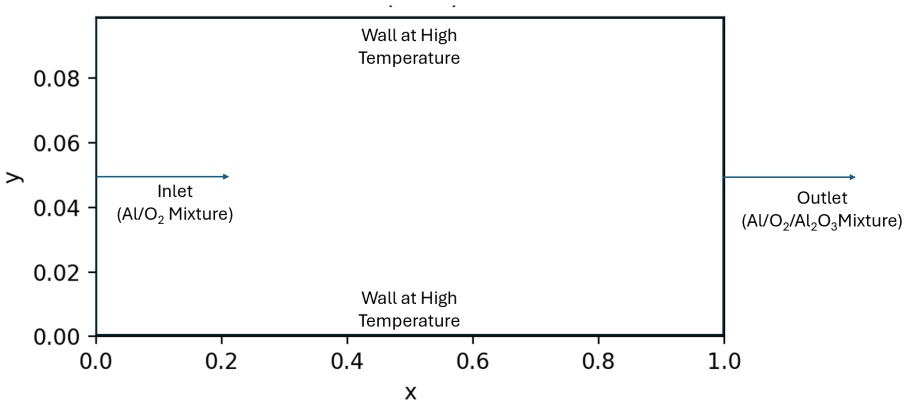
Reinforcement Learning Model for Heat-Diffusion/Reaction-Diffusion with Surrogate Environment

Overview

Aluminum combustion offers significant potential for sustainable energy applications, yet complex reaction-diffusion processes make optimization challenging [1]. This project develops a mathematical model for the combustion reaction, explores parameter ranges through batch simulations, and develops a hybrid neural network/gradient boosting surrogate model using CUDA on GPUs through Google Colab. Then, the Neural Network surrogate will be used as a simulated environment in an Actor-Critic Reinforcement Learning (RL) model. The intent of this RL model with a simulated surrogate neural network is to capture and control the complex stochastic behavior of combustion processes.

Mathematical Model

Let's say we have a 2-Dimensional area for the combustion reaction with $x \in [0, 1]$ and $y \in [0, 0.1]$.



The aluminum and oxygen mixture enters from the left-hand side and exists as an aluminum, oxygen, and aluminum oxide mixture. For optimal performance, the outlet flow would only consist of aluminum oxide. This is unrealistic, so the goal will be to ensure all aluminum is combusted by the outlet.

To simplify the math model, we introduce the inlet flow at high temperatures, 1500 K, to prevent large density and pressure changes within the given domain. This simplification allows for the mathematical model to avoid adding a fluid mechanics portion, and solely relies on the reaction-diffusion and heat-diffusion equations.

There will be two reaction-diffusion equations. The first is as follows and models the concentration of the oxygen. All partial differential equations for the mathematical model are taken from Fundamentals of Heat and Mass Transfer [2].

$$\frac{\partial c_{o_2}}{\partial t} + u \cdot \nabla c_{o_2} = D\nabla^2 c_{o_2} - \nu_{O_2} R(c_{Al}, c_{o_2}, T)$$

Where, ν_{O_2} is the stoichiometric factor for the reaction. We include advection to account for bulk flow of the oxygen as it moves throughout the domain. To define this, we refer to the reaction equation as follows:

$$4AL + 3O_2 \rightarrow 2Al_2O_3$$

This indicates that $\nu_{O_2} = \frac{4}{3}$. The second reaction-diffusion equation models the concentrations of aluminum throughout the domain.

$$\frac{\partial c_{Al}}{\partial t} = -R(c_{Al}, c_{O_2}, T)$$

Only one form of the heat-diffusion equation considered.

$$\frac{\partial T}{\partial t} + u \cdot \nabla T = \kappa \nabla^2 T + \frac{Q}{\rho C_P} R(c_{Al}, c_{O_2}, T)$$

Where, we define Q and the subsequent parameters as follows:

$$Q \;=\; \frac{-\,\Delta H_{\text{rxn}}}{\nu_{\text{ref}}}, \quad \nu_{ref} = \nu_{al} = 4, \quad \Delta H_{rxn} \approx -3.351x10^6 J$$

