UCONN
UNIVERSITY OF CONNECTICUT

Improving Operations of Wastewater Treatment Plants Using Economic MPC

Presented by Pengfei Xu
Dimitri Alston, Alireza Miraliakbar, Wenjun Xiang
Pratt & Whitney Associate Professor
Matthew D. Stuber

2025, Nov 3

2025 AIChE ANNUAL MEETING

Process Systems and Operations Research Laboratory

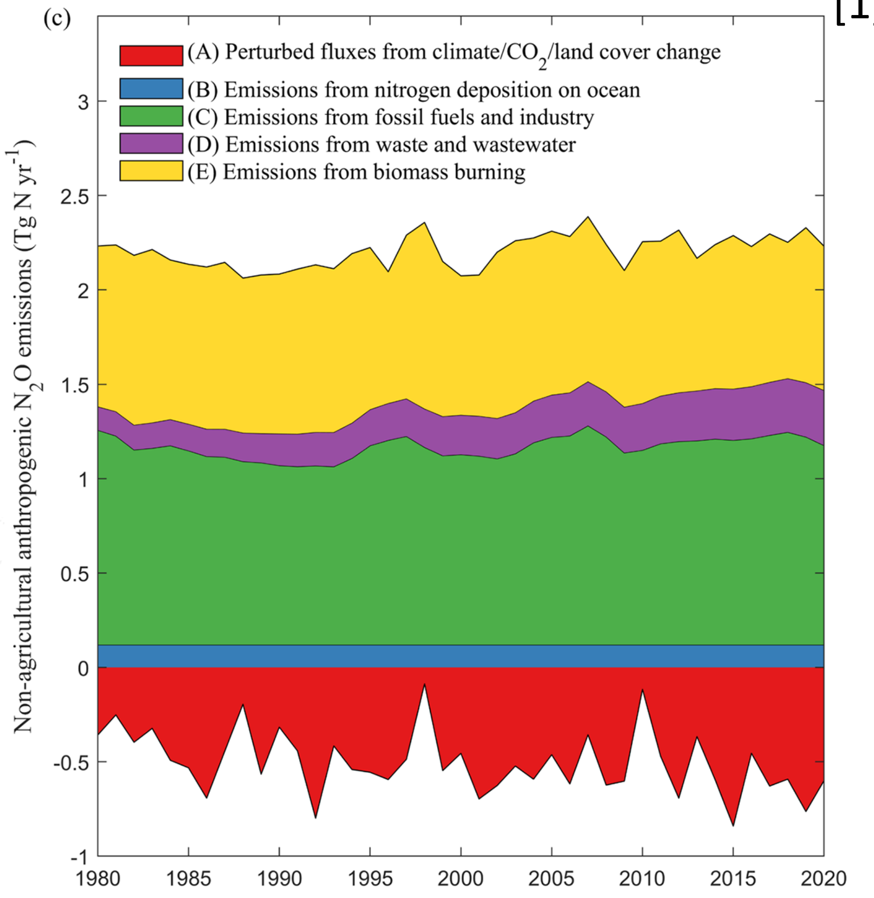
Motivation: Energy and Sustainability in WWTPs

- WWTPs emit **N_2O**, a greenhouse gas **273× stronger than CO_2**.
- **Aeration uses 50–90%** of total plant energy[2].
- Traditional (Dissolved Oxygen) DO control ignores the **energy–sustainability trade-off**.

➢ **Goal:** Operate WWTPs more efficiently — **reduce energy use while maintaining effluent quality.**

➢ **This talk:** Develops an **ASM3-based nonlinear MPC** and **eMPC** framework for **NH_4 control**, laying the groundwork for future **N_2O-integrated optimization.**

[1]



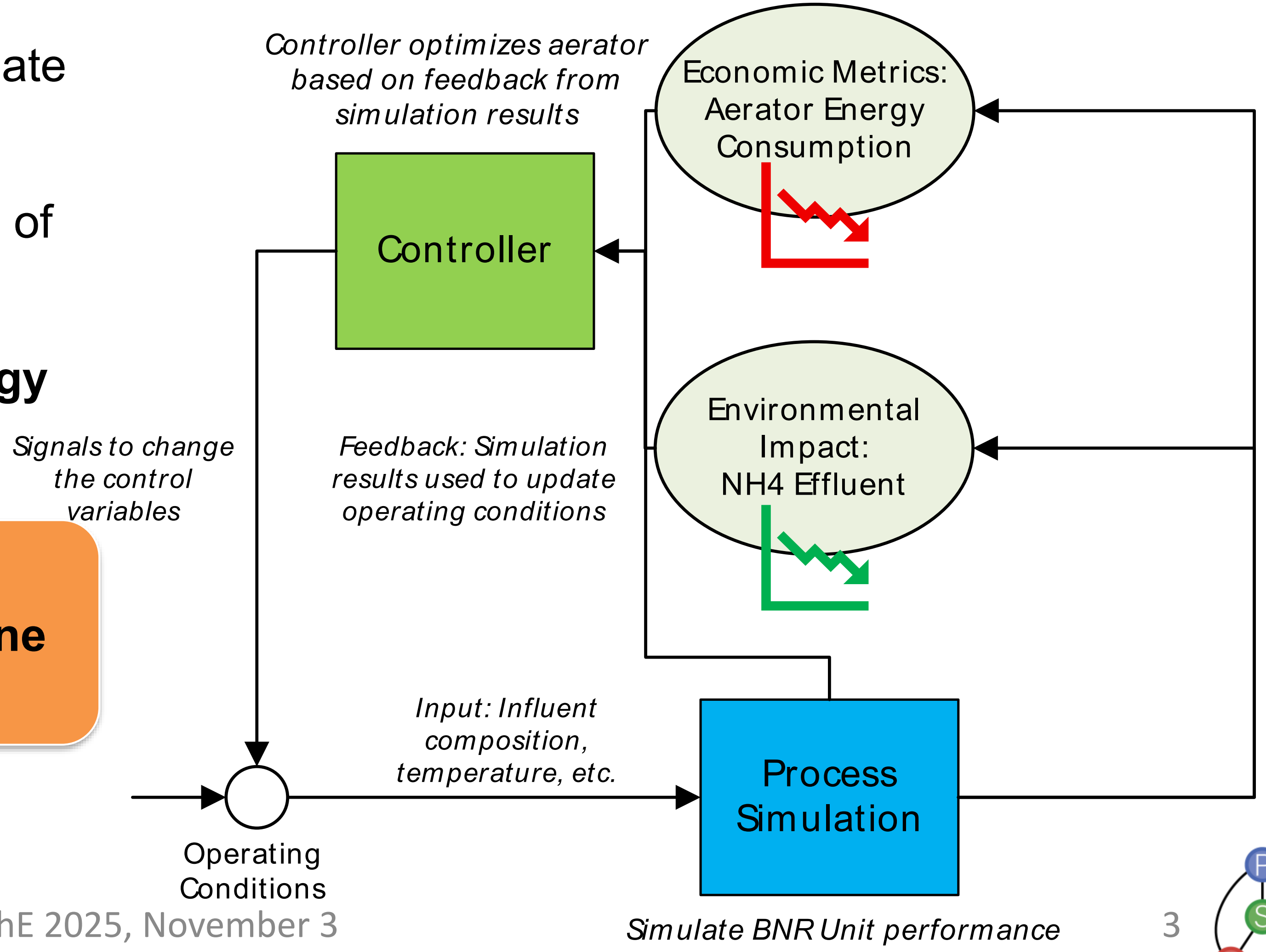
[1] Tian, H., Pan, N., Thompson, R. L., Canadell, J. G., Suntharalingam, P., Regnier, P., ... & Zhu, Q. (2024). Global nitrous oxide budget (1980–2020). *Earth System Science Data*, *16*(6), 2543-2604.

