Systems & Control Transactions

Original Research Article
Peer Reviewed Conference Proceeding
ESCAPE 36 - European Symposium on Computer Aided Process Engineering
Sheffield, UK. 21-24 June 2026

PSE PRESS

OpenAD-lib: Open-Source Framework for Uncertainty-Aware Anaerobic Digestion Digital Twins

Benaissa Dekhici[ab], Rohit Murali[a], and Michael Short[ab]*

[a] School of Chemistry and Chemical Engineering, University of Surrey, Guildford, GU2 7XH, Surrey, United Kingdom
[b] Supergen Bioenergy Impact Hub, Energy and Bioproducts Research Institute, UK
* Corresponding Author: m.short@surrey.ac.uk.

ABSTRACT

This paper presents OpenAD-lib, an open-source Python framework for anaerobic digestion (AD) digital twins, unifying mechanistic models, machine learning (ML) surrogates, and model predictive control (MPC) within a modular ecosystem. OpenAD-lib addresses the critical fragmentation in AD digitalisation by bridging mechanistic and data-driven paradigms under explicit uncertainty. By integrating uncertainty-aware feedstock characterisation with robust process control, the platform enables the transition from isolated research tools to fully integrated digital twins, delivering economic and environmental value in AD systems.

Keywords: Digitalisation, Bioenegry, Anaerobic digestion, Digital twins, Uncertainty quantification, Model Predictive Control, Machine learning, Open-source framework

INTRODUCTION

Anaerobic digestion (AD) is critical to the UK's energy transition and circular economy, producing renewable biogas and nutrient-rich digestates that enhance soil health and carbon sequestration [5]. However, most AD facilities lack production-ready computational tools integrating mechanistic modelling with data-driven methods, resulting in passive monitoring, sub-optimal biogas recovery, and underutilised operational potential.

Since the introduction of Anaerobic Digestion Model No. 1 (ADM1) in 2002 [2], mechanistic AD modelling has advanced substantially, with ADM1 establishing the international reference standard for capturing complex substrate heterogeneity, microbial dynamics, and inhibitory mechanisms across 32 state variables [1]. However, four interconnected limitations impede sector digitalisation. First, the AD modelling landscape remains fragmented [5]. No unified platform integrates mechanistic models, ML surrogates, and control strategies. Practitioners assemble tools ad-hoc with divergent input/output conventions, and the absence of standardised feedstock descriptors across six documented characterisation methodologies for the ADM1 inputs [13] necessitates resource-intensive site-specific calibration, preventing model transfer between facilities. Second, existing tools deliver deterministic predictions without quantifying uncertainty or confidence bounds. Real-time operational decisions including biogas commitments and control setpoints require probabilistic guarantees rather than nominal estimates. Third, mechanistic and data-driven components operate in isolation. Recent advances including Long Short-Term Memory networks (LSTM) surrogates [9], Physics-Informed Machine Learning (PIML)[12], and multi-task Gaussian processes [4] (MTGP) demonstrate that hybrid approaches achieve both computational efficiency and mechanistic fidelity, yet no integrated commercial or open-source platform exists, forcing practitioners to choose between high-fidelity simulation or fast data-driven methods [5]. Finally, scattered AD models [4, 7, 9, 12] across disconnected repositories lack unified access, standardised comparison frameworks, and industrial-grade implementation practices required for operational deployment.

OpenAD-lib, developed within the Supergen Bioenergy Impact Hub (SBIH) [11] is an open-source Python framework that bridges mechanistic understanding with data-driven approaches under explicit uncertainty. The framework integrates feedstock characterisation, mechanistic models, ML surrogates, and control strategies in seamless workflows (unified platform), well-documented, comprehensively tested, and designed for industrial extensibility (production-ready), propagates uncertainty explicitly from feedstock characterisation

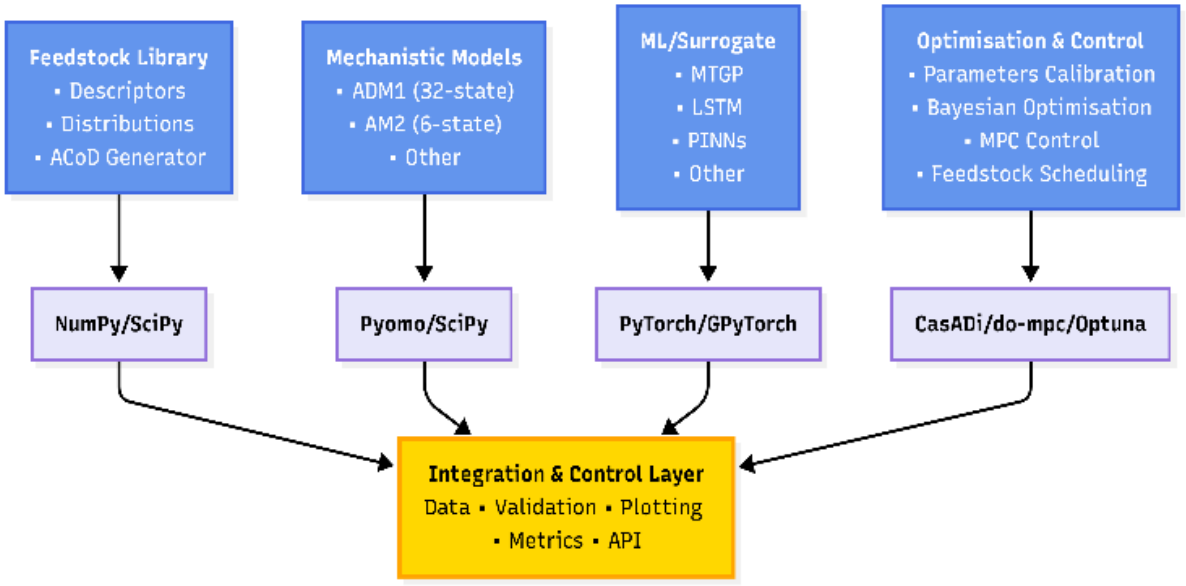

Figure 1. OpenAD-lib four-layer modular architecture.

through mechanistic simulation, surrogate approximation, and control optimisation to enable quantified confidence in operational decisions (uncertainty-aware by design), and enables users to select components (ADM1-only, surrogate-only, or hybrid approaches) matching their computational and accuracy requirements, avoiding unnecessary complexity while enabling gradual adoption (modular composability). Beyond AD, OpenAD-lib is designed for extensibility to complementary bioenergy pathways (gasification, pyrolysis) aligned with SBIH's partnership ecosystem.

The paper is organised as follows: Section 2 presents the software architecture, Section 3 describes key features and implementation, Section 4 demonstrates a case study, and Section 5 provides conclusions and future work.