Finally, $R(c_{Al}, c_{O_2}, T)$ is defined as follows:

$$R(c_{Al}, c_{O_2}, T) = k(T)\; c_{Al}^{\alpha}\; c_{O_2}^{\beta} = \; c_{Al}^{\alpha}\; c_{O_2}^{\beta} K_0 \exp\left(-\frac{E_a}{R_g\, T}\right)$$

We assume that this is a first order reaction, so $\alpha = \beta = 1$.

$$R(c_{Al}, c_{O_2}, T) = k(T)\; c_{Al}^{\alpha}\; c_{O_2}^{\beta} = \; c_{Al}\; c_{O_2} K_0 \exp\left(-\frac{E_a}{R_g\, T}\right)$$

Defining the rest of the nomenclature:

D_{O_2}: oxygen diffusion coefficient (m^2/s)=1

κ: thermal diffusivity (m^2/s)=1

Q: heat released per unit mass of Al reacted (J/kg)

ρC_p : volumetric heat capacity $(J/(m^3 K)$=1

k_0 : Pre-exponential factor =1000

E_a : Activation energy (J/mol) =1.25e5

R_g : Universal gas constant $(kJ/mol/K)$=8.314

Surrogate Model

Finite Difference Discretization of Partial Differential Equations

Let's say we have a domain $x \in [0, L]$ and $y \in [0, H]$ with constant flow in the x-direction. We can update the 2-dimensional oxygen reaction-diffusion equation for with advection to the following:

$$\frac{\partial c_{O_2}}{\partial t} + u_0 \frac{\partial c_{O_2}}{\partial x} = D \left(\frac{\partial^2 c_{O_2}}{\partial x^2} + \frac{\partial^2 c_{O_2}}{\partial y^2} \right) - \nu_{O_2} R(c_{Al}, c_{O_2}, T)$$

The aluminum 2-dimensional reaction-diffusion equation will be modeled with no advection since it is assumed to be in solid form. The aluminum particles are assumed to be embedded within the oxygen flow as a solid particulate phase. Unlike the gaseous oxygen, these solid particles do not exhibit significant diffusion through the medium, so the diffusion coefficient is neglected. The aluminum concentration changes only through the combustion reaction, not through diffusive transport. The equation is then simplified to the following:

$$\frac{\partial c_{Al}}{\partial t} = -R(c_{Al}, c_{O_2}, T)$$

The 2-dimensional heat-diffusion equation with advection can be written as follows:

$$\frac{\partial T}{\partial t} + u_o \frac{\partial T}{\partial x} = \kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + \frac{Q}{\rho C_P} R(c_{Al}, c_{O_2}, T)$$

Initial Conditions (all constant and uniform)

$$c_{O_2}(x, y, t = 0) = c_{O_2,0}, \quad c_{Al}(x, y, t = 0) = c_{Al,o}, \quad T(x, y, t = 0) = T_0$$

The concentrations are assumed to be in units of $\frac{kg}{m^3}$.

Boundary Conditions

At x = 0:

$$c_{O_2}(0, y, t) = c_{O_2,in}, \quad c_{Al}(0, y, t) = c_{Al,in}, \quad T(0, y, t) = T_{in}$$

At x = L, we have the following unknown boundary conditions:

$$c_{O_2}(L, y, t), \quad c_{Al}(L, y, t), \quad T(L, y, t)$$

Assuming the walls are heated to encourage combustion, and that the air is the same temperature at the wall. At y = 0:

$$\frac{\partial c_{o_2}}{\partial y}|_{x \in [0,L], y=0} = 0, \quad T(x, 0, t) = T_w(x)$$

At y = L:

$$\frac{\partial c_{o_2}}{\partial y}|_{x, y=H} = 0, \quad T(x, H, t) = T_w(x)$$

Now, for the numerical discretization of each equation. For spatial components, we will use the explicit and Crank-Nicolson methods for the spectral and spatial components, respectively.