[2] Drewnowski, J., Remiszewska-Skwarek, A., Duda, S., & Łagód, G. (2019). Aeration process in bioreactors as the main energy consumer in a wastewater treatment plant. Review of solutions and methods of process optimization. *Processes*, *7*(5), 311.

Objective: Toward Sustainable WWTP Operation

- Develop an **ASM3-based MPC** to regulate effluent **NH_4** via aeration control.
- Validate the **closed-loop performance** of the integrated model–optimizer system.
- Extend to an **eMPC** that balances **energy use** and **treatment efficiency**.

Our framework unifies modeling, optimization, and sustainability in one closed-loop system.



Toolbox: Modeling and Optimization in Julia

ModelingToolkit.jl

EOptInterface.jl

JuMP.jl

- Acausal, equation-based modeling in Julia.
- Enables symbolic formulation and simplification.
- Ideal for dynamic process simulation.

- Domain-specific optimization language in Julia.
- Supports nonlinear and mixed-integer solvers.
- Integrates with automatic differentiation tools.

[3] Ma, Y., Gowda, S., Anantharaman, R., Laughman, C., Shah, V., & Rackauckas, C. (2021). Modelingtoolkit: A composable graph transformation system for equation-based modeling. *arXiv preprint arXiv:2103.05244*.
[4] Lubin, M., Dowson, O., Garcia, J. D., Huchette, J., Legat, B., & Vielma, J. P. (2023). JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*, *15*(3), 581-589.
[5] https://github.com/PSORLab/EOptInterface.jl

EOptInterface: Connecting ModelingToolkit and JuMP

EOptInterface **automaticall**
into JuMP constraints.

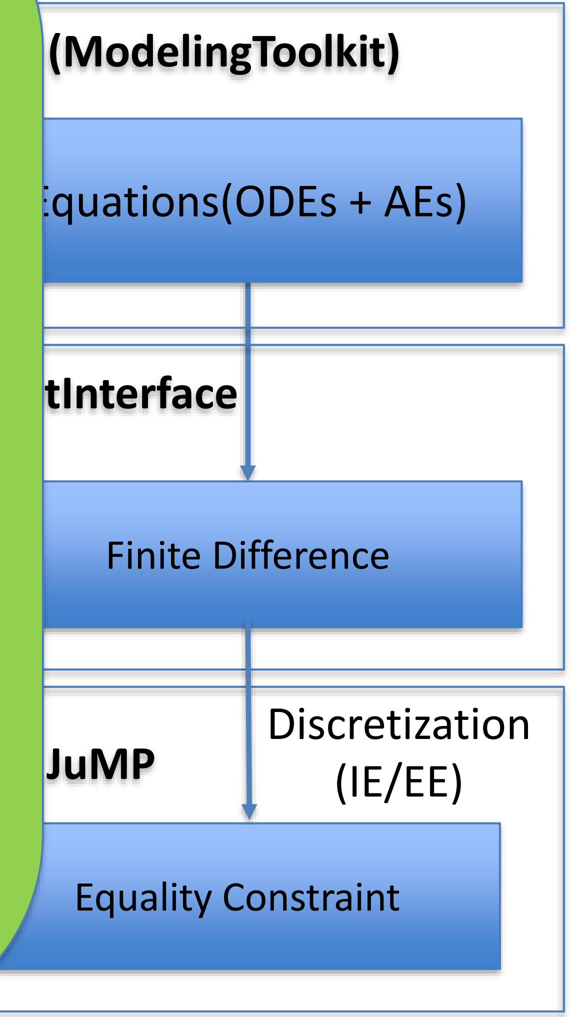
Core Functions:

- ➢ **Registers ODE systems**
  (Explicit / Implicit Euler)
- ➢ **Transforms differential**
  equality constraints.
- ➢ **Maintains consistency**
  JuMP variables.

EOptInterface.jl

Dimitri Alston (Main developer)
Optimization Abstraction Layer for
Acausal/Equation-Oriented Models in Julia
Nov 5, 2025