SOFTWARE ARCHITECTURE

OpenAD-lib implements a four-layer modular architecture that enables uncertainty propagation from feedstock characterisation through process prediction to optimisation and control. Figure 1 presents the layered technology stack of OpenAD-lib, demonstrating how the framework leverages Python's scientific computing ecosystem to integrate each layer within a cohesive software architecture. This layered design philosophy reflects a systems level approach to digitalisation of AD processes, enabling users to employ components independently or in combination to accommodate diverse use cases ranging from offline feasibility analysis to real-time adaptive control:

- **Layer 1 -Feedstock Characterisation Library:** Encodes statistical representations of feedstock properties.
- **Layer 2 -Mechanistic Modelling:** Wraps standard solvers for ADM1 (32-states) and AM2 (reduced-order only 6-states) simulations.
- **Layer 3 -ML/Surrogate Models:** Provides pre-configured PyTorch modules for fast inference.

Layer 4- Optimisation and Control: Integrates optimization solvers for calibration and MPC.

The modular interconnection of these layers is orchestrated through a unified *Integration & Control Layer* that manages data flow and uncertainty propagation across all operational modes. The OpenAD-lib can be accecced in Github via the following link: https://github.com/benmola/OpenAD-lib

IMPLEMENTATION AND KEY FEATURES

Feedstock Characteristics Library

Reliable characterisation of feedstock properties is fundamental to predictive accuracy in AD modelling [6]. OpenAD-lib addresses this through a structured feedstock library that captures the variability inherent in real-world substrates and translates readily available measurements into the required inputs for the mechanistic models. The library integrates comprehensive representations of diverse feedstock types commonly encountered in typical UK AD sites.

Table 1: OpenAD-lib Feedstock Library: Data Coverage Summary

Feedstock Category	Included Feedstocks
Agricultural/ Energy Crops	Maize Silage, Wholecrop Cereals, Grass, Fodder Beet
Agro-industrial	Rice Bran, Corn Gluten/Rapemeal, Crimped Wheat
Food Processing	Lactose (Whey), Apple Pomace, Potatoes
Manure & Residues	Chicken Litter, Animal Manure

Table 2: Standardised descriptor set for uncertainty-aware feedstock characterisation.

Symbol	Descriptor	Unit
TS	Total Solids	Mass % or kg m^{-3}
VS	Volatile Solids	g kg^{-1} TS or kg m^{-3}
COD_t	Total COD	kg COD m^{-3}
$COD_{s,p}$	Soluble / Particulate COD	kg COD m^{-3}
BMP	Biochemical Methane Potential	NmL CH_4 kg^{-1} VS
f_{bio}	Biochemical Composition	g kg^{-1} or kg m^{-3}
$C:N$	Carbon-to-Nitrogen Ratio	Molar ratio
TAN	Total Ammonia Nitrogen	g N m^{-3}
V_{vfa}	Volatile Fatty Acids	kg m^{-3}

Feedstock Characterisation and Uncertainty Representation

The OpenAD-lib feedstock library consists of 12+ substrates consitsting of agricultural residues, energy crops, agro-industrial by-products and food waste streams as shown in Table 1. To capture the inherent variability of biological substrates, OpenAD-lib encodes feedstock properties using the standardised descriptor set detailed in Table 2. Unlike deterministic approaches, the framework models these inputs as probabilistic distributions ($x_{feedstock} \sim \mathcal{P}(\theta)$), parametrised via Maximum Likelihood Estimation (MLE) using data from multi-facility laboratory measurements, literature meta-analysis, and expert elicitation. To ensure physical consistency during ensemble generation, the library automatically assigns Beta distributions to bounded fractions (e.g., biodegradability coefficients) and Log-normal or Gamma distributions to strictly positive quantities (e.g., TS, VS, COD), thereby preventing non-physical sampling while facilitating the propagation of measurement uncertainty into downstream process control.

Mechanistic Modelling

The mechanistic modelling layer simulates AD dynamics using coupled ordinary differential equation (ODE) systems [1]. OpenAD-lib adopts a multi-model approach, offering both comprehensive and reduced-order formulations to balance process fidelity with computational efficiency for varying application requirements. The complex nonlinear dynamics are represented by a general state-space mass balance equation [1]:

$$\dot{x} = D(x_{in} - x) + Kr(x) \quad (2)$$

where $x \in R^n$ is the state vector of concentrations were is n is the number of the model state variables. D is the dilution rate (scalar or diagonal matrix, $\frac{Q_{in}}{V}$ where Q_{in} is the flow rate and V is the reactor volume). x_{in} is the vector of influent concentrations. K is the stoichiometric matrix containing yield coefficients. r(x) is the vector of kinetic reaction rates (Monod/inhibition functions). The mechanistic layer provides Python wrappers for the ADM1 and the reduced-order AM2 model. The engine uses SciPy/Numpy for standard simulation and Pyomo/CasADi for symbolic gradient-based optimization tasks. OpenAD-lib implements the ADM1 as the primary high-fidelity representation. The implementation follows [2, 7] comprising a system of 32 ($x \in R^{32}$) coupled ODEs that describe seven biochemical pathways. ADM1 offers exceptional detail for hypothesis testing and steady-state design [2]. However, the requirement to estimate ~100 parameters and the computational stiffness of the 32-state system make it challenging for real-time control applications or scenarios requiring thousands of iterative simulations [1, 3]. To address the control-oriented limitations of ADM1, OpenAD-lib incorporates the Anaerobic Model for Control (AMOCO), hereafter referred to as the AM2 model [3]. AM2 aggregates the complex dynamics of ADM1 into a two-step biological process (acidogenesis and methanogenesis) governed by two lumped biomass populations. The AM2 dynamics follow the same general mass balance structure as in Eq (2), but with only 6 state variables ($x \in R^6$). A major barrier in applying mechanistic models is the complex mapping required to convert routine physicochemical measurements (e.g., TS, VS, COD) into the state variables required by ADM1/AM2. To resolve this, OpenAD-lib provides a standardised framework that mathematically transforms available feedstock data into stoichiometrically consistent ADM1 input fractions, ensuring mass conservation (COD, Nitrogen) across the model interface [6]. The framework also supports other widely-adopted characterisation protocols depending on available data fidelity [13]. For facilities operating with substrate mixtures, the library implements a mixing module that aggregates individual feedstock descriptors. Once

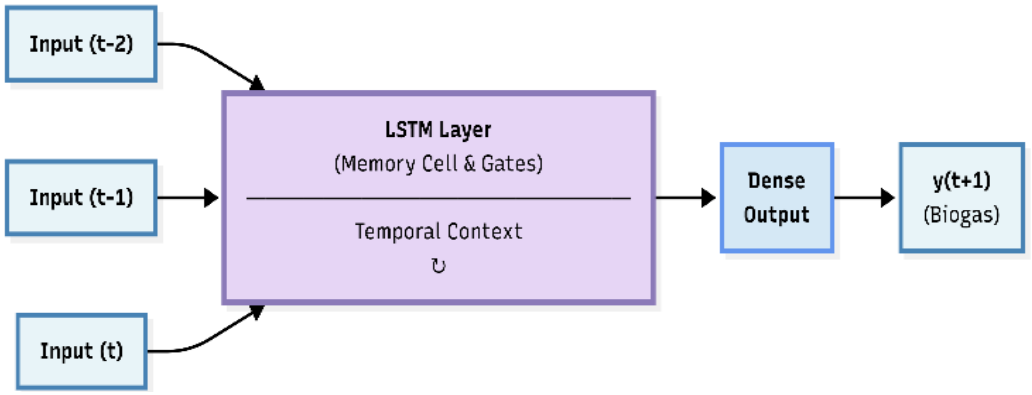

Figure 2. LSTM Architecture (Temporal Forecasting) [9].

