

2.1 核心抑制：核电，轻而易举啊！

龙之研究 (<https://www.mcmod.cn/class/423.html>) 是一个邪恶的模组，凭借一己之力创造了 Minecraft 科技模组中最容易爆炸的反应堆之一。本节中我们研究神龙反应堆的一些性质。



图 2.1: 使用通用机械的纸箱可以将失控的反应堆包装起来，这样你就获得了便携核弹。(图源 B 站 UP 主 @ 沐星团子)

2.1.1 反应堆的状态空间模型

据官方代码 (<https://github.com/Draconic-Inc/Draconic-Evolution>), 该反应堆可由离散时间状态空间模型建模：状态变量包括 $x_1(t)$ /反应堆温度, $x_2(t)$ /反应堆能量缓存, $x_3(t)$ /反应堆控制场, $x_4(t)$ /燃料消耗进度；控制变量包括 $u_1(t)$ /从反应堆缓存中提取的能量, $u_2(t)$ /向反应堆控制场中输入的能量。

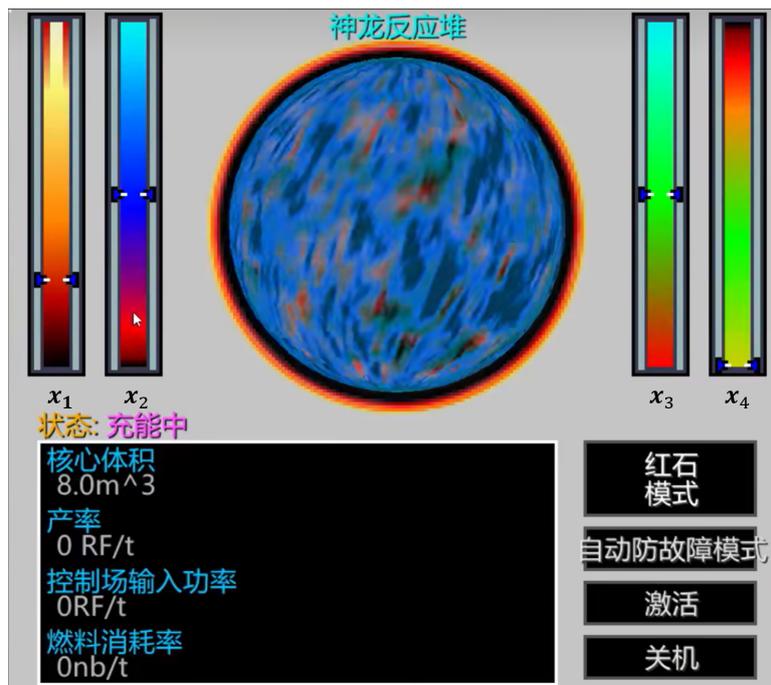


图 2.2: 在反应堆的 GUI 中，你可以读取四个状态变量（图源 B 站 UP 主 @ 安逸菌）

简便起见，在写出状态空间模型时使用以下辅助变量：

- 能量饱和度

$$\rho_{sat}(x_2) = \text{clamp}\left(\frac{x_2}{E_{max}}, [0, 1]\right)$$

- 燃料转化水平

$$\lambda_{conv}(x_4) = \left(\frac{x_4}{F_{total}} \times 1.3\right) - 0.3$$

- 温度乘数

$$\eta(x_1) = \begin{cases} 1 + (x_1 - 8000)^2 \times 2.5 \times 10^{-6} & x_1 > 8000 \\ 1 & 2000 < x_1 \leq 8000 \\ \frac{x_1 - 1000}{1000} & 1000 < x_1 \leq 2000 \\ 0 & x_1 \leq 1000 \end{cases}$$

- 基础发电量

$$G_{base} = \frac{E_{max}}{1000} \times 15$$

- 反向饱和度

$$N_{sat}(x_2) = (1 - \rho_{sat}(x_2)) \times 99$$

- 升温阻力系数

$$T_{50}(x_1) = \min\left(\frac{x_1}{T_{max}} \times 50, 99\right)$$

- 升温部分

$$T_{rise}(x_2) = \frac{N_{sat}^3(x_2)}{100 - N_{sat}(x_2)} + 444.7$$

- 降温部分

$$T_{resist}(x_1) = \frac{T_{50}^4(x_1)}{100 - T_{50}(x_1)}$$

- 能量产生速率

$$G_{rate}(x_2, x_4) = (1 - \rho_{sat}(x_2)) \cdot G_{base} \cdot (1 + 2\lambda_{conv}(x_4))$$

- 护盾消耗速率

$$D_{sh}(x_1, x_2) = \eta(x_1) \cdot \max(0.01, 1 - \rho_{sat}(x_2)) \cdot \frac{G_{base}}{10.923556}$$