EOptInterface.jl

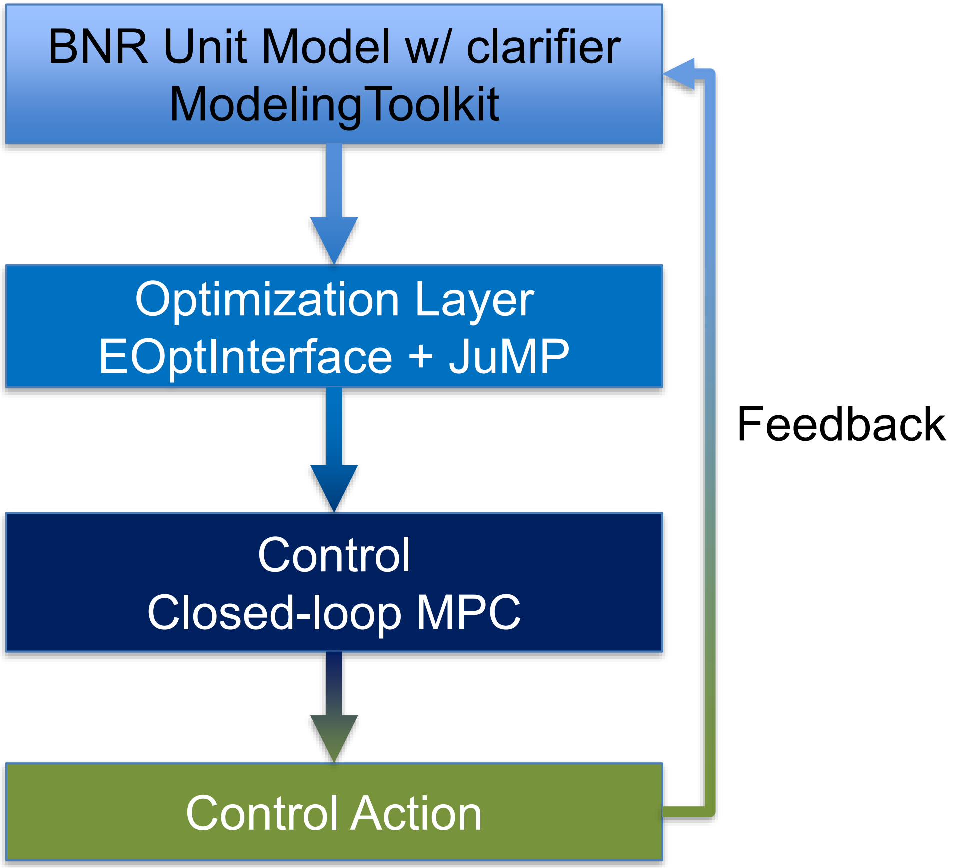


Framework Overview: From Model to Closed-loop Control

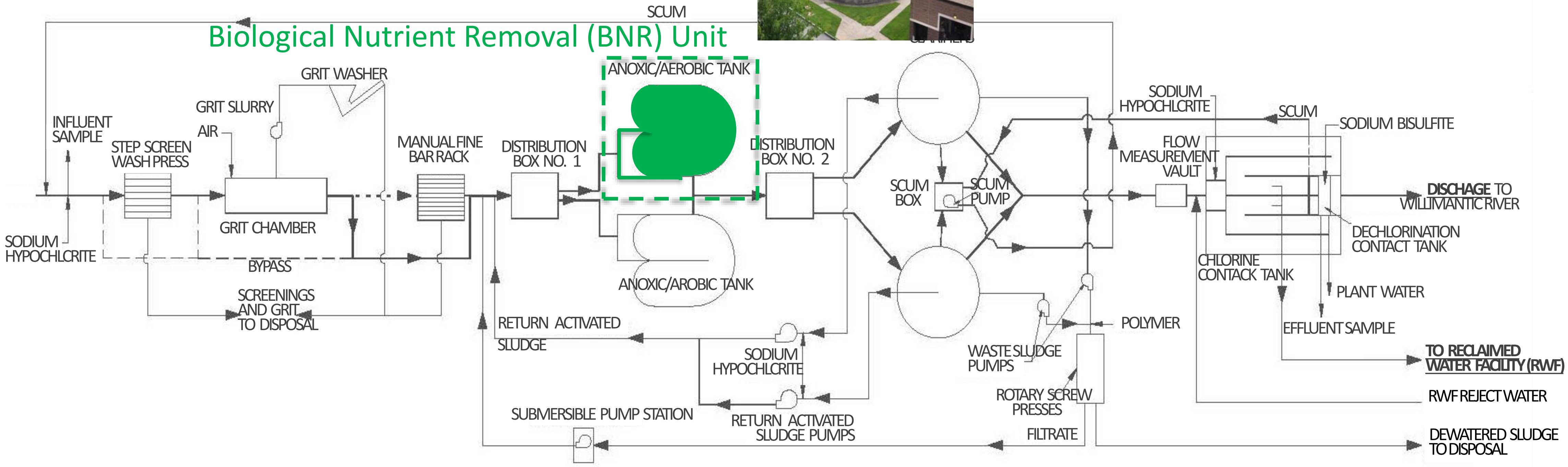
- **Mechanistic Model** (ASM3 in *ModelingToolkit*): builds the process-level ODE system.
- **Optimization Layer** (*EOptInterface* + *JuMP*): converts model equations into constraints for MPC
- **Closed-loop Control**: executes real-time MPC using updated sensor feedback.

Fully integrated feedback between simulation and optimization.

Our framework integrates modeling, optimization, and control into one unified, feedback-driven architecture.