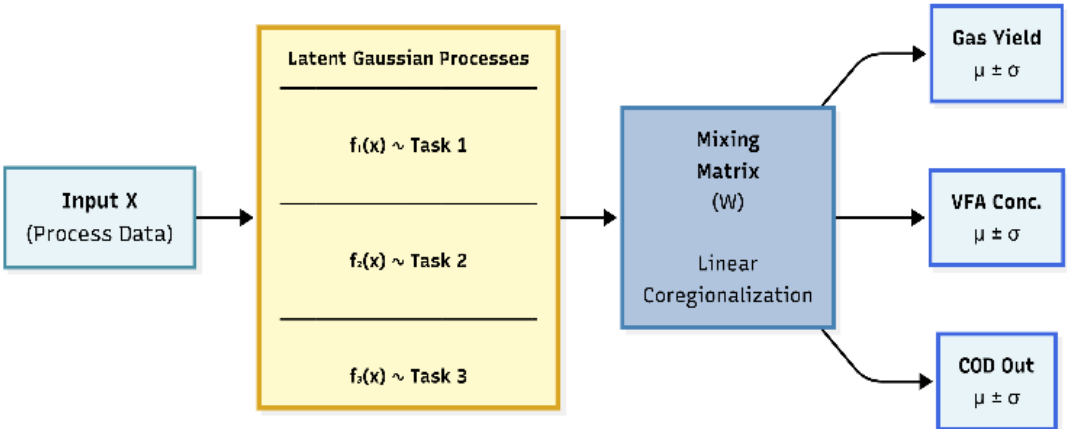

Figure 3. MTGP Architecture (Multi-OUTPUT & Uncertainty) [4].

characterised, the composite influent state vector $x_{mixture}$ is computed via weighted averaging based on volumetric or mass flow ratios (w_i) :

$$x_{mixture} = \sum_{i=1}^{N} w_i x_i \quad \text{with} \quad \sum_i w_i = 1 \tag{3}$$

This formulation preserves the mass balance of all state variables while naturally propagating the uncertainty from individual substrate characterisations to the final blended influent. By providing a single, standardised interface to multiple established characterisation protocols, OpenAD-lib removes the historical barrier whereby practitioners implemented ad-hoc, undocumented conversions from measurements to ADM1 inputs, often introducing unquantified error and reducing model transparency [6, 13].

ML/Surrogate Models

The ML surrogate approximation problem is formulated as learning an efficient mapping:

$$\hat{y} = \mathcal{S}(x; \phi) \tag{4}$$

where $\mathcal{S}$ denotes the learned surrogate function parameterised by ϕ (e.g. neural network weights, Gaussian process hyperparameters), and the ML surrogate is trained to approximate outputs $\hat{y}$ (e.g. cumulative methane/biogas production, VFA concentration trajectory, digestate composition) given inputs x (real system data such as feedstock composition and feeding, operating conditions). However, ML surrogates require hyperparameter tuning of the network and it remains a "black box" approach. OpenAD-lib serves as a repository for ML surrogates tailored to AD and biogas production forecasting. Currently, the framework implements two validated models: LSTM [9] and MTGP [4], both developed by the authors and natively integrated into the library. Future versions will progressively incorporate additional surrogate architectures from peer-reviewed studies in the AD and bioenergy domain. The LSTM (Figure 2) utilises a many-to-one architecture, ingesting a sliding window of historical operational data (e.g., 30 days) to explicitly model the retention time lag, predicting future yield based on past feeding trends [9]. In contrast, the MTGP (Figure 3) utilises a non-recurrent structure designed for robust state estimation. It maps current

process inputs to multiple correlated outputs (Gas, Stability, Effluent) through a set of shared latent Gaussian functions. Crucially, the MTGP output layer includes a covariance term, generating confidence intervals (shaded regions) that quantify the uncertainty of the prediction [4]. To address the computational stiffness of ODE solvers, OpenAD-lib provides native PyTorch modules that share the same API as the mechanistic models. To synergise the interpretability of mechanistic descriptions with the fitting power of deep learning, the library implements a Physics-Informed Machine Learning (PIML) framework. As illustrated in Figure 4, this hybrid approach integrates the "Mechanistic Model" (ADM1 or AM2) directly into the learning loop of the "ML Model." By embedding governing ODEs as physical regularisation terms or using mechanistic simulations to generate synthetic pre-training data, the framework constrains the neural network to biologically plausible solutions [12]. This interaction creates an "Improved Model" that balances the high fidelity of mass-balance equations with the computational speed of data- driven inference. The implementation supports "grey-box" configurations where the ML component compensates for unmodelled phenomena (e.g., temperature shocks) while the physical component guides extrapolation in data-sparse regimes, preventing the prediction of chemically impossible states (e.g., negative concentrations) outside the training domain.

Optimisation and Control

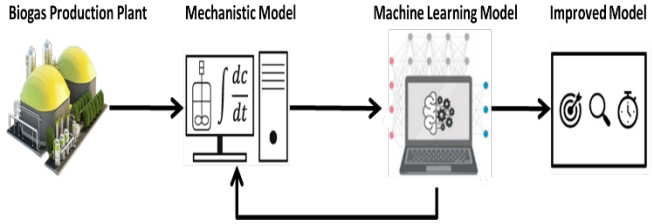

Figure 4. Physics-Informed Machine Learning workflow.

For real-time process optimisation and control, OpenAD-lib integrates a modular control layer designed to be model-agnostic. This architecture allows users to deploy any calibrated model, whether a mechanistic system (AM2, ADM1) or a data-driven surrogate (LSTM, MTGP) as the internal prediction engine (f_{model}) without altering the control logic.

Automated Calibration and Tuning

Prior to deployment, the library leverages the *Optuna* framework [8], a state-of-the-art Bayesian optimisation library to automate parameter estimation. For mechanistic models, this involves identifying kinetic constants (μ_{max}, K_s), while for ML surrogates, it tunes architectural hyperparameters (e.g., learning rate, hidden layers). The library utilises the Tree-structured Parzen Estimator (TPE), a Bayesian optimisation algorithm that constructs probabilistic density estimates of promising versus unpromising parameter regions [8]. TPE is chosen for its ability to converge in significantly fewer iterations than random search [8]. The objective is to minimise the deviation between simulated outputs and experimental time-series data:

$$J_{calib}(\Theta) = \sum_{t=1}^{T} ||y_{meas}(t) - y_{sim}(t, \Theta)||^2 \quad (4)$$

where θ is the vector of unknown kinetic parameters. This automated routine ensures that the internal model accurately reflects the biological reality of the specific digester before control authority is granted.

