A Software Tool for Identifying Kinetic Models from Concentration Data

Vignesh Kumar S[1] Nirav P Bhatt [1,2] Sridharakumar Narasimhan [1,2]

[1]Indian Institute of Technology Madras, Chennai, India [2]Robert Bosch Centre for Data Science and Artificial Intelligence, IIT Madras, India

Introduction

This project introduces PyKineMod, a toolbox for determining kinetic model structures and corresponding parameters from concentration data for homogeneous reaction systems with inlet and outlet streams. The software implements the incremental identification approach and overcomes the limitation of combinatorial complexity while determining the model structure discrimination for multiple reactions. The software tool uses incremental approaches to solve optimization problems and supports rate-based and extent-based parameter estimation techniques. It includes data preprocessing capabilities and can determine parameter confidence levels. Implemented in Python, it utilizes numpy, scipy, and pandas packages for a simplified and Pythonic approach to kinetic parameter determination. The optimization problem is solved using the scipy optimization solver, and confidence intervals are obtained through bootstrapping.

Data

Input data required for the software (with shape):

- **N**: Stoichiometry Matrix $(R * S)$
- **Mw**: Molecular weight Matrix $(S * S)$
- **V**: Volume of the reactor ($const$, or $(T,)$ shape)
- **Winhat**: Inlet composition matrix $(P * S)$
- **uin**: Mass Flow rate of P inlets $(P * 1)$ or $(P * T)$
- **uout**: Mass Flow rate of outlet $const$ or $(T,)$
- **n0**: Initial number of moles in the reactor $(S * 1)$

where S = number of species, R number of reactions, T timesteps, and P inlets

Methodology

The software primarily provides two ways of solving the kinetics-identification problem: rate-based and extent-based incremental identification methods.

- **Rate-Based method:**

$$\hat{\theta}_i = \min \Sigma_{h=0}^{H} (r_i(t_h) - r_i(\mathbf{c}(t_h), \theta_i))^2$$

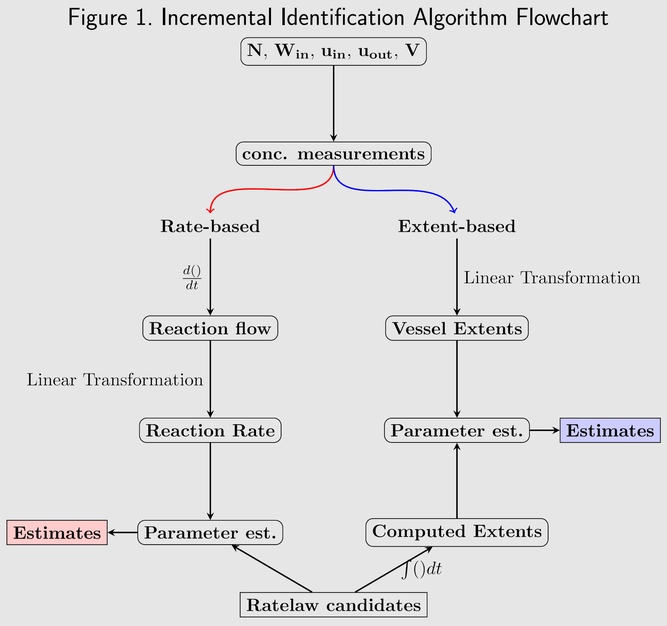
$$\begin{aligned} &\text{s.t. } \mathbf{r}(t_h) = (N^T)^{+}\mathbf{f}(t_h)V^{-1}(t_h) \\ &\mathbf{f}(t_h) = \frac{d\mathbf{n}}{dt}(t_h) - W_{in}u_{in}(t_h) + \frac{u_{out}(t_h)}{m(t_h)}\mathbf{n}(t_h) \\ &m(t_h) = \mathbf{1}_S^T \mathbf{M}_W \mathbf{n}(t_h) \\ &\text{(or)} \quad \dot{m} = \mathbf{1}_p^T \mathbf{u}_{in} - u_{out}, \quad m(0) = m_0 \end{aligned} \tag{1}$$

- **Extent-based method:**

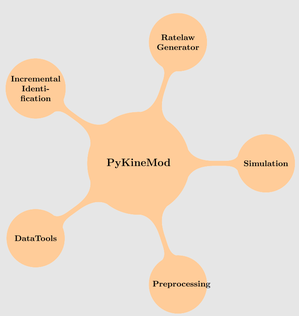
$$\hat{\theta}_i = \min \Sigma_{h=0}^{H} (x_{r,i}(t_h) - x_{r,i}(\theta_i, t_h))^2$$

$$\begin{aligned} &\text{s.t. } \mathbf{x}_r(t_h) = (S_0^T)\mathbf{n}(t_h); \quad S_0 \text{ through P3D alg.} \\ &\dot{x}_{r,i}(\theta_i, t) = r_i(\mathbf{c}(t), \theta_i)V(t) - \frac{u_{out}(t)}{m(t)} x_{r,i}(\theta_i, t) \\ &m(t_h) = \mathbf{1}_S^T \mathbf{M}_W \mathbf{n}(t_h) \\ &\text{(or)} \quad \dot{m} = \mathbf{1}_p^T \mathbf{u}_{in} - u_{out}, \quad m(0) = m_0 \end{aligned} \tag{2}$$

Figure 1. Incremental Identification Algorithm Flowchart



Software Package



Application of package is demonstrated with acetoacetylation of pyrrole reaction. To simulate realistic data, measurements are corrupted with additive zero-mean Gaussian noise. Code for creating object and adding concentration data is given below.