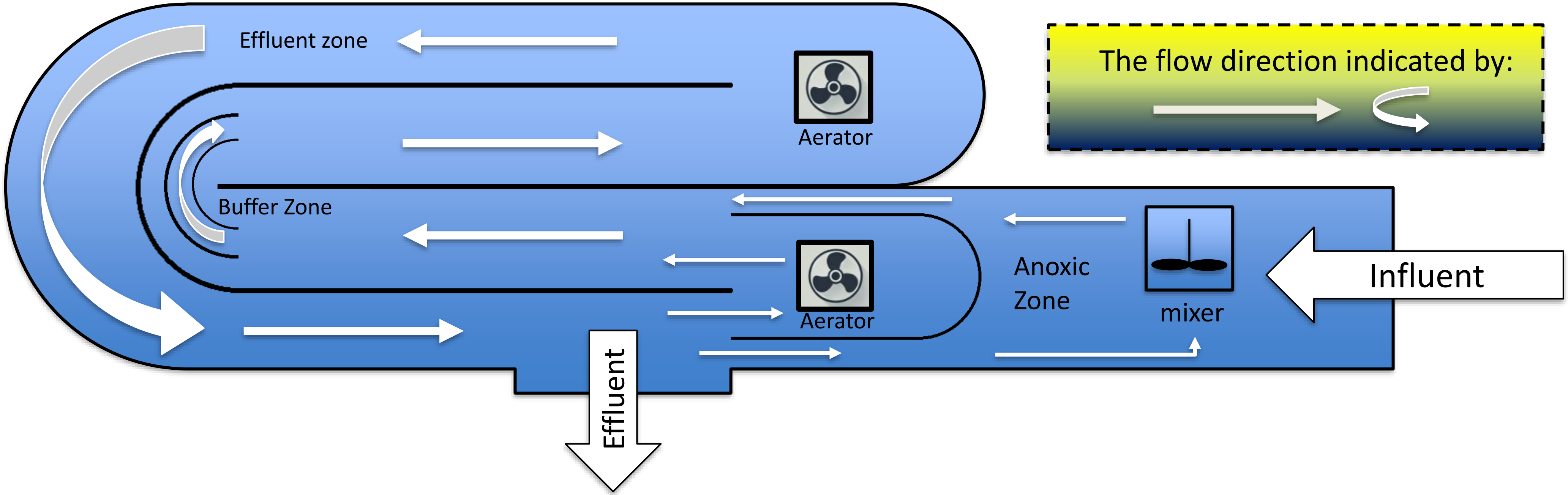


BNR System & Process Layout

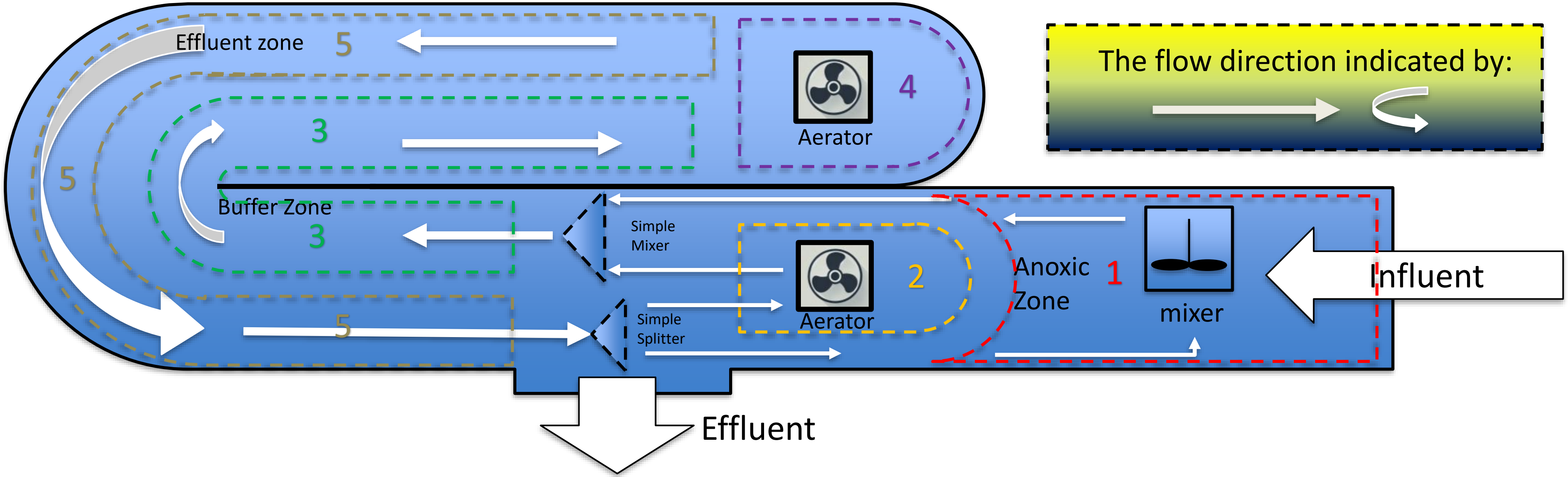


UConn Wastewater Treatment Plant (WWTP) Process Flow Diagram

Modeling of BNR Unit: Diagram



Modeling of BNR Unit: Divided Zones and ODEs



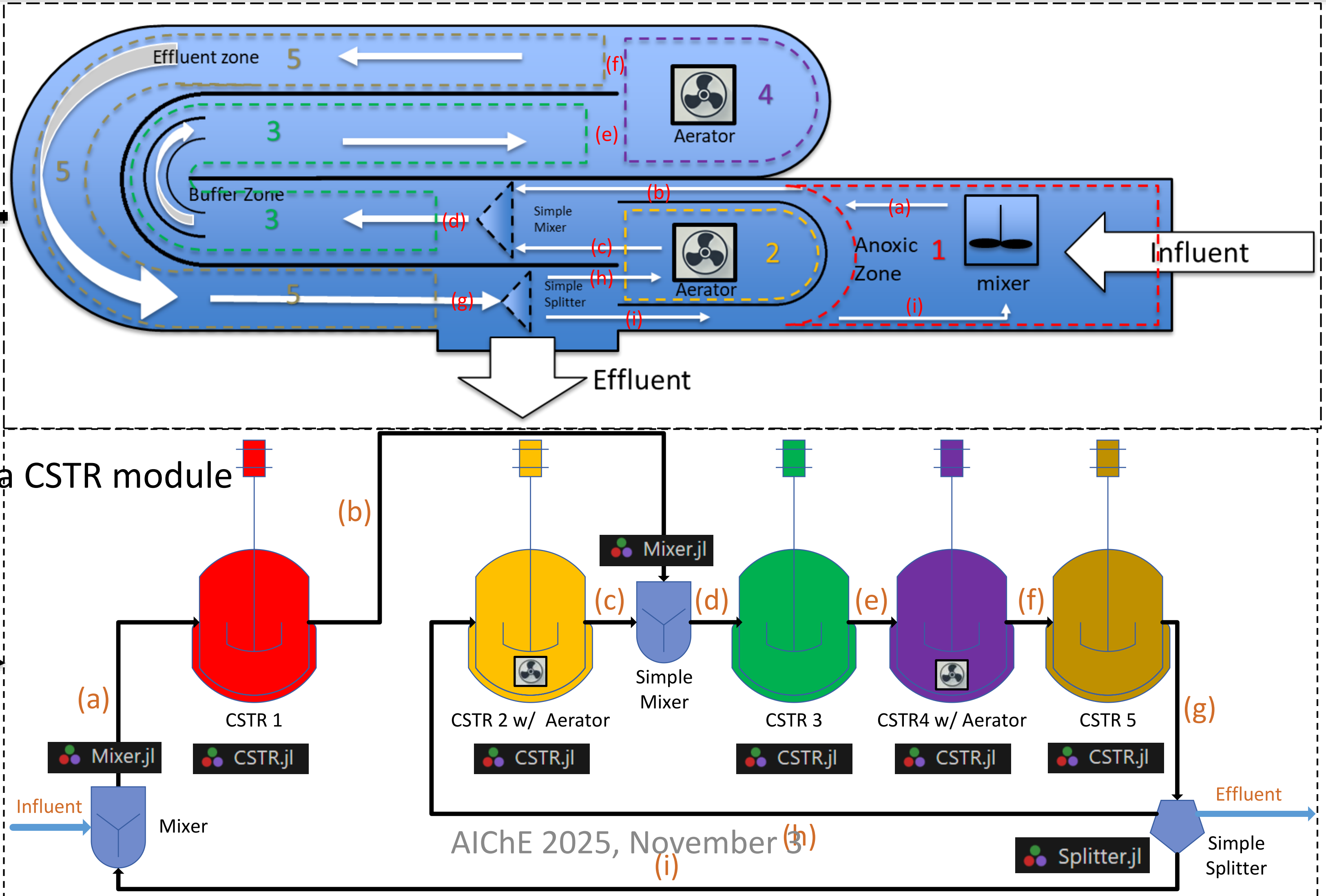
ODEs for State Variables $\mathbf{c}(t)$ in CSTRs:

$$\frac{d\mathbf{c}}{dt} = \mathbf{c}_{\text{in}}(t) - \mathbf{c}(t) + \mathbf{r}_{\text{ASM3}}(\mathbf{c}(t), t)$$