For the reaction-diffusion equation for oxygen:

$$\frac{\partial \phi}{\partial t} = \frac{\phi_{i,j+1} - \phi_{i,j}}{k}$$

$$\left(\frac{\partial \phi}{\partial x} \right)_{i,j}^{\text{CN}} := \tfrac{1}{2} \left[\delta_x \phi_{i,j}^{k+1} + \delta_x \phi_{i,j}^{k} \right] = \frac{1}{2} \left[\frac{\phi_{i+1,j}^{k+1} - \phi_{i-1,j}^{k+1}}{2h_x} + \frac{\phi_{i+1,j}^{k} - \phi_{i-1,j}^{k}}{2h_x} \right],$$

$$\left(\frac{\partial \phi}{\partial y} \right)_{i,j}^{\text{CN}} := \tfrac{1}{2} \left[\delta_y \phi_{i,j}^{k+1} + \delta_y \phi_{i,j}^{k} \right] = \frac{1}{2} \left[\frac{\phi_{i,j+1}^{k+1} - \phi_{i,j-1}^{k+1}}{2h_y} + \frac{\phi_{i,j+1}^{k} - \phi_{i,j-1}^{k}}{2h_y} \right],$$

$$\left(\frac{\partial^2 \phi}{\partial x^2} \right)_{i,j}^{\text{CN}} := \tfrac{1}{2} \left[\delta_{xx} \phi_{i,j}^{k+1} + \delta_{xx} \phi_{i,j}^{k} \right] = \frac{1}{2} \left[\frac{\phi_{i+1,j}^{k+1} - 2\phi_{i,j}^{k+1} + \phi_{i-1,j}^{k+1}}{h_x^2} + \frac{\phi_{i+1,j}^{k} - 2\phi_{i,j}^{k} + \phi_{i-1,j}^{k}}{h_x^2} \right],$$

$$\left(\frac{\partial^2 \phi}{\partial y^2} \right)_{i,j}^{\text{CN}} := \tfrac{1}{2} \left[\delta_{yy} \phi_{i,j}^{k+1} + \delta_{yy} \phi_{i,j}^{k} \right] = \frac{1}{2} \left[\frac{\phi_{i,j+1}^{k+1} - 2\phi_{i,j}^{k+1} + \phi_{i,j-1}^{k+1}}{h_y^2} + \frac{\phi_{i,j+1}^{k} - 2\phi_{i,j}^{k} + \phi_{i,j-1}^{k}}{h_y^2} \right].$$

Defining numerical solution for c_{O_2}:

$$\Delta t = k, \quad h_x = \Delta x, \quad h_y = \Delta y, \quad r_x := \frac{D\,\Delta t}{2h_x^2}, \quad r_y := \frac{D\,\Delta t}{2h_y^2}, \quad a := \frac{u_0\,\Delta t}{4h_x}$$

Which results in the following system:

$$A\,\mathbf{c}_j^{k+1} = B\,\mathbf{c}_j^{k} + \mathbf{d}_j^{k},$$

Where A and B are tridiagonal matrices with the following values:

$$A = \begin{bmatrix} 1+2r_x & -r_x+a & & \\ -r_x-a & 1+2r_x & \ddots & \\ & \ddots & \ddots & -r_x+a \\ & & -r_x-a & 1+2r_x \end{bmatrix}$$

$$B = \begin{bmatrix} 1-2r_x & r_x-a & & \\ r_x+a & 1-2r_x & \ddots & \\ & \ddots & \ddots & r_x-a \\ & & r_x+a & 1-2r_x \end{bmatrix}$$

Finally, d_j^k is a column vector that includes known values.

$$\mathbf{d}_j^k = \begin{bmatrix} r_y\left(c_{1,j-1}^k - 2c_{1,j}^k + c_{1,j+1}^k\right) - \Delta t\,\nu_{O_2} R_{1,j}^k \\ r_y\left(c_{2,j-1}^k - 2c_{2,j}^k + c_{2,j+1}^k\right) - \Delta t\,\nu_{O_2} R_{2,j}^k \\ \vdots \\ r_y\left(c_{n-1,j-1}^k - 2c_{n-1,j}^k + c_{n-1,j+1}^k\right) - \Delta t\,\nu_{O_2} R_{n-1,j}^k \end{bmatrix}.$$

For the reaction-diffusion equation for aluminum:

$$c_{Al}^{k+1} = c_{Al}^{k} - \Delta t\Big[R\big(c_{Al}^k,\, c_{O_2}^k,\, T^{\,k}\big)\Big]$$

