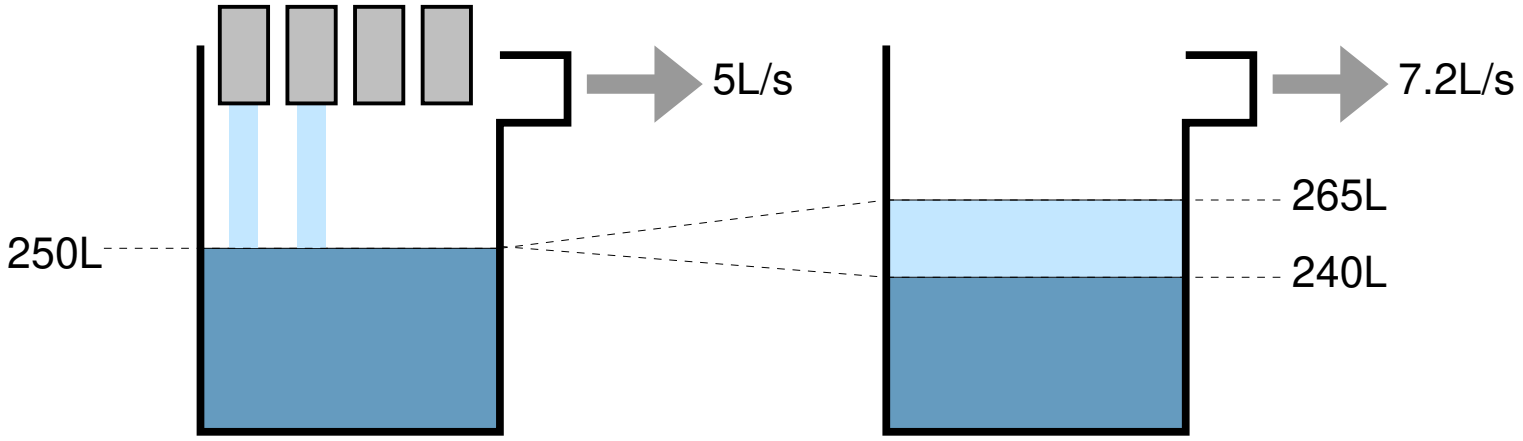
David J. Pearce

School of Engineering and Computer Science
Victoria University of Wellington

Simulation Software



Normal Mode



Normal mode is *relatively* straight forward:

- Each cycle, **determine** how many pumps to enable ...
- ... by **predicting** level of water at next clock cycle ...
- .. for **all possible** pump configurations!

Normal Mode: Example Calculation

$$\begin{aligned} L^{max} &= L + (5 \times C \times n) - (5 \times S) \\ &= 250 + (5 \times 4 \times 2) - (5 \times 5) = 265L \end{aligned} \tag{1}$$

$$\begin{aligned} L^{min} &= L + (5 \times C \times n) - (5 \times W) \\ &= 250 + (5 \times 4 \times 2) - (5 \times 10) = 240L \end{aligned} \tag{2}$$

n — number of pumps enabled (e.g. 2)

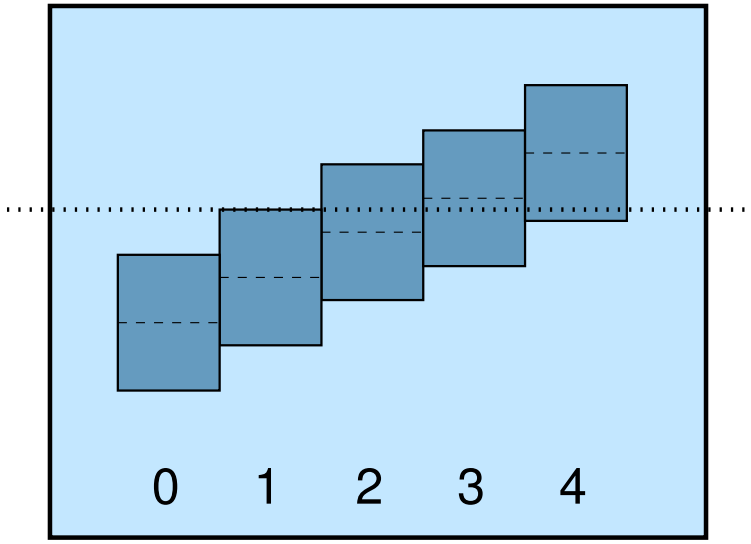
L — current water level (e.g. $250L$)

C — pump capacity (e.g. $4L/s$)

W — maximum steam rate is W (e.g. $10L/s$)

S — steam level reading (e.g. $(5L/s)$)

Normal Mode: Hitting the Target



- Simple strategy considers **midpoint** of prediction
- Some configuration **unsafe** and cannot be used
- When **no valid** configuration ...

Failure Mode

Assumption

In any given period, at most one component can fail

- Failure detection **hardest part** of assignment
- Above **assumption** critical to proper **failure analysis**
- Solution based around **predicting** what components will do next

Failure Detection

(easy) **Invalid Sensor Reading**. Negative or out-of-bounds readings *are obviously incorrect*

(easy) **Unexpected Pump Response**. If pump does not turn off or on when asked, then *has obviously failed*

(hard) **Unexpected Controller Response**. If pump and controller disagree then *something has failed* — but what?

(hard) **Unexpected Water Level**. If water level not within predicted range, then *something has failed* — but what?

(hard) **Unexpected Steam Output**. Assume steam output continues to rise, otherwise *something has failed*.

Note, the steam boiler specification states that pumps take 5s to open; however, our physical hardware does not reflect this!)

Failure analysis (part 1)

P_i	C_i	L	**Commentary**
—	—	—	Indicates physical units operating as expected.
—	—	×	Indicates water level is not within predicted range. Since everything else appears to be functioning correctly, indicates a failure of water level sensor. Observe could also indicate *valve failure* or *heating element failure*, but we are ignoring these possibilities here.
—	×	—	Controller *i* has responded incorrectly (e.g. showing water flowing when we not expected). This is *most likely* a controller failure, but could signal a pump failure. For latter, might expect to see a corroborating level signal. However, if predicted water range is large, then pump might have failed (e.g. stopped pumping) but water level remained within expected range.
—	×	×	Controller *i* has responded incorrectly (e.g. showing water not flowing when is expected to be), and water level is not within predicted range. Assuming controller response and water level *corroborate* each other, indicates a failure of pump *i*. For example, if controller indicates water is *not flowing* and water level *below* predicted levels, indicates pump *i* has failed. Otherwise, suggests a *multiple component failure* (which is ignored).

Failure analysis (part 1)

P_i	C_i	L	**Commentary**
×	—	—	Pump *i* has responded incorrectly (e.g. failed to open when requested) which indicates it has failed. In this case controller *does not corroborate* observation, suggesting pump is behaving as requested but just not responding correctly.
×	—	×	Pump *i* has responded incorrectly (e.g. failed to open when requested) which indicates it has failed. In addition, water level is not within predicted range. These responses could be consistent with a pump failure. For example, if pump is *not pumping* and water level is *below* expected levels. Otherwise, suggests a *multiple component failure* (which is ignored).
×	×	—	Pump *i* has responded incorrectly (e.g. failed to open when requested) which indicates it has failed. The *i*th controller *corroborates* observation.
×	×	×	Pump *i* has responded incorrectly (e.g. failed to open when requested) which indicates it has failed. The *i*th controller *corroborates* observation and water level *may corroborate* it. For example, if pump has failed off and water level is below prediction, then is consistent. But, if pump has failed off and water level is *above* predication, then suggests a *multiple component failure* (which is ignored).