Model Predictive Control (MPC) Formulation

The control layer wraps the *do-mpc* library [10] to solve nonlinear Optimal Control Problems (OCP). The controller discretises the process dynamics using orthogonal collocation on finite elements, ensuring numerical stability even for stiff biological systems (e.g., rapid pH changes). At each sampling instant k, the solver uses the calibrated prediction engine (f_{model}) to compute a sequence of optimal control inputs u (e.g., dilution rate, feedstock blend) over a finite horizon N_p:

$$\min_{u} J_{MPC} = \sum_{k=0}^{N_p} \left(l(x_k, u_k)\right) + V\left(x_{N_p}\right) \quad (5)$$

Subject to:

- **Model Dynamics:** $x_{k+1} = f_{model}(x_k, u_k)$ (Mechanistic ODEs or ML Inference)
- **Path Constraints**: $g(x_k) \leq 0$ (e.g., $pH > 6, VFA < 2000\ mg/L$)
- **Input Bounds:** $u_{min} \leq u_k \leq u_{max}$

where $l(x_k, u_k)$ is the running cost (penalising deviation from setpoints or maximising yield) and $V\left(x_{N_p}\right)$ represents the terminal cost, which penalises the final state deviation to ensure infinite-horizon stability. The optimisation problem is transcribed into a Non-Linear Programming (NLP) problem via CasADi and solved using IPOPT implemented within the do-mpc library [10]. The implementation natively supports two configurable control strategies: i) Yield Maximisation: the stage cost is defined to maximise energy recovery ($J = -Q_{gas}$), driving the system to the highest loading rate possible within stability constraints. ii) Setpoint Tracking: the controller minimises deviations from a reference trajectory ($J = ||y - y_{ref}||^2$), useful for maintaining effluent quality standards (e.g., COD limits).

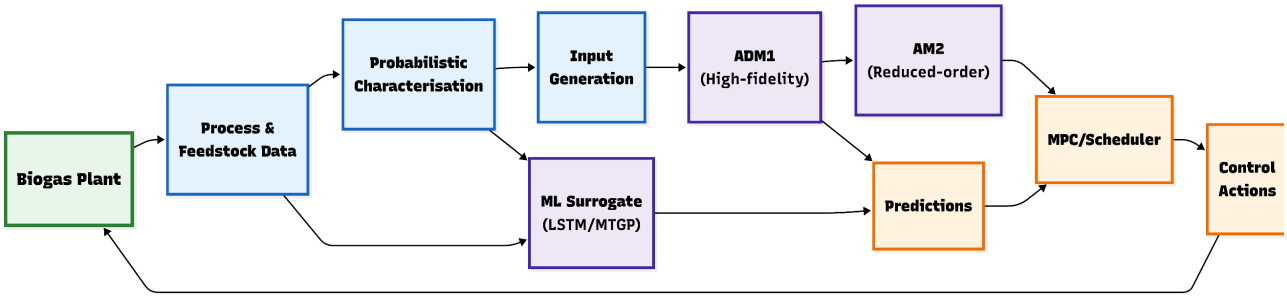

Figure 5: OpenAD-lib framework architecture showing the integrated workflow from feedstock characterization through mechanistic models, ML surrogates to control with closed-loop feedback.

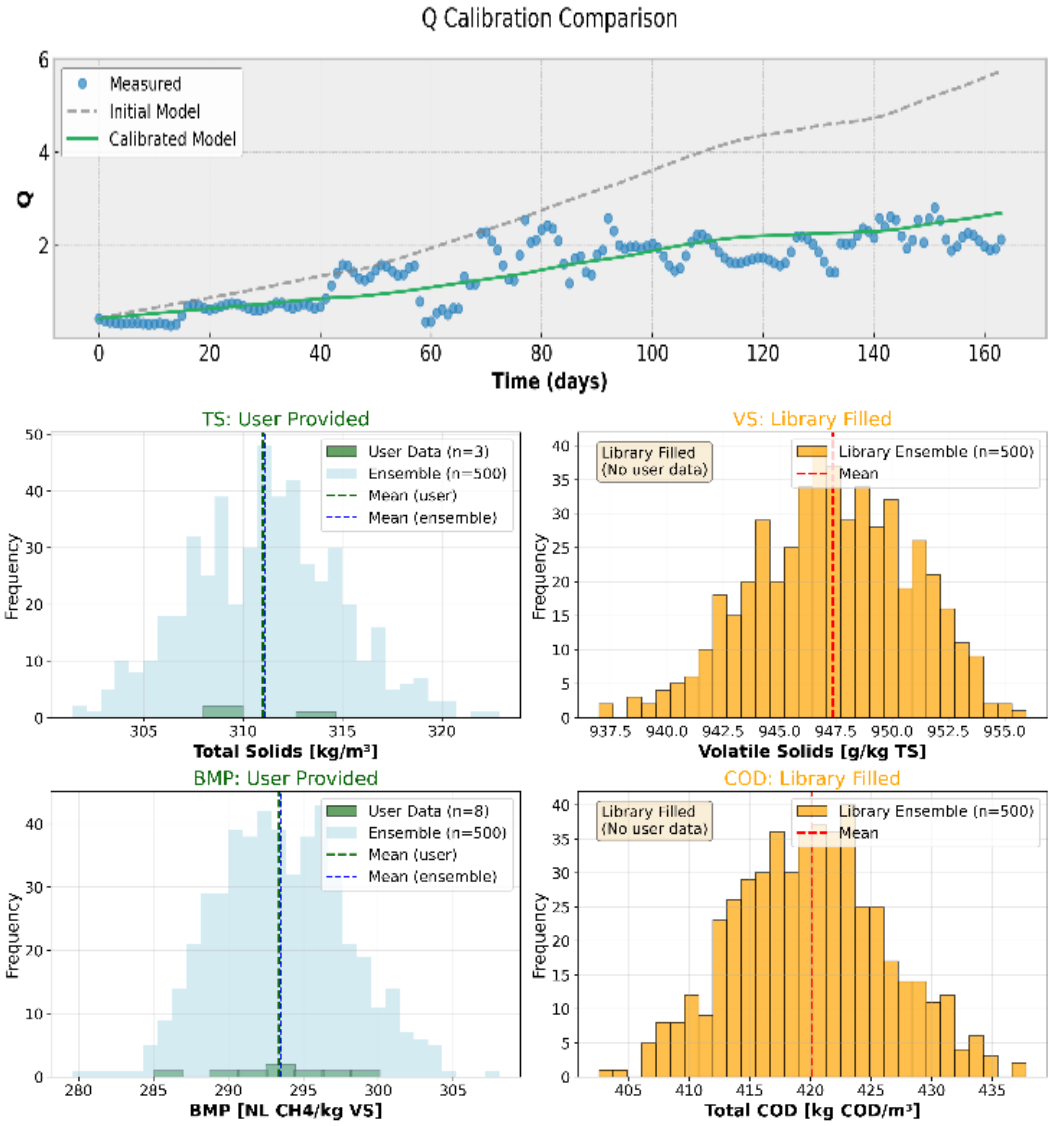

Figure 6. Probabilistic Feedstock Characterisation Sampling.