For the heat-diffusion equation:

$$r_x := \frac{\kappa\,\Delta t}{2h_x^2}, \qquad r_y := \frac{\kappa\,\Delta t}{2h_y^2}, \qquad a := \frac{u_0\,\Delta t}{4h_x}, \qquad \gamma := \Delta t\,\frac{Q}{\rho C_P}$$

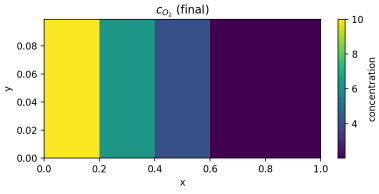
$$A\,\mathbf{T}_j^{k+1} = B\,\mathbf{T}_j^{k} + \mathbf{s}_j^{k},$$

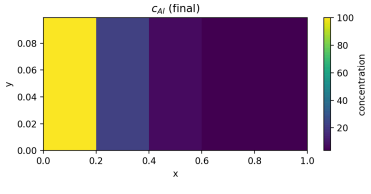
$$A = \begin{bmatrix} 1+2r_x & -r_x+a & & \\ -r_x-a & 1+2r_x & \ddots & \\ & \ddots & \ddots & -r_x+a \\ & & -r_x-a & 1+2r_x \end{bmatrix}, \qquad B = \begin{bmatrix} 1-2r_x & r_x-a & & \\ r_x+a & 1-2r_x & \ddots & \\ & \ddots & \ddots & r_x-a \\ & & r_x+a & 1-2r_x \end{bmatrix}.$$

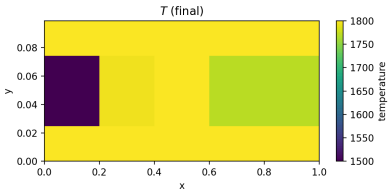
$$\mathbf{s}_j^k = \begin{bmatrix} r_y(T_{1,j-1}^k - 2T_{1,j}^k + T_{1,j+1}^k) + \gamma R_{1,j}^k \\ r_y(T_{2,j-1}^k - 2T_{2,j}^k + T_{2,j+1}^k) + \gamma R_{2,j}^k \\ \vdots \\ r_y(T_{n-1,j-1}^k - 2T_{n-1,j}^k + T_{n-1,j+1}^k) + \gamma R_{n-1,j}^k \end{bmatrix}.$$

To test the model, we start with a single case defined by the following parameters:

$u_0\ (\frac{m}{s})$	$K_0\ (\frac{mol}{m^3})$	$T(x=0,y,t)(^\circ C)$	$T(x,y=0=H,t)(^\circ C)$	$c_{O_2}(x=0,y,t)$	$c_{Al_in}(x=0,y,t)$
0.1	1	1500	1600	130	100







The three heat maps demonstrate the model's physical behavior. The oxygen concentration (left) decreases from inlet to outlet as it's consumed in the reaction. The aluminum concentration (center) shows similar consumption patterns with localized depletion zones where reaction rates are highest. The temperature distribution (right) shows heating throughout the domain, with peak temperatures occurring where the exothermic reaction is most active, validating the coupled heat-reaction physics.

Batch Generation of Parameters

The following parameters were added to the batch generation process. These are physical parameters calculated from the system metrics that may give the surrogate model more insight into the physical nature of the system. First, starting with the residence time, which represents the amount of time spent within our domain.

$$\tau = \frac{L}{u_0}$$

Next, we define the Mass Peclet Number, which compares advection and diffusion in the species [3].

$$PE_c = \frac{u_0 L}{D}$$

The Thermal Peclet number is similar, but compares the advection and diffusion for heat [4]

$$PE_T = \frac{u_0 L}{\kappa}$$

Finally, the Damkohler number will be defined, which compares the reaction versus transport rates [5].

$$Da = k\tau$$

The batch generation of parameters was conducted as follows:

Value	$u_0\ \left(\frac{m}{s}\right)$	$K_0\ \left(\frac{mol}{m^3}\right)$	$T(x, y = 0 = H, t)(K)$
Minimum	0.02	1	300
Maximum	2.0	1e8	1800

By changing the velocity above, the following extracted features will indirectly vary:

Feature	τ	PE_c	PE_T	Da
Note	Varies with u_0	Varies with u_0	Varies with u_0	Varies with u_0 and K

Each parameter was uniformly distributed to produce approximately 1600 total datapoint. The aluminum and oxygen inlet concentrations were held constant. If the data is too easy for the surrogate model to predict, then more batch simulations will be ran with varying inlet temperatures. An R^2 value of 1 across all outputs will be an indicator of this phenomenon.