Modeling of BNR Unit – From Physical Layout to Symbolic Flowsheet

Abstraction

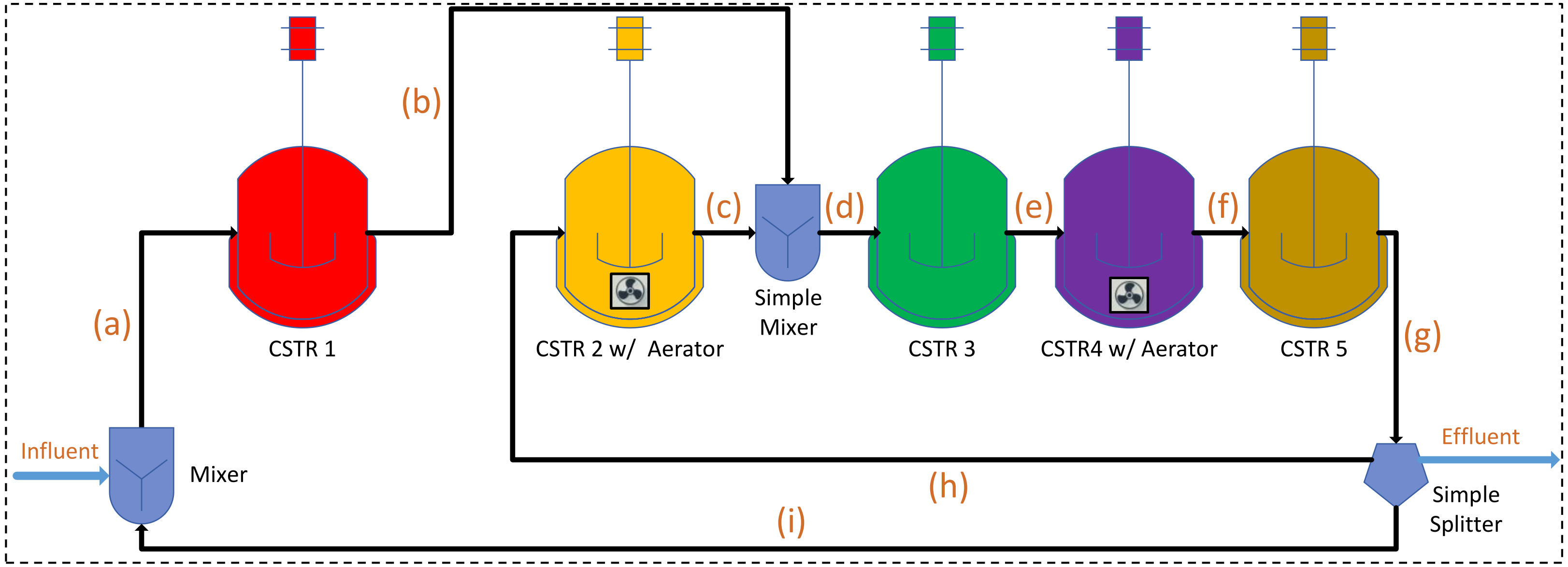
Each zone represented as a CSTR module



MaterialStream.jl

ModelingToolkit.jl

Hierarchical Modeling of the BNR Flowsheet

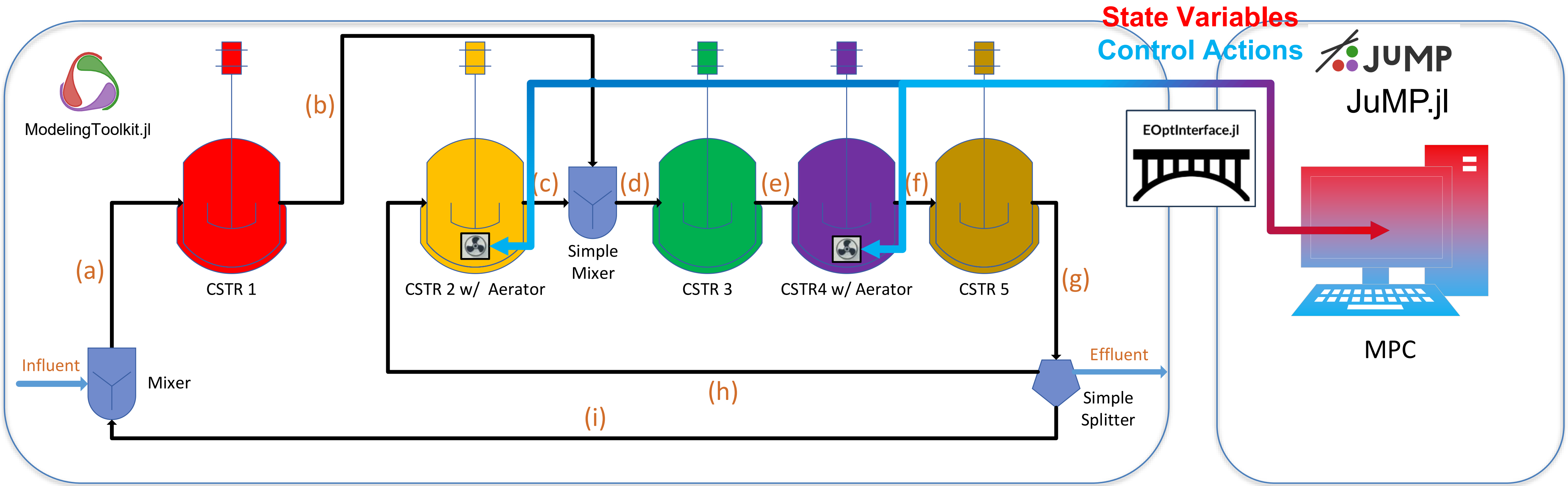



```julia
@mtkmodel WRRFPlantModel begin

    @components begin
            # Influent = DynamicInfluent()
            Influent = DynamicInfluent2()
            reactor1 = CSTR(vol=500, switch=0, Temp = inlet_temp)
            reactor2 = CSTR(vol=500, switch=0, Temp = inlet_temp)
            reactor3 = CSTR(vol=500, switch=1, Temp = inlet_temp)
            reactor4 = CSTR(vol=500, switch=0, Temp = inlet_temp)
            reactor5 = CSTR(vol=500, switch=1, Temp = inlet_temp)
            splitter1 = Splitter_1in_2out(out_factor_1 = 0.5)
            clarifier = Clarifier(R_1 = 0.4, w_1 = 0.03)
            mixer1 = Mixer_2in_1out()
            mixer3 = Mixer_2in_1out()
    end

    @equations begin
        connect(Influent.port,mixer1.In1)
        connect(splitter1.Out2,mixer1.In2)
        connect(mixer1.Out1, mixer3.In2)
        connect(clarifier.recycle_stream, mixer3.In1)
        connect(mixer3.Out1,reactor1.In)
        connect(reactor1.Out,reactor2.In)
        connect(reactor2.Out,reactor3.In)
        connect(reactor3.Out,reactor4.In)
        connect(reactor4.Out,reactor5.In)
        connect(reactor5.Out,splitter1.In)
        connect(splitter1.Out1, clarifier.inlet_stream)
    end
end

@mtkcompile sys = WRRFPlantModel()
```

ModelingToolkit.jl

- Each reactor, mixer, and splitter is a reusable ModelingToolkit component.
- All units are connected symbolically using simple syntax to form the BNR flowsheet.
- Mass and energy balances are enforced automatically.
- The full model runs directly for steady-state and dynamic simulation.

Adding MPC to the Flowsheet



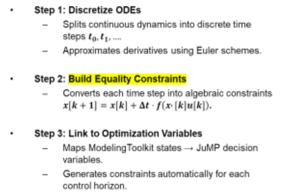
- The assembled model communicates directly with the MPC optimizer.
- **Control actions (Aerator)** are updated at each step using system feedback.
- This integration enables closed-loop dynamic control under the ASM3 framework.

MPC: Automating Discretization and Constraint Generation

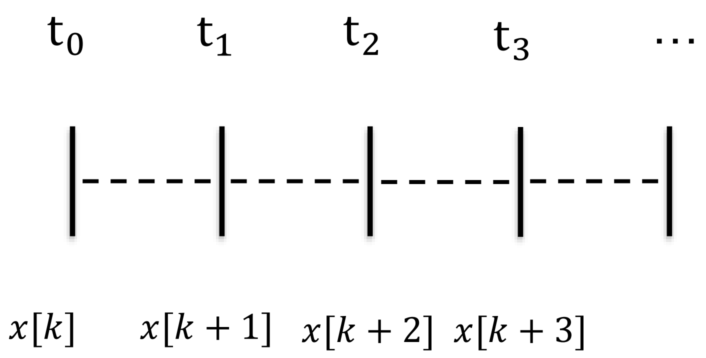
- **Step 1: Discretize ODEs**
  - Splits continuous dynamics into discrete time steps $\boldsymbol{t_0}, \boldsymbol{t_1}, ...$
  - Approximates derivatives using Euler schemes.

- **Step 2: Build Equality Constraints**
  - Converts each time step into algebraic constraints $\boldsymbol{x}[\boldsymbol{k}+\mathbf{1}] = \boldsymbol{x}[\boldsymbol{k}] + \boldsymbol{\Delta t} \cdot \boldsymbol{f}(\boldsymbol{x}, [\boldsymbol{k}]\boldsymbol{u}[\boldsymbol{k}])$.

- **Step 3: Link to Optimization Variables**
  - Maps ModelingToolkit states → JuMP decision variables.
  - Generates constraints automatically for each control horizon.