CASE STUDY

We present a comprehensive case study of a co-digestion biogas plant to showcase the 'plug-and-play' nature of the OpenAD-lib ecosystem. This digital twin integrates the entire process pipeline, transforming sparse feedstock inputs into optimised control trajectories. To

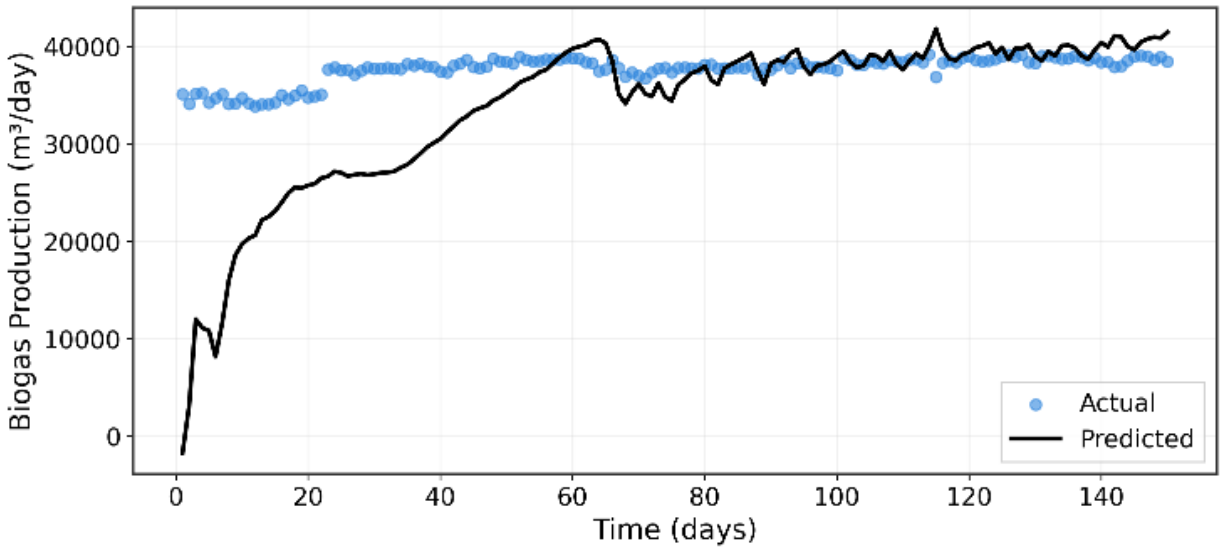

Figure 7. ADM1 mechanistic simulation validation.

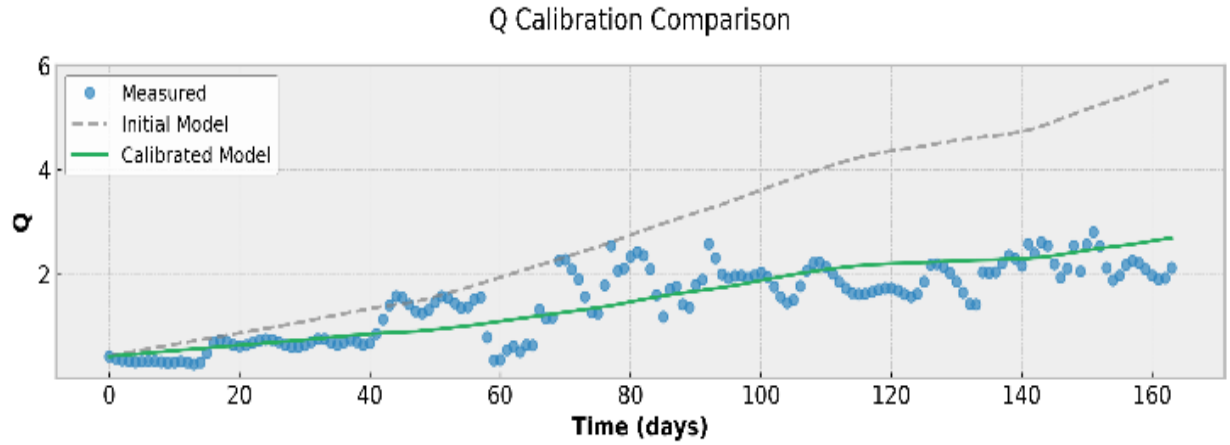

Figure 8. Tune simplified model to plant data with optimised parameters, before/after comparison.

validate the framework across varying degrees of information availability, the case study leverages a multi-source data strategy: it combines real-world sensor readings with synthetic data generated to simulate rare operational events, alongside standard datasets adopted from the original manuscripts of the ADM1/AM2 and MTGP /LSTM models [2, 3, 4, 9]. This hybrid approach ensures the digital twin is tested against both theoretical baselines and realistic operational noise. Figure 5 outlines the complete architecture and data flow of this integrated workflow.

Phase 0: Probabilistic Feedstock Characterisation

This phase Handles sparse user measurements with library uncertainty filling. Figure 6 shows the probabilistic feedstock characterisation with sparse user measurements (3-8 samples) combined with library-filled distributions for TS, VS, BMP, and COD parameters.

```
import openad_lib as openad
# Initialize feedstock library
lib = openad.FeedstockLibrary()
# User provides LIMITED measurements (real-
istic scenario)
user_maize_data = {
 'ts': [310, 315, 308], # Only 3 TS measure-
ments from lab
 'bmp': [285, 300, 293, 290, 295, 292, 298,
294] # 8 BMP measurements
# Missing: vs, cod_total, proteins, lipids
}
# Library fills missing parameters automati-
cally
maize_prob = lib.get_probabilistic("Maize",
user_data=user_maize_data)
```