$c_{Al}(x = 0, y, t)c_{Al}(x = 0, y, t)c_{Al}(x = 0, y, t)$	$c_{O_2}(x = 0, y, t)$	$T(x = 0, y, t)(K)$
100	130	1500

Initially, the batch generation was defined in Python using Visual Studio Code. However, the average run time per simulation was over 3 seconds. To achieve 2000 data points, it would have taken nearly two hours. Instead, the batch simulation utilized Google Colab's GPUs with CUDA programming. By using GPUs, the simulation time decreased from hours to seconds while generating 2000 simulations.

Surrogate Model Training and Performance

As outlined in the batch simulation process, without GPUs, the simulations would take over 3 seconds each, which is higher than some residence times in our system. Without GPUs, the RL model wouldn't be able to use the mathematical model to

determine outlet conditions in real time. We move forward with the surrogate model based on the assumption that it's unrealistic to have a GPU dependent simulated environment for the RL model. Instead, we use the batch simulated data to develop a surrogate model, which provides immediate state information at or quicker than real time. The features for the surrogate model are as follows.

	Primary	Features			Derived	Features	
Feature	u_0	K_0	T_{wall}	τ	PE_c	PE_T	Da
Note	Varied	Varied	Varied	Varies withu_0	Varies with u_0	Varies with u_0	Varies with u_0 and K

Then, the physic parameters of the system predicted by the surrogate model are as follows:

$$y_1 \rightarrow c_{Al,out} \quad y_2 \rightarrow c_{O_2,out} \quad y_3 \rightarrow T_{out}$$

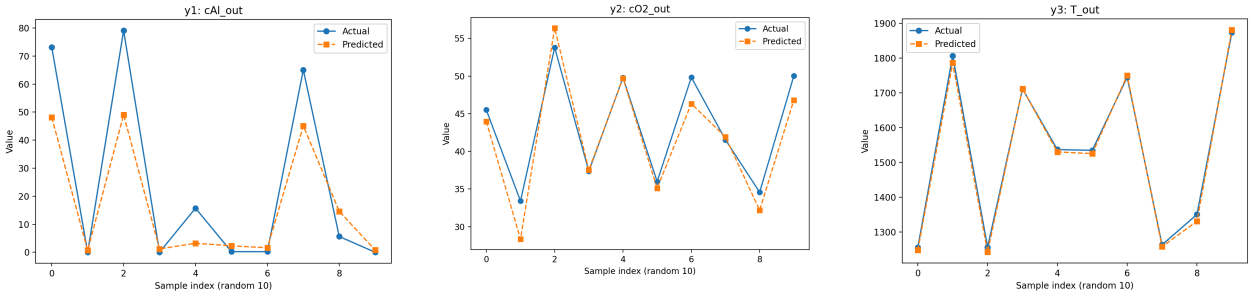
After hyperparameter tuning and analyzing performances of the model, it was determined that the best way to develop the surrogate model was to create a hybrid approach where y_1 is predicted separately from y_2 and y_3. The hybrid approach was necessary due to the different characteristics of the output variables. The aluminum outlet concentration (y_1) exhibited discontinuous spikes corresponding to uncombusted aluminum particles, making it well-suited for Gradient Boosting Regression, which handles non-smooth, irregular patterns effectively. In contrast, the oxygen concentration and temperature outputs showed smooth, continuous behavior that Neural Networks model well.

All input features were normalized to increase performance. The activation functions were applied as part of the hyperparameters for the tuning process, giving the option between ReLU, tanh, logistic, and identity functions. Error handling was applied to ensure feature order validation, invalid activation functions, data cleaning for nan and inf values, along with input shape handling. The latency of the model did not require the need for any speed enhancements - GPU use with CUDA would have been overkill.