Step 1: Discretize ODEs
Splits continuous dynamics into discrete time steps $t_0, t_1, ...$
Approximates derivatives using Euler schemes.
Step 2: Build Equality Constraints
Converts each time step into algebraic constraints $x[k+1] = x[k] + \Delta t \cdot f(x\cdot[k]u[k])$.
Step 3: Link to Optimization Variables
Maps ModelingToolkit states → JuMP decision variables.
Generates constraints automatically for each control horizon.



$$x[k+1] = x[k] + \mathrm{D}t \times f(x[k], u[k])$$

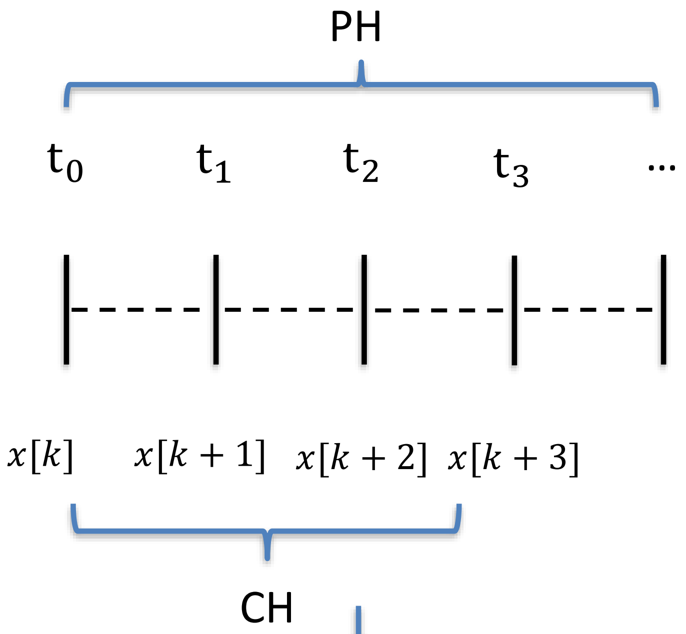
Time discretization with IE/EE to build equality constraints

MPC: Architecture

- **Prediction horizon (PH):** future simulation window based on ASM3 dynamics.
- **Decision variables:** oxygen transfer coefficients (kLa_2, kLa_4)
- **Control horizon (CH):** number of steps where aeration (kLa_2, kLa_4)
- **Objective function:**

$$J = w_1 \sum_{i=1}^{PH} \left| y_{NH_4}(kLa_{2,i}, kLa_{4,i}, X_i) - y_{\text{target}, NH_4, i} \right|^2 + w_2 \sum_{j=1}^{CH-1} \left(|\Delta kLa_{2,j}|^2 + |\Delta kLa_{4,j}|^2 \right)$$

- The optimization runs recursively at each step using new sensor feedback.



Measurement Update → Optimization → Apply first move (Kla)

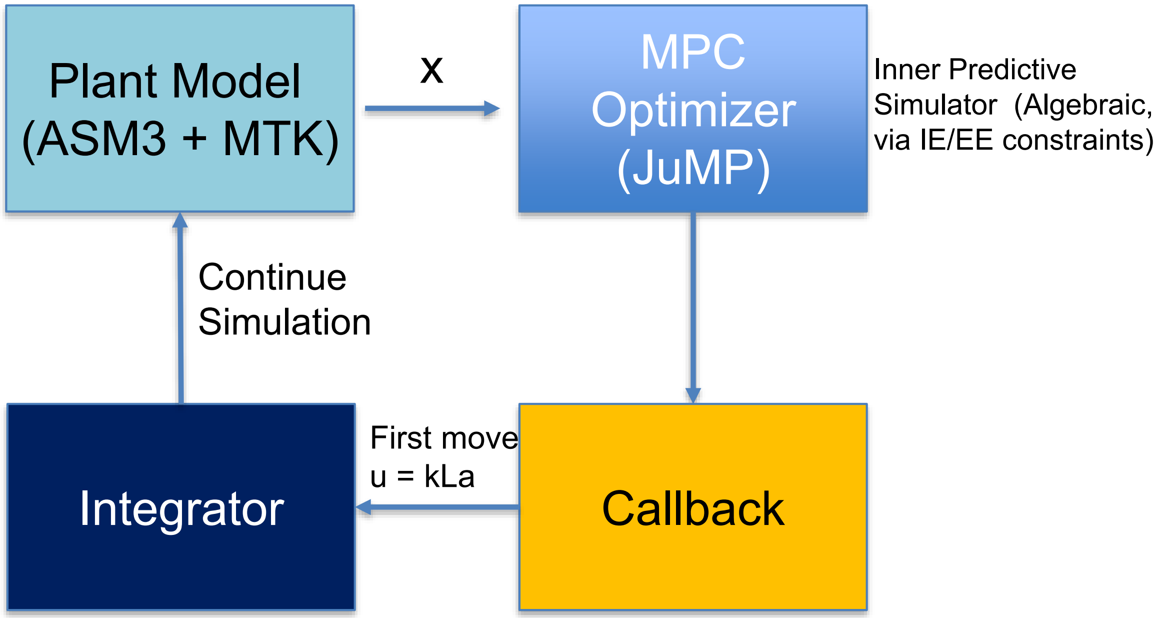
Objective: Track NH_4 while smoothing control.

Closed-loop MPC Execution in Real Time

Algorithm 1 Closed-loop MPC: Code-level Cycle

Initialize:
Build ASM3 model in `ModelingToolkit`
Create outer ODE integrator (continuous simulation)
Register ODE system with `EOptInterface` (IE/EE discretization)
Define optimization model and objective J
while simulation time $t < T_{\text{end}}$ **do**
Read current state x_k from outer integrator
Pin initial condition in JuMP: $x_{\text{pred}}[1] \leftarrow x_k$
Discretize and predict over horizon P_H
Solve MPC optimization problem
Apply first control move $u_k = (KLa_2, KLa_4)$
Advance outer integrator by Δt to obtain x_{k+1}
Feed back x_{k+1} and repeat
end while
Output: effluent NH_4 trajectory, control profiles, energy indicators

Real-time coupling of dynamic simulation and optimization.



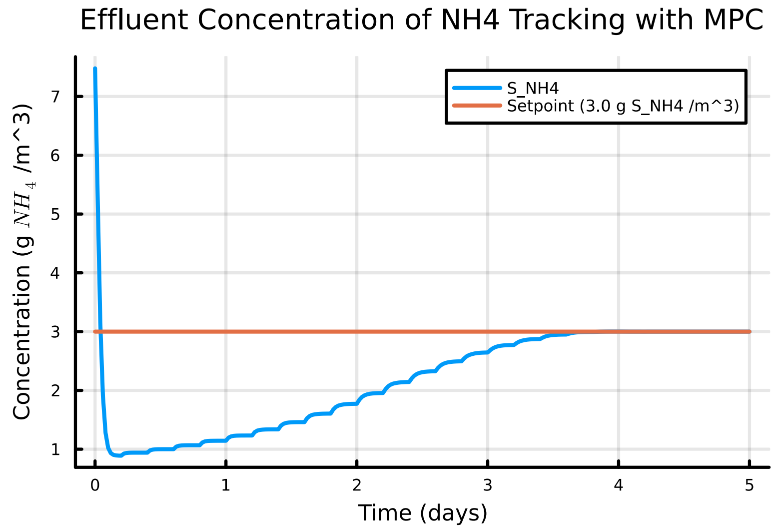
Real-time MPC coupling achieves dynamic consistency between model prediction and plant response.