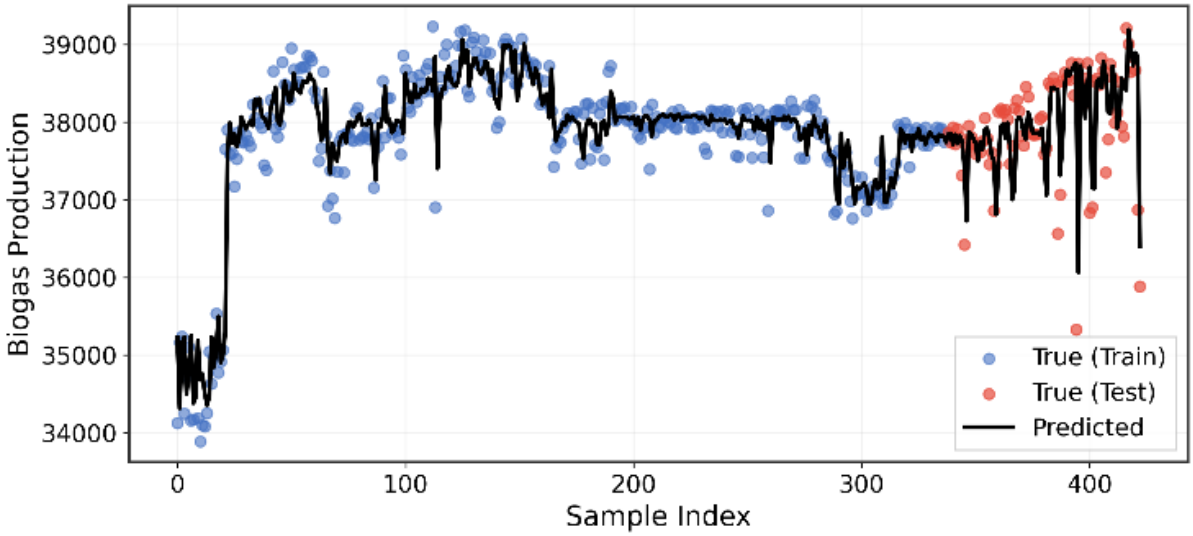

Figure 9. LSTM biogas production predictions.

```
# Generate ensemble for uncertainty propaga-
tion
maize_ensemble = maize_prob.sample(n=500,
random_state=42)
```

Phase 1: ADM1 Mechanistic Baseline

After generating the ADM1 model inputs from the probabilistic characterisation we establish the physics-based understanding by running ADM1 simulations and compare the results of the measured and simulated outputs as shown in Figure 7.

```
# Generate influent from feedstock ratios
influent_df = openad.acod.generate_influ-
ent_data(feedstock_data)
# Run ADM1 simulation (32 states)
adm1_model = openad.ADM1Model()
results = adm1_model.simulate(influent_df)
# Evaluate performance
metrics = openad.utils.metrics.compute_met-
rics(y_true, y_pred)
```

Phase 2: AM2 Parameters Calibration

In this step the AM2 model calibration using Optuna is achieved. Figure 8 show a comparison of pre-calibration predictions of biogas production showing poor fit versus post-calibration performance after 30 optimisation trials.

```
# Calibrate with Optuna (30 trials)
calibrator = openad.AM2Calibrator(am2_model)
best_params = calibrator.calibrate(
 params_to_tune=['m1', 'K1', 'm2', 'Ki',
'K2'], n_trials=30)
```

Figure 8. Tune simplified model to plant data with optimised parameters, before/after comparison.

Phase 3A: LSTM Time Series Prediction

LSTM time-series prediction for biogas production with 30-day lookback window. Figure 9 show the results with training phase (blue, days 0-60) demonstrates model learning on historical feeding patterns. Test phase (orange, days 60-90) validates generalisation capability. LSTM architecture: input dimension = 6 (operational + feedstock features), hidden dimension = 24, training epochs = 30 as shown in the code below.

```
# Create LSTM model
lstm = openad.LSTMModel(input_dim=6, hid-
den_dim=24)
# Prepare time series with lags
X, y, dataset = lstm.prepare_time_se-
ries_data(data, features, target, n_in=1)
# Train and evaluate
lstm.train(X_train, y_train, epochs=30)
metrics = lstm.evaluate(X_test, y_test)
```

Phase 3B: Multi-Task GP with Uncertainty

Next, multi-output prediction with confidence intervals using the MTGP of three correlated outputs with uncertainty quantification is shown. Figure 10 presentes the results of (a) soluble COD (SCOD_out) predictions with 95% confidence intervals (shaded region) achieving 94% empirical coverage; (b) VFA concentration predictions capturing process dynamics; (c) biogas production rate with mean ± 2σ bounds.

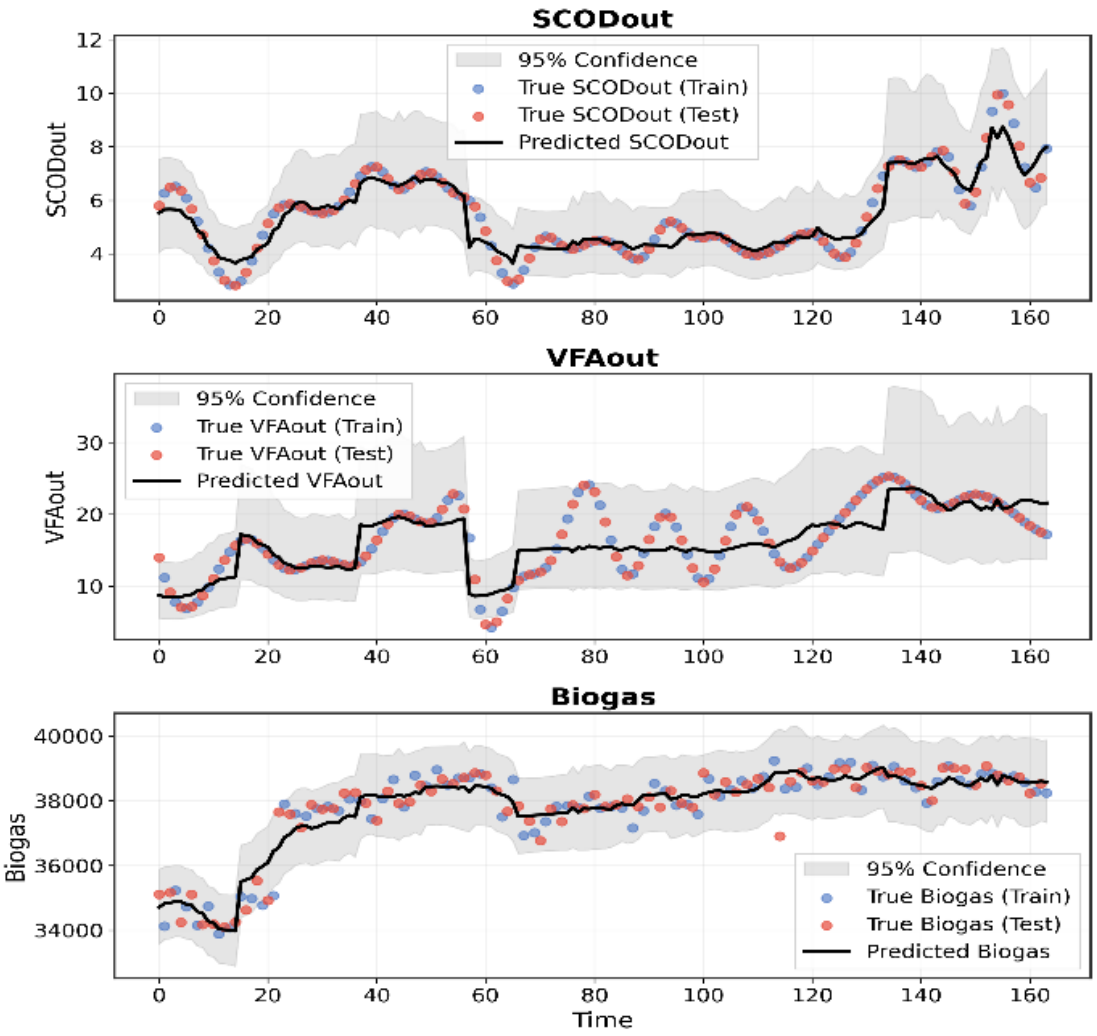

Figure 10. MTGP OF SCOD, VFA, and biogas with 95% confidence intervals.

```
# Initialize MTGP (3 outputs: SCOD, VFA, Bi-
ogas)
mtgp = openad.MultitaskGP(num_tasks=3,
num_latents=3)
# Train and predict with uncertainty
mtgp.train(X_train, Y_train, epochs=300)
mean, lower, upper = mtgp.predict(X_test,
return_std=True)
```