The R^2 and MSE are included in the table below.

$Subset(10)\ R^2$	$c_{Al_{Out}}\ R^2$	$c_{Al_{Out}}\ MSE$	$c_{O2_{Out}}\ R^2$	$c_{O_2,Out}\ MSE$	$T_{Out}\ R^2$	$T_{Out}\ MSE$
0.8878	0.7912	217.2	0.8746	6.506	0.9975	130.6

A subset of 10 data points were plotted with the actual and predicted values for $y_1, y_2,\ and\ y_3$.

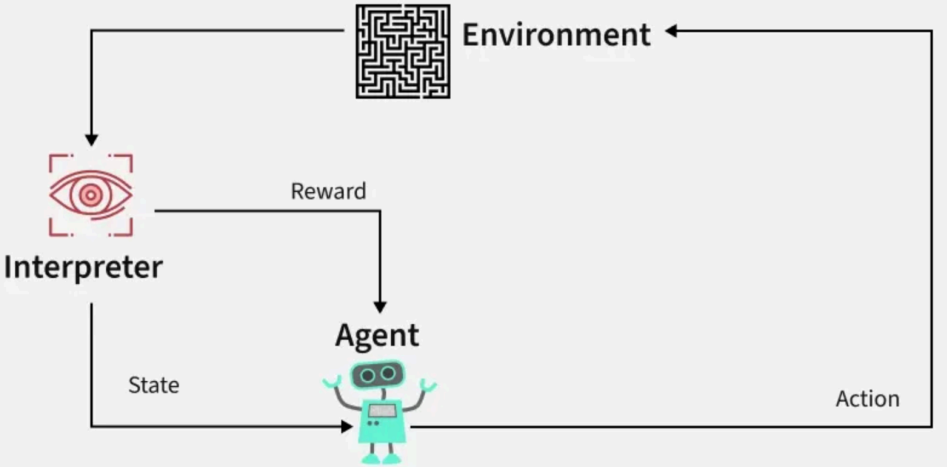


The plots above indicate that $y_2\ and\ y_3$ closely matched the actual data. However, due to the nature of the outlet aluminum concentrations, y_1 struggles to match the actual data, but we assume that it is close enough for the purposes of the RL model since it follows the relative pattern.

Reinforcement Learning Model

We use an Actor-Critic PPO model that starts with a surrogate model environment. The reward, state, and action all tie back to the surrogate model outputs and features.

GeeksforGeeks provides a visual of an RL model architecture [6].



The purpose-built interpreter is not needed, since the surrogate model will provide deterministic results. In scenarios where sensors provide noisy state measurement, a Kalman filter or signal processing technique could be used for the interpreter.

The state space includes the variables $c_{Al,out}$, $c_{O_2,out}$, and T_{out} which are anticipated to provide enough information for the action space. The reward is defined as follows:

$$Reward \; = \; -\epsilon[W_{temp}(T_{out} - 1500(K))^2 + W_{c_{Al}} c_{Al,out}^2 + W_{c_{O_2}} c_{O_2,out}^2]$$

Where, after testing, viable values for the weights were defined as follows:

$$W_{temp} = 1.0 \quad W_{c_{Al}} = W_{c_{O_2}} = 0.1$$

The weights prioritize energy efficiency over perfect combustion. $W_{temp} = 1.0$ heavily penalizes excessive wall heating, encouraging energy-efficient operation. The lower weights $W_{c_{Al}} = W_{c_{O_2}} = 0.1$ for concentration terms reflect that some residual unreacted material is acceptable if it significantly reduces energy input. This represents a practical trade-off between combustion completeness and operational costs.