- 燃料消耗速率

$$F_{use}(x_1, x_2) = \eta(x_1) \cdot (1 - \rho_{sat}(x_2)) \cdot 0.005$$

状态空间模型如下，其中时间 t 的单位是 Game Tick / $\frac{1}{20}$ sec.:

$$x_1(t+1) = x_1(t) + \frac{10}{10000} [T_{rise}(x_2(t)) - T_{resist}(x_1(t))(1 - \lambda_{conv}(x_4(t))) + 1000\lambda_{conv}(x_4(t))]$$

$$x_2(t+1) = x_2(t) - u_1(t) + G_{rate}(x_2(t), x_4(t))$$

$$x_3(t+1) = x_3(t) + u_2(t) - D_{sh}(x_1(t+1), x_2(t))$$

$$x_4(t+1) = x_4(t) + F_{use}(x_1(t+1), x_2(t))$$

2.1.2 瞬时最大功率控制器

一个很显然的想法是把这个反应堆稳定在某种平衡状态上，让它安稳地度过一生。首先，对于无输入系统 $u_1(t) = u_2(t) = 0$ ，不存在平衡态。对于有输入系统，由于你无法控制随时间单增的 $x_4(t)$ ，系统也没有平衡态。但是，相比于其他状态变量， $x_4(t)$ 的变化速度极慢，不妨假设在其余变量向平衡态弛豫的过程中 x_4 不变，这样仍可分析系统 (x_1, x_2, x_3) 的平衡态。

另一个可以观察到的重要性质是（在 x_4 慢变假设下） (x_1, x_2) 组成一个子系统，该子系统只能影响外界（此处特指 x_3 ），无法受外界影响。由 x_1 的平衡条件可以给出 $x_1(t), x_2(t)$ 间满足的方程，解这个方程可获得 $x_1 \mapsto x_2$ 的一一到上映射；由 x_2 的平衡条件可以给出 $u_1(t) = G_{rate}(x_2(t), x_4)$ ，由此可得到 $x_2 \mapsto u_1$ 的一一到上映射。因此， (x_1, x_2) 这个子系统的平衡态可以通过指定平衡态上的 x_1, x_2, u_1 中任意一者来唯一确定。另外，通过 $x_1(t)$ 满足的方程可知：若 $x_1(t)$ 受扰动上升，系统的升温速度 T_{rise} 不变，而降温速度 T_{resist} 上升，因此 $x_1(t)$ 将自动回到平衡态，反之亦然；根据 $x_2(t)$ 满足的方程可知：若 $x_2(t)$ 受扰动上升，能量产生速率 G_{rate} 将下降，而能量提取速度不变，因此 $x_2(t)$ 将自动回到平衡态，反之亦然。因此子系统 (x_1, x_2) 的任一平衡态都是稳定平衡（当然要严格说这件事的话，需要在平衡点附近线性化动力学方程，检查 $x(t+1) = Ax(t) + B$ 中矩阵 A 的本征值）。通过求解 x_3 的平衡方程立刻得到 $u_2(t) = D_{sh}(x_1(t+1), x_2(t))$ ，这给出平衡时需要的 $u_2(t)$ 。由于 $x_3(t+1) - x_3(t) = \dots$ 的等号右侧不含 x_3 ，且 x_3 无法影响 x_1, x_2 ，因此 $x_3(t)$ 是随遇平衡。

根据以上分析，我们很容易设计一种瞬时最高发电功率控制器。平衡态上的瞬时发电功率是：

$$P_{gen} = u_1(t) - u_2(t) = u_1(x_1) - D_{sh}(x_1(t+1), x_2(t)) = u_1(x_1) - D_{sh}(x_1)$$

寻找瞬时最大发电功率的问题是一个单变量优化问题，很容易对每个燃料消耗进度 x_4 找到瞬时发电功率最大的温度 x_1^* 。找到 x_1^* 后，通过向系统输入能量 $u_1(x_1^*)$ （这个值通过 x_1, x_2 的平衡方程解出）来将子系统 (x_1, x_2) 移动到平衡点上，设置抽取速率 $u_2 = D_{sh}$ ，再为此时随遇平衡的 x_3 单独设置一个比例控制器，将它稳定在某个目标值上即可。

我们给出不同燃料消耗进度 x_4 下反应堆的工作特性曲线（平衡点上， $u_1(x_1) - u_2(x_1)$ 和 $x_2(x_1)$ 曲线），以及在这种瞬时最大发电策略下反应堆的运行全过程（直到 90% 燃料被消耗为止）。