Phase 4A: MPC Biogas Maximisation

For control, MPC for maximum biogas production. Tow-panel trajectory showing (Figure 11): (a) optimal dilution rate (control input) computed at each MPC step subject to process constraints; (b) biogas production increase from 0.001 L/d (baseline) to 0.03 L/d under optimal feeding strategy. MPC configuration: prediction horizon = 10 steps, control update frequency = 1 day, with AM2 internal model.

```
# Setup MPC controller
mpc = openad.AM2MPC(params)
mpc.setup_controller(
 horizon=10,
 objective_type='maximize_biogas',
 D_max=0.5
)
# Run closed-loop simulation
for k in range(30):
 u_opt, y_next =
mpc.run_step(S1in_val=S1in_k, pH_val=pH_k)
```

Phase 4B: MPC VFA Tracking

MPC for VFA concentration setpoint tracking to maintain process stability. Figure 12 demonstrates: (a) VFA concentration regulated at 2.0 mmol/L target setpoint (dashed line) with tight tracking despite

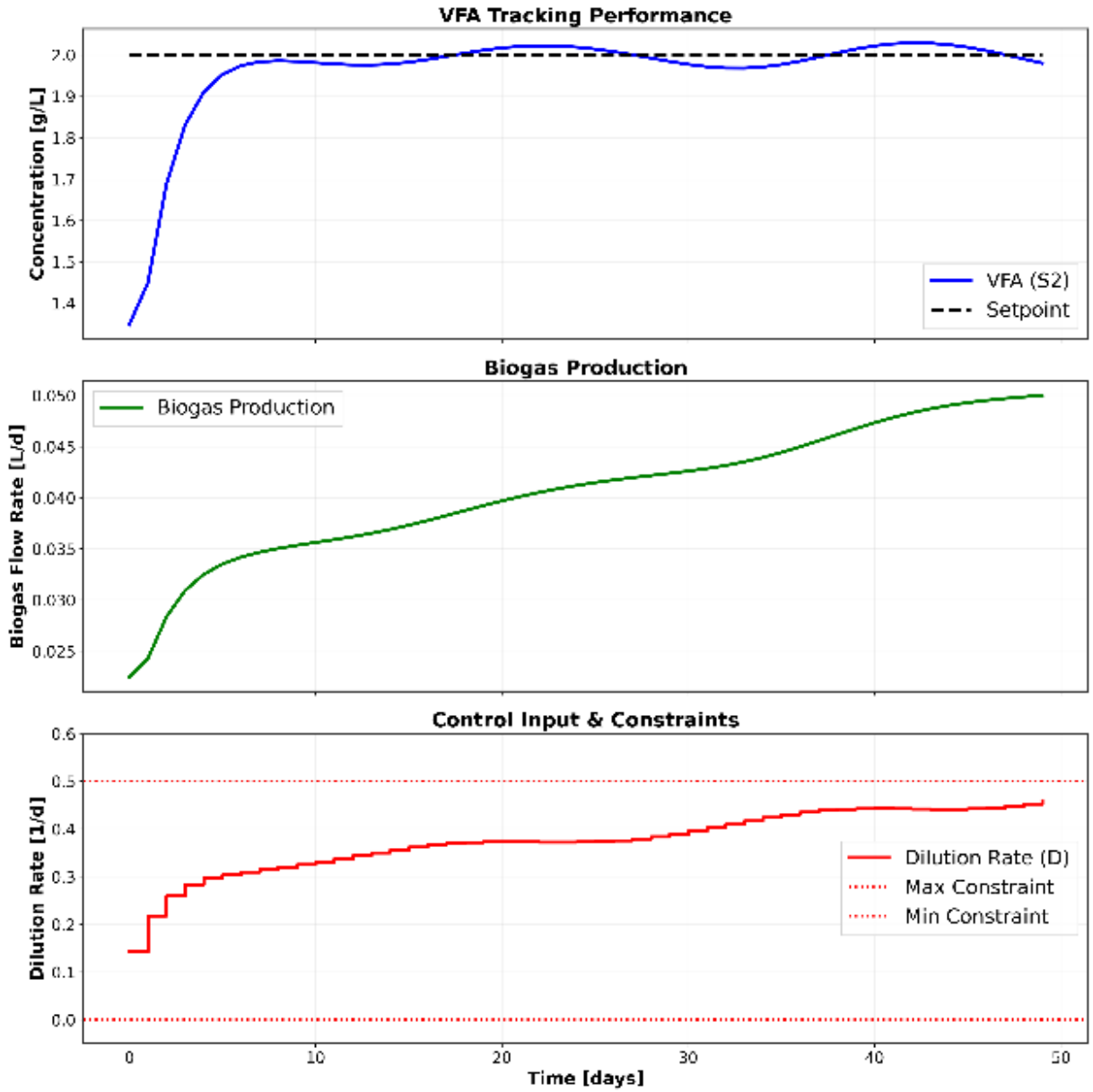

Figure 11. MPC for maximum biogas production.

feedstock composition disturbances; (b) control effort (dilution rate adjustments) showing smooth manipulated variable trajectory without excessive actuator action.

```
# Configure VFA tracking
mpc.setup_controller(
objective_type='track_vfa',
```

Figure 12. MPC for VFA concentration setpoint tracking to maintain process stability.

```
vfa_setpoint=2.0
)
# Run tracking simulation
for k in range(30):
   u_opt, y_next = mpc.run_step(...)
```

CONCLUSION

OpenAD-lib addresses the digital gap in AD industry by unifying mechanistic understanding, data-driven methods, and uncertainty quantification. By providing a production-ready, open-source ecosystem, it enables the development of robust strategies for real-world AD application. Future iterations will extend to bioenergy