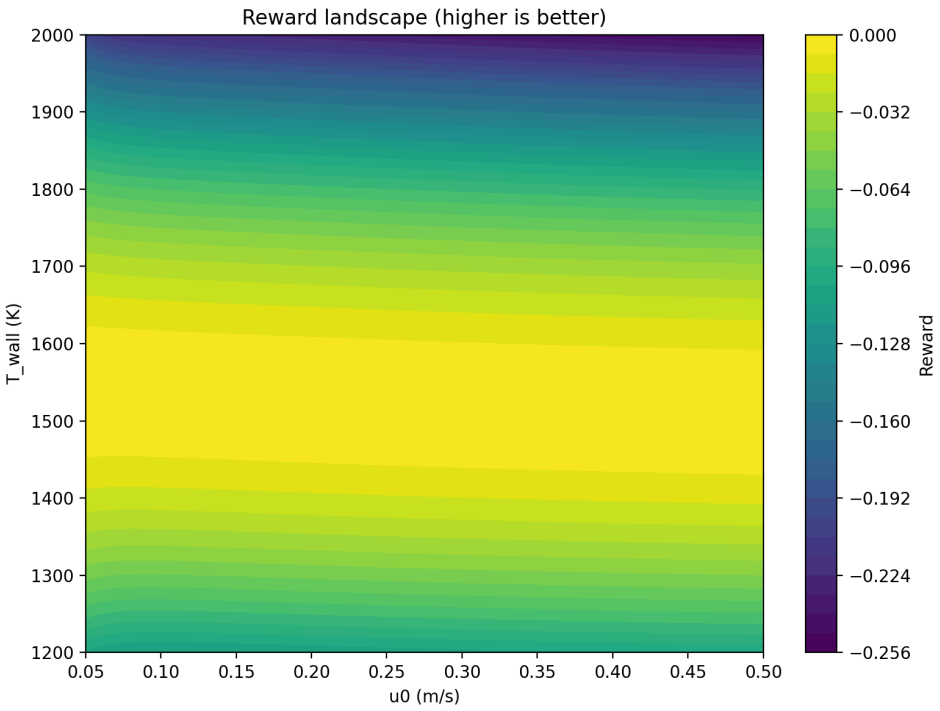
The reward is defined such that the agent is incentivized to minimize the temperature rise across the domain and the outlet concentration of aluminum and oxygen. Physically, these policies train the model to input the least amount of thermal energy while also ensuring that the aluminum is combusted. The policy network uses a 3D input and 2D output, ReLU hidden layers, and sigmoid output layer. The algorithm uses REINFORCE with baseline (policy gradient), exploration is Gaussian noise with annealing standard deviation.

Training details included a batch size of 1024 episode, learning rate of $5x10^{-4}$, logarithmic standard deviation annealing for exploration decay, and 2000 total training episodes.

Once the agent interprets the state and the reward, the policy implements the action by changing the flow of aluminum and oxygen, along with altering the temperatures of the walls, also known as the action space:

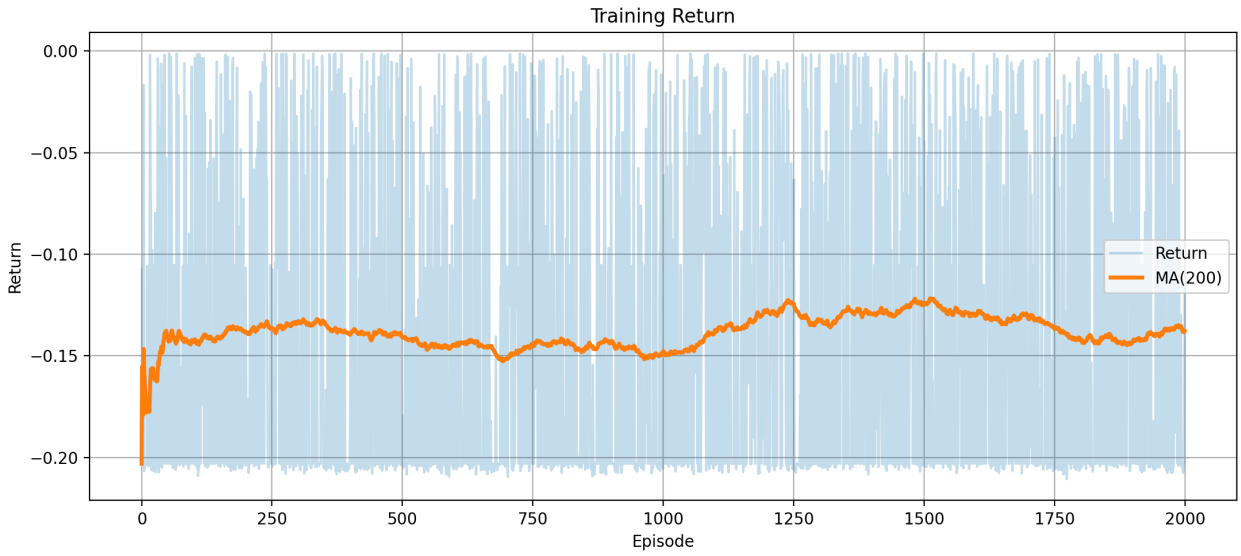
$$T_{wall} \in [1200 - 2000] \; (K) \quad u_0 \in [0.05 - 0.5 \,] \left(\frac{m}{s}\right)$$