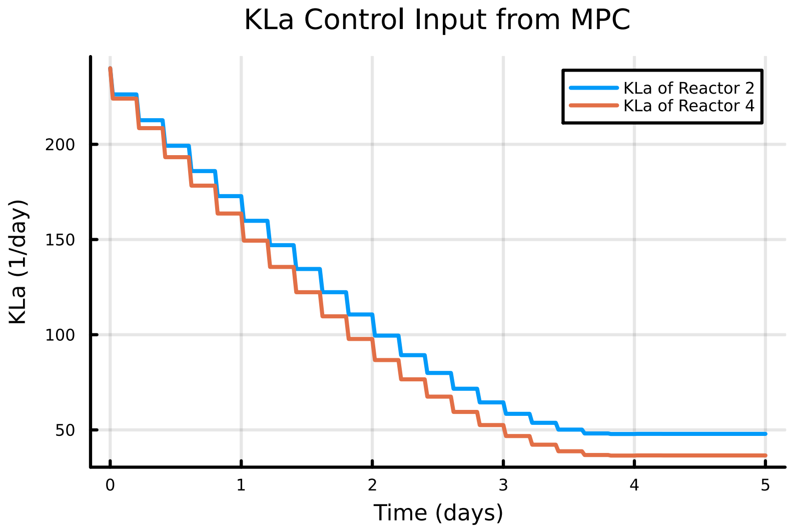
NMPC: Setpoint Tracking Performance

Goal: Regulate effluent NH_4 by adjusting aeration rates (kLa_2, kLa_4).

Result: Smooth convergence to 3.0 g NH_4/m^3 within 4 days, with no overshoot.

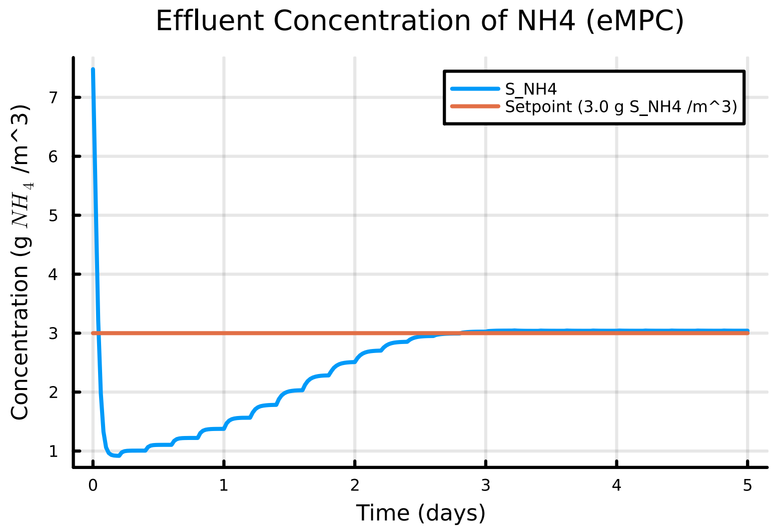
Insight: Stable, realistic control actions confirm robust MPC–simulation coupling.