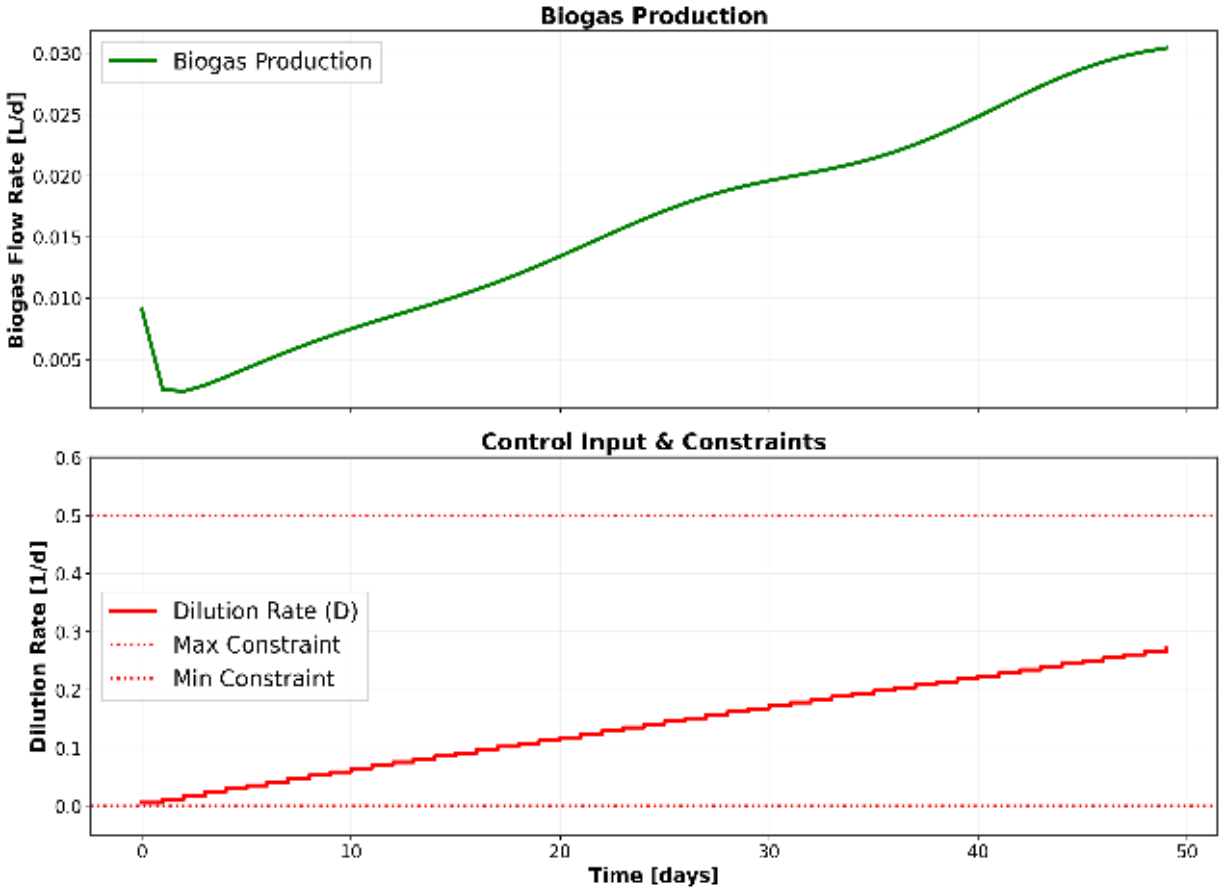

Figure 12. MPC for VFA concentration setpoint tracking to maintain process stability.

technologies (such as gasification and pyrolysis) demonstrating the platform's role in the wider bioenergy transition.

ACKNOWLEDGEMENTS

The authors thank both UKRI grants: the Supergen Bioenergy Impact Hub, EP/Y016300/1 and AI for Net Zero, EP/Y005600/1, for their support.

AUTHOR IDENTIFIERS

Author ORCIDs:
Benaissa Dekhici: 0000-0002-2660-1029
Rohit Murali: 0009-0001-5515-7721
Michael Short: 0000-0003-3227-1844

REFERENCES

1. Dekhici B. Data-driven modeling, order reduction and control of anaerobic digestion processes. Ph.D. thesis, University of Tlemcen (2024)
2. Batstone DJ, Keller J, Angelidaki I, Kalyuzhnyi SV, Pavlostathis SG, Rozzi A, Sanders WTM, Siegrist H, Vavilin VA. The IWA anaerobic digestion model no 1 (ADM1). Water Science and Technology 45:65-73 (2002). https://doi.org/10.2166/wst.2002.0292
3. Bernard O, Hadj?Sadok Z, Dochain D, Genovesi A, Steyer J. Dynamical model development and parameter identification for an anaerobic wastewater treatment process. Biotech & Bioengineering 75:424-438 (2001). https://doi.org/10.1002/bit.10036
4. Dekhici B, Short M. Data-driven modelling of biogas production using multi-task gaussian processes. Systems and Control Transactions 4:26-32 (2025). https://doi.org/10.69997/sct.121877
5. Murali R, Bywater A, Dolat M, Dekhici B, Zarei M, Hilton L, Sadhukhan J, Zhang D, Short M. Anaerobic digestion site-wide optimisation and decision-making: an industrial perspective and review. Renewable and Sustainable Energy Reviews 226:116402 (2026). https://doi.org/10.1016/j.rser.2025.116402

6. Arnell M, Astals S, Åmand L, Batstone DJ, Jensen PD, Jeppsson U. Modelling anaerobic co-digestion in benchmark simulation model no. 2: parameter estimation, substrate characterisation and plant-wide integration. Water Research 98:138-146 (2016). https://doi.org/10.1016/j.watres.2016.03.070
7. Sadrimajd P, Mannion P, Howley E, Lens PNL. PyADM1: a Python implementation of Anaerobic Digestion Model No. 1. bioRxiv (2021) https://doi.org/10.1101/2021.03.03.433746
8. Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A Next-generation Hyperparameter Optimization Framework. arXiv:1907.10902 (2019) https://arxiv.org/abs/1907.10902
9. Murali R, Dekhici B, Chen T, Zhang D, Short M. Mechanistic and data-driven models for predicting biogas production in anaerobic digestion processes. Systems and Control Transactions 4:388-393 (2025). https://doi.org/10.69997/sct.176459
10. Fiedler F, Karg B, Lüken L, Brandner D, Heinlein M, Brabender F, Lucia S. Do-mpc: towards FAIR nonlinear and robust model predictive control. Control Engineering Practice 140:105676 (2023). https://doi.org/10.1016/j.conengprac.2023.105676
11. Supergen Bioenergy Impact Hub.
12. https://www.supergen-bioenergy.net/research/digitalisation-and-ai-for-ad/
13. Shaw KM, Poh PE, Ho YK, Chen ZY, Chew IML. Modeling the anaerobic digestion of palm oil mill effluent via physics-informed deep learning. Chemical Engineering Journal 485:149826 (2024). https://doi.org/10.1016/j.cej.2024.149826
14. Mayowa F. Oladele, George M. Bollas, Leveraging Digital Twin Modeling for Anaerobic Digesters using Anaerobic Digestion Model No. 1 (ADM1) and Neural Network within the Pyomo Framework, Computer Aided Chemical Engineering, Elsevier, Volume 53, 2024. https://doi.org/10.1016/B978-0-443-28824-1.50191-5
15. Ficara E, Hassam S, Allegrini A, Leva A, Malpei F, Ferretti G. Anaerobic digestion models: a comparative study. IFAC Proceedings Volumes 45:1052-1057 (2012). https://doi.org/10.3182/20120215-3-at-3016.00186