The following plot indicates the reward as a function of the action space - wall temperature T_{wall} and u_0.



The reward landscape reveals the optimal operating regime. The contour pattern shows that slower velocities (longer residence times) allow more complete combustion, while wall temperatures around 1500K provide the optimal balance - sufficient activation energy without excessive energy waste. The negative slope indicates that velocity has less impact than wall temperature on performance, consistent with reaction-limited rather than transport-limited behavior.

The training return per episode is provided in the visual below that includes the instantaneous return, along with the moving average of 200 episodes, MA(200).



The training curve demonstrates successful learning convergence. The high variance in individual episode returns (blue) is typical of exploration-based learning, while the moving average (orange) shows steady improvement and stabilization around episode 300 at approximately -0.13 reward units. The lack of further improvement after episode 300 indicates the policy has found the optimal strategy within the given constraints.

Conclusion

The project successfully developed a framework for modeling and optimizing aluminum-oxygen combustion processes through physics based simulation, surrogate modeling, and reinforcement learning control.

The mathematical modeling and simulation portion of the project established a 2D reaction/heat-diffusion model based on coupled partial differential equations. The model incorporates Arrhenius kinetic, boundary conditions, and finite difference discretization using the Crank-Nicolson method. Temperature predictions achieved an R^2 value of approximately 0.9975, indicating that the surrogate model successfully captures the underlying combustion physics.

A breakthrough in latency was achieved through GPU parallelization using Google Colab's CUDA capabilities. This reduced simulation time from over 3 seconds per case to milliseconds, enabling the generation of 2000 training cases in seconds rather than hours.

The hybrid surrogate model architecture combined gradient boosting regression for aluminum concentration and neural networks for oxygen concentration and temperature achieved strong overall performance ($R^2 = 0.888$). The model successfully learned the complex physics of combustion reactions and outlet conditions, providing millisecond predictions suitable for control applications.

The reinforcement learning implement successfully learned to control reactor conditions through wall temperature and inlet velocity optimization. The REINFORCE algorithm with Gaussian exploration converged to stable policies within 300 episodes, achieving consistent performance around -0.13 reward units. The reward analysis revealed physically meaningful optimization patterns, with the best performance at approximately 1500 K wall temperature and moderate flow velocities.

Works Cited

[1] Trowell, K., Goroshin, S., Frost, D., & Bergthorson, J. (2020). Aluminum and its role as a recyclable, sustainable carrier of renewable energy. *Applied Energy*, *275*, 115112. https://doi.org/10.1016/j.apenergy.2020.115112

[2] Bergman, T. L. (2011). *Fundamentals of heat and mass transfer*. John Wiley & Sons.

[3] Chakraborty, S. (2012). *Microfluidics and microscale transport processes*. CRC Press.

[4] Chandra, Y. P., & Matuska, T. (2019). Stratification analysis of domestic hot water storage tanks: A comprehensive review. *Energy and Buildings*, *187*, 110–131. https://doi.org/10.1016/j.enbuild.2019.01.052

[5] Rehage, H., & Kind, M. (2020). The first Damköhler number and its importance for characterizing the influence of mixing on competitive chemical reactions. *Chemical Engineering Science*, *229*, 116007. https://doi.org/10.1016/j.ces.2020.116007

[6] GeeksforGeeks. (2025, February 24). *Reinforcement learning*. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/what-is-reinforcement-learning/