In the code, the model object takes stoichiometric matrix **N**, Molecular weight matrix $\mathbf{M}_w$, Volume of the reactor V, inlet composition matrix $\mathbf{W}_{inhat}$, inlet and outlet mass flow rate u_{in} and u_{out} and initial concentration n_0 as inputs.

```python
# Importing PyKineMod package
from PyKineMod import Incremental
# Creating Model Object
model = Incremental(N, Mw, V, Winhat, uin, uout, n0)
# Adding concentration data from csv file
model.add_concentration_data("aceto_pyrrole_basecase.csv")
```

Software Package

```python
# Generating Ratelaws from Stoichiometric Matrix
from ratelawgen import CandidateRateLaws
import numpy as np

# N is RxS Matrix
cand1 = CandidateRateLaws(N[0], type = "Irreversible") # For First Reaction
cand2 = CandidateRateLaws(N[1], type = "Reversible") # For second Reaction
cand3 = CandidateRateLaws(N[2], type = "Irreversible") # For Third Reaction

candidates_list = [cand1, cand2, cand3]
```

Creating candidate ratelaws is as simple as this. We can also use custom functions like below.

```python
# Pythonic way of creating custom rate laws
def cand_ratelaw11(y,K):
    Ca, Cb, _, _, _, _ = y
    Ck = 0.5
    return K[0]*Cb*Ck
candidates_list = [[cand_ratelaw11]]
```

Discriminating and estimating the parameters are done as below. Plot argument will plot loss versus ratelaws. Loss metrics can be RMSE, AIC or AICC as of now.

```python
# Rate-based Method
res1 = model.estimate_parameters(candidates_list, method = 'rate_based',
  conf_int = True, metric = 'aicc', plot = True, bootstraps = 1000)
# Extent-based Method
res2 = model.estimate_parameters(candidates_list, method = 'extent_based',
  conf_int = True, metric = 'aicc', plot = True, bootstraps = 100)
```

Results

Scenario 1: 1% noise, 151 points.
Scenario 2: 10% noise, 151 points.
Scenario 3: 1% noise, 21 points.

Correct rate laws are used in the parameter estimates. The parameters values used to generate data are $k_1 = 0.053, k_2 = 0.128, k_3 = 0.028$. Confidence intervals obtained are not shown.

Scenario	rate par.	rb-est.	eb-est.
1 Base case	k1	0.0538	0.0511
	k2	0.1279	0.1276
	k3	0.0279	0.0275
2 High noise	k1	0.0584	0.0425
	k2	0.1247	0.1245
	k3	0.0299	0.0291
3 Fewer points	k1	0.0583	0.0333
	k2	0.1371	0.1385
	k3	0.0311	0.0285

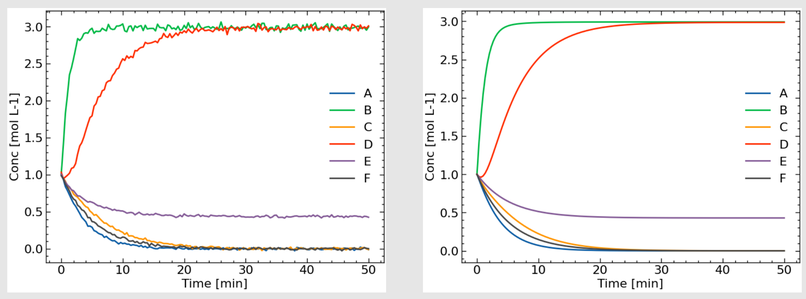


Figure 2. The simulated data on the left and the final fitted concentration profiles

Conclusion

Our software package simplifies the identification of optimal rate laws and parameters in open reactor systems, even with noisy data. By utilizing Python libraries like numpy, scipy, and pandas, it directly estimates kinetic parameters from concentration data. The software generates candidate rate laws, compares them to experimental data, and selects the best fit, while also calculating confidence intervals for the parameters. With its user-friendly interface and powerful libraries, our package streamlines data analysis, enabling informed decision-making in research.

References

- Amrhein, M., Bhatt, N., Srinivasan, B., Bonvin, D. (2010). Extents of reaction and flow for homogeneous reaction systems with inlet and outlet streams. AIChE Journal, 56(11), 2873-2886.https://doi.org/10.1002/aic.12125
- Michael Marquardt, Wolfgang Bonvin, Dominique. (2012). Incremental Identification of Reaction Systems - A Comparison between Rate-based and Extent-based Approaches. Chemical Engineering Science. 83. 24–38. 10.1016/j.ces.2012.05.040.
- Bhatt, Nirav Amrhein, Michael Bonvin, Dominique. (2011). Incremental Identification of Reaction and Mass–Transfer Kinetics Using the Concept of Extents. Industrial Engineering Chemistry