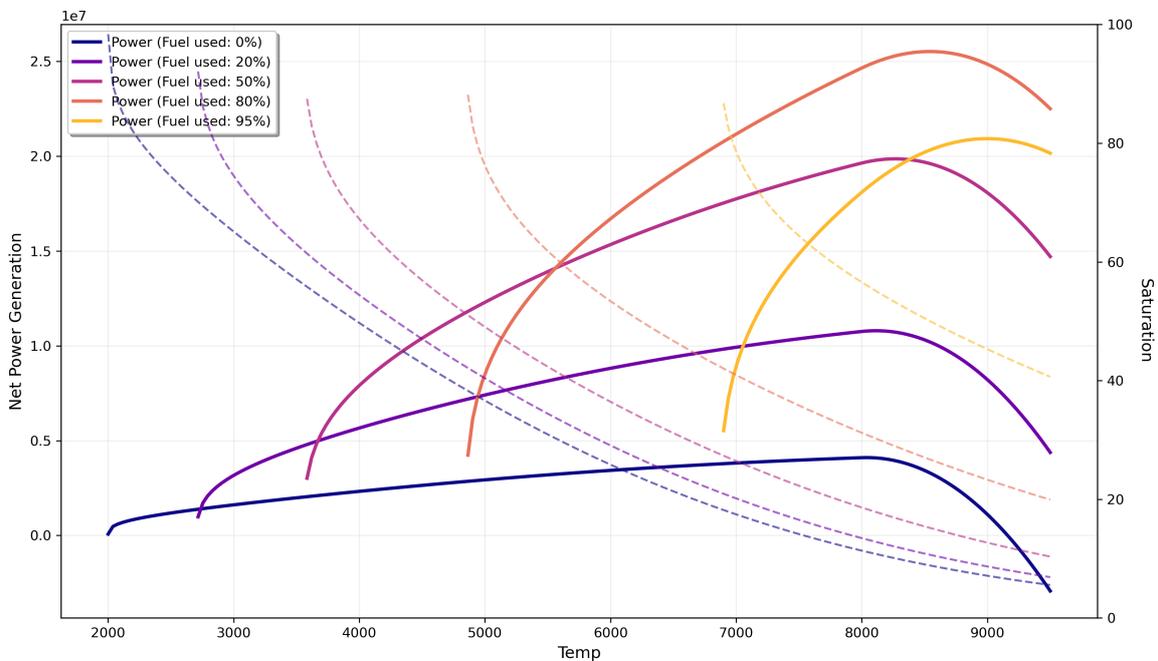


图 2.3: 不同燃料消耗进度下反应堆的工作特性曲线。随着燃料消耗，反应堆最大功率对应的温度上移；最大功率先增后减。（左轴：净发电量；右轴：电量缓存）

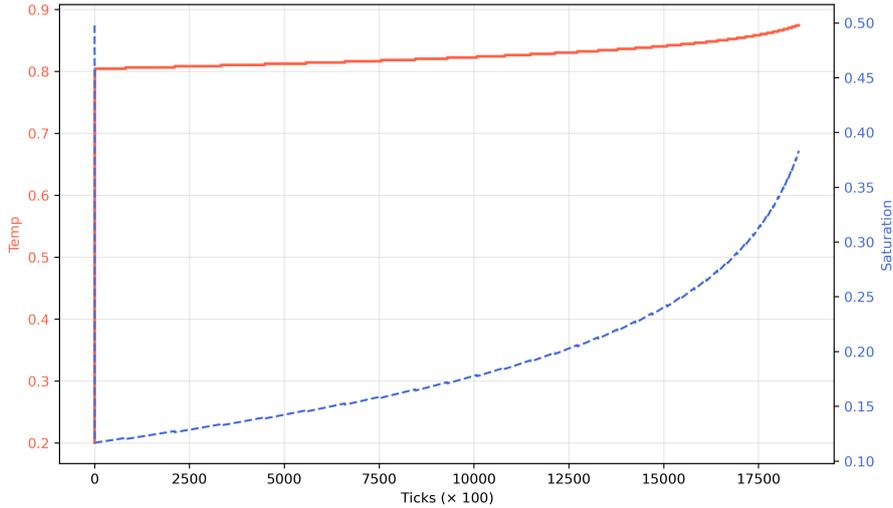


图 2.4: 最大瞬时功率控制下反应堆的运行全程。随着燃料消耗，控制器逐渐上移反应堆的工作温度 x_1 。(左轴：温度/最大温度；右轴：电量缓存/最大电量缓存)