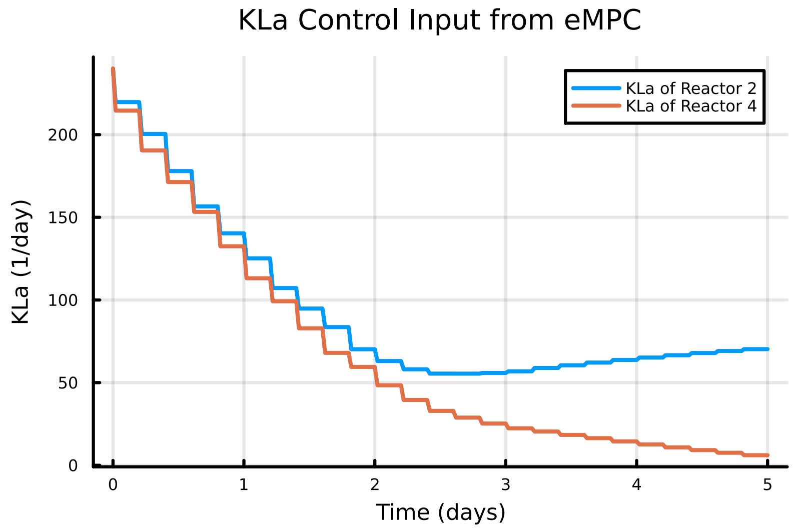




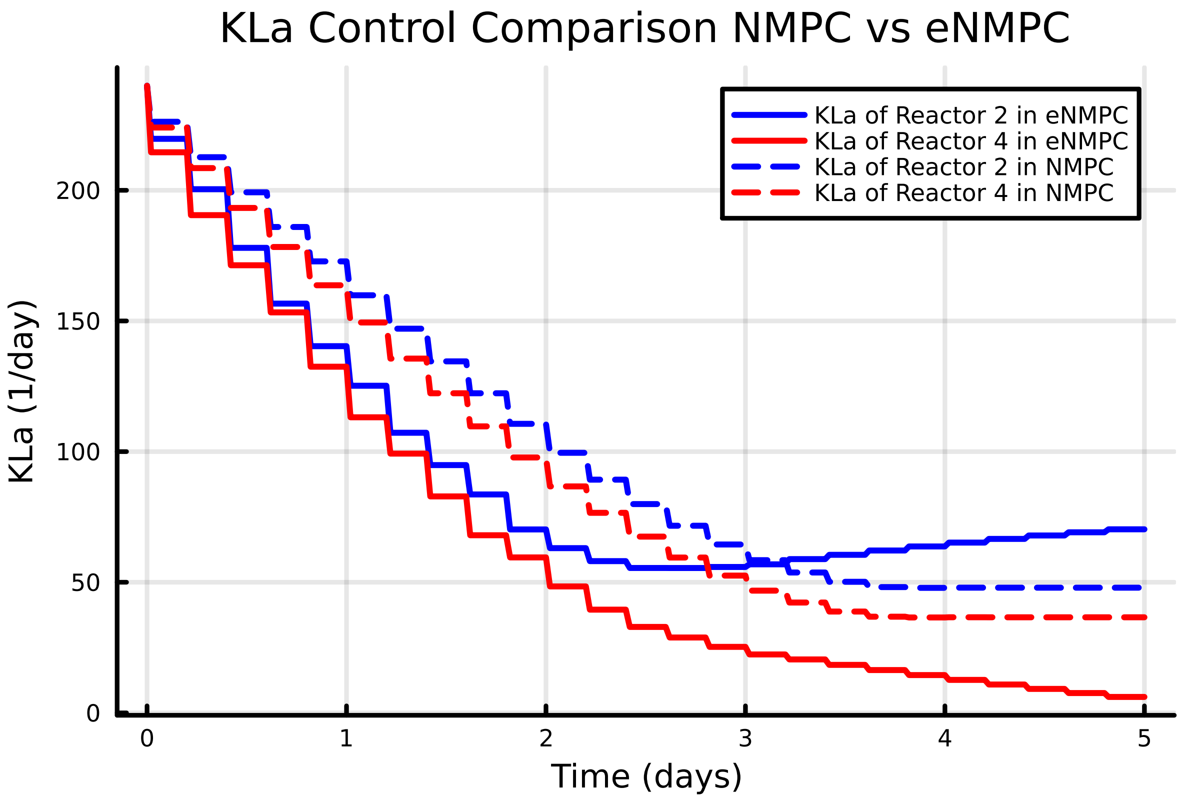
eNMPC Performance

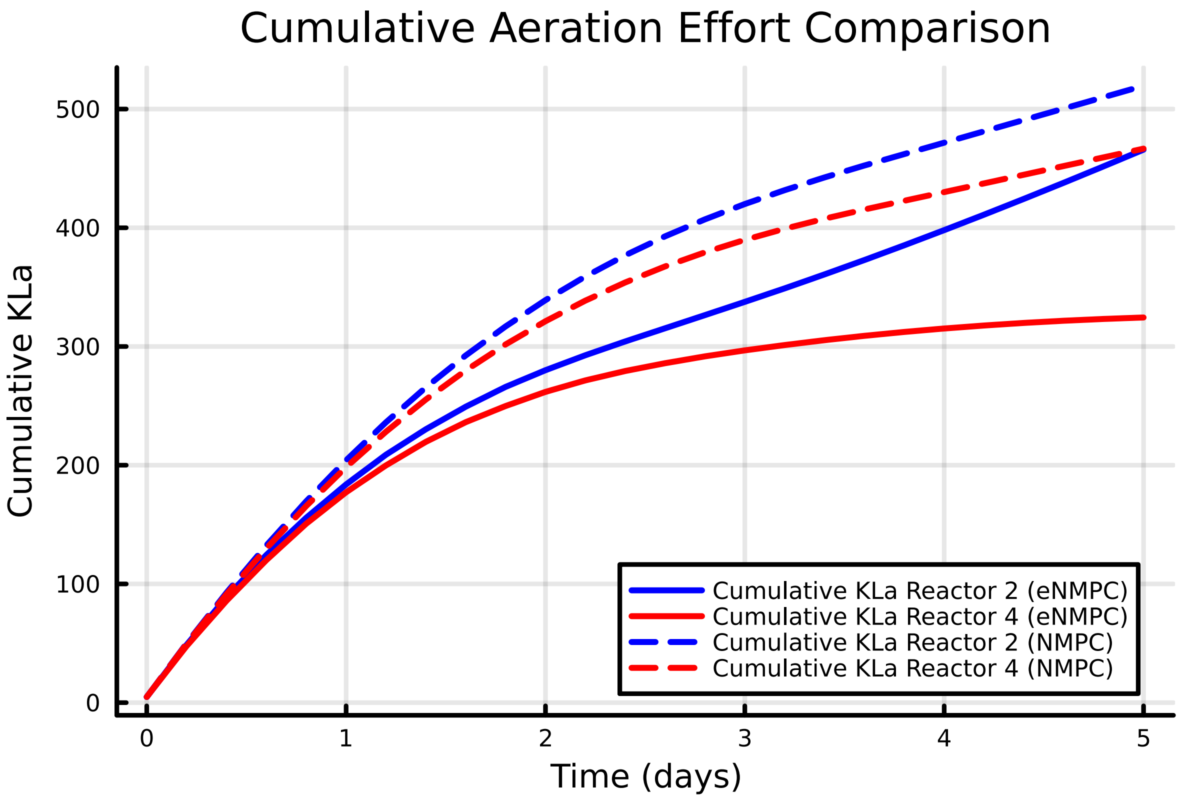
- **Goal:** Add an economic layer balancing energy use and sustainability.
- **Effect:** Slightly higher aeration improves NH_4 stability and reduces GHG impact.
- **Insight:** Establishes the energy–environment trade-off frontier for future N_2O integration.





Energy Efficiency: eNMPC vs NMPC

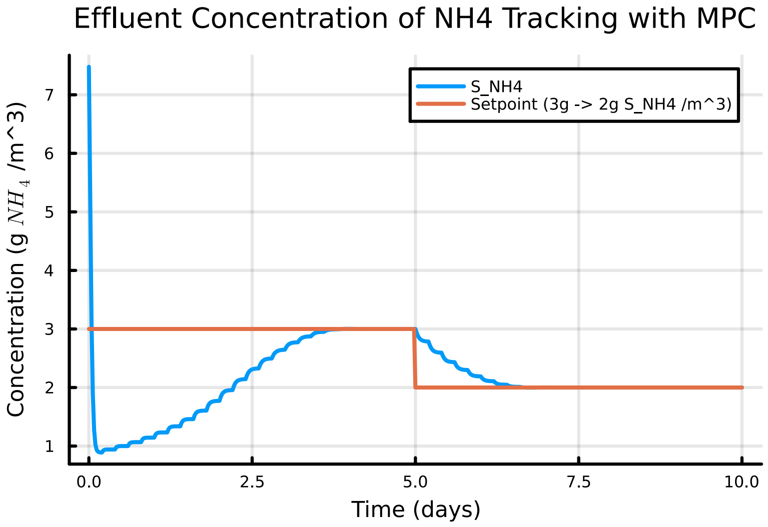


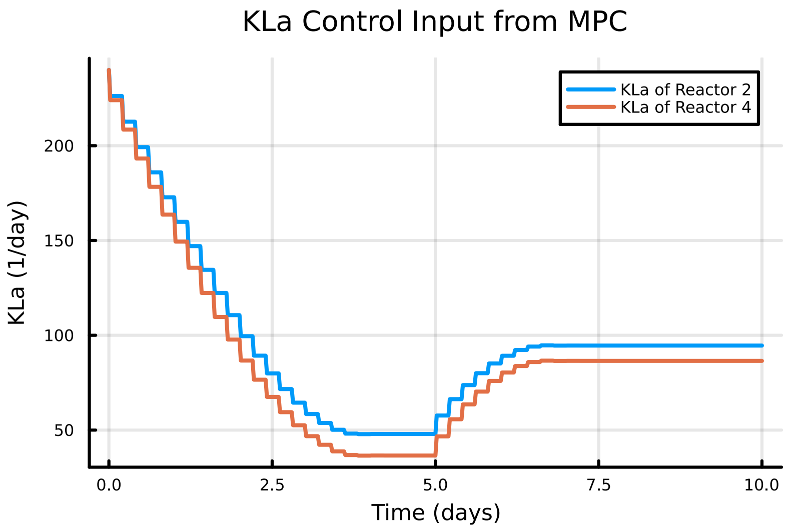


Cumulative KLa serves as a proxy for aeration energy use.

Controller Robustness and Practical Feasibility: NMPC

- **Test:** Step change (NH_4: 3 → 2 mg/L) shows smooth and stable transition.
- **Control:** kLa_2, kLa_4 adjust gradually within feasible limits.
- **Result:** Stable and energy-efficient response under realistic disturbances.





The NMPC framework stays feasible, stable, and efficient under varying setpoints and loads

Conclusions

- The ASM3-based eMPC framework successfully integrates mechanistic modeling, optimization, and control in a unified Julia environment.
- EOptInterface automates ODE discretization and constraint generation, enabling real-time closed-loop simulation.
- The controller achieves regulation of effluent NH_4 concentration.
- The framework remains feasible and robust under realistic operating variations, including setpoint changing.

Outlook

- Incorporate GHG emission objectives (N_2O-inclusive optimization).
- Introduce time-varying influent and recycle conditions for real-plant scenarios.

Acknowledgements

Members of the Process Systems and Operations Research Laboratory at the University of Connecticut (https://www.psor.uconn.edu)

UCONN UNIVERSITY OF CONNECTICUT
Process Systems and Operations Research Laboratory

2025 AIChE ANNUAL MEETING

Pratt & Whitney Institute for Advanced Systems Engineering UNIVERSITY OF CONNECTICUT

U.S. DEPARTMENT of ENERGY

Funding:

Department of Energy, Award No.: DE-EE0010991

Pratt & Whitney Endowed Professorship

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Department of Energy or the United States Government.