2.1.3 全程最大发电总量控制器

燃料充足时，你可以使用最大瞬时功率控制器；燃料不足时，你可能需要全程最大发电控制器：指定关机时燃料消耗的比例 $x_{4,f}$ ，最大化（原则上这里应该写求和）：

$$J[u_1, u_2] = \int_0^{t_f} [u_1(t) - u_2(t)] dt \quad x_4(t_f) = x_{4,f}$$

这是一个终端时间不固定，但是（部分）终端状态固定的问题。在求解这个问题之前，我们需要对其做两点简化：首先，（就算不采用 x_4 慢变近似） $x_3(t)$ 只能受到其他变量的影响，而不能影响其他变量，因此我们可以令 $u_2(t) = D_{sh}$ ，并将 $x_3(t)$ 稳定在任意合法值上。这样，我们只有一个控制变量：

$$J[u_1] = \int_0^{t_f} [u_1(t) - D_{sh}(x_1(t+1), x_2(t))] dt := \int_0^{t_f} [u_1(t) - \hat{D}_{sh}(x_1(t), x_2(t), x_4(t))] dt$$

其中 $\hat{D}_{sh}(x_1(t), x_2(t), x_4(t))$ 是将 $x_1(t+1)$ 带入 $D_{sh}(x_1(t+1), x_2(t))$ 中得到的。此外，作为一个终端时间自由的问题，不适合通过初始化终端时刻的值函数，然后反向递推的方式求解。当然，你可以使用 Value Iteration 算出值函数，但是对于本例，我们有随时间单增的 $x_4(t)$ ，且终端时 $x_4(t)$ 固定，因此我们可以将 $x_4(t)$ 作为新的时间。考虑 $x_4(t)$ 的动力学方程：

$$x_4(t+1) = x_4(t) + F_{use}(x_1(t+1), x_2(t)) := \hat{F}_{use}(x_1(t), x_2(t), x_4(t))$$

这里的 $\hat{F}_{use}(x_1(t), x_2(t), x_4(t))$ 也是将 $x_1(t+1)$ 带入 $F_{use}(x_1(t+1), x_2(t))$ 得到的。也就是说，每消耗 Δf 单位的燃料，时间前进

$$\Delta t = \frac{\Delta f}{\hat{F}_{use}(x_1(t), x_2(t), x_4(t))}$$

我可以将 $x_1(t), x_2(t)$ 的动力学方程改写为：

$$x_1(x_4 + \Delta x_4) = x_1(x_4) + \left(\frac{10}{10000} [T_{rise}(x_2(x_4)) - T_{resist}(x_1(x_4))(1 - \lambda_{conv}(x_4))] + 1000\lambda_{conv}(x_4) \right) \cdot \frac{\Delta x_4}{\hat{F}_{use}(x_1, x_2, x_4)}$$

$$x_2(x_4 + \Delta x_4) = x_2(x_4) - (u_1 + G_{rate}(x_2(x_4), x_4)) \cdot \frac{\Delta x_4}{\hat{F}_{use}(x_1, x_2, x_4)}$$

性能指标：

$$J[u_1] = \int_0^{x_{4,f}} [u_1(x_4) - \hat{D}_{sh}(x_1(x_4), x_2(x_4), x_4)] \frac{1}{\hat{F}_{use}(x_1, x_2, x_4)} dx_4$$

之后，这个问题可以通过离散化状态空间 (x_1, x_2) 、等效的时间步 x_4 以及控制变量 u_1 来求解。这里直接给出最大发电量控制律下反应堆的运行过程：

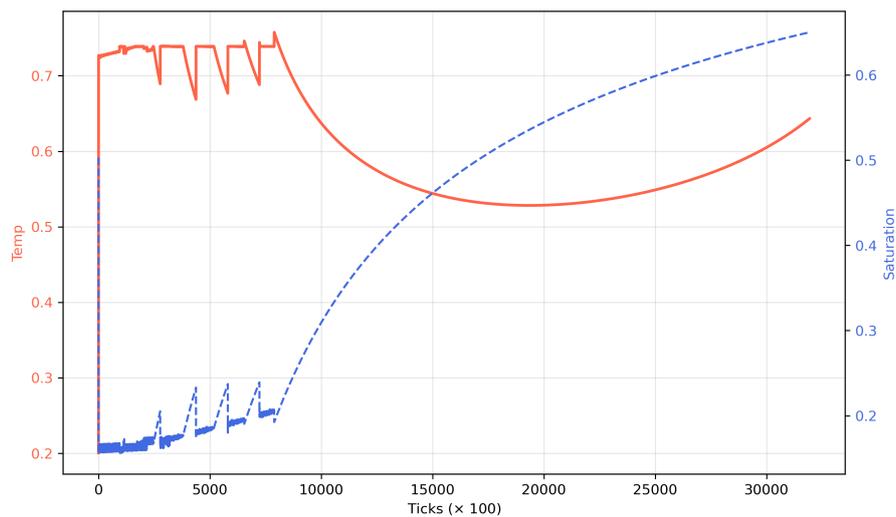


图 2.5: 最大发电量控制下反应堆的运行全程。(左轴：温度/最大温度；右轴：电量缓存/最大电量缓存)